Teoria dei sistemi e del controllo

LM in Ingegneria Informatica e Ingegneria Elettronica

(http://www.dii.unimore.it/~lbiagiotti/TeoriaSistemiControllo.html)

Esercitazione numero 3

Avvio di Matlab

La prova pratica viene svolta in ambiente Linux. Per accedere al programma Matlab e creare i propri file di lavoro (che dovranno essere inclusi dentro la stessa directory `cognome.nome`) eseguire la seguente procedura:

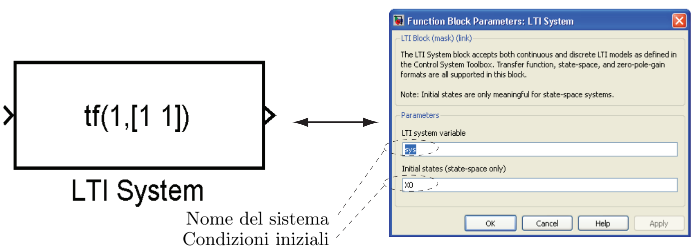
1. Accedere al pc utilizzando le seguenti username e password (sono quelle per accedere alla propria e-mail di ateneo):
   Username: <numero di tessera dello studente>
   Password: <password e-mail dello studente>
2. Sulla barra in alto, cliccare su `Applications`, quindi da `Accessories` selezionare `Terminal`
3. Nella propria home creare la propria directory di lavoro locale ed entrarvi con i comandi
   ```
   mkdir cognome.nome
   cd cognome.nome
   ```
4. Aprire il programma Matlab con il comando `matlab2006b`
5. Svolgere la prova chiamando il programma principale `prova.m` (nella prima riga del file prova.m specificare il proprio nome e cognome, opportunamente commentati)

NOTA BENE. Al termine della prova chiudere Matlab e attendere che la prova sia salvata dal docente. E' possibile (anzi è consigliabile) effettuare un backup della prova stessa copiandola sulla propria chiavetta o spedendola via mail al proprio indirizzo di posta.

Istruzioni e schemi Matlab/Simulink utili per questo esercizio

definizione di una funzione di trasferimento tempo-continua:	`Gs = tf(B,A)` dove `B` e `A` sono rispettivamente i vettori che definiscono i polinomi a numeratore e denominatore. La funzione `tf` invocata con un solo parametro che rappresenta un sistema nello spazio degli stati (`Gs = tf(Sys)`) consente di trasformare il sistema nella corrispondente funzione di trasferimento
definizione di una funzione di trasferimento tempo-discreta:	`Gz = tf(B,A,T)` dove `B` e `A` sono rispettivamente i vettori che definiscono i polinomi a numeratore e denominatore e `T` è il periodo di campionamento
realizzazione di una funzione di trasferimento:	`Sysd = ss(Gz)` (ovvero `Sys = ss(Gs)` per sistemi tempo-continui) fornisce una realizzazione di `Gz` (`Gs`) in forma canonica di controllo modificata. Il sistema `Sysd` (`Sys`) è chiaramente definito nello spazio degli stati
discretizzazione di un sistema (espresso come funzione di trasferimento o come sistema nello spazio degli stati):	`Sysd = c2d(Sys,T,'Metodo')` (ovvero `Gz = c2d(Gs,T,'Metodo')` nel caso di funzioni di trasferimento) discretizza il sistema in accordo con il `Metodo` scelto. Di default (quando omesso) vien utilizzato il metodo con ricostruttore di ordine zero 'zoh' (si veda l'help del comando per maggior dettagli).
estrazione delle matrici caratteristiche di un sistema dinamico tempo-discreto:	`[F,G,C,D,T] = ssdata(Gz)`

Blocco LTI (del Control System Toolbox) per l'introduzione in simulink di un sistema definito come funzione di trasferimento mediante il comando `tf` o nello spazio degli stati mediante il comando `ss`:



Testo dell'esercitazione

Si progetti con Matlab un m-file (prova.m) che (eventualmente con l'ausilio di altri m-file e di uno o più schemi Simulink) svolga le operazioni richieste.

Dato il sistema[1] contenuto nel file simulink `PlantModel_mdl.mdl`[2] (si veda la figura sottostante) e disponibile all'url: `http://www.dii.unimore.it/~lbiagiotti/TeoriaSistemiControllo.html`

Sistema Dinamico LTI

1. Definire l'ingresso forzante costituito da una somma di sinusoidi (con un tempo finale $T_f = 100$s, e un tempo di campionamento $T_s = 0.01$s)
$$u(t) = a_1\,\sin(\omega_1 t) + a_2\,\sin(\omega_2 t) + a_3\,\sin(\omega_3 t) + a_4\,\sin(\omega_4 t)$$
con $a_i = 1, i = 1, \ldots, 4$, $\omega_1 = 0.1$rad/s, $\omega_2 = 1$rad/s, $\omega_3 = 10$rad/s, $\omega_4 = 50$rad/s.
NOTA: al fine di simulare il modello, definire il tempo di campionamento con la variabile Matlab `Ts`.

2. Applicare, in simulazione, il segnale $u(t)$ al sistema dinamico e salvare l'andamento dell'uscita $y(t)$ negli istanti $t_k = kT_s$

3. Assumendo che l'ordine n del modello sia pari a 3, costruire la matrice $\mathbf{\Phi} = [\mathbf{\Phi}_1 \;\; \mathbf{\Phi}_2]$ con
$$\mathbf{\Phi}_1 = \begin{bmatrix} y(n) & y(n-1) & \cdots & y(1) \\ y(n+1) & y(n) & \cdots & y(2) \\ y(n+2) & y(n+1) & \cdots & y(3) \\ \vdots & & & \\ y(N-1) & y(N-2) & \cdots & y(N-n) \end{bmatrix}$$
$$\mathbf{\Phi}_2 = \begin{bmatrix} u(n) & u(n-1) & \cdots & u(1) \\ u(n+1) & u(n) & \cdots & u(2) \\ u(n+2) & u(n+1) & \cdots & u(3) \\ \vdots & & & \\ u(N-1) & u(N-2) & \cdots & u(N-n) \end{bmatrix}$$
e il vettore
$$\boldsymbol{y} = [y(n+1),\, y(n+2), \ldots,\, y(N-1),\, y(N)]^T$$

4. Utilizzando la funzione `pinv` di Matlab per il calcolo della pseudo-inversa, risolvere l'equazione in $\boldsymbol{\alpha} = [-a_1,\, -a_2,\, \ldots,\, -a_n,\, b_1,\, b_2,\, \ldots,\, b_n]$
$$\boldsymbol{y} = \mathbf{\Phi}\boldsymbol{\alpha}$$

5. Definire in Matlab la funzione di trasferimento discreta $G(z)$ del sistema identificato, e confrontare, sovrapponendoli nella medesima figura, gli andamenti dell'uscita del sistema identificato e di quello reale considerando un ingresso a gradino unitario applicato all'istante $t = 2$s (durata della simulazione 10 s).

[1] Il sistema considerato corrisponde a $G(s) = \dfrac{101(s+8)}{(s+4)(s^2+2s+101)}$.

[2] Nel caso in cui all'atto dell'apertura del file simulink si riceva un messaggio di errore, ad esempio in ambiente Linux, eseguire il comando

```
>> bdclose all; slCharacterEncoding('windows-1252')
```