

Università di Pisa  
Facoltà di Ingegneria  
Corso di Laurea in Ingegneria Informatica

# Model Order Reduction Toolbox

Riduzione di modello con implementazione in Scilab®

Progetto di  
Controllo Digitale  
Prof. Ing. Aldo Balestrino

*23 febbraio 2007*

Domenico de Giulii

## **Sommario**

In questo documento è presentato uno studio su alcune tecniche di riduzione di modello per sistemi dinamici complessi. In particolare, è proposto un toolbox in Scilab® in grado di effettuare riduzioni su sistemi tempo-continui e tempo-discreti, sia stabili che instabili, e mostrare le differenze tra il sistema completo e quello ridotto.

# Indice

|  |    |
|--|----|
| Introduzione .....   | 4  |
| 1. Realizzazione bilanciata .....  | 5  |
| 1.1 Note sui sistemi instabili .....   | 7  |
| 2. Decomposizione .....  | 9  |
| 3. Riduzione .....   | 10 |
| 3.1 Truncate .....   | 10 |
| 3.2 Match Gain .....   | 11 |
| 3.3 Indice di Qualità .....  | 13 |
| 4. Scilab <sup>®</sup> Model Order Reduction Toolbox .....                       | 14 |
| 4.1 Contenuto .....  | 14 |
| 4.2 Installazione .....  | 14 |
| 4.2.1 GNU/Linux .....  | 14 |
| 4.2.2 Microsoft <sup>®</sup> Windows <sup>®</sup> .....                          | 15 |
| 4.3 Funzionamento .....  | 16 |
| 5. Test ed esempi .....  | 21 |
| 5.1 Esempio di riduzione su modello tempo-discreto stabile del 4° ordine .....   | 21 |
| 5.1.1 Riduzione ad ordine 2 con il metodo Truncate .....                         | 22 |
| 5.1.2 Riduzione ad ordine 2 con il metodo Match Gain .....                       | 23 |
| 5.2 Esempio di riduzione su modello tempo-continuo stabile del 4° ordine .....   | 24 |
| 5.2.1 Riduzione ad ordine 2 con il metodo Truncate .....                         | 25 |
| 5.2.2 Riduzione ad ordine 2 con il metodo Match Gain .....                       | 26 |
| 5.3 Esempio di riduzione su modello tempo-discreto instabile del 4° ordine ..... | 27 |
| 5.3.1 Riduzione ad ordine 3 con il metodo Truncate .....                         | 28 |
| 5.3.2 Riduzione ad ordine 3 con il metodo Match Gain .....                       | 29 |
| 5.4 Esempio di riduzione su modello tempo-continuo instabile del 4° ordine ..... | 30 |
| 5.4.1 Riduzione ad ordine 3 con il metodo Truncate .....                         | 31 |
| 5.4.2 Riduzione ad ordine 3 con il metodo Match Gain .....                       | 32 |
| 6. Appendice: codice sorgente routine Scilab <sup>®</sup> .....                  | 33 |
| 6.1 mor.sci .....  | 33 |
| 6.2 modred.sci .....   | 38 |
| 6.3 dtsi_mod.sci .....   | 40 |
| 7. Bibliografia .....  | 41 |

## Introduzione

La riduzione di modello è di fondamentale importanza in diverse applicazioni della modellazione matematica e del controllo, poiché permette di ridurre la complessità di sistemi dinamici di ordine elevato, preservandone alcune caratteristiche peculiari, quali stabilità, guadagno, etc.

In particolare, nella progettazione di regolatori lineari per sistemi dinamici, può essere utile approssimare il modello originale da controllare con un modello di ordine minore, che conservi il più possibile il legame ingresso/uscita, ed effettuare la sintesi su questo modello.

Lavorare con un modello ridotto, infatti, comporta diversi vantaggi, quali la riduzione della complessità computazionale e del tempo di simulazione in ambiente CAD, ma anche il minor costo realizzativo nell'implementazione circuitale.

Sul mercato sono attualmente disponibili diversi prodotti commerciali *general purpose* in grado di automatizzare le procedure di riduzione. Tuttavia tali software sono coperti da diritto d'autore e richiedono spesso licenze molto costose.

Per questo motivo si è scelto di sviluppare un toolbox libero e completamente *open-source*, basato su Scilab<sup>®</sup>, un potente e versatile software scientifico libero sviluppato dai laboratori francesi dell'INRIA.

Grazie a questo toolbox è possibile ridurre sistemi tempo-continui o tempo-discreti, sia stabili che instabili, utilizzando diversi algoritmi. È inoltre possibile scegliere il grado di riduzione che si intende applicare al sistema ed analizzare l'efficienza delle riduzioni mediante lo studio della risposta temporale (risposta al gradino) e frequenziale (diagrammi di Bode) e l'analisi di un indice di qualità.

## 1. Realizzazione bilanciata

Al fine di individuare i modi “poco osservabili” o “poco controllabili”, quindi potenzialmente eliminabili, del sistema dinamico da ridurre, è necessario trasformare il sistema in una forma equivalente bilanciata.

Si definisce gramiano di controllabilità in tempo-discreto la matrice

$$W_C^d \triangleq \sum_{j=0}^{\infty} A^j B B^T (A^T)^j$$

Si definisce gramiano di osservabilità in tempo-discreto la matrice

$$W_O^d \triangleq \sum_{j=0}^{\infty} (A^T)^j C^T C A^j$$

Si definisce gramiano di controllabilità in tempo-continuo la matrice

$$W_C^c \triangleq \int_0^{\infty} e^{At} B B^T e^{A^T t} dt$$

Si definisce gramiano di osservabilità in tempo-continuo la matrice

$$W_O^c \triangleq \int_0^{\infty} e^{A^T t} C^T C e^{At} dt$$

Un sistema in forma di spazio di stati si dice bilanciato se il gramiano di controllabilità ( $W_C$ ) ed il gramiano di osservabilità ( $W_O$ ) sono uguali e diagonali.

Dato il sistema tempo-discreto in forma di spazio di stato

$$\begin{cases} x(k+1) = Ax(k) + Bu(k) \\ y(k) = Cx(k) + Du(k) \end{cases}$$

ipotizzando che sia stabile, completamente osservabile e controllabile, è possibile trovare una matrice  $T$  di trasformazione<sup>1</sup> tale per cui il sistema si porti in forma bilanciata

$$\tilde{A} = T^{-1}AT, \quad \tilde{B} = T^{-1}B, \quad \tilde{C} = CT, \quad \tilde{D} = D$$

$$\begin{cases} \tilde{x}(k+1) = \tilde{A}\tilde{x}(k) + \tilde{B}u(k) \\ \tilde{y}(k) = \tilde{C}\tilde{x}(k) + \tilde{D}u(k) \end{cases}$$

Si ottiene quindi

$$\tilde{W}_c^d = \tilde{W}_o^d = \Sigma, \quad \Sigma = \begin{bmatrix} \sigma_1 & 0 & \cdots & 0 \\ 0 & \sigma_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \sigma_n \end{bmatrix}$$

Per i sistemi tempo-continui vale una procedura analoga.

Dato il sistema tempo-continuo in forma di spazio di stato

$$\begin{cases} \dot{x}(t) = Ax(t) + Bu(t) \\ y(t) = Cx(t) + Du(t) \end{cases}$$

ipotizzando che sia stabile, completamente osservabile e controllabile, è possibile trovare una matrice  $T$  di trasformazione<sup>1</sup> tale per cui il sistema si porti in forma bilanciata

$$\tilde{A} = T^{-1}AT, \quad \tilde{B} = T^{-1}B, \quad \tilde{C} = CT, \quad \tilde{D} = D$$

$$\begin{cases} \dot{\tilde{x}}(t) = \tilde{A}\tilde{x}(t) + \tilde{B}u(t) \\ \tilde{y}(t) = \tilde{C}\tilde{x}(t) + \tilde{D}u(t) \end{cases}$$

Si ottiene quindi

$$\tilde{W}_c^c = \tilde{W}_o^c = \Sigma, \quad \Sigma = \begin{bmatrix} \sigma_1 & 0 & \cdots & 0 \\ 0 & \sigma_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \sigma_n \end{bmatrix}$$

---

<sup>1</sup> Per maggiori dettagli sul calcolo della matrice  $T$  di trasformazione per portare il sistema in forma bilanciata, si faccia riferimento a [6].

## 1.1 Note sui sistemi instabili

La procedura di bilanciamento appena descritta è valida esclusivamente per sistemi stabili. Al fine di generalizzare tale procedura anche per sistemi instabili, si è adottato un semplice espediente. Ipotizzando che sia possibile decomporre facilmente il sistema (di ordine  $n+m$ ) in una parte stabile (di ordine  $n$ ) ed in una parte instabile (di ordine  $m$ ), si applica la procedura di bilanciamento esclusivamente alla parte stabile.

Per tener conto dei modi instabili è sufficiente espandere le matrici di stato già bilanciate con le righe e le colonne relative agli stati instabili.

Per semplicità ipotizziamo che i due sottosistemi (stabile ed instabile) siano diagonali e strettamente propri ( $D=0$ ).

Il sottosistema stabile, che è stato precedentemente bilanciato, contiene  $n$  stati:

$$A_{stable}^{balanced} = \begin{bmatrix} \rho_1^s & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \rho_n^s \end{bmatrix}, \quad B_{stable} = \begin{bmatrix} b_1^s \\ \vdots \\ b_n^s \end{bmatrix}, \quad C_{stable} = [c_1^s \quad \cdots \quad c_n^s]$$

Poiché questo sottosistema è stato bilanciato, secondo la procedura descritta nel paragrafo precedente, è possibile ricavare il gramiano di controllabilità e l'equivalente gramiano di osservabilità:

$$W_C^{stable} = W_O^{stable} = \Sigma^{stable}, \quad \Sigma^{stable} = \begin{bmatrix} \sigma_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \sigma_n \end{bmatrix}$$

Il sottosistema instabile, che non può essere bilanciato, contiene  $m$  stati:

$$A_{unstable}^{unbalanced} = \begin{bmatrix} \rho_1^u & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \rho_m^u \end{bmatrix}, \quad B_{unstable} = \begin{bmatrix} b_1^u \\ \vdots \\ b_m^u \end{bmatrix}, \quad C_{unstable} = [c_1^u \quad \cdots \quad c_m^u]$$

Per il sottosistema instabile non è possibile calcolare il gramiano di controllabilità né quello di osservabilità.

A questo punto è possibile unire i due sottosistemi precedentemente decomposti, riempiendo di zeri gli spazi rimasti vuoti nella fusione della matrice  $A_{full}$ :

$$A_{full} = \begin{bmatrix} \rho_1^u & 0 & 0 & 0 & 0 & 0 \\ 0 & \ddots & 0 & 0 & 0 & 0 \\ 0 & 0 & \rho_m^u & 0 & 0 & 0 \\ 0 & 0 & 0 & \rho_1^s & 0 & 0 \\ 0 & 0 & 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & 0 & 0 & \rho_n^s \end{bmatrix}, \quad B_{full} = \begin{bmatrix} b_1^u \\ \vdots \\ b_m^u \\ b_1^s \\ \vdots \\ b_n^s \end{bmatrix}, \quad C_{full} = [c_1^u \quad \cdots \quad c_m^u \quad c_1^s \quad \cdots \quad c_n^s]$$

Per quanto riguarda la matrice gramiano (di controllabilità o di osservabilità), abbiamo soltanto informazioni inerenti la parte stabile del sistema. Per tener conto anche della parte instabile, inseriamo nella matrice tanti autovalori ad *infinito* quanti sono gli stati instabili:

$$W_{full} = \begin{bmatrix} \infty & 0 & 0 & 0 & 0 & 0 \\ 0 & \ddots & 0 & 0 & 0 & 0 \\ 0 & 0 & \infty & 0 & 0 & 0 \\ 0 & 0 & 0 & \sigma_1 & 0 & 0 \\ 0 & 0 & 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & 0 & 0 & \sigma_n \end{bmatrix}$$

Il motivo per cui gli autovalori del gramiano relativi ai modi instabili sono fissati ad *infinito* risiede nel significato che gli stati instabili assumono nel sistema da ridurre.

Come verrà meglio chiarito nel capitolo 3, il criterio di riduzione di modello si basa sugli autovalori della matrice gramiano. Stati trascurabili, quindi potenzialmente eliminabili, corrispondono ad autovalori  $\sigma_i$  del gramiano trascurabili.

Ponendo ad infinito gli autovalori relativi ad i modi instabili, infatti, si associa priorità massima agli stati instabili, che non potranno mai considerarsi trascurabili.

In genere, infatti, è preferibile non trascurare i modi instabili nella modellazione di un sistema dinamico e tanto più nella sintesi di un sistema di ordine ridotto. A tal scopo, si preferisce effettuare le riduzioni esclusivamente sulla parte stabile del sistema e preservare integra, nel modello finale, la parte instabile. In questo modo un eventuale controllore che agisce sul sistema ridotto può conservare inalterate le sue caratteristiche di stabilizzazione.

Qualora il numero di modi instabili sia dominante, questo approccio non risulta particolarmente efficiente. Tuttavia in quasi tutte le applicazioni reali il numero di poli instabili è di gran lunga inferiore al numero di poli stabili.

## 2. Decomposizione

Una volta che il sistema è in forma bilanciata, è possibile decomporlo in due sottosistemi, che contengono, rispettivamente, gli stati non trascurabili e quelli trascurabili. Consideriamo il vettore di stato e le matrici del sistema tempo-discreto nella seguente rappresentazione:

$$x(k) = \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix}, \quad A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}, \quad B = \begin{bmatrix} B_1 \\ B_2 \end{bmatrix}, \quad C = [C_1 \quad C_2]$$

dove  $x(k) \in \mathbb{R}^n$ ,  $x_1(k) \in \mathbb{R}^p$ ,  $x_2(k) \in \mathbb{R}^q$ , con  $n = p + q$ .

Per semplicità ipotizziamo che il sistema sia strettamente proprio, si ha quindi  $D = 0$ .

Possiamo riscrivere il sistema di partenza nella forma

$$\begin{cases} x(k+1) = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} x(k) + \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} u(k) \\ y(k) = [C_1 \quad C_2] x(k) \end{cases}$$

⇓

$$\begin{cases} x_1(k+1) = A_{11}x_1(k) + A_{12}x_2(k) + B_1u(k) \\ x_2(k+1) = A_{21}x_1(k) + A_{22}x_2(k) + B_2u(k) \\ y(k) = C_1x_1(k) + C_2x_2(k) \end{cases}$$

Per i sistemi tempo-continui vale una procedura analoga.

Consideriamo il vettore di stato e le matrici del sistema tempo-continuo nella seguente rappresentazione:

$$x(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix}, \quad A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}, \quad B = \begin{bmatrix} B_1 \\ B_2 \end{bmatrix}, \quad C = [C_1 \quad C_2]$$

Dove  $x(t) \in \mathbb{R}^n$ ,  $x_1(t) \in \mathbb{R}^p$ ,  $x_2(t) \in \mathbb{R}^q$ , con  $n = p + q$ .

Per semplicità ipotizziamo che il sistema sia strettamente proprio, si ha quindi  $D = 0$ .

Possiamo riscrivere il sistema di partenza nella forma

$$\begin{cases} \dot{x}(t) = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} x(t) + \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} u(t) \\ y(t) = [C_1 \quad C_2] x(t) \end{cases}$$

⇓

$$\begin{cases} \dot{x}_1(t) = A_{11}x_1(t) + A_{12}x_2(t) + B_1u(t) \\ \dot{x}_2(t) = A_{21}x_1(t) + A_{22}x_2(t) + B_2u(t) \\ y(t) = C_1x_1(t) + C_2x_2(t) \end{cases}$$

Ugualmente si può scomporre il gramiano in due sottomatrici diagonali  $\Sigma_1$  e  $\Sigma_2$

$$W_C = W_O = \Sigma = \begin{bmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_2 \end{bmatrix}$$

### 3. Riduzione

Il criterio di riduzione di modello che adotteremo si basa sugli autovalori della matrice gramiano  $\Sigma$ . Stati trascurabili, quindi potenzialmente eliminabili, corrispondono ad autovalori  $\sigma_i$  del gramiano trascurabili.

Eventuali autovalori ad infinito sono da considerarsi infinitamente grandi, quindi assolutamente non trascurabili.

$$\Sigma = \begin{bmatrix} \infty & 0 & 0 & 0 & 0 & 0 \\ 0 & \ddots & 0 & 0 & 0 & 0 \\ 0 & 0 & \infty & 0 & 0 & 0 \\ 0 & 0 & 0 & \sigma_1 & 0 & 0 \\ 0 & 0 & 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & 0 & 0 & \sigma_n \end{bmatrix}$$

È possibile, quindi, decomporre la matrice gramiano in due sottomatrici  $\Sigma_1$  e  $\Sigma_2$ , che contengono, rispettivamente, gli autovalori non trascurabili e quelli trascurabili:

$$\Sigma = \begin{bmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_2 \end{bmatrix}$$

È ovvio che tutti gli autovalori ad infinito apparterranno alla sottomatrice  $\Sigma_1$ .

#### 3.1 Truncate

Ipotizzando che gli autovalori della matrice  $\Sigma_2$  siano trascurabili rispetto a quelli della matrice  $\Sigma_1$ , è possibile eliminare completamente i modi corrispondenti ad  $x_2$ . Si ottiene quindi il seguente sistema tempo-discreto ridotto:

$$\begin{cases} x_1(k+1) = A_{11}x_1(k) + B_1u(k) \\ y(k) = C_1x_1(k) \end{cases}$$

Per i sistemi tempo-continui vale una procedura analoga.

$$\begin{cases} \dot{x}_1(t) = A_{11}x_1(t) + B_1u(t) \\ y(t) = C_1x_1(t) \end{cases}$$

Le caratteristiche strutturali del sistema di partenza (stabilità, osservabilità, controllabilità) sono rimaste inalterate nel sistema ridotto, tuttavia non è stato conservato il guadagno.

Per ovviare a questo problema si utilizza un'altra tecnica di riduzione, chiamata *Match Gain*.

### 3.2 Match Gain

Ipotizzando che gli autovalori della matrice  $\Sigma_2$  siano trascurabili rispetto a quelli della matrice  $\Sigma_1$ , al fine di conservare il guadagno del sistema di partenza, si considerano i modi corrispondenti ad  $x_2$  come infinitamente veloci. Per sistemi tempo-discreti si ha quindi:

$$x_2(k+1) = x_2(k)$$

Da qui otteniamo il seguente sistema ridotto:

$$\begin{cases} x_1(k+1) = A_{11}x_1(k) + A_{12}x_2(k) + B_1u(k) \\ x_2(k) = A_{21}x_1(k) + A_{22}x_2(k) + B_2u(k) \\ y(k) = C_1x_1(k) + C_2x_2(k) \end{cases}$$

$$\Downarrow$$

$$\begin{cases} x_1(k+1) = A_{11}x_1(k) + A_{12}x_2(k) + B_1u(k) \\ x_2(k) = (I - A_{22})^{-1}(A_{21}x_1(k) + B_2u(k)) \\ y(k) = C_1x_1(k) + C_2x_2(k) \end{cases}$$

$$\Downarrow$$

$$\begin{cases} x_1(k+1) = (A_{11} + A_{12}(I - A_{22})^{-1}A_{21})x_1(k) + (B_1 + A_{12}(I - A_{22})^{-1}B_2)u(k) \\ y(k) = (C_1 + C_2(I - A_{22})^{-1}A_{21})x_1(k) + C_2(I - A_{22})^{-1}B_2u(k) \end{cases}$$

Si possono ora porre le seguenti uguaglianze:

$$\hat{A} = A_{11} + A_{12}(I - A_{22})^{-1}A_{21}$$

$$\hat{B} = B_1 + A_{12}(I - A_{22})^{-1}B_2$$

$$\hat{C} = C_1 + C_2(I - A_{22})^{-1}A_{21}$$

$$\hat{D} = C_2(I - A_{22})^{-1}B_2$$

Da cui deriva il seguente sistema ridotto:

$$\begin{cases} x_1(k+1) = \hat{A}x_1(k) + \hat{B}u(k) \\ y(k) = \hat{C}x_1(k) + \hat{D}u(k) \end{cases}$$

Per i sistemi tempo-continui vale una procedura analoga.

Si considerano i modi corrispondenti ad  $x_2$  come infinitamente veloci, per cui

$$\dot{x}_2(t) = 0$$

Da qui otteniamo il seguente sistema ridotto:

$$\begin{cases} \dot{x}_1(t) = A_{11}x_1(t) + A_{12}x_2(t) + B_1u(t) \\ 0 = A_{21}x_1(t) + A_{22}x_2(t) + B_2u(t) \\ y(t) = C_1x_1(t) + C_2x_2(t) \end{cases}$$

$\Downarrow$

$$\begin{cases} \dot{x}_1(t) = A_{11}x_1(t) + A_{12}x_2(t) + B_1u(t) \\ x_2(t) = -A_{22}^{-1}A_{21}x_1(t) - A_{22}^{-1}B_2u(t) \\ y(t) = C_1x_1(t) + C_2x_2(t) \end{cases}$$

$\Downarrow$

$$\begin{cases} \dot{x}_1(t) = (A_{11} - A_{12}A_{22}^{-1}A_{21})x_1(t) + (B_1 - A_{12}A_{22}^{-1}B_2)u(t) \\ y(t) = (C_1 - C_2A_{22}^{-1}A_{21})x_1(t) - C_2A_{22}^{-1}B_2u(t) \end{cases}$$

Si possono ora porre le seguenti uguaglianze:

$$\hat{A} = A_{11} - A_{12}A_{22}^{-1}A_{21}$$

$$\hat{B} = B_1 - A_{12}A_{22}^{-1}B_2$$

$$\hat{C} = C_1 - C_2A_{22}^{-1}A_{21}$$

$$\hat{D} = -C_2A_{22}^{-1}B_2$$

Da cui deriva il seguente sistema ridotto:

$$\begin{cases} \dot{x}_1(t) = \hat{A}x_1(t) + \hat{B}u(t) \\ y(t) = \hat{C}x_1(t) + \hat{D}u(t) \end{cases}$$

Il sistema che si ottiene con questa procedura ha l'indubbio vantaggio di conservare inalterato il guadagno del sistema di partenza, oltre che preservare le caratteristiche strutturali.

### 3.3 Indice di Qualità

Al fine di valutare l'efficacia della riduzione, si utilizza un indice di qualità  $J$  di tipo ISE (Integral of Squared Error). Tale indice misura lo scostamento tra le risposte al gradino unitario del sistema completo e del sistema ridotto.

In tempo-discreto l'indice  $J$  è valutato come segue:

$$J_d = \sum_{k=0}^{\infty} (y_{red} - y)^2$$

dove  $y_{red}$  è il vettore dell'uscita del sistema ridotto, mentre  $y$  è il vettore dell'uscita del sistema completo.

Nei sistemi simulati, tuttavia, il tempo di esecuzione non può arrivare all'infinito, per cui si sceglie di arrestare il calcolo della risposta al gradino unitario ad un valore  $k = N$  sufficientemente grande da catturare tutte le dinamiche transitorie del sistema. L'indice di qualità diventa perciò:

$$J'_d = \sum_{k=0}^N (y_{red} - y)^2$$

Considerazioni analoghe si possono fare per sistemi tempo-continui.

L'indice  $J$  è valutato come segue:

$$J_c = \int_0^{\infty} (y_{red} - y)^2 dt$$

dove  $y_{red}$  è il vettore dell'uscita del sistema ridotto, mentre  $y$  è il vettore dell'uscita del sistema completo.

Nei sistemi simulati, tuttavia, il tempo di esecuzione non può arrivare all'infinito, per cui si sceglie di arrestare il calcolo della risposta al gradino unitario ad un valore  $t = N$  sufficientemente grande da catturare tutte le dinamiche transitorie del sistema. L'indice di qualità diventa perciò:

$$J'_c = \int_0^N (y_{red} - y)^2 dt$$

Si fa notare che un modello ridotto è tanto più fedele al modello completo quanto più è vicino a 0 l'indice di qualità  $J$ .

## 4. Scilab® Model Order Reduction Toolbox

Grazie a questo toolbox per Scilab®, è possibile ridurre sistemi tempo-continui o tempo-discreti, sia stabili che instabili, utilizzando il metodo *Truncate* o il metodo *Match Gain* presentati nel capitolo precedente.

È inoltre possibile scegliere il grado di riduzione che si intende applicare al sistema ed analizzare l'efficienza delle riduzioni mediante lo studio della risposta temporale (risposta al gradino) e frequenziale (diagrammi di Bode) e l'analisi dell'indice di qualità.

### 4.1 Contenuto

Nel pacchetto applicativo *Scilab® Model Order Reduction Toolbox* sono contenuti i file necessari al funzionamento del toolbox e la relativa documentazione.

|                     |   |
|---------------------|---|
| <i>doc.pdf</i>      | Documentazione (questo file)                              |
| <i>mor.sci</i>      | File principale del toolbox                               |
| <i>loader.sce</i>   | File di installazione                                     |
| <i>modred.sci</i>   | Funzioni con gli algoritmi di riduzione                   |
| <i>dtsi_mod.sci</i> | Funzione per la decomposizioni in stati stabili/instabili |

### 4.2 Installazione

Il toolbox *Model Order Reduction* è stato sviluppato per Scilab® versione 3.x, 4.x e superiori in ambiente GNU/Linux e Microsoft® Windows®.

Verrà di seguito presentata una procedura di installazione del toolbox per entrambi i tipi di Sistema Operativo.

#### 4.2.1 GNU/Linux

1. Copiare il contenuto della cartella MOR (*loader.sce; modred.sci; mor.sci*) in una directory di cui si hanno permessi in scrittura  
(ad es. */home/utente/MOR*);
2. Avviare Scilab®;
3. Posizionarsi nella path in cui è stato estratto l'archivio  
(ad es. *--> cd /home/utente/MOR* );
4. Eseguire il file *loader.sce*  
*--> exec loader.sce;*

Si fa notare che il file *loader.sce* va eseguito ogni volta che si avvia una nuova sessione di Scilab o si esegue un reset del workspace (comando *clear*).

#### 4.2.2 Microsoft® Windows®

1. Copiare il contenuto della cartella MOR (*loader.sce*; *modred.sci*; *mor.sci*) in una directory di cui si hanno permessi in scrittura  
(ad es. C:\MOR);
2. Avviare Scilab®;
3. Posizionarsi nella path in cui è stato estratto l'archivio  
(ad es. --> `cd c:\MOR` );
4. Eseguire il file *loader.sce*  
--> `exec loader.sce`;

Si fa notare che il file *loader.sce* va eseguito ogni volta che si avvia una nuova sessione di Scilab® o si esegue un reset del workspace (comando `clear`).

### 4.3 Funzionamento

Dopo l'installazione, è possibile avviare il toolbox mediante il comando `mor` nella console:

```
--> mor ;
```

Qualora si voglia salvare il risultato di una riduzione, è necessario chiamare il toolbox come una funzione ed assegnare il valore di ritorno ad una variabile:

```
--> SysRed=mor( ) ;
```

In questo caso, al termine dell'operazione, la variabile *SysRed* conterrà la rappresentazione nello spazio degli stati dell'ultimo sistema ridotto.

Se si è interessati anche alla rappresentazione bilanciata del sistema di partenza (non ridotto), è possibile chiamare la funzione *mor()* con un secondo argomento di ritorno:


```
--> [SysRed, SysBal]=mor( ) ;
```

In questo caso, al termine dell'operazione, la variabile *SysRed* conterrà la rappresentazione nello spazio degli stati dell'ultimo sistema ridotto, mentre *SysBal* conterrà la rappresentazione nello spazio degli stati del sistema originale bilanciato.

All'avvio del toolbox viene richiesto se si vuole lavorare con un sistema tempo-continuo o tempo-discreto:



In entrambi i casi si dovranno inserire i parametri caratteristici del sistema, ovvero le matrici A, B, C, D. Nel caso discreto si dovrà specificare anche il tempo di campionamento.



Inserisci i parametri del sistema tempo-discreto


A

B

C

---

< Indietro   **Avanti >**   Annulla



Inserisci i parametri del sistema tempo-discreto

D

Tempo di campionamento

---

< Indietro   **Fine**   Annulla

I valori default per le matrici A, B, C, D sono rispettivamente le variabili A, B, C, D, che possono essere inizializzate nella console prima dell'avvio del toolbox. Il valore default per il tempo di campionamento (per sistemi tempo-discreti) è di 0.1 secondi.

Se il sistema non è completamente controllabile o non è completamente osservabile il toolbox termina. Allo stesso modo, se il sistema non ha stati stabili, non è possibile effettuare riduzioni e il toolbox termina.

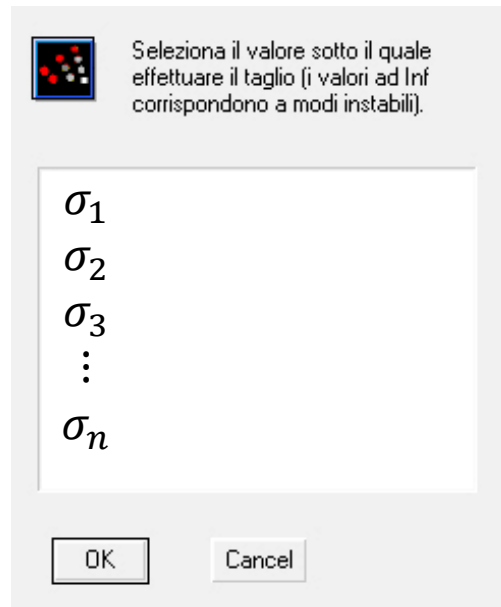
Eventuali stati instabili presenti nel sistema vengono notificati con un messaggio di *warning*.

Il sistema inserito viene quindi bilanciato, mediante la trasformazione presentata nel capitolo 1, ed è mostrato l'elenco ordinato degli autovalori  $\sigma_i$  del gramiano di controllabilità (equivalente al gramiano di osservabilità):

$$\tilde{W}_C = \tilde{W}_O = \Sigma, \quad \Sigma = \begin{bmatrix} \sigma_1 & 0 & \cdots & 0 \\ 0 & \sigma_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \sigma_n \end{bmatrix}$$

$$\sigma_i = \text{diag}(\Sigma) = \begin{bmatrix} \sigma_1 \\ \sigma_2 \\ \vdots \\ \sigma_n \end{bmatrix}$$

Se nel sistema ci sono  $m$  stati instabili, vengono inseriti in testa  $m$  autovalori ad infinito, come chiarito nel paragrafo 1.1.

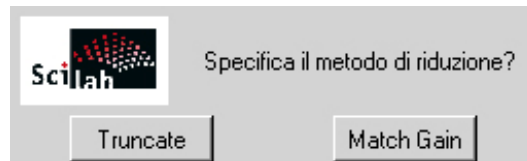


È richiesto di selezionare l'autovalore sotto il quale effettuare il taglio, secondo la scomposizione descritta nel capitolo 2. Se si seleziona l' $i$ -esimo autovalore, i primi  $i-1$  modi del sistema resteranno inalterati, mentre gli ultimi  $i$  stati verranno eliminati nella procedura di riduzione.

Se non si seleziona alcun autovalore, non verrà effettuata alcuna riduzione. Se si seleziona il primo autovalore  $\sigma_1$ , il toolbox restituirà un messaggio d'errore, dal momento che non è possibile ridurre un sistema all'ordine 0.

Eventuali autovalori ad infinito, corrispondenti a modi instabili, non possono essere selezionati come valore di taglio.

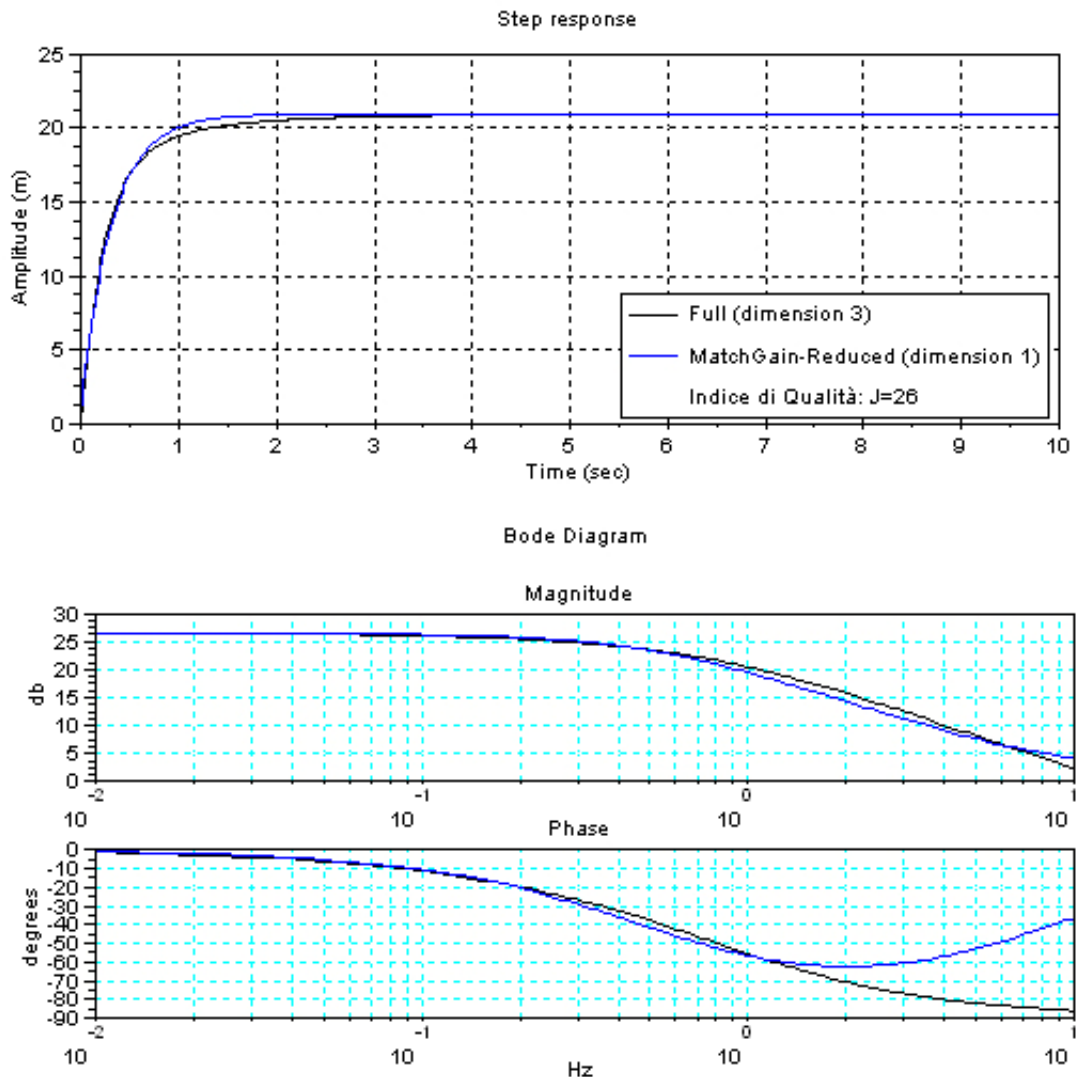
Una volta selezionato il valore di taglio è possibile scegliere il metodo di riduzione che si intende utilizzare:



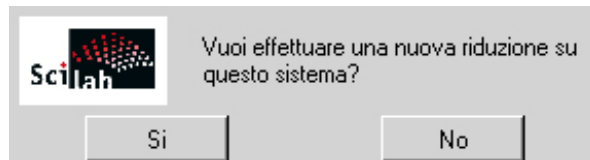
Per maggiori dettagli sugli algoritmi relativi ai metodi *Truncate* e *Match Gain*, si rimanda ai paragrafi 3.1 e 3.2 rispettivamente.

Dopo aver scelto il metodo di riduzione vengono stampate la risposta temporale (risposta al gradino) e la risposta frequenziale (diagrammi di Bode di modulo e fase) del sistema completo (in nero) e del sistema ridotto (in blu). Viene inoltre indicato il valore dell'indice di qualità  $J$  corrispondente all'approssimazione corrente.

Di seguito è presentato un esempio di riduzione di un sistema tempo-continuo stabile di ordine 3 mediante il metodo *Match Gain* con approssimazione ad ordine 1.



Il toolbox permette, inoltre, di effettuare riduzioni multiple sullo stesso sistema, al fine di verificare e confrontare visivamente l'efficacia della procedura e la fedeltà dell'approssimazione. Per questo motivo, al termine di ogni procedura di riduzione viene chiesto all'utente se si ha intenzione di effettuare una nuova riduzione sullo stesso sistema di partenza:



In caso di risposta affermativa, verrà riavviata la procedura di riduzione a partire dalla scelta dell'autovalore di taglio dal gramiano del sistema bilanciato. In caso di risposta negativa, il toolbox termina.

Il risultato dell'ultima riduzione effettuata, ovvero la rappresentazione nello spazio degli stati dell'ultimo sistema ridotto, è salvato nella variabile di ritorno della funzione, se specificata, altrimenti è salvato nella variabile di sistema *ans*.

## 5. Test ed esempi

Al fine di testare il corretto funzionamento del toolbox, verranno di seguito effettuati alcuni test di riduzione su modelli stabili ed instabili, tempo-discreti e tempo-continui. Verranno mostrati i risultati delle riduzioni e si verificherà l'efficienza degli algoritmi utilizzati mediante l'analisi delle risposte e degli indici di qualità.

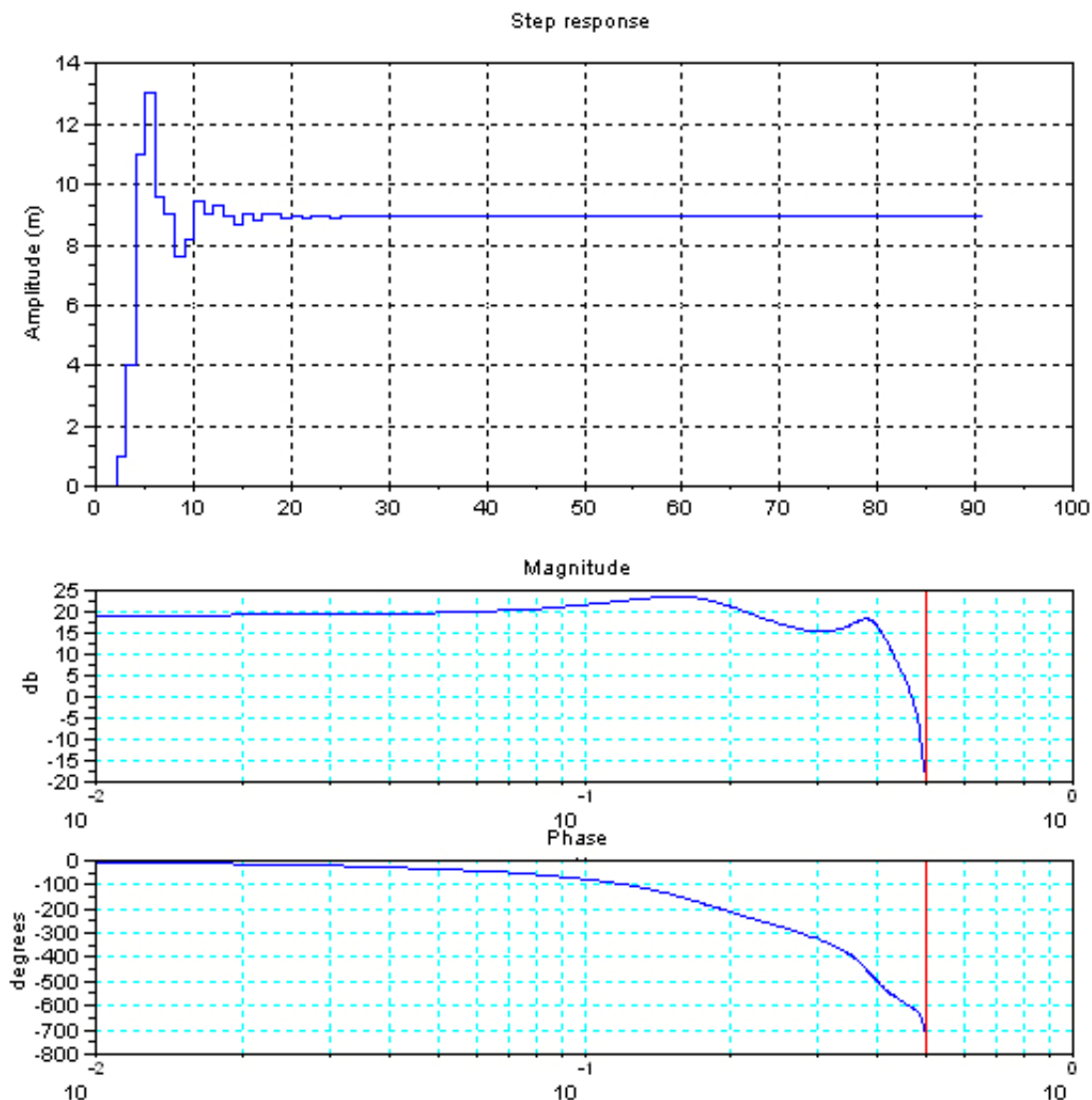
### 5.1 Esempio di riduzione su modello tempo-discreto stabile del 4° ordine

Consideriamo il seguente sistema tempo-discreto stabile del 4° ordine descritto nello spazio degli stati:

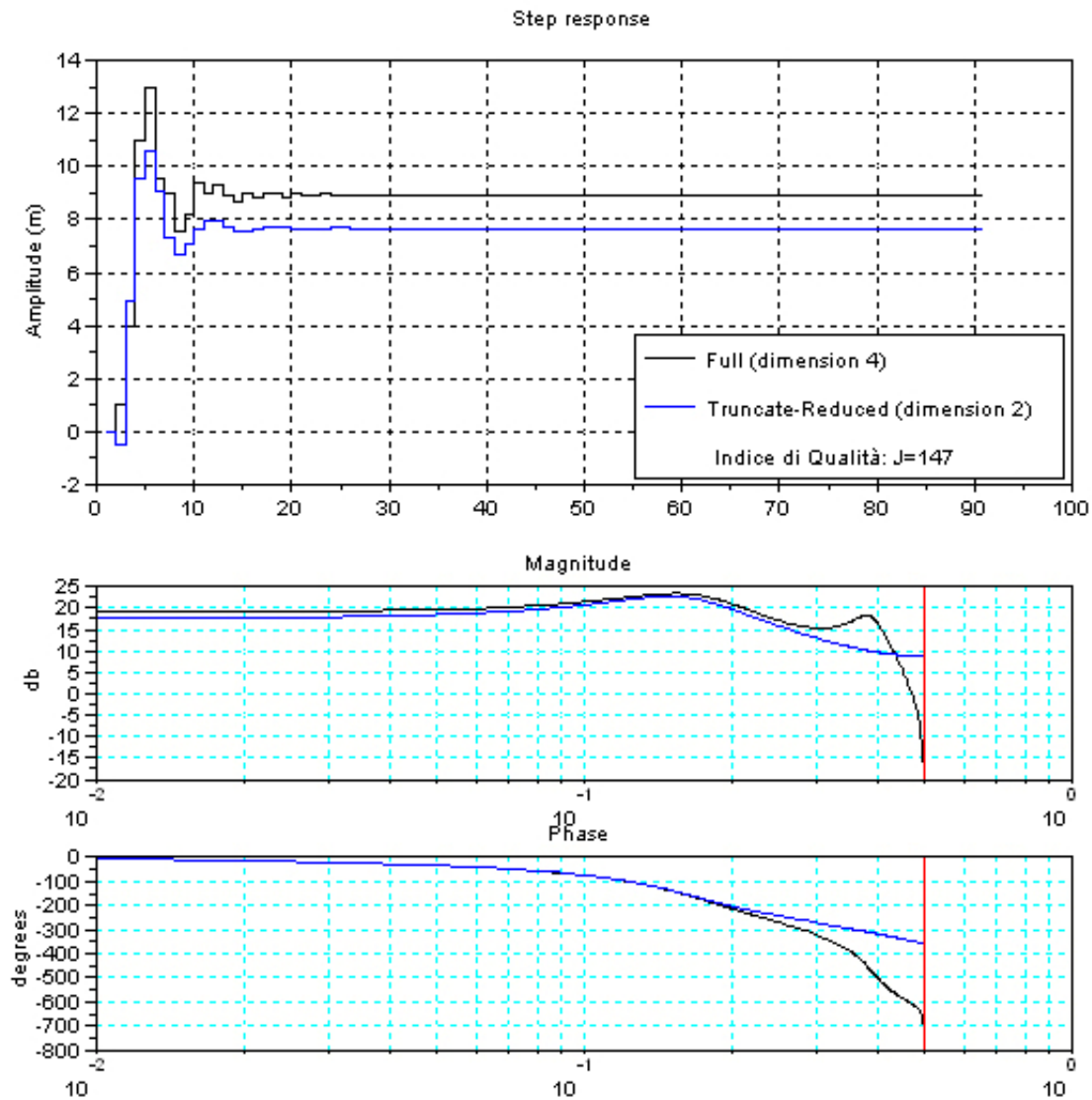
$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -1/3 & -1/9 & -1/4 & -1/2 \end{bmatrix}, \quad B = \begin{bmatrix} 1 \\ 3 \\ 7 \\ 2 \end{bmatrix}, \quad C = [1 \quad 0 \quad 0 \quad 0], \quad D = 0$$

Tempo di campionamento: 0.1 secondi

Sono di seguito riportate la risposta al gradino e la risposta in frequenza del sistema completo:



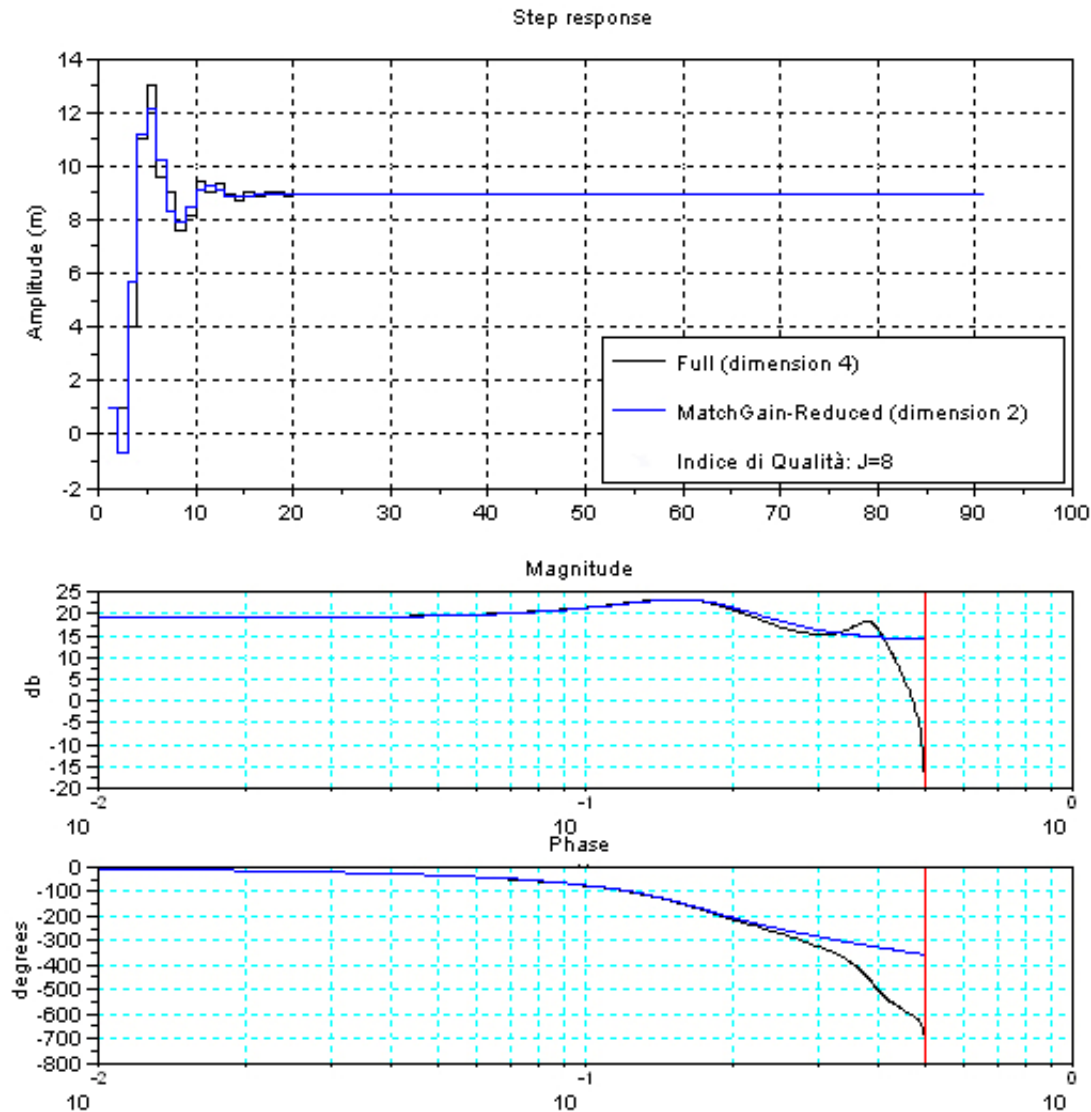
### 5.1.1 Riduzione ad ordine 2 con il metodo Truncate



Come si nota dai grafici delle risposte, l'approssimazione non è molto efficiente, poiché il sistema ridotto ha un comportamento molto diverso da quello intero. Come ci si aspettava, inoltre, il guadagno del sistema ridotto è diverso da quello del sistema originale.

L'indice di qualità è molto alto ( $J = 147$ ), poiché l'errore a regime del sistema non è nullo.

### 5.1.2 Riduzione ad ordine 2 con il metodo Match Gain



Utilizzando il metodo di riduzione *Match Gain* l'approssimazione è decisamente migliore rispetto al caso precedente. Il guadagno, infatti, è perfettamente conservato e la risposta al gradino nel sistema ridotto è molto simile a quella nel sistema intero. La risposta in frequenza, tuttavia, mostra grandi disparità per frequenze superiori ad  $1/10$  Hz.

L'indice di qualità è molto ridotto ( $J = 8$ ) ad indicare un'ottima approssimazione nella risposta al gradino unitario.

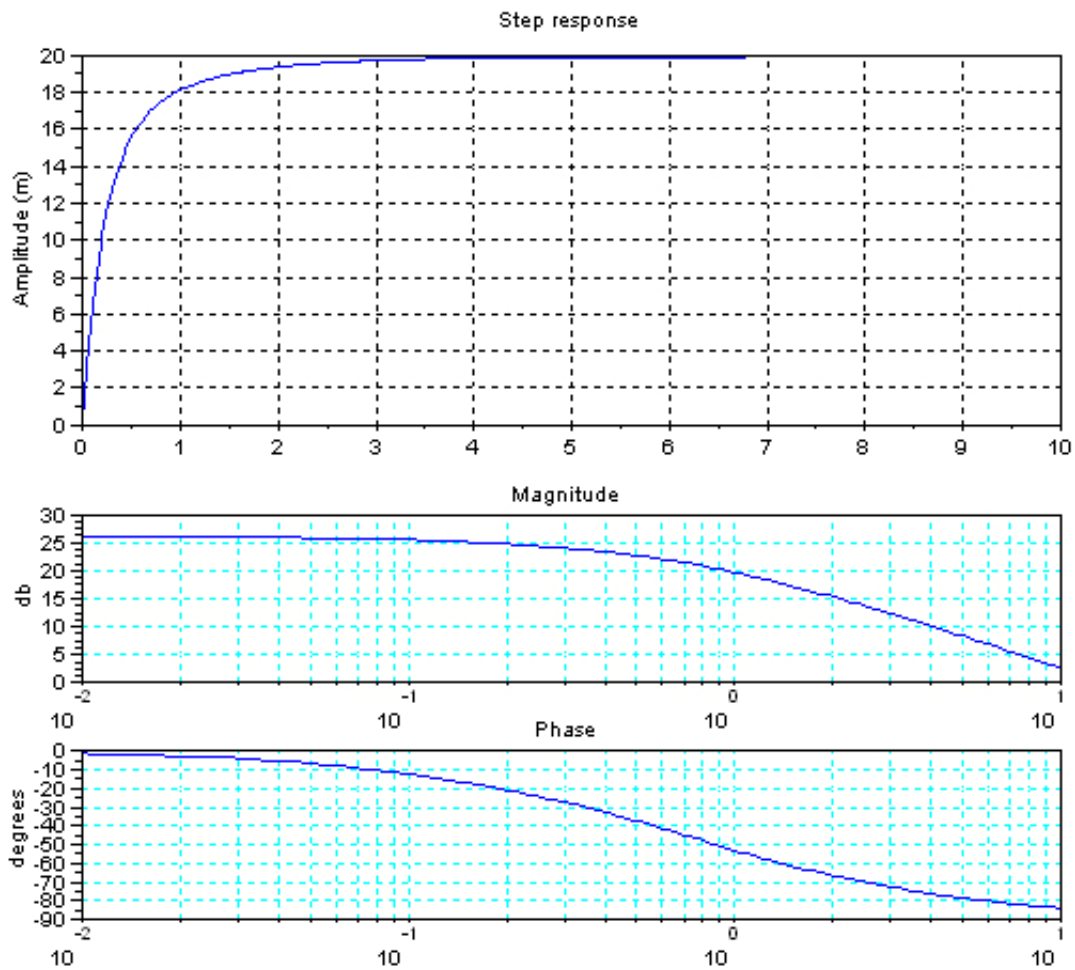
Da notare che, nonostante l'ordine del sistema ridotto sia dimezzato rispetto all'ordine del sistema di partenza, per basse frequenze l'approssimazione è molto fedele, e ciò a dimostrare l'efficacia dell'algoritmo utilizzato.

## 5.2 Esempio di riduzione su modello tempo-continuo stabile del 4° ordine

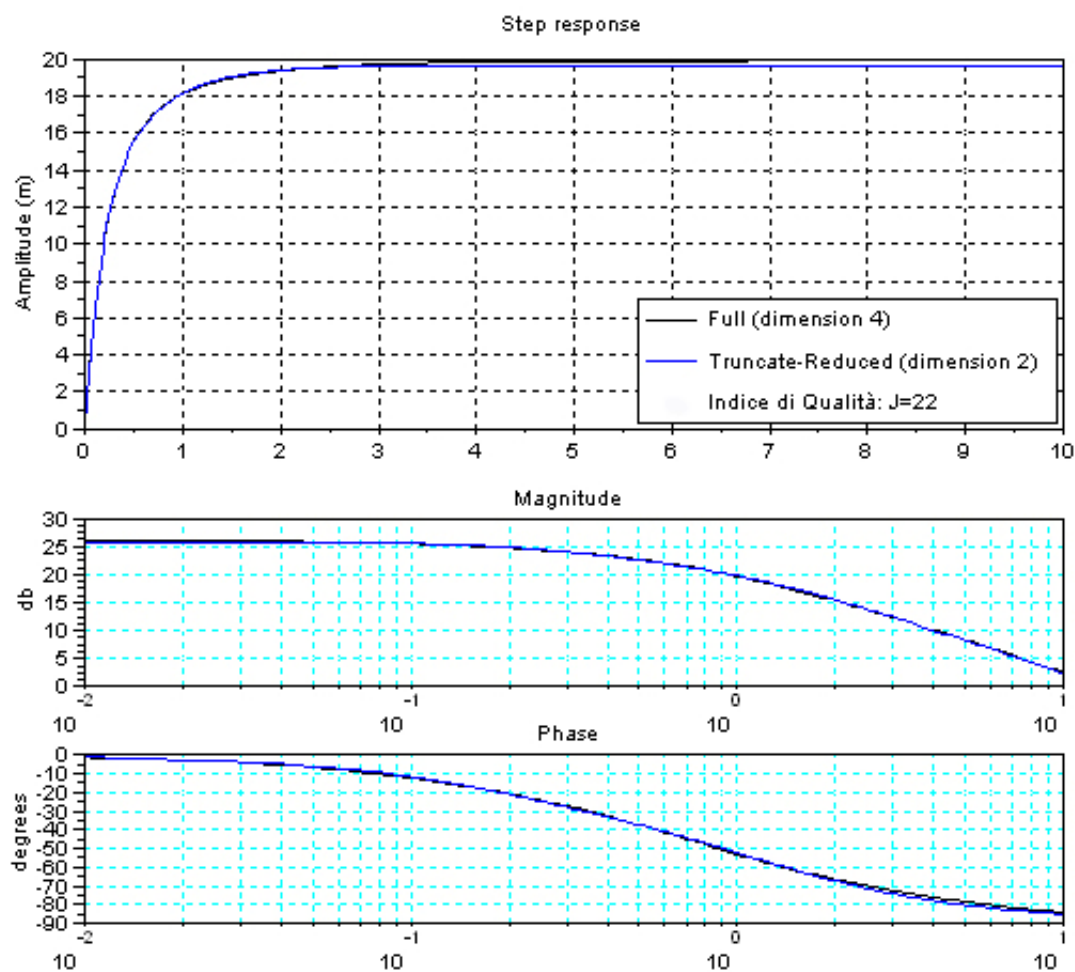
Consideriamo il seguente sistema tempo-continuo stabile del 4° ordine descritto nello spazio degli stati:

$$A = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & -3 & 0 & 0 \\ 0 & 0 & -5 & 0 \\ 0 & 0 & 0 & -16 \end{bmatrix}, \quad B = \begin{bmatrix} 1 \\ 3 \\ 7 \\ 2 \end{bmatrix}, \quad C = [4 \quad 5 \quad 7 \quad 9], \quad D = 0$$

Sono di seguito riportate la risposta al gradino e la risposta in frequenza del sistema completo:



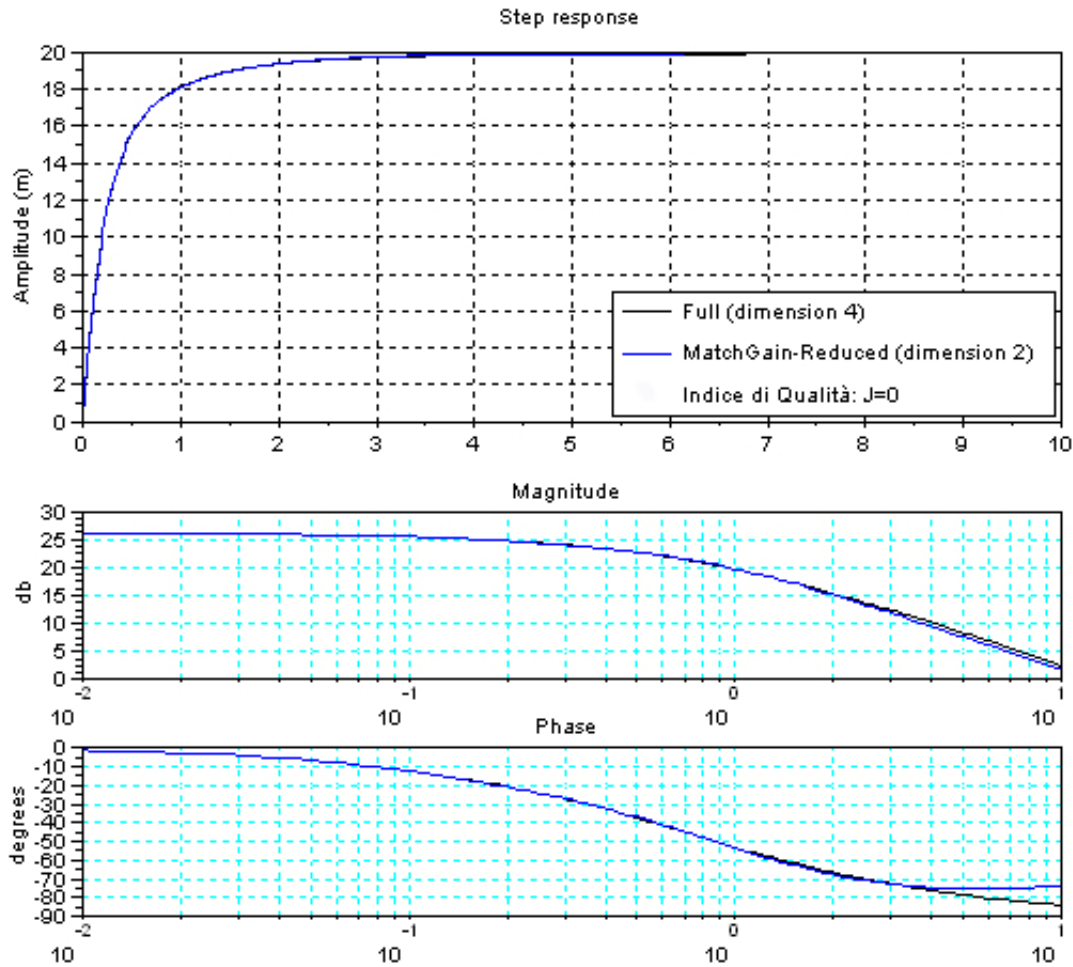
### 5.2.1 Riduzione ad ordine 2 con il metodo Truncate



Come si nota dai grafici delle risposte, l'approssimazione non è molto efficiente, poiché il sistema ridotto ha un comportamento molto diverso da quello intero. Come ci si aspettava, inoltre, il guadagno del sistema ridotto è diverso da quello del sistema originale.

L'indice di qualità è abbastanza alto ( $J = 22$ ), poiché l'errore a regime del sistema non è nullo.

### 5.2.2 Riduzione ad ordine 2 con il metodo Match Gain



Utilizzando il metodo di riduzione *Match Gain* l'approssimazione è decisamente migliore rispetto al caso precedente. Il guadagno, infatti, è perfettamente conservato e la risposta al gradino nel sistema ridotto è identica a quella nel sistema intero. La risposta in frequenza, tuttavia, è fedele fino a frequenze di poco inferiori ai 10 Hz.

L'indice di qualità è praticamente nullo ( $J = 0$ ) ad indicare un'ottima approssimazione nella risposta al gradino unitario.

Da notare che, nonostante l'ordine del sistema ridotto sia dimezzato rispetto all'ordine del sistema di partenza, per basse frequenze l'approssimazione è molto fedele, e ciò a dimostrare l'efficacia dell'algoritmo utilizzato.

### 5.3 Esempio di riduzione su modello tempo-discreto instabile del 4° ordine

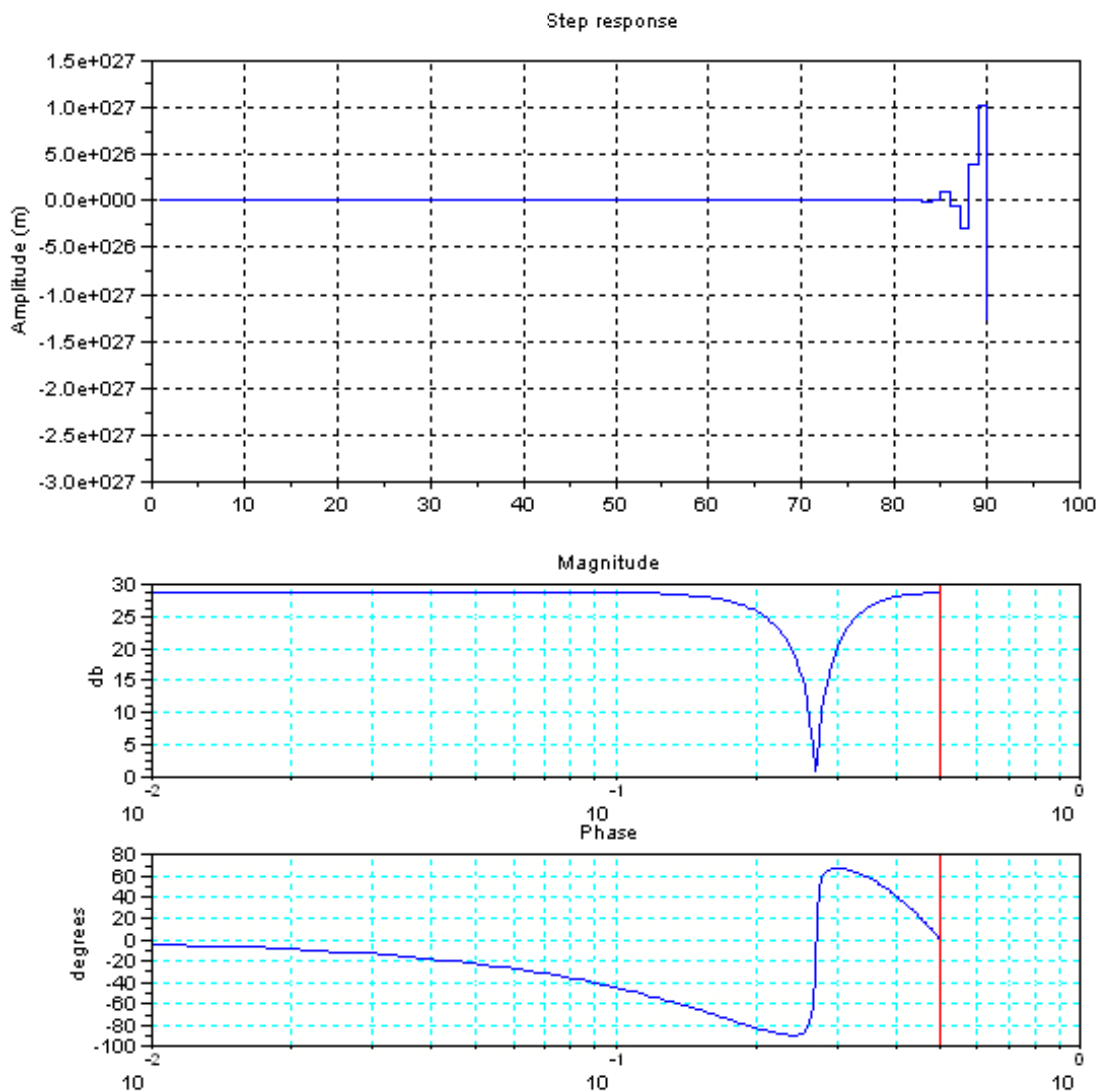
Consideriamo il seguente sistema tempo-discreto instabile del 4° ordine descritto nello spazio degli stati:

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -1/3 & -1/9 & -4 & -1/2 \end{bmatrix}, \quad B = \begin{bmatrix} 1 \\ 3 \\ 7 \\ 2 \end{bmatrix}, \quad C = [1 \quad 0 \quad 0 \quad 0], \quad D = 0$$

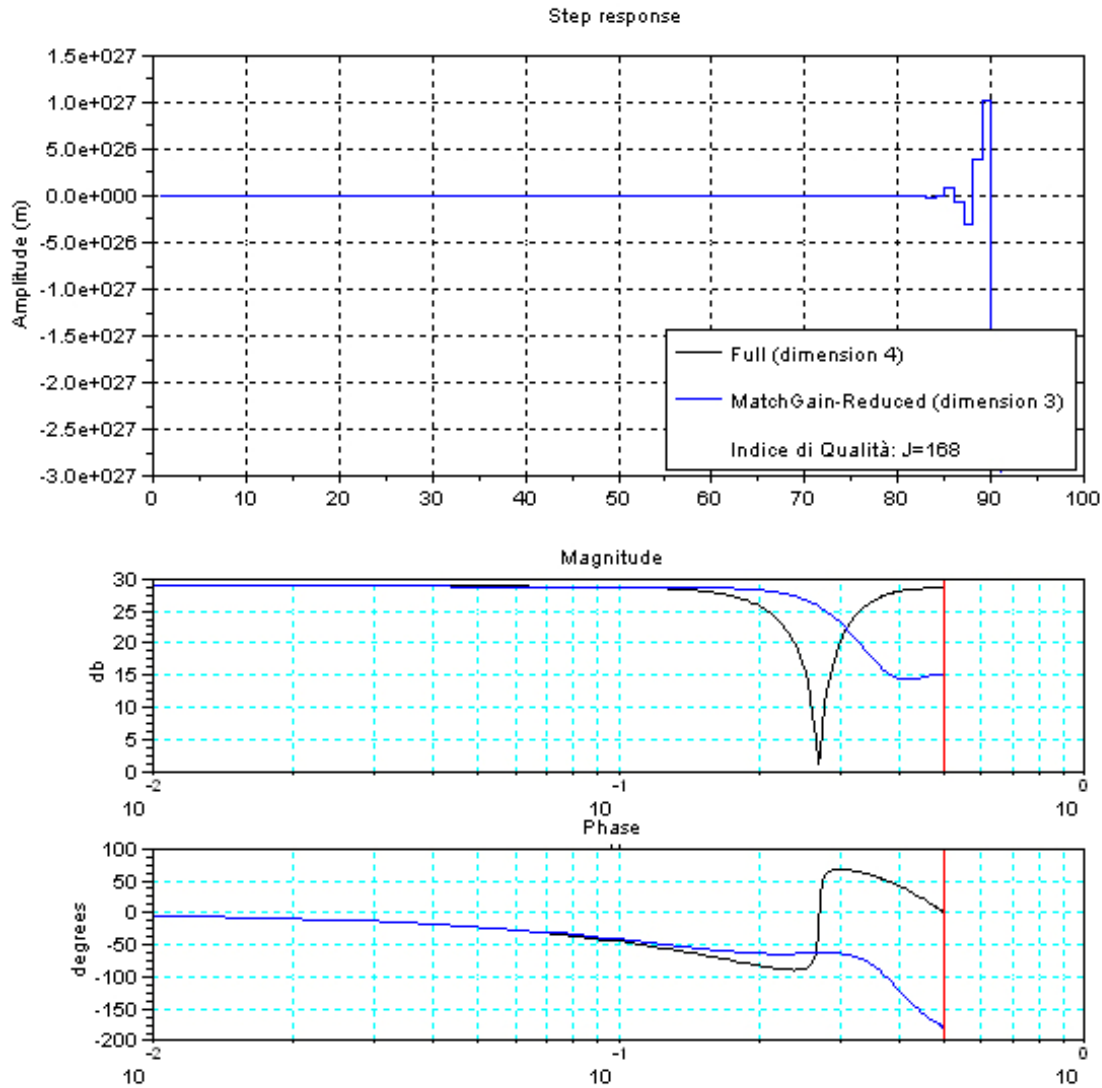
Tempo di campionamento: *0.1 secondi*

Il sistema presenta 2 stati stabili e 2 stati instabili. Nella procedura di riduzione entrambi gli stati instabili verranno conservati intatti, come chiarito nel paragrafo 1.1.

Sono di seguito riportate la risposta al gradino e la risposta in frequenza del sistema completo:



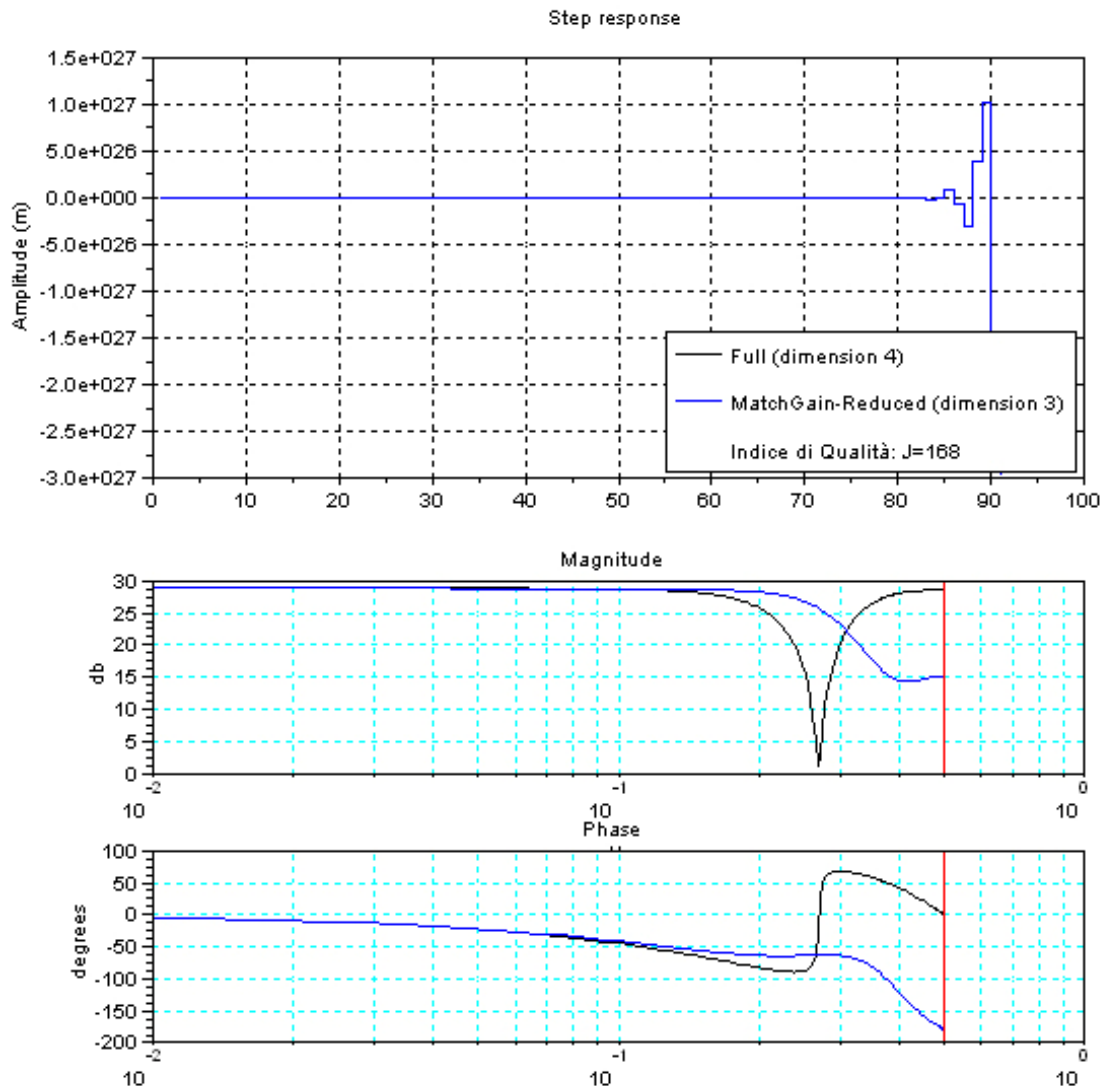
### 5.3.1 Riduzione ad ordine 3 con il metodo Truncate



Come si nota dai grafici delle risposte, l'approssimazione non è molto efficiente, poiché il sistema ridotto ha un comportamento abbastanza diverso da quello intero. Come ci si aspettava, inoltre, il guadagno del sistema ridotto è diverso da quello del sistema originale.

L'indice di qualità non è significativo, poiché la risposta al gradino diverge all'infinito e l'errore è destinato ad aumentare all'infinito.

### 5.3.2 Riduzione ad ordine 3 con il metodo Match Gain



Utilizzando il metodo di riduzione *Match Gain* l'approssimazione è decisamente migliore rispetto al caso precedente. Il guadagno, infatti, è perfettamente conservato e la risposta al gradino nel sistema ridotto è molto simile a quella nel sistema intero. La risposta in frequenza, tuttavia, mostra grandi disparità per frequenze superiori ad  $1/10$  Hz.

L'indice di qualità non è significativo, poiché la risposta al gradino diverge all'infinito e l'errore è destinato ad aumentare all'infinito.

Da notare che, nonostante l'ordine del sistema ridotto sia dimezzato rispetto all'ordine del sistema di partenza, per basse frequenze l'approssimazione è molto fedele, e ciò a dimostrare l'efficacia dell'algoritmo utilizzato anche per sistemi instabili.

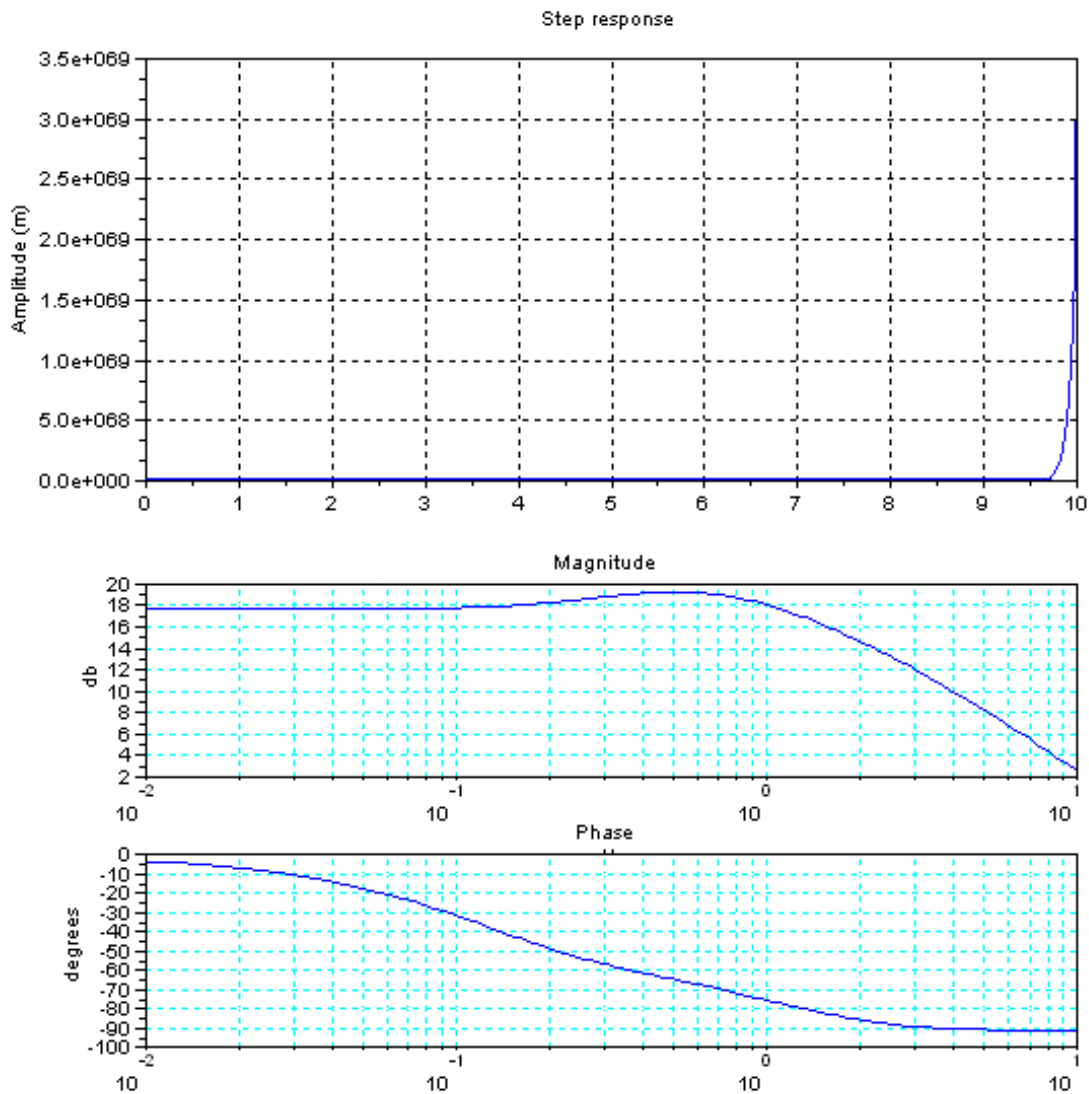
## 5.4 Esempio di riduzione su modello tempo-continuo instabile del 4° ordine

Consideriamo il seguente sistema tempo-continuo instabile del 4° ordine descritto nello spazio degli stati:

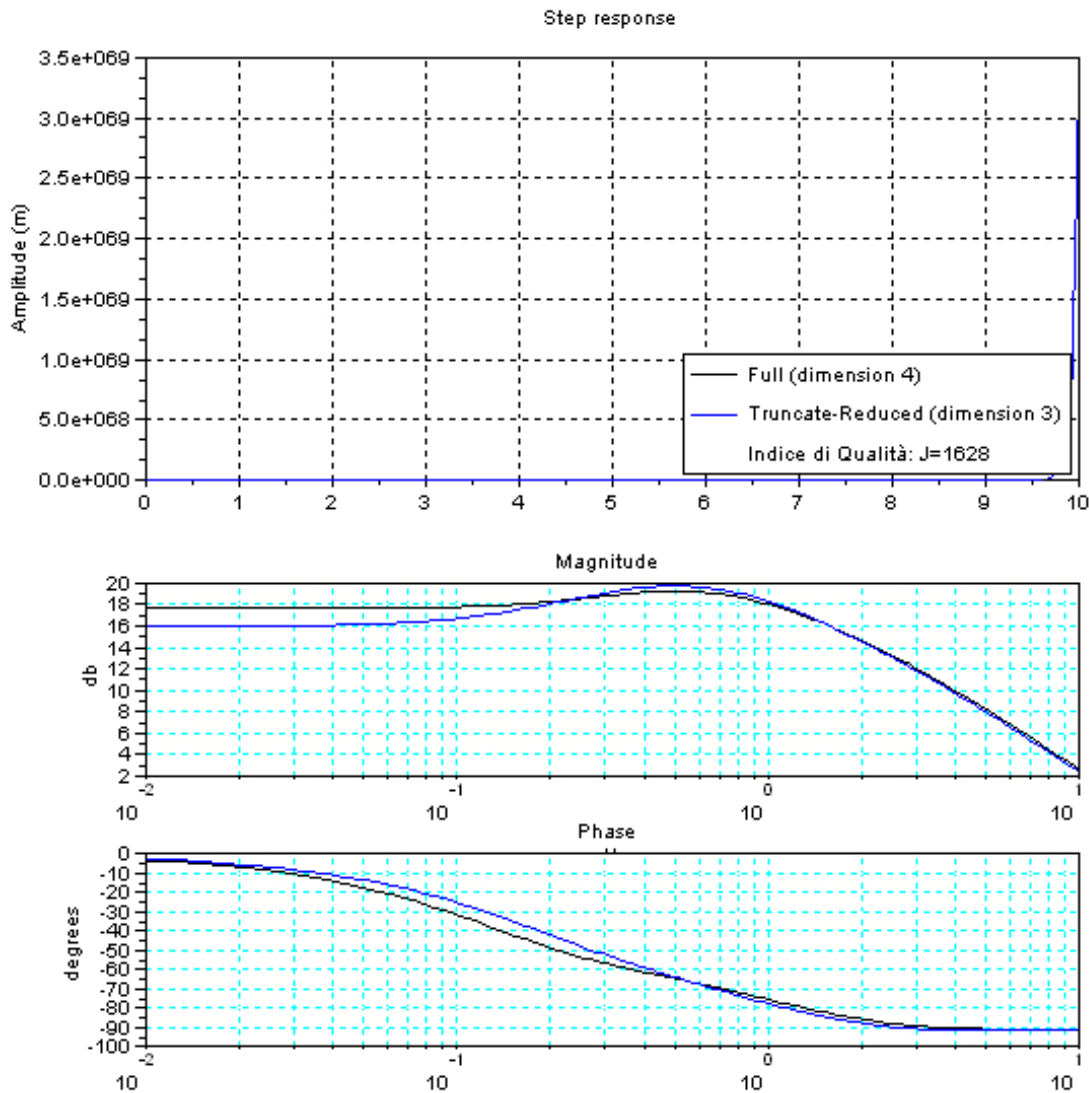
$$A = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & -5 & 0 \\ 0 & 0 & 0 & 16 \end{bmatrix}, \quad B = \begin{bmatrix} 1 \\ 3 \\ 7 \\ 2 \end{bmatrix}, \quad C = [4 \quad 5 \quad 7 \quad 9], \quad D = 0$$

Il sistema presenta 2 stati stabili e 2 stati instabili. Nella procedura di riduzione entrambi gli stati instabili verranno conservati intatti, come chiarito nel paragrafo 1.1.

Sono di seguito riportate la risposta al gradino e la risposta in frequenza del sistema completo:

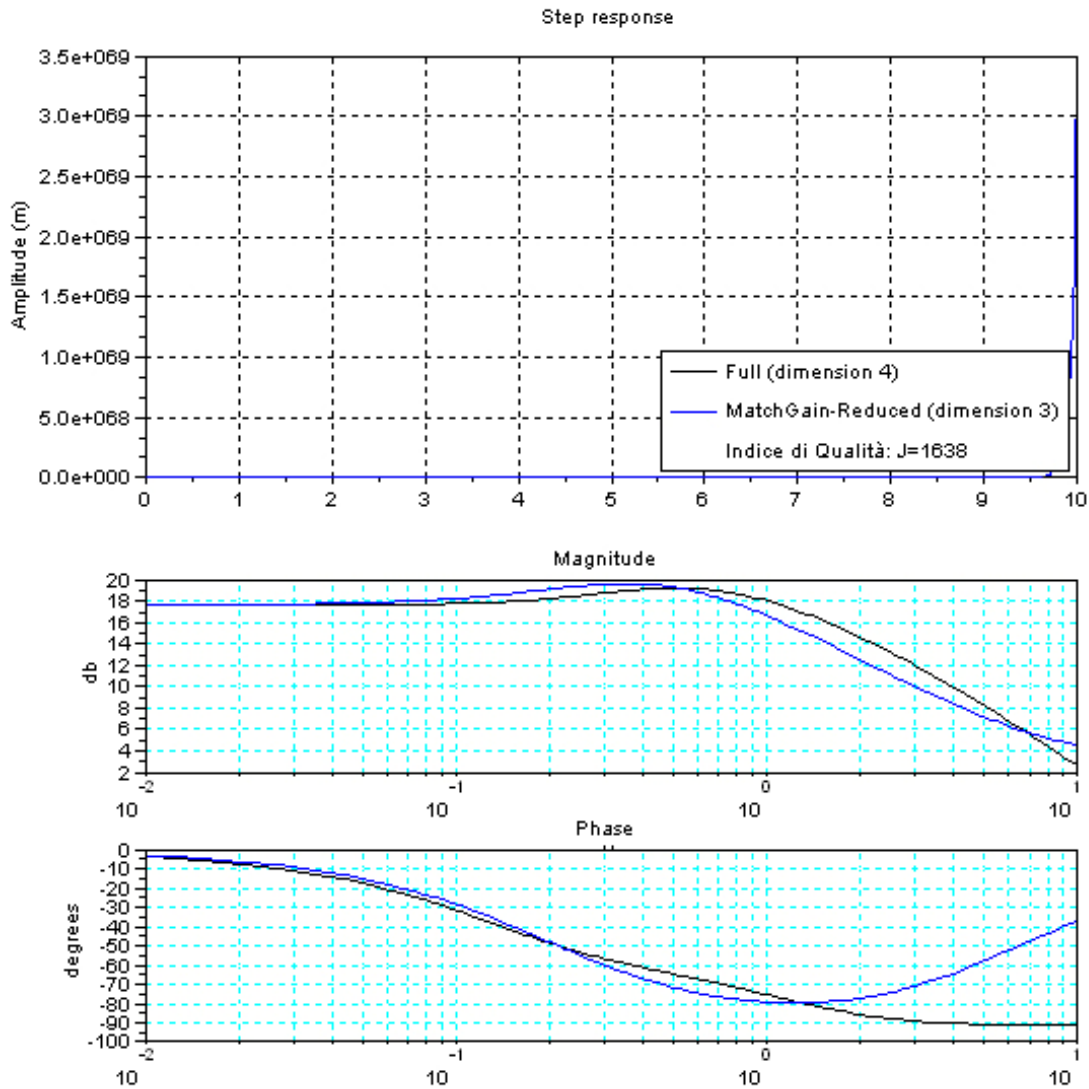


### 5.4.1 Riduzione ad ordine 3 con il metodo Truncate



Come si nota dai grafici delle risposte, l'approssimazione non è molto efficiente, poiché il sistema ridotto ha un comportamento abbastanza diverso da quello intero. Come ci si aspettava, inoltre, il guadagno del sistema ridotto è diverso da quello del sistema originale. L'indice di qualità non è significativo, poiché la risposta al gradino diverge all'infinito e l'errore è destinato ad aumentare all'infinito.

### 5.4.2 Riduzione ad ordine 3 con il metodo Match Gain



Utilizzando il metodo di riduzione *Match Gain* l'approssimazione è decisamente migliore rispetto al caso precedente. Il guadagno, infatti, è perfettamente conservato e la risposta al gradino nel sistema ridotto è molto simile a quella nel sistema intero. La risposta in frequenza, tuttavia, mostra grandi disparità per frequenze superiori ad 1 Hz.

L'indice di qualità non è significativo, poiché la risposta al gradino diverge all'infinito e l'errore è destinato ad aumentare all'infinito.

Da notare che, nonostante l'ordine del sistema ridotto sia dimezzato rispetto all'ordine del sistema di partenza, per basse frequenze l'approssimazione è molto fedele, e ciò a dimostrare l'efficacia dell'algoritmo utilizzato anche per sistemi instabili.

## 6. Appendice: codice sorgente routine Scilab®

È di seguito presentato il codice sorgente delle routine Scilab® utilizzate nel toolbox Model Order Reduction.

### 6.1 mor.sci

```
//Funzione principale del toolbox
function [slbr,slb] = mor();

// Display mode
mode(0);

// Display warning for floating point exception
ieee(1);

//Faccio scegliere all'utente se vuole lavorare in tempo-continuo o in tempo-
discreto
tipo=buttndialog("Specifica il tipo del sistema?","Continuo|Discreto");

//Controllo se l'utente ha scelto il tempo continuo o il tempo discreto;
if (tipo == '1')
    tipo = 'c';
    nome = 'continuo';
else
    tipo = 'd';
    nome = 'discreto';
end;

//Leggo i parametri del sistema continuo
if (tipo == 'c')
    txt=['A';'B';'C';'D'];
    sig=x_mdialog('Inserisci i parametri del sistema tempo-
'+nome,txt,['A';'B';'C';'D']);
    A=evstr(sig(1));
    B=evstr(sig(2));
    C=evstr(sig(3));
    D=evstr(sig(4));

//Leggo i parametri del sistema discreto
else
    txt=['A';'B';'C';'D';'Tempo di campionamento'];
    sig=x_mdialog('Inserisci i parametri del sistema tempo-
'+nome,txt,['A';'B';'C';'D';'.1']);
    A=evstr(sig(1));
    B=evstr(sig(2));
    C=evstr(sig(3));
    D=evstr(sig(4));
    tc=evstr(sig(5));
end;

if (rank(cont_mat(A,B)) < size(A,1))
    buttndialog("Attenzione: il sistema non è completamente
controllabile.", "OK", "error");
    return;
end
```

```

if (rank(observ_mat(A,C)) < size(A,1))
    buttndialog("Attenzione: il sistema non è completamente
osservabile.", "OK", "error");
    return;
end

//Creo il sistema
sl=syslin(tipo,A,B,C,D);

//Estraggo il sottosistema stabile
[sl_unstable,sl_stable] = dtsi_mod(sl);

//Calcolo del dimensioni del sistema completo (parte stabile + parte instabile)
TotalModes = size(sl.a,1);
if (sl_stable == 0)
    StableModes = 0;
else
    StableModes = size(sl_stable.a,1);
end

UnstableModes = TotalModes - StableModes;

if (StableModes == 0)
    buttndialog("Il sistema contiene solo modi instabili. Impossibile bilanciare
un sistema completamente instabile!", "OK", "error");
    return;
end

if (UnstableModes > 0)
    if (UnstableModes == 1)
        msg = 'Attenzione: il sistema contiene 1 modo instabile.';
    else
        msg = sprintf('Attenzione: il sistema contiene %i modi
instabili.',UnstableModes);
    end
    buttndialog(msg,"OK", "warning");
end

//Bilancio il sottosistema stabile
slbS=balreal(sl_stable);

FirstStableMode = UnstableModes+1;

//Inizializzo a 0 il sistema bilanciato finale
slb = sl;
slb.a = zeros(TotalModes,TotalModes);
slb.b = zeros(TotalModes,1);
slb.c = zeros(1,TotalModes);

//Aggiungo i modi stabili al sistema bilanciato
if(StableModes > 0)
    tmp = slb.a;
    tmp(FirstStableMode:TotalModes,FirstStableMode:TotalModes)=slbS.a;
    slb.a = tmp;

    tmp = slb.b;
    tmp(FirstStableMode:TotalModes)=slbS.b;
    slb.b = tmp;

    tmp = slb.c;
    tmp(FirstStableMode:TotalModes)=slbS.c;
    slb.c = tmp;
end

```

```

//Aggiungo i modi instabili al sistema bilanciato
if(UnstableModes > 0)
    tmp = slb.a;
    tmp(1:UnstableModes,1:UnstableModes)=sl_unstable.a;
    slb.a = tmp;

    tmp = slb.b;
    tmp(1:UnstableModes)=sl_unstable.b;
    slb.b = tmp;

    tmp = slb.c;
    tmp(1:UnstableModes)=sl_unstable.c;
    slb.c = tmp;
end

//Calcolo il gramiano di controllabilità
//E' superfluo il calcolo del gramiano di osservabilità, poichè
//dopo il bilanciamento il gramiano di osservabilità è coincidente con il
gramiano di controllabilità
Wc=clean(ctr_gram(slbS));
//Wo=clean(obs_gram(slbS));

//Estraggo gli autovalori dal gramiano ed inserisco Inf per indicare i modi
instabili
Gc(1:TotalModes) = %inf;
Gc(UnstableModes+1:TotalModes) = diag(Wc);

//Converto il vettore Gramiano in vettore di stringhe
L = [];
for i=1:size(Gc,1),
    L(i)=string(Gc(i));
end

//Avvio un ciclo infinito di riduzione per permettere all'utente di eseguire
//tutte le riduzioni che vuole sul modello creato
counter = 0; //Porto il conto di quante riduzioni si effettuano sullo stesso
sistema
while 1
    counter = counter + 1;
    index=x_choose(L,['Seleziona il valore sotto il quale effettuare il taglio (i
valori ad Inf corrispondono a modi instabili).']);

    //Controllo se l'utente ha selezionato il primo autovalore, ovvero vuole
ridurre ad ordine 0 il sistema
    if (index == 1)
        buttondialog("Impossibile ridurre il sistema ad ordine 0!","OK","error");
        counter = counter - 1;
        continue;
    end;

    //Controllo se l'utente ha selezionato un modo instabile
    if (index > 0 & L(index) == 'Inf')
        buttondialog("Impossibile selezionare un modo instabile come valore di
taglio!","OK","error");
        counter = counter - 1;
        continue;
    end;
end;

```

```

//Controllo se l'utente non ha selezionato alcun valore di taglio
if (index == 0)
    choise = buttndialog("Non è stato selezionato il valore di taglio. Se si
sceglie di continuare il modello non verrà
ridotto!", "Continua|Indietro", "warning");
    if (choise == '1')
        soglia = -1;
    else
        counter = counter - 1;
        continue;
    end
end

//Faccio scegliere all'utente il metodo di riduzione
method=buttndialog("Specifica il metodo di riduzione?", "Truncate|Match
Gain");
if (method == '1')
    method = 'Truncate';
else
    method = 'MatchGain';
end;

if (index > 0)
    //Identifico il valore di soglia.
    soglia = Gc(index);
end;

elim = (Gc <= soglia);

//Effettuo la riduzione
slbr = modred(slb,elim,method);

//Stampo il grafico della risposta al gradino
if (tipo == 'c') //Tempo continuo
    t=linspace(0,10,500);
    stepresp=csim('step',t,slb);
    stepresp_rid=csim('step',t,slbr);
    xset('window',counter); xbascc(); // create and clear window for subplots
    subplot(2,1,1);
    plot2d(t,stepresp, 1); xgrid();
    plot2d(t,stepresp_rid, 2);
    xtitle('Step response',' ','Amplitude (m)');

else //Tempo discreto
    u = ones(1:tc:10);
    stepresp=flts(u,slb);
    stepresp_rid=flts(u,slbr);
    xset('window',counter); xbascc(); // create and clear window for subplots
    subplot(2,1,1);
    plot2d2("onn",(1:size(u,'c'))',stepresp',1); xgrid();
    plot2d2("onn",(1:size(u,'c'))',stepresp_rid',2);
    xtitle('Step response',' ','Amplitude (m)');
end;

//Calcolo l'indice di qualità
J=0;
tmp=0;
for i=1:size(stepresp,2),
    tmp = stepresp_rid(i) - stepresp(i);
    J = J + tmp^2;
end

```

```

J = iconvert(J,14);    //Converto in intero il fattore di qualità

//Stampo la legenda
FullOrder = sprintf('Full (dimension %i)',size(Gc));
if (index == 0) //Controllo se il sistema non è stato ridotto
    index = size(Gc) + 1;
end
ReducedOrder = sprintf('%s-Reduced (dimension %i)',method,index - 1);
Js = string(J);
Quality = sprintf('Indice di Qualità: J=%s',Js);
legends([FullOrder;ReducedOrder;Quality],[[1;1],[2;1],[0;0]], opt="lr");

//Stampo la risposta in frequenza
subplot(2,1,2);
bode([slb;slbr],0.01,10);

//Chiedo se si vuole effettuare una nuova riduzione sul sistema
choise=buttondialog("Vuoi effettuare una nuova riduzione su questo
sistema?","Si|No");
if (choise == '2')
    break;
end

end

endfunction

```

## 6.2 modred.sci

```
// Funzione con gli algoritmi di riduzione
function [Dr] = modred(DD,elim,method)

// Output variables initialisation (not found in input variables)
Dr=[];

// Display mode
mode(0);

// Display warning for floating point exception
ieee(1);

Dr = DD;
ns = size(DD.A,1);
if ns==0 then
    return;
end;
keep = 1:ns;

//Seleziono gli indici da conservare
L = [];
j=0;
for i=1:ns,
    if (elim(i) == %F)
        j=j+1;
        L(j)=i;
    end
end

keep=L;

//Seleziono gli indici da eliminare
L = [];
j=0;
for i=1:ns,
    if (elim(i) == %T)
        j=j+1;
        L(j)=i;
    end
end

nokeep=L;

a=DD.A;
b=DD.B;
c=DD.C;
d=DD.D;

select method
case "MatchGain" then
    // Match gain: divido in x1, da conservare, e x2, da eliminare
    a11 = a(keep,keep);
    a12 = a(keep,nokeep);
    a21 = a(nokeep,keep);
    a22 = a(nokeep,nokeep);
    b1 = b(keep,:);
    b2 = b(nokeep,:);
    c1 = c(:,keep);
    c2 = c(:,nokeep);
    n2 = size(nokeep,1);
```

```

// Creo le matrici ridotte
tolsing = %eps^0.75;
if (DD(7) == 'd')
    // Sistema tempo-discreto
    a22 = a22 - eye(n2,n2);
end;
[l,u,p] = lu(a22);
norml1 = norm(a11,1);
%v03 = bool2s(%f);
if bool2s(norm(a22,1)>(tolsing*norml1)) then %v03 =
bool2s(mtlb_rcond(u)>tolsing);end;
if %v03 then

    A21 = u\ (l\ (p*a21));
    B2 = u\ (l\ (p*b2));
    ar = a11 - a12 * A21;
    br = b1 - a12 * B2;
    cr = c1 - c2 * A21;
    dr = d - c2 * B2;
else

    [u,s,v] = svd(a22);
    s = diag(s(1:n2,1:n2));
    nnz = sum(bool2s(mtlb_logic(s,">",tolsing*mtlb_max([s;norml1],"m"))),"m");
    u = u(:,mtlb_imp(1,nnz));
    v = v(:,mtlb_imp(1,nnz));
    s = s(mtlb_imp(1,nnz));
    A21 = (diag(1 ./s)*u')*a21;
    B2 = (diag(1 ./s)*u')*b2;
    A12 = a12 * v;
    C2 = c2 * v;
    ar = a11 - A12 * A21;
    br = b1 - A12 * B2;
    cr = c1 - C2 * A21;
    dr = d - C2 * B2;
end;

case "Truncate" then
    // Elimino gli stati selezionati
    ar = a(keep,keep);
    br = b(keep,:);
    cr = c(:,keep);
    dr = d;
end;

// Preparo le matrici da mandare in uscita
Dr.A = ar;
Dr.B = br;
Dr.C = cr;
Dr.D = dr;
Dr.X0 = zeros(size(keep,1),1);
endfunction

```

## 6.3 dtsti\_mod.sci

```
//Funzione per la decomposizioni in stati stabili/instabili
function [ga,gs]=dtsti_mod(g)

tipo=g(7);
if (tipo == 'c')
    [a,b,c,d]=g(2:5),
    [n1,n2,t]=size(g),
    [a,u0]=balanc(a);b=u0\b;c=c*u0;
    [u,n]=schur(a,'c'),
    a=u'*a*u,
    if n==t then ga=0,
        gs=g,return,
    end,
    if n==0 then gs=0,
        ga=g,return,
    end,
    //      [ab,w,bs]=bdiag(a);
    a1=a(1:n,1:n),a4=a(n+1:t,n+1:t),x=a(1:n,n+1:t),
    z=sylv(a1,-a4,-x,'c'),
    w=[eye(n,n),z;0*ones(t-n,n),eye(t-n,t-n)],
    wi=[eye(n,n),-z;0*ones(t-n,n),eye(t-n,t-n)],
    tr=u*w,tri=wi*u';
    bb=tri*b,b1=bb(1:n,:),b2=bb(n+1:t,:),
    cc=c*tr,c1=cc(:,1:n),c2=cc(:,n+1:t),
    ga=syslin('c',a4,b2,c2),
    gs=syslin('c',a1,b1,c1);
end

if (tipo == 'd')

    [a,b,c,d]=g(2:5),
    [n1,n2,t]=size(g),
    [a,u0]=balanc(a);b=u0\b;c=c*u0;
    [u,n]=schur(a,'d'),
    a=u'*a*u,
    if n==t then ga=0,
        gs=g,return,
    end,
    if n==0 then gs=0,
        ga=g,return,
    end,
    //      [ab,w,bs]=bdiag(a);
    a1=a(1:n,1:n),a4=a(n+1:t,n+1:t),x=a(1:n,n+1:t),
    z=sylv(a1,-a4,-x,'d'),
    w=[eye(n,n),z;0*ones(t-n,n),eye(t-n,t-n)],
    wi=[eye(n,n),-z;0*ones(t-n,n),eye(t-n,t-n)],
    tr=u*w,tri=wi*u';
    bb=tri*b,b1=bb(1:n,:),b2=bb(n+1:t,:),
    cc=c*tr,c1=cc(:,1:n),c2=cc(:,n+1:t),
    ga=syslin('d',a4,b2,c2),
    gs=syslin('d',a1,b1,c1);

end

endfunction
```

## 7. Bibliografia

- [1] Aldo Balestrino, Appunti di "Controllo Digitale", Università di Pisa, 2006.
- [2] "Model Order Reduction" Wiki, MIT, 2007.  
[<http://scripts.mit.edu/~mor/wiki>]
- [3] Jacob White, "Model Order Reduction", MIT, 2003.  
[<http://web.mit.edu/mor/papers/6336/lec24.pdf>]  
[<http://web.mit.edu/mor/papers/6336/lec25.pdf>]
- [4] Alberto Bemporad, "Model Reduction", Università di Siena, 2006.  
[[http://www.dii.unisi.it/~bemporad/teaching/controllo digitale/pdf/13-model\\_reduction.pdf](http://www.dii.unisi.it/~bemporad/teaching/controllo digitale/pdf/13-model_reduction.pdf)]
- [5] Paolo Mantegazza, "Riduzione Bilanciata dei Modelli", Politecnico di Milano, 2006.  
[<http://www.aero.polimi.it/~mante/bacheca/DinamicaEControllo/bigino12.pdf>]
- [6] A.J. Laub, M.T. Heath, C.C. Paige, and R.C. Ward, "Computation of System Balancing Transformations and Other Applications of Simultaneous Diagonalization Algorithms," *IEEE Trans. Automatic Control*, AC-32 (1987), pp. 115-122.
- [7] P. Benner, M. Castillo, E. S. Quintana, G. Quintana, ParCo2003, "Parallel Model Reduction of Large-Scale Unstable Systems", Dresda (Germania), 2003
- [8] Scilab User's Guide, INRIA, 2006.  
[<http://www.scilab.org/product/man-eng/index.html>]
- [9] C. Bonivento, C. Melchiorri, R. Zanasi, "Sistemi di Controllo Digitale", Editrice Esculapio, Bologna, 1995.
- [10] Andras Varga, "Model reduction software in the SLICOT library" in "Applied and Computational Control, Signals, and Circuits, Ed. B. Datta", Kluwer Academic Publishers, Boston, 2001.