

Progetto di Controllo dei Sistemi Incerti: Tecniche per il controllo di un veicolo a guida autonoma per l'inseguimento di una traiettoria



Francesco Bechelli

Cesare Palamara

Alberto Regoli

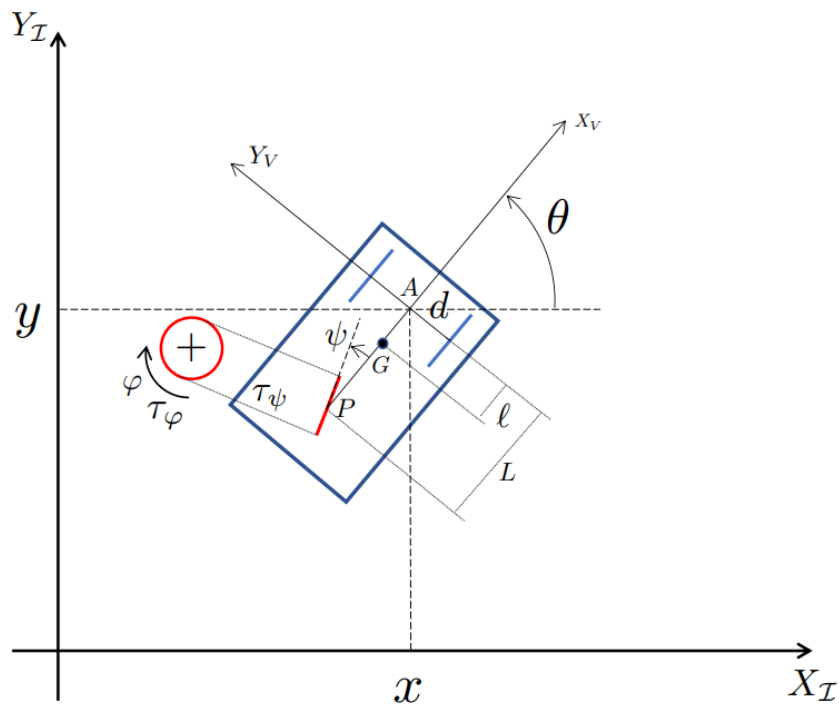
Anno Accademico 2022/2023

Università di Pisa, Ingegneria Robotica e dell' Automazione

Indice

1 - Introduzione e descrizione del problema	3
Equazioni del modello	4
Parametri geometrici e sensori	5
2 - Obiettivo del controllo	6
3 - Modello linearizzato	7
Condizioni iniziali	8
4 - Modellazione incertezze	9
LQG	9
Mix-Sensitivity e D-K iteration	9
Implementazione Matlab	9
5 - LQG	11
Introduzione teorica	11
Implementazione Matlab-Simulink	13
Risultati	14
LQG con integratore	15
LQG con modello non lineare	16
6 - Mixed-Sensitivity \mathcal{H}_∞	17
Introduzione teorica	17
Peso sulle prestazioni	18
Peso sugli ingressi	19
Implementazione Matlab-Simulink	20
Sistema non lineare con Mixed-Sensitivity \mathcal{H}_∞	22
7 - D-K iteration	23
Introduzione teorica	23
Implementazione Matlab-Simulink	24
Sistema non lineare con D-K iteration	27
8 - Risultati e confronti	28
Risultati caso lineare \mathcal{H}_∞ e D-K	28
Analisi di robusta stabilità e robusta prestazione	30
Implementazione Matlab	30
Confronto robusta stabilità e robusta prestazione	33
9 - Bibliografia e sitografia	38

1 - Introduzione e descrizione del problema



Il sistema in figura rappresenta un veicolo industriale a guida autonoma (AGV) la cui configurazione sul piano di moto è indicata dalle coordinate x e y e dalla direzione θ rispetto ad un sistema di riferimento global $\{I\}$.

L'AGV è dotato di tre ruote: due folli anteriori ed una motoruota posteriore. Quest'ultima è attuata per mezzo della coppia di rotolamento τ_φ che ne permette i movimenti di traslazione ed attraverso la coppia di sterzo τ_ψ che ne permette la manovrabilità agendo sull'angolo di sterzo ψ della motoruota posteriore.

Siano, r_P ed r_A i raggi della ruota posteriore ed anteriore, rispettivamente, m_P ed m_A le loro masse, M_V la massa dell'intero veicolo, L la distanza tra le ruote anteriori e quella posteriore, l la distanza del centro di massa dall'asse delle ruote anteriori, d la lunghezza del semiasse delle ruote anteriori ed infine, I_{GZ} il momento di inerzia lungo l'asse Z del veicolo, I_{PZ} il momento di inerzia dello sterzo alla ruota posteriore, I_{PY} il momento di inerzia della ruota posteriore e I_{AY} quello della ruota anteriore.

Si assuma la presenza di attuatori per imprimere la coppia motrice τ_φ alla ruota posteriore e la coppia di sterzo τ_ψ .

Equazioni cinematica

$$\dot{x} = r_P \dot{\varphi} \cos(\theta) \cos(\psi)$$

$$\dot{y} = r_P \dot{\varphi} \sin(\theta) \cos(\psi)$$

$$\dot{\theta} = -\frac{r_P}{L} \dot{\varphi} \sin(\psi)$$

Equazioni dinamica

$$\tau_{\varphi} = (ar_P^2 \cos^2(\psi) + b \frac{r_P^2}{L^2} \sin^2(\psi) + I_{PY}) \ddot{\varphi} - I_{PZ} \frac{r_P}{L} \ddot{\psi} \sin(\psi) + (b \frac{r_P^2}{L^2} - ar_P^2) \dot{\varphi} \dot{\psi} \cos(\psi) \sin(\psi)$$

$$\tau_{\psi} = -I_{PZ} \frac{r_P}{L} \ddot{\varphi} \sin(\psi) + I_{PZ} \ddot{\psi} - I_{PZ} \frac{r_P}{L} \dot{\varphi} \dot{\psi} \cos(\psi)$$

Con

$$a = (M_V + m_P + 2m_A + 2 \frac{I_{AY}}{r_A^2})$$

$$b = \frac{1}{2} (M_V l^2 + m_P L^2 + 2m_A d^2 + I_{GZ} + I_{PZ} + 2I_{AZ} + 2I_{AY} \frac{d^2}{r_A^2})$$

Funzioni di trasferimento degli attuatori

Le funzioni di trasferimento nominali degli attuatori sono ben approssimabili con un modello del primo ordine con ritardo del tipo

$$\bar{G}_{\tau\psi} = \frac{\bar{K}_{m_1} e^{-T_1 s}}{\bar{T}_{m_1} s + 1}$$

per lo sterzo posteriore e uno del secondo ordine del tipo

$$\bar{G}_{\tau\varphi} = \frac{\bar{K}_{m_2}}{(\bar{T}_{m_2} s + 1)(\bar{T}_{m_2}/100 s + 1)}$$

per il rotolamento della motoruota posteriore.

Parametri geometrici e inerziali

<i>Parametro</i>	<i>Valore Nominale</i>	<i>Incertezza</i>	<i>Unità di misura</i>
l	0.4	-	[m]
L	1	-	[m]
d	0.3	-	[m]
r_p	0.3	$\pm 5\%$	[m]
r_a	0.3	$\pm 5\%$	[m]
M_v	15	-	[Kg]
m_p	0.5	-	[Kg]
m_a	0.5	-	[Kg]
I_{Gz}	0.655	-	[kg*m ²]
I_{Pz}	0.23	-	[kg*m ²]
I_{Py}	0.15	-	[kg*m ²]
I_{Ay}	0.12	-	[kg*m ²]
I_{Az}	0.8	-	[kg*m ²]
\bar{K}_{m1}	1.08	$\pm 10\%$	-
\bar{T}_{m1}	0.005	$\pm 20\%$	[s]
\bar{K}_{m2}	0.335	$\pm 10\%$	-
\bar{T}_{m2}	0.002	$\pm 20\%$	[s]
\bar{T}_1	$\bar{T}_{m1} / 2$	$\pm 5\%$	[s]

Si considerano i valori in tabella per quanto riguarda i valori nominali e le loro corrispondenti incertezze dei parametri geometrici e inerziali.

Sensori

Per lo sviluppo della trattazione si usano i seguenti sensori per l'acquisizione di dati necessari per il problema d'identificazione:

- Sensore per l'ordinata del punto medio assale anteriore y
- Sensore per l'orientazione nel piano θ
- Sensore per lo sterzo ψ
- Sensore per la velocità angolare della ruota posteriore $\dot{\phi}$

Per i sensori si utilizza un modello lineare.

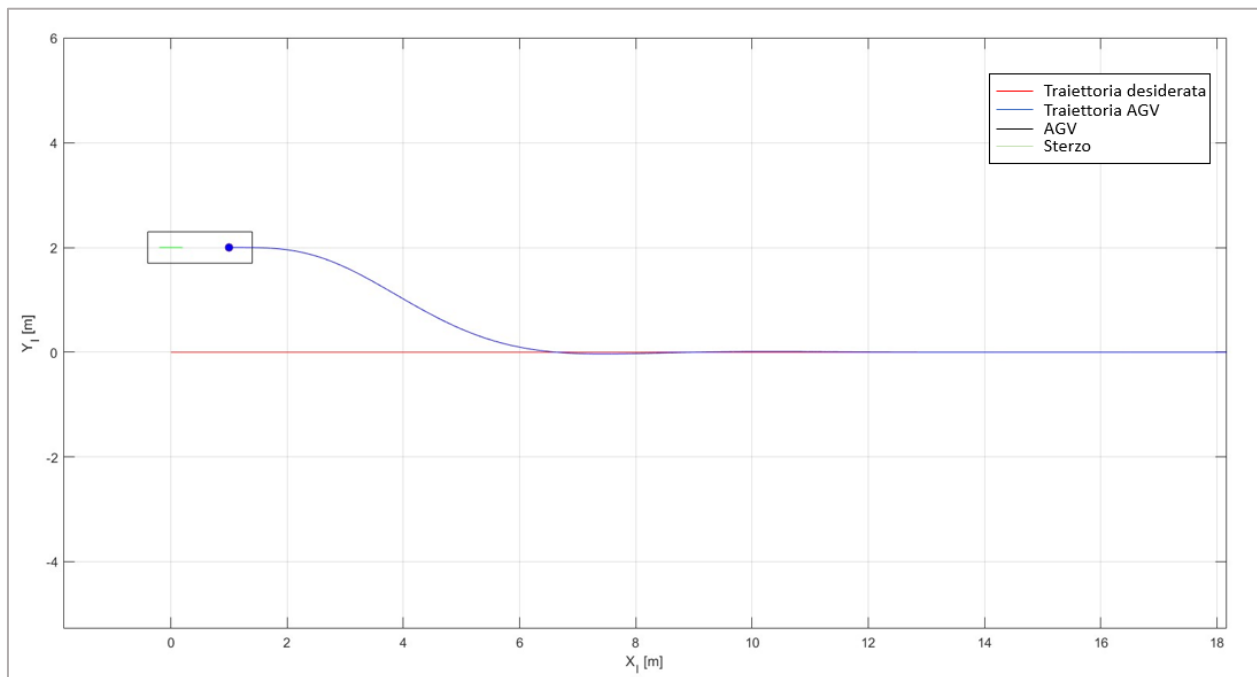
2 – Obiettivo del controllo

Lo scopo del progetto è di costruire con varie tecniche un controllore che permetta l'inseguimento della seguente traiettoria con determinata velocità:

$$\begin{aligned}y_I &= 0 \\ \theta_I &= 0 \\ \dot{\varphi} &= v/r_p\end{aligned}$$

dove v è la velocità desiderata di avanzamento dell' AGV.

Inoltre, si pone attenzione ai valori massimi assunti dagli ingressi agli attuatori per evitare un'eccessiva sollecitazione.



Le tecniche utilizzate per il controllo sono

- LQG (Linear Quadratic Gaussian)
- Mixed-Sensitivity \mathcal{H}_∞
- μ -synthesis e D-K iteration

Infine, si riporta il confronto in termini di robusta stabilità e performance degli ultimi due controllori.

3 – Modello Linearizzato

Per l'implementazione dei controllori si utilizza un modello linearizzato del problema intorno alla traiettoria desiderata.

Si definisce lo stato del sistema con le coordinate global:

$$\begin{pmatrix} y_I \\ \theta_I \\ \psi_I \\ \dot{\psi}_I \\ \dot{\varphi}_I \end{pmatrix}$$

Successivamente si trasla il sistema rispetto alla traiettoria desiderata:

$$\begin{pmatrix} y \\ \theta \\ \psi \\ \dot{\psi} \\ \dot{\varphi} \end{pmatrix} = \begin{pmatrix} y_I \\ \theta_I \\ \psi_I \\ \dot{\psi}_I \\ \dot{\varphi}_I - v/r_P \end{pmatrix}$$

Si riscrivono le equazioni della dinamica e della cinematica rispetto alle nuove coordinate e si linearizza il modello valutandolo all'equilibrio:

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx + Du \end{aligned}$$

con x si indica tutto lo stato, y le uscite e u gli ingressi.

```
% Dinamica e q=[phi_ddot; psi_ddot] e tau=[tau_phi; tau_psi];
Mass=[a*rp^2*(cos(psi))^2+b*(rp/L)^2*(sin(psi))^2+Ipy -Ipz*rp/L*sin(psi) ; -Ipz*rp/L*sin(psi) Ipz];
N=[(b*(rp/L)^2-a*rp^2)*psi_dot*(phi_dot+v/rp)*sin(psi)*cos(psi) ; -Ipz*rp/L*(phi_dot+v/rp)*psi_dot*cos(psi)];
q=(Mass)\(tau-N);

% Cinematica e c=[x_dot y_dot theta_dot]
c=[rp*(phi_dot+v/rp)*cos(psi)*sin(theta); -rp*(phi_dot+v/rp)*sin(psi)/L];

% Funzione di transizione di stato
f=[rp*(phi_dot+v/rp)*cos(psi)*sin(theta); -rp*(phi_dot+v/rp)*sin(psi)/L; psi_dot; q(2); q(1)];

% Funzioni di trasferimento attuatori
G_tau_psi=Km1*exp(-T1*s)/(Tm1*s+1); % controllo sterzo
G_tau_phi=Km2/(Tm2*s+1)/(Tm2/100*s+1); % controllo velocità angolare ruota posteriore

% Modello nominale linearizzato dinamica e cinematica AGV
syms psi_dot psi tau_psi tau_phi phi_dot y theta;

% Matrice A
A=jacobian(f,stato);
% Valutazione all'equilibrio
A=double(subs(A,[y theta psi psi_dot phi_dot tau_phi tau_psi],[0 0 0 0 0 0]));

% Matrice B
B=jacobian(f,tau);

% Valutazione all'equilibrio
B=double(subs(B,[y theta psi psi_dot phi_dot tau_phi tau_psi],[0 0 0 0 0 0]));
```

```

% Matrice C
C=[1 0 0 0 0 ; 0 1 0 0 0; 0 0 1 0 0; 0 0 0 0 1];

% Matrice D
D=zeros(4,2);

% Funzione di trasferimento AGV
G=C/(s*eye(5)-A)*B;
G=minreal(zpk(G));

% Modello lineare e nominale attuatori
% Funzioni di trasferimento attuatori
G_tau_phi=Km2/(Tm2*s+1)/(Tm2/100*s+1);
G_tau_pade=Km1*(1-T1*0.5*s)/(1+T1*0.5*s)/(Tm1*s+1);
% e^(-T1*s) ~ (1-T1*0.5*s)/(1+T1*0.5*s) approssimazione di Padè

% Attuatori in forma di spazio di stato
[Aphi, Bphi, Cphi, Dphi]=ssdata(G_tau_phi);
[Apsi, Bpsi, Cpsi, Dpsi]=ssdata(G_tau_pade);

```

Per completare il modello lineare si va a connettere in serie degli attuatori ottenendo un' unica forma di stato. Il sistema finale adesso ha uno stato aumentato che comprende le 5 coordinate precedenti e le 4 relative alla dinamica degli attuatori:

$$(y \ \theta \ \psi \ \dot{\psi} \ \dot{\phi} \ x_{1\phi} \ x_{2\phi} \ x_{1\psi} \ x_{2\psi})^T$$

```

% Modello lineare interconnessione in serie AGV-attuatori
% stato_int=[x; y; theta; psi; psi_dot; phi_dot; x1_phi; x2_phi; x1_psi; x2_psi]; stato interconnessione
Atot=[A B(:,1)*Cphi B(:,2)*Cpsi; zeros(2,5) Aphi zeros(2,2); zeros(2,7) Apsi];
Btot=[zeros(5,2); Bphi zeros(2,1); zeros(2,1) Bpsi];
Ctot=[C zeros(4,4)];
Dtot=D;
sys=ss(Atot,Btot,Ctot,Dtot); % forma di stato sistema interconnessione

% Funzione di trasferimento interconnessione in serie AGV-attuatori
Gtot=zpk(tf(sys));

```

Condizioni iniziali

Si considera come condizioni iniziali per le simulazioni i seguenti valori:

Condizione iniziale	Valore	Unità di misura
$y(0)$	2	[m]
$\theta(0)$	0	[rad]
$\psi(0)$	0	[rad]
$\dot{\psi}(0)$	0	[rad/s]
$\dot{\phi}(0)$	0	[rad/s]
$x_{1\phi}$	0	
$x_{2\phi}$	0	
$x_{1\psi}$	0	
$x_{2\psi}$	0	

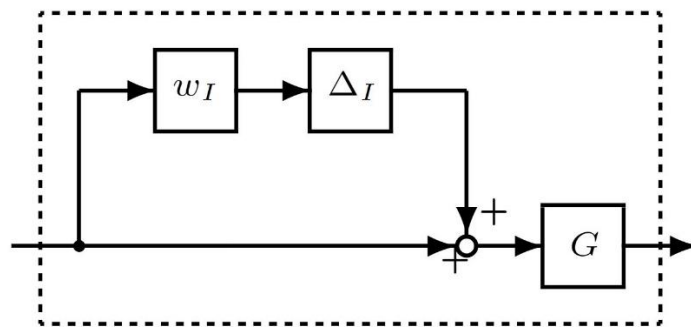
4 - Modellazione incertezze

LQG

Per il controllore LQG la modellazione dei disturbi di processo e i rumori di misura viene fatta attraverso dei segnali stocastici, per la precisione attraverso dei rumori bianchi incorrelati a media nulla e matrice di covarianza costante.

Mix-Sensitivity e D-K iteration

Le incertezze del modello derivano solamente dai parametri e dall' approssimazione del ritardo con Padè, ed è stato scelto di raggrupparle tutte quante in un' unica incertezza moltiplicativa in ingresso non strutturata.



Questa scelta è dovuta al fatto che se si fosse utilizzata la forma $M - \Delta$ di Doyle si sarebbe ottenuto un sistema con dimensione elevata (matrice $\Delta \in \mathbb{C}^{33 \times 33}$).

Per il calcolo del peso w_i si utilizza la seguente procedura:

$$G_p = G(I + w_I \Delta_I), \quad \|\Delta_I\|_\infty \leq 1$$
$$l_I(\omega) = \max_{G_p \in \Pi} \bar{\sigma} \left(G^{-1}(G_p - G)(j\omega) \right); \quad |w_I(j\omega)| \geq l_I(\omega) \quad \forall \omega$$

dove G_p è il set di impianti incerti e G l'impianto nominale, che nel nostro caso è comprensivo degli attuatori.

Implementazione Matlab

```
% Calcolo della funzione peso Wi per le incertezze (finire)
% Si ipotizza una modello di incertezza moltiplicativa
% Wi > sigma(Gnom^(-1)*(Gp-Gnom)) per ogni frequenza

nomega=50; % numero di campioni di frequenze
nsample=70; % numero di campioni di impianti incerti Gp

Gi=Ci*inv(eye(6)*s-Ai)*Bi; % funzione di trasferimento impianti incerti
Gtoti=Gi*blkdiag(G_tau_phi,G_tau_pade); % generico impianto interconnessione AGV-attuatori incerto Gp
Gtotisample=usample(Gtoti,nsample); % 70 campioni di impianti incerti Gp
omega=logspace(0,6,nomega); % 50 campioni di frequenze fra 10^0 rad/s e 10^6 rad/s
```

```

% Calcolo di Wi

% Differenza fra i campioni della Gp con l' impianto nominale Gtot
for i=1:1:nsample
    Gtotivett(:,i)=Gtotisample(:,i)-Gtoti.NominalValue;
    Gvettfrd(:,i)=frd(Gtotivett(:,i),omega); % valutazione in frequenza
end

% Valutazione in frequenza dell' impianto nominale Gtot
Gnomfrd=frd(Gtoti.NominalValue,omega);

% Pseudoinversa di Gnomfrd
for i=1:1:nomega
    Gnompinv(:,i)=pinv(Gnomfrd.ResponseData(:,i));
end

% Calcolo del valore singolare massimo per ogni campione ad ogni frequenza di pinv(Gtot)*(Gtoti-Gtot)
for i=1:1:nsample
    for k=1:1:nomega
        Greterr(:,k,i)=Gnompinv(:,k)*Gvettfrd.ResponseData(:,k,i);
        [U,S,V]=svd(Greterr(:,k,i));
        SigmaMat(k,i)=S(1,1);
    end
end

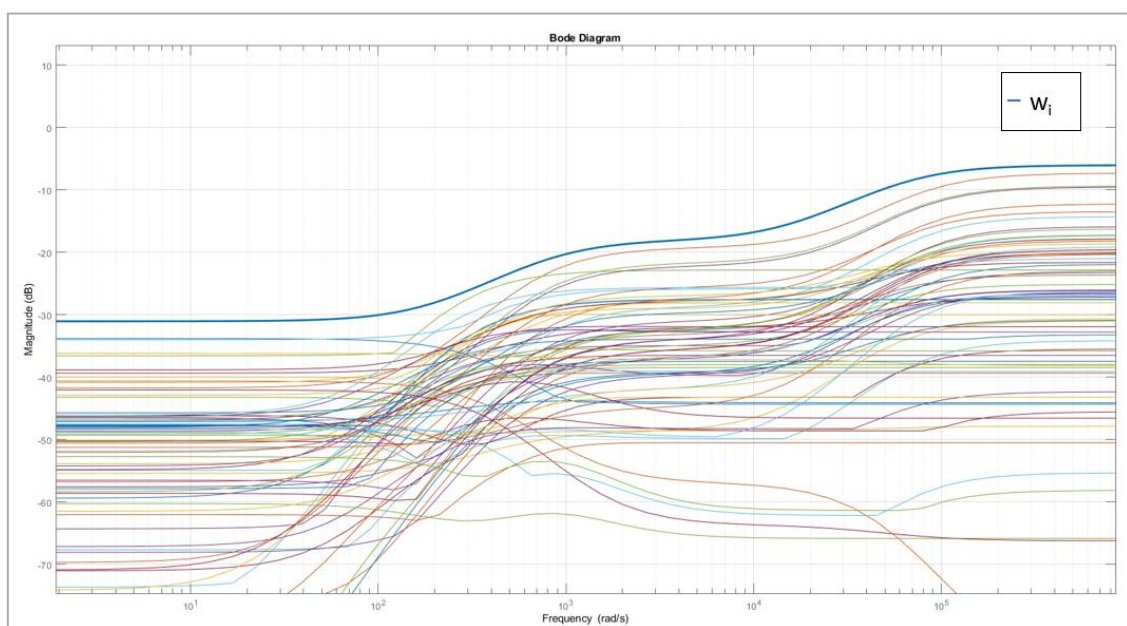
% Plot dei valori singolari massimi
hold on
for i=1:1:nsample
    loglog(omega,20*log(SigmaMat(:,i)));
end

% Peso sull' incertezza Wi
Wi=(0.5*s^2 + 7983*s + 15e05)/(s^2 + 6.424e04*s + 5.345e07);
bodemag(Wi) % plot di Wi
% [freq,resp]=ginput(20); acquisizione grafica di 20 punti di Wi
% sys=frd(10.^(resp/20),freq);
% Wi=fitmagfrd(sys,2); fitting della funzione di trasferimento Wi

Wi=(0.5*s^2 + 7983*s + 15e05)/(s^2 + 6.424e04*s + 5.345e07)*eye(2);

grid on
clear Gtotivett
clear Gvettfrd
clear Gnomfrd

```



Le linee colorate sono il massimo valore singolare dell' errore relativo tra G e G_p .

5 – LQG

Introduzione teorica

Il metodo Linear Quadratic Gaussian è una tecnica per progettare un controllore dinamico per sistemi MIMO (Multiple Input Multiple Output).

Le ipotesi su cui si basa sono le seguenti:

- modello lineare e noto del sistema
- funzionale di costo di forma quadratica
- rappresentazione dei segnali di disturbo e i rumori sull' ingresso e sullo stato come rumori bianchi a media nulla e matrice di covarianza costante

Dato il modello lineare

$$\begin{aligned}\dot{x} &= Ax + Bu + w_d \\ y &= Cx + Du + w_n\end{aligned}$$

dove w_d è il disturbo in ingresso con media nulla e covarianza W e w_n è il disturbo dato dai sensori con media nulla e covarianza V , la tecnica va a sintetizzare il controllo $u(t)$ minimizzando il funzionale di costo

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

dove Q e R sono opportune matrici peso simmetriche per lo stato e l'ingresso, con Q positiva semidefinita e R positiva definita.

Il metodo può essere diviso in due fasi:

- sintesi del controllore $u(t)$ con LQR
- implementazione del filtro di Kalman

Linear Quadratic Regulator (LQR)

In questa sezione si determina un controllore ottimo basandosi su un modello lineare non soggetto a disturbi e minimizzando

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

Il controllore ottenuto si basa sulla retroazione dello stato $u(t) = -K_r x$, in cui x è lo stato, $K_r = R^{-1} B^T X$ e X matrice simmetrica semidefinita positiva è soluzione dell' equazione algebrica di Riccati

$$A^T X + X A - X B R^{-1} B^T X + Q = 0$$

La soluzione al problema è ottenibile se e solo se il sistema è stabilizzabile e rilevabile.

Filtro di Kalman

Non sempre è possibile osservare lo stato, quindi si utilizza un filtro di Kalman per stimare le componenti per il controllo ottenendo $u(t) = -K_r \hat{x}$.

Scrivendo il filtro nella forma di Osservatore di Luenberger si ottiene il guadagno $K_f = YC^T V^{-1}$, con Y matrice simmetrica semidefinita positiva è soluzione dell'equazione algebrica di Riccati

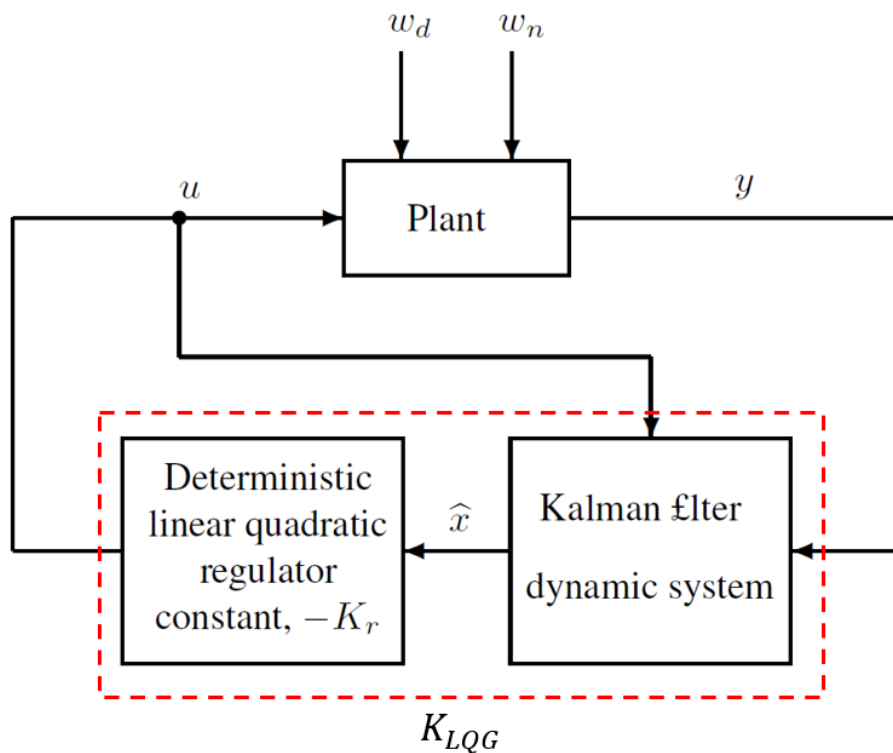
$$YA^T + AY - YC^T V^{-1}CY + W = 0$$

Controllore K_{LQG}

Il controllore K_{LQG} è un sistema dinamico rappresentato da quattro matrici

$$\begin{aligned} A_{LQG} &= A - BK_r - K_f C \\ B_{LQG} &= K_f \\ C_{LQG} &= -K_r \\ D_{LQG} &= 0 \end{aligned}$$

$$K_{LQG}(s) = \left[\begin{array}{c|c} A - BK_r - K_f C & K_f \\ \hline -K_r & 0 \end{array} \right]$$



Rappresentazione dell'impianto soggetto a controllo LQG

Il controllo LQG, a differenza di LQR e filtro di Kalman, non garantisce buone proprietà di robustezza, poiché non è possibile rappresentare correttamente tutte le incertezze come rumori bianchi e il metodo di sintesi può essere interpretato in termini di norma \mathcal{H}_2 .

Implementazione Matlab-Simulink

Si riporta il codice Matlab per la sintesi del controllore K_{LQG} e i grafici delle simulazioni in ambiente Simulink.

```
% Calcolo del controllore LQG

% Caricamento dei parametri necessari
AGV_Parametri;

% Test Stabilizzabilità e rilevabilità
poli_t=eig(Atot); % poli sistema interconnessione AGV-Attuatori
rank_zero_st_t=rank([Atot-0*eye(size(Atot,1)) Btot]); % lemma PBH per la stabilizzabilità dei poli nell'origine
rank_v_st_t=rank([Atot-v*eye(size(Atot,1)) Btot]); % lemma PBH per la stabilizzabilità del polo in +v
rank_zero_det_t=rank([Atot-0*eye(size(Atot,1)) Ctot]); % lemma PBH per la rilevabilità dei poli nell'origine
rank_v_det_t=rank([Atot-v*eye(size(Atot,1)) Ctot]); % lemma PBH per la rilevabilità del polo in +v
```

Le condizioni di stabilizzabilità e rilevabilità sono verificate sia per i quattro poli nell'origine sia per il polo in + 2, che è pari alla velocità di avanzamento del muletto.

```
% Passo LQR

% Matrici peso LQR
Q=blkdiag(15,20,1,1,1,0,0,0,0); % peso sullo stato interconnessione AGV-attuatori
Q1=blkdiag(15,20,1,1,1,0.001,0.001,0.001,0.001,0.001,0.001,0.001);
R=[1 0;0 20]; % peso sull' ingresso di controllo u = [u_phi u_psi]

% Controllo LQR
Kr=lqr(Atot,Btot,Q,R); % calcolo della costante Kr del controllo in feedback u=-kr*stato_int
```

```
% Passo Filtro di Kalman

% Matrici di covarianza disturbi e errori di misura
B_n=blkdiag(zeros(5),1,0,1,0); % matrice disturbo di processo x_dot=A*x+B*u+B_n*wd
W=0.0001*blkdiag(zeros(5),1,0,1,0); % matrice covarianza disturbi di processo
v_y=0.01; % [m] deviazione standard sensore di y
v_theta=0.02; % [rad] deviazione standard sensore di theta
v_psi=0.02; % [rad] deviazione standard sensore di psi
v_phi_dot=0.01; % [rad/s] deviazione standard sensore di phi_dot
V=blkdiag(v_y^2,v_theta^2,v_psi^2,v_phi_dot^2); % matrice covarianza errori di misura

% Filtro di Kalman
Dkal=zeros(4,11); % Matrice D della forma di stato del filtro di Kalman
Estss = ss(Atot,[Btot B_n],Ctot,Dkal); % Forma di stato del filtro di Kalman
[Kess, Ke] = kalman(Estss,W,V); % Ke è il guadagno del filtro di Kalman
```

```
% Controllo KLQG = [ Alqg Blqg ; Clqg Dlqg ]
```

```
Alqg=Atot-Btot*Kr-Ke*Ctot;
```

```
Blqg=Ke;
```

```
Clqg=-Kr;
```

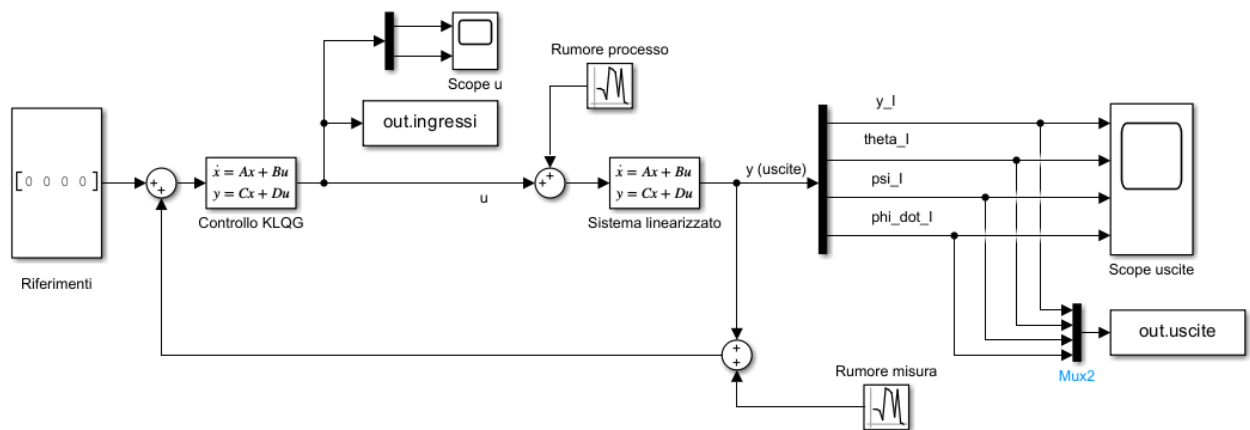
```
Dlqg=zeros(2,4);
```

```
Klqg=[Alqg Blqg;Clqg Dlqg];
```

```
% Forma di stato KLQG
```

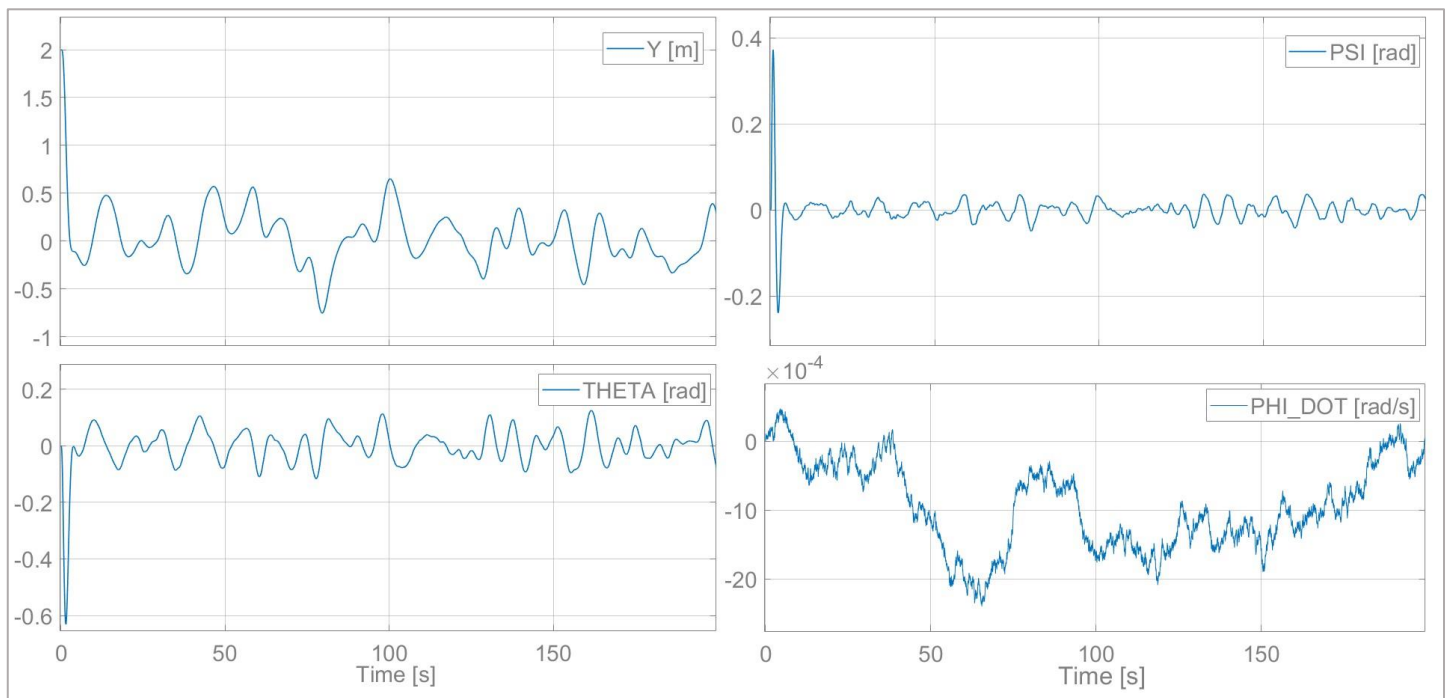
```
KLQG=ss(Alqg,Blqg,Clqg,Dlqg);
```

```
KLQGs=tf(KLQG);
```



Schema simulink del modello linearizzato con controllore K_{LQG} .

Risultati

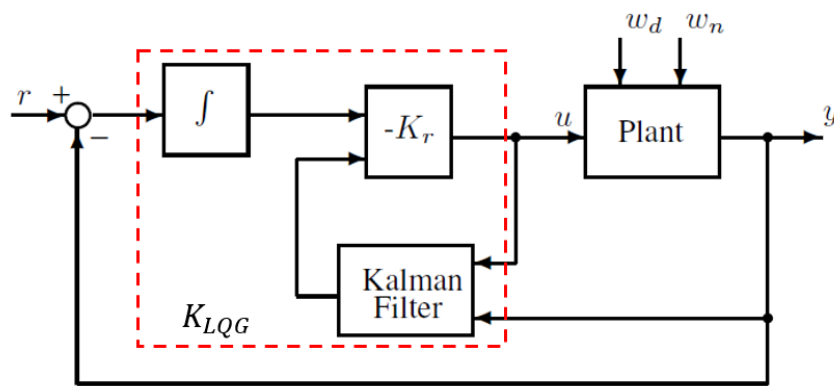


I grafici presentati mostrano l'andamento delle componenti osservate partendo dalla condizione iniziale $[2 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0]^T$ e con i valori nominali dei parametri.
Il controllore K_{LQG} permette di rispettare le specifiche richieste in meno di 20 secondi.

LQG con integratore

Il controllo LQG standard non fornisce un'azione integrale necessaria per annullare gli errori a regime o per reiettare disturbi.

È possibile sopperire a questa mancanza aumentando l'impianto con l'aggiunta di un integratore e progettare un controllore su questo nuovo sistema. Dopodiché l'integratore viene spostato nel blocco del controllo per ottenere l'impianto iniziale.



Nella relazione non verrà riportato un controllore con integratore perché il sistema perde osservabilità aggiungendo quattro poli in zero (tanti quanti le uscite).
Si presenta comunque due metodi per l'implementazione Matlab.

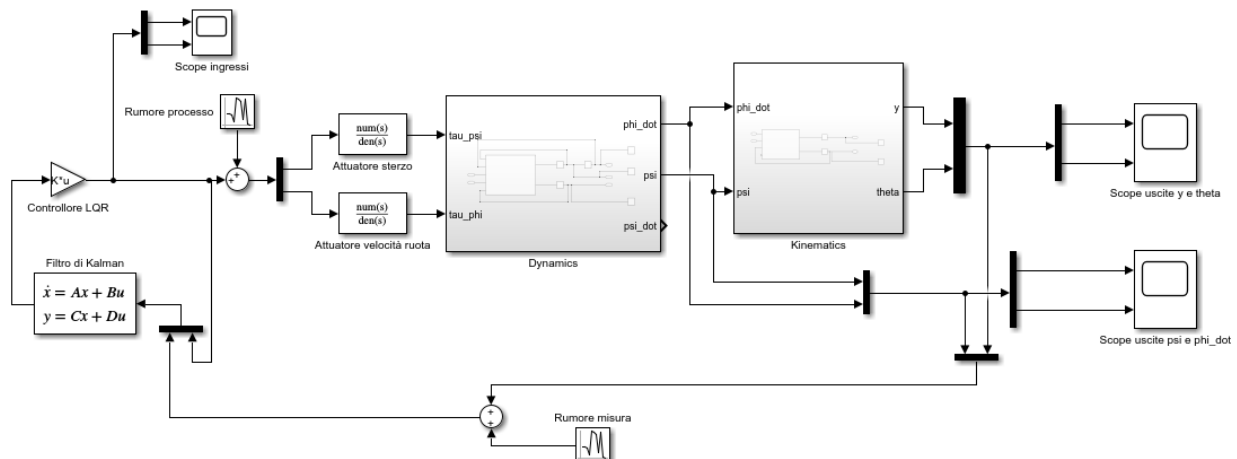
```
% Controllo LQG con integratore

% Metodo 1
[K,S,CLP]=lqi(sys,Q,R);

% Metodo 2
n=9; % dimensione stato sistema iniziale
m=4; % numero uscite sistema iniziale
Znm=zeros(n,m); Zmm=zeros(m,m);
Znn=zeros(n,n); Zmn=zeros(m,n);

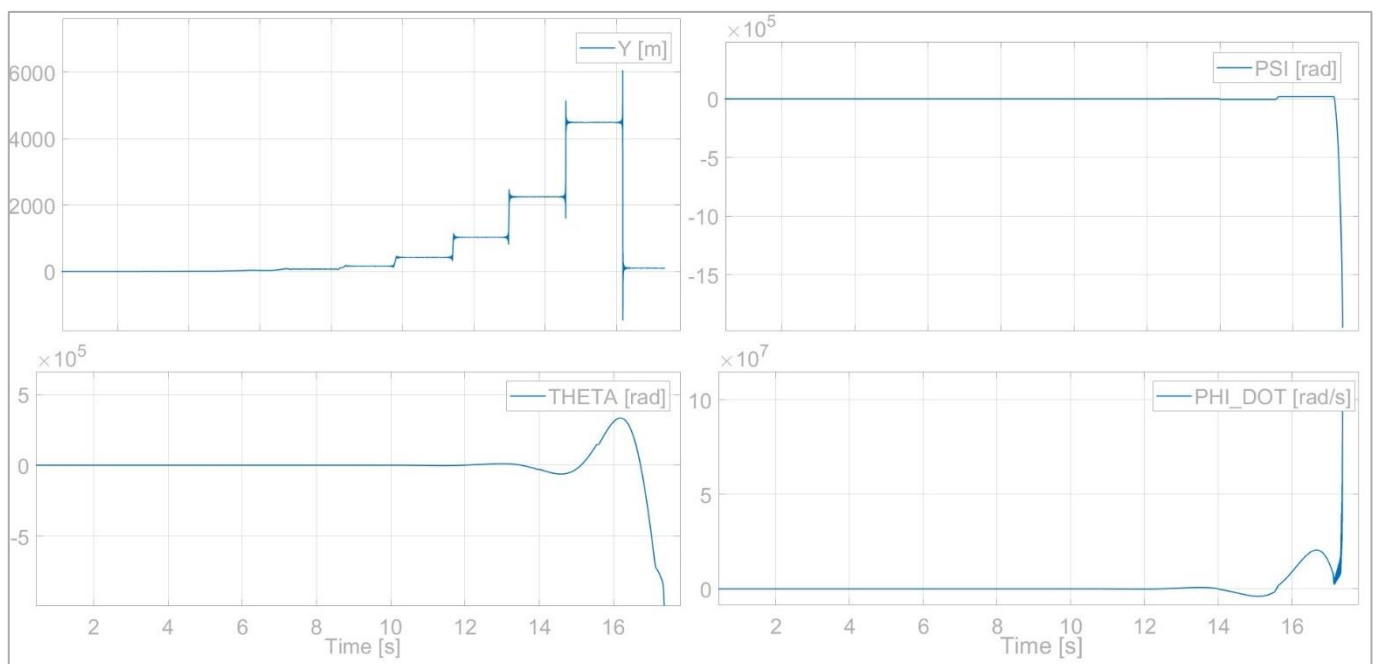
A=[Atot Znm;-Ctot Znm]; % matrice A per sistema aumentato
B=[Btot;-Dtot]; % matrice B per sistema aumentato
Q=[Znn Znm;Zmn eye(m,m)]; % matrice pesi stato aumentato
R=eye(2); % matrice pesi ingressi
Kr=lqr(A,B,Q,R); % controllore LQR
Krp=Kr(1:m,1:n); % controllore LQR su sistema iniziale
Kri=Kr(1:m,n+1:n+m); % controllore LQR su integratori
Bnoise= 0.001*eye(n); % disturbi di processo
Estss = ss(Atot,[Btot B_n],Ctot,Dkal);
[Kess, Ke] = kalman(Estss,W,V); % filtro di Kalman sistema aumentato
Ac=[ZmmZmn;-b*Kri Atot-Btot*Krp-Ke*Ctot]; % matrice A controllore LQG con integratore
Bcr= [eye(m);Znm]; Bcy = [-eye(m);Ke];
Bc= [Bcr Bcy]; % matrice B controllore LQG con integratore
Cc = [-Kri -Krp]; % matrice C controllore LQG con integratore
Dcr = Zmm; Dcy= Zmm;
Dc=[Dcr Dcy]; % matrice D controllore LQG con integratore
Klqg= ss(Ac,Bc,Cc,Dc); % controllore LQG con integratore
```

LQG con modello non lineare



Sistema con funzione di transizione di stato non lineare e controllore LQG

Si valuta l' utilizzo del controllore trovato precedentemente sul modello non lineare del sistema partendo dalle solite condizioni iniziali e con valori nominali dei parametri.



Il sistema in anello chiuso dopo un breve lasso di tempo (10 secondi) diverge, anche partendo da condizioni iniziali più vicine alla traiettoria di equilibrio.

È necessario implementare un controllore migliore basato su tecniche non lineari e utilizzare un filtro di Kalman esteso per garantire le specifiche richieste.

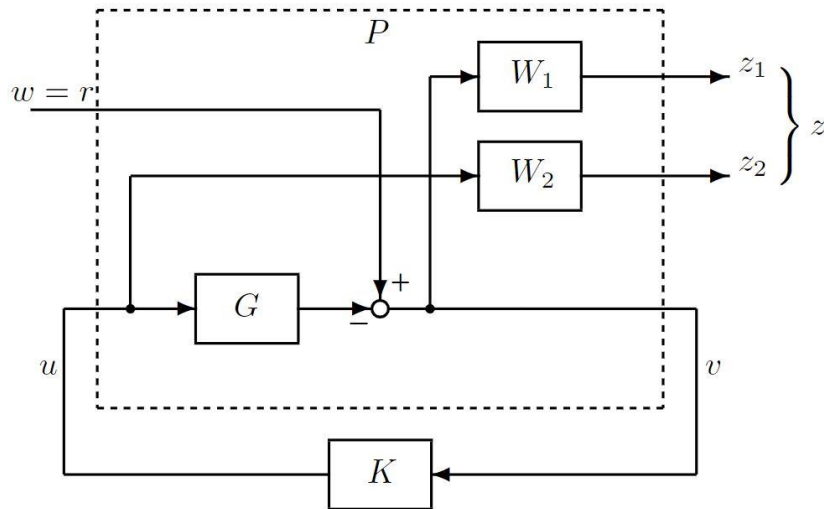
6 – Mixed-Sensitivity \mathcal{H}_∞

Introduzione teorica

Questa tecnica di sintesi permette la realizzazione di un controllore K tale da minimizzare la norma H_∞ della matrice N riportata in figura, in generale andando a considerare non solo la funzione di sensitività S , e quindi la reiezione di disturbi o l'inseguimento del riferimento, ma anche le funzioni KS , limitando così i segnali di controllo in ingresso al sistema, e di sensitività complementare T per il tracking. Gli elementi di N sono inoltre moltiplicati per una serie di funzioni peso (W_p, W_u, W_T), scelte in modo tale che il sistema controllato in ciclo chiuso possieda determinate caratteristiche.

$$K \text{ t.c. } \left\| N = \begin{bmatrix} w_p S \\ w_u KS \\ w_T T \end{bmatrix} \right\|_\infty \leq \gamma$$

Lo stesso problema di sintesi può essere riformulato, usando la forma di Doyle:



$$z = \begin{bmatrix} W1(r - y) \\ W2u \end{bmatrix} \quad P = \begin{bmatrix} W1 & -W2G \\ 0 & W2 \\ I & -G \end{bmatrix}$$

$$\|F_l(P, K)\|_\infty = \left\| \begin{bmatrix} W1S \\ W2KS \end{bmatrix} \right\|_\infty$$

Nel caso specifico del controllo dell'AGV, sono stati considerati un problema di tracking, scegliendo come ingresso esogeno r , e sfruttando il fatto che S , oltre ad essere la funzione di trasferimento tra d e y , è anche la funzione di trasferimento tra r ed $e = (r - y)$, ed un problema di limitazione del segnale di controllo u .

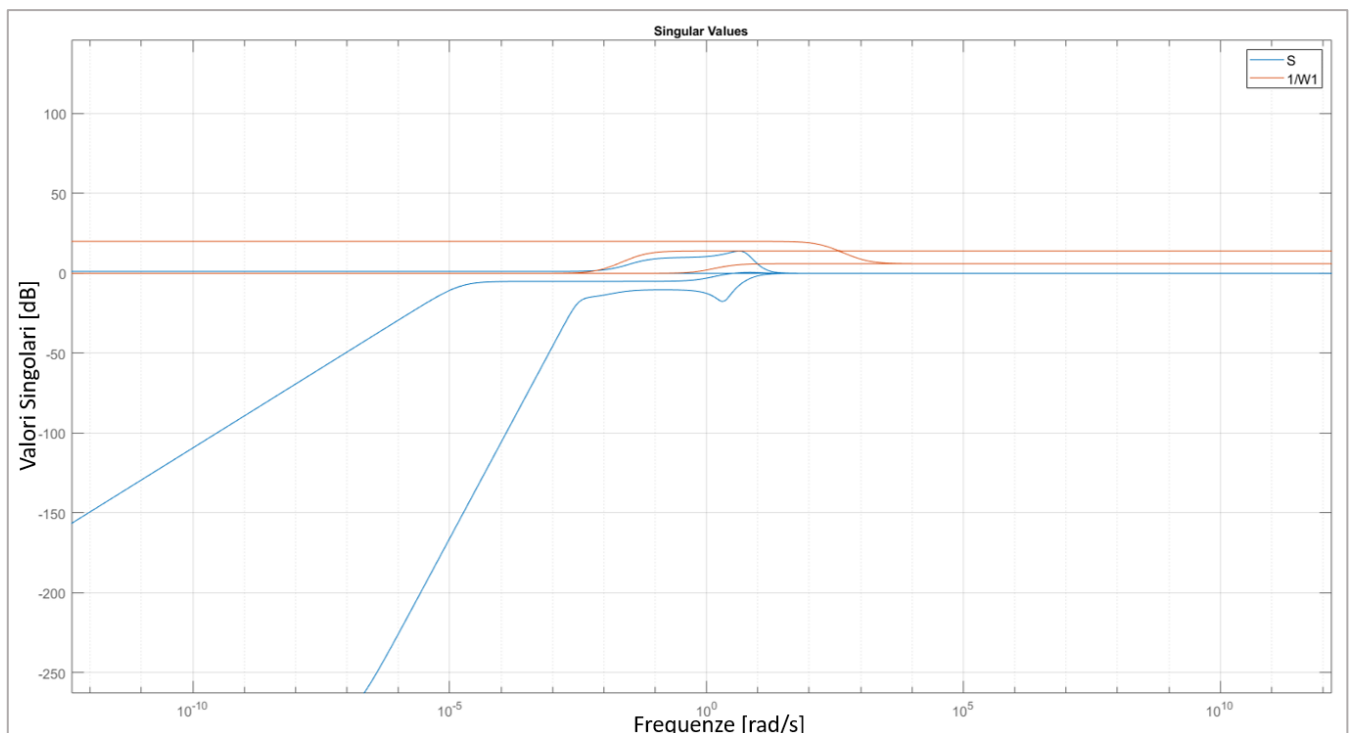
Peso sulle prestazioni

Come funzione peso per S è stata scelta la matrice $W1$ diagonale nella forma:

$$W1 = \begin{bmatrix} w1_y & 0 & 0 & 0 \\ 0 & w1_\theta & 0 & 0 \\ 0 & 0 & w1_\psi & 0 \\ 0 & 0 & 0 & w1_\phi \end{bmatrix}, \text{ con } w1_i = \frac{\left(\frac{s}{M^{\frac{1}{n}}} + \omega_b\right)^n}{\left(s + A^{\frac{1}{n}} * \omega_b\right)^n}$$

La scelta dei parametri è basata su tecniche note in letteratura: nello specifico, è stato inizialmente realizzato un controllore H_∞ sub-ottimo, e si è poi sfruttato il modulo degli elementi sulla diagonale della funzione di sensitività S per assegnare ai parametri i valori riportati in tabella, sapendo che M è legato al guadagno in alta frequenza di S , ω_b alla banda passante, e A all'errore a regime. In questo modo il sistema con il nuovo controllore ottimo soddisfa la proprietà di prestazione nominale, come mostrato in figura.

	M	ω_b	A	n
Y	5	0.01	1	1
θ	2	100	10	1
ψ	2	100	10	1
ϕ	2	1	1	1



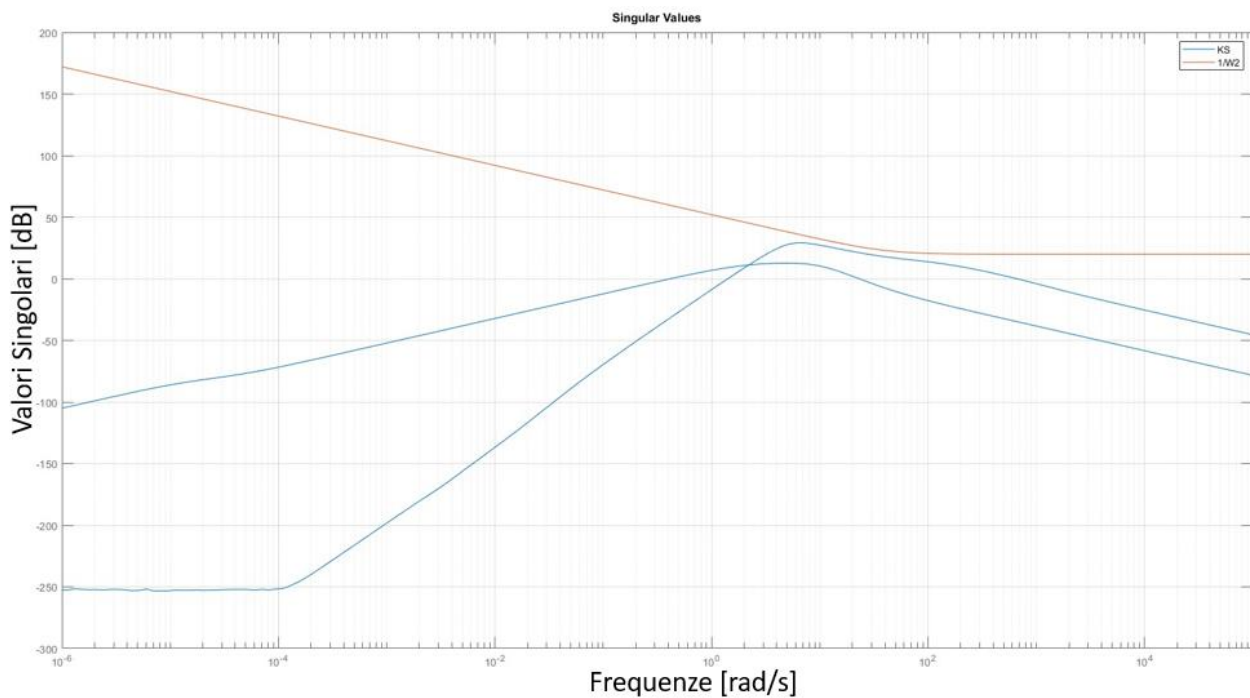
Peso sugli ingressi

Per l'ingresso u è stata inizialmente scelta una matrice di costanti, con elementi sulla diagonale pari a 0.08, in modo tale da assicurare almeno la prestazione nominale; tuttavia, è stato osservato che alzando la costante di guadagno a 0.1, e aggiungendo su ogni componente un filtro passa-alto con ω_u pari 40 rad/s, si ha un miglioramento delle prestazioni del sistema, specialmente per quanto riguarda il controllore realizzato con il metodo "DK-iteration", che verrà trattato in seguito.

La funzione peso finale avrà quindi la forma:

$$W2 = 0.1 \begin{bmatrix} w_{2u1} & 0 \\ 0 & w_{1u2} \end{bmatrix}, \text{ con } w_{2i} = \frac{s}{s + \omega_u}$$

Una volta definite le funzioni peso, e portato il sistema in forma di Doyle, il controllore viene sintetizzato utilizzando il comando `'hinfsyn'`.



Implementazione Matlab-Simulink

Di seguito sono riportati i codici Matlab ed il modello Simulink utilizzato per le simulazioni.

```
% Calcolo del controllore Hinf

% Caricamento dei parametri necessari
AGV_Parametri;

%%Modello lineare interconnessione in serie AGV-attuatori
sys=ss(A,B,C,D); % modello di stato AGV
sys=sys*blkdiag(G_tau_phi,G_tau_pade); % interconnessione
```

```
% Calcolo dei pesi Wp per le performance su y, theta, psi, phi_dot
% Struttura del peso Wp=((s/M^(1/n)+w)^n)/(s+w*A^(1/n)^n

% Peso coordinata y
My=5;
wy=1e-2;
Ay=1;
ny=1;
W1y=((s/My^(1/ny)+wy)^ny)/(s+wy*Ay^(1/ny)^ny;

%Peso coordinata theta
Mtheta=2;
wtheta=100;
Atheta=10;
ntheta=1;
W1theta=((s/Mtheta^(1/ntheta)+wtheta)^ntheta)/(s+wtheta*Atheta^(1/ntheta)^ntheta;

%Peso coordinata psi
Mpsi=2;
wpsi=100;
Apsi=10;
npsi=1;
W1psi=((s/Mpsi^(1/npsi)+wpsi)^npsi)/(s+wpsi*Apsi^(1/npsi)^npsi;

%Peso coordinata phi_dot
Mphi=2;
wphi=1;
Aphi=1;
nphi=1;
W1phi=((s/Mphi^(1/nphi)+wphi)^nphi)/(s+wphi*Aphi^(1/nphi)^nphi;

W1=blkdiag(W1y,W1theta,W1psi,W1phi);
```

```
% Calcolo dei pesi Wu per le performance su u_phi, u_psi
% Struttura del peso Wu=s/(s+wb)

% Peso su u_phi
W_phi=tf([1 0],[1 40]);

% Peso su u_psi
W_psi= tf([1 0],[1 40]);

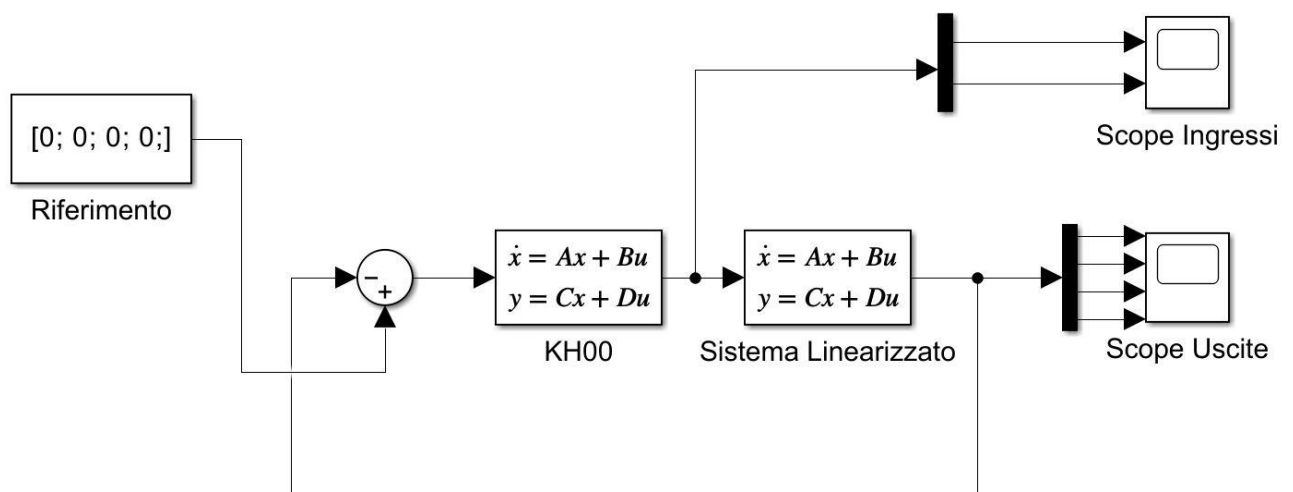
%Peso sul segnale di controllo
W2=0.1*[W_phi 0; 0 W_psi]; % riduzioni effetto peso di un fattore 0.1
```

```
% Sistema in forma di Doyle (Delta-P-K)
```

```
systemnames = 'sys W1 W2'; % blocchi nel sistema
inputvar = '[r(4) ; u(2)]'; % ingressi al blocco P
outputvar = '[W1; W2; -sys+r]'; % uscite dal blocco P
input_to_sys = '[u]'; % ingressi al sistema sys
input_to_W1 = '[r-sys]'; % ingressi alle performance W1
input_to_W2 = '[u]'; % ingressi alle performance W2
sysoutname = 'P';
cleanup_sysic = 'yes';
sysic;
```

```
% Sintesi controllore Hinf
```

```
nmeas=4; % numero di misure dai sensori y
ncon=2; % numero di ingressi u
[KH00,CL,GAM,INFO]=hinfsyn(P,nmeas,ncon); % Controllore Hinf
KH00=(minreal(KH00));
```

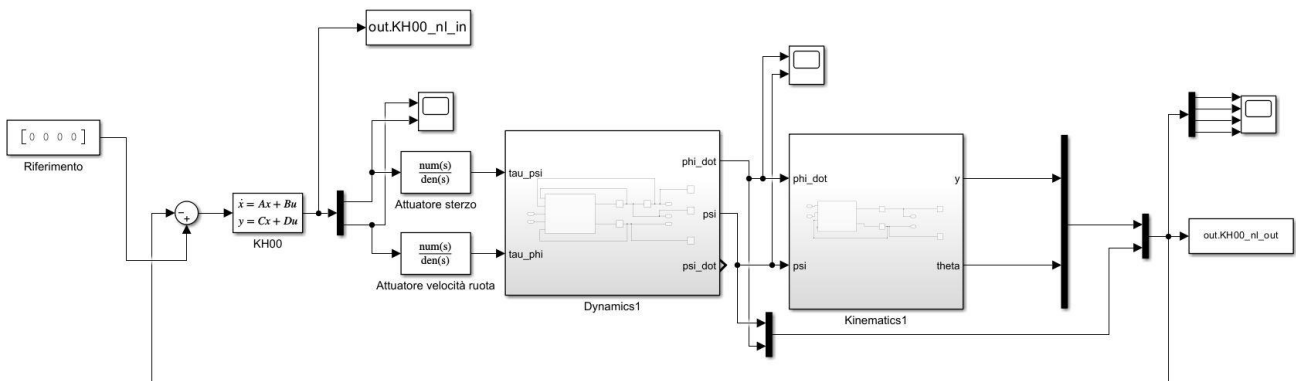


Schema simulink del modello linearizzato con controllore Mixed-Sensitivity \mathcal{H}_∞ .

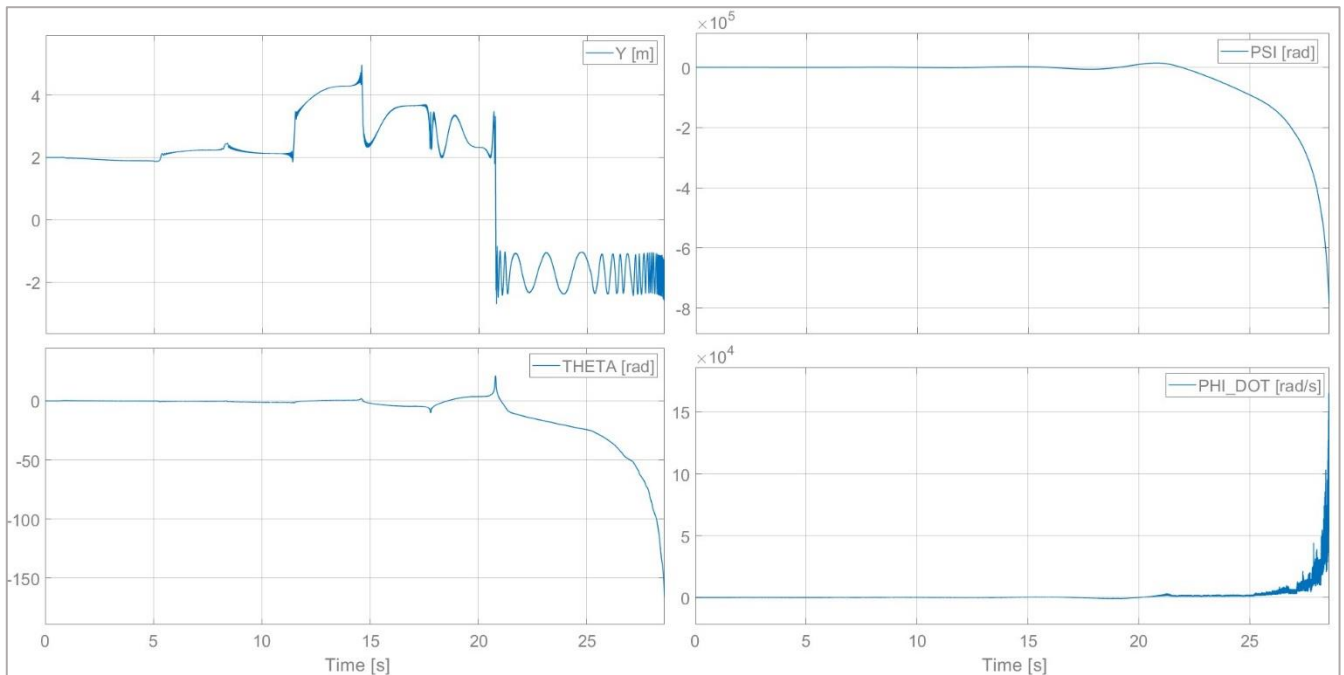
Si è verificato che il controllore ottenuto rende il sistema nominalmente stabile in quanto gli autovalori del sistema a ciclo chiuso sono tutti a parte reale negativa.

I risultati del caso lineare sono analizzati nel capitolo *‘Risultati e confronti’*.

Sistema non lineare con Mixed-Sensitivity \mathcal{H}_∞



Schema simulink del modello non lineare con controllore Mixed-Sensitivity \mathcal{H}_∞ .



Si valuta l' utilizzo del controllore trovato precedentemente sul modello non lineare del sistema partendo dalle solite condizioni iniziali e con valori nominali dei parametri.

Come osservato per il controllore LQG , anche in questo caso il sistema a ciclo chiuso diverge dopo pochi secondi, ma in maniera "ridotta" almeno nei primi 20 secondi.

7 – D-K iteration

Introduzione teorica

La DK-iteration è una tecnica di μ -sintesi che consente di ottenere buoni risultati, e va a generare un controllore combinando la sintesi H_∞ con la μ -analisi.

Prima di tutto si va a definire un upper bound su μ , in termini di valore singolare strutturato

$$\mu(N) \leq \min_{D \in \mathcal{D}} \bar{\sigma}(DND^{-1})$$

L'idea alla base del metodo è quella di cercare un controllore che vada a minimizzare il valore di picco al variare delle frequenze di questo upper bound, ovvero:

$$\min_K \left(\min_{D \in \mathcal{D}} \|DN(K)D^{-1}\|_\infty \right)$$

Questo viene fatto minimizzando il valore $\|DN(K)D^{-1}\|_\infty$ rispetto a K e D in maniera alternata.

Sinteticamente, il metodo prevede tre step:

- K-step: si sintetizza un controllore H_∞ per il problema $\min_K \|DN(K)D^{-1}\|_\infty$ con $D(s)$ fissato
- D-step: si trova $D(j\omega)$ che minimizza $\bar{\sigma}(DND^{-1}(j\omega))$ ad ogni frequenza con N fissato
- Fit-step: si va a fittare il modulo di ogni elemento di $D(j\omega)$ con una funzione di trasferimento $D(s)$ che sia stabile e a fase minima

Il metodo è iterativo, pertanto una volta raggiunto il *Fit-step* si riparte dal *K-step*, fintanto che non si riesce a trovare un controllore che minimizzi l'upper bound, o che comunque soddisfi le prestazioni desiderate.

In generale quello trovato è un controllore che risulta sub-ottimo. Questo perché il *D-step* e il *K-step* sono convessi, ma non è garantito che si raggiunga un minimo globale per il problema.

Ad ogni modo, i risultati che si ottengono con questa tecnica sono in genere buoni, anche se il controllore risultante ha spesso un ordine abbastanza elevato.

Implementazione Matlab-Simulink

L'implementazione inizia con una prima fase di inizializzazione, composta dal caricamento dei parametri dell'AGV e l'interconnessione con gli attuatori.

```
% Calcolo del controllore D-K Iteration

% Caricamento dei parametri necessari
AGV_Parametri;

%%Modello lineare interconnessione in serie AGV-attuatori
sys=ss(A,B,C,D); % modello di stato AGV
sys=sys*blkdiag(G_tau_phi,G_tau_pade); % interconnessione
```

Segue poi la scelta dei pesi per il problema, che vengono scelti come spiegato nel paragrafo relativo alla *Mixed-Sensitivity*.

```
% Calcolo dei pesi Wp per le performance su y, theta, psi, phi_dot
% Struttura del peso Wp=((s/M^(1/n)+w)^n)/(s+w*A^1/n)^n

% Peso coordinata y
My=5;
wy=1e-2;
Ay=1;
ny=1;
W1y=((s/My^(1/ny)+wy)^ny)/(s+wy*Ay^1/ny)^ny;

%Peso coordinata theta
Mtheta=2;
wtheta=100;
Atheta=10;
n=1;
W1theta=((s/Mtheta^(1/n)+wtheta)^n)/(s+wtheta*Atheta^1/n)^n;

%Peso coordinata psi
Mpsi=2;
wpsi=100;
Apsi=10;
npsi=1;
W1psi=((s/Mpsi^(1/n)+wpsi)^n)/(s+wpsi*Apsi^1/n)^n;

%Peso coordinata phi_dot
Mphi=2;
wphi=1; %10 (finire)
Aphi=1;
nphi=1;
W1phi=((s/Mphi^(1/n)+wphi)^n)/(s+wphi*Aphi^1/n)^n;

%Blocco unico per i pesi
W1=blkdiag(W1y,W1theta,W1psi,W1phi);

% Calcolo dei pesi Wu per le performance su u_phi, u_psi
% Struttura del peso Wu=s/(s+wb)

% Peso su u_phi
W_phi=tf([1 0],[1 40]);

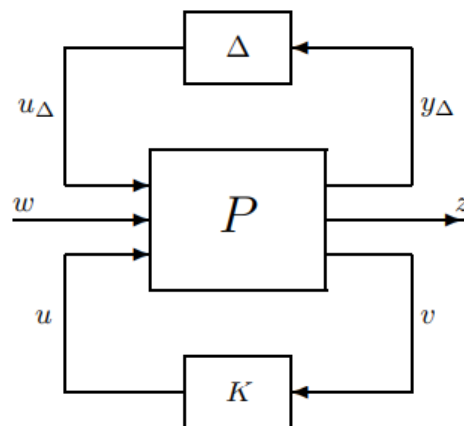
% Peso su u_psi
W_psi= tf([1 0],[1 40]);

%Peso sul segnale di controllo
W2=0.1*[W_phi 0; 0 W_psi]; % riduzioni effetto peso di un fattore 0.1
```


Si definisce poi il peso da dare alle incertezze di modello.

```
% Calcolo peso delle incertezze
Wi=(0.5*s^2 + 7983*s + 15e05)/(s^2 + 6.424e04*s + 5.345e07)*eye(2);
```

Si procede poi all'interconnessione del sistema in forma di Doyle Δ - P - K , ovvero rifacendosi allo schema:



L'implementazione, utilizzando il comando 'sysic', è la seguente

```
% Sistema in forma di Doyle (Delta-P-K)

systemnames = 'sys W1 W2 Wi'; % blocchi nel sistema
inputvar = '[udel(2); r(4) ; u(2)]'; % ingressi al blocco P
outputvar = '[Wi; W1; W2; -sys+r]'; % uscite dal blocco P
input_to_sys = '[u+udel]'; % ingressi al sistema sys
input_to_W1 = '[r-sys]'; % ingressi alle performance W1
input_to_W2 = '[u]'; % ingressi alle performance W2
input_to_Wi = '[u]'; % ingressi alle incertezze Wi
sysoutname = 'P';
cleanupsysic = 'yes';
sysic;
```

Adesso si ha la vera e propria DK-iteration. Dopo aver definito la forma del blocco legato alle incertezze e averlo interconnesso col sistema in maniera coerente alla formulazione Δ - P - K schematizzata sopra, si procede alla sintesi del controllore utilizzando il comando 'musyn'.

```

% Controllo D-K iteration

% Parametri per il calcolo
nmeas = 4; % numero di misure dai sensori y
nu = 2; % numero di ingressi u
Delta=ultidyn('Delta',[2 2]); % forma del blocco legato alle incertezze
Fu=lft(Delta,P); % upper linear fractional trasformation

% Controllore D-K iteration
[K,CLPERF,INFO] = musyn(Fu,nmeas,nu);
K=minreal(K); % Controllore D-K

```

Il comando '*musyn*' restituisce un report che riassume i risultati ottenuti ad ogni iterazione. Come si può vedere le performance migliorano ad ogni passo, anche se l'ordine del controllore è molto elevato.

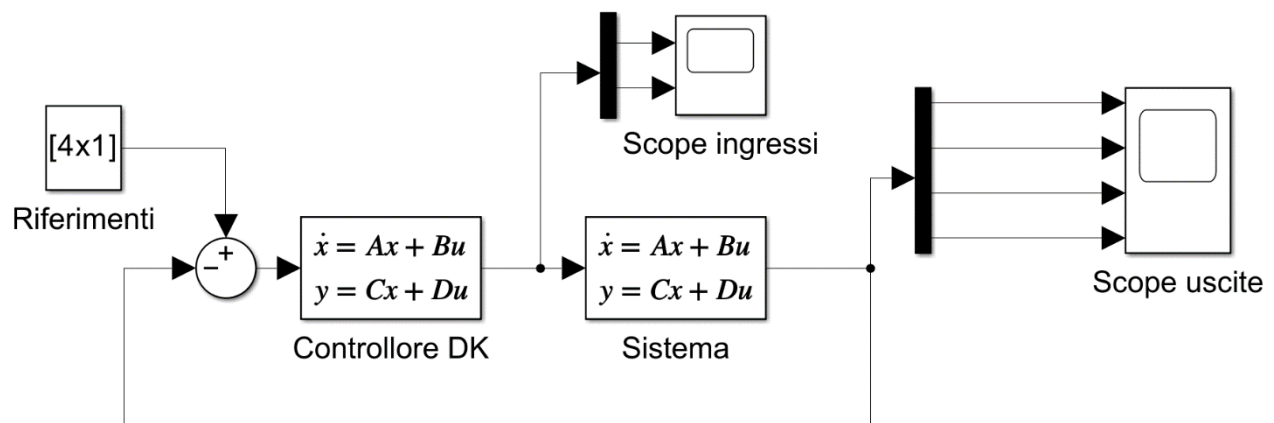
D-K ITERATION SUMMARY:

Robust performance				Fit order
Iter	K Step	Peak MU	D Fit	D
1	0.7777	0.7776	0.7859	16
2	0.671	0.6709	0.6849	16
3	0.6577	0.6576	0.6658	16
4	0.6525	0.6525	0.6646	16
5	0.652	0.6521	0.6627	16

Best achieved robust performance: 0.652

4 states removed.

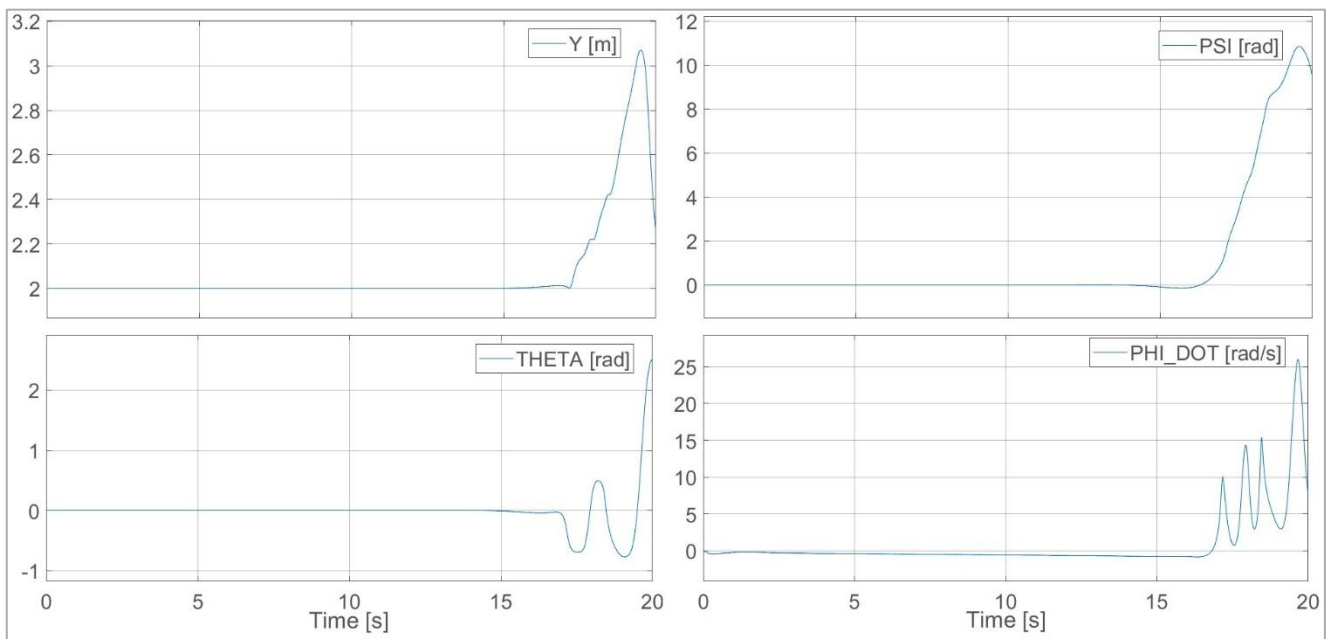
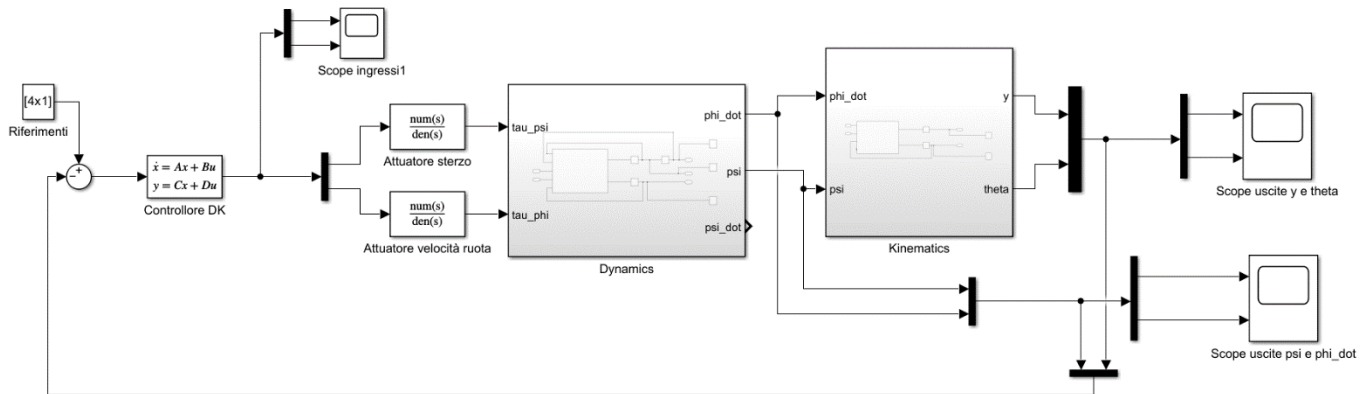
Si riportano infine gli schemi implementati su Simulink per simulare il sistema con il controllore ricavato dalla DK-iteration.



Anche in questo caso il controllore ottenuto rende il sistema nominalmente stabile in quanto gli autovalori del sistema a ciclo chiuso sono tutti a parte reale negativa.

I risultati del caso lineare sono analizzati nel capitolo *‘Risultati e confronti’*.

Sistema non lineare con D-K iteration

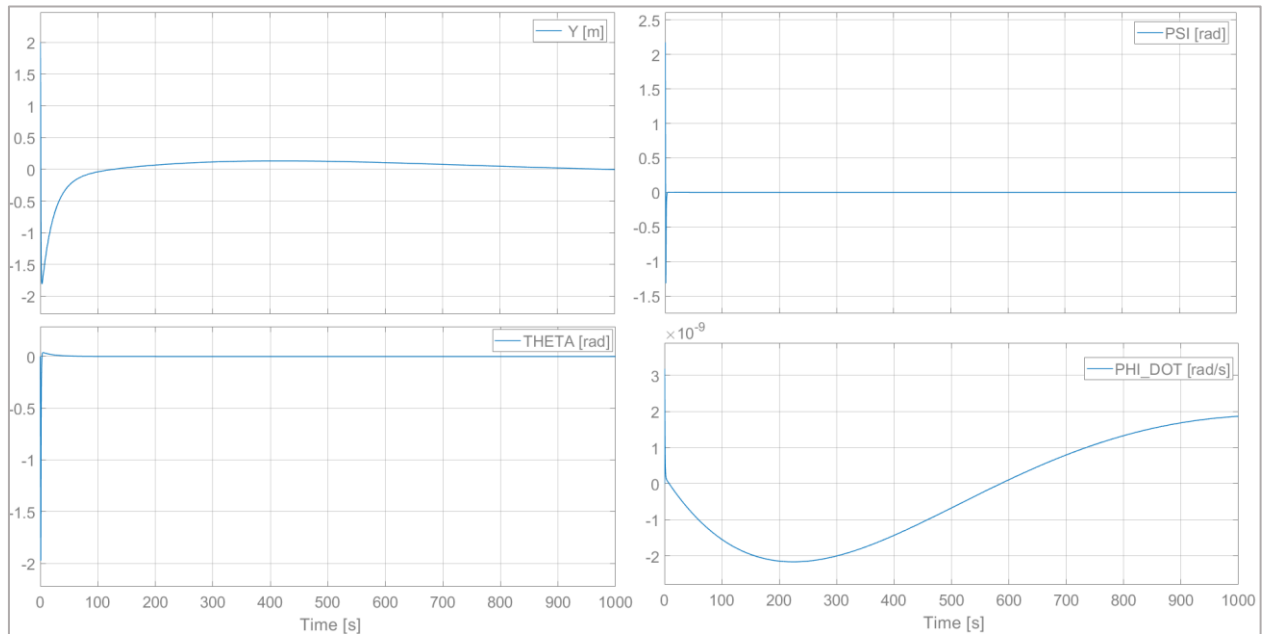


Si valuta l' utilizzo del controllore trovato precedentemente sul modello non lineare del sistema partendo dalle solite condizioni iniziali e con valori nominali dei parametri.

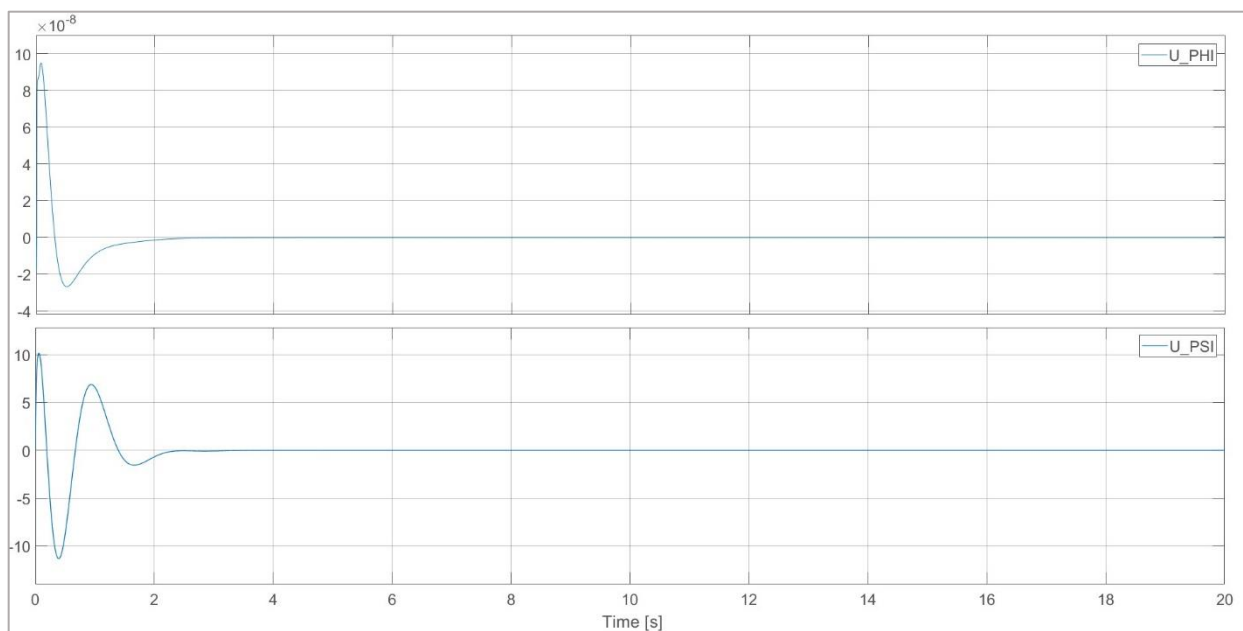
Come visto anche per la sintesi del controllore \mathcal{H}_∞ , nel caso non lineare si ha instabilità dopo pochi secondi. Anche avvicinando le condizioni iniziali allo stato di equilibrio il risultato non varia molto.

8 – Risultati e confronti

Risultati caso lineare \mathcal{H}_∞



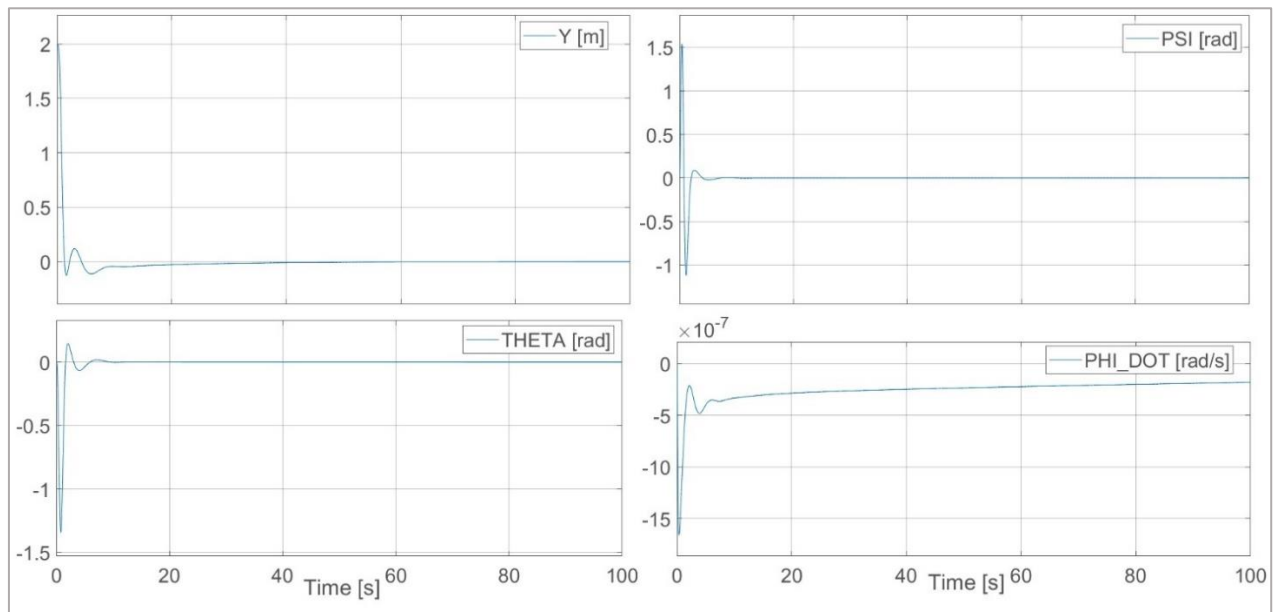
Dai grafici si osserva che tutte le componenti osservate vanno a zero in un tempo breve, eccetto la y che ci arriva in circa 100 secondi. Inoltre, si rileva la presenza di una sovraelongazione e un' inversione della traiettoria. La componente $\dot{\phi}$ oscilla, ma ciò non influisce sulle performance richieste perché i valori assunti sono molto piccoli.



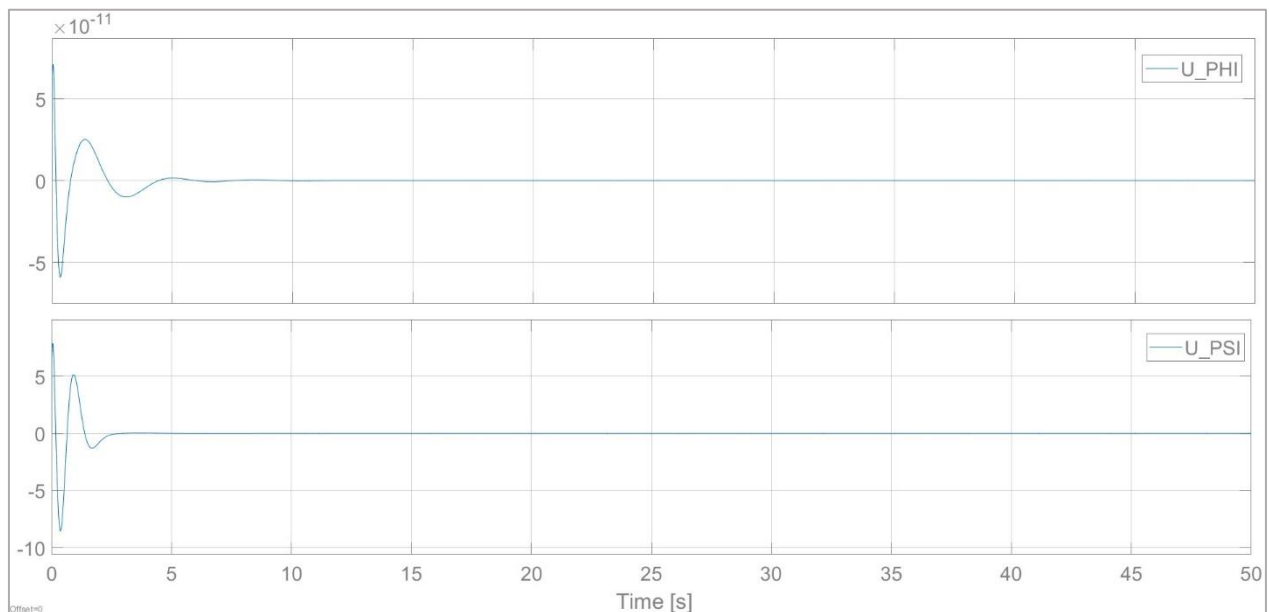
Grafici degli ingressi di controllo

Per quanto riguarda gli ingressi si nota che il controllo dello sterzo è quello più critico.

Risultati caso lineare D-K



Simulazioni per sistema AGV-Attuatori con parametri nominali.



Grafici degli ingressi di controllo

Le considerazioni per il controllore $D - K$ relative alle prestazioni delle componenti osservate sono simili a quelle del controllore \mathcal{H}_∞ .

La differenza, che si può notare, è il minor tempo per il raggiungimento della condizione di regime ed infatti anche il tempo di simulazione scelto è un decimo di quello dell' \mathcal{H}_∞ .

Inoltre, si presenta una minore sovraelongazione per tutte le componenti.

Per quanto riguarda gli ingressi si nota che il controllo dello sterzo è quello più critico.

Analisi di robusta stabilità e robusta prestazione

Per l'analisi di robustezza sono stati utilizzati su Matlab due approcci:

- verifica con plot del valore singolare strutturato *'mussv'*
- verifica con le funzioni *'robuststab'* e *'robustperf'*

Implementazione Matlab

\mathcal{H}_∞

```
clc
% Analisi di robusta stabilità e robusta performance per il controllore D-K itertaion

% Eseguire il codice per la sintesi del controllore (DK_musyn)
H00_MixSen_4sen;

% Pesì relativi alle incertezze
Wi1=(0.5*s^2 + 7983*s + 15e05)/(s^2 + 6.424e04*s + 5.345e07);
Wi=blkdiag(Wi1,Wi1);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Sistema in forma di Doyle (Delta-P-K)
systemnames = 'sys W1 W2 Wi'; % blocchi nel sistema
inputvar = '[udel(2); r(4) ; u(2)]'; % ingressi al blocco P
outputvar = '[Wi; W1; W2; -sys+r]'; % uscite dal blocco P
input_to_sys = '[u+udel]'; % ingressi al sistema sys
input_to_W1 = '[r-sys]'; % ingressi alle performance W1
input_to_W2 = '[u]'; % ingressi alle performance W2
input_to_Wi = '[u]'; % ingressi alle incertezze Wi
sysoutname = 'P';
cleanupsysic = 'yes';
sysic;

%Controllore MixSen
K=KH00;

% Sistema in forma Delta-M
M = lft(P,K); % lower linear fractional trasformation

% Valutazione in frequenza della matrice M
omega = logspace(-6,3,200); % 200 campioni di frequenze tra 10^(-6) rad/s e 10^3 rad/s
Mf = frd(M,omega); % M valutata in omega

% Struttura matrice Delta
Delta=ultidyn('Delta',[2 2]); % forma del blocco legato alle incertezze

% Valutazione in frequenza della matrice N
N = lft(Delta,M); % upper linear fractional trasformation (sistema incerto)
Nf = ufrd(N,omega); % N valutata in omega
```

```
% Analisi di robusta stabilità

%Analisi di robusta stabilità con valore singolare strutturato (SSV)
Mrs = Mf(1:2,1:2); % blocco M11 della matrice Mf
[mubnds,muinfo] = mussv(Mrs,[2 2],'a'); % Robusta stabilità con SSV
muRS = mubnds(:,1); % bound inferiore di SSV
[muRSinf,muRSw] = norm(muRS,inf); % norma H_inf del bound inferiore di SSV (max valore)
figure(1)
bodemag(mubnds(:,1),'r-',mubnds(:,2),'b-') % plot dei bound di SSV
grid on

% Analisi di robusta stabilità con la sensitività complementare in ingresso Ti usando il comando robuststab
Gp = sys*(eye(2)+Delta*Wi); % generico impianto incerto Gp
looptransfer = loopsens(Gp,K); % calcolo delle funzioni Si, Ti, So e To
Ti = looptransfer.Ti; % sensitività complementare in ingresso Ti
Tif = ufrd(Ti,omega); % sensitività complementare in ingresso Ti valutata per omega
opt = robopt('Display','on');
[stabmarg,destabunc,reportrs,infor] = robuststab(Tif,opt); % Robusta stabilità
```

```
% Analisi di robusta performance

BlockStructure = [2 2; 4 6]; % Struttura della matrice Delta_hat=blkdiag(Delta_i,Delta_P)
[mubnds,muinfo] = mussv(Mf,BlockStructure,'a'); % Robusta performance con SSV
muRS = mubnds(:,1); % bound inferiore di SSV
[muRSinf,muRSw] = norm(muRS,inf); % norma H_inf del bound inferiore di SSV (max valore)
figure(2)
bodemag(mubnds(:,1),'r-',mubnds(:,2),'b-') % plot dei bound di SSV
grid on

% Analisi di robusta performance con il comando robustperf
opt = robopt('Display','on');
[stabmarg,destabunc,reportrp,inforp] = robustperf(Nf,opt); % Robusta performance
```

D-K

```
% Analisi di robusta stabilità e robusta performance per il controllore D-K iterativo

% Eseguire il codice per la sintesi del controllore (DK_musyn)
DK_musyn

% Pesì relativi alle incertezze
Wi1=(0.5*s^2 + 7983*s + 15e05)/(s^2 + 6.424e04*s + 5.345e07);
Wi=blkdiag(Wi1,Wi1);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Sistema in forma di Doyle (Delta-P-K)
systemnames = 'sys W1 W2 Wi'; % blocchi nel sistema
inputvar = '[udel(2); r(4) ; u(2)]'; % ingressi al blocco P
outputvar = '[Wi; W1; W2; -sys+r]'; % uscite dal blocco P
input_to_sys = '[u+udel]'; % ingressi al sistema sys
input_to_W1 = '[r-sys]'; % ingressi alle performance W1
input_to_W2 = '[u]'; % ingressi alle performance W2
input_to_Wi = '[u]'; % ingressi alle incertezze Wi
sysoutname = 'P';
cleanupsysic = 'yes';
sysic;

% Sistema in forma Delta-M
M = lft(P,K); % lower linear fractional transformation

% Valutazione in frequenza della matrice M
omega = logspace(-3,3,300); % 300 campioni di frequenze tra 10^(-3) rad/s e 10^3 rad/s
Mf = frd(M,omega); % M valutata in omega

% Struttura matrice Delta
Delta=ultidyn('Delta',[2 2]); % forma del blocco legato alle incertezze

% Valutazione in frequenza della matrice N
N = lft(Delta,M); % upper linear fractional transformation (sistema incerto)
Nf = ufrd(N,omega); % N valutata in omega
```

```

% Analisi di robusta stabilità

%Analisi di robusta stabilità con valore singolare strutturato (SSV)
Mrs = Mf(1:2,1:2); % blocco M11 della matrice Mf
[mubnds,muinfo] = mussv(Mrs,[2 2],'a'); % Robusta stabilità con SSV
muRS = mubnds(:,1); % bound inferiore di SSV
[muRSinf,muRSw] = norm(muRS,inf); % norma H_inf del bound inferiore di SSV (max valore)
figure(1)
bodemag(mubnds(:,1),'r-',mubnds(:,2),'b-') % plot dei bound di SSV
grid on

% Analisi di robusta stabilità con la sensitività complementare in ingresso Ti usando il comando robuststab
Gp = sys*(eye(2)+Delta*Wi); % generico impianto incerto Gp
looptransfer = loopsens(Gp,K); % calcolo delle funzioni Si, Ti, So e To
Ti = looptransfer.Ti; % sensitività complementare in ingresso Ti
Tif = ufrd(Ti,omega); % sensitività complementare in ingresso Ti valutata per omega
opt = robust('Display','on');
[stabmarg,destabunc,reportrs,infor] = robuststab(Tif,opt); % Robusta stabilità

```

```

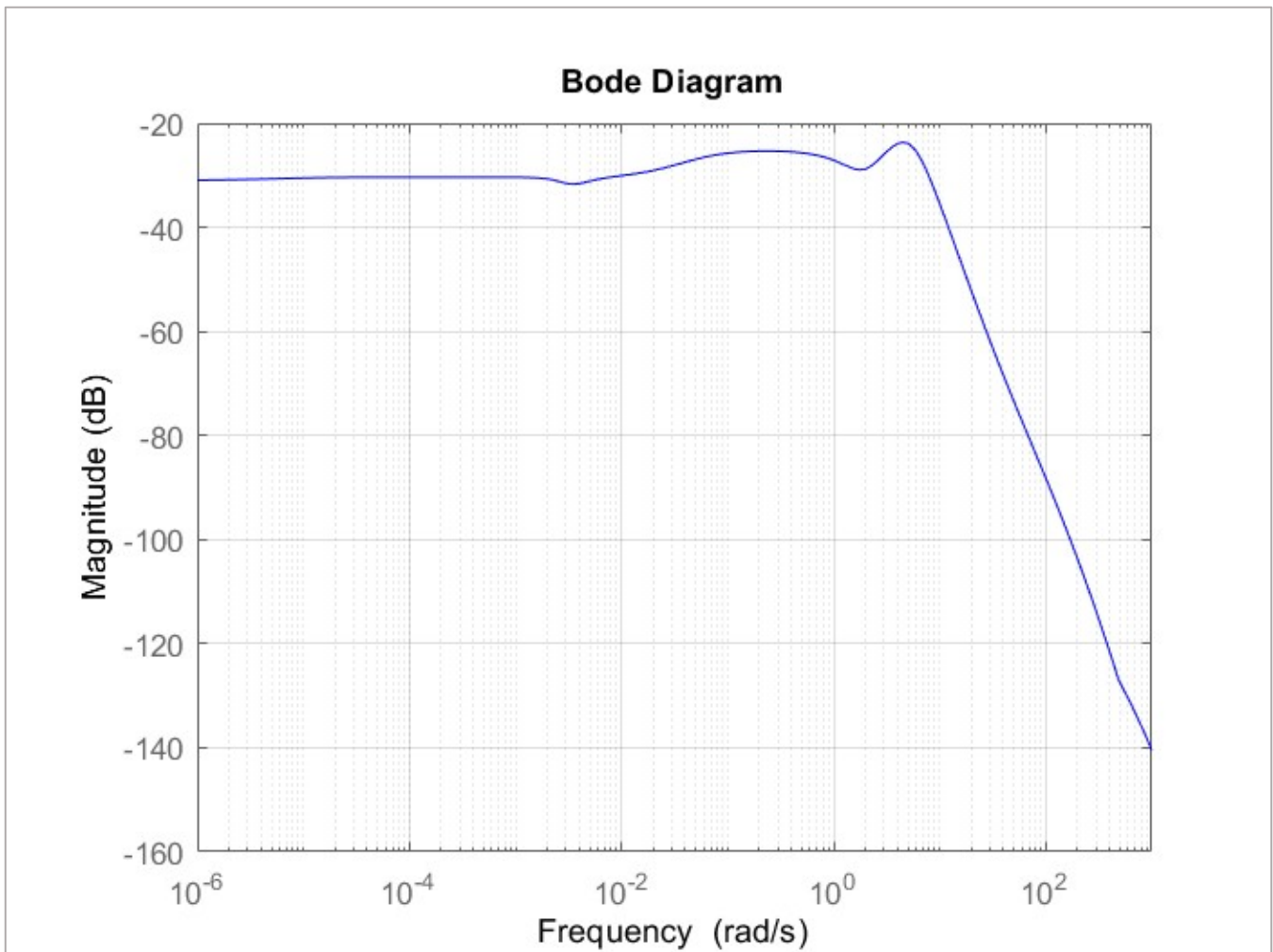
% Analisi di robusta performance

% Analisi di robusta performance con il valore singolare strutturato (SSV)
BlockStructure = [2 2; 4 6]; % Struttura della matrice Delta_hat=blkdiag(Delta_i,Delta_P)
[mubnds,muinfo] = mussv(Mf,BlockStructure,'a'); % Robusta performance con SSV
muRS = mubnds(:,1); % bound inferiore di SSV
[muRSinf,muRSw] = norm(muRS,inf); % norma H_inf del bound inferiore di SSV (max valore)
figure(2)
bodemag(mubnds(:,1),'r-',mubnds(:,2),'b-') % plot dei bound di SSV
grid on

% Analisi di robusta performance con il comando robustperf
opt = robust('Display','on');
[stabmarg,destabunc,reportrp,inforp] = robustperf(Nf,opt); % Robusta performance

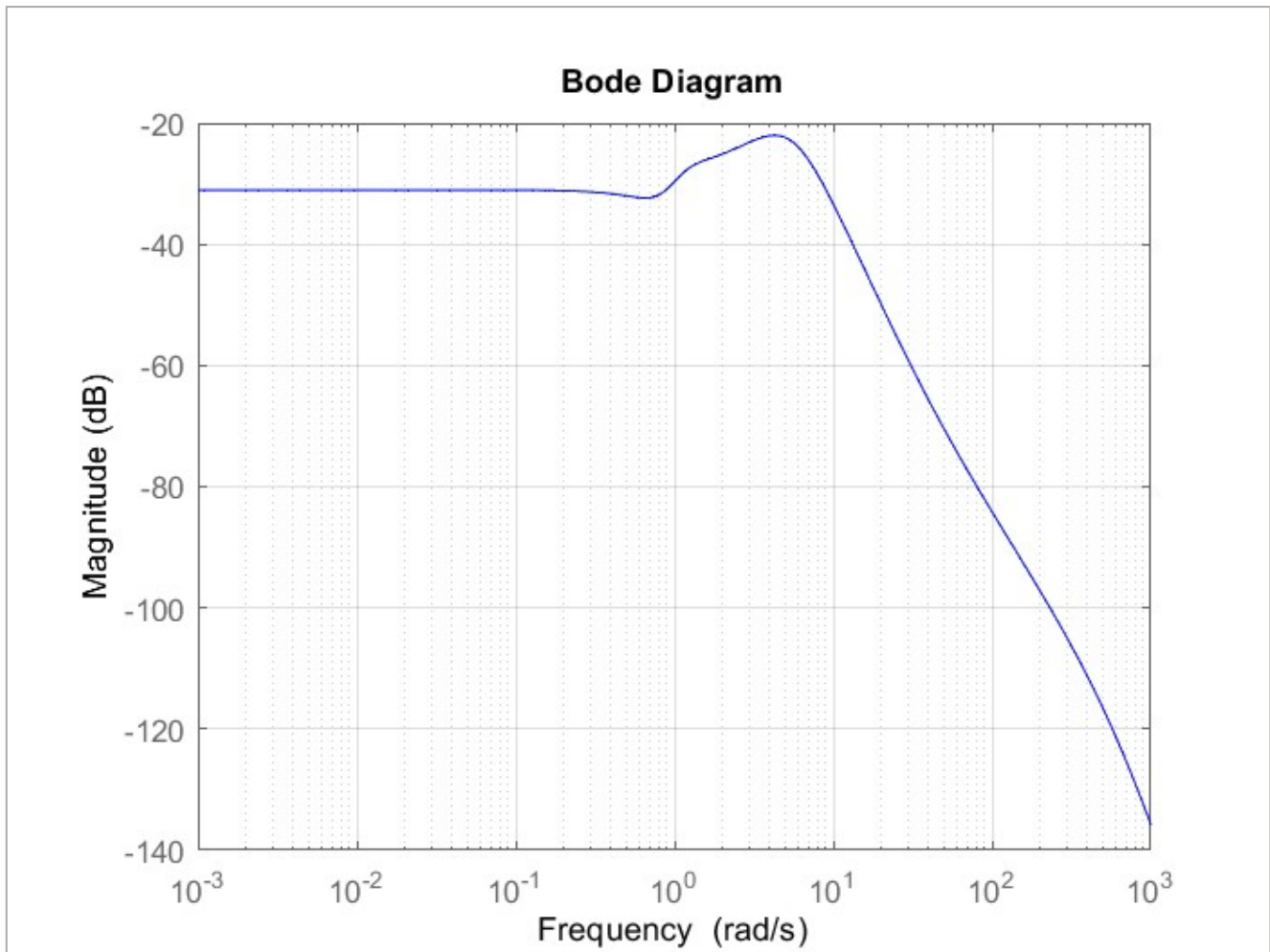
```


Confronto robusta stabilità



```
reportrs =  
  
6×101 char array  
  
'Assuming nominal UFRD system is stable...↵Uncertain system is robustly stable to modeled uncertainty.'  
' -- It can tolerate up to 1.53e+03% of the modeled uncertainty. '  
' -- There is a destabilizing perturbation amounting to 1.53e+03% of the modeled uncertainty. '  
' -- This perturbation causes an instability at the frequency 4.45 rad/seconds. '  
' -- Sensitivity with respect to each uncertain element is: '  
' 100% for Delta. Increasing Delta by 25% decreases the margin by 25%. '
```

Grafico del valore singolare strutturato di M nel caso \mathcal{H}_∞ per la robusta stabilità e report del comando `'robuststab'`.



```
reportrs =
6x101 char array

'Assuming nominal UFRD system is stable...↵Uncertain system is robustly stable to modeled uncertainty.'
' -- It can tolerate up to 1.26e+03% of the modeled uncertainty.
' -- There is a destabilizing perturbation amounting to 1.26e+03% of the modeled uncertainty.
' -- This perturbation causes an instability at the frequency 4.29 rad/seconds.
' -- Sensitivity with respect to each uncertain element is:
'     100% for Delta. Increasing Delta by 25% decreases the margin by 25%.
```

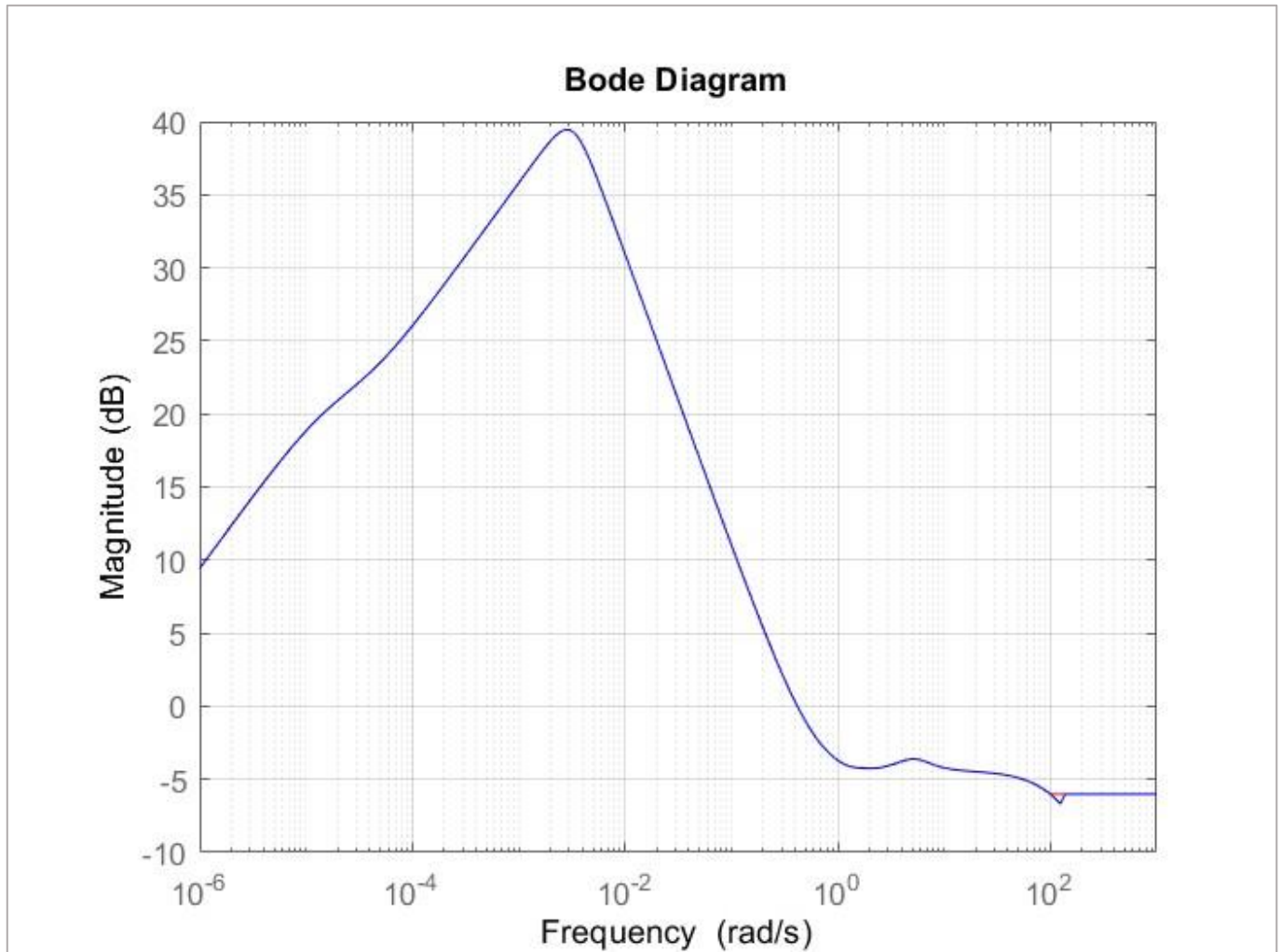
Grafico del valore singolare strutturato di M nel caso $D - K$ per la robusta stabilità e report del comando `'robuststab'`.

In entrambi casi si hanno margini di stabilità molto ampi.

Per il controllore \mathcal{H}_∞ è tollerabile un' incertezza pari a 15 volte quella data prima di raggiungere l' instabilità, mentre per il controllore $D - K$ l'incertezza tollerabile è pari a 12 volte.

Si conclude quindi che il controllore migliore è l' \mathcal{H}_∞ per la robusta stabilità.

Confronto robusta prestazione



```
reportrp =  
  
5×113 char array  
  
'Uncertain system does not achieve performance robustness to modeled uncertainty.'  
' -- The tradeoff of model uncertainty and system gain is balanced at a level of 1.06% of the modeled uncertainty.'  
' -- A model uncertainty of 1.06% can lead to input/output gain of 93.9 at 0.00274 rad/seconds.'  
' -- Sensitivity with respect to each uncertain element is:  
' 47% for Delta. Increasing Delta by 25% decreases the margin by 11.8%.
```

Grafico del valore singolare strutturato di M nel caso \mathcal{H}_∞ per la robusta prestazione
e report del comando *'robustperf'*.

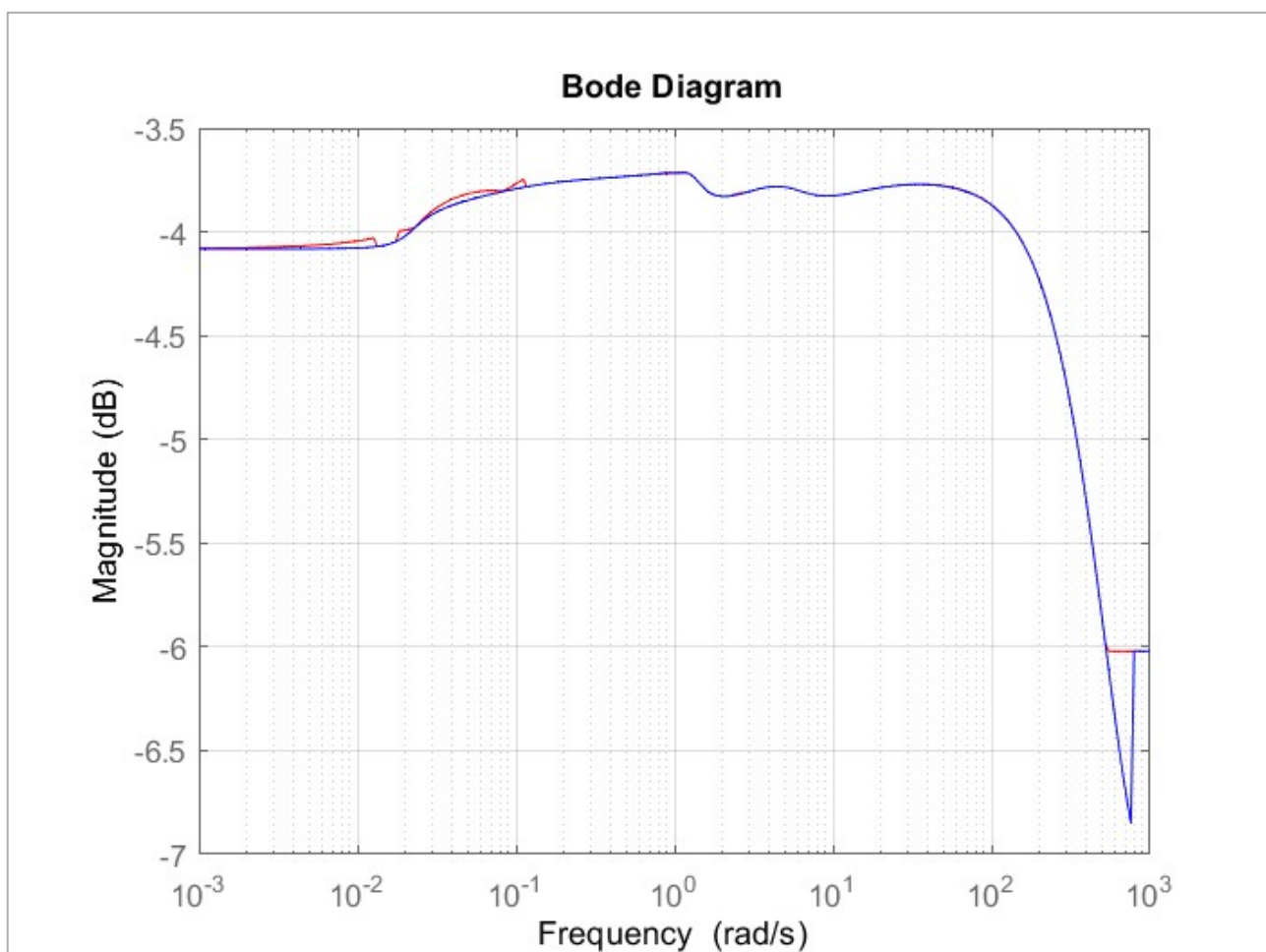
Per quanto riguarda la robusta prestazione per il caso \mathcal{H}_∞ questa non viene raggiunta; infatti, il valore singolare strutturato ha un picco molto elevato a una frequenza di circa 10^{-3} rad/s. La verifica fatta si basa su ipotesi molto conservative perché tutte le incertezze sono state raggruppate in un' unica incertezza non tenendo conto di eventuali strutture.

Ripetendo la verifica con incertezze non concentrate, ovvero passando direttamente il sistema incerto a Matlab il margine migliora del 10%, ma comunque non garantisce la robusta prestazione.

```
reportrp =
11x131 char array

'Assuming nominal UFRD system is stable...Uncertain system possibly does not achieve performance robustness to modeled uncertainty.'
' -- The tradeoff of model uncertainty and system gain is balanced at a level of 81.9% of the modeled uncertainty.
' -- A model uncertainty of 138% can lead to input/output gain of 0.724 at 6.75 rad/seconds.
' -- Sensitivity with respect to each uncertain element is:
'   150% for Km1. Increasing Km1 by 25% decreases the margin by 37.5%.
'   41% for Km2. Increasing Km2 by 25% decreases the margin by 10.2%.
'   7% for T1. Increasing T1 by 25% decreases the margin by 1.75%.
'   2% for Tm1. Increasing Tm1 by 25% decreases the margin by 0.5%.
'   28% for Tm2. Increasing Tm2 by 25% decreases the margin by 7%.
'   392% for ra. Increasing ra by 25% decreases the margin by 98%.
'   317% for rp. Increasing rp by 25% decreases the margin by 79.2%.
```

Report robusta prestazione con sistema incerto.



```
reportrp =
5x114 char array

'Assuming nominal UFRD system is stable...Uncertain system achieves performance robustness to modeled uncertainty.'
' -- The tradeoff of model uncertainty and system gain is balanced at a level of 153% of the modeled uncertainty.
' -- A model uncertainty of 153% can lead to input/output gain of 0.652 at 1.18 rad/seconds.
' -- Sensitivity with respect to each uncertain element is:
'   11% for Delta. Increasing Delta by 25% decreases the margin by 2.75%.
```

Grafico del valore singolare strutturato di M nel caso $D - K$ per la robusta prestazione e report del comando 'robustperf'.

Nel caso del controllore $D - K$ si riesce a garantire la robusta prestazione anche con le ipotesi conservative.

Riassumendo la trattazione:

- Il controllore LQG è semplice da implementare e utilizza un' unico modello per tutte le incertezze, ma spesso non è una corretta modellazione e non ha margini di robustezza.
- Il controllore \mathcal{H}_∞ , a sua volta, permette un migliore approccio alla modellazione delle incertezze, ma se si basa su ipotesi conservative (incertezza non strutturata) non riesce a garantire la robusta prestazione.
- Il controllore $D - K$, basandosi sulla μ -analisi, riesce a garantire la robusta prestazione dopo varie iterazioni.

In conclusione, per le specifiche richieste sulla traiettoria dell' AGV solamente il controllore $D - K$ riesce a garantirle.

9 – Bibliografia e Sitografia

- *'Multivariable feedback control: Analysis and Design'*, Skogestad and Postlethwaite.
- *'Robust control design with MATLAB®'*, Gu, D. W., Petkov, P., and Konstantinov, M. M.
- <https://it.mathworks.com/help/matlab/>
- Cartella drive con i codici Matlab:
https://drive.google.com/drive/folders/1ocZr8xAVj3k6ijOuOh6_SLJcXj8qBG9X?usp=sharing