

PROGETTO ESAME AUTOMATICA BALL&BEAM

Bendazzoli Luca	10661018
Guala Pietro	10629553
Tamburini Tommaso	10631799

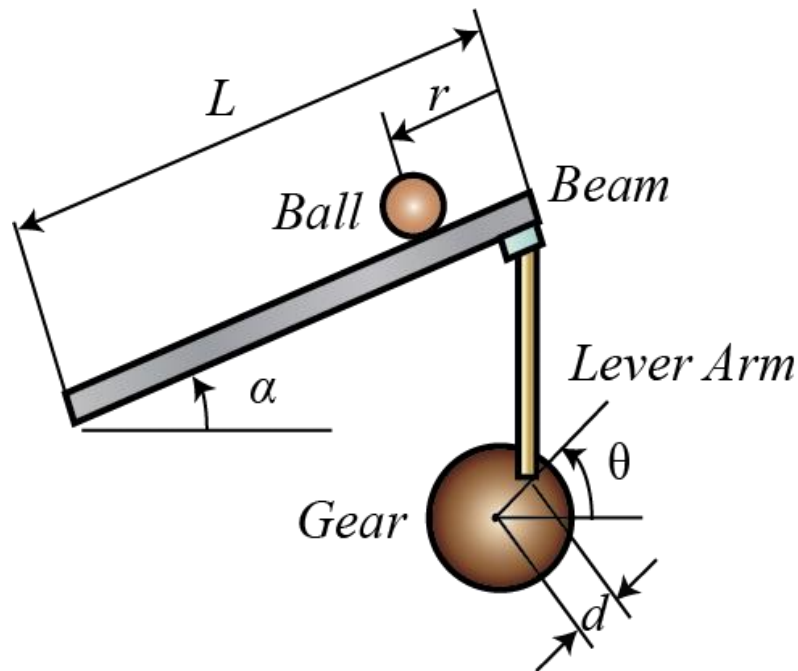
INDICE GENERALE

1. Introduzione *[Pag.2]*
2. Premessa progettuale *[Pag.4]*
3. Parte I *[Pag.6]*
4. Parte II *[Pag.12]*

Il testo presenta dei collegamenti ipertestuali per facilitarne la visualizzazione e la comprensione

Introduzione

L'argomento di questo progetto riguarda lo studio della stabilità di un sistema meccanico e la progettazione di un controllore che rispetti i vincoli sull'uscita imposti dal problema, applicando le nozioni acquisite durante il corso di Fondamenti di Automatica.



Il sistema iniziale non è lineare ma l'equazione fornita descrive il sistema linearizzato nell'intorno del punto di equilibrio ($\theta = 0$).

Sistema non lineare:

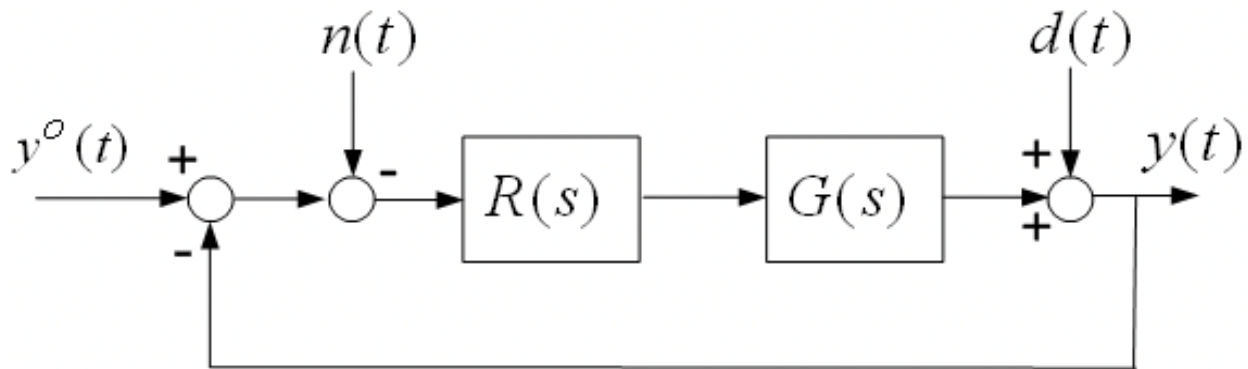
$$mg \sin(\theta) - \left(\frac{J}{R^2} + m \right) \ddot{r} - m(L - r)\dot{\theta}^2 = 0$$

Sistema linearizzato:

$$\left(\frac{J}{R^2} + m \right) \ddot{r} = mg \frac{d}{L} \theta$$

Per rispettare gli assi nel disegno fornito abbiamo cambiato di segno il secondo membro dell'equazione con lo scopo di avere concordanza tra l'aumento dell'angolo θ e la coordinata ' r ' che individua la discesa della pallina.

Ci teniamo a fornire un breve accenno alla teoria dei sistemi retroazionati:



Schema a blocchi con:

- $R(s)$ = Regolatore
- $G(s)$ = FdT del nostro sistema

$L(s) = R(s) \cdot G(s)$ = FdT d'anello retroazionato

Lo scopo del nostro lavoro sarà quello di progettare $R(s)$ in modo che il sistema di controllo verifichi delle specifiche assegnate.

Lo schema a blocchi sopra rappresenta un sistema retroazionato completo, con tutti i possibili segnali ingresso ($n(t)$, $d(t)$, $y^o(t)$) e uscita ($y(t)$, $u(t)$, $e(t)$).

Nel nostro caso il sistema risulta semplificato perché $n(t) = 0$ e $d(t) = 0$.

Premessa Progettuale:

L'obiettivo del progetto è quello di individuare un regolatore che stabilizzi il sistema dato in anello chiuso, per realizzarlo abbiamo scritto un codice in MatLab che iterativamente esegua tentativi e fornisca la soluzione in modo preciso e efficace (metodo a tentativi implementato da un calcolatore che ha una capacità di risoluzione di gran lunga più precisa).

Per garantire l'asintotica stabilità sappiamo che è consigliato attraversare l'asse a 0dB con una pendenza di -20 dB/dec, siccome il nostro sistema non regolato avrebbe un attraversamento a -40 dB/dec ci rendiamo conto di dover inserire uno zero con $\text{Re}(z) < 0$ (per garantirne la stabilità) precedentemente alla pulsazione iniziale.

Poiché il regolatore deve rispettare i vincoli di realizzabilità il suo grado relativo r non deve essere negativo è necessario dunque inserire anche un polo con pulsazione maggiore di quella iniziale.

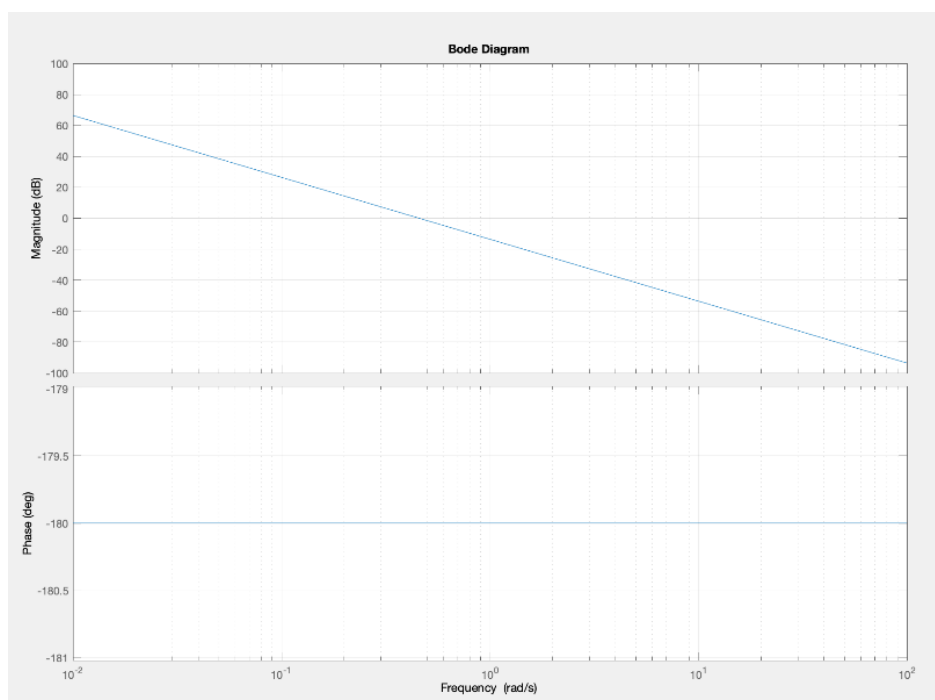


Diagramma di Bode del sistema $G(s)$

Il problema fornisce due vincoli dinamici principali:

- Sovraelongazione percentuale ($S_{\%}$)
- Tempo di assestamento (T_a)

Per soddisfare i vincoli abbiamo costruito un ciclo iterativo che a partire da una pulsazione ω imposta, posiziona uno zero con una frequenza più bassa e un polo con una frequenza più alta in modo simmetrico, fino a quando vengono rispettate le richieste progettuali.

La pulsazione di partenza l'abbiamo stabilita con un codice MatLab secondario (<CODICE OMEGA>), non appartenente al programma principale, che calcola quale sia una pulsazione ragionevole affinché il programma riesca a restituire una soluzione robusta.

Dai risultati generati abbiamo dedotto che per $\omega = 3$ il programma principale esegue il numero minimo di iterazioni però il margine di fase è basso rispetto ad altri valori di ω considerati.

Per $\omega = 35$ infatti, il margine di fase risulta il più grande valore ottenibile, questo però comporta un numero troppo elevato di casi analizzati prima di trovare la soluzione.

Abbiamo scelto $\omega = 10$ perché richiede un numero ragionevole di tentativi e il margine di fase risulta altrettanto appropriato.

<i>Omega (ω)</i>	<i>Margine di fase (φ_m)</i>	<i>Numero tentativi*</i>
3	70.6°	173
35	86.3°	375
10	84.1°	259

*numero di casi analizzati prima di trovare una soluzione che rispetti i vincoli.

L'obiettivo principale del progetto è regolare la sfera in una posizione di equilibrio; con il codice sotto riportato sarà l'utente a decidere in quale posizione la pallina dovrà assestarsi inserendo il valore in una finestra di dialogo:

```
%Scelta posizione di assestamento da parte dell'utente
while true
    PROMPT = {'Inserire un valore compreso tra 0 e 1 attorno al quale si vuole
assestare la pallina'};
    NAME = 'POSIZIONE FINALE';
    r_fin_cell = inputdlg(PROMPT, NAME);
    r_fin = str2double(r_fin_cell{1, 1});
    if r_fin > 0 && r_fin < 1
        break;
    end
end
```

È inoltre presente un ciclo *while* per assicurarsi che la posizione di assestamento sia valida, infatti l'esecuzione non esce dal ciclo fino a quando non viene inserito un valore corretto ovvero compreso tra 0 e 1 (lunghezza dell'asta).

Analogamente per la seconda richiesta della [PARTE 2] è l'utente a dover scegliere la frequenza del disturbo sinusoidale dell'ingresso.

Abbiamo inoltre deciso di generare un file ('PRESTAZIONI.txt') contenente le prestazioni e le specifiche del regolatore ottenuto per un'analisi immediata e più semplice dei risultati notevoli.

Nota: Tutti i grafici e i risultati riportati sono stati calcolati per una posizione finale $r_{fin} = 0.25$.

PARTE 1:

1. Scrivere il sistema in forma di stato, con ingresso l'angolo $\theta(t)$ e uscita la posizione della massa.
2. Realizzare in ambiente Matlab/Simulink un modello del sistema con i parametri dati. Effettuare un test sul sistema con variazione a scalino dell'angolo $\theta(t)$.
3. Calcolare la FDT del sistema con ingresso $\theta(t)$ e uscita $r(t)$.
4. Tracciare la risposta del sistema ad un ingresso a scalino, verificando la coerenza della risposta ottenuta con quella risultante dal test fatto sul sistema in simulazione, commentando i risultati.
5. Stimare la banda del sistema in anello aperto e caratterizzare la risposta a scalino in termini di valore iniziale, valore finale, tempo di assestamento e entità delle oscillazioni.

SOLUZIONE:

1. Scrivere il sistema in forma di stato, con ingresso l'angolo $\theta(t)$ e uscita la posizione della massa:

Ingresso = $\theta(t) = U(s)$

Uscita = $r(t) = Y(s)$

$$\begin{bmatrix} \dot{r}(t) \\ \ddot{r}(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} r(t) \\ \dot{r}(t) \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{mgd}{L(\frac{J}{R^2} + m)} \end{bmatrix} \theta(t)$$

$$y(t) = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} r(t) \\ \dot{r}(t) \end{bmatrix}$$

Tenendo conto del sistema:

$$\begin{cases} \dot{x}_1(t) = x_2(t) = \dot{r}(t) \\ \dot{x}_2(t) = \ddot{x}_1(t) = \ddot{r}(t) \\ y(t) = x_1(t) = r(t) \end{cases}$$

2. Realizzare in ambiente Matlab/Simulink un modello del sistema con i parametri dati. Effettuare un test sul sistema con variazione a scalino dell'angolo $\theta(t)$:

Qui di seguito viene riportato il codice MatLab del sistema dove abbiamo inserito i dati del problema all'interno di una struct, successivamente abbiamo scritto le matrici del sistema per poterlo definire in spazio di stato:

```
%dati%
DATI = struct('m', 0.11, 'g', 9.81, 'L', 1, 'd', 0.03, 'R', 0.015, 'J', 9.99*10^-6);
m_eq = DATI.m + (DATI.J / DATI.R^2);
k = (DATI.m * DATI.g * DATI.d) / (m_eq * DATI.L);

%matrici del sistema
A = [0 1; 0 0];
B = [0, k]';
C = [1 0];
D = 0;
sist = ss(A, B, C, D);

%avvio simulazione simulink
SimOut = sim('ball_beam_simulink_diff_eq', 'ReturnWorkspaceOutputs', 'on');
```

Dal sistema in spazio di stato abbiamo rappresentato il modello in ambiente Simulink, simulando l'uscita in risposta ad un riferimento a scalino e salvando le variabili nel workspace:

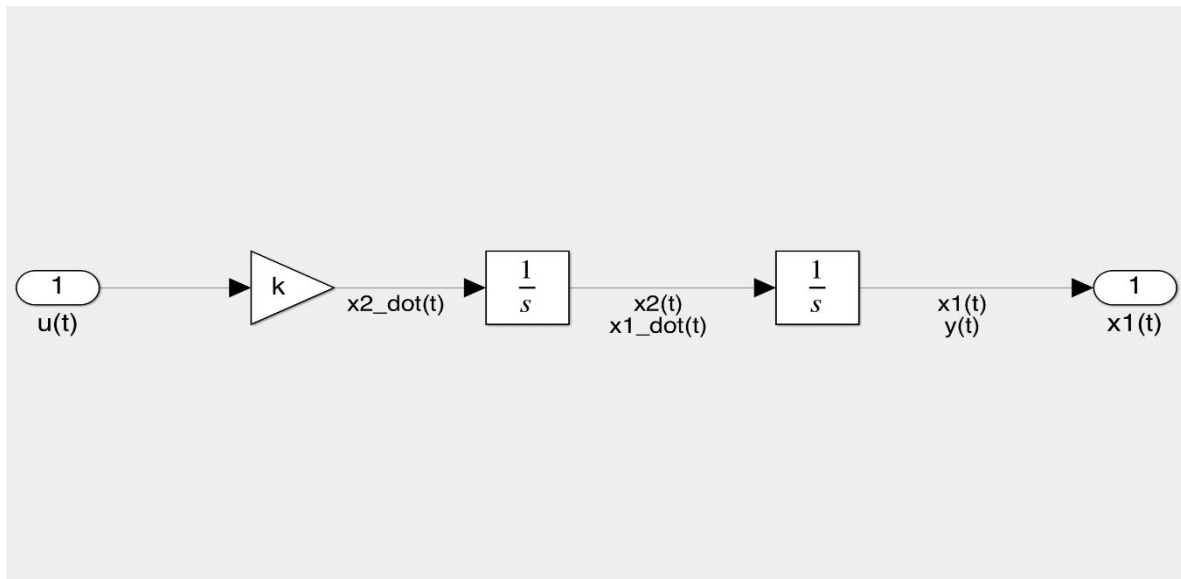


Figura 1: modello Ball&Beam.

Il modello 'ball_beam_simulink.slx' presenta in ordine un blocco 'inport' per il segnale in ingresso; blocco 'Gain' per la specifica del guadagno; due blocchi 'integrator' per definire la funzione di trasferimento insieme al blocco 'Gain'; infine il blocco 'outport' per definire il segnale d'uscita.

Gli schemi Simulink sono stampati a video nel dock sotto forma di figura durante l'esecuzione del codice.

3. Calcolare la FDT del sistema con ingresso $\theta(t)$ e uscita $r(t)$:

Risolvendo il sistema trasformandolo nel dominio di Laplace (in funzione di s) e con condizioni iniziali nulle troviamo:

$$Y(s) = \frac{k\Theta}{s^2}$$

Con

$$k = \frac{mgd}{l(\frac{J}{R^2} + m)}$$

Da cui possiamo ricavare la Funzione di Trasferimento:

$$G(s) = \frac{Y(s)}{U(s)} = \frac{k}{s^2} \quad [1.3.1]$$

Notiamo che la FdT ha:

- Tipo $g = 2$
- Due poli in $s = 0$

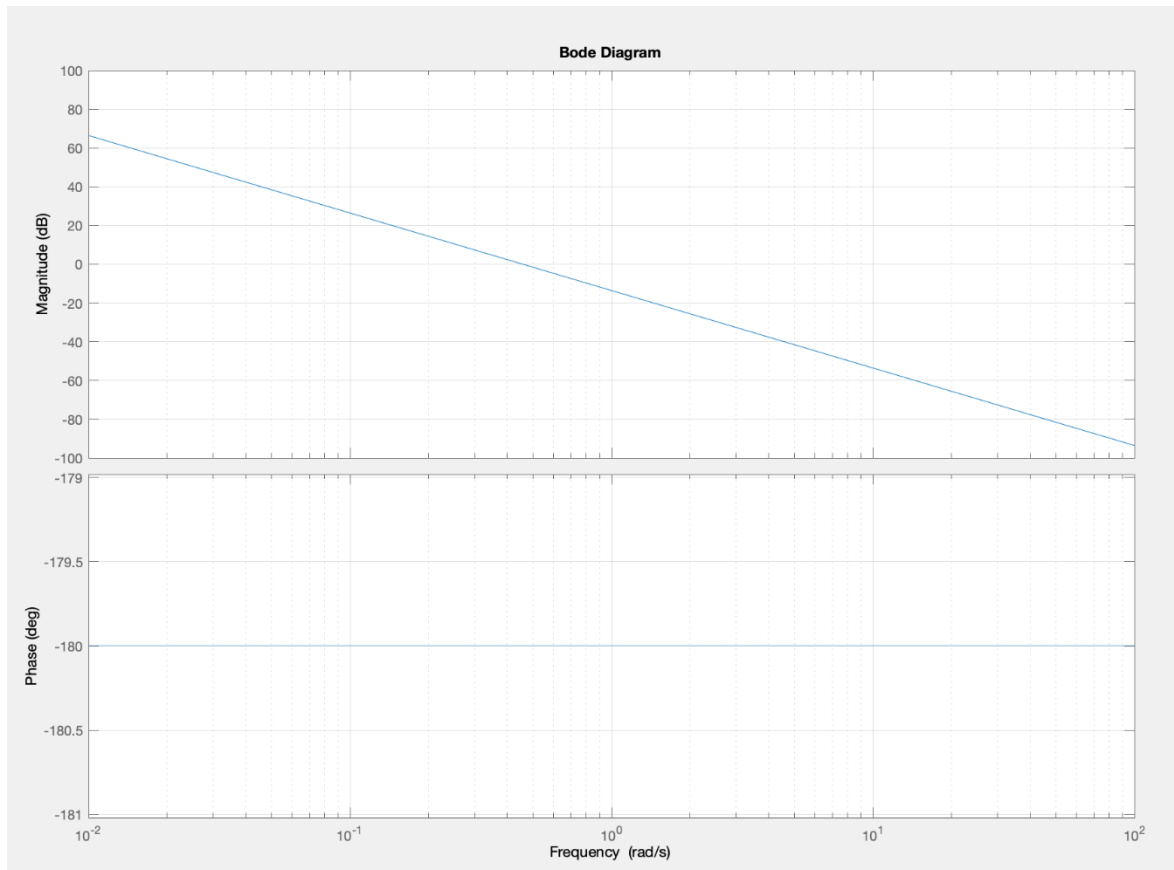
Ricordiamo:

Ingresso = $\theta(t) = U(s)$

Uscita = $r(t) = Y(s)$

Codice MatLab riferito:

```
%funzione di trasferimento  
G = tf(sist);  
set(figure , 'WindowStyle' , 'docked');  
  
bode(sist , {0.01, 100});  
grid on;
```



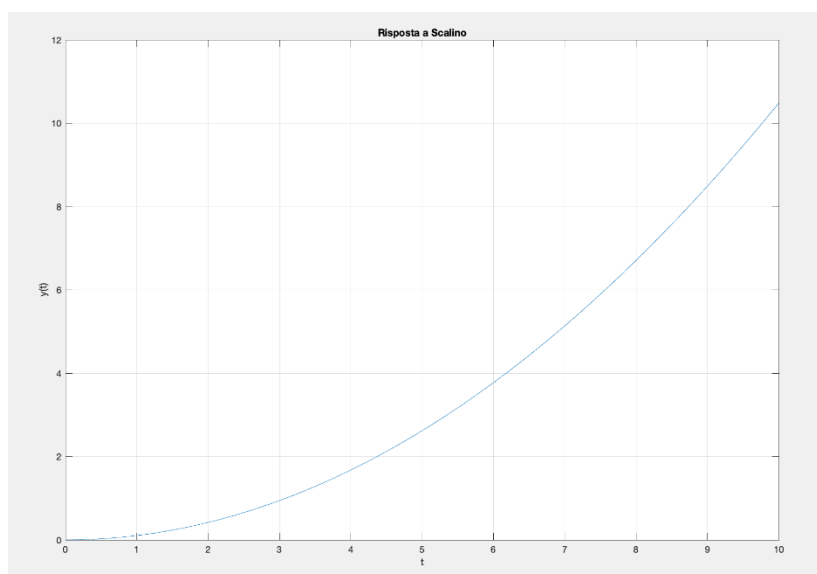
Dal diagramma di Bode si può notare che il sistema non è stabile perché la FdT [1.3.1] non ancora regolata, presenta due poli nell'origine che portano la fase ad essere costante al valore di -180° rendendo nullo il margine di fase.

Per stabilizzare un sistema di questo tipo è necessario utilizzare il procedimento descritto nella premessa.

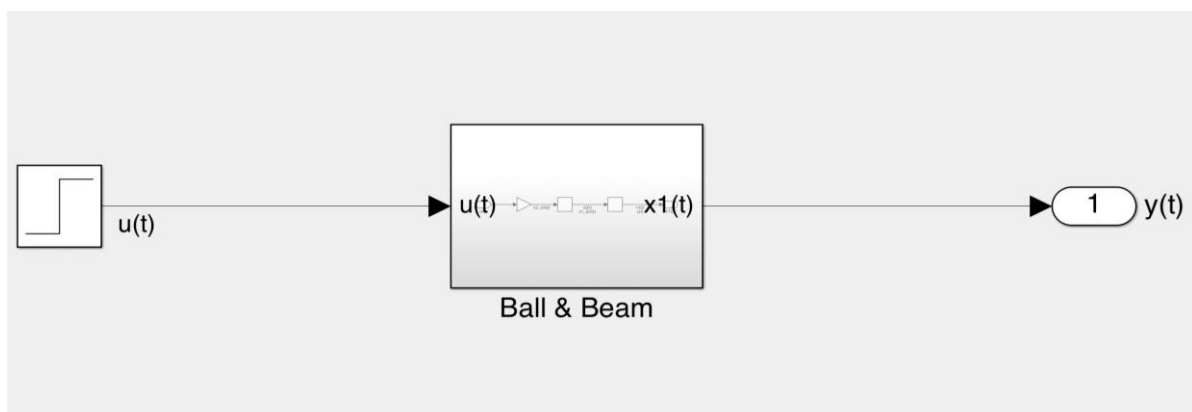
4. Tracciare la risposta del sistema ad un ingresso a scalino, verificando la coerenza della risposta ottenuta con quella risultante dal test fatto sul sistema in simulazione, commentando i risultati:

Codice MatLab riferito:

```
%risposta a scalino
t = 0 : 0.01 : 10;
[y , t , x] = step(sist , t);
set(figure , 'WindowState' , 'docked');
plot (t , y);
grid on;
title('Risposta a Scalino');
xlabel('t');
ylabel('y(t)');
```



In modo equivalente, su Simulink poniamo il blocco 'step' al modello *Ball&Beam* per ottenere la risposta allo scalino come abbiamo fatto su MatLab



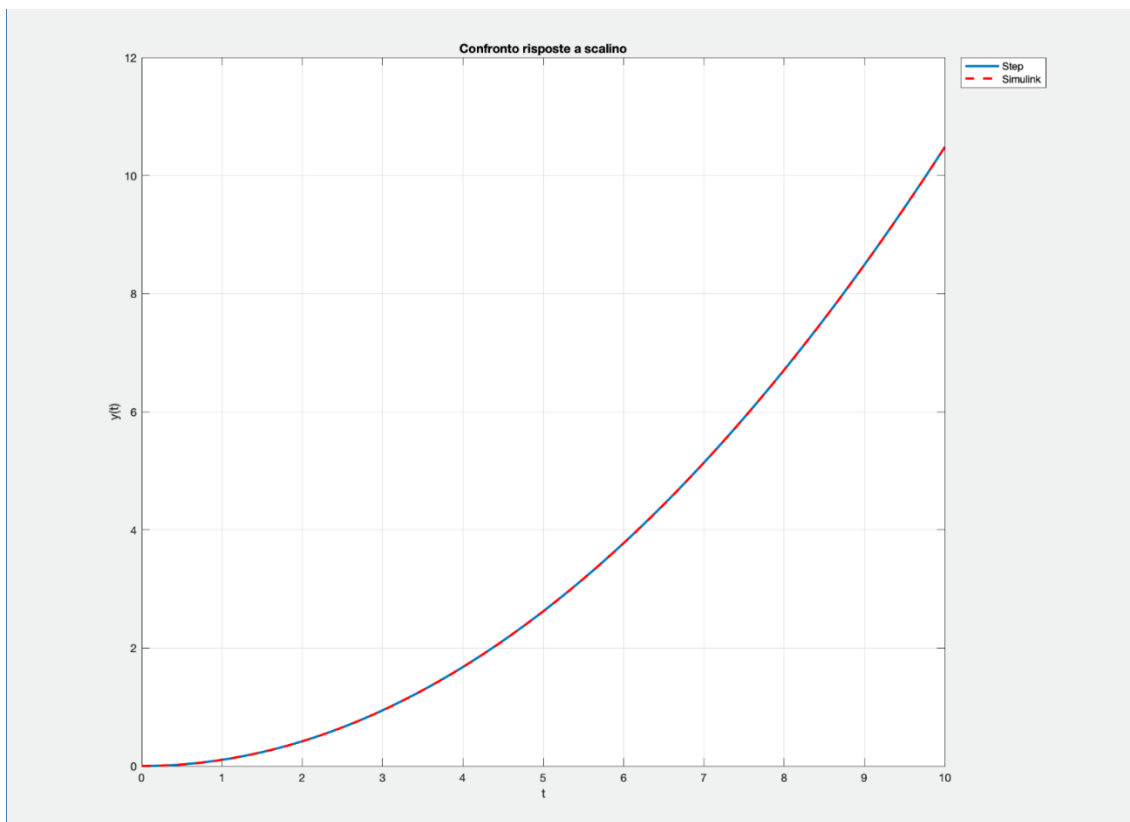
*il blocco all'interno della casella *Ball&Beam* in figura è rappresentato dalla [Figura 1]

Confronto risposte a scalino [MatLab](#)/ [Simulink](#):

```
%confronto simulazioni
set(figure, 'WindowStyle', 'docked');
plot(t, y, 'linewidth', 2);

hold on

plot (SimOut.tout ,SimOut.yout,'r--', 'linewidth', 2);
grid on;
title('Confronto risposte a scalino');
xlabel('t');
ylabel('y(t)');
legend('Step', 'Simulink');
set(legend, 'Location', 'bestoutside');
```



Viene verificato anche a livello grafico che le risposte a scalino del sistema calcolate sia con MatLab che con Simulink non differiscono nonostante i diversi risolutori utilizzati.

Nell'ambiente MatLab il risolutore calcola il valore dell'uscita in funzione di un vettore t con passo 0.01
 $t = 0 : 0.01 : 10;$

Simulink invece utilizza un *variable step* che definisce il passo in modo dinamico a seconda del segnale da calcolare inoltre è possibile definire il risolutore da utilizzare.

Noi abbiamo preferito affidarci al risolutore automatico di Simulink inserendo come limite massimo del passo 0.01, ovvero lo stesso valore che abbiamo utilizzato su MatLab.

Abbiamo inoltre cambiato il valore iniziale della simulazione, da quello di default ($t=1$) a $t=0$ per ottenere un confronto migliore delle due risposte.

5. Stimare la banda del sistema in anello aperto e caratterizzare la risposta a scalino in termini di valore iniziale, valore finale, tempo di assestamento e entità delle oscillazioni:

La banda passante (BP) del sistema in anello aperto è $[0, \omega_c]$ con ω_c = pulsazione di attraversamento del grafico intersecato con l'asse a 0 dB:

analizzando la risposta in frequenza $L(j\omega) = \frac{k}{(j\omega)^2} = 1$

dunque:

$$\omega_c = \sqrt{k}$$

Da cui possiamo evincere che BP= $[0; 0,5]$ circa.

Mediante il Teorema del Valore Iniziale (TVI) e il Teorema del Valore Finale (TVF) possiamo studiare la risposta ad un ingresso a scalino:

Ricordiamo che possono essere utilizzati perché soddisfano entrambi le rispettive condizioni di applicabilità, infatti:

- Hp TVI:
Y(s) funzione razionale
 $r \geq 1$
- Hp TVF:
Y(s) funzione razionale
Poli con $Re() < 0$ oppure nulli in $s = 0$

Dunque:

Ingresso a scalino: $\Theta = \frac{1}{s}$

$$Y(s) = \frac{k\Theta}{s^2} = \frac{k}{s^2} \frac{1}{s} = \frac{k}{s^3}$$

- TVI: si nota che la FdT ha grado relativo $r = 2$, dunque constatiamo che:

$$\begin{aligned} y(0) &= 0 \\ \dot{y}(0) &= 0 \\ \ddot{y}(0) &\neq 0, \end{aligned}$$

infatti:

$$\ddot{y}(0) = \lim_{s \rightarrow \infty} s^2 [sY(s) - \dot{y}(0)] = k$$

- TVF: rappresenta il valore a regime che la nostra FdT avrà:
 $y_\infty = \lim_{s \rightarrow 0} s Y(s) = \infty$

T_a non esiste perché non ha un valore di equilibrio non essendo stabile il sistema in anello aperto.

$S_{\%}$ non esiste perché, essendo divergente, non esiste il picco.

PARTE 2:

1. Progettare un controllore che permetta di regolare la posizione della massa sferica ad un valore costante desiderato, soddisfacendo i seguenti requisiti:

1. (a) Sistema in anello chiuso asintoticamente stabile.
2. (b) Tempo di assestamento non superiore a 3 s.
3. (c) Sovraelongazione massima $S\% = 5\%$.
4. (d) Errore a transitorio esaurito nullo a fronte di un riferimento a scalino.

2. In un setting non ideale, la movimentazione dell'ingranaggio può provocare delle vibrazioni. Tenuto conto di ciò, mostrare in simulazione il comportamento del sistema in presenza di disturbi sinusoidali additivi sull'ingresso $u(t)$, dimensionandone ampiezza e pulsazione in modo che il sistema in anello chiuso possa o non possa attenuarli, e discutere i risultati ottenuti.

RISOLUZIONE:

1. Progettare un controllore che permetta di regolare la posizione della massa sferica ad un valore costante desiderato, soddisfacendo i seguenti requisiti:

Nota: visto il metodo utilizzato per risolvere il problema è stato necessario in primis imporre la condizione d) per permettere di definire il codice che in seguito risolve la parte dinamica del progetto.

d) Errore a transitorio esaurito nullo a fronte di un riferimento a scalino:

Come spiegato nell'introduzione abbiamo un vincolo statico sulla la funzione d'anello che impone $g_L > 0$. Sapendo che il tipo della funzione d'anello equivale a:

$$g_L = g_G + g_R$$

Poiché $g_G = 2$ abbiamo scelto per comodità $g_R = 0$.

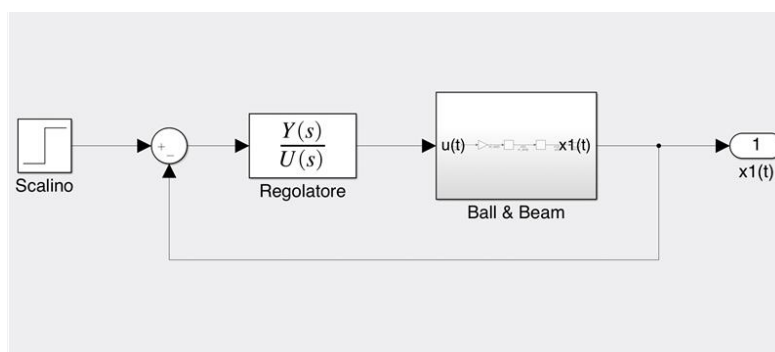
a) Sistema in anello chiuso asintoticamente stabile:

b) Tempo di assestamento non superiore a 3 s.

c) Sovraelongazione massima $S\% = 5\%$.

Mettiamo in pratica le premesse fatte in precedenza attraverso questo Codice MatLab grazie al quale aumentiamo la frequenza del polo e diminuiamo quella dello zero iterativamente; ad ogni iterazione è fatto variare anche il guadagno in un intervallo da 1 a 20.

L'esecuzione si interrompe una volta che il regolatore sintetizzato permette al sistema di rispettare i vincoli riguardanti il tempo di assestamento, la sovravelongazione e la stabilità.



Modello Simulink del sistema retroazionato

Codice MatLab:

```
while true
    cost = cost + 0.1;
    num = [1 / ((w) * 10^(-cost) ), 1];
    den = [1 / ((w) * 10^(cost)), 1];
    R2 = tf (num, den);
    for mu = 1 : 20
        R1 = tf(mu, 1);
        R = series(R1, R2);
        L2 = series(R, G);
        F = feedback(L2, 1);
        ys = r_fin * step(F, t);
        y_inf = r_fin * ones(size(t));
        Info = lsiminfo(ys, t);
        Ta_reale = Info.SettlingTime;
        S_reale = (Info.Max - r_fin) / r_fin;
        PRESTAZIONI = allmargin(L2);
        if (Ta_reale < Ta && S_reale < S && PRESTAZIONI.PhaseMargin > 60)
            break
        end
    end
    if (Ta_reale < Ta && S_reale < S && PRESTAZIONI.PhaseMargin > 60)
        break
    end
end
```

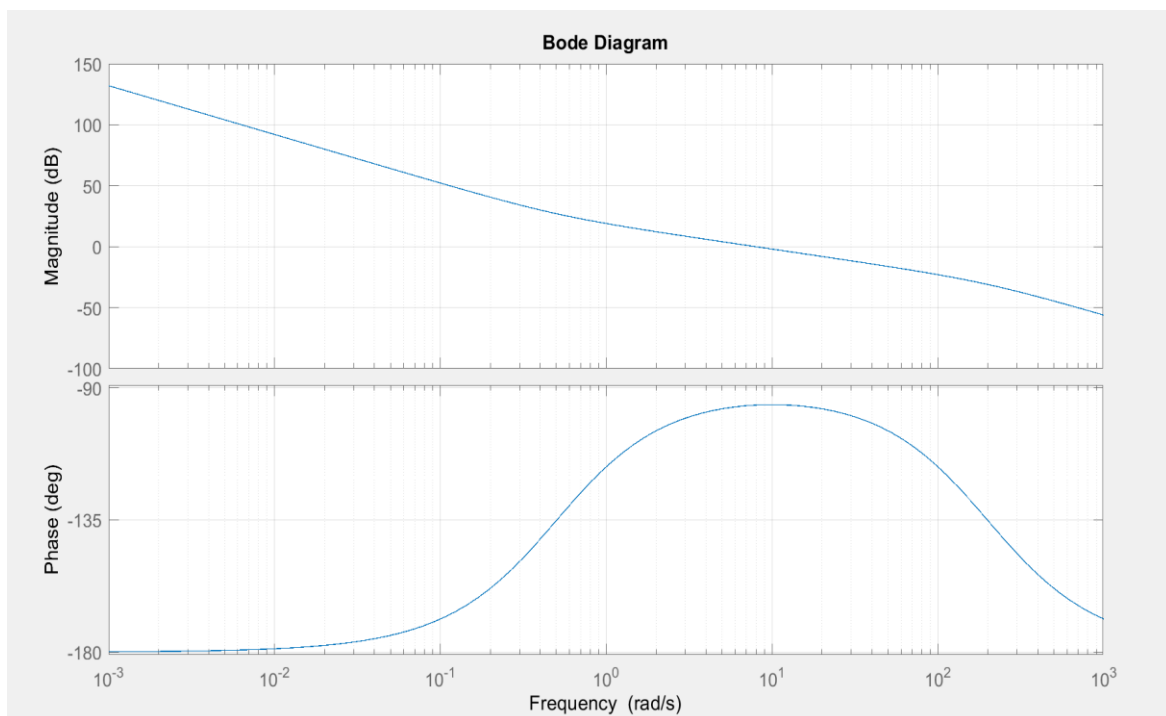


Diagramma di Bode della funzione di anello del sistema in serie al regolatore

Dal diagramma di Bode del sistema regolato dal controllore ottenuto grazie al programma possiamo notare che l'attraversamento ora avviene con una pendenza di -20 dB/dec aumentandone notevolmente il margine di fase.

Il sistema ora è stabile, rispetta quindi il requisito del punto a).

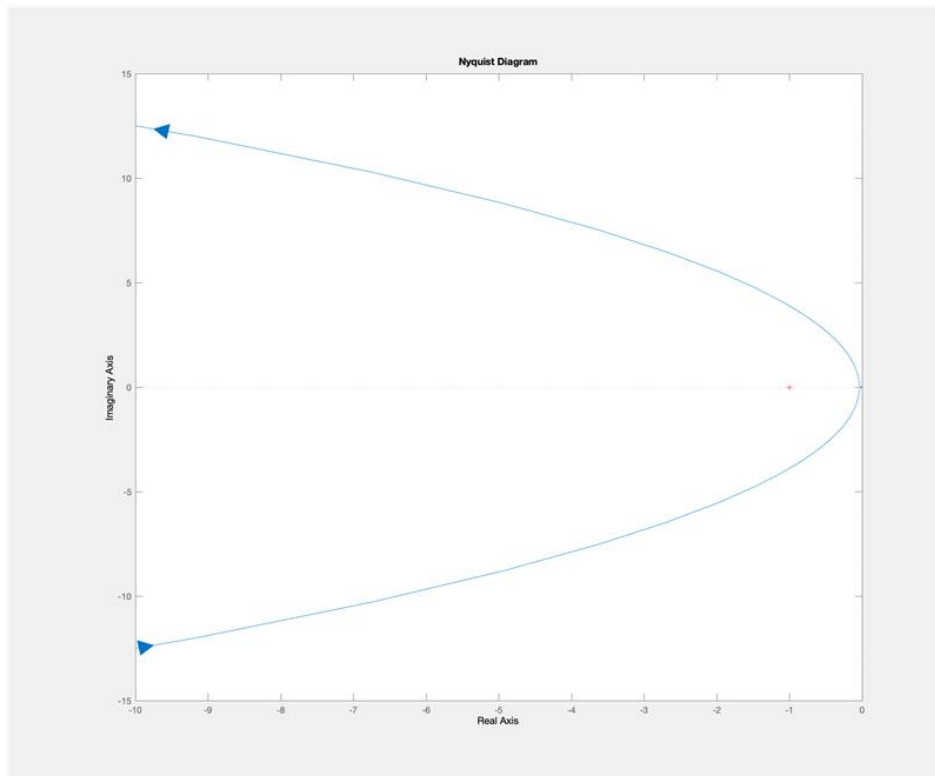
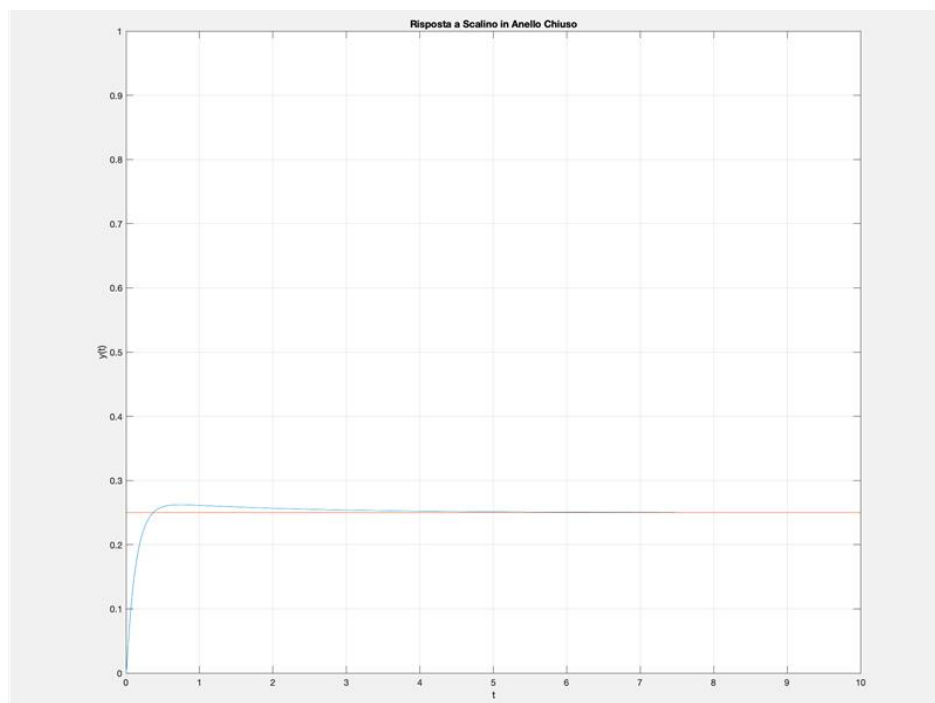


Diagramma di Nyquist della funzione d'anello

Per completezza, durante l'esecuzione viene 'plottato' anche il diagramma di Nyquist, applicandone il relativo criterio abbiamo ulteriore conferma riguardo la stabilità in anello chiuso, non essendo presente un attraversamento del diagramma nel punto $(-1, 0)$ né tantomeno Poli della funzione di Anello con $Re(p_i) > 0$



Risposta a scalino del sistema in Anello Chiuso

Il diagramma della risposta a scalino ci permette di comprendere meglio, tramite la visualizzazione grafica, come il regolatore assesti nel tempo la posizione della pallina in un intorno di quella desiderata.

Qui di seguito vengono riportati i risultati ottenuti in seguito alla progettazione del regolatore

PRESTAZIONI IN ANELLO CHIUSO:

Pulsazione di Attraversamento: 7.961 rad/s
 Margine di Fase: 84.113 °
 Tempo di Assestamento 2.497 s
 Sovraelongazione Percentuale 4.868 %

2. In un setting non ideale, la movimentazione dell'ingranaggio può provocare delle vibrazioni. Tenuto conto di ciò, mostrare in simulazione il comportamento del sistema in presenza di disturbi sinusoidali additivi sull'ingresso $u(t)$, dimensionandone ampiezza e pulsazione in modo che il sistema in anello chiuso possa o non possa attenuarli, e discutere i risultati ottenuti.

Come fatto per la scelta della posizione di assestamento della pallina abbiamo inserito una finestra di dialogo con l'utente che chiede di inserire un valore della pulsazione per il disturbo sinusoidale in ingresso.

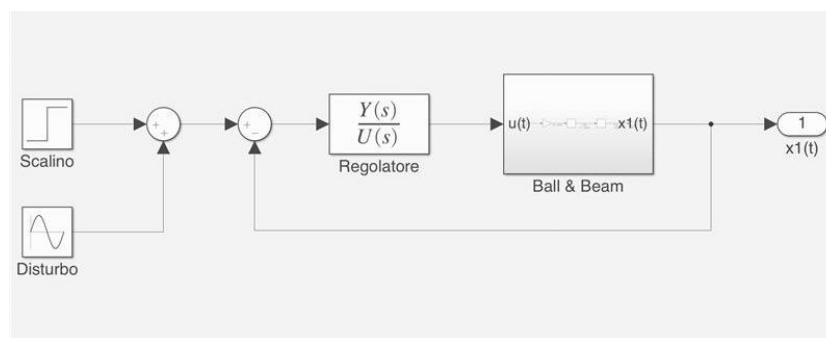
```
%scelta pulsazione da parte dell'utente
PROMPT2={'Inserire il valore della pulsazione del disturbo sinusoidale in
ingresso:'};
NAME2='PULSAZIONE DISTURBO';
wd_cell=inputdlg(PROMPT2, NAME2);
wd=str2double(wd_cell{1, 1});
```

Viene successivamente calcolata l'ampiezza massima di tale segnale affinché il sistema in anello chiuso rimanga asintoticamente stabile.

```
%scelta ampiezza massima per stabilità tramite modello simulink
amp=0.1;
while true
    amp=amp+0.1;

SimOut2=sim('ball_beam_retroazionato_simulink.slx','ReturnWorkspaceOutputs','on'
); %ingresso con disturbo
ys2=SimOut2.yout;
if max(ys2)>1 || min(ys2)<0
    amp=amp-0.1;

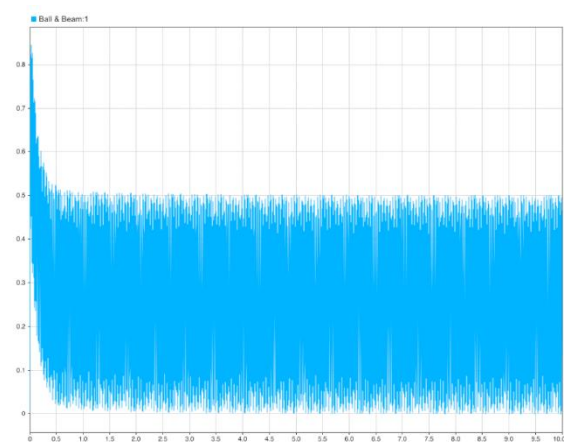
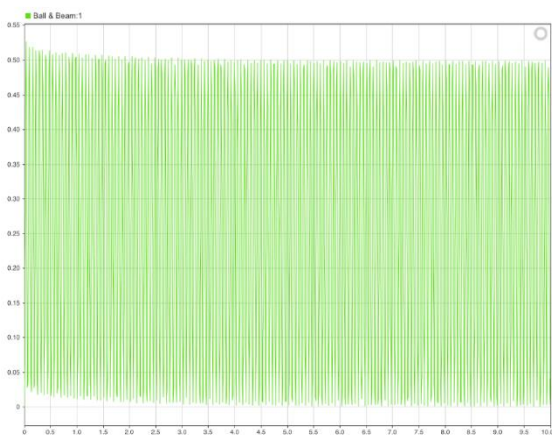
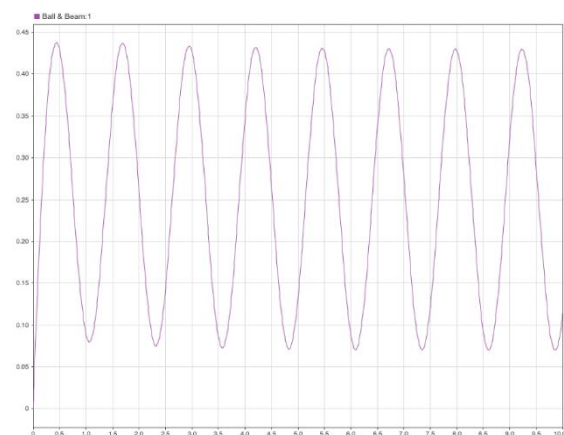
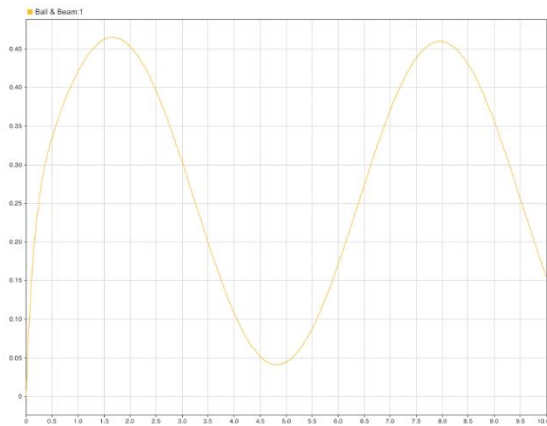
SimOut2=sim('ball_beam_retroazionato_simulink.slx','ReturnWorkspaceOutputs','on'
);
    break
end
end
```



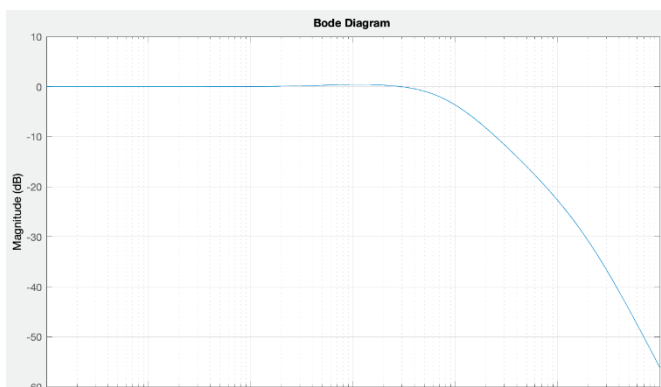
Modello Simulink retroazionato con disturbo sinusoidale in ingresso

Facciamo ora vedere quattro diversi grafici aventi frequenza di disturbo sinusoidale differenti

Pulsazione disturbo (ω_d)	Ampiezza massima (A_{max})
[rad/s]	[m]
1	0.2
5	0.2
100	3.4
500	42.2



Come possiamo notare dai grafici riportati, all'aumentare della frequenza del segnale sinusoidale di disturbo in ingresso, l'ampiezza massima che il segnale può avere la sinusoide aumenta.



Quanto detto sopra è avvalorato dal diagramma di Bode della funzione di sensibilità complementare $F(s)$ in quanto possiamo vederla agire come un filtro passa basso di conseguenza è spiegato il comportamento osservato tramite l'aiuto dei grafici sopra riportati.