Robust design of attitude and position control system for a multirotor UAV

Aerospace Control Systems
Final exam project
AA 2022-2023

Pietro Dal Lago
Lorenzo Cucchi
Alberto Armanni

Summary

- Formulation of the problem
- Attitude control of simplified model
- Attitude control with motor and battery dynamics
- Robust analysis
- Monte Carlo analysis

Formulation of the problem

Given the lateral-directional dynamics of a quadrotor drone, in terms of a statespace LTI system with stability and control derivatives, the purpose of the project is to design a control system for the roll angle ϕ.

Lateral dynamics:

Dynamic equation

$$\begin{cases} \dot{x} = Ax + Bu \\ y = Cx + Du \end{cases} \longrightarrow$$

$$\begin{bmatrix} \dot{v} \\ \dot{p} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} Y_v & Y_p & g \\ L_v & L_p & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} v \\ p \\ \phi \end{bmatrix} + \begin{bmatrix} Y_d \\ Y_d \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} p \\ \varphi \\ a_y \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ Y_v & Y_p & 0 \end{bmatrix} \begin{bmatrix} v \\ p \\ \phi \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ Y_\delta \end{bmatrix}$$

Formulation of the problem

Followed procedure

Simplified model

Task 1
- Nominal performance
- Control effort

Task 2
- Robustness analysis

Complete model

Task 2
- Nominal performance
- Control effort
- Robustness analysis

Task 3
- Observer design

Task 4
- Monte Carlo Validation

Formulation of the problem

Data

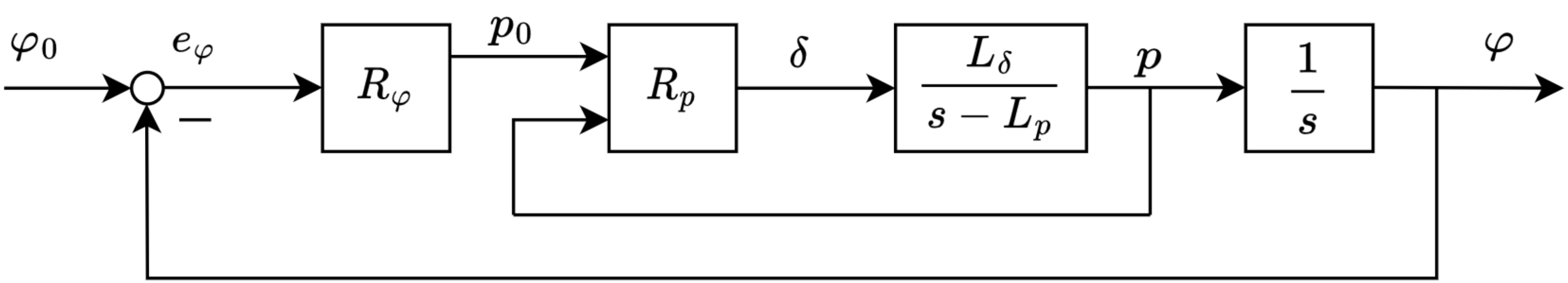
Derivatives	Nominal Value	Units	Standard deviation [%]
Y_v	-0.1068	s^{-1}	4.26
Y_p	0.1192	$m\ s^{-1} rad$	2.03
L_v	-5.9755	$rad\ s\ m^{-1}$	1.83
L_p	-2.6478	s^{-1}	2.01
Y_d	-10.1647	$m\ s^{-2}$	1.37
L_d	450.785	$rad\ s^{-2}$	0.81

The nominal values for stability and control derivatives are listed in the table on the left

For each one the standard deviations is expressed as percentage of the corresponding nominal values (assuming Gaussian distribution for each derivative)

Attitude control

(with assumptions)



Control-oriented assumptions:

- $Y_\delta = 0$
- $Y_p = 0$
- $L_v = 0$

Resulting equations:

- $\dot{\varphi} = p$
- $\dot{p} = L_p p + L_\delta \delta$
- $\dot{v} = Y_v + g\varphi$

Transfer functions:

- $\varphi = \frac{1}{s} p$
- $p = \frac{L_\delta}{s - L_p} \delta$
- $v = \frac{g}{s - Y_v} \varphi$

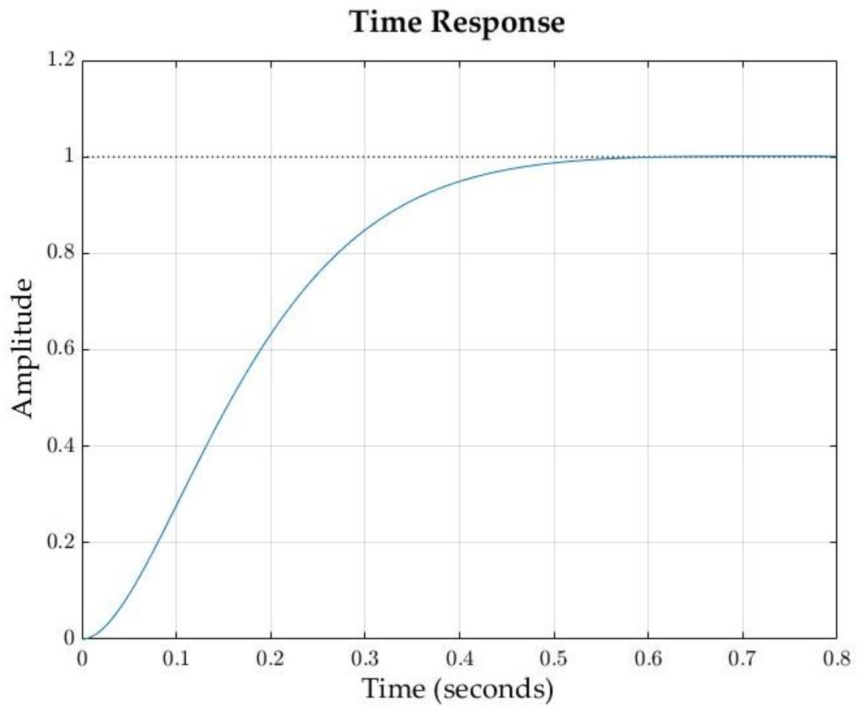
Requirements:

- Nominal performance: response of ϕ to its setpoint equivalent to a second order system with $\omega_n \geq 10$ rad/s and ξ ≥ 0.9
- Control effort limitation: 5% for a doublet setpoint (+10° between 1÷3 sec and -10° between 3÷5 sec)
- Robust stability
- Robust performance

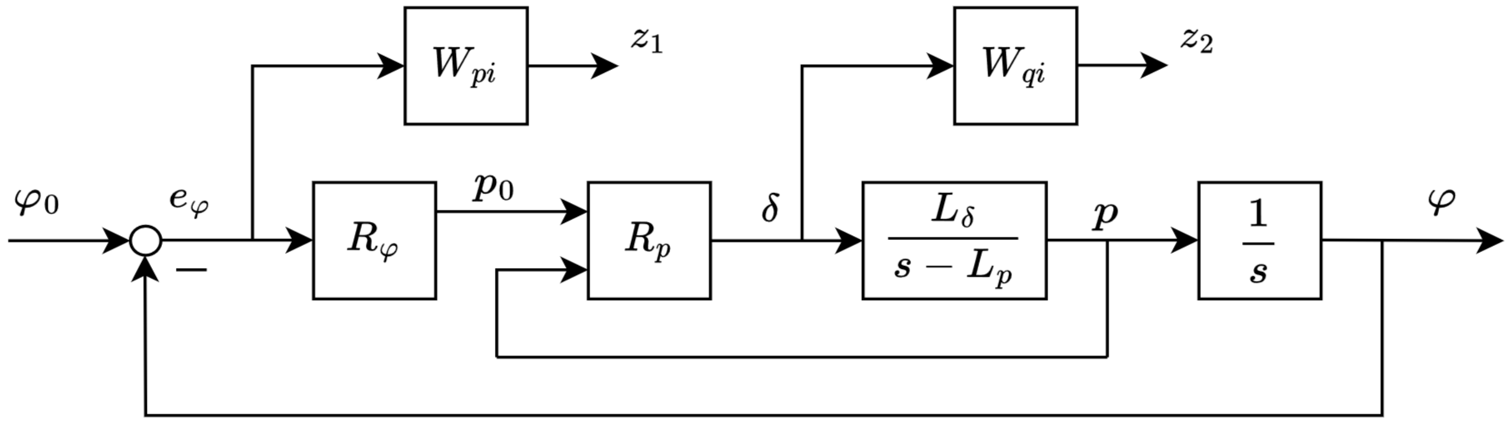
Task 1 (with assumptions)
weight selection for nominal response

$$F_{\varphi^0 \to \varphi}^{target} = \frac{\omega_n^2}{s^2 + 2\xi\omega_n s + \omega_n^2}$$

Step Info	
Rise Time	0.2883
Transient Time	0.4700
Settling Time	0.4700
Settling Min	0.9024
Settling Max	1.0015
Overshoot	0.1524
Undershoot	0
Peak	1.0015
Peak Time	0.7215



Task 1 (with assumptions)
Augmented Plant, tuning

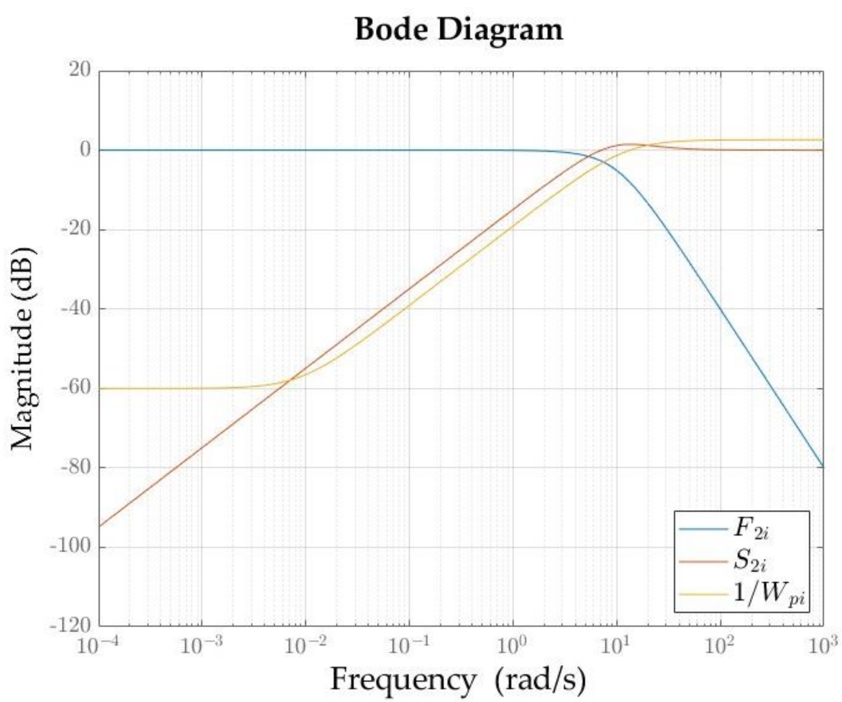


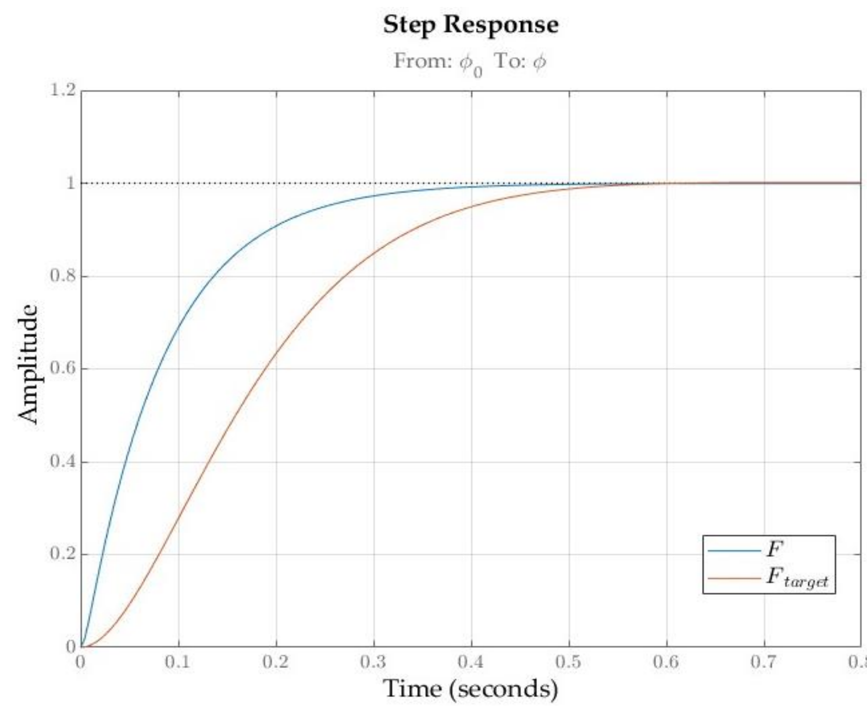
- The control-oriented assumptions allow a cascade architecture
- R_φ is a proportional controller
- R_p is a 2-degree of freedom PID
- Two weighting functions are added to the augmented plant for the structured H∞ synthesis of the controllers

Task 1 (with assumptions)
Weight selection for nominal response

$$W_p = \frac{\frac{s}{M} + \omega_b}{s + A\omega_b}$$

M =1.35
A=0.001
$\omega_b = 0.9\omega_n$

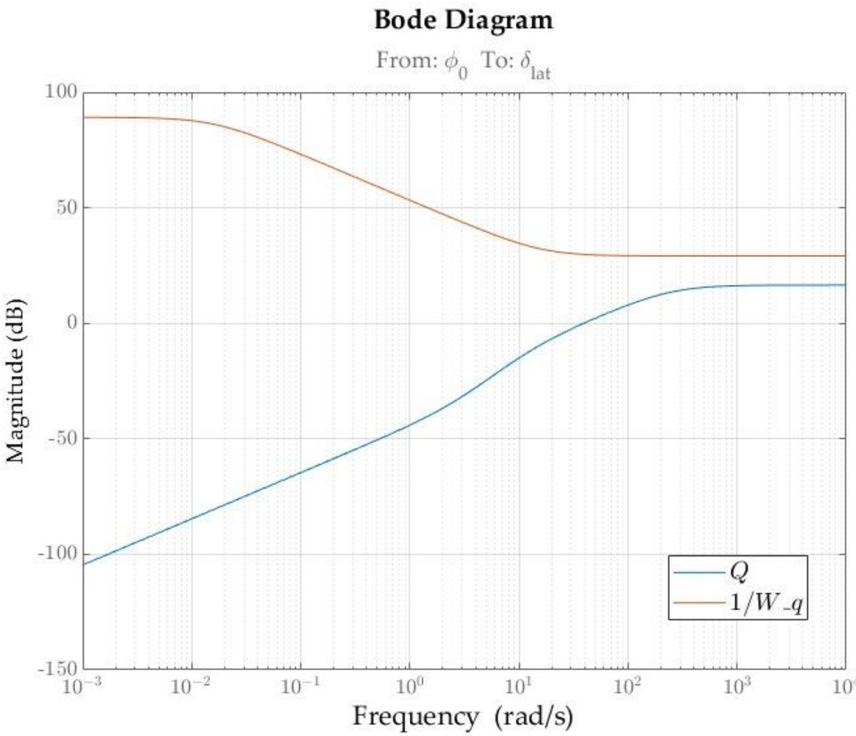


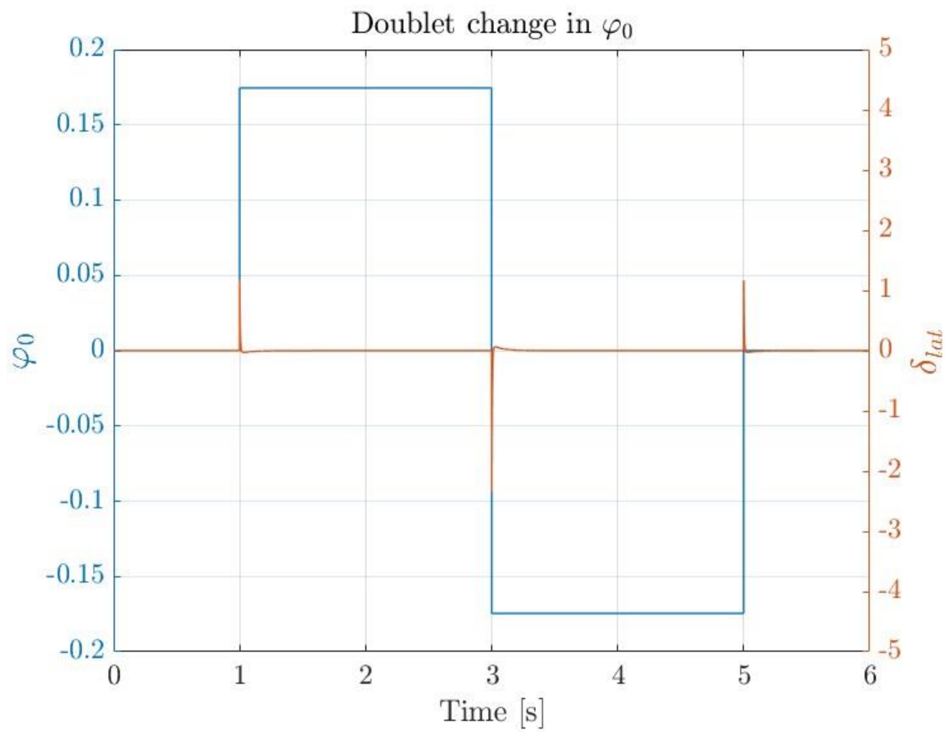


Task 1 (with assumptions)
Weight selection for control effort limitation

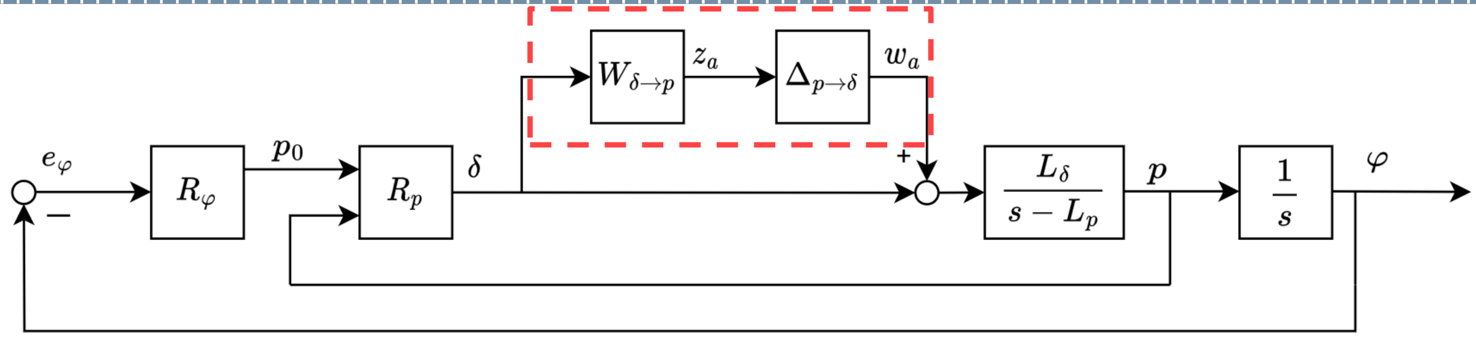
$$W_q = \alpha \frac{s + \omega_{act} 10^{-3}}{s + \omega_{act}}$$

$$\alpha = \frac{10 \frac{180}{\pi}}{5}$$
$$\omega_{act} = 16$$





Task 1 (with assumptions)
Robustness Verification



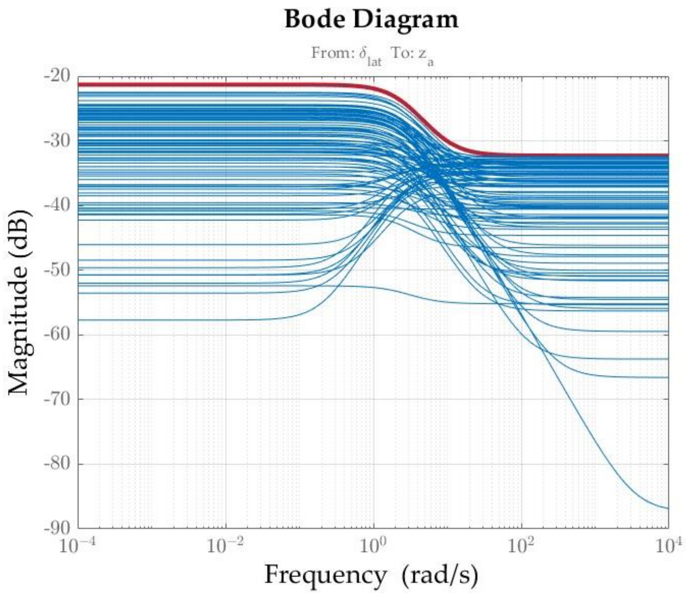
The weight for robust stability and robust performance is constructed with 'usample' and 'ucover' as an upper limit for the uncertain model

```
delta_p_array = usameple(sys_delta_p,100);
[~,Info] = ucover(delta_p_array, sys_delta_p_nom, 3);
W_delta_p = Info.W1;


SumInner2 = sumblk('uili = \delta_{lat} + w_a')

CL = connect(R_phi, R_p, sys_p_phi_nom, sys_delta_p_nom, SumInner1,...
SumInner2, W_delta_p, 'w_a', 'z_a',{'\delta_{lat}','\phi'});

M = getIOTransfer(CL, 'w_a', 'z_a');
```



Task 1 (with assumptions)
Robustness analysis

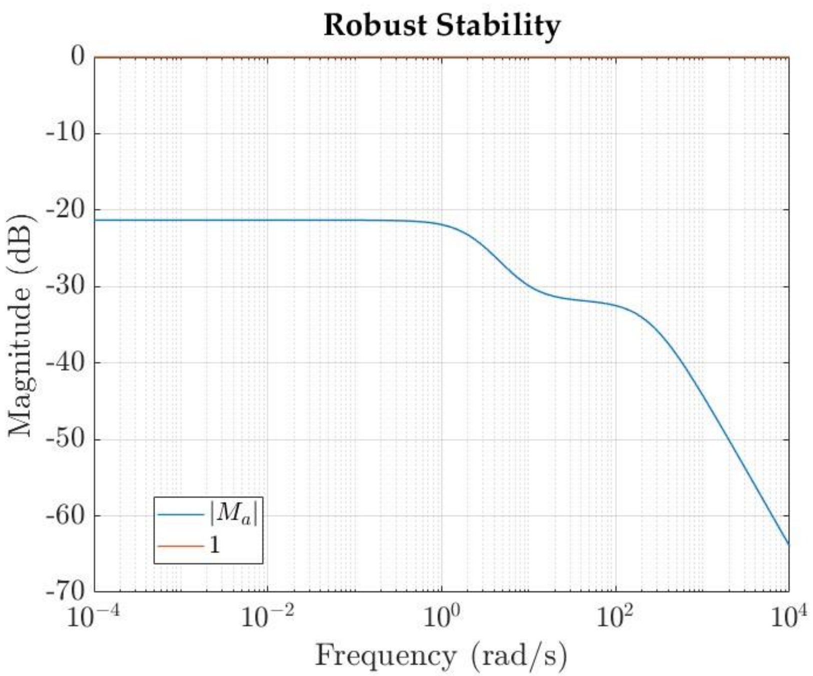
$$R_\varphi = k_p(\varphi_0 - \varphi)$$

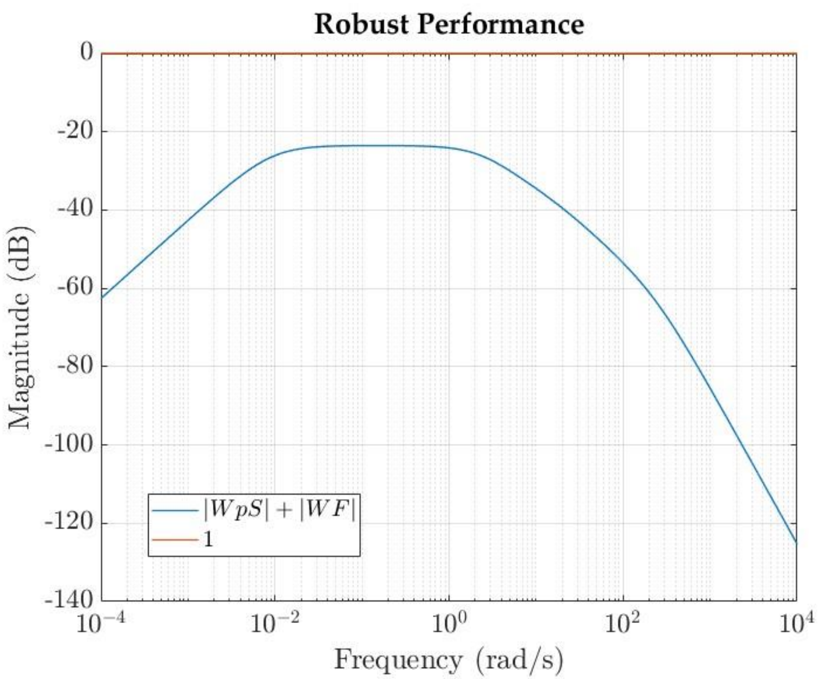
$$R_p = k_p(bp_0 - p) + \frac{k_i}{s}(p - p_0) + k_d \frac{s}{Ts+1}$$

T= 0.001s, b =1, c=1

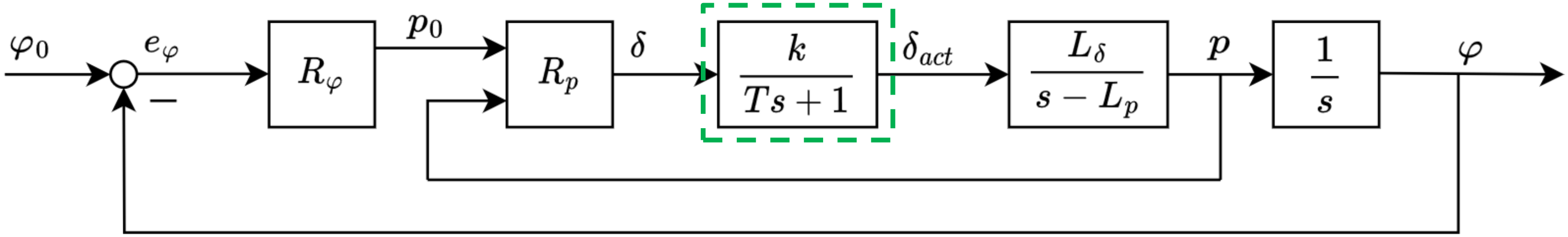
Proportional controller	
k_p	11.5857
Settling Time	0.3261 [s]

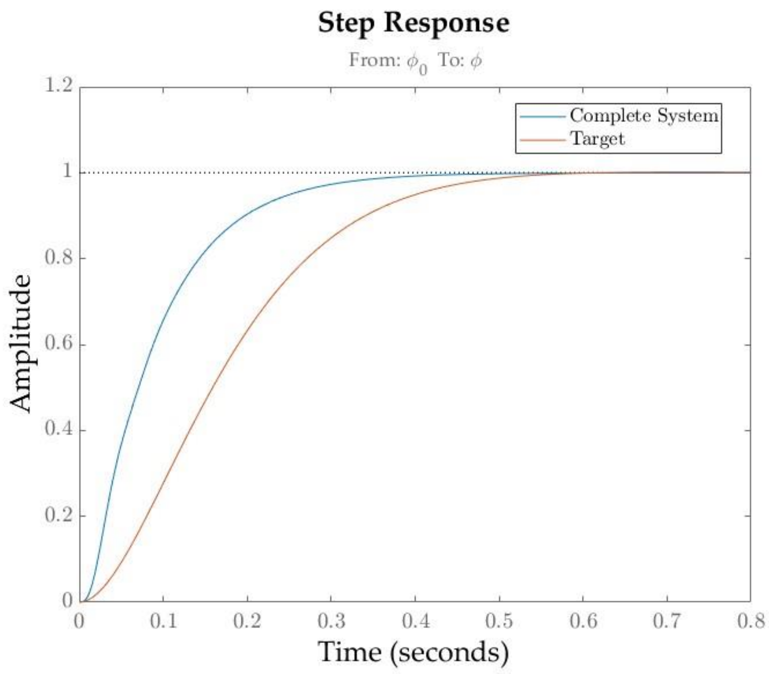
PID	
k_p	0.5782
k_d	1.5613
k_i	2.745e-7

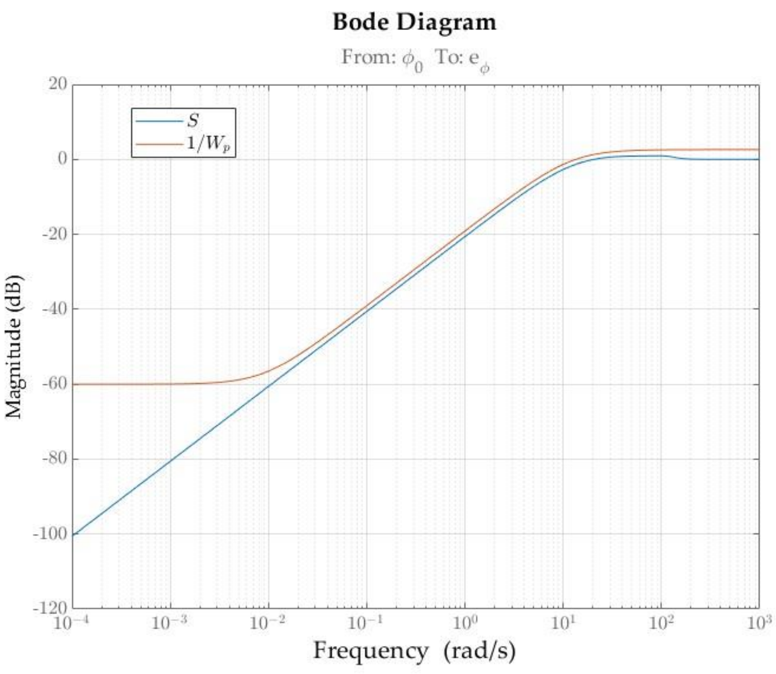




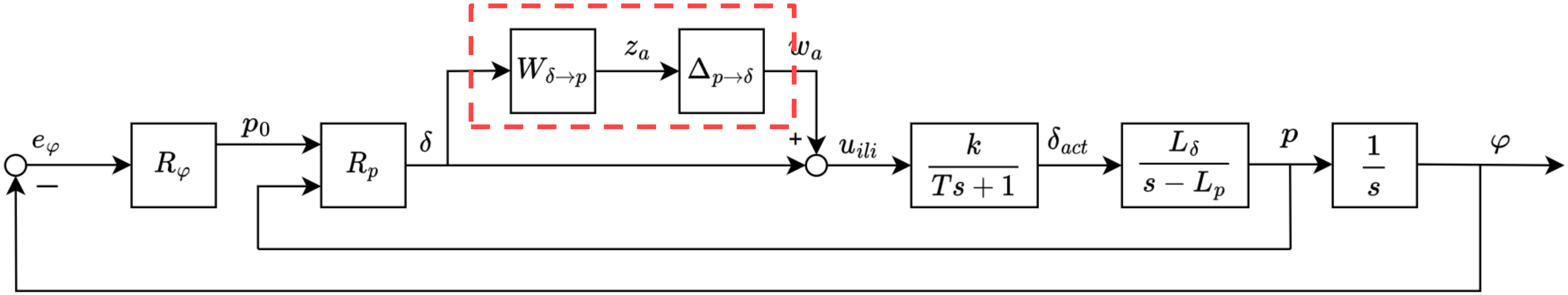
Task 2 (with assumptions)
Propulsive unit and battery addiction, tuning







Task 2 (with assumptions)
Robustness Verification

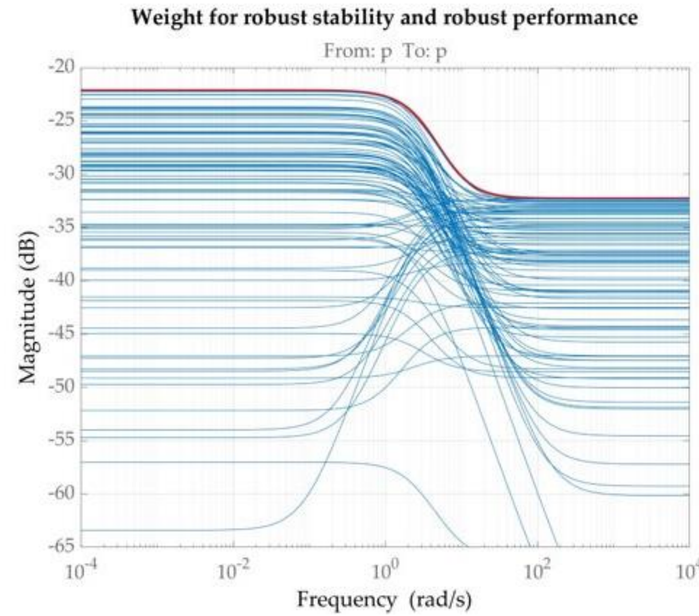



```
delta_p_array = usameple(sys_delta_p,100);
[~,Info] = ucover(delta_p_array, sys_delta_p_nom, 3);
W_delta_p = Info.W1;


SumInner2 = sumblk('uili = \delta_{lat} + w_a')

CL = connect(R_phi, R_p, sys_d_db_nom, sys_db_da, sys_delta_p_nom,...
SumInner1, SumInner2, W_delta_p, 'w_a', 'z_a',{'\delta_{lat}','\phi'});

M = getIOTransfer(CL, 'w_a', 'z_a');
```



Task 2 (with assumptions)
Propulsive unit and battery addiction, tuning

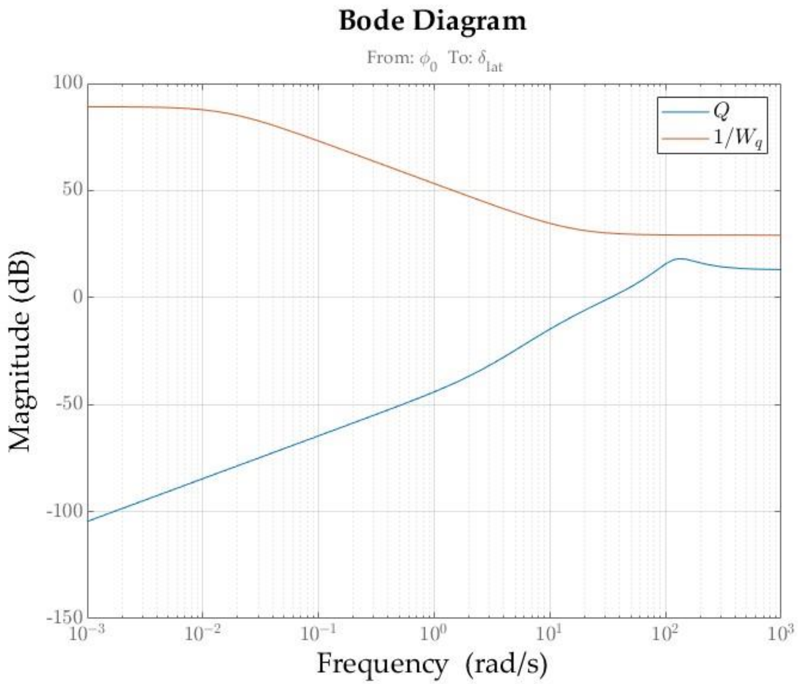
$$R_\varphi = k_p(\varphi_0 - \varphi)$$

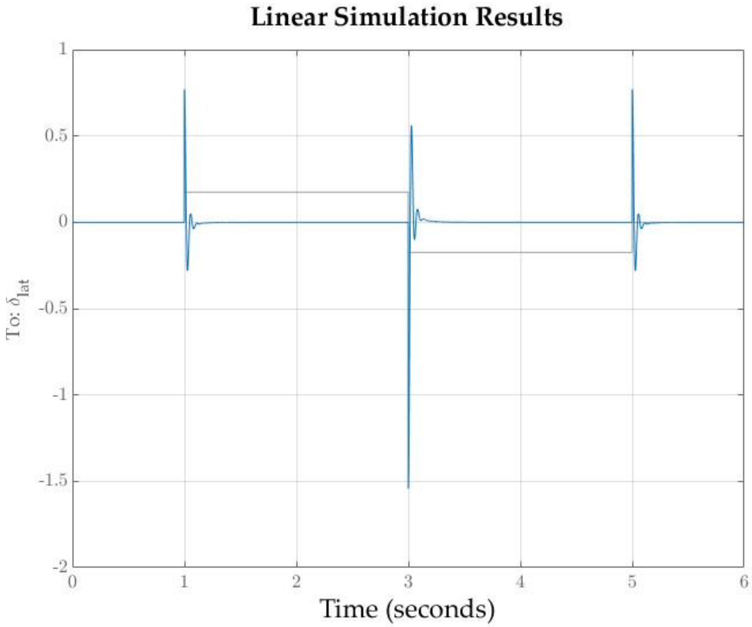
$$R_p = k_p(bp_0 - p) + \frac{k_i}{s}(p - p_0) + k_d\frac{s}{Ts+1}$$

T= 0.001s, b =1, c=1

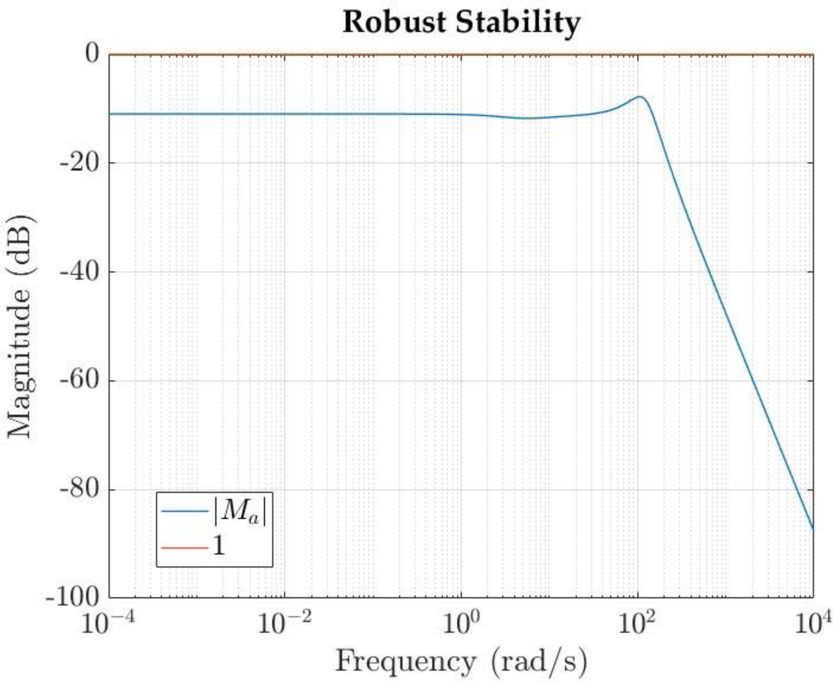
Proportional controller	
k_p	10.8

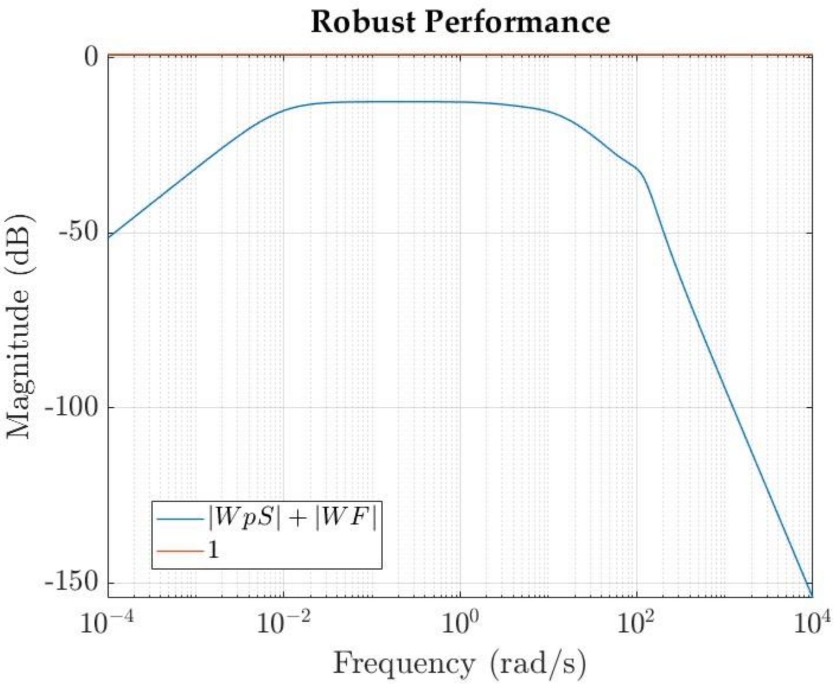
PID	
k_p	0.47
k_i	4.3e-8
k_d	0.00303





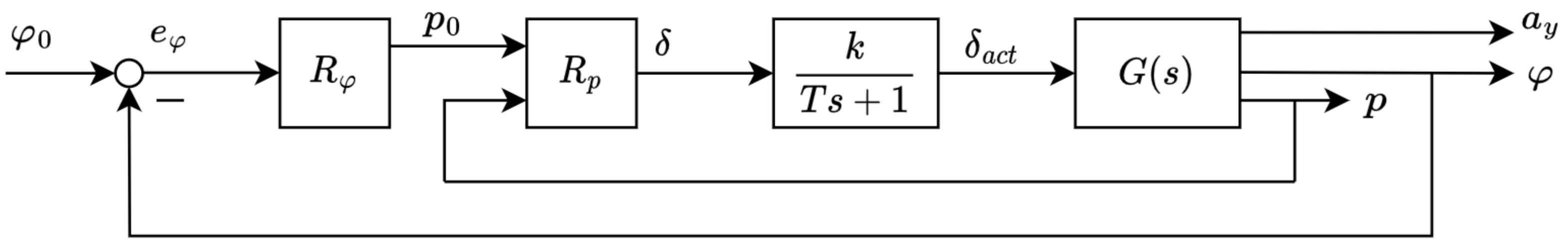
Task 2 (with assumptions)
Results and comparison with the requirements



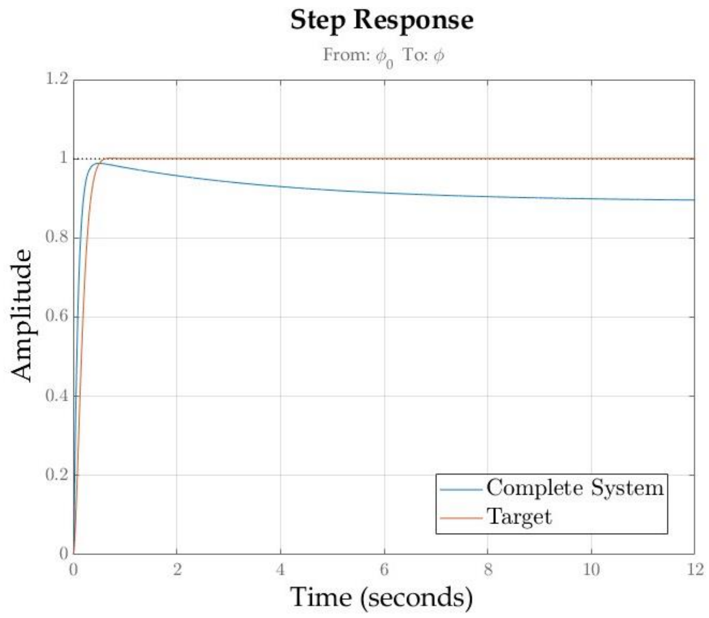


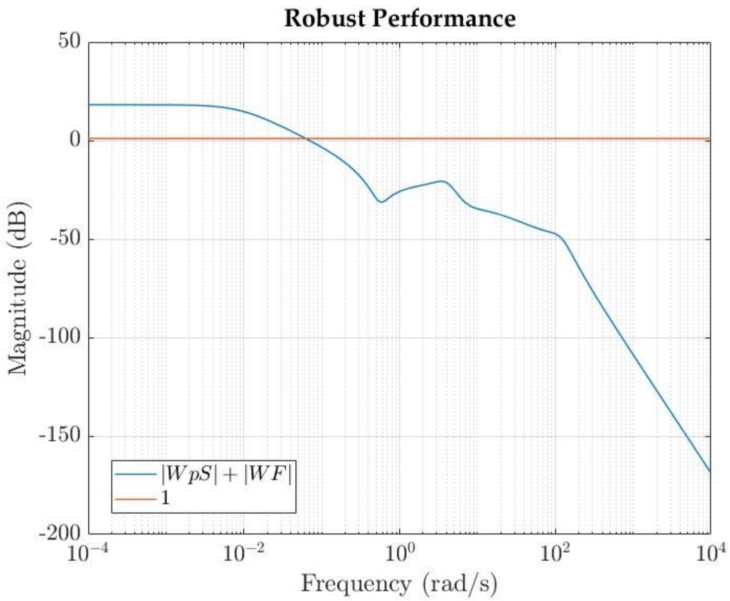
Relaxing assumptions

Task 2 (complete)
Pid from simplified system



The Integral gain of the Rp pid obtained from the simplified system is 0, with the complete system we need an integral gain to track the error.





Task 2 (complete)

Results and comparison with requirements

$$R_\varphi = k_p(\varphi_0 - \varphi)$$

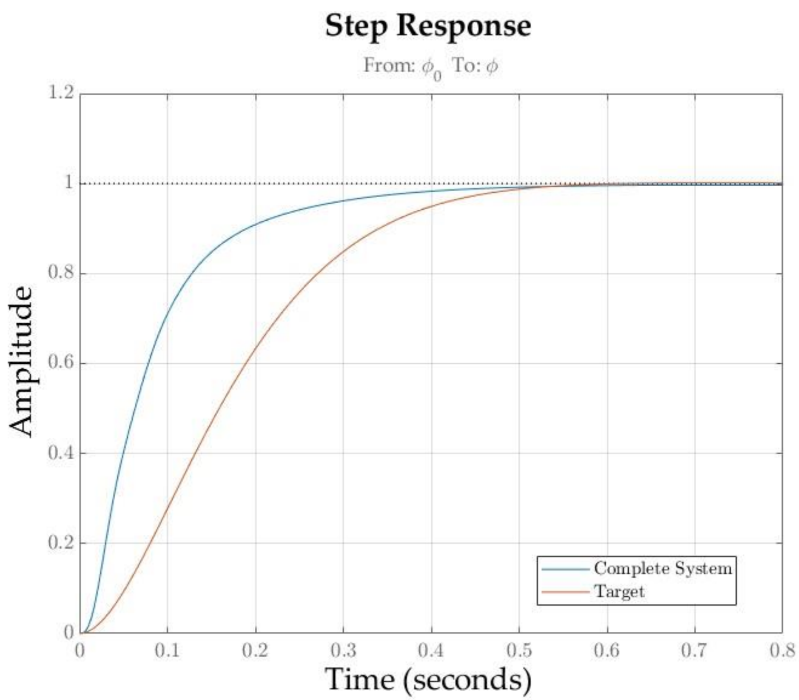
$$R_p = k_p(bp_0 - p) + \frac{k_i}{s}(p - p_0) + k_d\frac{s}{Ts+1}$$

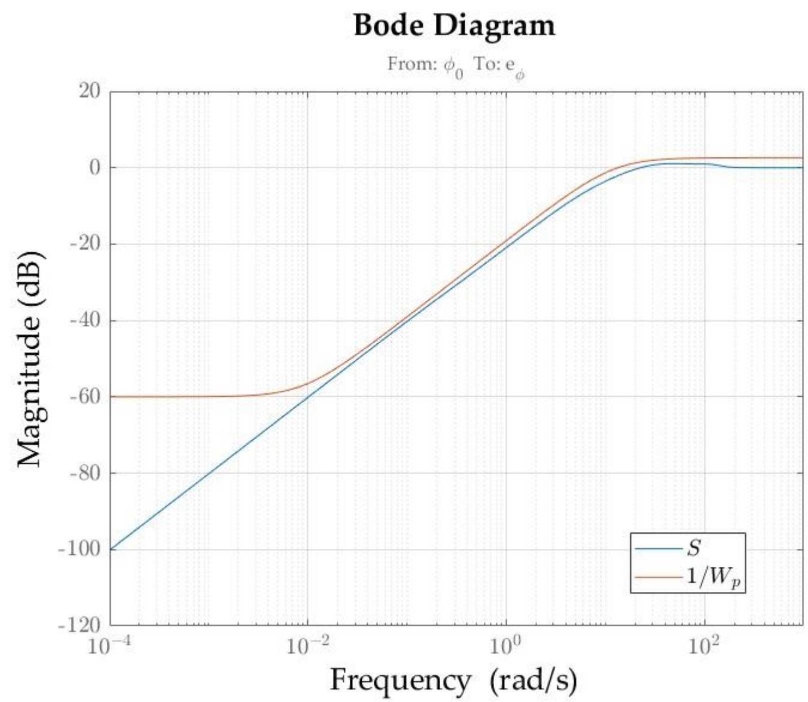
T= 0.001s, b =1, c=1

Proportional controller	
k_p	11.1

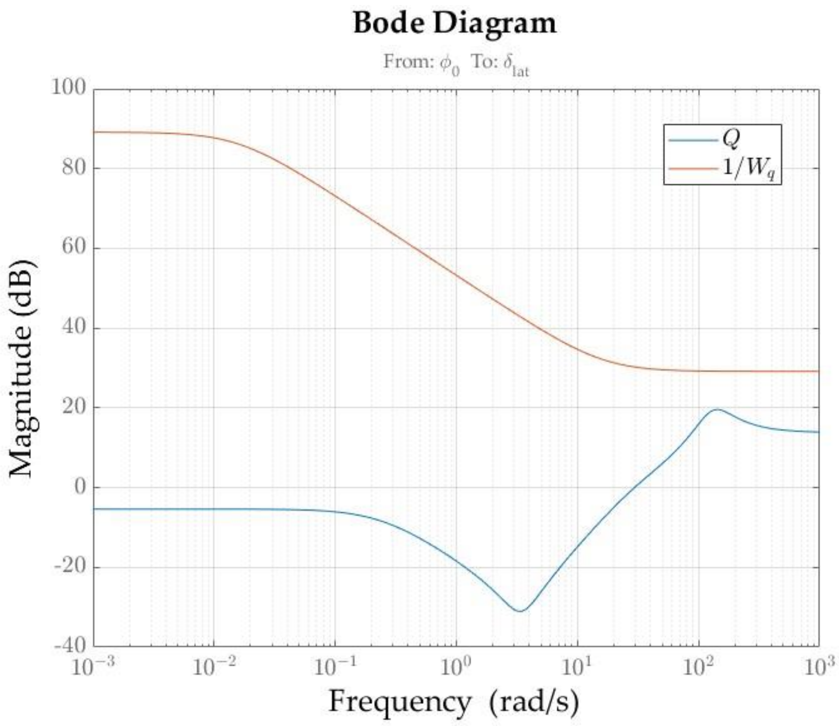
Settling Time	0.3805 [s]

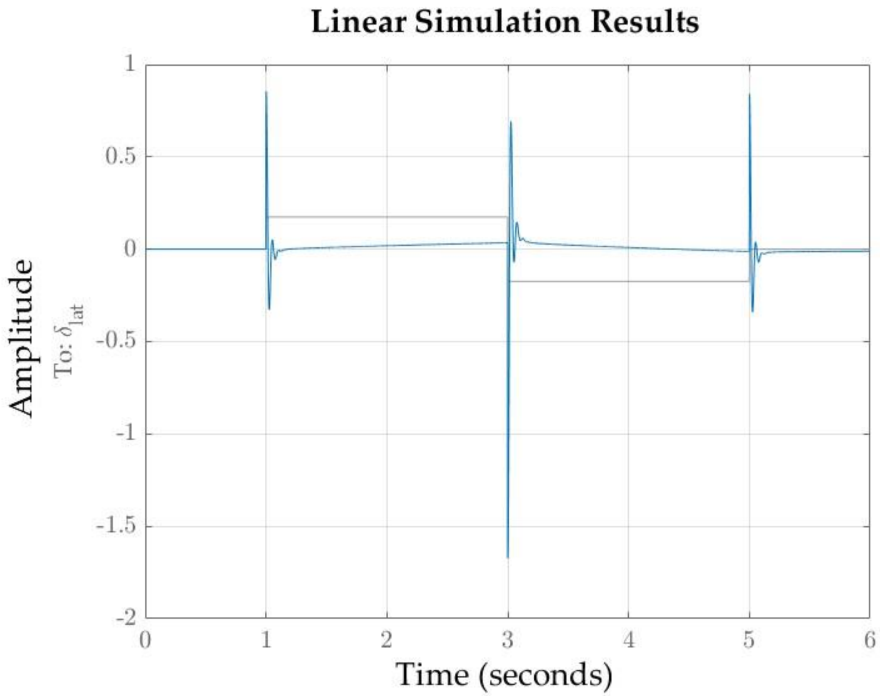
PID	
k_p	0.438
k_i	5.67
k_d	0.00415



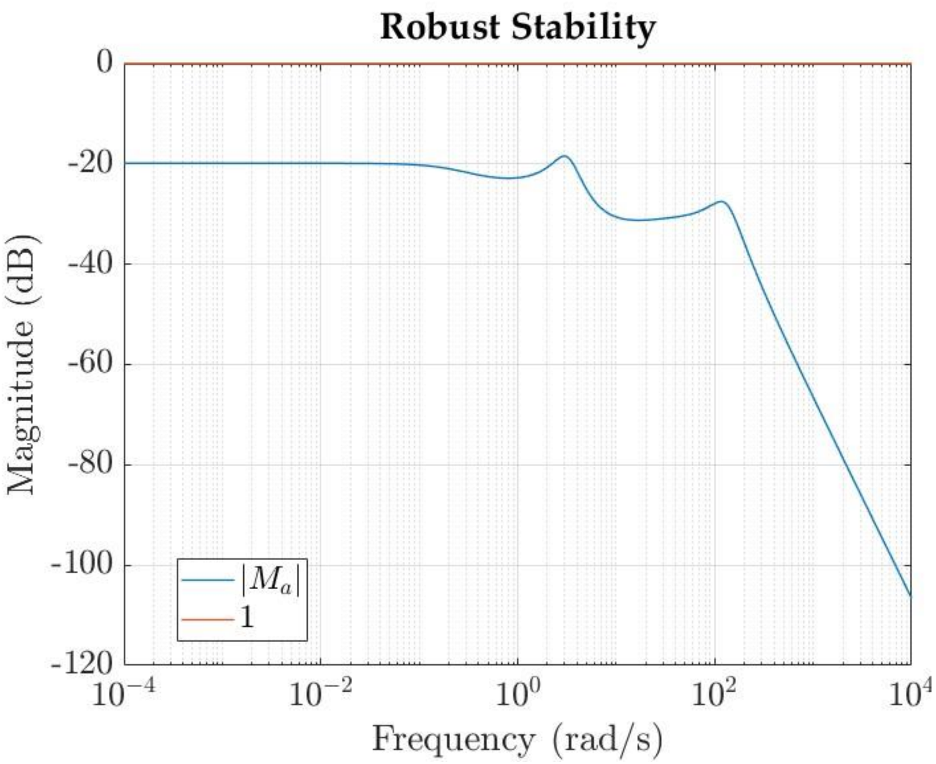


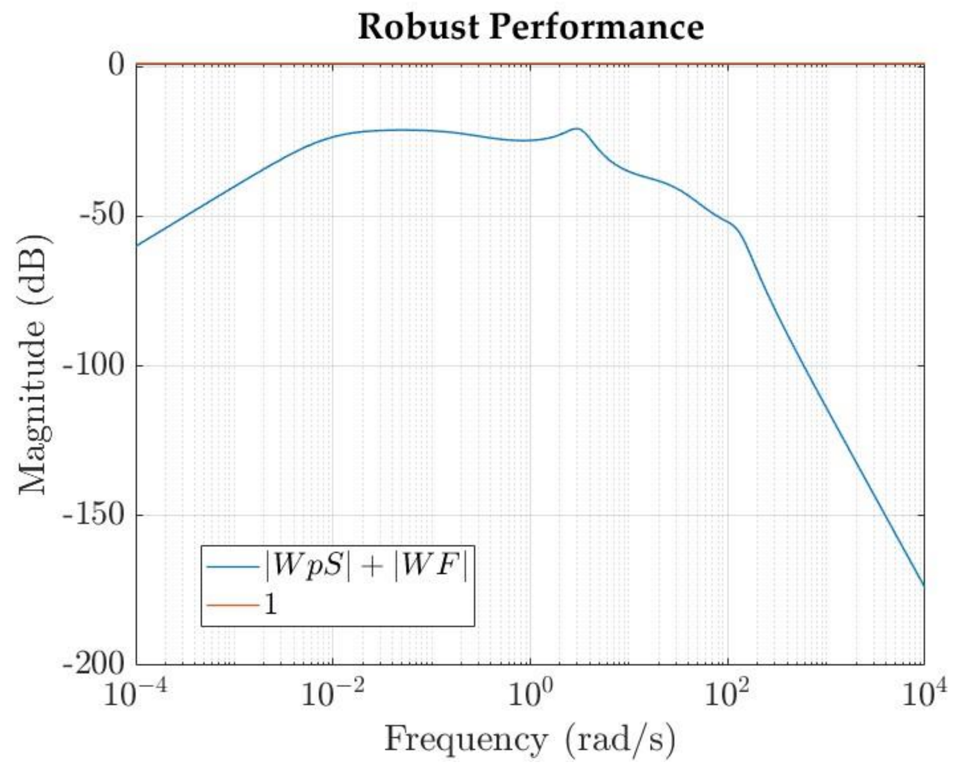
Task 2 (complete)
Re-tuning and compliance with requirements



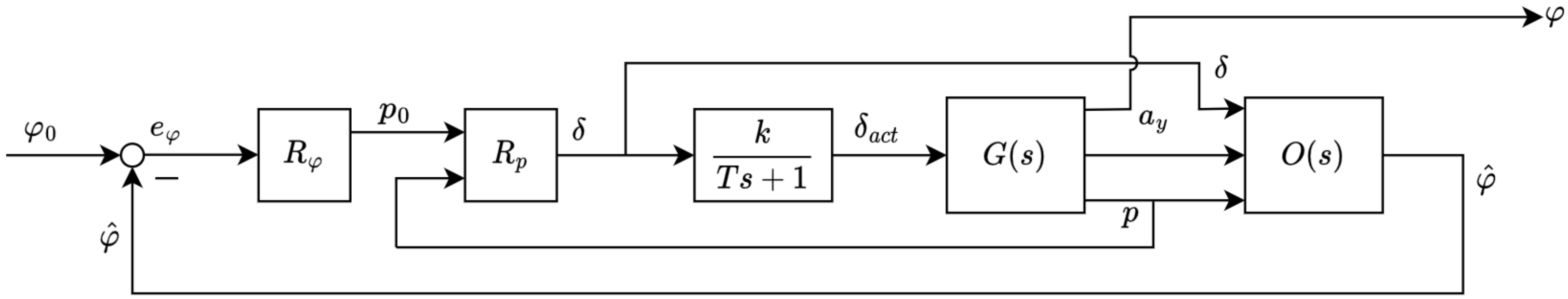


Task 2 (complete)
Robustness analysis





Task 3 (complete)
Observer design



A Luenberger state observer θ estimates ϕ through the measurements of δ, a_y and p.

The dynamics of the observer is governed by the state matrix (A – L*C).

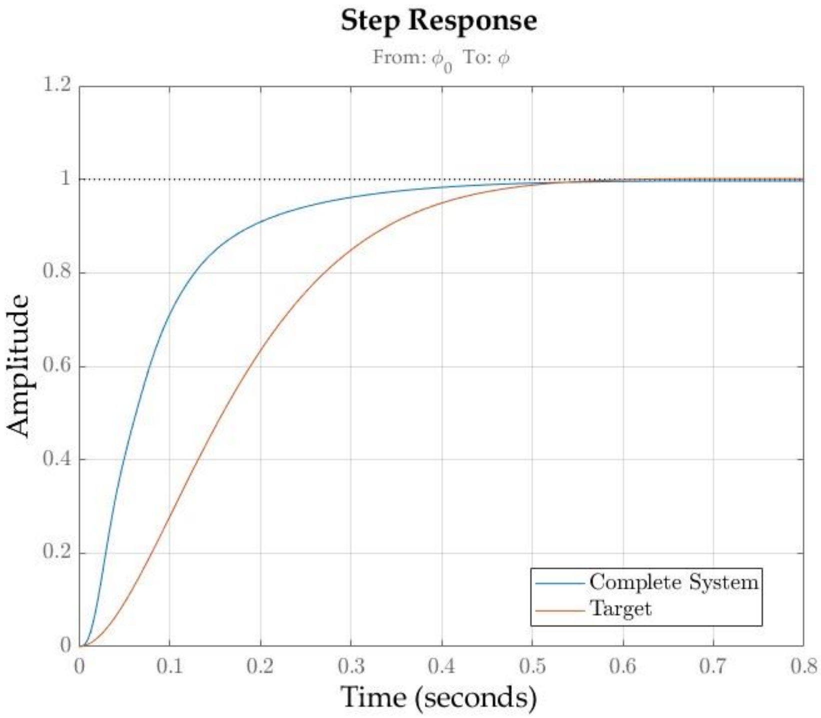
The problem of defining the L matrix is an eigenvalue assignment, to solve it Matlab's command *place* is used

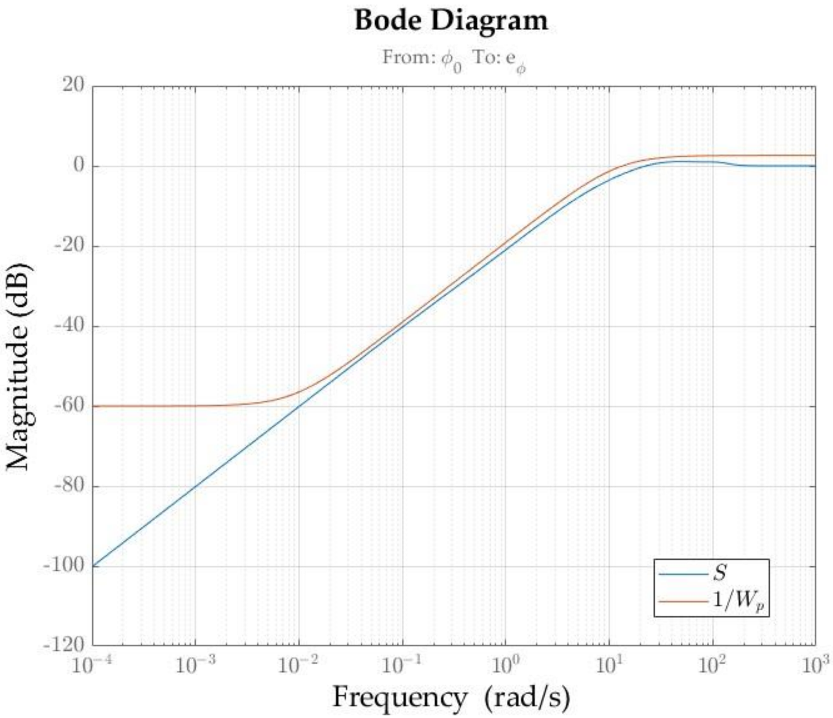
$$\begin{cases} \dot{\hat{x}} = (A - LC)\hat{x} + (B - LD)u + Ly \\ \hat{y} = C\hat{x} + Du \end{cases}$$

$$L = \begin{bmatrix} 0.5692 & -3.7753 \\ 0.6829 & 55.9504 \\ 1.000 & -0.0048 \end{bmatrix}$$

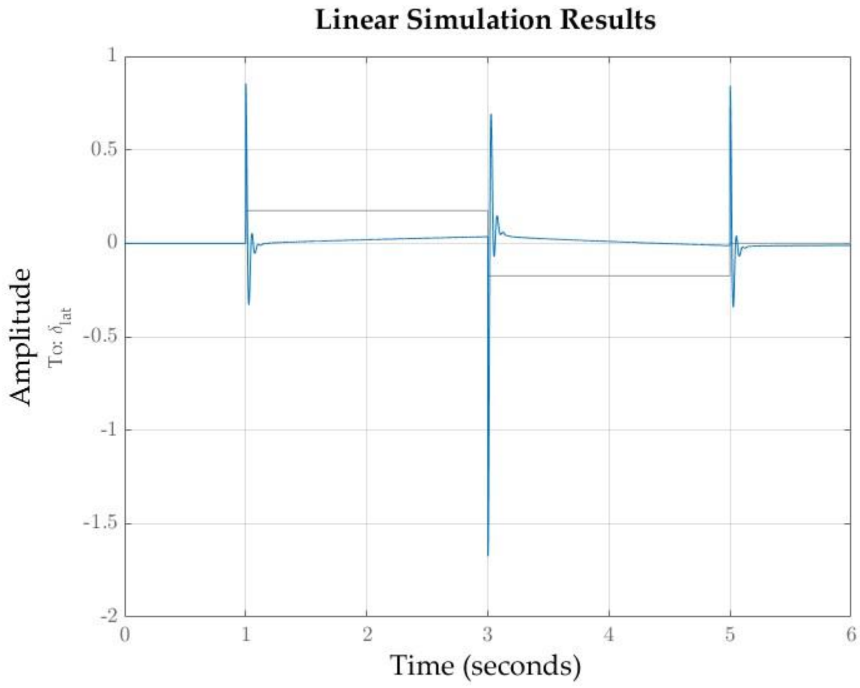
$$P = \begin{bmatrix} -0.5 \\ -0.01 \\ -10 \end{bmatrix}$$

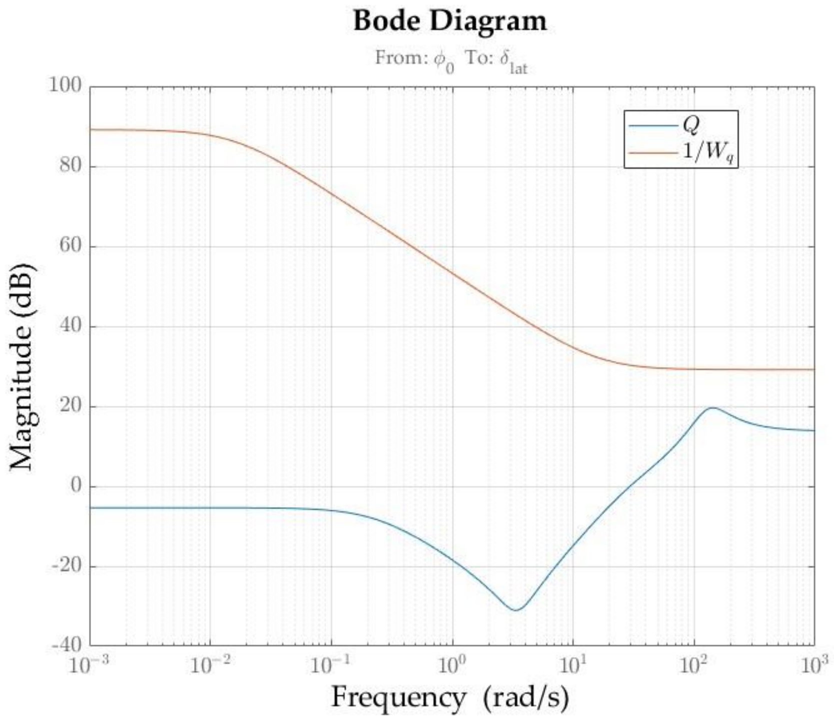
Task 3 (complete)
Robust stability analysis





Task 3 (complete)
Compliance with the requirements





Monte Carlo Validation

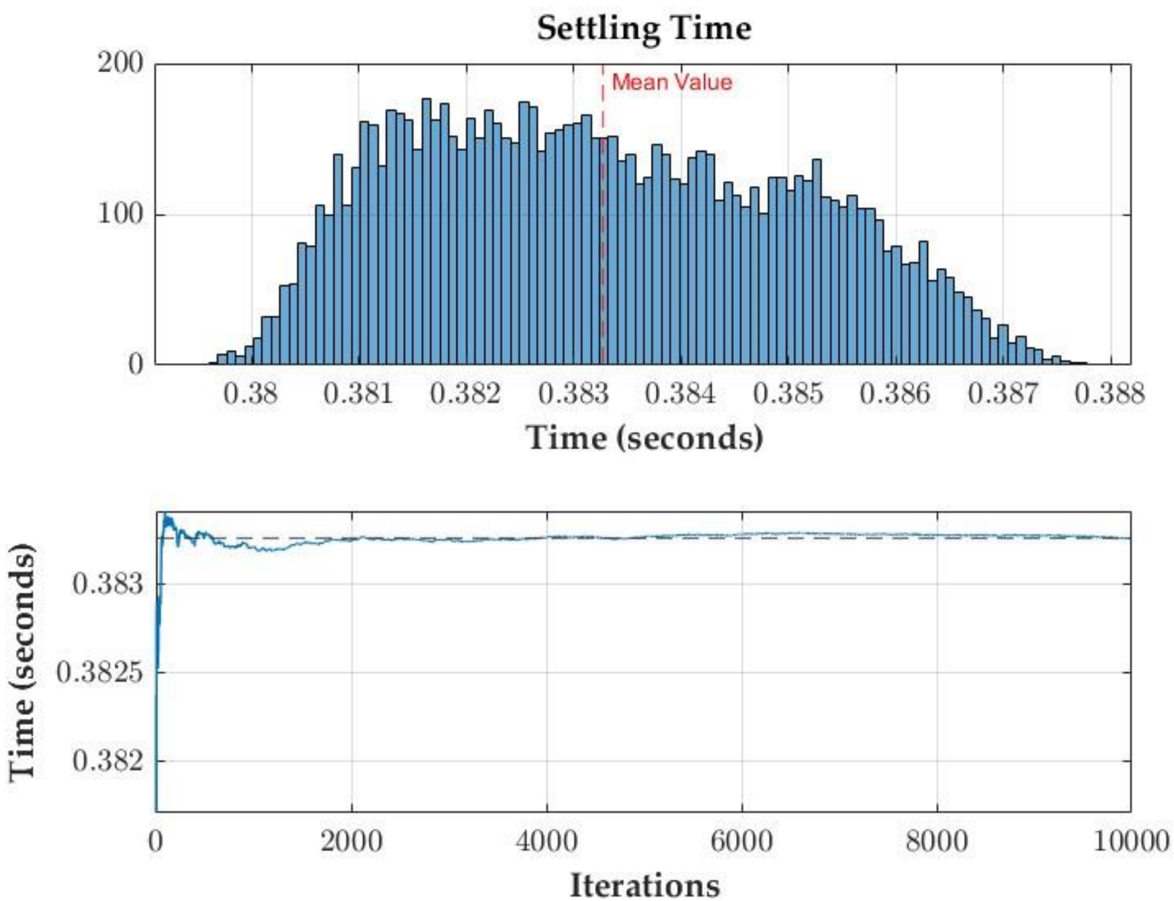
A Montecarlo method with 10000 iterations is used to validate the observer design and the system tuning:

- Step response of ϕ in terms of percentage overshoot and settling time
- Control effort

The plots present a graph of the convergence of the mean value, this is used to show after how many iteration the results are stable.

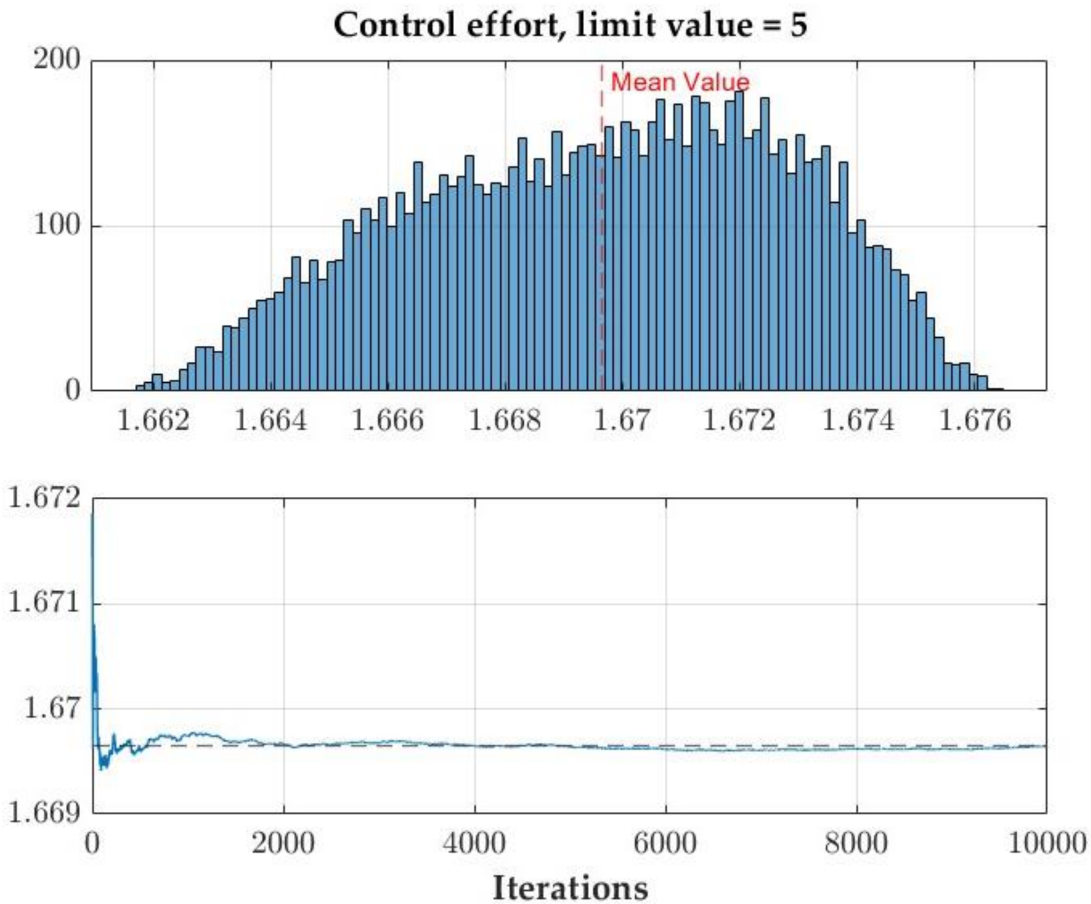
Indicatively two thousands iteration are enough to have an accurate convergence of the mean and of the standard deviation

Task 4 (complete)
Monte Carlo validation



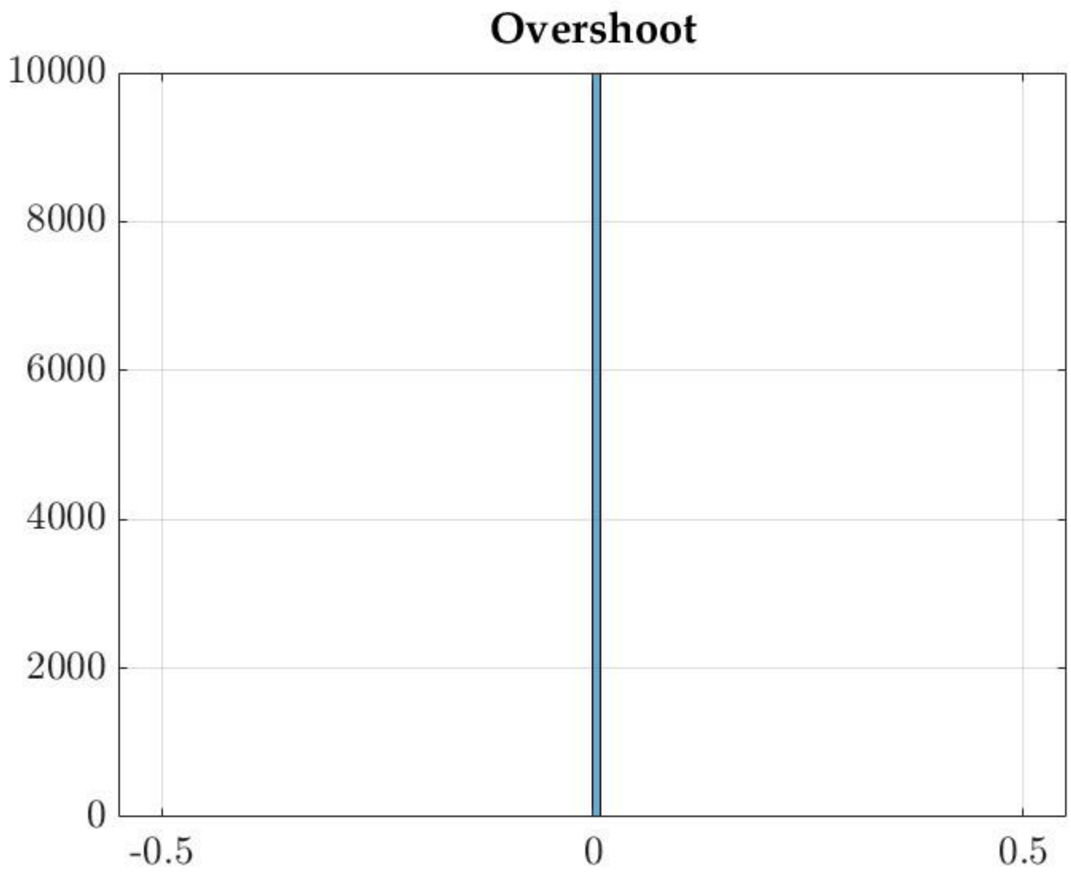
Settling Time	
Mean Value:	0.383262 [s]
Standard Deviation:	0.001755 [s]

Task 4 (complete)
Monte Carlo validation

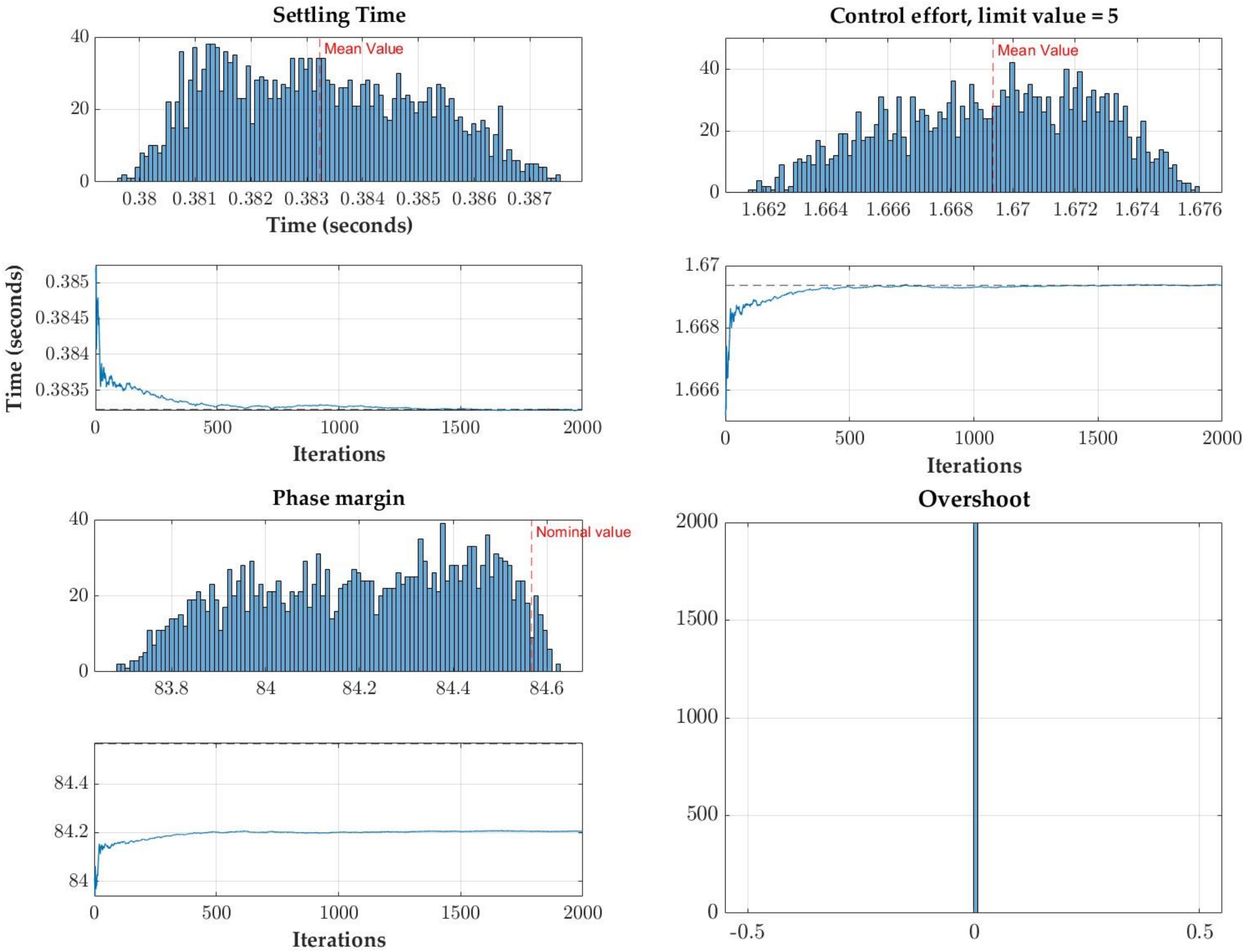


Control Effort	
Mean Value:	1.669646 [deg]
Standard Deviation:	0.003148 [deg]

Task 4 (complete)
Monte Carlo validation



Task 4 (complete)
Monte Carlo validation



Samples	2000

Step Info	
Mean Value:	0.383232 [s]
Standard Deviation:	0.001794 [s]

Control Effort	
Mean Value:	1.669373
Standard Deviation:	0.003209

Phase Margin	
Mean Value:	84.20500 [deg]
Standard Deviation:	0.235579 [deg]

Fine