Compte rendu de TP
Commande articulaire d'un robot manipulateur de type RP

Aurélien Bernier Levalois, Fleytoux Yoann

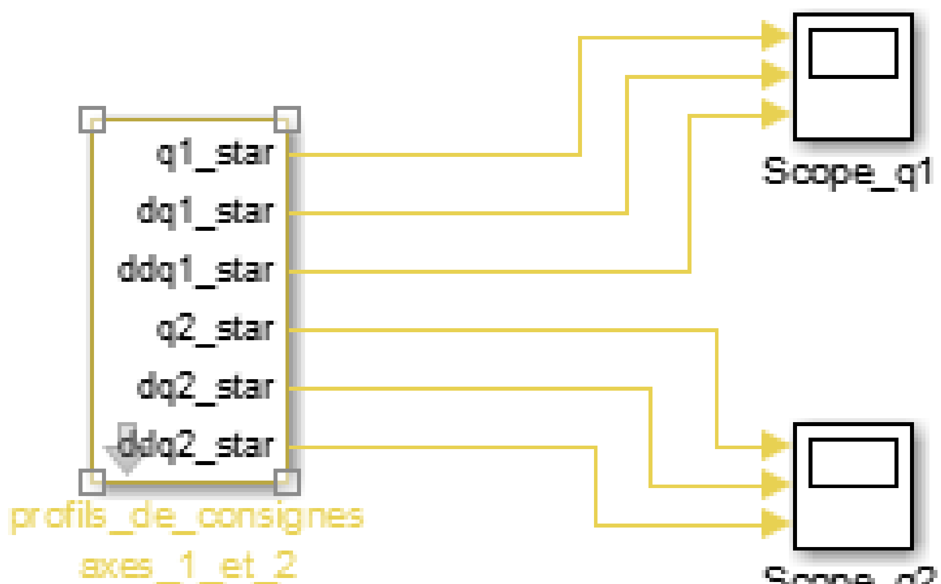
21 novembre 2016

Table des matières

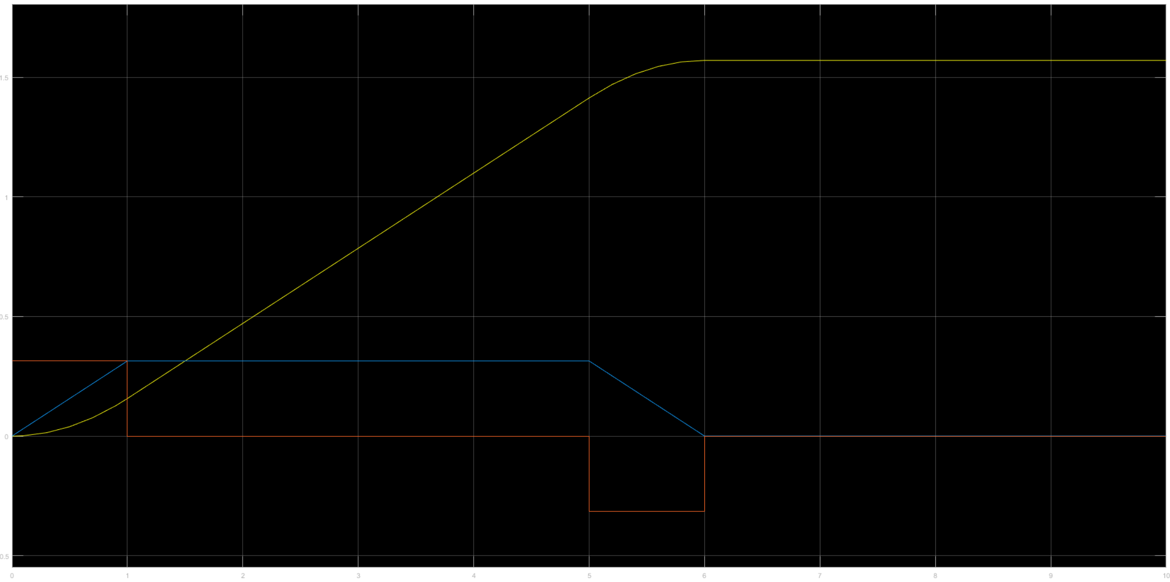
1 Travail demandé

1.1 Prise en main de l'outil de simulation et calculs préliminaires

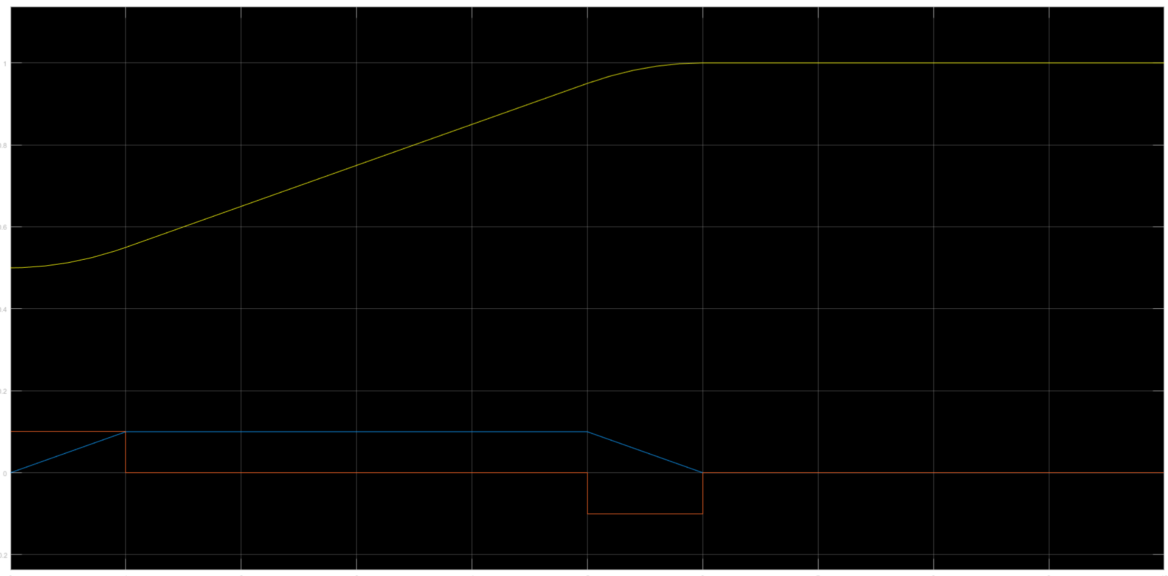
1.Relever l'évolution temporelle des profils de consigne
Simulink



$Scope_q1$: on peut observer le profil de position, vitesse et accélération de q1



$Scope_q2$: on peut observer le profil de position, vitesse et accélération de q2



2.

```

m=15

Km_R = 0.3;

Beff = 1/80;

Jm = 1/100;

R = [1/200 1/30 1];

Jeff200 = Jm+(R(1)^2)*m;
Jeff30 = Jm+R(2)^2*m;
Jeff1 = Jm+R(3)^2*m;
```

1.2 Commande en vitesse de type PD

3. On cherche les coefficients K et KD de la loi de commande Proportionnelle Dérivée, pour $di(t) = 0$, confère à la boucle fermée un amortissement unité $\zeta = 1$ et une erreur de vitesse $/epsilon_1 i$ donnée en réponse à une consigne rampe $\theta_{mi}^{*}(t) = \theta_{mi}^{1} t * U(t)$

avec $wn = 4$

$K = (wn^2) * Jeff/(Km/R) = 16 * Jeff/(Km/R)$

$KD = ((K/2) - (Beff/(Km/R))) = (8 * Jeff - Beff)/(Km/r)$

4.

a)

```
Ti = 0.8
zeta=1
wn=4

K200 =(wn^2)*Jeff200/Km_R
K30 =(wn^2)*Jeff30/Km_R
K1 =(wn^2)*Jeff1/Km_R

KD200 = ((K200/2)-(Beff/Km_R))
KD30 = ((K30/2)-(Beff/Km_R))
KD1 = ((K1/2)-(Beff/Km_R))
```

Avec r=1/200 Jeff= 1/100+15/40000=0.010375 ; on trouve K=0.5533 et KD=0.2350

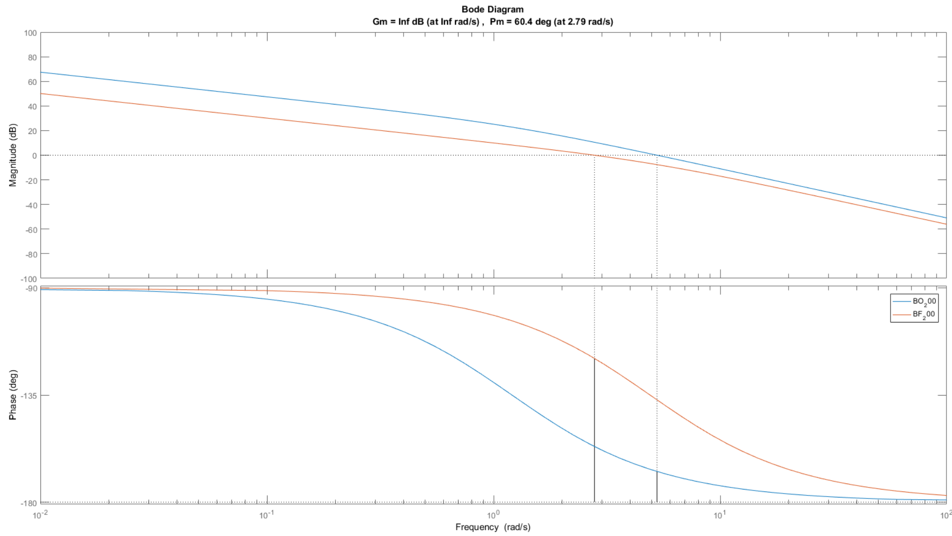
Avec r=1/30 Jeff= 2/75=0.02666666666 ; on trouve K=1.4222 et KD=0.6694

Avec r=1 Jeff= 1/100+15=15.01 ; on trouve K=800.5333 et KD=400.2250

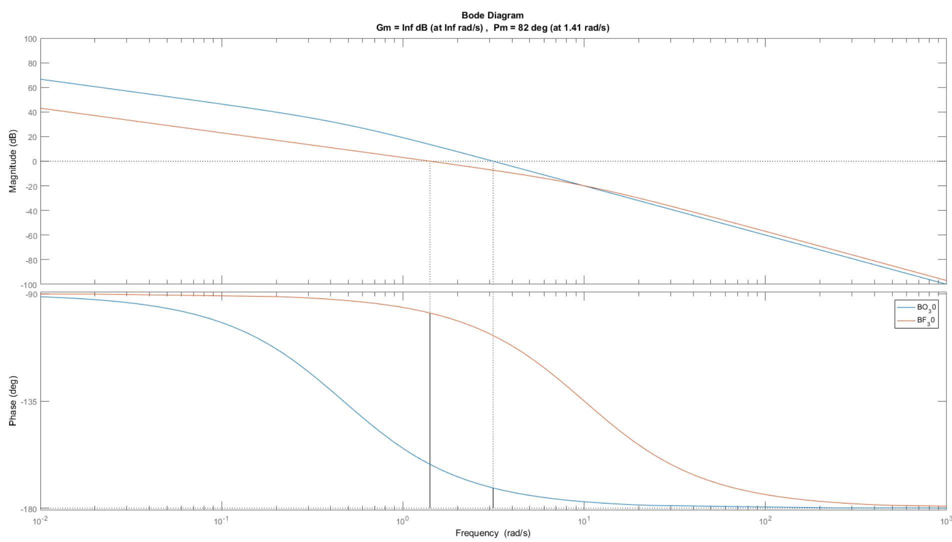
b) lieux de transfert de la boucle ouverte avant correction :

```
G_200=tf(1,[Jeff200 Beff 0])
G_30=tf(1,[Jeff30 Beff 0])
G_1=tf(1,[Jeff1 Beff 0])

(on suppose que d=1)

BO_200=series((Km_R-R(1)),G_200)
BF_200=feedback(K200*BO_200,tf([KD200 0],1))

BO_30=series((Km_R-R(2)),G_30)
BF_30=feedback(K30*BO_30,tf([KD30 0],1))

BO_1=series((Km_R-R(3)),G_1)
BF_1=feedback(K1*BO_1,tf([KD1 0],1))

margin(BO_200);
title('Marge de phase et marge de gain');
hold on;
margin(BF_200);
legend('BO_200','BF_200');
hold off;

figure
margin(BO_30);
title('Marge de phase et marge de gain');
hold on;
margin(BF_30);
legend('BO_30','BF_30');

figure
margin(BO_1);
title('Marge de phase et marge de gain');
hold on;
margin(BF_1);
legend('BO_1','BF_1');
```

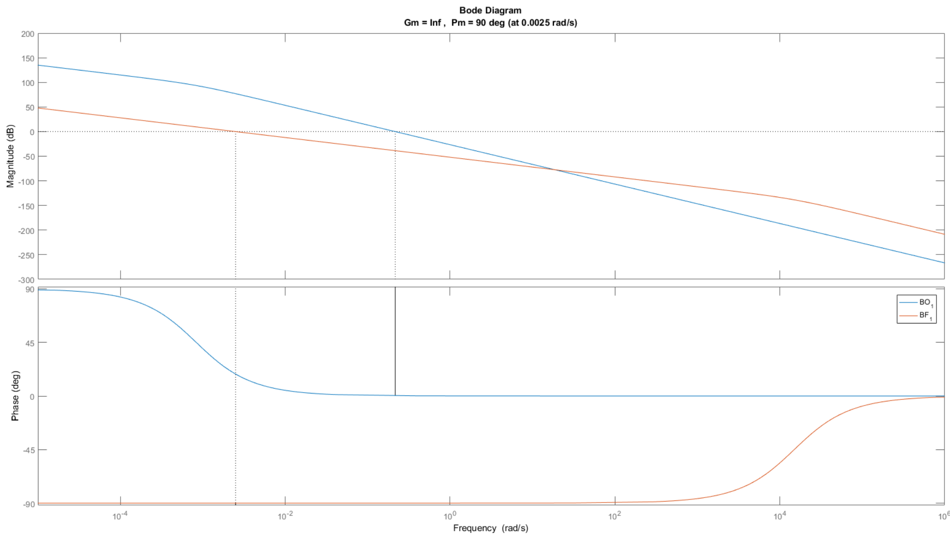
$r = 1/200$


Bode Diagram
Gm = Inf dB (at Inf rad/s) , Pm = 60.4 deg (at 2.79 rad/s)
Magnitude (dB)
100
80
60
40
20
0
-20
-40
-60
-80
-100
Phase (deg)
-90
-135
-180
BO$_2$00
BF$_2$00
10^{-2}
10^{-1}
10^{0}
10^{1}
10^{2}
Frequency (rad/s)

$r = 1/30$:


Bode Diagram
Gm = Inf dB (at Inf rad/s) , Pm = 82 deg (at 1.41 rad/s)
Magnitude (dB)
100
80
60
40
20
0
-20
-40
-60
-80
-100
Phase (deg)
-90
-135
-180
BO$_3$0
BF$_3$0
10^{-2}
10^{-1}
10^{0}
10^{1}
10^{2}
10^{3}
Frequency (rad/s)

$r = 1$



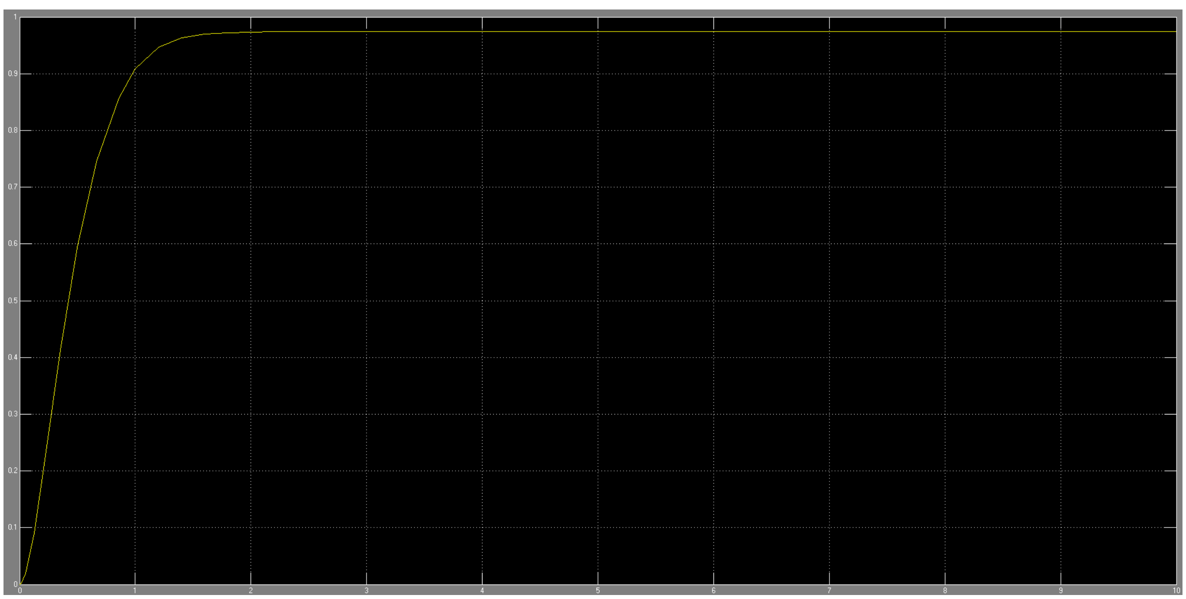
c) Réponses temporelles en fonction des rapports de réduction et des perturbations r=200 sans perturbations

r=30 sans perturbations

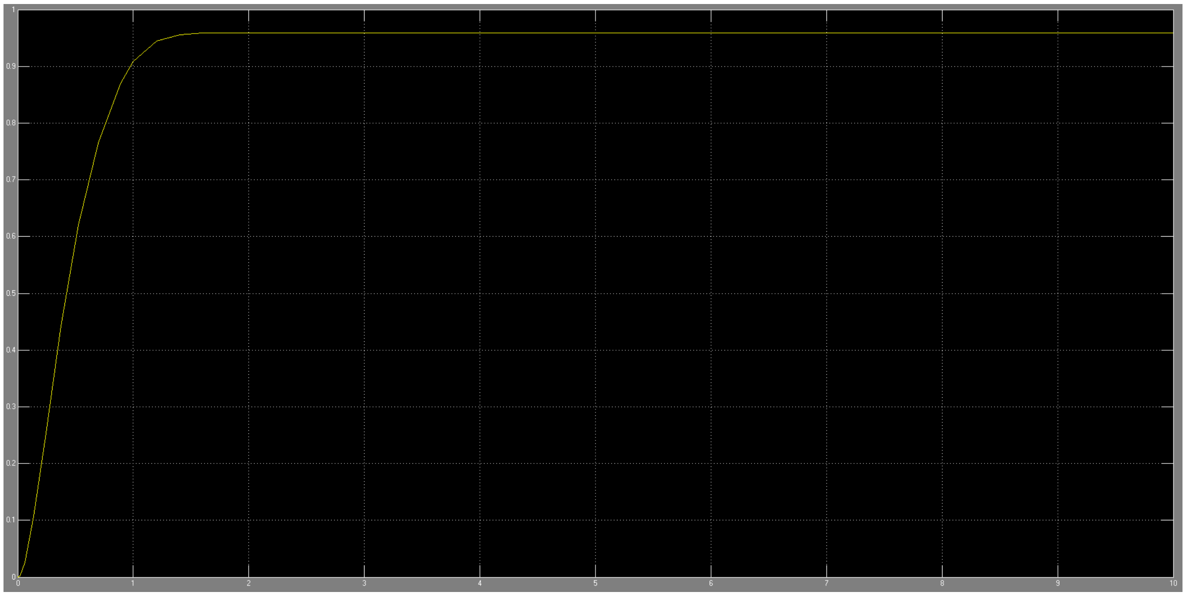
r=1 sans perturbations

r=200 avec rdi=10

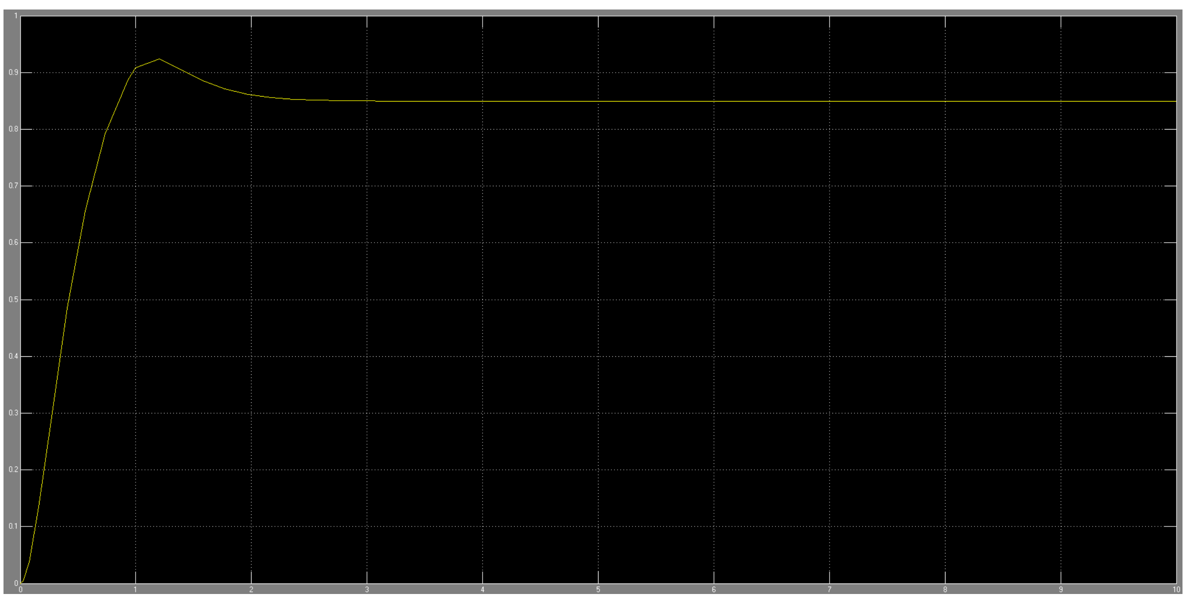
r=30 avec rdi=10



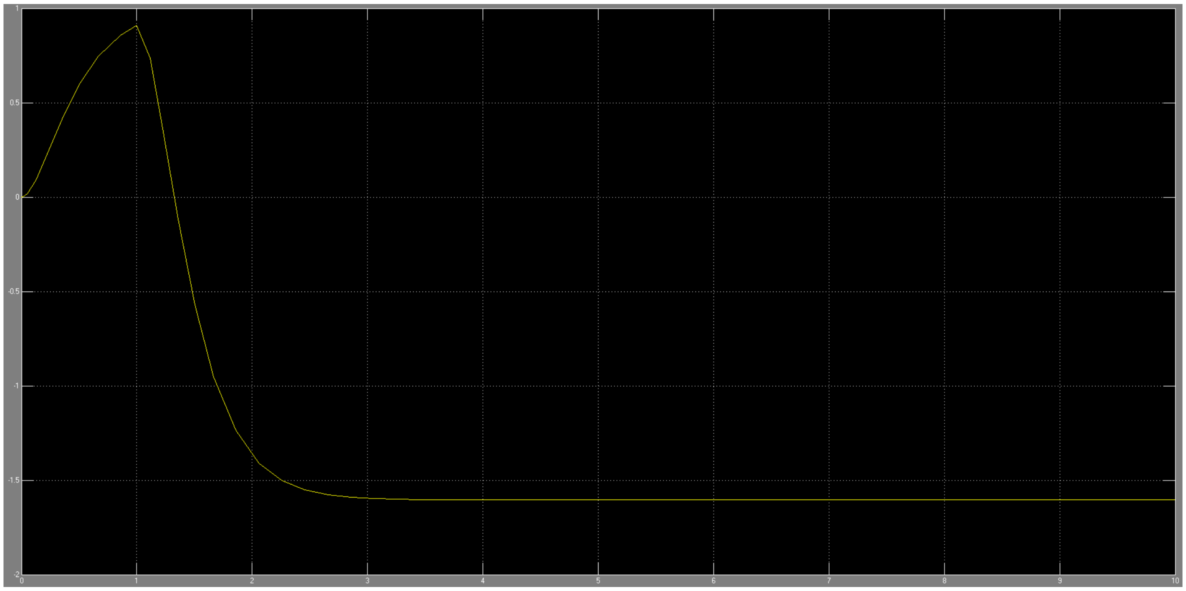
r=1 avec rdi=10



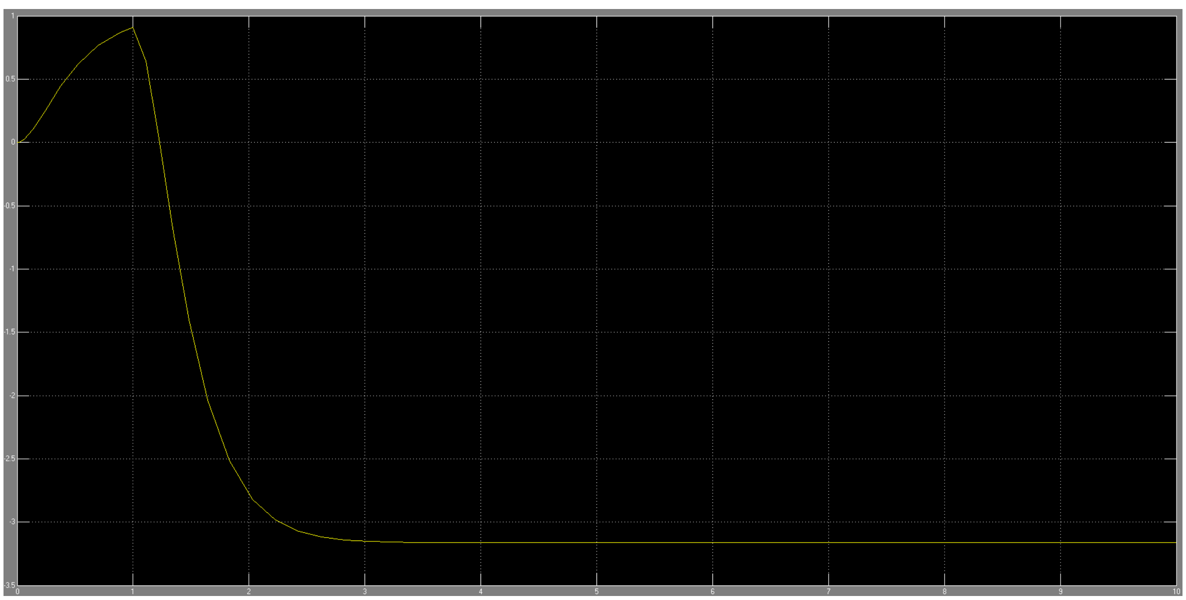
r=200 avec rdi=1000



r=30 avec rdi=1000

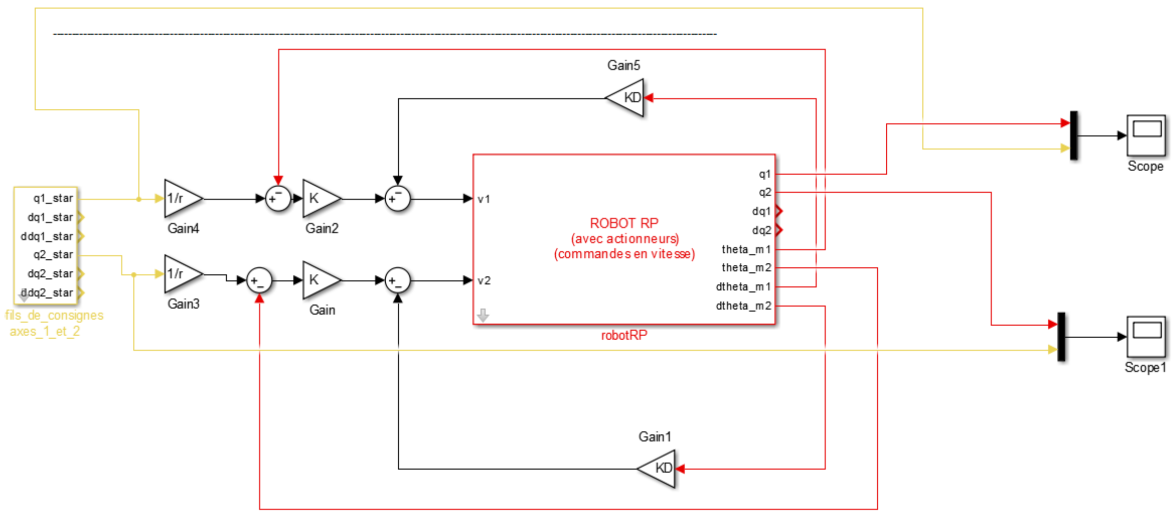


r=1 avec rdi=1000

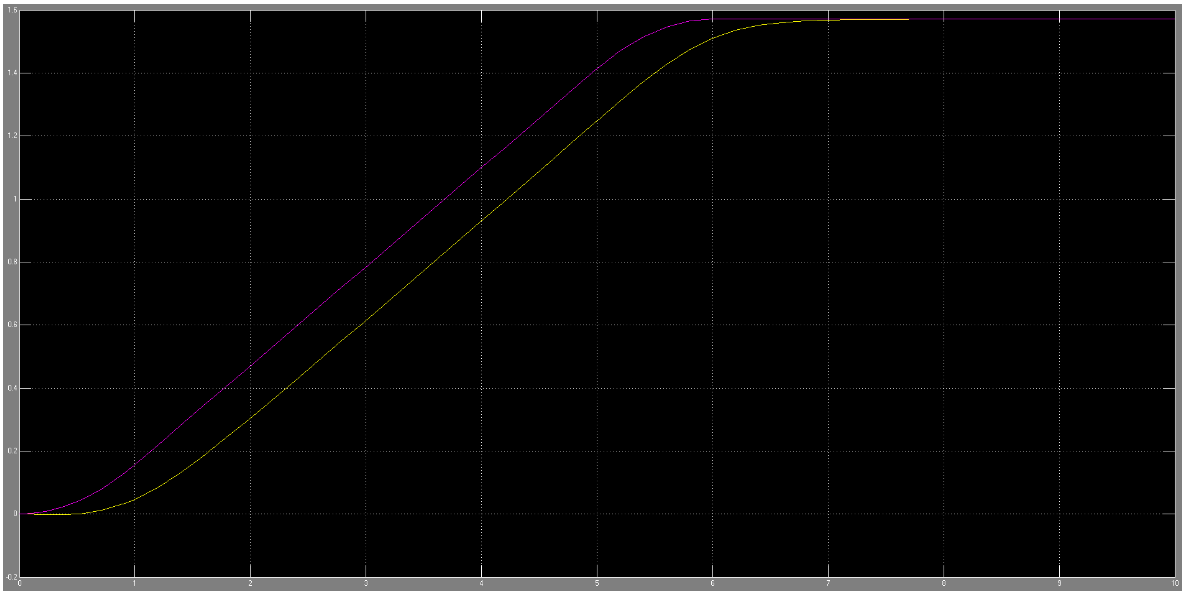


Pour conclure, on voit que l'erreur est beaucoup plus facilement compensée par un rapport de réduction petit.

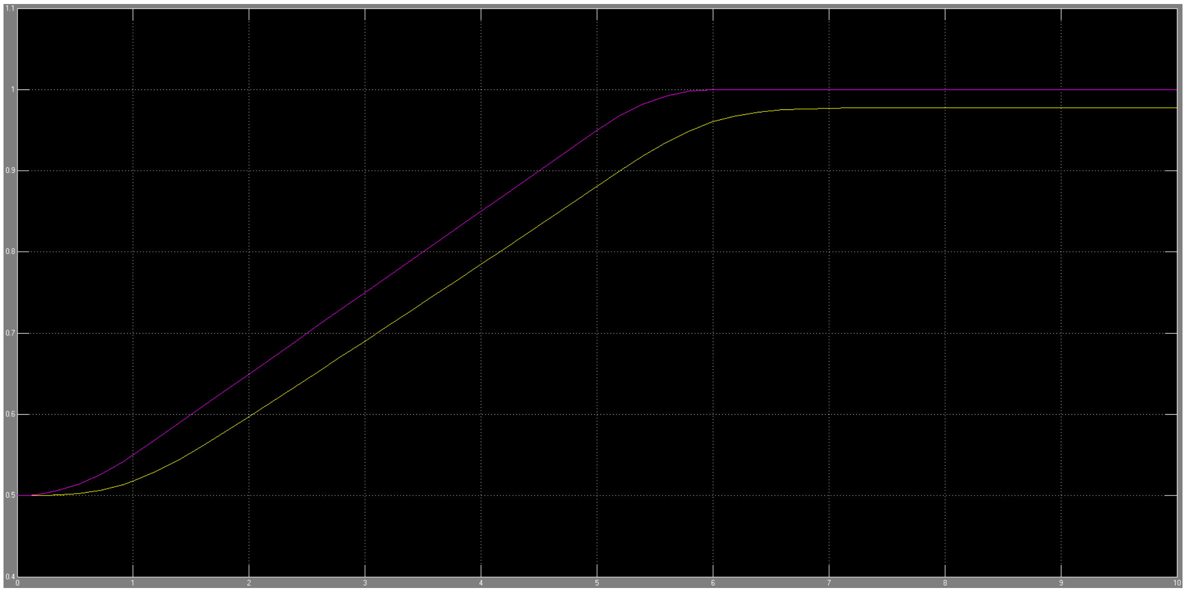
d) Comparaison avec modèle non-linéaire
Schéma simulink avec bloc robot RP



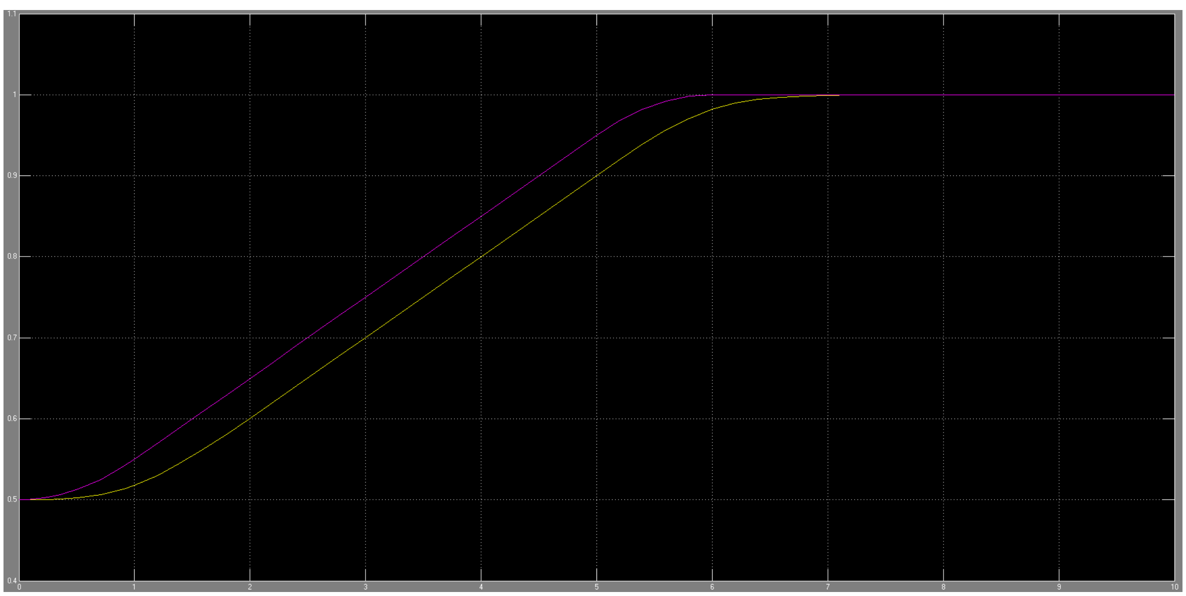
Sortie q1 comparée à l'entrée avec perturbations



Sortie q2 comparée à l'entrée avec perturbations (présence d'erreur)



Sortie q2 comparée à l'entrée sans perturbations



On voit que l'erreur due à la gravitation ne touche que q2.

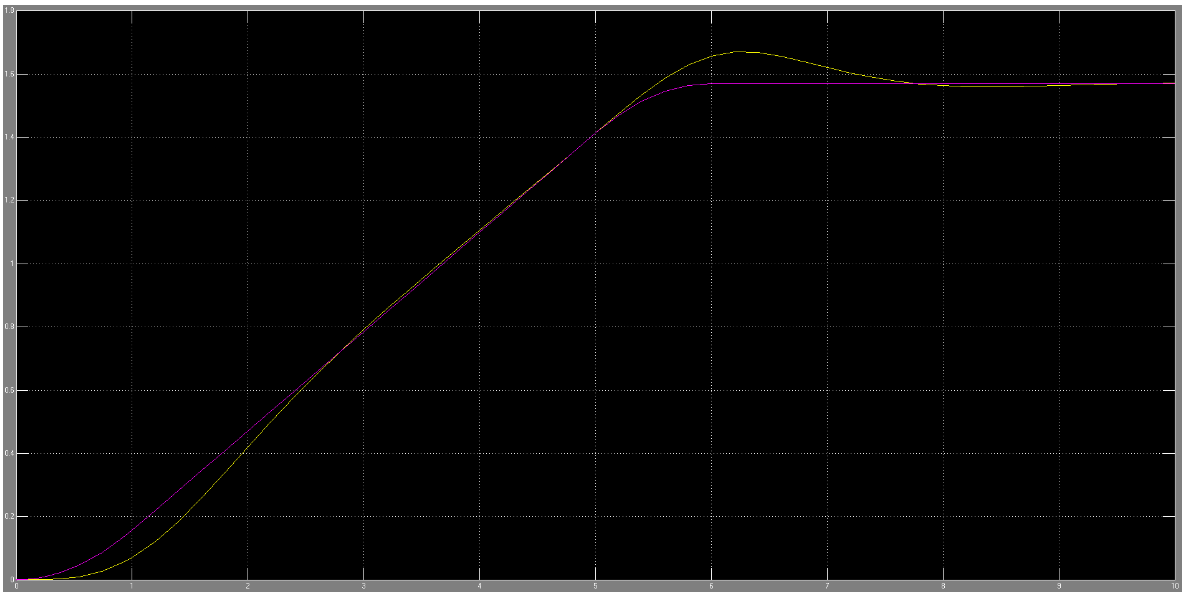
5.

On voit donc que q2 est affecté par la gravitation mais pas q1. Notre modèle linéaire ne fonctionne pas avec des rapports de réductions trop grands. Le modèle non linéaire se stabilise moins vite.

1.3 Commande en vitesse de type PD

6.

On choisit Ti = 0.8 car la marge de phase du diagramme de Bode est de 45.1 degrés. On obtient cette réponse :



1.4 Retour sur la modélisation

7.

8.

```
q2min=0.5;
q2max=1;
Dmin=[m*q2min^2 0; 0 m]
Dmax=[m*q2max^2 0; 0 m]

Jeff1min=Jm+R(1)*Dmin(1,1)
Jeff1max=Jm+R(1)*Dmax(1,1)
Jeff2min=Jm+R(1)*Dmin(2,2)
Jeff2max=Jm+R(1)*Dmax(2,2)
```

On choisit les valeurs Jeff1min et Jeff2min, car ils représentent des systèmes avec moins d'inertie.

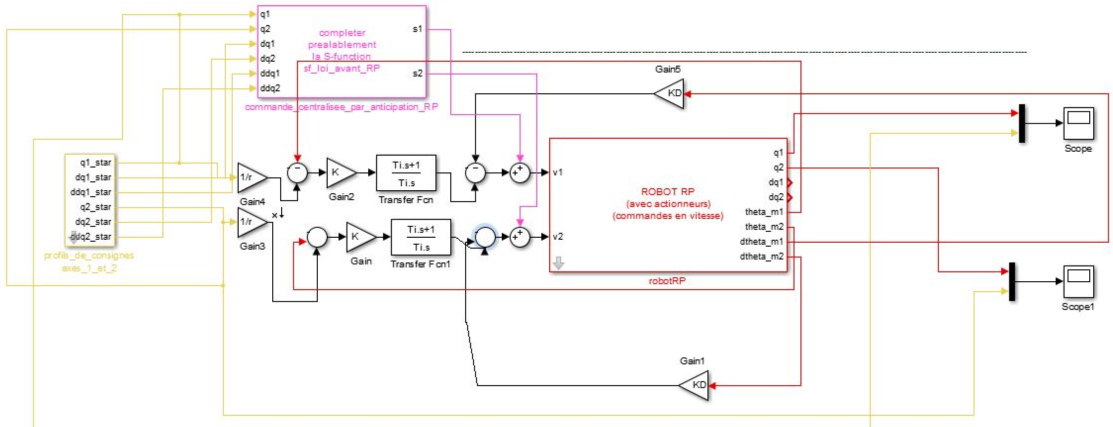
```
Jeff1min =

    0.0287


Jeff1max =

    0.0850


Jeff2min =

    0.0850


Jeff2max =

    0.0850
```

1.5 Commande non linéaire centralisée par anticipation

Code sf_loi_avant_rp.m

```
function sys = mdlOutputs(t,x,u)
%
q1 = u(1); q2 = u(2); dq1 = u(3); dq2 = u(4); ddq1 = u(5);
   ddq2 = u(6);
%
%
%

gy = -9.81;
d1 = 2*m*q2*dq1*dq2-m*q2*gy*cos(q1);
d2 = -m*q2*dq1*dq1-m*gy*sin(q1);
s1 = ddq1/r * Jeff/Km_R + dq1/r * Beff/Km_R + r*d1/Km_R;
s2 = ddq2/r * Jeff/Km_R + dq2/r * Beff/Km_R + r*d2/Km_R;
sys = [s1;s2];
```

Soit le schéma Simulink suivant :



1.6 Commande non linéaire centralisée par découplage

Code sf_modele_dyn_inverse.m

```
function sys = mdlOutputs(t,x,u)
%
ddq1_star = u(1); ddq2_star = u(2);
q1 = u(3); q2 = u(4); dq1 = u(5); dq2 = u(6);
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%

gy = -9.81;
d1 = 2*m*q2*dq1*dq2-m*q2*gy*cos(q1);
d2 = -m*q2*dq1*dq1-m*gy*sin(q1);
s1 = ddq1_star/r * Jeff/Km_R + dq1/r * Beff/Km_R + r*d1/
   Km_R;
s2 = ddq2_star/r * Jeff/Km_R + dq2/r * Beff/Km_R + r*d2/
   Km_R;
sys = [s1;s2];
```

Soit le schéma Simulink suivant (incomplet) :