Progetto Corso Filtraggio e Identificazione per il Controllo

Nicola Corea , matricola : 217069

Analisi del Modello

Per lo svolgimento del progetto si è scelto il modello matematico del pendolo, sistema meccanico non lineare.

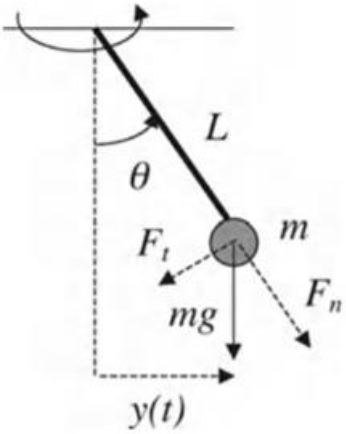


Figura 1.1 Sistema meccanico pendolo

Si consideri dunque il pendolo della **figura 1.1** , costituito da un corpo di massa **m** = 0.55 (Kg) posto all'estremità di un'asta rigidia di lunghezza **L** = 0.40 (m) e massa trascurabile. La posizione del corpo è individuata dall'angolo $\boldsymbol{\theta}$**(t)** che l'asta forma con la verticale. Il movimento del pendolo è determinato dall'azione della forza peso cui è soggetto , la cui componente nella direzione tangenziale è data da $\boldsymbol{F_t}(\boldsymbol{t}) = -mg\sin\big(\theta(t)\big)\ (N)$, e dall'effetto di una coppia meccanica esterna $\boldsymbol{u}(\boldsymbol{t})$. Supponiamo infine che il pendolo sia incernierato attraverso un giunto rotoidale che oppone al moto del corpo stesso una forza di attrito viscoso di coefficiente **β** = 0.5, proporzionale alla velocità angolare.

Detta $I = mL^2$, l'opposizione del corpo al moto (inerzia) , dal secondo principio della meccanica rotazionale segue

$$M_{tot}(t) = I\,\frac{\partial^2}{\partial t^2}\theta(t)$$

Nel nostro caso avremo

$$I\frac{\partial^2}{\partial t^2}\theta(t) = -mgLsin\big(\theta(t)\big) - \beta L\frac{\partial}{\partial t}\theta(t) + u(t)$$

Dall'analisi è noto che : data una generica equazione differenziale ordinaria di ordine n , questa può sempre essere vista come un sistema di equazioni differenziali ordinarie del primo ordine. Infatti , ponendo

$$x_1(t) = \theta(t) \quad e \quad x_2(t) = \frac{\partial}{\partial t}\theta(t)$$

e supponendo inoltre di poter misurare la distanza della massa dalla verticale $y(t) = Lsin\big(x_1(t)\big)$, la rappresentazione del nostro sistema in termini di variabili di stato diventa

$$\begin{cases} \frac{\partial}{\partial t}x_1(t) = x_2(t) \\ \frac{\partial}{\partial t}x_2(t) = -\frac{g}{L}\sin\big(x_1(t)\big) - \frac{\beta}{mL}x_2(t) + \frac{1}{mL^2}u(t) \\ y(t) = Lsin\big(x_1(t)\big) \end{cases}$$

il risultato è un modello in variabili di stato del secondo ordine non lineare.

Discretizzazione del Modello

Il primo passo verso la discretizzazione del modello è la scelta del periodo di campionamento. A tal proposito ricordiamo che detto T_{a1} il tempo di assestamento all'1% , la scelta del periodo di campionamento del sistema deve essere tale che

$$\frac{T_{a1}}{10\alpha} \leq T_s \leq \frac{T_{a1}}{\alpha}$$

con α di solito compreso tra 5 e 10. Se scegliessimo ad esempio $\alpha = 5$, la condizione precedente suggerisce che la scelta del periodo di campionamento deve essere fatta in maniera tale da avere dai 5 ai 50 campioni nell'intervallo di tempo di assestamento.

Andiamo dunque a determinare il tempo di assestamento all'1% graficando la risposta al gradino unitario del sistema. A tal proposito utilizzeremo la funzione ode45(..) di matlab che appunto integra il sistema di equazioni differenziali sull'intervallo di tempo passatogli come parametro.

Nota

Possiamo già predire l'andamento della risposta al gradino. Avendo supposto la presenza di una forza di attrito opposta al moto , la risposta al gradino avrà un andamento oscillatorio smorzato rispetto alla posizione di equilibrio.

$$\begin{cases} x_2(t) = 0 \\ -\frac{g}{L}\sin\big(x_1(t)\big) + \frac{1}{mL^2} = 0 \\ y(t) = L\sin\big(x_1(t)\big) \end{cases} \quad , \quad \begin{cases} x_2(t) = 0 \\ x_1(t) = \sin^{-1}\left(\frac{1}{mLg}\right) = 27.6 \\ y(t) = Lsin\big(x_1(t)\big) = 0.185 \end{cases}$$

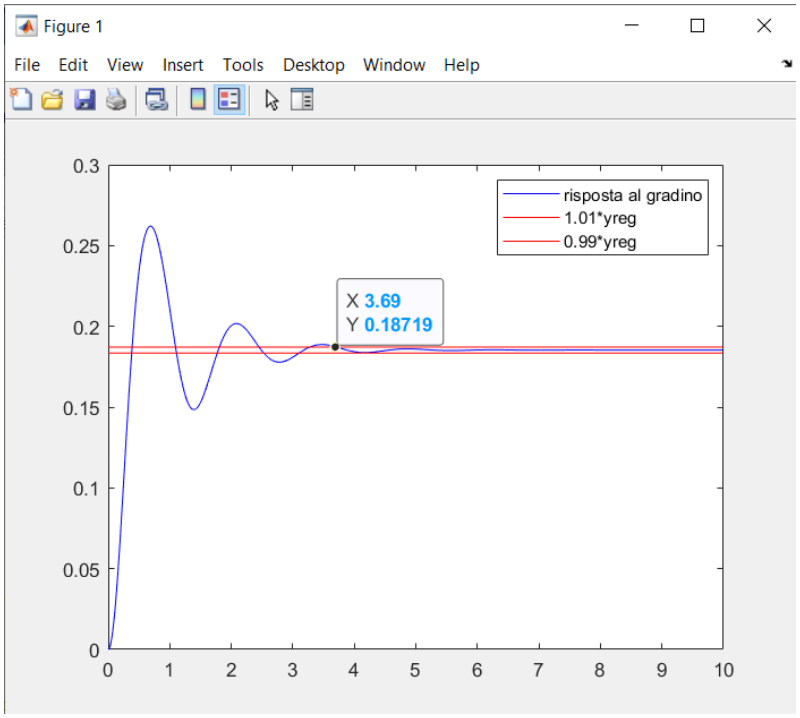
riportiamo di seguito la parte principale del codice matlab per la generazione della risposta al gradino

- **Codice MatLab**

```
function [xp] = generaSistema(t,x,L,m,g,b,u)
xp = [x(2);(-g/L)*sin(x(1))-(b/(m*L))*x(2)+(1/(m*L*L))*u];
end

[t,x] = ode45(@generaSistema,t,[0 0],[],L,m,g,b,u);
y     = L*sin(x(:,1))';
figure(1),grid,plot(t,y,'b'),legend("risposta al gradino");
```

ricordando che il tempo di assestamento all'1% è il tempo necessario affinchè l'uscita sia compresa nell'intervallo $[(1-0.01)y_\infty, (1+0.01)y_\infty]$



e con una scelta $\alpha = 8$, se ne deduce un tempo di campionamento $Ts = \frac{3.69}{80} \cong 0.05$ (s).

Come tecnica per discretizzare il nostro sistema , useremo la più naif , eulero in avanti , andremo cioè ad approssimare la derivata con il suo rapporto incrementale, ottenendo così

$$\begin{cases} x_1(k+1) = x_1(k) + Ts * x_2(k) \\ x_2(k+1) = x_2(k) - \frac{gTs}{L}\sin\big(x_1(k)\big) - \frac{bTs}{mL}x_2(k) + \frac{Ts}{mL^2}u(k) \\ y(k) = L\sin\big(x_1(k)\big) \end{cases}$$

riportiamo di seguito uno schema simulink della versione discretizzata del processo , che utilizzeremo nel punto successivo.

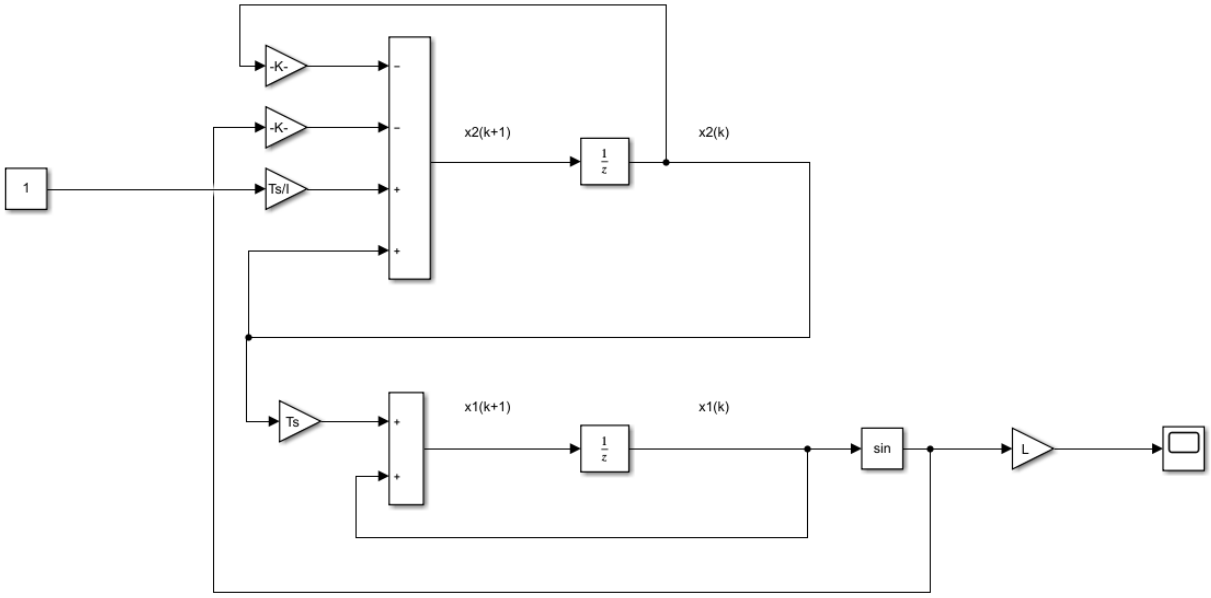
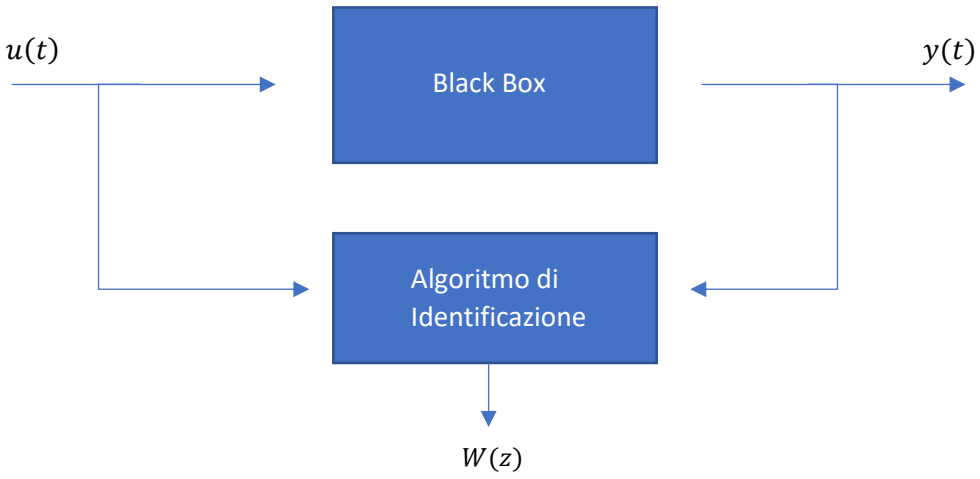


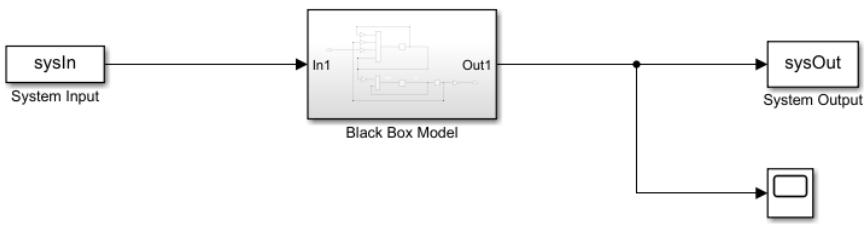
Figura 1.2 Schema simulink modello discretizzato

Identificazione Del Modello

Procediamo ora all'identificazione del modello. Ovvero , supponiamo di non conoscere il modello del sistema (black box) e cerchiamo di individuare un modello a partire dalle misurazione delle coppie ingresso – uscita.



A tal proposito costruiamo uno schema simulink in cui supponiamo di con conoscere il modello del sistema.



Dove nel blocco Black Box Model si è riportato lo schema simulink della versione discretizzata del sistema in esame.

Poniamoci ora una domanda molto importante : “che tipo di ingresso bisogna dare in input al sistema?”. Ovviamente non sempre si può scegliere l’ingresso da fornire al sistema, si pensi ad esempio ad uno schema di controllo in retroazione; ma quando è possibile farlo , è bene scegliere un ingresso che vada ad eccitare tutte le frequenze del sistema. Il segnale per eccellenza è il “rumore bianco” ma per nostra sfortuna è un segnale puramente ideale.

Ma la caratteristica che rende il rumore bianco ideale per questo tipo di problema è il fatto che esso gode di uno spettro totalmente piatto, ogni frequenza contribuisce alla trasformata di Fourier in modo uniforme , e cosa veramente più importante è il fatto che esso contiente tutte le frequenze.

Viste queste peculiarità del rumore bianco , dobbiamo scegliere un segnale che abbia quanto meno uno spettro biatto il più completo possibile.

La scelta tipica è un segnale “PRBS” , un segnale che oscilla casualmente tra -1 e 1. Andiamo dunque ad applicare al processo in esame un segnale di questo genere e grafichiamone la risposta.

- **codice MatLab**

```
u          = idinput(N,'prbs');
sysIn      = [t u];
sim("blackBoxModel");
```

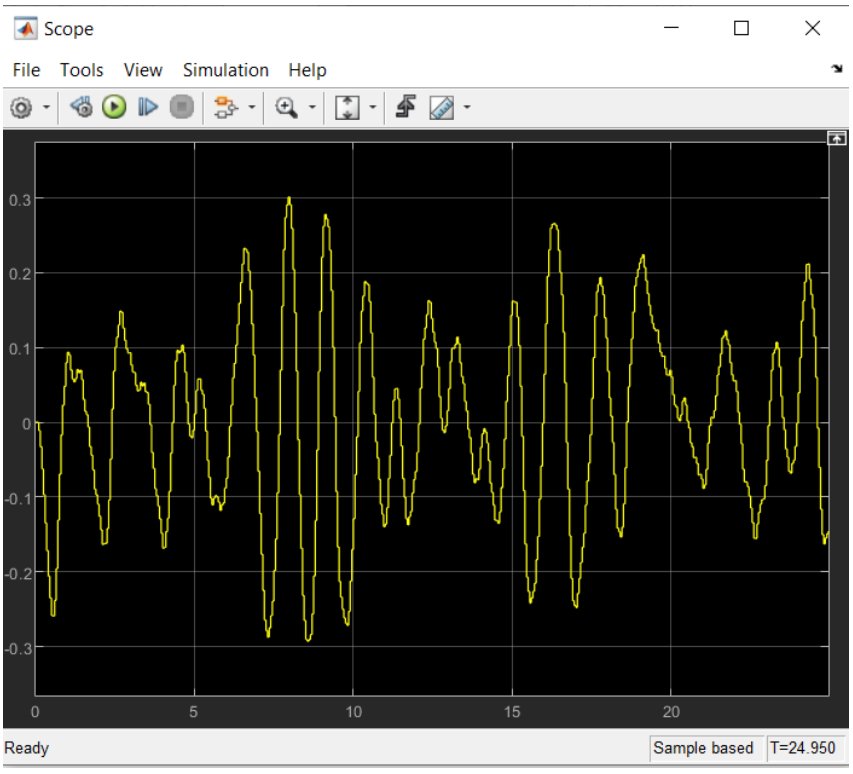


Figura 1.3 Risposta Sistema segnale PRBS

Al fine di stimare la funzione di trasferimento del processo in esame , metteremo a punto un modello ARX, un modello del tipo

$$y(k) + a_1 y(k-1) + \cdots + a_{na} y(k-n_a) = b_1 u(k-n_k) + b_2 u(k-1-n_k) + \cdots + b_{nb}(k-n_k-n_b+1) + e(k)$$

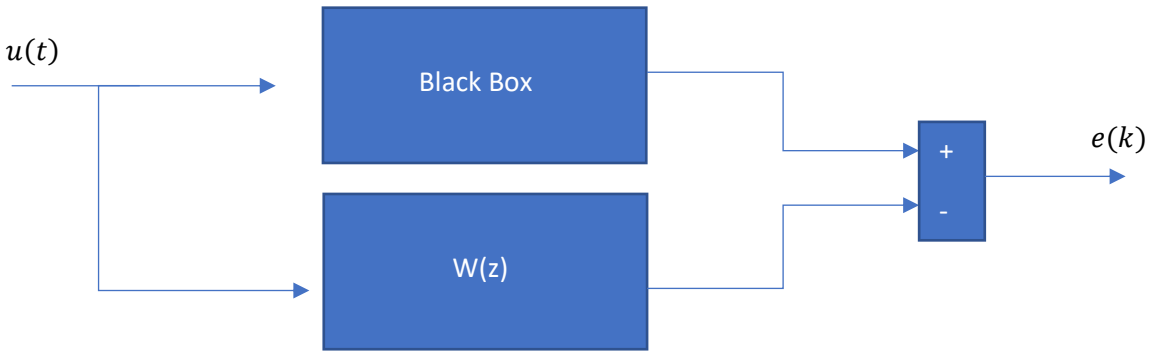
con $e(k)$ un rumore bianco a media nulla.

Come si intuisce dal modello appena definito , i parametri da stimare sono i coefficienti delle combinazioni lineari. Il problema di identificazione può essere facilmente trasformato in un problema di ottimizzazione andando a definire una cifra di merito (funzionale)

$$J(\theta) = \sum_{j=n}^{N} e(n)^2 = \sum_{j=n}^{N} (y(j) - \varphi(j)^T \theta)^2$$

dove con φ si è indicato il vettore dei regressori e con θ il vettore dei parametri. Un buon modello (identificato dal vettore θ) allora sarà quel modello che porterà all'ottimizzazione del funzionale. Essendo un problema di regressione lineare , l'approccio classico a tale problema sono i "minimi quadrati" , metodo inventato da Gauss per determinare l'orbita di Cerere.

Matlab ci viene incontro a questo problema mediante il comando $arx(data, [n_a\ n_b\ n_k])$. Come si vede la scelta del modello ricade sulla scelta dei massimi ritarti accettabili sull'uscita e sull'ingresso.
La domanda sorge spontanea : "Come si fa a stabilirli?". Lo si fa andando ad analizzare l'errore che si commette tra l'uscita effettiva e l'uscita stimata.



Dalla definizione del modello precedente, si capisce che più questo errore si avvicina ad un rumore bianco è più sarò fiducioso del fatto che $W(z)$ è un buon modello del processo in esame.

- **codice MatLab**

```
u           = idinput(N,'prbs');
sysIn       = [t u];
sim("blackBoxModel");
yv          = sysOut';              %uscita priva di rumore

%definizione rumore di misura
d           = normrnd(0,0.1,1,N);
ym          = yv + d;               %uscita rumorosa

%modello arx
data        = [ym(1:N)' u];
system      = arx(data,[14 14 2]);
ys          = lsim(system,u,(0:N-1))'; %uscita stimata
```

Andando ad analizzare l'andamento dell'errore di identificazione si è trovato che la scelta dei parametri migliori per il modello sono rispettivamente 14 14 2.

Andiamo a riportare una serie di andamenti dell'uscita del processo e del modello stimato. Si ricordi che ogni esecuzione corrisponde ad esiti diveresi del processo aleatorio considerato , quindi si avranno diversi andamenti in funzione dell'esito (processo stocastico). In particolare si graficherà il confronto tra le due risposte al gradino unitario.

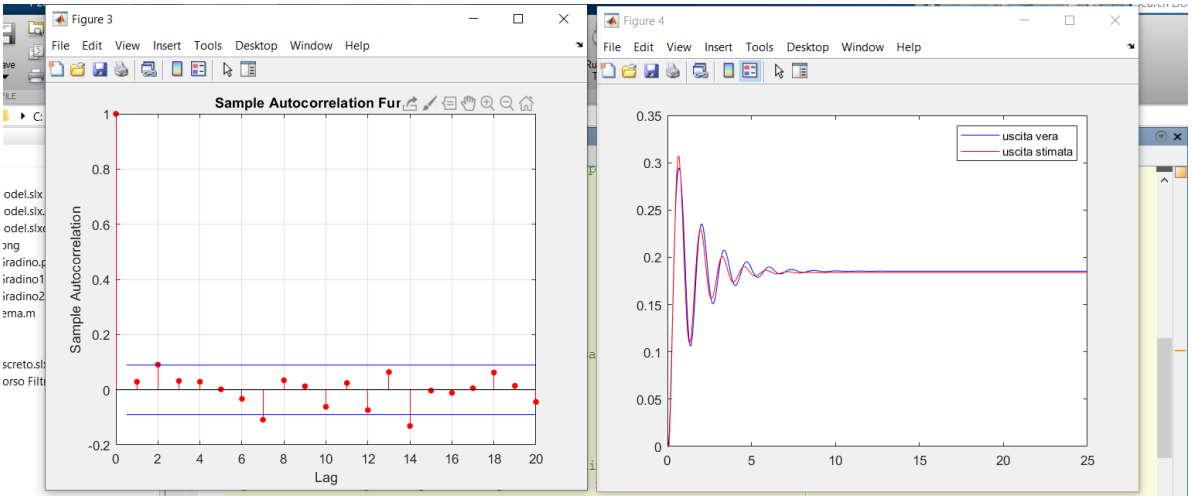


Figura 1.4 Confronto risposta al gradino

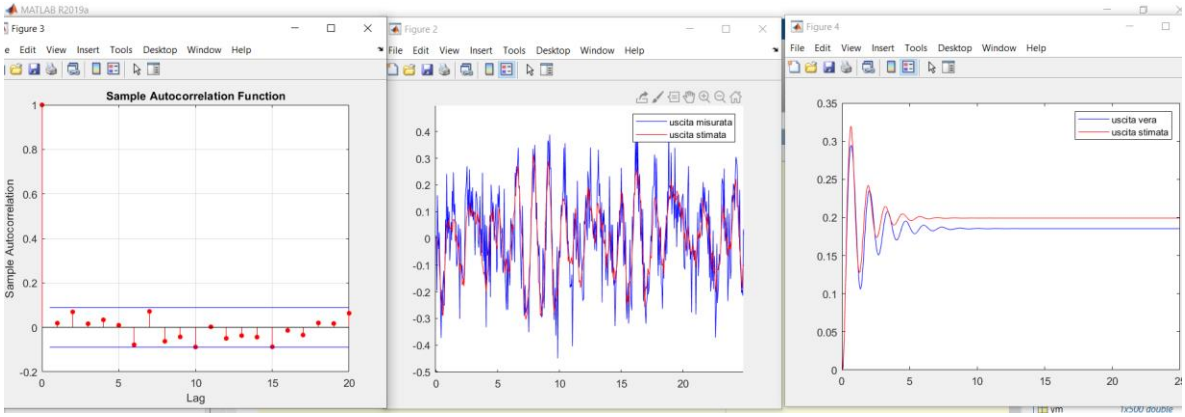


Figura 1.5 Confronto risposta al gradino (1)

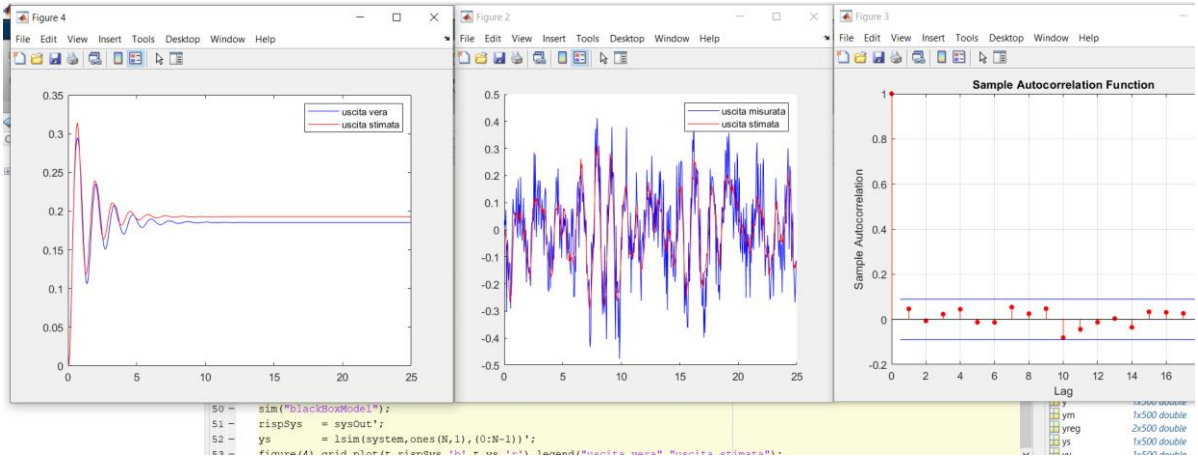


Figura 1.6 Confronto risposta al gradino (2)

Riportiamo di seguito la struttura della funzione di trasferimento individuata ad una delle esecuzioni

```
Discrete-time ARX model: A(z)y(t) = B(z)u(t) + e(t)

  A(z) = 1 - 0.09546 z^-1 - 0.1334 z^-2 - 0.07887 z^-3 - 0.1029 z^-4 - 0.06592 z^-5 + 0.03505 z^-6 + 0.06905 z^-7
         + 0.0127 z^-8 + 0.1009 z^-9 + 0.07827 z^-10 + 0.008568 z^-11 - 0.01218 z^-12 + 0.1728 z^-13 + 0.08448 z^-14


  B(z) = 0.01139 z^-2 + 0.02669 z^-3 + 0.02996 z^-4 + 0.02194 z^-5 + 0.02855 z^-6 + 0.02785 z^-7 + 0.02325 z^-8
         + 0.02125 z^-9 + 0.007345 z^-10 + 0.01029 z^-11 + 0.003615 z^-12 + 0.007141 z^-13 + 0.002463 z^
                                                                                              -14 - 0.009177 z^-15
```

Figura 1.7 Funzione trasferimento modello ARX(14,14,2)

Andiamo ora a graficare l'andamento delle risposte dei due sistemi dinamici ad un particolare ingresso sinusoidale

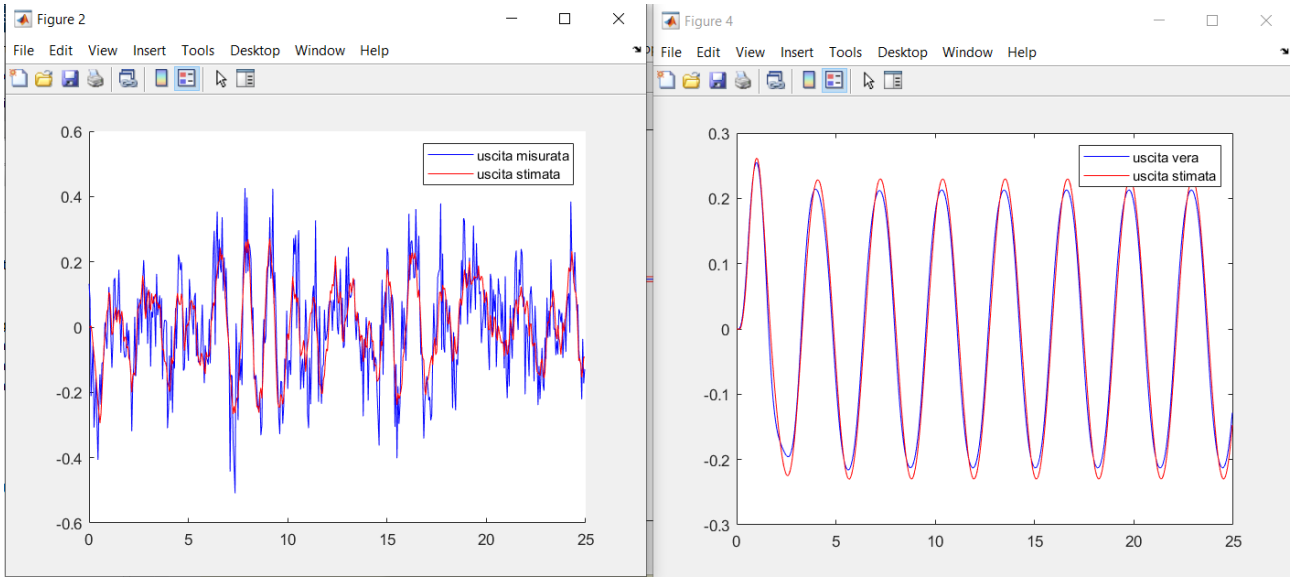


Figura 1.8 Confronto risposte ingresso u(t) = sin(2*t).

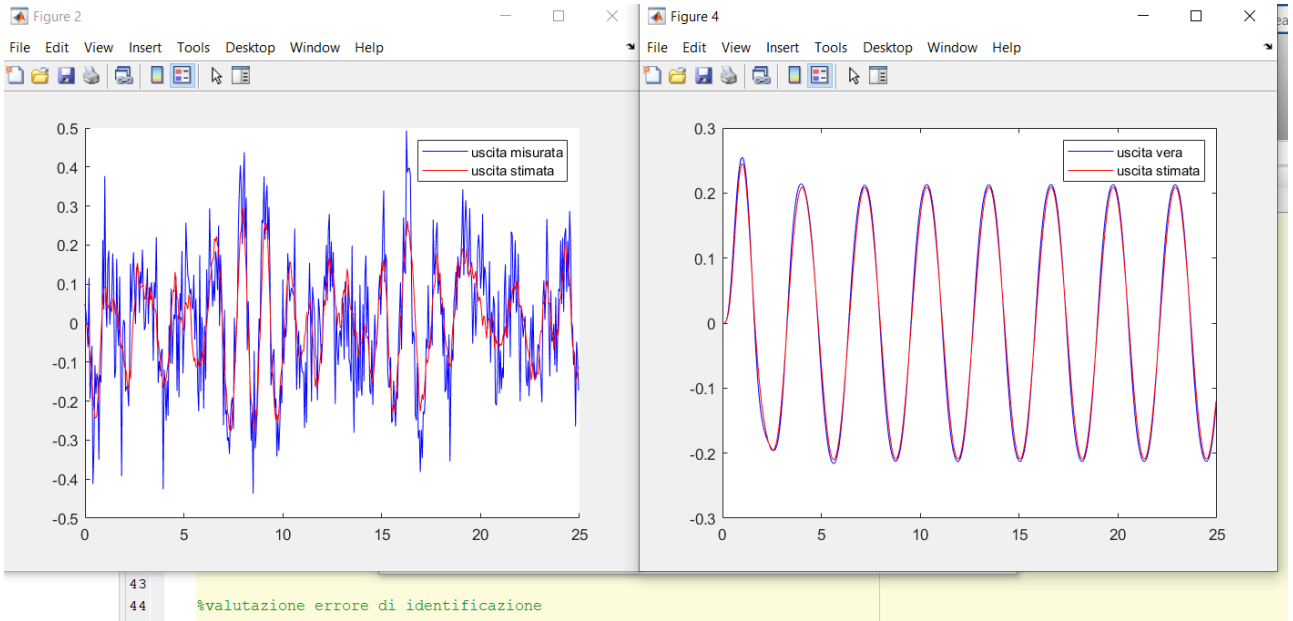


Figura 1.9 Confronto risposte ingresso u(t) = sin(2*t).

Stima dello Stato (Extended Kalman Filter)

Prima di procedere alla linearizzazione del sistema dinamico e alla scrittura del codice MatLab per l'implementezione del filtro di Kalman esteso, diamo brevemente un'idea dell'estensione del filtro di Kalman a sistemi non lineari.

Partiamo dunque con il considerare il seguente sistema non lineare

$$\begin{cases} x(k+1) = f\big(x(k), u(k)\big) + v_1(k) \\ \quad y(k) = h\big(x(k)\big) + v_2(k) \end{cases}$$

dove come al solito $v_1(.) \sim W_n(0, V_1)$ e $v_2(.) \sim W_n(0, V_2)$ tra loro scorrelati. Inoltre supponiamo che V_2 sia definita positiva .

Supponiamo di disporre di una traiettoria nominale del sistema

$$\begin{cases} \bar{x}(k+1) = f\big(\bar{x}(k), \bar{u}(k)\big) \\ \quad \bar{y}(k) = h\big(\bar{x}(k)\big) \end{cases}$$

notiamo che il calcolo di $\bar{x}(k), \bar{y}(k)$ può essere effettuato a priori , risolvendo un'equazione alle differenze deterministica.
Andiamo ora ad effettuare la linearizzazione del sistema intorno a questa traiettoria nominale. A tal proposito indichiamo con

$$\Delta x(k) = x(k) - \bar{x}(k)$$

$$\Delta y(k) = y(k) - \bar{y}(k)$$

la discrepanza tra l'andamento nominale è quello perturbato.

Supponendo che f, h siano funzioni abbastanza regolari , esse possono essere sviluppate in serie di Taylor rispetto a $\bar{x}(k), \bar{u}(k)$. Sostituendo questo sviluppo arrestato al primo ordine si ottiene (a meno dei rumori bianchi)

$$x(k+1) = \Delta x(k+1) + \bar{x}(k+1) \cong f\big(\bar{x}(k), \bar{u}(k)\big) + \frac{\partial}{\partial x} f|_{x=\bar{x}(k), u=\bar{u}(k)} \Delta x(k) + \frac{\partial}{\partial u} f|_{x=\bar{x}(k), u=\bar{u}(k)} \Delta u(k)$$
$$y(k) = \Delta y(k) + \bar{y}(k) = h\big(\bar{x}(k), \bar{y}(k)\big) + \frac{\partial}{\partial x} h|_{x=\bar{x}(k), u=\bar{u}(k)} \Delta x(k)$$

Indicando con

$$F(k) = \frac{\partial}{\partial x} f|_{x=\bar{x}(k), u=\bar{u}(k)} \qquad G(k) = \frac{\partial}{\partial u} f|_{x=\bar{x}(k), u=\bar{u}(k)} \qquad H(k) = \frac{\partial}{\partial x} h|_{x=\bar{x}(k), u=\bar{u}(k)}$$

si ottiene

$$\begin{cases} \Delta x(k+1) = F(k)\Delta x(k) + G(k)\Delta u(k) + v_1(k) \\ \quad \Delta y(k) = H(k)\Delta x(k) + v_2(k) \end{cases}$$

che è un sistema lineare tempo variante. A tale sistema lineare si può applicare la teoria di Kalman vista, ottenendo così il predittore

$$\widehat{\Delta x}(k+1|k) = F(k)\, \widehat{\Delta x}(k|k-1) + G(k)\Delta u(k) + K(k)e(k)$$

$$e(k) = \Delta y(k) - \hat{\Delta} y(k|k-1)$$

$$\widehat{\Delta y}(k+1|k) = H(k)\widehat{\Delta x}(k+1|k)$$

ottenendo così

$$\hat{x}(k+1|k) = \hat{\Delta}x(k+1|k) + \bar{x}(k+1) = f\big(\bar{x}(k), \bar{u}(k)\big) + F(k)\widehat{\Delta x}(k|k-1) + G(k)\Delta u(k) + K(k)e(k)$$

osservando che

$$f\big(\hat{x}(k|k-1), u(k)\big) = f\big(\bar{x}(k), \bar{u}(k)\big) + F(k)\widehat{\Delta x}(k|k-1) + G(k)\Delta u(k)$$

$$h(\hat{x}(k|k-1) = h\big(\bar{x}(k)\big) + H(k)\widehat{\Delta x}(k|k-1)$$

si ottiene

$$\begin{cases} \hat{x}(k+1|k) = f\big(\hat{x}(k|k-1), u(k)\big) + K(k)e(k) \\ e(k) = y(k) - \hat{y}(k|k-1) = y(k) - h\big(\hat{x}(k|k-1)\big) \end{cases}$$

che prende il nome di filtro linearizzato o filtro tangente. Nella pratica si è visto che risultati nettamenti migliori li si ottengono andando a linearizzare le due funzioni intorno all'uscita stima effettuata

$$F(k) = \frac{\partial}{\partial x} f|_{x=\hat{x}(k|k-1), u=u(k)} \qquad G(k) = \frac{\partial}{\partial u} f|_{x=\hat{x}(k|k-1), u=u(k)} \qquad H(k) = \frac{\partial}{\partial x} h|_{x=\hat{x}(k|k-1), u=u(k)}$$

Il predittore corrispondente viene chiamato **predittore di Kalman Esteso**.

Nella pratica però si utilizza una forma più affidabile del predittore di Kalman , la cosiddetta forma **predictor/corrector .** Con questa forma del predittore il passaggio da $\hat{x}(k|k-1)$ a $\hat{x}(k+1|k)$ viene effettuata in due passi. Prima si passa $\hat{x}(k|k-1)$ a $\hat{x}(k|k)$ facendo uso del guadagno del filtro $K_0(k)$; dopo di che si passa al calcolo di $\hat{x}(k+1|k)$.
L'affidabilità di questa forma del predittore sta per il fatto che nel calcolare $\hat{x}(k|k)$ si fa uso di una misura in più dell'uscita, andando così a ridurre l'incertezza sulla stima dello stato al passo k. E da qui il nome correttore (corrector).

$$\begin{cases} \hat{x}(k|k) = \hat{x}(k|k-1) + K_o(k)e(k) & (corrector) \\ \hat{y}(k|k-1) = h\big(\hat{x}(k|k-1)\big) \\ e(k) = y(k) - \hat{y}(k|k-1) \\ K_o(k) = P(k)H(k)'(H(k)P(k)H(k)' + V2)^{-1} \\ P_o(k) = \big(I - K_o(k)H(k)\big)P(k) \\ \hat{x}(k+1|k) = f\big(\hat{x}(k|k), u(k)\big) \\ P(k+1) = F(k)P_o(k)F(k)' + V1 \\ F(k) = \frac{\partial}{\partial x} f\ |_{\left(x=\hat{x}(k|k), u=u(k)\right)} \end{cases}$$

- **Linearizzazione Del Modello**

Al fine di ottenere la versione estesa del filtro di Kalman determiniamo $F(k) e\, H(k)$

$$F(k) = \frac{\partial}{\partial x} f = \begin{bmatrix} 1 & Ts \\ -\frac{gTs}{L}\cos\big(x(1)\big) & \left(1 - \frac{bTs}{mL}\right) \end{bmatrix} \in R^{2x2}$$

$$H(k) = \frac{\partial}{\partial x} h = \begin{bmatrix} L * \cos\big(x(1)\big) & 0 \end{bmatrix} \in R^{1x2}$$

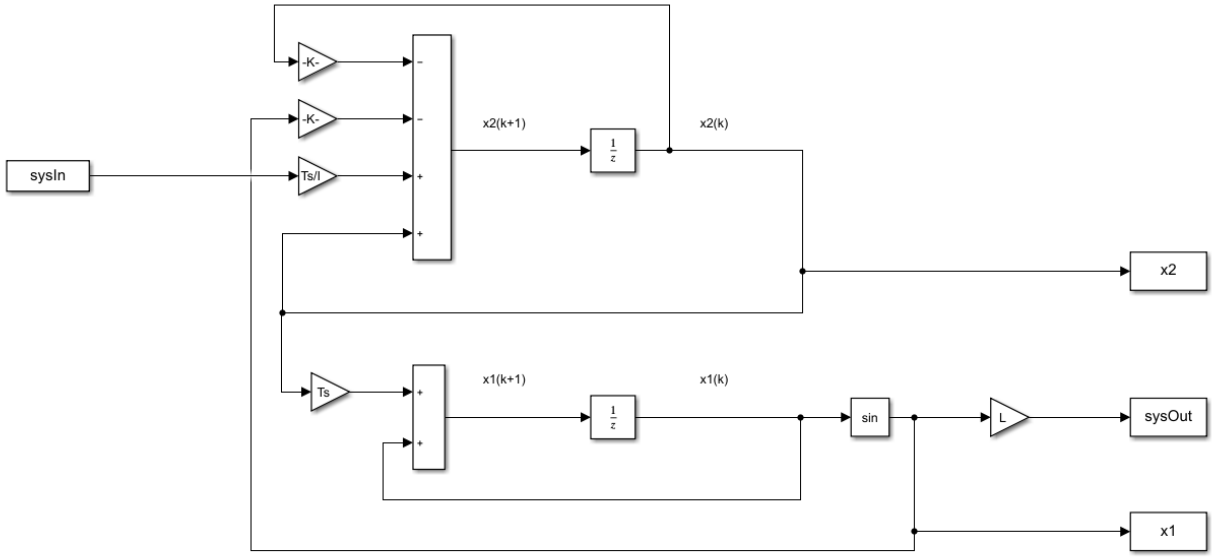
$$G(k) = \frac{\partial}{\partial u} f = \begin{bmatrix} 0 \\ \frac{Ts}{mL^2} \end{bmatrix} \in R^{2x1}$$

Il nostro modello linearizzato con l'aggiunta dei rumori bianchi sarà

$$\begin{cases} x(k+1) = F(k)x(k) + G(k)u(k) + v_1(k) \\ y(k) = H(k)x(k) + v_2(k) \end{cases}$$

Andiamo dunque ad applicare Kalman Esteso per stimare lo stato del sistema , quindi posizione e velocità angolare.

Costruiamo uno schema simulink che mi fornisca le misurazioni dell'uscita e dello stato reale del sistema



Procediamo poi alla scrittura di 4 funzioni che mi restituiscano ad ogni passo k $F(k), H(k), f, h$

```
function [H] = generaMatriceH(x,k,L)
H = [L*cos(x(1,k)),0];
end

function [F] = generaMatriceF(x,k,L,Ts,m,g,b)
a = ((g*Ts)/L)*cos(x(1,k));
b = 1-((b*Ts)/(m*L));
F = [1,Ts;-a,b];
end

function [val] = h(x,k,L)
val = L*sin(x(1,k));
end

function [xp] = f(x,k,L,Ts,m,g,b,u)
a  = x(1,k)+Ts*x(2,k);
b  = x(2,k)-((g*Ts)/L)*sin(x(1,k))-((b*Ts)/(m*L))*x(2,k)+(Ts/(m*L*L))*u;
xp = [a;b];
end
```

Andiamo ad effettuare la predizione dello stato dando al sistema in ingresso un gradino di ampiezza unitario

```
%risposta al gradino unitario
u     = ones(N,1);
sysIn = [t' u];
sim("modelloForKalman");
```

di seguito riportiamo il codice MatLab dell'implementazione del filtro di Kalman Esteso.

```
%definizione variabili stato stimato
Xkk       = zeros(2,N);                 %x^(k|k)
Xkk1      = zeros(2,N);                 %x^(k|k-1)
Xkk1(:,1) = xv(:,1);                    %x^(1|0)

%implementazione filtro di Kalman esteso
for k     = 1 : N
    %generazione matrice H(k)
    H         = generaMatriceH(Xkk1,k,L);
    %generazione innovazione
    e         = ym(k) - h(Xkk1,k,L);
    %guadagno del Filtro Ko(k)
    Ko        = P*H'*inv(H*P*H' + V2);
    %stima dello stato al passo k
    Xkk(:,k) = Xkk1(:,k) + Ko*e;
    %measurement update
    Po        = (I-Ko*H)*P;
    %generazione matrice F
    F         = generaMatriceF(Xkk,k,L,Ts,m,g,b);
    %predizione stato al passo k + 1
    if k < N
        Xkk1(:,k+1) = f(Xkk,k,L,Ts,m,g,b,u(k));
        P           = F*Po*F'+V1;
    end
    %memorizzazione innovazione
    evect(k) = e;
    %memorizzazione guadagni
    KK(:,k)   = Ko;
end
```

grafichiamo l'andamento dello stato effettivo e di quello stimato dal filtro di Kalman esteso.

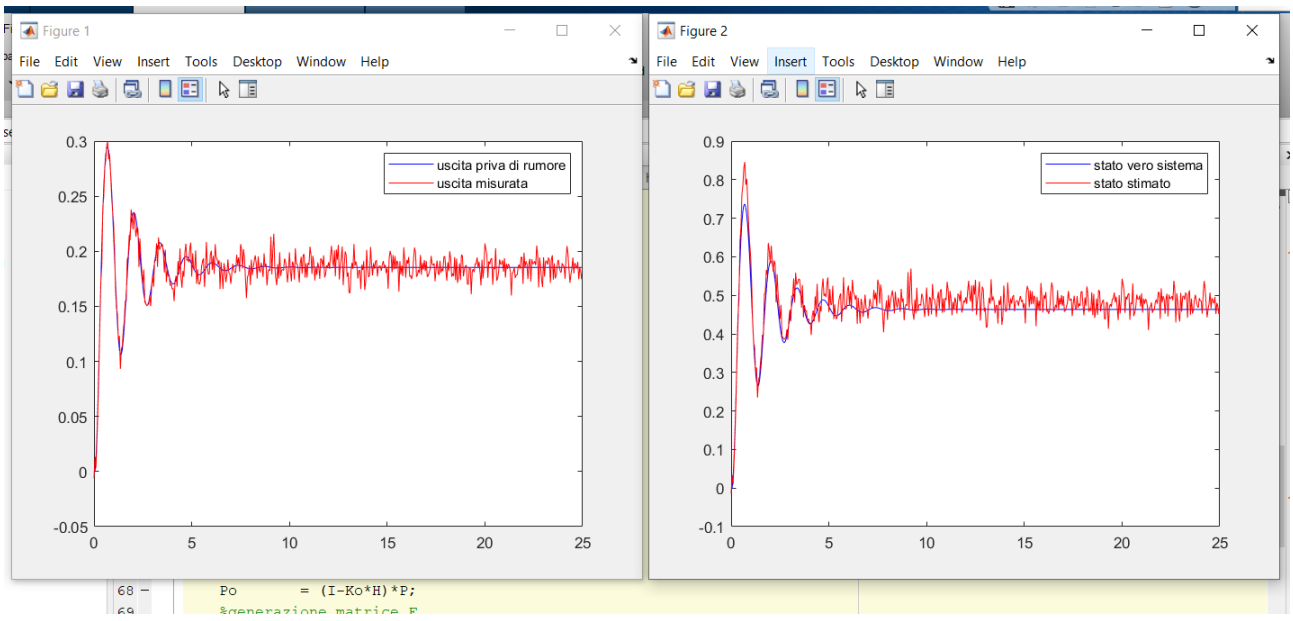


Figura 2 Confronto stato vero e stato stimato da Kalman

Prima cosa da detenere presente , quando abbiamo troncato la serie di Taylor al primo ordine, abbiamo perso tutte le proprietà del nostro filtro di Kalman. A conti fatti non esiste nessuna garanzia di convergenza del filtro , nessuno ci garantisce che l'innovazione tende al rumore di misura.

Ma andiamo a vedere cosa succede ai guadagni e all'errore nel nostro caso

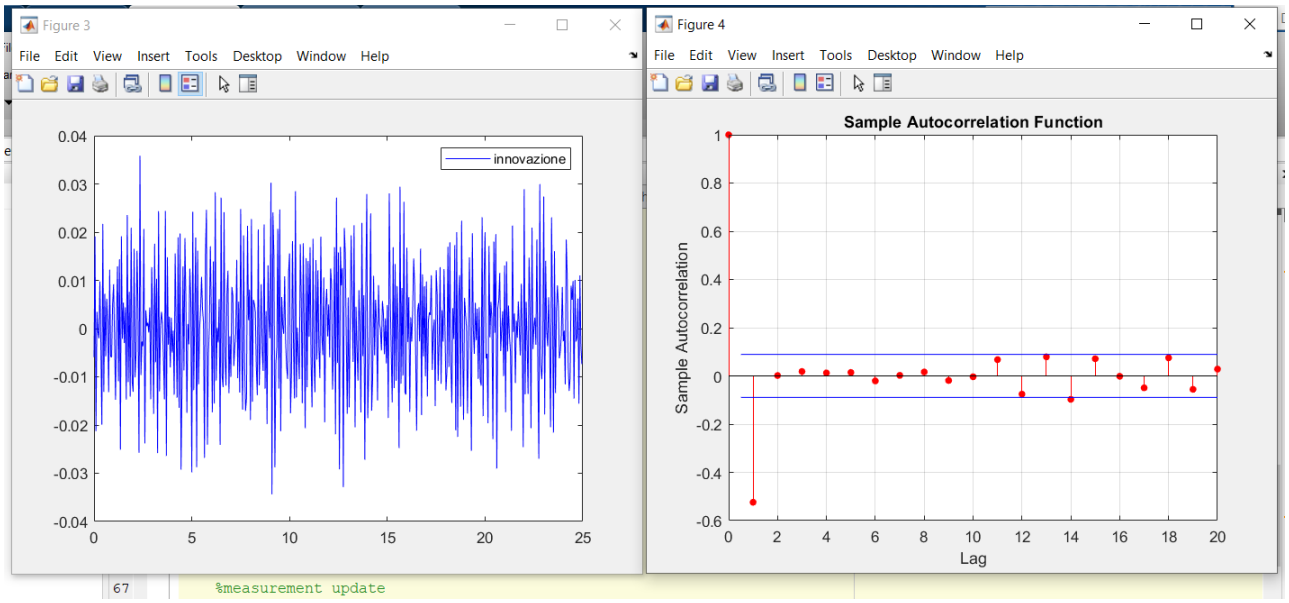


Figura 2.1 innovazione

Nel nostro caso il filtro di Kalman esteso ha funzionato , l'innovazione tende al rumore di misura del sistema.

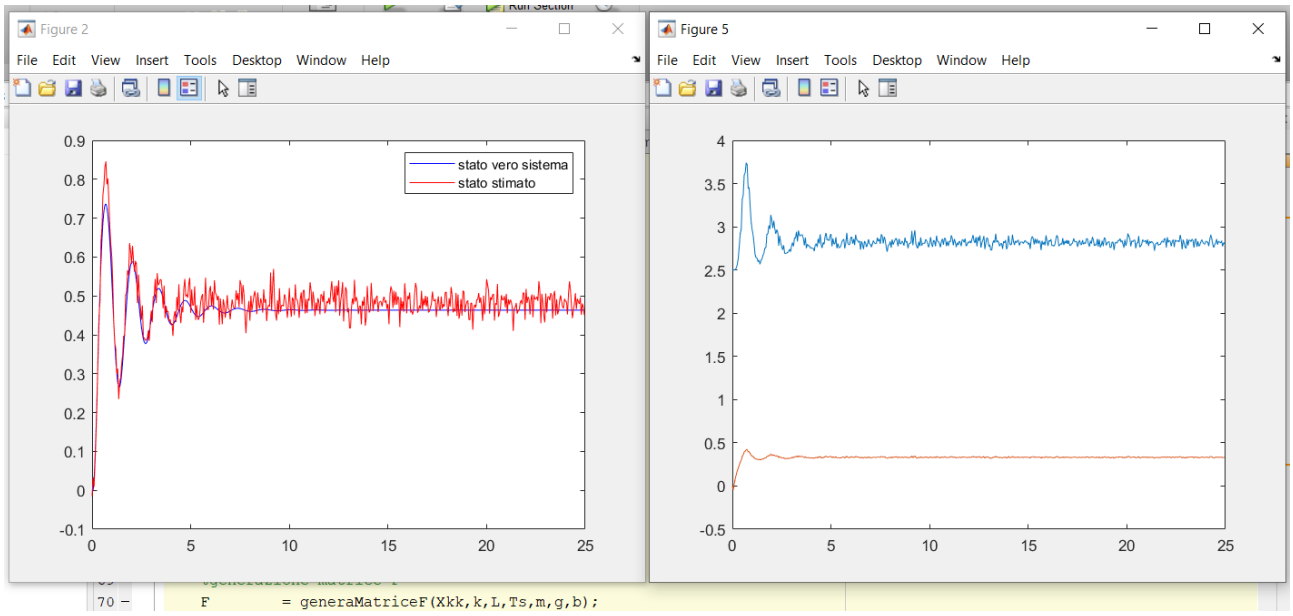


Figura 2.2 guadagni del filtro

Notiamo che anche il guadagno del filtro non diverge. Questo significa che anche la soluzione dell'equazione di Riccati non diverge (equazione alle differenze non lineare a soluzione matriciale). Ma ricordiamo che la soluzione dell'equazione di Riccati altro non è che la matrice di varianza dell'errore di predizione dello stato. La non divergenza indica perciò la capacità del predittore di fornire una predizione dello stato con un errore di entità limitata (in senso probabilistico).