



DIE
TI.

UNI
NA

VERSITA' DEGLI STUDI DI
POLI FEDERICO II

DIPARTIMENTO DI INGEGNERIA ELETTRICA
E TECNOLOGIE DELL'INFORMAZIONE

SCUOLA POLITECNICA E DELLE SCIENZE DI BASE

TESINA DI COMPLEMENTI DI CONTROLLO

Controllo di un sistema
meccanico costituito da due
carrelli ed un corpo che rotola
senza strisciare collegati
mediante molle e smorzatori

Prof.

Alfredo Pironti

Fiorella Maria Romano

P38000265

Anno Accademico 2023/2024

Sommario

Introduzione	3
Modello matematico	4
Controllabilità e Osservabilità dell'impianto	7
Assegnamento Autovalori	7
Progettazione impianto Aumentato	7
Scelta degli Autovalori	8
Simulazioni	10
Osservatore	12
Simulazioni	14
Controllo LQR	17
LQR con azione integrale	18
I tentativo	18
II tentativo	19
Funzione di trasferimento a ciclo chiuso	20
Allocazione autovalori vs controllo LQ	21
Controllo LQG	24
Filtro di Kalman	24
Osservatore vs filtro di Kalman	29
Controllo H^∞ e Mixed Sensitivity Design	31

Introduzione

Lo scopo di questo elaborato è quello di descrivere il processo di progettazione e simulazione di vari algoritmi di controllo affrontati nel corso di Complementi di Controlli.

Il sistema fisico di riferimento è costituito dalla serie di un carrello di massa m_1 , un corpo di massa M che rotola senza strisciare ed un altro carrello di massa m_2 , tenuti assieme da un sistema di molle e smorzatori.

Il controllo verrà effettuato attraverso l'uso di diverse tecniche:

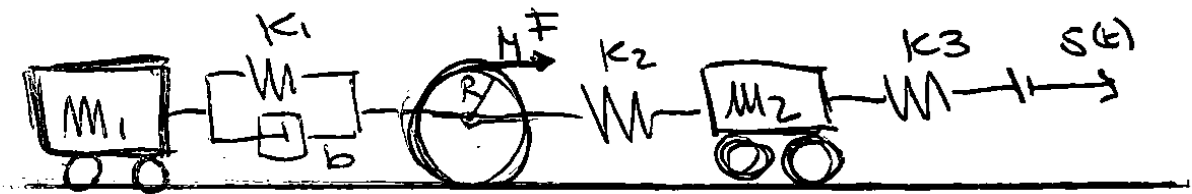
- Controllo con retroazione dello stato e assegnamento degli autovalori con integratore
- Controllo con ricostruzione dello stato mediante un osservatore
- Controllo ottimo LQ
- Controllo LQG con filtro di Kalman
- Controllo H^∞ Mixed Sensitivity Design

Le specifiche da rispettare sono:

- Tempo di assestamento al 2% $< 8s$
- Sovraelongazione $< 10\%$
- Errore a regime nullo per riferimenti a gradino

L'obiettivo sarà controllare la posizione del primo carrello attraverso l'ingresso di controllo u_1 che agisce sulla molla collegata al secondo carrello, considerando la forza che agisce sul corpo che rotola come un disturbo.

Modello matematico



Equazioni del moto:

$$m_1 \dot{s}_1 = -b(\dot{s}_1 - R\dot{\theta}) - k_1(s_1 - R\theta)$$

$$\left(MR + \frac{J}{R}\right) \ddot{\theta} = -b(R\dot{\theta} - \dot{s}_1) - k_1(R\theta - s_1) - k_2(R\theta - s_1) + 2F$$

$$m_2 \dot{s}_2 = -k_2(s_2 - R\theta) - k_3(s_2 - s)$$

Variabili di stato:

$$x_1 = s_1; x_2 = \dot{s}_1; x_3 = \theta; x_4 = \dot{\theta}; x_5 = s_2; x_6 = \dot{s}_2; u_1 = s; u_2 = F; y_1 = s_1; y_2 = s_2$$

Equazioni del moto in variabili di stato:

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = -\frac{k_1}{m_1}x_1 - \frac{b}{m_1}x_2 + \frac{k_1 R}{m_1}x_3 + \frac{b R}{m_1}x_4 \\ \dot{x}_3 = x_4 \\ \dot{x}_4 = \frac{k_1}{K}x_1 + \frac{b}{K}x_2 - \frac{k_2 + k_1}{K}R x_3 - \frac{b}{K}x_4 + \frac{k_2}{K}x_5 + \frac{2}{K}u_2 \\ \dot{x}_5 = x_6 \\ \dot{x}_6 = k_2 \frac{R}{m_2}x_3 - \frac{k_3 + k_2}{m_2}x_5 + \frac{k_3}{m_2}u_1 \end{cases}$$

Uscite:

$$\begin{cases} y_1 = x_1 \\ y_2 = x_5 \end{cases}$$

Su MATLAB:

```
m1=2;
m2=1.5;
M=0.8;
R=0.5;
k1=0.9;
k2=0.8;
k3=0.8;
b=0.35;
J=M*R^2/2;
const=M*R+J/R;
```

```
A=[ 0 1 0 0 0 0;
    -k1/m1 -b/m1 k1/m1*R b*R/m1 0 0;
    0 0 0 1 0 0;
    k1/const b/const -(k1*R+k2*R)/const -b*R/const k2/const 0;
    0 0 0 0 0 1;
    0 0 k2*R/m2 0 -(k3+k2)/m2 0];

B=[0 0 0 0 0 k3/m2;
    0 0 0 2/const 0 0]';

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

D=zeros(2);
```

Andiamo a valutare anche i poli dell'impianto a ciclo aperto:

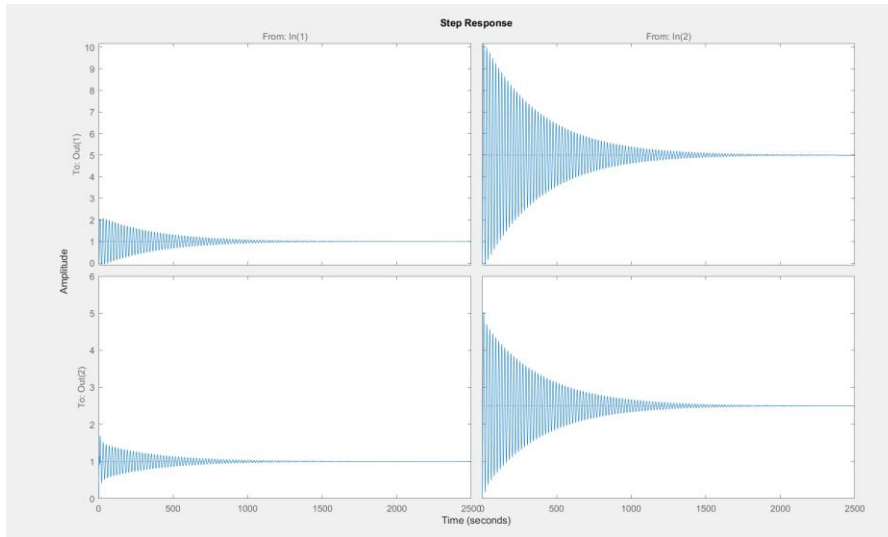
```
eig(A)
```

```
ans = 6x1 complex
    -0.1567 + 1.3793i
    -0.1567 - 1.3793i
    -0.0740 + 0.9250i
    -0.0740 - 0.9250i
    -0.0026 + 0.3105i
    -0.0026 - 0.3105i
```

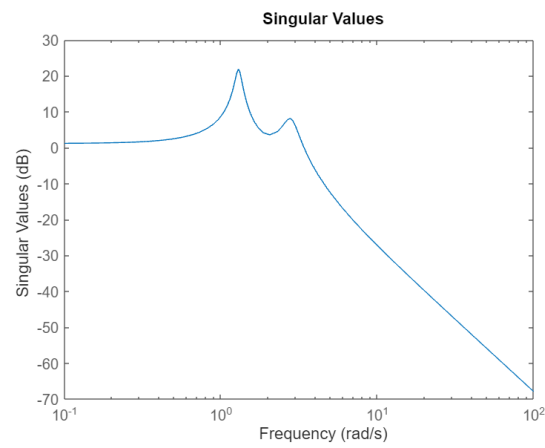
Presenta 3 coppie di poli complessi e coniugati, tutti a parte reale negativa.

Possiamo concludere dicendo che il sistema a ciclo aperto è *asintoticamente stabile*.

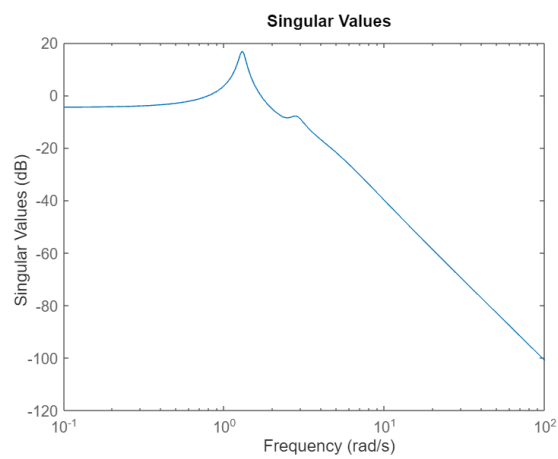
```
sys=ss(A,B,C,D);
figure
step(sys)
```



```
figure
sigma(sys(:,1))
```



```
sigma(sys(:,2))
```



```
hinfnorm(sys(:,1))
ans = 12.3446
```

```
hinfnorm(sys(:,2))
ans = 6.9349
```

Controllabilità e Osservabilità dell'impianto

Affinché le strategie di controllo abbiano l'effetto desiderato, bisogna assicurarsi della controllabilità e dell'osservabilità del sistema considerato.

Un sistema si dice “completamente controllabile” se e solo se la sua matrice di controllabilità ha rango pieno, il quale dovrà essere pari all'ordine del sistema stesso.

Un sistema si dice “completamente osservabile” se e solo se la sua matrice di osservabilità ha rango pieno; il ragionamento è quindi analogo al caso precedente.

Tramite i seguenti comandi, abbiamo valutato la controllabilità e l'osservabilità dell'impianto in esame:

```
ctr=ctrb(A,B);  
rank(ctr)    %risulta completamente controllabile  
>> ans=6  
  
obs=obsv(A,C);  
rank(obs)    %risulta completamente osservabile  
>> ans=6
```

Con il comando *rank* andiamo a valutare la dimensione delle matrici di controllabilità e di osservabilità.

Possiamo concludere che entrambe le matrici presentano rango pari all'ordine del sistema; di conseguenza quest'ultimo è completamente controllabile e completamente osservabile.

Assegnamento Autovalori

La prima strategia di controllo presentata è basata sull'assegnamento degli autovalori tramite retroazione di stato. Questa tecnica consiste in uno schema di controllo con retroazione statica dello stato, la cui caratteristica fondamentale è quella di poter scegliere la dinamica del sistema a ciclo chiuso allocando i poli della matrice di trasferimento; in questo modo risulta possibile stabilizzare il sistema e soddisfare alcuni requisiti della risposta transitoria. Questa flessibilità è resa possibile dalla completa controllabilità del sistema, che consente di manipolare tutti i poli del sistema a ciclo chiuso. Inoltre, per garantire errore a regime nullo, inserisco un'azione integrale.

Progettazione impianto Aumentato

Inserendo un'azione integrale mi assicuro astatismo con disturbi costanti ed errore a regime nullo per riferimenti a gradino.

Si può dimostrare che se la coppia (A,B) è completamente controllabile e se la matrice W ha rango pari alla somma del numero di stati e del numero di uscite, allora il sistema aumentato è completamente controllabile.

La matrice W è definita nel seguente modo:

$$W = [A \ B; -C \ 0];$$

ciò equivale a dire che il numero di uscite deve essere minore o al più uguale al numero di ingressi di controllo. In questo caso, poiché abbiamo un solo ingresso di controllo, è possibile retroazionare un'unica uscita.

```
Aaum=[A zeros(6,1);
-C(1,:) 0];
Baum=[B;zeros(1,2)];
Caum=[C zeros(2,1)];
Daum=D;

sysAum=ss(Aaum,Baum,Caum,Daum);
rank(ctrb(Aaum,Baum(:,1)))
```

```
>>ans = 7
```

Il Sistema Aumentato risulta completamente controllabile attraverso l'ingresso u1.

Posso quindi dividere l'azione di controllo in due parti; un termine proporzionale alla retroazione di stato, un altro proporzionale all'azione integrale:

$$u = -Kf x - Ki xi$$

Scelta degli Autovalori

Bisogna a questo punto individuare il vettore degli autovalori che vogliamo raggiungere tramite retroazione di stato. Questo vettore verrà scelto in relazione alle caratteristiche dinamiche e statiche richieste.

$$PO = 100e^{-\frac{\pi\zeta}{\sqrt{1-\zeta^2}}}$$

$$Ta_{2\%} \approx 4\zeta\omega_n$$

Da tali espressioni è possibile ricavare ω_n e ζ , dalle relazioni inverse.

$$\zeta = \frac{|\log(\frac{PO}{100})|}{\sqrt{\pi^2 + \log^2(\frac{PO}{100})}}$$

$$\omega_n \approx \frac{4}{\zeta Ta_{2\%}}$$

Suppongo di voler approssimare il mio sistema del settimo ordine con un sistema del secondo ordine con due poli dominanti. Questi ultimi saranno soluzione dell'equazione di secondo grado:

$$\lambda^2 + 2\zeta\omega_n\lambda + \omega_n^2 = 0$$

Impongo i parametri $PO = 10\%$ e $T_a = 8s$ in modo da rispettare tutte le specifiche assegnate al sistema.

```
P0=10;
Ta=8;
zita=abs(log(P0/100))/(sqrt(pi^2+log(P0/100)^2));
wn=4/(zita*Ta);
p=roots([1 2*zita*wn wn^2]);
% a questi aggiungo gli altri 5 poli ad alte frequenze:
Poli=[p real(p(1)*10) real(p(1)*11+1) real(p(1)*11+2) real(p(1)*11+3) real(p(1)*11+4)]

Poli = 1x7 complex
    -0.5000 - 0.6822i    -0.5000 + 0.6822i    -5.0000 + 0.0000i    -4.5000 + 0.0000i    -3.5000 + 0.0000i    -2.5000 + 0.0000i    -1.5000 + 0.0000i
```

A questo punto non resta che utilizzare il comando *place* per ottenere la matrice di retroazione di stato K. Inoltre, verifico che il sistema a ciclo chiuso abbia effettivamente i poli nelle posizioni desiderate.

```
Kaum=place(Aaum,Baum(:,1),Poli)

Kaum = 1x7
    -0.6423    1.5503   -0.6958   -0.2511    4.4921    2.8889   -0.9507
```

```
eig(Aaum-Baum(:,1)*Kaum)

ans = 7x1 complex
    -5.0000 + 0.0000i
    -4.5000 + 0.0000i
    -3.5000 + 0.0000i
    -0.5000 + 0.6822i
    -0.5000 - 0.6822i
    -2.5000 + 0.0000i
    -1.5000 + 0.0000i
```

Adesso divido la retroazione tra azione proporzionale e quella integrale:

```
Kpaum=Kaum(:,1:6)

Kpaum = 1x6
    -0.6423    1.5503   -0.6958   -0.2511    4.4921    2.8889

Kiaum=Kaum(:,7)

Kiaum = -0.9507
```

Lo schema di controllo in Simulink è il seguente:

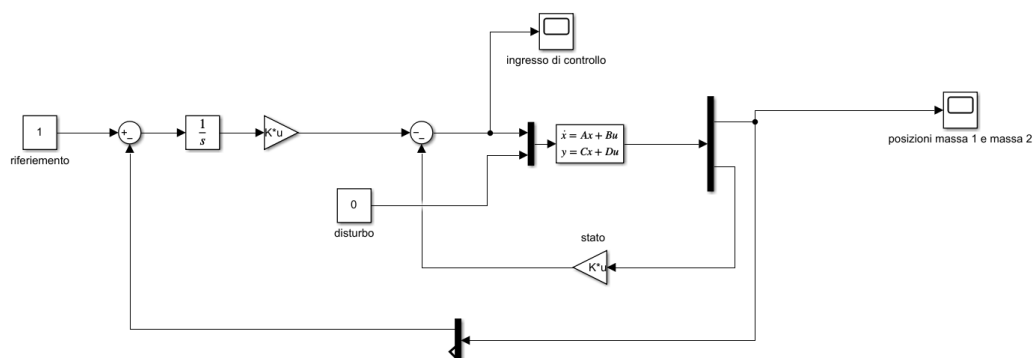


Figura 1 schema Simulink con retroazione ed assegnamento degli autovalori

Simulazioni

Considero come stato iniziale del sistema: $x_0 = [0 \ 0.1 \ 0 \ 0.02 \ 0 \ 0.5]$

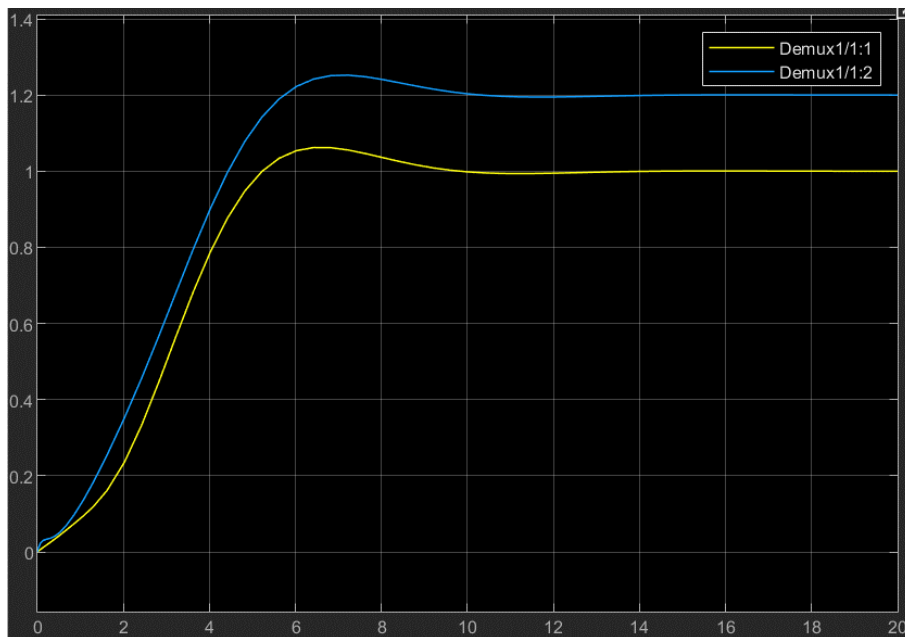


Figura 2 posizioni massa 1 e massa 2 senza disturbo

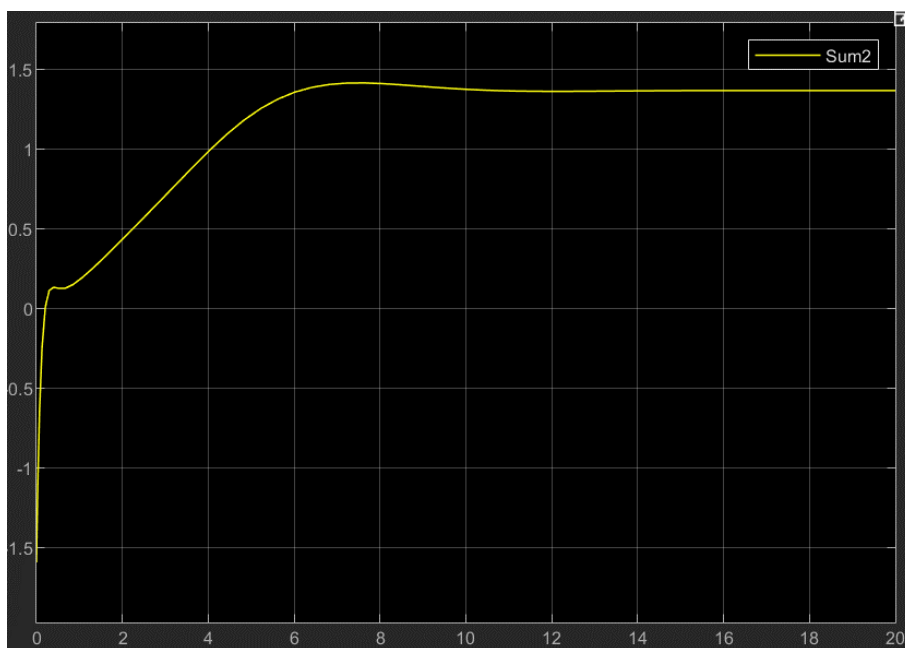


Figura 3 ingresso di controllo senza disturbo

Noto che le specifiche del sistema vengono rispettate, errore a regime nullo per ingressi a gradino viene ottenuto e lo sforzo di controllo non è eccessivo.

A questo punto aggiungo un disturbo non nullo.

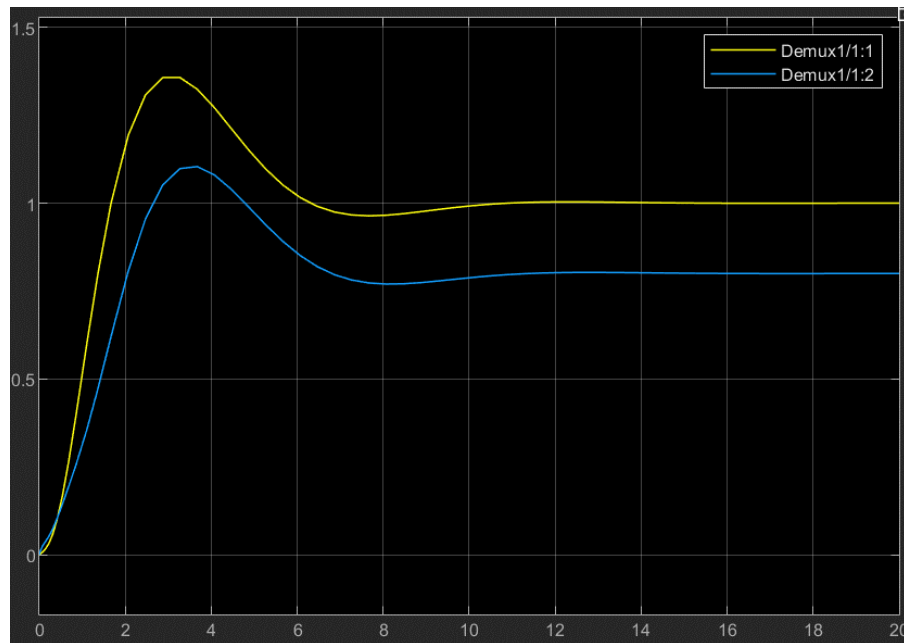


Figura 4 posizioni massa 1 e massa 2 con disturbo = 1

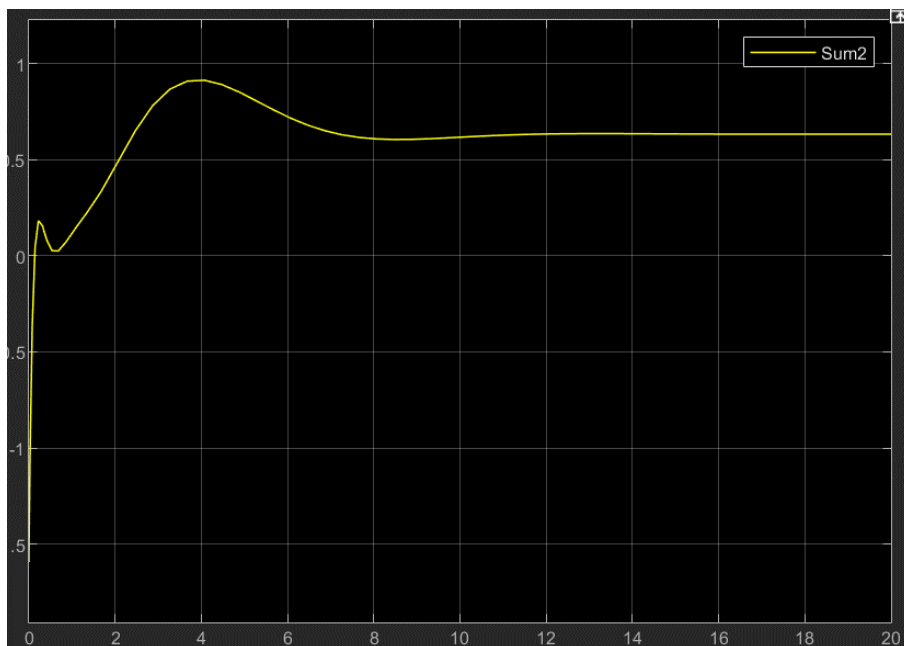


Figura 5 ingresso di controllo con disturbo = 1

L'uscita presenta un'importante sovraelongazione rispetto al caso precedente, mentre il controllo, sebbene presenti un picco, ha un'ampiezza minore. Tuttavia, le richieste statiche e dinamiche vengono ampiamente soddisfatte e l'astatismo viene ottenuto.

Osservatore

Durante l'utilizzo delle tecniche presentate nelle pagine precedenti ci siamo avvalsi della possibilità di retroazionare completamente gli stati del sistema in esame; infatti, negli schemi Simulink visti fino ad ora, la matrice C del blocco State-Space prevedeva una matrice identità in grado di fornire tutti gli stati in uscita all'impianto stesso.

Nella pratica non è sempre possibile misurare tutti gli stati del sistema che si vuole controllare o ciò può essere particolarmente costoso.

Per risolvere il problema si utilizza un osservatore.

Un osservatore è un sistema che prende come input solo gli ingressi di controllo e le uscite dell'impianto principale, per poi restituire in uscita il vettore di stato "osservato" di quest'ultimo.

L'errore di stima è dato dalla differenza tra lo stato effettivo che l'impianto presenta e lo stato misurato che l'osservatore sta restituendo.

L'equazione di stato tipica di un osservatore classico è la seguente:

$$\dot{\hat{x}} = A\hat{x} + Bu + L[y - C\hat{x} - Du]$$

Dove L è la matrice che fa in modo che l'errore converga a zero una volta che il sistema raggiunge la condizione di regime.

L'errore è in evoluzione libera:

$$e'(t) = \dot{x}(t) - \dot{\hat{x}}(t) = (A - LC)e(t)$$

Poiché la coppia (A,C) è completamente osservabile posso assegnare gli autovalori di (A-LC) a piacimento. Gli autovalori saranno scelti in modo da avere un transitorio molto più veloce rispetto a quello del sistema controllato, in modo da ottenere delle buone prestazioni dinamiche.

È importante notare che il controllore progettato finora è ancora valido, perché è applicabile il principio di separazione: la retroazione di stato e l'osservatore, sebbene facciano parte dello stesso sistema di controllo, possono essere progettati separatamente.

È quindi possibile costruire in maniera indipendente controllori e osservatori, cioè assegnare autonomamente gli autovalori di (A-BK) e (A-LC). Per ottenere buone prestazioni dinamiche, è opportuno scegliere gli autovalori di (A-LC) in modo che siano trascurabili rispetto a quelli del sistema controllato.

Il progetto dell'osservatore consiste quindi nella determinazione della matrice L, la quale si otterrà risolvendo un problema di allocazione degli autovalori per il sistema dell'impianto duale:

$$\dot{x} = A^T x + C^T u$$

```
obs=obsv(A,C(1,:));
rank(obs)
```

```
ans = 6
```

```
Poss=[-14 -16 -19 -20 -22 -24]
```

```
Poss = 1x6
```

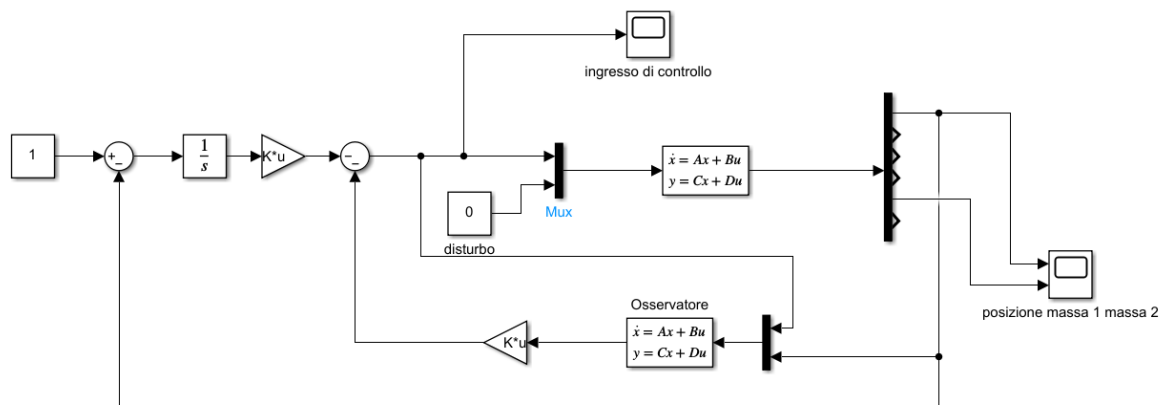
```
-14 -16 -19 -20 -22 -24
```

```
L=place(A',C(1,:)',Poss)';
Aoss=A-L*C(1,:);
Boss=[B(:,1) L];
Coss=eye(6);
Doss=zeros(6,2);
oss=ss(Aoss,Boss,Coss,Doss);
eig(Aoss)
```

```
ans = 6x1
```

```
-24.0000  
-22.0000  
-20.0000  
-19.0000  
-16.0000  
-14.0000
```

Costruisco lo schema in Simulink:



Simulazioni

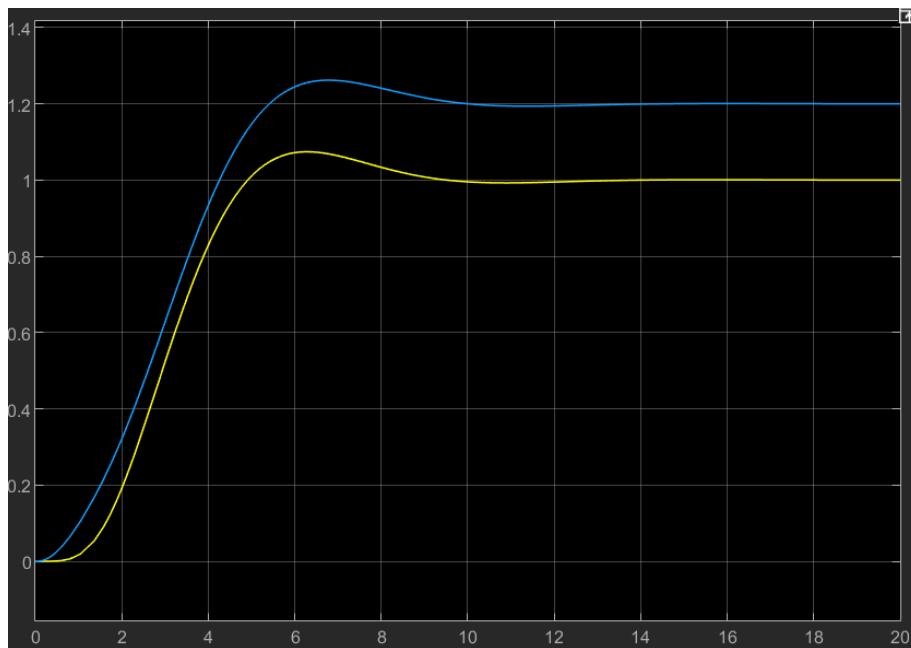


Figura 6 posizione massa 1 e massa 2

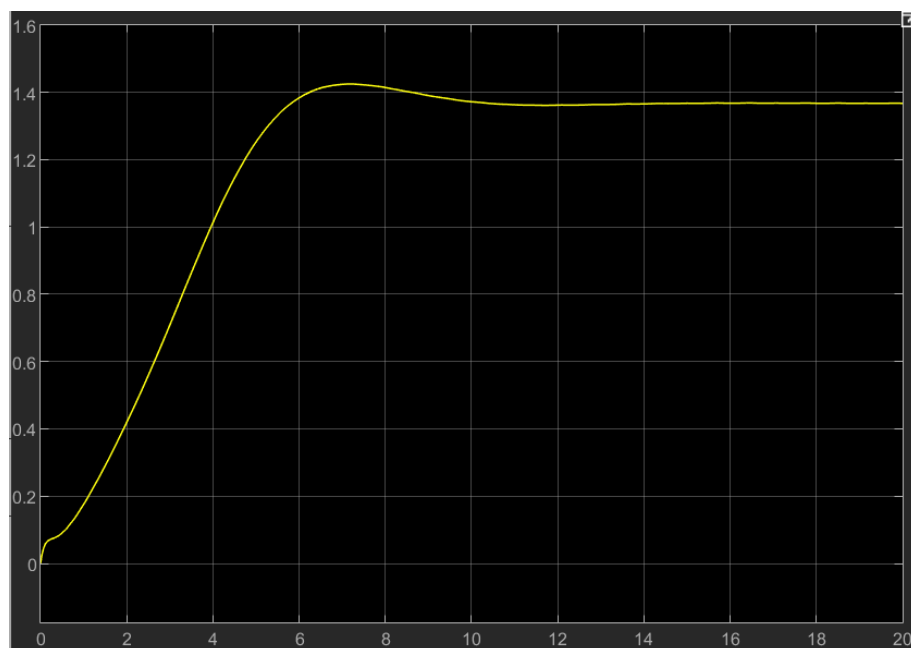


Figura 7 ingresso di controllo

Noto che le prestazioni statiche e dinamiche sono ampiamente rispettate anche in questo caso. L'ingresso di controllo è invece più 'smooth' nella regione transitoria, rimanendo molto contenuto.

considero adesso un disturbo diverso da 0.

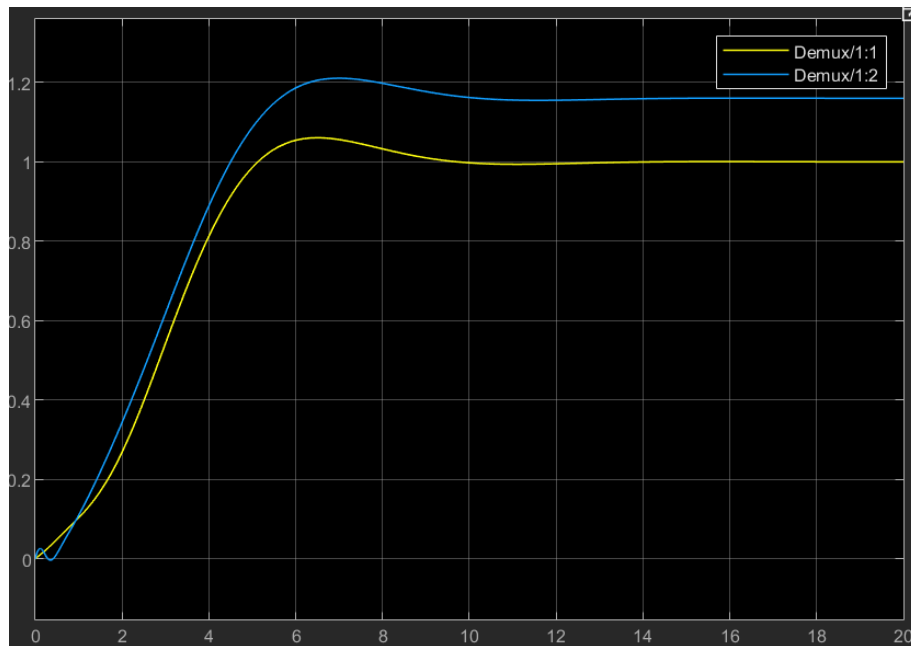


Figura 8 posizioni massa 1 massa 2 con disturbo

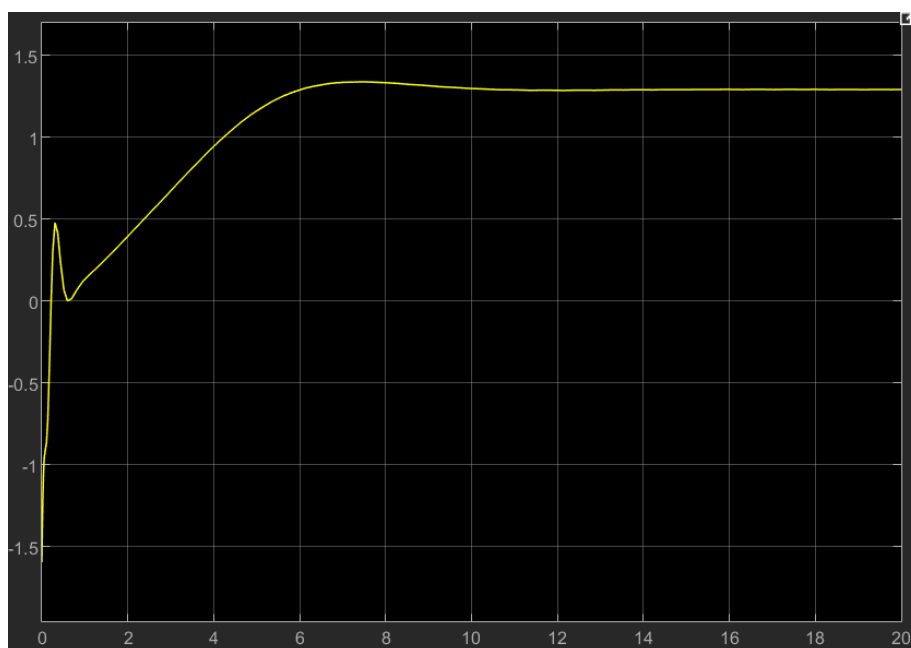


Figura 9 ingresso di controllo con disturbo

Noto che non ho grosse variazioni dell'uscita, per cui l'osservatore si rivela robusto nei confronti di disturbi costanti, ma l'ingresso di controllo è aumentato in maniera ripida nei primi istanti della simulazione, per poi portarsi ad un valore più o meno costante.

Osserviamo l'errore di stima, Se imposto una condizione iniziale per l'osservatore diversa da quella dell'impianto $x(0)=x_0$, si può osservare una piccola deformazione della traiettoria dovuta alla presenza dell'errore di stima abbastanza grande durante il transitorio.

Grazie all'asintotica stabilità della sua matrice dinamica (A-LC), è garantito comunque a regime un errore di stima dello stato dell'impianto nullo.

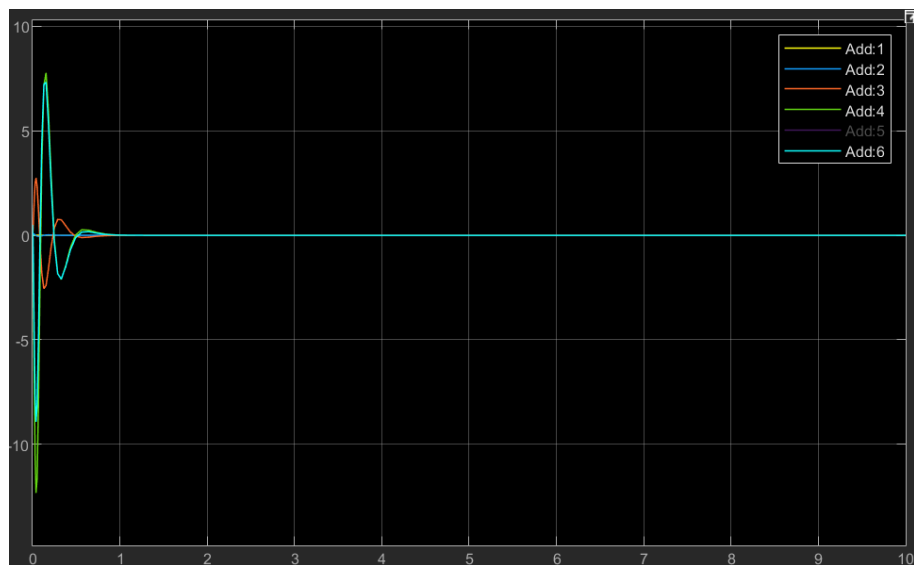
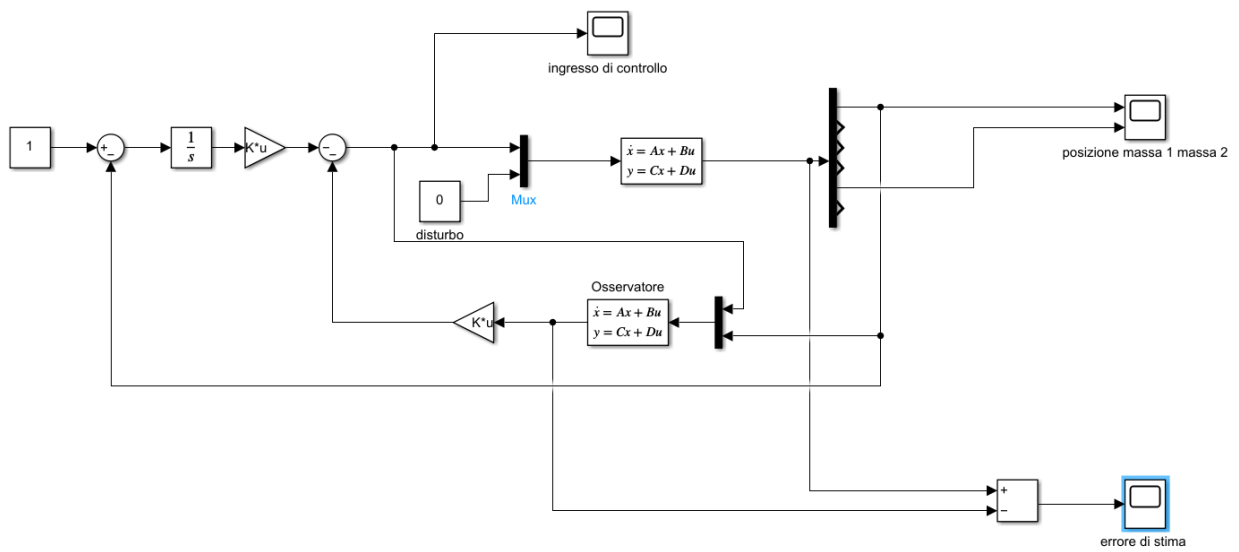


Figura 10 errore di stima

In ogni caso, dopo un certo transitorio, l'errore converge a zero e, essendo la stima degli stati corretta, il sistema di controllo riesce comunque a portare le uscite al valore desiderato.

Controllo LQR

Il controllo ottimo si occupa di trovare una legge di controllo che ottimizzi un indice di qualità. Nel caso del controllore LQR, si va a minimizzare la funzione di costo J definita come segue:

$$J(x_0, u) = \int_0^{+\infty} (x^T Q x + u^T R u) dt$$

dove Q è una matrice semidefinita positiva che pesa lo stato, mentre R è una matrice definita positiva che pesa l'ingresso di controllo. La dinamica del sistema viene vista come vincolo di uguaglianza del problema di ottimizzazione vincolato. Si è particolarizzata la trattazione al caso stazionario ed estendendo l'orizzonte temporale all'infinito.

Così facendo la matrice P che risolve l'equazione algebrica di Riccati risulta stazionaria e di conseguenza il guadagno $K = R^{-1}B^TP$ sarà anch'esso stazionario. Si ricava dunque un controllo in retroazione di stato del tipo $u = -Kx$, proprio come nel caso dell'assegnamento degli autovalori, dove però in questo caso K non è ottenuta andando a scegliere la posizione degli autovalori a ciclo chiuso desiderati, ma è ottenuta risolvendo il problema di ottimizzazione.

È importante sottolineare che si può progettare la legge di controllo solo dopo aver verificato che la coppia (A,B) sia stabilizzabile e la coppia (A,H) sia rilevabile, dove $Q = H^TH$.

Per poter ottenere i risultati desiderati, lo strumento con cui si può lavorare è la scelta dei pesi Q ed R che compaiono nella funzione obiettivo, tenendo conto che il primo pesa le variabili di stato mentre l'altro pesa l'ingresso di controllo. Ad esempio, aumentando il peso su R , si realizza una legge di controllo di ampiezza più bassa e di conseguenza una dinamica più lenta, mentre al contrario diminuendo il peso su R si può favorire un aumento della legge di controllo e di conseguenza una risposta più veloce.

Si può anche scegliere, invece di far comparire una forma quadratica dello stato, di includere nella funzione obiettivo una forma quadratica dell'uscita:

$$J = \int y' Q y + u' R u dt$$

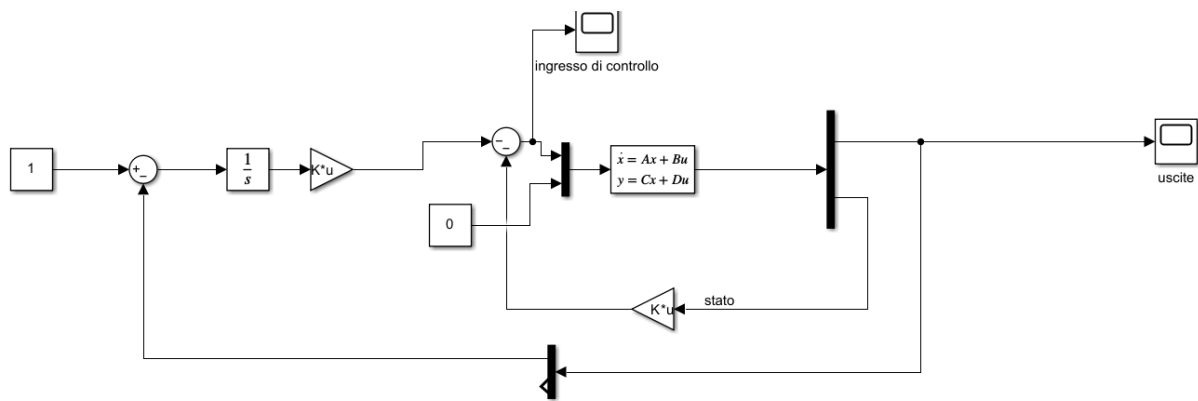
Dove $Qy \in R^{p \times p}$ è la matrice che pesa direttamente le uscite del sistema. Tale scelta risulta essere piuttosto intuitiva, anche perché in base ai valori delle uscite possiamo decidere se penalizzare o meno una determinata azione di controllo, mentre spesso non si è grado di fare lo stesso tipo di valutazione sullo stato perché questo potrebbe non essere interamente disponibile.

Nel momento in cui consideriamo il sistema aumentato, la matrice Q assume la forma:

$$Q = \begin{bmatrix} C' Q_y C & 0 \\ 0 & Q_I \end{bmatrix}$$

dove il termine che pesa lo stato dell'integratore viene considerato separatamente, ma la matrice Q ha comunque una struttura di tipo diagonale.

In Simulink:



LQR con azione integrale

Per determinare i pesi delle matrici Q ed R si procede con la tecnica TRIAL and ERROR, ovvero per tentativi.

I tentativo

Voglio ottenere una legge di controllo contenuta ed un'uscita piuttosto smooth, ma che ovviamente soddisfi le specifiche imposte. Allora immagino di pesare di più la matrice Q_y che la matrice R .

```
Qy=20;
Qi=1;
Q=[C'*Qy*C zeros(6,1);
  zeros(1,6) Qi];
R=10;

[Klq,P,e]=lqr(sysAum(:,1),Q,R);
```

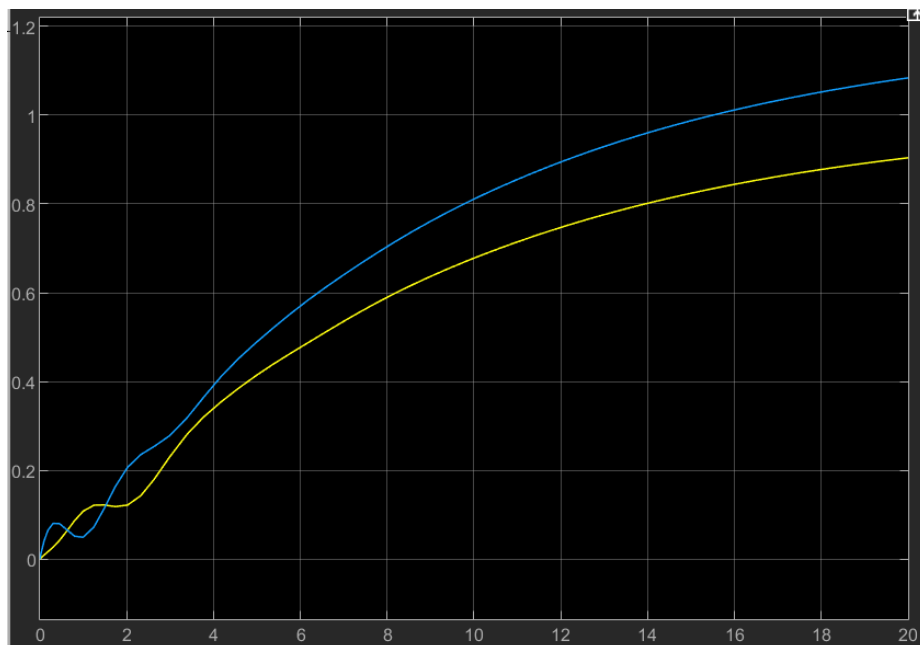


Figura 11 posizione massa 1 e massa 2 con LQR I

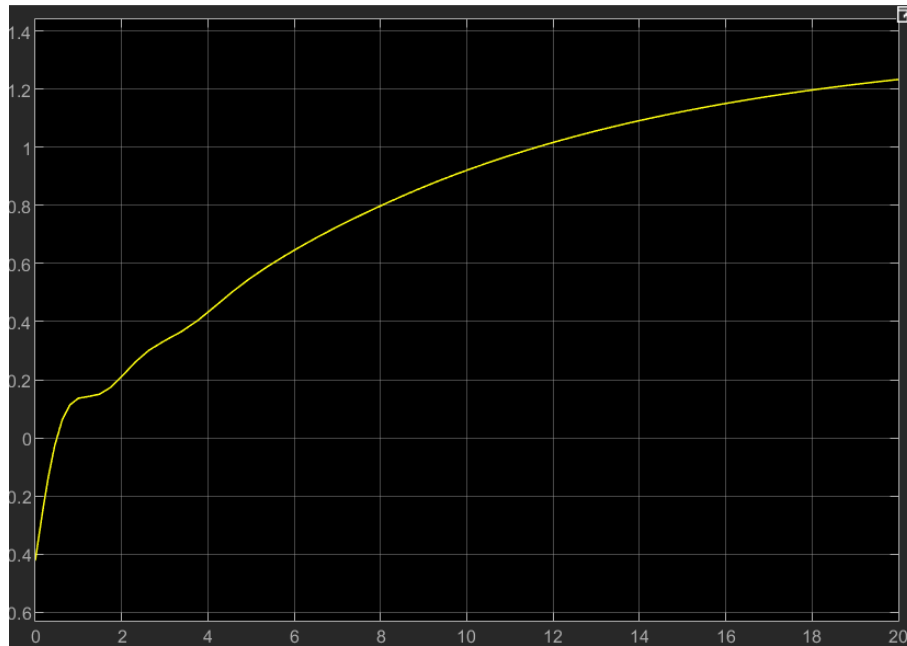


Figura 12 ingresso di controllo con LQR I

Notiamo che l'ingresso di controllo è sicuramente contenuto e non presenta picchi, ma d'altra parte l'uscita non soddisfa le richieste. In particolare, non presenta sovraelongazione, ma ha una dinamica piuttosto lenta e non soddisfa il requisito sul tempo di assestamento.

II tentativo

Allora diminuisco il valore di R ed aumento Q_i , così da velocizzare il controllo.

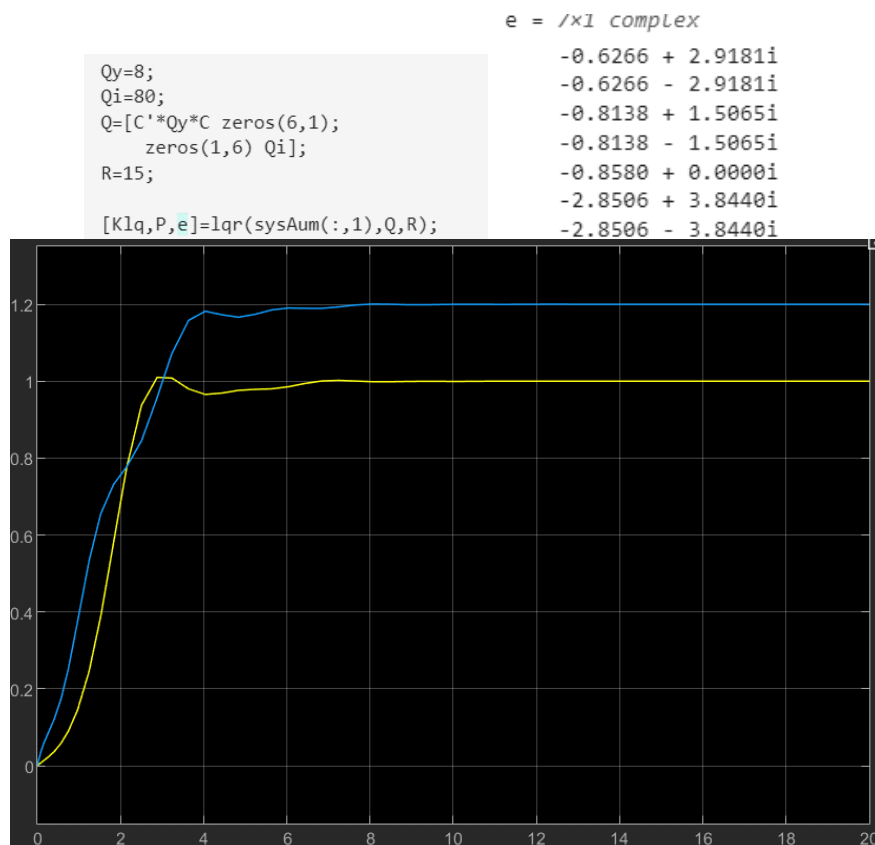


Figura 13 posizione massa 1 e massa 2 con LQR II

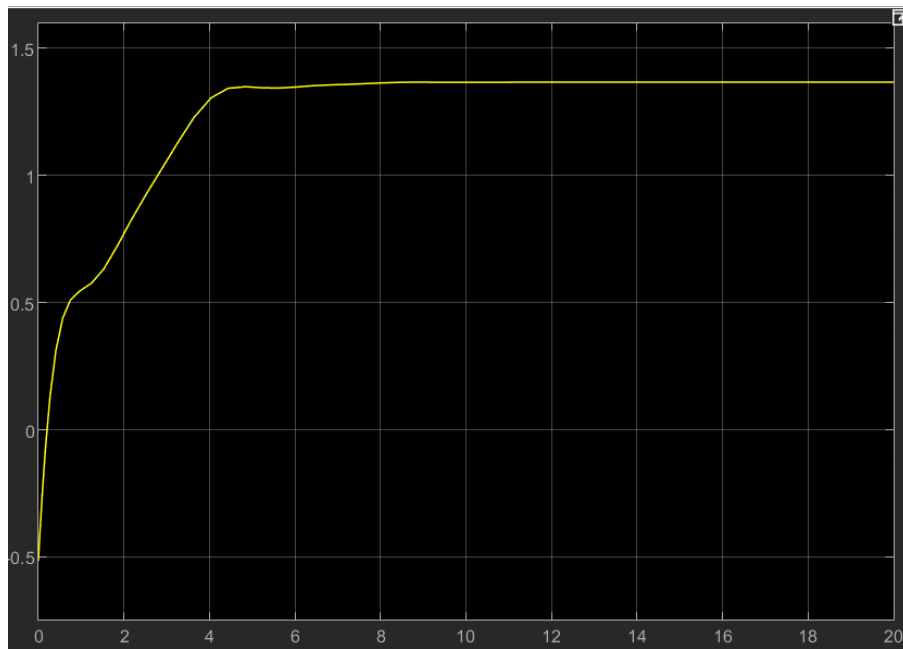


Figura 14 ingresso di controllo con LQR II

Questo sembra essere il giusto trade-off, poiché l'ingresso di controllo è accettabile e l'uscita soddisfa tutti i requisiti imposti.

Funzione di trasferimento a ciclo chiuso

```
function clLoop=closedLoop(A,B,C,D,Kaum)
G=ss(A,B,C,D);
G.InputName={'ForceU','ForceW'};
G.OutputName={'Pos1','Vel1','Pos_ruota','Vel_ruota','Pos2','Vel2'};

Ki=Kaum(1,7);

sysItg=ss(0,1,Ki,0);
sysItg.InputName='errorPos1';
sysItg.OutputName='IntegralAction';

Kf=Kaum(:,1:6);
StFeedMatrix=ss([],[],[],Kf);
StFeedMatrix.InputName={'Pos1','Vel1','Pos_ruota','Vel_ruota','Pos2','Vel2'};
StFeedMatrix.OutputName='StateFeedbackAction';

Sum1=sumblk('errorPos1=Ref-Pos1');
Sum2=sumblk('ForceU=-IntegralAction-StateFeedbackAction');
clLoop=connect(G,sysItg,StFeedMatrix,Sum1,Sum2,{'Ref','ForceW'},{'Pos1','Vel1','Pos_ruota','Vel_ruota','Pos2','Vel2','ForceU'});
end
```

```
ClLQ=closedLoop(A,B,eye(6),zeros(6,2),Kaum)
```

```
ClLQ =
```

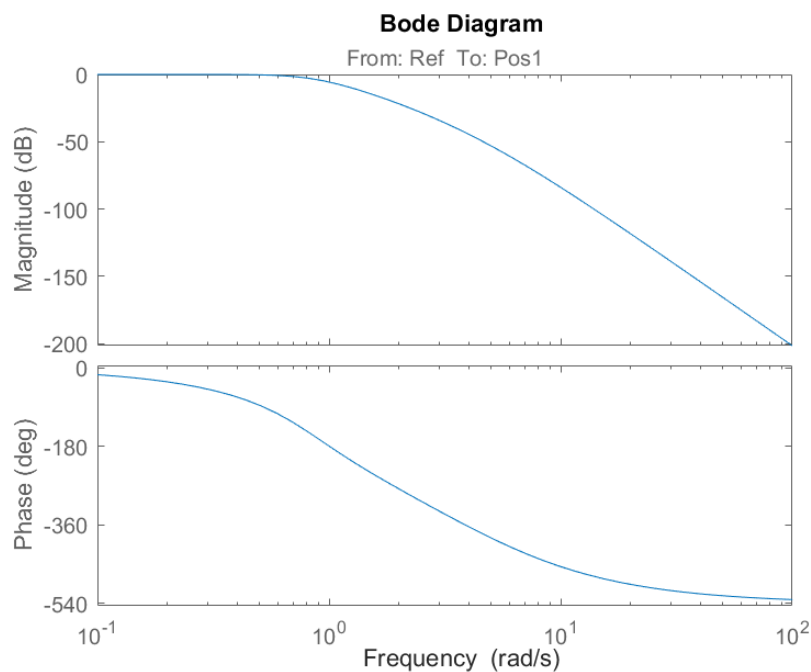
```
A =
```

	x1	x2	x3	x4	x5	x6	x7
x1	0	1	0	0	0	0	0
x2	-5	-2	1	0.4	0	0	0
x3	0	0	0	1	0	0	0
x4	55.56	22.22	-24.44	-4.444	55.56	0	0
x5	0	0	0	0	0	1	0
x6	2.569	-6.201	3.45	1.005	-25.3	-11.56	3.803
x7	-1	0	0	0	0	0	0

```
B =
```

	Ref	ForceW
x1	0	0
x2	0	0
x3	0	0
x4	0	22.22

```
bode(ClLQ(1,1))
```



La funzione a ciclo chiuso ha una banda passante a -3dB di circa 0.82 rad/s.

Allocazione autovalori vs controllo LQ

Appreziamo le differenze che ci sono nel caso in cui la matrice di retroazione di stato viene calcolata con il metodo di assegnamento degli autovalori e quando viene calcolata mediante il controllo ottimo.

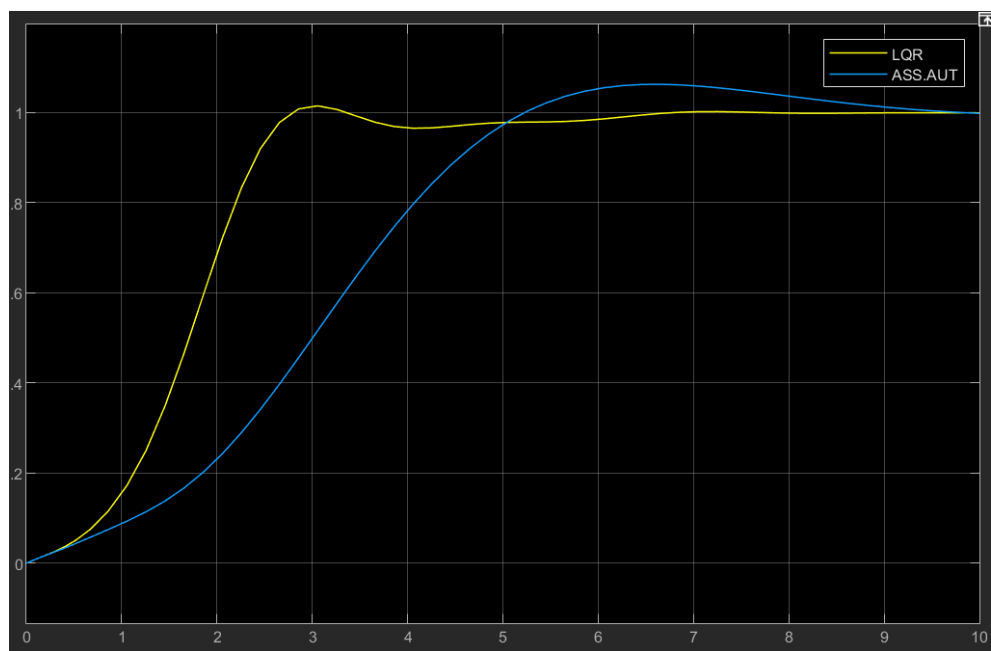
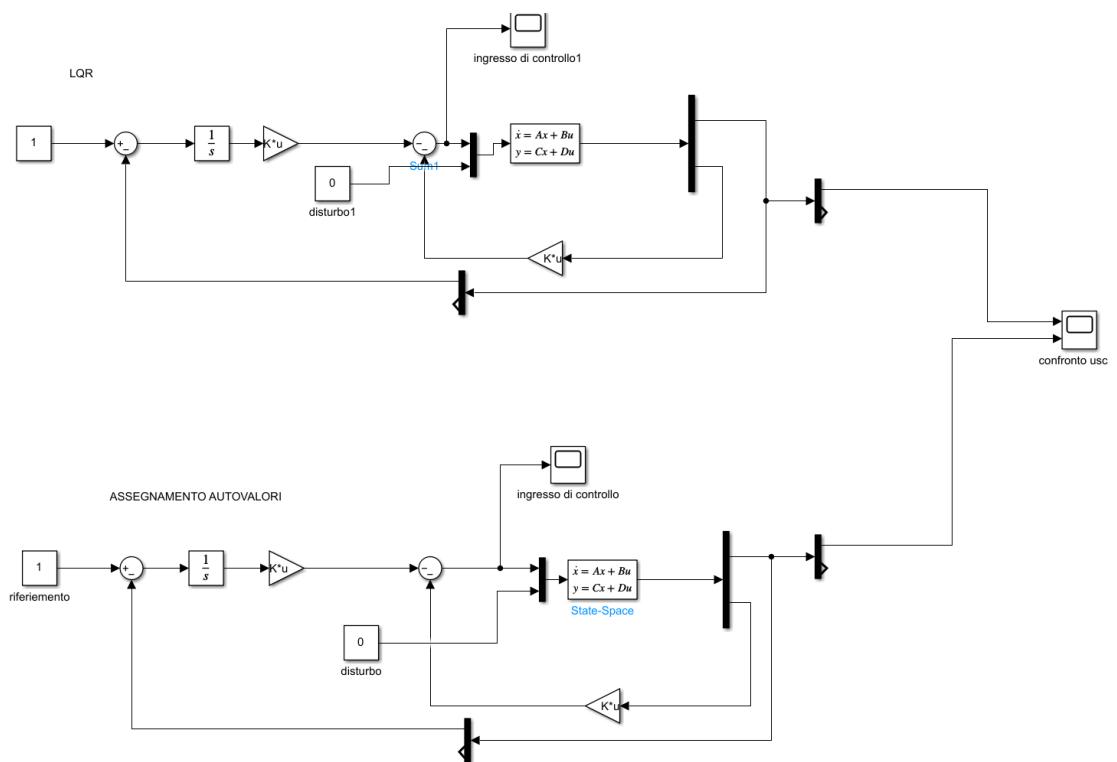


Figura 15 confronto uscite

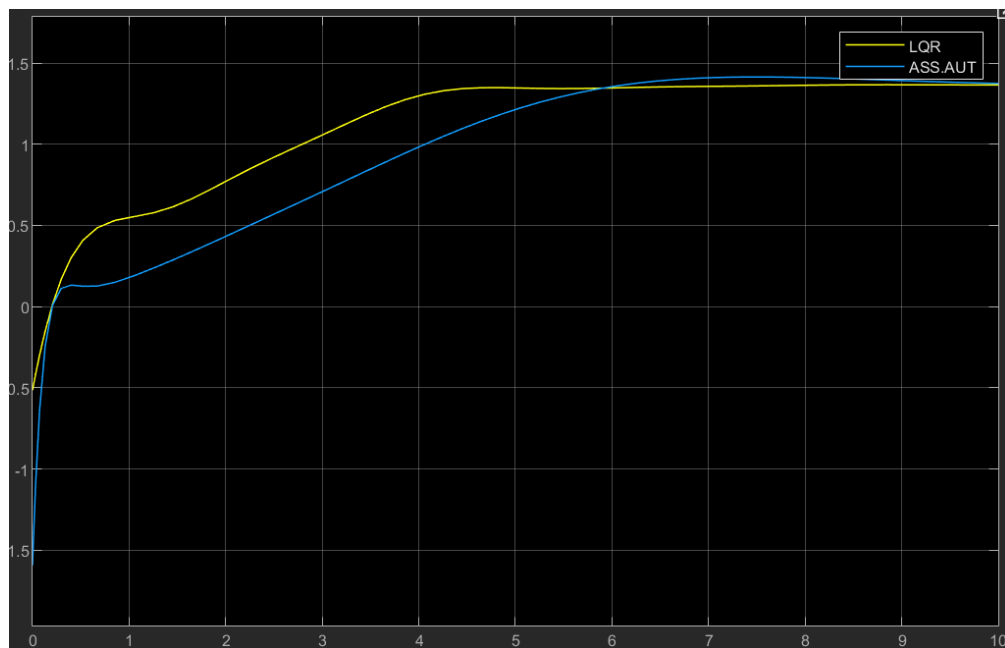


Figura 16 confronto ingresso di controllo

Attraverso il controllo LQ la dinamica dell'uscita è migliore e presenta una sovraelongazione minima. Nonostante ciò, anche l'alternativa classica con l'assegnamento degli autovalori è molto valida.

Controllo LQG

Il controllo LQG (Lineare Quadratico Gaussiano) è un controllo utilizzato per processi stocastici, ovvero sistemi del tipo:

$$\dot{x} = Ax + Bu + v_x$$

$$y = Cx + Du + v_y$$

$$x(t_0) = x_0$$

dove v_x è il rumore di processo mentre v_y il rumore di misura.

In questo caso il funzionale di costo da minimizzare è

$$J = \lim_{T \rightarrow \infty} \frac{1}{T} E \left[\int_0^T x^T Q x + u^T R u \, dt \right]$$

essendo ora J una variabile aleatoria e non deterministica

Questa strategia di controllo adotta lo schema che presenta azione integrale ed osservatore. Tuttavia, questa strategia sostituisce l'allocazione degli autovalori con il controllo LQR e sostituisce l'osservatore classico con il Filtro di Kalman.

Filtro di Kalman

Il filtro di Kalman è un osservatore stocastico; esso tiene conto del rumore di misura (presente sull'uscita dell'impianto) e del rumore di processo (presente in ingresso all'impianto).

Questi rumori (bianchi e gaussiani) sono caratterizzati dalla matrice di covarianza V , la quale è così definita:

$$E[v(t_1)v^T(t_2)] = V\delta(t_1 - t_2)$$

Dove V è formata dalle seguenti sottomatrici:

$$V = \begin{pmatrix} \tilde{Q} & Z \\ Z^T & \tilde{R} \end{pmatrix}$$

La matrice \tilde{Q} è la matrice relativa alla varianza del rumore di processo; mentre \tilde{R} è la matrice relativa alla varianza del rumore di misura. La matrice Z verrà considerata nulla poiché supponiamo che rumore di processo e rumore di misura siano segnali aleatori incorrelati fra loro.

Minimizzare la varianza di un segnale aleatorio si traduce nella riduzione dell'incertezza di quest'ultimo; di conseguenza, la minimizzazione della varianza rende il mio osservatore in grado di offrire reiezione al rumore ottima.

Quindi il filtro di Kalman è un osservatore ottimo che si cura di minimizzare la traccia della matrice di covarianza. La minimizzazione della traccia di V si traduce anche nella minimizzazione di tutti i termini presenti al di fuori della diagonale, i quali rappresentano il grado di correlazione tra i segnali aleatori in gioco.

Mettendo insieme un controllore ottimo LQ ed un filtro di Kalman si ottiene un controllo ottimo LQG. Si dimostra che ciò è possibile se sono verificate le seguenti ipotesi

1. $Q = H^T H \geq 0$
2. $R > 0$
3. $\tilde{Q} = MM^T \geq 0$
4. $\tilde{R} > 0$
5. (A, H) rilevabile
6. (A, B) stabilizzabile
7. (A, M) stabilizzabile
8. (A, C) rilevabile

A questo punto devo utilizzare il comando *kalman*. Ho bisogno delle matrici Q ed R.

ERRORE DI MISURA

Poiché il mio obiettivo è controllare la posizione di un carrello, suppongo di utilizzare dei sensori di posizione con precisione 5cm. Voglio che la varianza dell'errore di misura sia contenuto nel mio range di interesse. Prendo la banda passante a ciclo chiuso calcolata nel paragrafo precedente e la approssimo a 0.85 rad/s. allora:

$$f = w/2\pi$$

Da cui ricavo $f = 0.136 \text{ Hz}$ e quindi:

```
sys=ss(A,B(:,1),C(1,:),D(1,2));
```

```
Rk=sqrt(5e-2/3)/2/0.136
```

```
Rk = 0.4746
```

ERRORE DI PROCESSO

Se la misura è poco rumorosa, al filtro di Kalman conviene usare molto la misura e poco il modello. Invece, e è molto rumorosa, la velocità di convergenza sarà simile alla velocità di convergenza del sistema a ciclo aperto quindi più lento.

Per avere una buona matrice di convergenza utilizzo una matrice Q più piccola rispetto alla matrice R:

```
Qk=1e-3*eye(6)
```

```
Qk = 6x6
10^-3 x
    1.0000    0    0    0    0    0
    0    1.0000    0    0    0    0
    0    0    1.0000    0    0    0
    0    0    0    1.0000    0    0
    0    0    0    0    1.0000    0
    0    0    0    0    0    1.0000
```

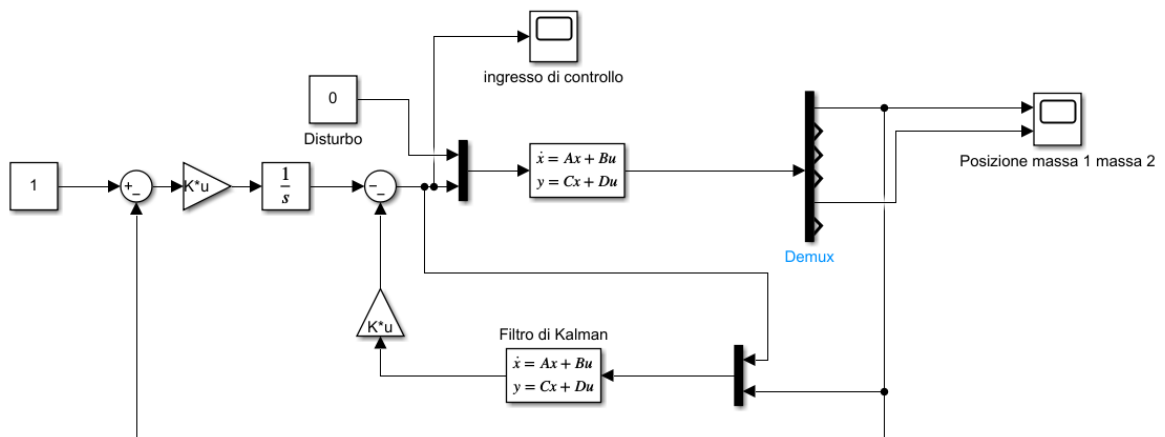
```
sys4Kalman=ss(A,[B(:,1) eye(6)],C(1,:),[D(1,1) zeros(1,6)]);
[kalmf,Lk,Pk] = kalman(sys4Kalman,Qk,Rk);
norm(Lk)
```

```
ans = 0.0342
```

```
eig(kalmf)
```

```
ans = 6x1 complex
-2.8525 + 3.8453i
-2.8525 - 3.8453i
-0.3090 + 2.7900i
-0.3090 - 2.7900i
-0.0655 + 1.2953i
-0.0655 - 1.2953i
```

```
AobsK=kalmf.A;
BobsK=kalmf.B;
CobsK=kalmf.C(1:end,:);
DobsK=kalmf.D(1:end,:);
```



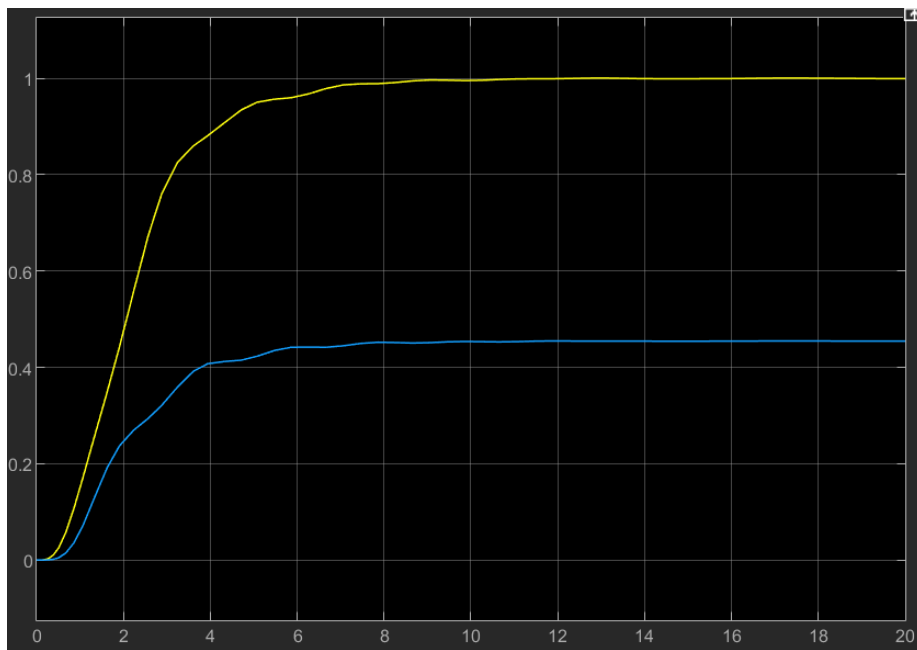


Figura 17 posizione massa 1 e massa 2

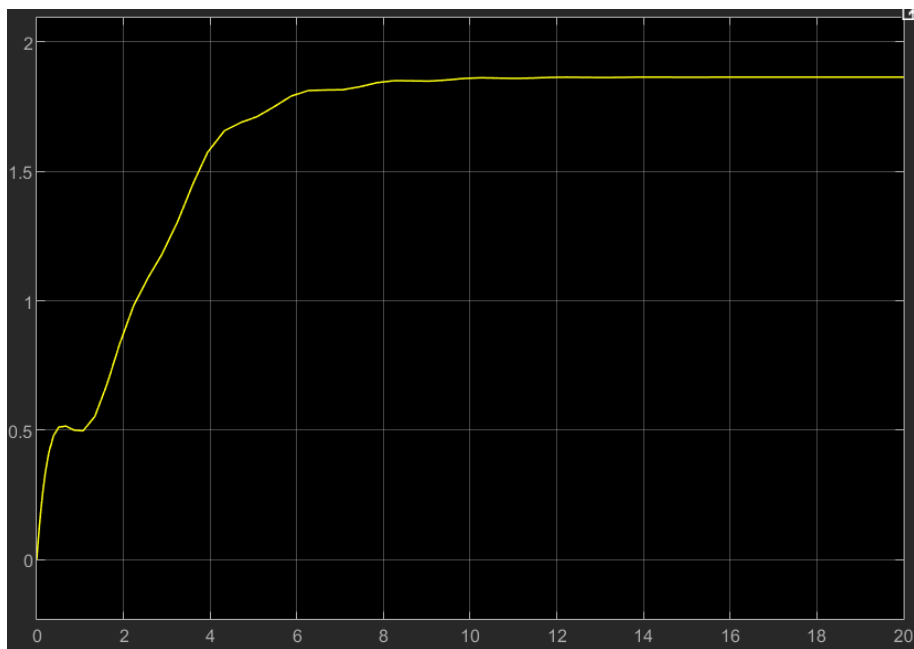


Figura 18 ingresso di controllo

Le uscite rispettano le specifiche imposte, il controllo è contenuto ma è leggermente aumentato in ampiezza rispetto al controllo LQ.

Adesso andiamo ad aggiungere l'errore di misura e l'errore di processo utilizzando il blocco *BAND LIMITED WHITE NOISE*.

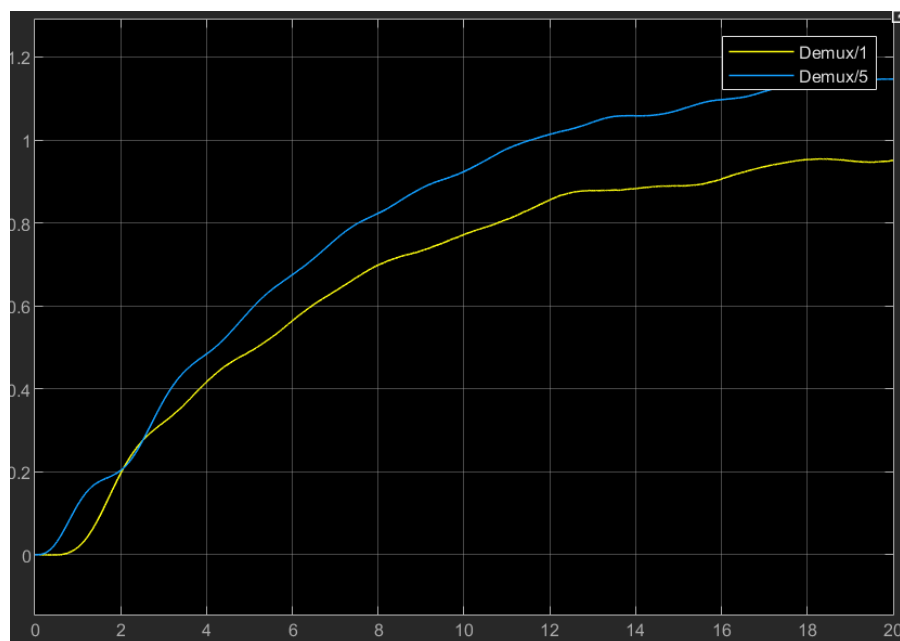
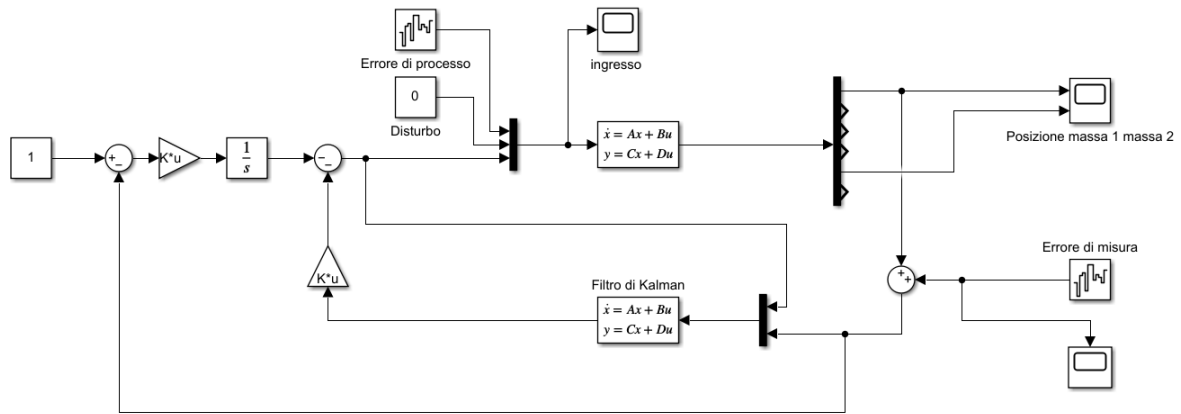


Figura 19 posizione massa 1 massa 2

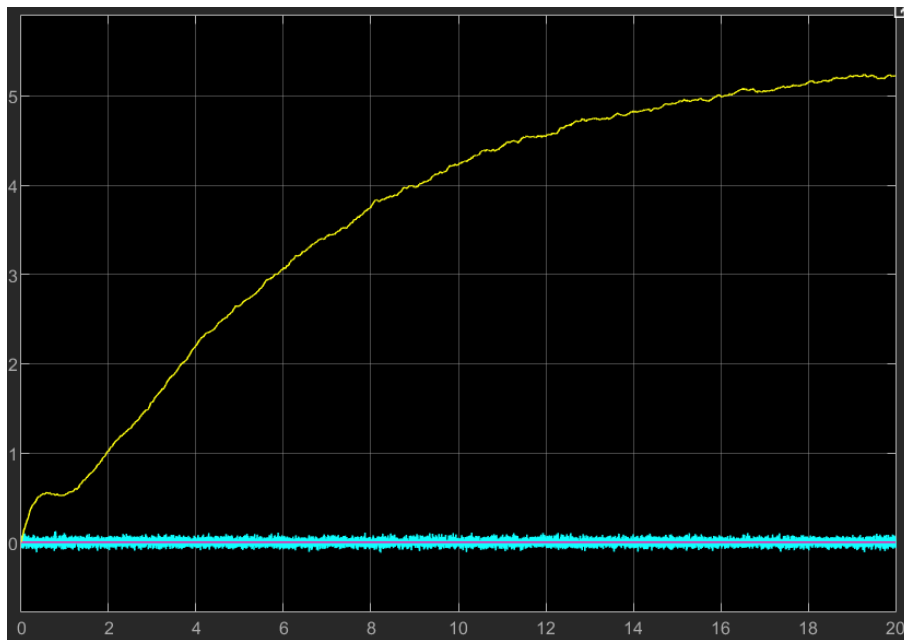


Figura 20 ingresso di controllo e disturbo

In questo caso, rispetto al controllo fatto con allocazione degli autovalori, si nota un rallentamento della dinamica del sistema, ma una più forte reiezione al disturbo

Osservatore vs filtro di Kalman

Vediamo cosa succede se inserisco rumore di misura in uno schema con osservatore classico e confrontiamo i risultati con lo schema precedente.

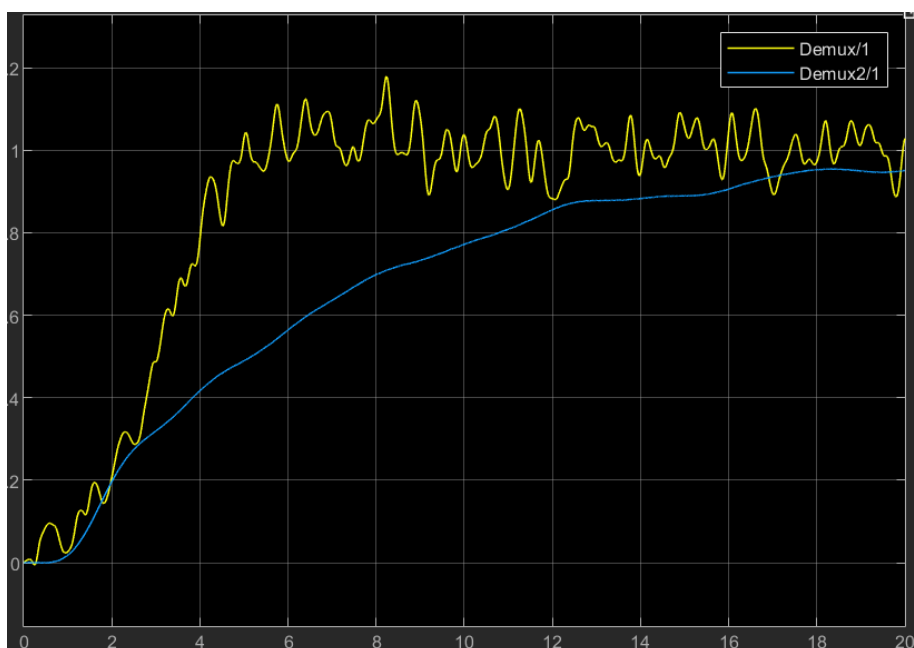


Figura 21 confronto uscite

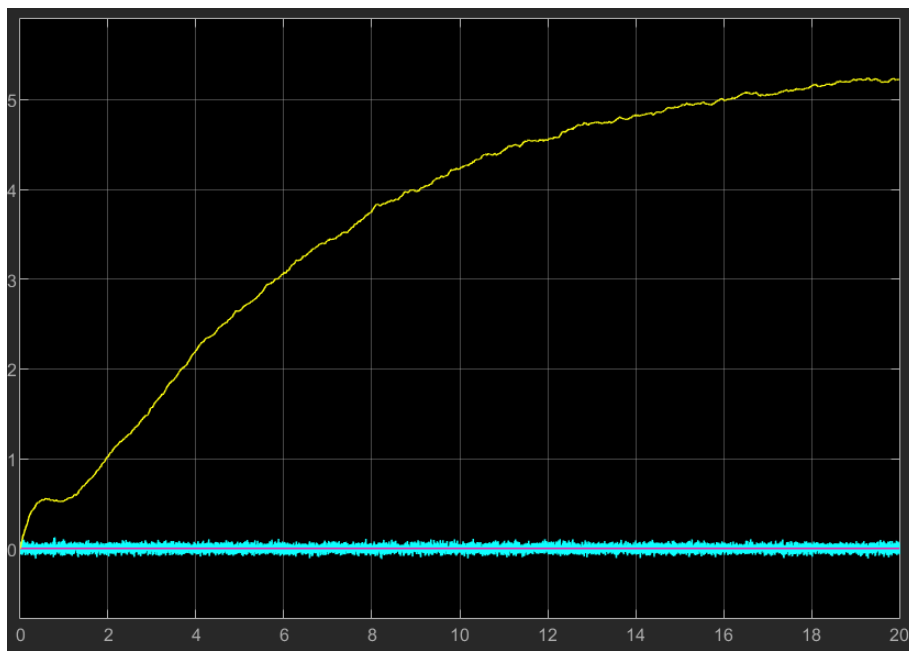
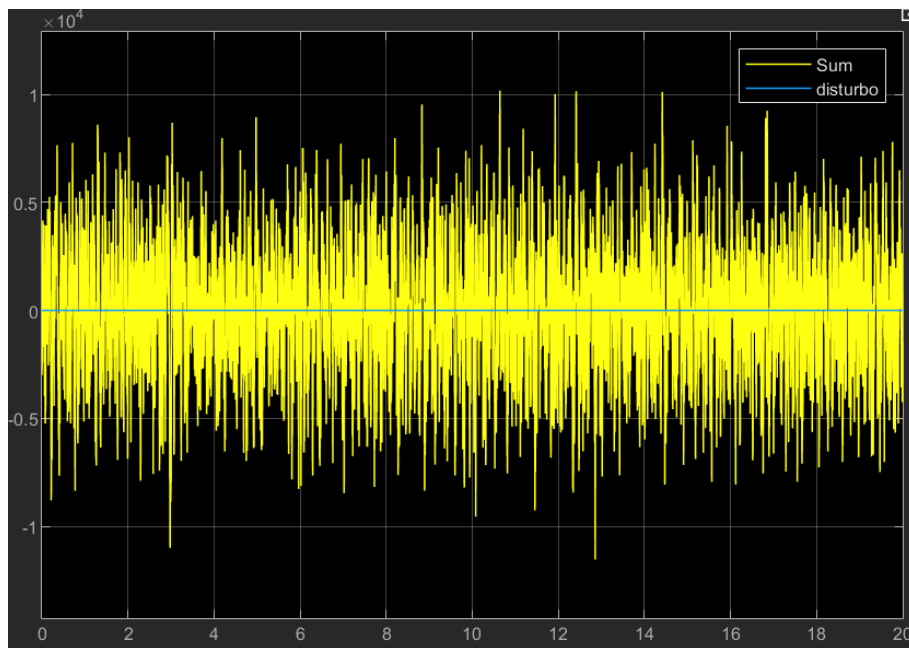


Figura 22 ingresso con filtro di Kalman



Il filtro di Kalman offre prestazioni ottimali in presenza di rumore di misura e di processo a differenza dell'osservatore classico. C'è però da sottolineare il fatto che quest'ultimo funziona molto meglio rispetto a filtro di Kalman in presenza di disturbi costanti.

Dalle simulazioni portate avanti si evince che la tecnica di controllo LQG è la migliore implementabile in termini di prestazioni dinamiche, di azione di controllo e di reiezione al rumore.

Controllo H^∞ e Mixed Sensitivity Design

L'obiettivo del controllo H^∞ con mixed sensitivity design è ottenere un controllore che offre prestazioni elevate in presenza di incertezze e perturbazioni, bilanciando efficacemente la sensibilità alle variazioni del segnale di riferimento e alle perturbazioni nel sistema controllato.

In altre parole, mira a progettare un controllore per un sistema dinamico in modo da massimizzare le prestazioni nel contesto della massima tolleranza possibile verso perturbazioni e incertezze nel sistema.

Parto con il creare il mio sistema e le 3 diverse matrici di peso da utilizzare.

W_t è la matrice diagonale che va a pesare la matrice di sensitività complementare, cioè la matrice del sistema a ciclo chiuso. Questa dovrà avere un guadagno statico molto basso per garantire fedeltà alle basse frequenze. Man mano che la frequenza aumenta, la fedeltà diminuisce e quindi W_t assumerà valori più elevati. Sulla diagonale troviamo:

$$W_{S_{i,i}}(s) = \frac{1 \frac{s}{\omega_2} + 1}{\alpha \frac{s}{\omega_1} + 1}$$

W_s è la matrice diagonale che va a pesare la matrice di sensitività. Ha un comportamento speculare a quello di W_t ; presenterà quindi prima uno zero e poi un polo. Sulla diagonale troviamo gli elementi:

$$W_{T_{i,i}}(s) = \beta \frac{\frac{s}{\omega_1} + 1}{\frac{s}{\omega_2} + 1}$$

Infine, W_k è una matrice diagonale costante che pesa l'azione del controllo su tutta la banda. È equivalente alla matrice R per il controllo LQ.

In conclusione, troverò:

$$G_{zw}(s) = \begin{bmatrix} W_S(s)S(s) \\ W_K(s)R(s)S(s) \\ W_T(s)T(s) \end{bmatrix}$$

Una volta scelte le opportune matrici di peso, non resta che trovare il controllore $R(s)$ che minimizzi la norma hinfinito della matrice di trasferimento:

$$\|G_{zw}(s)\|_\infty < \gamma$$

```
sysInf=ss(A,B(:,1),C(1,:),D(1,1));
Ws= zpk(makeweight(100,[1,1],1/100))
```

Ws =

$$\frac{0.01 (s+100)}{(s+0.01)}$$

Continuous-time zero/pole/gain model.
[Model Properties](#)

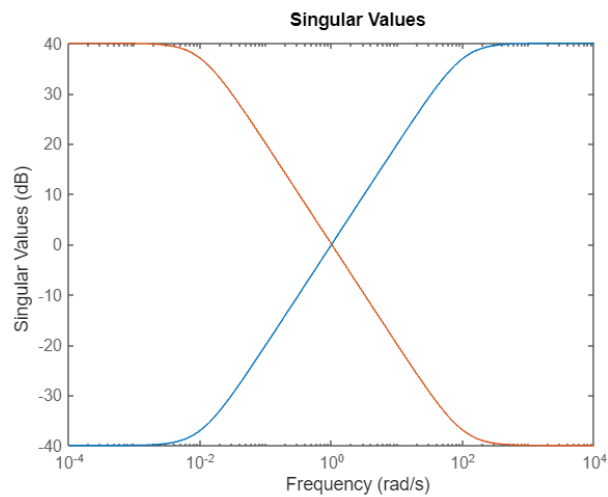
```
Wt=zpk(makeweight(1/100,[1,1],100))
```

Wt =

$$\frac{100 (s+0.01)}{(s+100)}$$

Continuous-time zero/pole/gain model.
[Model Properties](#)

```
Wk=11;
sigma(Wt,Ws)
```



```
[Kinf,CL,gamma]=mixsyn(sysInf,Ws,Wt,Wk)
```

Kinf =

```
A =
      x1      x2      x3      x4      x5      x6      x7      x8
x1  -0.01  1.981e-17  3.669e-16  7.49e-17 -6.242e-18  4.144e-18  6.615e-17  2.3e-17
x2   5.839  -6.267  -7.667  -7.108  0.7736  -0.416  -5.541  -4.062
x3  -9.569e-18  1.278e-18  2.367e-17  1  -4.026e-19  2.673e-19  4.266e-18  1.483e-18
x4  -1.539e-18  2.055e-19  -5  -2  1  0.4  6.862e-19  2.386e-19
x5  -3.861e-17  5.156e-18  9.551e-17  1.95e-17 -1.625e-18  1  1.722e-17  5.987e-18
x6  1.653e-17 -2.207e-18  55.56  22.22  -24.44  -4.444  55.56 -2.563e-18
x7  -2.304e-18  3.077e-19  5.7e-18  1.164e-18 -9.696e-20  6.437e-20  1.028e-18  1
x8   0.292   4.687  -0.3834  -0.3554  0.7054  -0.0208  -7.61  -0.2031
```

```
B =
      u1
x1      1.25
x2  6.184e-20
x3  1.432e-17
gamma = 10.9919
```

```
L=sysInf*Kinf;
S=1/(1+L);
T=minreal(L*S);
```

21 states removed.

```
sigma(L,S,T);
grid
```

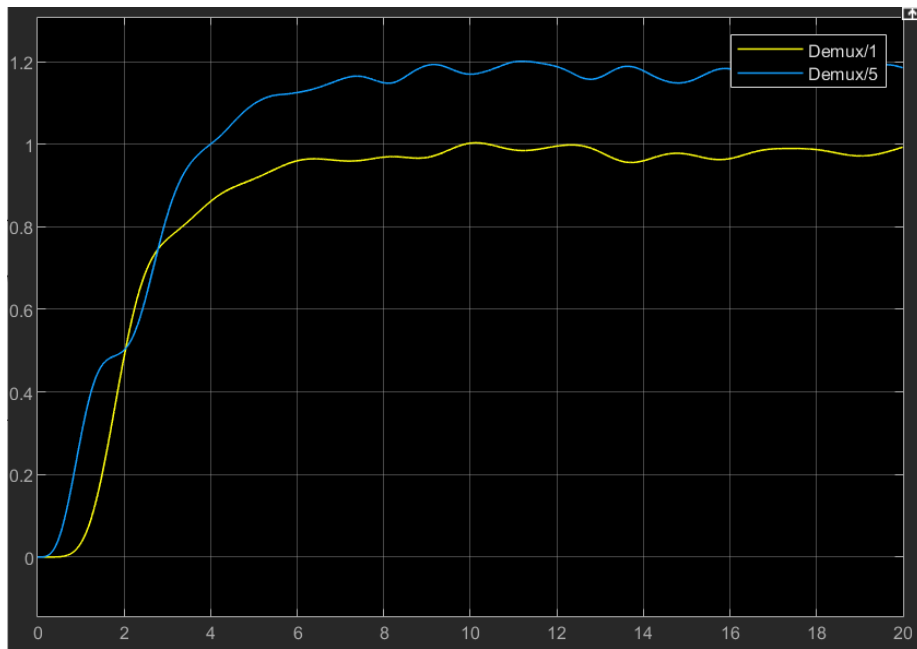



Figura 24 posizione massa 1 e massa 2

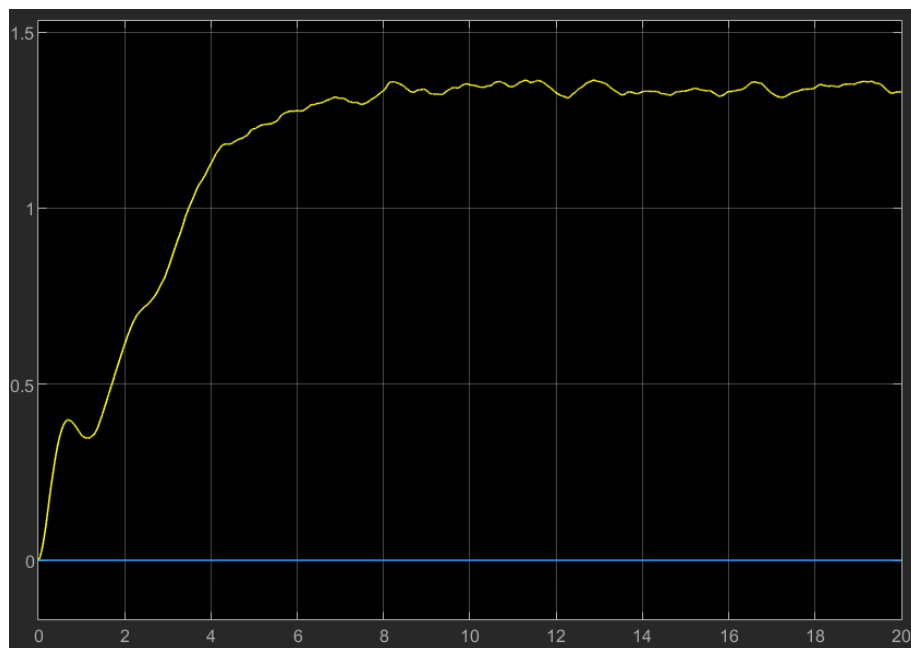


Figura 25 ingresso di controllo + disturbo

Come si evince dai risultati, il tempo di assestamento è aumentato e l'uscita non va a regime, ma ha un moto oscillatorio. Ciò è dovuto al fatto che in questo tipo di controllo non è prevista l'azione integrale.

Per eliminare l'errore a regime creo una '*quasi azione integrale*' andando a sfruttare i poli della matrice W_s . In particolare, porto il polo molto vicino all'origine, così da simulare appunto un polo in 0 e quindi un'azione integrale.

```
[Ak,Bk,Ck,Dk]=ssdata(Kinf)
```

```
Ak = 8x8
    -0.0100    0.0000    0.0000    0.0000   -0.0000    0.0000    0.0000    0.0000
     5.8392   -6.2670   -7.6670   -7.1081    0.7736   -0.4160   -5.5412   -4.0616
    -0.0000    0.0000    0.0000    1.0000   -0.0000    0.0000    0.0000    0.0000
    -0.0000    0.0000   -5.0000   -2.0000    1.0000    0.4000    0.0000    0.0000
    -0.0000    0.0000    0.0000    0.0000   -0.0000    1.0000    0.0000    0.0000
     0.0000   -0.0000   55.5556   22.2222  -24.4444   -4.4444   55.5556   -0.0000
    -0.0000    0.0000    0.0000    0.0000   -0.0000    0.0000    0.0000    1.0000
     0.2920    4.6869   -0.3834   -0.3554    0.7054   -0.0208   -7.6104   -0.2031
```

```
Bk = 8x1
     1.2499
     0.0000
     0.0000
     0.0000
     0.0000
    -0.0000
     0.0000
    -0.0000
```

```
Ck = 1x8
     0.0730     1.1717    -0.0958    -0.0889     0.0097    -0.0052    -0.0693    -0.0508
```

```
Dk = 0
```

```
[z,p,k]=zpkdata(Kinf)
```

```
z = 1x1 cell array
    {7x1 double}
p = 1x1 cell array
    {8x1 double}
k = 0.0912
```

```
p{1}
```

```
ans = 8x1 complex
    -2.7625 + 3.8771i
    -2.7625 - 3.8771i
    -0.4926 + 3.2887i
    -0.4926 - 3.2887i
    -0.0100 + 0.0000i
    -2.1594 + 2.1289i
    -2.1594 - 2.1289i
    -2.0854 + 0.0000i
```

```
%sposto polo in 0.01 nell'origine
p{1}(5)=0;
p{1}
```

```
ans = 8x1 complex
    -2.7625 + 3.8771i
    -2.7625 - 3.8771i
    -0.4926 + 3.2887i
    -0.4926 - 3.2887i
     0.0000 + 0.0000i
    -2.1594 + 2.1289i
    -2.1594 - 2.1289i
    -2.0854 + 0.0000i
```

```
Kinf1=ss(zpk(z,p,k));
[Ak1,Bk1,Ck1,Dk1]=ssdata(zpk(z,p,k));
L1=sysInf*Kinf1;
S1=minreal(1/(1+L));
```

```
7 states removed.
```

```
T1=minreal(L*S)
```

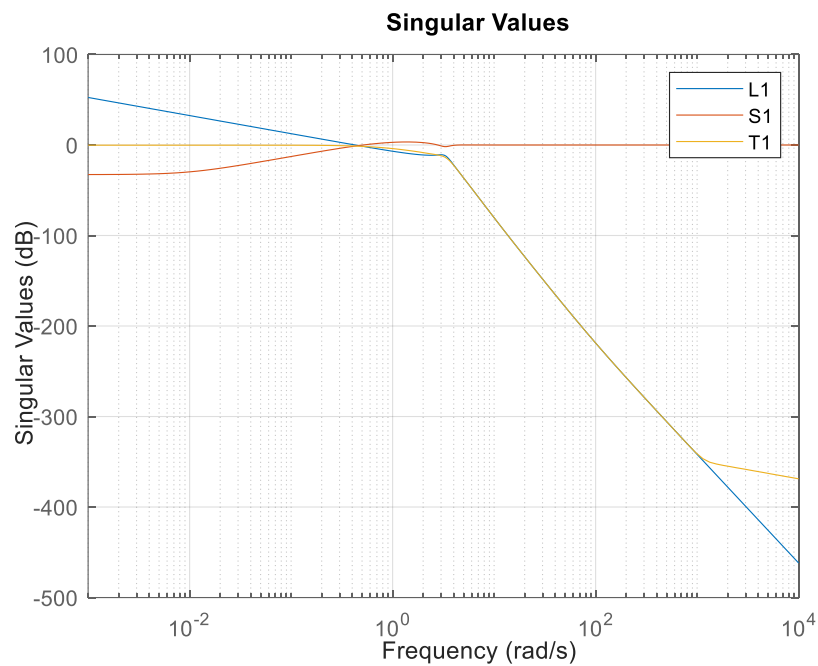
```
21 states removed.
```

```
T1 =
```

```
A =
      x1      x2      x3      x4      x5      x6      x7      x8
x1    2.189   -8.668   -1.751   -0.7746    0.7554   -4.027    1.514   -6.136
x2    8.751   -7.205    6.499    2.92     3.16   -9.387   -2.339   -12.88
x3   -2.158    0.4083   -3.27   -0.5617   -1.18    2.729    0.7538    3.175
x4    2.998   -10.27   -21.37   -1.545   15.39   -0.9116   -10.73   -2.632
x5     1.11    0.926    5.883    0.3585   -2.689   -0.6041    1.902   -1.901
x6   0.07197  -0.2736    1.462   -1.618   -5.621   -0.935    3.622   -1.408
x7   -3.211    3.768    0.557   -0.05   -2.029    2.741    1.181    3.178
x8    0.1459   -5.594   -7.905   -0.6732    5.985    0.343   -2.697   -0.6507
```

```
B =
      u1
x1   -0.3414
```

```
sigma(L1,S1,T1, {1e-3,1e4})
grid
```



Noto che adesso L1 non ha più pendenza nulla, ma di -20dB/dec proprio come un integratore.

In simulink, se confronto l'uscita precedente con quella attuale vedo che non ha subito grossi cambiamenti, ma l'uscita adesso oscilla attorno ad 1.

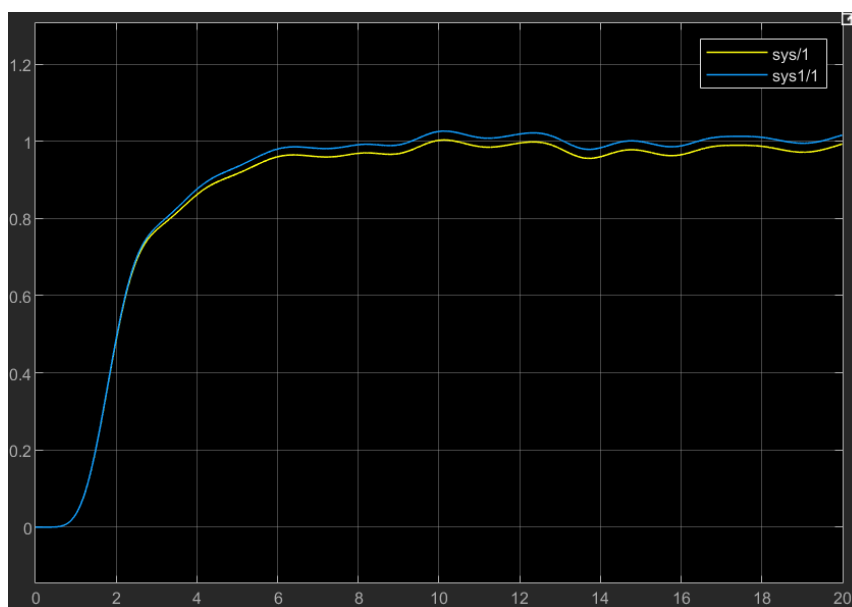


Figura 26 confronto uscite sysInf e sysInf1

Aggiungo l'errore di processo ed un disturbo diverso da 0:

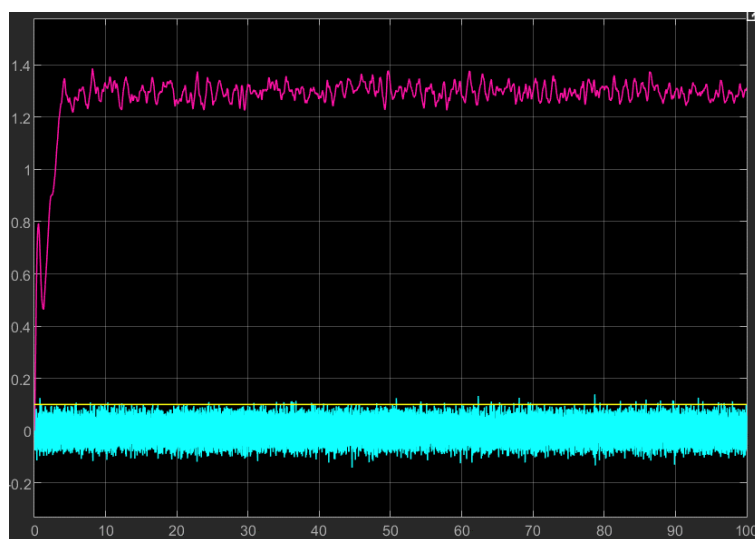
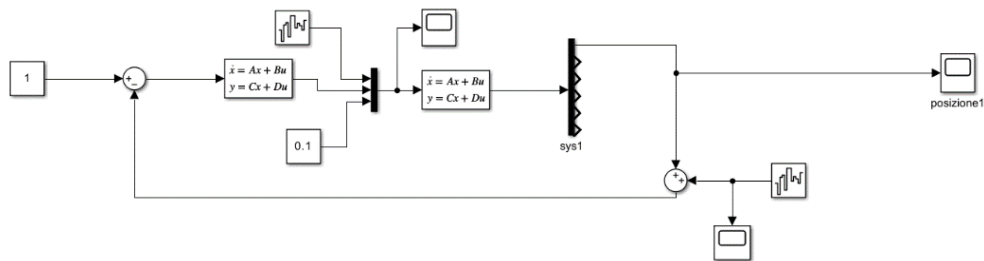


Figura 27 ingresso con errore di processo

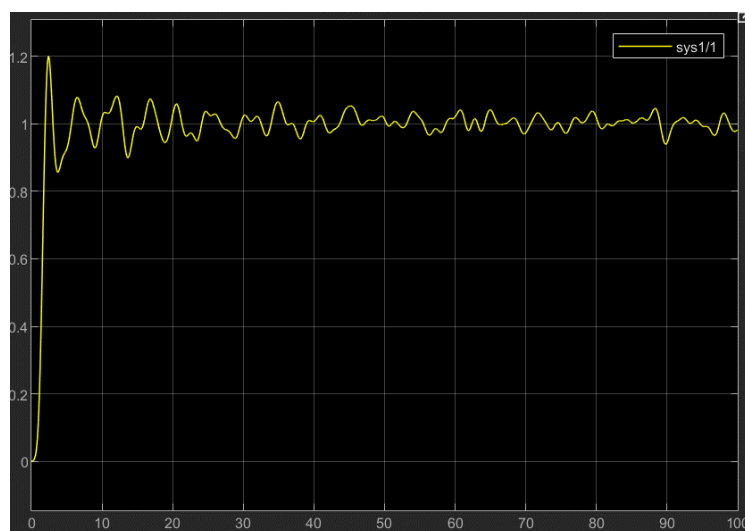


Figura 28 uscita con errore di processo e disturbo

Il disturbo eccita l'osservatore, il quale influenza la retroazione di stato e quindi l'evoluzione del sistema. Di conseguenza aumenta il tempo di assestamento dell'uscita.