

Preface

With the development of computer and its popularization, scientific computation has evolved into one of the three magic weapons of scientific research with its own right. In addition, industrials in various domains have considered scientific computation as a key technology. In the light of this ongoing development, it is expected that scientific computation software will play an important role for enlarging wide applications. Scilab, as a free open-source software package for scientific computation, provides a powerful and interactive environment with extensive mathematical capabilities, sophisticated graphics, and high-level programming language for rapid prototyping. Due to its distinguished features of “open source”, Scilab has been widely used by more and more engineers and scientists in both academia and industries.

As a matter of fact, since it was developed in 1990s by INRIA (Institut National de Recherche en Informatique et en Automatique) and ENPC (Ecole Nationale des Ponts et Chaussees), France, Scilab has been distributed freely along with the source code via the Internet and used in educational and industrial environments around the world. Meanwhile, Scilab is also being disseminated in all over the world by various introductory documentations in several languages such as French, English, Spanish, Portuguese, Chinese and so on.

Since 2002, an annual event of software development contest based on Scilab platform and environment has been carrying on in China, which is no doubt to promote and advance a worthy cause for the development of open source software engineering, especially for the research, development and applications of Scilab. In order to promote an in-depth research and development in this field and enrich scientific achievements, LIAMA and ZheJiang University, China jointly organize 2006 International Workshop on Open Source Software Scilab and its Engineering Application (OSSS-EA'06) on September 28-29, in Hangzhou, China under the sponsorship of Zhejiang University,

INRIA/Scilab Consortium-France, LIAMA in China and so on. During the workshop the Award Ceremony of 2006 SCILAB Contest is celebrated so that some young and promising prize winners can take a good opportunity to exchange knowledge and experience with participants in the workshop. It is indispensable and very important to boost up the strength of R&D and further popularize the applications of scientific achievements in this field.

In this book, 30 papers are included, which are selected from papers submitted to 2006 International Workshop on Open Source Software Scilab and its Engineering Application (OSSS-EA'06). According to their corresponding topics the papers are classified as 3 different parts which may be introduced as follows.

Part 1 is about some researches on Scilab and Open Source Software Engineering, which contains 7 papers and the major contents the Integrating modelica in scicos; Evaluating Interpolation Kernels for InSAR Using Scilab; Scilab Grid Toolkit with ProActive; Some practical experiences with Scilab Scicos and RTAI-Lab at the SUPSI laboratory; Targeting the Scicos Code Generator HDL Model Example; The Present and Future of Scilab's Engineering Application; and Tools for building Scilab toolboxes.

Part 2 focuses on the design and development of toolbox for Scilab which contains 9 papers and the major contents A Complex Network Toolbox; A Testing Tool for Character Recognition; An ITS Dynamics Model Simulator; Audio Editor Toolbox, Campus Emergency Management System; Scilab scicos toolboxes; SciSpeech Toolbox; Telecommunication Toolbox; The realization of SCADA based on Scilab .

Part 3 consists of engineering applications based on Scilab which contains 14 papers and the major contents A Stereo Matching Method for Color Image; Dynamic Analysis and Simulation of Water Management; Electrostatic precipitator modeling and simulation;

Function versus geometry; Secondary Clarifier and A2O Process Modeling; Modeling of the Activated Sludge Process; Modeling and Simulation of Adaptive Optics; Modeling and Simulation for Transpiration Cooling Control System with Distributed Parameters Based on Scilab; Identification and simulation of mechanical structures; Simulation of cars waiting at a toll booth using Scilab; Sustainable Cooktop Design; The Counting Algorithm of Transistor Based On Image Detect Technology; Two efficient numerical methods for finding global minimizers of continuous functions of one variable; Using Scilab to Perform a Cogeneration System Simulation and Optimization.

We are grateful to all contributors for making the workshop a success. We would like to express our sincerely thanks to France Telecom, R & D Beijing; Thomson Broadband R & D Beijing; Bull Information Systems Beijing; Hangzhou Zheda Feiteng Technology Co., Ltd; Jinhua Hansheng Mechanical & Electric Control Engineering Co., Ltd; Zhejiang JiaHuan Electronic Co., Ltd. for their support. We also thank Mr. Zhe Peng for his elaborately composing and design in the publication of this book; and Ms. Li-fang Zhou, Mr. Sheng-ming Chen and Mr. Ke-jie Fang for their beneficial helps in organizing the workshop.

Long-Hua Ma, Shi Li

Claude Gomez

Hangzhou, Aug.29, 2006

CONTENTS

Part1. Researches on Open Source Software Scilab

Challenges in Integrating Modelica in the Hybrid System Formalism Scicos	3
<i>Ramine Nikoukhah</i>	
Evaluating Interpolation Kernels for InSAR Using Scilab	9
<i>Jinghui Fan, Guoqiang Li, Xiaofang Guo, Guang Liu , Huadong Guo, Daqing Ge, and Shengwei Liu</i>	
Scilab Grid Toolkit with Open Source ProActive	21
<i>Denis Caromel, and Arnaud Mangin</i>	
Some Practical Experiences with Scilab/Scicos and RTAI-Lab at the SUPSI Laboratory	31
<i>Roberto Bucher, and Silvano Balemi</i>	
Targeting the Scicos Code Generator HDL Model Example	49
<i>Zhang Dong, and Kang Cai</i>	
The Present and Future of Scilab's Engineering Application	61
<i>Zhe Peng, and Longhua Ma</i>	
Tools for Building Scilab Toolboxes, Present and Future	69
<i>Farid Belhacene, and Serge Steer</i>	

Part2. Design and Development of Toolbox for Scilab

A Complex Network Toolbox based on Scilab	79
<i>Bo Wu, Wei Chen, and Gang Yan</i>	
A Testing Tool for Character Recognition Based On BP Network Using JAVASCI Interface	89
<i>Ding Liu, Rui Xiao, and Xin Zhou</i>	
An ITS Dynamics Model Simulator Based on Scilab	97
<i>Yanlin Wang, Shuqin Lee, and Panfeng Xue</i>	
Audio Editor Toolbox for Scilab --- Scilab Melody	107
<i>Yuanhang Pan, Yongpeng Lin, Min Dechun , Xianming Xu, Wenfeng Zeng, Yan Wang, Yanjie Li, Zhihui Wang, Yang Liu, Jun Cai, and Shengrui Li</i>	

Campus Emergency Management System based on Scilab	119
<i>Xin Wang, and KeJun Wang</i>	
Scilab/Scicos Toolboxes for Telecommunications	129
<i>Sebastien Mons, Alan Layec, Abderrazak.Benadji, Tibault Reveyrand, Raphael Sommet, Edouard Ngoya, and Raymond Quere</i>	
Speech Analysis, Synthesis and Recognition Toolbox for Scilab---SciSpeech	143
<i>Cong Wu, XiDa Chen, HuiTing Su, WeiJie Liu, LongLing Luo, Kun Yang, Jing Liu, TieZheng Lin, RongPing Zhuang, and Jun Cai</i>	
Telecommunication Toolbox Design Based on Scilab	157
<i>Yongshun Chen, and Yanhua Zhang</i>	
The Realization of SCADA based on Scilab	175
<i>Zhe Peng, and Longhua Ma</i>	

Part3. Engineering Applications Based on Scilab

A Stereo Matching Implementation Method for Uncalibrated Color Image Based on Scilab	189
<i>Zhulin LI, Zongtao ZHAO, Genyao ZHANG, and Yan MA</i>	
Dynamic Analysis and Simulation of Water Management in PEM Fuel Cell based on Scilab	197
<i>Liyang Zhang, Mu Pan, and Shuhai Quan</i>	
Electrostatic Precipitator Modeling and Simulation	211
<i>Kejie Fang, Longhua Ma, Weiguang Chen, Qinglong Jiang, and Zhiping Zhang</i>	
Function Versus Geometry – A dichotomy for Design Engineers	221
<i>Mr Gareth Black, Dr David Allanson, and Dr Steve Douglas</i>	
Improved Model of Secondary Clarifier and A2/O Process Modeling	233
<i>Shengming Chen, and Longhua Ma</i>	
Modeling and Simulation for Transpiration Cooling Control System with Distributed Parameters Based on Scilab	247
<i>CongChao Zhu, ShiYin Qin</i>	
Modeling of the Activated Sludge Process by Multi-model Approach	269
<i>Linling zhao, Lifang zhou, and Luwen zhou</i>	
SAO: Modeling and Simulation of Adaptive Optics in the Scilab/ Scicos Environment	279
<i>Jingyuan Chen, Guangyong Gan, and Yingxue Tao</i>	

Scilab Tools for Identification and Simulation of Mechanical Structures	297
<i>Maurice Goursat, and Laurent Mevel</i>	
Simulation of Cars Waiting at a Toll Booth Using Scilab	309
<i>Jianfeng Li, and Lifang Zhou</i>	
Sustainable Cooktop Design Using Scilab	317
<i>T.T. Wong, C.W. Leung, D.Y. Liu, and H.B. Li</i>	
The Counting Algorithm of Transistor Based On Image Detect Technology	327
<i>Peng Zhou, Xiangbo Yang, Hong Zhu, and Ruirui Ji</i>	
Two Efficient Numerical Methods for Finding Global Minimizers of Continuous Functions of One Variable	335
<i>Yu He</i>	
Using Scilab to Perform a Cogeneration System Simulation and Optimization	345
<i>Fonseca Jr., J.G.S.1, Asano H.1, Fujii T.1, and Hirasawa S.</i>	

Part 1

Researches on Open Source Software Scilab

Challenges in Integrating Modelica in the Hybrid System Formalism Scicos

Ramine Nikoukhah

INRIA-Rocquencourt, Domaine de Voluceau, 78153 Le Chesnay Cedex, France
ramine.nikoukhah@inria.fr,
WWW home page: <http://www.scicos.org>

Abstract. Both Modelica and Scicos propose formalisms for modeling hybrid dynamical systems. Despite many differences, their integration within a single tool is considered. It is shown in particular that even though the two methods of constructing models are very different, their implementations of hybrid dynamics are so similar that an integration at a very low level can be considered.

1 Introduction

Many different formalisms have been used for modeling systems with both continuous-time and discrete-time dynamics. Such systems, referred to as hybrid dynamical systems, are encountered in many systems and control applications. Scicos and Modelica propose two such formalisms that allow easy model construction, and can be used efficiently for numerical simulation and real-time code generation. They consider very general dynamic behaviors and are suited for general purpose modeling and simulation tools.

Modelica is a language based formalism in which the goal is to be able to express all the dynamics of the system without the use of any other programming language. Scicos on the other hand is based on a hierarchical graphical editor in which the model is constructed by connecting elementary blocks, the dynamics of which can be expressed in standard programming languages such as C. Despite these fundamental differences, the treatment of the hybrid dynamics in both formalisms are so similar that it is only natural to consider the integration of the two within a single tool. It is of course of no surprise that the two formalisms are similar, in particular as far as the discrete dynamics is considered, since both are largely inspired by synchronous languages such as the Signal language [1].

Even though it is possible to consider extensions to the Modelica language to allow the use of Scicos models, in this paper we focus on the integration of Modelica within the Scicos formalism. This integration is already started; in the current version of Scicos, one can define sub-diagrams using Modelica components (for example an electrical circuit). This however is only a first step since only continuous-time dynamics are allowed to be defined in Modelica. The use of discrete dynamics within the Modelica subpart is not allowed since it poses synchronization problems with the rest of the diagram defined in Scicos formalism.

In this paper, we examine the way hybrid dynamics modeling is implemented in Scicos and Modelica, and examine the similarities and the differences. We then propose solutions for a harmonious integration of Modelica in the Scicos tool. We assume the reader is familiar with Modelica and Scicos; information about Modelica can be obtained from www.modelica.org. For information about Scicos, visit www.scicos.org; see also [2].

2 Why use Modelica in Scicos?

Standard Scicos is not well suited for physical component level modeling. For example, when modeling an electrical circuit, it is not possible to construct a Scicos diagram with a one to one correspondence between the electrical components (resistor, diode, capacitor,...) and the blocks in the Scicos diagram. In fact, the Scicos diagram does not resemble the original electrical circuit .

Consider the electrical circuit depicted in Figure 1. This circuit contains a voltage source, a resistor and a capacitor. To model and simulate this diagram in Scicos, we have to express the dynamics explicitly.

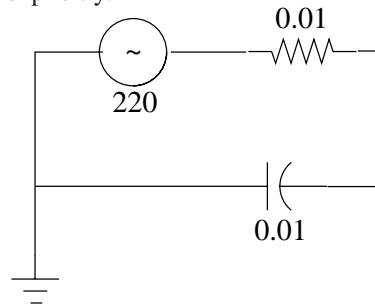


Fig. 1. A simple electrical circuit.

The resulting Scicos diagram is depicted in Figure 2. Note that this Scicos diagram does not look anything like the original electrical circuit in Figure 1.

Recently an extension of Scicos has been developed to allow modeling of physical components directly within the standard Scicos diagrams. This has been done, in particular, by lifting the causality constraint on Scicos' blocks and by introducing the possibility of describing block behaviors in the Modelica language. This extension allows us to model naturally not only systems containing electrical components but also mechanical, hydraulic, thermal, and other systems. For example, the electrical circuit in Figure 1 can be modeled and simulated by constructing the Scicos diagram in Figure 3. The electrical components come from the `Electrical` palette.

The electrical components in this case are realized as *implicit blocks*. Implicit blocks have implicit ports. An implicit port is different from an input or an output port in that connecting two such ports imposes a constraint on the values at these ports but does not imply the transfer of information in an a-priori known direction as it is the case when

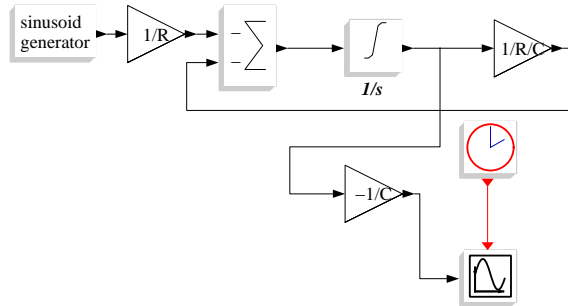


Fig. 2. Scicos diagram realizing the dynamics of the electrical circuit in Figure 1.

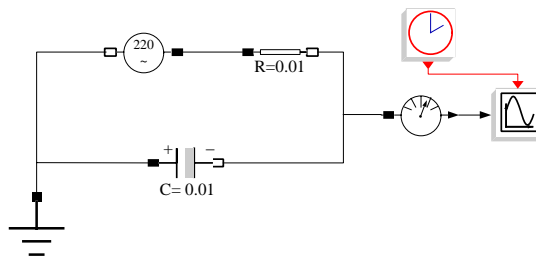


Fig. 3. Scicos diagram using Modelica component.

we connect an output port to an input port. For example, the implicit block `Capacitor` used in diagram of Figure 3 has current and voltage values on its ports but there is no way a-priori, without analyzing the full diagram, to designate any of them as input or output.

Implicit blocks and the construction of diagrams based on these blocks is fully in the spirit of the object oriented language Modelica. The description of block behavior is done in Modelica. Only a subset of the Modelica language can be used however since all the implicit blocks are grouped in a first phase of diagram compilation to obtain a single standard Scicos block with continuous-time dynamics. As long as the Modelica based components of the diagram contain only continuous-time dynamics, the conversion into a single Scicos block is straightforward, but Modelica is a rich language allowing the modeling of discrete and hybrid dynamics. These latter capabilities, if used within a Scicos diagram, poses many challenges. The main problem is how to assure synchronism among discrete-time dynamics of the Modelica and Scicos parts of the diagram.

3 Full integration of Modelica

We study here the possibility of a full integration of Modelica within the Scicos environment. The first step is to identify the similarities between the two and in particular to establish a sort of a dictionary for translating models from one formalism to the other.

3.1 Hybrid dynamics: Modelica and Scicos

We have presented an overview of some aspects of hybrid dynamics modeling in Scicos earlier in this paper; we now present the way Modelica implement similar dynamics.

Continuous-time dynamics As far as the continuous-time dynamics is concerned, both Scicos and Modelica work with systems of DAEs (Differential-Algebraic Equations). Modelica has the advantage of having the possibility to manipulate the equations symbolically and thus for example reducing the index if necessary, and optimizing the code. But at the end, the type of continuous-time dynamics obtained by Modelica is fully integrable in Scicos; in fact as we have previously seen, this integration is already realized in Scicos.

The zero crossing detection capabilities of Scicos are realized using the `when` statement in Modelica. For example

```
when x>0 then
```

where x is a continuous-time variable, corresponds to a zero crossing test in Scicos. This feature is also already integrated in the current Modelica integration in Scicos.

Events in Modelica are in general similar to events in Scicos. An event can cause a jump in a continuous-time variable. In Modelica this is done via the `reinit` operator. In Scicos, an always-active block having a continuous-time state can also have an input activation port. An event received on this input activation port could cause a jump in the continuous-time state. One such a block is the integrator block with re-initialization available in the `Linear` palette. The output of this block is the integral of its first input but when it is activated by an event through its activation input port, the value on the second input is copied into the state of the integrator.

In Modelica, as it is the case in Scicos, events are used to help the numerical solver integrate the system more efficiently. This is done in particular by generating events at points in time where the continuous-time dynamics become non-smooth. For example the `abs` function (absolute value), is not differentiable at zero. So when the input value of this function crosses zero, a cold restart signal should be sent to the numerical solver. Both in Modelica and Scicos, these events are generated by default but can be turned off. In Modelica, this is done using the `noEvent` directive, and to some extent using the `smooth` directive. In Scicos, each block has direct control over the number of its zero-crossings.

Discrete dynamics In Modelica, the `when` statement is also used to implement the discrete dynamics. In particular a statement such as `when x>0 then` seen earlier applied to a continuous-time x can also be used with a discrete-time x . The event triggered

in this case is synchronous with the discrete-time variable x . This usage of `when` can also be realized using the `if` and the `edge` statements. Conditional sub-sampling in particular is realized in this way.

In Scicos, discrete dynamics is realized using events and conditional sub-sampling is realized using the special blocks `IfThenElse` and `ESelect`. The main difference with Modelica is that in Modelica the primitive operator `sample` is used to implement an event clock. In Scicos, event clock is realized using a delay and a feedback. This type of realization is also possible in Modelica and the necessity of the primitive `sample` is not clear; it can very well be implemented as a model.

Another important language feature in Modelica is the possibility of performing an action when one, or more, of certain events is activated. For example the statement

```
when {x > 2, sample(0,2), x < 5} then
```

becomes true if any of the three conditions within the curly brackets becomes true. In Scicos, this union of events is realized using the event `sum` “block”. See Fig. 4. Note that both in Scicos and Modelica, if two of these events become true synchronously, the corresponding operations are executed only once.

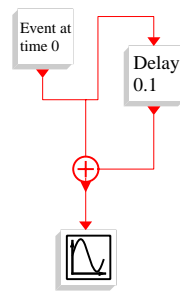


Fig. 4. Union of events realized with the sum operator.

An important conditional constructor in Modelica is `elsewhen` (or equivalently `elseif edge`). This constructor is used as follows:

```
when cond1 then
  ...
elsewhen cond2 then
  ...
end when;
```

In case `cond1` and `cond2` become synchronously true, this construction specifies unambiguously the operations that must be executed. In Scicos, the same construction can be realized in a more general form. In particular, a block can have more than one input

activation port; this is the case for the `Select` block, which has two regular and two activation input ports. This block depending on the activation input by which it has been activated, copies one or the other of its inputs on its output (it selects one of the inputs). To be able to realize such operations, the Scicos block, when activated, receives the an integer specifying exactly the way by which it had been activated. In this case, there are three possibilities: activated through the first activation input port, through the second or both. The block can then execute different operations in each case. Note that this implementation is more flexible than the `elsewhen` construct in Modelica.

3.2 Scicos blocks expressed in Modelica

Considering the similarities between the two formalisms, it is natural to consider the construction of most, if not all, Scicos blocks in Modelica language. This task is a lot more ambitious than the integration already available in Scicos where only blocks with continuous-time dynamics can be modeled in Modelica.

To achieve the full integration, certain adaptations must be considered both in Scicos and in Modelica. We have already encountered the problem with the generation of events using the `sample` primitive and the dual usage of `when`. These however do not pose any real problem because they can always be implemented, in Modelica, using other primitives, for which we have counterparts in Scicos.

Another difference encountered previously is the counterpart of the `elsewhen` statement, which is more general in Scicos. But again this doesn't pose any problem because the Scicos implementation is strictly more general; it would have been a problem if a Scicos block were to be integrated in Modelica as an external function.

4 Conclusion

We have discussed a number key issues in integrating Modelica in the Scicos environment. We have shown that thanks to the similarity between the two formalisms, this task is realizable as long as some Modelica primitives are implemented as models.

The full integration of Modelica considered in this paper would be very valuable to Scicos since having access to formal description of blocks is essential for model reduction and optimal code generation. On the other hand, the Scicos environment provides a natural setting for Modelica language by providing a free Simulink like model editor where the user has the option of constructing new blocks either using the Modelica language or standard programming languages such as C based on Scicos formalism. This latter option is more efficient in some cases, but also can be easier for developers who are not familiar with the complex object oriented language Modelica.

References

1. A. Benveniste, and P. L. Guernic, "Hybrid dynamical systems theory and the signal language", *IEEE Transactions on Automatic Control*, AC-35, 1990.
2. S. L. Campbell, J. P. Chancelier and R. Nikoukhah, *Modeling and Simulation in Scilab/Scicos*, Springer-Verlag, 2005.

Evaluating Interpolation Kernels for InSAR Using Scilab

Jinghui Fan^{1 2 3}, Guoqiang Li⁴, Xiaofang Guo³, Guang Liu^{1 2}, Huadong Guo²,
Daqing Ge³, and Shengwei Liu³

¹ Graduate School of Chinese Academy of Science, Beijing 100049, China;

² State Key Laboratory of Remote Sensing Science, Institute of Remote Sensing Applications, CAS and Beijing Normal University, Beijing 100101, China;

³ AGRS (China Aero Geophysical Survey & Remote Sensing Centre for Land & Resources), Beijing, 100083, China;

⁴ School of Geophysics and Geoinformation Technology, China University of Geosciences (Beijing), Beijing, 100083, China

jhfan2004@yahoo.com

Abstract. Interferometric Synthetic aperture radar (InSAR) has become one of the most powerful tools in environment microwave remote sensing for its prominent applications. Interpolation is necessary for registration, oversample and calibration in conventional and latter-day advanced InSAR technology. The interferometric decorrelation and phase noise can objectively quantify the interpolation accuracy of various kernels. Based on the free software Scilab, numerical results are compared for nearest neighbor, piecewise linear, four points cubic convolution, six points cubic convolution, Knab SW and Raised cosine kernels. In the procedure of the simulation, the range signal parameters of ERS AMI and ENVISAR ASAR are used. All 7 kernels perform better for the signals based on the range signal parameters of ERS AMI than of ENVISAT ASAR and the 6 points RC kernel is recommended in practical SAR data processing. The symbolic function of the Scilab still needs to be enhanced.

Keywords: InSAR, interpolation kernel, phase noise, correlation, Scilab

1. Introduction

Synthetic aperture radar (SAR) is a coherent imaging system, recording both the intensity and the phase of the backscattered signal. As the phase is related to the distance between the sensor and the target, interferometric SAR (InSAR) exploits this coherence, comparing the phase measurements to infer useful geodetic information in two or more SAR images of the same surface.

For its prominent applications including generation of digital elevation models, measurement of glacier flows and mapping of earthquakes, volcanoes and subsidence, InSAR has become one of the most powerful tools in environment microwave remote sensing^{[1][2][3][4]}.

Interpolation is required during the data processing of InSAR. A major reason is that, as InSAR is based on the comparison of the phases of two or more SAR acquisitions of the same surface, it is necessary to register the complex images to within an accuracy of about a tenth of a resolution element. For the purpose of registration, one image is chosen as the master, and the slave (s) should be resampled and mapped onto the grid of the master. In mathematical terms this process is equivalent to an interpolation procedure, that is, reconstruction of the continuous signal from the slave complex data by convolution with an interpolation kernel $i(x,y)$ and sampling of the reconstructed signal at the sampling grid of the master image^[5].

Another reason for the interpolation in InSAR is the prevention of aliasing. While the complex multiplication of two SAR images are applied to get the interferogram, a doubled bandwidth is produced. To prevent aliasing from deteriorating the interferogram, the chosen SAR images should be oversampled by factor 2 or more^{[6][7]}.

Some special requirements exist in the process of the latter-day advanced InSAR technology, so-called PSInSAR (Permanent Scatterers InSAR)^[8] and CRInSAR (Corner Reflectors)^[9]. The first step of the PSInSAR is to identify and isolate in the SAR images the pixels which behave somewhat like point scatterers. And before statistical analysis of the amplitude values of the pixels in the SAR acquisitions, images must be radiometrically corrected in order to make them comparable. According to^[10], oversampling of the images is necessary for the purpose of the radiometric correction, that is, calibration of them. As for the CRInSAR technology, man-made corner reflectors deployed in the field should be recognized in the SAR images. Although the pulse response of a corner reflector shows a Sinc-like shape in the SAR intensity image, the peak value of a corner reflector's backscatter signal may not be sampled by

the digital signal processor of the SAR system for its limited sampling frequency. Only by interpolation may the accurate position of the corner reflector be found in the SAR acquisitions.

In this paper, the requirement for the interpolation in InSAR technology is introduced. And the effective theory^[5] of interpolation errors is briefly summarized. After some interpolation kernels^{[11][12]} are listed, the numerical experimental results based on Scilab are reported and discussed.

2. Model

The proper choice of the interpolation kernel is very important, since the interpolation accuracy and computational efficiency depends on the interpolation kernel and its length. Two types of errors are generally considered in the interpolation process^[5]. One is the distortion of the useful spectral band $|f| \leq B/2$ and the other is the unsatisfactory suppression of the distortion of its replicas $|f - nf_s|, n \neq 0 \leq B/2$, where f_s is the SAR image sampling frequency (Fig.1).

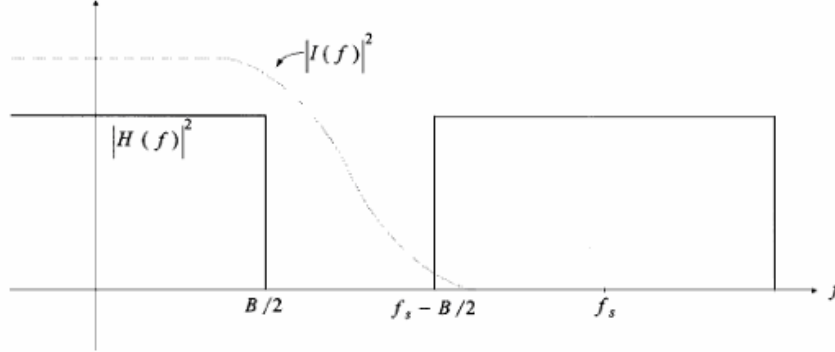


Fig. 1. Fourier transform $I(f)$ of interpolation kernel $i(x)$ and the replicated signal spectrum

Because of the un-strictly low-pass limited interpolated signal, aliasing terms are produced. If the aliasing terms are superposed incoherently, they can be treated as noise and the SNR (Signal to Noise Ratio) is given by:

$$\frac{S}{N} = \frac{\int_{-B/2}^{+B/2} |H(f)|^2 |I(f)|^2 df}{\sum \int_{nf_s - B/2}^{nf_s + B/2} |H(f - nf_s)|^2 |I(f)|^2 df} \quad (1)$$

Under the assumption that the signals are backscattered from circular Gaussian distribution targets, the coherence degradation, resulting from the registration of one SAR complex image onto the second one, $\Delta\gamma = 1 - \gamma$ where the coherence γ is given by:

$$\gamma = \frac{1}{\sqrt{1 + \frac{1}{(S/N)}}} \frac{\int |H(f)|^2 I(f) df}{\sqrt{\int |H(f)|^2 df \cdot \int |H(f)|^2 |I(f)|^2 df}} \quad (2)$$

The coherence degradation $\Delta\gamma$ leads to interferometric phase noise, the standard deviation of which is known to be:

$$\sigma_\phi^2 = \int_{-\pi}^{+\pi} \phi^2 \text{pdf}(\phi) d\phi \quad (3)$$

where, in the N-look case,

$$\text{pdf}(\phi; \gamma, N) = \frac{\Gamma(N+1/2)(1-\gamma^2)^N \gamma \cos \phi}{2\sqrt{\pi}\Gamma(N)(1-\gamma^2 \cos^2 \phi)^{N+1/2}} + \frac{(1-\gamma^2)^N}{2\pi} \cdot {}_2F_1(N, 1; 1/2; \gamma^2 \cos^2 \phi) \quad (4)$$

and in which $\Gamma(\cdot)$ is the Eulerian gamma function and ${}_2F_1(\cdot, \cdot; \cdot; \cdot)$ is the hypergeometric function.

3. Interpolation Kernels

Seven interpolation kernels and their corresponding spectra are summarized as below.

- ◆ Nearest neighbor^[5]:

$$i(x) = \text{rect}(x) = \begin{cases} 0, & |x| > \frac{1}{2} \\ \frac{1}{2}, & |x| = \frac{1}{2} \\ 1, & |x| < \frac{1}{2} \end{cases} \quad (5)$$

$$I(f) = \sin c(f)$$

- ◆ Piecewise linear interpolation ^[5]:

$$i(x) = \text{tri}(x) = \begin{cases} 0, & |x| > 1 \\ 1 - |x|, & |x| < 1 \end{cases} \quad (6)$$

$$I(f) = \sin c^2(f)$$

- ◆ Four-point cubic convolution ($\alpha = -1$) ^[5]:

$$i(x) = \begin{cases} (\alpha + 2)|x|^3 - (\alpha + 3)|x|^2 + 1, & 0 \leq |x| < 1 \\ \alpha|x|^3 - 5\alpha|x|^2 + 8\alpha|x| - 4\alpha, & 1 \leq |x| < 2 \\ 0, & 2 \leq |x| \end{cases} \quad (7)$$

$$I(f) = \frac{3}{(\pi f)^2} [\sin c^2(f) - \sin c(2f)] \\ + \frac{2\alpha}{(\pi f)^2} [3 \sin c^2(2f) - 2 \sin c(2f) - \sin c(4f)]$$

- ◆ Six-point cubic convolution ($\alpha = -\frac{1}{2}, \beta = \frac{1}{2}$) ^[5]:

$$i(x) = \begin{cases} (\alpha + \beta + 2)|x|^3 - (\alpha + \beta + 3)|x|^2 + 1, & 0 \leq |x| < 1 \\ \alpha|x|^3 - (5\alpha - \beta)|x|^2 + (8\alpha - 3\beta)|x| - (4\alpha - 2\beta), & 1 \leq |x| < 2 \\ \beta|x|^3 - 8\beta|x|^2 + 21\beta|x| - 18\beta, & 2 \leq |x| < 3 \\ 0, & 3 \leq |x| \end{cases} \quad (8)$$

The corresponding $I(f)$ expression is not available in closed analytical form.

◆ Truncated sinc^[5]:

$$i(x) = \sin c(x) \text{rect}(x/L) \quad L = 6, 8, 16 \quad (9)$$

$$I(f) = \frac{1}{\pi} (Si(\pi(f + 0.5)L) - Si(\pi(f - 0.5)L))$$

where $Si = \int (\sin(x)/x) dx$ and L is the window size.

◆ Knab SW^[11]:

$$i(x) = \text{sinc}(x) \frac{\cosh\left\{\pi M \nu \sqrt{1 - (x/M)^2}\right\}}{\cosh(\pi M \nu)} \quad (10)$$

where $\cosh(\cdot)$ is the hyperbolic cosine function, and the number of retained sam-

ples is $L=2M$, and $\nu = 1 - 1/\chi$, in which χ is the oversampling ratio of the SAR system. The corresponding $I(f)$ expression is not available in closed analytical form.

◆ Raised cosine (RC)^[12]:

$$i(x) = \text{sinc}(x) \frac{\cos(\alpha\pi x)}{1 - 4\nu^2 x^2} \text{rect}(x/L) \quad (11)$$

where L is the window size and ν , which is called the roll-off factor in [12], is same to the one in the expression of Knab SW. The corresponding $I(f)$ expression is not available in closed analytical form.

4. Experiment using Scilab and the results

In this section we follow the system model ^[5] of Fig.2 and implement some numerical experiments using the free software Scilab 4.0. And the 7 interpolation kernels' impacts to the accuracy of the oversampling of SAR images are evaluated making reference to the ERS AMI and ENVISAT ASAR one-look range signal parameters. Because the function of the symbolic toolbox in Scilab is still limited until the work of this paper, the theoretical evaluation of (2) and (3) is not implemented. And the theoretical findings from [5][11][12] are listed by this paper.

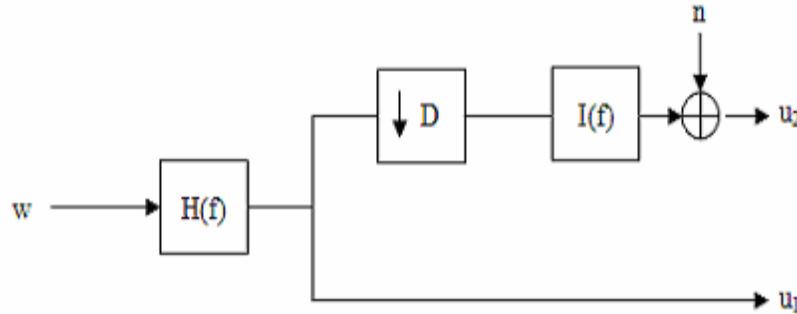


Fig. 2. System model of the numerical experiment for evaluating interpolation errors

```

roll_off = 0.1823; // the roll-off factor in [12]
x = -15:0.1:15;
n = length(x);
filter = zeros(1,n);
xa = abs(x);
a = find(xa<6); // retained points
filter(a) = sinc(%pi.*xa(a)).*cos(roll_off*%pi*xa(a))...
./ (1-4*roll_off^2*xa(a).*xa(a));

```

Fig. 3. The example code for the 6 points RC kernel in Scilab

The simulation procedure in this paper follows the detailed in [5]. In Scilab using 'rand' function we generate two uniformly distributed vector R_1 and R_2 , and then a 1-D white circular Gaussian complex signal w is computed with amplitude $|\omega| = \sqrt{-\ln R_1 [\cdot]}$ and phase $\arg(\omega) = 2\pi R_2 [\cdot]$. To reduce the reference u_1 in Fig.2 the transfer function $H(f) = \text{rect}(f/B)$ should be known. The sampling frequency

$$\phi = \arg[u_1 \cdot u_2^*] \quad (12)$$

$$\hat{\gamma} = \frac{\left| \sum_{k=1}^N u_1^{(k)} u_2^{(k)*} \right|}{\sqrt{\sum_{k=1}^N |u_1^{(k)}|^2} \sqrt{\sum_{k=1}^N |u_2^{(k)}|^2}} \quad (13)$$

Table 1. The coherence and the interferometric phase error for various schemes of interpolation

	Coherence			Phase error [deg] std.		
	Theory (AMI)	Simulate (AMI)	Simulate (ASAR)	Theory (AMI)	Simulate (AMI)	Simulate (ASAR)
Nearest neighbor	0.9132	0.9112	0.9079	37.4	36.3293	36.8555
Piecewise linear	0.9773	0.9770	0.9751	21.4	20.1852	20.9042
4-point cubic	0.9949	0.9948	0.9937	11.3	11.1618	12.0519
6-point cubic	0.9995	0.9987	0.9984	4.19	5.9552	6.5961
6-point tru sinc	0.9975	0.9974	0.9973	8.3	8.0607	8.1258
8-point tru sinc	0.9980	0.9979	0.9979	7.4	7.2499	7.2539
12-point tru sinc	---	0.9990	0.9990	---	5.2737	5.6160
16-point tru sinc	0.9995	0.9994	0.9993	4.1	4.0679	4.2047
6-p Knab SW	0.9999	0.9989	0.9986	1.61	5.3454	6.0101
8-p Knab SW	0.9999	0.9996	0.9995	0.95	3.1666	3.6812
12-p Knab SW	0.9999	0.9998	0.9998	0.30	1.5877	1.8918
16-p Knab SW	---	0.9998	0.9998	---	1.3134	1.5154
6-point RC	0.9999	0.9998	0.9997	0.067	1.9546	2.4578
8-point RC	0.9999	0.9998	0.9998	0.038	1.3421	1.5526
12-point RC	0.9999	0.9998	0.9998	0.010	1.3063	1.4994

f_s and the system bandwidth B are 18.96 MHz and 15.5 MHz for ERS AMI range signal parameters^[5], and the two parameters for ENVISAT ASAR, which can be found in the head file from an ASAR image, are 19.20768 MHz and 16 MHz sepa-

rately. By using the oversampling ratio 12.23 for simulating ERS range signals and 12.00 for ASAR we low-pass filtering w and create u_1 . Downsampling u_1 using a ratio of 1/10 results in a test signal u_t , the oversampling ratio of which is resembling $H(f)$ of ERS AMI(or ENVISAT ASAR). Then using the 7 kernels, implemented in Scilab according to (5)-(11) (the Scilab code of 6-point RC is shown in Fig. 3), we interpolate u_t and get u_2 , the estimate of the reference signal u_1 . To evaluate the kernels the total coherence $\hat{\gamma}$, the interferometric phase error ϕ and its standard deviation are calculated by (11) and (12)^{[5][6]}.

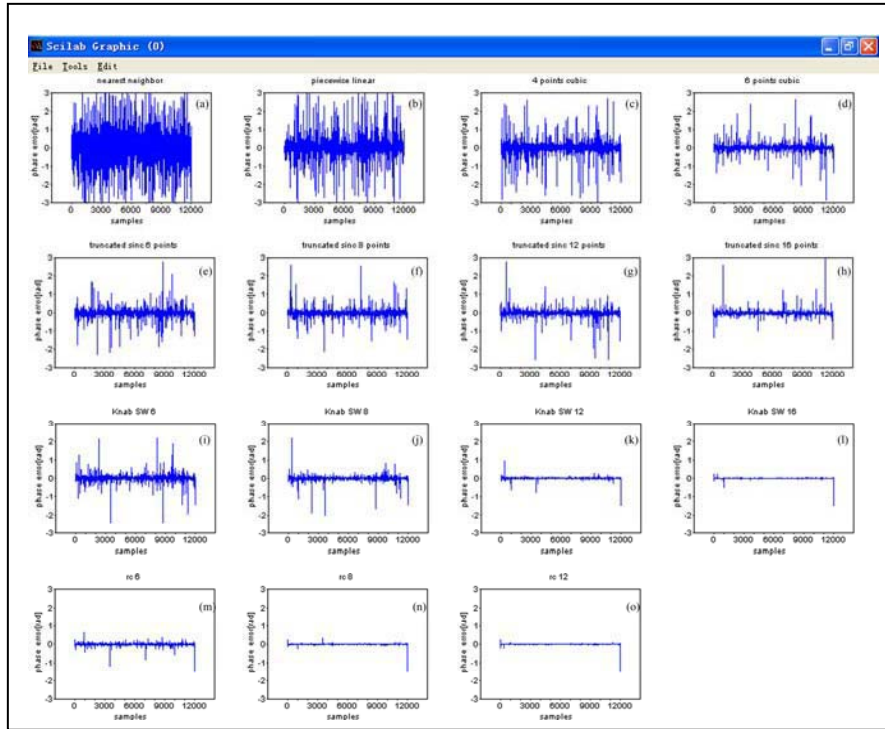


Fig. 4. Simulated phase errors for various interpolation kernels (the name of each kernel on the top of every subplot)

The interferometric phase error of single experiment for the 7 kernels, several of which are with different length, are shown in Fig. 4. And in Table 1 the coherence $\hat{\gamma}$ and the standard deviation of interferometric phase error ϕ , which are the averaged

values from 500 simulation loops, are compared with the theoretical values^{[5][11][12]} from (2) and (3). All the simulation results in Fig. 4 are based on the range signal parameters of ENVISAT ASAR, while the simulated values in Table 1 are calculated from the parameters making reference to the ones of both the ERS AMI and the ENVISAT ASAR range signal.

Table 1 shows all 7 kernels perform better for the range signal parameters of ERS AMI than for ENVISAT ASAR, especially from the simulated standard deviation of the interferometric phase error. It seems that the lower oversampling ratio of ENVISAT ASAR decrease the coherence value and produce more phase error. And in Table 1 and Fig.4, it is obviously shown that with more points retained the interpolation kernel perform better, which is according to the conclusion of [5][11][12].

From the phase error plots in Fig. 4 the better performance of the Knab SW and RC kernel than the other 5 kernels can be observed. For the limited decimal precision, in Table 1 the theory coherence values of the calculated Knab SW and RC kernel, with more than 6 points retained, are all 0.9999. But from the simulated results, especially the standard deviation of the phase error, we can find that the 6 points RC kernel performs better than 8 points Knab SW kernel, and the 8 points RC kernel is almost as excellent as 16 points Knab SW. Although the choice of an optimal interpolation kernel depend on the correlation properties of the signal^[5], based on the numerical experiment results of this paper, 6 points RC is recommended for its accuracy and computation efficiency. But for the more decorrelation in the beginning and ending segments of the interpolated signal, just as shown in Fig. 4, it seems to be wise to interpolate a SAR image with 6 points RC kernel and then discard the pixels at borders.

5. Conclusion

In this paper, we introduced the theory of evaluating interpolation kernels for InSAR data processing illustrated in [5] and listed 7 kernels, that is, nearest neighbor, piecewise linear, four points cubic convolution, six points cubic convolution, Knab SW and Raised cosine kernels. Using Scilab we implemented these kernels and calculated the coherence and the interferometric phase error based on simulation signals making reference to the range signal parameters of both ERS AMI and ENVISAT ASAR. While interpolating the simulated signals using all kernels, the lower oversampling

ratio of ENVISAT ASAR seems decrease the coherence value and produce more phase error. As for the interpolation accuracy, RC and Knab SW kernels are better than the other conventional kernels. 6 points RC kernel performs better than 8 points Knab SW kernel and is recommended in practical SAR data processing. It's shown that Scilab is a useful tool for InSAR application because of its powerful functions and free spirit. But until the work of this paper, the symbolic function of the Scilab still needs to be enhanced.

Acknowledgment. This work was supported by China Geological Survey (CGS), the Ministry of Land and Resources. And the authors would like to thank Dr. Ramon Hassen for his kindly help.

Reference

1. Graham L., "Synthetic interferometric radar for topographic mapping," *Proceedings of IEEE*, 6, 763– 768, 1974.
2. Zebker, H. A. and R. M. Goldstein, "Topographic mapping from interferometric synthetic aperture radar observations," *Journal of Geophysical Research-B5*, 91, 4993– 4999, 1986.
3. Massonnet, D., P. Briole, and A. Arnaud," Etna monitored by spaceborne radar interferometry," *Nature*, 375, 567– 570, 1995.
4. Bamler, R., M. Eineder, B.M. Kampes, H. Runge, and N. Adam. "SRTM and beyond: current situation and new developments in spaceborne INSAR. (Invited paper),"in *Proceedings ISPRS Workshop*, Hannover, Germany, October 2003: Joint Workshop High resolution mapping from space, 8 pp., 2003.
5. Hanssen, R. and R. Bamler, "Evaluation of interpolation kernels for SAR interferometry," *IEEE Trans. Geosci. Remote Sensing*, Vol. 37, 318–321, 1999.
6. Hanssen, R. F.,*Radar Interferometry-Data Interpretation and Error Analysis*, Kluwer Academic Publishers, New York, 2002.
7. Kampes, B., Hanssen R. and Zbigniew Perski. "Radar Interferometry with Public Domain Tools," In: *Proceedings of FRINGE 2003*, December 1-5, Frascati, Italy, 2003.
8. Ferretti, A., C Prati, and F Rocca," Permanent scatterers in SAR interferometry," *IEEE Transactions on Geoscience and Remote Sensing*, 39(1):8-20, January 2001.
9. Xia Ye, Kaufmann H and Guo Xiaofang," Differential SAR interferometry using corner reflectors," *IEEE Proceedings of IGARSS' 2002*, Toronto (CD-ROM) ,2002.

10. Laur H., Bally P., Meadows P., Sanchez J., Schaettler B., Lopinto E., and D. Esteban, "Derivation of the backscattering coefficient sigma-nought in ESA ERS SAR PRI," ESA Document No: ES-TN-RS-PM-HL09, http://earth.esa.int/esc_intro.htm, Ispra, Italy, Sept. 1998.
11. Migliaccio, M. and F. Bruno, "A new interpolation kernel for SAR interferometric registration," IEEE Trans. Geosci. Remote Sensing, Vol. 41, 1105 - 1110, 2003.
12. Byung Lae Cho ,Young Kyun Kong and Young Soo Kim, "interpolation using optimum NYQUIST filter for SAR interferometry," J. of Electromagn. Waves and Appl., Vol. 19, No. 1, 129 - 135, 2005.

Scilab Grid Toolkit with Open Source ProActive

Denis Caromel and Arnaud Mangin

INRIA Sophia-Antipolis, CNRS, I3S, UNSA. 2004, Route des Lucioles, BP 93,
F-06902 Sophia-Antipolis Cedex, France.

`First.Last@sophia.inria.fr`

Abstract. Scilab is an Open Source library and environment for scientific applications. In this paper, we present an Open Source extension that supports the execution of parallel and independent tasks in a Grid setting.

This extension uses the Java ProActive middleware to manage parallel and distributed task executions. Using a exible API, this extension allows to build parallel and distributed Java applications taking advantage of the powerful computing environment of Scilab, as well as on a cluster or a network of workstations, giving access to computational power and a huge aggregate memory sizes. It also provides a seamless way to run unmodified Scilab scripts in a Grid context.

1 Introduction

Grid computing is becoming a crucial aspect of scientific computing. Major scientific environments now offer Grid related packages. The Java environment is suitable to build parallel and distributed programs in an object-oriented programming model. Portability, multi-threading, and remote method invocations (RMI) are key features. RMI presents two main drawbacks. Firstly, it only provides synchronous and point-to-point communication. Secondly, it does not provide a standard for the deployment of distributed applications. In this paper, we present an extension of Scilab [10], a software for numerical computations, with a generic and portable toolkit for the Grid. It includes a exible application programming interface (API), and a graphical user interface (GUI). This extension allows to build parallel and distributed Java applications taking advantage of the powerful computing environment of Scilab and to run usual Scilab scripts in a parallel and distributed way.

This toolkit was designed with the ProActive middleware [17] to benefit of the Java platform and to address the limits of the RMI model. The other important points about ProActive is its integration with many Grid computing standards (LSF, PBS, NorduGrid, Globus, EGEE, etc). The synergy created between the ObjectWeb consortium, the ProActive Grid community, and the Scilab consortium is another important aspect of this overall work.

This document is organized as follows. In section 2 we provide some background on the Scilab environment and on the ProActive middleware. In section 3 we describe the implementation of the Scilab Grid API, then, in section 4

we present some performances and benchmarks. Next in section 5 we give an overview of different parallel computing architectures and finally we conclude in section 6 and we present the ongoing work: the integration of a framework in order to design complex applications thank to recurring patterns for task and data parallelism.

2 Background

2.1 Scilab

Scilab [10] is an open source software similar to Matlab for numerical computations, providing a powerful open interactive environment. It provides an extensible set of primitives and data structures (Fortran or C subprograms) for a wide variety of calculations. This set is easily mapped by a script language for an use in the interpreted environment of Scilab. Scilab also provides the possibility to use these features by an external application. For instance in Java, Scilab is viewed like a single library linked to the application.

2.2 ProActive

ProActive [17] is a Java Grid middleware (under LGPL licence) for concurrent, distributed, and mobile computing. ProActive is part of the ObjectWeb consortium for Open Source middleware. It provides a comprehensive API to simplify the programming of Grid Computing applications distributed on Local Area Network (LAN), on clusters of workstations, or on Internet Grids.

ProActive uses an *active object* (AO) programming model. AO are remotely accessible via method invocation, automatically stored in a queue of pending requests. Each AO has its own thread of control and is granted the ability to decide in which order incoming method calls are served (FIFO by default). Method calls on AO are asynchronous with automatic synchronization (including a *rendezvous*). This is achieved using automatic *future objects* as a result of remote methods calls, and synchronization is handled by a mechanism known as *wait-by-necessity* [7].

In distributed programming, the deployment of grid applications is a central problem, rather difficult to achieve. ProActive succeeds in completely avoiding scripting (for configuration, getting computing resources, etc.). ProActive provides a *Descriptor Deployment Model* [5], which allows the deployment of applications onto any kind of parallel machines or Grid using heterogeneous protocols, without changing the application source code. All information related to the deployment of the application is described in the XML Deployment Descriptor. Thus, eliminating references inside the code to: machine names, submission protocols (rsh, ssh, Globus, LFS, PBS, etc.) and communication protocols (rmi, jini, http, etc.). It also features tunneling of communications, for sakes of both security and firewall crossing.

The communication ow that takes place between the various entities is one of the most significant features to track in distributed applications. This is why

ProActive provides a graphical environment within IC2D [4] for the monitoring of ProActive applications. IC2D tracks, on demand, all events that relate to remote method calls between *active objects*.

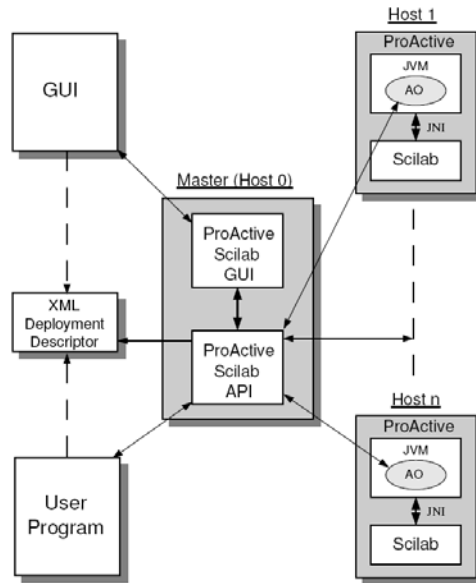


Fig. 1. Organization of the Grid Scilab Toolkit

3 Scilab Grid Toolkit

The objectives of the Scilab Grid Toolkit is to extend Scilab for parallel and distributed computing in the Java environment, and to benefit from the programming and deployment models of ProActive. This framework allows the deployment of Scilab runtimes over a set of hosts and to use these instances like computing engines monitored by a central ProActive application according to the *master/slave* model. The choice of a centralized view aims to ease the managing of the application. Each node is an independent entity, i.e. one task computation is achieved by a single engine without communication with any other nodes. The topology of the application can evolve at run-time: any node can be deleted/created at any time. The client interacts directly with the monitor to deploy the topology, to submit tasks, and to retrieve results.

Figure 2.2 summarizes the principles and the architecture of the Scilab Grid Toolkit:

- A set of Scilab instances can be launched on a set of desktop or cluster machines.
- Each Scilab instance communicates with a ProActive Java Virtual Machine through JNI, and an Active Object living in each JVM provides remote access to the Scilab instance.
- A dedicated API allows from a master JVM:
 - to create a set of remote Scilab engine,
 - to asynchronously send scripts to be executed on a given Scilab engine,
 - to monitor and to synchronize with the script completion.
- A GUI, its implementation using the API above, allows to:
 - configure where to launch remote Scilab engine from an external XML descriptor (allowing to use simultaneously desktop machines, dedicated clusters under various protocols, and ssh tunneling when needed),
 - launch, monitor, cancel, and visualize the execution of Scilab scripts.

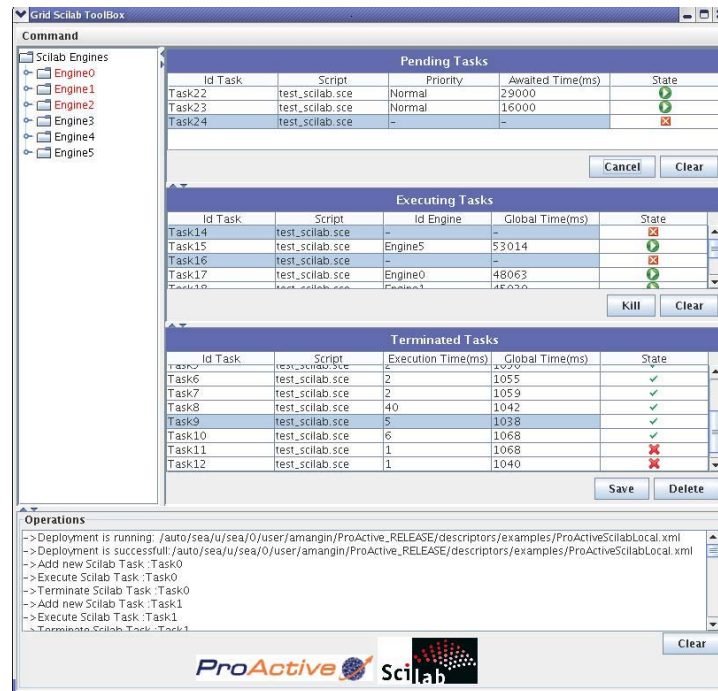


Fig. 2. The Scilab Grid GUI

We have implemented the Scilab Grid API as service methods available in the ProActive library. Figure 3 represents the set of methods to deploy Scilab engines with ProActive and to free the resources. Figure 4 represents the methods to manipulate a given Scilab Engine instance (activation, execution, ending).


```

// Deploy ScilabEngine instances for each node defined in the xml descriptor passed as
parameter
public synchronized static HashMap deploy(String[] arrayIdEngine, String
pathDescriptor);

// Deploy a new ScilabEngine instance over the node passed as parameter
private synchronized static ScilabEngine deploy(String idEngine, Node node);

// Kill the node passed in parameter
public static synchronized void kill(Node node);

// Kill all deployed nodes
public static synchronized void killAll();

```

Fig. 3. Scilab Grid API: Deployment

```

// Execute the Scilab task passed as parameter
// the method call is asynchronous
public ScilabResult execute(ScilabTask scilabTask);

// activate the ScilabEngine instance
public BooleanWrapper activate();

// kill the current Scilab task
public void killTask();

// Exit the ScilabEngine instance
public void exit();

```

Fig. 4. Scilab Grid API: Scilab Engine

4 Benchmarks

The goal of this section is to compare the overhead and the speed-up introduced by this new interface. These benchmarks were achieved from a parallel algorithm that creates 2-dimensional pictures of the Mandelbrot set. The parallelizing of the Mandelbrot set was chosen for the following reasons:

1. It does not require interprocess communication (only for the termination detection).
2. It is easy to make a parallel implementation, each point can be calculated independently from all others.
3. It is irregular, the amount of work required to compute each task, is not known.
4. For a large number of iterations and a large size of matrix the algorithm becomes CPU intensive.
5. For a large size of matrix the communications become significant.

These benchmarks were achieved onto a Desktop Grid, with 20 workstations based on the kernel Linux 2.6 and interconnected by a Fast Ethernet network switched (100 Mbps/sec). The Table 1 describes the characteristics of the test platform.

#Workstations	CPU	RAM
7	Intel Pentium 4 3.40 GHz	1GB DDR
8	Intel Pentium 4 3.40 GHz	2GB DDR
5	Intel Pentium 4 3.40 GHz dual	2GB DDR
total 20		

Table 1. Characteristics of the Desktop Grid

Figure 5 shows the computation time for both sequential and parallel algorithm functions of the size of square matrices. The limit is fixed at 10. For the parallel version the number of tasks is constant (60 tasks). Figure 6 shows the overhead inferred on the performances of the parallelizing; the curves represent the computation time of the parallel algorithm functions of the number of tasks. The size of square matrices is 100*100 and the limit is fixed at 1000. These benchmarks were computed over 5, 10, 15 and 20 nodes

Comparing the timing results of the Scilab Grid Interface and Scilab, we see the efficiency of the *master/slave* mechanism that enables some linear performance. The number of task can affect the performances. The Figure 6 shows that:

- For a large number of tasks and a small size of matrix or a small limit: the algorithm spends more time to send/receive a task than to compute it.
- For small number of tasks and a large size of matrix or a large limit: the *master/slave* mechanism is not efficient.

5 Related Work

To satisfy the requirements of modern engineering with large simulation problems, we are seeing increasing interest in developing Grid related packages for computing environments. There are many organizations that would benefit these environments of Grid technologies. A survey [8], was done about parallel Matlab projects. These projects vary in their scope.

- The *Embarrassingly Parallel* approach makes use multiple processes of the computing environment, running without coordination. A parent process passes off data to the child processes. Then all processes work on its local data and return the result to the parent process. With this model, a limited number of functions can be parallelized but it has the advantage of simplicity. Software that makes use this approach: Multi [14], Paralize [1], and PLab [13].
- The *Message Passing* approach provides message passing routines between processes. This enables to write a user parallel programs in a fashion similar to parallel programming in MPI [16]. Indeed the routines provided are wrappers for MPI. This approach has the advantage of exibility. Software that makes use this approach: MPITB/PVMTB [3] , CMTM [19], and MultiMatlab [18].

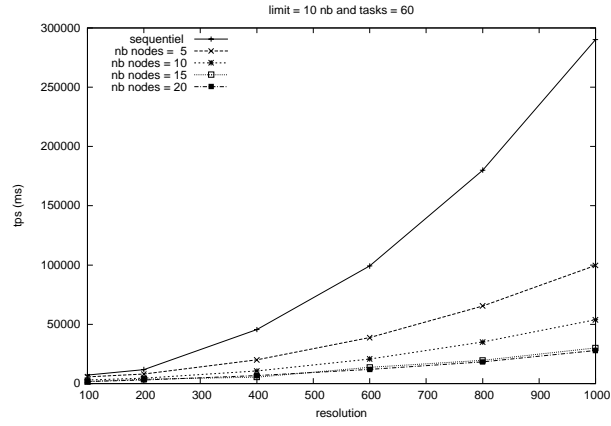


Fig. 5. Computation time (limit = 10 & nb tasks = 60)

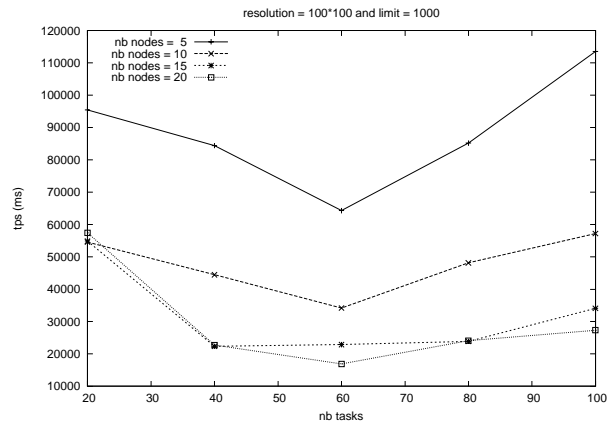


Fig. 6. Computation time (limit = 1000 & size = 100*100)

- The *Back End Support* approach use the computing environment as a front end for a parallel computation. This computation is achieved by an engine, usually using high-performance numerical computation library like ScallAPACK [6]. The advantage of this approach is that it only requires one session of the environment (therefore, only one licence), and usually does not require the user to have knowledge of parallel programming. Software that makes use this approach: MATLAB *P [9], NetSolve [2], and DLab [15].

The Scilab software has also an interface for parallel computing based on the *Message Passing* approach [11]. This interface uses PVM [12] (Parallel Virtual Machine): a set of packages that enables to define a heterogeneous collection of computers hooked together by a network to be used as a single large parallel computer. The PVM library offers a large set of routines accessible via Fortran or C user programs such as process initialization, message transmission and reception, and synchronization.

Compare to the works above, the ProActive Scilab Grid Toolkit features the following advantages:

- deployment of remote Scilabs with a large variety of protocols (LSF, PBS, Globus, Sun Grid Engine, OAR, EGEE gLite, NorduGrid, etc.), including deployment on parallel machines, clusters, and desktop machines,
- addressing machines behind firewall or NAT,
- no pre-configurations are needed since ProActive deploys without installation, and Scilab itself could be deployed on the y if needed,
- a powerful Java API allows to control the asynchronous execution, with, for instance, group communication, and the management of groups of futures,
- a powerful GUI, with exible configuration of the machines to be used being given in an external XML file.

6 Conclusion and Future Work

In this paper, we have given an overview of the Scilab Grid Toolkit. The main objective was to extend the Scilab environment towards the parallel and distributed programming. We have used the ProActive Grid middleware. As it was described in Section 3, we have designed the API and have taken advantage of both the simplicity and the powerfulness of this model for some applications.

Another extension being currently designed and implemented is the capacity to execute parallel Scilab scripts that communicate. The principal is to provide a few simple primitives allowing from within Scilab scripts to send/receive information to/from other scripts. Using the Object-Oriented SPMD capacity of ProActive to handle the addressing of script instances. The ProActive feature of full connectivity will enable inter-script communications even in the case of firewall, NATs, or mixed cluster/desktop deployments. Finally, the Java API will provide full-etched Object-Oriented communication and synchronization when the user will address the Scilab library from the Scilab Java API.

References

1. T. Abrahamsson. Paralize, 1998.
<ftp://ftp.mathworks.com/pub/contrib/v5/tools/paralize/>.
2. D. ARNOLD, S. AGRAWAL, S. BLACKFORD, J. DONGARRA, M. MILLER, K. SAGI, Z. SHI, and S. VADHIYAR. Users' guide to netsolve v, 2001.
<http://citeseer.ist.psu.edu/475297.html>.
3. Javier Fernandez Baldomero. Mpi/pvm toolbox for matlab, 2000.
<http://atc.ugr.es/javier-bin/mpitb>.
4. F. Baude, A. Bergel, D. Caromel, F. Huet, O. Nano, and J. Vayssière. IC2D: Interactive control and debugging of distribution. *Lecture Notes in Computer Science*, 2179:193+, 2001.
5. F. Baude, D. Caromel, L. Mestre, F. Huet, and J. Vayssière. Interactive and descriptor-based deployment of object-oriented grid applications. In *Proceedings of the 11th IEEE International Symposium on High Performance Distributed Computing*, pages 93–102, Edinburgh, Scotland, July 2002. IEEE Computer Society.
6. Laura Susan Blackford, J. Choi, A. Cleary, A. Petitet, R. C. Whaley, J. Demmel, I. Dhillon, K. Stanley, J. Dongarra, S. Hammarling, G. Henry, and D. Walker. ScaLAPACK: A portable linear algebra library for distributed memory computers — design issues and performance. pages ??–??. 1996.
7. D. Caromel. Toward a method of object-oriented concurrent programming. *Communications of the ACM*, 36(9):90–102, 1993.
8. R. Choy. Parallel matlab survey, 2001. <http://theory.lcs.mit.edu/cly/survey.html>.
9. "R. Choy. *Matlab*p 2.0: Interactive supercomputing made practical*. PhD thesis, Massachusetts Inst, 2002. 67p.
10. Scilab Consortium. <http://www.scilab.org>.
11. F. Despres, C. Gomez E. Fleury, S. Steer, and S. Ubeda. Bringing metacomputing to scilab. pages 398 – 403, 1999.
12. Al Geist, Adam Beguelin, Jack Dongarra, Weicheng Jiang, Robert Manchek, and Vaidyalingam S. Sunderam. *PVM: Parallel Virtual Machine: A Users' Guide and Tutorial for Networked Parallel Computing*. MIT Press, Cambridge, MA, USA, 1994.
13. Ulrik Kjems. Plab, 2000. <http://bond.imm.dtu.dk/plab/>.
14. Tom Krauss. Multi - a multiple matlab process simulation engine, 2000. http://www.lapsi.eletr.ufrgs.br/Disciplinas/ENG_ELETRICA/CAD-ENG/Matlab/CommSim/.
15. B.R. Norris. Dlab: An environment for interactive parallel numerical computing. Technical report, Urbana, Illinois, 1999.
16. Jeffrey M. Squyres. Definitions and fundamentals – the message passing interface (MPI). *ClusterWorld Magazine, MPI Mechanic Column*, 1(1):26–29, December 2003.
17. ProActive INRIA Sophia Antipolis OASIS Team. <http://proactive.objectweb.org>.
18. A.E. Trefethen, V.S. Menon, C.C. Chang, G.J. Czajkowski, C. Myers, and L.N. Trefethen. An environment for interactive parallel numerical computing. Technical report, Multimatlab: Matlab on multiple, 1996.
19. J. Zollweg. Cornell multitask toolbox for matlab, 2001.
<http://www.tc.cornell.edu/Services/Software/CMTM/>.

Some practical experiences with Scilab/Scicos and RTAI-Lab at the SUPSI laboratory

Roberto Bucher and Silvano Balemi

Scuola Universitaria Professionale della Svizzera Italiana (SUPSI),
Dipartimento Tecnologie Innovative,
CH-6928 Lugano-Manno,
[roberto.bucher, silvano.balemi]@supsi.ch,
WWW home page: <http://www.dti.supsi.ch/~bucher>

Abstract. This paper shows how the Scilab/Scicos and the RTAI-Lab environments can successfully be used in a didactic laboratory as a Rapid Control Prototyping (RCP) system to generate hard real-time code for different control activities. In particular, these environments have been used at SUPSI laboratory to perform different tasks: data acquisition and identification of plant models, design and implementation of complex control algorithms, monitoring of the real-time task for data collection and parameters tuning. These activities are presented with concrete examples on three different laboratory systems

1 Introduction

The Scuola Universitaria Professionale della Svizzera Italiana (SUPSI) is a university of applied sciences which offers students the possibility to get a specialization in mechatronic in the third study year. One of the most important parts of this study is represented by the mechatronic laboratory, where students can put all the tasks connected to the courses of *Modelling and Identification*, *Control Design*, *Mechatronic systems* and *Sensors and actuators* into practice. Rapid Control Prototyping methods (RCP) are used to achieve these objectives.

Section 2 gives an overview about the mechatronic laboratory at SUPSI. Section 3 presents a state feedback control of a simple servo DC motor. The identification of a voice coil motor for nanopositioning is described in section 4. The classical inverted pendulum is described in section 5, where the PC is connected to the plant using a CAN bus. Some conclusions about the experiences collected in the past 4 years are reported in section 6.

2 The SUPSI laboratory

2.1 Equipment

Each labor place is composed of a PC (Pentium 4, 2.8 GHz) with Linux RTAI as real-time operating system, a CAN bus interface (Peaks PCAN or a self built

EPP CAN Dongle) and an embedded system (Compact PCI) as supplementary RT target for distributed control. Some PCs are equipped with AD/DA cards (Computer Measurements PCI-1200 DAQ cards).

2.2 Linux RTAI

The RTAI extension was created as an environment for implementing low cost data acquisition and digital controller systems ([1], [2]) and is intended as an open source replacement of other hard real-time OSs such as QNX or VxWorks. This extension adds hard real-time capabilities to Linux, allowing sample frequencies up to several thousands of cycles per second with a jitter of just a few microseconds. Real-time processes can run both in the kernel and in the user area. The low latencies guaranteed by Linux RTAI allows us to implement controllers for systems with a bandwidth up to a few hundred hertz.

2.3 RTAI-Lab

RTAI-Lab is an open source project which integrates the code generated by a CACSD (Computer Aided Control System Design) environment into a Linux RTAI hard real-time task. At present, three CACSD suites are supported: the commercial MATLAB/Simulink/RealTime-Workshop (RTW) suite, the commercial EicasLab suite and the open source SCILAB/Scicos suite. The system has been widely described in [3], [4], [5], [6] and [7]. The document [8] describes in detail how to install the full RTAI-Lab toolchain. RTAI-Lab provides a GUI application for remote monitoring and controlling of hard real-time generated tasks.

3 Control of a DC servo motor

3.1 Plant description

The plant is represented by a servo DC motor from Siemens AG (Fig. 1)

This plant is part of an Universal Experimenter Didactic System, which allows the investigation of PID controllers. In our case, the controller is implemented as a digital controller in the Linux RTAI OS and only the current amplifier and the plant are considered. The interface between the PC and the plant is implemented by a AD/DA card of Computer Measurements (PCI-1200) using the COMEDI interface([9]) which sends the control signal to the motor and reads the position using a potentiometer.

3.2 Identification

A simplified model of the servo DC motor is represented by

$$G(s) = \frac{K}{s^2 + \alpha \cdot s} \quad (1)$$

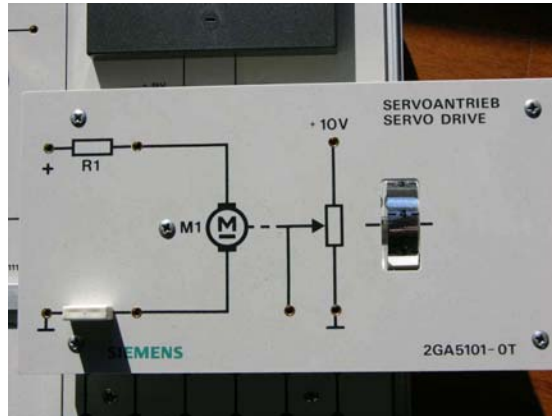


Fig. 1. Servo DC motor

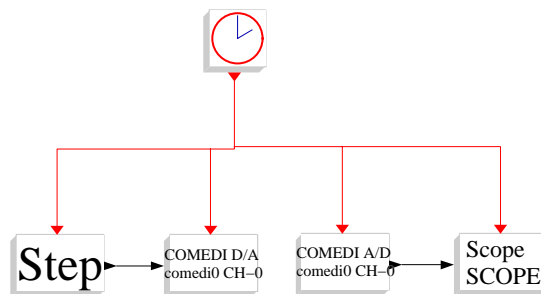


Fig. 2. Block diagram for the identification

The Scicos block diagram used for the identification is represented in Fig. 2.

The motor can get control signals between $-10V$ and $10V$; a step signal of $8V$ is sent to the real plant and the response is collected using the *xrtailab* application. A simple script (see 3.6) performs the identification using the *leastq* command of the Scilab environment. The measured data and the step response of the identified plant are shown in Fig. 3.

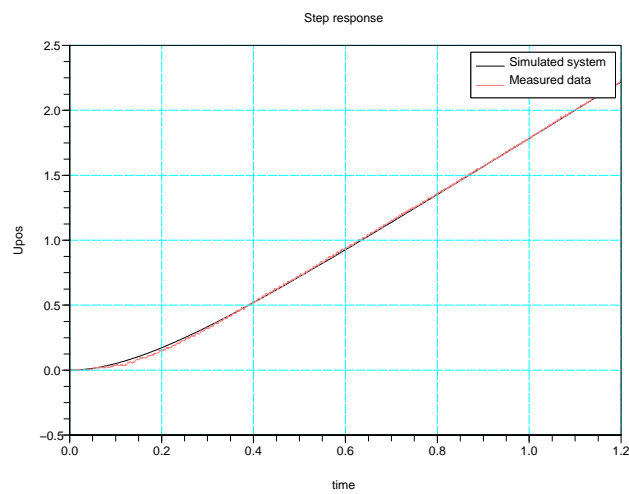


Fig. 3. Measured and simulated step response

3.3 Controller design

The state feedback controller has been implemented using the pole placement method. The 3 poles of the controlled system are placed near to the position of the plant pole. The script which calculates the controller is the same used for the identification (see subsection 3.6). A discrete reduced order observer is used to generate the unknown state of the plant (see 7.1).

3.4 Simulation

The same Scicos block diagram has been used for simulation and realtime control; the only difference between the two block diagrams is represented by the superblock *plant* as shown in Fig. 4 and by the *scope* block.

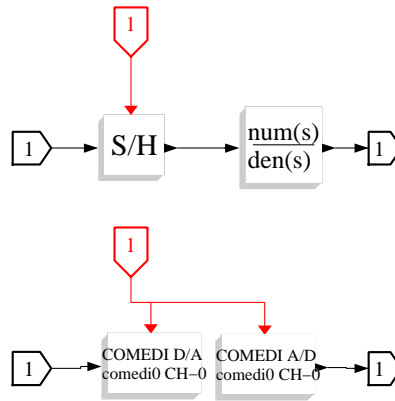


Fig. 4. Scicos blocks for simulation (top) and real-time control (bottom)

3.5 Real-time controller

In the case of the real-time control, the mathematical representation of the plant is substituted by two COMEDI blocks, in order to send and collect data from the AD/DA card (see Fig. 4).

The controller has been realized under Scicos (see Fig. 5). It contains a discrete reduced order observer and an integral part with anti-windup to eliminate the steady state error.

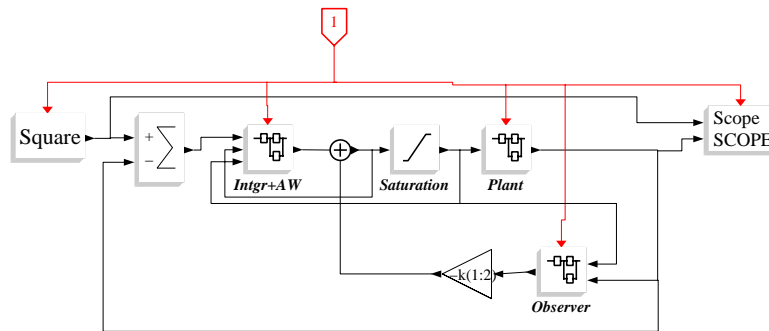


Fig. 5. Block diagram for the controller

Fig. 6 shows the plot of the simulation vs. the plot of the real plant.

3.6 Scilab script

```
// == Servo DC: Identification and Control design ==
```

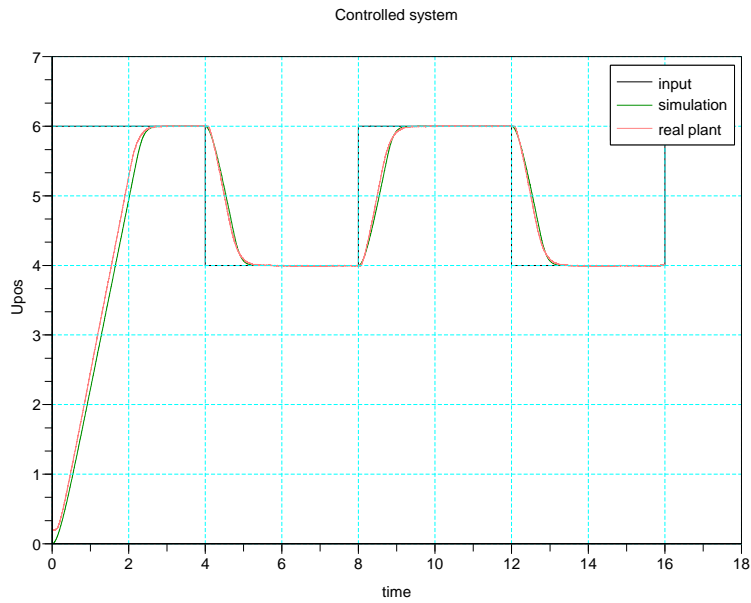


Fig. 6. Controlled system - Response of the simulation and of the real plant

```

x=read('SCOPE',-1,2);
t=x(:,1);
y=-x(:,2);
y=y-y(1);
t=t(1:4001);
y=y(1:4001);
xbascc()
plot(t,y)

ts=1e-3;

function z=fun(p,t,y)
z=y+p(1)/p(2)^2-p(1)/p(2)*t-p(1)/p(2)^2*exp(-p(2)*t);
endfunction

Uo=8;
p0=[1,4];
p02=[1,4,1];

[ff,p]=leastsq(list(fun,t,y/8),p0);

g=syslin('c',p(1)/(s*(s+p(2))));
xbascc()
bb_step(Uo*g,4)
plot(t,y)

num=g.num;
den=g.den;

g_ss=tf2ss(g);
p_g=roots(den);
g_dss=dscr(g_ss,ts);

```

```

p_max=max(abs(p_g));

p_c=[-p_max,
      -p_max*cos(pi/6)+%i*p_max*sin(pi/6),
      -p_max*cos(pi/6)-%i*p_max*sin(pi/6)];
k_poles=1.2;
p_c=k_poles*p_c;

p_d=exp(p_c*ts);
[Phi,G,C,D]=abcd(g_dss);

[m1,n1]=size(Phi);
[m2,n2]=size(G);
Phi_f=[Phi,zeros(m1,1);-C(1,:)*ts,1];
G_f=[G;zeros(1,n2)];

k=ppol(Phi_f,G_f,p_d);
k_sat=k(3)/30;

p_oc=[-5*p_max];
p_od=exp(p_oc*ts);

T=[0,1];
[Ao,Bo,Co,Do]=redobs(Phi,G,C,D,T,p_od);

```

4 Identification of a voice coil motor

4.1 Plant description

One of the most difficult tasks in control system design is represented by the identification of the plant model. This section describes how Scilab/Scicos/RTAI has been used to find the non-parametric and the parametric identification of a voice coil motor. SUPSI has great experience in controlling high precision systems ([10]). The plant to be identified is a voice coil motor (Fig. 7) used for high precision positioning. It can perform displacements up to $1mm$ with a precision of about $50nm$.

The controller is implemented in a Compact PCI system, where a 10MB Linux-RTAI with full network support is stored into a Flash Card. The two cards to interface the real plant (a DA analog card and a sincos encoder card) have been designed and built at SUPSI. The hard realtime drivers have been also created.

The controller is designed and simulated on a second PC. The generated code is then downloaded and executed into the Compact PCI. The *xrtailab* monitor application can be started on each PC in the LAN to monitor the hard real-time task.

4.2 Identification

The scicos block diagram for the identification is shown in Fig. 8.

The plant is excited by a white noise signal of $10s$ which is sent repeatedly to the plant. This signal and the sensor values are collected by the block *signal means*. This block collects the input data and creates a cumulative sum of packet of $10s$. When the generated hard realtime task is stopped, this block creates a

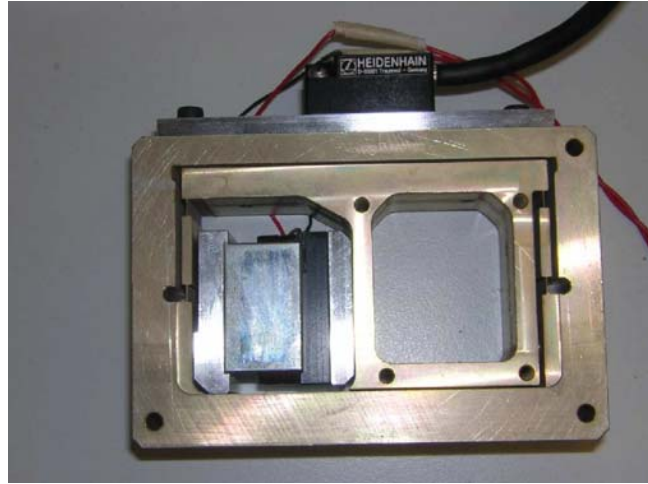


Fig. 7. Voice coil motor

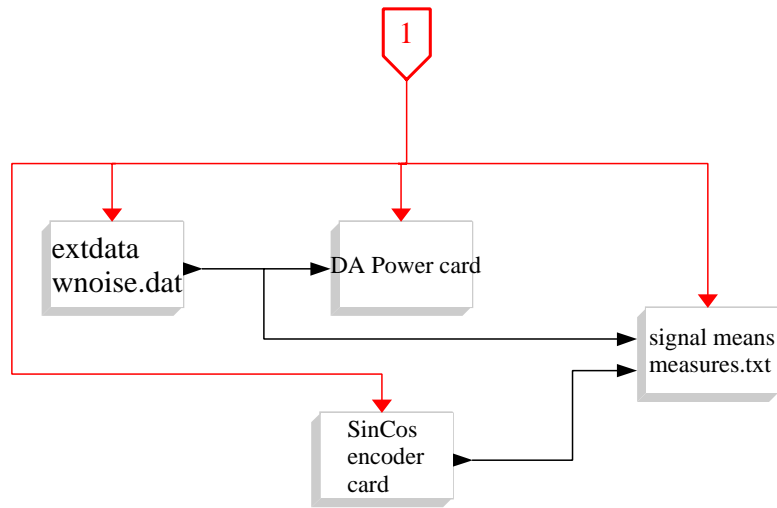


Fig. 8. Block diagram for the identification

mean of the collected signals in order to reduce the effect of sensor noise, and store it in the file *measures.txt*. The script in subsection 4.3 performs the identification of the plant. The core of the script is the function *xspectrum* (see 7.4) which performs the non-parametric identification of the system (represented in Fig. 9); the parametric identification of the plant is calculated using the scilab function *frep2tf* (the 2. order term in the numerator has been eliminated). The non parametric model and the gainplot of the parametric model are represented in Fig. 9.

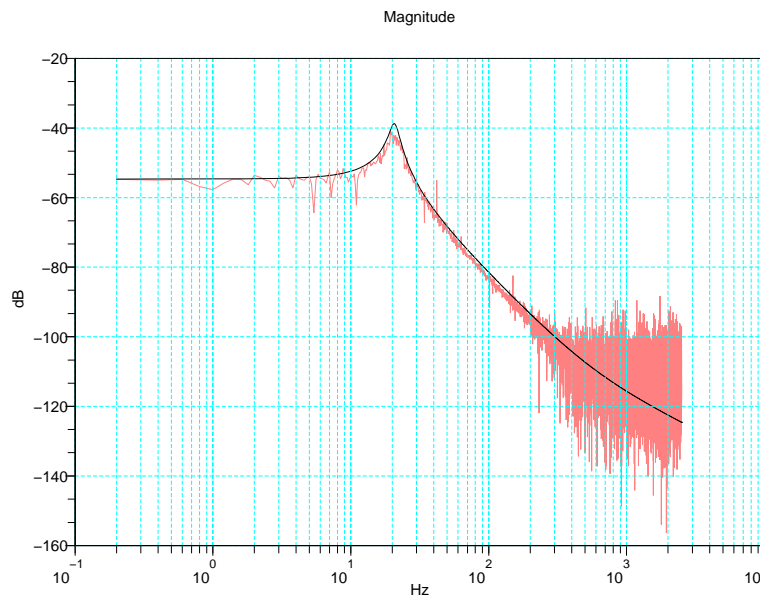


Fig. 9. Plot of the measured and identified system

4.3 Identification script

```

data=read('measdata.dat',-1,3);
x=data(:,2);
y=data(:,3)/12;
ts=1/5000;

P=xspectrum(x,y);
frq=0:1/ts/2/length(P):(1/ts/2-1/ts/2/length(P));
xbasc()
gainplot(frq(2:$),P(2:$)')

N=2835;
h=frep2tf(frq(2:N),P(2:N),2);
h=ss2tf(tf2ss(h));

```

```

num=h.num
den=h.den
coeff_n=coeff(num)
coeff_n=coeff_n(1:2)
num=poly(coeff_n,'s','coeff')
sys=syslin('c',num/den)
repf=repfreq(sys,frq(2:$));
gainplot(frq(2:$),repf)

```

5 Control of the inverted pendulum

5.1 Plant description

The inverted pendulum is a classic application in a student laboratory. Our system is connected to the controller (a normal PC with Linux RTAI) through a CAN Bus. The CAN interfaces under Scicos have been realized in a three layer hierarchy: a middle layer has been introduced as a *generic CAN interface* to connect the different CAN devices to the different CAN cards (as shown in Fig. 10). In the SUPSI laboratory we have different CAN cards: for most of them we rewrote the driver in order to use them in a hard real-time environment. In particular, we have real-time drivers for CAN cards of Peak System ([11]) and for a self-built parallel port dongle.

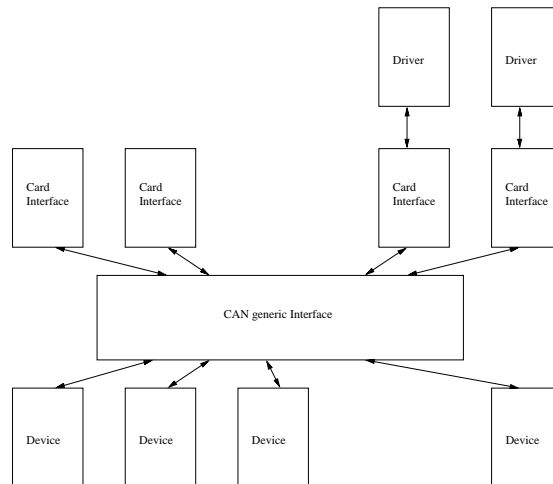


Fig. 10. Three layer CAN implementation

The actuator is a DC motor from Maxon (RE 40) which is controlled by current through an Epos driver (EPOS P 24/5) ([12]). The same driver is used to count the impulses of the incremental encoder of the motor to get its position. The angle of the pole is measured using a second incremental encoder, which sends via RF the impulses to a receiver. The power supply to this encoder and to the transmitter has been realized by current induction.

5.2 Identification

The linear state-space model of the inverted pendulum is

$$\begin{bmatrix} \dot{\varphi} \\ \dot{\omega} \\ \dot{x} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \frac{m \cdot r \cdot (M+m) \cdot g}{\Theta \cdot (M+m) - m^2 \cdot r^2} & 0 & 0 & \frac{-m \cdot r \cdot d}{\Theta \cdot (M+m) - m^2 \cdot r^2} \\ 0 & 0 & 0 & 1 \\ \frac{m^2 \cdot r^2 \cdot g}{\Theta \cdot (M+m) - m^2 \cdot r^2} & 0 & 0 & \frac{-\Theta \cdot d}{\Theta \cdot (M+m) - m^2 \cdot r^2} \end{bmatrix} \cdot \begin{bmatrix} \varphi \\ \omega \\ x \\ v \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{m \cdot r}{\Theta \cdot (M+m) - m^2 \cdot r^2} \\ 0 \\ \frac{\Theta}{\Theta \cdot (M+m) - m^2 \cdot r^2} \end{bmatrix} \cdot F \quad (2)$$

where M is the mass of the car, m is the mass of the pole, r is the distance to the center of mass of the pole, Θ is the inertia of the pole and d is the friction constant of the car.

The force F is obtained by the torque given by the motor and it is proportional to the motor current:

$$F = K_t \cdot \frac{I_{mot}}{r_p} \quad (3)$$

where r_p represents the radius of the motor gear.

The distance r to the pole center of mass is calculated from the geometry of the pole (different volumes V_a and V_m of the pole parts)

$$r = \frac{V_a \cdot \frac{r_a}{2} + V_m \cdot (r_a + \frac{r_m}{2})}{V_a + V_m} \quad (4)$$

The value of Θ can be identified by measuring the swing frequency (\rightarrow the oscillation time T_{osc}) of the pole as

$$\Theta = m \cdot g \cdot r \cdot \frac{T_{osc}^2}{(4 * \pi^2)} \quad (5)$$

5.3 Simulation

By the simulation the inverted pendulum has been implemented using a *C-Block2* block in Scicos. The C-code of this block contains the non linear description of the inverse pendulum ([13], pages 245-247).

Fig. 11 shows the superblock of the pendulum used for the simulation. The subsection 5.5 shows the code of the *C-Block2* block.

5.4 Real-time controller

Fig. 13 shows the scicos block diagram of the implemented controller.

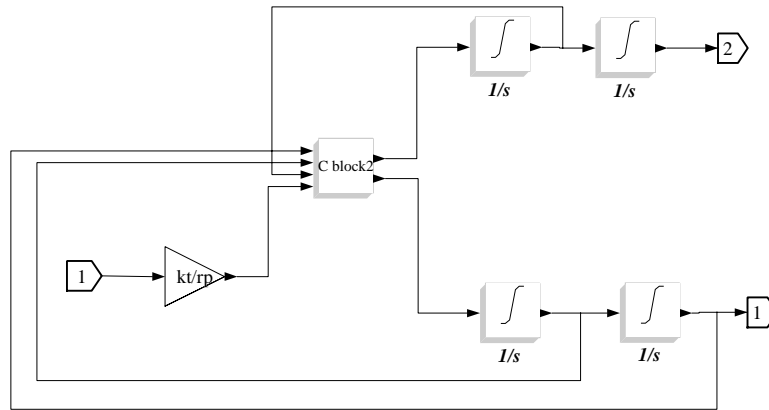


Fig. 11. Inverted pendulum - Block used for simulation

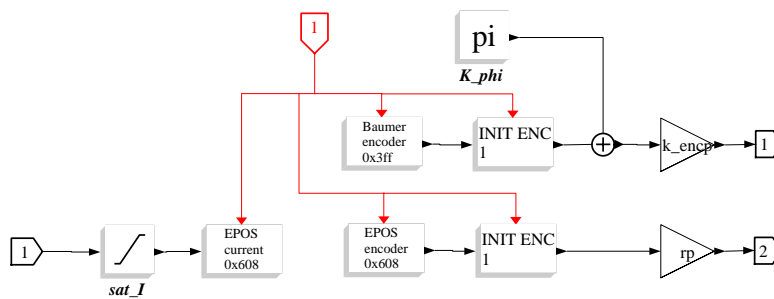


Fig. 12. Inverted pendulum - Block used for simulation (top) and for realtime controller (bottom)

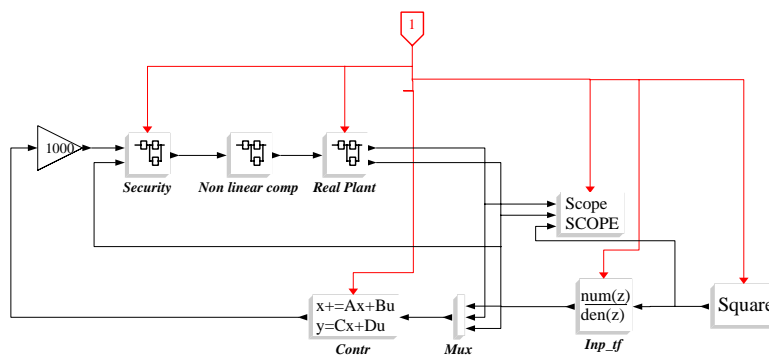


Fig. 13. Controller of the inverted pendulum

All the scilab commands needed to implement the controller are shown in the subsection 5.6. The state feedback controller has been implementing using a LQR approach. The function *redobs.sci* creates the 4 matrices for the reduced order observer. As a last step, the reduced order observer, the integral part and the state feedback gains are concentrated in a single state space system, in order to avoid any possible algebraic loop in the scicos block diagram. The block of the pendulum for the real controller is represented in Fig. 12. The *Init encoder* block fixes the position of the encoder after 1s as 0 position. A special block is represented by the *Security* block (see Fig. 14).

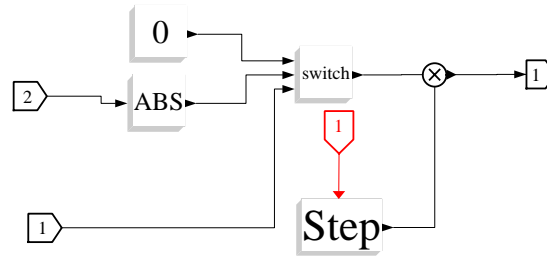


Fig. 14. Inverted pendulum - Security block

The "switch" block stops the controller if the car position exits from a given interval. The *step* block is used to start the controller after an initialization time.

5.5 C-code of the non linear model

```

/* Inverted pendulum - Nonlinear model */
#include <math.h>
#include <stdlib.h>
#include <scicos/scicos_block.h>
void inv_pend(scicos_block *block,int flag)
{
    double M,m,g,Theta,r,d,th,thd,xd,u,delta,thdd,zdd;

    M=block->rpar[0];
    m=block->rpar[1];
    r=block->rpar[2];
    Theta=block->rpar[3];
    d=1.2571;
    g=9.81;
    th=block->inptr[0][0];
    thd=block->inptr[1][0];
    xd=block->inptr[2][0];
    u=block->inptr[3][0];

    delta=Theta*(M+m)-m*m*r*r*pow(cos(th),2);
    thdd=m*r*((M+m)*g*sin(th)-m*r*thd*thd*sin(th)*cos(th)+(u-d*xd)*cos(th))/delta;
    zdd=(m*m*r*r*g*sin(th)*cos(th)-Theta*m*r*thd*thd*sin(th)+Theta*(u-d*xd))/delta;
    block->outptr[0][0]=zdd;
    block->outptr[1][0]=thdd;
}

```

5.6 Control design of the inverted pendulum

```

// Inverted pendulum - State feedback control design
M=0.62336
m=0.08377+0.10834;
g=9.81;
Tosc=29/21;

r=0.026+.297;
Theta=m*g*r*Tosc^2/(4*pi^2);
d=1.2165
kt=60.3e-3;
rp=0.6/20.91044;
k_encp = -1;

Ts=5e-3;

Imin=0; // For nonlinear compensation

A=[ 0, 1, 0, 0;
    m*r*(M+m)*g/(Theta*(M+m)-m*m*r*r), 0, 0, -m*r*d/(Theta*(M+m)-m*m*r*r);
    0, 0, 0, 1;
    m*m*r*r*g/(Theta*(M+m)-m*m*r*r), 0, 0, -Theta*d/(Theta*(M+m)-m*m*r*r)];
B=[0;
    m*r*kt/(rp*(Theta*(M+m)-m*m*r*r));
    0;
    kt*Theta/(rp*(Theta*(M+m)-m*m*r*r))];

C=[1 0 0 0; 0 0 1 0];
D=[0;0];

sys=syslin('c',A,B,C,D);

sysd=dscr(sys,Ts);
[Ad,Bd,Cd,Dd]=abcd(sysd);

// LQR discrete controller
// weights
Q=diag([250,1,200,1,250]);
R=[1];

[m1,n1]=size(Ad);
[m2,n2]=size(Bd);
[m3,n3]=size(Cd);

// Add suppl. state (integral) to SS form
Ad_f=[Ad,zeros(m1,1);-Cd(2,:)*Ts,1];
Bd_f=[Bd;zeros(1,n2)];
Cd_f=[Cd,zeros(m3,1);zeros(1,n3),1];
Dd_f=[Dd;zeros(1,n2)];

k_lqr=bb_dlqr(Ad_f,Bd_f,Q,R);

preg=max(abs(spec(A)));

// Reduced order observer
poli_oss=exp([-preg,-preg]*10*Ts);
T=[0,0,0,1;0,1,0,0];
[Ao,Bo,Co,Do]=redobs(Ad,Bd,Cd,Dd,T,poli_oss);

// Compact form
Contr=comp_form_i(Ao,Bo,Co,Do,Ts,k_lqr,[0,1]);

// Filetr for input square signal
g=syslin('c',10/(s+10));

```

```
gz=ss2tf(dscr(tf2ss(g),Ts));
Ainp=0.3
```

6 Conclusion

This paper presented three practical examples explaining some basic activities connected with control system design. The Scilab/Scicos/RTAI suite has demonstrated to have reached enough maturity and to be an interesting alternative to other expensive commercial suites. This tool can be used for data acquisition, analysis, identification, control design and simulation. Just with a few steps, the same scicos block diagram used for simulation can be transformed into a real-time controller running on a standard PC.

7 Some useful functions

7.1 Function *redobs.sci*

```
function [A_redobs,B_redobs,C_redobs,D_redobs]=redobs(A,B,C,D,T,poles)
P=[C;T]
invP=inv([C;T])

AA=P*A*invP

ny=size(C,1)
nx=size(A,1)
nu=size(B,2)

A11=AA(1:ny,1:ny)
A12=AA(1:ny,ny+1:nx)
A21=AA(ny+1:nx,1:ny)
A22=AA(ny+1:nx,ny+1:nx)

L1=ppol(A22',A12',poles)';

nn=nx-ny;

A_redobs=[-L1 eye(nn,nn)]*P*A*invP*[zeros(ny,nn); eye(nn,nn)];
B_redobs=[-L1 eye(nn,nn)]*[P*B P*A*invP*[eye(ny,ny);L1]]*[eye(nu,nu) zeros(nu,ny); -D, eye(ny,ny)];
C_redobs=invP*[zeros(ny,nx-ny);eye(nn,nn)];
D_redobs=invP*[zeros(ny,nu) eye(ny,ny);zeros(nx-ny,nu) L1]*[eye(nu,nu) zeros(nu,ny); -D, eye(ny,ny)];
```

7.2 Function *compform.sci*

```
function [Contr]=comp_form(A,B,C,D,Ts,K)
// Create the compact form of the Observer ABCD and the
// gain K,
//
// A,B,C,D: Observer matrices
// Ts: sampling time
// K: stte feedback gains

ss_sys=sslin('d',A,B,C,D);
ss_sys(7)=Ts;
g_sys=ss2tf(ss_sys);

gu=g_sys('num')(:,1)./g_sys('den')(:,1);
gy=g_sys('num')(:,2:$)./g_sys('den')(:,2:$);

Greg=[1/(1+K*gu),-K*gy/(1+K*gu)];
Contr=tf2ss(Greg);
```

7.3 Function *compform_i.sci*

```
function [Contr]=comp_form_i(A,B,C,D,Ts,K,Cy)
// Create the compact form of the Observer ABCD and the
// gain K, using an integrator at the input
// to eliminate the steady state error
//
// A,B,C,D: Observer matrices
// Ts: sampling time
// K: state feedback gains
// Cy: matrix to extract the output for the steady
// state feedback

[larg,rarg]=argn(0);

if rarg ~= 7 then
    Cy = [1]; // only 1 output
end

ss_sys=syslin('d',A,B,C,D);
ss_sys(7)=Ts;
g_sys=ss2tf(ss_sys);

g_int=syslin('d',Ts/(%z-1));
g_int(7)=Ts;

gu=g_sys('num')(:,1)./g_sys('den')(:,1);
gy=g_sys('num')(:,2:$)./g_sys('den')(:,2:$);

nn=size(K,2);

Ke = K(1,nn);
K = K(1,1:nn-1);

Greg=[-Ke*g_int/(1+K*gu), (Ke*Cy*g_int-K*gy)/(1+K*gu)];
Contr=tf2ss(Greg);
```

7.4 Function *xspectrum.sci*

```
function Txy=xspectrum(x,y)

nfft=int(size(x,'*')/2);
wind>window('hn',nfft)';
n=size(x,'*');
nwind=size(wind,'*');
index=1:nwind;
k=fix(n/nwind);

Pxx=zeros(nfft,1);
Pxy2=Pxx;Pxy=Pxx;

for i=1:k
    xw=wind.*detrend(x(index));
    yw=wind.*detrend(y(index));
    index=index+nwind;
    Xx=fft(xw(1:nfft));
    Yy=fft(yw(1:nfft));
    Xx2=abs(Xx).^2;
    Xy=Yy .* conj(Xx);
    Pxx=Pxx+Xx2;
    Pxy=Pxy+Xy;
end

if modulo(nfft,2)==1 then
    selct=(nfft+1)/2;
else
    selct=nfft/2+1;
```

```
end  
  
Pxx=Pxx(1:selct);  
Pxy=Pxy(1:selct);  
  
Txy = Pxy./Pxx;
```

References

- [1] D. Beal, E. Bianchi, L. Dozio, S. Hughes, P. Mantegazza, and S. Papacharalambous, "RTAI: real time applications interface," *Linux Journal*, April 2000.
- [2] E. Bianchi and L. Dozio, "Some experience in fast hard real-time control in user space with RTAI-LXRT," in *Real Time Linux Workshop*, Orlando, 2000.
- [3] R. Bucher and L. Dozio, "CACSD with Linux RTAI and RTAI-Lab," in *Real Time Linux Workshop*, Valencia, 2003.
- [4] R. Bucher, "Interfacing Linux RTAI with Scilab/Scicos," in *Real Time Linux Workshop*, Singapore, 2004.
- [5] R. Bucher, L. Dozio, and P. Mantegazza, "Rapid Control Prototyping with Scilab/Scicos and Linux RTAI," in *International Scilab Conference*, Paris, 2004.
- [6] R. Bucher and S. Balemi, "Scilab/Scicos and Linux RTAI - A unified approach," in *IEEE conference on Control Applications*, Toronto, 2005.
- [7] R. Bucher, "Targeting the Scicos Code Generator - The Linux RTAI Example," in *SCILAB Research, Development and Applications*, Wuhan, China, 2005.
- [8] R. Bucher, S. Mannori, and T. Netter. (2006) RTAI-Lab tutorial: Scilab, Comedi and real-time control. [Online]. Available: <https://www.rtai.org/RTAILAB/RTAI-Lab-tutorial.pdf>
- [9] COMEDI. Linux Control and Measurement Device Interface. [Online]. Available: <http://www.comedi.org>
- [10] S. Balemi, J. Moerschell, J.-M. Breguet, D. Braendlin, S. Bottinelli, and I. Beltrami, "Surface Inspection System for Industrial Applications," in *Conf. on Robotics and Mechatronics*, Aachen, Germany, Sept. 2004, pp. 1597–1602.
- [11] Peaks System. PCAN Linux website. [Online]. Available: <http://www.peak-system.com/linux/index.htm>
- [12] Maxon motor. Maxon motor website. [Online]. Available: <http://www.maxonmotor.com/>
- [13] S. Campbell, J.-P. Chancelier, and R. Nikoukhah, *Modeling and Simulation in Scilab/Scicos*. Springer, 2006.

Targeting the Scicos Code Generator HDL Model Example

Zhang Dong, Kang Cai

School of Physics & Electrical Information, Ningxia University, Ningxia, 750021, R.P.
China
Email:scicoshdl@gmail.com

Abstract. This paper shows how the original Scicos code generator has been modified to obtain a code for a specific HDL target, and how to design digital circuit in Scicos. And then a toolbox Scicos-HDL which integrates the hardware circuit, algorithm and Scilab/Scicos environment as a plat for digital circuit design, simulation and Hardware Description Language generation is introduced, we also give two examples to describe the design flow of Scicos-HDL.

Keywords: VHDL, Verilog, EDA, SCILAB/SCICOS, Hardware/Software co-design

1 Introduction

As the development of many kinds of EDA(Electronic Design Automatic) design tools, the design period time of digital circuit has been shortened, many tools can modeling, simulating ,synthesizing and validating the every process of electronic system design, but with higher-priced, impacting their applications. So it is of a huge meaning to develop an open source EDA Tools.

2 The Target

2.1 The predominance of Scicos as a plat for digital circuit design

Scilab is a scientific software package for numerical computations providing a powerful open computing environment for engineering and scientific applications developed by researchers from INRIA and ENPC since 1990. Its function and the form of language are close to Matlab, but emphasize particularly on control system CAD.

Scicos is a Scilab toolbox included in the Scilab Package: a dynamic systems modeler and simulator. Scicos has been successfully used in a number of large industrial applications as the Simulink of MATLAB.

To sum up, it has a large predominance to make Scicos environment as a plat for digital circuit design: 1) a Graphic User Interface (GUI) software and an algorithm-friendly development environment; 2) easily operating, simply link every module, no need to understand the internal principles; 3) Powerful computing function. 4) Open source software.

2.2 The solution in Scicos-HDL

Scicos-HDL is a tool to design digital circuit system; it can help the hardware engineers and other digital system engineers to design digital circuit faster and with lower cost. At present, Scicos-HDL has four libraries: sequential logic library, combinational logic library and IPcore blocks, it supports two kind of Hardware Description Language: VHDL&Verilog, and the Test Bench for them. You can design interface circuit of digital system, micro-digital system etc. With a good development we expect that Scicos-HDL can be as a Digital signal processing (DSP) system design tool in the future. It is a bridge between Digital Circuit and Scicos.

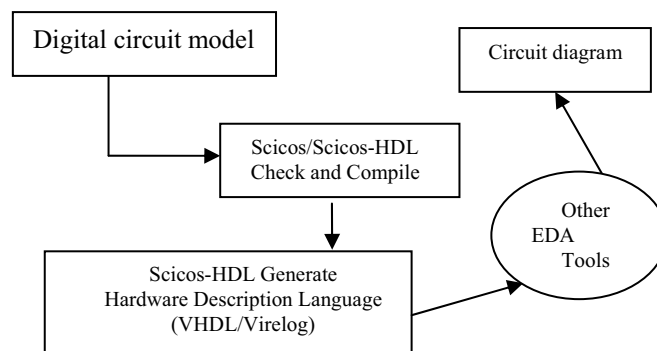


Figure 2.1 Digital circuit design flow in Scicos-HDL

The feature of Scilab-HDL:

- ◆ Scicos Hardware circuit model design develop toolbox
- ◆ Generate Hardware Description Language
- ◆ Support VHDL&Verilog language
- ◆ Visual digital circuit and the simulation library
- ◆ Support Test Bench of VHDL&Verilog and design documents
- ◆ It is free
- ◆ The LICENSE is under the Scilab 's LICENSE

- ◆ Support many OS: Windows XP, 2000, 98, Linux, UNIX, and Mac OS X.
- ◆ Provide the interface for users to add blocks
- ◆ LGPL

Note: The VHDL / Verilog codes generated by this compiler can be used in following EDA tools.

Synplify Pro 7.6, Quartus® II, Mux+plus II, ISE, Modelsim, etc.

2.3 Install Scicos-HDL

- A. Add following to the last of <scilab.star> file:
 - ◆ Load('SCI/macros/nxuEmb/libi')
 - ◆ Load('SCI/macros/scicos_blocks/nxuEmbc/libi')
 - ◆ exec('SCI/macros/scicos-HDL/global_define.sci');
 - ◆ getf('SCI+\'/macros/scicos-HDL/vhdl/vhdl.sci');
 - ◆ getf('SCI+\'/macros/scicos-HDL/verilog/verilog.sci');
 - ◆ write(%io(2),' scicos-hdl vision 0.3 inside !');
- B. following files or folders to appointed directory
 - ◆ Scicos-HDL block → computing function
 - a) Copy “nxuEmb” to 'SCI/macros' directory.
 - b) In “buildmacros.sce” file, add "nxuEmb" to “Directories “.
 - ◆ Scicos-HDL block → interface function
 - a) Copy “nxuEmbc” to 'SCI/macros/scicos_block' directory.
 - b) In “buildmacros.sce”, add "nxuEmbc" to “SubDirectoriesScicosBlocks ”.
 - ◆ Scicos-HDL block → palette
 - a) Copy the four files of “palette” directory to 'SCI/macros/scicos' directory.
 - ◆ Scicos-HDL → compiler
 - a) Copy “scicos-HDL” paper file to 'SCI/macros' directory.
- C. Install Scicos-HDL
Simply do, in Scilab/macros directory: **make**

3 The structure and design principles

How to make Scicos has the function of Hardware circuit model design, the solution is that we develop the digital circuit design blocks as the design libraries in Scicos, you can use the blocks in Scicos-HDL to create a hardware implementation of a system modeled in Scicos in short time. The Scicos-HDL Compiler block reads Scicos Model Files (.cos) that are built by Scicos-HDL and then the compiler will generate VHDL / Verilog code, hardware implementation, and simulation.

See the following figure:

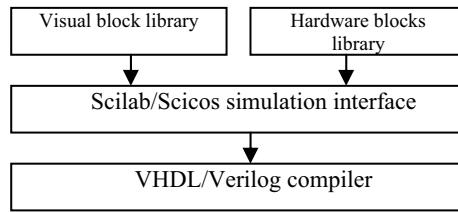


Figure 3.1 the internal principles of Scicos-HDL

We have developed 4 libraries for Scicos-HDL to design sequential logic circuit and combinational logic circuit, what innovative thought in Scicos-HDL is that we combine Scicos, Scicos-HDL with IPcore, this make Scicos and IPcore of EDA work together.

◆ Combinational logic library **EMBlibC**

This library has the basic components of digital system design, including Gate circuit, Multiplexer, Encoder, Decoder and BUS related components.

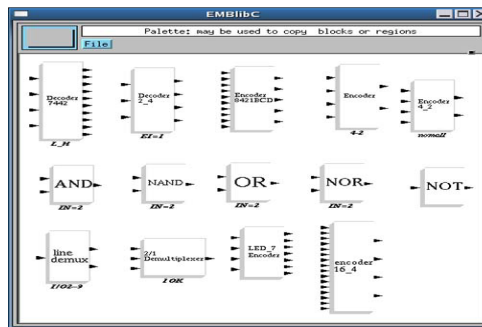


Figure3.2 Combinational logic library

◆ Sequential logic library **EMBlibT**

This library includes D trigger,JK trigger,RS trigger without CR port, and D trigger,JK trigger,RS trigger with CR port, 4-bit counter with self-set up.

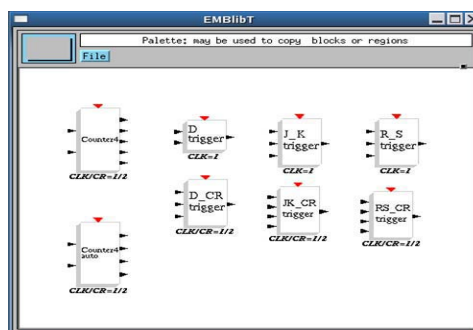


Figure 3.3 Sequential logic library

◆ Ipcore library **IPnxuEMB**

At present, we have Multiplexer, and decoder74138 and BUS related components.

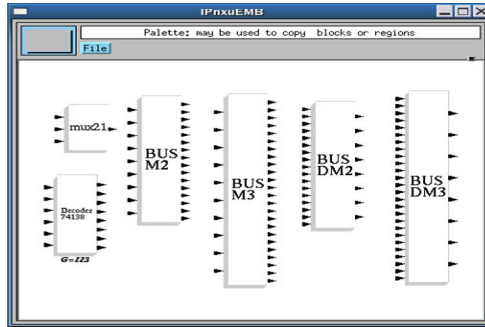


Figure 3.4 Ipcore library

◆ Digital system design basic library **EMBbaselib**

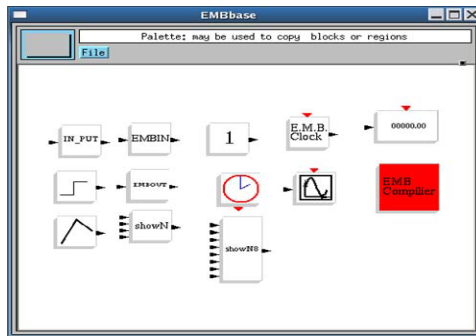


Figure 3.5 Digital system design basic library

NOTE:

1) EMBIN



This is the input port of E.M.B., the number of EMBIN is equal to the one of the whole system input ports, each in-signal must through this block, it must be in every model file.

2) EMBOUT



This is the output port of E.M.B., the number of EMBOUT is equal to the one of the whole system output ports, each out-signal must through this block, it must be in every model file.

3) showN



Binary bit → decimal bit converter

4) Scicos-HDLClock



E.M.B. clock, used in sequential logic circuit

5) input



E.M.B level input, high level “1” , low level “0”

◆ EMB Compiler

The compiler of Scicos-HDL--EMB Compiler, is the heart of Scicos-HDL, its main function is to convert Scilab/Scicos model file to standard **VHDL / Verilog** RTL code. And you can load these **VHDL / Verilog** code to FPGA.

(I) How to operate the compiler



It must be at every model file, when the whole system is finished, click this block to start compiler, then follow the dialog box

(II) NOTE

- Make sure you have put the block **EMBIN** as **the input ports**, signals are transferred in through this block;
- Make sure you have put the block **EMBOUT** as **the output ports**, signals are transferred out through this block;
- Self-connected is not allowed in every block;
- If one output port is needed to be connected with many other blocks ,use the block **LineDemux of EMBlibC** library;
- Make sure the path and name of model file are correct as the rule in Scicos;
- Make sure the directory of saving **VHDL / Verilog** code file is correct, empty or at least without the same name to the model file you will compile;
- Make sure no super block in the model file.

The **VHDL / Verilog** codes generated by this compiler can be used in following EDA tools. **Synplify Pro 7.6, Quartus® II, Mux+plus II ,ISE, Modelsim, etc.**

4 Applications

4.1 4-bit shift register

Register is a logic component as a memorizer in computer and other digital system, the main element of it is trigger, every trigger can store one binary bit, so N bits binary code need the same number of triggers.

This is a 4 bits shift register, it is composed of 4 triggers, and we use the fall edge of the clock, figure 4.1

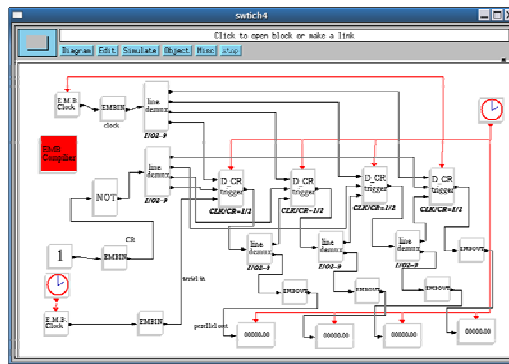


Figure 4.1 4 bits shift register

Synthesized the VHDL / Verilog codes in Synplify Pro 7.6

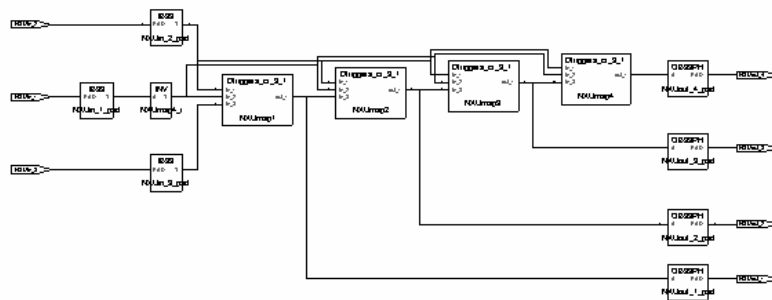


Figure 4.2 Synthesized in Synplify Pro 7.6

```
VHDL CODE:
library ieee;
use IEEE.std_logic_1164.all;
use IEEE.std_logic_arith.all;
use IEEE.std_logic_unsigned.all;
entity switch is
port(
```

```

        NXUin_1: in std_logic;
        NXUin_2: in std_logic;
        NXUin_3: in std_logic;
        NXUout_1: out std_logic;
        NXUout_2: out std_logic;
        NXUout_3: out std_logic;
        NXUout_4: out std_logic);
end swith;
architecture NXU_beh of swith is
component Dtriggers_cr_3_1
port(
    in_1,in_2,in_3: in std_logic;
    out_1: out std_logic);
end component;
component notgate_1_1
port(
    in_1: in std_logic;
    out_1: out std_logic);
end component;
component LineDemux_1_4
port(
    in_1: in std_logic;
    out_1,out_2,out_3,out_4: out std_logic);
end component;
component LineDemux_1_2
port(
    in_1: in std_logic;
    out_1,out_2: out std_logic);
end component;

```

4.2 Peripheral circuit of single chip

Scicos-HDL can help hardware engineer, especially the single chip engineer design the peripheral circuit of single chip or other digital circuit. When some digital systems are high speed system and need the high interface circuit, then FPGA is first choice, so what we do just use the blocks of Scicos-HDL to finish the interface circuit design, Scicos-HDL compiler will convert the hardware design to standard VHDL / Verilog, then load them to FPGA. It can shorten the design cycle for a project; what's more better, it can help some engineers although they are not familiar with VHDL / Verilog use FPGA.

This is a peripheral circuit for a single Chip; it has an A/D interface, LED interface, key board interface, D/A interface, with a flip-latch integrated, decoders and encoders, designed by Scicos-HDL

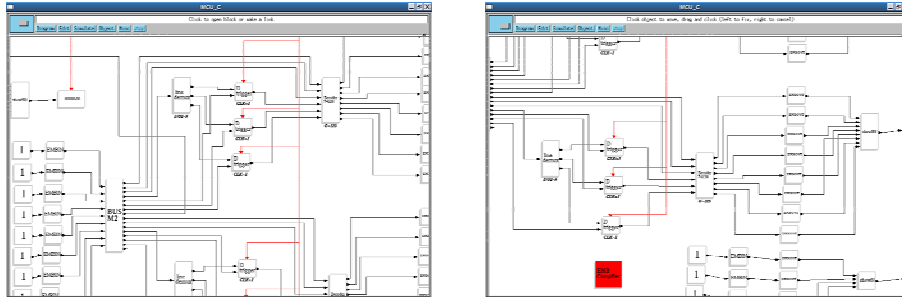


Figure 4.3 peripheral circuits for a single chip

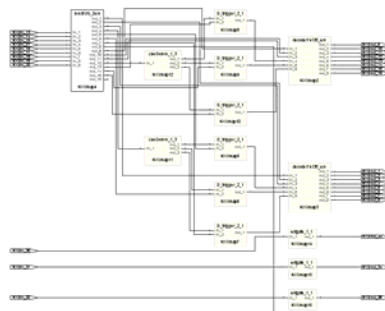


Figure 4.4 Synthesized in Synplify Pro 7.6

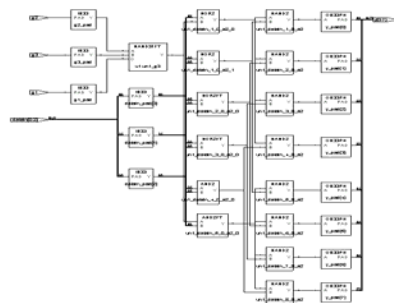


Figure 4.5 Synthesized in Synplify Pro 7.6

Figure 4.4 is the **VHDL / Verilog** code of the interface circuit;
 Figure 4.4 is the VHDL / Verilog code of one block named decoder74138 of the interface circuit.

```

Verilog CODE
module decoder74138(g1,g2,g3,datain,y);
    input g1,g2,g3;
    
```

```

input [2:0] datain;
output [7:0] y;
reg [7:0] y;
always @(g1 or g2 or g3)
begin
    if (g1==1'b0)
        y=8'b11111111;
    else if (g2==1'b1)
        y=8'b11111111;
    else if (g3==1'b1)
        y=8'b11111111;
    else
        case (datain)
            3'b000: y=8'b01111111;
            3'b001: y=8'b10111111;
            3'b010: y=8'b11011111;
            3'b011: y=8'b11101111;
            3'b100: y=8'b11110111;
            3'b101: y=8'b11111011;
            3'b110: y=8'b11111101;
            3'b111: y=8'b11111110;
            default: y=8'bxxxxxxx;
        endcase
    end
endmodule

```

5 Conclusion

We introduce a Digital System Design tool based on Scicos, Scicos-HDL which makes Scicos has hardware design and simulation function. As open-resource software, Scicos-HDL has a unique meaning in the filed of open-resource EDA tools. We hope it contribute to our China's IC design.

At present Scicos-HDL is free. You can download Scicos-HDL vision 0.3.

Download address: <https://sourceforge.net/projects/scicoshdl>

Welcome to contact us: ScicosHDL@Gmail.Com

References

1. Scilab, INRIA, <http://www-rocq.inria.fr/scilab>.
2. Fang Yu, Huang Jianwen., "Application of Scilab/Scicos in Hardware/Software Co-design", Computer Engineering Vol.28 June 2002, pp.227-229.
3. Hu BaoGang, ZhaoXing, KangMengzhen, "Introduction to Scientific Computing Software", Tsinghua University Press 2003.1, pp.165-203.
4. SjöholmS, LindhL, VHDL for designers.[M].BianJinian, XueHongxitransl. Beijing: TsinghuaUniversityPress, 2000.(in Chinese).
5. Synplify Pro <http://www.Synplicity.com/support>.

The Present and Future of Scilab's Engineering Application

Zhe Peng, Longhua Ma

Institute of Industrial Control of Zhejiang University

310027 Hangzhou, China

[Pengzhe1113@gmail.com](mailto:pengzhe1113@gmail.com) , lhma@iipc.zju.edu.cn

Abstract. The paper mainly describes the present and future of Scilab's engineering application. In the past, Scilab only can used in discrete and off-line control in laboratory because of not ability of interface function. Nowadays Scilab can exchange data with device layer and Ethernet layer convenience due to the interface toolbox. It transforms Scilab from discrete and off-line control to continuous and on-line control. ESP and A2/O are taken for example to describe the modeling and optimization with Scilab. The future of Scilab's engineering application is prospected in the end.

Key words: Scilab, engineering, application, continuous, on-line

1 Introduction

Developed at INRIA, Scilab as scientific compute software is freely distributed in source code format. Nowadays, Scilab is widely used in much area, Such as simulation, parallel compute, modeling, optimization, video and image process, signal process, fuzzy network, and so on.

2 Scilab's present engineering application

In modeling and simulation area, Scicos is a block diagram graphical editor for the construction and simulation of dynamical systems. Scilab/Scicos is the only

open-source alternative to commercial products such as Matlab/simulink and MATRIXx/SystemBuild [1]. The common Scicos diagram is shown as fig.1.

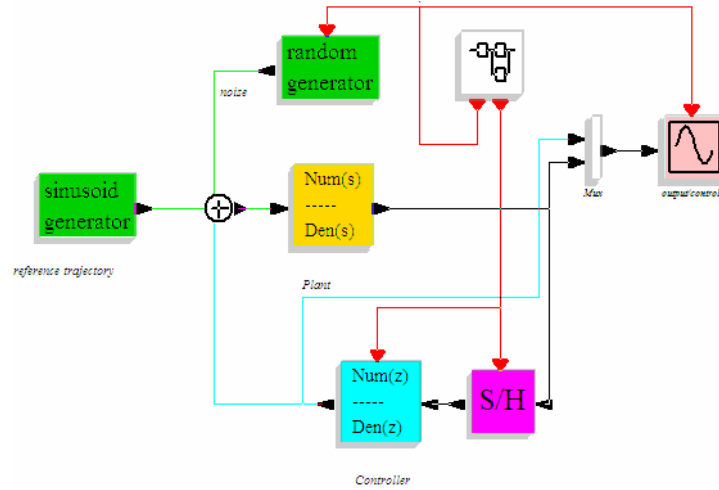


Fig. 1. Common scicos diagram

In control area, Scilab is competent for much including PID, LQG, Robust control, Fuzzy control, Genetic Algorithm, Neural networks, etc. Take PID simulation for example, you can easily use Scilab PID toolbox for PID simulation shown as fig2.

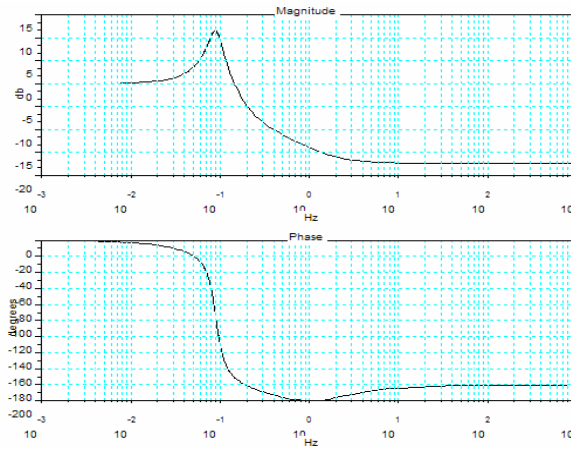


Fig. 2. PID simulation result

In optimization aspect, you can use Lmitool, Multiflow problem etc. toolbox to solve problem. If your problem cost too much time and resource to solve, you can consider using parallel compute method. In the field of scientific computing, the limit

of the developing speed of the hardware can be counterbalanced by the parallel computing software. Scilab also support parallel compute.

Also Scilab can perfect solve other problem in image, video, signal process area and so on.

Anyway, you can easily found that Scilab mainly used in laboratory for simulation optimization and so on, not really used in factory. There is much cause for it, such as, the not enough ability of exchange data, the not sufficient of toolbox, the interface of Scilab is not widely.

3 Transform from discrete and off-line control to continuous and on-line control

To transform control from discrete and off-line to continuous and on-line, the core is realizing exchange data between factory and Scilab. And in arithmetic aspect, we should prefer time cost than precision.

The first and important step of transform control from discrete and off-line control to continuous and on-line control is acquisition data from factory. In past, Scilab doesn't support acquisition data through serial and Ethernet, also Scilab's exchange with other software ability is very infirmness. The important of acquisition data from factory is shown as fig3.

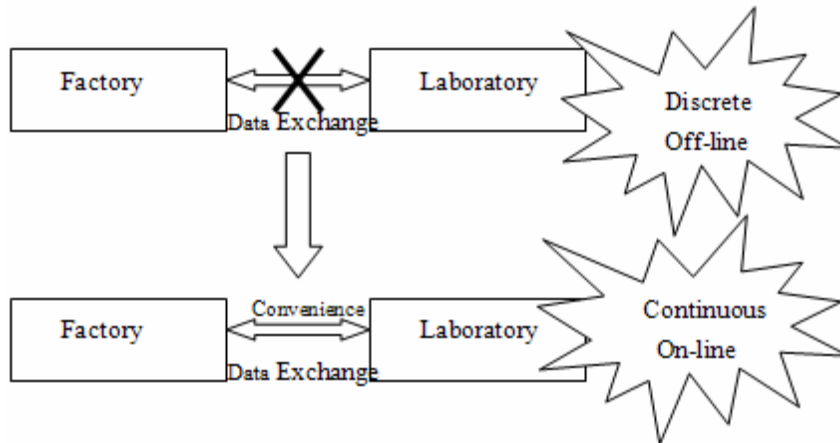


Fig. 3. Change from discrete and off-line control to continuous and on-line control

Different factory or device may be support different method of data acquisition. But nowadays, there are much data exchange specification in control area, such as OPC, Field bus, Industrial Ethernet, etc. Scilab can communicate with most devices if Scilab supports these interfaces. Also if Scilab can offer a toolkit for hardware and software, Scilab's user can communicate with new device.

4 What we have done

4.1 The software frame we have designed

In the continuous and on-line system, Scilab is the core. It can execute control and optimization due to its strength scientific compute ability. Data can be exchanged between Scilab and much device such as PLC, Sensor, Motor, etc. Also Scilab can storage data to database. And interface with other software should be developed.

The continuous and on-line syetem frame can be divided into 3 layers: Device Layer, Control Layer, and Ethernet Layer, which is shown as fig4.

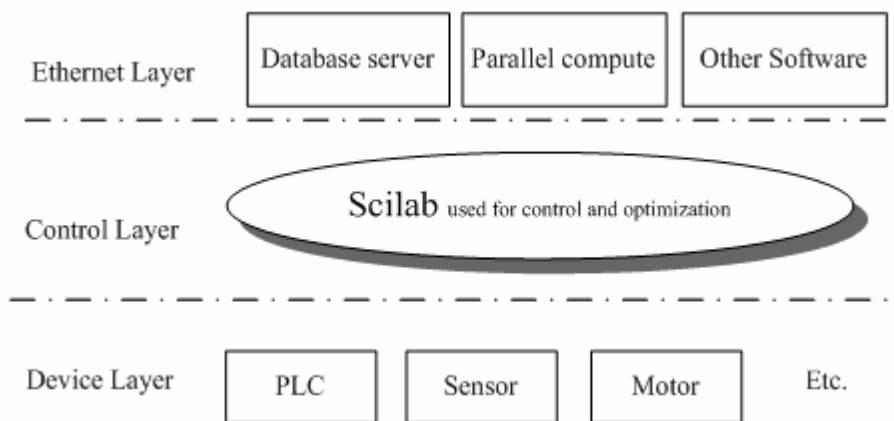


Fig. 4. Continuous and on-line system framework

With TCP/IP protocol, Scilab can storage data to remote database server , and realizing parallel compute, also can exchange data with other software if need.

4.2 Data acquisition from factory

Scilab can be easily interfaced with C programs. This allows the use of standardized packages and libraries in the interpreted environment of Scilab. Also it is useful to have to use specific numerical code for, e.g., the simulation or optimization of user defined systems, or some interface modules. In fact, interfacing numerical code appears necessary in most nontrivial applications [2].

In the system we developed OPC, Serial, Modbus, and Ethernet interface to communicate with device layer.

Take Modbus interface for example. Modbus protocol is made by Modicon Corporation. With Modbus protocol, you can exchange data with device which supporting Modbus protocol easily. So now Scilab also can communicate with device supporting Modbus protocol due to the toolbox. The configure interface is shown as fig5.

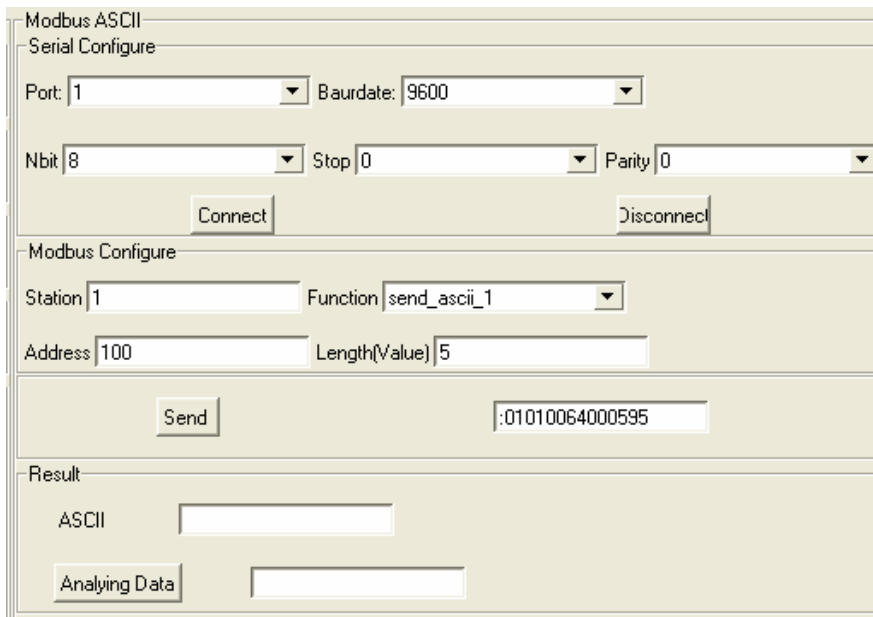


Fig. 5. The Modbus configuring interface

4.3 Modeling and optimization

Now with the Scilab's feature – enabling acquisition data from device, we can model and optimization base on it.

Take Electrostatic precipitator modeling and Improved Model of Secondary Clarifier and A2/O Process Modeling for example.

The electrostatic precipitator (ESP) is a commonly used device for removing gas-particulates mixtures exhausted by various industrial processes in so many different domains. According the mechanism and modeling of ESP, we design a ESP simulation platform which is based on Scilab. We enter the values of all needed variables, and do the computation by Scilab, we get the results and the figures help us to do analysis easily such as fig7, which is the distribution of the electric field from three-dimensional view.

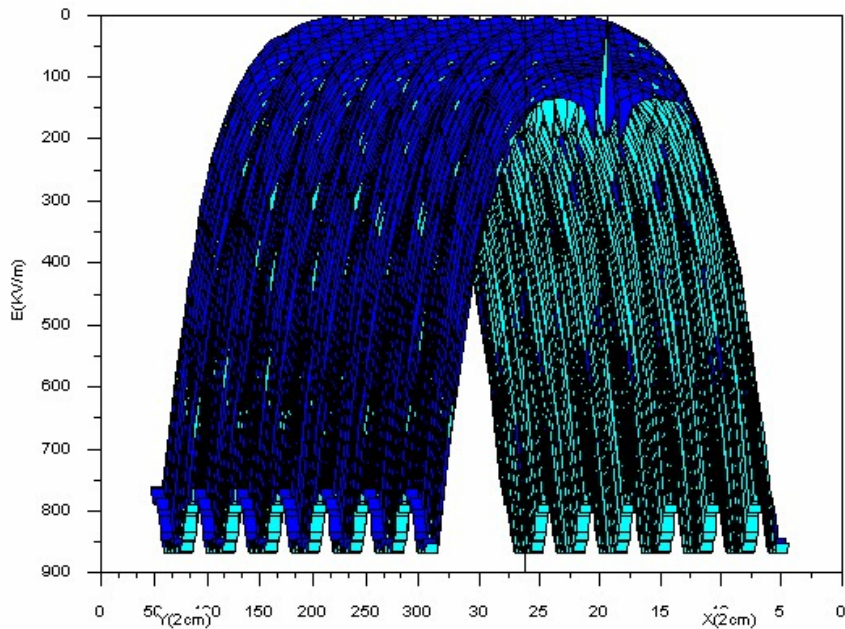


Fig. 6. Three-dimensional view of electric field distribution

Activated Sludge Process (ASP) is the most commonly used biological wastewater treatment process. It is considered to be the most cost-effective way to remove the organic materials from wastewater. Besides that, it is very flexible, reliable, and has the capability of producing high quality effluent. But the design and operation of ASP depend on experience highly. So establishing the activated sludge mechanism mathematic model is necessary. Also we do much work in Improved

Model of Secondary Clarifier and A2/O Process Modeling. Due to the limit of the paper length, the particular is omitted.

4.4 The future

In the future, Scilab will offer toolbox for data exchange, which supporting most of nowadays industrial protocol, with our great effort. And the core of Scilab should be modified to be real-time for real-time control. And in data monitor and alarm function aspect, should be realized through TCL script language.

5 Conclusion

In the past, Scilab only can used in discrete and off-line control in laboratory. Through the interface toolbox, Scilab can exchange data with device layer and Ethernet layer convenience. It transforms Scilab from discrete and off-line control to continuous and on-line control. And the paper takes ESP and A2/O for example to describe the modeling and optimization with Scilab. In the end, the future of Scilab's engineering application is prospected.

Acknowledgments. This work is supported by a grant from the National Natural Science Foundation of China (No. 60474064) and the Fund of produce-study-research Science and Technology Cooperation of Hangzhou City to Zhou LiFang and Ma LongHua.

References

1. M. Najafi, S. Furic, R. Nikoukhah, SCICOS: a general purpose modeling and simulation environment, Proc. the 4th International Modelica Conference, 2005
2. Scilab Group, *INTRODUCTION TO SCILAB*
3. Lydia E. van Dijk and Christoph L. Spiel, *Scilab Bag Of Tricks: The Scilab-2.5 IAQ*, 2000
4. Baogang Hu, Xing Zhao and Mengzhen Kang, *Introduction to Scientific Computing Software- SCILAB*, Tsinghua University Press, Beijing, 2003.

Tools for building Scilab toolboxes, present and future

Farid Belhacene and Serge Steer

INRIA, BP 153, 78153 le Chesnay CEDEX, France
Farid.Belhacene@inria.fr, Serge.Steer@inria.fr

Abstract. This article first describes how to build a new Scilab toolbox and the available tools¹ which can ease this work and make the toolbox platform independent, easy to use and well integrated. Finally we present new ideas to ease extern C or FORTRAN code use in Scilab.

1 Introduction

Scilab toolboxes are generally built over a set of Scilab functions. Some of these functions are written in Scilab language and some other build over external procedures written in C, C++, FORTRAN, Java ...). Building a toolbox from this material consist in giving users all the elements for loading these functions in Scilab and having them fully integrated into the Scilab environment with their documentation.

External codes have to be interfaced with Scilab. The interface is a piece of code which allows calling the code through Scilab function syntax. The interface code have to check the correctness of the arguments given when calling the Scilab function (number, types and dimensions of the Scilab variables), then it calls the external procedure to produce the expected result and finally it stores the results in Scilab variables. In the sequel we will present the currents tools for writing these interface and some new ideas for automatic interface generation.

The toolbox loading process should be, as far as possible, made independent of the operating system and as simple as possible to use for the end user. The connections with the operating system are mainly the file path definition and the handling of external source codes (compilation and incremental linking with Scilab). Tools for handling these aspects are also presented above.

It is recommended to organize all the toolbox files under a common directory. This directory should also contain two Scilab script files. The `builder.sce` file will be called once, at the toolbox installation time, to perform all the tasks requested to build the toolbox from its sources and allow an easy plugging of the toolbox in a running Scilab session by the `loader.sce` script. Some convenient tools are included into

¹ Due to the length of this article, only the main tools are presented without entering in the details which can be found in the Scilab help or documentation.

the Scilab distribution to perform the tasks that the builder and loader files commonly have to do.

2 Basic Tools

First we present the most common tools that can be used for all types of Scilab toolboxes

2.1 Building and loading Scilab functions

The functions written in Scilab language of a toolbox should be organized in at least one directory. Each main function should be stored in a file with *.sci* extension and whose name is the name of the function, subsidiary functions can also be defined here. According to these rules the Scilab `genlib(<lib_name>, <lib_path>)` instruction can be used in the `builder.sce` file to generate (or update) the pseudo-code files (with extension *.bin*) and build the Scilab *library* variable `<lib_name>` which is saved in the directory under the name `lib`. It is then enough that the `loader.sce` script loads this file (`load <lib_path>/lib`) into Scilab to make it know all the functions defined in the library.

2.2 Building and loading on line help

The simplest way of providing on line help information for Scilab function is to add comment lines just after the first line of the function. By default the Scilab on line help manager extract these comment lines up to the first non comment one and display them when requested. But this method does not allow hypertext navigation nor apropos search nor localization.

For a full integration in the Scilab help system the on line helps have to be written in XML files following the Scilab help DTD which is given in `<SCIDIR>/man/manrev.dtd` file. The XML syntax is quite heavy, fortunately the Scilab function `help_skeleton` can be used to generate all the XML stuff for a given function. The programmer has only to edit the generated file to enter the relevant information. XML file editors like `xxe2` can also be used to write these XML files in very convenient way.

It is recommended to collect these XML files in a sub-directory of the toolbox hierarchy. Once XML files have been written the corresponding HTML file can be generated by the `builder.sce` script using the `xmltohtml` Scilab function (`xmltohtml help_subdir`).

The instruction `add_help_chapter(<Title>, <Help_dir_path>)` can then be added to the `loader.sce` script.

² www.xmlmind.com/xmleditor

2.3 Building and loading demos

The on line help DTD includes an example section, where the developer can put Scilab instruction illustrating the function use, but it is often convenient to add some global demonstration of the toolbox. The demo files can be Scilab script files stored in a demos sub-directory and a demo driver script based on a demo selection dialog. Adding demos can then be done easily in the loader.sce file adding the instruction: `add_demo(<Title demo>, <driver_script_path>)`. This way the toolbox demos will appear in the Scilab demos menu.

3 Interfacing external procedures with Scilab

Using external codes written in C or FORTRAN is more demanding. The developer has first to write an interface between the external procedure. The interface is a piece of code which allows calling the code through Scilab function syntax. The interface code first checks the correctness of the arguments given when calling the Scilab function (number, types and dimensions of the Scilab variables), it allocates memory in the Scilab stack for the results, then it calls the external procedure to produce them and finally it stores the results in Scilab variables. This interface can be coded either in Scilab language either in C.

3.1 Scilab coded interface and call function

This interfacing method is based on the Scilab function `call` which is able to execute a FORTRAN or C procedure provided that all the arguments of the procedure are pointers on double, float or int arrays or scalars. As an example, suppose that one wants to interface the following C procedure with compute the sum of two vectors:

```
void vsum(int *n, double *a, double *b, double *r)
{
    int i;
    for (i=0;i<*n;i++) r[i]=a[i]+b[i];
}
```

If A and B are two Scilab vectors of real floating point numbers with the same size the C procedure can be called within Scilab by the instruction:

```
R=call("vsum", size(A,"*"), 1, "i", A, 2, "d", B, 3,...
      "out", size(A), 4, "d")
```

The `call` arguments contains: the name of the procedure to be called, a sequence of triples (<value>, <position_in_the_calling_sequence>, <type_of_data>) associated with the inputs. The input argument list is closed by the string "out" which is followed by a sequence of triples associated with the left hand side arguments of call: (<array_sizes>, <position_in_the_calling_sequence>, <type_of_data>)

To associate the Scilab function `Vsum` to the `vsum` procedure one then need to write `Vsum` function as follow:

```

function R=Vsum(A,B)
  if type(A)~=1|type(B)~=1 then
    error("Invalid argument types")
  end
  if or(size(A)~=size(B)) then
    error("Argument dimensions should be the same")
  end
  if ~isreal(A)|~isreal(B) then
    error("Arguments should be real")
  end

  R=call("vsum", size(A,"*"), 1, "i", A, 2, "d", B, 3,...
        "out", size(A), 4, "d")
endfunction

```

Once these codes are written, the `ilib_for_link` Scilab function can be called by the `builder.sce` file to compile the C code and build a shared library. This function has been designed to be platform independent but it obviously requires a C compiler installed on the computer! The execution of `ilib_for_link` also produces a Scilab script file which can be used, within the main `loader.sce` file, to perform the incremental linking of the `vsum` procedure into Scilab. Incrementally linked functions can be passed as argument to some special function like `call`, `ode`, `optim`, ... using their name here the `"vsum"` argument of `call`.

Using the `call` function to interface external codes is easy to use but it has limitations mainly relative to the procedures that can be called. In particular it is not possible to pass string array arguments, pointers, ... Moreover this solution should be inefficient because of the interpreted Scilab code used for testing, in particular when the cost of the C function execution is low.

3.2 C coded interfaces

This method is much more flexible and efficient, but it requires the interface code being written in C. When the interface is not too complicated, the interface code can be automatically generated by the `intersci-n` Scilab tool given a formal description file. The C interface of the example above can be described by the file `Vsum.desc`:

```

Vsum      a      b
a          vector n
b          vector n
r          vector n

vsum      n a      b      r
n          integer
a          double
b          double
r          double

out       sequence r
*****

```

Given this description file the `intersci-n` tool produces the interface C procedure in file `Vsum.c`:


```
#include "stack-c.h"
int intsVsum(char *fname)
{
    int m1,n1,l1,mn1,m2,n2,l2,mn2,un=1,mn3,l3;
    CheckRhs(2,2); /*Number of input arguments checking*/
    CheckLhs(1,1); /*Number of output arguments checking*/
    /* checking input variable a (first argument) */
    GetRhsVar(1,"d",&m1,&n1,&l1);
    CheckVector(1,m1,n1);
    mn1=m1*n1;
    /* checking input variable b (second argument) */
    GetRhsVar(2,"d",&m2,&n2,&l2);
    CheckVector(2,m2,n2);
    mn2=m2*n2;
    /* cross variable size checking */
    CheckDimProp(1,2,m1*n1 != m2*n2);
    /* creation of a variable to store result */
    CreateVar(3,"d", (un=1,&un), (mn3=mn1,&mn3),&l3);
    /* procedure call */
    vsum(&mn1,stk(l1),stk(l2),stk(l3));
    /* set returned variables */
    LhsVar(1)= 3;
    return 0;
}
```

This interface can be built using the Scilab function `ilib_build` called by the instructions:

```
//<Scilab_name>, <interface_name> table: one row per
function
table =["Vsum","intsVsum"];
objectfiles = ["vsum.o", "Vsum.o"];
otherlibs=[];
ilib_build('libVsum',table,objectfiles,otherlibs);
```

The `ilib_build` function will produce a gateway for all the function that are described in the `.desc` file (here just one), a shared library named `libVsum.dll` (or `.so`) with the requested object files and libraries and a Scilab script for shared library loading.

As shown above the generated interface code is based on a library of procedures like `CheckRhs`, `CheckLhs`, `GetRhsVar`, `CreateVar` ... In case where the interface cannot be described with the `intersci` formalism, the developer can write the interface manually using the same library. Many examples are included in the Scilab distribution (see `<SCIDIR>/examples/interface-tour-so`) but this method is not error prone while coding the interface and requires C programming knowledge.

4 Future tools for interfacing external procedures with Scilab

C code Interface generators like `swig`[1], `MEXGEN`[2], `Matwrap`[3] are available but they do not fulfill all the objectives of Scilab interfaces.

Taking profit of the experience with the existing tools we started thinking to a new interface generator based on an extension of the current `call` function which can support all data types and can work by reference to avoid copy of data. To be able to design interface using Scilab language it will be necessary to create new Scilab data type and functions for pointer handling.

As an example, suppose one want to define a Scilab interface for the following C procedure which returns a point on an array containing the index of positive values within a given vector.:

```
void findpositive(int n, double *v, int *np, int *ptr)
{
    int i,k;
    *np=0;
    for (i=0;i<n;i++) if (v[i]>0) *np++;
    k=0;
    for (i=0;i<n;i++) if (v[i]>0) {*ptr[k]=i;k++;}
}
```

Sample of what should be the Scilab interface:

```
function l=findpositive(v)
    if type(v)~=1|~isreal(v) then
        error("Invalid argument"),
    end
    nv=size(v,'*')
    v_ptr=ref("v") // get pointer on the variable v
    np=0; // allocate np number of positive entries
    np_ptr=ref("np") // get pointer on the variable np
    h=handle()
    extproc("findpositive",nv,v_ptr.data,np.data,h)
    y=zeros(1,np)//memory allocation for the result
    y_ptr=ref("y") // get pointer on the variable y
    copy(n,h,y_ptr.data) //copy the index array
    free(h)
endfunction
```

Once the Scilab code interface is validated in the interpreted mode, the Scilab interface function should be automatically translated to a C code interface, using the Scilab to C translator which will be developed in the hArtes FP6 European project.

5 Conclusion

The tools currently available for building, easy to use, well integrated, toolboxes for Scilab have been summarized. The limits of these tools have been pointed out and analyzed to produce ideas for future evolutions.

References

1. <http://www.swig.org/>
2. <http://svi.cps.utexas.edu/mexgen.htm>

3. <http://lnc.usc.edu/~holt/matwrap>

Design and Development of Toolbox for Scilab

A Complex Network Toolbox based on Scilab

Bo Wu¹ Wei Chen² Gang Yan¹

¹ Department of Electrical Science and Technology, University of Science and Technology of China, Hefei, Anhui, 230026, P. R. China.

ninja@mail.ustc.edu.cn,

² Department of Electronic Engineering and Information Science, University of Science and Technology of China, Hefei, Anhui, 230026, P. R. China.

Abstract. The discovery of small-world and scale-free properties of many natural and artificial complex networks has in recent years stimulated a great deal of interest in studying the underlying organizing principles of various complex networks, which has led to dramatic advances in this emerging and active field of research. Here we create one visual project using the scilab to analyze the topological properties of complex networks including average degree, community value, degree-degree correlation, betweenness, clustering coefficient, degree distribution etc. Moreover, we provide the dynamic method in order to make it convenient to investing the dynamical processes on the networks such as traffic flow dynamics, synchronization, evolutionary game, epidemic dynamics and so on.

Key words: complex networks, small-world, scale-free, scilab, toolbox

1 Introduction

The term “complex network” (*see Figure 1*) refers to a graph that has non-trivial topological structure. Most real networks are complex on account of their interesting features that do not exist in simple networks. Examples include the the World Wide Web, social networks of acquaintance or other connections between individuals, organizational networks and networks of business relations between companies, neural networks, metabolic networks, food webs, distribution networks such as blood vessels or postal delivery routes, networks of citations between papers, and many others.

The study of network based on the classic graph theory ago. For the last decade, however, the science of complex networks has focused on describing the structural complexity of the real network [1–4]. In this Scilab project, we provide some parameter functions to compute the fundamental status of models, dynamics model functions to observe the whole process of evolution and some interface functions for users that they could import their own data from their original program generated by C, C++, Fortran etc. We also establish similar rules like *Pajek scripts* [5] in which people could combine different models more effectively.

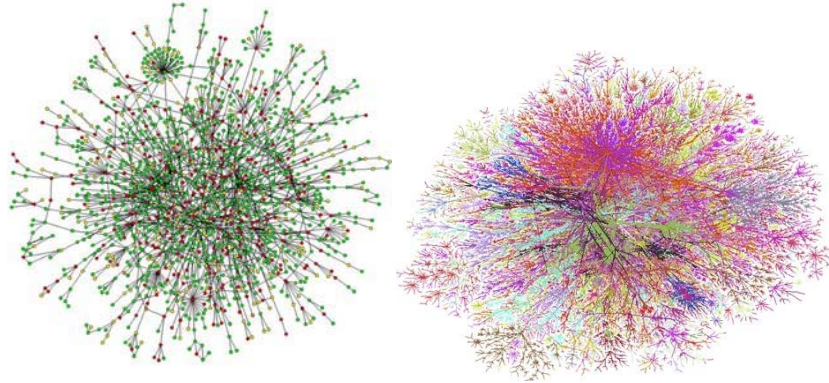


Fig. 1. Two examples of real-world complex networks: (left) cell metabolic network; (right) Internet topology (router level).

2 Frame of The Toolbox

The original graph toolbox *Metanet* [27] provides some basic and simple functions to characteristics such as neighbors of some node, the shortest path between some two nodes and so on. However there are much more topological properties in the field of complex networks. Furthermore, it's essential and meaningful to provide some functions to analyze the frequently researched dynamical processes taking place on the complex networks. In the following subsections, we show the topological and dynamical characteristics of complex networks and the corresponding functions in our project based on Scilab.

2.1 Different Parameters

1. Degree Distribution:

One fundamental characteristic of complex networks is the degree distribution $p(k)$, which represents the fraction of the number of nodes that have k neighbors in the network. Equivalently, $p(k)$ is the probability that a node chosen uniformly at random has degree k . A plot of $p(k)$ for any given network can be formed by making a histogram of the degrees of vertices. This histogram is the degree distribution for the network. (see *Figure 2*)

2. Assortativity Coefficient and Degree Correlations:

In most kinds of networks there are at least a few different types of vertices, and the probabilities of connection between vertices often depends on types. Many networks show “assortative mixing” on their degrees, i.e., a preference for high-degree nodes to attach to other high-degree nodes, while others show disassortative mixing – high-degree vertices attach to low-degree ones. The Assortativity coefficient $-1 < r < 1$, given by Newman [6], represents the very properties of complex

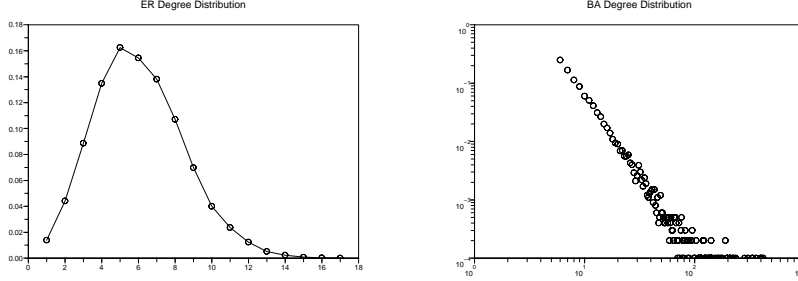


Fig. 2. (left) ER degree distribution generated by *Scilab* (node number=10000), the distribution function $P_k \sim e^{-\langle k \rangle} \frac{\langle k \rangle^k}{k!}$ which is approximated by a Poisson distribution. (right) BA degree distribution generated by *Scilab* (node number=10000), the distribution function $P_k \sim k^{-\alpha}$, α is about 3.

networks.

$$r = \frac{M^{-1} \sum_i j_i k_i - (M^{-1} \sum_i \frac{1}{2}(j_i + k_i))^2}{M^{-1} \sum_i \frac{1}{2}(j_i^2 + k_i^2) - (M^{-1} \sum_i \frac{1}{2}(j_i + k_i))^2} \quad (1)$$

where j_i, k_i are the degrees of the nodes at the ends of the i th edge, with $i = 1, \dots, M$.

Another representation of degree correlation is a two-dimensional histogram, proposed by Pastor-Satorras *et. al* [7], which shows the mean degree of the network neighbors of a node as a function of the degree k of that node.

3. Community Structure and Rich-Club Phenomenon:

Empirical studies have revealed that most real-world networks show “community structure”, i.e., groups of nodes that have a high density of edges within them, with a lower density of edges between groups [8]. For example, communities in an airport network might represent related airlines in one nation or region.

In the toolbox, some functions could be used for creating the community structures. Interestingly the latest article indicated that although in an absolutely random graph, if we select some strategies to recognize these communities, links between them also represent a power-law statistic result, for example, this will help people to choose different search strategies in different levels of one network. Luckily we find the *supernode* function could help us to combine some nodes easily, and we also provide more advanced functions based on the *supernode* function.

Recent researches also provide one analytical expression and the correct null models that allow for a quantitative discussion of the rich-club phenomenon [9], which could respectively measure the rich-club ordering and its relation with the function and dynamics of networks in biological, social and technological domains:

$$\theta(k) = \frac{2E_{>k}}{(N_{>k})(N_{>k} - 1)}$$

4. **Betweenness:**

A good measure of the centrality of each node has to incorporate a more global information such as its role played in the existence of paths between any two given nodes in the network. A natural definition of the betweenness centrality (BC) is given by the socialists [10], which counts the fraction of shortest paths going through a given node. More precisely, the BC of a node v is given by

$$g(v) = \sum_{s \neq v \neq t} \frac{\sigma_{st}(v)}{\sigma_{st}} \quad (2)$$

where σ_{st} is the total number of shortest paths from node s to node t and $\sigma_{st}(v)$ is the number of shortest paths from s to t going through v .

In the toolbox, we select the faster algorithm by Brandes [11], users could call the function *betweenness(i,g)* to calculate the betweenness of node i in the graph g quickly. For the edge betweenness, users could use the *split_edge function* to transfer one edge to one node, so that they could use the node betweenness function to obtain the edge betweenness quickly.

5. **Degree fluctuations:**

In the complex network, we provide a heuristic characterization of the level of heterogeneity of networks by defining the parameter $\kappa = \frac{\langle k^2 \rangle}{\langle k \rangle}$, for scale-free networks $\kappa \rightarrow \infty$. Users could call the *kba_para(g)* and *kfang_para(g)* to get the value of $\langle k \rangle$ and $\langle k^2 \rangle$.

6. **Percolation transition:**

Percolation theory focuses on the behavior of the percolation transition close to the critical point where it takes place. A basic quantity describing the system structure is the cluster number distribution $n_s(p)$, which measure the number of clusters of size s per lattice node, at the percolation probability p . The probability that any given node belongs to a cluster of size s is therefore $sn_s(p)$, where the fact that the node can be any one of the cluster's elements has been considered.

2.2 **Complex Network Models**

1. **ER Model (Random Networks):**

The simple and most studied network with undirected edges was introduced by Erdős and Rényi (ER model) [12]. In the network: (i)the total number of nodes, N , is fixed; (ii)the probability that two arbitrary vertices are connected equals p where $0 \leq p \leq 1$. The degree distribution of ER random networks is showed in Fig. 2, which is Poisson distribution.

2. **WS Model (Small-World Networks):**

Inspired by that many networks are highly clustered, while at the time they exhibit a small average distance between nodes. The Watts and Strogaz model [13] starts with a ring with N nodes in which each vertex is symmetrically connected to its $2m$ nearest neighbors. Then, for every node, each edge connected to a clockwise

neighbor is rewired with probability p , and preserved with probability $1 - p$.

3. Link Preferment Mechanism:

The most famous example is the *Barabási-Albert* (BA) model [14] which is based on two basic ingredients: growth and preferential attachment. The main idea is that in the WWW sites with high degrees acquire new links at higher rates than low-degree nodes that means “richer get richer”. Other models which include nonlinear preferential attachment, dynamic edge rewiring, fitness and exponential elements such as *Dorogovtsev-Mendes-Samukhin* model and *Krapivsky* model are also used frequently more recently. So we provide the following function with tunable parameters, users can obtain the different link preferment mechanism in the following.

$$\prod(k_i) = \frac{(k_i + k_0)^\alpha}{\sum_l (k_l + k_0)^\alpha} \quad (3)$$

where $\prod(k_i)$ is the probability that a new edge connects to node i , the degree of which is k_i and k_0 is a constant representing the initial attraction of a newly added node.

For the weighted network, we also provide the similar functions, only use the weight $S = \sum_l w_l$ instead of the total degree coefficient k of one node. Users could find functions to create the *Barrat-Barth'elemy-Cespignani*(BBV) model [15] and other similar models [16]. Moreover, since the Metanet toolbox provide the ‘*node_x*’ and ‘*node_y*’ property, it is easy to introduce the concept that scale-free networks in Euclidean space. The toolbox includes these functions too.

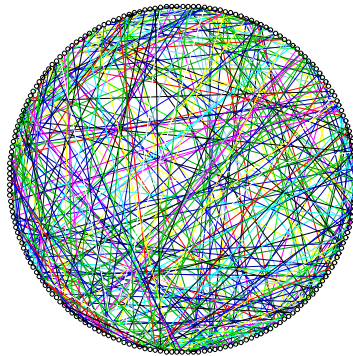


Fig. 3. ER model generated by *Scilab*. In ER model, each node own the equal possibility to be linked by other node.

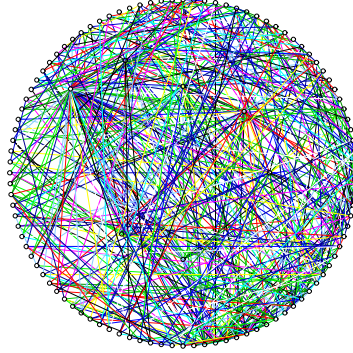


Fig. 4. BA model generated by *Scilab*. We could observe that some nodes in BA model have large degree because of the link preferment mechanism.

2.3 Dynamical Processes on Complex Networks

1. Traffic Dynamics and Attack:

Unlike *CiudadSim* [17], a toolbox designed for solving the traditional urban traffic problems, we focus more attentions on the traffic dynamics on complex networks which may reveal the underlying principles of Internet, Airline Transportation Networks *etc.* The representative example is the traffic of data packages on the network [18, 19]. We give different rules to send these packages and set different bandwidths of the edge to simulate the congestion of real networks. And users could observe simultaneously the data about the state of the traffic.

Another famous experiment is the attack of the scale-free networks which displays the interesting influence of the traffic capability of the networks [20]. In the computer simulation, scholars select different approaches such as to remove the nodes in the decreasing, increasing, or random degrees order and find some interesting results. In the toolbox, some attack functions selecting different strategies are able to be used.

2. Epidemic Dynamics:

Epidemic dynamics is used to measure the spread of an infection disease in a population of human, animals or computers with the aim of reproducing the actual dynamics of the disease, designing the strategies to control the propagation [21–23]. We will introduce three classic models that describe disease spreading through a contacts between infected and susceptible individuals: the susceptible-infected-removed (SIR), the susceptible-infected-susceptible (SIS) model and the susceptible-infected-rescued-susceptible (SIRS) model.

The dynamics in SIS model can be expressed as,

$$S(i) + I(j) \longrightarrow I(i) + I(j) \quad (4)$$

where i and j are two neighbors. And a infected node recover with the probability μ ,

$$I(i) \xrightarrow{\mu} S(i) \quad (5)$$

3. Game Theory:

Game theory could used to capture the intern essential elements of the characteristic interactions among individuals. In game theory, player must simultaneously decide wether to cooperate or defect. In the simulation in computers, people find these game theory in structured populations which represent the small world state will reveals critical phase transitions which fall into the universality class of directed percolation on square lattices and mean-field-type transitions on regular scale-free networks and random regular graphs. Recently scholars employ the prisoner's dilemma (PD) and the snow drift theory to discuss new insights gained in behavior ecology [24, 25]. Our toolbox include some similar strategies function (*figure 5*).

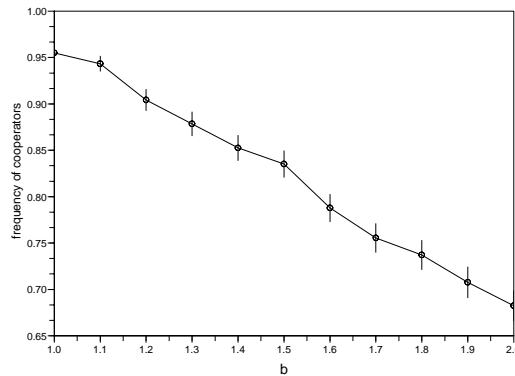


Fig. 5. Average frequency of cooperators as a function of b which represent benefit generated by *Scilab*. We choose BA graph and PD rules in our model.

2.4 Summary of the script interpreter

The TCL script language could manage some complex strings excellently. With the command “regexp” [26], programmers could write the special script parse soft. In the view of point, the toolbox include the main script interpreter and the conversion function which could translate the edge graph data file into the ‘.graph’ file used in Scilab. We also create the convenient GUI for users, (*see figure 6*) In the panel, people could add,

edit, delete these functions and parameters. In our simulation of dynamic evolutions of some networks, the mechanism works well. People could use the special “player of the script” and the calculation functions of parameters to observe details about the networks.

1. “Preview” button could get rough results via the known algorithm which we wrote in the *scilab script*, so that Scilab could display the possible results, but the method has its restriction that it would be inaccurate even wrong when it is faced with the new model made by users.
2. “Step Over” button could run the special scripts step by step and view changes of results caused by one step.
3. “Run” button would run all the scripts without interrupt if users are sure that every line of their script will work well. This method will give the accurate result eventually, but sometimes it will cost more time without using *PVM* technology when users calculate some parameters such as betweenness in one graph which have many nodes or links.

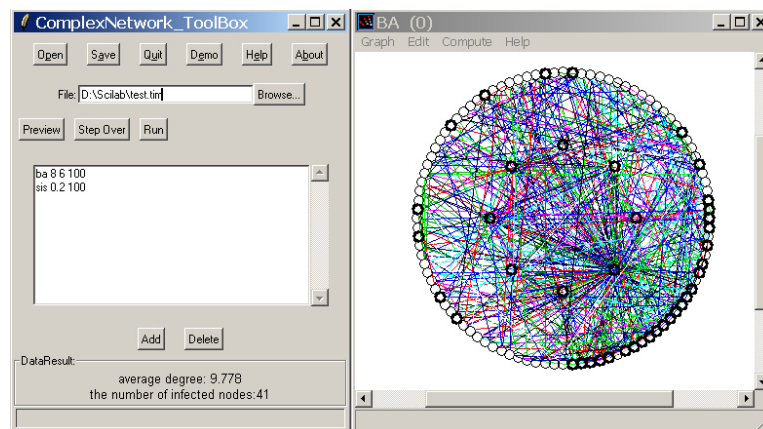


Fig. 6. GUI wrote by *TK* and *Scilab*. The script interpreter function is running one script. The example is representing SIS strategy process in a BA network. The highlighting nodes stand for the infected individuals, the normal nodes stand for susceptible individuals.

3 The flexible approaches for users

In researches, when people have an idea, they want to find whether there are something interesting immediately, the strategy functions could help them view the parameters of

their model directly and quickly. However, sometimes the experimental model without long time steps perhaps induce people to inaccurate results. In this case, people could use the function interfaced with C. Although people could not observe the visual details every step when considered the problem of the refresh speed, but they could obtain the data from the output file, and of course they could view the final statistic result using graph function in Scilab.

In the second place, with the development of the complex network, users would find more parameters or dynamic models. Although we will update our function library periodically, the functions library provided by the toolbox inevitably lack something latest. But users could add their ideas into the toolbox easily. The toolbox includes *Model*, *Strategy*, *Parameter*, *Dynamic function*, *User Interface* library. If people want to change or add one function, they could add their codes quickly.

Sometimes, people like using C or C++ or special topological graph softwares to realize their models, they also probably get the actual data from relevant institutes. The toolbox includes convert tools wrote by TK script which could convert the edge graph to the *.graph*. Then users could use the command '*load_graph*' to load their graph. If the graph data file is too large to be loaded, users could use '*stacksize()*' function to increase the amount of memory. After that users could use the toolbox to calculate parameters of the graph.

4 Conclusion

Scilab is an excellent scientific calculation software. It can be easily interfaced with Fortran, C, or JAVA programs which is useful to have faster code. It also provides the PVM module in that users could assign different duties to other computers. In our project, we use TK to write GUI, and through call the parsing string function wrote by TCL to analyze the script created by users, the result will enable *ScilabEval* to load relevant functions, get and display the final result. Current tests indicated that the toolbox works well on Windows XP and Debian Linux system.

References

1. Strogatz, S.: Exploring complex networks. *Nature* **410** (2001) 268-276
2. Albert, R., Barabási, A. -L.: Statistical mechanics fo complex networks. *Rev. Mod. Phys.* **74** (2002) 59-62
3. Wang, X. -F., Chen, G.: Complex Networks: Small-World, Scale-free and Beyond. *IEEE Circuits and Systems Magazine*. **3(1)** (2003) 6-20
4. Newman, M. E. J.: The structure and function of complex networks. *SIAM* **45**, (2003), 167-256
5. Networks /Pajek.: <http://vlado.fmf.uni-lj.si/pub/networks/pajek/>
6. Newman, M. E. J.: Assortative mixing in networks. *Phys. Rev. Lett.* **89** (2002) 208701
7. Pastor-Satorras, R., Vázquez, A., Vespignani, A.: Dynamical and correlation properties of the Internet. *Phys. Rev. Lett.* **87** (2001) 258701

8. Girvan, M., Newman, M. E. J.: Community structure in social and biological networks. *PNAS*. **99** (2002) 7821-7826
9. Colizza, V., Flammini, A., Serrano, M.-A., Vespignani A.: Detecting rich-club ordering in complex networks. *Nature Phys.* **2**, (2006) 110-115
10. Freeman, L.: A set of measures of centrality based upon betweenness, *Sociometry* **40** (1977) 35C41
11. Brandes, U.: A faster algorithm for betweenness centrality. *Journal of Mathematical Sociology*. **25(2)** (2001) 163-177
12. Erdős, P., Rényi, A.: On random graphs. *Publicationes Mathematicae*. **6** (1959) 290-297
13. Watts, D.-J., Strogatz, S.: Collective dynamics of 'small-world' networks. *Nature*. **393** (1998) 440-442
14. Barabási, A.-L., Albert, R.: Emergence of scaling in random networks. *Science*. **286** (1999) 509C512
15. Barrat, A., Barthélemy M., Vespignani, A.: Weighted evolving networks: coupling topology and weight dynamics. *Phys. Rev. Lett.* **92** (2004) 228701
16. Wang, W.-X., Wang, B.-H., Hu, B., Yan, G., Ou, Q.: General dynamics of topology and traffic on weighted technological networks. *Phys. Rev. Lett.* **94** (2005) 188702
17. CiudadSim.: <http://www-rocq.inria.fr/metalau/ciudadsim/welcome.htm>
18. Tadic, B., Thurner, S., Rodgers, G. J.: Traffic on complex networks: Towards understanding global statistical properties from microscopic density fluctuations. *Phys. Rev E*. **69** (2004) 036102
19. Zhao, L., Lai, Y.-C., Park, K., Ye, N.: Onset of traffic congestion in complex networks. *Phys. Rev E*. **71** (2005) 026125
20. Albert, R., Jeong, H., Barabási, A.-L.: Attack and error tolerance of complex networks. *Nature*. **406** (2000) 378C382
21. Pastor-Satorras, R., Vespignani, A.: Epidemic spreading in scale-free networks. *Phys. Rev. Lett.* **86** (2001) 3200
22. Kuperman, M., Abramson, G.: Small world effect in an epidemiological model. *Phys. Rev. Lett.* **86** (2001) 2909
23. Newman, M. E. J.: Threshold effects for two pathogens spreading on a network. *Phys. Rev. Lett.* **95** (2005) 108701
24. Santos, F. C., Pacheco, J. M.: Scale-free networks provide a unifying framework for the emergence of cooperation. *Phys. Rev. Lett.* **95** (2005) 098104
25. Lieberman, E., Hauert, C., Nowak, M. A.: Evolutionary dynamics on graphs. *Nature*. **433** (2005) 312-316
26. Brent B. Welch.: *Practical Programming in TCL/TK*. PH PTR, (2002)
27. Scilab Group.: *Scilab Reference Manual*. (2006)

A Testing Tool for Character Recognition Based On BP Network Using JAVASCI Interface

Ding Liu¹, Rui Xiao², Xin Zhou³

^{1,2}Research Center of Information and Control Engineering
Xi'an University of technology, Xi'an, China

³Systems Engineering Institute Xi'an Jiaotong University, Xi'an, China

¹Liud@xaut.edu.cn, ²sharrie_2001@163.com, ³xinzhou@sei.xjtu.edu.cn

Abstract. Character recognition is a very important and active research in pattern recognition. BP network recognition method is a new method of the research field. The testing tool for character recognition based on BP network using JavaSci interface is presented. The tool is implemented in JAVA language, a general programming language. We use Java libraries to make GUI and use SCILAB as compute engine. The structure, interface and function of the tool are introduced. The validity of the tool is proved by experiment.

Keywords: Character recognition, BP network, JavaSci interface, SCILAB

1 Introduction

Character recognition [1] is a very important and active research in pattern recognition. Theoretically, it is not an isolated technique. It concerns with the problem that all the other areas of pattern recognition must confronted. Practically, being a kind of information processing measure, character recognition has a very broad application background and vast need of market. Thus, it is of both theoretical and practical significance.

BP (Back Propagation) network recognition method is a new method of the research field in recent years, it can be used to solve or implement a problem if that problem can be reduced to pattern recognition. This method has some merits that traditional techniques do not have, such as good tolerance for error, strong sorting ability, strong parallel handling ability and strong self-learning ability.

Based on SCILAB[2], a good clone of MATLAB, but can be used freely, the paper introduces a general and all-powerful Character recognition tool, which is open source and free, supports as many features as possible, and with JavaSci interface, which written to allow calling SCILAB computational engine from JAVA. It eases the burden of a prior knowledge by utilizing a user-friendly GUI so the casual or new user can immediately begin operation. An efficient algorithm which is BP network is offered. Furthermore, the tool is designed to enable the serious user to run individual algorithms separately as well as interface his or hers own particular algorithms.

2 Introduction to Character Recognition on BP Network

2.1 BP Network Algorithm

Character recognition algorithm is the kernel of the tool, BP network algorithm [3] as recognition algorithm is applied by the tool in this paper, so we introduce the BP network first. Fig.1. shows a three-layers BP network, as it shows, La is input layer, Lb is hidden layer, Lc is output layer.

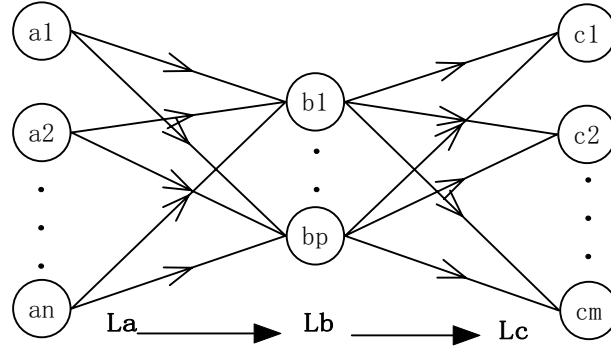


Fig. 1. Three-layers BP network.

The learning process of BP algorithm consists of two stages: The first stage is propagating the input information forward. The second stage is propagating the output errors backward. In the forward propagation stage, the information is input into the input layer. After the input is calculated and processed in turn at hidden layers, an output in the output layer and an error between calculating output and objective output are obtained. If the error value does not satisfy the desired error, i.e. no desired output is obtained, the process will turn to the stage of propagating backward. In the backward propagation stage, from output layer to input layer, error values are propagated backward along the same way and the weights and thresholds of neurons in each layer are modified layer by layer. The two stages of forward propagation stage and backward propagation stage work by turns until the desired output is obtained.

If the input vector is represented as a with n input neurons, the transfer function is Sigmoid function represented as $f(*)$. The output vector as c with m output neurons, the object output as \hat{c} , then

$$c_j = f(I_j) = f\left(\sum_{i=1}^n w_{ij}a_i - \theta_j\right). \quad (1)$$

where , w_{ij} represents the connecting weight value from neuron i to neuron j , θ_j is the threshold value. When input a learning sample, its sum square error is:

$$E = \frac{1}{2} \sum_{j=1}^m (c_j - \hat{c}_j)^2 . \quad (2)$$

According to gradient method, and the transfer function adopted in neurons is Sigmoid function, the adjusting function of weight is:

$$w_{ij}(n+1) = w_{ij}(n) + \eta \delta_j c_j . \quad (3)$$

where, $w_{ij}(n)$ is the n th modified weight value from neuron i to neuron j , c_j is the output of neuron j , η is the constant of learning rate, δ_j is the deviation of neuron j , we first adjust the connected weight value of output layer, then propagate back this weight value and δ value connecting to output neurons, calculate δ values of each intermediate layer and adjust intermediate weight values layer by layer using formula (3) until the desired output is obtained.

2.2 Application BP Network on Character Recognition

The process of application BP network on character recognition shows in Fig.2.

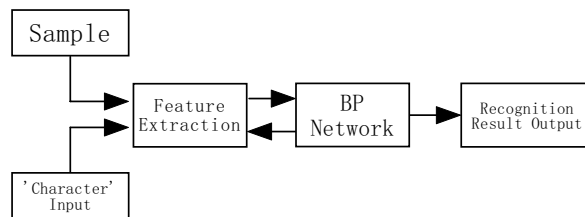


Fig. 2. Flow chart of BP network on character recognition

The sample was a digitized version of a character, divided into 8x8 pixels, each with a value of 0 or 1. The value of each pixel represents the features of sample. Input the extracted features of sample, then train the features using BP network, adjust weight values and δ values until the desired convergence of BP network is obtained. After training the sample, we could input a pending character for recognition, multiply input by weight values we obtained, recognize the character, then we could get the result of recognition.

3 Tool Running Mechanism and Its Structure

SCILAB offer JavaSci interface, which is SCILAB tool to interface SCILAB functions to JAVA, This thus allows, for instance to use Interface Builders such as Netbeans to design the graphic user interface (GUI). The interface built directly with JAVA is looking-nice, multi-look and feels, and much faster than the ones built with the SCILAB Graphic Object library. Furthermore, SCILAB is a scientific software package for numerical computations providing a powerful open computing environment. SCILAB includes hundreds of mathematical functions, which will be called easily. JavaSci interfaces are written to allow calling SCILAB from JAVA. SCILAB offer JavaSci interface like follows:

- SciStringArray – Class to use String matrix in SCILAB.
- SciDouble - Class to use double object with SCILAB
- Scilab - SCILAB Class

Example:

```
public double SFunction(double x) {
    double s = 0;
    SciDouble xf = new SciDouble("x",x);
    Scilab.Exec("s = 1 / ( 1.0 + exp( -x ) )");
    return s;
}
```

This example is a Java method, possessing sigmoid function. Java method call Scilab computational engine. As it shows, two interfaces SciDouble and Scilab were implemented.

The main panel of the tool illustrates the concept of recognition algorithms and can even serves as a flow chart:

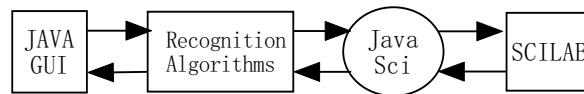


Fig. 3. Main structure of the tool

The main structure of the tool is given in Fig.2. As it shows, character recognition algorithm is the kernel of the tool, the next frame is the JavaSci interface, which allow calling SCILAB computational engine from JAVA. The graphical user interface is Java GUI [4]. The tool on a BP network-based algorithm can be viewed as a routine, in which sample are loaded and new sample are added, ten sample in all are trained by BP network and character input are then recognized. In addition, accordingly, tool use

such a structure: framework & extendable modules (plug-ins). This tool provides JAVA GUI allowing users to flexibly and interactively incorporate their own algorithms into it so as to test them. Tool offers an environment for storing, accessing, manipulating and visualizing character sample. For creative research programmers and methods developers, character recognition tool offers an extensible, open-source platform through which they can share new methods by contributing character recognition 'plug-in' functions that can be loaded easily.

4 Demonstration

Fig.3 shows the main window of the tool. As it shows, four menu items, i.e., “File”, “Operation”, and “Help”, are included character recognition tool. By clicking “File” and its submenu, users can initialize the tool and loading sample data, etc. “Operation” mainly devotes to the work related to sample training and a pending character was recognized. Three table items, i.e., “Sample Loading View”, “Training View”, and “Recognizing View”, user can view the results of character recognition. “Operation” Three button region items, user can operate the tool directly, and choose different look and feels, etc.

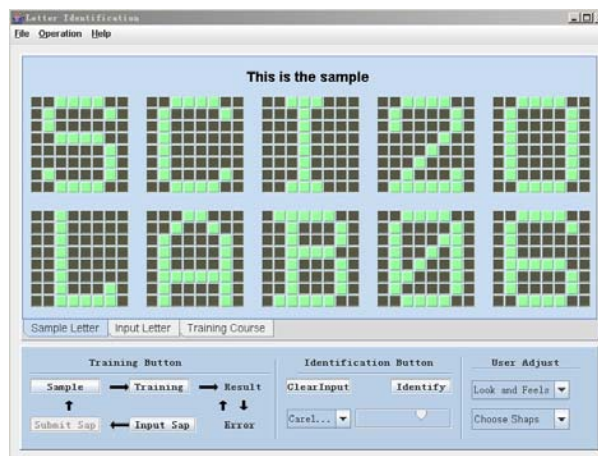


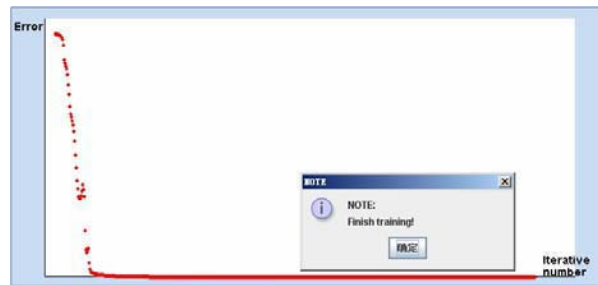
Fig. 4. The main window of character recognition tool

Simulations were done with the character data, the sample were digitized versions of ten characters, divided into 8x8 pixels, each with a value of 0 or 1 representing black or green respectively, which could be represent features of the sample. This produced quite crude representations of the characters, i.e., ‘SCILAB2006’, shown in Fig. 4(a). After training features of sample using BP network, we got the curve of convergence error shown in Fig.4 (b), we could input pending character for

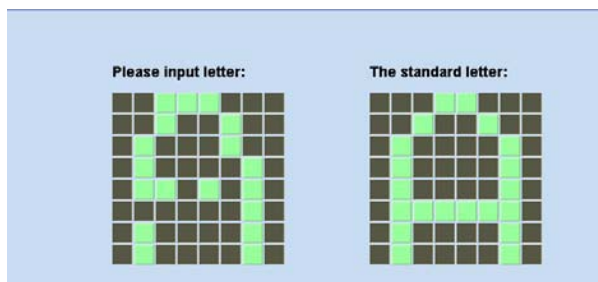
recognition. The input character may be a noised or defect character. If recognition succeeded, the output indicated which character has been recognized as the input. So we have an input output shown in Fig. 4 (c). If the tool couldn't recognize the input character, tool display an error message shown in Fig. 4 (d).



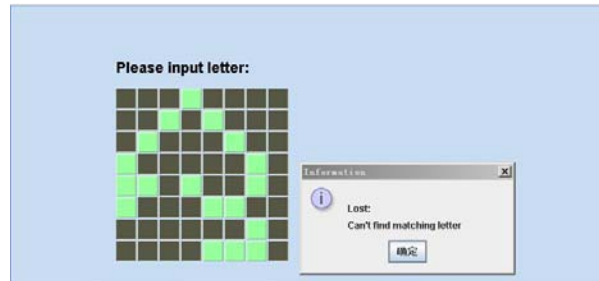
(a) Sample character



(b) The curve of convergence error



(c) The successful result of recognition



(d) The unsuccessful result of recognition

Fig. 5. Experiment results

6 Conclusion

In this paper, a tool for character recognition has been presented. JavaSci interface enhance the use of SCILAB, JAVA GUI made the tool easy to use. The kernel of the tool, Character recognition algorithm based on BP network, has been verified by the testing tool. BP network is a fast and effective method for character recognition. Furthermore, the tool is designed easy to be extended, it's convenient to add new plug-ins. This tool is under active open-source development. We plan to extend its capabilities to include more novel algorithms for character recognition problems.

References

1. Govindan V. K.: Character recognition-a review. Pattern Recogn, Artif.Intel, Lecture Vol. 5. pp1-24 (1991)
2. <http://www.scilab.org>
3. Phill Picton: Neural Network , second edition. PALGRAVE, New York (2000) 37-49
4. <http://www.java.com>

An ITS Dynamics Model Simulator Based on SCILAB

Yanlin Wang¹, Shuqin Lee¹, and Panfeng Xue¹

Associated with the other competition participants from USTC who submit an ITS-DMSS work

¹ Department of Electronic Engineering and Information Science, University of Science and Technology of China,
P.O.Box4, 230027, Hefei, Anhui, China
wylwendy@hotmail.com, lisq@mail.ustc.edu.cn, xpfg@mail.ustc.edu.cn

Abstract: Our surface transportation infrastructure has evolved to a level of complexity where Intelligent Transportation Systems (ITS) are essential for large urban environments nowadays. Thus a well-performing simulator towards ITS is in dire need. In this paper, we describe an ITS model simulator which is called ITS Dynamics Model Simulator based on SCILAB (ITS-DMSS) for building up the ITS model and simulating the real situations. Through this simulator, a best solution considering the distance, time, charge, and many more will be found out when driving. Also, a notion of Best-Way-Direction (BWD) is proposed in this article, which best embodies the dynamic characteristics of this simulator. With the Best-Way-Direction (BWD), vehicles can avoid sudden traffic congestion better.

Keywords: Intelligent transportation systems (ITS), ITS Dynamics Model Simulator based on SCILAB (ITS-DMSS), Best-Way-Direction(BWD), Data Gathering Part (DGP), Information Management Platform (IMP), Car Navigation Part (CNP), Dijkstra's Algorithm, Link State Routing Algorithm (LS)

1 Introduction

Transportation systems are playing a more and more important and critical role in all facets of modern life. Topics such as how to eliminate the traffic congestion, how to reduce the great quantities of collisions and so on have been ardently discussed. To further improve the efficiency and safety of the current transportation systems of all kinds, researches are done on a new and developing field of study, which is called Intelligent Transportation Systems (ITS).

Intelligent transportation systems (ITS) encompass a broad range of wireless and wire line communications-based information and electronics technologies. When integrated into the transportation system's infrastructure, and in vehicles themselves, these technologies relieve congestion, improve safety and enhance productivity. ITS

is made up of 16 types of technology-based systems. These systems are divided into intelligent infrastructure systems including Arterial Management, Freeway Management, Transit Management, Incident Management, Emergency Management, Electronic Payment, Traveler Information, Information Management, Crash Preventions and Maintenance, Road Weather Management, Commercial Vehicle Operations, Inter modal Freight and intelligent vehicle systems that contains Collision Avoidance Systems, Collision Notification Systems, and Driver Assistance Systems.

Since 1994, various organizations, including first the Ad Hoc Committee on Intelligent Transportation Systems (ITS) and then the ITS Council, were formed to explore the use of electrical and electronics engineering, systems and control engineering, and information technologies in ITS applications. Great supports are provided by the governments of different countries all over the world. Thus, significant progress on ITS has been made. Many ITS, such as ITS in Japan, ATIS, AVCS, ACVO in Europe, have already been put into daily use and many other countries are planning to build such systems as well.

The greatly increasing demands on the design of ITS in particular areas calls for a better contriving scheme to lower the design cost and make the design much more reasonable and persuasive, eliciting the simulation of Intelligent Transportation Systems (ITS) based on SCILAB.

In this article, we are going to discuss a new idea of using SCILAB to build an ITS model simulator, which easily adapt to the real time changes at any moment, and help the transportation managers better design the whole system.

2 ITS Dynamics Model Simulator based on SCILAB (ITS-DMSS)

2.1 ITS Dynamics Model Simulator based on SCILAB (ITS-DMSS)

ITS Dynamics Model Simulator based on SCILAB (ITS-DMSS) is a simulator that can build up an ITS model automatically after inputting a few basic data of transportation details of an area. Once the model is built, it can simulate the real-time situations of different sections of roads. As the user inputs the destination and other needs, the simulator, according to the instant traffic situations, can immediately select a Best-Way-Direction (BWD). At the time the virtual vehicle follows the instructions given by the system and drives along the Best-Way-Direction(BWD), the system helps to figure out the next Best-Way-Direction(BWD) at the next intersection considering several different aspects such us time, distance, safety, fees, and so on at that moment. After such dynamic path-choosing procedure, the ITS model built up by the ITS-DMSS can finally help to find out the best route.

2.2 Function Analysis

The ITS Dynamics Model Simulator based on SCILAB (ITS-DMSS) is designed to be a multifunctional system. Its major function is to find out a Best-Way-Direction(BWD) with all aspects considered, along with many accessional functions. The details are listed in the table below.

Table 1. Functions of the ITS-DMSS

Major Service					Accessional Services
Distance Consideration	Time Facet	The Basic Situation of Roads	Charge Facet	Other Special Consideration	Accident Disposal, Map Display, Traffic Information Supply, Weather Reports, Tour Guided, Video and Music Player Collision Avoided
Consider the Distance Between Set-off Point and the Destination	Congestion Avoided	The Width of the Roads, one-way and two-way traffic, Etc.	Lower the Charge	Emergency, Special Road Controlled, etc.	
With All Facets Considered, Find Out a <i>Best-Way-Direction (BWD)</i> at a time					

3 The Structure of ITS Dynamics Model Simulator based on SCILAB (ITS-DMSS)

The ITS Dynamics Model Simulator based on SCILAB (ITS-DMSS) is made up of three basic components: Car Navigation Part (CNP), Information Management Platform (IMP) and Data Gathering Part (DGP).

3.1 Car Navigation Part (CNP)

Car Navigation Part (CNP) is the visual terminal loaded to the supposed vehicles. It can exchange information with the Information Management Platform (IMP), and receive the driving info such as Best-Way-Direction(BWD), real-time traffic information display, etc. and accessional info such as weather reports, tour guided information etc. The unit sends out the messages to IMP about where the driver plans to go, which facet of cost he or she mostly considers and even sends the emergency help requests to the accidents as well.

The exact display of the real time map engendering and picking-up of local area will be a sparkle point but also a technical difficulty to the simulator.

3.2 Information Management Platform (IMP)

Information Management Platform (IMP) is the central disposal unit of the whole system. It receives the feed-back information from vehicles and processes the data to give CNPs further driving instructions.

3.3 Data Gathering Part (DGP)

As it is named, Data Gathering Part (DGP) is to collect data and information, then send them to the IMP. A Data Gathering Part (DGP) can be a traffic monitor, a speed monitor, or even a revised traffic light. Moreover, some useful additional data like the news report, the weather forecast, the tour guides, etc. can also be acquired from the Ethernet, LAN or WAN. That is to say, in some sense, the networks including Wireless LANs and Ethernet can also be an abstract Data Gathering Part (DGP).

3.4 The Relations Between the Components

The relations between these components and the major working principle of this simulator are shown in the graph below:

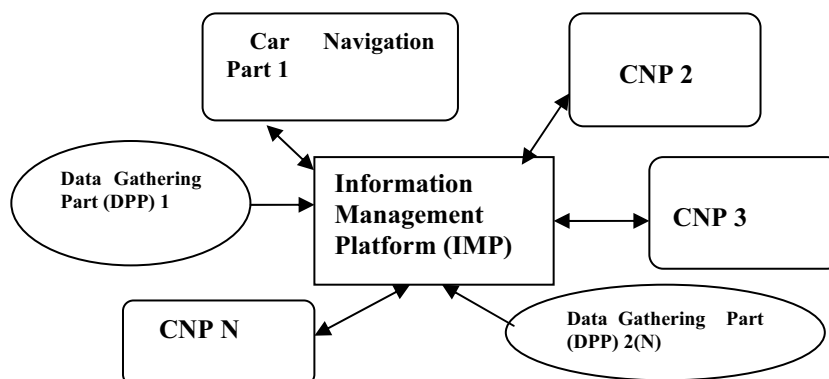


Fig. 1. The Relations Between the three components of ITS-DMSS

From the graph given above, we can easily see how these three components act on each other. The Information Management Platform (IMP) is the heart of the whole system. And the Car Navigation Part (CNP) is the indispensable component. Since the CNPs can already bring feed-back informations which portray out the real time traffic situation to the Information Management Platform (IMP), the Data Gathering Parts (DPPs) can be judged as optional items.

4 The Combination with SCILAB & Major Algorithm

4.1 The Combination with SCILAB

As the name indicated, the ITS_DMSS is based on SCILAB. We all know that SCILAB is a scientific software package for numerical computations providing a powerful open computing environment for engineering and scientific applications. Since the disposal towards the Intelligent Transportation Systems (ITS) contains gigantic data processing, such computation platform is needed.

Moreover, some other sparkle advantages of SCILAB attract us to use it to build up the ITS-DMSS. The kernel of this software is quite small compared to some other similar softwares, which is to say the calculation capacity and complexity in some sense can be lower. It has a lot of practical meaning to such huge system when the simulation is close to the real situation. And think it further more, if the system is applied in the real world, such small-kernel software can lower the cost of the maintenance, also enhance the efficiency of the system.

4.2 Major Algorithm to realize the ITS-DMSS

The major algorithm is based on an improved Dijkstra's Algorithm. Dijkstra's algorithm is known to be a good algorithm to find a shortest path statically. But to solve the dynamic path-seeking problem, an improved Dijkstra's Algorithm with dynamic feature is needed. We mainly use a loop clause to actualize this function. And for the consideration of those gigantic simulation systems, we proposed another function, which is based on Link State Routing Algorithm, to find out the best way much more efficiently in that situation. Here we list the major functions appear in the algorithm:

IMP(source_str,dest_str) It is the main function of the whole program. It uses a while clause as the loop clause in order to realize the dynamic function. The loop condition will be the match of the next crossing according to the Best-Way-Direction(BWD) (source_pos) and the supposed destination(dest_pos).

Caculate_Shortest_Path(start_pos,end_pos) It is the most important part of the whole program which helps to find out the shortest path. It is based on the Dijkstra's Algorithm.

LS_Caculate_Shortest_Path(start_pos,end_pos) It is a function which will take the place of the Caculate_Shortest_Path(start_pos,end_pos) function when the scale of the simulation is gigantic. The function is based on the Link State Routing Algorithm. As we know, the LS Algorithm has some good features for the distributed network. Thus, when the system is so enormous that it may contain several IMPs, the former Caculate_Shortest_Path(start_pos,end_pos) function is not efficient enough. The LS_Caculate_Shortest_Path(start_pos,end_pos) function would be a better choice then.

Cost_Input_Fuction() It is a function which defines the cost functions of the system such as distance, time, charge, etc..

CSMA/CA() It is a function to avoid communication collisions between different Car Navigation Parts and between the Car Navigation Parts and the Information Management Platform. It is based on the protocol CSMA/CA(Carrier Sense Multiple Access with Collision Avoidance) for the 802.11, thus enhances the system's efficiency in maximum.

Here is the most important part of the program:

```
function IMP(source_str,dest_str) % Information Management Platform
    global Node_Num;
    global Max_Cost;
    global node;
    global cost;

    node=node_input_function();
    cost=cost_input_function();
    S=char( node(1:Node_Num,:) );
    disp('all crossings: ');
    disp(S);
    source_find=0;dest_find=0;
    for i=1:Node_Num
        if( strcmp( deblank(node(i,:)) , source_str ) )
            source_pos=i;source_find=1;
        end
        if( strcmp( deblank(node(i,:)),dest_str) )
            dest_pos=i;dest_find=1;
        end
    end

    else
        while(source_pos = dest_pos)
            source_pos=get_right_pos();
        dest_pos=get_dest_pos();
        Caculate_Shortest_path(source_pos,dest_pos);
        source_pos=dest_pos;
```

```

CSMA_CA();
    end
end

```

```

function Caculate_Shortest_path(start_pos,end_pos)
    %function Caculate_Shortest_path(v0,v_destination)
    global Node_Num;
    global Max_Cost;
    global node;
    global cost;
    %%%%%% Dijkstra Algorithm %%%%%%
    v0=start_pos;v_destination=end_pos;
    %%%%%%Initiation%
    flag=zeros(1,Node_Num);
    flag(v0)=1;
    path=ones(Node_Num,Node_Num);
    path=path*(-1);
    dist=cost(v0,:);
    for i=1:Node_Num
        if( cost(v0,i)~=Max_Cost )
            path(i,1)=v0;path(i,2)=i;
        end
    end
    %%%%%%Initiation Ending%
    k=1;k1=1;
    for j=2:Node_Num
        while( dist(k)~=Max_Cost )
            k=k+1;
        end
        min=k ;
        for i=1:Node_Num
            if( ~flag(i) & dist(i)<dist(min) )min=i;
        end
        end
        flag(min)=1;
        for i=1:Node_Num
            if( ~flag(i) & dist(i)>dist(min)+cost(min,i) )
                dist(i)=dist(min)+cost(min,i);
                while( path(min,k1)~=(-1) )
                    path(i,k1)=path(min,k1);
                    k1=k1+1;
                end
                path(i,k1)=i;
                k1=1;
            end
        end
    end
end

```

```

end
%%%%%%%%% Dijkstra Algorithm Ending%%%%%%%%%

%%%%%%%%% output %%%%%%%%%%
i=1;way='The introduced way: ';
while( path(v_destination,i)~=(-1) )
    if(path(v_destination,i+1)==(-1))
        way=strcat(way,node( path(v_destination,i),: ));
    else
        way=strcat(way,node( path(v_destination,i),: ),'->');
    end
    i=i+1;
end
disp (way);

```

5 Concluding Remarks and Future Work

In this paper, we put forward an original designed simulator which is called ITS Dynamics Model Simulator based on SCILAB (ITS-DMSS) to build up the ITS model and simulate the actual situations. The simulator can search out a best way considering the distance, time, charge, and many more for the terminal driver users. And a notion of Best-Way-Direction (BWD) is proposed in this article, which best embodies the dynamic characteristic of this simulator. Vehicles can avoid the sudden traffic congestion better according to the Best-Way-Direction (BWD). The algorithm to realize such simulator is mainly based on Dijkstra's Algorithm for the general scale of simulation and LS Routing Algorithm for the gigantic scale of the simulation. Moreover, the accessional services such as accident disposal, map display, traffic information supply, etc. make the simulator keep pace with the future society's information integration.

To make the system more convenient, much work, such as algorithm optimization and expansion, service complementation, etc., should be done in the future. Significant challenges remain to further improve the efficiency and safety of the system. A real ITS Dynamics Model Simulator based on SCILAB (ITS-DMSS) will be submitted by us in this competition later in September.

References

1. F-Y. Wang and S. Tang, "A Framework for Artificial Transportation Systems: From Computer Simulations to Computational Experiments," Proc. 8th IEEE Int'l Conf. Intelligent Transportation Systems (ITSC 05), IEEE Press, 2005, pp. 1130–1134.
2. R.L. Bertini et al., "The German Autobahn: An ITS Test Bed for Examining Dynamic Traffic Flow Phenomena," Proc. 8th IEEE Int'l Conf. Intelligent Transportation Systems (ITSC 05), IEEE Press, 2005, pp. 502–507.

3. Figueiredo, I. Jesus, J. Machado, J. Ferreira, J. Santos, "Towards the Development of Intelligent Transportation Systems", in Proc.4th IEEE Intelligent Transportation Systems Conference, Oakland (CA), USA, pp. 1207-1212, 2001.
4. J. Sussman, Introduction to Transportation Systems, Norwood, MA: Artech House, ISBN 1-58053-141-5, 2000.
5. S. Ghosh, T. Lee, Intelligent Transportation Systems – New Principles and Architectures. Boca Raton, FL: CRC Press, 2000.
6. E. Lieberman, Ajay K. Rathi, "Traffic Simulation" in Traffic flow theory, Oak Ridge National Laboratory, 1997, ch

Audio Editor Toolbox for Scilab --- Scilab Melody

Pan Yuanhang , LinYongpeng , Min Dechun , Xu Xianming , Zeng Wenfeng ,
Wang Yan , Li Yanjie , Wang Zhihui , Liu Yang , Cai Jun , Li Shengrui

Computer Science Department of Xiamen University, Fujian, P.R.C
Panpan0000@sina.com

Abstract. Multimedia processing requires a great deal of scientific computing, some complex numerical computing arithmetic in particular. And Scilab, with great power in computing, can easily fulfill this requirement. Scilab Melody is a digital audio editor toolbox developed for Scilab. It strengthens the function on audio processing in Scilab. It includes rich effects, powerful audio edit function, wave analysis, user-friendly GUI. This paper will show you the algorithms how Scilab Melody edits the sound, and some techniques it uses.

Keywords: audio edit, scilab, wav , snack , echo, pitch shift

1 Introduction

Scilab Melody is a digital audio editor toolbox specifically developed for Scilab.

The goal of the toolbox is to extend the application of audio processing field for Scilab.

For we have find Scilab had only few functions about audio. It can only do simple playing, writing and reading without any other effects, also with some bugs. So Scilab Melody appears.

Scilab has powerfully advantages on matrix computation. So when facing audio effects, we can make good use of Scilab. We can read in a music file in the form of matrix, and do various operations on it.

Scilab Melody toolbox can do works about audio effects---echo, noise-reduce, pitch-shift, chorus, fade-in, fade-out and so on, recording, playing audio without being restricted by the stack size of Scilab and play the music.

And it can make comprehensive analysis on the time-domain and frequency-domain to the sound.

The current version of Scilab Melody is 1.0. It is developed under Scilab 3.0 and supports both Linux and Windows platforms.

2 WAV SOUND IN SCILAB

2.1 wav sound file format

Natural sound – the continuous analog signals of sound , can be converted into discrete digital signals by sampling and quantification. Then it will be saved into computer in RIFF format (Resource Interchange File Format).It is the criterion of the interface of multimedia program.

'WAV' is one format of RIFF, and it is also a basic sound format of WINDOWS.

It contains two parts, one is Format Chunk, the other is Sound Data Chunk . Format Chunk contains some important parameters of waveform, such as sampling frequency, precision of sample, size of file etc. While the Sound Data Chunk contains the actual data of waveform .

You can see an example from the following figure.

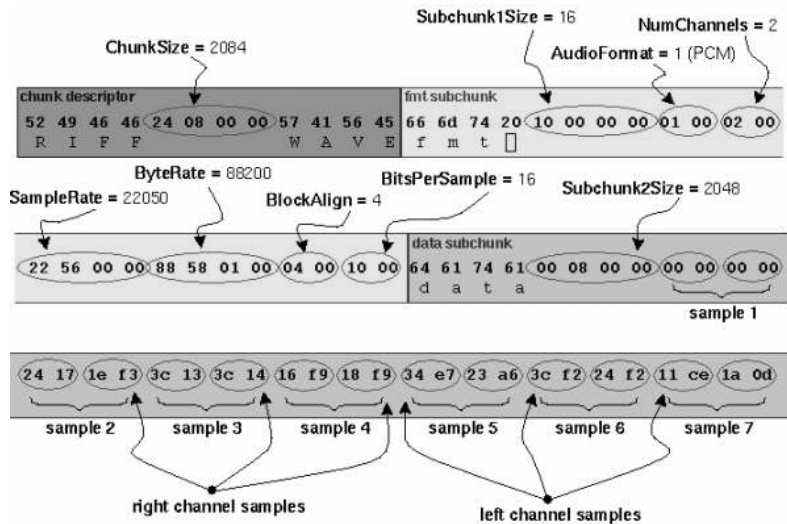


Fig. 1. wav file format example

Usually there are two kind of audio: monophony (2-tracks) and stereo (1-track). As to monophonic audio files, the sample is 8-bits (short integer 00H-FFH).And to stereo, it's 16-bits integer, high 8 bits respectively represent left track and low 8 bits for right track. The more the sampling bits are, the more exactly the sound is recorded, and the more data are required. There are usually three sample precisions: 8-bits, 16-bits, 32-bits.

The data chunks of WAVE files contain the samples which are in the form of PCM (Pulse Code Modulation).

2.2 wav file in scilab

Because WAVE file is non-compression, it takes up large storage disk.

$$\text{FILE SIZE} = (\text{SAMPLE RATE} * \text{BITS}) / 8(\text{bytes}). \quad (1)$$

But we still use it, because when we process other format, we have to first transform them into non-compression format such as .wav. Thus, wav is enough for processing.

Then, we use scilab function `wavread()` to read WAVE files. The function will convert the wave file into matrix. That means the music file is stored in the form of matrix in Scilab. If the music is monophony, then the matrix is a 1-D vector. And if the music is stereo, then the matrix is $n*2$ matrix. Each column stands for one channel. Each element in the matrix represents the quantification level value. The values range from -1 to 1.

According to the definition of sample-rate, there are FS sample points for each track in 1 second's music. So we can take it as this: a 3.2-seconds, 22.1KHz sample-rate stereo music is a 70720-rows, 2-lines matrix in Scilab ($70720=3.2*22100$).

Here we want to explain "sample rate". It is the times of sampling every time unit during the transform from analog signals to digital signals. Usually, the bigger the sample rate is, the better quality the sound is.

In general human can hear from 20Hz to 20 KHz, according to "Nyquist Functions", we can completely record the samples under 20KHz by using the sample rate above 40KHz in theory. And according to the early standard, the CD with the best timbre takes 44.1 KHz as its sample rate.

3 DESIGN OF THE SOFTWARE STRUCTURE

Just like figure.2, we read in a wav file as a sound matrix. Then we handle the matrix by Scilab. And the GUI layer responds to us the result, such as wave form or plays the music. Also we control the operation through the GUI. Scilab communicate with GUI through TK/Scilab Script. We can also save our operation to file by `wavewrite()`.

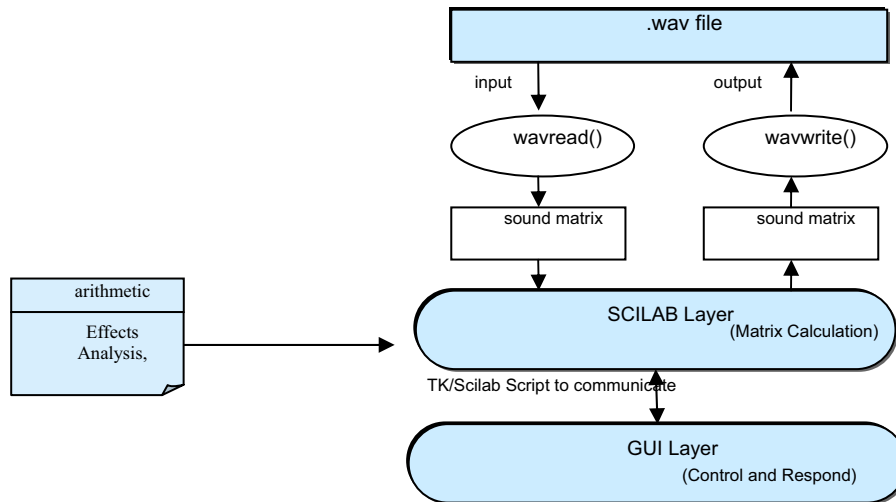


Fig. 2. Software structure

So, in scilab, the sample points of music are in the form of matrix. You can think the values are sample points of the sound wave's amplitude.

4 Linear Editing

It's the basic of audio processing .Simple but important.

- Amplitude Process

Try to magnify or shrink the sound wave. In scilab, it can be achieved easily by multiplying the sound matrix by a modulus, So that each sample point would be linear changed at the same time. The result is the volume of the sound raised or reduced.

The fade in and fade out effect: When the wave is multiplied by a curve vector which first rising then decline, the fade effect got.

If you want a piece of selected sound silence, just set the selected part of sound matrix to zero.

- Wave Form Process

To invert the sound wave, you can exchange the positive and negative part of the matrix (multiply -1).

To reverse the sound's head and tail, make matrix element [1] as [n], and [n] as [1], just so on.

- Normalize

Because there's always bound for digital sound data, just like [-1,+1] in scilab matrix or [-255,+255] when written to wav format file at 8 bits. And data out of the range will be lost. It means that the music is distorted and noise may appear. So we normalize the result of our effects so that the music would keep in the valid range. You'll find the peak value of matrix, if it's too large; multiply the matrix with a proper value like 0.9. So the data won't lose.

5 Frequency Domain Editing

For time domain signal, Fourier Transform is a main instrument to study its characteristic in frequency domain. We usually use Fast Fourier Transform (FFT). There are many frequencies in sound. Through FFT, we could get each frequency's information, such as frequency value, phase and amplitude and so on.

System transfer function is used to express a system's function. Usually denoted as $h(n)$, called impulse response. The result can be calculated as convolution between $h(n)$ and $x(n)$. I.E: $h(n)*x(n)$. here '*' represents convolve.

After $h(n)$ and $x(n)$ are Fourier transformed, it returns $x(k)$ and $h(k)$. The desired output could gain by simple multiplication. The z-transform of $h(n)$ is usually written as $h(z)$. (z-domain representation)

5.2 Equalizer

Equalizer is just to modulate the energy of the different frequencies of the sound. It can reach various effects through the way of strengthening or weakening certain frequency section.

First we divided the sound frequency into several sections, filtrate them out through the FIR filter. Multiply them by corresponding coefficient given by the user (in db), then combine them.

In scilab, we can achieve the result by following steps:

Denote x as the input signal.

Denote $z(i)$ and $z(i+1)$ as the section of frequency; and $a(i)$ as coefficient for each section.

```
1.wft(i)=wfirm('bp',50,[z(i) z(i+1)], 'hm', []); (i=1 2...n)
```

```
2.y(i)=a(i)*convol(x, wft(i))';
```

```
3. y = y(1)+y(2)+...+y(n);
```

In 1 step, 'bp' means filter type is band-pass, 50 is filter order, 'hm' means Hamming window. We filter out frequencies from $z(i)$ to $z(i+1)$.

For it will spend additional time to do step 2, we optimize it as following:

```
1.wft(i)=wfirm('bp',50,[z(i) z(i+1)], 'hm', []); (i=1 2...n)
```

```
2.wft = a(1)*wft(1)+a(2)*wft(2)+...+a(n)*wft(n);
```

```
3.y = convol(x,wft)';
```

5.3 Pitch Shift (tone changing) algorithm

Pitch is an effect that the music's tone will change but the playing speed and playing time remain.

5.3.1 Time and transpose:

Playtime = (num of sample points) / sample rate.

In another word, according to the formula above, the playtime alters when the sample rate changes. So if we want to speed up or slow down the music, just like change the playing speed of the cine-projector in cinema.

But there are other methods besides changing sample rate.

5.3.2 Frame

First of all, I'll introduce the concept of 'frame'. It's more or less the same as the frame in film. Playing film in cinema is to switch still pictures quickly. And the switching is quick enough that the film is dynamic.

The sound heard by human can be considered as a periodic signal in 10-20 ms. so when we deal with the music in unit of frame, the rhythm of music will remain. Another reason for using frame is that when Fourier Transform (FFT) each frame separately, the computing speed is much quicker than FFT the whole music.

5.3.3 Speed

We cut tail of each frame. Then the music becomes shorter. When play it at the primary sample rate, the music sounds like being accelerated. At the mean time, you would feel the tonality rising.

5.3.4 pitch shift

But the music of 5.3.3 above is speedup and play-time becomes shorter. We wonder whether we can make the transpose effect in Kara OK. And we make it.(Figure 3)

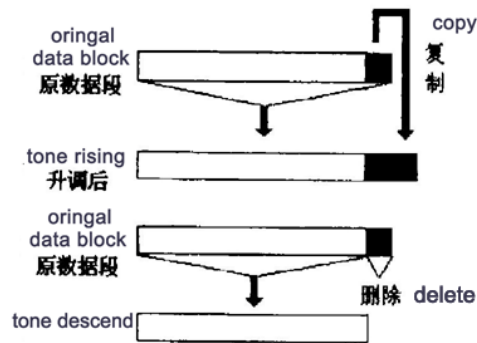


Fig. 3. pitch shift method

The algorithm is to change the sample rate to make it in company with the music's length. By adding a copy of tail to each frame, we prolong the music with no change to the rhythm. And increase the sample rate according to the new length of music. In term to Playtime = (num of sample points) / sample rate, the play speed remained, but the tonality rising(pitch up). Just like a man's voice becomes a woman's .

The add-tail Algorithm in scilab (using function matrix()):

denote **fn** as the frame num of a music, **fs** as the frame size (we can use “sample rate”*0.01 to be 10ms frame)

temp is an intermediate variable. It’s another form of the music.

assume input **x** is single track. **y** is output

```
temp = matrix(x(1:fn*fs), fs, -1)';
//the input x is divided into fn row, fs column .
//one column is a frame.
[rn, cn] = size(temp);
total = cn - ceil(cn*cof);
frame_tail = temp(:, cn - total/2 + 1 : cn);
//cut the tail of frames
y = [temp, frame_tail]'; //add the tails
y = matrix(t, 1, -1)';
//convert to column vector, so matrix turns to
// column vector--our familiar format
```

After that , we can increase the sample rate of music , the tone-rising is done.

It’s almost the same with falling tone. Throw away the tail of frame, Then reduce the sample rate accordingly. The result is the tone fell and the speed remained.

According to music principle, when increasing each frequency of music by $2^{(1/12)}$ times, it equals to increasing a Semitone. Contrarily, frequency reduces by $2^{(1/12)}$ times, one Semitone fall.

So increase music size by add tails and increase sample rate by $2^{(1/12)}$ time at the same time, 1 Semitone rise.

5.4 noise reducing

Usually the noise is high frequency signal. We can take out the high frequency part by a low-pass filter.

If we know the frequency of the noise added to the sound. For example, a 5K Hz sine wave noise was added. We can use band-pass filter and hamming window to eliminate this noise in 5K Hz alone.

If we don’t know the noise’s frequency, but a noise sample is available, we can analyze the FFT result of the sample, then pick up the peak-value(the strongest value). The value may be the main frequency of the noise. Then we can eliminate it from the mixed signal as above.

5.5 Sci-filter and effects

5.5.1 Filter

We choose to convolve the time-domain representation of the signals with the filter.

5.5.2 Sci-filter

In Scilab3.0 document, there is filter design function ‘filter’. But we could find it. In matlab, filter is an important function. And the similar one in scilab is ‘flts’, it’s too slow to bear.

So we design a ‘scifilter’ to deal with the filter process in scilab. Its usage is almost the same as matlab’s filter. So that users familiar with Matlab can easily use it.

And the scifilter’s algorithm based on “long exact divide” Algorithms:

1. get transform function -- $H(z)$
2. Inverse z-transform the $H(z)$ (using `ldiv()`) then we got impulse response $h(n)$
3. convolve the input x with $h(n)$, the result is the filter output.

Denote $H(z)$ to be the transfer function .

In scilab, let variable b is numerator of the transfer function $H(z)$, and a is denominator. For $H(z)$ above , $b = [1 \ 2 \ 0 \ 0 \ 3]$; $a = [1 \ 0 \ 0 \ -5]$;

The follow is scilab script for sci-filter algorithm:

```
//The coefficient of the polynomial A & B:
A = poly(a, 'z', 'coeff' ); //denominator
B = poly(b, 'z', 'coeff' ); //numerator
H = B./A; //H is the transfer function H(z)
h=ldiv(B,A,max(size(a,2),size(b,2)));
// inverse z-transform : h is the result of inverse
y =convol(h,x); //convolution between input x and h
```

So we can use it as : $y = \text{scifilter}(b,a,x)$ x :input, y :output

we will see that it's almost the same with the scilab function `flts` and our `scifilter`. The error is inconspicuous. And the little error in the music processing can be ignore .Though the music distorted, but don't forget that the effects like echo in fact are also making the music distortion in some aspect. So they are almost the same. Little difference can be ignored to your ears. But the time is 10 times different.

5.5.3 The delay effects

With “sci-filter” algorithm, we could easily apply many transfer function of z-field expression to music, to gain various of effects.

Here we demonstrate many transfer function of several delay effects which we use in Scilab Melody.

- Echo

An echo of a signal can be implemented by the following filter, which adds to the original signal an attenuated and delayed copy of itself .

The FIR filter demonstrated the effect of a single echo is of the form:

$$y(n) = x(n) + bx(n-D) \quad (|b| \leq 1)$$

x is original signal. b is the weighting factor or the attenuated coefficient and reflection and propagation losses. The delay D represents the round-trip travel time from the source to a reflecting wall.

The transfer function in z -domain representation is following:

$$H(z)=1+bz^{(-D)}$$

Then we apply transfer function $H(z)$'s numerator B as $[1,\text{zeros}(1,D-1),a]$ and denominator A as 1 to function $Y = \text{scifilter}(B,A,X)$.

- Multiple (FIR) Echo

As the name implies, a multiple echo is nothing but adding a number of delay units to the original signal. It is closer to the reality.

For example adding 3 successive echoes using the following comb filter:

$$y(n) = x(n) + b_1 * x(n-D) + b_2 * x(n-2D) + b_3 * x(n-3D) + \dots$$

$$H(z)=1 + az^{(-D)} + a^2 * z^{(-2D)} + a^3 * z^{(-3D)} + \dots$$

We assume that the propagation losses will be double to former time.

- Reverberate

Reverberation is defined as "the combined effect of multiple sound reflections within a room."

Echo is a distinct delayed version of the signal whereas a reverb has a very short delay time that makes it difficult to distinguish each reflection. The FIR reverb is

$$y(n) = x(n) + b_1 x(n-D) + b_2 x(n-2D) + \dots + k \text{ terms}$$

And a simple IIR reverberation involves saving the past values of the output instead of the input values.

$$y(n) = x(n) + a * y(n - D)$$

$$H(z) = 1 / (1 - a * z^{(-D)})$$

Refer to `scifilter()`: $B = 1$; $A = [1,\text{zeros}(1,D-1),-a]$

And a flat-response reverb has a flat response for all values of frequencies. We know that for an all-pass filter the gain is ideally unity for all values. The flat response filter is derived from the all-pass filter for which the poles and the zeros are conjugate reciprocals of each other.

$$y(n) = a * y(n-D) - a * x(n) + x(n-D)$$

$$H(z) = (z^D - a) / (1 - a * z^{(-D)})$$

- Chorusing

Just as a chorus is a group of singers, chorusing imitates the effect of a group of musicians playing the same piece simultaneously. They are more or less in sync with each other except for minor variations. The small variations in the time delays and amplitudes can be simulated by varying them slowly and randomly. A low frequency random time delay was generated.

$$y(n) = x(n) + b * x(n-d(n))$$

The function $d(n)$ is obtained from a slowly varying random signal and after passing it through a low pass filter for producing the minor changes in amplitude and phase.

$$d(n) = D_1 + (D_2 - D_1)v(n)$$

Where $v(n)$ is a low pass random signal varying between 0 and 1.

6 Audio signal analysis

First do FFT on the specific signal matrix. Return the result **F** and get absolute value **F1**.

Denote the y axis as $F1(\text{min_freq}*\text{points}/\text{rate} : \text{max_freq}*\text{points}/\text{rate})$ (rate is the sample rate), and x axis as frequency from min to max frequency of the signal .Plot them and you'll get the frequency-spectrum figure .

If you apply F to function `angle()`, you'll get the phase of the signal.

If get the square of the modulus of F and plot it , it will be power-spectrum result.

Through the FFT result , we could obtain:

- min / max sample value : immediately get from the result.
- min / max RMS power:

This value is the min/max Root Mean Squared amplitude (closer to what the ear hears)of the waveform that is scanned using a window as the RMS Window Width.

7 Large file processing

The audio file stores in Scilab in the form of matrix. Hundreds of thousands of samples are needed to play a 1-second file. That means much more bytes are needed. And usually a music file is much longer than 1s.

The stack size of Scilab is limited. So it's impossible to load all the music into the scilab stack.

We can load a block (section) of music into scilab, after processing ,write it back, then load next block, process...etc . Thus no matter how large the file is, it won't be limited by the stack size of Scilab. So we implement `wavreadblock()` and `wavsavblock()` function .

But when we process echo effect block by block, noise will appear in the juncture between blocks. To smooth the block juncture, we always read in 1~2 blocks ahead and after the current block, then "echo" them together

8. Snack toolkit

8.1 Introduction

The Snack Sound Toolkit is designed to be used with a scripting language such as Tcl/Tk or Python, developed by Sweden Royal Institute of Technology. Using Snack you can create powerful multi-platform audio applications with just a few lines of code.

We use it to create waveform and play music(in linux) for our GUI .

Snack has commands for basic sound handling, such as playback, recording, file and socket I/O. Snack also provides primitives for sound-visualization, e.g. waveforms and spectrograms.

For example, in Tcl/Tk you can use it like this:

```
canvas .c -width 400 -height 200

sound snd -file test.wav

.c create waveform 0 0 -sound snd
```

8.2 Interface between snack and scilab

We know scilab can communicate with Tcl/Tk interface to pass simple integer or float value, but matrix variable passing is impossible. It's uncomfortable.

We develop an interface for matrix access. It give Scilab the ability to access sample matrix in an object Called "sound" in TCL/TK, what is provided in Snack package.

The interface bases on Scilab's C interface tech and snack's C/c++ extension. (Scilab can share the same .dll file) So we can allocate room by "c program" as a "mid - agent ". We call it "sounddata".

They all provided "get", and "set" access to the c allocated room. They all have the ability to get and modify it. That's to say that it connects the two systems. For example: you can set sound matrix in Tcl/Tk and get it in scilab.

The Tcl/Tk command is developed by Snack's Tcl/Tk "C" extension, called "sounddata"; It build in snack's sound command. It has a strict index check. The "get" command allocates "C" space to store the data. To "get", you should provide the start index, end index and the channel (default is left channel); Once the end index is out of range, a default value given, that's the length of sound object, such as

```
"TK_EvalStr("snd sounddata get 0 2000000");
```

The "set" command free the room. If you set the processed "sounddata", the room allocated by "C" is free, and then you can't "get" the "sounddata" in "Scilab" any more. To "set", you should point out the start index of the "sound" structure, like "TK_EvalStr ("snd sounddata set 0");

We can take full advantage of Scilab's ability on matrix computation to process the signal. By the outstanding advantage on matrix computation of Scilab, and Snack's excellent audio-visual, supporting various file formats, we can make the music world of scilab more soundful and changeful!

References

1. Hu Baogang, Kang Mengzhen and Zhao Xin : Introduction to Scientific Computing Software—Scilab. Tsinghua Press, Beijing (2003)
2. Sophocles J.Orfanidis : Introduction to Signal Processing Tsinghua Press, Beijing (2001)
3. Scilab Group (INRIA Meta2 Project/ENPC Cergrene): Signal Processing with Scilab

4. Arun Krishnan Unnikrishnan : Digital audio experiments
5. Di Hongwei, Chen Longyun, Zhang Yonglin : “research and implement for digital music pitch shift” computer engineering and application pp 27 (1998)
6. Luo junhui, Luo Yongjiang ,Bai Yichen, Pang Na : Matlab7.0: Application in digital signal processing China Machine Press (2004)

Campus Emergency Management System based on Scilab

Wang Xin, Ke-Jun Wang

Automation College, Harbin Engineering University, Harbin, Heilongjiang, 150001
bafferlo@yahoo.com.cn

Abstract. This paper describes the designing of campus emergency management systems based on Scilab. The framework and all modules of this system are introduced in detail, in which GIS module plays a crucial role. Besides, a system for Harbin Engineering University is discussed as an example, which proves to be fast, stable and efficient.

Keywords: CEMS; GIS; Scilab; TCL/TK

1 Introduction

In times of crisis, the President of the University, and other senior officials has the primary responsibility for rapidly and effectively managing any emergency that might occur on or affect the campus. To assist the president and officials, a campus emergency management system, based on Geographic Information System, is needed urgently.

A Geographic Information System (GIS) is a flexible tool that can be utilized to provide detailed situational information for response to an emergency situation. There are several reasons why GIS is well suited for an emergency management application. First, GIS systems produce information quickly, which is particularly important for emergency managers who need to make time-critical decisions. GIS applications can create maps quickly integrating data into one single picture, both dynamically on a computer screen and in the form of a printed map. Second, GIS data is typically portrayed in a map with structured data, and thus the information is well organized, accessible and easy to understand, promoting informed decision-making and reporting capabilities. A map with critical areas identified can communicate data quickly and in a manner easily understood compared with a strictly tabular data output. Third, in GIS, data is easy to update, and it is easy to maintain current files and data integrity. It can provide a visual query interface that makes it is easy to enter and access the data stored in a structured format.^[1]

GIS has made great success in almost every field throughout the world, such as Resource Management, Resource Configuration, Urban Planning and Management, Land Information System and Cadastral Application, Emergency Response, Application in Geosciences, Business and Marketing, Facilities Management, Site

Selecting Analysis, Network System Analysis, Visualization Application and Distributed Geographic Information Application.

However, most of these systems are developed using commercial software like MapInfo, ArcInfo, and SuperMap, which are excellent but so expensive for researches in colleges. Another way of designing GIS is to seek help from Open Source GIS software, such as Grass and Terralib, but they are difficult to grasp, not mention design systems. Fortunately, Scilab gives us another choice. With its great functions on numerical computing and visualization of results^[2], especially a powerful toolbox called Metanet, we designed a system named Intelligent Digital Map in Sep.2005, which got the first prize in 2005 Scilab Contest in China. Our works prove that it is probable, though difficult, to construct GIS based on Scilab. Besides, we do further research on GIS and CEMS, and have developed a CEMS for HEU (Harbin Engineering University) to manage emergency more rapidly and effectively.

2 Structure of CEMS

The general framework of CEMS is shown in Fig.1, including 5 main parts, GIS Module, Emergency Management Module (EMM), GUI (programming in Tcl/Tk), Network Communication Module and database.

Actually, our CEMS have two editions, one of which is server, and the other is client. The former one (server) is designed for emergency managers/directors (usually college/university president and other senior official), and the latter one is for emergency reporters (also rescuing teams) distributed all over the campus. Clients and server communicate through Network Communication Module, which is a package written in tcl/tk. There are several reasons why C/S framework is adopted, rather than B/S, which is popular in GIS field. First and foremost, it is almost impossible (maybe possible but so difficult) for Scilab to construct a B/S framework. Secondly, to avoid attacks from the Internet, C/S structure is a better choice. Besides, it will be a burden for CEMS server if this system is distributed on the Internet.

As we've discussed in part 1, GIS module plays a key role of the whole system. There are two kinds of digital map in GIS, raster map and vector map. Since no vector map toolbox has been developed for Scilab, we have to design the system based on raster map. Raster maps are digital maps on the basis of individual pixels. The foundation of GIS module is GIS database, constituted of a raster map, a graph and other data. When the system is initialized, the raster map and the graph will be load into Scilab, and then you can search objects, compute shortest path, and do other analysis.

Emergency Management Module (EMM) consists of three sub modules, Preparedness Module, Response Module and Recovery Module. Preparedness Module is connected directly to EMM database, and with this module, you are able to view emergency plans for all kinds of emergencies occur on campus. When emergencies happen, reports are sent from clients, and then Response Module works together with GIS Module. The digital map will locate the risk areas automatically, and display reports sent from clients. At the same time, emergency managers can see

related emergency plans to direct rescuing procedures. In order to record information after the emergency, Recovery Module is designed.

The GUI of this system is written in Tcl/Tk, a powerful script language has the advantages of platform-independent, fast, efficient and easy to master^[3]. Raster map, overview map, information panel and menu are four parts of the interface, which is shown in Fig.2.

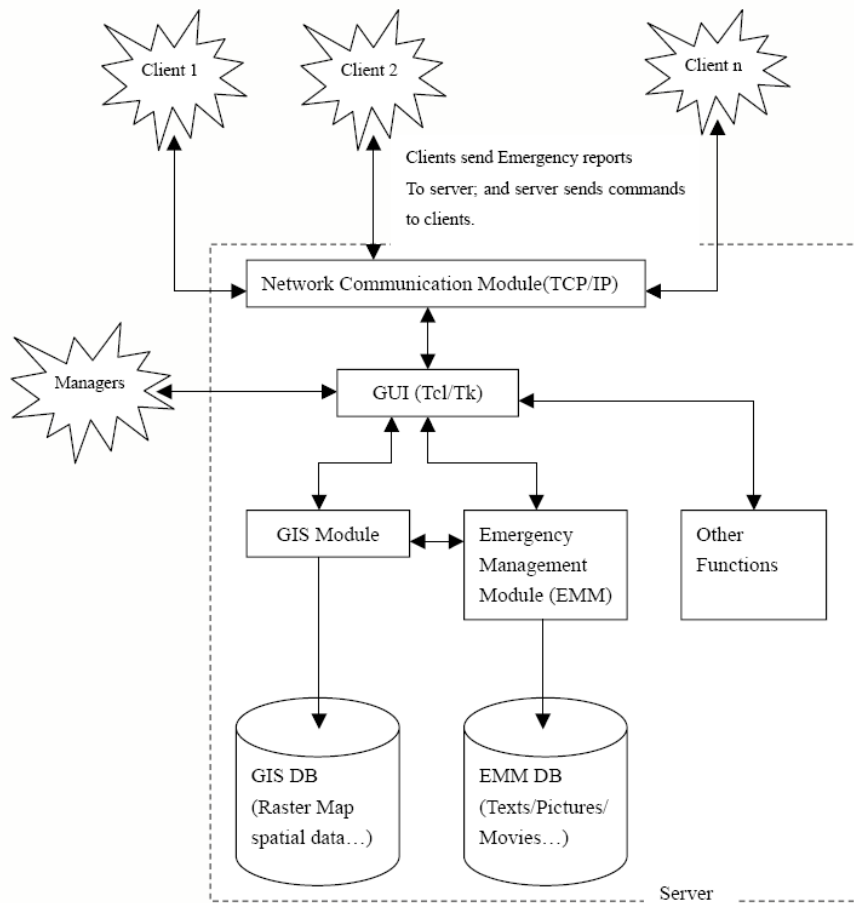


Fig. 1. Structure of CEMS

3 Detail Design of each module

3.1 GIS Module

Before GIS Module is constructed, the names, coordinates and indices of nodes in the graph should firstly be collected. A Tcl/Tk package named “pkgDotCollection.tcl” is written and used to fulfill this tough work.

The structure of GIS Module is shown in Fig.3. At the beginning, the indices, coordinates and names of nodes are read into Scilab, and the raster map, overview map are read into Tcl/Tk. After that, the graph is built with the help of metanet toolbox. The computing works, such as searching objects, computing shortest path, are achieved by Scilab functions. Meanwhile, the visualization of computing results and operation of digital map are realized by Tcl/Tk packages.

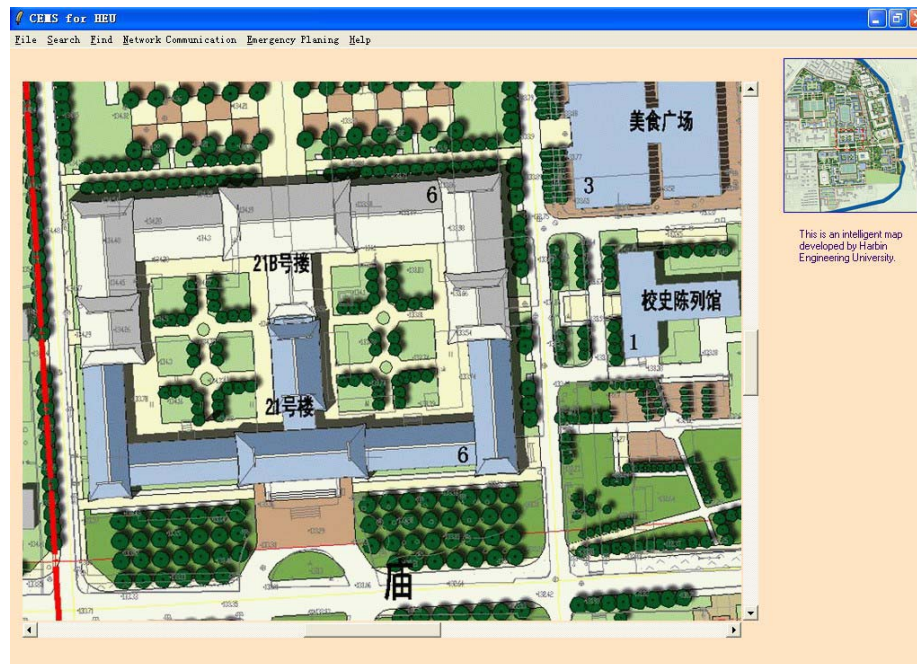


Fig.2. GUI of CEMS

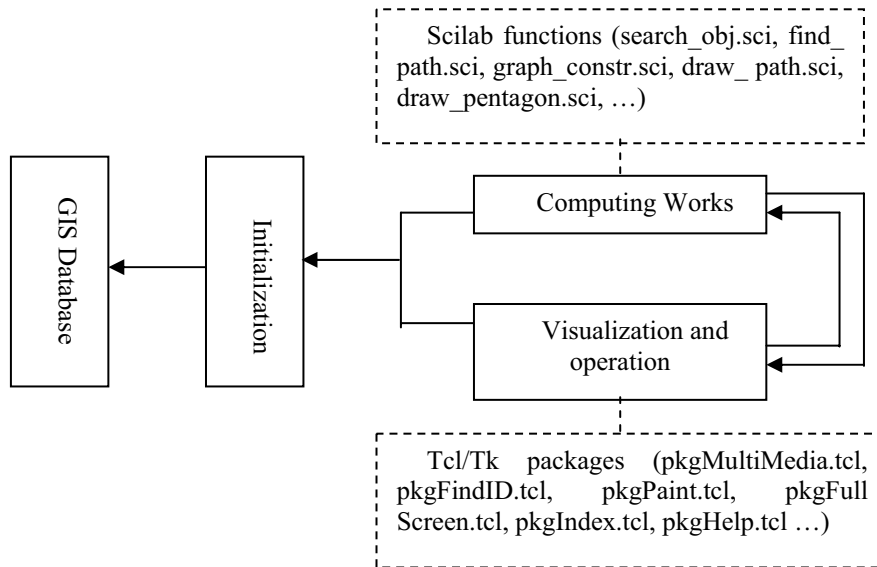


Fig.3. Structure of GIS Module

The most important data structure in GIS Module is the structure of graph. A graph is a set of arcs and nodes. It is constituted by the relationship of the arcs and nodes, where the arcs express the streets; the nodes denote the junctions of the streets, the prominent buildings beside the streets (such as teaching buildings, gymnasiums, libraries...), and so on. Some of the nodes are useless for the users, because the users are just interested in the target location and the shortest path they want to search. But these nodes must be contained in the program because the algorithm of the shortest paths needs these nodes.

The Metanet toolbox has this graph structure. The data input of the graph are: the number of nodes, row vector of the tail node numbers, row vector of the head node numbers, row vector of the x coordinates of the nodes, row vector of the y coordinates of the nodes and row vector of the sizes of the arcs. The following is the code in the file "graph_constr.sci", which constructs the graph structure of the map:

```

...
//Construct tail vector and head vector.
...
g=make_graph('mapheu',1,146,tail,head);
nodexy=read(DIR+'data\coordinates.dat',146,2);
g.node_x=(nodexy(:,1))';
g.node_y=(nodexy(:,2))';
B=2*ones(1,size(g('tail'),'*'));
//compute length of edges

g('edge_length')=sqrt(abs((g('node_x')(g('tail'))-
g('node_x')(g('head'))).^B)+abs((g('node_y')(g('tail'))-
-g('node_y')(g('head'))).^B))

```

3.2 Emergency Management Module

As is mentioned before, Emergency Management Module (EMM) consists of three sub modules, Preparedness Module, Response Module and Recovery Module. The relationships can be seen in Fig.4.

Preparedness Module depends on Emergency database, which includes emergency plans about various kinds of campus emergencies, such as food poisoning, fire, SARS, political emergencies, collective activity and serious infection.

Response Module and Recovery Module will be invoked when server receives reports from clients. Preparedness Module will be invoked by Response Module to display related emergency plans, and GIS Module will be called to show emergency area and compute shortest path. In the background, Recovery Module saves information about the emergency.

After emergency manager's analysis with the help of emergency plan, commands will be sent to clients to take actions immediately.

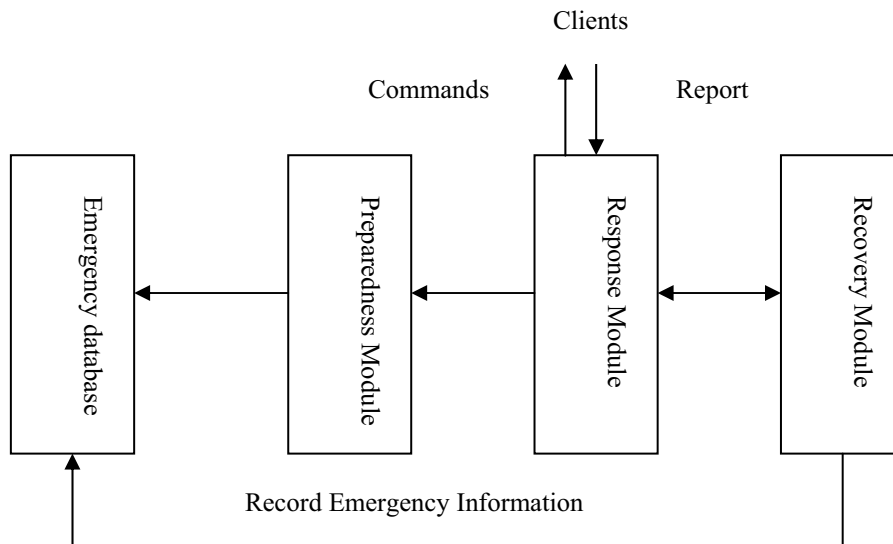


Fig.4. Structure of EMM

3.3 Network Communication Module

Fig.5. shows the mechanism of Network Communication Module.

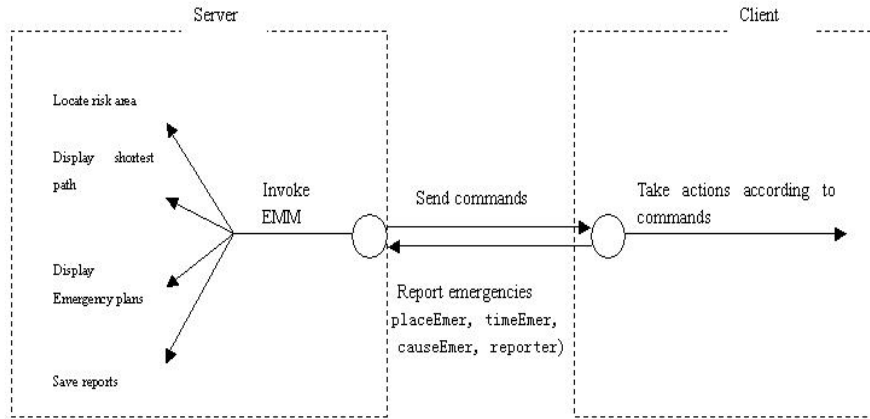


Fig.5. Mechanism of Network Communication Module

The Network Communication Module is based on TCP/IP protocol, so it is necessary to configure the IP address and port of server when client wants to connect. The following codes are part of “pkgNetworkCommunication.tcl”, which is the package fulfill the task of network communication.

```

#Network Configuration
proc NetConfigure {} {
    variable echo
    variable WinNetConf
    variable ipini
    variable portini

    catch {destroy $WinNetConf}
    toplevel $WinNetConf
    wm title $WinNetConf "Network Configuration"
    wm iconname $WinNetConf "Network Configuration"

    frame $WinNetConf.f1 -borderwidth 10
    frame $WinNetConf.f2 -borderwidth 10
    label $WinNetConf.f1.l1 -text "Server's IP address:
" -font 2
    entry $WinNetConf.f1.e1 -textvariable ipcmd
    label $WinNetConf.f1.l2 -text "Apply Port: " -font 2
    entry $WinNetConf.f1.e2 -textvariable portcmd
    button $WinNetConf.f2.b1 -text "OK" -relief ridge -
width 15 -font 2 -command {NetworkCommunication::NetSave
$ipcmd $portcmd}
    button $WinNetConf.f2.b2 -text "Default
Configuration" -relief ridge -width 15 -font 2 -command
{NetworkCommunication::NetSave
$NetworkCommunication::ipini
$NetworkCommunication::portini}
    grid $WinNetConf.f1.l1 -row 0 -column 0 -sticky w
    grid $WinNetConf.f1.e1 -row 0 -column 1 -sticky w
    grid $WinNetConf.f1.l2 -row 1 -column 0 -sticky w
    
```

```

grid $WinNetConf.f1.e2 -row 1 -column 1 -sticky w
grid $WinNetConf.f2.b2 -row 2 -column 1 -sticky w
grid $WinNetConf.f2.b1 -row 3 -column 1 -sticky w
pack $WinNetConf.f1 $WinNetConf.f2 -side top
}

```

Note that the server should open the port that clients apply for.

After the connection is successful, clients are able to send emergency report to server, including the reporter, place, time, and cause of the emergency. When server gets this information, it will automatically invoke EMM. The risk area will be displayed immediately; the shortest path from any other place to emergency area will be computed; the emergency plan will be displayed, and you can choose texts or movies about emergency; at the same time, the report will be record at the background for future use.

4 Example

Suppose one day in HEU, a client nearby finds No.11 teaching building on fire, and a report about this emergency is sent immediately. Fig.6 shows details about this report.

Manager will receive this report if server's port is open and the client has the right network configuration. Fig.7 shows server's status after it has received the report.



Fig.6. Reports window



Fig.7. Server's status when receiving report

At this time, the manager has many choices, read emergency plan, watch movie about this emergency or find shortest path to emergency spot. The shortest path from manager to object is shown in Fig.8.

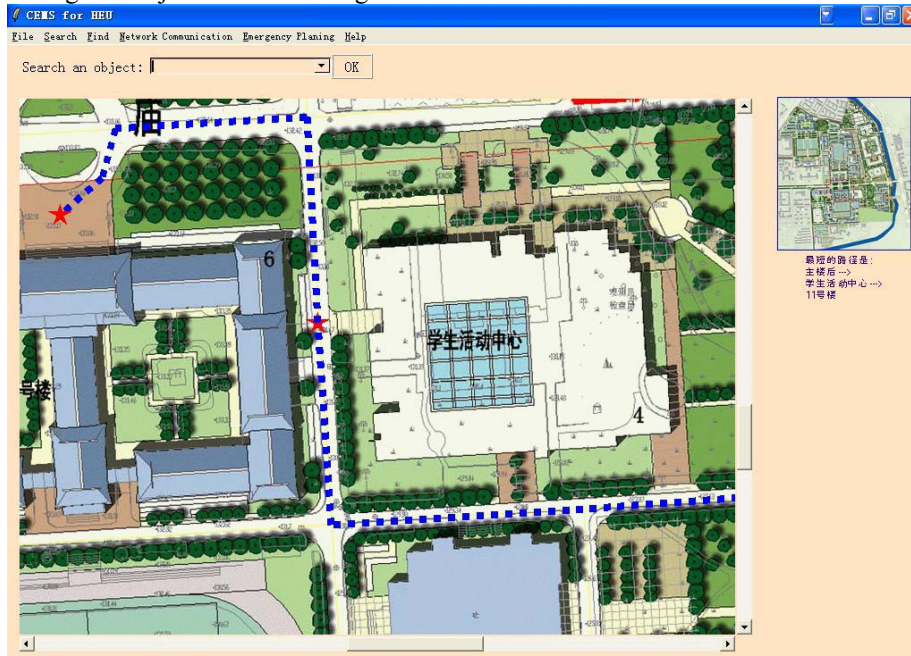


Fig.8. Shortest path from manager to emergency spot

After a comprehensive analysis of this emergency, commands are sent to clients all over the campus to take actions.

5 Conclusion

A new way of designing campus emergency management system (CEMS) is discussed in this paper. It is proved that the system for Harbin Engineering University is fast, stable and efficient.

Compared to other CEMS based on commercial GIS software like ArcInfo, the functions of this system is simpler. It is partly because Scilab has no GIS toolbox, and building a perfect GIS toolbox needs a long time. Many basic functions of GIS cannot be achieved by using raster map, such as zoom in and zoom out. Our future work is to build a GIS toolbox for Scilab, supporting vector map. In that way, any application based on GIS toolbox will be more powerful.

References

1. Kevin Johnson: GIS Emergency Management for the University of Redlands. ESRI International User Conference (2003) Paper Submission Paper Number: 1178
2. Hu Baogang , Zhao Xing ,Kang Mengzhen: Introduction to Scientific Computing Software -- SCILAB, Tsinghua Press, Beijing (2003)
3. Brent Welch and Ken Joes: Practical Programming in Tcl and Tk (Electric Edition), Prentice Hall PTR, 200

Scilab/Scicos toolboxes for Telecommunications

Sebastien Mons, Alan Layec*, Abderrazak. Benadji, Tibault Reveyrand, Raphael Sommet, Edouard Ngoya, Raymond Quéré.

XLIM/C2S2 – Université de Limoges, 123 av. A. Thomas, 87060 Limoges, France
*INRIA –Domaine de Voluceau-Rocquencourt, B.P. 105, 78153 Le Chesnay Cedex, France
e-mail : sebastien.mons@xlim.fr

Abstract. Interactions between devices and circuits on one hand and circuits and systems on the other hand are key points of the design of high frequency communications or Radar systems. In order to accurately predict those interactions there is a need for high level tools that are able to cope with different kind of formalisms and equations. Up to now only specialized tools are available for the simulation of devices, circuits and systems individually. A major drawback of these tools is their inability to communicate together. So we begin the integration of a number of tools in the high level Scilab/Scicos in order to develop an integrated and open research tool which will allow to make the various softwares to communicate together. This paper presents an overview of research activities on the simulation, characterization and modeling of High frequency devices, circuits and systems using the Scilab/Scicos Environment.

1 Introduction

The strategy of our lab consists on the use of a development platform for both modelling, simulation and instrumentation [1]. The kernel of this platform was developed under Scilab/Scicos environment. In order to make the kernel communicate with device or circuit simulators a set of so-called system simulation functions have to be added to this kernel. The general organisation of this tool can be viewed in three axes :

First, Links between those functions and the electromagnetic, thermal or physical models are performed through the use of Model Order Reduction techniques in the linear cases or by direct integration of the simulators in Scilab in the non linear cases as it will be seen in section-1. This toolbox integrates circuit simulation, physics based simulation of semiconductors equation as well as the 3D thermal simulation.

Section-2 presents two complementary toolboxes dedicated to system level simulation using scilab/Scicos environment. First one is a system level simulation tool for RF and microwave devices which includes many low pass discrete time representation elements as filters, modulators, demodulators, signal and noise generators, samplers but also some analyzers like spectrum analyzers, constellation and eye diagram scopes, or BER calculators. This toolbox is well suited for digital communications system level simulations. The second toolbox concerns respectively the

implementation of PM's behaviour models in system simulation environments and the integration of commercial high frequency circuit simulation tools using MPI. This will allow the cosimulation of circuits and systems.

This open-source development tool drives also instrumentation algorithms as shown in section 3. Bridges have to be established between measurement test set-ups and the kernel. This is done by integrating GPIB calls in Scilab functions. Toolboxes respectively dedicated to drive a complete Load-Pull measurement setup with pulse capabilities and to generate or measure RF Times envelopes signals are presented here.

Finally, The use of Scilab/Scicos for both characterization and simulation enables an "hardware in the loop" in system level simulations in order to allow an high flexibility for device characterization and modeling processes.

2 Advanced vic simulation

Scilab is an open and free environment for numerical computation. We have chosen this environment for its ability to link dynamically external functions, for its numerous primitives (graphical, linear algebra, eigenvalues...) and for the advantages it gives to link numerous toolboxes including the best linear libraries like MTLMS, MRPMCK, UMFPCK, TMUCS...

1.1 Heterojunction Bipolar Transistor simulation (HBT)

The first toolbox is dedicated to physics-based semiconductor simulation by means of interface functions with our (HBT) simulator. These calling functions are used to fill in the jacobian matrices of the problem. SCILMB is used to describe the problem and to solve it by powerful solvers. Many different parameters can be achieved with this toolbox : [S] parameters but also currents, voltages, carrier densities inside active devices, under large signal conditions [2].

1.2 Pin Diode impedance computation

The second toolbox concerns an improvement of the frequency-dependent impedance calculation for p-i-n diodes. Under forward bias condition, for thin p-i-n diodes occur important recombinations in the heavily doped regions which affect seriously the diode impedance. This effect is taken into account to increase the accuracy of existing models and to extend the validity domain to any I-region thicknesses. SCILMB is used to solve nonlinear system as well as linear systems obtained from ambipolar semiconductor equation and recombination current definition [3].

1.3 Model order reduction

The third toolbox is dedicated to model order reduction of linear differential equation system by means of Ritz vector technique [4]. This technique is a projection technique based on Gram-Schmidt Orthogonalization algorithm. It has been developed to create reduce models from thermal Finite Element Simulation for electrothermal simulation in circuit simulator. Matrices extracted from MNSYS are read in SCILMB thanks to SCILIN toolbox. Sparse format is used. SCILMB linear algebra as well as eigenvalues computation are made in order to derive the reduced model.

2 Advanced circuit-system simulation

A communication chain is constituted of pure analog circuits such as Power Amplifiers (PA), Low Noise Amplifiers (LNA), filters and antenna as well as pure digital circuits such as source and channel coders and decoders. At the interface between those two worlds we find some mixed circuits such as Analog to Digital Converters (ADC) or modulators and Local Oscillators. The Scicos environment allows to describe all those components shown fig-1 in a simple way. The option chosen in our lab has been to integrate models already developed in our laboratory as "C" routines linked with Scicos. Beyond the integration of known models an effort has been done to develop specific scopes used in telecommunications designs [5] such as eye-diagram scope, constellation scope, spectrum analyser, probability densities scope...

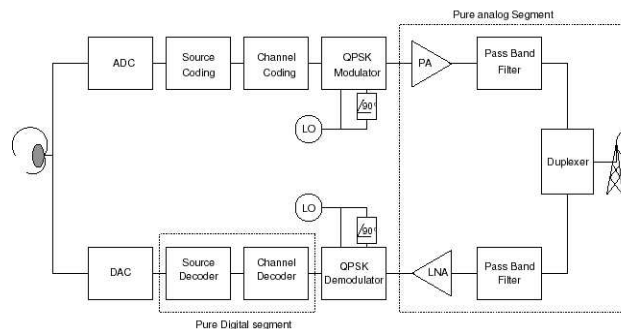


Fig. 1. Simplified architecture of a communication channel

We present a communication toolbox in Scilab dedicated to modelling and simulation of digital transmission. The modelling effort is focused on the improvement of time computation, and on the description in time domain of the constituent components in order to allow future compatibility with models realized in other design environment [6]. A description of a rudimentary communication chain using a DS spread spectrum technique, for a Quadrature Phase Shift Keying (QPSK) transmission shows the returned time domain waveforms in the Scicos environment for a given set of

parameters. The use of the scicos simulate function allows to drive a Scicos diagram from the Scilab prompt, so we are able to realize efficient post-processing operations by directly access to the simulated data. We finally present a laboratory experience where we simulate transmission chains with an heterogeneous network of computers using the Parallel Virtual Machine (PVM) communication toolbox in order to accurate the simulation time of BER estimation.

The « Modnum » toolbox has been developed at our laboratory and is now available [7], its internal architecture is illustrated in fig.-2. It is an open and free built-in Scilab/Scicos toolbox for the modelling and the simulation of communication systems. This toolbox contains several elements : Mlow-level computational routines library (in C language) including common functions of base band communication and tri-state PFD, a set of Scicos palettes including 71 blocks with computational functions and extended interfacing functions (Communication palette for PSK/QMM transmissions, Phase Locked Loop palette for the designing of integer and fractional frequency synthesizer, sinks palette to display frame-based signal and 3d trajectory in Scicos, Sources palette to compute random process, on-linear palette with common nonlinearities encountered in digital processor, ...).

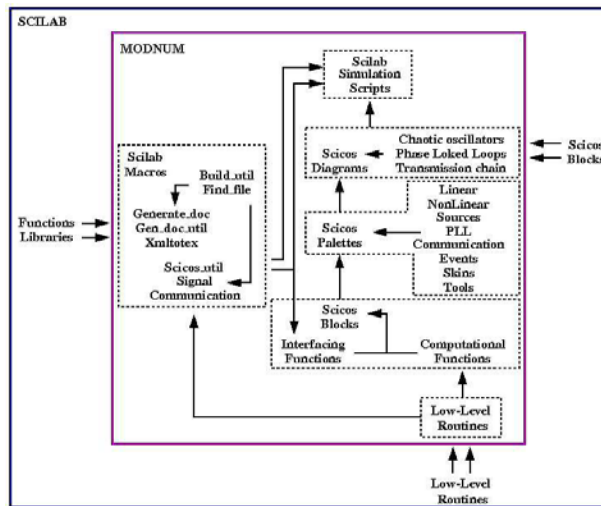


Fig. 2. Toolbox internal architecture

This toolbox proposes blocks, schematics and in-line functions of base-band PSK/QMM modulations in order to build communication chains in Scilab/Scicos environment. Components used to build spread-spectrum communication systems, such Pseudo Noise sequence generators (Quasi-Chaotic, PN and Gold sequence generators) are also included. This toolbox also includes miscellaneous scopes for Scicos, such 3D trajectory Scope and other scopes used for analysis of digital transmission: Eye Diagram Scope, Scattered Diagram Scope,...

The Scicos diagram of the transmission chain is shown in Fig. 3. Blocks that don't have event inputs are direct feedthrough marked. The original binary vector, computed by the random binary generator, goes through all the user's blocks in one

time of simulator process (one event). After being separate in two I/Q channel, scale by the over-sampling block, filter by a Root Raised Cosine impulse response, the vector is added up to a white gaussian noisy samples vector. The receptor is respectively composed by a RRC filter, an adaptive filter, an under-sampling block and an I/Q demodulator. Final binary vector is compared with the original delayed binary vector and a counter block memorize number of errors.

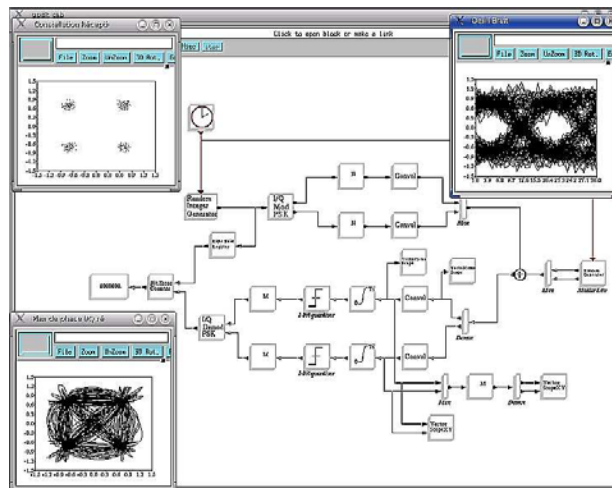


Fig. 3. PSK chain diagram (phase plane, eye & scatter diagram)

The realization of FIR is a critical point : overlap-add method which uses FFT convolution (instead of standard convolution) is used to decrease the execution time. In Scilab we have the scicosim function in order to externally launch the simulation of a pre-compiled Scicos diagram with the Scilab prompt. To launch a parameterized simulation of a Scicos diagram, for example by changing a parameter in the context (such as the variance of the white Gaussian noise random generator or the size of the original binary word), the diagram must be evaluated and compiled before invoking scicosim [8].

The simulation of the BER with this simple chain can tragically become long when we use a single computer. Indeed, to have a good estimation of the BER, we have to simulate a sequence comporting several millions of symbols, according to some values of the variance of the noise source.

Here, the developed idea is to use several computers to simulate a set of ten points of 10e8 symbols each. The laboratory experience is illustrated by the Fig. 4 where we use four computers. M1 computers are arranged with Scilab and PVM and are connected together via an intranet network correctly configured. They have also got interfacing function and compiled computational function of user's blocks and the Scicos diagram.

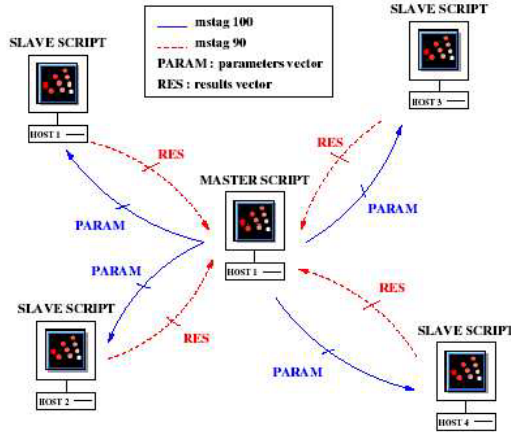


Fig. 4. Laboratory experience

M single master script drive slave scripts that are in charge of executing the simulation. The master script sends parameter vectors, via a first virtual PVM canal, to slave scripts and waits response vectors via a second canal. After realizing the simulation parametrized by the sending vector, slave scripts respond a result vector, and then prevent the master script that they are free to realize another simulation. The resulting BER plot of the preceding experience is shown in Fig. 5. With tree computers instead of one we reduce by a factor two the time calculation.

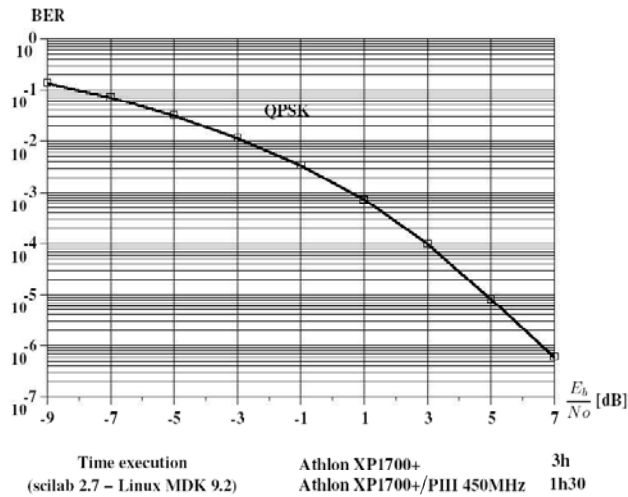


Fig. 5. TEB estimation

The presented transmission chain with the vectorial treatment are able to be completed with other user's blocks. We think now about realizing a spread spectrum transmission chain by adding Gold or chaotic sequences generator to simulate a

CDMM communication system. The Scilab/Scicos environment used with PVM communication toolbox shows that we are in a privileged environment to realize efficient simulation of numerical communication systems

Another key circuit of transmissions channels is the frequency synthesizer. Indeed low phase noise of the Local Oscillator is mandatory for minimizing the BER in communications systems or for increasing the spatial and velocity resolution of Radar Systems. While the simulation of phase noise of microwave oscillators is now a well established topic, there is still a lot of work to be performed in order to include those phase noise characteristics into the overall frequency synthesizer which is constituted of both digital frequency dividers, phase-frequency detectors and analog Voltage Controlled Oscillator and loop filter. Schematics and blocks of integer and fractional frequency synthesizer components (Phase/Frequency Detector, VCO, Delta-Sigma modulators,...) are provided. This kind of circuit is a good example of mixed digital analog circuit. For the modeling of such subsystems the event driven simulation capability of Scicos is very useful [9] as the overall phase noise can be calculated as the fluctuations of zero crossing times around the periodic sequence $\{T, 2T, \dots\}$.

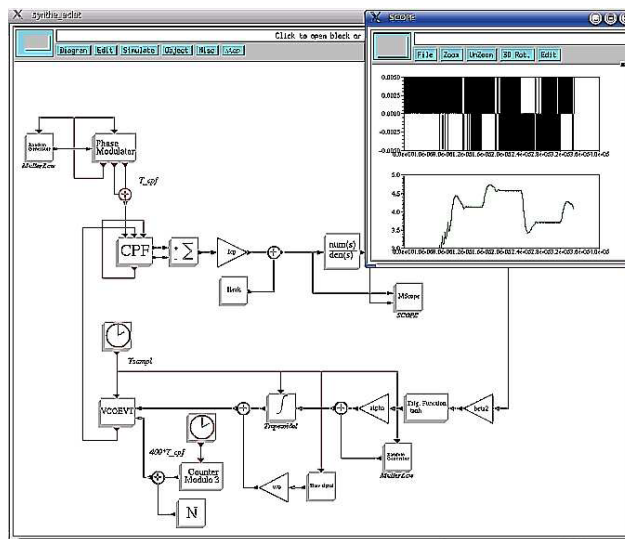


Fig. 6. Frequency synthesiser

This toolbox also focus on the simulation of chaotic systems and gives schematics of simulation of Chua's, Rössler's, Van Der Pol's systems (and others) as shown in fig-7. The models developed in the framework of frequency synthesizer simulation are used in our laboratory to do research on chaotic self synchronizing chaotic generators in the framework of a collaborative action of CNRS for the use of chaotic techniques in cryptography.

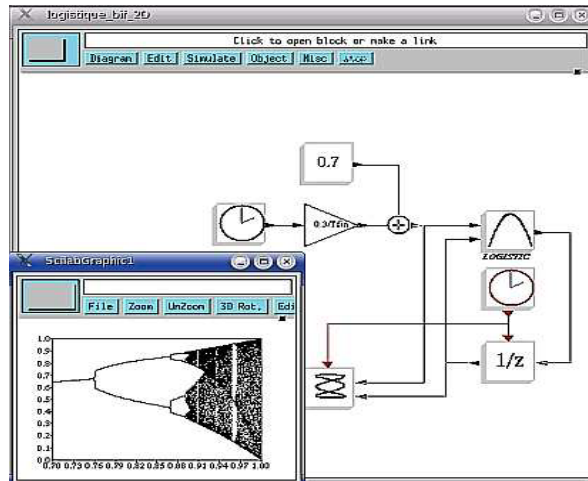


Fig. 7. Chaotic sub-system (bifurcation map)

This toolbox is can be installed indifferently on linux (using gcc/g77) or windows (using MSVC++/lcc-win32) platform. Future improvement of builder script should also build the toolbox for Solaris/HP-Unix platform. Source version and binary version of the toolbox include also a English documentation in html/pdf format. Future improvements are Scilab macros to generate scilab simulation script files of scicos diagram with PVM (for BER estimation), Interfacing functions of low-level computational routines, Spectrum analyzer scope for Scicos...

In the design process of modern communication systems, it is more and more required to predict accurately the performances of the full transceiver chain. That means that the needs in terms of system level simulation tools and of macro-models become important. That signifies also that these tools must be associated with models able to describe the behaviors of the different functionalities (amplifiers, mixers, etc.) with a high level of precision. In this context, implementation of the models based on nonlinear convolution integrals in system environment is a critical point

These models have proven to be accurate for all types of signals and their extraction is affordable in common circuit simulators and measurement benches. In an effort to improve modeling accuracy, we have developed a co-simulation interface that allows the system simulator to access a circuit simulator for each time sample. In our example we have considered co-simulation between Scilab/Scicos™ and Xpedion/Goldengate™ circuit simulator. We will briefly present the numerical implementation of advanced model in system-level environment such as Scilab/Scicos and the cosimulation interface between system simulator and circuit simulator.

The figure 8 shows the flowchart which recalls the advance of our principle of modeling. The files resulting from measurement or simulation must have a standard format (.dat). They contain all information of the extracted data of the model (Volterra or nonlinear impulse response).

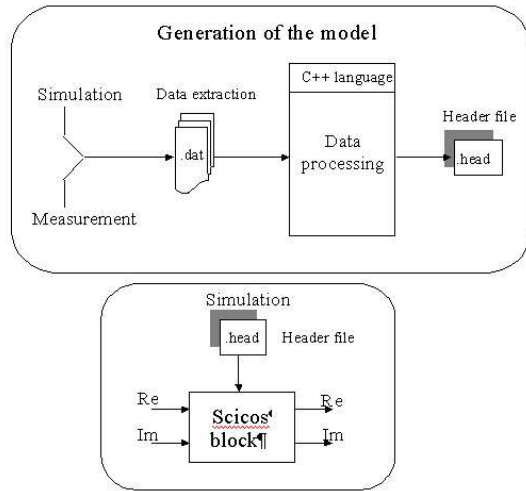


Fig. 8. Principle of modeling

In co-simulation, the system simulator access an Envelope transient [10] circuit simulator to compute the output of the critical part of the system. The interface between the two simulators is based on a parent/child interaction (fig. 9). Scicos acts as the parent simulator, which calls GoldenGate™ at the beginning of the simulation as its child. The co-simulated amplifier block is instantiated into Scicos model thanks to a bloc based on a generic Function, during the simulation, the two simulation kernels are synchronized and share data on their I/O. On the Scicos side a special generic function block acts as the coupling element and it is configured for shared memory communication between Scicos and GoldenGate™, running on a single computer.

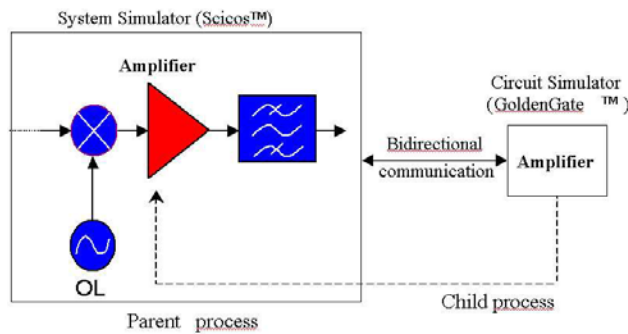


Fig. 9. Principle of modeling

We have run a Scicos simulation using alternatively PM's behavior models developed in our laboratory and the co-simulation interface for a satellite communication chain (Fig. 1). This chain represent a satellite link composed by a Satellite Downlink Transmitter, Downlink Path, and Ground Station Downlink Receiver. The signal

transmitted over the channel is 16-QAM waveform flows with speed of 5 Mbits/s, the carrier frequency of the link equal to 8 GHz. The Volterra kernel and the impulse response of the LNM (operating at 1.96 GHz) placed after the reception antenna were extracted from GoldenGate™ simulator. The simulation performances (CPU) is summarized in table 1. We observe that using the Volterra and nonlinear impulse response models has reduced the CPU time. The performance of nonlinear impulse response model show a simulation speed up sacrificing accuracy.

	Co-simulation	MM/MM MM/PM model	Volterra model	Impulse response model
MCPR	1h. 12 min.	14 sec.	2min. 18 sec.	2 min. 23 sec.

Tabl 1. Simulations performances

Faster and more accurate system level simulations of transmission chain is more and more important in modern communication system design. We have considered here the modeling and simulation of complete transmission chain as we have demonstrated that this can be achieved with better accuracy using an advanced power amplifier model that accounts memory effects, while keeping the simulation time well reasonable.

3 measurement setups

The Scilab environment for circuit-system interactions calculation have to be complemented by measurement interface capabilities. This has lead to the development of a Scilab toolboxes for RF and microwave measurements and analysis. Those toolboxes constitute a complete GUI based environment for measurements. The structure of the measurement software is shown in fig. 10. This approach is an open-source alternative to VEE, Labview and Matlab which enables communications between VISM for Windows (either Mgilent or National Instrument) or Linux (NI-VISM) and Scilab in order to control your instruments. The software includes a layer between VISM and Scilab which is freely downloadable [11] : The GPIB toolbox includes several drivers (written in scilab script) for RF and microwave instrumentation.

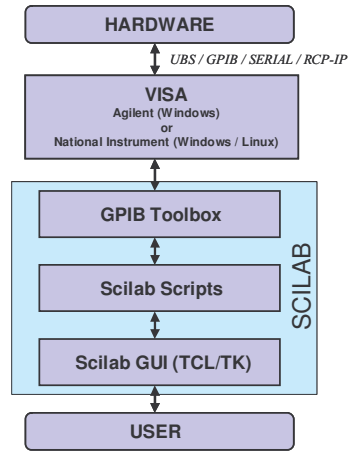


Fig. 10. Structure of the measurement softwares (Scilab)

Three different measurement set-ups are powered with Scilab at our laboratory : a low noise measurement setup [12] ; a time domain envelope load pull setup [13] (as shown in fig. 11) and a VNM based load-pull measurement setup with pulse capabilities in the frequency domain [14] (illustrated on fig. 12).

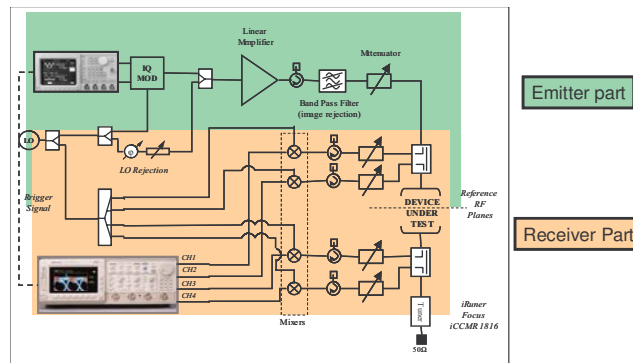


Fig. 11. Time domain envelop load pull measurement setup

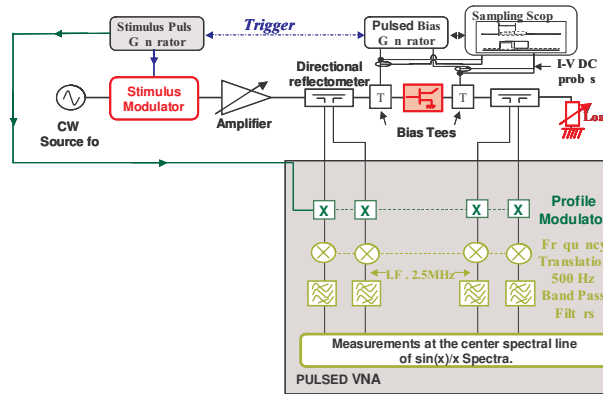


Fig. 12. Frequency domain pulsed load pull measurement setup

If modeling become to complex for a device, one can use either a co-simulation process which enables a link between Scicos and a circuit level simulator (as shown in section 2) or a link in quasi realtime between Scicos and the time domain instrumentation : the “hardware in the loop” process. In this last case, the signal located at the input of the amplifier under simulation is applied to the amplifier under test. This signal is pre-distorted according to the calibration of the emitter part of the bench and sent to the MWG. The calibrated response of the amplifier under test is feed backed to Scicos. Then, the simulation process carry on under the system simulator. This method provides a friendly environment in order to compare easily and simultaneously a device under test and its associated model with a modulated signal excitation.

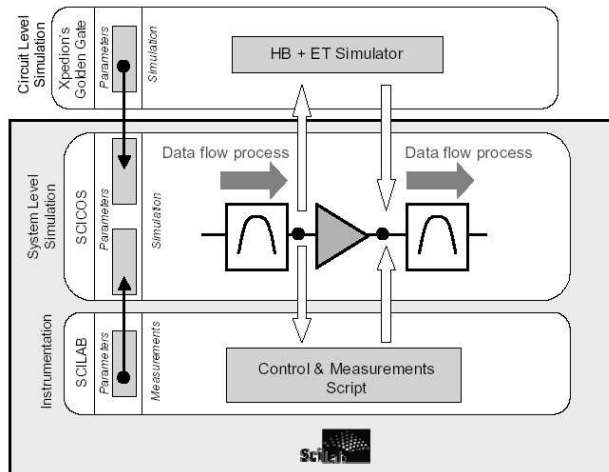


Fig. 13. Dynamic links between simulation tools and instrumentation

The toolboxes for communication are well suited for digital communications system level simulations. Power amplifiers may be included in the chain from their associated

system level behavioral models [15], a circuit level cosimulation principle [16] or a link through measurements [13].

5 Conclusion

This paper has presented an overview of the current software development and strategy at XLIM laboratory in order to federate different tools within a system level hierarchic design framework. Scilab/Scicos has been identified as an efficient tool for high frequency device, circuit and system simulation. Therefore, some Scilab/Scicos toolboxes were presented respectively dedicated to advanced devices, circuits and system simulation. Those toolboxes are open source and freely usable and distributable. To complete this task XLIM/C2S2 has received strong support from the CNES-Toulouse, the Délégation générale à l'Armement (DGM), the CNRS and the Conseil Régional du Limousin. Two PhD students are currently working within this framework.

R e f e r e n c e s

1. Sommet, R., Quere, R., Ngoya, E., Mons, S., Nallatamby, J.-C., Reveyrand, T., Mallet, M.: Overview of research activities on the simulation of high frequency devices, circuits and systems using the Scilab/Scicos environment. 1st Scilab International Conferences, Rocquencourt (2004)
2. Riah, Z., Sommet, R., Nallatamby, J.-C., Prigent, M., Obregon, J.: Coherent Tools for Physics-Based Simulation and Characterization of Noise in Semiconductor devices oriented to Nonlinear Microwave Circuit CMD. Noise in Devices and Circuits II, SPIE, (2004) 507-518.
3. Submitted in IEEE Microwave and Wireless Components Letters
4. Sommet, R., Chang, C., Dueme, P., Quéré, R.: Electrothermal model of transistors based on finite element analysis for Radar applications. Thermal and Thermomechanical Phenomena in Electronic Systems IThERM (2004).
5. Layec, M., Quere, R., Nallatamby, J.-C., Mons, S., Guittard, J.: Modelling and simulation of a communication chain in the Scilab/Scicos environment. Conference Scilab, Rocquencourt (2004)
6. Bennadji, M., Soury, M., Ngoya, E.: Implementation of dynamic Volterra model in system Malysis. International Workshop on Electronics and System Malysis, Bilbao (2004) 15-18.
7. Layec, M.: available at http://www.lesia.insa-toulouse.fr/nouveau/transchaos/mod_num_web/web/fr/eg.htm
8. *De la compilation à la simulation sous scicos*, Scilab group. [Online]. Mavailable : <http://www.saphir-control.org>.
9. Layec, M., Quere, R., Nallatamby, J.-C., Mons, S., Guittard, J.: Behavioral model of integer-N frequency synthesizer. In International Workshop on Electronics and System Malysis, Bilbao (2004)
10. Ngoya, E., Larcheveque, R.: Envelope Transient Malysis: MNew Method for the Transient and Steady State Malysis of Microwave Communication Circuits and Systems. IEEE MTT Symposium Digest (1996) 1365-1368.
11. Reveyrand, T.: available <http://membres.lycos.fr/treveyrand/>

12. Lisboa de Souza, M, Nallatamby, J.-C., Prigent, M.: Low-Frequency Noise Measurements of Bipolar Devices Under High DC Current Density: Whether Transimpedance or Voltage Amplifiers, accepted at European Microwave Week, Manchester (2006)
13. Reveyrand, T., Barataud, D., Neveux, G., Nebus, J.-M., Ngoya, E., Teyssier, J.-P., Quere, R.: Overview on System Level Simulation Environment for Characterization, Modeling and Simulation of RF and Microwave devices. IEEE InMMIC (2006)
14. Gasselting, T., Barataud, D., Mons, S., Nebus, J.-M., Villotte, J.-P., Obregon, J., Quere, R.: Hot small-signal S-parameter measurements of power transistors operating under large-signal conditions in a load-pull environment for the study of nonlinear parametric interactions. Microwave Theory and Techniques, IEEE Transactions on, Vol. 52. (2004) 805-812
15. Soury, M, Ngoya, E., Rousset, J.: Behavioral Modeling of RF and microwave circuit blocs for hierarchical simulation of modern transceivers. IEEE MTT-S International Microwave Symposium (2005)
16. Benadji, M, Soury, M, Ngoya, E., Quere, R.: Implementation of Volterra Model In System Simulation Environments and Rf Circuit/System Co-simulation. European Microwave Week, Manchester (2006)

Speech Analysis, Synthesis and Recognition Toolbox for Scilab --- SciSpeech

Cong Wu, Xi-Da Chen, Hui-Ting Su, Wei-Jie Liu, Long-Ling Luo, Kun Yang,

Jing Liu, Tie-Zheng Lin, Rong-Ping Zhuang, Jun Cai

Computer Science Department of Xiamen University, Fujian, P.R.C
xmuscilab@gmail.com

Abstract. Speech is an important method in human and computer communication. We develop SciSpeech, a speech analysis, synthesis and recognition toolbox, for Scilab, which has great power in scientific computing. This toolbox provides powerful speech analysis functions, including time analysis, frequency analysis, cepstral analysis and WRLS-VFF analysis. It uses the PSOLA algorithm to adjust the prosodics, which can adjust the pitch and energy of speech signal freely while maintaining the basic audio elements of the original speech. This toolbox also includes a practical speech recognition system, which can be applied in Voice Dial System. Besides, It has a friendly GUI. However SciSpeech has a high recognition rate while offering effective analysis functions and we hope it can help to stimulate the idea of using Scilab in the domain of human and computer communication.

1 Introduction

With the popularization of wireless communicating equipments, developing a human and machine speech communication system in the mobile phone, PAD and robot is becoming applicable. In this situation, more and more company focus on the design of the human and machine speech communication system. There have been some famous voice systems being developed such as the "voicecommand" which is designed by the Microsoft, the "IBM Viavoice" which can operate personal computer by speech command and the speech synthesis system "interphonic" which is developed by USTC iFLYTEK CO. However many users and researchers are not able to use the above systems due to their expensive price, close source and the lack of providing speech analysis functions. So we have developed Scispeech, which is an open sourced speech analysis, synthesis and recognition toolbox especially designed for Scilab. The SciSpeech toolbox can be easily used or extended by scholars who is doing research in the realm of speech analysis, synthesise and recognition.

SciSpeech provides various analysis functions including time analysis, frequency analysis, cepstral analysis and WRLS-VFF analysis, which provides great convenience for the both developers and users, while the famous software MATLAB voicebox toolbox only offers few speech analysis functions,

SciSpeech use the PSOLA algorithm to supply the prosodics adjusting function in

the module of speech synthesis. And this complex function is open sourced so that it can be easily applied and modified in a speech synthesis system.

SciSpeech also includes a practical speech recognition system, which can recognize speech command within a short time and the telephone number recognition ratio is very good. In fact it contains our open source versions of “speech command” and “speech dialing” functions provided by the "IBM Viavoice" that developed by Microsoft, which is the fundamental functions of IBM Viavoice.Speech Porcessing

2.1 Windowing

The *time-windowing* process is illustrated in Fig. 1.

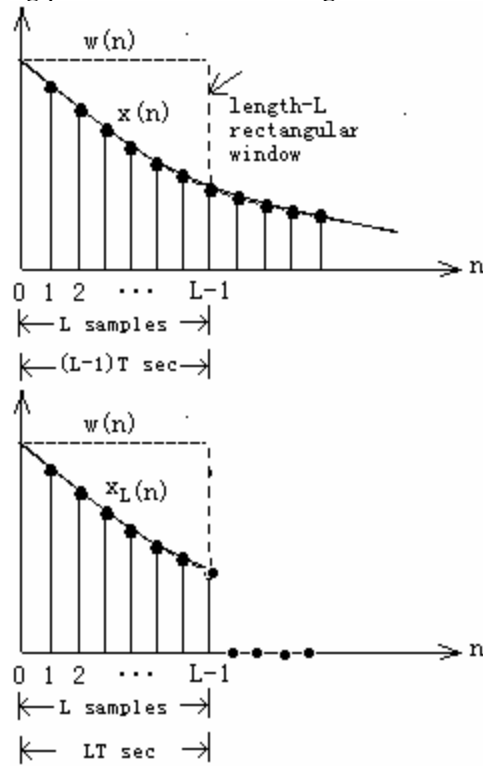


Fig. 2. Time windowing

In terms of the time samples $x(nT)$, the original sampled spectrum $\hat{X}(f)$ and its time-windowed version $\hat{X}_L(f)$ are given by:

$$\hat{X}(f) = \sum_{n=-\infty}^{\infty} x(nT)e^{-2\pi ifnT} \tag{1}$$

$$\hat{X}_L(f) = \sum_{n=0}^{L-1} x(nT) e^{-2\pi jfnT} \quad (2)$$

As seen in **Fig. 3**, the duration of the windowed data record from the time sample at $n=0$ to the sample at $n=L-1$ is $(L-1)T$ seconds, where T is the sampling time interval $T = 1/f_s$. Because each sample lasts for T seconds, the last sample will last until time LT . Therefore, we may take the duration of the data record to be $T_L = LT$.

The windowed signal may be thought of as an infinite signal which is zero outside the range of the window and agrees with the original one within the window. To express this mathematically, we define the *rectangular window* of length L :

$$w(n) = \begin{cases} 1, & \text{if } 0 \leq n \leq L-1 \\ 0, & \text{otherwise} \end{cases} \quad (3)$$

Then, define the windowed signal as follows:

$$x_L(n) = x(n)w(n) = \begin{cases} x(n), & \text{if } 0 \leq n \leq L-1 \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

The multiplication by $w(n)$ ensures that $x_L(n)$ vanish outside the window. Equation(1) and Equation(2) can now be expressed more simply in the form:

$$\begin{aligned} X(\omega) &= \sum_{n=-\infty}^{\infty} x(n) e^{-j\omega n} \\ X_L(\omega) &= \sum_{n=0}^{L-1} x(n) e^{-j\omega n} = \sum_{n=-\infty}^{\infty} x_L(n) e^{-j\omega n} \end{aligned} \quad (5)$$

where $\omega = 2\pi f / f_s$. Thus, $X_L(\omega)$ is the DTFT of the windowed signal $x_L(n)$ and is computable for any desired value of ω .

2.2 Energy and ZCR

One of the most common short-time analysis methods is the short-time average energy of a signal. If $w(n)$ is the data window for $0 \leq n \leq N-1$ and is zero otherwise, and $x(n)$ is the data record, then the short-time average energy is

$$E_n = \sum_m [x(m)w(n-m)]^2 = \sum_m x^2(m)h(n-m) \quad (6)$$

where $h(n) = w^2(n)$. Thus, $h(n)$ is a window (filter) that affects the short-time energy function. If this window is very short, then its bandwidth is large and there is

little filtering of the data by the window. If the window is very long, then the window bandwidth is narrow, and there is a great deal of filtering of the data. We select the window parameters (or type of window) to provide “good” data smoothing. The type of window selected can depend on the application or task but for speech, the window used is usually the hamming window.

The short-time average zero-crossing rate is used to estimate the frequency of voicing. Suppose we are given a sampled sine wave of frequency F_0 that is sampled at rate F_s . The number of samples in a period is $N = F_s / F_0$. For a periodic signal, we have two zero-crossings per period, which is $2F_0 / T$, where $T = 1 / F_s$. To estimate F_0 , count the number of zero-crossings in the sampled data over some time interval, T_1 . Then T_1 is the number of samples in T_1 divided by F_s , that is $T_1 = N_1 / F_s$ or $N_1 = F_s T_1$. The average number of zero-crossings per sample is $2F_0 / F_s$, thus $F_0 = (\text{average number of zero-crossings}) / (2T_1)$. So the zero-crossing rate algorithm is the following.

- Count the number of samples in the data between sign changes. This interval (number of samples) represents 1/2 the period or $\frac{1}{2F_0}$.
- The average number of samples between sign changes is N_1 .
- $N_1 / F_s = \frac{1}{2F_0}$
- $F_0 = \frac{F_s}{2N_1}$

The short-time energy and zero-crossing functions are useful for estimating word and phoneme boundaries.

2.3 Autocorrelation

Various applications call for autocorrelation estimates, including data modeling such as linear prediction and power spectral estimation.

The biased autocorrelation estimate, is expressed as

$$\hat{R}_{XX}(k) = \frac{1}{N} \sum_{i=0}^{N-1-|k|} X(i+|k|)X(i), \quad \text{for } |k| < N \quad (7)$$

where $X(i)$ is a zero mean Gaussian random sequence. Although this estimate is biased, It is asymptotically unbiased as N increases for k fixed. And the variance is approximately

$$VAR[\hat{R}_{xx}(k)] = \frac{1}{N^2} \sum_{i=-(N-1-k)}^{N-1-k} \{(N-|i|-|k|)(R_{xx}^2(i) + R_{xx}(i+|k|)R_{xx}(i-|k|))\} \quad (8)$$

This estimate is also consistent as N increases, provided k is fixed. It can also be shown that this estimate does have its largest value at the origin, which in correlation matrix terms means that this estimate is positive definite.

2.4 Linear Prediction

Linear prediction analysis of speech is historically one of the most important speech analysis techniques. The basis is the source-filter model where the filter is constrained to be an all-pole linear filter. This amounts to performing a linear prediction of the

next sample as a weighted sum of past samples: $\hat{s}_n = \sum_{i=1}^p a_i s_{n-i}$

Given N samples of speech, we would like to compute estimates to a_i that result in the best fit. One reasonable way to define “best fit” is in terms of mean squared error. These can also be regarded as “most probable” parameters if it is assumed the distribution of errors is Gaussian and a priori there were no restrictions on the values of a_i .

The error at any time, e_n , is:

$$e_n = s_n - \hat{s}_n = s_n - \sum_{i=1}^p a_i s_{n-i} \quad (9)$$

Hence the summed squared error, E , over a finite window of length N is:

$$E = \sum_{n=0}^{N-1} e_n^2 = \sum_{n=0}^{N-1} (s_n - \sum_{k=1}^p a_k s_{n-k})^2 \quad (10)$$

The minimum of E occurs when the derivative is zero with respect to each of the parameters, a_k . As can be seen from equation (11) the value of E is quadratic in each of the a_k therefore there is a single solution. Very large positive or negative values of a_k must lead to poor prediction and hence the solution to $\partial E / \partial a_k = 0$ must be a minimum. Hence differentiating equation(12) with respect to a_k and setting equal to zero gives the set of p equations:

$$\frac{\partial E}{\partial a_j} = - \sum_{n=0}^{N-1} (2(s_n - \sum_{k=1}^p a_k s_{n-k}) s_{n-j}) = 0 \quad (13)$$

then we can get :

$$\frac{\partial E}{\partial a_j} = -2 \sum_{n=0}^{N-1} s_n s_{n-j} + 2 \sum_{n=0}^{N-1} \sum_{k=1}^p a_k s_{n-k} s_{n-j} \quad (12)$$

rearranging the equation gives

$$\sum_{n=0}^{N-1} s_n s_{n-j} = \sum_{k=1}^p a_k \sum_{n=0}^{N-1} s_{n-k} s_{n-j} \quad (13)$$

Define the covariance matrix, r with elements $r_{i,k} : r_{i,k} = \sum_{n=0}^{N-1} s_{n-i} s_{n-k}$. Now we can

write the following equation as: $r_{i,0} = \sum_{k=1}^p r_{i,k} a_k$ or in matrix form:

$$\begin{bmatrix} r_{1,0} \\ r_{2,0} \\ \dots \\ r_{p,0} \end{bmatrix} = \begin{bmatrix} r_{1,1} & r_{1,2} & \dots & r_{1,p} \\ r_{2,1} & r_{2,2} & \dots & r_{2,p} \\ \dots & \dots & \dots & \dots \\ r_{p,1} & r_{p,2} & \dots & r_{p,p} \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \dots \\ a_p \end{bmatrix} \quad (14)$$

or simply $r_0 = r_a$.

Hence the *Covariance method* solution is obtained by matrix inverse: $a = r^{-1} r_0$.

Note that:

- This method can be used on much smaller sample sequences (as end discontinuities are less of a problem)
- There is no need to window the data while calculating and it has high precision
- There is no guarantee of stability (but you can check for instability)

Fig. 2. shows the result for the *Covariance method*.

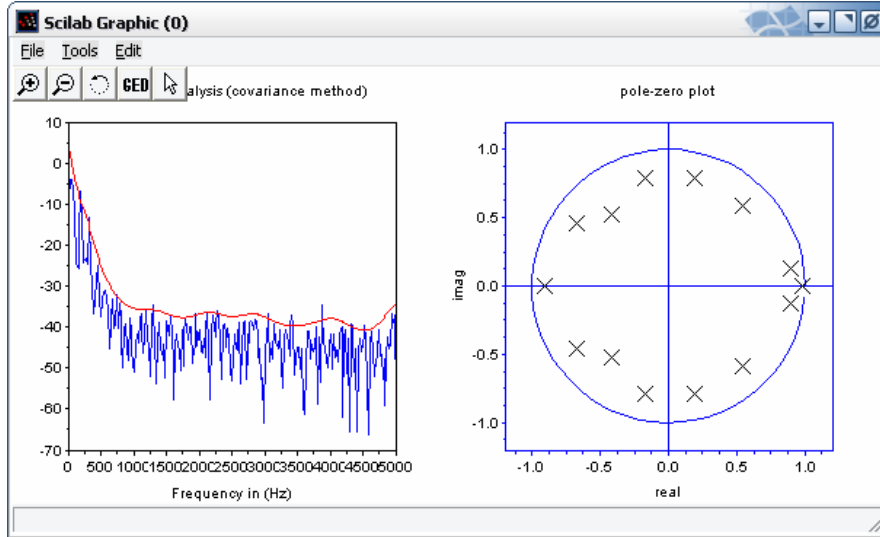


Fig. 2. Linear Prediction analysis by using Covariance method

Another linear prediction method is called Burg method. In contrast to the *Covariance method*, which estimate the LP parameters directly, the Burg method estimates the reflection coefficients first, and then uses the Levinson recursion to obtain the LP parameter estimates. The reflection coefficient estimates are obtained by minimizing estimates of the prediction error power for different order predictors in a recursive manner.

$$\tilde{E}^{(i)} = \sum_{n=0}^{N-1} \{ [f^{(i)}(n)]^2 + [b^{(i)}(n)]^2 \} \quad (15)$$

make $\frac{\partial \tilde{E}^{(i)}}{\partial k_i} = 0$, then we have:

$$\begin{aligned} \frac{\partial \tilde{E}^{(i)}}{\partial k_i} &= -2 \sum_{n=0}^{N-1} \{ f^{(i-1)}(n) - k_i b^{(i-1)}(n-1) \} b^{(i-1)}(n-1) \\ &\quad - 2 \sum_{n=0}^{N-1} \{ b^{(i-1)}(n-1) - k_i f^{(i-1)}(n) \} f^{(i-1)}(n) = 0 \end{aligned} \quad (16)$$

→

$$k_i = \frac{2 \sum_{n=0}^{N-1} [f^{(i-1)}(n)b^{(i-1)}(n-1)]}{\sum_{n=0}^{N-1} [f^{(i-1)}(n)]^2 + \sum_{n=0}^{N-1} [b^{(i-1)}(n)]^2} \quad (17)$$

Fig. 3. shows the result for the burg method.

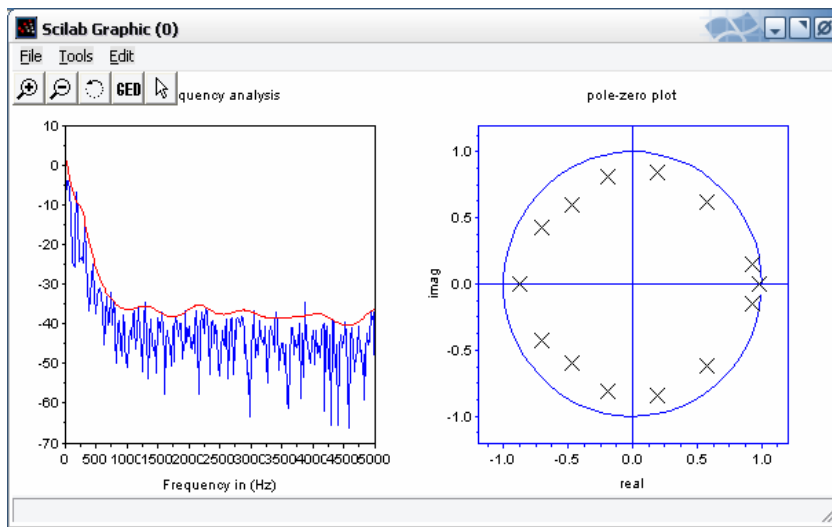


Fig. 4. Linear Prediction analysis by using Burg method

3 Speech Synthesis

There are mainly three steps to implement speech synthesis by PSOLA.

(1).Pitch synchronous analysis:

Set the synchronized mark of the unit of speech synthesis as the center, choose a proper window length and window the unit of synthesized speech, then we can get a group of short-time signal $x_m(n)$:

$$x_m(n) = h_m(t_m - n)x(n) \quad (18)$$

where t_m is the pitch mark, the default value of $h_m(n)$ is Hamming window. The window length is often chose to be 2~4 times as a pitch period.

(2).Pitch synchronous modification:

According to the TD-PSOLA method, short-time synthesize signal is a copy of its short-time analysis signal. If the short-time analysis signal is $x(t_a(s), n)$, then the short-time synthesize signal is

$$x(t_s(s), n) = x(t_a(s), n) \tag{19}$$

where $t_a(s)$ is analysis pitch mark and $t_s(s)$ is synthesize pitch mark.

(3).Pitch synchronous synthesize:

Pitch synchronous synthesize is implemented by superposition of short-time synthesize. There are a lot of method about superposition of short-time synthesize, we use the Least-Square Overlap-Added Scheme. The final synthesize signal is:

$$\bar{x}(n) = \sum_q a_q \bar{x}_q(n) \bar{h}_q(\bar{t}_q - n) / \sum_q \bar{h}_q^2(\bar{t}_q - n) \tag{20}$$

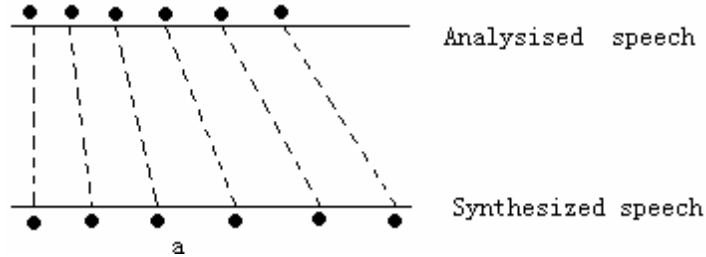
it can also be expressed as:

$$\bar{x}(n) = \sum_q a_q \bar{x}_q(n) / \sum_q \bar{h}_q^2(\bar{t}_q - n) \tag{21}$$

If we set $a_q = 1$, then we have:

$$\bar{x}(n) = \sum_q \bar{x}_q(n) \tag{22}$$

Using the Equation (21) and (22), we can extend or compress the relatively distance between the pitch synchronized mark (t_m) of the original speech, and finally we can have another pitch synchronized mark (t_q) of the synthesized speech. The POSLA method can be illustrated in Fig. 4.



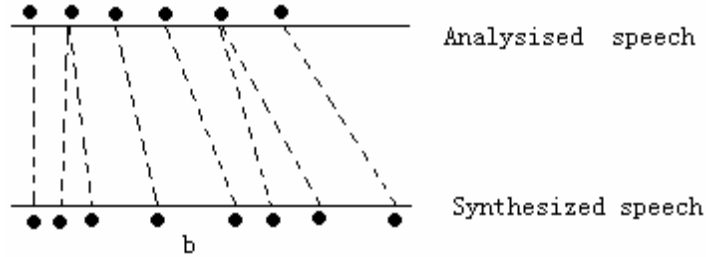


Fig.4.

- (a) Pitch frequency was decreased
- (b) Speech was extended but the pitch frequency remains the same

4 Speech Recognition

4.1 MFCC

The human hearing system is a special nonlinear system. The sensitiveness it response to different signals is different, primarily a logarithm. In recent years, a parameter called Mel-scaled Cepstrum Coefficients, which can make use of the specialty of human ears effectively, has been applied widely, In short, MFCC. Mass researches indicate that MFCC parameter can improve the discriminating ability of the system more efficiently.

MFCC parameter is calculated using “bark” as its frequency unit. The translation formula of bark and the linear frequency is

$$f_{mel} = 2595 \log_{10} \left(1 + \frac{f}{700} \right) \tag{23}$$

MFCC parameter is calculated according to frames. First we get the power spectrum $S(n)$ of the signal frame using FFT and translate it into power spectrum of the MEL frequency. This need to set several band-pass filters in the area of vocal frequency spectrum.

$$H_m(n), \quad m = 0, 1, \dots, M - 1, \quad n = 0, 1, \dots, \frac{N}{2} - 1$$

M is the number of filters, normally set to 24. N is the number of vocal signal points. Always set to 256 for calculating convenience. The frequency area of filters is a simple triangle with a center frequency f_m . They distribute uniformity in the MEL frequency axis, while the distance between f_m changes in proportion with m in the linear frequency axis. And there is a linear area between f_m and f in the low-frequency area. The parameters of the band-pass filter are figured out before calculating the MFCC parameters.

MFCC parameters are always calculated through the following processes.

- (1) First determine the number of sampling points in every frame's vocal sampling sequence, Our system set $N=256$ points. Set pre-emphasize and then DFFT to each frame of $s(n)$ to get $s'(n)$. Mod $s'(n)$ and square the result, then we obtained the discrete power spectrum $s(n)$.
- (2) Calculate the products of $s(n)$ and $H_m(n)$ in every discrete frequency point and add them together. Then we obtain M parameters of $P_m, m=0,1,2,M-1$.
- (3) Calculate the natural logarithm of P_m as $L_m, m=0,1,\dots,M-1$.
- (4) Calculate the discrete cosine transition of L_0, L_1, \dots, L_{M-1} , we have the result $D_m, m=0,1,\dots,M-1$.
- (5) Abandon D_0 representing the direct current part. Use D_1, D_2, \dots, D_k as MFCC parameters. here $K=12$.

4.2 DTW

Speech is a time-dependent process. Several utterances of the same word are likely to have different durations, and utterances of the same word with the same duration will differ in the middle, due to different parts of the words being spoken at different rates. To obtain a global distance between two speech patterns (represented as a sequence of vectors) a time alignment must be performed.

This problem is illustrated in figure 5, in which a "time-time" matrix is used to visualize the alignment. As with all the time alignment examples the reference pattern (template) goes up the side and the input pattern goes along the bottom. In this illustration the input "SsPEEhH" is a 'noisy' version of the template "SPEECH". The idea is that 'h' is a closer match to 'H' compared with anything else in the template. The input "SsPEEhH" will be matched against all templates in the system's repository. The best matching template is the one for which there is the lowest distance path aligning the input pattern to the template. A simple global distance score for a path is simply the sum of local distances that go to make up the path.

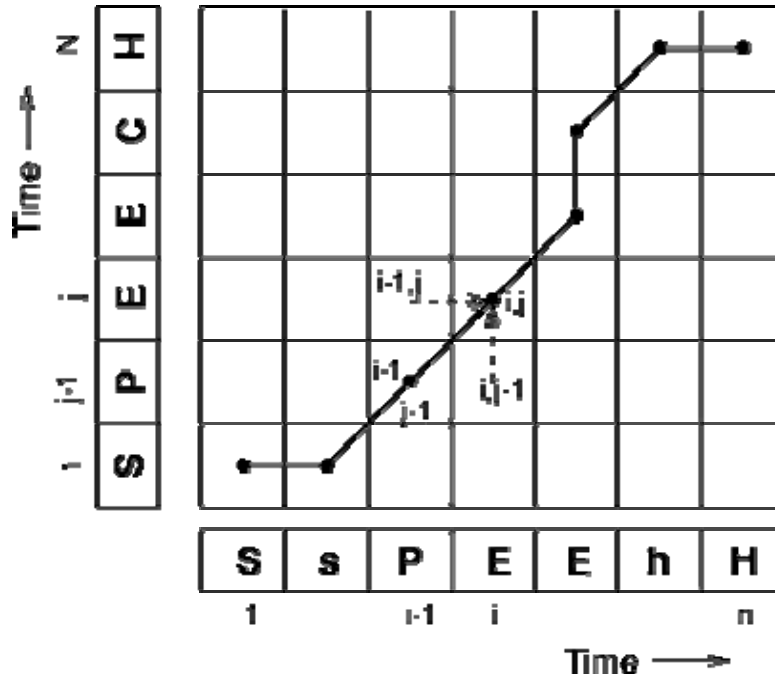


Fig. 5. Illustration of a time alignment path between a template pattern "SPEECH" and a noisy input "SsPEEhH"

In order to find the best-matching (= lowest global distance) path between an input and a template, we could evaluate all possible paths - but this is extremely inefficient as the number of possible paths is exponential in the length of the input. Instead, we will consider what constraints there are on the matching process (or that we can impose on that process) and use those constraints to come up with an efficient algorithm. The constraints we shall impose are straightforward and not very restrictive:

- Matching paths cannot go backwards in time;
- Every frame in the input must be used in a matching path;
- Local distance scores are combined by adding to give a global distance.

For now we will say that every frame in the template and the input must be used in a matching path. This means that if we take a point (i, j) in the time-time matrix (where i indexes the input pattern frame, j the template frame), then previous point must have been $(i-1, j-1)$, $(i-1, j)$ or $(i, j-1)$. The key idea in dynamic programming is that at point (i, j) we just continue with the lowest distance path from $(i-1, j-1)$, $(i-1, j)$ or $(i, j-1)$.

This algorithm is known as Dynamic Time Warping (DTW). DP is guaranteed to find the lowest distance path through the matrix, while minimizing the amount of computation. The DP algorithm operates in a time-synchronous manner: each column

of the time-time matrix is considered in succession (equivalent to processing the input frame-by-frame) so that, for a template of length N , the maximum number of paths being considered at any time is N .

If $D(i,j)$ is the global distance up to (i,j) and the local distance at (i,j) is given by $d(i,j)$

$$D(i,j)=\min[D(i-1,j-1), D(i-1,j), D(i,j-1)]+d(i,j) \quad (24)$$

Given that $D(1,1) = d(1,1)$ (this is the initial condition), we have the basis for an efficient recursive algorithm for computing $D(i,j)$. The final global distance $D(n,N)$ gives us the overall matching score of the template with the input. The input word is then recognized as the word corresponding to the template with the lowest matching score. (Note that N will be different for each template.)

For basic speech recognition DP has a small memory requirement, the only storage required by the search (as distinct from the templates) is an array that holds a single column of the time-time matrix.

The pseudocode for this process is:

```

calculate first column (predCol)

for i=1 to number of input feature vectors
    curCol[0]=local cost at (i,0) + bal cost at (i-1,0)
    for j=1 to number of template feature vectors
        curCol[j]=local cost at (i,j) + minimum of global
costs at (i-1,j), (i-1,j-1) or (i,j-1)
    end
    predCol=curCol
end

minimum global cost is value in curCol[number of
template feature vectors]
```

In order to effect a recogniser, the input pattern is taken and the above process is repeated for each template file. The template file which gives the lowest global cost is the word estimate.

5 Conclusion

In this paper, all the powerful speech toolbox has been presented. It can run on Linux and Windows platform without any modification. The result of the analysis is precise. The recognition ratio is very good in a relatively short time.

However, there are still a lot of works that can be done: 1) Write more useful functions focused on synthesis, develop a text-to-sound system. 2) Speed up the process of recognition. 3) Develop a Speaker Independent Recognition system.

References

1. Baogang Hu, Xing Zhao and Mengzhen Kang: Introduction to Scientific Computing Software—Scilab. Tsinghua Press, Beijing (2003)
2. John Holmes and Wendy Holmes: Speech Synthesis and Recognition. Second Edition.
3. Sophocles J. Orfanidis: Introduction to Signal Processing. Tsinghua Press, Beijing (2003)
4. Jan P.H. van Santen, Richard W. Sproat, Joseph P. Olive, Julia Hirschberg: Progress in Speech Synthesis. China Machine Press, Beijing (2005)
5. Tianren Yao: Digital Speech Processing. Hust University Press (1992)
6. Jiqing Han, Lei Zhang and Tieran Zheng: Speech Signal Processing. Tsinghua Press, Beijing (2004)
7. Simon Haykin and Barry Van Veen: Signals and Systems. Second Edition. Publishing House of Electronics Industry, Beijing (2004)
8. Li Zhao: Speech Signal Processing. China Machine Press, Beijing (2003)

Telecommunication Toolbox Design Based on Scilab

Yongshun Chen, Yanhua Zhang

College of Electronic Information and Control Engineering,
Beijing University of Technology, China, 100022
yongshunchen@gmail.com

Abstract. Simulation is a controllable, harmless, reusable, environmental-restriction-free, and fully-functional method of experiment, which can help effectively research in early system development, and so does in system examination, detection, and evaluation. Certain aspects of simulation require to be carried out by simulation software. Communication simulation is a cross-subject action, which involves multiple theories from different fields. Hence, based on integrating simulation skill and communication theory, with plentiful research on free software technique, this research is thereafter aimed at providing an simulation research platform for complicated system analysis and design, by using Scilab's visible simulation environment Scicos, calling its internal functions and mathematics cells, researching and integrating a modern communication simulation toolbox, which made up of sub models libraries upon modern communication theory and the characteristics of Scilab, therefore provides with an open platform for materializing an open, refillable, and renewable modern communication toolbox.

Keywords: Modeling, Models Libraries, Simulation, Scilab, Scicos

1 Introduction

Along with the rapid development and the increase of requirements in IT industry, the traditional communication principles can no longer meet modern communication requirements. In the trend of 3-C union and 3-net union, modern communication technique is developing rapidly. Hence, under the construction of future Infobahn, modern telecom technique doubtlessly plays a key role, with more mature functionalities, and wider range of applications. Because of the fact that one of the

main parts in modern communication technique is to emulate arithmetic with software, communication toolbox, the basis of system simulation, therefore becomes a significant component in communication simulation system.

1.1 Brief Introduction of Software Based Simulation Technique

Simulation software is a kind of specific software. Its development has ceaselessly been aiming at improving its task-oriented and user-oriented ability in model description, and enhancing its functionalities in model establishment, detection, design, and evaluation. In fact, simulation software is not only a reflection of the simulation method, but the embodiment of arts and sciences of software simulation technique as well. It comprises of all programs and documentations which are designed for modeling, detecting, executing, and result analyzing on the simulation computer. Starting from the 80s, by the combination of modeling technique with AI (Artificial Intelligence), Knowledge Engineer, graphic and image technique, and automatic program design, different sorts of simulation supported systems and simulation environments arises. Its target is to provide users with a better consistent interface for simulation research, so that more support and enhancement can be given.

1.1.1 Development and Relevant Software of Simulation Technique. Since 1995, when the first digital simulation software came in to being, simulation software development can be orderly divided into following 6 parts:

- a* General programming language
- b* Emulate program package and junior simulation language
- c* Senior completed commercial simulation language
- d* Incorporated (partly intelligential) modeling and simulation environment
- e* Intelligential modeling and simulation environment
- f* Complex simulation environment which supports distribute interactive simulation

Till now, there have already been many program languages and simulation languages throughout the world. Among them, the commonly used for modern computer simulation are:

- Programming language: FORTRAN, BASIC, C VB, VC++, Java, etc.
- Analog system simulation language: GSMP III, ACSL, MPSE, YFSIM, etc.
- Discrete events system simulation language: GPSS, SIMSCRIPT, SIMULA, etc.

Analog-Discrete complex system simulation language: GASP IV, SLAM, IHSL, SIMAN, etc.

Different sorts of specialty simulation language: VRML, UML, etc.

Traditional simulation language is narrowly focused on system model description, namely solving system model computationally. Although providing API (Application Programming Interface) for simulation results analyzing, their functionalities remain limited, only processing data in simple ways and output in 2 dimensions. Along with AI improvement and development, simulation software for the next stage is elicited: encompass the knowledge of modeling and simulation methods with that of specific field together, to construct a diversity of expert system; induce machine learning ability and object-oriented knowledge processing ability to software; apply techniques of natural language, image and graphic, etc. to the new man-machine intellectual API, so that a highly intelligential modeling and simulation environment, which is user-oriented, detection-oriented, problem-oriented, and object-oriented, can be created. In this way, a new generation of expert simulation software system will renew the classical one.

1.1.2 Usage of Simulation Technique. Simulation is a means of detection which is controllable, ruinous-free, repeatable, external-restriction-free, and fully functional. Simulation technique is an integrated technique which basing on control theory, similarity principle, and computer technique, using computer and multiple physics effects as tools, and researching real or imagined system by means of system models. It is an effective method to research early system development and system detect, experiment, as well as evaluation. It can also effectively ascertain system specific function blocks and parameters, compensate for the lack of real experimental experiences, and supply technical proof for management decision and technical decision. Simulation experiments can deepen the designer's understanding to certain system, and speed up the research. Advanced simulation technique is of great significance in communication system, national defense system, biomedical field, and so on.

1.2 Communication Simulation Development and Challenges

One of the key tasks in modern communication technology is the arithmetic simulation by means of software. In the past, it was commonly practiced with commercial mathematics software, such as MATHEMATICA, MATLAB, MAPLE,

SAS, etc. Although the application of certain open source mathematics software is getting increasingly popular recent years, its application at communication remains limited.

1.2.1 Communication Simulation Development. A whole process (lifecycle) of simulation is constitutionally a process of knowledge processing. It includes: activities based on models, activities related to models behaviors, and activities concerning qualities. For tens of years, researchers in this field have been ceaselessly research system models which are described as user-oriented or problem-oriented, and an abundance of techniques for the experiments. Their main achievements are modeling methods and simulation software which are equation-oriented, block-oriented, event-oriented, process-oriented, activity-oriented, etc.

1.2.2 Communication Simulation Challenges. Since the core of simulation system is the computational system, the sole is the simulation software. The basis of computational simulation is the simulation models, which represented by simulation software. Many simulation researches are executed by simulation software. Computational simulation system equally needs simulation software to execute its functionalities through series of experiments. Hence, the development of computational simulation technique is doubtlessly strongly related to the research of simulation software. Moreover, in view of the whole simulation system design, the biggest and hardest workload is the research of simulation software.

To improve efficiency, shorten cycle, as well as protect software system reliability, improve software reusability, we are in need of a good software developing and integrating environment. Focusing on the requirement for simulation system, the environment should have following characteristics:

- a* Support real time execution, parallel development, and aperture-free integration.
- b* Enable simulation software well reusable and maintainable.
- c* Use object-oriented technique for the system structure design.
- d* Compatible for currently existing mature simulation software blocks.
- e* Both general and specific.
- f* Have friendly user interface and visible tools environment.

1.3 Communication System Simulation

The development of modern communication system complicates communication system and equipment ceaselessly. To ensure that the communication system has the best performance under certain conditions, we need to be familiar with different parameters when design, and their possible influences on system, as well as complicated interactive relations. In fact, communication system simulation is indeed an approach to some specific aspects of a communication system, by means of applying methods as real as possible.

Along with the development of computer technique, communication system simulation is getting increasingly common, and becoming the main tool to design and analysis communication system. Before communication system simulation widely used, traditional analyzing method was widely used for early analysis and design. Such method was to create a function to conjunct all different design parameters with system performance. To supply a better analysis result, simplification is a must for most real systems, and this confronts the analyzing method with many difficulties when coping with complicated systems. However, computer simulation can generally free from these, and this is why computer simulation widely used.

Board as the meaning of communication system is, it symbols any system that transmits information, which can be analog or digital. Communication system simulation is a multi-subject technique, and comprises of theories from various fields. Some are about traditional communication subject, others the range of simulation. Among others, the knowledge relating to traditional fields is: communication system, communication theory, mathematical model for system, and various sources, etc. By the same token, the subjects that have direct influence on simulation are: pseudo random coding, Monte Carlo estimation, and other estimation techniques.

1.4 Research Contents

This research is aimed using Scilab's modeling environment, Scicos, calling Scilab internal functions and mathematical calculation cells, developing a modern communication simulation toolbox based on Scicos, and expanding as well as constructing blocks for the sub libraries, in order to supply a platform for the open, expandable, and renewable modern communication toolbox, provides conditions for the update, and offers a newly open software environment for using Scilab in modern

communication system. In the meanwhile, through the process of constructing sub libraries and materializing the arithmetic, we also want to venture and evaluate Scilab's effects on communication system modeling.

2. Communication Modeling and Simulation Technique

Constructing simulation models needs to combine the academic and quantitative knowledge together with simulation methods. This chapter will base on the research about simulation technique, mainly discuss constructing modeling simulation environment, and applying modeling simulation technique at communication systems.

2.1 Brief Introduction of Modeling Simulation Environment

Simulation methodology research mainly carried in the direction of content and mechanism. Now I will discuss some simulation environment related issues focusing on simulation principle. In view of modeling simulation environment, the environment which is independent of problems is more transplantable and more commonly used. The details of its own characteristics and structures will be given below.

2.1.1 Modeling Simulation Environment Characteristics. The first step for communication system simulation is to create the computational program, which commonly comprises of main system programs. They connect a set of sub-programs or different types of functional block models together. These functional blocks are programmed and saved in one or more models libraries. According to the logic topology of the simulation system's principles, models establisher or configuration cell will select and connect relevant sub-blocks. Each functional block's parameters should be defined when system configuration, or announced ahead of simulation execution. Simulation parameters values (except design parameters) such as simulation length, sample rate, random number's core (seed), etc. also need to be set before simulation program execution.

At the execution step, signal sample is created, processed, and stored. Signal processing and the calculation's functional estimation during emulating would be processed in the end. The last step is called post-processing, which may include other

functional estimation, calculation, and result display. If the simulation result indicates that the requirement was not meet, system parameters or the topology logic will need to be adjusted iteratively, until the satisfying design completed.

For the advantage of simulation based communication system analysis as well as design, the whole software environment for communication system simulation should have following characteristics:

a. comprehensive graphic environment. Besides the interactivity of analysis and design, the simulation environment should be interactive and graphical. Recent development trends to the visible environment, allowing to model by means of multi-layers blocks application and interactive graphic modeling, for the sake of the convince of display and analysis of simulation result.

b. multi-layers. If the simulation environment allows applying multi-layers structure, complicated system modeling would be easier to materialize. In the multi-layers modeling environment, complicated models can be comprised of simple blocks, as simple blocks' programming and testing are easy to finish. It can start from creating basic blocks, such as adder, multiplier, integrator, modulator, filter, etc. then further apply them to construct transmitter and receiver. Besides, using this multi-layer structure to design system is convenient for teamwork, so that users can handle their most skillful problems separately.

Moreover, multi-layers visual modeling is one of the most common methods in current communication system simulation research. In commercial software packages, users rarely code, whereas instead, they use functional cells to construct simulation models upon multi-layer, and graphic modeling topology logic.

c. transplantable. Software environment should be highly transplantable, so that execution on different hardware platforms can be possible. Current development trends to compile program standardizing and apply window/graphic environment, so that the transplant of simulation software packages and programs cross hardware platforms can be a great convenience.

d. User Interface. Comparing with the past, user interface has been greatly improved. Menu drive commands, online collaboration, error collation, etc. are all the standard characters. They have considered the details of simulation mechanism, so that users can narrowly focus on modeling, analyzing, and designing.

e. blocked. Simulation environment should support blocked software structure. It makes it easier to adjust and design the simulation environment, and moreover is

helpful for the development of hardware, and optimization software environment for new applications.

2.1.2 Modulation and Simulation Environment Structure. Simulation environment is mainly comprised of: simulation host language, graphic user interface (GUI), simulation supported environment, toolbox, models libraries. These integrate an environment which is free of aperture, with appropriate user interface. Its system structure is given in Fig. 1.:

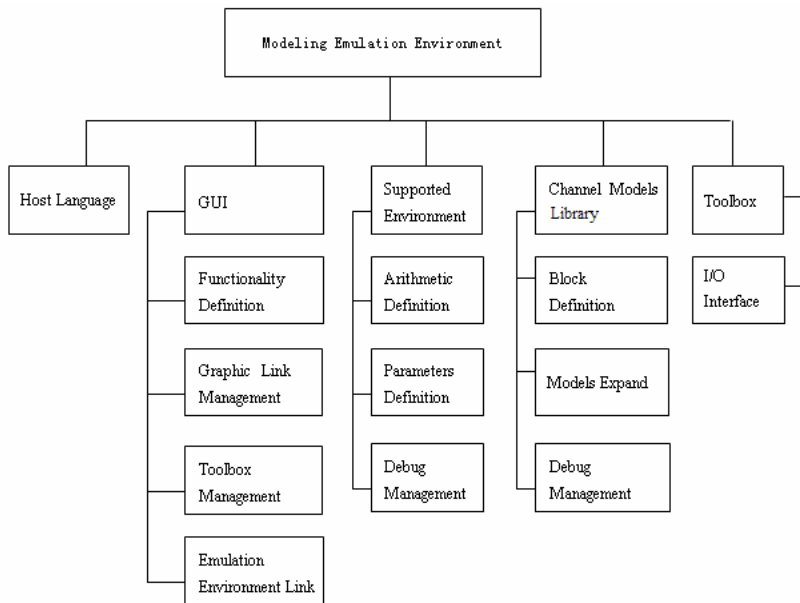


Fig. 1. Modeling Simulation Environment Structure

a. Simulation Host Language. Simulation Host Language's functionality is to read information from modeling simulation examples, identify models which construct the system, create internal link between models, load necessary parameters, arrange modeling visit sequence, materialize data transfer between models, and execute simulation.

b. Graphic User Interface. Based on models libraries, GUI constructs system budgetary models. It is an open task management and resources management system, namely capable to schedule and manage multi-task, multi-thread, and multi-user system, including schedule and manage models libraries, arithmetic libraries, and simulation database.

c. Simulation Supported Environment. It allows user to freely adjust budgetary model's each layer's structure, parameter, and iterative initial values etc. Moreover, independent execution and test of each layer's sub-system are allowed, so that user can emulate, analysis, locate errors of each sub-system, and pause or stop simulation at any time through simulation, as well as further process simulation results after it ended.

d. Toolbox. Toolbox comprises of programmable blocks to describe system functional models' capabilities. These blocks include original blocks (independent program), and the layered models which consist of original blocks as well as lower leveled simple blocks. Models libraries can be expanded by users, namely using the existing blocks in the toolbox, or the independent program (original), to create new blocks. The usage of toolbox can be measured in the following angles: the quality and quantity of the blocks, the form of the blocks (flat or multi-layered), the difficulty in expand, online file, online collaboration, and data certification.

e. Models Libraries. Models libraries are the elements after toolbox upper delamination, and also the sub systems of toolbox's each branch. Each model library materializes specific functionality. Models libraries can be consisted of numbers of functional blocks or of sub models libraries, but can also be a mixture of the two. The elements of the models libraries' bottom level are plenty of functional blocks, so users can achieve specific simulation functionality by using them. It is necessary to point out that a good models libraries' block needs to be multi-used, in order to meet users' needs of constructing systems.

Models need to be reused (namely used more than once in a single simulation), possible to be driven, and able to be executed conditionally by using holding and enabling control signals. It should also support simulation under topology model or point-by-point model. In the latter mode, whenever a block is used, a sampling of its all input ports would be accepted, and commutating a value for each output. In the topology mode, the model would accept a sampling sequence (vector) from its each input port.

Visual modeling advanced computation efficiency, and can materialize multi-rate sampling if combined with enable signals.

2.2 Research about Communication Toolbox Modeling and Simulation based on Scicos

For modern communication system, blocks construction often use mathematical software to materialize. In the field of engineer and mathematics, as open-source software, Scilab has gained increasing amounts of focuses. The software itself includes a massive number of internal functions, and some mathematics function application blocks, but the specific communication sub-system is not supplied. This research taking the advantage of Scilab itself, constructed an open and expandable communication toolbox, and moreover for the expansion of Scilab communication toolbox, supplied an applicable platform, so as to make its content update and functionalities expansion a great convenience.

2.2.1 Overview of Scicos. Scicos default toolbox includes some basic mathematics computation blocks and mathematics functions, both of which were stored at Scicos default sources. Input ‘scicos()’, or click Menu>Applications>scicos, a new diagram would be open. At the diagram Menu>Edit>Palettes, we can get a list of Scicos mathematics calculation units. To query Scicos function sources, click Main Menu>?>Scilab Help, view Scicos: Bloc diagram editor and simulator.

The following are all default Scicos default palettes: Sources, Sinks, Linear, Non_linear, Events, Treshold, Others, Branching, Electronical, Thermo Hydraulics, Old Blocks, Demo Blocks.

2.2.2 Communication Toolbox Modeling Methods. *a Communication Simulation Toolbox Platform.* For Scicos simulation, each block executed once at each time step. So to speak, all blocks executed simultaneously at each time step, namely uses time flow simulation. For Scilab simulation, function executed according to data flow sequence. In other words, data would pass a computation steps, then activate the next, namely uses data flow simulation. Though these two methods would have the same result (under acceptable accuracy), as their different treatments to time, there must be one solution better than the other. Hence, Scicos is selected as the environment to construct communication toolbox.

b Communication Simulation Toolbox Files. Scicos has two ways to create files. One is to call Scilab internal functions to compile files, while another is to convert a set of MATLAB files automatically. To convert from MATLAB files, SCILAB help:

'Matlab to Scilab conversion tips' would be helpful. The function 'translatepaths' is used for this functionality. The details of help documentation are given at Fig. 2.



Fig. 2. Convert MATLAB M-files to Scilab

Through this process we can convert a set of functions. Similarly, all files from MATLAB toolbox could be translate to Scilab file. At Scicos, the block for reading files is given at Fig. 3.

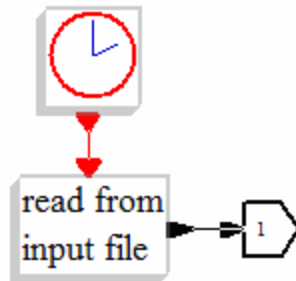


Fig. 3. Read from Input File

Hence, with function `translatepaths()`, we can direct translate multiple MATLAB toolboxes to Scicos toolbox. It not only saves time, enriches Scicos toolbox, but also embodies Scilab's great transplant ability, and the advantage of Scilab toolbox's expandability.

3. Communication Toolbox Based on Scicos

As the foundation of communication system simulation, the toolbox plays a key role in the communication system. Therefore, in a communication simulation system environment, communication toolbox's capability would directly influence simulation environment's practicability and convenience. The details would be discussed below.

3.1 System Introduction

Communication means to transfer information from one place to another. It has many forms, such as talking is a common way for interpersonal communication. However, when the two people are far apart, conversations would be impossible to be sent to the listener through long distance. Therefore, people must rely on some communication systems to materialize long distance information transmission. Among them, the most widely used methods is to use wireless electronic signal to communicate through long distance.

A communication system (see Fig. 4.) includes three parts: signal sender, channel (the media that transmits signals), and the receiver. The source and sink are also viewed as a component of communication system.



Fig. 4. Communication System

Communication system engineer’s task is to design and maintain a communication system, making sure of the receiver’s accurately demodulation, coding, and correction of errors which caused by transmission.

Generally speaking, communication system can be divided into two parts, analog communication system and digital communication system.

Analog communication system’s sender includes signal modulation and multiple access these two parts. Multiple access is a sharing technique, depending on sequential signal sending, numbers of multiple accesses can share the same channel resource. Analog communication system’s receiver includes signal filter units and signal demodulate. The whole system structure is given at Fig. 5.

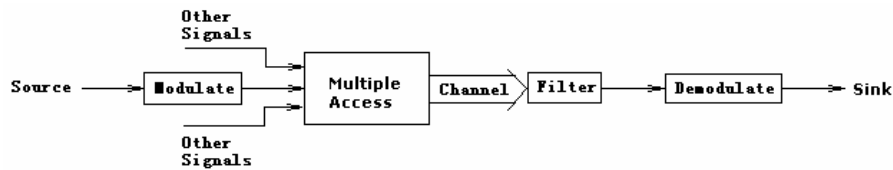


Fig. 5. Analog Communication System

In digital communication system, message is represented by digital signals. System sender includes source coding, error control coding, digital modulation, multiple access, etc. Source coding process uses digital signal to take the place of analog signal. Data compress process translates original message to compressed information in order to decrease the bit amount, and the receiver can take back the original message from the compressed file by data decompress. Error control coding process adds extra error check up and/or error reclaim bit, so the receiver can use the excessive bit information to check errors which caused by transmitting, or correct some of them. Digital modulation process can map digital information to analog information, and use high frequency signal to carry this analog information. The transmitting process can be seen at Fig. 6.

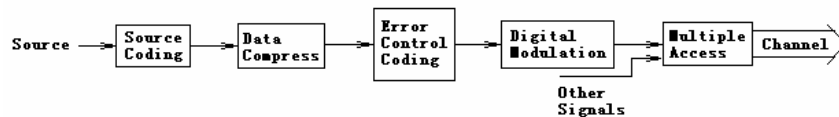


Fig. 6. Digital Communication System Sender

Theoretically, receiving is a contradictory process of sending. Under most circumstances, the receiver’s structure should match the sender, so that the original information can be regained correctly. Fig. 7. shows the receiving process. Among others, channels have different models in different application environments. It would bring signal influences such as noise, decline, phase excursion, disturbance etc.

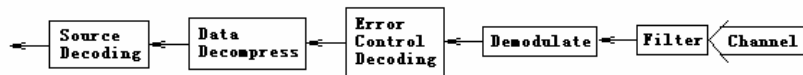


Fig. 7. Digital Communication System Receiver

Technique in digital communication field can also be used in others. Such as source coding, data compress and error control coding are widely used in computer data storage.

It is necessary to point out that the communication system toolbox mentioned here includes analog communication system, and digital communication system as well.

3.2 Communication Simulation Toolbox Requirement Analysis

3.2.1 Communication Simulation Toolbox Structure. The upper part of Fig. 8. is the communication system's sending part, and the down part is the receiving part. It can be clearly told that the receiving part is completely symmetry to the sending, and it is a mirror image of that. In many cases (not all), receiving side’s calculation is a contradictory process of the sending’s.

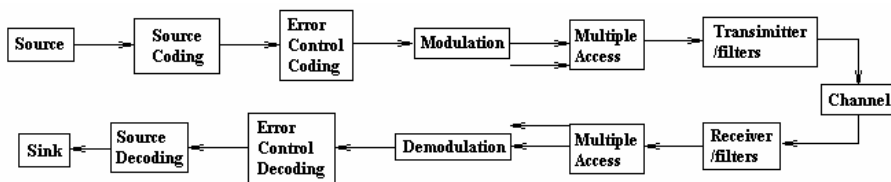


Fig. 8. Communication Toolbox Upper Simulation Modules Library

By double clicking any of the squares, the next level models libraries can be seen. By clicking the sending side's any block, or its mirror at receiving side, we will get the same sub models library.

User can use the default functions as well as blocks to construct their own communication system. By clicking open the block with mouse, can they adjust the

parameters at dialogs. In some sub-libraries, there is a line between the sending and receiving block, and it tells that they are a pair of matched sending and receiving processing. Generally speaking, these modules also need to be used in pair.

3.2.2 Communication Simulation Toolbox Sub Models Libraries and Their Elements. In each sub library of the communication toolbox, there are many basic functional modules. Now I will give an overall description for each.

a Modulation and Demodulation Models Library. It contains digital and analog Modulation and Demodulation separately. For analog, it moreover includes baseband transmission library and passband transmission. For each, the basic modulation and demodulation technique related models are: DSB-SC Modulation and Demodulation, DSB-TC Modulation and Demodulation, SSB-AM Modulation and Demodulation, QAM Modulation, QADM Demodulation, Frequency Modulation and Demodulation, Phase Modulation and Demodulation, etc. Likewise, Digital Modulation and Demodulation library also includes baseband transmission library and passband transmission. In each, the main models are: General QAM, M-PAM, Rectangular QAM, CPFSK, CPM, GMSK, MSK, M-FSK, BPSK, DBPSK, M-DPSK, M-PSK, DQPSK, OQPSK, QPSK.

b Source Models Library. It basically includes random signal generator library, and the function generator library. For random signal generator, they are: Gauss noise generator, Rayleigh noise generator, Poisson noise generator, Equipartition noise generator, complex binary noise generator, etc. Function generator includes saw tooth signal generator, square wave generator, sinusoid signal generator, VCO, discrete VCO, etc.

c Channel Models Library. It includes additive Gauss white noise channel, binary symmetric channel, Rayleigh noise generator, Fading channel, etc.

d Sink Models Library. It mainly includes continuous time eye pattern, discrete time eye pattern, digital signal oscilloscope, event signal oscilloscope, error rate analysis, etc.

e Synchronization Models Library. It includes charge pump PLL, Baseband signal PLL, Linearized Baseband PLL, Phase-locked Loop, Passband signal PLL, etc.

f Customized Tools Models Library. It includes models which can help define interactively, encapsulate sub-block, input comments, etc.

g Coding and Decoding Models Library. It mainly includes source encoding and decoding sub models library, error control encoding and decoding sub models library,

interlace coding sub models library, etc., including: A-Law Compressor, A-Law Expander, Differential Decoder, Differential Encoder, DPCM Decoder, DPCM Encoder, Enabled Quantizer Encoder, Mu-Law Compressor, Mu-Law Expander, Quantizer Decoder, Sampled Quantizer Encoder, etc.

h Filter Models Library. It includes butterworth filter, 1st-Order Filter, 2nd-Order Filter, ellipse filter, etc.

i Basic Communication Functions Models Library. It includes models which carries on basic communication functions for further use.

3.3 Communication Toolbox Based on Scilab

Finally, the communication toolbox can be successfully load into Scicos, and the models libraries as well as the sub models libraries can all the time be comfortably used. See Fig. 9. for an example of the final toolbox. Each of the functional models libraries can be click open, and can supply further specific models or even sub models libraries.

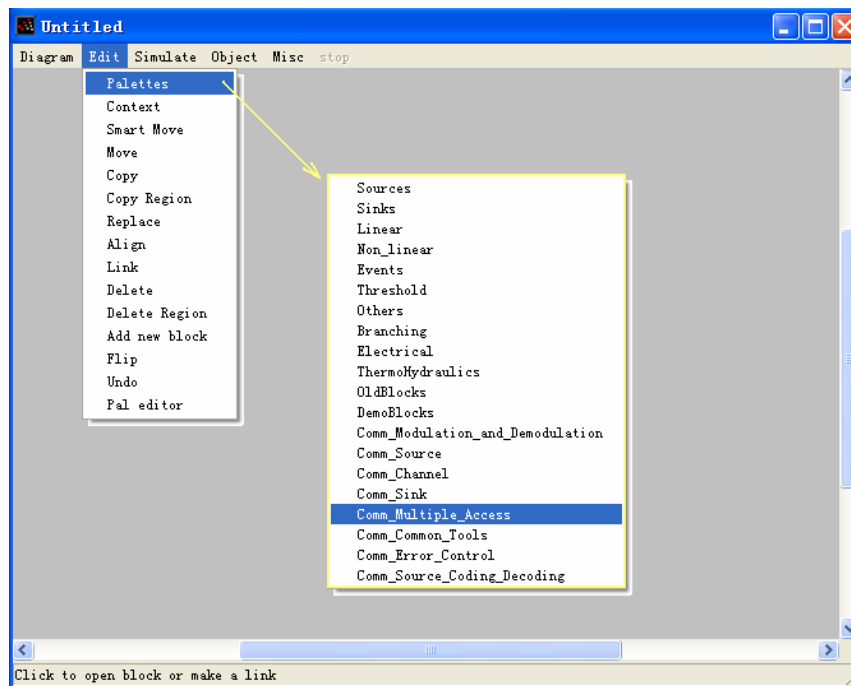


Fig. 9. Communication Toolbox based on Scicos

4 Characteristics of the Toolbox

This Toolbox has following characters:

a Under Scicos, this toolbox can emulate linear or nonlinear system, continues time and discrete time system, or a mixture of the two. It can also simulation multiple rate system.

b This toolbox is greatly renewable, and easy to be updated. Since each block is created by using functional blocks or functions, new blocks can be add to expand the toolbox.

c GUI friendly. Users can simply use mouse and Scilab functions calls to construct their system model, and carry out simulations interactively, free from the complicated arithmetic based on differential equations, and the simulation result can be well saved.

d Use Scicos as the environment, whose GUI is similar to MATLAB, so that easy to be understood.

e Supply different ways to construct system: using block, using functions, calling Scilab files, convert file from MATLAB, FORTRAN, C language files and use it.

References

1. Bingjun Wang, etc. : Modern Communication System Principles. Tianjin Tianjing University Press. (1999.8).
2. Yanhua Zhang: A Graphic Modeling Technology for Satellite Radio Path Computing (2002.6.)
3. FuZhen Xue, etc.: Simulation Software of Control System Diagram Configuration Based on Elements. Journal of System Simulation (2002.7).
4. Scilab, A Free Scientific Software Package: <http://www.scilab.org>
5. Scicos: Scilab's block diagram modeler/simulator: <http://www.scicos.org>

The realization of SCADA based on Scilab

Zhe Peng, Longhua Ma

Institute of Industrial Control of Zhejiang University

310027 Hangzhou, China

[Pengzhe1113@gmail.com](mailto:pengzhe1113@gmail.com) , lhma@iipc.zju.edu.cn

Abstract. Nowadays, optimization and control are considered more and more important in the field of industrial control. However, Traditional SCADA (Supervisory Control and Data Acquisition) software can only implement the functions of data acquisition and data monitor, which can not do optimization and control because of not enough ability of scientific computation. In this paper, a method of SCADA realization is proposed which is based on scientific computation software Scilab. The SCADA system consists of data acquisition module, data monitor module, optimization and control module, and data storage module. In control area, PID simulation, Genetic Algorithm and Fuzzy control can be done by Scilab. Hence, we could write OPC, Modbus, Mysql and Industrial Ethernet interface to achieve the communication between SCADA and devices or other software. Finally, the paper describes the application of SCADA system in a water enterprise.

Key words: SCADA, interface, Scilab, OPC, Modbus, Industry Ethernet, Database

1 Introduction

In industrial control area, the SCADA (Supervisory Control and Data Acquisition) system is widely used because of its convenience [1]. But the traditional SCADA system can only implement the functions of data acquisition and data monitor, having no enough scientific computation ability to do complicated control and optimization.

To solve the scientific computation ability of SCADA, compute module should be imported in the SCADA system, or developing the SCADA system based on

scientific computation software. Most scientific computation software is commercial and without source code, which increases the cost and the difficulties of SCADA system.

A method of realizing SCADA system is proposed which is based on scientific computation software Scilab. Developed at INRIA, Scilab has been developed for system controlling and signal processing [2]. It is freely distributed in source code format.

Scilab's scientific computation ability enables SCADA's complicated control and optimization be realized. Scilab is easily interfaced with C subprograms. This allows the use of standardized packages and libraries in the interpreted environment of Scilab. OPC, Mysql, Modbus and Industrial Ethernet modules can be realized through DLL which is built by Visual Studio.

2 The introduction and designing of SCADA system

SCADA systems are widely used in industry for Supervisory Control and Data Acquisition of industrial processes, such as electric power system, water supply system, petroleum and chemical engineering, etc. System is composed of main control stations and RTU (Remote Terminal Unit).

The core of SCADA system is data acquisition and data monitor. The SCADA system is made up of data acquisition module, data monitor module, optimization module and control module.

In this system, Supervisory interface is developed with TCL/TK, optimization and control, such as PID simulation and Fuzzy control which are realized through Scilab. With Modbus protocol, it can communicate with devices. The provision of OPC client functionality for SCADA to access devices in an open and standard manner is developing. Server-client communication is in use with Modbus/TCP protocol.

The software architecture of this SCADA system can be described as figure 1.

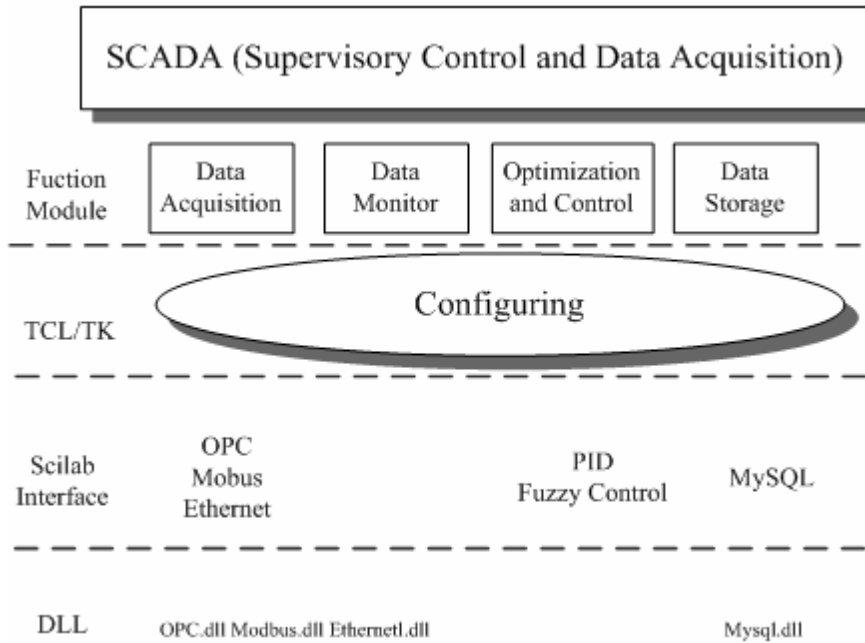


Fig. 1. SCADA system based on Scilab software frame

3 The realize of SCADA system’s interface with Scilab

3.1 Scilab’s C programs interface and TCL/TK interface

Scilab can be easily interfaced with C programs. This is useful to have faster code or to use specific numerical code for, e.g., the simulation or optimization of user defined systems, or specific Lapack or netlib modules [2]. In fact, interfacing numerical code appears necessary in most nontrivial applications. For interfacing C or Fortran programs, it is of course necessary to link these programs with Scilab. This can be done by a dynamic (incremental) link or by creating a new executable code for Scilab. For executing a C or Fortran program linked with Scilab, its input parameters must be given specific values transferred from Scilab and its output parameters must be transformed into Scilab variables.

An interface between Scilab and Tk-Tcl exists. A Graphic User Interface object can be created by the function `uicontrol`. Basic primitives are `TK_EvalFile`, `TK_EvalStr` and `TK_GetVar`, `TK_Setvar`.

3.2 The significance of realizing OPC, Modbus and Mysql interface

The OPC Foundation has established a set of standard OLE/COM interface protocols intended to foster greater interoperability between automation/control applications, field systems/devices, and business/office applications in the process control industry [3].

This specification describes the OPC COM Objects and their interfaces implemented by OPC Servers. An OPC Client can connect to OPC Servers provided by one or more vendors.

Different vendors may provide OPC Servers. Vendor supplied code determines the devices and data to which each server has access, the data names, and the details about how the server physically accesses that data. Specifics on naming conventions are supplied in a subsequent section. With Mysql interfacing, in can storage data, alarm to database, which is the base of report.

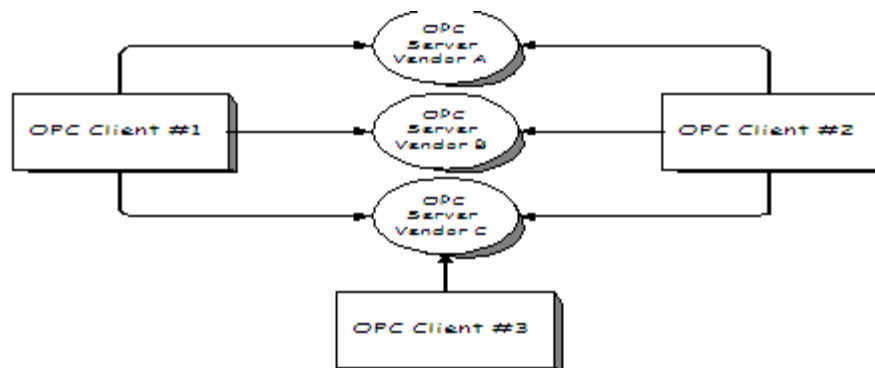


Fig. 2. OPC Client/Server Relationship

Although OPC is primarily designed for accessing data from a networked server, OPC interfaces can be used in many places within an application. At the lowest level they can get raw data from the physical devices into a SCADA or DCS, or from the SCADA or DCS system into the application.. The architecture and design makes it possible to construct an OPC Server which allows a client application to access data

from many OPC Servers provided by many different OPC vendors running on different nodes via a single object.

Controllers supporting Modbus protocol can communicate with each other and with other devices over a variety of networks. Supported networks include the Modbus and Modbus Plus industrial networks, and standard networks such as MAP and Ethernet. Networks are accessed by built-in ports in the controllers or by network adapters, option modules, and gateways [4].

During communications on a Modbus network, the protocol determines how each controller will know its device address, recognize a message addressed to it, determine the kind of action to be taken, and extract any data or other information contained in the message. If a reply is required, the controller will construct the reply message and send it using Modbus protocol.

MODBUS/TCP is a variant of the MODBUS family of simple, vendor-neutral communication protocols intended for supervision and control of automation equipment. Specifically, it covers the use of MODBUS messaging in an 'Intranet' or 'Internet' environment using the TCP/IP protocols. The most common use of the protocols at this time are for Ethernet attachment of PLC's, I/O modules, and 'gateways' to other simple field buses or I/O networks [5].

Mysql is open source database software. Establishing distribution real time database and relation database can build data report and provide data for DSS (Decision Support System).

3.3 The design and realization of OPC, Modbus, Mysql and Industrial Ethernet interface

The simplest way to call external programs is to use the link primitive (which dynamically links the user's program with Scilab) and then to interactively call the linked routine by call primitive which transmits Scilab variables (matrices or strings) to the linked program and transforms back the output parameters into Scilab variables. An other way to add C or Fortran code to Scilab is by building an interface program.

It is possible to add a permanent new primitive to Scilab by building an interface program as above and making a new executable code for Scilab. This is done by

updating the fundef file [6]. The interfacing of Mysql, OPC and Modbus is shown as figure 3, and the configuring is shown as figure 4.

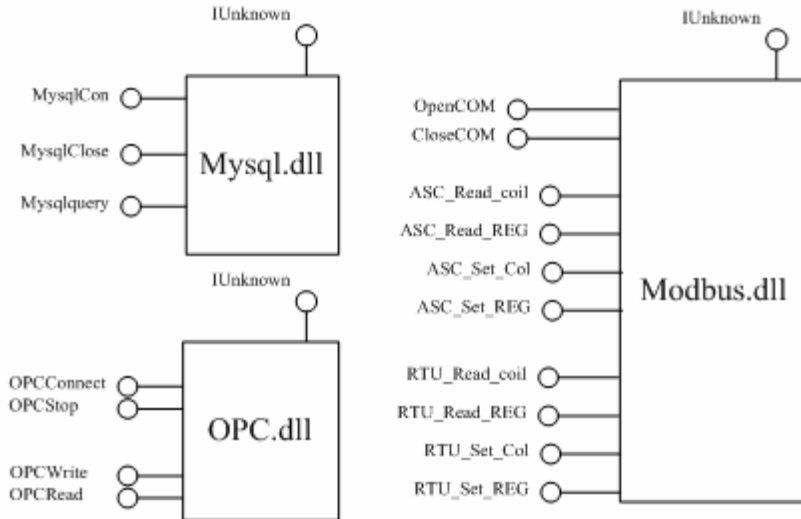


Fig. 3. The Interfacing of Mysql, OPC, and Modbus

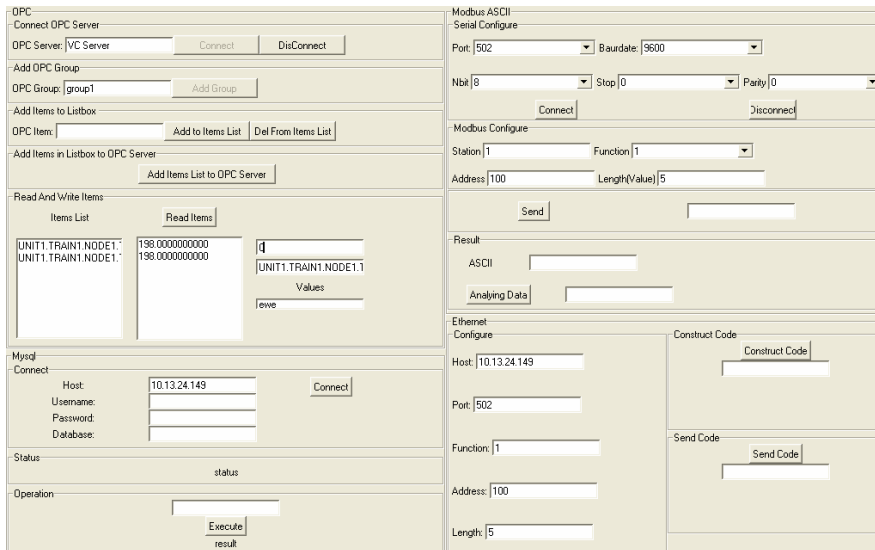


Fig. 4. The SCADA Interface Configuring

4 The realization of control and optimization with Scilab

4.1 Scilab's application in control area

Scilab provides a variety of powerful primitives for the analysis of non-linear systems. Integration of explicit and implicit dynamic systems can be accomplished numerically [7].

The scicos toolbox provide the function of the graphic definition and simulation of complex interconnected hybrid systems. There exist numerical optimization facilities for non linear optimization (including non differentiable optimization), quadratic optimization and linear optimization.

4.2 PID simulation with Scilab

The proportional-integral-derivative (or PID) control technique has been in existence for over eighty years. Up to now, however, this technique is still keeping a dominant role in control engineering. Both the past history of control engineering as well as PID control technique itself can support this belief. PID control takes the advantages of the “feedback” idea with most intuitive yet simplest means. As long as the feedback control remains, the principle of PID will be behind in working either explicitly or implicitly, or at least partially. In control engineering, PID control technique has been considered as a matured technique in comparing with other control techniques. With Scilab, we can do PID simulation. The configuring and result is shown as figure 5 and figure 6.

Base variable

continue System discrete System

Please enter the expression:

Kp:

T0:

T1:

-Stability

stable

Frequency	Continuous Time	Discrete Time
<input type="button" value="Frequency response"/>	Sampling period: <input type="text" value="0.1"/>	Tmax: <input type="text"/>
	Tmax: <input type="text" value="50"/>	<input type="button" value="OK"/>

Fig. 5. PID Simulation Configuring

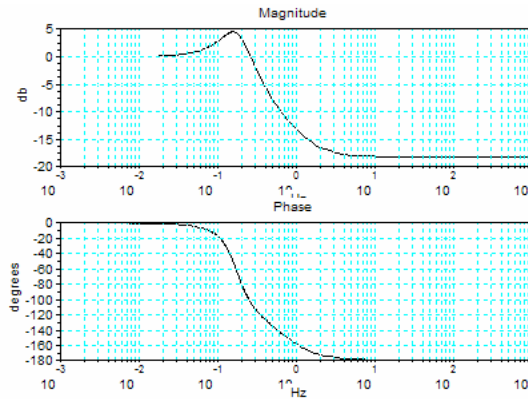


Fig. 6. PID Simulation Result

5 The SCADA system's application in water enterprise

In TongLu's water enterprise, this SCADA system is used and proved running good. The network of the SCADA system is shown as figure 7.

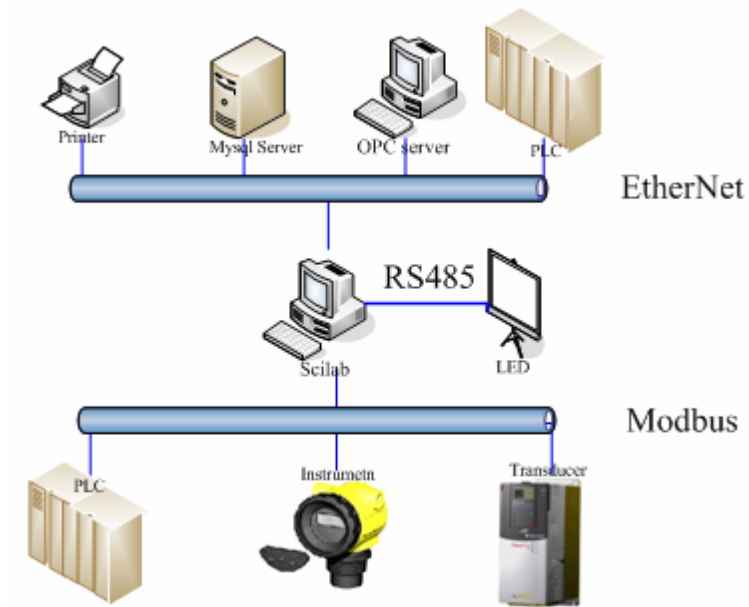


Fig. 7. The SCADA System Network in TongLu's Water Treatment

It can Supervise the whole enterprise, including fetching water, filtrating, remote control.

With OPC interfacing, it can exchange data with PLC (Programmable Logic Controllers), or other commercial hardware modules. With Modbus interfacing, it can write to Modbus slave device data, and read data, which is enable manual operation. With Mysql interfacing, in can storage data, alarm to database, which is the base of report. The SCADA software is shown as figure 8.



Fig. 8. The SCADA Software

6 Conclusion

- (1) SCADA system can be developed with Scilab.
- (2) SCADA system, which is based on Scilab, can do control and optimization because of its scientific computation ability.
- (3) SCADA system can extend its interfaces through dll, which is built by Visual studio, such as OPC, Modbus, Mysql, Ethernet, etc.
- (4) The stability of SCADA system is proved to be good in industry enterprises.

Acknowledgments. This work is supported by a grant from the National Natural Science Foundation of China (No. 60474064) and the Fund of produce-study-research Science and Technology Cooperation of Hangzhou City to Zhou LiFang and Ma LongHua.

References

1. A. Daneels, W.Salter, *WHAT IS SCADA?*, Trieste, Italy, International Conference on Accelerator and Large Experimental Physics Control Systems, 1999
2. Scilab Group, *INTRODUCTION TO SCILAB*
3. OPC Foundation, *Data Access Custom Interface Standard*, December 17, 2001
4. MODICON, Inc., Industrial Automation Systems, *Modicon Modbus Protocol Reference Guide*, June 1996
5. Andy Swales Schneider Electric. *OPEN MODBUS/TCP SPECIFICATION* , 29 March 1999
6. Lydia E. van Dijk and Christoph L. Spiel , *Scilab Bag Of Tricks: The Scilab-2.5 IAQ*, 2000
7. Baogang Hu, Xing Zhao and Mengzhen Kang, *Introduction to Scientific Computing Software- SCILAB*, Tsinghua University Press, Beijing, 2003.
8. Brent B.Welch, *Practical Programming in Tcl/Tk* , PH PTR, 2002.

Engineering Applications Based on Scilab

A Stereo Matching Implementation Method for Uncalibrated Color Image Based on Scilab

Zhu-lin LI^{1,2} Zong-tao ZHAO¹ Gen-yao ZHANG² Yan MA²

(1. 402 Division, Xian Research Inst. Of Hi-Tech , Xi'an, China, 710025)
(2. Institute of Computer Science, Yan'an University , Yan'an, China, 716000)
E-mail: ydlizhulin@163.com

Abstract. It is crucial problem How to resolve the correspondence in stereo matching. Traditional methods are to calibrate camera firstly, then to compute the projection matrix. But its application is limited in many areas. In this paper, a stereo matching implementation method for color image uncalibrated camera based on Scilab is proposed, which is based on geometric affine invariant and photometric invariant. The master idea is to find correspondences of feature points between left image and right image, which are measured by gradient invariance under some constraints and Mahalanobis distance, and false matched points are rejected by topology property. The results show the method expands its application and improves matching accuracy, right ratio is 83.1%.

Keywords: affine invariant, photometric invariant, uncalibrated camera parameter, stereo matching

1 Introduction

Depth information obtaining of 3D-Scenes is one of vital technologies of computer vision. Stereo vision is a significant passive method, its key technology is how to resolve correspondence problem. Traditional methods are to calibrate camera firstly, then to compute the projection matrix. Yet, this limits its application in many areas. In this paper, a stereo matching method for color image uncalibrated camera is proposed and implemented on Scilab workshop. Scilab is a scientific software package for numerical computations providing a powerful open computing environment for engineering and scientific applications^[1], it includes hundreds of mathematical functions with the possibility to add interactively programs from various languages, for example, C, VC++ and so on. The master idea of this paper is finding correspondences of feature points between left image and right image, which are measured by gradient invariance under some constraints and Mahalanobis distance, and false matched points are rejected by topology property. Then fundamental matrix is computed and epipolar geometry is estimated.

2 Invariant Detection

It is chief work that affine invariant and photometric invariant are extracted in wide baseline stereo matching. Interest points detected by SUSAN detector can be adapted

to affine transformations^[2] and give repeatable results when scale $s \in [2/3 \sim 3/2]$. Gradient of color image can be adapted to illumination changes when photometric changes $\Delta L \in [-15 \sim +39]$, see table 1 and figure 1. Otherwise, gradient is invariant in other photometric changes interval, see figure 2 .

Interest points include edge points, corners, and so on. In this paper, we use SUSAN detector to detect feature points, SUSAN operator^[3] is a robust detector. The basic idea of SUSAN operator is to count pixels having same intensity with nucleus in circle panel, get initial points. The local maximum of these points is interest point.

The following is to analyze photometric invariance.

We can describe a color image by tone, color, and intensity. The intensity describes reflection ability of object, its expression i

$$P = L/L_0 \quad (1)$$

where L is the intensity of object surface, L_0 is the intensity of white object surface. Obviously, the reflection ability relates to illumination. When illumination is same, let L_A denote the intensity of object A, and L_B denote the intensity of object B, then the ratio is

$$P_{AB} = L_A/L_B \quad (2)$$

P_{AB} correlates with the change-rate of intensity between A and B. Gradient of point describes the intensity change of the point and adjacent points, the gradient amplitude and direction are

$$|G(x, y)| = \sqrt{f'_x{}^2 + f'_y{}^2}, \quad \theta(x, y) = \arctan(f'_y/f'_x) \quad (3)$$

We extract interest points of 2D color image using SUSAN operator based on Scilab workshop, table 1 and figure 1 show results of photometric changes, gradient amplitude and direction for a random interest point.

Table 1. Correlative data of IC and gradient

IC Gradient	-40	-30	-20	-15	0	10	20	39	40	50
RGB- GA	70.23	71.47	74.33	77.79	77.79	77.79	77.79	77.79	77.20	79.40
RGB- GD	0.35	0.31	0.34	0.31	0.31	0.31	0.31	0.31	0.29	0.29

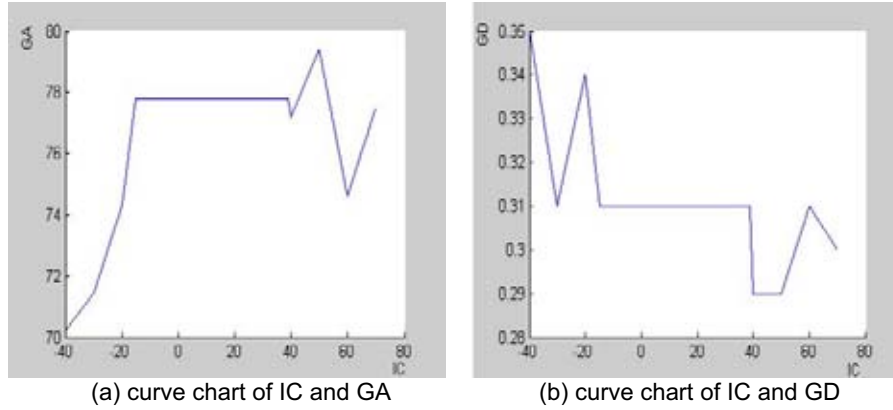


Fig. 1. Curve chart of IC and gradient

where PC denotes intensity change, GA denote gradient amplitude, GD denote gradient direction

We get the conclusion by experiment results: the gradient amplitude and direction are invariant when $\Delta L \in [-15 \sim +39]$.

A different intensity image is selected, its gradient can be adapted to illumination changes when photometric changes $\Delta L \in [-26 \sim +58]$, see Figure 2. The result show the gradient is invariant under other photometric changes interval.

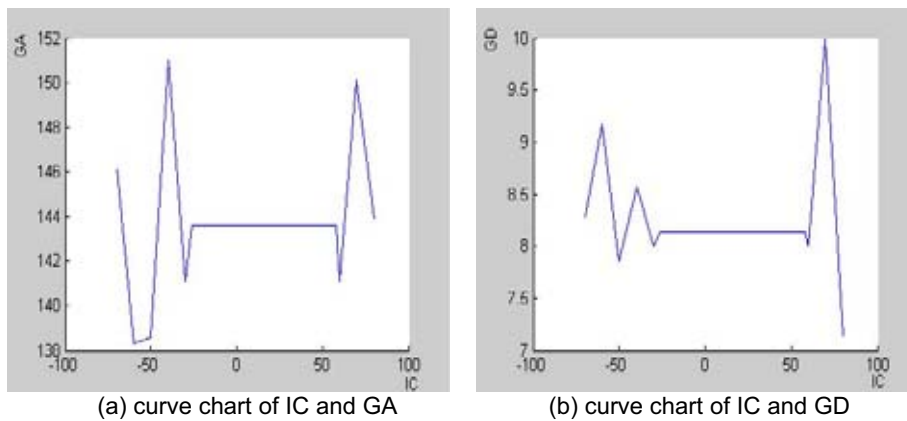


Fig. 2. Curve chart of IC and gradient

3 Similarity Metric Algorithm

Mahalanobis distance is covariant distance of two-vector, the distance of vector a and b is given by

$$d_m(a,b) = \sqrt{(b-a)^T \Lambda^{-1} (b-a)} \quad (4)$$

where Λ is covariant matrix of reference image region.

A set of points u_i^L of the left image, u_j^R of the right image, when scale $s \in [2/3 \sim 3/2]$ and photometric change $\Delta L \in [-15 \sim +39]$, similarity metric algorithm of searching correspondences is:

1. Compute the gradient amplitude of u_i^L and u_j^R by formula (3) respectively, if $\|G_L| - |G_R\| \leq \delta$, where δ is given amplitude threshold, then go to step2.

Otherwise, select u_{j+1}^R , go to step1;

2. Compute the gradient direction of u_i^L and u_j^R by formula (3) respectively, if $\|\theta_L| - |\theta_R\| \leq \phi$, where ϕ is given direction threshold, then go to step3.

Otherwise, select u_{j+1}^R , go to step1;

3. Compute Mahalanobis distance d_m of u_i^L and u_j^R by formula (4), if $d_m \leq \varepsilon$, where ε is given threshold, then marks u_i^L as u_i^{L*} , marks u_j^R as u_j^{R*} , u_i^{L*} and u_j^{R*} are a pair of corresponding points.

4. Select u_{i+1}^L , go to step1.

The results are initial correspondences set.

4 Rejecting False Matched Points

Due to the wide range of geometric and photometric transformations allowed, false correspondences are inevitable. In this paper, false matched points are rejected by the idea proposed by Ferrari et al^[4], which is an extension of the topological filter. Instead, our filter can be only sets of matches containing points. The experiment show the result is effective.

The triple interest points (D_1^1, D_1^2, D_1^3) of the left image are not in line, their correspondences are (D_2^1, D_2^2, D_2^3) of the right image. Their topological relation is

$$side(D_v^1, D_v^2, D_v^3) = sign((D_v^2 \times D_v^3) D_v^1), \quad (v=1,2) \quad (5)$$

if D_v^1 lies on the left side of the directed line $D_v^2 \times D_v^3$, then. $side = 1$. Otherwise, $side = -1$.

The same side equation is

$$side(D_1^1, D_1^2, D_1^3) = side(D_2^1, D_2^2, D_2^3) \quad (6)$$

In initial matched points, if points are not satisfied with function (6), then reject these points.

We compute the Fundamental matrix F using 8 points algorithm^[5], then detect the points not satisfied with epipolar geometry, that is function (7), to get robust correspondences.

$$u'^T F u = 0 \quad (7)$$

where u and u' are correspondences of two-view image.

5 Experiment Results and Analysis

The Experiment Results

We have done the simulation experiment on Scilab(Pentium 4 at 1.8GHz) workshop, the result see Figure 3.

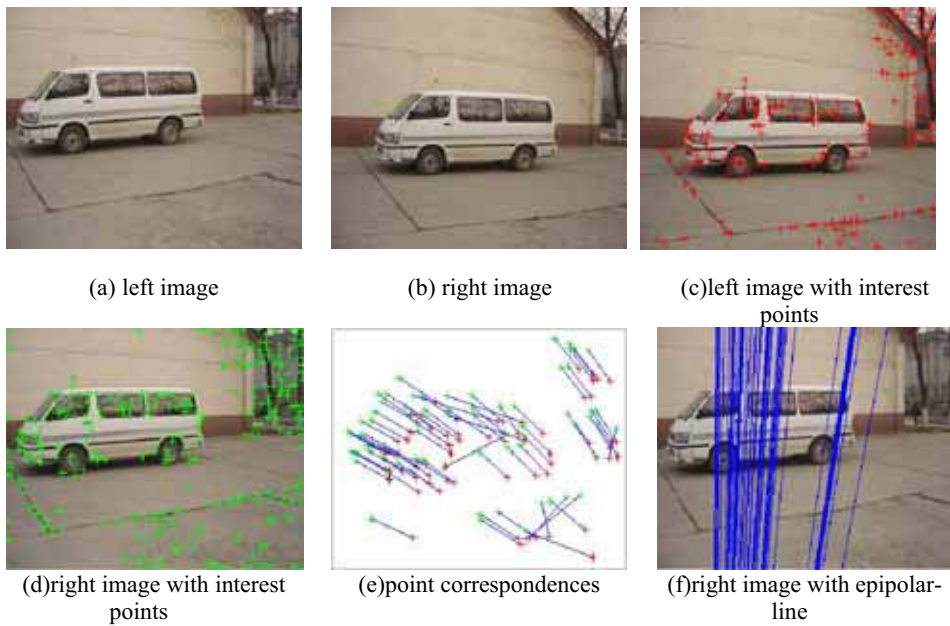


Fig. 3. Results for stereo matching of color image

Invariant Analysis

The extraction of affine invariant and photometric invariant is one of key technologies in wide baseline stereo matching. Interest points detected by SUSAN-

affine detector are invariant when scale $s \in [2/3 \sim 3/2]$. Gradient of color image is photometric invariant when intensity variant $\Delta L \in [-15 \sim +39]$, see Figure 1, Figure 2 and Table 1.

Capability Analysis

Stereo matching should do three aspects work mainly: extracting invariant features, finding initial correspondences, and rejecting false matched units. In this paper, we have finished these work using Scilab language, results see figure 1, 2 3 and table 1, 2.

Table 2. Results of stereo matched algorithm

interest points of left image	interest points of right image	correspondences	correspondences after rejected	right-ratio	time(ms)
236	283	83	69	83.1%	274

6 Conclusion and Future Work

In this paper, we select the feature point as matching unit, do initial matching of color image by gradient invariant under some constraints and Mahalanobis distance, rejected false matched points by topology property, and resolve the correspondences problem for color image uncalibrated camera, The results show the method expands its applicability. The method has the following advantages:

Comparing this method and dense matching method, this is a robust method, having high speed and good ability of resisting noise.

- geometric and photometric constraints have been proposed ,this allows to reject falsely matched correspondences at an early stage, so we can get more exact initial correspondences.

Matching accuracy is more robust because of rejecting false correspondences by using topological relation of triple points not in line.

But, it fails for images with large scale changes and big intensity changes, therefore the design of a more robust and distinctive algorithm is required. At the same time, searching correspondences for image with smooth texture surface or repetitive texture surface is a future area of work.

7 References

[1] Scilab, A Free Scientific Software Package, <http://www.scilab.org/>, 2006

- [2] Mikolajczyk K, Tuytelaars T, Schmid C, Zisserman A, Matas J, Schaffalitzky F, Kadir T, Van Gool L. A Comparison of Affine Region Detectors[J], IJCV,2005, 65(1-2):43-72.
- [3] Smith S M, Brady J M. SUSAN-A New Approach to Low Level Image Processing [J].Int. Journal of Computer Vision, 1997,23(1):45-78.
- [4] Ferrari V, Tuytelaars T, Gool L V. Wide-baseline multiple-view correspondences[A]. Proceedings of Computer Vision and Pattern Recognition[C], Madison, USA,2003:718-725.
- [5] Longuet-Higgins. A computer algorithm for reconstructing a scene from two projectio

Dynamic Analysis and Simulation of Water Management in PEM Fuel Cell based on SCILAB

Liyan Zhang^{1,2}, Mu Pan¹ and Shuhai Quan²

1. State Key Laboratory of Advanced Technology for Materials Synthesis and Progressing, Wuhan University of technology, Wuhan, Hubei Province, China
2. School of automation, Wuhan University of technology, Wuhan, Hubei Province, China
E_mail: zeroplus@tom.com

Abstract. In this paper, water balance equation is developed to simulate the transient phenomena in a proton exchange membrane fuel cell (PEMFC). This model is built by investigating water distribute of membrane and water transport process in cathode and anode. The proposed model is implement in Scilab environment. This paper discuss the parameters (return manifold in cathode, mass flow rate of inject water and current of stack) of water management in PEMFC stack. The simulation results provide good guidance in designing the structure of return manifold and optimal control of operation parameter. The simulation results also show that Scilab is a powerful and convenient tool.

1 Introduction

Fuel cell stack systems are under intensive development for mobile and stationary power applications. In particular, Proton Exchange Membrane (PEM) Fuel Cells (also known as Polymer Electrolyte Membrane Fuel Cells) are currently in a relatively more mature stage for ground vehicle applications. In PEM fuel cells the membrane must be sufficiently hydrated because its conductivity depends critically on the humidity level. Too little water causes membrane drying, which increases the ionic resistance, and exacerbates the voltage

drop due to ohmic losses. Too much water causes “flooding”, that is blocking of porous passages, which reduces the transport rate of reactants to the catalyst site.

In this paper, we build model of water management in PEM fuel cell and implement in SCILAB. This paper is organized as follow: In section 2 we describe the water management systems in PEM fuel cell, in section 3 we build water balance equation in PEM fuel cell stack and implement in SCILAB and we give simulation result in section 4 and discuss the parameters of effect water management by drawing figure in SCILAB.

2 Water Management System

In PEM fuel cell stack oxygen and hydrogen are consumed, and water and heat are generated. Generally, a fuel cell stack is integrated with four auxiliary systems: hydrogen supply system, air supply system, cooling system and humidification system. Water management system, that is humidification system, is to maintain hydration of the polymer membrane and to balance water consumption in the system. The amount of reactant flow and the water injected into the anode and cathode flow streams affect the humidity of the membrane. Dry membranes and flooded fuel cells cause high polarization losses. As the current is drawn from the fuel cell, water molecules are both produced in the cathode and dragged from the anode to the cathode by the hydrogen protons. As the concentration of water in the cathode increases, the concentration gradient causes water to diffuse from the cathode to the anode. Perturbation in fuel cell humidity can be caused by different mechanisms: water generated while load increases, changes in the absolute and relative reactant pressure across the membrane, changes in air flow rate and changes in stack temperature, which change the vapor saturation pressure [1]. These mechanisms indicate strong and nonlinear interaction among the humidity control tasks. A 20 to 40% drop in voltage can be occur if there is no proper humidification control [2].

3 Model of Water Management System

We build water balance equation by investigating water distribute of membrane and water transport process in cathode and anode. In this paper humidification method is direct inject water into cathode. The following equations are water distribute of membrane and water balance equation of cathode and anode.

a) Water balance equation in anode

The governing equation for hydrogen and water in the anode can be written as

$$\frac{dm_{H_2}}{dt} = W_{H_2,in} - W_{H_2,out} - W_{H_2,ret} \quad (1)$$

$$\frac{dm_{\omega,an}}{dt} = W_{v,an,in} - W_{v,an,out} - W_{v,mbr} \quad (2)$$

The hydrogen and water flows to the anode are calculated by

$$W_{H_2,in} = (1 - \alpha)W_{an,in} \quad (3)$$

$$W_{v,an,in} = \alpha W_{an,in} \quad (4)$$

Where α is content of vapor.

The outlet hydrogen and vapor mass flow rate is calculated by the following equations.

$$W_{H_2,an,out} = \frac{1}{1 + \omega_{an,out}} W_{an,out} \quad (5)$$

$$W_{v,an,out} = \frac{\omega_{an,out}}{1 + \omega_{an,out}} W_{an,out} \quad (6)$$

$$\omega_{an,out} = \frac{M_v}{M_{H_2}} \cdot \frac{m_{H_2} R_{H_2}}{m_{v,an} R_{H_2}} \quad (7)$$

The rate of water flow across the membrane $W_{v,mbr}$ is determined in the membrane hydrogen model. The rate of hydrogen consumed in the reaction is a function of the stack current

$$W_{H_2,ret} = M_{H_2} (nI_{st}) / (2F) \quad (8)$$

b) Water balance equation in anode

The mass continuity is used to balance the mass of the three elements: oxygen, nitrogen and water, inside the cathode volume [3].

$$\frac{dm_{O_2}}{dt} = W_{O_2,in} - W_{O_2,out} - W_{O_2,ret} \quad (9)$$

$$\frac{dm_{N_2}}{dt} = W_{N_2,in} - W_{N_2,out} \quad (10)$$

$$\frac{dm_{w,ca}}{dt} = W_{v,ca,in} - W_{v,ca,out} + W_{v,gen} + W_{v,mbr} \quad (11)$$

Where inlet mass flow rate of oxygen, nitrogen and vapor is related with mass flow rate and humidity of inlet air.

$$W_{O_2,in} = y_{O_2} / (1 + \Omega_{atm}) W_{ca,in} \quad (12)$$

$$W_{N_2,in} = y_{N_2} / (1 + \Omega_{atm}) W_{ca,in} \quad (13)$$

$$W_{v,ca,in} = \Omega_{atm} / (1 + \Omega_{atm}) W_{ca,in} \quad (14)$$

y_{O_2} and y_{N_2} is oxygen mole fraction and nitrogen mole fraction of dry air.

$$y_{O_2} = x_{O_2} \frac{M_{O_2}}{M_a^{atm}} \quad (15)$$

$$y_{N_2} = (1 - x_{O_2}) \frac{M_{N_2}}{M_a^{atm}} \quad (16)$$

Where x_{O_2} is volume content of oxygen in dry air. The air molar mass is calculated by

$$M_a^{atm} = x_{O_2} M_{O_2} + (1 - x_{O_2}) M_{N_2} \quad (17)$$

The humidity ratio is then

$$\Omega_{atm} = \frac{M_v}{M_a} \frac{\phi_{atm} P_{sat}^{atm}}{P_{atm} - \phi_{atm} P_{sat}^{atm}} \quad (18)$$

The pressure of cathode is the sum of oxygen partial pressure, nitrogen partial pressure and vapor partial pressure.

$$P_{ca} = P_{O_2} + P_{N_2} + P_{v,ca} \quad (19)$$

According to ideal gas law [4], the partial pressure of oxygen, nitrogen and vapor is

$$P_{O_2} = \frac{m_{O_2} R_{O_2} T_{st}}{V_{ca}} \quad (20)$$

$$P_{N_2} = \frac{m_{N_2} R_{N_2} T_{st}}{V_{ca}} \quad (21)$$

$$P_{v,ca} = \frac{m_{v,ca} R_v T_{st}}{V_{ca}} \quad (22)$$

Outlet gas flow rate is calculated by [1]

$$W_{ca,out} = k_{ca,out} (P_{ca} - P_{rm,ca}) \quad (23)$$

Where the mass flow rate of each species out of the cathode is calculated as

$$W_{O_2,out} = \frac{m_{O_2}}{m_{ca}} W_{ca} \quad (24)$$

$$W_{N_2,out} = \frac{m_{N_2}}{m_{ca}} W_{ca} \quad (25)$$

$$W_{v,ca,out} = \frac{p_{v,ca} V_{ca} M_v}{RT_{st} m_{ca}} W_{ca} \quad (26)$$

where $m_{ca} = m_{O_2} + m_{N_2} + \frac{p_{v,ca} V_{ca} M_v}{RT_{st}}$ is the total mass of the cathode gas.

The rate of hydrogen consumed in the reaction and water generated in the reaction is calculated by

$$W_{H_2,rc} = M_{H_2} (nI_{st}) / (2F) \quad (27)$$

$$W_{v,gen} = M_{H_2O} (nI_{st}) / (2F) \quad (28)$$

c) Water transport process in membrane

The water transport across the membrane is achieved through two distinct phenomena [5]. First, water molecules are dragged across the membrane from anode to cathode by the hydrogen proton. This phenomenon is called electro-osmotic. Secondly, the gradient of water concentration across the membrane results in “back-diffusion” of water, usually from cathode to anode. The water concentration is assumed to change linearly over the membrane thickness. Combining the two water transport mechanisms, the water flow across the membrane from anode to cathode is

$$W_{v,mem} = M_v A_{fc} n \left(\frac{n_d I_{st}}{F} - D_w \frac{c_{v,an} - c_{v,ca}}{t_m} \right) \quad (29)$$

The electro-osmotic drag coefficient n_d and the diffusion coefficient D_w vary with water content in the membrane.

$$n_d = 0.0029 \lambda_m^2 + 0.05 \lambda_m - 3.4 \times 10^{-19} \quad (30)$$

$$D_w = D_\lambda \exp \left(2416 \left(\frac{1}{303} - \frac{1}{T_{st}} \right) \right) \quad (31)$$

Where

$$D_\lambda = \begin{cases} 10^{-6}, & \lambda_m < 2 \\ 10^{-6}(1 + 2(\lambda_m - 2)), & 2 \leq \lambda_m \leq 3 \\ 10^{-6}(3 - 1.67(\lambda_m - 3)), & 3 < \lambda_m < 4.5 \\ 1.25 \times 10^{-6}, & \lambda_m \geq 4.5 \end{cases} \quad (32)$$

Membrane average water content λ_m is calculated by [6]

$$\lambda_m = \begin{cases} 0.043 + 17.81 a_m - 39.85 a_m^2 + 36.0 a_m^3, & 0 < a_m \leq 1 \\ 14 + 1.4(a_m - 1), & 1 < a_m \leq 3 \end{cases} \quad (33)$$

Where

$$a_m = \frac{a_{an} + a_{ca}}{2} \quad (34)$$

water activities in anode and cathode a_{an} and a_{ca} is calculated by

$$a_{an} = \frac{P_{v,an}}{P_{sat,an}}, \quad a_{ca} = \frac{P_{v,ca}}{P_{sat,ca}} \quad (35)$$

4 Simulation Result

The feature of this model is the integration of all possible water dynamic equation by including inlet vapor in anode and cathode, outlet water in anode and cathode, water generated in the reaction and water across the membrane. The dynamic model is implement using Scilab.

Scilab is a scientific software package for numerical computations providing a powerful open computing environment for engineering and scientific applications. Developed since 1990 by researchers from INRIA and ENPC, it is now maintained and developed by Scilab Consortium since its creation in May 2003. Distributed freely and open source via the Internet since 1994, Scilab is currently being used in educational and industrial environments around the world [7]. The differential equation is solved by ode function and figure is drawn by polt2d function.

Typical values of parameters used in the model are tabulated in Table 1.

Table 1. Parameter used in the model

Symbol	Variable	Value
M_{H_2}	hydrogen molar mass	2.016 $\times 10^{-3}$ kg/mol
M_{O_2}	oxygen molar mass	32×10^{-3} kg/mol
M_{N_2}	nitrogen molar mass	28×10^{-3} kg/mol
R_{H_2}	hydrogen gas constant	4124.3 J/(kg · K)
R_{N_2}	nitrogen gas constant	296.8 J/(kg · K)
R_{O_2}	oxygen gas constant	259.8 J/(kg · K)
R_v	vapor gas constant	461.5 J/(kg · K)
R_a	air gas constant	286.9 J/(kg · K)
P_{atm}	atmospheric pressure	101.325kPa
n	number of cells in fuel cell stack	381
A_{fc}	fuel cell active area	280cm ²

V_{an}	anode volume	0.005 m ³
V_{ca}	cathode volume	0.01 m ³
F	Faraday number	96485

Simulation result show that the factors of affect water content in membrane are constant k_{out} , inlet air flow rate, water flow rate inject into cathode and current of fuel cell stack. First, we discuss constant k_{out} . Note that Equation (23) indicates that k_{out} can be viewed as a conductivity for an electrical circuit (inverse of a resistance), so the larger the k_{out} is, the larger the flow is through the fuel cell for a given back-pressure. Figure 1 shows the water mass in cathode when k_{out} change. The simulation result show that water mass of cathode decreases when k_{out} increases. Moreover, water mass increases as linearity when k_{out} is equal to 0.75×0.00001 . This shows that water in cathode loses its balance and accumulate in cathode when k_{out} less than set point. Figure 2 shows the water mass in cathode in different stack current. The simulation results show that water mass of cathode increases when stack current increases. Figure 3 shows the water mass of cathode in different inject water flow rate W_{inj} in cathode. The simulation results show that water mass of cathode increases when W_{inj} increases. Moreover, water mass increases as linearity when W_{inj} is equal to 70.5×0.001 . This shows that water in cathode loses its balance and accumulate in cathode when W_{inj} larger than set point.

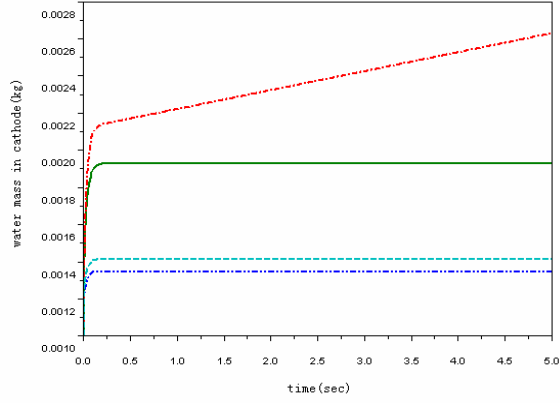


Fig. 1. water mass of cathode in different k_{out} .
 bigdash dot line: $k_{out} = 10 \times 0.00001$, dash line: $k_{out} = 5 \times 0.00001$, solid: $k_{out} = 1 \times 0.00001$,
 longdash dot: $k_{out} = 0.75 \times 0.00001$.

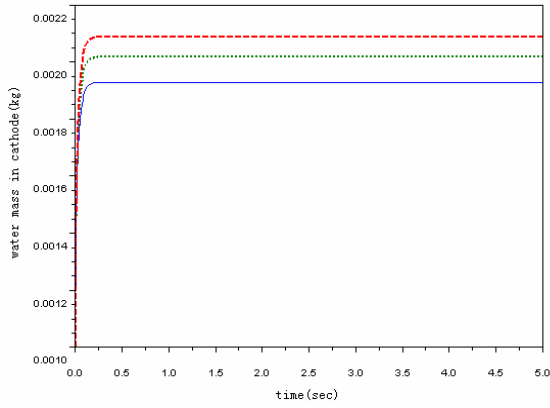


Fig. 2. water mass of cathode in different stack current

dash line: $I=280$, dash dot line: $I=200$, solid: $I=100$.

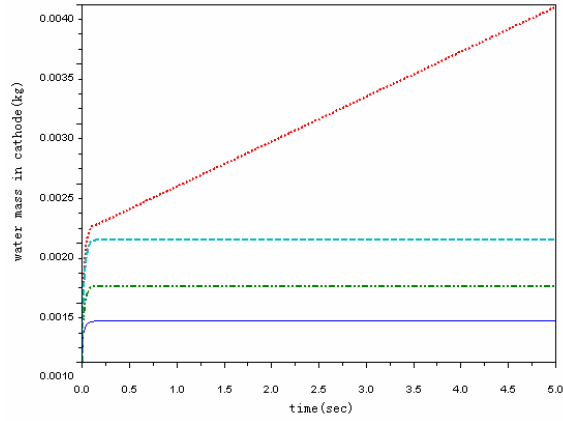


Fig. 3. water mass of cathode in different inject water flow rate

dash dot line: $W_{inj} = 70.5 \times 0.001 \text{ kg/s}$, dash line: $W_{inj} = 60 \times 0.001 \text{ kg/s}$, longdash dot: $W_{inj} = 30 \times 0.001 \text{ kg/s}$, solid: $W_{inj} = 10 \times 0.001$

5 CONCLUSION

In this paper, water balance equation is developed to simulate the transient phenomena in a proton exchange membrane fuel cell (PEMFC). This model is built by investigating water distribute of membrane and water transport process in cathode and anode. The proposed model is implement in Scilab environment. This paper discuss the parameters (return manifold in cathode k_{out} , mass flow rate of inject water W_{inj} and current of stack I)

of water management in PEMFC stack. The simulation results show that k_{out} and W_{inj} are

important in water management. k_{out} is determined by structure of PEMFC stack and W_{inj} is

determined by operation condition which need to control in stack work. This is used in designing the structure of return manifold and optimal control of operation parameter. The simulation results show that Scilab is a powerful and convenient tool.

NOMENCLATURE

A_{fc}	Fuel cell active area (cm^2)
F	Faraday's number (Coulombs)
I	Stack current (A)
J	Rotational inertia ($\text{kg} \cdot \text{m}^2$)
M	Molecular Mass (kg/mol)
P	Power (Watt)
R	Gas constant or electrical resistance (Ω)
T	Temperature (K)
V	Volume (cm^3)
W	Mass flow rate (kg/sec)
a	Water activity
c	Water concentration (mol/cm^3)
m	Mass (kg)
n	Number of cells
p	Pressure (Pa)
t	Time (sec)
t_m	Membrane thickness (cm)
ϕ	Relative humidity

Subscripts

act	Activation Loss
air	Air
an	Anode
ca	Cathode
$conc$	Concentration Loss
fc	Fuel cell
gen	Generated
in	Inlet

<i>m</i>	Membrane
<i>out</i>	Outlet
<i>rm</i>	Return manifold
<i>sm</i>	Supply manifold
<i>st</i>	Stack
<i>v</i>	Vapor
<i>w</i>	Water

REFERENCES

- [1] J T Pukrushpan, A G Stefanopoulou and Huei Peng. Control of fuel cell power systems: Principles, model, analysis and feedback design. Springer-Verlag London, 2004.
- [2] Buchi F and Srinivasan S. Operation proton exchange membrane fuel cells without external humidification of the reactant gases, *Journal of Electrochemical Society*, 144(8): 2767-2772, 1997.
- [3] Jay T. Paukrushpan, Anna G. Stefanopoulou and Huei Peng, "Control of fuel cell breathing" *IEEE Control Systems Magazine*, vol. 24, issue 2, pp. 30-46, April 2004.
- [4] Sonntag, R., Borgnakke, C., and Van Wylen, G. Fundamentals of Thermodynamics. Wiley, New York, fifth edition, 1998.
- [5] Nguyen T and White R. A water and heat management model for proton-exchange-membrane fuel cells. *Journal of Electrochemical Society*, 140(8), 2178-2186, 1993.
- [6] T E Springer, T A Zawodzinski and S Gottesfeld. Ploymer electrolyte fuel cell model. *Journal of Electrochemical Society*, 138(8): 2334-2342, 1991.
- [7] A Free Scientific Software Package, www.scilab.org.

Electrostatic precipitator modeling and simulation

Kejie Fang¹, Longhua Ma¹, Weiguang Chen¹, Qinglong Jiang², Zhiping Zhang²

¹Institute of Industrial Control, Zhejiang University, Hangzhou, China
{kjfang, lhma, wgchen}@iipc.zju.edu.cn

²Department of research and development, Zhejiang Jiahuan Electronic Co. Ltd, Jinhua, Zhejiang Province, China
jhjql@126.com, chenyingzhang@sina.com

Abstract. Electrostatic precipitator is a device for removing gas-particles. In this paper, we study on the mechanism and mathematical model of ESP which contains several parts such as electric field, particle charge, particle collection, and design a platform about ESP modeling and simulation based on Scilab. As a result, it has chosen to neglect some uncertainty factors, but the simulation results accord with the literarily data.

Keywords: electrostatic precipitator (ESP); complexity; modeling; simulation;

1 Instruction

The electrostatic precipitator (ESP) is a commonly used device for removing gas-particulates mixtures exhausted by various industrial processes in so many different domains. The most commonly used electrostatic precipitators are the wire-plate form as shown in Fig.1 [1]. A series of equally spaced vertical corona wires are placed between two vertical parallel plates. The flat plates are grounded and the corona wires are at a high voltage. Under the work of researchers and experts, electrostatic precipitators have reached fairly good collection efficiency. However, the complexity of the process of the ESP limits the capacity of increasing efficiency further. Since seeing the limitation, more work need to be done and have to be done. Knowing that increase the complexity of the model, better results can be achieved, different ways and ideas on the modeling emerged and are proved to be effective.

2 Principles of ESP

Generally, the process of electrostatic precipitator is known as three main stages: particle charging, transport and collection [2] and these are interacted that originated from the complexity of the process of precipitator. To characterize all these stages determines to take a great number of basic phenomena into account from a physical point of view when they occurred.

Usually the following presentations are all based on plate-wire construction shown as Fig.1 [1]. Particle charging is the first and foremost beginning in the process. The particle charging is described as two main processes: the field charging (dominant at particles diameter $d > 2 \mu\text{m}$) and the diffusion charging (diameter $d < 0.2 \mu\text{m}$) [1]. When the flue mixed-gases are injected through the inlet of the precipitator, the motion of the suspended particulates is largely dominated by the motion of the gas flow. Apply high voltage on the precipitator when work, the situation in interelectrode space will be changed significantly. As the applied voltage reach the threshold value, the space inside can be divided into two parts, ionization region and drift region (the wire diameter is much smaller compared with the distance to the plate). The electric field magnitude around the negative electrode is so strong that the electron is possible to get enough energy and speed to crash towards the neutral molecule and help the electron run out of the control of the molecule. This kind of process so-called as corona discharge produces abundant electrons and positive ions. Under the influence of electric field, the positive ions move towards the corona, while the negative ions and electrons towards the collecting plates. In the moving way, negative ions cohere and charge the particles, make the particles be forced towards collecting-plate as well.

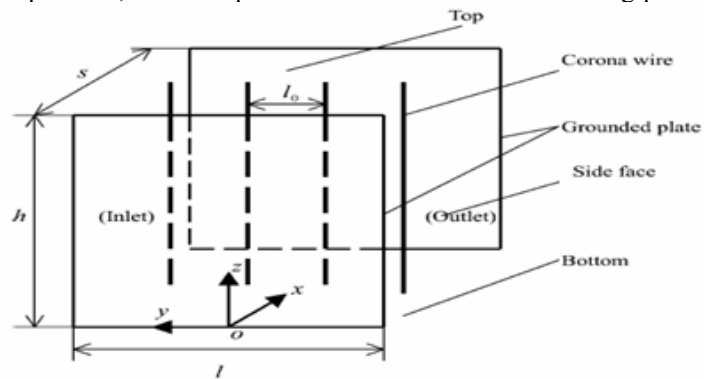


Fig. 1. Schematic of wire-plate electrostatic precipitator

Fig.2 shows the mechanism of charging. Particle charging considers the charge transfer from the ions to the particles and their progressive charging, starting from the inlet conditions up to saturation [3].

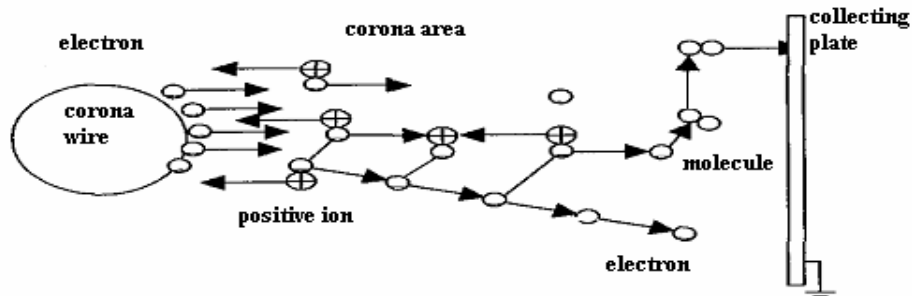


Fig. 2. Mechanism of electrostatic precipitator

In generally say, the corona discharge at these wires cause ions to traverse the interelectrode space to charge dust particles. The charged particles move toward the grounded plates under the influence of the electric field and are deposited on the plates. In the process of particle collection in an electrostatic precipitator, the particle motion is affected by both the electrostatic field and gas flow.

3 Research and design in ESP modeling

The numerical model describes in time and space the relevant mechanical, electrical, physical and chemical processes that are involved in transport, charging, migration and collection of fly ash. To represent the complete processes, the model is therefore structured into several modules, each one dealing with a particular process [3].

The model here is organized into the following three sections:

- electric field and discharge processes
- particle charging
- particle collection

3.1 electric field and discharge processes

The particle collection in electrostatic precipitators is largely dominated by the distribution of the electric field in the interelectrode space, which influences almost all the physical processes that occur in electrostatic precipitation [4]. The field distribution defines the characteristics of the corona discharges (either continuous glow or pulsed streamer) and therefore the ionic charge flux; the local electric field influences the particle charging and their saturation charge; the field magnitude defines the force that drags the particles towards the collecting plates; the field at the plane defines the electrostatic pressure that keeps compact the surface layer of by ash and therefore influences the particulate reentrainment by hydrodynamic erosion in correspondence of rapping.

In the absence of particles, neglecting the transport gas velocity and by assuming that the magnetic field due to the corona current is negligibly small, the electrical conditions are described by next three equations [7]:

$$\nabla^2 V = -\frac{\rho_e}{\epsilon_0} \quad (1)$$

$$\rho_e^2 = \epsilon_0 (\nabla V \cdot \nabla \rho_e) \quad (2)$$

$$\vec{E} = -\nabla V \quad (3)$$

Where V is the electric potential, ρ_e is the space-charge density, ϵ_0 is the permittivity of free space and E the electric field.

In this paper, we adopt equations (4) (5) (6) to describe and compute the electric field distribution with the initial and boundary conditions [8].

$$V(x, y) = V_0 \frac{\sum_{m=-\infty}^{\infty} \ln \left\{ \frac{\cosh[\pi(y - 2mS_y) / 2S_x] - \cos(\pi x / 2S_x)}{\cosh[\pi(y - 2mS_y) / 2S_x] + \cos(\pi x / 2S_x)} \right\}}{\sum_{m=-\infty}^{\infty} \ln \left\{ \frac{\cosh(\pi m S_y / S_x) - \cos(\pi a / 2S_x)}{\cosh(\pi m S_y / S_x) + \cos(\pi a / 2S_x)} \right\}} \quad (4)$$

Where $V(x, y)$ is the electric potential of the position (x, y) , V_0 is the initial potential on the wire, S_x is distance between collecting plate and wire, S_y is half length of the two nearest wires, a is the radius of particle, when x, y means the coordinate lines direction, shown as Fig.3.

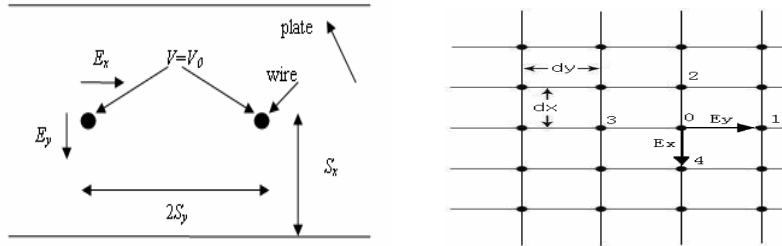


Fig. 3. Sketch of precipitator geometry and computed grid

$$\rho_0^2 = \epsilon_0 \left(\frac{\Delta V_0}{\Delta x} \frac{\Delta \rho_0}{\Delta x} + \frac{\Delta V_0}{\Delta y} \frac{\Delta \rho_0}{\Delta y} \right) = -\epsilon_0 \left(E_{0x} \frac{\rho_4 - \rho_2}{2d_x} + E_{0y} \frac{\rho_1 - \rho_3}{2d_y} \right) \quad (5)$$

$$V_0 = \frac{d_y^2 (V_4 + V_2) + d_x^2 (V_1 + V_3) + d_x^2 d_y^2 \rho_0 / \epsilon_0}{2(d_x^2 + d_y^2)} \quad (6)$$

ρ and V_0 mean the charge density and electric potential at the position as Fig.3 shown.

First calculate the electric potential by equation (4), then compute equation (5) (6) iteratively to get the last electric field distribution.

3.2 particle charging

The field charging refers to the local distortion caused near the particle surface by the difference in dielectric constants: as a consequence the ions moving along the electric field lines fall on the particle surface. This process continues until the particle goes up

to the saturation charge, which produces an electric field on particle surface equal and opposite to the external field.

Equation (7) is chosen to describe the model of particle charging [8].

$$q_s = 12\pi\epsilon_0 R^2 \frac{\epsilon_r}{\epsilon_r + 2} E_0 \quad (7)$$

Where ϵ_r is the relative dielectric constant and E_0 is the external field, q_s and R are the particle charge and radius.

The diffusion charging takes into account the possible impact of ions on the particle surface due to their stochastic thermal velocity.

3.3 particle collection

The module simulates in detail the boundary layer near the collecting plates and the interchange that take place.

The following boundary conditions are assigned [3]:

- Towards the bulk of the gas flow: as defined on the surface of the last cell used in the gas flow module.
- Towards plate surface: as computed by the rapping-reentrainment module which describes the reentering of particulate into the interelectrode space.

At each time step the rate of dust particles attached on the collecting plates is evaluated for every size class k and charge class j . In this way the characteristic of the deposited layer can be determined.

Here, we choose equation (8) to describe particle collection [7].

$$C = C_0 \cdot \exp\left(-\frac{a}{f} \left(\frac{w \cdot y}{v} - x\right)\right) \quad (8)$$

Where C is the particle density at every assigned point in ESP, C_0 is the entry density of particle, a is the unit collecting area in the flow way, f is area of ESP cross section, when w means particle velocity towards plate and v is the velocity moving to outlet.

4 Simulation results and analysis

According the above analysis of the mechanism and modeling of ESP, we design a simple ESP simulation platform which is based on Scilab [5] [6]. We enter the values of all needed variables as Fig.4 shown, and do the computation by Scilab, we get the results and the figures help us to do analysis easily.

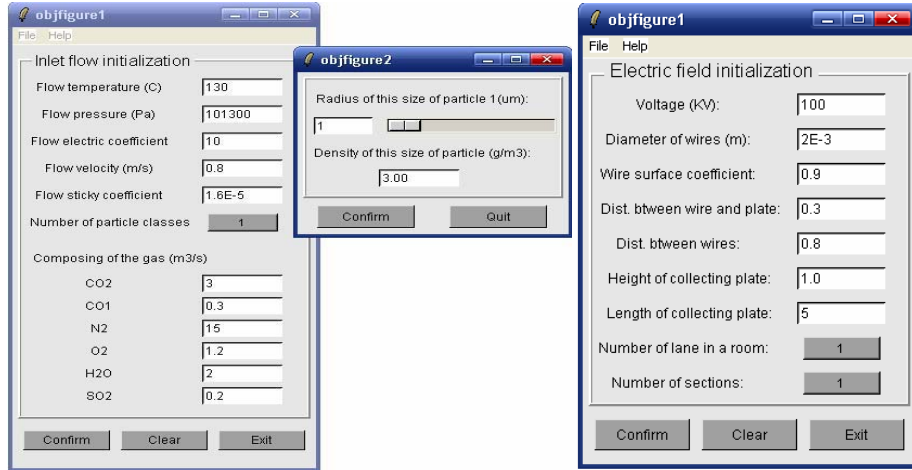


Fig. 4. Initial and boundary conditions

4.1 Simulation of electric field

Fig.5 is the distribution of the electric field in the direction of coordinate line x, we can find that around the wires E_x get a largest value, when at the connecting way of two wires, E_x is no more than zero. The cause of this distribution is the potential, at the connecting way of wires, nearly zero. E_x is decreased regularly from the wire at the coordinate line x, but larger when close to the collecting plate.

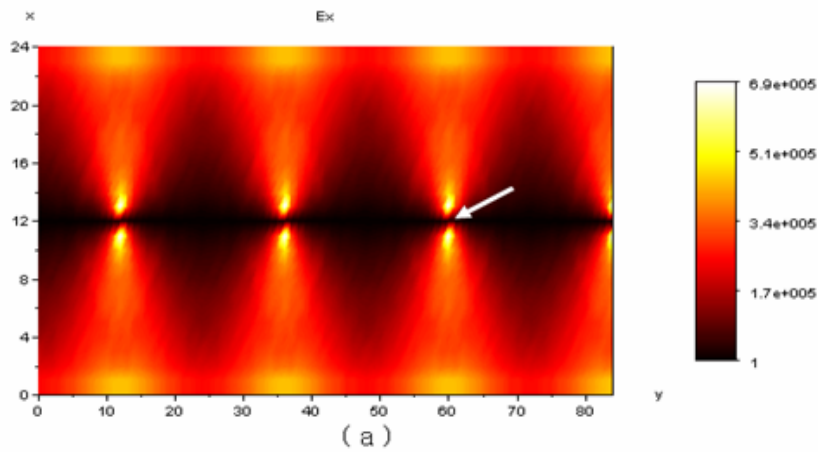


Fig. 5. Distribution of electric field E_x

Fig.6 is the distribution of the electric field in the direction of coordinate line y.

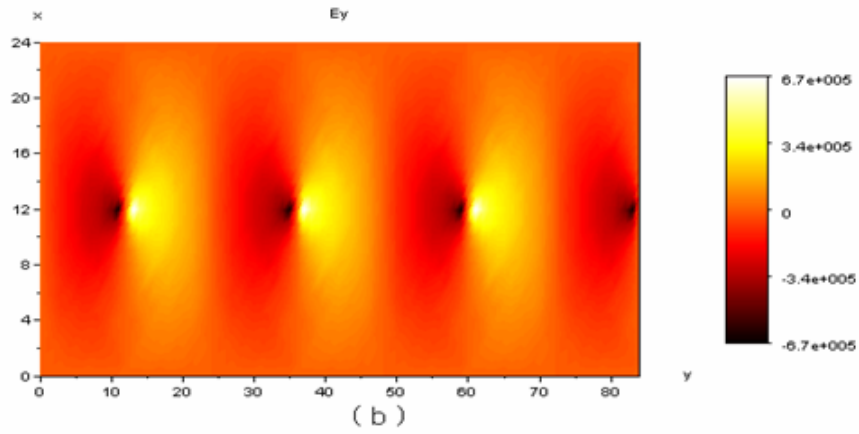


Fig. 6. Distribution of electric field E_y

Fig.7 is the distribution of the electric field from three-dimensional view.

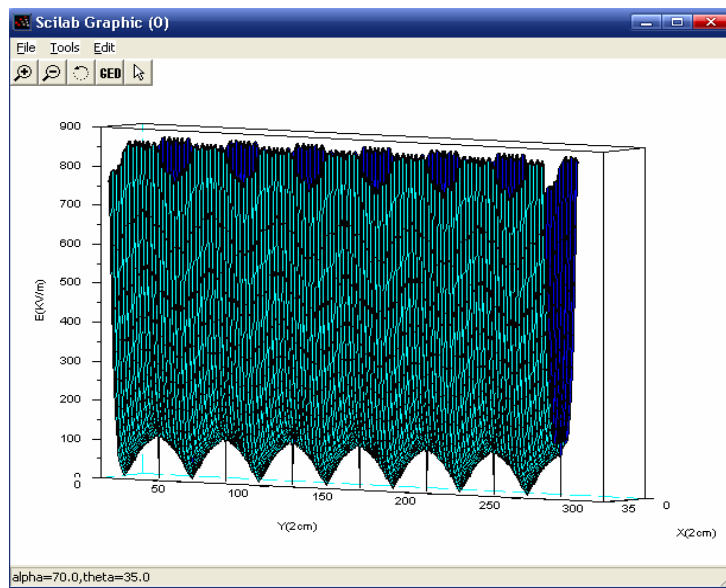


Fig. 7. Three-dimensional view of electric field distribution

This phenomenon is the same as the thesis of Leutert and Bohlen, and has the same result with Tassicker [9].

4.2 Simulation of particles density distribution

This part we present particles density distribution as Fig.7 and Fig.8 shown from two and three dimensional views respectively.

From Fig.8, we see the particles density distribution obviously. The density reaches the largest value at the entry of the ESP under the influence of electric wind. The value of density gets smallest near the wire at the direction to collecting plate. For at this point and this direction, electric potential is large and the electric wind press particles move to plate swiftly. We also can see density at the way of wires is nearly no change with the entry value. It is because we neglect the influence of gravity and other factor from the point of simplify. In real condition, this will not happen, and the particles can reach the outlet along the wires way.

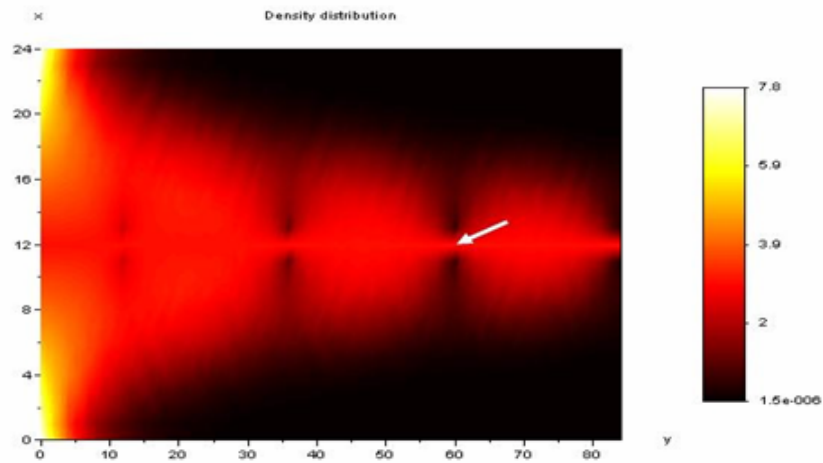


Fig. 8. Particle density distribution in ESP

Fig.9 shows us the particle density distribution from three-dimensional view to help us understand this phenomenon clearly.

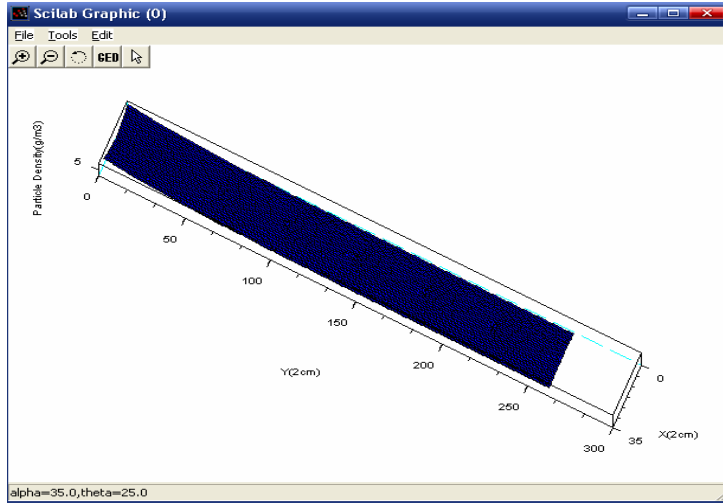


Fig. 9. Three-dimensional view of particle density distribution

4.3 Simulation of deposit density

Fig.10 shown the deposit density, we see along the collecting plate the deposit density is decreased definitely, since with the time go on, the particle is collected by the plate continuously. So at the later part, the deposit density is lower, and reasonable.

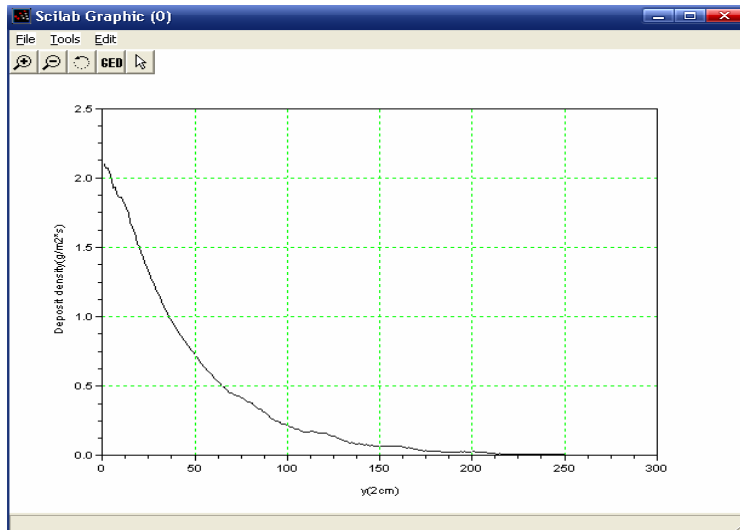


Fig. 10. The distribution of deposit density

5 Conclusion

In this paper, we construct a numerical model of electrostatic precipitator, it takes into account the principal physical processes, and designed base on Scilab. The simulation of these processes including electric field, particle charging, and particle collection aim to help us understand the complexity of ESP. Simulation results are according with laboratory experimental tests to obtain physical information and useful validations. Under deeply work, considering other influencing factors, we can describe different physical processes more accurately occurring within the precipitator cells.

Acknowledgments. This work is supported by a grant from the National Natural Science Foundation of China (No. 60474064) and the Fund of Science and Technology Cooperation of Jinhua City.

References

1. Hong Lei, Lian-Ze wang, Zi-Niu Wu: Applications of upwind and downwind schemes for calculating electrical conditions in a wire-plate electrostatic precipitator. *Journal of Computational Physics* 193 (2004)697-707
2. I.Kiss, I.Berta: New concept of ESP modeling based on fuzzy logic. *Journal of Electrostatics* 51-52 (2001) 206-211
3. I.Gallimberti: Recent advancements in the physical modeling of electrostatic precipitator. *Journal of Electrostatics* 43 (1998) 219-247
4. E. Lami, F. Mattachini, R.Sala, H. Vigl: A mathematical model of electrostatic field in wire-plate electrostatic precipitator. *Journal of Electrostatics* 39(1997)1-21
5. Baogang Hu, Xing Zhao, Mengzhen Kang: Introduction to Scientific Computing Software-SCILAB. Tsinghua University Press, Beijing (2003)
6. Brent B. Welch: Practical programming in Tcl/Tk, Prentice Hall, Inc (2002)
7. Guoquan Zhang: aerosol mechanics, China environmental science press, Beijing (1987)
8. Sabert Oglesby, Jr., Grady B.Nichols: Electrostatic Precipitation, Marcel Dekker, Inc, New York (1978)
9. O.J.Tassicker, Ph.D.Dissertation, Dept.of Elec. Eng., Wallongong University College, Univ. New South Wales, Australia (1972)

Function versus geometry – A dichotomy for design engineers

Mr Gareth Black, Dr David Allanson and Dr Steve Douglas

School of Mechanical Engineering, Liverpool John Moores University, Liverpool, L3 3AF, UK
G.Black@ljmu.ac.uk, D.R.Allanson@ljmu.ac.uk, S.S.Douglas@ljmu.ac.uk

Mr Jonathan Fox and Mr Brian Travis

Delta Fluid Products, Delta Road, St Helens, WA9 2ED, UK
Jon.Fox@DeltaFluidProducts.com, Brian.Travis@DeltaFluidProducts.com

Abstract: This paper presents an overview of the traditional design process which can be found in many SME's (small to medium sized enterprises) and discusses the dilemma of how to improve their design processes without recourse to expensive CAE software. The approach advocated in this paper is to extract the essence of the design into an engineering science (mathematical) model to create a virtual design. This approach is compatible with transformation into an integrated simulation model developed in Scilab and Scicos. The advantages of initiating the design process via simulation are shown particularly in regard to demanding global economic constraints. The concept of applying simulation methods to capture design knowledge is also discussed. This paper hopes to banish the mystique that surrounds the application of simulation in the industrial design context.

1 Introduction

“Better products in less time and for less money”. In the face of tight economic constraints, these are the seemingly conflicting goals facing many Small to Medium sized Enterprises (SME's) in the UK. To achieve these goals, the focus is placed firmly on engineering departments to maximize the efficiency of their design processes. An efficient design process will ensure that the final design is realized with informed decisions and is based upon sound engineering principles. This final design will be completed in a timely fashion, ensuring that the time to market is minimized, thereby maximizing potential profits. The tooling costs associated with mass produced items are large, so getting the design right first time is imperative to avoid unnecessary expenditure.

The majority of design processes are tightly constrained by standards, regulations and customer requirements. The design process must be able to handle changes in standards or regulations and be able to respond quickly to changes in customer requirements.

In the section two of this paper, current design methods employed by most SME's are introduced and their associated shortcomings discussed. Presented in section three is an approach for achieving substantial cost savings within the design process. This is followed by an example of the successful application of this improved design process. Finally, in section four, the concept of embedding knowledge within the design process is introduced.

2 Current design methods

Stringent economic constraints and rising labour costs mean that SME's are invariably under pressure to cut costs and reduce staff. Engineers within SME's routinely perform multiple functions, such as attending to manufacturing and materials issues and to changing designs. These activities, though necessary all serve to reduce the amount of time that the engineer can spend focused upon the development of new or improved products.

2.1 The traditional design process

In respect to the design process, most SME's still employ a very traditional methodology. This is shown schematically in figure 1 and consists of a series of linear steps leading to the communication of the solution. This process is led by experienced engineers using rules of thumb, intuition and their past experience to drive new product development. The solution picked from the alternatives generated is subject to analysis, evaluation, refinement and development (Cross, 1998). Occasionally, the analysis leads to the discovery of fundamental flaws in the selected alternative, which then has to be abandoned and the design cycle started again. According to Boothroyd, 55% of the process is spent doing design changes. Most SME's generally employ 3D CAD throughout this process as a vehicle for generating alternative solutions and communicating the final design.

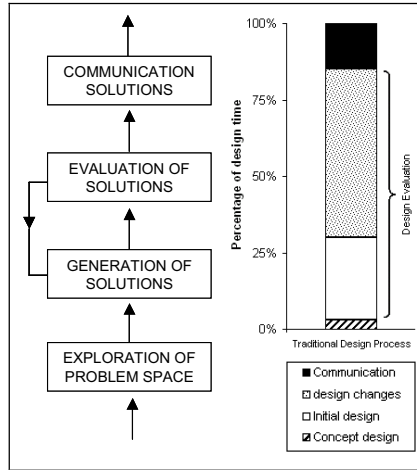


Figure 1 - The traditional design process (Cross, 1998 and Boothroyd et al)

The effectiveness of the traditional design process can be assessed by considering three performance parameters: design freedom, cost committed and design knowledge respectively. Shown by figure 2, in the traditional design process, engineers seemingly cause most of the product cost when, paradoxically, they know least about the product, which in the extreme may result in poor performing and expensive products. Boothroyd estimates that 70% of the product cost is committed during the conceptual phase of the design process.

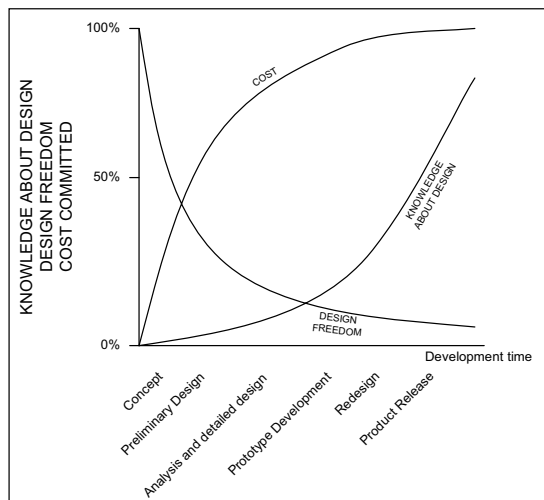


Figure 2 - Effectiveness of the traditional design process (Pinfold et al, 1998 and Shah, 1996)

Within this strategy, new products are relatively successful if the designer has the in-depth knowledge associated with years of experience within a particular field (Shaw,

2006), but less experienced engineers require input from senior members of staff. This is all well and good if there is sufficient senior staff. However, if this is not the case, who does the less experienced engineer turn to for advice?

Once the geometry has been fixed, the product will undergo a number of design/development iterations, usually based upon the performance of a physical test. These tests entrain substantial costs. In this approach, a body of knowledge regarding new products is derived through this physical testing, which means that most of the useful information is not made available until the end of the design process! Unfortunately, by this time, there is “no going back” for the engineer and if their product does not function as well as desired it is made to work via design modification and the disadvantages of this are obvious. A famous example of “making the design work” was the original Mercedes A-Class, which famously failed the Elk-test in 1997. The Elk-test assessed the ability of a vehicle to avoid an obstacle at speed, apparently designed to ensure that drivers can avoid the elks which roam freely on rural roads in Sweden. Mercedes had to make the car more stable by applying some complex electronics and stiffening the suspension, all of which lead to a less than optimal ride for the driver and passengers and served to increase the cost of the final product.

2.2 Levering simulation into the design process

In an attempt to improve the efficiency of development processes, many SME’s are looking to apply advanced simulation tools into their design system, such as multibody simulation, CFD, rapid prototyping technology and Finite Element Analysis (FEA).

One of the first simulation tools was FEA. Initially, these demanding analysis tools were the province of the analyst who used their knowledge to run them (Thilmany, 2004 and Pullin, 2005). In the context of design, the disadvantage of this approach was that the designer was not actively involved in the simulation effort. Advances in the graphical user interfaces (GUIs) have now led to the state where FEA can be embedded with 3D modelling packages such as SolidWorks and Inventor (Horspool, 2005).

Embedding the FEA capability within the solid modelling software obviously means that geometry is needed before any simulation can be performed. The idea that the “CAD is the star” (Uchitel) i.e. all analysis is based around a CAD model, is not limited to FEA, but is the same for all of the advanced simulation. So, similarly to the traditional design process, engineers need to commit to a particular geometry using their own past experience before any knowledge has been derived through the analytical tools. Thus even with this approach, many iterations involving the design geometry are necessary.

To quantify the improvements in efficiency gained by employing these advanced simulation tools, the key performance parameters can be compared with the traditional design process.

Figure 3 quantifies the improvements in efficiency that can be gained over the traditional design process by employing advanced simulation tools. The graph shows that the knowledge about the design is improved by implementing the advanced simulation, indeed some product knowledge may never be obtained without resorting to FEA or CFD. The cost committed is not vastly different with the improved design process because the engineer commits to geometry before any simulation even takes place. Design freedom is marginally improved because models remain in the “virtual” domain for much longer and therefore reduce cost as design changes can be made at a later stage.

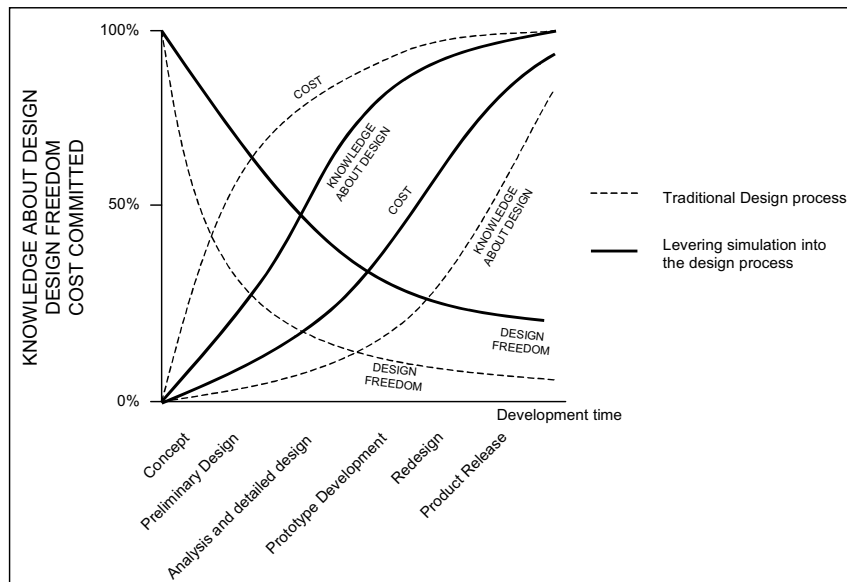


Figure 3 – Comparing the traditional design process and the improved process

Such advanced simulation tools are usually employed to validate the designs of an engineer and replace conventional physical testing. In this respect, these tools are extremely useful to companies where large-scale prototypes are very expensive, but for many SME’s the licensing costs and intellectual effort associated with the tools prohibit their widespread usage. The cost of deploying some of the simulation tools across even a small design department is significant. Single licenses usually cost tens of thousands of pounds to buy and have relatively high annual maintenance fees. Companies selling this software are characterized by strong corporate images, stylish websites and GUI’s and 24-hour global help lines. There is also a misconception

amongst many in the design community that experienced analysts are required to drive the simulation tools and convert CAD geometry.

The benefits associated with employing these advanced simulation tools are undisputable, but because of the costs and misconceived idea that analysts are required, simulation has not really moved into general industrial use.

3 Improving the design process

To achieve substantial reductions in design cost and time, a more elegant approach to simulation than those described above must be employed. The ideal goal of any design process would be to provide sufficient information to the engineer as early as possible, allowing informed decisions to be made and product costs to be minimized. Improved design process performance parameters are shown in figure 4 and contrasted with the simple and costly expedient of leveraging standard simulation into the design process to provide knowledge to the designer after geometry has been committed to.

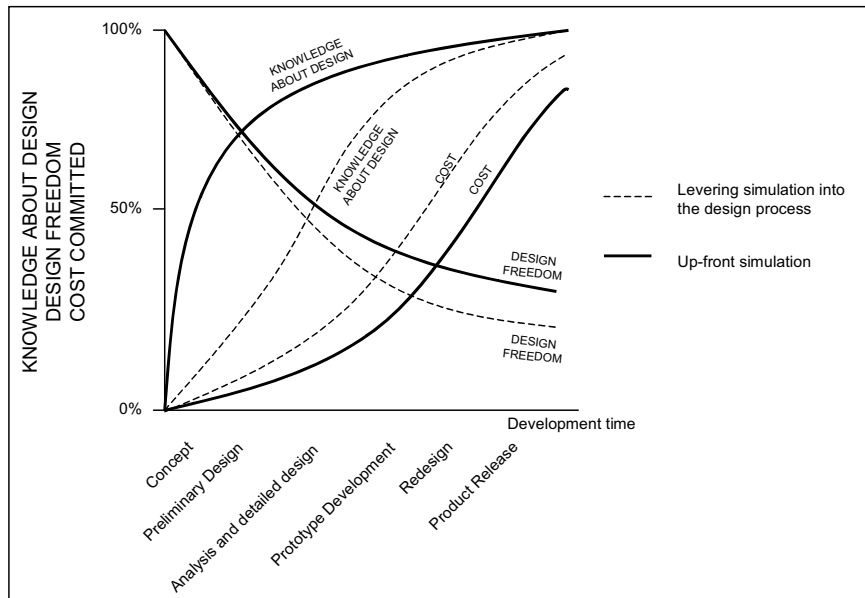


Figure 4 – Comparison of the improved design process and the process using upfront simulation

3.1 Using system simulation in the design process

It is the author's opinion and experience that simulation must be embedded at the start of the design process to achieve the performance of the improved design system shown in figure 4. The improvements are achieved by extracting the essence of the design from the geometry into a set of functional parameters representing both geometrical features and operating conditions. These parameters then form the basis of a design/system simulation. The virtue of this strategy is that the individual functions of a design alternative can be modelled and integrated to form a functional simulation of the design without committing to geometry and fixing cost into the product. Informed decisions can then be made upfront in the design process and a solution communicated much earlier than in the traditional approach. Figure 5 shows the estimated benefits of this approach and shows that by spending more time in conceptual analysis and simulating each alternative, more informed decisions can be made to ensure the design is right first time. There is an estimated potential time saving of 40% to be gained by adopting this approach, giving reduced time to market and increased profits. The advocated approach also enables the engineer to check the performance of alternatives against the requirements of the customer, thereby reducing the risk of a product that nobody will buy.

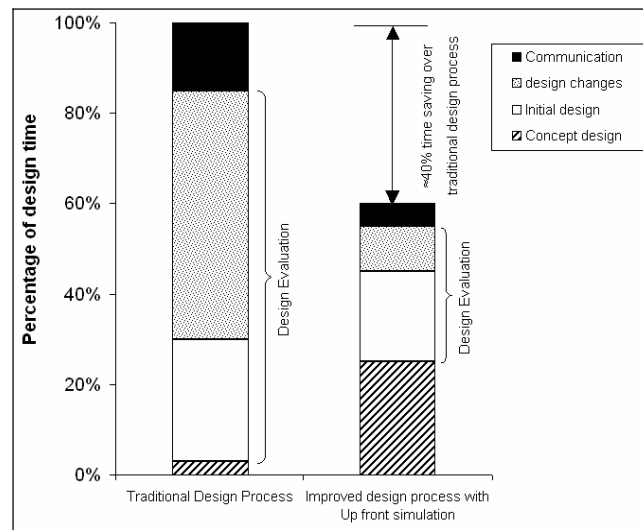


Figure 5 - The Improved design process

3.2 Open source software

A financial constraint upon the approach advocated is the commercial software packages used to implement these system simulations tend to be very expensive. This prohibits its widespread usage amongst the cost conscious engineer. However, in recent years, the use of open source software has emerged and is slowly finding

favour in academia and commerce. Even the UK government acknowledges its existence and recognises open source as “a viable and credible alternative to proprietary software” (OGC, 2003).

Scilab is an open source software package for scientific computation, which includes within it, the dynamical systems simulation package called Scicos. Scicos represents the ideal platform for implementing parametric system models, not only because it is free, but because of its good performance, ease of use and large online support community. Also, the free licence means that the simulations are not limited to individual computers, but can be used simultaneously across engineering departments, facilitating knowledge-based engineering.

The models are based upon parameters, which means that it can be directly linked into a CAD model. The result is that system simulation and solid modelling can be carried out concurrently with the results of each feeding into the other.

3.3 Practical application of the advanced design process

An example of the practical application of the advanced design tool, used by the author is within the analysis of gas pressure regulators to achieve improvements in product performance. An idealized model of a simple pressure regulator system to regulate pressure is shown schematically in figure 6. The regulator’s function is to maintain a constant outlet pressure regardless of flow rate and inlet pressure. The gas pressure is reduced from the inlet to the outlet pressure as it passes over the orifice, which is formed between the valve and the seat. The regulator is made up of three main elements: the spring, the plate and the valve respectively. Each of these elements is simulated individually by reducing its operation down to fundamental engineering science. Once each control element has been developed and validated, they are added to a palette of reusable elements.

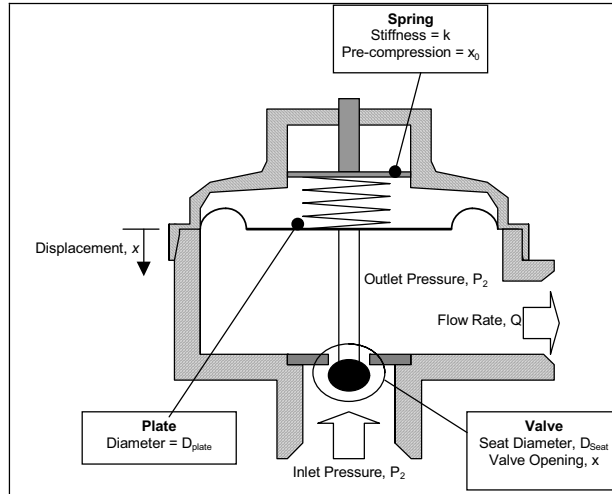


Figure 6 - Simple pressure regulator

Figure 7 shows the simulation of the simple pressure regulator. Here, the appropriate elements are selected off the palette and reconnected to represent the full pressure regulation system. The model can then be used to predict the performance of a system and test what-if scenarios, without the necessity of developing a solid or physical model. Insight is gained about the system performance by placing scopes or graphs within the model. In this way, during the conceptual analysis phase of the design process, the designer gains a detailed understanding of the product’s functional behaviour prior to any commitment to physical geometry.

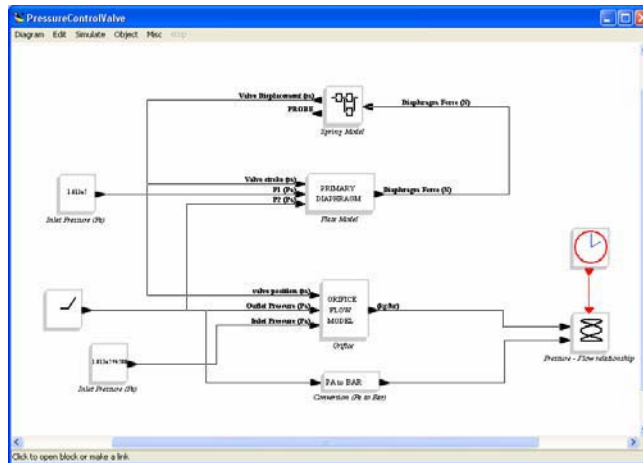


Figure 7 - Scicos model of the simple pressure regulator

Confidence in the models was developed by building simulations of current products and comparing the predictions with real physical results. Using the simulation, key

design changes were recommended and implemented, enhancing the physical product's performance.

4 Simulative methods and knowledge based engineering

One of the major drawbacks of the traditional design process is its reliance on the experience of senior members of staff and when one of these engineers' leaves/retires, a substantial portion of a company's knowledge is lost. To protect a company against the loss of these 'knowledge silos', experience and understanding must be embedded within the design process (Lovett et al, 1999).

The definition of a knowledge based engineering system is one that captures the expertise of individuals within a particular field of expertise (Lovett et al). Hence in our simulation strategy, knowledge is captured and represented through palettes of reusable control elements and system simulations. The palette is designed such that junior design staff could use this material with confidence. Furthermore, the reusable palette of control elements can be placed into a central store within the engineering department ready for deployment on new and current designs. By doing this, the knowledge is embedded into the design process and it is no longer necessary for the engineer to spend their time developing understanding of a company's standard components. With the engineer freed from the time consuming task of developing new understanding of current products, they can focus on improving a products design for manufacture (DFM), thereby reducing the manufacturing costs and maximizing profits.

The approach has many implications upon how design is taught within universities. Currently, the taught techniques follow the traditional design process. Simulation is taught as a discrete subject and this really needs to be embedded within the teaching of design. The implication of this is that university taught designers would have an understanding of both form and simulated function. This technique is really an updated form of the Bauhaus principles (Wikipedia).

5 Conclusions

In the author's opinion, SME's wishing to achieve better products in less time and for less money should consider integrating simulation in the initial stages and throughout the design process. The platform upon which these simulations should be implemented is Scilab/Scicos, which represents the best in open source software.

As a result of implementing simulations within the Scicos environment, the loss of knowledge associated with the departure of senior members of staff is minimized. SME's should place a priority on simulating their components and distributing models across their engineering departments.

Acknowledgements: The authors would like to thank, the Knowledge Transfer Partnership (Grant 665. www.ktponline.org.uk) and Delta Fluid Products (St Helens, UK. www.deltafluidproducts.com) for supporting this work.

References

1. Cross, N. (1998). *Engineering Design Methods*. John Wiley & Sons.
2. Chapman, C.B. & Pinfold, M. 1998. *Design engineering—a need to rethink the solution using knowledge based engineering*. Proceedings of the 1998 18th SGES International Conference on Knowledge-Based Systems and Applied Artificial Intelligence (ES98)
3. Shah, J., Hazelrigg H., “Research Opportunities in Engineering Design,” NSF Strategic Planning Workshop, Final Report, April 1996.
4. Boothroyd, G., Dewhurst, P. and Knight, W. (1994). *Product Design for Manufacture and Assembly*. Marcel Dekker, Inc.
5. Shaw, C. Feng and Eugene Y. Song. (2006). *Information Modelling of Conceptual Design*. Journal of Engineering Design. Taylor Francis.
6. Thilmany, J. (2004). *Flowing into the future*. Mechanical Engineering Magazine Online. ASME. <http://www.memagazine.org/backissues/feb04/features/flowinto/flowinto.html>
7. Pullin, J. *Stimulating Simulation*. Professional Engineering 10/5/2005, Vol. 18 Issue 18, p33
8. Horspool, John. *Give us the tools*. Professional Engineering, 3/9/2005, Vol. 18 Issue 5, p41, 2p
9. Uchitel, V. *Coupled Design and Analysis (CDA): Win the Race to Market*. ESI Group. Available online at http://www.esi-group.com/SimulationSoftware/CFD_CADalyzer/win_the_race.html
10. OGC (Office of Government Commerce). *Open Source Software Trials in Government – Final report*. Available online at http://www.ogc.gov.uk/embedded_object.asp?docid=1003914
11. Lovett, P., Ingram, A., Bancroft, C. (1999). *Knowledge Based Engineering for SME's – A methodology*. Proceedings of the 15th Annual Conference on Computer-Aided Production Engineering
12. Wikipedia. *Form Follows Function*. Available online at http://en.wikipedia.org/wiki/Form_follows_function

Improved Model of Secondary Clarifier and A²/O Process Modeling

Shengming Chen, Longhua Ma

Institute of Systems Engineering, Zhejiang University, Hangzhou, China, 310027
shmchen@iipc.zju.edu.cn

Abstract. The complex behavior of the secondary clarifier and its great importance for the successful operation of the activated sludge process have made the settling process a major issue for researchers working within the field of mathematical modeling. So a precise model of secondary clarifier can describe the factual behaviors of a sewage plant. But in ASM2, the solid/liquid separation that is high non-linear course in secondary clarifier is not considered. The solid/liquid separation affects not only the biology reaction but also the quality of output water. In this paper based on a one-dimensional settler model of secondary clarifier an improved secondary clarifier model including ASM2 is established in terms of the solid flux concept and a mass balance, Through giving some assumptions, a model including secondary clarifier is constructed on the A²/O process, and then simulates it with SCILAB.

Keywords: ASM2, Secondary Clarifier, Activated Sludge Process, A²/O Process

1 Introduction

Activated Sludge Process (ASP) is the most commonly used biological wastewater treatment process. It is considered to be the most cost-effective way to remove the organic materials from wastewater. Besides that, it is very flexible, reliable, and has the capability of producing high quality effluent. But the design and operation of ASP depend on experience highly. So establishing the activated sludge mechanism mathematic model is necessary. The model can instruct practice with theory and reduce the reliance on experience, and it is the outstanding handling tool of design and operation in municipal sewage plant^[12].

Currently, researchers have focused their all energy on Activated Sludge Model ((ASM1^[1], ASM2^[3], ASM3^[2]) published by International Water Association (IWA). Because of the absence of phosphorus removal in ASM1 and ASM3, it's a limitation especially in now days of more attention putting on phosphorus removal. So it is required that researching ASM2, building the suitable model for factual process, and putting the model into actualize on sewage plant^[11].

A secondary clarifier is an important part of the entire activated sludge system. The efficiency of system is closely combined with the design and running of the secondary clarifier. So a precise model of secondary clarifier will describe the behaviour of a

sewage plant. But in ASM2, the solid/liquid separation that is high non-linear course in secondary clarifier is not considered. The solid/liquid separation affects not only the biology reaction but also the quality of output water^[4]. In this paper based on a one-dimensional settler model of secondary clarifier an improved secondary clarifier model based on ASM2 is established in terms of the solid flux concept and a mass balance, Through giving some assumptions, a model including secondary clarifier is constructed on the A²/O process, and then simulates it.

2 ASM2 Kinetic Equations

ASM2 describe nitrogen removal and phosphorus removal processes in addition to COD removal, it incorporates 19 components and 19 process rate equations that require 22 stoichiometric coefficients and 42 parameters. The model has been introduced in a matrix form to accommodate the components considered with the complex stoichiometric and kinetic relations. The matrix gives all stoichiometric relations for the reactions (formation/decay) of all considered components considered in the process so that mass balance equations can be formulated to solve for the values of such components.

Every component (excluding inert components) in ASM2 participates in one process at least. So the rate of production of every component can be computed from the sum of the reaction rate of every process in which the component participates. For instance, component S_A participate in No. 5, 7, 8, 10 and 15 process, its production rate can be expressed:

$$\frac{dS_{O_2}}{dt} = \left(\frac{dS_{O_2}}{dt}\right)_4 + \left(\frac{dS_{O_2}}{dt}\right)_5 + \left(\frac{dS_{O_2}}{dt}\right)_{11} + \left(\frac{dS_{O_2}}{dt}\right)_{12} + \left(\frac{dS_{O_2}}{dt}\right)_{16} \quad (1)$$

According to the matrix of model:

$$\frac{dS_{O_2}}{dt} = \left(1 - \frac{1}{Y_H}\right) \cdot (\rho_4 + \rho_5 + \rho_{12}) - Y_{PHA} \cdot \rho_{11} - \frac{4.57 - Y_A}{Y_A} \cdot \rho_{16} \quad (2)$$

So the production rate of other components will be:

$$\begin{aligned}
\frac{dS_F}{dt} &= (1 - f_{S_I}) \cdot (\rho_1 + \rho_2 + \rho_3) - \frac{1}{Y_H} \cdot (\rho_4 + \rho_6) - \rho_8 & (3) \\
\frac{dS_A}{dt} &= -\frac{1}{Y_H} \cdot (\rho_5 + \rho_7) + \rho_8 - \rho_{10} + \rho_{15} \\
\frac{dS_{NH_4}}{dt} &= v_{1,NH_4} \cdot (\rho_1 + \rho_2 + \rho_3) - (i_{NBM} + \frac{1}{Y_A}) \cdot \rho_{16} + v_{17,NH_4} \cdot \rho_{17} \\
\frac{dS_{NO_3}}{dt} &= -\frac{1 - Y_H}{2.86Y_H} \cdot (\rho_6 + \rho_7) + \frac{1}{Y_A} \cdot \rho_{16} \\
\frac{dS_{PO_4}}{dt} &= v_{1,PO_4} \cdot (\rho_1 + \rho_2 + \rho_3) + Y_{PO_4} \cdot \rho_{10} - \rho_{11} - i_{PBM} \cdot (\rho_{12} + \rho_{16}) \\
&\quad + v_{13,PO_4} \cdot \rho_{13} + \rho_{14} + v_{17,PO_4} \cdot \rho_{17} \\
\frac{dS_I}{dt} &= f_{S_I} \cdot (\rho_1 + \rho_2 + \rho_3) \\
\frac{dS_{ALK}}{dt} &= v_{1,ALK} \cdot (\rho_1 + \rho_2 + \rho_3) + v_{18,ALK} \cdot \rho_{18} + v_{19,ALK} \cdot \rho_{19} \\
\frac{dS_{N_2}}{dt} &= \frac{1 - Y_H}{2.86Y_H} \cdot (\rho_6 + \rho_7) \\
\frac{dX_I}{dt} &= f_{X_I} \cdot (\rho_9 + \rho_{13} + \rho_{17})
\end{aligned}$$

$$\frac{dX_S}{dt} = -(\rho_1 + \rho_2 + \rho_3) + (1 - f_{X_1}) \cdot (\rho_9 + \rho_{13} + \rho_{17})$$

$$\frac{dX_H}{dt} = \rho_4 + \rho_5 + \rho_6 + \rho_7 - \rho_9$$

$$\frac{dX_{PAO}}{dt} = \rho_{12} - \rho_{13}$$

$$\frac{dX_{PP}}{dt} = -Y_{PO4} \rho_{10} + \rho_{11} - \rho_{14}$$

$$\frac{dX_{PHA}}{dt} = \rho_{10} - Y_{PHA} \cdot \rho_{11} - \frac{1}{Y_H} \cdot \rho_{12} - \rho_{15}$$

$$\frac{dX_{AUT}}{dt} = \rho_{16} - \rho_{17}$$

$$\frac{dX_{TSS}}{dt} = v_{1,TSS} \cdot (\rho_1 + \rho_2 + \rho_3) + 1.42 \cdot (\rho_{18} - \rho_{19})$$

$$\frac{dX_{MeOH}}{dt} = -3.45 \cdot (\rho_{18} - \rho_{19})$$

$$\frac{dX_{MeP}}{dt} = 4.87 \cdot (\rho_{18} - \rho_{19})$$

Where ρ_i is the rate equation of process i , v_{ij} is the stoichiometric coefficient for component j in process i .

3 Improved Model of Secondary Clarifier

The separation and concentration of the active biomass in an activated sludge process is performed in the secondary clarifier. The force that makes the sedimentation of the particles in the liquid possible, originates from gravity and the density differences between the particles and the liquid. From the bioreactor the mixed liquor enters the secondary clarifier where it should be sufficiently clarified in order to produce an effluent of acceptable quality. The sludge should also be adequately thickened so that the desired solids level in the bioreactors can be maintained through the sludge recirculation and enhance an effective treatment of the wasted activated sludge. This means that the settler combines the functions of clarification and thickening into one unit.

The mixed liquor is a flocculent suspension in which larger particles can be formed by the coalescing of particles, which have collided. The settling properties of a particular sludge depend both on the distribution of primary and floc particles and on how easily the primary particles are entrapped into larger flocs. Other factors that influence the settling behaviour are the hydraulic regime, temperature, basin design,

flow and feed variations, sludge characteristics, predators consuming dispersed bacteria, etc^[6].

Based on the mass balance, the rate of concentration diversification of solid equates the sum of solid flux change along the height and the reaction rate of solid. It can be expressed^[5]:

$$\frac{\partial X}{\partial t} = \frac{\partial \Phi}{\partial z} - \frac{\partial}{\partial z} \left(D \frac{\partial X}{\partial z} \right) + r_X \tag{4}$$

Where,

X: solid concentration

Φ: solid flux

D: dispersion coefficient

rX: reaction rate

Various simplifying assumptions, based on conditions known to exist in a continuous thickener, were applied to facilitate a solution, and initial and boundary conditions were established.

Assumptions:

- (1) Divide the secondary clarifier into n layers;
- (2) The continuous thickener does not exhibit vertical dispersion;
- (3) The concentration of suspended solids is completely uniform within any layer within the secondary clarifier;
- (4) The solids flux due to gravitational settling is zero at the bottom;
- (5) Component XTSS does not participate in reaction;
- (6) Soluble components do not consider gravitational flux;
- (7) The mass flux into a differential volume can not exceed the mass flux the volume is capable of pass;

As Figure 1, set Q_e is output water flow of secondary clarifier, is Q_w waste sludge flow, Q_r is recycle flow, input water flow $Q_{in}=Q_e+Q_w+Q_r$, S_{in} (X_{in}) is component concentration of input flow, divide the secondary clarifier into n layer, and inlet is at k layer, S_i (X_i), z_i , A_i is the component concentration, height and cross-sectional area respectively. Apply equation (2) in every layer, we can get:

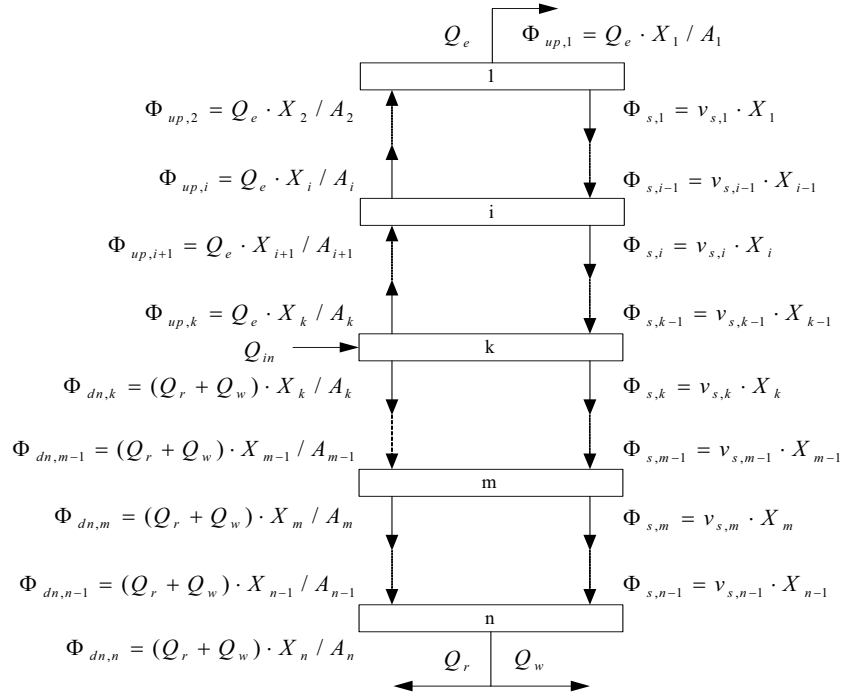


Fig. 1. Description of solids flux

(1) Top layer (No. 1 layer):

$$\begin{aligned}
\frac{dS_{O2,1}}{dt} &= (Q_e \cdot S_{O2,2} / A_2 - Q_e \cdot S_{O2,1} / A_1) / z_1 + r_{SO2,1} \\
\frac{dS_{F,1}}{dt} &= (Q_e \cdot S_{F,2} / A_2 - Q_e \cdot S_{F,1} / A_1) / z_1 + r_{SF,1} \\
\frac{dS_{A,1}}{dt} &= (Q_e \cdot S_{A,2} / A_2 - Q_e \cdot S_{A,1} / A_1) / z_1 + r_{SA,1} \\
\frac{dS_{NH4,1}}{dt} &= (Q_e \cdot S_{NH4,2} / A_2 - Q_e \cdot S_{NH4,1} / A_1) / z_1 + r_{SNH4,1} \\
\frac{dS_{NO3,1}}{dt} &= (Q_e \cdot S_{NO3,2} / A_2 - Q_e \cdot S_{NO3,1} / A_1) / z_1 + r_{SNO3,1} \\
\frac{dS_{PO4,1}}{dt} &= (Q_e \cdot S_{PO4,2} / A_2 - Q_e \cdot S_{PO4,1} / A_1) / z_1 + r_{SPO4,1} \\
\frac{dS_{I,1}}{dt} &= (Q_e \cdot S_{I,2} / A_2 - Q_e \cdot S_{I,1} / A_1) / z_1 + r_{SI,1} \\
\frac{dS_{N2,1}}{dt} &= (Q_e \cdot S_{N2,2} / A_2 - Q_e \cdot S_{N2,1} / A_1) / z_1 + r_{SN2,1} \\
\frac{dS_{ALK,1}}{dt} &= (Q_e \cdot S_{ALK,2} / A_2 - Q_e \cdot S_{ALK,1} / A_1) / z_1 + r_{SALK,1} \\
\frac{dX_{I,1}}{dt} &= (Q_e \cdot X_{I,2} / A_2 - Q_e \cdot X_{I,1} / A_1 - v_{s,1} \cdot X_{I,1}) / z_1 + r_{XI,1} \\
\frac{dX_{S,1}}{dt} &= (Q_e \cdot X_{S,2} / A_2 - Q_e \cdot X_{S,1} / A_1 - v_{s,1} \cdot X_{S,1}) / z_1 + r_{XS,1} \\
\frac{dX_{H,1}}{dt} &= (Q_e \cdot X_{H,2} / A_2 - Q_e \cdot X_{H,1} / A_1 - v_{s,1} \cdot X_{H,1}) / z_1 + r_{XH,1} \\
\frac{dX_{PAO,1}}{dt} &= (Q_e \cdot X_{PAO,2} / A_2 - Q_e \cdot X_{PAO,1} / A_1 - v_{s,1} \cdot X_{PAO,1}) / z_1 + r_{XPAO,1} \\
\frac{dX_{PP,1}}{dt} &= (Q_e \cdot X_{PP,2} / A_2 - Q_e \cdot X_{PP,1} / A_1 - v_{s,1} \cdot X_{PP,1}) / z_1 + r_{XPP,1} \\
\frac{dX_{PHA,1}}{dt} &= (Q_e \cdot X_{PHA,2} / A_2 - Q_e \cdot X_{PHA,1} / A_1 - v_{s,1} \cdot X_{PHA,1}) / z_1 + r_{XPHA,1} \\
\frac{dX_{AUT,1}}{dt} &= (Q_e \cdot X_{AUT,2} / A_2 - Q_e \cdot X_{AUT,1} / A_1 - v_{s,1} \cdot X_{AUT,1}) / z_1 + r_{XAUT,1} \\
\frac{dX_{MeOH,1}}{dt} &= (Q_e \cdot X_{MeOH,2} / A_2 - Q_e \cdot X_{MeOH,1} / A_1 - v_{s,1} \cdot X_{MeOH,1}) / z_1 + r_{MeOH,1} \\
\frac{dX_{MeP,1}}{dt} &= (Q_e \cdot X_{MeP,2} / A_2 - Q_e \cdot X_{MeP,1} / A_1 - v_{s,1} \cdot X_{MeP,1}) / z_1 + r_{MeP,1}
\end{aligned} \tag{5}$$

(2) Layer upward inlet (No. i layer):

$$\frac{dS_{c,i}}{dt} = (Q_e \cdot S_{c,i+1} / A_{i+1} - Q_e \cdot S_{c,i} / A_i) / z_i + r_{c,i} \quad (6)$$

$$\frac{dX_{c,i}}{dt} = (Q_e \cdot X_{c,i+1} / A_{i+1} - Q_e \cdot X_{c,i} / A_i + v_{s,i-1} \cdot X_{c,i-1} - v_{s,i} \cdot X_{c,i}) / z_i + r_{c,i}$$

(3) Layer of inlet (No. k layer):

$$\frac{dS_{c,k}}{dt} = (Q_{in} \cdot S_{c,in} - Q_e \cdot S_{c,k} - (Q_r + Q_w) \cdot S_{c,k}) / (A_k \cdot z_k) + r_{c,k} \quad (7)$$

$$\begin{aligned} \frac{dX_{c,k}}{dt} = & (Q_{in} \cdot X_{c,in} - Q_e \cdot X_{c,k} - (Q_r + Q_w) \cdot X_{c,k}) / (z_k \cdot A_k) \\ & + (v_{s,k-1} \cdot X_{c,k-1} - v_{s,k} \cdot X_{c,k}) / z_k + r_{c,k} \end{aligned}$$

(4) Layer downward inlet (No. m layer):

$$\frac{dS_{c,m}}{dt} = ((Q_r + Q_w) \cdot S_{c,m-1} / A_{m-1} - (Q_r + Q_w) \cdot S_{c,m} / A_m) / z_m + r_{c,m} \quad (8)$$

$$\begin{aligned} \frac{dX_{c,m}}{dt} = & ((Q_r + Q_w) \cdot X_{c,m-1} / A_{m-1} - (Q_r + Q_w) \cdot X_{c,m} / A_m) / z_m \\ & + (v_{s,m-1} \cdot X_{c,m-1} - v_{s,m} \cdot X_{c,m}) / z_m + r_{c,m} \end{aligned}$$

(5) Bottom layer (No. n layer):

$$\frac{dS_{c,n}}{dt} = ((Q_r + Q_w) \cdot S_{c,n-1} / A_{n-1} - (Q_r + Q_w) \cdot S_{c,n} / A_n) / z_n + r_{c,n} \quad (9)$$

$$\begin{aligned} \frac{dX_{c,n}}{dt} = & ((Q_r + Q_w) \cdot X_{c,n-1} / A_{n-1} - (Q_r + Q_w) \cdot X_{c,n} / A_n) / z_n \\ & + v_{s,n-1} \cdot X_{c,n-1} / z_n + r_{c,n} \end{aligned}$$

Equations (5)-(9) construct the improved model of secondary clarifier based on ASM2 Where $S_{c,n}$, $X_{c,n}$ is soluble and particulate component concentration in layer n respectively.

$v_{s,n}$ is settling velocity, expressed by by^{[9][10]}:

$$v_{s,n} = 10.59 \cdot \frac{e^{-(0.839+0.00131SVI)X_n}}{X_n} \quad (10)$$

Where SVI is sludge volume index^{[9][10]}.

4 A2/O Process Model

Now we got the ASM2 kinetic equations and improved model of secondary clarifier based on ASM2. But the models just describe the reactions of every single reactor, it is necessary to establish the model of specific activated sludge process.

The modeling principle of Mass Balance System is “accumulation=input-output+reaction”^{[7][8]}. In wastewater treatment process, the equation is:

$$V\left(\frac{dC_j}{dt}\right) = q_i C_{i,j} - q_o C_{o,j} \pm \sum_n V r_{n,j} \tag{11}$$

Where, V is volume of reactor, dC_j/dt is the rate of concentration diversification of component j in reactor V , q_i and q_o are input flow and output flow respectively, $C_{i,j}$ and $C_{o,j}$ are concentration of component j in input and output flow respectively, r_{nj} is the reaction rate of component j in reaction n .

Activated sludge A²/O process is shown as Figure 2.

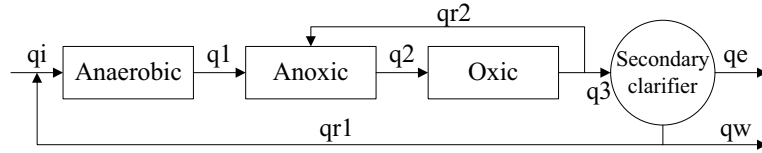


Fig. 2. the activated sludge process

According to equation (9), the mass balance equation in anaerobic is:

$$V_1\left(\frac{dC_j}{dt}\right) = q_i C_{i,j} - q_o C_{o,j} \pm V_1 \sum \left(\frac{dC_j}{dt}\right)_{V_1} \tag{12}$$

Where $\sum \left(\frac{dC_j}{dt}\right)_{V_1}$ is kinetic equation of component j in anaerobic, see equation (3) So the mass balance equations in anaerobic in anoxic and oxidic are:

$$V_2\left(\frac{dC_j}{dt}\right) = q_i C_{i,j} - q_o C_{o,j} \pm V_2 \sum \left(\frac{dC_j}{dt}\right)_{V_2} \tag{13}$$

$$V_3\left(\frac{dC_j}{dt}\right) = q_i C_{i,j} - q_o C_{o,j} \pm V_3 \sum \left(\frac{dC_j}{dt}\right)_{V_3} \tag{14}$$

Integrate equations (11), (12), (13) and (5)~(9), the A²/O process model including improved model of secondary clarifier is established.

5 Simulation

For simplifying the model, some assumptions is given:

- (1) Dissolved oxygen does not be considered;
- (2) Alkalinity does not be considered;
- (3) Chemical precipitation of phosphates does not be considered;
- (4) Component S_I does not participate in reaction;
- (5) Component X_{TSS} does not participate in reaction;
- (6) Secondary clarifier is divided into 10 layers, and the inlet is in No. 4 layer.

Refer to the literature^[11], the concentration of components of input flow and parameters of reactors are given:

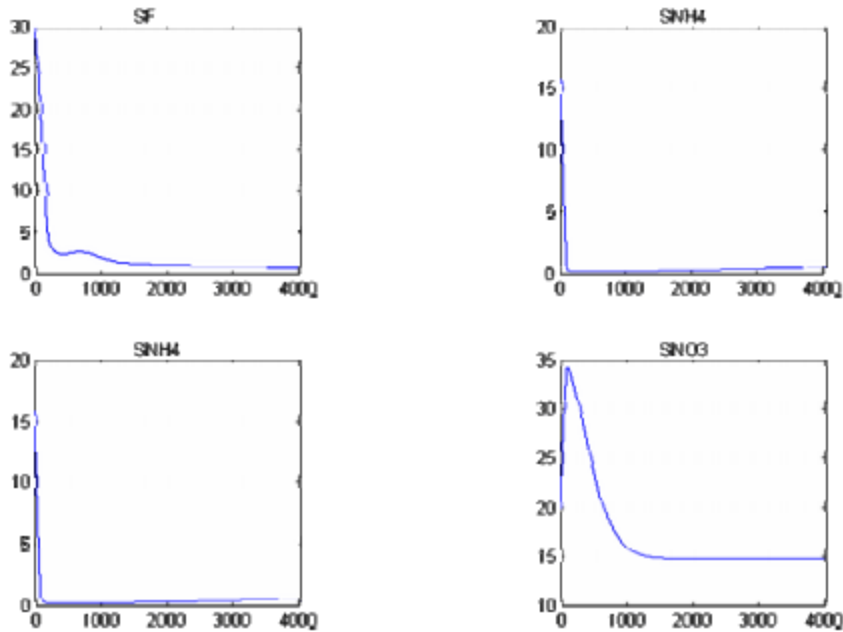
Volume of anaerobic=3, Volume of secondary clarifier=3, Volume of anoxic=3, Sludge Volume Index=800, Volume of oxic=3, Wastewater flow=10, Depth of secondary clarifier=5, Recycle mixed flow=0.2, Cross-sectional area of secondary clarifier=0.6, Recycle sludge flow=0.07, Waste sludge flow=0.4.

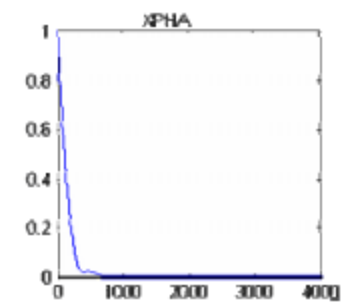
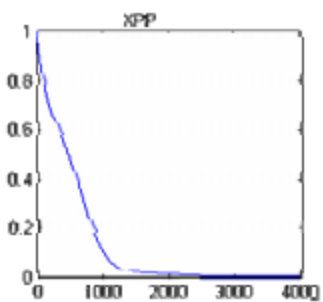
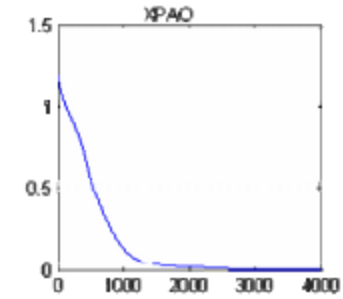
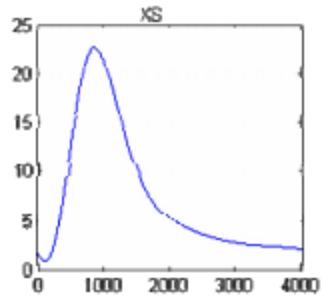
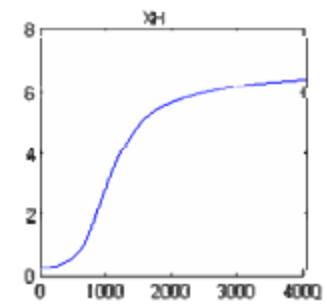
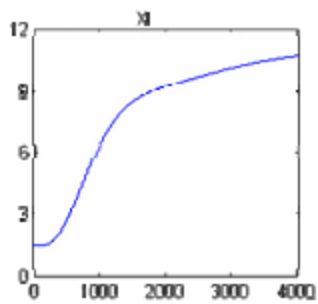
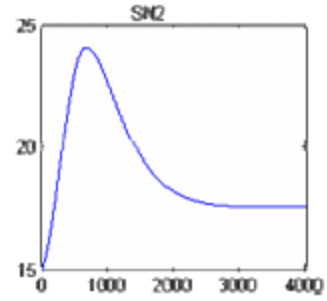
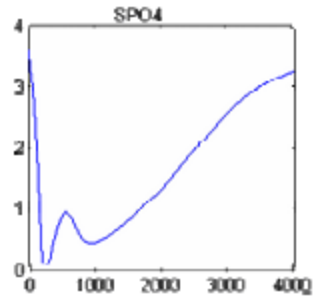
Table 1. the initial value of components

S_F	S_A	S_{N2}	S_{NH4}	S_{NO3}	S_{PO4}	X_I
30	20	15	16	0	3.6	25
X_S	X_H	X_{AUT}	X_{PAO}	X_{PP}	X_{PHA}	
125	30	0	0	0	0	

The stoichiometric coefficients and parameters refer to literature^[3].

The curves of component concentration of output flow are shown below:





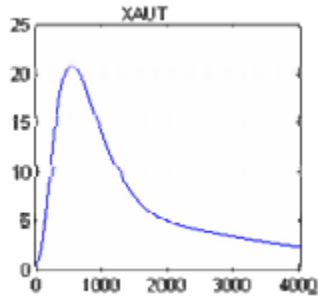


Fig. 3. curves of component concentration of output

Table 2. the stable values of components

SF	SA	S_I	S_{N2}	S_{NH4}	S_{NO3}	S_{PO4}
0.79	0.35	30	17.62	0.52	14.73	3.27
X_I	X_S	X_H	X_{AUT}	X_{PAO}	X_{PP}	X_{PHA}
10.76	2.32	6.43	2.53	0	0	0

From the simulation we can make out:

- (1) Most COD in wastewater can be removed efficiently, and the concentration of suspended solids is in reasonable range. These illuminate that the model of secondary clarifier is established in effect;
- (2) The effect of nitrogen removal is satisfied. It can be seen that the total nitrogen mainly compose of nitrate nitrogen, so the removal rate of total nitrogen can be enhanced through increasing the recycle mixed flow.
- (3) The phosphates is removed inefficiently because the chemical precipitation of phosphates is not be considered;

6 Conclusion

This paper puts forward one improved model of secondary clarifier based on ASM2. The improved model describes the biology reactions in secondary clarifier as well as the dynamical solid/liquid separation. Through summarizing the modeling methods, the mechanism model of A2/O process is established, and is simulated. Because of some assumptions, the simulating result must have error in comparison with the real result.

Acknowledgments. This work is supported by a grant from the National Natural Science Foundation of China (No. 60474064) and the Fund of produce-study-research Science and Technology Cooperation of Hangzhou City to Zhou LiFang and Ma LongHua.

References

1. Henze, M., et al, Activated Sludge Model No.1, IAWPRC Scientific and Technical report. No.1, London: IAWPRC,(1987)
2. Gujer, W., et al, Activated Sludge Model No.3, Water Sci&Tec (1999) 39(1): 183-193.
3. Henze, M. et al., Activated Sludge Models ASM1, ASM2, ASM2D and ASM3, scientific and technical report No .9 (2000)
4. R. Dupont, M. Henze, W. Gujer, Modeling of the secondary clarifier combined with the activated sludge model No.1, Water Sci&Tec (1991) 25(6)
5. Yoshioka, N., et al., Continuous Thickening of Homogeneous Flocculated Slurries, Chem. Eng. (Tokyo) (1957) 21, 66
6. Ulf. Jeppsson, Stefan, Diehl, Evaluation of a dynamic model of the secondary clarifier. Water Sci&Tec (1996) 34(5-6): 19-26
7. Williamson, K.J. and McCarty, P.L., Rapid measurement of Monod half-velocity coefficients for bacteria kinetics, Biotechnol Bioeng (1975) 17, 915-924
8. Fesabeth V. Munch et al, Simultaneous nitrification and denitrification in bench-scale sequence batch reactor, Wat. Res (1996) Vol. 30
9. Cho S.H., Colin F., Sardin M. and Prost C., Settling Velocity Model of Activated Sludge. Wat. Res (1993) Vol.27, 1237-1241
10. Huang, Dynamic mathematic model of secondary clarifier, Journal of Suzhou Institute of Urban Construction and Environmental Protection (1992) Vol.11,No.3,6-11
11. Wu, J.Q., Wang, H.Z., Introduction of Activated Sludge Model No.2 (ASM2), China Water and Wastewater (1998) 24(6)
- 12.. Lu, P.L., et al., Development of Activated Sludge Kinetic Model Studying, Journal of Chongqing University (2005) Vol.25,No.3

Modeling and Simulation for Transpiration Cooling Control System with Distributed Parameters Based on Scilab

Cong-Chao Zhu, Shi-Yin Qin

School of Automation Science and Electrical Engineering,
Beihang University, Beijing 100083, China
zhcch@asee.buaa.edu.cn, qsy@buaa.edu.cn

Abstract. Transpiration cooling control is an important and efficient means of thermal protection, which can be employed to avoid the deformation of a high-speed vehicle resulting from ablation of the nose and the change of the performance of aerodynamics. In this paper, a set of models with distributed parameters for transpiration control process are built and some strategies and algorithms of numerical solutions are studied in depth. Then a series of numerical simulations including fixed domain (non-ablation) and variable domain (ablation) cases are completed and the results are analyzed in detail. The numerical simulation results indicate that both temperature field and boundary moving law can be controlled by adjusting the transpiration control parameters. Furthermore, a toolbox of the transpiration cooling control based on Scilab is designed and developed, which enhances and enriches capabilities of Scilab tool boxes and can be extended to be used in the modeling, analysis, synthesis and design of complex systems with distributed parameters.

Keywords: Ablation, Transpiration cooling, Distributed parameter control, Crank-Nicholson, Scilab

1 Introduction

When a high speed space vehicle flies through the air, the high temperature due to the friction between the vehicle and the air may cause ablation of the material in its front

surface which could result in the damage of the vehicle. In engineering, a thermal shield must be designed to prevent this event from happening. Therefore how to resolve the problem of heat barrier is of great significance for the research of aircraft [1], [2].

Ablation thermal protection is the primary means of thermal protection. It is not new and has attracted considerable attentions both in engineering and in mathematics since the 19th century [3], [4]. Basically, this process can be described by a heat conduction equation with moving boundary condition. Through the process of ablation, some substance, by means of melting or vaporizing, will be injected into the gas boundary lay around the ablation surface which is called mass injection. It can be classified to three kinds: (a) transpiration cooling; (b) film cooling including liquid film and gaseous film; (c) ablation, for instance, sublimation, fusion, denudation and so on. All the above methods except film cooling have the same principle that the substance is injected into the gas boundary lay with a velocity vertical to the ablation surface resulting in influencing the performance of aerodynamics. At the same time, the temperature gradient is reduced and heating the ablation material is blocked partly or completely.

Transpiration cooling control is an effective method to thermal protection, which can avoid the deformation of a high-speed vehicle resulting from ablation of the nose and the change of the performance of aerodynamics. Applying the principle of transpiration to temperature field control has prospects for both military and commercial application [5] such as in the thermal protection of the cabin of supersonic aircraft, in dealing with the ablation problem of heat protection shield of missile nose, in the design of thermal protection for the throat of rocket jets or the gas rudder, in the control of air suspension train [6], etc. Hence it is of practical importance.

Based on the mathematical models for transpiration cooling control system, this paper gives some numerical solutions to them, including line method and Crank-Nicholson method. By using the open source scientific computing software SCILAB, a series of numerical simulations are implemented and the corresponding simulation results are analyzed in depth.

The rest of this paper is organized as follows. In section 2, a series of mathematical models are built and the respective engineering significance is expounded; and then some resultful strategies and algorithms of numerical solutions are studied in detail in section 3; successively section 4 introduces the open source software platform Scilab

based on which a series of numerical simulation results are completed and some corresponding analysis and comments are made. Finally section 5 concludes the whole paper and gives some remarks.

2 Modeling and Analysis

2.1 Problem statement

As the structure temperature depends on variables of both space and time, the transpiration cooling control system is a typical distributed parameter system. Thus the initial conditions and the boundary conditions play an important role in the procedures of analysis and solution. Therefore, as a powerful tool of modeling, the partial differential equation (PDE) with variable coefficients should be employed in the analysis, synthesis, design and simulation.

It is impossible to find some feasible analytical solutions based on such a class of models of PDE with variable coefficients. Thus it is necessary to find some effective numerical methods so as to deal with difficulties in the resolving and simulations.

2.2 An open loop control model with variable domain heat transfer

Variable domain transpiration cooling control model can be obtained in a two-dimension coordinate system (y, t) which is composed of one-dimension space coordinate y and one-dimension time coordinate t . And the coordinate origin O is set at the initial outer boundary of the porous thermal shield. Let l denote the thickness of the porous material, y -axis be vertical to the initial outer boundary, and a part of length $s(t)$ is bite off on the y -axis because of ablation. The coolant flows through the porous thermal shield and injects into the gas boundary lay, which may be shown in Fig. 1.

The basic assumptions in this paper are as follows. (1) The coolant flowing through the porous thermal shield is incompressible, and there is no chemical reaction and no mass change also. (2) The thermal shield and the coolant have the same local temperature because of full heat exchange. (3) The porous medium is homogeneous and isotropic. (4) All material parameters are constant.

Suppose $T(y, t)$ is the temperature distribution of porous material at any position y and at any time t , based on dynamic infinitesimal method [7], the variable domain transpiration cooling control equation can be established as follows:

$$\frac{\partial T}{\partial t} = \alpha \frac{\partial^2 T}{\partial y^2} + \beta \frac{\partial T}{\partial y} + \frac{\dot{m}_L}{l - s(t)} \frac{l - y}{l - s(t)} \frac{\partial T}{\partial y}, \quad y \in (s(t), l) \tag{1}$$

where $\alpha = K / \rho c_p$, $\beta = \dot{m}_L c_{pL} / \rho c_p$. K , ρ and c_p represent the thermal conductivity, the density and the specific heat of porous material respectively. \dot{m}_L is the transpiration mass flux per second which is constant or time variant as adjustable control parameter. c_{pL} is the specific heat of the coolant.

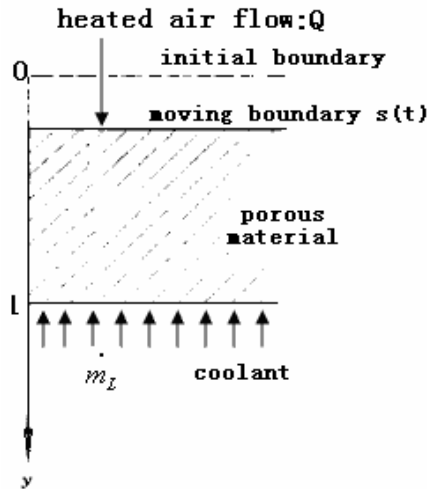


Fig. 1. Schematic representation of the ablation-transpiration cooling process

The first term on the right side of the equation is the diffusion factor of temperature in porous material, the second is the convection factor as a result of coolant flow, and the third is the compensation factor for the temperature difference due to the moving

boundary. The convection factor coefficient β is linearly related with the transpiration mass flux m_L^g which is actually a control variable because both temperature field T and boundary moving law $s(t)$ can be controlled by it. In other words, by regulating the control variable m_L^g in the equation (1), not only can the temperature field be controlled, but the variable domain can also be regulated.

The initial condition can be expressed as

$$t = 0, T = \varphi(y), \quad y \in (0, l) \quad (2)$$

where $\varphi(y)$ is a known function of y .

On the outer boundary, the surface of porous material is heated by the friction between the space vehicle and the air. At the same time, the heating is blocked partly or completely due to the existence of coolant. Assume L as the latent heat of ablation material, according to energy balance, outer boundary condition can be expressed as

$$\rho L \dot{s}(t) - \left(K \frac{\partial T}{\partial y} \right) \Big|_{y=s(t)} = \psi(m_L^g) q(t) \quad (3)$$

where $q(t)$ is practical heat flow, $\psi(m_L^g)$ is blocking coefficient, which is a quadric function of m_L^g as

$$\psi(m_L^g) = 1 - \left[0.724 \frac{h_e}{q_0} m_L^g - 0.13 \left(\frac{h_e}{q_0} m_L^g \right)^2 \right] \quad (4)$$

where h_e is the inner boundary enthalpy, q_0 is the theoretical heat flow.

The blocking coefficient $\psi(m_L^g)$ represents physical meaning that heating is blocked due to transpiration which is in the range of zero to one. If $m_L^g = 0$, then $\psi = 1$ which indicates heating is not blocked at all and all of the heat flows into the

porous material. On the other hand, if $\frac{h_e m_L^g}{q_0}$ is 3.033 or 2.536, then $\psi = 0$ which denotes heating is blocked completely and none of the heat flows into the porous material. If $\frac{h_e m_L^g}{q_0}$ is between 2.536 and 3.033, then $\psi < 0$ can be obtained easily which means hotter gas is heated by the cooler porous material, however it is impossible in practice. Therefore, $\frac{h_e m_L^g}{q_0}$ has its practical meaning only when its value is between 0 and 2.536 and equation (4) can only be used under small transpiration quantity. When large transpiration quantity need be researched in theory, $\psi = 1$ could be employed considering a conservative estimate in engineering. In fact, all the simulation results obtained in section 4 are subjected to this condition.

On the inner boundary, considering coolant's decalescence from controller icebox to the inner boundary of porous material, inner boundary condition can be expressed as

$$-(K \frac{\partial T}{\partial y})|_{y=l} = m_L^g (\Delta h_N - \Delta h_c) \tag{5}$$

where $\Delta h_c = c_{pL} T_c$ represents coolant enthalpy in icebox and T_c is the temperature of coolant; $\Delta h_N = c_p T_N$ represents inner boundary enthalpy in porous material and T_N is the temperature of inner boundary. This condition indicates the difference between the state of icebox and the state of inner boundary. If the difference exists, heat transfer between them occurs and the quantity of heat exchange is in direct proportion to the control variable m_L .

Combining the control equation (1) with the boundary condition (3) and (5), the variable domain transpiration open loop control model with the second boundary condition can be expressed as

$$\frac{\partial T}{\partial t} = \alpha \frac{\partial^2 T}{\partial y^2} + \beta \frac{\partial T}{\partial y} + s(t) \frac{l-y}{l-s(t)} \frac{\partial T}{\partial y}, \quad y \in (s(t), l) \tag{6a}$$

$$\rho L \dot{s}(t) - (K \frac{\partial T}{\partial y})|_{y=s(t)} = \psi(m_L^g)q(t) \quad (6b)$$

$$-(K \frac{\partial T}{\partial y})|_{y=l} = m_L^g(\Delta h_N - \Delta h_c) \quad (6c)$$

It is obvious that the basic mathematical model for variable domain transpiration control is given in (6). As a matter of fact, the control law governed by variable m_L^g can be set to a form of step function or a known time variant function, independent of the control process, thus in this case, the control strategy for the model (6) belongs to an open loop one.

The variable domain transpiration open loop control model with the third boundary condition can be expressed as [7]

$$\frac{\partial T}{\partial t} = \alpha \frac{\partial^2 T}{\partial y^2} + \beta \frac{\partial T}{\partial y} + s(t) \frac{l-y}{l-s(t)} \frac{\partial T}{\partial y}, \quad y \in (s(t), l) \quad (7a)$$

$$(\frac{\partial T}{\partial y} - aT)|_{y=s(t)} = c \quad (7b)$$

$$(\frac{\partial T}{\partial y} + bT)|_{y=l} = dT_c \quad (7c)$$

where $a = c_p \cdot \psi \cdot q_0 / (K \cdot h_r)$, $b = m_L^g \cdot c_p / K$, $c = -\psi \cdot q_0 / K$, $d = m_L^g \cdot c_{pL} / K$.

2.3 An open loop control model with fixed domain heat transfer

In fact, it is possible and feasible to make the boundary without ablation, that means $s(t)=0$, by increasing transpiration quantity. In this case, the fixed boundary control is actually equivalent to the temperature field control.

In equations (6), let $s(t)=0$, it is easy to get the fixed domain transpiration open loop control model with the second boundary condition as follows:

$$\frac{\partial T}{\partial t} = \alpha \frac{\partial^2 T}{\partial y^2} + \beta \frac{\partial T}{\partial y}, \quad y \in (0, l) \tag{8a}$$

$$-(K \frac{\partial T}{\partial y})|_{y=0} = \psi(m_L^g)q(t) \tag{8b}$$

$$-(K \frac{\partial T}{\partial y})|_{y=l} = m_L^g(\Delta h_N - \Delta h_c) \tag{8c}$$

With the same method, the fixed domain transpiration open loop control model with the third boundary condition can be educed as follows

$$\frac{\partial T}{\partial t} = \alpha \frac{\partial^2 T}{\partial v^2} + \beta \frac{\partial T}{\partial v}, \quad y \in (0, l) \tag{9a}$$

$$(\frac{\partial T}{\partial y} - aT)|_{y=0} = c \tag{9b}$$

$$(\frac{\partial T}{\partial v} + bT)|_{y=l} = dT_c \tag{9c}$$

2.4 A closed loop control model

An architecture of closed loop control system can be shown in Fig. 2.

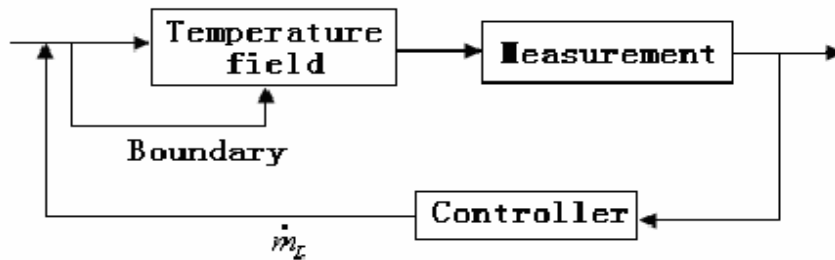


Fig. 2 . An architecture of closed loop control system

Usually, the transpiration quantity function m_L^g can be chosen in the form of temperature series [7] which may be expressed as

$$m_L^g = m_{L0}^g + m_{Lu}^g \sum_{i=1}^n \frac{U_{k,i} - T_t}{T_t} \quad (10)$$

where m_{L0}^g is the initial transpiration quantity, m_{Lu}^g is the transpiration quantity per unit, $T_{k,i}$ is the real temperature at the position k in porous material, and $U_{k,i}$ is the sampling temperature which is usually a function of $T_{k,i}$ based on its neighborhood temperature. However, $U_{k,i}$ is assumed to be the same as $T_{k,i}$ in simulation experiments, T_t is the target temperature.

3. Numerical Solutions: Strategies and Algorithms

The transpiration cooling control systems governed by equations (6) (7) (8) (9) are a class of typical distributed parameter control system with variable-domain boundary conditions. From the point of view of mathematics, their models are uniformly represented by a set of second-order variant coefficient quasilinear parabolic partial differential equations. In view of the complexity of their inner mechanism and boundary conditions, some effective strategies and algorithms of numerical solutions are employed instead of pursuing some miscellaneous analytical solutions.

3.1 A linear method for open loop control problem based on the third boundary condition

3.1.1 Temperature field control for fixed domain

For the fixed domain transpiration open loop control model (9), using linear method [8] [9], a discrete control equation can be deduced as follows

$$\frac{dT_i}{dt} = \left(\frac{\alpha}{h^2} - \frac{\beta}{2h}\right)T_{i-1} - \frac{2\alpha}{h^2}T_i + \left(\frac{\alpha}{h^2} + \frac{\beta}{2h}\right)T_{i+1} \tag{11}$$

where $h (=l/n)$ is the space step length of space.

Let $y_{-1} = -h$, $y_{n+1} = l + h$ substituting $\frac{(T_1 - T_{-1})}{2h}$ for $\frac{\partial T(0,t)}{\partial y}$ in the boundary condition (9b) and solving it, T_{-1} can be found. Recalling the equations (11) for $i=0$, a substitution of T_{-1} can be made to eliminate it from the equation. Thus an ordinary differential equation can be obtained

$$\frac{dT_0}{dt} = -\left(\frac{2\alpha}{h^2} + \frac{2\alpha}{h}a - \beta a\right)T_0 + \frac{2\alpha}{h^2}T_1 - c\left(\frac{2\alpha}{h} - \beta\right) \tag{12}$$

In the same way, the boundary condition (9c) can be transformed as

$$\frac{dT_n}{dt} = \frac{2\alpha}{h^2}T_{n-1} - \left(\frac{2\alpha}{h^2} + \frac{2\alpha}{h}b + \beta b\right)T_n + T_c d\left(\frac{2\alpha}{h} + \beta\right) \tag{13}$$

Combining the equations (11) for $i=1, 2, \dots, n-1$ with equation (12) (13), we have a set of ordinary differential equation group including $n+1$ equations with $n+1$ unknown terms. And the initial condition can be acquired from equation (2).

Using implicit Euler method, the above equation set will become

$$BT^{j+1} = T^j + f, j = 0, 1, 2, \dots \tag{14}$$

where τ is the step length of time, $r = \tau\alpha / h^2$, $H = \tau\beta / 2h$

$$B = \begin{bmatrix} 1+2r+2ha(r-H) & -2r & & & \\ -(r-H) & 1+2r & -(r+H) & & \\ & \dots & \dots & \dots & \\ & & -(r-H) & 1+2r & -(r+H) \\ & & & -2r & 1+2r+2hb(r+H) \end{bmatrix} \tag{15}$$

$$f = (-2hc(r-H), 0, \dots, 0, 2hdT_c(r+H))^T \quad (16)$$

$$T^j = (T_0^j, T_1^j, \dots, T_n^j)^T \quad (17)$$

Therefore the numerical solutions of above discrete linear equations can be found by effective double sweeping method.

3.1.2 Boundary control for variable domain

For the open loop control model of variable domain transpiration (7), the control process can be divided into two stages:

(a) From the beginning to time t_0 , where t_0 is the time when the temperature of the outer boundary T_0 reaches the ablation temperature T_m of the porous material. Here, $s(0)=0$, $T_0 < T_m$, the algorithm is the same as that of the fixed domain.

(b) After the time t_0 , $T_0(t) \equiv T_m$. The space step length h will be changed to $h(t) = (l-s(t))/n$ and β will be substituted by $\beta_1(t) = \beta + \frac{g}{s(t)}(1 - \frac{i}{n})$, $i = 0, 1, \dots, n$. Thus the equation (7a) can be reduced to

$$\frac{\partial T}{\partial t} = \alpha \frac{\partial^2 T}{\partial y^2} + \beta_1 \frac{\partial T}{\partial y} \quad (18)$$

As $T_0(t) \equiv T_m$ is known, the discrete equation concerning $T_0(t)$ can be eliminated. And the differential equation (19) with the constraint of initial condition (20) can be employed to find the recursive results of moving boundary $s(t)$.

$$\frac{ds}{dt} = \frac{1}{\rho L} [(K \frac{\partial T}{\partial y})|_{y=s(t)} + \psi(m_L^g)q(t)] \quad (19)$$

$$s(0) = 0 \quad (20)$$

where $\frac{\partial T}{\partial y} \Big|_{y=s(t)}$ can be computed by numerical differential formula

$$\frac{\partial T}{\partial y} \Big|_{y=s(t)} = \frac{1}{h} \left[-\frac{11}{6}T_0 + 3T_1 - \frac{3}{2}T_2 + \frac{1}{3}T_3 \right] \quad (21)$$

with truncation error $O(h^2)$.

3.2 Crank-Nicholson method for open loop control based on the second boundary condition

In order to accelerate the convergence speed of the numerical simulation, Crank-Nicholson method is employed to discretize the open loop control model with the second boundary condition along space and time. In this method, two-layer six-point symmetrical difference scheme is adopted for the second-order diffusion factor, and two-layer four-point central difference scheme is used for the first-order convection factor and the boundary conditions. As a result, a linear algebraic equation set with sparse coefficient matrix is obtained and the double sweeping method can be employed to solve it.

3.3 The solution of closed loop control problems

Without loss of generality, the simulation process for closed loop control problem for temperature field control is expressed simply as follows: initialize the temperature and the initial transpiration quantity m_{L0}^g and calculate the transpiration control parameter m_L^g using formula (10); compute the coefficients in the equation set (14) with m_L^g ; solve it and get the temperature next time; sample the temperature with sampling equation; take the sampling temperature into formula (10) and get m_L^g ; do the above steps in turn and constitute close loop control.

An algorithm of the above process may be shown in Fig.3.

```

Initialize the temperature  $T$  and the initial transpiration quantity  $\dot{m}_{20}$ ,
set the target temperature  $T_t$ , the step length of time  $\tau$ , the step
length of space  $h$  and the simulation time  $t_{end}$ ,  $j=0, t=0$ ;

Compute the coefficients  $B$  and  $f$  using formulas (15) and (16);

LOOP:
    Solve the algebraic equation (14) and save  $T^j$ ;
     $j=j+1, t=t+\tau$ ;
    Calculate  $\dot{m}_z$  with formula (10);
    Compute the coefficients  $B$  and  $f$  using formulas (15) and (16);
IF  $t \geq t_{end}$ 
    Break and finish, the temperature of control point is stored in
     $T^j, j=0, 1, 2, \dots$ 
ELSE
    Goto LOOP;

```

Fig. 3. An algorithm of closed loop control process

4. Simulation Results

4.1 Scilab: an open source software platform and its advantages

Scilab is a scientific software package for numerical computations providing a powerful open computing environment for engineering and scientific applications [10], [11]. Developed since 1990 by researchers from INRIA and ENPC, it is now maintained and developed by Scilab Consortium since its creation in May 2003.

Distributed freely and open source via the Internet since 1994, Scilab is currently being used in educational and industrial environments around the world and is the only open-source alternative to commercial products for scientific computing such as MATLAB.

Scilab includes hundreds of mathematical functions with the possibility to add interactively programs from various languages (C, Fortran...). It has sophisticated data structures (including lists, polynomials, rational functions, linear systems...), an interpreter and a high level programming language.

Scilab has been designed to be an open system where the user can define new data types and operations on these data types by using overloading.

Scilab provides a variety of powerful and available toolboxes: 2-D and 3-D graphics, animation; Linear algebra, sparse matrices; Simulation: ODE solver and DAE solver; Scicos: a hybrid dynamic systems modeler and simulator; Interface with Computer Algebra: Maple package for Scilab code generation, MuPAD 3.0 includes Scilab; Interface with Tcl/Tk, and so on. These toolboxes make the software powerful and convenient to use.

Moreover, it has many advantages in its openness and adaptability and runs under different types of operating systems in a user-friendly environment. It works on most Unix systems (including GNU/Linux) and Windows (9X/NT/2000/XP). It comes with source code, on-line help and English user manuals. Binary versions are available.

Considering the above advantages, especially its two dominant functions, numerical computing and visualizing computed results, it is an ideal simulation platform for solving the transpiration control models and visualizing the simulation results.

4.2 Main simulation results with analysis and comments

Take a combinational parameter TPC ($= m_L^g c_{pL} l / 2K$) as transpiration control parameter, the numerical simulation is carried out in the Scilab software environment under the simulation condition:

$$l = 0.015m, c_{pL} = 1.25kcal / kg^{\circ}, T_0 = 288K, T_c = 42.23K$$

And the simulation results are as follows.

Fig.4 shows the boundary temperature varying with time and TPC. The x-axis of the figure represents the simulation time, and the y-axis represents the corresponding temperature. As we can see from the figure, the boundary temperature increases when

the time increases for any value of transpiration control parameter TPC. Comparing the several curves with different values of TPC, it is easy to know that when TPC increases, the acceleration of the boundary temperature change is decreasing. If TPC is greater than or equal to five, the temperature will become stable soon and this is the target of temperature control.

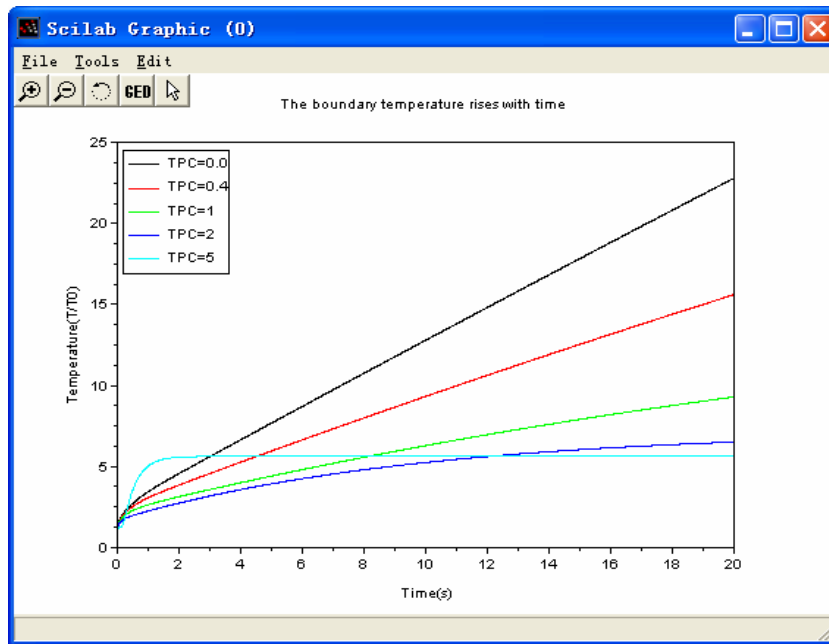


Fig. 4. Boundary temperature varying with time and TPC

Fig.5 shows temperatures of several positions varying with time when TPC=1 without ablation. The x-axis of the figure represents the simulation time, and the y-axis represents the corresponding temperature. We can see the trend of temperature change: at any time, the temperature is declining from the outer boundary to inner boundary of the porous material.

Fig.6 gives the temperature distribution varies with TPC at $t=20s$, the x-axis of which represents the relative position, and the y-axis refers to the corresponding temperature. From the four curves, the transpiration control parameter TPC plays an important role in deciding the shapes of the curves. The temperature is lower when the value of TPC is larger. When the value of TPC increases, the temperature is declining immediately and approaching the initial temperature.

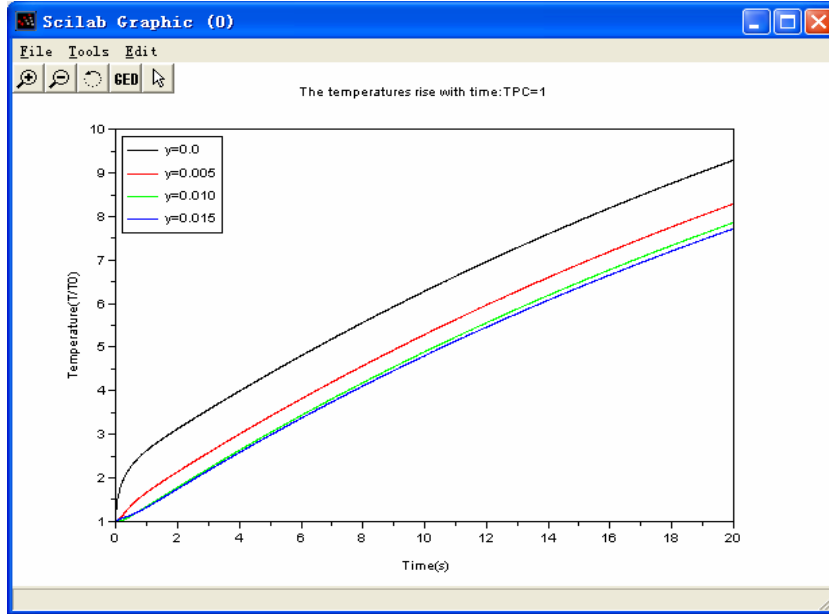


Fig. 5. Temperatures of several positions varying with time (non-ablation) when TPC=1

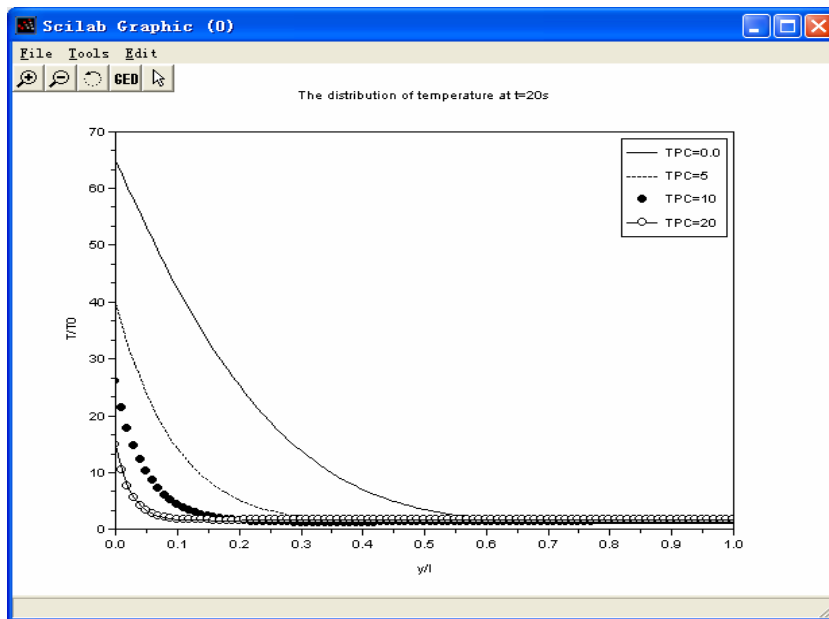


Fig. 6. Temperature distribution varying with TPC at t=20s

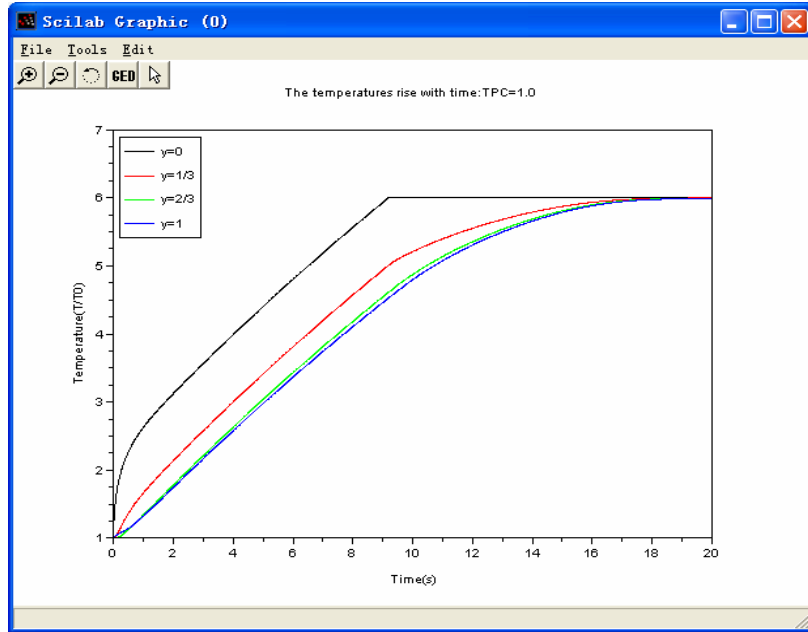


Fig. 7. Temperatures of several positions varying with time (ablation) when TPC=1

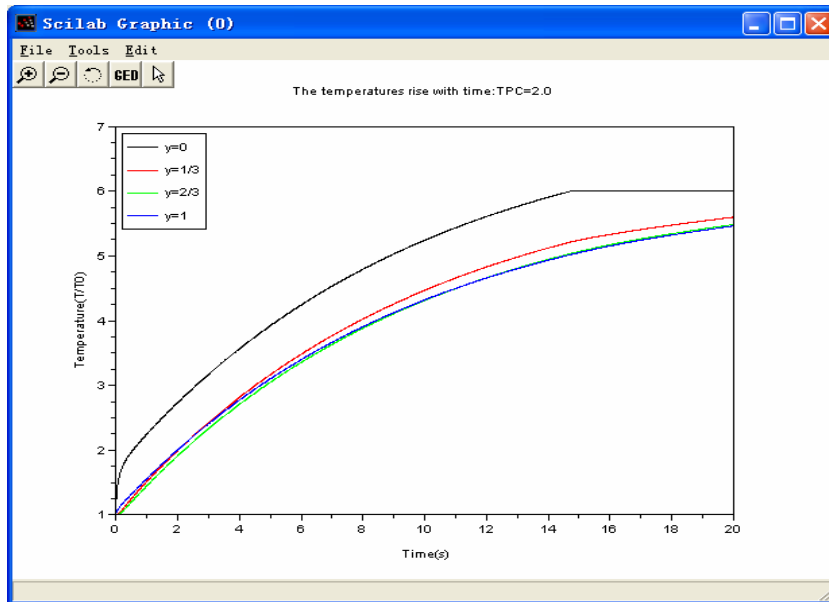


Fig. 8. Temperatures of several positions varying with time (ablation) when TPC=2

Temperatures of several positions varying with time with ablation temperature six times the initial temperature when $TPC=1$ and $TPC=2$ are shown in Fig.7 and Fig.8 respectively. When $TPC=1.0$, the outer boundary temperature reaches the ablation temperature at the time of about 9s, and the temperature will be maintained after that. Also, the inner boundary temperature reaches it at about 19s, which means the whole porous material will be burned out at that moment. Whereas when $TPC=2.0$ is implemented, the outer boundary begins to ablate at about 15s, later than that of $TPC=1.0$, and the inner boundary temperature is away from the ablation temperature throughout the simulation time.

Fig.9 and Fig.10 show the law of the ablation amount and the ablation velocity varying with time under different values of TPC respectively. With the transpiration quality increasing, the time begin to ablate becomes later and the ablation velocity turns lower. Supposing that the value of TPC large enough to maintain the outer boundary without ablation is achieved, the control target of maintaining $s=0$ will come true. Thus, it is an efficient method to control a moving boundary by adjusting the transpiration control parameter.

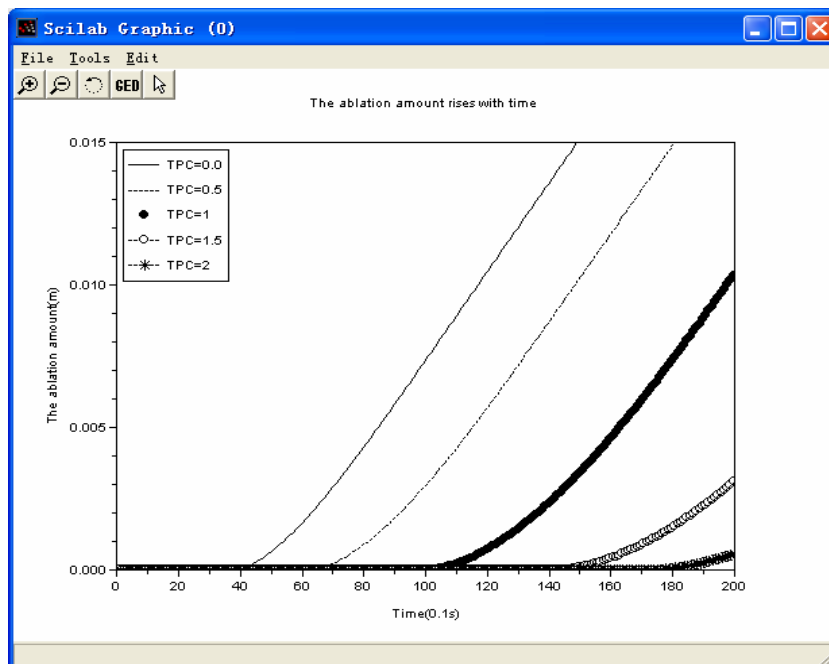


Fig. 9. Law of the ablation amount varying with time

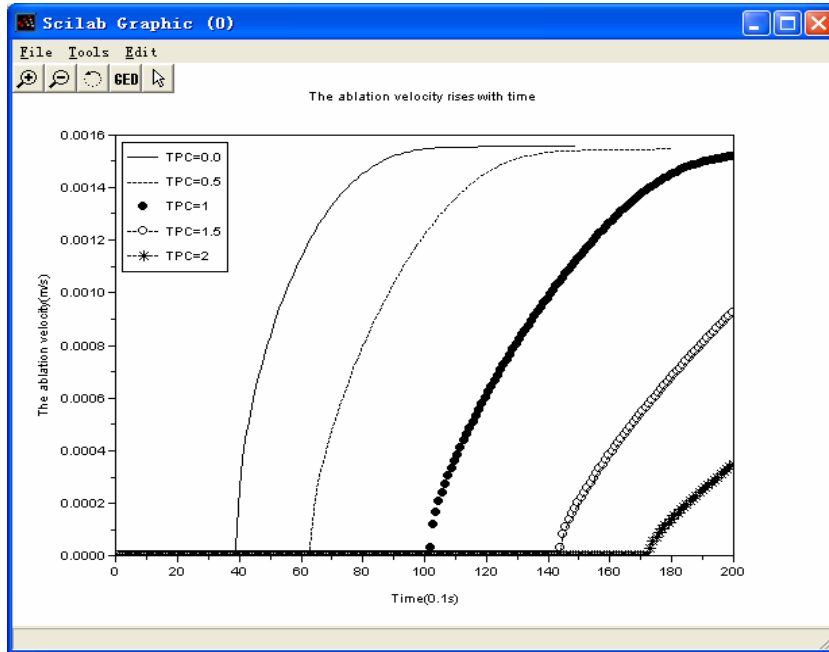


Fig. 10. Law of the ablation velocity varying with time

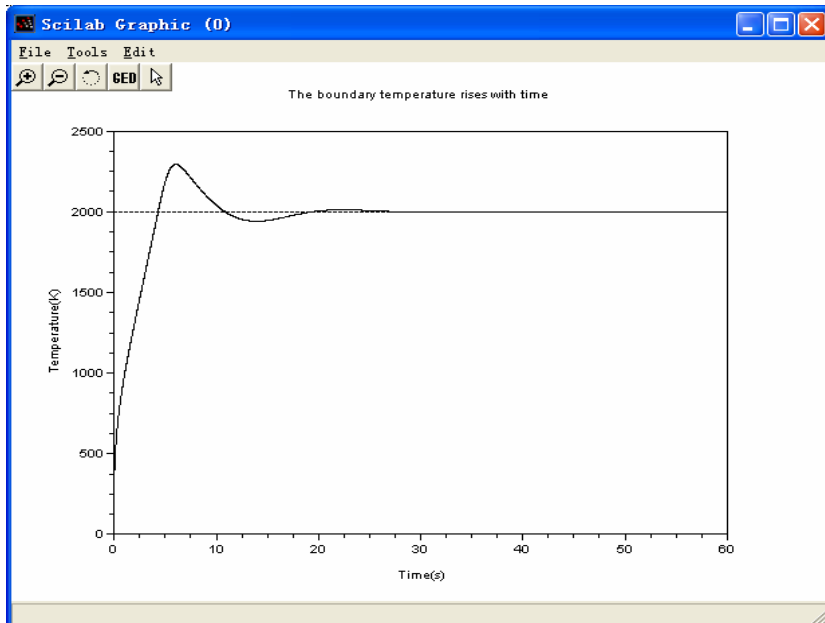


Fig. 11. Temperature of control point varying with time

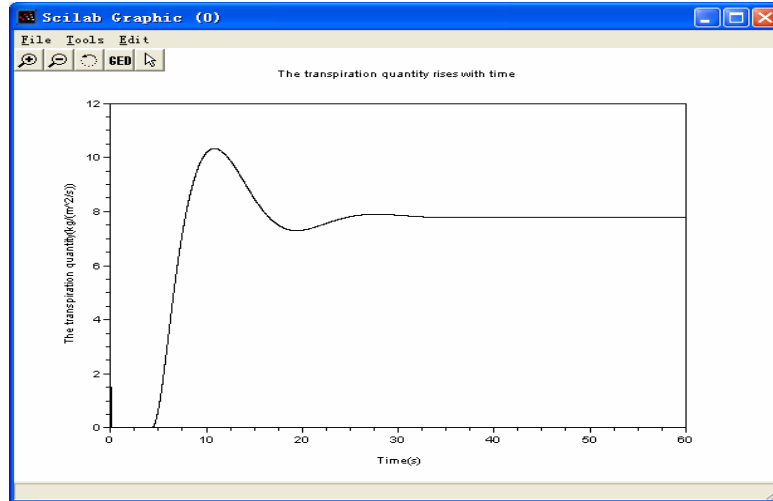


Fig. 12. Transpiration quantity varying with time

Fig.11 shows the temperature of control point varying with time with closed loop control under a target temperature of 2000K and the corresponding transpiration quantity is given in Fig.12. Though the initial temperature is far away from the target temperature, it still can approach the goal in less than 20 seconds and maintain it.

4.3 Design and development of toolbox with Scilab

In order to make the above simulation work more powerful in functions and more convenient to analyze the characteristics of porous material and coolant for researchers, transpiration cooling control toolbox is designed and developed with Scilab. This toolbox is developed in Scilab 4.0 environment, and provides the interface to set the simulation conditions and control parameters in a user-friendly environment.

The basic function of the toolbox is to display the above simulation result figures in menu items "Fixed Domain", "Variable Domain" and "Closed Control" shown in Fig. 13. Beside that, there are some other functions: menu item "File" shown includes sub items "Exit" used to quit the toolbox environment and "Quit Scilab" to quit the Scilab software and return to windows; menu item "Help" shown in Fig. 14 has the

function to show some help information about the transpiration cooling control toolbox and some demos.

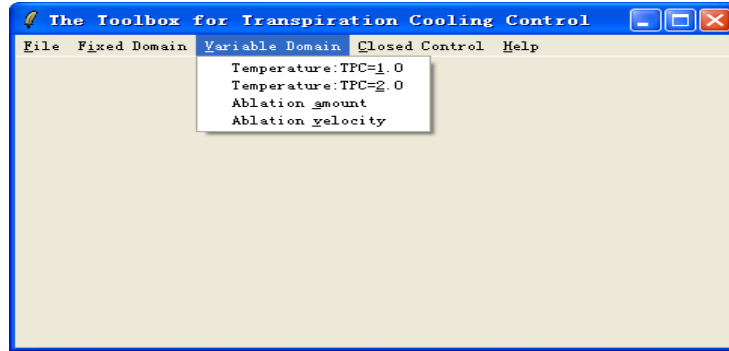


Fig. 13. Toolbox interface with menu "Variable Domain"

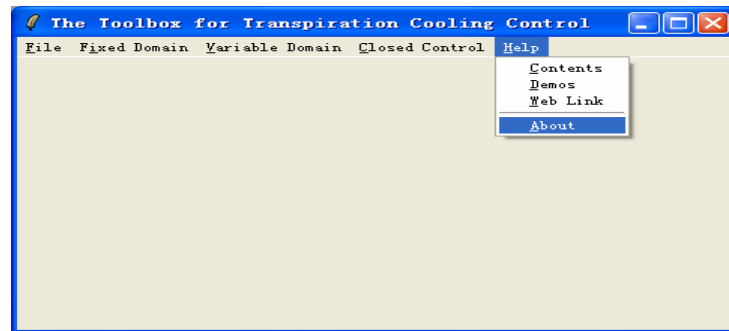


Fig. 14. Toolbox interface with menu "Help"

5. Conclusion and Remarks

In this paper, a set of models with distributed parameters for transpiration control process are established and some strategies and algorithms of numerical solutions are studied in depth.

Based on the open source scientific computing software Scilab, a series of numerical simulations are carried out. These simulation results demonstrate the dynamic response process and the distinguished control effect of the transpiration cooling control system in the thermal protection. With the transpiration quantity function in the series form of temperature, more efficient and more practical control effect on the temperature field can be obtained by closed loop control.

Furthermore, in order to enrich capabilities of Scilab tool boxes, a toolbox of the transpiration cooling control based on Scilab is designed and developed, which can be extended to be used in the modeling, analysis, synthesis and design of complex systems with distributed parameters.

References

1. Bing Sun, Baozhu Guo: Maximum principle for the optimal control of an ablation-transpiration cooling system. *Journal of System Science and Complexity*, Vol. 18, No. 3, (2005) 285-301
2. Xueshi Yang, Jingyuan Yu, Guangtian Zhu, et al: Notes on Research into Transpiration Cooling Control. *System Engineering and Electronics*, Vol. 18, No. 4, (1996) 2-11
3. Yanhou Xu, Guangyu Wu, Guanghui Gong, Xueshi Yang: The Analysis and Numerical Simulation of Dynamical Functions of the Transpiration Cooling Control System with Surface Ablation. *Chinese J. of Systems Engineering & Electronics*, Vol. 5, No. 1, (1994) 44-52
4. Lei Yu, Peixue Jiang, Zepei Ren: Experimental investigation and numerical simulation of convection heat transfer in transpiration cooling. *Journal of Engineering Thermophysics*, Vol. 26, No. 1, (2005) 95-97
5. Ji Sun, Xueshi Yang, Huolin Liu: A Survey of Variable Domain Transpiration Control Theory. *Control and Decision*, Vol. 21, No. 7, (2006) 721-725
6. Xueshi Yang: New Developments in Research of Air Suspension Trains. *Journal of Transportation Systems Engineering and Information Technology*, Vol. 3, No. 3, (2003) 66-70
7. Xueshi Yang: Variable Domain Heat Transfer Transpiration Control Theory. Beijing University Press, Beijing (2002)
8. Zaoding Yuan, Yajun Xie, Degui Liu, et al: Line-method for Transpiration Control Problem. *System Engineering and Electronics*, Vol. 18, No. 4, (1996) 12-19
9. Yajun Xie, Degui Liu, Zaoding Yuan: Line-method for Transpiration Control Problems of One Dimension. *COMPUTER ENGINEERING AND DESIGN*, Vol. 18, No. 1, (1997) 58-64
10. Baogang Hu, Xing Zhao, Mengzhen Kang: Introduction to Scientific Computing Software---SCILAB. Tsinghua University Press, Beijing (2003)
11. <http://www.scilab.org/>

Modeling of the Activated Sludge Process by Multi-model Approach

Linling zhao¹, Lifang zhou¹, Luwen zhou¹

¹Institute of Industrial Control, Zhejiang University, Hangzhou 310027

Email: linlingzhao@iipc.zju.edu.cn

Abstract. The activated sludge method No.1 is a complex biological wastewater treatment process with feature of nonlinear dynamic, large uncertainty, and multi input-output structure. ASM1 is widely accepted as biodegradation model. In order to simplify the nonlinearity of variables in ASM1, multi-model approach based on the clustering and partial least squares (PLS) algorithm is studied and applied into this field. The simulation is carried out by SCILAB[16], and the results reveal the high predictive quality of the multi-model proposed in this paper.

Keywords: ASM1, multi-model, clustering, PLS

1. Introduction

With the continuing enhancement of requirements on effluent quality, many researchers have paid more attention to the design, control and optimization of biological wastewater treatment systems (WWTPs). In 1983, the International Association on Water Quality (IAWQ, formerly IAWPRC) formed a task group, which was to promote the development and application of practical models for design and operation of biological wastewater treatment systems. The final results were presented as the activated sludge Model no.1 (ASM1), ASM2 and ASM3. ASM1 is widely accepted as biodegradation model, incorporating the basic biotransformation processes of an activated sludge wastewater treatment plant [1].

Although ASM1 is commonly regarded as state of the art, it have the following disadvantages: firstly, the calibration of all the kinetic parameters is a hazardous task; secondly, the model is highly nonlinear due to the appearance of Monod like kinetic in the mass balance equation; thirdly, the model is deterministic, while the influent characteristics and the kinetic parameters is unknown variation. To deal with these defects, some simplified ASM1 models are proposed. Kabouris and Georgakakos [8] assumed the dissolved oxygen concentration to be fairly high which enables the decoupling of the oxygen dynamics from the rest of the process dynamics, and only 5 ASM1 state variables are retained in their model. Zhao et al [9] introduced a simplified ASM1 model which S_{nh} and S_{no} are on-line measurable to describe the nitrogen dynamics in an alternating activated sludge process (ASP). Along the same line, Julien et al [10] developed a similar simplified ASM1-based model. The model

is characterized by the state variables S_{no} , S_{nh} and S_o and consists of two sub-models: one for the aerobic and one for the anoxic phase.

A second frequently encountered model reduction strategy is the application of singular perturbation theory. This technique allows the reduction of the dynamic state equations which are faster than the time scale of interest to algebraic equations and those which are slower to constants. Steffens et al [11] introduce an eigenvalue to state association technique that enables the quantification of the speed of a state, which is then used for reducing the model via singular perturbation analysis. Kabouris and Georgakakos [8] demonstrate that the incorporation of the multirate nature of the system into integration and discretization schemes, enhances computational efficiency which is a prerequisite for the successful implementation of optimal control and uncertainty propagation algorithms.

The main disadvantage of most of the above-mentioned reduced order models is that they start from a linearized model. The classical approach to obtain a (locally valid) linear approximation of model ASM1 is based on a Taylor series expansion (around some reference trajectory), of which only first-order terms are retained [13]. But the linearization approach is not convenient because the resulting model is only valid in the neighborhood of the selected reference trajectory and the choice of such a reference trajectory is not straightforward in practice. Instead of linearizing the model around a single steady-state (which rarely occurs in daily operation practice), Cheeruy et al [14] described the state dynamic with linear Gaussian differential equation, resulting from the linearization around three anticipated operational conditions. Along the same lines, Ilse Y. Smets et al [2] introduced a multi-model approach to identify a globally valid model based on time-varying plant data. Satu-Pia Mujunen et al [5] introduce partial least squares regression (PLSR) into ASP to find out the most relevant process or control parameters affecting the effluent quality and the filamentous bulking state of the biosludge.

In order to improve the control accuracy, multi-model method is developed as an efficient strategy to deal with nonlinear problem. Clustering algorithm, which digs the relationship from characteristic data of object, is successfully applied in the field of identification of multiple models. In this paper, K-means clustering and PLS algorithm are introduced into the ASP to solve the linearization of ASM1. The clustering method is to classify the states of complex control systems and PLS algorithm is used to identify dynamic equations of every state.

The simulation program in this paper is written in SCILAB 3.0. SCILAB is a kind of open source environment for scientific computation, and it is diffusely utilized in science and engineering.

The outline of this paper is as follow: Firstly, the ASM1 is introduced by some assumptions. Then the simulated data from ASM1 are decomposed to several fractions by K-means clustering, and each fraction captured is modeled using PLS algorithm. Finally, the simulation is done to validate the efficiently of the proposed method.

2. ASM1 model

ASM1 model is formed by 13 state variables and 19 parameters that describe eight biological processes. The differential equations of the 8 dynamic processes are given in ASM1.

To reduce complexity of the model, we give some assumptions as follow [1]:

- The system operates at constant temperature;
- The pH is constant and near neutrality, so the concentration of the alkalinity(S_{alk}) is constant;
- There is no bioreaction in the settler which is only a tank separating the sludge from the treated wastewater;
- The soluble inert organic matter (S_i) and the particulate inert organic matter (X_i) don't take part in the biodegradation, so they are discarded in the reduced model.

Consider a representative activated sludge process (Fig.1). Anoxic-oxic(A/O) technique is applied in it. Based on description of IAWQ for ASM1, the system model can be obtained.

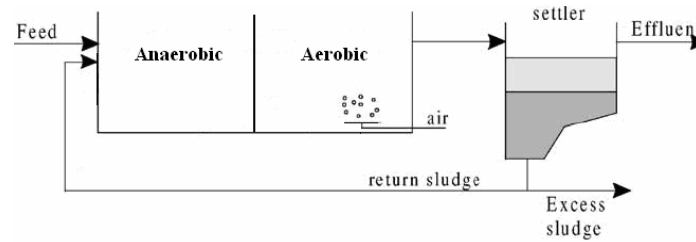


Fig. 1. Activated sludge process with anaerobic and aerobic tanks

Each state equation in the tanks has the following form:

$$V \left(\frac{dC_j}{dt} \right) = q_i C_{i,j} - q_o C_{o,j} \pm \sum_n VR_{n,j} \quad (1)$$

where C_j is concentration of state variable j , $C_{i,j}$ and $C_{o,j}$ are the concentration of variable j in the inflow and in the outflow respectively, q_i and q_o are the inflow and the outflow respectively, V is the bioreactor volume and $R_{n,j}$ is the reaction rate of variable j in the biological process n .

Two kinds of models should be made up: one for the anaerobic tank and the other for the entity of the aerobic tank and the secondary settler.

Based on the above assumptions, the reduced models are excluding three components- S_{alk} , S_i and X_i . The dynamic equations of the 9 remaining components in the anaerobic tank are as under:

$$\begin{aligned} \frac{dS_s}{dt} = & \frac{q_i}{V} S_{s,i} + -\frac{q_1}{V} S_s + \left\{ -\frac{\hat{u}_h}{Y_h} \left(\frac{S_s}{K_s + S_s} \right) \left[\eta_g \left(\frac{K_{o,h}}{K_{o,h} + S_o} \right) \left(\frac{S_{no}}{K_{no} + S_{no}} \right) \right] + \right. \\ & \left. K_h \frac{X_s / X_{bh}}{K_x + (X_s / X_{bh})} \left[\eta_h \left(\frac{K_{o,h}}{K_{o,h} + S_o} \right) \left(\frac{S_{no}}{K_{no} + S_{no}} \right) \right] \right\} X_{bh} \end{aligned} \quad (2)$$

$$\frac{dX_s}{dt} = \frac{q_i}{V} X_{s,i} + \frac{q_r}{V} 3 X_{s,o} - \frac{q_1}{V} S_s + (1 - f_p)(b_h X_{bh} + b_a X_{ba}) - \quad (3)$$

$$K_h \frac{X_s / X_{bh}}{K_x + (X_s / X_{bh})} \left[\eta_h \left(\frac{K_{o,h}}{K_{o,h} + S_o} \right) \left(\frac{S_{no}}{K_{no} + S_{no}} \right) \right] X_{bh}$$

$$\frac{dS_{nd}}{dt} = \frac{q_i}{V} S_{nd,i} - \frac{q_1}{V} S_{nd} + \{ -K_a S_{nd} + \quad (4)$$

$$K_h \frac{X_{nd} / X_{bh}}{K_x + (X_s / X_{bh})} \left[\eta_h \left(\frac{K_{o,h}}{K_{o,h} + S_o} \right) \left(\frac{S_{no}}{K_{no} + S_{no}} \right) \right] \} X_{bh}$$

$$\frac{dX_{nd}}{dt} = \frac{q_i}{V} X_{nd,i} + \frac{q_r}{V} 3 X_{nd} - \frac{q_1}{V} X_{nd} + \quad (5)$$

$$(i_{xb} - f_p i_{xp})(b_h X_{bh} + b_a X_{ba}) - K_h \frac{X_{nd} / X_{bh}}{K_x + (X_s / X_{bh})} \left[\eta_h \left(\frac{K_{o,h}}{K_{o,h} + S_o} \right) \left(\frac{S_{no}}{K_{no} + S_{no}} \right) \right] X_{bh}$$

$$\frac{dS_{nh}}{dt} = \frac{q_i}{V} S_{nh,i} - \frac{q_1}{V} S_{nh} + \quad (6)$$

$$\left\{ -i_{xb} \hat{u}_h \left(\frac{S_s}{K_s + S_s} \right) \left[\eta_g \left(\frac{K_{o,h}}{K_{o,h} + S_o} \right) \left(\frac{S_{no}}{K_{no} + S_{no}} \right) \right] + K_a S_{nd} \right\} X_{bh} - \hat{u}_a \left(i_{xb} + \frac{1}{Y_a} \right) \left(\frac{S_{nh}}{K_{nh} + S_{nh}} \right) \left(\frac{S_o}{K_{o,a} + S_o} \right) X_{ba}$$

$$\frac{dS_{no}}{dt} = \frac{q_i}{V} S_{no,i} - \frac{q_1}{V} S_{no} - \quad (7)$$

$$\hat{u}_h \eta_g \left(\frac{1 - Y_h}{2.86 Y_h} \right) \left(\frac{S_s}{K_s + S_s} \right) \left(\frac{K_{o,h}}{K_{o,h} + S_o} \right) \left(\frac{S_{no}}{K_{no} + S_{no}} \right) X_{bh}$$

$$\frac{dX_{bh}}{dt} = \frac{q_i}{V} X_{bh,i} + \frac{q_r}{V} 3 X_{bh,o} - \frac{q_1}{V} X_{bh} + \quad (8)$$

$$\left\{ \hat{u}_h \left(\frac{S_s}{K_s + S_s} \right) \left[\eta_g \left(\frac{K_{o,h}}{K_{o,h} + S_o} \right) \left(\frac{S_{no}}{K_{no} + S_{no}} \right) \right] - b_h \right\} X_{bh}$$

$$\frac{dX_{ba}}{dt} = \frac{q_i}{V} X_{ba,i} + \frac{q_r}{V} 3 X_{ba,o} - \frac{q_1}{V} X_{ba} + -b_a X_{ba} \quad (9)$$

$$\frac{dX_p}{dt} = \frac{q_i}{V} X_{p,i} + \frac{q_r}{V} 3 X_{p,o} - \frac{q_1}{V} X_p + f_p (b_h X_{bh} + b_a X_{ba}) \quad (10)$$

In aerobic condition, the reaction rates differ from the ones in anaerobic condition, because biodegradation rate varies along with the concentration of soluble oxygen. The anaerobic and the secondary settler are taken as an integer here. The equations of the components in the integer of the aerobic tank and the settler are approximate to

the ones in the anaerobic tank, and the differences between the two models owe to differences of the flows and the reaction rates in the two tanks. Readers can obtain the model in the aerobic condition according to the equation (1) and refer to [1].

The effluent quality is generally characterized by total COD (for organic compound concentration) and/or total nitrogen (for nitrogen concentration). The expression for total nitrogen is:

$$N_{tot} = S_{nh} + S_{no} + S_{nd} + X_{nd} + X_{ni} + i_{xb}(X_{bh} + X_{ba}) + i_{xp} X_p . \quad (11)$$

In this paper, dissolved oxygen concentration (S_o) in the anaerobic tank, the inflow (q_i) and recycle rate (r) are taken as the input variables, and total nitrogen as the output variable. The relationship of the input and output is characterized based on the dynamic equations of components. Let the input varies in a certain area, then we can get the dynamic output change along with the inputs.

After ASM1 system model has been made up, some practical parameters are imported to be simulated by ASM1. Then the simulated data will be operated by K-means clustering and PLS algorithms. Next, K-means clustering and PLS are briefly introduced.

3. Identification of multiple models for ASM1 based on K-means clustering and partial least squares (PLS)

3.1 Sampling

The variables are determined with certain sampling interval. The sampling intervals of the inputs (S_o , q_i , r) and output (concentration of total nitrogen) are all 0.05 days.

3.2 Clustering

Clustering, as an important unsupervised learning problem, could find a structure in a collection of unlabeled data. Clustering is used abroad in many fields such as data mining, pattern recognition, statistics and identification of multiple models. The basic course is that the clustering algorithm is used to distinguish different states, and then each different state will be described in dynamic equations.

K-means clustering, developed by Mac Queen in 1967, is a simple clustering algorithm with fast convergence rate. However, K-means clustering has some disadvantages in it. Firstly, a priori knowledge of number of clusters is needed for K-means clustering algorithm. Secondly, K-means clustering is an iterative method, and it has the immanent disadvantage of iterative methods that there could be different clustered results for different initial selections. Some improvement of the initialization and data condensation in K-means clustering was made by Zhou Lu-wen et al [6], and the experiment result show that the improved method incorporating partial least

square is good at multiple modeling. In this paper, the same method is used to make up the multiple models approximate to ASMI.

The computational steps of the advanced K-means clustering in this paper are as follow:

Input: D—the data set of n sample elements described with 5 attributes(the first three attributes are sampled values of S_o , q_i , r for the current sampling point, and the others are values of total nitrogen sampled at current point and at last sampling point; K—predefined number of clusters, K=4 here;

Output: C—the set of K cluster centers; q—the number of clusters for the current instant time; S—the set of unique pattern strings for data set D;

Step 1): Normalize the complete data set D and set desired number of clusters K;

Step 2): Using CCIA[9] algorithm to compute the initial cluster centers for K-means algorithm, create initial partitions using Euclidean distance and execute K-means over data set D; finally, store the cluster labels for each element of data set D;

Step 3): Repeat step 2) till all elements have their own pattern strings, then find the unique strings and store it into S. if the data points have the same pattern strings, we consider that they should be distributed into the same cluster. And the number of the unique strings is the current number of clusters; we can note it as q ($K < q < Km$);

Step 4): if $q > K$, go to step 5); else go to step 6);

Step 5): Compute each cluster center $\{c_1, c_2, \dots, c_K\}$, the distance $dist(c_i, c_j)$ and the threshold

$$dist(c_i, c_j) = \sum_{p=1}^5 (c_{ip} - c_{jp})^2 \quad p = 1, 2, \dots, 5 \cdot \quad (12)$$

$$threshold = \frac{1}{2} \max(dist(c_i, c_j)) \quad i, j = 1, 2, \dots, 4 \cdot \quad (13)$$

If the distance between two of q clusters is shorter than the threshold, merge those clusters and $q=q-1$; repeat again until all of distance between any two are larger than the threshold; then, set $K=q$, and go to step 6); the rule of merging is following;

$$c_{merge} = \frac{n_i \times c_{ip} + n_j \times c_{jp}}{N}, \quad p = 1, 2, \dots, 5 \quad N = n_i + n_j \cdot \quad (14)$$

Where: c_i, c_j express the clusters which will be syncretized, and n_i, n_j denote the number of data elements which belong to c_i, c_j respectively.

Step 6): Find the means of these K-clusters to get final K-centers that will be used as initial cluster centers, and classify all samples using K-means clustering algorithm [3] to obtain ultimate divisions.

3.3 Modeling using PLS

The data set has been divided to four clusters, and next each cluster will be regressed using partial least squares (PLS). PLS, which has recently attracted much attention is a comparatively new method of constructing regression equations. It has some great advantages. Firstly, it may be used with any number of explanatory variables, even far

The simulated data are divided into 4 clusters by K-means clustering algorithm. Then the characteristic of each cluster is analyzed by PLS algorithm. The local linear equations of data set are as follow:

$$y1(k)=[-0.0122 \ -0.0092 \ 0.000297]x(k)+0.385y1(k-1)+0.470y1(k-2)+4.487 . \quad (17)$$

$$y2(k)=[0.159 \ 0.119 \ -0.150]x(k)+0.272y2(k-1)+0.375y2(k-2)+9.071 . \quad (18)$$

$$y3(k)=[-0.103 \ -0.0775 \ 0.0131]x(k)+0.279y3(k-1)+0.483y3(k-2)+8.535 . \quad (19)$$

$$y4(k)=[-0.0727 \ -0.0545 \ -0.168]x(k)-0.0523y4(k-1)+0.450y4(k-2)+20.250 . \quad (20)$$

where $x(k)$ is the input vector, that is, $x(k)=[S_o \ q_i \ r]^T$, $y(k)$ is the output, concentration of the total nitrogen and $y(k-1)$, $y(k-2)$ are samples before $y(k)$.

The multiple models by PLS are simulated in SCILAB, which is a powerful open source software diffusely used in science and engineering, and the results are shown in Fig.2. The dashed curves are from ASM1, and the real curves are by the multiple models.

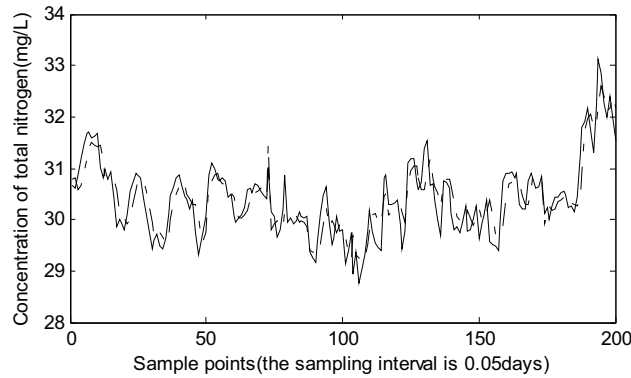


Fig. 2. the simulated results of multi-model (data from ASM1 (-) and multiple models(-))

From Fig.3, there is a little distinction between the multiple models and the ASM1 model. Fig.3 depicts the error charts between data set and results of multi-model. Four fractions in Fig 3 represent error curves of four cluster models respectively.

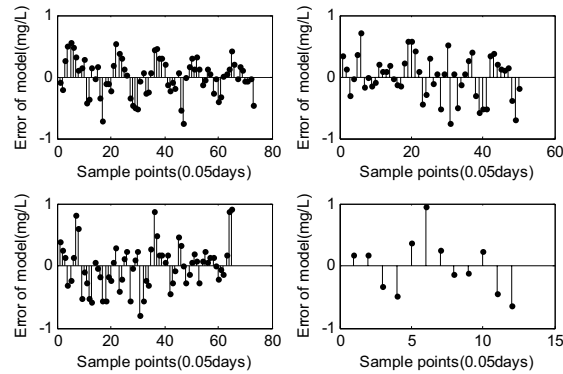


Fig. 3. Error charts respectively for 4 clusters of data set 1 between the simulated data of ASM1 and the multiple models

5. Conclusion

Based on practical data of wastewater plant, three variables (S_o , q_i , r) in a representative activated sludge process with A/O technique are taken as the input of ASM1 model and the concentration of total nitrogen is regarded as output. Then the linear multiple models is obtained by using K-means clustering and PLS algorithms in order to get approximate. The simulation results using SCILAB reveal that the multiple models by the K-means clustering algorithm incorporating with PLS can fit the nonlinear system of ASM1.

State variables in the ASM1

S_{alk}	alkalinity	S_i	soluble inert organic matter
S_{nd}	soluble organic nitrogen	S_{nh}	soluble ammonium nitrogen
S_{no}	soluble nitrate–nitrite nitrogen	S_o	dissolved oxygen concentration
S_s	soluble biodegradable substrate	X_{ba}	active autotrophic biomass
X_{bh}	active heterotrophic biomass	X_i	particulate inert organic matter
X_{nd}	particulate organic nitrogen		
X_s	particulate substrate biodegradable		
X_p	particulate products from biomass decay		

Acknowledgments. This work is supported by a grant from the National Natural Science Foundation of China (No. 60474064) and the Fund of produce-study-research Science and Technology Cooperation of Hangzhou City to Zhou LiFang.

References

1. Ulf-Jeppsson, et al. A General Description of the IAWQ Activated Sludge Model No. 1
2. Ilse Y. Smets, et al. Linearization of the activated sludge model ASM1 for fast and reliable predictions[J]. *Water Research* 37 (2003) 1831-1851
3. Shehroz S. Khan, Amir Ahmad. Cluster center initialization algorithm for K-means clustering[J]. *Pattern Recognition Letter*, 2004, 25
4. Paul H. Garthwaite. An Interpretation of Partial Least Squares[J]. *Journal of the American Statistical Association*, 1994, 89(425)
5. Satu-Pia Mujunen, et al. Model of activated sludge plants treatment efficiency with PLSR: a process analytical case study. *Chemometrics and Intelligent Laboratory Systems* 41 (1998) 83-94
6. Zhou Lu-wen, Zhou Li-fang. Multiple modeling method based on advanced k-means clustering. *Journal of university of science and technology of China*, 2005, Vol.35
7. Shang Miner, Du Shuxin, Wu Tiejun. An introduction of control strategies for activated sludge wastewater treatment. *Techniques and Equipment for Environmental Pollution Control*, 2002
8. Kabouris JC, Georgakakos AP. Accounting for different time scales in activated sludge process control. *Water Sci Technol* 1992;26(5-6):1381-90.
9. Zhao H, Isaacs SH, S[^]eberg H, Kummel M. An analysis of nitrogen removal and control strategies in an alternating activated sludge process. *Water Res* 1995;29(2):535-44.
10. Julien S, Lessard P, Babary JP. A reduced order model for control of a single reactor activated sludge process. *Math Compute Model Dyn Systems* 1999;5(4):337-50.
11. Steffens MA, Lant PA, Newell RB. A systematic approach for reducing complex biological wastewater treatment models. *Water Res* 1997;31(3):590-606.
12. Keesman KJ, Spanjers H, van Straten G. Analysis of endogenous process behaviour. *Water Sci Technol* 1998;37(12):227-35.
13. Leith DJ, Leithead WE. On the identification of nonlinear systems by combining identified linear models. In: Troch I, Breitenacker F, editors. *Proceedings third MathMod Vienna*, ISBN 3-901608-15-X, ARGESIM Report No. 15, vol. 1. Vienna, Austria, 2000. p. 407-10.
14. Ch!eruy A, Panzarella L, Denat JP. Multimodel simulation and adaptive stochastic control of an activated sludge process. In: *Proceedings of the IFAC symposium on modelling and control of biotechnological processes*. Helsinki, Finland, 1982.
15. Wang huafeng. The research on modeling and control for A/O technics of wastewater treatment based on ASM2. Dissertation for the degree of Master of Engineering in Zhejiang university, March, 2005
16. Baogang Hu, Xing Zhao, Mengzhen Kang: *Introduction to Scientific Computing Software-SCILAB*. Tsinghua University Press, Beijing (2003)

SAO: Modeling and Simulation of Adaptive Optics in the Scilab/ Scicos Environment

Chen jingyuan, Gan guangyong, Tao yingxue

The institute of Applied Physics and Computational Mathematics, Beijing, 100088, China

jingyuan_chen@yahoo.com.cn

Abstract: SAO is an open source, cross-platform, and user-friendly toolbox based on the Scilab and Scicos environment for modeling and simulation of wave optics, especially for the adaptive optics system. This paper describes the functionality available in SAO, gives some application examples, and discusses future plans for this toolbox.

Keywords: Modeling and Simulation, Adaptive Optics, Scilab/Scicos, Open Source

1 Introduction

Through nearly 40-years development, Adaptive optics (AO) [1] is now a mature technology widely used in optical astronomy, high energy laser (HEL) systems and other modern optical equipments. At the same time, AO is also a dynamic field of research, new architectures, algorithms, and applications being proposed have pushing the original concept (first suggested by Babcock at 1953) to the technological limits (e.g. MCAO: Multi-Conjugate AO system).

To grasp the behavior and the performance analysis of these optical systems, a theoretical analyse is obviously the first choice, while this approach can't work well in practice, partly because the overly complexity of AO systems and its complex working environment, partly because a number of problems are not open to analytical solution. In fact, this kind of study involves the complex nature of the turbulent atmospheric environment, and also the complexity of interactions of many connected instruments and concepts in a whole AO system. Within this background of the

growing complexity of these system architectures and the expense of their implementation, numerical simulations can play a very useful role in providing a quantitative evaluation of their capabilities. The advantage of a simulation lies in the fact that absolute truth is known, and the fact it is very cheap and controllable. By generating random realizations of atmospheric turbulence and performing wave propagation through this turbulence, a simulation can model the effects of turbulence degradation, and can preserve correlations between beams that arise from tilt and focal anisoplanatism. A simulation can accurately represent the effects of scintillation, which are important in high precision applications such as direct planetary imaging and in atmospheric monitoring using Scidar. With models of the adaptive optics system components and by replicating the reconstructor and control law, a simulation can accurately represent the process of wavefront sensing and correction. Such functionality may be used to generate performance predictions for an adaptive optics system, and to evaluate the system design. The highly controlled environment provided by simulation affords the opportunity of performing quantitative comparisons between different reconstruction and control algorithms. Finally, a simulation can generate field dependent PSFs (Point Spread Function) in the science focal plane, which may be used to evaluate scientific performance metrics other than residual wavefront error. And together with a simulated set of telemetry data, these PSF's may be used to test PSF reconstruction algorithms.

There are a large number of softwares that can execute adaptive optics simulation. Most of them are commercial or proprietary products, and their inner mechanisms are not known to other researchers. Some simulation softwares are open source, while these codes are almost all written in FORTRAN, or C/C++, and they have not good interfaces for other people to use. Many of today's codes (e.g. AOTools [2] developed by tOSC) are written entirely or partly in MATLAB, which has the advantages of an extensive library of math routines, very fast matrix operations, plenty of other toolboxes and very well graphical outputs. The main limitations of MATLAB are the cost of the licenses and the slow speed for non-matrix iterative calculations.

Recently we have been developing a toolbox (we called it SAO) based on Scilab/Scicos [3] environment in which we can model and simulate most adaptive optics devices or other AO-related problems. Although this toolbox is mainly designed to satisfy the requirement of AO simulations, we also moderately consider

the needs to some other optics simulation system (e.g. the simulation of general Wave optics and imaging devices) and designed some software modules for them, so this toolbox can also be used to simulating other optics problems beyond the AO field. Most of the programs of our SAO toolbox are written by C/C++ language or based on some excellent open source C/C++ libraries, so it is efficient and cross-platform. At the same time, we also fully make use of the powerful graphical and interface program ability, especially the dynamic modeler and simulator (Scicos) of the Scilab environment, so it is very easy for other researchers or beginners to learn and use this toolbox to simulate their peculiar adaptive or other optical problems. Our toolbox will be released under open-sourced GPL license and all other people who are interested in this software can download, modify, or use it if they abide this license.

In this paper, we will give a general description of the SAO toolbox (now the steady version is 0.2) and give some examples of simulation applications. At the last we also give some plans of the functional extension for SAO. While first we need give an introduction of the basic concept of adaptive optics and its simulation, which is an unacquainted field to most of other researchers.

2 Adaptive optics concept and some relevant simulation software

Adaptive optics is a newly developed technology in order to improve the performance of the reflecting telescopes by reducing the effects of optical aberration or distortions, especially the distortions caused by the Earth's atmosphere. An classical adaptive optics system tries to correct these distortions, using quality-of-image detector (sensors), tip/tilt mirrors and deformable mirrors (correctors), and a controllable computer that receives input from the detector and calculates the optimal deformation of the mirror, and the corresponding control signals which will be used to make the deformable (or other corrector devices) change its shape or move. Figure 1 is an outline plot of the main components and its basic work diagram of a classical adaptive optics (SCAO: Single-Conjugate AO system).

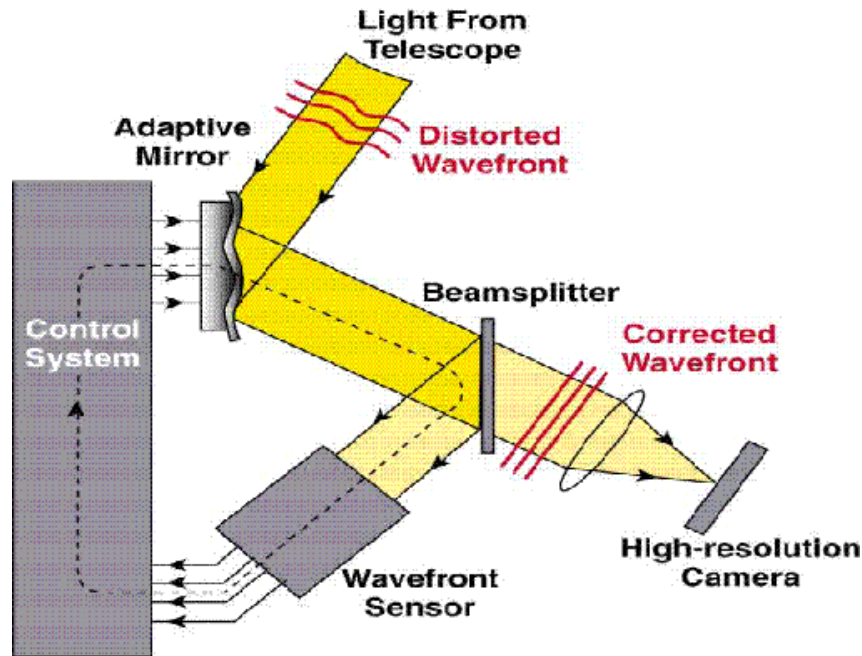


Fig. 1. structure of a classic SCAO system

Generally say, AO simulation code can be roughly divided into two types. At the lower level are the physical and engineering codes that try to model and predict the complex interactions or performance based on first principles. They can predict detailed system performance characteristics such as evolution of wave field phase, intensity profiles, and a number of other information such as deformable mirror (DM) actuator commands or wavefront sensor pixel output, etc. These are also known as wave optics, or time domain codes. At the top level are the scaling law or systems engineering codes. These provide much faster predictions based on scaling laws or empirical fits to simulation or experimental results. At below, we will give a briefly description of some AO simulation codes which are relevant to our SAO toolbox. They are WaveTrain, CAOS, LightPipes, and Arroyo. All of them belong to the physical or engineering codes.

WaveTrain [4] is a user-friendly wave and adaptive optics propagation code mainly developed and distributed by MZA Associates Corporation, USA, under funding from Air Force Research Laboratory, USA. It has been implemented using Tempus (another main product of MZA), which is a general purpose simulation

development environment similar to the Simulink toolbox of Matlab. Using Tempus as the foundation, WaveTrain construct a lot of component libraries for wave and adaptive optics, where each type of software component models a different system component or physical effect.

CAOS [5] (Code for Adaptive Optics System) is a set of software tools specifically designed to allow the modeling and simulating of any kind of adaptive optics system originally developed in the framework of the “TMR Network on Laser guide star for 8 meter class telescopes” funded by the European Community. Similar to WaveTrain, CAOS is also a user-friendly AO simulation tool, and is essentially a set of modules designed to be used within a graphical programming environment – the CAOS Application builder (another clone of Simulink), where the data flow between modules can be defined, and the parameters of each module can be set. Dissimilar to WaveTrain, CAOS is an open source software, but it is implemented in the IDL (Interactive Data Language, designed by RSI Corp).

LightPipes [6] (written by Dr. Gleb Vdovin) is a tool that can model free space diffractive propagation and the effects of various optic devises such as apertures and lenses. The simulation used the pipe command to transfer the wavefront data between individual programs. This tool contains no support for modeling atmospheric turbulence and other adaptive optical devises. Most of this programs are written in C and is open-sourced. It has also a Matlab version, but this version is only can run under limited mode (only 64x64 grid dimension at most).

Arroyo [7], developed by Dr. Matthew Britton at California Institute of Technology, is an open source C++ class library designed for modeling and simulation of electromagnetic wave propagation through atmospheric turbulence and adaptive optics system. Based on the object-oriented programming methodology and other modern software engineering technology (e.g. coordinate free geometric programming, the object factory design pattern, etc), this library is an excellent software project which achieves powerful abstractions and good modularity, reusability, and extensibility. This library also supports parallelized computation.

3 SAO: goals and schemes

WaveTrain and CAOS are user-friendly, but they are proprietary in essential (Though SCAO is open source, the IDL is proprietary), and their efficiencies are not very well,

because they are implemented by higher language. LightPipes and Arroyo are more efficient, while it is hard to use them to simulate particular optics problems.

We want to develop a user-friendly simulation tool which can model and simulate wave or adaptive optics easily, at the same time, it must be as efficient and credible as other tested FORTRAN or C/C++ code. We also expect it is open-sourced and all of other people inside or outside the field of adaptive optics can use, modify, and improve it arbitrarily.

We select the Scilab software as our fundamental developing platform (so it called), because Scilab is open-sourced, and easy to learn and use for beginner. More important, Scilab offers interfaces to external libraries written in other lower level languages such as C and FORTRAN, which make it possible to use legacy code and other open-source software. Furthermore, Scilab has a powerful visual modeling and simulation environment, called Scicos, which is very semblable to the Simulink of Matlab, or the Tempus of WaveTrain, or the Application Builder of CAOS. Using Scicos, a simulation can be built by connecting together the required occurrences of the desired modules (blocks), respecting only the logical constraints given by their formalized input and output type, and then the block diagram will be analysed and executed by Scicos itself. Complex simulation application are thus simply created by assembling the elementary building blocks in a straightforward manner, so that the user can concentrate on the scientific aspects of his special problems, while mundane coding problems are managed by some automatic mechanism.

In developing this tool we did not "reinvent the wheels". All of the basic techniques and algorithms used in wave optics which had already been implemented and tested for previous wave optics codes, and all of the useful open-source codes which we can acquire, can be used by our toolbox. Where it made sense, we used the legacy or open-source code directly. In some cases minor modifications were needed to satisfy integration requirements, and in some cases it was simpler to rewrite the code. For open-source codes which we have used in SAO, we refer to Arroyo and LightPipes.

4 Status and functional description of SAO (version 0.2)

We have worked on this toolbox almost a year and now it has been able to model and simulate some simple wave and adaptive optics problems. In this part, we will describe the basic structure and function of SAO.

SAO contains several elements, the mains are:

- Some external optics routines libraries. They are kernels of the SAO, and all of the optics computations are actuated by them. Now two external library has been integrated into SAO: lightPipes and arroyo. We have made some modifications to their original codes in order to let them accord with Scilab/Scicos environment, or let them be able to cross-platform run under Microsoft Windows/Visual Studio .net 2003 and Gnu/Linux. Optics libraries also include another routines library we called Wrapper, which is some C wrappers to Arroyo C++ class library. We need such a C wrapper library because Scilab/Scicos now has not been able to interface C++ class library directly.
- C routines libraries that interface external optics library to Scilab/Scicos. There are two such C routines libraries: the interface function library of Scilab functions for Optics, and the computational function library of the Scicos blocks for Optics. Now we have constructed about one hundred Scilab functions to optics, and about thirty Scicos blocks to optics.
- Some Scilab scripts, include the interface functions of Scicos blocks for Optics, and some other useful functions written by original or SAO' Scilab functions. There are also some TCL/TK scripts we use them to construct some beautiful GUI.
- Many examples of Scicos diagrams and Scilab simulation scripts. We have written these examples using the Scilab functions and Scicos blocks of our SAO toolbox, so that beginners can learn and grasp the use of our toolbox. Further, users can construct their special simulation models of wave optics or adaptive optics at the base of these examples we have provided.

At below, we will give an outline description of the main Scilab functions and Scicos blocks of our SAO toolbox. We start from the Scicos blocks of SAO. Figure 2 gives out the Scicos palette of SAO.

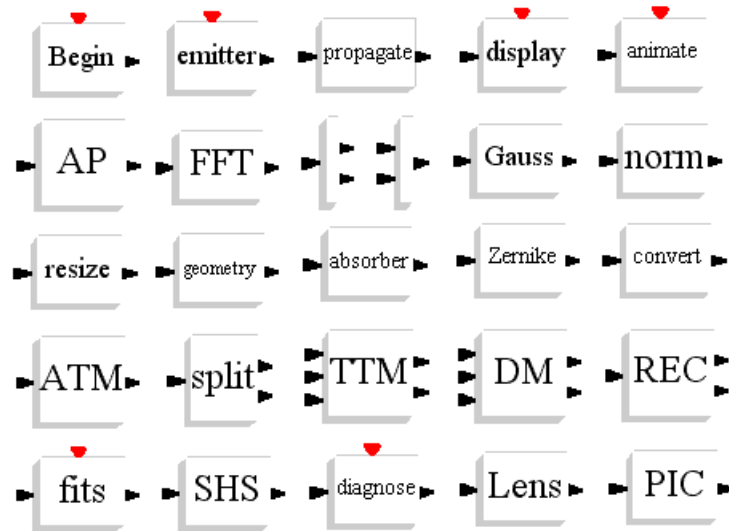


Fig. 2. the optical Scicos palette in SAO

As Figure 2 points out, there are now more than twenty optical Scicos blocks in SAO. In them, the blocks labeled Begin and Emitter can be used to model optical “sources”, while Display, Animate, Fits, and Diagnose blocks model optical “sink” (or say it detectors). All the other optical blocks in this Figure are used to model and simulate various optical equipment, such as atmospheric refractive layers (ATM), deformable mirrors (DM), Shack-Hartmann sensors (SHS), apertures (AP), beam splitter (Split), lens (Lens), reconstructor of wavefronts (REC), proportional integral controller (PIC), or other optical transforms, such as geometrical optics transform (the Geometry block), the normalization of intensity (Norm), and frame transforms (Convert), fast Fourier transform (FFT), and so on. We will add more and more blocks to let it be able to model other optical devices or transform at later versions of SAO.

Besides above Scicos blocks for optics, we also write a number of Scilab function to model various optical equipments or optical processes and now the number of these functions is about one hundred. Now we give them some description. They can be include in these types (we list most of them, and give some brief description for its function in brackets):

- Emitters: `begin` (form unit plane wave); `emitter` (construct light emitter); `gauss` (form Gauss beam); `pl_wave` (form general plan wave); `sp_wave` (form spherical wave or point source); etc.
- Detector or Output: `dwf_fits` (write wavefront to fits file); `cros_out` (write the cross sections of a field); `file_pgm` (write field to a pgm files); `file_ps` (write field to a ps file); `file_int` (write the intensity of field to a file); `file_pha` (write the phase of field to a file); `znk_fits` (write Zernike mode to a fits file); `ref_atm_layer_fits` (write a refractive atmospheric layer to a fits file); `atm_mod_layers_fits` (write all the layers of a atmospheric model to fits files); `reconstructor_fits` (write the reconstructor of a AO system to a fits file); etc.
- Wavefronts and Propagators: `wavefront_header` (create the wavefront header of a field); `set_dwfh_timestamp` (set the timestamp of a wavefront); `set_dwfh_curvature` (set the curvature of a wavefront); `fresnel` (Fresnel propagator); `forward` (FFT or spectra method propagator); `forward` (Direct integration method propagator); `lens_forward` (FFT or spectra method propagator for spherical wave); `lens_fresnel` (Fresnel propagator for spherical wave); `steps` (Finite difference method propagator); `geom_propagation` (geometric optics propagator); `near_angular` (near field angular propagator); `near_fresnel` (near field Fresnel propagator); `far_fresnel` (far field Fresnel propagator); `far_fraunhoffer` (far field Fraunhoffer propagator); `fresnel_GRT` (far field Fresnel and Goertzel-Reinsch tranform propagator); `fraunhoffer_GRT` (far field Fraunhoffer and Goertzel-Reinsch tranform propagator); etc.
- Telescope aperture and its transforms: `circ_ap` (circular aperture); `rect_ap` (rectangular aperture); `hex_ap` (hexagonal aperture); `annul_ap` (annular aperture); `spid_annul_ap` (spidered annular aperture); `til_hex_ap` (tilted hexagonal aperture); `aperture_transform` (transform a wavefront by aperture);
- Atmosphere models: `power_law` (create power law of turbulent spectrum); `von_karman_power` (Von-Karmman power spectrum); `greenwood_power` (Greenwood power spectrum); `null_inner` (null inner scale of a power law); `exponent_inner` (Tartaskii exponent inner scale); `frehlich_inner` (Frehlich exponent inner scale); `power_spectrum` (isotropic power law spectrum of turbulence); `null_subm` (null subharmonic method); `quad_pix_subm` (quad-pixel subharmonic method); `Lane_subm` (Lane subharmonic method); `Johansson_Gavel_subm` (Johansson-Gavel subharmonic method);

`general_subm` (generalized subharmonic method); `ref_atm_lay` (create refractive atmospheric layer); `ref_atm_lay_transform` (transform a wavefront by refractive atmospheric layer); `hardy_wind` (create Hardy wind); `get_hardy_wind_vectors` (get the wind vectors of Hardy wind model); `ref_atm_model` (create generalized refractive atmospheric model); `Ellerbroek_Cerro_Pachon` (Ellerbroek Cerro Pachon atmospheric model); `Ellerbroek_Mauna_Kea` (Ellerbroek Mauna Kea atmospheric model); `Palomar_DIMM_MASS` (Palomar DIMM MASS atmospheric model); `Hufnagel_Valley` (Hufnagel Valley atmospheric model); `SLCSAT_day` (SLCSAT daytime atmospheric model); `SLCSAT_night` (SLCSAT nighttime atmospheric model); `TMT_SRD_v13_Cn2` (TMT SRD v13 Cn2 atmospheric model); `Gemini_GLAO_study` (the Gemini GLAO study model); `get_atm_mod_dwfh` (get the wavefront of an atmospheric model); `get_atm_mod_layers` (get all the refractive layers of an atmospheric model); `atm_mod_lay_number` (get the number of refractive layers of an atmospheric model); `atm_mod_lay_heights` (get the heights vectors of a model); etc.

- Common optical devices and their corresponding transforms: `lens` (create a lens); `file_ter` (general file filter); `zernike` (Zernike mode filter); `random` (random filter); `b_mix` (beam mixer); `b_split` (beam splitter); `l_amplify` (Laser amplifiers); `tilt` (tilt modes of the field); `absorber` (beam absorber or amplifier); etc.
- AO appropriate devices and their transforms: `lnslt_array` (Shack Hartmann lenslet array); `lnslt_arr_transform` (transform wavefront by Shack Hartmann Sensor); `create_shcentroids` (create Shack-Hartmann centroids); `actuator_array` (create actuator array for a deformable mirror); `ideal_dm` (create ideal deformable mirror); `ideal_dm_transform` (transform wavefront by a ideal deformable mirror); `set_dm_timestamp` (set the timestamp of a deformable mirror); `set_dm_actuator_positions` (set the positions of the actuators for deformable mirror); `set_dm_actuator_commands` (set the command vectors of the actuators for deformable mirror); `ideal_ttm` (create ideal tip-tilt mirror); `set_ttm_timestamp` (set the timestamp of a tip-tilt mirror); `set_ttm_commands_vector` (set the command vectors of a tip-tilt mirror); `ideal_ttm_transform` (transform a wavefront by a tip-tilt mirror); `pi_controller` (create a general proportional integral controller); `ttm_pi_controller` (create proportional integral controller for a tip-tilt mirror);

dm_pi_controller (create proportional integral controller for a deformable mirror); ttm_pi_update (update the proportional integral controller of a tip-tilt mirror); dm_pi_update (update the proportional integral controller of a deformable mirror); create_ttm_commands (create the command vector of a tip-tilt mirror); create_dm_commands (create the command vector of a deformable mirror); reconstructor (create a wavefront reconstructor for a AO system); zernike_residuals (get Zernike modal residual of the corrected wavefront); zonal_residuals (get zonal residual of the corrected wavefront); atm_mod_lays_transform (transform wavefront by an atmospheric model); etc.

- Other function: znk_mod (create Zernike polynomial); set_znk_cos_coef (set the cosine coefficient of a Zernike polynomial); set_znk_sin_coef (set the sine coefficient of a Zernike polynomial); pixel_array (create a 2D pixel array); set_pix_arr_data (set the data of a 2D pixel array); three_point (create a point in 3D space); three_vector (create a vector in 3D space); three_frame (create the frame in 3D space); three_translation (create a translation in 3D space); three_rotation (create a rotation in 3D space); convert (convert the frame between plane and spherical frame); normal (normalize the wavefront to unit intensity); pip_fft (FFT spacial filter); interpol (interpolate a field); strehl (get the Strehl ratio of a field); etc.

For more information about the Scilab functions or Scicos blocks which we have constructed for our SAO toolbox, please consult the online help documents.

5 Application examples

In this part, we will give two examples to demonstrate how to model and simulate optical problems using the SAO toolbox. The first example is about Young's interferogram, while the second relates to SCAO systems. For more examples, consult the "examples" directory in our SAO release version.

(1) Young's interferogram

Consider such an experiment: suppose a completely coherent plane light, having unit amplitude and 550 nm wavelength, vertically irradiate to a board possessing two

circular apertures, apart from 0.5 mm and both have radius $R=0.25\text{mm}$, we want to observe the interferogram at the distance 30 cm behind the board.

We can easily model such an experiment using the Scilab functions we have designed for SAO. Below is a Scilab script which simulate such a physical process:

```
m=1; cm=1e-2*m; mm=1e-3*m; nm=1e-9*m;
ngrid=256; size=5*mm; lamda=550*nm;
radius=0.25*mm; shift=0.5*mm; distance=30*cm;
init_field = begin(size,lamda,ngrid);
circ_ap_1 = circ_ap(init_field,radius,shift);
circ_ap_2 = circ_ap(init_field,radius,-shift);
mix_field = b_mix(circ_ap_1,circ_ap_2);
dwf_fits(mix_field,"board");
interf_field = forward(mix_field,distance);
dwf_fits(interf_field,"interferogram");
intensity=field_int(interf_field);
x=1:ngrid; y=1:ngrid; grid=[1,1,ngrid,ngrid];
grayplot(x,y,intensity,strf="030",rect=grid);
```

In above script, we have written the results to two FITS files (Flexible Image Transport System, it is the standard data format used in astronomy. Of cause, you can write them into other formats, such as PS or PGM, etc., and we have also provided some other functions to support these formats in SAO): board.fits and interferogram.fits, and at the same time plotted the interferogram directly.

A more intuitional method to model and simulate in SAO is using the optical Scicos blocks which we have designed for SAO. Figure 3 gives out a diagram that can model above Young's experiment. In this diagram, we have packed the board to a super-block for concision, whose content is revealed in Figure 4.

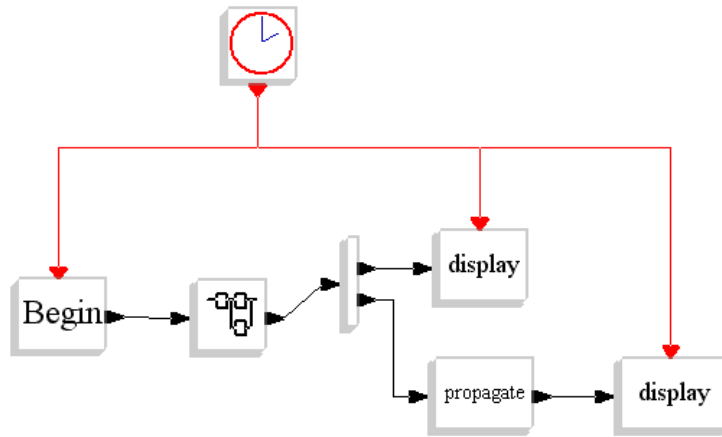


Fig. 3. the diagram to model Young's interfere

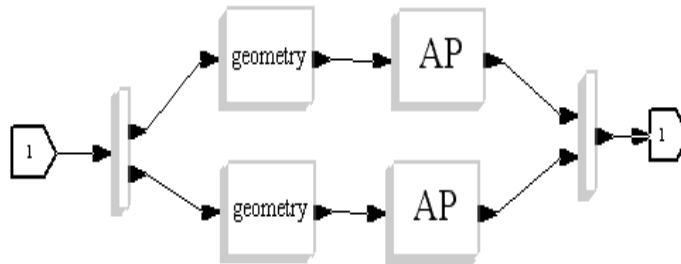


Fig. 4. the contents of the super-block in figure 3

Two methods (using Scilab functions or using Scicos blocks of SAO) is equivalence. We give out the results of simulation in figure 5:

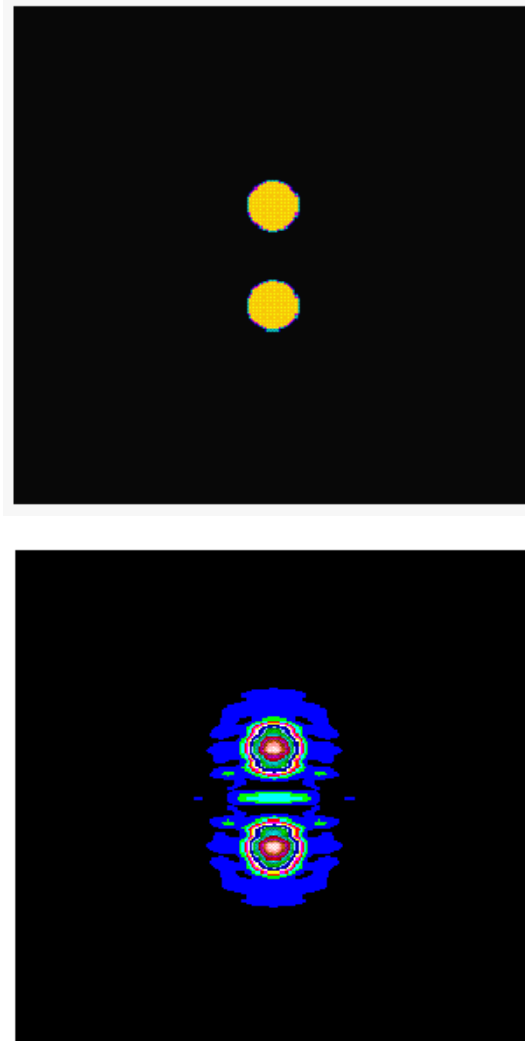


Fig. 5. the left is the field at the behind of the board, and the right is the interferogram.

(2) SCAO system

Analogously, we can model and simulate AO systems by two equivalent methods in SAO. We first give a diagram example in figure 6 to model a simple SCAO system.

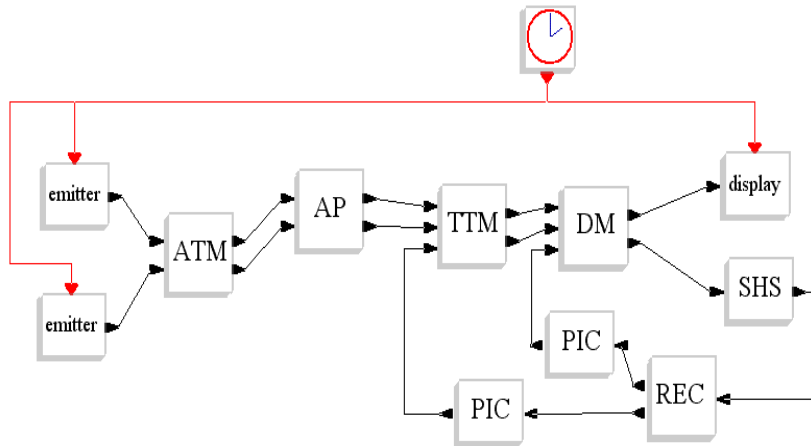


Fig. 6. the diagram to model a SCAO system

The Scilab scripts that model this simple SCAO system is lengthy, so we do not give them out here. The total number of the parameters to simulate such a system is about 70, and we have designed a GUI to simplify the input process. The GUI is plot at Figure 7:

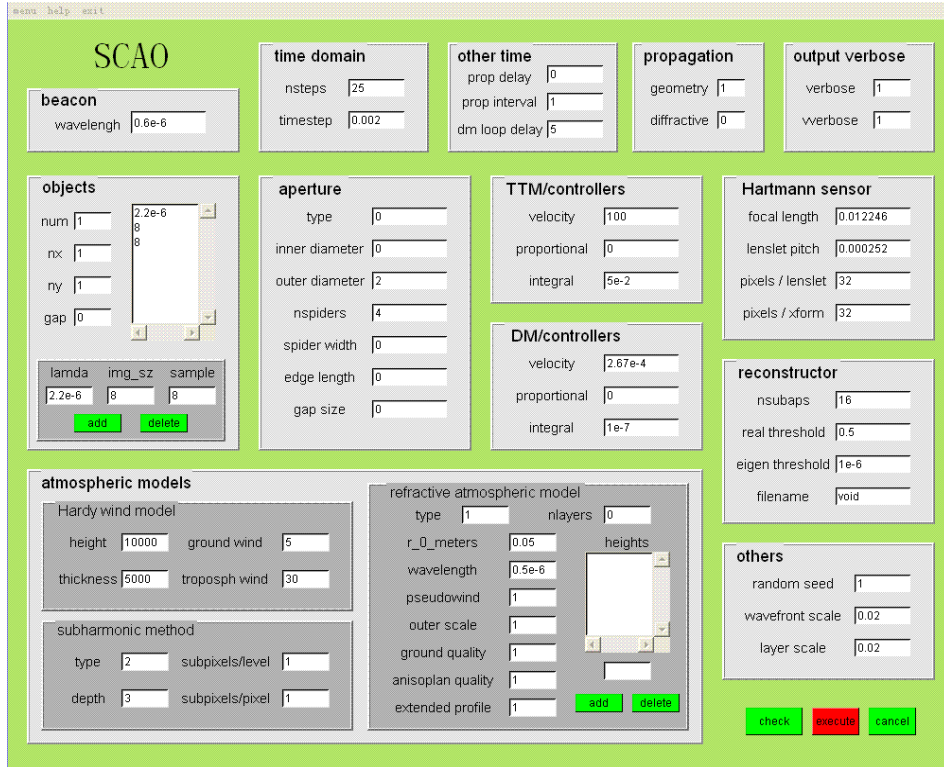


Fig. 7. the GUI to model a SCAO system

In Figure 8 we give out two figures to demonstrate the process of adaptive correction:

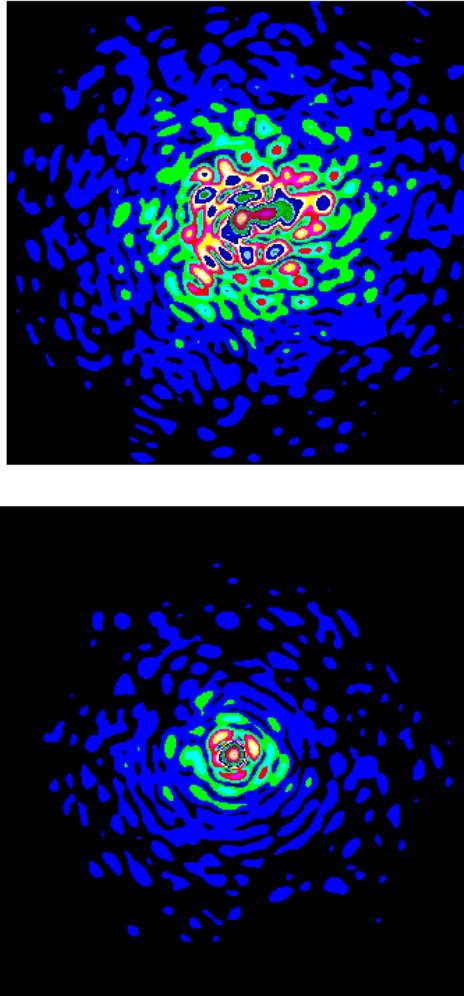


Fig. 8. the left is an uncorrected image and the right is a corrected image.

6 Conclusion, future extensions and plans

The objective that we select the Scilab/Scicos environment and design SAO toolbox for it is to make an optical simulation much more easier and user-friendly than original FORTRAN, or C/C++ codes, without sacrificing fidelity or efficiency. We also expect other researchers in the field of modeling and simulation of adaptive optics to use, correct, and improve this toolbox, so we release it under GPL license.

We believe we have met our original objective. Users now can build models from scratch, or simply copy an existing model, then modify it to suit their needs. More advanced users can create their own components at the top of our fundamental C/C++ routines or our optical Scilab functions, which can then be used and reused just like those in our Optics library or other Scilab library.

At the same time, there remains considerable room for improvement. Now SAO is only at its infancy and is very much under development. To model and simulate more complex application problems in wave or adaptive optics, we need construct more other software modules. Other than, the documentation is never up to date; The component libraries is far from complete, and many of component interfaces could be improved; The GUI is getting better and better, but there is always more to do; And so on. We are hard at work on all these things.

References

- [1] Hardy, *Adaptive Optics for Astronomical Telescopes*, Oxford Press, 1998
- [2] <http://www.tosc.com/software/software.html/>
- [3] <http://www.scilab.org/> and <http://www.scicos.org/>
- [4] <http://www.mza.com/>
- [5] <http://www.arcetri.astro.it/caos/>
- [6] <http://www.okotech.com/software/lightpipes/>
- [7] <http://eraserhead.caltech.edu/>

SCILAB TOOLS FOR IDENTIFICATION AND SIMULATION OF MECHANICAL STRUCTURES

Maurice Goursat and Laurent Mevel
INRIA, Domaine de Voluceau
Rocquencourt BP 105 78153 Le Chesnay Cedex, France
Email: maurice.goursat@inria.fr
INRIA-IRISA, Campus de Beaulieu, 35042 Rennes Cedex, France
Email: laurent.mével@irisa.fr

No Institute Given

KEYWORDS: modal analysis, covariance-driven, subspace methods, output-only, Scilab

1 ABSTRACT

In this paper we present a toolbox COSMAD and some other Scilab functionalities for identification of modal parameters of mechanical structures and simulation of specific cases for the development and adaptation of new algorithms. Modal analysis is a process needing the expertise of mechanical engineers for choosing the best version of algorithms, the tuning of parameters... The Scilab environment with a specific GUI is a tool of paramount interest for the quality of results and to have a precise evaluation of the reliability of the estimated parameters.

2 INTRODUCTION

Identification of modal parameters of mechanical structures is a standard process for design of cars, bridges, aircrafts... Forced excitations and input-output correlations are classical techniques.

There exists many situations where controlled excitation is not a feasible way: this is the case for in-operation structures such as rotating machines for electricity, space launchers during the flight, bridge with traffic (see [1]). During real operation, the conditions may be completely different from the ones used in laboratory simulations. So there is an increasing need for techniques adapted to these real cases.

Many examples have shown that the output-only modal analysis of complex structures give results with a quality similar to the case of observed and controlled input.

We present here a comprehensive identification and detection Scilab toolbox. We briefly recall the main theoretical results and the foundations of the covariance based subspace identification and detection methods. We present some other Scilab tools

sometimes not directly used in the final identification and diagnosis process but necessary for the development of new algorithms for specific problems.

Then, we present the main functions of the toolbox illustrated them with some dialog windows and results.

3 IDENTIFICATION PROCEDURE

3.1 Modeling

The mechanical system is assumed to be a time-invariant linear dynamical system:

$$\begin{cases} M\ddot{\mathcal{Z}}(t) + C\dot{\mathcal{Z}}(t) + K\mathcal{Z}(t) = v(t) \\ Y(t) = L\mathcal{Z}(t) \end{cases}$$

\mathcal{Z} : displacements;

M, C, K : mass, damping, stiffness matrices;

t : continuous time;

v : excitation (external non measured forces) modeled as a non-stationary white noise;

L : observation matrix (Y corresponds to the locations of the sensors on the structure).

The modal characteristics are the vibration modes or eigen-frequencies μ and the modal shapes or eigenvectors Ψ_μ solution of:

$$(M\mu^2 + C\mu + K)\Psi_\mu = 0, \quad \Psi_\mu = L\Psi_\mu$$

By stacking \mathcal{Z} and $\dot{\mathcal{Z}}$ and sampling at rate $1/\delta$ we get the following equivalent state-space model:

$$\begin{cases} X_{k+1} = FX_k + V_k \\ Y_k = HX_k \end{cases}$$

$F = \exp(A\delta)$ and $H = [L \ 0]$ with

$$A = \begin{bmatrix} 0 & I \\ -M^{-1}K & -M^{-1}C \end{bmatrix}$$

The input noise is zero-mean and its covariance is:

$$Q_V(k) = \int_{k\delta}^{(k+1)\delta} \exp(As)\tilde{Q}(s)\exp(A^T s)ds,$$

with

$$\tilde{Q}(s) = \begin{bmatrix} 0 & 0 \\ 0 & M^{-1}Q_v(s)M^{-T} \end{bmatrix},$$

and $Q_v(s)$ is the covariance matrix of v . The dimensions of the state X and the observations Y are respectively $2m$ and r with r much smaller than $2m$ (e.g. 20 sensors and 100000 degrees of freedom).

2. Compute the following SVD

$$\hat{\mathcal{H}}_{p-1,q} = (U_1 U_2) \begin{pmatrix} S_1 & 0 \\ 0 & S_2 \end{pmatrix} V^T$$

with the m dominant singular values in S_1 .

3. Get the estimate of the observability matrix with:

$$\hat{\mathcal{O}}_{p-1}(\hat{H}, \hat{F}) = U_1 S_1^{1/2} T$$

4. Extract the transition matrix from $\hat{\mathcal{O}}_{p-1}$ for increasing orders and get the stabilization diagram for the successive redundant models: separate the frequencies common to many models from the spurious ones.

4 IDENTIFICATION WITH THE COSMAD TOOLBOX

The features of the toolbox have been developed and adapted in order to solve problems for different domains. An example is the COST benchmark described in [5]. We summarize now the steps of the processing.

4.1 Data preprocessing

The figure 1 shows the Scilab window with the on-line help during the selection of sensors, possible filtering, and choice of the part of the signal to be processed.

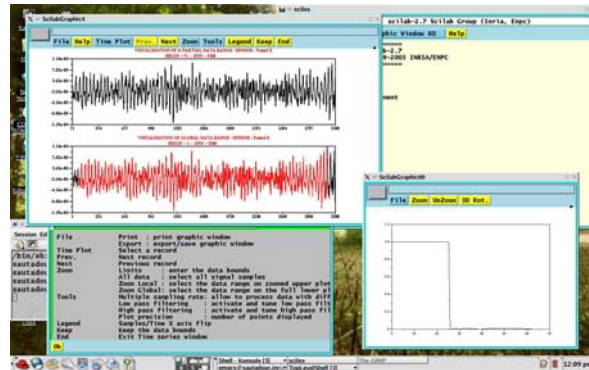


Fig. 1. Basic signal processing and selection

4.2 Identification with COSMAD

Computational steps of the subspace method

1. Data manipulation and preprocessing. The first step is the creation of a “project” to define the file locations, parameters for data and algorithm. The best results for a global analysis are obtained with a small number of sensors: 2 to 10. It can be useful to start with a single sensor data in order to get sensor pools giving a good observation for the same modal characteristics. Fig. 1 shows the screen for the selection of the part of the record used for the identification process (with the help window).
2. For the classical subspace case algorithm the steps are : (a) compute the empirical estimates of the covariance matrices, (b) build the Hankel matrix, (c) compute the observability matrix and (d) compute the transition matrices (and eigen-structures) for increasing orders. By this way we obtain a collection of signatures (modal parameters).

Visual exploitation of the results The final selection is done by a visual exploitation. An automatic selection procedure has been implemented for monitoring, or as a first step for subsequent visual improvements. Monitoring is mandatory for long sequences and time-varying systems.

1. Stabilization diagram. All the obtained frequencies are plotted on an “histogram” referred to as the *stabilization diagram*. The “true” eigenfrequencies are assumed to be present throughout the different model orders, whereas spurious “computational modes” are not. The “true” modes are given by the “consistent” alignments of the diagram.
2. Local examination. For every frequency selected with the previous procedure, we plot w.r.t. the model order the result of the identification i.e. the triple {eigenfrequency, damping coefficient, MAC factor}. Visual zooming can be performed to help for properly selecting and matching close, coupled, eigenfrequencies.

Throughout the process, mode shapes can be shown and animated.

Fig. 2 shows the dialog window for a detailed inspection of the results of a global automatic identification procedure and Fig. 3 the correlation of mode shapes.

4.3 Other Tools

FRFs as data In many industrial cases the recorded data are simply condensed in the FRF (as shown on Fig. 4).

These FRFs are, in the simplest way, used for manual peak peaking. It is possible to use directly the FRFs in our identification procedure. It is also interesting to build the impulse responses and to use these responses to easily compute the correlations and compute the Hankel matrix. By this procedure we have a more precise identification, we get the damping parameters and we can construct the mode shapes. The Fig. 5 shows the impulse response corresponding to the previous FRF.

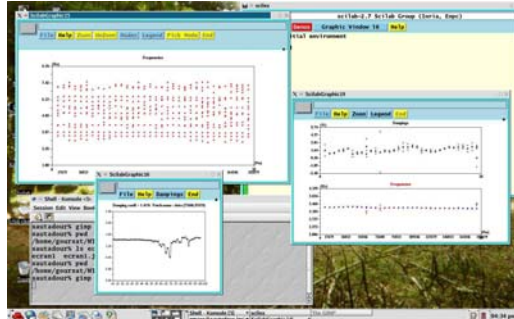


Fig. 2. Monitoring results examination

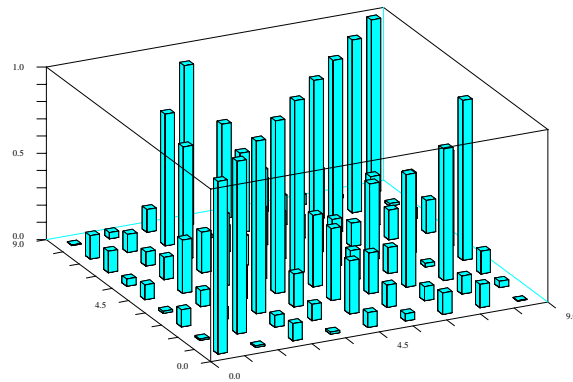


Fig. 3. MAC values for the mode-shapes of the identified modes

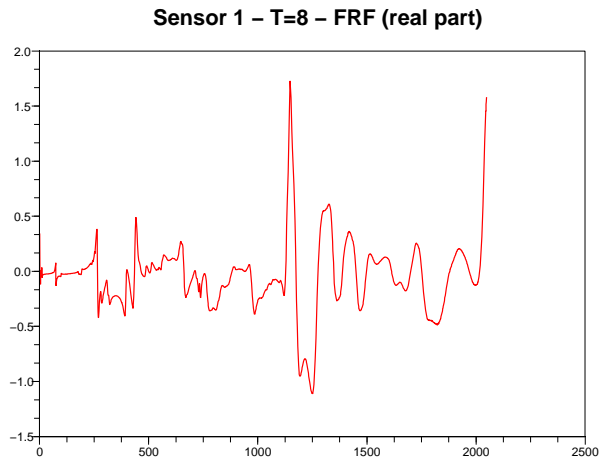


Fig. 4. An example of FRF (real part - sensor 1).

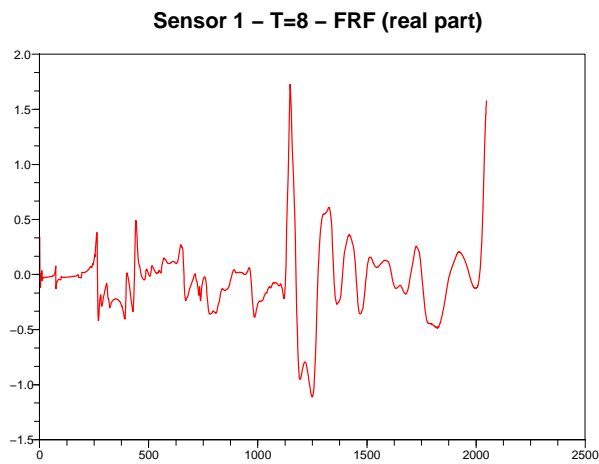


Fig. 5. Impulse response (sensor 1)

Multi-patch data (see [6]) . This is the case of non simultaneously measured data for input-output analysis with moving the sensors but keeping at least one fixed reference sensor. In this case we need a suitable renormalization of the covariance matrices to

accommodate for possible changes in the excitation (see [7] and [8]).

Monitoring for time-varying modal characteristics (see [9]) Both increasing window with forgetting factor, and sliding window modes can be applied for on-line identification (with automatic mode extraction).

Fig. 6 shows an example with the evolution of the parameters (with a local peak in of the damping corresponding to a shock to be avoided).

Fig. 7 illustrates one of the capabilities of the toolbox for choosing the best parameters: example of the effect of the size of the Hankel matrix on the identification (zoom on every point).

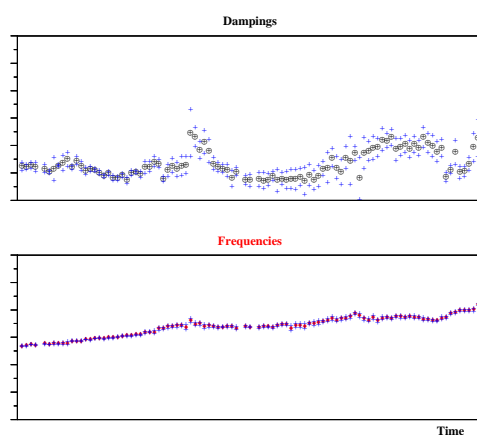


Fig. 6. Evolution of a frequency and the corresponding damping

Animation of mode shapes When the geometry of the structure is available, it is interesting to animate the mode shapes when we have many close poles (large civil aircrafts are typical examples of such a situation).

We give here a simple example with the COST F3 steel-quake. The sensor locations are given by Fig. 8. Animation of one of the modes is shown on Fig. 9.

Simulation of time varying systems Many systems are time varying (aircrafts during flights, space launchers, bridges...) and we want to design algorithms for the monitoring of such structures and adapted tests to detect failures or dangerous phenomena

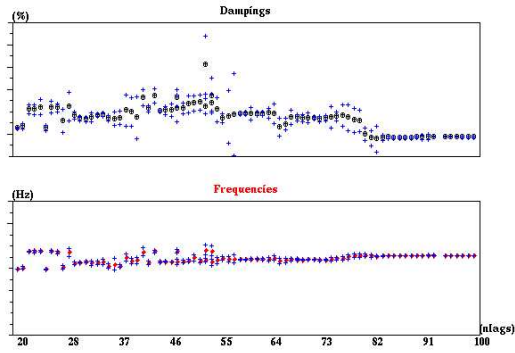


Fig. 7. Mode evolution as a function of the number of computed autocorrelations

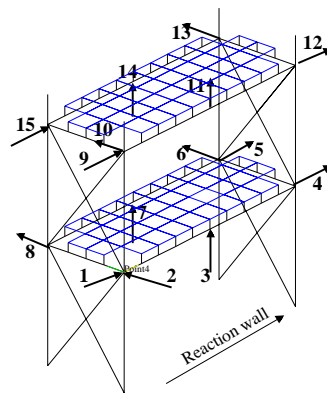


Fig. 8. Steel-quake. Location of the sensors.

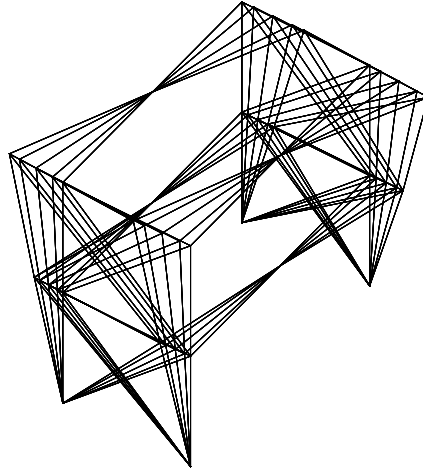


Fig. 9. Animation of the first torsion mode.

(e.g. flutter). We need data to develop the algorithms. We use a simulator with the state-space model described in the modal basis.

The identification phase gives us the modes of the structure: the eigenvalues λ and the modal shapes ϕ_λ .

Our data are collected by the simulation of the system:

$$\begin{cases} X_{k+1} = F(\lambda, k) X_k + V_{k+1} \\ Y_k = H(\phi) X_k + W_k \end{cases} \quad (1)$$

where the system matrix F is bloc-diagonal of the identified eigenvalues λ , the observation matrix H is composed by the identified modal shapes, V and W are white noises. We supposed that the modal shapes are stable during the process and the simulation of the variation of frequencies or damping parameters is simply achieved by time varying simulation of the corresponding elements of the diagonal matrix F . The identified eigenvalues and the eigenvectors are not real. We can operate a simulation with the complex values or after an appropriate rotation of the eigenvectors get “near real” values and perform a simpler simulation.

5 CONCLUSIONS

We have presented the capabilities of COSMAD and some other Scilab tools for modal analysis of mechanical structures and the key role of the and interactivity for the reliability of results and the choices of different algorithms. COSMAD is freely available on the Scilab website (www.scilab.org) and on www.irisa.fr/sisthem for current version.

References

- [1] L. Hermans and H. Van der Auweraer, "Modal Testing and Analysis of Structures under Operational Conditions: Industrial Applications" *Mechanical Systems and Signal Processing*, vol. 2, 1999 (Special issue on modal testing).
- [2] A. Benveniste and J.-J. Fuchs, "Single Sample Modal Identification of a Non-stationary Stochastic Process" *IEEE Trans. Aut. Cont.*, vol. AC-30, pp. 66-74, 1985
- [3] P. Van Overschee and B. De Moor, "Subspace Algorithms for the Identification of Combined Deterministic-stochastic Systems" *Automatica*, vol. 30, no 1, pp. 75-93, 1994
- [4] M. Abdelghani, M. Goursat and T. Biolchini, "Online Monitoring of Aircraft Structures under Unknown Excitation" *Mechanical Systems and Signal Processing*, 1999.
- [5] F.J. Molina, R. Pascual and J.-C. Golinval, "Description of the Steelquake Benchmark", *Mechanical Systems and Signal Processing*, vol. 17(1), 2003.
- [6] A. Guyader and L. Mevel, "Covariance driven subspace methods: input/output vs output only" *Proc. IMAC XXI, Kissimmee, USA*, 2003.
- [7] L. Mevel, M. Basseville, A. Benveniste and M. Goursat, "Merging sensor data from multiple measurement setups for non-stationary subspace-based modal analysis", *Journal Sound and Vibration*, vol. 249, 2002.
- [8] L. Mevel, M. Goursat and M. Basseville, "Stochastic subspace-based structural identification and damage detection - Application to steel-quake benchmark" *Mechanical Systems and Signal Processing*, vol. 17, 2003 (Special Issue on COST F3 benchmarks).
- [9] A. Sam, L. Mevel and M. Goursat, "Blind Modal Identification for Large Aircrafts - The Case of High Number of Close Poles" *Proc. IMAC XXII, Dearborn, USA*, 2004.

Simulation of cars waiting at a toll booth using Scilab

Jianfeng Li, Lifang Zhou

Institute of industrial process control, Zhejiang University,
Hangzhou, China, 310027, {jfli, lfzhou}@iipc.zju.edu.cn

Abstract. Based on the basic theories and methods of queuing theory, a cars waiting at a toll booth was developed applying the tools of scientific computational software SCILAB. In order to ensure the velocity of cars passing through the toll booth, the number of servers should be calculated during the stage of the system designing. After analyzing the arrival status of the cars flow, the probability of the vehicle reaching the toll booth was calculated based on queuing theory and the optimal number of server is obtained. The process of cars waiting in a toll booth is simulated in the form of animation and the gliding process can be observed.

Keywords: Queuing Theory, SCILAB, Toll Booth

1. Introduction

Queuing theory is the study of the waiting line phenomena. It is a branch of applied mathematics utilizing concepts from the field of stochastic processes. It is a system that customers require service, and some restrictions exist on the service. The service times are random variable. If the number of servers is limited, some customers will have to wait. Queuing phenomenon encompasses all activities of our lives. We have to wait in line whenever the number of servers or the service rate of the server does not match the rate at which the customers arrive in the queue. Queuing theory is the formal analysis of this phenomenon in search of finding the optimum solution to this problem so that everybody gets service without waiting for a long time in line.

The queue system is composed of the arrival process, the queue discipline and the service process. If there is one server in all service process, it is called single-server system. If the server number is more than one and each server can serve customers independently, it is called multi-server system. Usually, the service system is

evaluated by performance indexes, such as service intensity, average queuing length, average queuing time, average stay time, probability of the service desks being idle, and the probability of the customers having to wait.

The basic service process is following: the customers arrive at a queuing system. If the servers are busy, they join the queue in a waiting line; otherwise, they are served according to the prescribed queuing discipline. When the service is finished, they leave the system. The terms customer, server, service time, and waiting time may acquire different meanings in different applications. In this article, customers are cars arriving at a toll booth, the servers are the workers in the toll booth, and the service times are the payment times.

2. The queue system's Performance index

In the simulation of cars waiting at a toll booth, some necessary condition assumptions and notation definitions should be made.

Assumptions:

The arrival time and service time follow exponential distribution.

The waiting queue is infinite.

Notations:

λ : the arrival rate of the car

μ : the service rate of the car

$1/\lambda$: the average interval of two cars arrival the toll booth

$1/\mu$: the average service time of each car

ρ : service intensity

L_s : the average quantity of cars in the system

L_q : the average quantity of cars in the waiting queue

W_s : the average time of each car stay in the system

W_q : the average time of each car stay in the waiting queue

Here, according to John DC Little's formula, we only discuss two usual models.

1. Standard M/M/1 service model.

The arrival time and service time of cars can be discretional distribution. The number of server is one, car source and service space are infinite, and the queuing discipline is first come first service. So the major performance indexes are:

Service intensity: $\rho = \lambda/\mu$;

Probability of the server being idle: $P_0 = 1 - \rho$;

Probability of n cars in the system: $P_n = \rho^n (1 - \rho)$

$$L_s = \frac{\lambda}{\mu - \lambda}; \quad L_q = \frac{\rho\lambda}{\mu - \lambda}; \quad W_s = \frac{1}{\mu - \lambda}; \quad W_q = \frac{\rho}{\mu - \lambda}$$

2. Standard M/M/C service model.

Except the number of server is C and they are parallel connection, the other terms are the same as M/M/1 service model. Servers work independently and the average service rate is the same. So the major performance indexes are:

Service intensity: $\rho = \lambda/c\mu$;

Probability of the service desks being idle:

$$P_0 = \left[\sum_{k=0}^{c-1} \frac{1}{k!} \left(\frac{\lambda}{\mu}\right)^k + \frac{1}{C!} \frac{1}{1-\rho} \left(\frac{\lambda}{\mu}\right)^c \right]^{-1};$$

The average quantity of cars in the waiting queue: $L_q = \frac{\rho(c\rho)^c P_0}{C!(1-\rho)^2}$

The average quantity of cars in the system: $L_s = L_q + c\rho$

The average time of each car stay in the system: $W_s = W_q + \frac{1}{\mu}$

Probability of the car has to waiting after arriving:

$$P(n \geq c) = \sum_{n=c}^{\infty} \frac{1}{c!c^{n-c}} \left(\frac{\lambda}{\mu}\right)^n P_0$$

3. The optimal number of servers

According to queuing theory, the probability of no cars in single queuing multi-server system is:

$$P(0) = \frac{1}{\sum_{k=0}^{N-1} \frac{\rho^k}{k!} + \frac{\rho^N}{N!(1-\rho/N)}}$$

N is the number of server that can provide service in the queue system. According to the data we measured, the minimum servers number can be calculated by $N_{\min} = \lceil \lambda/\mu \rceil$, then the performance indexes of queue system can be obtained. If

they didn't satisfy the needs, then let $N = N_{\min} + 1$, compute the performance indexes again, until the satisfied solution is found.

After the above analysis, the algorithm approach how to find the optimal number of servers in a toll booth is summarized as following:

Step 1, according to $N_{\min} = \lceil \lambda / \mu \rceil$, find out the initialization number of servers

$$N = N_{\min} ;$$

Step 2, compute the performance indexes, such as the average quantity of cars in the waiting queue, the average time of each car stay in the system, and the average time of each car stay in the waiting queue. Comparing them with desired indexes, if it is dissatisfied, go to step 3; otherwise, the optimal number of servers is N and the process is end.

Step 3, let $N = N + 1$, go to step 2.

4. Simulation of cars waiting at a toll booth using scilab

In this system, the arrival time and service time of car are similar to the exponential distribution. Every car's arrival, accepting service and leaving after service are the events that have the system state changing. Usually, assume that the left time of the i th car is the begin service time of the $(i+1)$ th car. So, all the events can be classified to two sorts: one is the arrival event of cars; the other is the leave event of car. When the arrival event and the leave event occur at a same time, disposed the former first.

In Scilab simulation environment, the average arrival rate of cars, the average service rate of cars and the simulation terminate time are given by the fig 1.

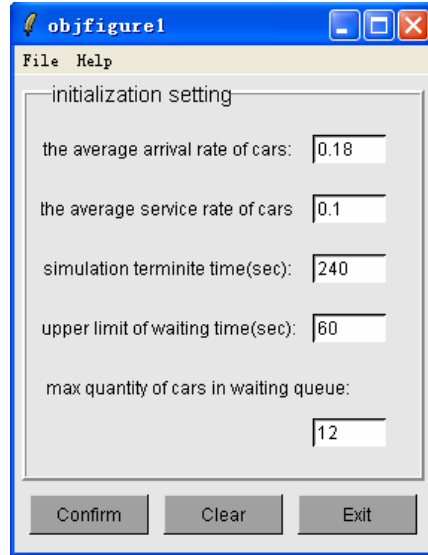


fig 1. The initialization setting

In the simulation platform we present, the draw mode of SCILAB is set to be animate all along so that a gliding process can be observed. The animation demo can be described as fig 2 and fig 3.

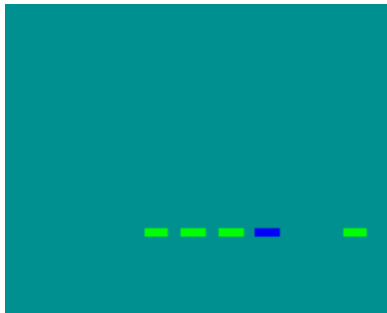


fig 2. Single queuing single server

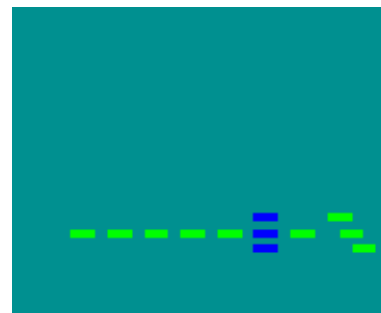


fig 3. Single queuing multi-server

Table 1 expresses the process of single queuing single server simulation. Assume that the terminate time of simulation is 240 sec, the average arrival rate is 0.18 and the average service rate is 0.1. The minimum servers' number can be found out:

$$N_{\min} = \lceil \lambda / \mu \rceil = 1. \text{ Cars' arrival event is 1 type event; cars' leave event is 2 type event.}$$

The queuing discipline is first come first serve.

Table 1. the simulation table of cars waiting at a tool booth

simulation clock	car	arrival time	queue length	cars number in system	server begin time	waiting time	service time	leave time	stay time	served cars number
0	—	—	—	—	—	—	—	—	—	0
10	1	10	0	1	10	0	4	14	4	0
12	2	12	1	2	14	2	4	18	6	0
14	1	—	0	1	—	—	—	14	—	1
16	3	16	1	2	18	2	6	24	8	1
18	2	—	0	1	—	—	—	18	—	2
27	3	27	0	1	27	0	8	35	8	2
29	4	29	1	2	35	6	2	37	8	2
35	3	—	0	1	—	—	—	35	—	3
36	5	36	1	2	37	1	14	51	15	3
37	4	—	0	1	—	—	—	37	—	4
37	6	37	1	2	51	14	15	66	29	4
39	7	39	2	3	66	27	1	67	28	4
:	:	:	:	:	:	:	:	:	:	:
233	22	—	22	23	—	—	—	233	—	22
233	46	233	23	24	533	300	7	540	307	22
234	47	234	24	25	540	306	11	551	317	22
237	48	237	25	26	551	314	10	561	324	22
240	49	240	26	27	561	221	4	565	325	22
245	—	—	—	—	—	—	—	—	—	—

The simulation processes can be described as follows: simulation begin, simulation clock is set to zero, queue length, cars number in system and served cars number are all set to zero. The first event is the first car arrival event, arrival time is 10sec, and the next car arrival time (=the current car arrival time + arrival interval) is 12sec. Because the sever state is idle, the first car receive service immediately, the number of car in the system is 1, the car's waiting time is 0. Service time is 4sec, service finish time (=arrival time + waiting time + service time) is 14sec, the car's stay time in the system (=waiting time + service time) is 8sec. compare the next car arrival time(12sec) with leave time (14sec), we will find that the next earlier event is 1 type event. The simulation clock is pushed to 12sec, the second car arrival. The third car arrival time is 16sec. Because the first car is still in the system, the second car has to waiting, the length of waiting queue is 1, the number of car in system is 2. Because the second car only can be serviced after the first one have left, the second car's

beginning service time is equal to the first car's leave time, we can calculate that the second car's waiting time (=service begin time – arrival time) is 2sec. service time is 4sec, then leave time is 18sec, stay time is 6sec. Compare the third car's arrival time (16sec) with the first car's leave time (14sec), it can be found that the next earlier event is 2 type event. The simulation clock is pushed to 14sec, the first car leaves the toll booth, the leave time is 14sec, and the number of served car is 1. At the same time, the second car receives service, so the server is still busy, the length of queue and the number of car in the system minus one respectively. Then go on comparing the third car's arrival time (16sec) with the second car's leave time (18sec), it can be found that the next earlier event is still 1 type event.....go along like this continually, till the simulation clock (245sec) is more than terminate time (240).

The simulation results can be calculated:

The average quantity of cars in the waiting queue is: $Lq = \frac{1}{T} \sum_{i=1}^m q_i (T_i - T_{i-1}) = 14.7 \approx 15$

The max quantity of cars in the waiting queue is: 26

The average quantity of cars in the system is: $\sum_{i=1}^m SN_i (T_i - T_{i-1}) / T = 15.78 \approx 16$

The average time of each car stay in the waiting queue is: 146sec

The average time of each car stay in the system is: 162sec

All the served cars number is: 22

Let $N = N_{\min} + 1$, the number of server become 2. The simulation process is similar to the above single servers systems. The different is that, in the simulation, when the second car arrival, it can be serviced immediately because there are two servers in the system. So the car will pass the toll booth with less waiting time. The simulation results with two servers can be calculated:

The average quantity of cars in the waiting queue is: 3

The max quantity of cars in the waiting queue is: 6

The average quantity of cars in the system is: 4

The average time of each car stay in the waiting queue is: 29sec

The average time of each car stay in the system is: 41sec

All the served cars number is: 42

Comparing the above results with desired performance index, it is found that when there are two servers, the demands can be satisfied. So, it is recommended to open two servers.

5. Conclusion

Queuing theory have been used in almost every domain of social and natural science as a tool to solve many problems. In this article, queuing theory is used to solve the problem of cars waiting in a tool booth. Using the tools of Scilab, the simulation of a single queuing multi-servers system is done and the optimal number of server is obtained.

SCILAB is suitable for the demonstration and simulation on cars waiting in a tool booth for its advantages in matrix computing and graphic/animation display. The simulation process of cars waiting in a tool booth system in Scilab environment can be observed expediently and visually.

Acknowledgments. This work is supported by a grant from the National Natural Science Foundation of China (No. 60474064) and the Fund of produce-study-research Science and Technology Cooperation of Hangzhou City to Zhou LiFang.

References

1. Paul E.Joustra and Nico M.Van Dijk, simulation of check-in at airports, *Proceedings of the 2001 Winter Simulation Conference*, 2001.
2. Longhua Ma and Weiguang Chen, Building Cellular Automata Simulation Platform With Scilab, *Proceeding of the 2005 International Workshop on SCILAB and Open Source Software Engineering*. Wuhan, China. 2005.
3. Jun Liang and Yong Zhao. Systems Engineering Introduction. Chemical Industry Press. 2005.
4. Zhou Zai-ling, Shi Hong-wen, Wu Meng, Chang Jun-qian, Application of Queuing Theory to calculate number of BRT parking space, *Railwang Computer Application*. 2006, 15(2)

Sustainable Cooktop Design Using SCILAB

T.T. Wong, C.W. Leung, D.Y. Liu & H.B. Li

Department of Mechanical Engineering,
The Hong Kong Polytechnic University, Hong Kong
E-mail: mmtwong@polyu.edu.hk

Abstract. Ensuring a better quality of life for everyone, now and for future generations, lies at the heart of every responsible Government's sustainable development strategy. There are two major considerations in designing domestic cooktop burners, namely, energy utilization and indoor air pollution. Researches conducted to optimize the thermal performance of cooktop burners are rather limited. Considering the large number of cooktop burners being used in the world, a slight improvement in thermal performance resulting from a better design will certainly lead to a significant impact on the domestic and commercial energy consumption. In view of the need to optimize thermal performance and to minimize indoor air pollution, an attempt has been made to apply a combination of analytical techniques, viz. design of experiments, multiple regression and optimization, via the Scilab. Confirmatory experiments have been carried out to verify the validity of Scilab findings. The methodology used suggests a feasible approach to the design of a fuel-efficient and environmental-friendly cooktop burner.

Keywords: cooktop burner; thermal efficiency; pollutant emissions; Scilab.

1 Introduction

Most homes have more than one source of indoor air pollution. For example, pollutants come from tobacco smoke, building materials, decorating products, home furnishings, and activities such as cooking, heating, cooling, and cleaning. Living in areas with high outdoor levels of pollutants usually results in high indoor levels. Combustion pollutants are one major category of indoor air pollutants.

Combustion pollutants are gases or particles that come from burning materials. Common fuels burned in these appliances are natural or LP gas, fuel oil, kerosene, wood, or coal. Some of the common pollutants produced from burning these fuels are carbon monoxide, nitrogen dioxide, particles, and sulfur dioxide. Other pollutants that can be produced by some appliances are unburned hydrocarbons and aldehydes. The health effects of combustion pollutants range from headaches and breathing difficulties to death. The health effects may show up immediately after exposure or occur after being exposed to the pollutants for a long time. Further studies to better define the release of pollutants from combustion appliances and their health effects are needed.

Each year, according to US Consumer Product Safety Committee, there are more than 200 carbon monoxide deaths related to the use of all types of combustion appliances in the home. (CPSC, 2004). Exposure to carbon monoxide reduces the blood's ability to carry oxygen. Often a person or an entire family may not recognize that carbon monoxide is poisoning them. The chemical is odorless and some of the symptoms are similar to common illnesses. This is particularly dangerous because carbon monoxide's deadly effects will not be recognized until it is too late to take action against them. Carbon monoxide exposures especially affect unborn babies, infants, and people with anemia or a history of heart disease. Breathing low levels of the chemical can cause fatigue and increase chest pain in people with chronic heart disease. Breathing higher levels of carbon monoxide causes symptoms such as headaches, dizziness, and weakness in healthy people. Carbon monoxide also causes sleepiness, nausea, vomiting, confusion, and disorientation. At very high levels it causes loss of consciousness and death.

Breathing high levels of nitrogen dioxide causes irritation of the respiratory tract and causes shortness of breath. Compared to healthy people, children, and individuals with respiratory illnesses such as asthma, may be more susceptible to the effects of nitrogen dioxide. Some studies have shown that children may have more colds and flu when exposed to low levels of nitrogen dioxide. When people with asthma inhale low levels of nitrogen dioxide while exercising, their lung airways can narrow and react more to inhaled materials.

Domestic gas-fired cooktop burners operating with low-pressure and low-Reynolds-numbers have been extensively used. Since most of the cooktop burners usually have no flue to vent combustion particles and gases directly to outside, they should be carefully designed and properly used. At the same time, thermal efficiency is one of the important requirements of domestic cooking. Slight improvements in thermal performance resulting from a better design will certainly lead to a significant impact on the decreasing world fossil fuel resource. Hence research on simultaneously improving cooktop burner efficiency and reducing pollutant emissions of cooktop burners can be very useful to the society [7].

Hou and Ko [3] examined the influences of nozzle-to-plate distance on thermal efficiency for low-Reynolds-number and fuel-rich flames. The results showed the following phenomena: (1) With an increasing nozzle-to-plate distance, the thermal efficiency first increases to a maximum value, which occurs when the impingement plate is at a location slightly lower than the tip of the inner rich premixed flame, and then it decreases. Stubington et al. [10] used the factorial experimental design method to investigate the thermal efficiency and emissions produced by each of the three cooktop burners under consideration. It was found that the emissions and thermal efficiency were significantly affected by either the thermal input, or the nozzle-to-plate distance to flame-length ratio. In general, an increase in either one of them reduces the thermal efficiency.

Pollutant emissions generated from premixed impinging flame jets have also been investigated. Mishra [8] reported the result of emissions from impinging premixed flames. It was found that the level of an indirect greenhouse gas, carbon monoxide (CO), increases with the Reynolds number for the nozzle-to-plate distance, $H/d > 12$. However, the CO level also increases with the equivalence ratio at the same nozzle-to-plate distance. Ko et al. [5] assessed the effect of changes in gas composition on

burner performance. The significant parameters reported include gas composition, primary aeration, gas flow rate, gas supply pressure and the nozzle-to-plate distance. It was found that burner using natural gas with a high heating value results in a decrease in thermal efficiency and an increase in CO emissions. On the contrary, for a burner using natural gas with a low heating value, the results are reversed. But the problem can be solved by several means, such as decreasing the gas pressure, by enlarging the primary aeration, by choosing a proper thermal input or by adjusting the nozzle-to-plate distance.

It can be seen that while there are studies performed to analyze the effects of the design parameters on cooktop burner thermal efficiency and pollutant emissions. There is very little work done to optimize these important parameters. Therefore, the present experimental work was aimed to fill this gap. In the present study, the effects of four design parameters were investigated: Reynolds number, equivalence ratio, nozzle-to-plate distance and jet-to-jet spacing distance. Since these four parameters might be interrelated, the Box-Behnken Design (BBD) was employed to design the experiment [9] and multiple regression was used to model the experimental data. Its application and comparison with response surface methodology (RSM) as an empirical modeling tool has already been reported by many in the field of process and product design [12]. Then the regression model was used to characterize the burner performance within the experimental region and considered as objective functions for optimization.

The free and open scientific environment: Scilab (INRIA, ENPC), which provides many scientific functions and tools, can be very usable and valuable in engineering and scientific applications. In the present work, from modeling and statistics, to visualization, to optimization, Scilab and its functions are used to complete the process of characterizing and improving performance of domestic gas cookers. It is found that Scilab not only can be competent for scientific analysis, but also can accelerate the process.

2 Experimental Procedure

The experimental setup consists of a heat generation system and a heat absorption system, as shown schematically in Fig. 1. In the heat generation system, the domestic gas-fired burner fabricated from cast iron is a ring burner with 48 circular holes of a 3mm inner diameter each. During the experiments, only 10 holes were used and the other 38 holes were blocked in order to ensure the “operating holes” were spaced uniformly round the chosen ring. LP gas was used as fuel in the experiment. The composition of the LPG is 70% butane (C_4H_{10}) and 30% propane (C_3H_8) by volume, with a low heating value of 117 MJ/m^3 approximately. Metered compressed air and LP gas were premixed in a stainless steel chamber and then delivered to the burner via a 2500 mm long high-pressure pipe with a 9.5 mm inner diameter. The premixed chamber was fully filled with very small stainless steel wire to ensure a uniform mix of the air and the LP gas. The air/LPG mixture was then supplied to the burner and ignited at the ports. The burner was fixed on a three-dimensional position controller

so that the flame can be moved to any desired position relative to the bottom surface of the loading vessel.

For the heat absorption system, a loading vessel with an outer diameter of 250 mm containing water at a temperature below 30 °C was fixed on a metallic frame. There were four uniformly spaced thermo-junctions used to measure the water temperature. These thermo-junctions were prepared according to the experience obtained from previous studies [14]. They were placed with equal distance at four points on the same ring with a diameter of 125 mm, and were positioned approximately at the mid-point of the water depth. Whenever the difference in temperature readings was found to exceed 0.2 °C, a glass stirrer was used to stir the heated water to ensure uniformity of the water temperature. Heat loss due to the stirring was assumed negligible and therefore was not considered in the thermal efficiency calculations.

In accordance with the Chinese National Standard on domestic gas appliances (GB16410-1996), the CO emissions were measured with a circular sampling tube. The sampling tube was located vertically at a height of 30 mm from the bottom of the loading vessel and 1 mm away from the outer surface of the loading vessel. The concentration of the combustion product components were measured continuous using a commercial pollutant analyzer (model 300 CO/CO₂ analyzers (CAI) and model EU 200/4 O₂ analyzer (S&V)).

In this study the thermal efficiency of a burner is defined as the percentage of the heat input transferred to the water in the loading vessel. It was determined by measuring the elapsed time for a standard 4 kg load of water to be heated through a 50°C temperature rise (30°C to 80°C) and the heating value of the corresponding consumption of LPG. Mathematically, the thermal efficiency is expressed as:

$$\eta = \frac{M \times C_p \times \Delta T}{Q \times H_v} \times 100\% \quad (1)$$

In Equation (1), M (kg) is the load mass of water, C_p (kJ/kg°C) is the specific heat of water, Q (m³) is the LPG consumption, ΔT (°C) is the temperature rise and H_v (kJ/m³) denotes the heating value of LPG. The measured CO emissions were then corrected for 0 % excess oxygen and dry-basis. We use $CO(@0\% O_2)$ to designate CO emissions, which is defined by the following equation:

$$CO (@ 0\% O_2) = \left[\frac{20.9}{(20.9 - \%O_2)} \right] * CO_{measured} \quad (2)$$

In Equation (2), $CO(@0\% O_2)$ is the theoretical concentration of dry combustion products corresponding to 0 % excess oxygen. In addition, $\%O_2$ is the measured O_2 concentration and $CO_{measured}$ is the measured CO concentration. In the present study, the thermal efficiency and CO emissions of cooktop burner have been hypothesized to be influenced significantly by the following parameters: (1) Reynolds number (Re); (2) equivalence ratio ($Equiv.$); (3) jet-to-jet-spacing between the centers of two adjacent nozzles (S/d); and (4) nozzle-to-plate-distance (H/d), which is defined as the vertical distance from the bottom of the loading vessel to the top surface of the burner nozzle. These factors are coded at various levels and they are shown in Table 1.

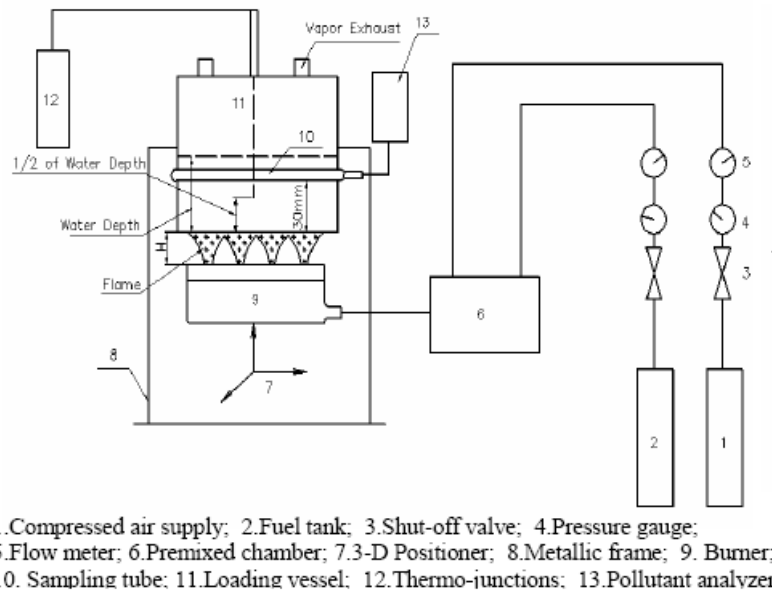


Fig. 1. Experimental set-up for cooktop design

Table 1. Levels of variables chosen for the Box-Behnken design

No.	Code	Factors	Variable level (coded value)		
			Low (-1)	Middle (0)	High (+1)
1	A	Re	200	350	500
2	B	$Equiv.$	1.4	1.7	2
3	C	H/d	1	4	7
4	D	S/d	5.2	10.4	15.6

3 Multiple Regression

The relationship between each response of thermal efficiency, CO or NO_x and control variables of *Re*, *Equiv*, *H/d* and *S/d* is respectively estimated by a quadratic model. The model is described in Equation (3).

$$y_k = a_0 + \sum_{i=1}^n a_i x_i + \sum_{i=1}^n \sum_{j=i}^n a_{ij} x_i x_j \tag{3}$$

where y_k is the predicted value of one response; $x_i (i = 1, 2, \dots, n)$ represent control variables; and the coefficient parameters of a_0, a_i, a_{ij} are estimated by least-square error method. The function *datafit* provided by Scilab is employed to estimate the coefficients. In order to facilitate the identification of important factors, the data are normalized before built into quadratic models. The control factors are normalized to [0.1, 0.9] and the response variables are normalized to [-0.9, 0.9].

Fig. 2, 3 and 4 respectively shows, with the function *bar* plot provided in Scilab, the normalized coefficients of each quadratic response model for thermal efficiency, CO and NO_x respectively. Root mean square errors (RMSE) and squared correlation coefficient (R^2) shown in the figures indicate that the quadratic models are adequate.

From Fig. 2 to 4, the contribution of each control factor to the responses can be seen. The linear, interaction, and quadratic effects of each factor on responses can be estimated. For example, it can be seen that, *H/d* is the most important design factors. Its positive large linear coefficient and negative medium quadratic coefficient on thermal efficiency indicate that as *H/d* is increased the thermal efficiency will firstly increase then decrease. Its interaction with *s/d* on thermal efficiency is also significant.

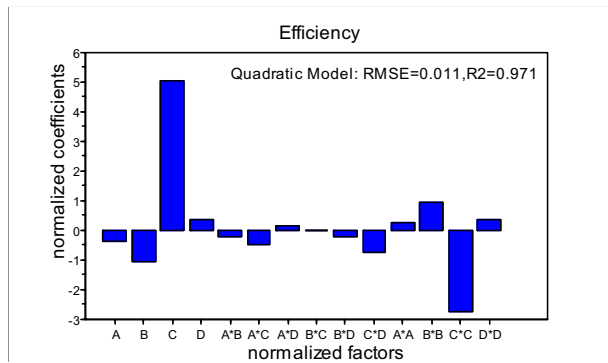


Fig. 2. Coefficients in the quadratic thermal efficiency model - showing the linear, interaction, and quadratic effects of each control factor

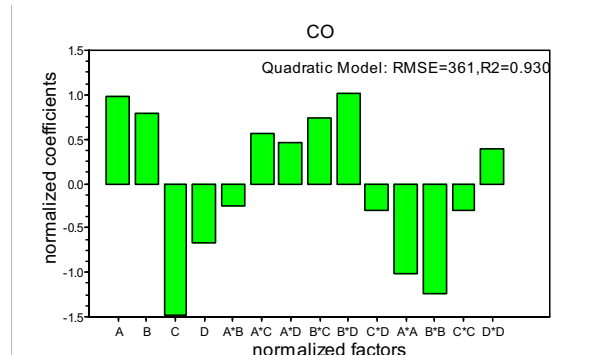


Fig. 3. Coefficients in the quadratic CO model -showing the linear, interaction, and quadratic effects of each control factor

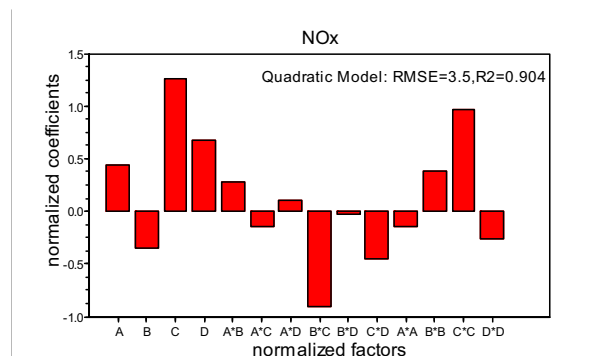


Fig. 4. Coefficients in the quadratic NO_x model -showing the linear, interaction, and quadratic effects of each control factor

4 Multi-objective Optimization

The optimization is performed under the experimental regions and emission requirements issued by the State Environmental Protection Administration of China (1997), viz. thermal efficiency $\geq 55\%$, CO ≤ 300 ppm@0%O₂ and NO_x ≤ 100 ppm@0%O₂. The problem is summarized in Equation (4).

$$\begin{aligned}
 & \max \text{ Efficiency} = f_1(\text{Re}, \text{Equ}, H/d, s/d) \\
 & \min \text{ CO} = f_2(\text{Re}, \text{Equ}, H/d, s/d) \\
 & \min \text{ NO}_x = f_3(\text{Re}, \text{Equ}, H/d, s/d) \\
 & \text{s.t.} \\
 & 200 \leq \text{Re} \leq 500, \quad 1.4 \leq \text{Equ} \leq 2.0, \\
 & 1 \leq H/d \leq 7, \quad 5.2 \leq s/d \leq 15.6; \\
 & \text{In code form: } (\text{Re})^2 + (\text{Equ})^2 + (H/d)^2 + (s/d)^2 \leq 2; \\
 & \text{Efficiency} \geq 55\%, \quad \text{CO} \leq 300, \quad \text{NO}_x \leq 100
 \end{aligned} \tag{4}$$

where f_1, f_2, f_3 are respectively the quadratic models for thermal efficiency, CO and NO_x . Based on Equation (4), the approximate feasible region of the control factors is estimated:

$$D = \left\{ (\text{Re}, \text{Equ}, H/d, s/d) \left| \begin{array}{l} 200 \leq \text{Re} \leq 330 \\ 1.4 \leq \text{Equ} \leq 1.6 \\ 6 \leq H/d \leq 7 \\ 5.2 \leq s/d \leq 12.4 \end{array} \right. \right\} \tag{5}$$

It can be seen that the value of *Equiv.* in the feasible condition set is at low levels. Fuel-rich conditions are generally tried to be avoided because it produces more emission of unburned hydrocarbon and CO. When fuel type and equivalence ratio is fixed for a burner, the flame length is determined by Reynolds number [16]. The value of *H/d* should be designed according to the flame length to ensure that the loading vessel is positioned in touch with the hottest zone of flames.

Simultaneous optimization of thermal efficiency and CO, NO_x emissions is performed according to the standardized Euclidian distance method. In this method, all three response variables are incorporated into a single function based on the distance between the predicted and optimum values estimated, as shown in Equation (6). The optimal solution can be found by minimizing the generalized distance function.

$$S(X) = \left(\sum_{k=1}^K \left\{ \frac{FD_k - FO_k(X)}{SD_k} \right\}^2 \right)^{\frac{1}{2}} \tag{6}$$

where $SD_k, FD_k, FO_k(X)$ are respectively, the standard deviation, the optimum value estimated individually, and the predicted value given in the input factors X , of the *k*th response variable.

The constrained optimization problem is solved by transforming it to an unconstrained optimization problem based on the penalty-function method. Then the function *optim* provided by Scilab is employed to search for the optimal solution. The predicted optimal solutions are presented in Table 2 and the condition at this solution is configured experimentally. At this condition, the flame length is about 25mm and appears to be blue-green in colour. The loading vessel is placed where its bottom is

exactly touch the tip of the inner premixed flame. Table 2 also presents comparisons of the observed and predicted optimal solutions. The results indicate an error range of less than 10%, which is acceptable for this study.

Table 2. Predicted (P) and configured (C) values of the optimum solution

	Control Factors				Responses		
	Re	Equ	H/d	S/d	Efficiency	CO	NO _x
P	203	1.64	6.6	5.3	0.613	300	60
C	200	1.6	6.5	5.2	0.602	284	56

5 Conclusions

Ensuring a better quality of life for everyone, now and for future generations, is a major aim in every responsible Government's sustainable development strategy. Hence accordance to the recommendations of the State Environmental Protection Administration of China (1997) on cooktop design, viz. thermal efficiency should be at least 55%; CO emissions should be less than or equal to 300 ppm@0%O₂ and NO_x emissions should be less than or equal to 100 ppm@0%O₂, a study on optimizing cooktop design parameters was proposed and approved by the Research Grants Council of Hong Kong SAR. With the aid of Scilab, a set of optimized design parameters was found through an optimized regression model and their ranges are:

- the values of *Re* should be between 200 and 330;
- the values of *H/d* would be between 6 to 7,
- the range of *S/d* is not restricted, indicating that *S/d* is not a control factor as crucial as *H/d* or *Re*, *Equiv*.
- the values of *Equiv* would be between 1.4 and 1.6.

Acknowledgements. The Authors wish to acknowledge the full financial support from the Research Grants Council of The Hong Kong SAR to the present project (CERG Project: PolyU 5276/04E).

References

1. Freeman, J.A. and Skapura, D.M. (1991). Neural Networks: Algorithms, Applications, and Programming Techniques. Addison-Wesley, New York, USA
2. Hagan, M.T., Demuth, H.B., Beale, M.H. (1996). Neural network design, PWS Pub, Boston, USA
3. Hou, S.S. and Ko, Y.C. (2004). Effects of heating height on flame appearance, temperature field and efficiency of an impinging laminar jet flame used in domestic gas stoves. Energy Conversion and Management, 45:1583-1595.

4. Hush, D.R. and Horne, B.G. (1993). Progress in Supervised Networks. *IEEE Signal Processing Jan*: 8-39.
5. Ko, Y.C. and Lin, T.H. (2003). Emissions and efficiency of a domestic gas stove burning natural gas with various compositions. *Energy Conversion and Management*, 44:3001-3014.
6. Lau, H.C.W., Wong T.T. and Ning A. (2001). Incorporating machine intelligence in a parameter-based control system: a neural-fuzzy approach. *Artif. Intell.Eng*, 15(3):253-264.
7. Li, H.B., Wong, T.T., Leung, C.W. and Probert S.D. (2006). Thermal performances and CO emissions of gas-fired cooktop burners. *Applied Energy*, in press.
8. Mishra, D.P. (2004). Emission studies of impinging premixed flames. *Fuel*, 83:1743-1748.
9. Myers, R.H. and Montgomery, D.C.(1995). *Response surface methodology: process and product optimization using designed experiments*. Wiley, New York, USA.
10. Stonington, J.F., Beashel, G., Murphy, T., Junus, R., Ashman, P.J. and Sergeant, G.D. (1994). Emissions and efficiency from production cooktop burners firing natural-gas. *J. Inst. Energy*, 67:143-155.
11. Takayama, K., Fujikawa, M. and Nagai, T. (1999). Artificial neural network as a novel method to optimize pharmaceutical formulations. *Pharmaceutical Research* 16: 1-6.
12. Takayama, K., Fujikawa, M. and Obata, Y. (2003). Neural network based optimization of drug formulations. *Advanced drug delivery reviews*, 55:1217-1231.
13. Turns, S. R. (2000). *An introduction to combustion: concepts and application*. McGraw-Hill, Boston, USA.
14. Wong, T.T. and Leung, C.W. (2002). Forced convection augmentation of turbulent flow in triangular duct with artificially roughened internal surfaces. *Experimental Heat Transfer*, 15:89-106.
15. Zhang, J., Smith, K.R., Uma, R., Ma, Y., Kishore, V.V.N., Lata, K., Khalil, M.A.K., Rasmussen, R.A. and Thorneloe, S.T.(1999). Carbon monoxide from cookstoves in developing countries: Emission factors. *Chemosphere: Global Change Science*, 1:353-366.
16. Zhao Z., Wong, T.T. and Leung, C.W. (2004). Impinging premixed butane/air circular laminar flame jet-influence of impingement plate on heat transfer characteristics. *Int. J. Heat and Mass Transfer*, 47:5021-503.

The Counting Algorithm of Transistor Based On Image Detect Technology

Peng Zhou, Xiangbo Yang, Hong Zhu, Ruirui Ji

Xi'an University of Technology,
Xi'an P. R. China 710048

zhoupeng8223@yahoo.com.cn yangxiangbo525@yahoo.com.cn
zhuhong@xaut.edu.cn jirui@xaut.edu.cn

Abstract. Aiming at the inaccuracy of the electronic scale, this paper proposed a counting method of transistor based on image recognition technology. Firstly, binary image can be got by OTSU. Secondly distance transformation is used to transform the position information of pixels to the gray information. Next conglomerate transistor region is segmented by watershed algorithm, and finally over-segmentation region is combined together to accomplish correct detection of transistor. Experimental results indicate the efficiency of this algorithm.

Keywords: Transistor, Counting, Distance transformation, Watershed algorithm, Conglutination segmentation

1 Introduction

At present, on transistor subpackage producing line, the transistors' number is often counted by the electronic scale, whose error is 2% for 1000 transistors. In order to improve measurement accuracy, this paper proposed a transistor counting method based on image detecting technology.

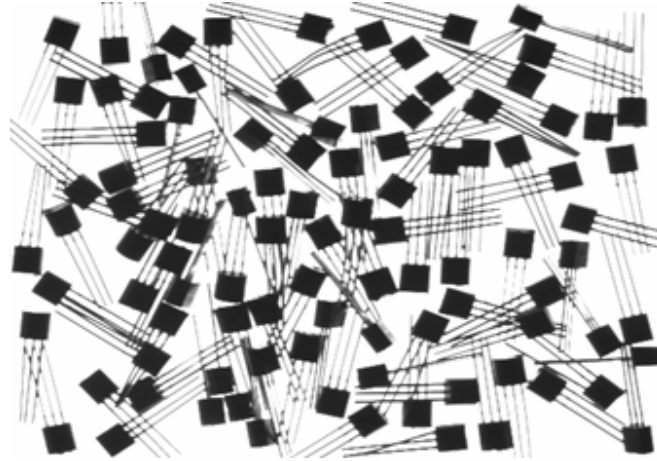


Fig. 1. Transistor image

As show in Figure 1, it is an image taken by the backlight illuminant. Sometimes the transistors neighbored conglutinate each other, and single transistor's area is difference owing to the laid form, Aiming at this difficulty, this paper firstly thresholds the image [3] by OTSU, then according to the statistic average value of area, counts the approximately transistor number, next recognizes the connected zones, segments them again and combines over-segmentation region [2] to acquire the correct number of transistors.

2 The detecting method of transistor

Transistor is an instruction with black head and three metal pins end. To counting transistor, only considering the black head is enough. So, the paper gets binary images by means of OTSU and separates the transistor head from other parts. Figure.2 (b) is the resulting image after threshold, it can be seen that transistor can be segmented clearly and some pins are also extracted, and however, there exists conglutination between transistors. So continued processing is needed.

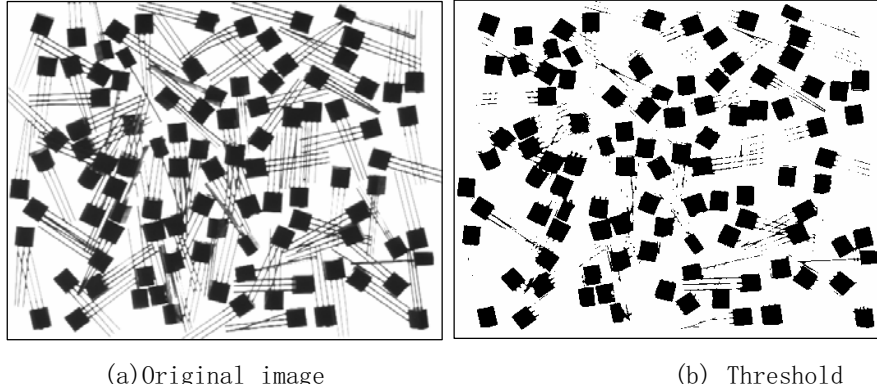


Fig. 2. The result image of threshold

2.1 Detecting the single transistor

Firstly, the binary image is eroded to cut off the part of feet, which are connected to the transistor head, and then labeling is used in order to filter weak noise, such as feet of small area. After this process, the result as shown in Figure.3 (a) can be obtained, where only existing the head of the transistor, including the head of single transistor and conglutination transistors.

Single transistor should be extracted firstly, for their big percent in image. Extracting is based on distributed characters of odd transistor area.

Firstly, as shown in Figure 3(a), the area of connected zones is counted. Supposing there are N connected zones, the areas of i -th connected zone is $S_i, i = 1, 2, \dots, N$. The average value and variance are μ_S and σ_S^2 . The connected zones of $S_i - \mu_S > \mu_S + \sigma_S$ are found out as the connecting transistor from

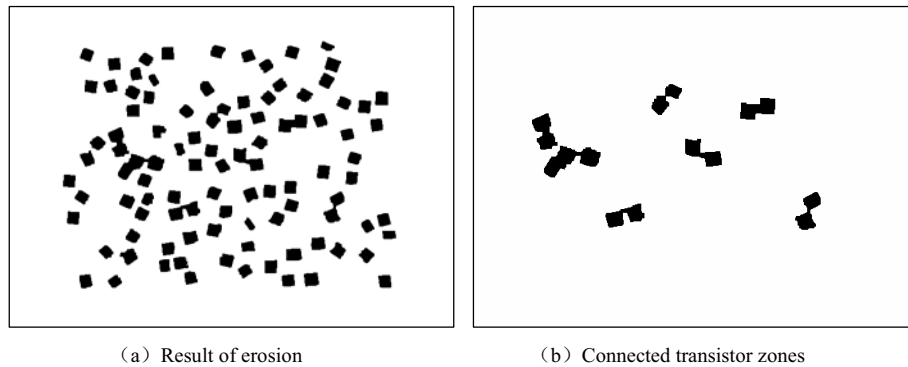


Fig. 3. Detecting the single

original image. In this way, circulating the same operation all the time until the variance is as small as possible, which demonstrates all the part of connected zones has been extracted. Then the resting connected zones is counted as the number of single transistor. After counting the number of single transistor, as shown in Figure 3(b), the connected transistors should be counted subsequently.

2.2 Detecting the conglutination transistor

Two or more than two conglutinated transistors form transistors conglutination region where must exist the shape or gray change. Therefore, different position information of the transistor pixel point is transformed to the gray information by distance transformation, and then separated transistors are obtained from the conglutination region with the method of the watershed algorithm [1] of adaptive threshold.

Distance transformation is the process, which converts threshold images into gray images in which pixels value is the distance to the nearest background.

As shown in Figure4, (b) is the result of applying distance transformation to (a). From this figure, different position information of pixels in the binary image is transformed to the gray information by distance transformation.

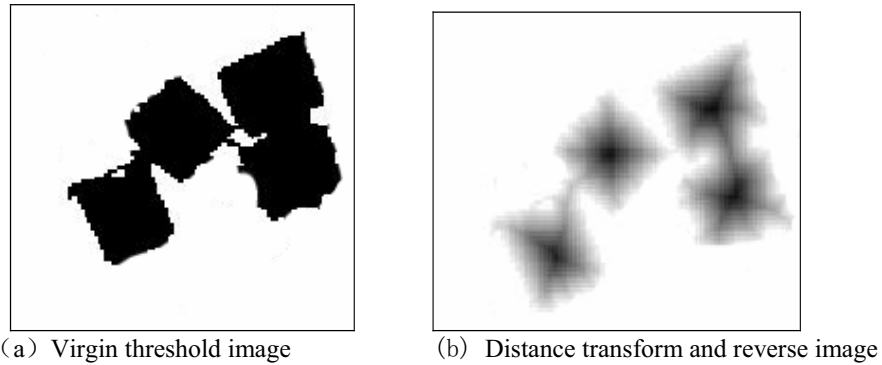


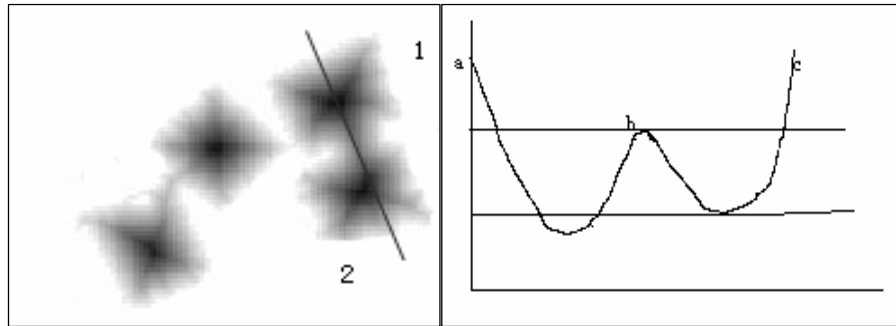
Fig. 4. Distance transform theory

Watershed algorithm is the method of mathematical morphology segmentation, which has been developed recently. The Figure 6 shows its theory drawing. Supposing the gray scale value of the objects is low, and the gray scale value of the background is high. (As shown in Figure 5(a)). The Figure 5(b) shows gray scale distribution along the line. b, and object 2 is from point b to point c. When edge touches each other, connecting characters can be got from gray distribution, i.e. the first touching point converts into the final edge of connecting objects. This process doesn't end until the threshold reach the gray scale of the background, in other word, when the edge point of the conglutination objects is correctly confirmed.

As show in Figure 6(a), it is the segmentation result of the conglutination transistor by the watershed algorithm. It can be seen that watershed algorithm is very sensitive to faint edge and overcomes the disadvantage of losing faint edge of traditional segmentation methods. But the sensitivity to noises of this algorithm results in over-segmentation as shown in Figure 6(a). So the detection of over-segmentation regions is needed, and to merge these regions reasonably.

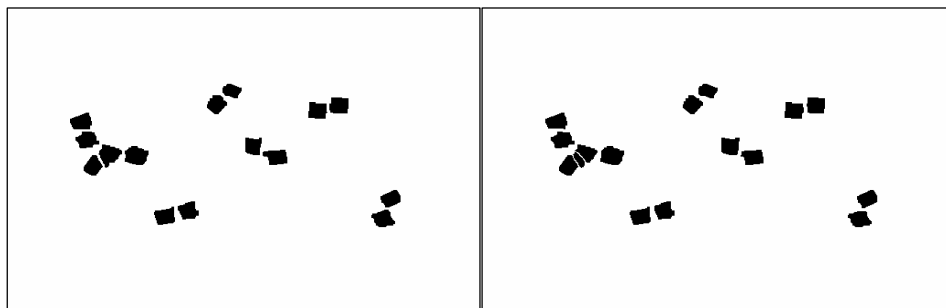
Firstly, all the connected areas after watershed segmentation are calculated and their average values \bar{S} are computed, then each connected area S_i is judged whether satisfies

$S_i < \bar{S}/k$ (k is experienced value) or not. When $S_i < \bar{S}/k$, it is an over segmentation area, therefore, this areas are merged to accomplish split connected region merging, as shown in Figure 6(b). As a result, all conglutinated transistors are exactly segmented. The number of transistors in the detected images can be obtained through counting the connected area after segmentation.



(a) Distance transform and reverse image (b) Gray scale distribution

Fig. 5. Sketch of watershed



(a) Initial segment image (b) The result of region merging

Fig. 6. Region merging

3 Experimental conclusion and analysis

A large numbers of effectively experiments are carried out using the above-mentioned method. As shown in Figure 7; (a) is the counted distributing histogram of transistor before conglutination process, in contrast to which, (b) is the result after conglutination process. Where horizontal axis denotes the area of the transistor; and vertical axis denotes the numbers of transistor of certain area. From the figure and the comparison of detected number with real number, the method of this article is proved that can exactly compute the number of transistors and deal with the conglutination phenomenon.

4. Conclusion

Considering the low accuracy of the electronic scale method used in semiconductor device factory at present, this paper proposed a transistor counting method based on image recognition technology. Firstly, binary image can be got by OTSU. Secondly, distance transformation is used to transform the position information to the gray information. Thirdly, conglutinate transistor region is segmented by watershed algorithm, and then over-segmentation region is combined. Experimental results indicate that this method can detect transistors correctly.

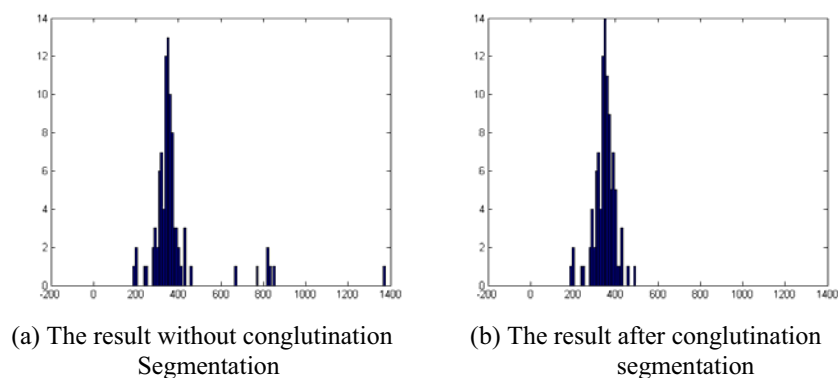


Fig. 7. Number distribution histogram of the triode

Reference

- [1] Vincent L , Soille P1 Watersheds in digital space : An efficient algorithms based on immersion simulation [J] 1 IEEE Transactions on Pattern Analysis and Machine Intelligence , 1991 , 13(6) : 583 ~ 598
- [2] Haris K, Efstratiadis S , *et al*1 Hybrid image segmentation using watersheds and fast region merging [J] 1 IEEE Transactions on Image Processing , 1998 , 7 (12) : 1684 ~ 1698
- [3] Ostu N A. Threshold Selection Method from Gray-Level Histograms[J]. IEEE Trans. On System, Man and Cybernetics, 1979,9(1):62-66.

Two efficient numerical methods for finding global minimizers of continuous functions of one variable

Yu He

College of Computer Science and Technology
Beijing Technology and Business University
Beijing 100037, China
heyu@th.btbu.edu.cn

Abstract. This paper presents two efficient numerical methods for finding global minimizers of continuous functions of one variable. Their convergence and complexity are discussed. Many representative examples are tested. Global problem is transformed to a local one so as to reduce difficulty. Methods are especially suitable for multimodal functions and they can be expanded to continuous functions of several variables. Theoretical analysis and tested examples show that they have particular character and effect compared with other global optimal methods.

Keywords: global minimizer; 1D continuous function; multimodal function

It is known that a continuous function on close set must possess of minimum and maximum, i.e. if $f(x)$ is continuous, $a \leq x \leq b$, it must possess of minimum and maximum on $[a,b]$. In fact, it is very difficult to solve out a numerical solution of the minimum or maximum. So is a multimodal function especially.

The study related to it often comes down to solve the problem finding global and local minimum of functions. The latter is easier to be solved then the former not only in theory but also practicality. Up to now, there had been many good algorithms to solve the latter problems, but the former. Even so, a good global algorithm is more important and needed.

Two methods for finding global minimizers of a continuous function of one variable are introduced fellow. They are especially suitable for multimodal functions. That finds minimum of $f(x)$ only is discussed because that finds maximum of it is equivalence to minimum of $-f(x)$. For conveniently, let $f(x)$ is a real multimodal continuous function of a variable on $[a,b]$ and its numbers of minimums and maximums are finite.

1. Idea of methods

1.1 Method 1

If function values of a continuous function are not equality, its graph is like continuous, stretched and rugged mountains. Look into Fig. 1. The process of finding maximum of $f(x)$ can be regarded to climbing one and one peak of mountains. When all peaks are crossed, the highest peak must be crossed. So that, whole climbing can be regarded to repeat same process many times: start off one end of $f(x)$, go down a valley, climb to a peak, go down next valley, climb to next peak,, arrive another end of $f(x)$ finally. The process is over. The maximum of $f(x)$ is found (Fig. 1).

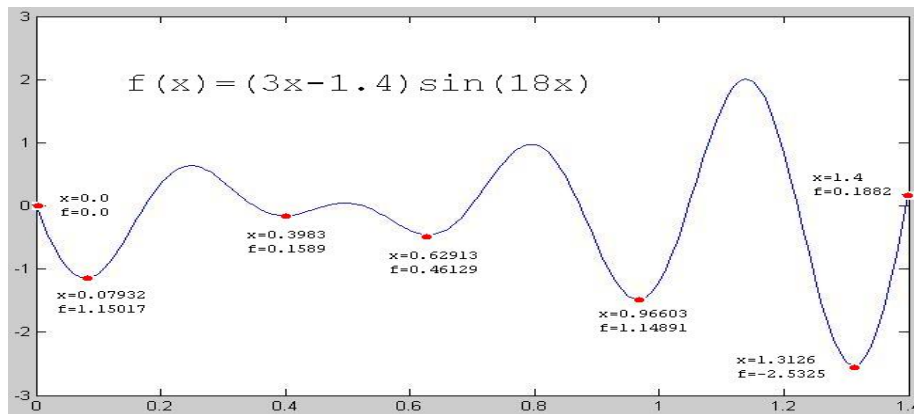


Fig. 1. graph and minimizers of $f(x)=(3x-1.4)\sin(18x)$, $x \in [0,1.4]$

Because process of going down valley and climbing to peak is equivalence, they should be treated with one. Every process is finding local extremum of $f(x)$. It is able to select to existing various mature feasible good local algorithms. The global problem is translated local one and the difficulty reduced (look into test examples). The method called Valley-Peak Method^[4,5].

1.2 Method 2

The process of finding global minimizers of a continuous multimodal function is like go downstairs from top floor of a skyscraper to ground of it. It is impossible to finish it one time. It should spend many times in a general way. The most difficult problem is that stairways in different floor are not in same position and it is not easy to find them. Suppose you go downstairs from the highest floor of a skyscraper. You find a stairway and rush down at first. You only go downstairs one floor or two floors at one time. Then you must look for another stairway. If another can be looked for, continue

run down. And look for next stairway. To repeat it until arrive first floor of the skyscraper. The process of finding minimizer of $f(x)$ repeats 2 steps: Step 1. To search out a stairway and go downstairs (make use of appropriate local algorithm to find a local minimizer); Step 2. After arrive lower floor, search out a new stairway which can go lower downstairs from (search out a point which can look for smaller function value make used of). It can construct an appropriate equation and solve it. A root of the equation is a point which finds smaller function value at. If a smaller minimizer is found, return to Step 1. To repeat Step 1 and Step 2 until the equation is not able to be solved. If so, the process^[10,11] is over (Fig. 3).

2. Realization of the algorithm

2.1 Method 1

Finding a global minimizer of $f(x)$ defined on $[a,b]$. Let step $h>0$. Selecting an appropriate local minimization algorithm based on property of $f(x)$, such as Gold Section Method^[1,2]. The process is as follows:

Step 1 Figure out function value $f(a)$ at left end a . Save as $f(mx)$ and mx respectively.

Step 2 Estimate trend of graph of $f(x)$. If downtrend, go to step 3, othewise, go to step 4.

Step 3 Find a local minimizer of $f(x)$, such as m . If $m \geq b$, go to step 5, else if $f(m) \leq f(mx)$, let $mx=m$ and $f(mx)=f(m)$.

Step 4 Step h forward. Solving a local minimizer of $-f(x)$, such as m . If $m \geq b$, go to step 5, else step h forward again and go to step 3.

Step 5 Figure out function value $f(b)$ at right end b . If $f(b) \leq f(mx)$, let $mx=b$ and $f(mx)=f(b)$.

Global minimizer and Global function value of $f(x)$ are mx and $f(mx)$ respectively.

2.2 Method 2

Finding a global minimizer of $f(x)$ defined on $[a,b]$. Let step $h>0$. Selecting an appropriate local minimization algorithm based on property of $f(x)$, such as Gold Section Method^[1,2] and a global algorithm to solve to the equation $f(x)-c=0$ (such as Interval Method^[15-17,22-26]), c is a constant. The process is as follows:

Step 1 Select a initial point $x_0 \in [a,b]$ arbitrarily(it is the first stairway).

Step 2 Find a local minimizer $m \in [a,b]$.

Step 3 Construct an equation $f_1(x)=f(x)-f(m)-h=f(x)-c=0$ and solve it. If there is a solution (it is possible that there are many solutions that all are belong to $[a,b]$. They must be in an interval which there is a smaller minimizer of $f(x)$ in), solve out one

solution x_1 and $x_0 = x_1$, go to step 2; otherwise, if there is no solution of the equation $f_1(x)$, $x \in [a, b]$, the process is over.

The local minimizer m and local function value $f(m)$ that solved out at the last time are global minimizer and global function value of $f(x)$.

Note: Because algorithms used to solve the equation $f_1(x)$ are iterative algorithms in most case, the solution of $f_1(x)$ does not need to be solved indeed. The work need to do is to find a function value $f(x')$ that is smaller than the last $f(m)$. The point x' must be in an interval that includes smaller function value than the last one. That makes x' a new x_0 can greatly decrease times of calculation and enhance efficiency.

3. Convergence and complexity

3.1 Method 1

Let $f(x)$ has n extrema (including 2 ends). It is foregoing know that the process to find minimizer of $f(x)$ is to call local algorithms many times. All local extrema of $f(x)$ can found out with them and global extremum will be got. That whether or not to find out global extremum is rest with property of idiographic $f(x)$ and local algorithm selected. Because of it, the property of $f(x)$ and local algorithm selected decide whether the Method 1 is convergence. So do the complexity. If Method 1 calls local algorithm N times, its complexity is N times to local algorithm. If local algorithm selected is a good one, Method 1 also is good. The complexity does not change in inbeing.

3.2 Method 2

Let $f(x)$ has k extrema (including 2 ends). It is foregoing know that Method 2 is repeats of 2 processes. One is to find local minimizers of $f(x)$. The process is called k times at best. The convergence and complexity of the algorithm called are decided by property of $f(x)$ and local algorithm selected and the same with Method 1. Another is to call algorithm that solves the equation $f_1(x) = f(x) - c = 0$ k times at best. The algorithm, such as Interval Method, generally has global convergence. It has good property. It can theoretically ensure that if there is a solution of equation, it must find out it; if there is no solution of equation, the process is over, i.e. it also is a condition of the process ceasing and global minimizer is gotten at same time. The convergence and complexity of Method 2 also relate to property of $f(x)$ and the algorithm that solves the equation $f_1(x) = 0$.

4. Testing examples

A lot of programs made with VC++ 6.0 , MATLAB V6.5 and SCILAB 4.0 in PC to test many testing functions. Result shows the Methods are successful and effect is satisfaction. Gold Section Method and 3 Points Interpolation Method^[1,2] are used to find an interval that includes a local minimizer of $f(x)$. Gold Section Method also is used to find a local minimizer and a function value of $f(x)$. Parts of testing functions, testing results and brief of testing are follows. Numerical values underlined are global function values of testing $f(x)$.

4.1 Method 1

1. Make $f(x)$ in 2-dimensional Shubert function(I)^[11], $x_1=-10, x_2=x_1$.

$$f(x) = \left\{ \sum_{i=1}^5 i \cos[-10(i+1) + i] \right\} \cdot \left\{ \sum_{i=1}^5 i \cos[(i+1)x + i] \right\}, x \in [-10, 10]$$

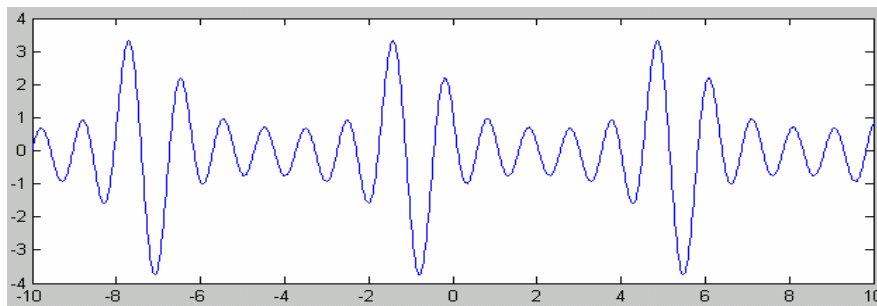


Fig. 2. one of sections of 2-dimensional Shubert function(I)

The $f(x)$ has 38 extrema (do not include 2 ends a, b). There are 3 global minimizers and minimum values in them. Table 1 shows part of testing data. There are 3 rows in one group. The first row includes 2 initialization points to find an interval which one minimizer in. The first 3 numerical values in the second row are 3 points in the interval which a local minimizer located in and the last values are their function values respectively. Numerical values in the third row are local minimizer in the interval and extremum found out. It is very expressly shows that maximum and minimum are found out alternately. This is one of characteristic of Method 1. Testing data shows that -3.748042 is minimum of objective function at -0.783506, -0.800321, and 5.482864. The data also includes maximums of the function.

Table 1. Testing results of testing example 1

$$f(a) = f(-10) = 0.0667$$

k	ax	bx	cx	fa	fb	fc
1	-10.00000	-9.999999				
	-9.999837	-9.983979	-9.104610	-0.067488	-0.139569	0.401753

	Xmax=	-9.780436		Fmax=	0.681933	
2	-9.780436	-9.780435				
	-9.508997	-9.341238	-8.932948	-0.233934	-0.882266	0.562571
	Xmin=	-9.286344		Fmin=	-0.933570	
3	-9.286344	-9.286343				
	-9.014905	-8.847145	-8.408635	-0.113911	-0.869023	1.226455
	Xmax=	-8.794063		Fmax=	0.926857	
4	-8.794063	-8.794062				
	-8.522623	-8.354864	-7.610124	-0.413830	-1.477963	2.922541
	Xmin=	-8.290388		Fmin=	-1.593927	
5	-8.290388	-8.290387				
	-7.851189	-7.709693	-7.579749	-2.553339	-3.325024	-2.640242
	Xmax=	-7.708314		Fmax=	3.325103	
6	-7.708314	-7.708313				
	-7.269115	-7.062625	-6.997675	-2.302840	-3.728780	-3.430890
	Xmin=	-7.083506		Fmin=	<u>-3.748042</u>	
7	-7.083506	-7.083505				
	-6.812067	-6.644308	1.330002	1.115922	-1.259553	0.754733
	Xmax=	-6.478571		Fmax=	2.200522	
.....						
38	9.069120	9.069121				
	9.340559	9.508318	9.916608	-0.233934	-0.882266	0.562571
	Xmin=	9.563212		Fmin=	-0.933570	

$$f(b) = f(10) = 0.8638$$

$$2. f(x) = -\sum_{p=1}^5 p \sin((p+1)x + p), \quad x \in [-10, 10]$$

It has 19 local minimizers and 3 global ones.

$$f(-6.774576)=f(-0.491391)=f(5.791794)=-12.03149$$

$$3. f(x)=(3x-1.4)\sin(18x), \quad x \in [0, 1.2]$$

It has 4 local minimizers and 1 global one.

$$f(0.079352)=-1.150173, \quad f(0.398387)=-0.158888,$$

$$f(0.629167)=-0.461289, \quad f(0.966086)=-1.489073$$

$$4. f(x)=2x^2-1.05x^4+x^6/6-|x|, \quad x \in [-1, 1]$$

It has 4 local minimizers and 2 global ones.

$$f(-1.0)=0.116667, f(-0.270447)=-0.129716,$$

$$f(0.270379)=-0.129716, f(1)=0.116667$$

4.2 Method 2

Because practice to global algorithm of solving equation is very difficult, global algorithm is not used to the example bellow. It only expresses the thought of Method 2. Function $f(x)=(3x-1.4)\sin(18x)$ is tested (picture 2). Because the function is downtrend at point a, let $x_0=a$. Find out minimizer $m_1=0.0793$. Let $h=0.03$. Solve equation $f_1(x)=f(x)-f(m_1)-h=0$. There are several solutions, such as $x_1=0.8727$ and $x_2=1.0472$. They are included in interval $[0.8,1.1]$ which there is smaller minimizer in. Make the solution of $f_1(x)=0$ new initialize point (stairway) to find a smaller minimizer m_2 . Start on next circle of finding a newer initialize point (stairway) x_0 . Creating a new equation $f_1(x)=f(x)-f(m_1)-h=0$ and solving it,, until there is not any solution in the last equation $f_1(x)=f(x)-f(m_3)-h=0$. The process is over. The last local minimizer m_3 and minimum function value $f(m_3)$ are global ones respectively.

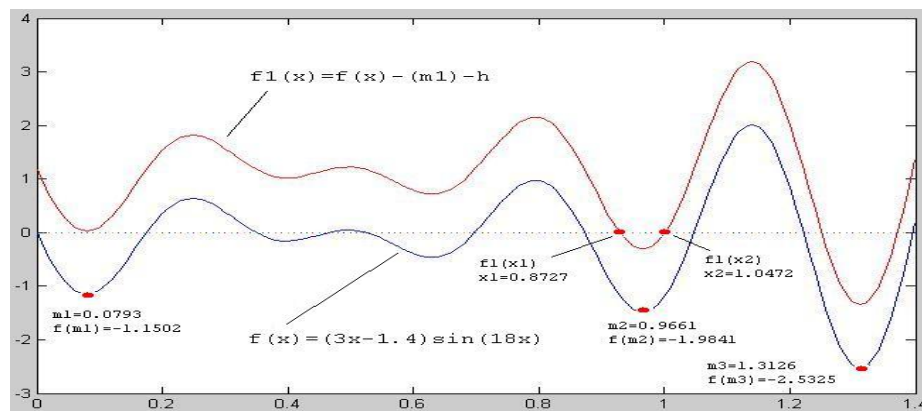


Fig. 3. Applying Method 2 to $(3x-1.4)\sin(18x)$

Summary

The effect of Method 1 that is used to find global minimizer of a continuous function, multimodal ones especially based on theory analysis and testing results is evidence. The objecting functions can be nonsmooth or weaker. Due to local algorithm is need in Method 1, different local ones can be selected according characters of objective

function. Do anything possible to select simple and convenient and quick stringent local algorithm.

Method 1 comparing with other global algorithms is obvious advantage. It is simple, practicable, fast, high efficient, easy applicable.

Method 2 needs 2 kinds of algorithm. One which is used to find local minimum of object function is the same with Method. Another which is used to solve equation is complex. Although restricted in practice, it is feasible in theory.

Method 1 and 2 can be applied to multidimensional cases.

References

1. YUAN Ya-xiang, SUN Wen-yu, Optimization Theory and Method, Academic Press, Beijing, 2001. (in Chinese)
2. XI Xiao-lin, ZHAO Feng-zhi. Optimization Method. Shanghai Academic Press, Shanghai. 1983. (in Chinese)
3. XI Xiao-lin. Nonlinear Optimization Method. Higher Education Press. Beijing. 1992.(in Chinese)
4. HE Yu. Numeral Algorithms and Programs in C++. Posts & Telecom Press. Beijing. 2003.(in Chinese)
5. HE Yu. Valley-Peak Method Applied to One-dimension functions. Journal of Beijing Technol and Business University. 2004,22(5):38-40.(in Chinese)
6. F.H.Branin. Solution of nonlinear DC network problem via differential equations. International I.E.E.E. Conference on systems networks & computers. Oaxtepey, Mexico. 1971.
7. A.V.Levy. The tunneling algorithm for the global minimization of functions. Present at the Dundee Conference on Numerical Analysis. Dundee, Scotland. 1977.
8. R.P.Ge. A filled function method for finding a global minimizer of a function of several variablesC.. Presented at the Dundee Biennial Conference on Numerical Analysis. Dundee. 1983.
9. R.Fletcher. Practical Method of Optimization (second edition). John Wiley and Sons. Chichester. 1987.
10. HE Yu. A further approach to the downstairs method for finding a global minimizers of a function. Numerical computation and Application of Computer. 1991,12(2):124-126(in Chinese); Chinese Journal of Numerical Mathematics and Applications. 1992,14(2):88-95(in English).
11. HE Yu. A downstairs method for finding a global minimizer of a function of several variable. Numerical computation and Application of Computer. 1989,10(4):193-207(in Chinese); Chinese Journal of Numerical Mathematics and Applications. 1990,12(1):8-22(in English).
12. A.Torn, A.Ziliskas. Global Optimazation. Berlin:Speinger-Verlag. 1990.
13. WANG De-ren, ZHANG Lian-sheng. Interval Methods of Nonlinear Systems. Shanghai Science and Technology Press. Shanghai. 1987(in Chinese).
14. LI Qing-yang, MUO Zhi-zhong, QI Li-qing. Numerical Methods of Nonlinear Systems. Scince Press. Beijing. 1987.(in Chinese)
15. E.R.Hansen. Global optimization using interval analysis, the multidimensional case. Numer. Math. 34(1980),247-344.

Two efficient numerical methods for finding global
minimizers of continuous functions of one variable 343

16. E.R.Hansen. Global optimization using interval analysis, the one-dimensional case. *J. Optimiz. Theory Appl.* 29(1979),331-344.
17. R.E.Moore. *Interval Analysis*. Prentice-Hall. New York. 1966.
18. R.Horst, H.Tuy. *Global Optimization: Deterministic Approaches*. Second, Revised Edition. Berlin: Springer-Verlag. 1993.
19. D-H Li, M.Fukushima. A modified BFGS method and its global convergence in nonconvex minimization. *Journal of Computational and Applied Mathematics*. 129(2001),15-35.
20. D-H Li, M.Fukushima. On the global convergence of BFGS method for nonconstrained optimization problems. *SIAM Journal on Optimization*. 11(2001),1054-1064.
21. J.Nocedal, S.Wright. *Numerical Optimization*. New York:Springer. 1999.
22. R.E.Moore. A test for existence of solutions to nonlinear systems, *SIAM J. Numer. Anal.*, 14:4(1977),611-615.
23. R.E.Moore. *Methods and Applications of Interval Analysis*, SIAM publication, Philadelphia, 1979.
24. E.Hansen. Interval forms of Newton's method. *Computing*. 20(1978),153-163.
25. K.Nickel. *Interval Mathematics*. Academic Press. New York. 1980.
26. K.Nickel. A globally convergent ball Newton method. *SIAM J, Numer. Anal.*, 18:6(1981),989-1003.
27. L.C.W.Dixon, G.P.Szego. *Towards Global Optimization*. Vol. 1,2. Holland: North-Holland. Amsterdam. 1975,1978.
28. Zhang Jiang-she, Wang Wei, Chen Bai-li. A cell exclusion algorithm for finding all global minimizers of functions in multiple variables. *Computing Mathematics*. 1995(4):443-455.(in Chinese)
29. Shen Pei-ping. An interval algorithm for finding all global minimizers of nonsmooth functions. *Numerical computation and Application of Computer*. 1998(3):212-217.(in Chinese)

Using Scilab to Perform a Cogeneration System Simulation and Optimization

Fonseca Jr., J.G.S.¹, Asano H.¹, Fujii T.¹, Hirasawa S.¹

¹ Mechanical Systems Department, Kobe University,
1-1 Rokkodai, Nada, 657-8501, Kobe Japan
{Fonseca Jr., J.G.S, Tel/Fax: +81-78-803-6153, E-mail: 022d893n@stu.kobe-u.ac.jp}

Abstract. The objective of this work is to perform the simulation and optimization of a cogeneration system operation strategy while meeting the typical energy demand of a 20000 m² hotel for the period of one year using Scilab. The simulated system has a boiler, a gas turbine, an auxiliary steam turbine, a water heating system as well as air conditioning ones. The operational strategy optimization was focused on fuel energy saving ratio and the results show that the use of cogeneration decreased the energy consumption in 7.23% when compared with the normal solution. In terms of running cost the savings reached 35.75%. Moreover, the meaningful gain in the calculations using Scilab 4.0 when compared with version 3.0, as well as the effect of Scilab matrix techniques to replace loops structures in the program will be shown.

Keywords: Scilab, Cogeneration System, Operational Strategy, Simulation, Optimization.

1 Introduction

Scilab is a powerful tool that can be used to approach many different problems in the engineering field. One of the areas where Scilab can be applied is in the simulation and modelling of energy systems.

Cogeneration systems are those able to generate heat transfer to a certain use and power in sequence [1]. Depending on the heat and electricity demand conditions, such kind of systems can present substantial energy savings [2]. One of the available ways to assess potential gains in efficiencies and savings generated by cogeneration is the use of computational programming. In this way, the system design, the interaction between its several equipments, as well as the best way to operate them, for example, can be modelled and simulated.

In cogeneration systems' modelling and simulation, many works are available. For instance, when the system present many components and the demand conditions are uncertain, mixed-integer linear programming can be used to tackle the design problem like in [3]. Moreover, to analyze cogeneration systems performance, information theory or evolutionary algorithms can also be used, like in [4] and [5] respectively.

This work, has by purpose to develop a program to simulate a small-scale cogeneration system working under different demand conditions. Moreover, it is

expected the developed program will not only simulate a cogeneration system, but also perform the optimization of its operational strategy focused on fuel-energy savings. Scilab was chosen as the programming language to build the simulator due the advantages of its built in functions application and programming techniques.

2 Problem Description

The hypothetical system simulated is composed by a gas turbine, a boiler, an auxiliary steam turbine (which use the exhaustion gases and is coupled to the refrigeration system), a water and an air heater. The system uses as fuel natural gas (modelled in this work as methane) and there is also the possibility to buy electricity from the network. For this work was assumed that none electricity generation surplus can be sold. A simple system's sketch is shown on fig. 1.

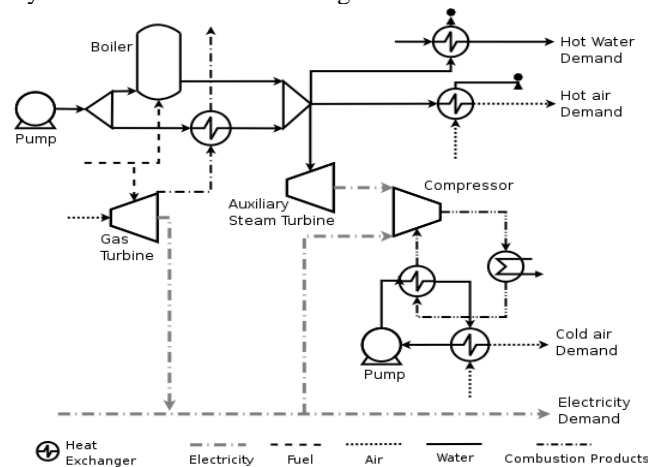


Figure 1 - Sketch of the simulated cogeneration system.

The main components were characterized with partial load model and the system was designed to attend the typical hourly demand of a hotel in Japan during one year. The demand data correspond to 24 hours of one typical day per month and was provided by the society of heating, air conditioning and sanitary engineers of Japan. The gas turbine was sized to work from a maximum load corresponding to the average steam demand needed by the hotel to a minimum of 40% of this demand. Moreover, the refrigeration system attached to the auxiliary turbine and the boiler were both sized to attend the hotel maximum cooling and steam demand respectively. The capacities and efficiencies data for the main equipments forming the system are on table 1.

Table 1. Cogeneration system's main equipments capacities.

Main Components	Capacity (full load)	Performance
Gas Turbine	Power output: 746 kJ/s	1 st Law efficiency: 18.7%
Boiler	Heat output: 5.43 MJ/s	1 st Law efficiency: 90.0%
Auxiliary Steam Turbine	Power output: 461 kJ/s	1 st Law efficiency: 91.4%
Compressor	Power input: 614 kJ/s	COP : 3.41

The partial load models used for the main equipments are available on [6]. Equations 1, 2 and 3 show the gas turbine model used. In this case, for instance, the gas turbine's work output (W_{gt} in J/s) is related not only with the fuel mass flows (m_f and $m_{f,max}$ in kg/s) but also with the turbine size (L_{gt} , A_{gt} and B_{gt}) and the fuel heating value (NHV in J/kg).

$$W_{gt} = n_{gt} \cdot m_f - W_{gt,int} \quad (1)$$

$$n_{gt} = \frac{L_{gt} + 1}{B_{gt}} \cdot \left(NHV - \frac{A_{gt}}{m_{f,max}} \right) \quad (2)$$

$$W_{gt,int} = \frac{L_{gt}}{B_{gt}} \cdot (NHV \cdot m_{f,max} - A_{gt}) \quad (3)$$

The main performance parameter regarded in this work was the fuel-energy saving ratio (*FESR*), defined by [2] as the amount of energy that a cogeneration system saves, when compared with the conventional solution (use of a boiler to meet the heat demand and purchasing all the electricity needed), to attend a certain demand condition. Another important parameter is the system's running cost. For this work, the running cost rc (in yen), showed in equation 4, was regarded as the sum of the fuel cost (C_{fuel}) and the purchased electricity one (C_{elec}) in every hour n throughout the year.

The basic tariffs used for fuel and electricity were 37.21 yen/m³ and 12.11 yen/kWh respectively. Both values were referenced by [7].

$$rc = \sum_1^n C_{fuel} + \sum_1^n C_{elec} \quad (4)$$

3 Program Development

To deal with the problem of how to use the system to have the best possible running cost condition during the whole year, while attending an hourly variable demand, an optimization procedure was implemented in a computer program developed in Scilab. Each equipment was described as independent functions having

some parameters initialized from the main program. For instance, the functions get data as fuel, air or water mass flow, enthalpy or temperature and pressures values and return data as work output, steam production, heated water flow, etc., according the equipment that is being modelled. The output of certain functions act as input data for subsequent ones and so on until the power and steam generation cycle is completely described. Moreover, in addition to these functions and in order to model the combustion products thermodynamic properties in different pressures and temperatures, a set of extra functions were written. The thermodynamic properties equations for methane, air, water, nitrogen, oxygen, carbonic gas as well as hydrogen, described by these functions are referenced by [8]. The simulator validation was done comparing its results in different points of the cycle with those provided by a commercial simulator for the system operating at full load.

Once the cycle is simulated, values of efficiencies, as well as running cost are obtained. The program then, computes all feasible operational strategy conditions (with an accuracy of 1% of the gas turbine maximum load) that satisfy the hourly demand requirements, storing the corresponding performance values. After this step, using a maximization function available in Scilab, the best operational strategy for FESR was located. For the simulations, a laptop pentium 1.6 GHz with 256 MB of RAM memory was used.

4 Results

Examples of the results obtained are on figure 2A, where the operational strategy for a summer month is showed. From that figure, it can be seen how the gas turbine, boiler and auxiliary steam turbine are used in several demand conditions. Also in figure 2A, the hourly running cost using the conventional system and the cogeneration one, optimized for the best FESR, can be compared. The advantage, from the running cost point of view, of using cogeneration instead the conventional system, in the simulations done for July, was clear.

From the results provided by the developed program, the main equipments annual average load were obtained and they are showed on figure 2B. Furthermore, other system's performance parameters were also calculated by the simulator and are available on table . It is interesting to see in table that although the FESR value was low, the amount of energy saved was meaningful. Moreover, although the system can produce the electricity demand, the amount of electricity bought from public grid during the year was expressive. This happened for two reasons: due equipments sizing, in some occasions, the electricity demand was higher than the system's capacity and then electricity had to be purchased; the second is related with the fact that for some demand conditions and from the FESR point of view, it is not good to use cogeneration at all, but the conventional solution.

Table 2: Cogeneration system's performance parameters for the best FESR solution.

	EconomE (Ecn)	FESR	Energy Saved (Wh)	Power Generated (Wh)	Power Bought (Wh)
Annual Value	3.74E+07	7.23E-02	2.27E+09	1.36E+08	6.14E+07

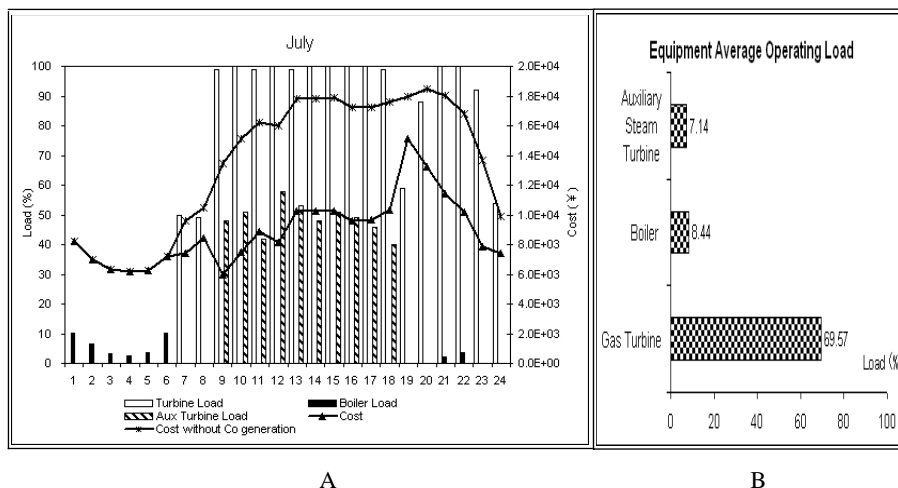


Figure 2 - Optimal Operation Strategy focused on FESR.

5 Simulator Speed Test

The simulator was originally built in Scilab 3.1.1 and it was originally conceived with a set of loops to calculated all feasible solutions. However, simulations were done using version 3.1.1 and 4 and the results on table 3. From this table, one can note that the time needed to run the simulation decreased substantially. Namely, the difference reached 19.3% from the 3.1.1 version.

To verify the effect of the replacement of some loops structures by matrix operations, as mentioned by [9], on the program execution time, part of the program related with the gas turbine function was altered and just the altered part execution time was 66.7% faster than using a loop structure to perform the same kinds of operations. For both tests, the Scilab time measurement function *time()* was used.

Table 3: Calculation speed using different Scilab versions.

Program	Scilab 3.1.1*	Scilab 4*	% Difference
Execution time (s)	24 771	19 978	19.3%

*This value represents the average of 3 simulations run in different occasions.

6 Conclusion

In this work, the development of a program to simulate and optimize a cogeneration system using Scilab was done. The program developed was able to simulate the cogeneration system operation under several water and air heating, electricity consumption and air cooling demand conditions.

It was also showed that given a demand data, the program developed was able not only to simulate the cogeneration system, but also to find its best operational strategy from the FESR point of view. Furthermore, the cogeneration solution could be

compared with the conventional system one making possible to assess potential efficiencies improvements and savings. With the developed program, it is also possible to find the best operation strategy for other objectives as the running cost or the exergetic efficiency for example.

Finally, Scilab proved to be a suitable tool to build the simulator. Its new version (4) decreased substantially the program execution time when compared with the previous one (3.1.1). Also, as showed in this work, its matrix operation features, when replacing some loops structures can enable even higher gains on the program execution time. Due to those results, modifications in the simulator are being studied in order to decrease its execution time.

References

1. Moran M. J., Shapiro H. S., *Principios de Termodinamica para Engenharia*, LTC (portuguese version), Rio de Janeiro-Brazil, (2002).
2. Horlock J. H., *Cogeneration - Combined Heat and Power*, Krieger Publishing Company, Malabar, USA, (1997).
3. Gamou S., Yokoyama R., Ito K., Optimal unit sizing of cogeneration systems in consideration of uncertain energy demands as continuous random variables, *Energy Conversion & Management: An International Journal*, Pergamon, n 43, (2002), 1349 – 1361.
4. Takahashi K., Ishizaka T., Application of the Information Theory for the Analysis of the cogeneration-system performance, *Applied Energy*, n 61, (1998), 147 - 162.
5. Tsai M., Lin W., Application of evolutionary programming to optimal operational strategy cogeneration system under time of use rates, *Electrical Power and Energy Systems*, n 22, (2000), 367 - 373.
6. Varbanov P.S., DoEle S. and Smith R., Modelling and Optimization of Utility Systems, *Chemical Engineering Research and Design*, n 82, (2004), 561 - 578.
7. Asano H., Fujii T., Hisazumi Y., Tsuda H., Simulation and Optimization of Districted Energy Supply System Based on Exergy, *Proceedings of the International Conference on Power Engineering-03*, Kobe, Japan, (2003).
8. Bejan, A., Tsatsaronis, G., Moran M., *Thermal Design & Optimization*, John Wiley and Sons Inc., New York, USA, (1996).
9. Campbell, S. L., Chancelier, J., Nikoukhah, R., *Modeling and Simulation in Scilab/Scicos*, Springer, USA, (2006).

Author Index

A		Guo, X.F.	9
Abderrazak, B.	129		
Alan, L.	129	H	
Arnaud, M.	21	He, Y.	335
Asano, H.	345	Hirasawa, S.	345
C		J	
Cai, J.	143, 107	Jiang, Q.L.	211
Chen, J.Y.	279	Ji, R.R.	327
Chen, S.M.	233	J.G.S.	345
Chen, W.	79	K	
Chen, W.G.	211	Kang, C.	49
Chen, X.D.	143	L	
Chen, Y.S.	157	Laurent, M	297
D		Leung, C.W.	317
David, A.	221	Li, J.F.	297
Denis, C.	21	Li, G. Q.	9
E		Li, H.B.	317
Edouard, N.	129	Li, S.Q.	97
F		Li, S.R.	107
Fan, J. H.	9	Li, Y.J.	107
Fang, K.J.	211	Liu, D.	89
Farid, B.	69	Liu, D.Y.	317
Fonseca, Jr.	345	Liu, G.	9
G		Liu, J	143
Gan, G.Y.	279	Luo, L.L.	143
Gareth, B.	221	Liu, S.W.	9
Ge, D.Q.	9	Lin, T.Z.	143
Guo, H.D.	9	Liu, W.J.	143
		Liu, Y.P.	107
		Liu, Y.	107

Liu, Z.L.	189	Wang, K.J.	119
		Wang, X	119
M		Wang, Y.	107
Ma, L.H.	61,175,211,233	Wang, Y.L.	97
Ma, Y.	189	Wang, Z.H.	107
Maurice, G	297	Wong, T.T.	317
Min, D.C.	107	Wu, B.	79
		Wu, C.	143
P		X	
Pan, Y.H.	107	Xiao, R.	89
Pan, W.	197	Xu, X.M.	107
Peng, Z.	61,175	Xue, P.F.	97
Q		Y	
Qin, S.Y.	247	Yan, G.	79
Quan, S.H.	197	Yang, K.	143
		Yang, X.B.	327
R		Z	
Ramine, N.	3	Zeng, W.F.	107
Raphael, S.	129	Zhang, D.	49
Raymond, Q.	129	Zhang, G.Y.	189
Roberto, B.	31	Zhang, L.Y.	197
		Zhang, Y.H.	157
S		Zhang, Z.P.	211
Serge, S.	69	Zhou, L.F.	269,309
Sebastien, M.	129	Zhao, L.L.	269
Silvano, B.	31	Zhou, L.W.	269
Steve, D.	221	Zhou, P.	327
Su, H.T.	143	Zhou, X.	89
		Zhao, Z.T.	189
T		Zhu, C.C.	247
Tao, Y.T.	279	Zhu, H.	327
Tibault, R.	129	Zhuang, R.P	143
W			