ADVANCED AEROSPACE CONTROL

Exam project AA 2019/2020

A.Occhipinti, S.Piacquadio, L.Quadri

July, 27 2020

Meet our team

Anna Occhipinti

944789

Salvatore Piacquadio

953108

Lorenzo Quadri

944515

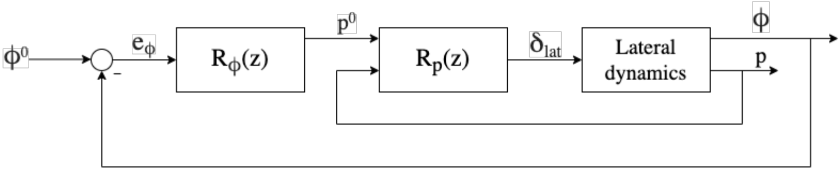
Abstract

The aim of the project is to design the control system for the lateral dynamics of an ANT-R quadrotor starting from grey-box model. After a brief plant analysis, the nominal design is carried out and robustness of the obtained controller is verified both in terms of stability and performance.

Table of contents

Presentation of the model

Scheme of the model



$$\dot{x} = Ax + Bu \qquad u = \delta_{lat}; \quad x = \begin{bmatrix} v & p & \phi \end{bmatrix}'$$

$$y = Cx + Du \qquad y = \begin{bmatrix} p & \phi \end{bmatrix}'$$

$$A = \begin{bmatrix} Y_v & Y_p & g \\ L_v & L_p & 0 \\ 0 & 1 & 0 \end{bmatrix}; \; B = \begin{bmatrix} Y_\delta \\ L_\delta \\ 0 \end{bmatrix}; \; C = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}; \; D = \begin{bmatrix} 0 & 0 \end{bmatrix}$$

where:

Stability derivatives

$Y_v = -0.264 s^{-1} (4.837\%)$

$Y_p = 0 m/s$

$L_p = 0 s^{-1}$

$L_v = -7.349 rad \cdot s/m (4.927\%)$

Control derivatives

$Y_d = 9.568 m/s^2 (4.674\%)$

$L_d = 1079.339 rad/s^2 (2.762\%)$

The controllers are digitally implemented and the sampling time is set at $T_s = 0.004s$.

$$R_p(z): \quad \begin{aligned} x_p(k+1) = &\quad A_p x_p(k) + B_p u_p(k) \\ \delta_{lat}(k) = &\quad C_p x_p(k) + D_p u_p(k) \end{aligned}$$

where:

$$u_p = \begin{bmatrix} p^o \\ p \end{bmatrix}$$

$$A_p = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \; B_p = \begin{bmatrix} b & -b \\ 0 & 0.5 \end{bmatrix} \; C_p = \begin{bmatrix} c_1 & c_2 \end{bmatrix} \; D_p = \begin{bmatrix} d_1 & d_2 \end{bmatrix}$$

Outer loop controller

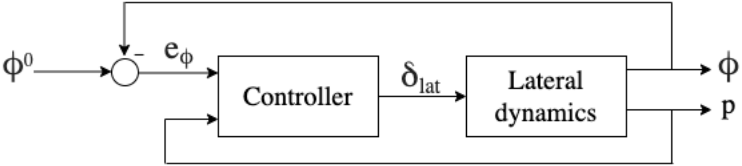
$R_\phi(z)$: $\quad p^o(k) = \quad D_\phi e_\phi(k)$

where :

$e_\phi(k) = \phi^o(k) - \phi(k)$

Scheme of the re-defined model

Controllers are merged into one:



Re-definition of the controller

The state space matrices are:

$$A_R = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$$

$$B_R = \begin{bmatrix} b & -b \cdot D_\phi \\ 0 & 0.5 \end{bmatrix}$$

$$C_R = \begin{bmatrix} c_1 & c_2 \end{bmatrix}$$

$$D_R = \begin{bmatrix} d_1 & d_2 \cdot D_\phi \end{bmatrix}$$

Matlab implementation(1)

Lateral dynamics

- Definition of uncertain parameters, for example:

```
Y_v = ureal('Y_v',-0.264,'Percentage',3*4.837);
```

- Build of matrices `A,B,C,D`
- Definition of the uncertain plant:

```
G = uss(A,B,C,D);
G.InputName = 'Delta_lat';
G.OutputName = {'p';'phi'};
```

Matlab implementation(2)

Controller

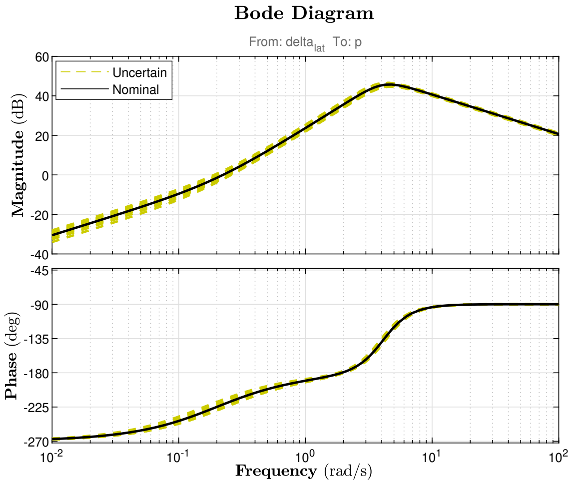
- Definition tunable parameters, for example:

```
b = realp('b',1);
```

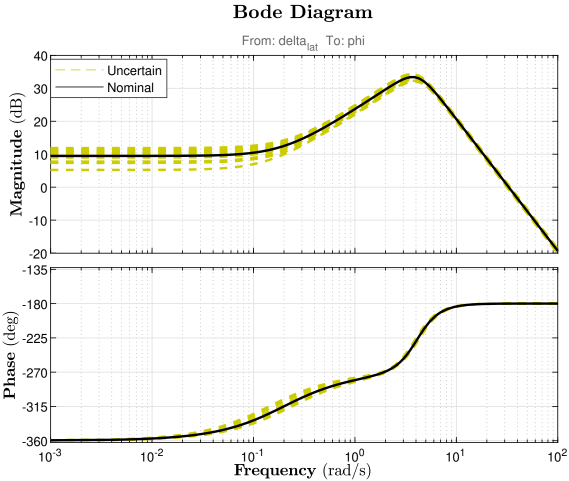
- Build of matrices of the controller A_R,B_R,C_R,D_R
- Definition of the controller:

```
R0 = ss(A_R,B_R,C_R,D_R,Ts);
R0.InputName = {'e_phi';'p'};
R0.OutputName = 'Delta_lat';
```

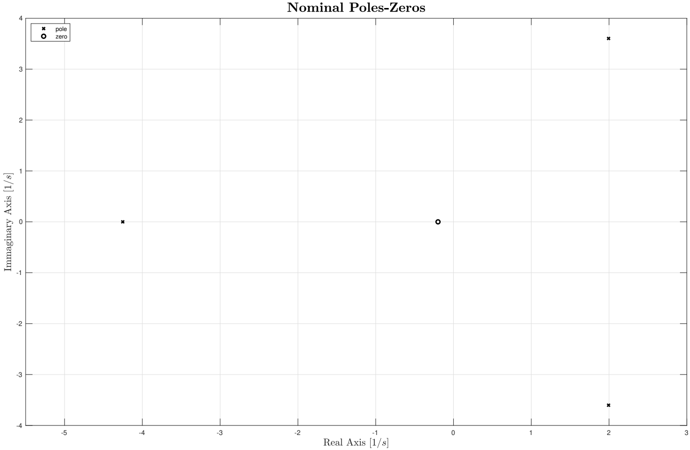
Bode plots of G(1)



Bode plots of G(2)

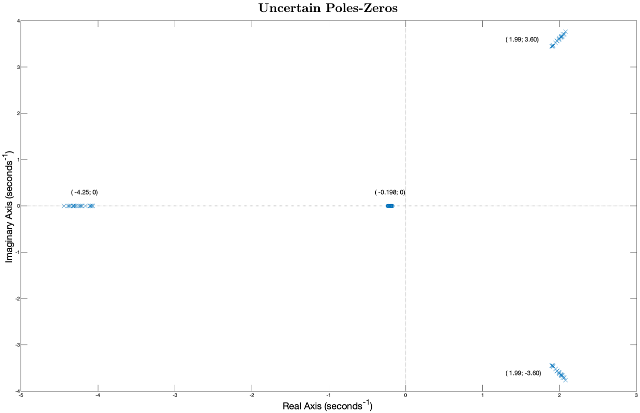


Nominal poles and zeros of G



Uncertain poles and zeros of G


```
pzplot(G)
```



Nominal design - H_∞

Tuning with H_∞

The tuning of the controller is carried out using H_∞ and the nominal values of the plant.

Weight functions

The performance requirement is of this type

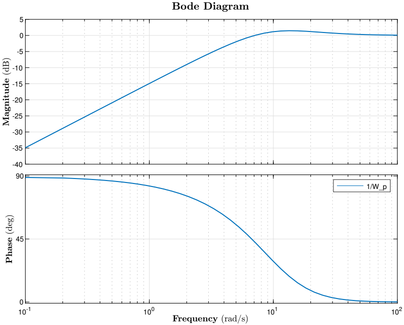
$$\frac{1}{W_p(s)} = \frac{s^2 + 2\xi\omega_b s}{s^2 + 2\xi\omega_b s + \omega_b^2}$$

where

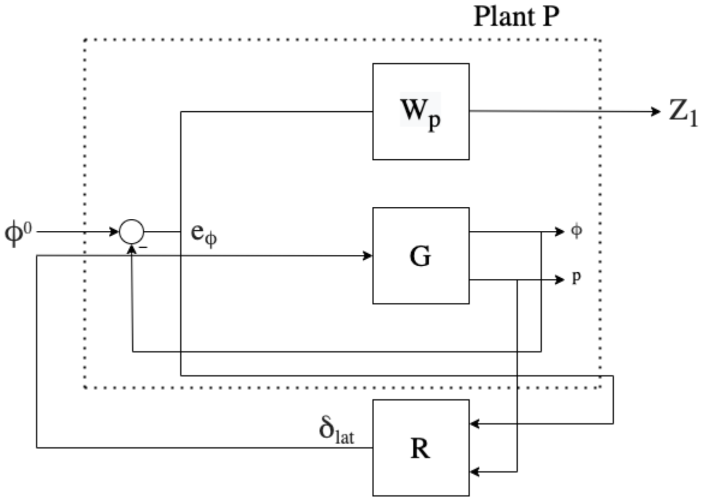
- ξ : minimum damping ratio.
- ω_b: lower bound of crossover frequency.

Weight functions - choice of parameters

- ξ : 0.9
- ω_b: 10 [rad/s]



Definition of the augmented plant



Definition of the augmented plant

The summing junction can be created with

```
S = sumblk('%e_phi=phi0-%phi',R0.InputName(1), ...
G.OutputName(2));
```

W_p is defined:

```
F_w = omega_w^2/(s^2+2*csi_w*omega_w*s+omega_w^2);
S_w = 1-F_w;
Wp = c2d(1/S_w,Ts);
Wp.InputName = 'e_phi';
Wp.OutputName ='z1';
```

The augmented plant is finally obtained through

```
P = connect(S,G,Wp,{'phi0','Delta_{lat}'}, ...
{'z1';'e_phi';'p'});
P.InputGroup.U1 =[1];
P.InputGroup.U2 =[2];
P.OutputGroup.Y1 = [1];
P.OutputGroup.Y2 = [2,3];
```

Application of H_∞

Applying the command

```
opt = hinfstructOptions('Display','final', ...
'RandomStart',5);
R = hinfstruct(P,R0,opt);
```

it's possible to find the structured controller which minimizes the Hinf norm of the closed-loop transfer function.

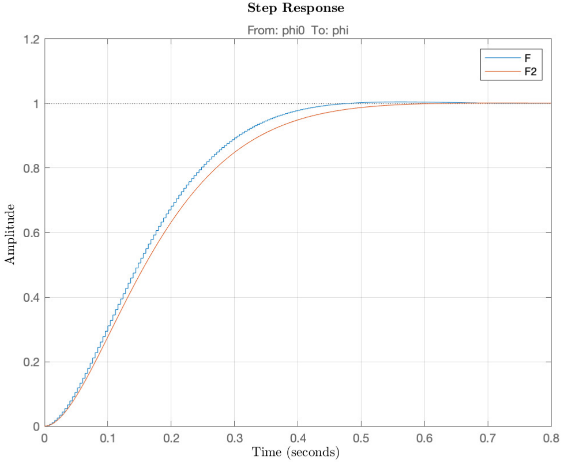
Typical output in command window:

```
Final: Peak gain = 1, Iterations = 60
```

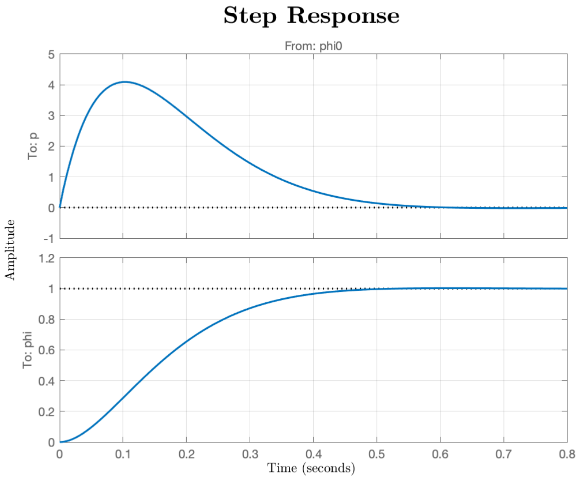
Results


```
D_phi = 0.1705;
b = 1.6208;
c1 = 0.08620;
c2 = 0.1032;
d1 = 0.1947;
d2 = -2.4833;
```

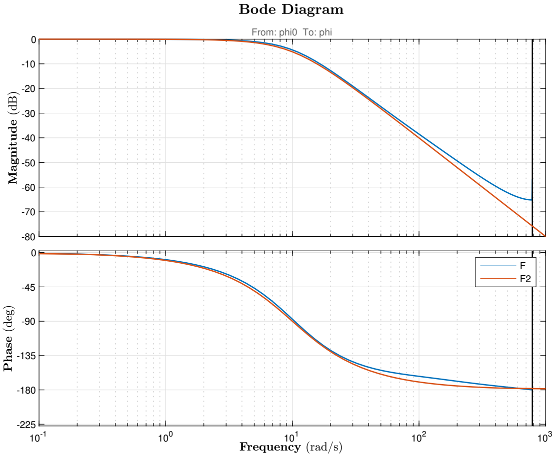
Response to step function (1)



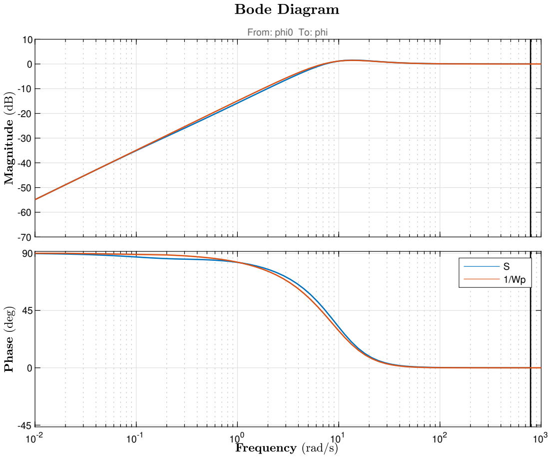
Response to step function (2)



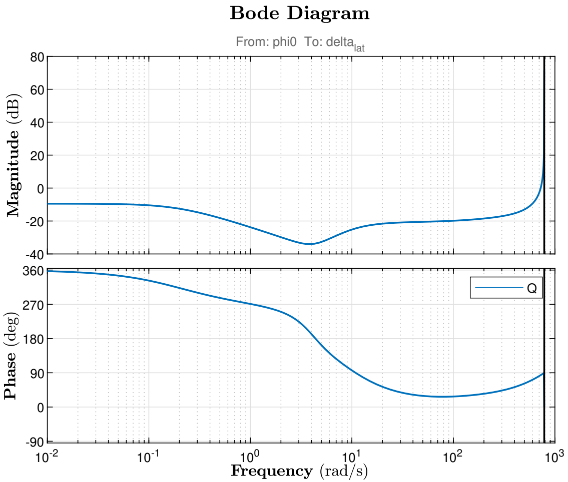
Comparison between F and F2



Comparison between S and Wp



Control sensitivity function



Nominal design - Systune

Application of Systune


```
SoftReqs = [req1];
HardReqs = [];
opt = systuneOptions('Display','final', ...
'RandomStart',5);
[F,fSoft,gHard,info] = ...
systune(F0,SoftReqs,HardReqs,opt);
```

Tuning goal

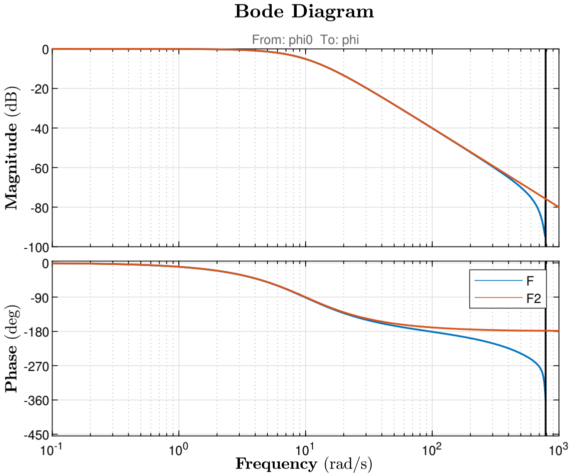

```
req1 = ...
TuningGoal.Transient(F0.InputName,F0.OutputName,F2);
```

$$F2 = \frac{\omega_b^2}{(s^2 + 2\xi\omega_b s + \omega_b^2)};$$

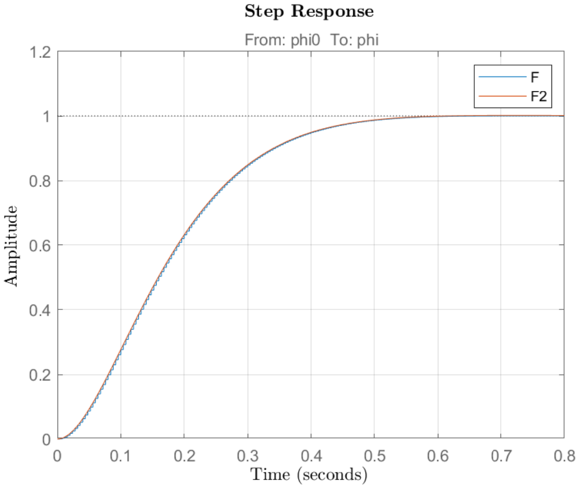
Where:

$\omega_b = 10[rad/s];$
$\xi = 0.9;$

Presentation of results (1)



Presentation of results (2)



Selection of method

H_∞ vs Systune

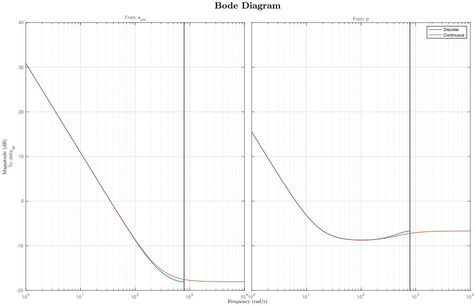
Robust stability analysis

A.Occhipinti, S.Piacquadio, L.Quadri

Conversion of R

From discrete to continuous time using Tustin method.

```
R = d2c(R,'tustin');
```



The maximum considered frequency: $\omega \approx 10^3$ [rad/s]

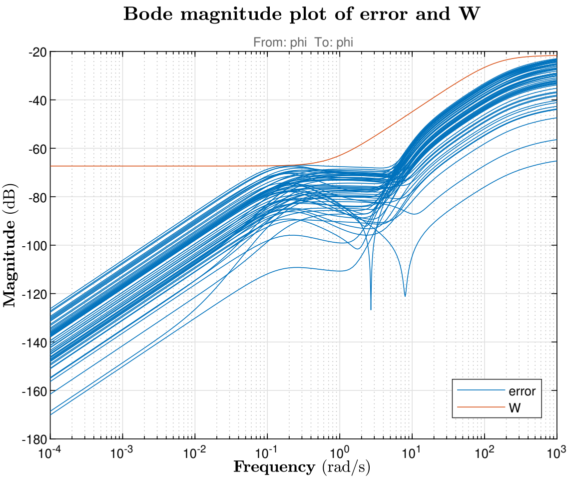
Complementary sensitivity function


```
S1 = sumblk('%e_phi = phi0-%phi', R.InputName(1), ...
G.OutputName(2));

F_nominal = connect(G.NominalValue,R,S1, ...
S1.InputName(1),{'p';'phi'});

F_uncertain = ...
connect(G,R,S1,S1.InputName(1),{'p';'phi'});
```

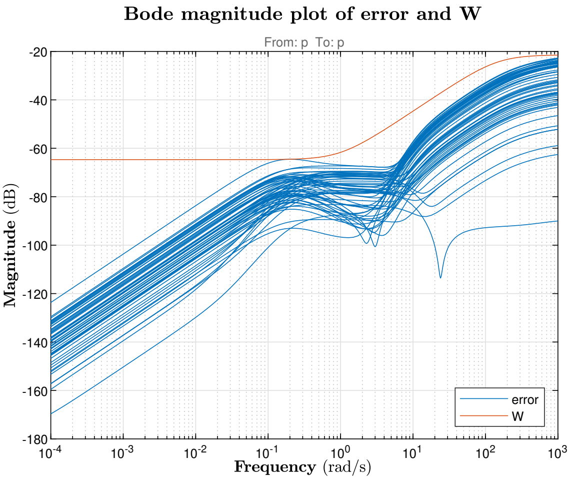
Creation of uncertainty function W (2)


```
F_array = usample(F_uncertain,60);
error = (F_nominal-F_array)/F_nominal;
[F,Info] = ucover(F_array,F_nominal,1);
W = Info.W1;
```

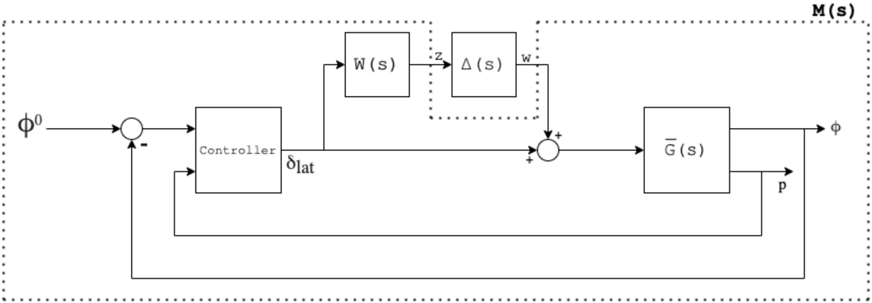
Presentation of results (1)



Presentation of results (2)



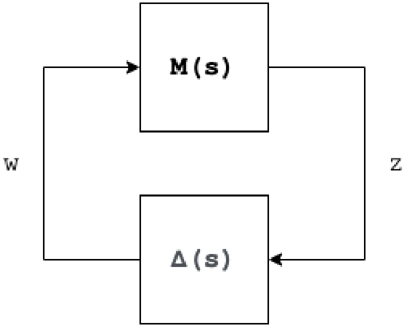
Model in M-Δ form



Creation of M


```
G_nominal = G.NominalValue;
G_nominal.InputName = 'Delta_lat2';
G_nominal.OutputName = {'p','phi'};

R.InputName = {'e_phi','p'};
R.OutputName = 'Delta_lat';

W.InputName = 'Delta_lat';
W.OutputName = 'z';

S1 = sumblk('%e_phi = phi0-%phi', R.InputName(1), ...
G.OutputName(2));
S2 = sumblk('%Delta_lat2 = %Delta_lat + w', ...
G_nominal.InputName, R.OutputName);

M = connect(G_nominal,R,W,S1,S2,'w','z');
```

Model in M-Δ form

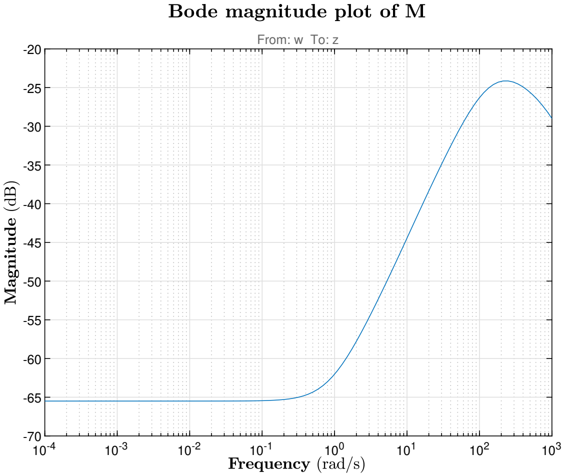


Condition

The condition of robust stability in M-Δ form is verified by the inequality:

$$|M(j\omega)| \leq 1, \quad \forall\omega$$

Presentation of results



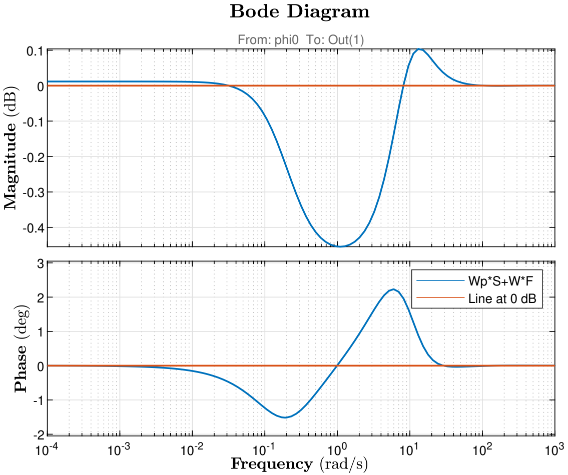
Verification of robust performance

Condition

The condition of robust performance is verified by the inequality:

$$|W_p(j\omega)\overline{S}(j\omega)| + |W(j\omega)\overline{F}(j\omega)| \leq 1 \quad \forall\omega$$

Presentation of results

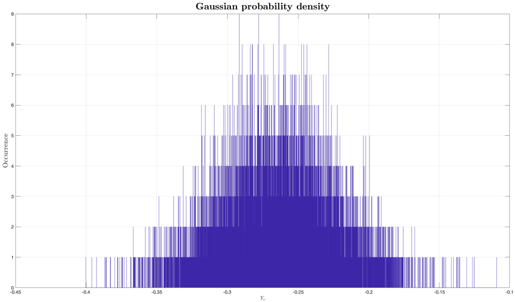


Montecarlo analysis

Re-sampling of uncertain parameters

An array of parameters of the plant can be created using a Gaussian probability distribution, for example:

```
Y_v_mc(n)= Y_v.NominalValue + Y_v.PlusMinus(2)* ...
randn(1,1);
```



Procedure

For each re-sampled parameter (N=100):

- Creation of the state space matrices.
- Creation of the state space model.
- Creation of the open loop transfer function.
- Computation of Gain and Phase margins.

```
[Gm(n),Pm(n)] = margin(L(n));
```

- Computation of step response and its information (settling time and overshoot).

```
y = step(minreal(L(n)/(1+L(n))),t);
S = stepinfo(y,t,1);
Sett(n)= S.SettlingTime;
Over(n)= S.Overshoot;
```

Presentation of results - Step response

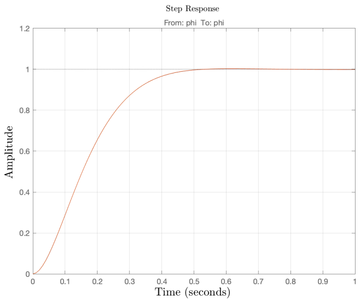


Figure: Step response

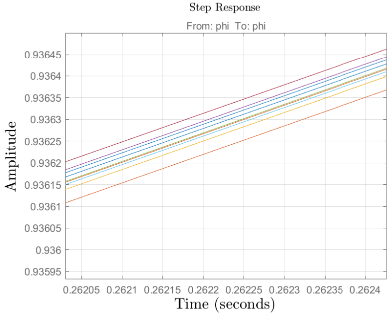
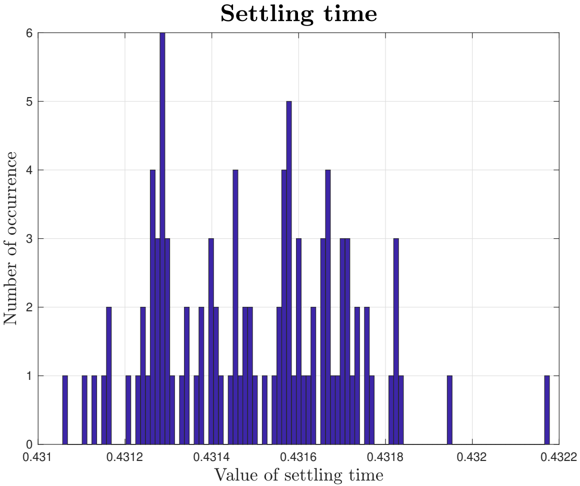
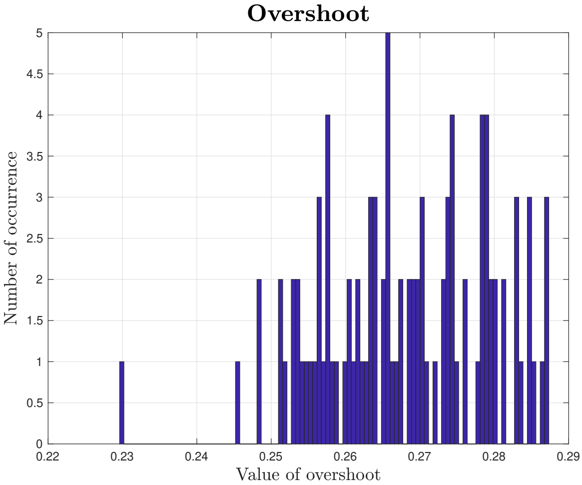


Figure: Zoom on step response

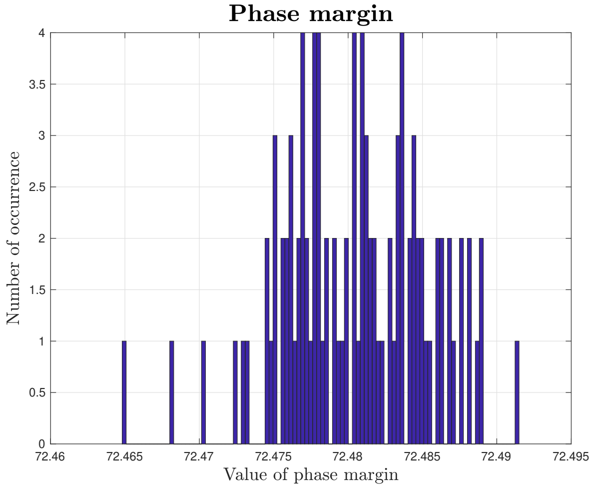
Presentation of results - Step Info (1)



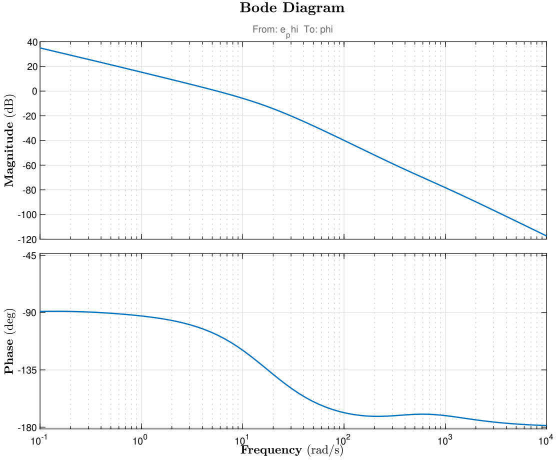
Presentation of results - Step Info (2)



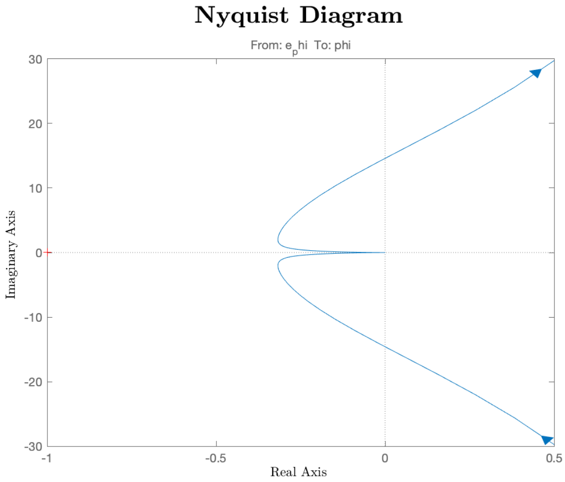
Presentation of results - Bode Info



Presentation of results - Bode Info



Presentation of results - Bode Info

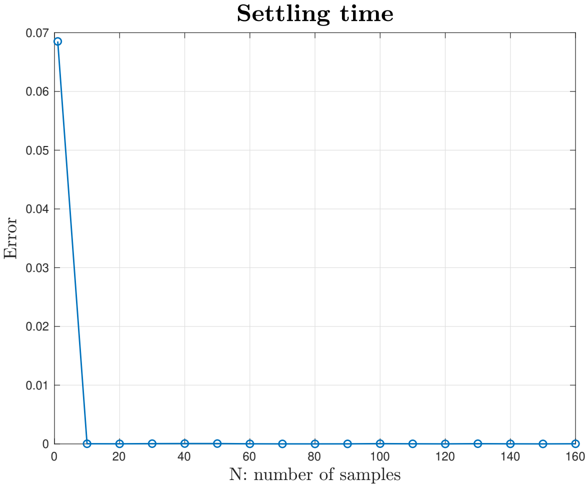


Convergence of Montecarlo method

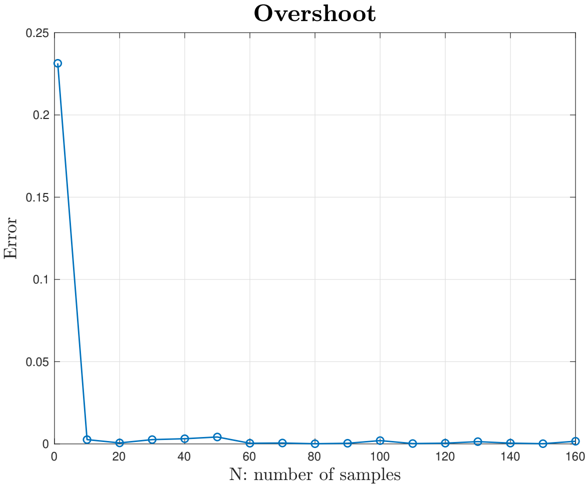
A.Occhipinti, S.Piacquadio, L.Quadri

What is the effect of increase the number of samples in Montecarlo analysis?

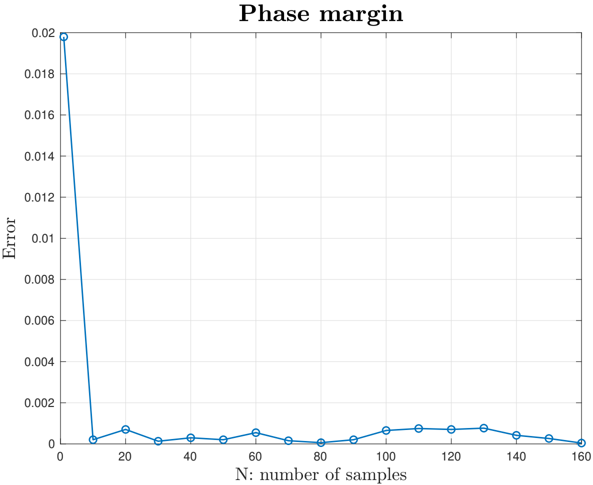
Presentation of results



Presentation of results



Presentation of results



μ-analysis

A.Occhipinti, S.Piacquadio, L.Quadri

Structured Singular Value

Computation of the the array of the frequencies:

```
omega = logspace(−2,3,500);
```

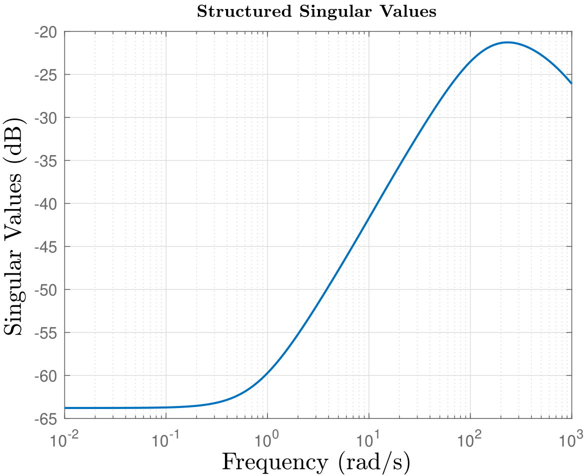
Computation of the bounds:

```
bounds = mussv(frd(M,omega),[1,0]);
```

The condition for RS is:

$$\mu(M(j\omega)) < 1, \quad \forall\omega$$

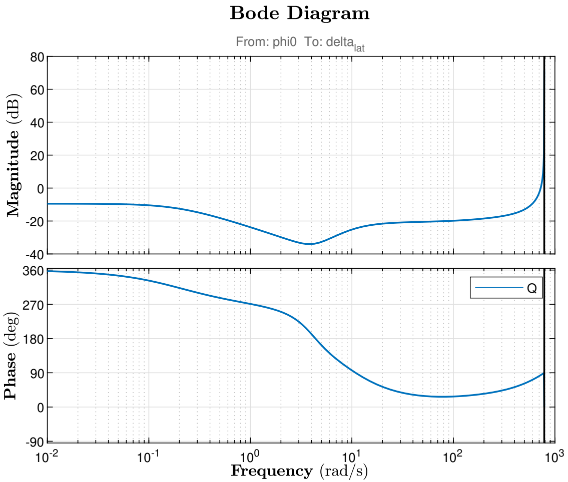
Presentation of results



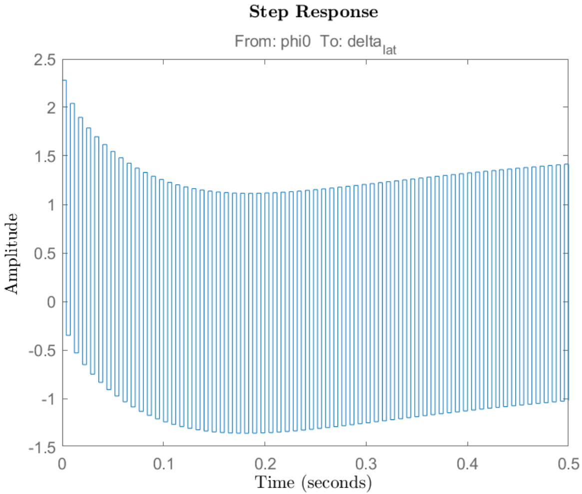
Critical observations on Control

A.Occhipinti, S.Piacquadio, L.Quadri

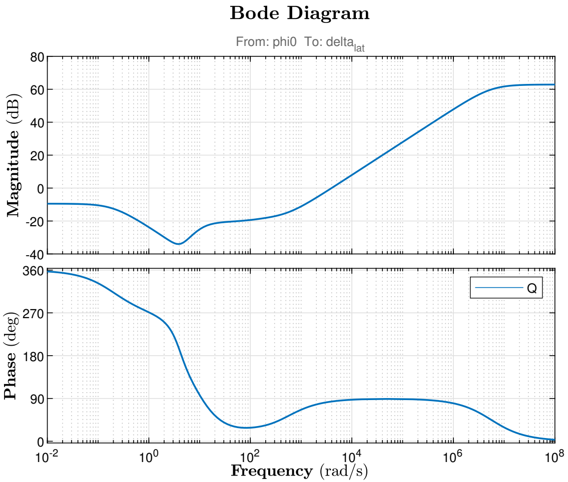
Control sensitivity function (1)



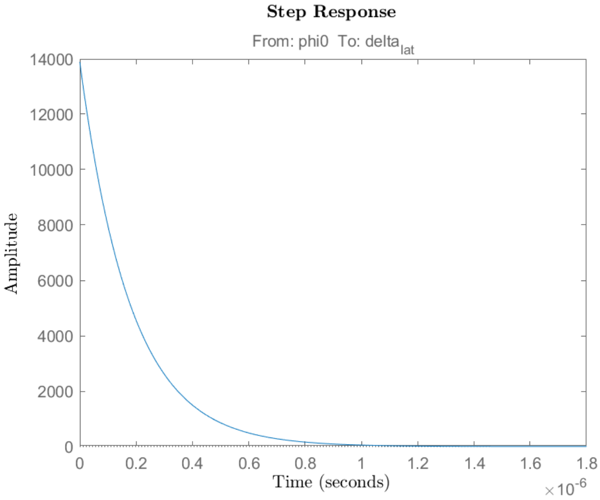
Control sensitivity function (2)



Control sensitivity function (3)



Control sensitivity function (4)



Thanks for your attention!