Control of a Hydro Power Plant

Advanced Process Control – Assignment 2024

Professor: Francesco Casella
Group: Davide Jarik de Rosa
Lorenzo Pagano
Francesco Zanella

Index

Point 1

Run open-loop simulations of the response of the system to step changes of the nozzle discharge surface An. The mass flow rate w(x,t) and the pressure p(x,t) are discretized in N finite volumes, from the inlet to the outlet. Observe the flow and pressure wave propagations and reflections during the simulation, and try to explain them qualitatively and quantitatively, based on the theoretical results of wave propagation in a pipe. Also observe the simulated behavior with different numbers of volumes N, comparing them to the behavior expected from the theory.

In order to obtain the analytical solution of pressure and flow rate variations in the 1D model, it is necessary to consider and apply properly the boundary conditions. In the scenario taken into account, the boundary conditions have been used as follows:

- Fixed pressure at the Inlet ($x = 0$)

$$\Delta p(0,s) = 0 = \frac{c}{A}k_1(s) - \frac{c}{A}k_2(s)$$

(therefore, guaranteeing that $k_1 = k_2$)

- Presence of a Nozzle at the Outlet ($x = 1200$)

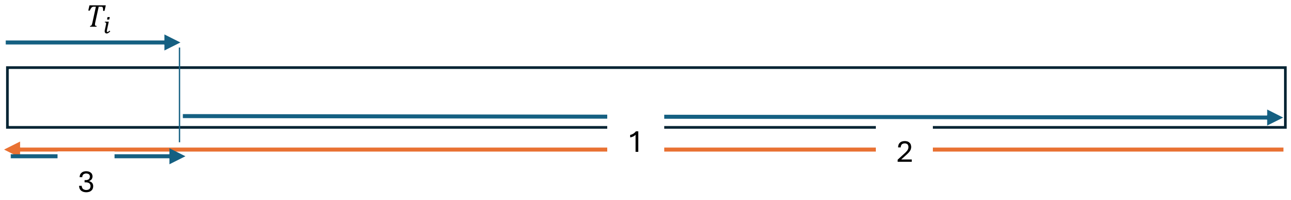
$$\begin{cases} \Delta p(L,s) = \frac{c}{A}k(s)e^{-\frac{s}{c}L} - \frac{c}{A}k(s)e^{+\frac{s}{c}L} \\ \Delta w(L,s) = k(s)e^{-\frac{s}{c}L} + k(s)e^{+\frac{s}{c}L} \\ \Delta w(L,s) = \frac{\overline{w}}{\bar{A}_N}\Delta A_N(s) + \frac{1}{R_N}\Delta p(L,s) \end{cases}$$

With $\Delta A_N(s) = -\frac{0{,}029}{s}$ given that we are dealing with a step change.

The result obtained is the following:

$$k_1 = k_2 = -\frac{4002{,}5}{s}\frac{1}{0{,}614e^{-0{,}908s} + 1{,}386e^{0{,}908s}}$$

The “discrete” time variations of these 2 functions start to be relevant after approximately 1.8 seconds. An explanation of this phenomenon may lie in the time required by the waves (forward or backward waves) to perform a whole path through the pipe. The waves, specifically, travel through the entire pipe length, get reflected backward (or forward), reach the other end of the pipe, get reflected forward (or backward) once again, and finally affect the pressure and flow rate of the section considered in the first place. In other words, the wave needs to travel through the pipe twice before affecting the initial section (while being considered as a wave of the same type).

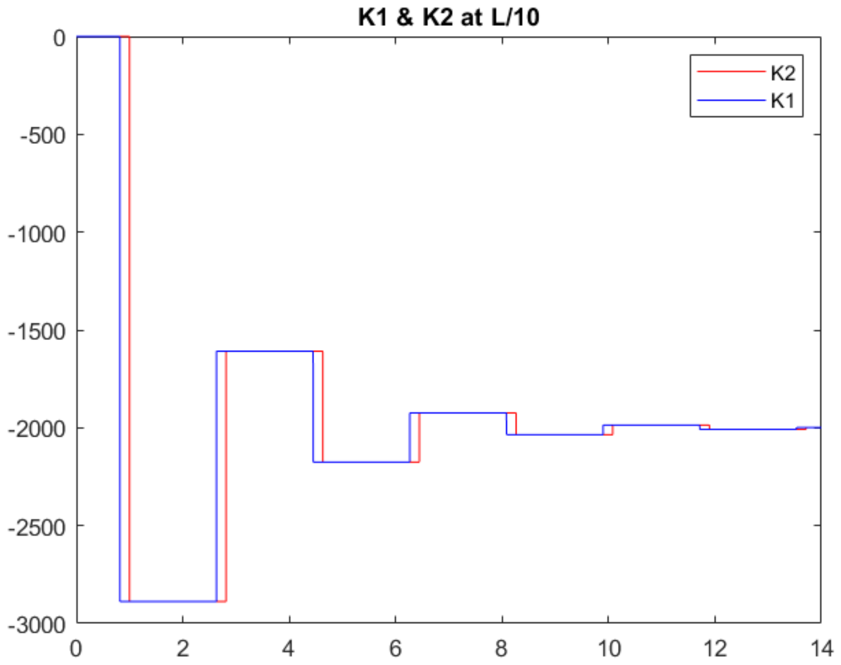


T_i = time required to affect the pressure and flow rate at first.

Blue Arrows The wave is travelling as a Forward Wave.

Orange Arrows The wave is travelling as a Backward Wave.

k_1 takes into account only the forward waves, if the wave is going backwards, it will not cause a change in amplitude for k_1 (but it will cause a change in k_2 instead).



The “exact result” obtained from the computation of $k_1$and k_2 is coherent with the simulations. In particular, the presence of a pole in the origin in the Laplace transform of k guarantees that the steady state value will be reached in finite time, regardless of the effects imposed by the waves. This can be evinced as well from the results obtained in the Modelica simulation.

$$k = k_1 = k_2 = -\frac{4002{,}5}{s}\frac{1}{0{,}614e^{-0{,}908s} + 1{,}386e^{0{,}908s}}$$

Pressure

Regarding **pressure**:

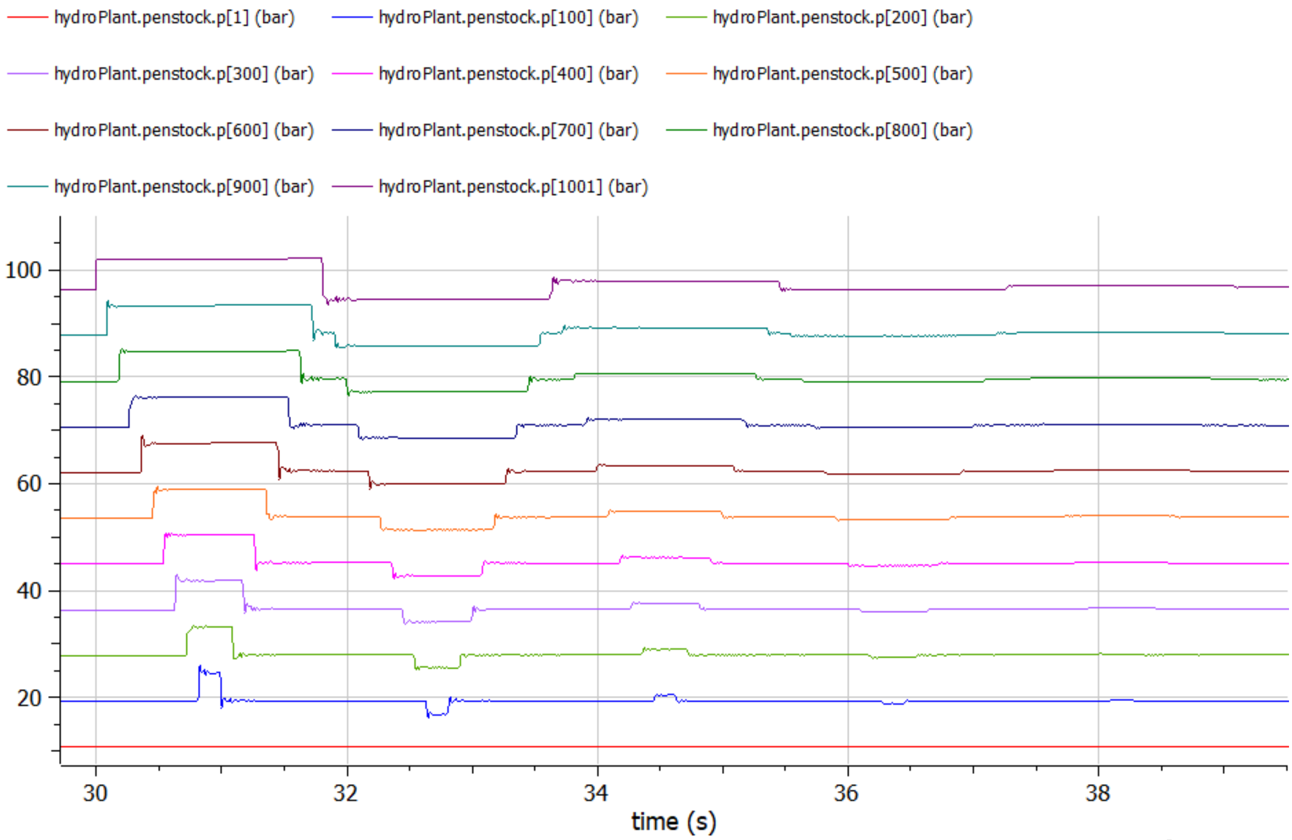
Only considering the represented values, the effect of the waves lasts the least at $1/10^{th}$ of the length of the pipe; this can be explained by its closeness to the fixed pressure boundary condition: here the effect of the forward waves is quickly compensated by the backward waves. This is perfectly in line with the theory since the equation of the pressure variations in the 1D model is the following:

$$\Delta p = \frac{c}{A}k_1\left(t - \frac{x}{c}\right) - \frac{c}{A}k_2\left(t + \frac{x}{c}\right)$$

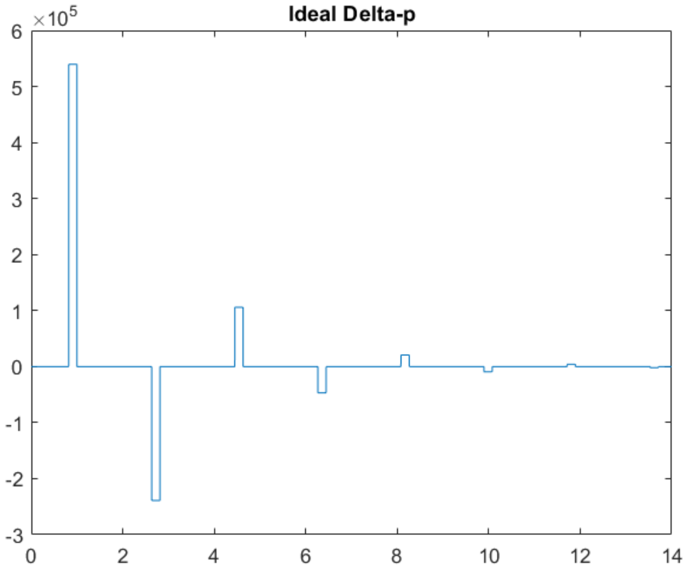
Where, as we can see, the forward and backward waves have the same amplitude but opposite signs and are applied at different times. In particular, the forward waves are applied at time $T - \frac{(L/10)}{c}$ while the backward waves at time $T + \frac{L/10}{c}$, with $T = \frac{L}{c}$. Therefore, the difference between the 2 times (which determines for how long the wave’s effects are visible in the graph) is the lowest achievable (lasting $2 \cdot \frac{L/10}{c}$).

Following this logic, it is possible to also understand the behavior of the pressure at the other ‘snapshots’. Particularly relevant is the pressure at the end of the pipe, where the forward wave effect is instantaneous (given that $T - \frac{L}{c}$ =0 by definition), while the backward wave influences the pressure only after $2T$.

The last comment on the pressure must refer to the increasing average value observable while moving towards the end of the pipe: this effect is caused by the gravitational term in the Bernoulli equation, given the inclination of the pipe.
(Regarding the intervals in which the pressure is at lower values than the steady state this is caused by the greater amplitude of the backward waves with respect to the forward waves.)



The following is the exact analysis obtained from the computation of k_1 and k_2 at $x = \frac{L}{10}$(please do note that unlike in the case of the Modelica analysis, the results refer to the variations and not to the overall values).



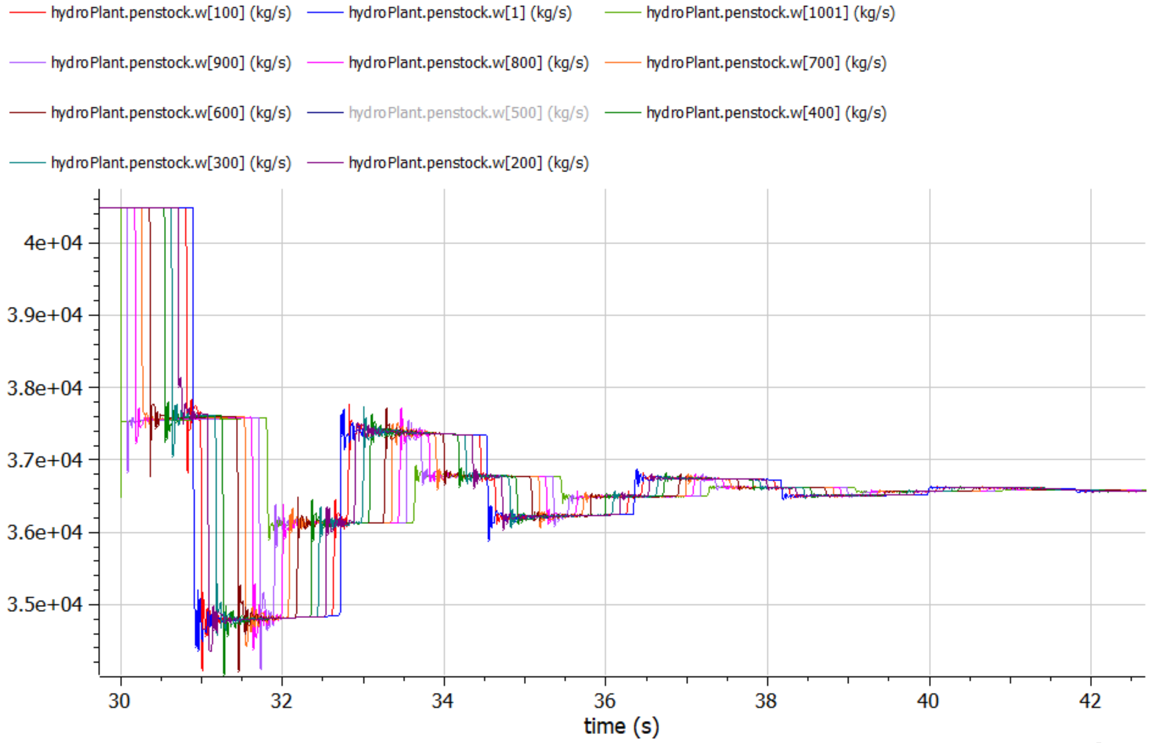
Flow rate

Regarding the **flow rate**:

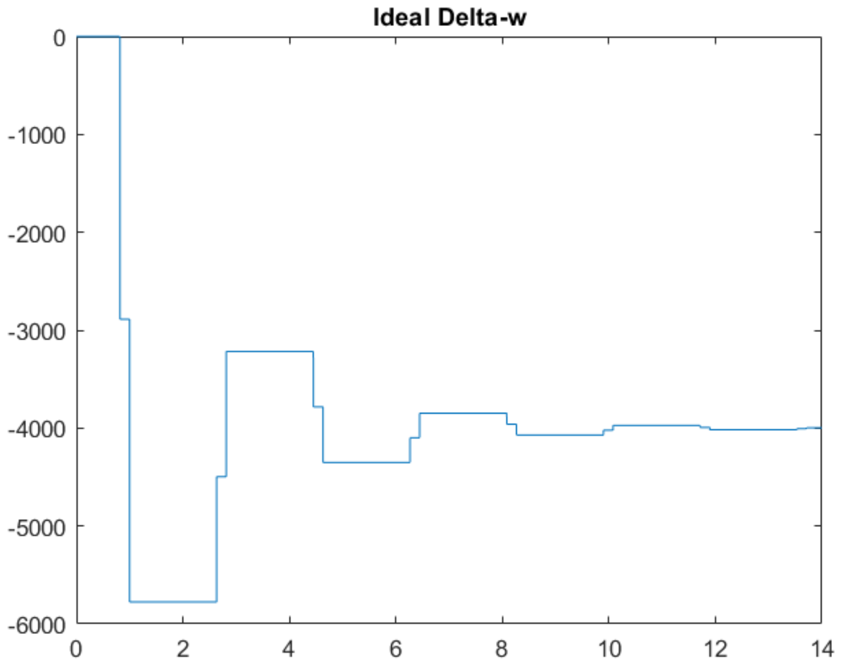
$$\Delta w = k_1\left(t - \frac{x}{c}\right) + k_2\left(t + \frac{x}{c}\right)$$

The flow rate variations can be better understood from an analysis of k_1 and k_2, given that Δw is just the summary of the 2 functions and there is no 'destructive' effect between backward and forward waves.

Exactly as it has been observed in the pressure variations, the wave effect at the outlet is instantaneous, meanwhile at the inlet a change can be recorded after T seconds.



The following is the exact analysis obtained from the computation of k_1 and k_2 at $x = \frac{L}{10}$(please note that, unlike in the case of Modelica analysis, the results refer to the variations and not the overall values)



Varying **N**:
Based on the number of volumes in which the pipe has been divided, the result will be more or less precise. Ideally, with infinitesimal dV volumes, we could fully represent the dependency on time and position of the pressure and flow rate variations (the best we could achieve is $N = 1000$). This would provide several snapshots (fixed in position) where it would be possible to observe a singular volume variation in time.
The only relevant difference is the precision of the computations, and the frequency of the oscillations, which are the artefacts to the necessary spatial discretization (the simulations realized at half-length, with $N = 200$ and $N = 10$ respectively, are reported below).

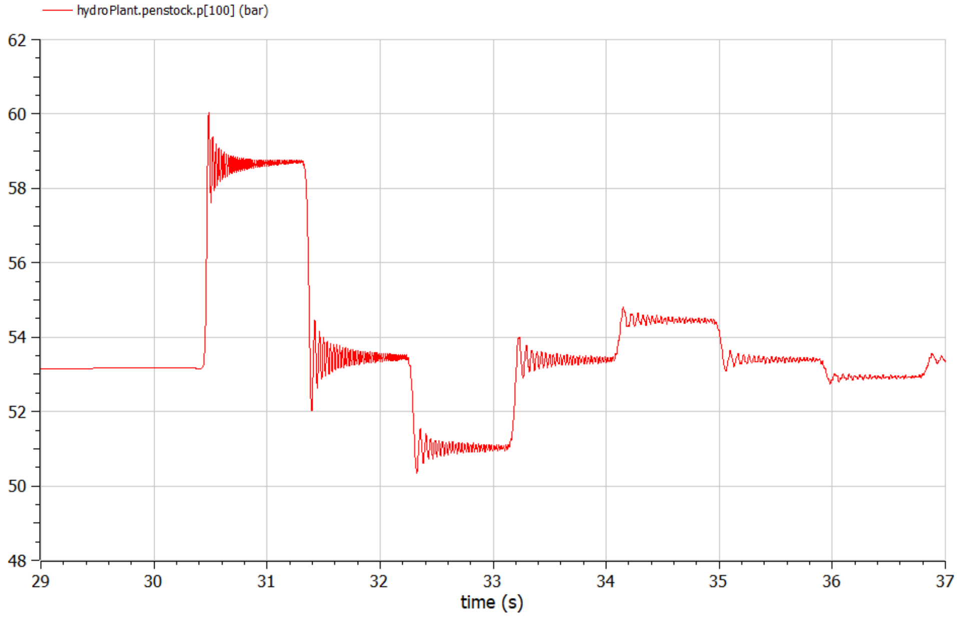


Figure 1: Simulation with $N = 200$

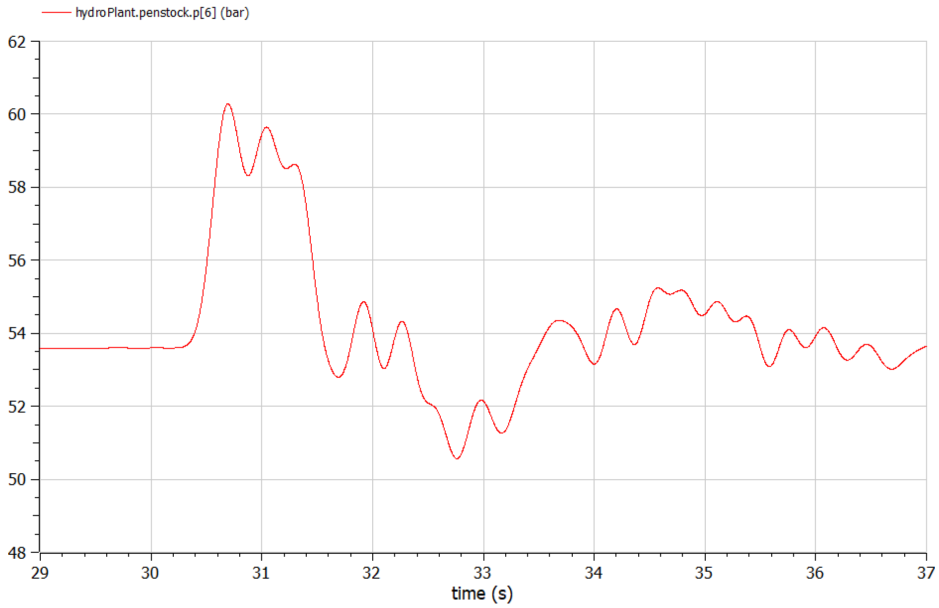


Figure 2: Simulation with $N = 10$

Script for MATLAB plotting: Appendix 1

Point 2

Design a PI frequency controller which is as fast as possible, with at least 50° phase margin, using the approximated transfer function obtained by neglecting the compressibility and friction effects.

In order to compute a suitable Regulator for the 1D model, one can work on the approximated model obtained in a 0D setting. Later, a regulator can be designed on the following Transfer Functions:

$\delta P_{hyd} = \frac{1-2Ts}{1+Ts}\delta A_N$ Modelling the hydraulic components

$\delta\omega = \frac{1}{\alpha+T_A s}\delta P_{hyd}$ Modelling the electric components

$\delta A_N = \frac{1}{1+T_a s}\delta A_N^o$ Modelling the actuator

$R(s) = K_p\frac{1+T_i s}{T_i s}$ PI regulator

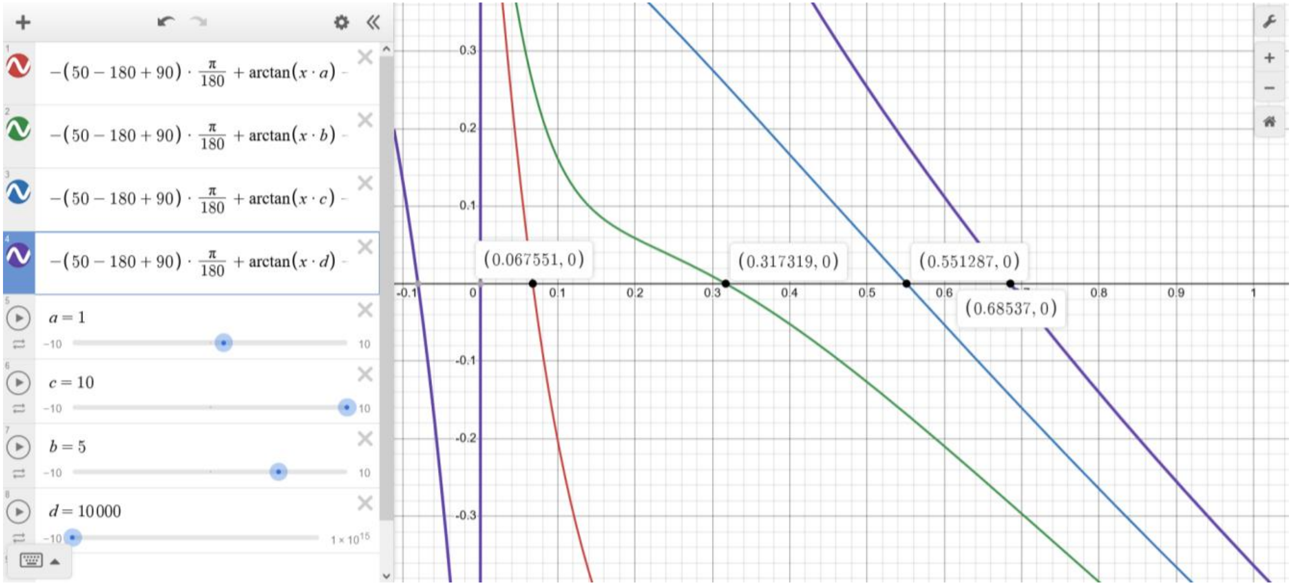
This means that, in order to achieve a minimum phase margin ($\varphi_m \geq 50°$), the constraints one needs to apply are the following

$$50° \leq 180° - 90° - \operatorname{atan}(2T\omega_c) - \operatorname{atan}(T\omega_c) - \operatorname{atan}(T_a\omega_c) - \operatorname{atan}\left(\frac{T_A\omega_c}{\alpha}\right) + \operatorname{atan}\,(T_i\omega_c)$$

4 different approaches may now be taken into consideration, every approach will lead to different degrees of quality in terms of results.

1. Highest achievable cutoff frequency

From an analysis of the previous inequality, 2 degrees of freedom may be identified. One could freely set either T_i or ω_c and compute the remaining variable from the phase margin constraint. Therefore, a theoretically viable option would be imposing the maximum achievable cutoff frequency by choosing an appropriate integral time. This mathematical approach brings forth an interesting result.



1 $-(50-180+90)\cdot\frac{\pi}{180}+\arctan(x\cdot a)$ -

2 $-(50-180+90)\cdot\frac{\pi}{180}+\arctan(x\cdot b)$ -

3 $-(50-180+90)\cdot\frac{\pi}{180}+\arctan(x\cdot c)$ -

4 $-(50-180+90)\cdot\frac{\pi}{180}+\arctan(x\cdot d)$ -

5 $a = 1$ (−10 to 10)

6 $c = 10$ (−10 to 10)

7 $b = 5$ (−10 to 10)

8 $d = 10000$ (−10 to 1×10^{15})

As observable in the image, increasing values of T_i generate greater and greater values of ω_{cmax} until an asymptotical limit value ($\omega_{cMAX} \cong 0.685485\ rad/s$).

$\omega_{cMAX} = 0.685485\ \frac{rad}{s}$ has been obtained with a chosen $T_i = 10000$ s. Despite being an exaggerated result, its relevance remains. Considering a smaller T_i (e.g. 1000 s), the resulting ω_{cmax} would be equal to 0.684335 rad/s.
The real problem brought by this approach is the resulting time required to reject disturbances, which unfortunately is the main goal of this controller. By applying an integrator with such a significant time constant, the poles of the sensitivity transfer function will be completely dominated, resulting in a settling time $t_s = 5T_i$. This would undoubtedly be an unacceptable result, even for $T_i = 100\ s$ ($\omega_{cmax} = 0.67385\ \frac{rad}{s}$).

2. Limitations on the difference between 0D and 1D model

Focusing on the discrepancies between the 0D and 1D model, it is possible to identify the maximum cutoff frequency which minimizes the differences between models (in terms of modulus). The goal of achieving a variation smaller than 0.01 would require a limitation on the cutoff frequency at 0.5570 rad/s (the difference in terms of phase margin is about 2 to 4 degrees). This results in a maximum integral time constant of $T_i = 10.3721\ s$.

3. Reduction of the settling time for disturbance rejection

Considering that the aim of control for Hydro Power Plants is mainly disturbances rejection (which represents the changes in the Grid Power Demand), a viable alternative for the design of the controller could be the reduction of the settling time given step changes in disturbances. Therefore, the analysis of the step response could be shifted to the transfer function

$$\frac{\delta\omega}{\delta P_{el}} = \frac{-1}{\alpha + sT_A} S(s)$$

which relates the frequency to the Power Load. We can immediately exclude the values of T_i greater than $12s$ given that they would end up dominating the dynamics of $S(s)$ and slow down the response to step changes. We should therefore focus on values below $12s$ (even $12s$ is a good choice, but not the best in terms of settling time) that allow also us to keep the integrator around 1 decade before the other eigenvalues. If the last condition does not hold, the surge of oscillations will end up ruining the settling time. By means of trial and error, the best result (while operating with a 0D model) has been found for $T_i = 5.12\ s$, which guarantees a settling time of about 15 seconds. Unfortunately, these results do not work properly in the 1D model, where oscillations can be already perceived for $T_i = 9s$. In this case, the fastest settling time is achieved with $T_i = 10\ s$.

4. Balanced results (seen in class)

A more balanced result can be obtained by avoiding an excessive phase loss in the non-minimum phase component (maximum acceptable imposed at 50°), considering the TF:

$$G_{nmp}(s) = \frac{1 - 2sT}{1 + 2sT}$$
$$\angle G_{nmp}(j\omega_c) = -\operatorname{atan}(2\omega_c T) - \operatorname{atan}(2\omega_c T) \geq -50°$$
$$\operatorname{atan}(2\omega_c T) \leq 25°$$
$$2\omega_c T \leq \tan(25°)$$

The cutoff frequency is forced to be lower than

$$\omega_c = \frac{0{,}22}{T}\ rad/s$$

This results in an integral time constant of about $T_i = 21.5110\ s$ (which we could approximate to $T_i = 20\ s$) and a settling time of around 105 seconds, an acceptable result.

$$T_i = 20\ s$$

With this value in mind the previous inequality yields $\omega_{cmax} = 0{,}6255\ rad/s$ as the maximum achievable cutoff frequency, which forces K_p to satisfy the following equation:

$$K_p \left| \frac{1 + T_i j\omega_c}{T_i j\omega_c} \frac{1 - 2Tj\omega_c}{1 + Tj\omega_c} \frac{1}{\alpha + T_A j\omega_c} \right| = 1$$

$$K_p = 7.1334$$

Important note: these same results could have been achieved through approximations that would have significantly reduced the complexity of the computations with little or no loss in the quality of the results (e.g. the higher frequency poles could have been neglected). Despite this, with the availability of software-based support, a more precise solution has been opted for.

The resulting Regulator transfer function is:

$$R(s) = 7.1334 \frac{1 + 20s}{20s}$$

Script used for computations: Appendix 2

H -> transfer function of the hydraulic component (la diga)

A = pipe surface (sezione tubo)

w = superfice del tubo

c = speed of sound

Rn = 2*ro*g*h/An - resistence of the nozzle

Point 3

Rp = resistance of the pipe

Check that the crossover frequency and phase margin obtained with the more accurate transfer function considering the 1D distributed compressibility and inertia and the controller designed at point 2. are not too different from those obtained with the incompressible fluid model for the process. You can use Matlab, Python, or even the Complex Modelica library to compute the modulus and phase of the frequency response of the 1D transfer function.

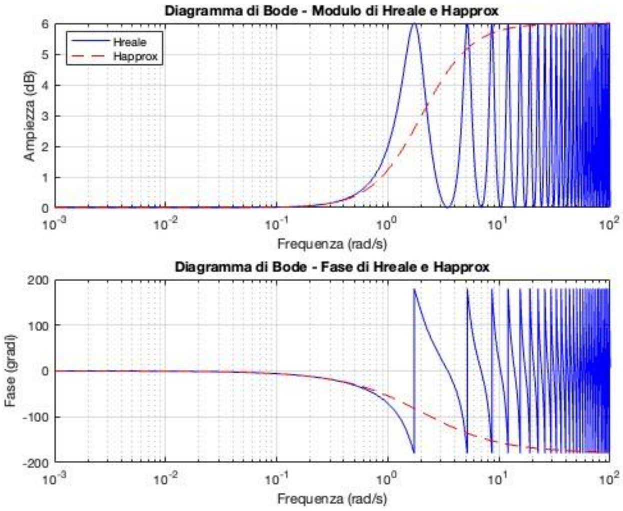
Using the following Transfer Function for the Hydraulic component of the System, it is possible to evaluate and compare the 0D and 1D models:

$$H_{approx}(s) = \frac{1-2Ts}{1+Ts} \qquad H_{real}(s) = \frac{1-2\,\beta\tanh(s\,\tau)}{1+\beta\tanh(s\,\tau)}$$

Where:

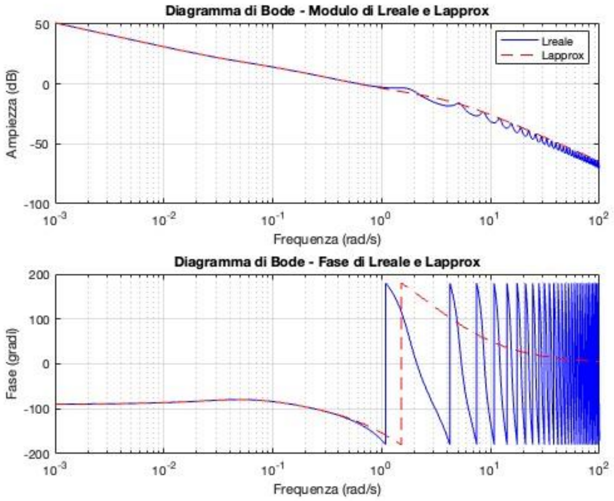
$$\beta = \frac{c}{A(R_n+R_p)} \qquad \tau = \frac{L}{c} \qquad R_p = \overline{w}\,\frac{c_f\,\omega\,L}{A^3\,\rho}$$

Notice that $R_p = 6.4894 \ll R_n = 484.0795$, the effect of R_p can be therefore considered negligible.



Alla cutoff frequency sono sovrapposte!!

In terms of phase and magnitude, the bode diagrams are practically identical at low frequencies. This is valid until the cutoff frequency: the approximations hold correctly in the range of frequencies of interest.



For higher frequencies, the effect of nonlinearities is reduced due to the structure of the sensitivity transfer function, which causes its gain to decrease to 0dB.

Therefore, the bandwidth in which the differences are relevant is rather limited.

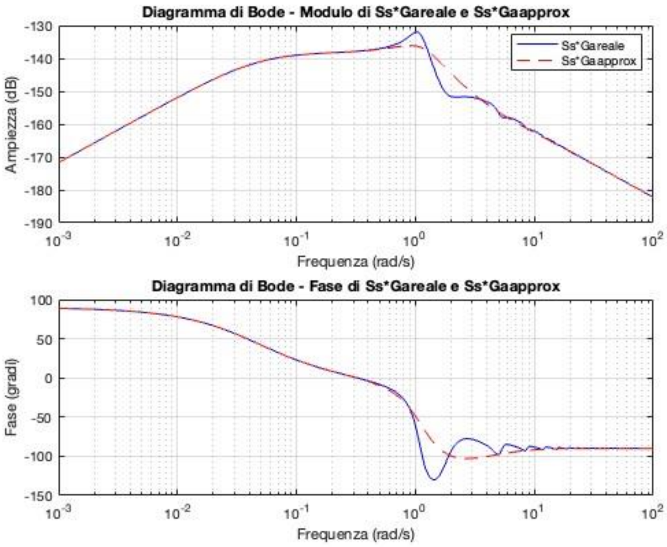


Diagramma di Bode - Modulo di Ss*Gareale e Ss*Gaapprox

Diagramma di Bode - Fase di Ss*Gareale e Ss*Gaapprox

For the nature of Ss the main differences are places in small frequency ranges!

Ss * -1/(alpha*sa)

Nel mezzo c'è il closed loop

Point 4

Implement the designed controller in the Modelica simulator. Please note that the simulator uses physical variables as controller inputs and outputs, not normalized deviations around the reference steady state, so you need to properly de-normalize all the gains. An offset is also added to the controller output in the Modelica model of the controller, so that the dynamic part of the controller can be initialized with zero output if the offset corresponds to the steady-state value.

The process of denormalization of the 0D model is fully described below. Considering the equation of the normalized transfer function

Denormalization!

$$\frac{\delta P_{hyd}}{\delta A_n} = \frac{1-2sT}{1+sT}$$

And knowing that

$$\delta * = \frac{\Delta *}{\overline{*}}$$

The following is obtained

$$\frac{\Delta P_{hyd}}{\Delta A_n} = \frac{1-2sT}{1+sT} \cdot \frac{\overline{P_{hyd}}}{\overline{A_n}}$$

The operation must be repeated for every Transfer Function present in the overall system, thus obtaining

$$\frac{\Delta\omega}{\Delta A_n} = G(s)\frac{\overline{\omega}}{\overline{A_n}} = \frac{1}{1+sT_a}\frac{1-2sT}{1+sT}\frac{1}{\alpha+sT_A}\frac{\overline{\omega}}{\overline{A_n}}$$

As the denormalization of

$$\frac{\delta\omega}{\delta A_n} = G(s) = \frac{1}{1+sT_a}\frac{1-2sT}{1+sT}\frac{1}{\alpha+sT_A}$$

Taking into account the definition of the closed loop Transfer Function $L(s)$

With the procedudure we got:

$$L(s) = K_p\frac{1+sT_i}{sT_i}\frac{1}{1+sT_a}\frac{1-2sT}{1+sT}\frac{1}{\alpha+sT_A}$$

It is trivial that, in order to respect the requirements also in the denormalized setting, the denormalization of the controller must be obtained as follows

$K_p = 7.1176$ (designed on the normalized system with $T_i = 20$)

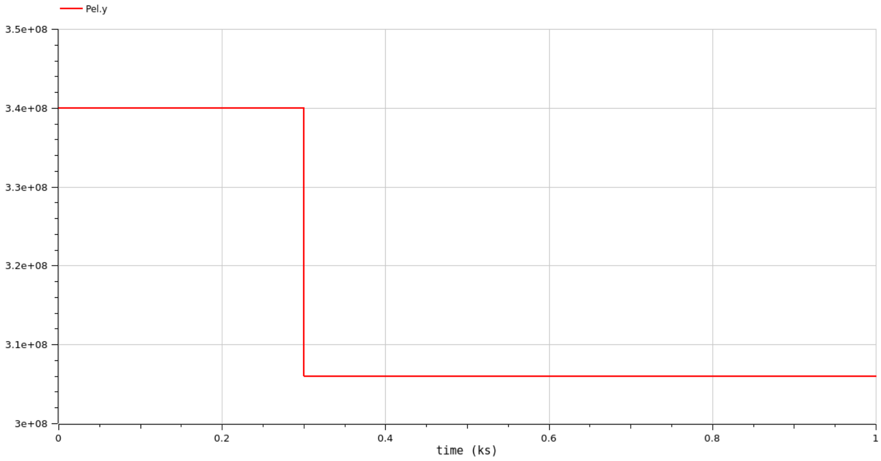
$$K_{P_{denorm}} = K_P\frac{\overline{A_n}}{\overline{\omega}} = 0.00662$$

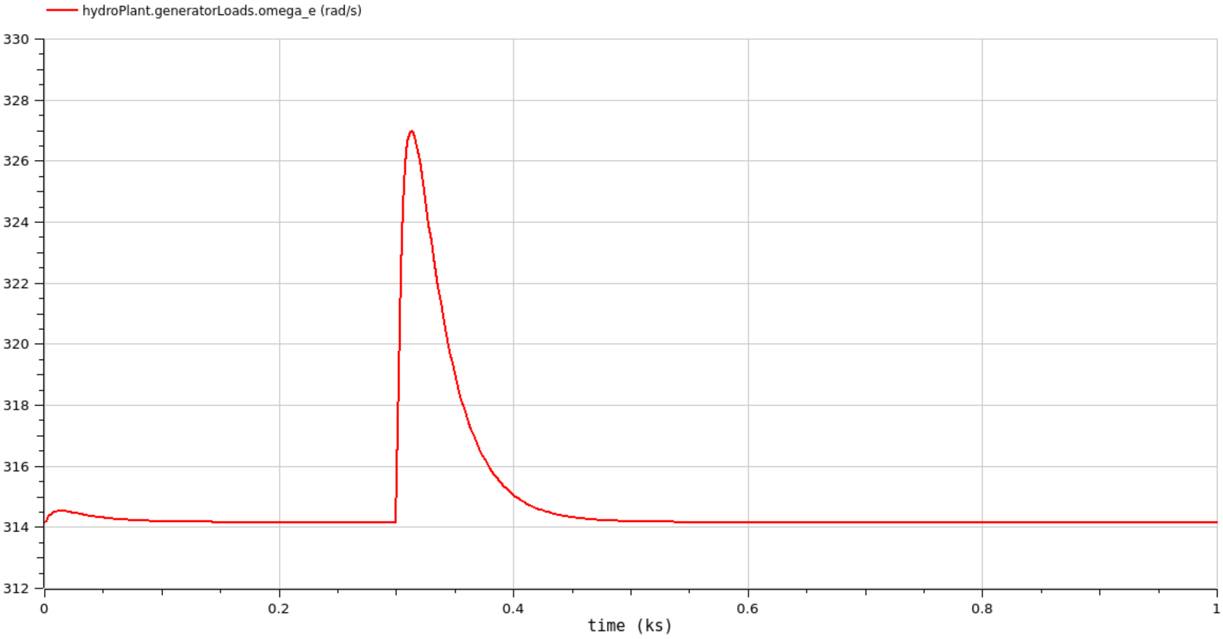
To mantain the results, the cutoff frequency we gotta denormalize the Kp as well!

Point 5

Simulate the response of the closed-loop system to a step change of -10% of the reference value of the active power consumed by the loads, starting from full load. Make sure you apply the step only when the system has reached a steady state, to avoid spoiling the results with the effect of initial conditions not corresponding to equilibrium. Discuss the obtained response with respect to the results you would expect based on the analysis carried out on the simplified linearized model, drawing some conclusions on the appropriateness of using that simplified model for control design.

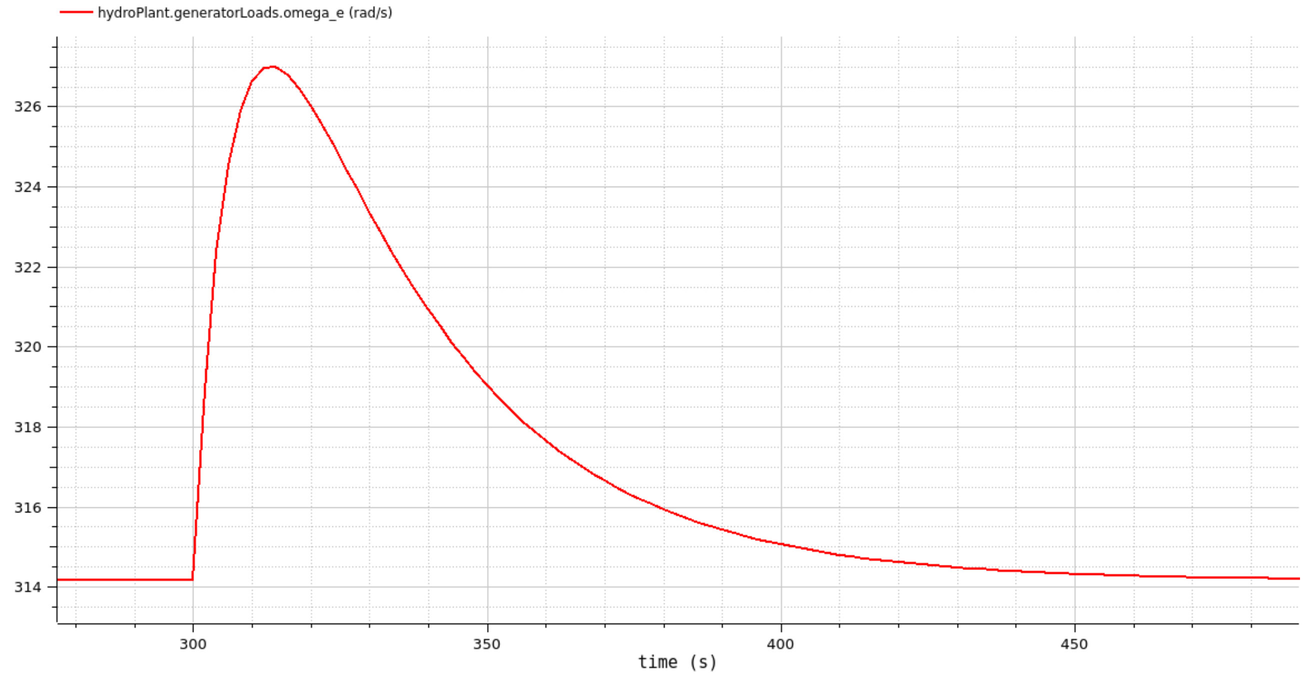
Please note that the step change has been applied at 300s. This was necessary in order to ensure that the system was in a steady state condition (in this case, due to the difference between the initial and nominal value of the nozzle discharge surface A_n, it was indeed a requirement).



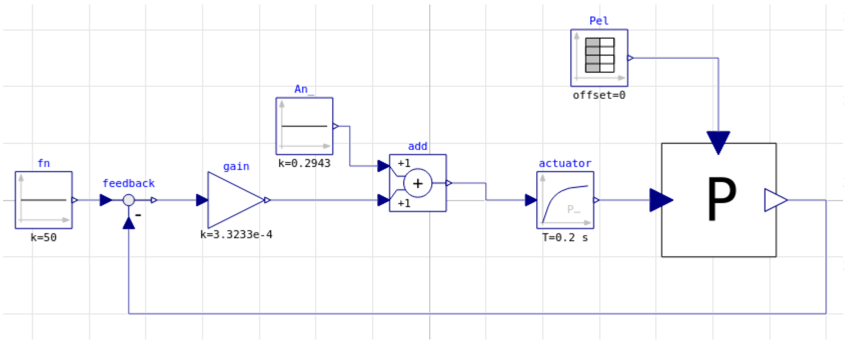


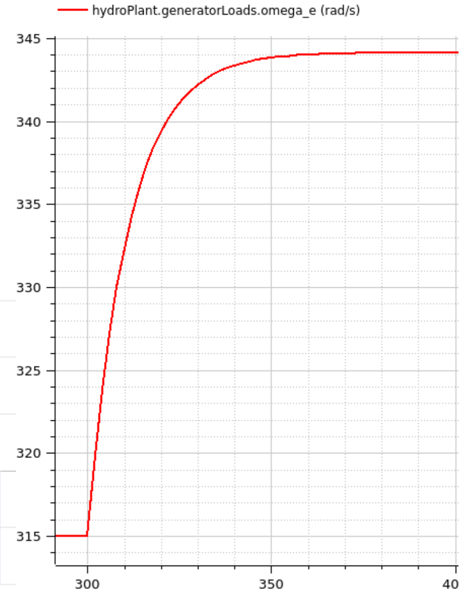
For what concerns the comparison with the approximated system:

The signal initially presents a proportional trend with a peak value of 327 rad/s, afterwards the intervention of the integral action may be appreciated, which brings the frequency back to the reference in a settling time of approximately $100s$. The result is obtained in line with the study performed on the 0D model, where the dominant pole is introduced by the controller, thus leading to an expected settling time of $T_s \cong 5 \cdot T_i = 100$ s.



To further underline the Proportional effect in the PI controller, the same simulation has been carried out with a purely Proportional controller.





From the 0D model analysis, the expected behavior of the step response can be approximated as the superimposition of the step response of two different transfer functions. These transfer functions may be found by means of the application of the Heaviside function on the approximation of the closed loop transfer function $\frac{\delta\omega}{\delta P_{el}}$

Approx of S(s), needed to divide in two components the others function

$$S(s) = \frac{T_i}{\omega_c T_A} \frac{s(1 + sT_A)}{(1 + sT_i)(1 + \frac{s}{\omega_c})}$$

$$\frac{\delta\omega}{\delta P_{el}} = -\frac{1}{(1 + sT_A)} \cdot S(s)$$

Please note that this approximation of $S(s)$ is not extremely accurate (it simplifies the dynamics of the closed loop transfer function to a 1st order system). Despite this, the final result is nevertheless acceptable considering the simulation goal.

Hslow is the blue; Hfast is the red

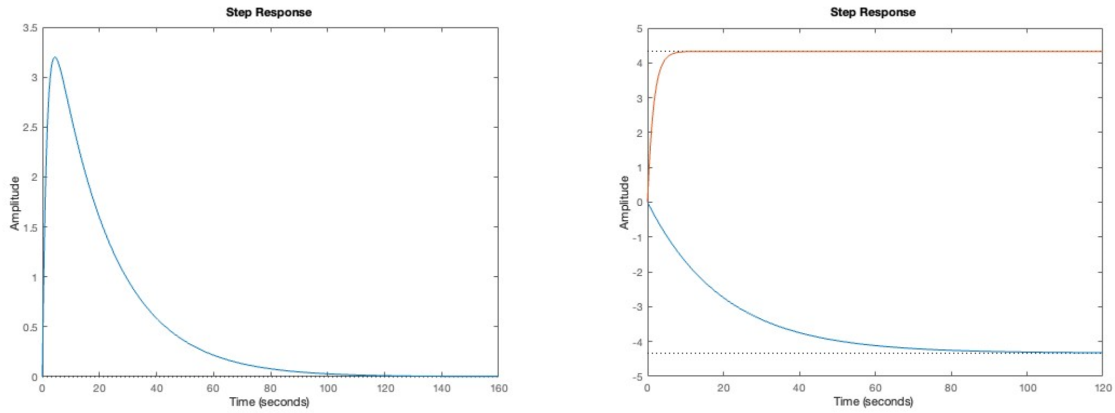
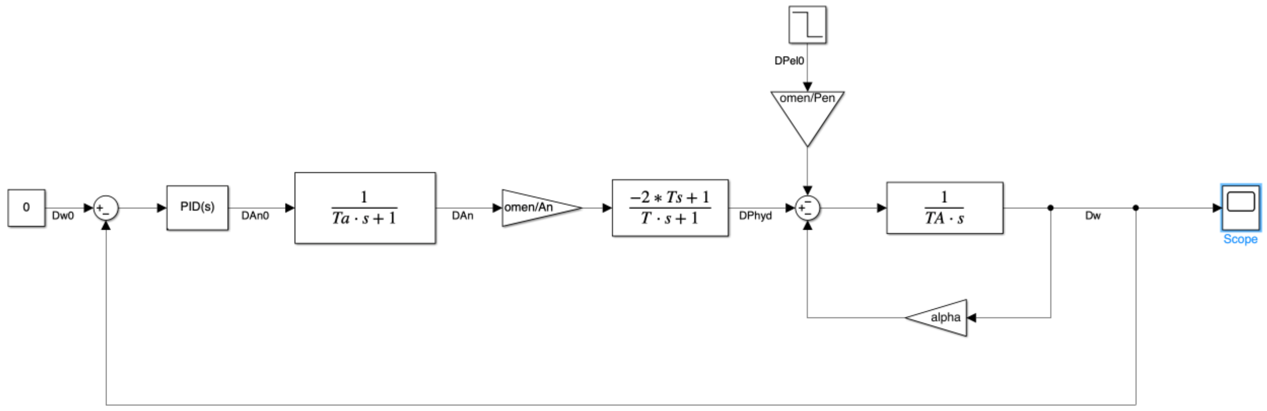
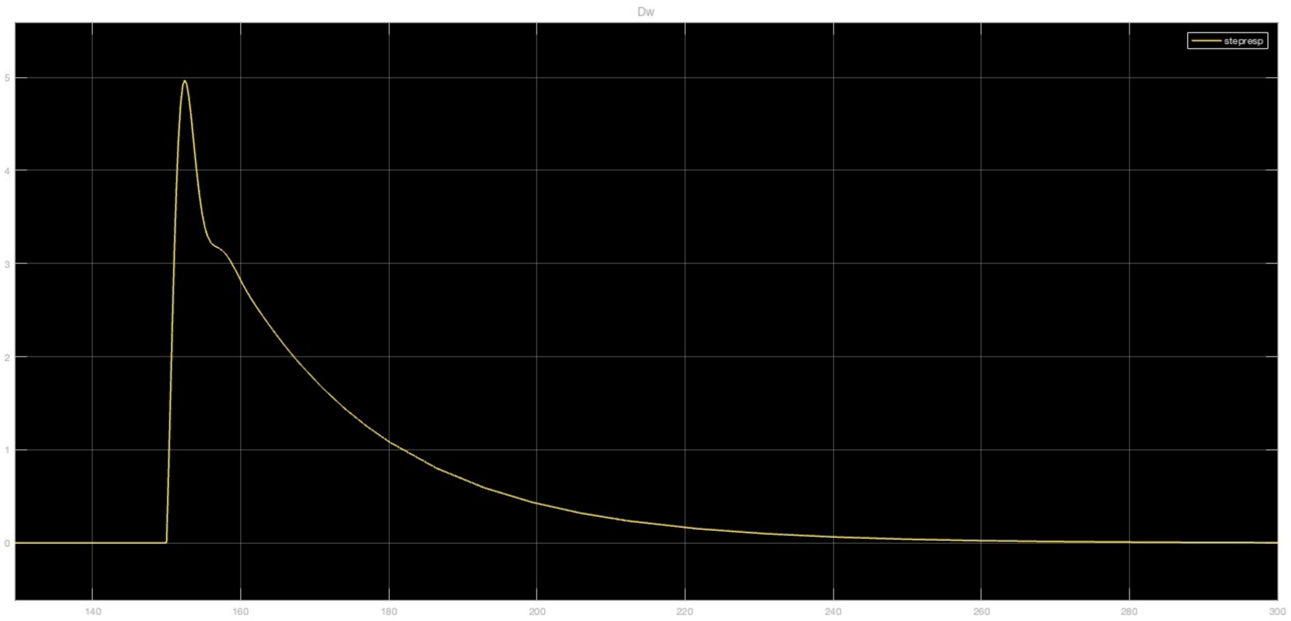


Figure 3: Computation in Appendix 4

$HS_{slow} = \frac{4.3310}{20s+1}$ $\qquad$ $HS_{fast} = \frac{-4.3310}{1.6s+1}$ $\qquad$ Hslow + Hfast

Reported below are the results of a more accurate Simulink simulation (Appendix 7**Errore. L'origine riferimento non è stata trovata.**):



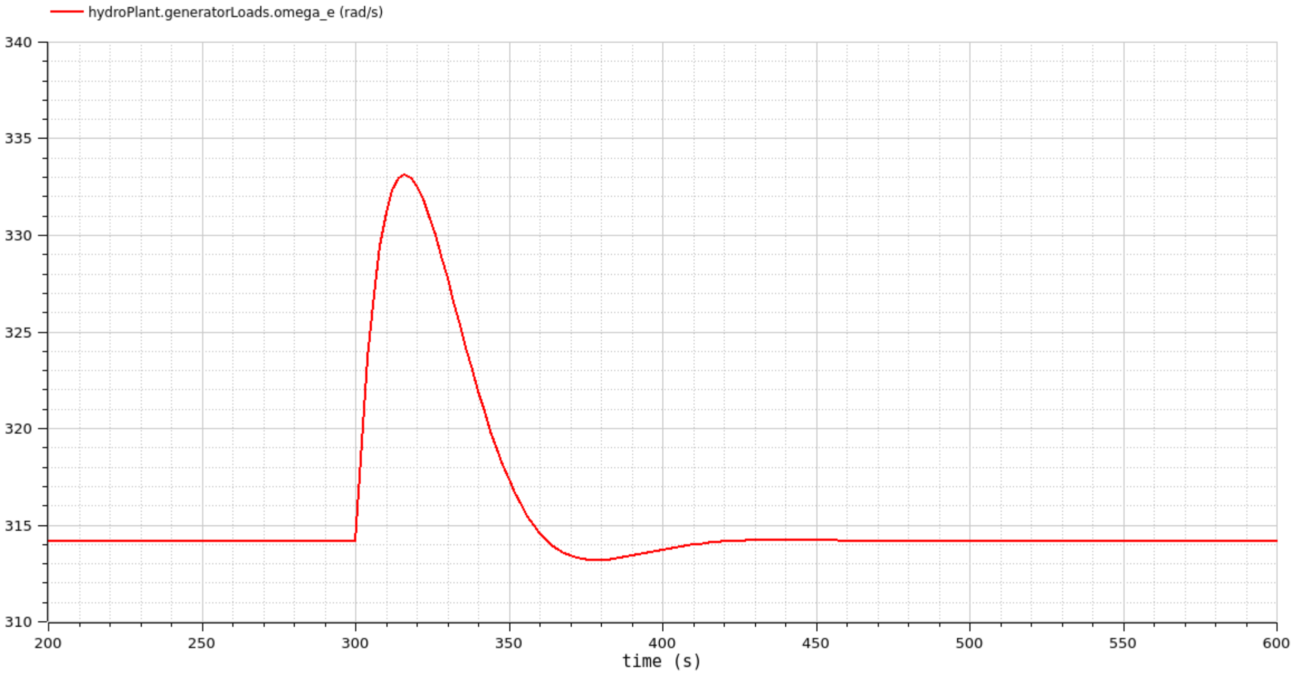


Point 6

Repeat the analysis of point 5. with $\alpha = 0.0$ (all loads driven by inverters) and $\alpha = 1.5$ (larger share of directly connected motors with torque proportional to the frequency). Compare the obtained results and try to justify them based on the simplified model of the control system.

- $\alpha = 0$ -> reduction in the phase margin, small oscillations (a = 0 wcs -> pole in the origin, loss of 90°

With this new α, a phase margin reduction of about $7.6°$ with respect to the original system may be observed. This difference is caused by the introduction of a pole in the origin, which corresponds to a loss of phase margin of 90°, a higher loss compared to the previous one of $82.4°$ (due to the pole in $\frac{1}{12}\ rad/s$). Considering that the controller has been designed with a phase margin $\varphi_m = 50°$ (a limit value on its own), this phase margin loss justifies the presence of oscillations in the overall result:

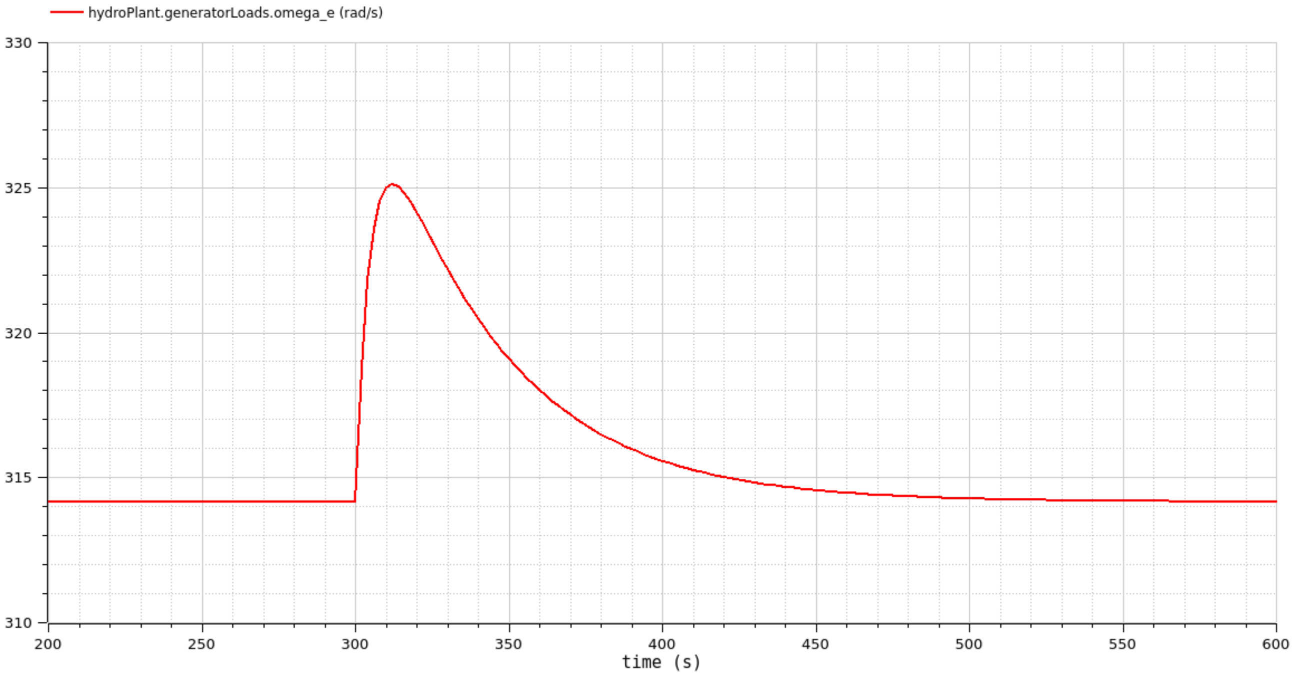


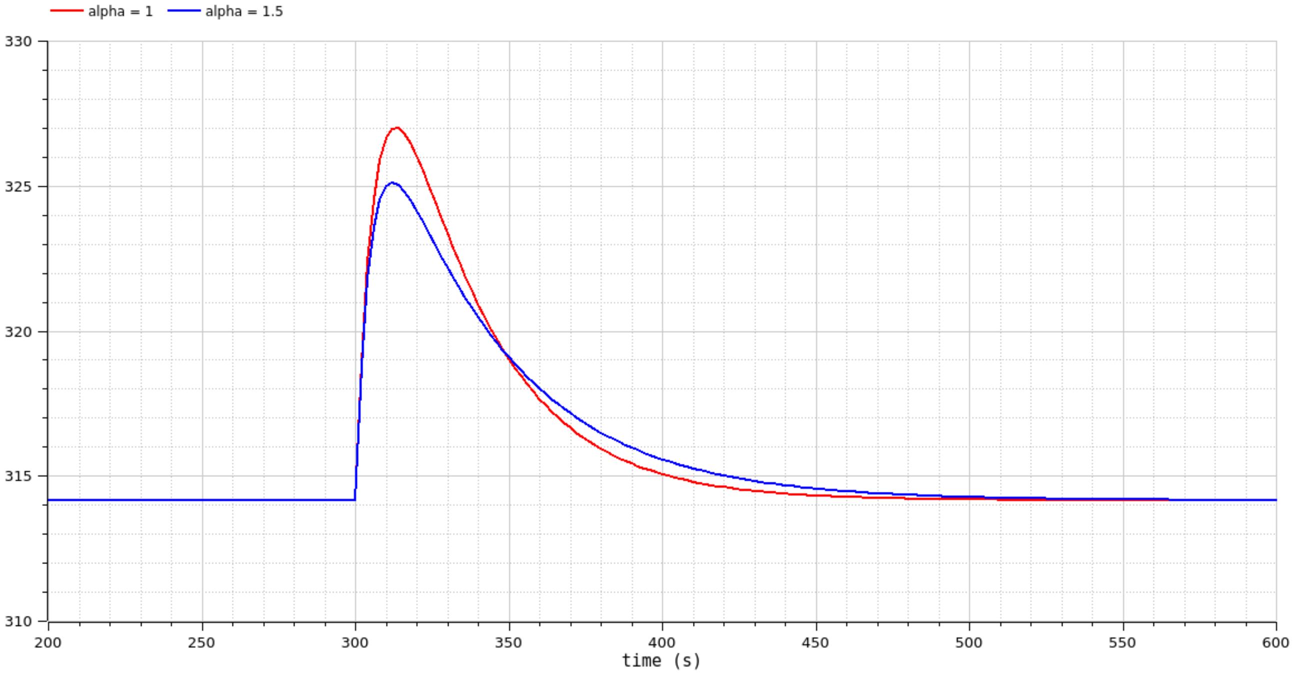
Furthermore, the peak value is increased due to the presence of oscillations.

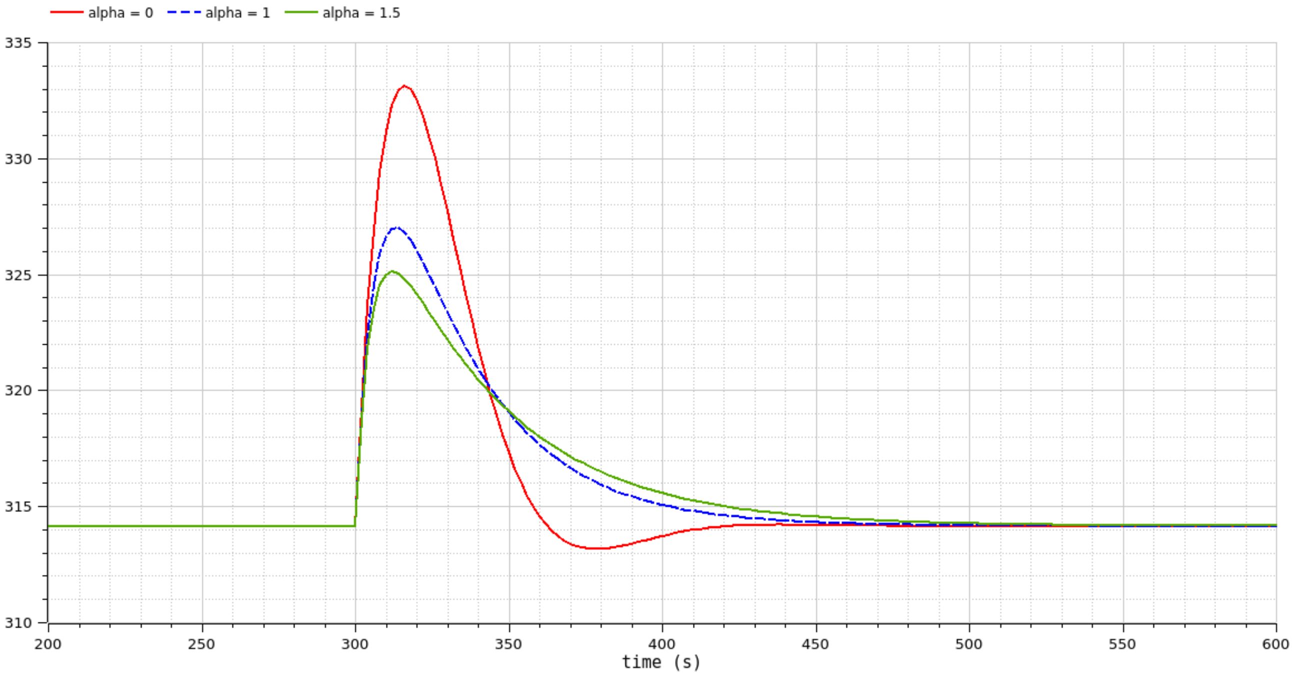
- $\alpha = 1.5$ -> the phase margin increases, we are moving the pole to the left

With this value of α, the phase margin is instead slightly increased. This increase does not constitute a significant change in the effectiveness of the designed controller: the only discernable differences in the step response are a slight reduction in the peak value (given its relationship with the gain $\mu = \frac{K_P}{T_i \cdot 1.5}$) and a negligible increase in the settling time.

These results prove the necessity of designing the controller on a system with $\alpha = 0$ (the worst-case scenario) so that the maximum robustness is guaranteed.


hydroPlant.generatorLoads.omega_e (rad/s)
330
325
320
315
310
200
250
300
350
400
450
500
550
600
time (s)


alpha = 1
alpha = 1.5
330
325
320
315
310
200
250
300
350
400
450
500
550
600
time (s)



FOR A MORE ROBUST CONTROLLER: ALPHA = 0 - WCS

100-> 90 prima
100 -> adesso

Diverso perché cambia l'operating point!

Point 7

Simulate the response of the system with $\alpha = 1.0$ to a ramp reduction of the electrical power consumed by the loads from 100% to 20% in 10 minutes, followed by a constant request for 2 minutes, and finally by a sudden increase from 20% to 30%. Compare the step response of point 5. with the step response at t = 12 minutes and try to explain why they are as they are, based on the analysis of the simplified model.

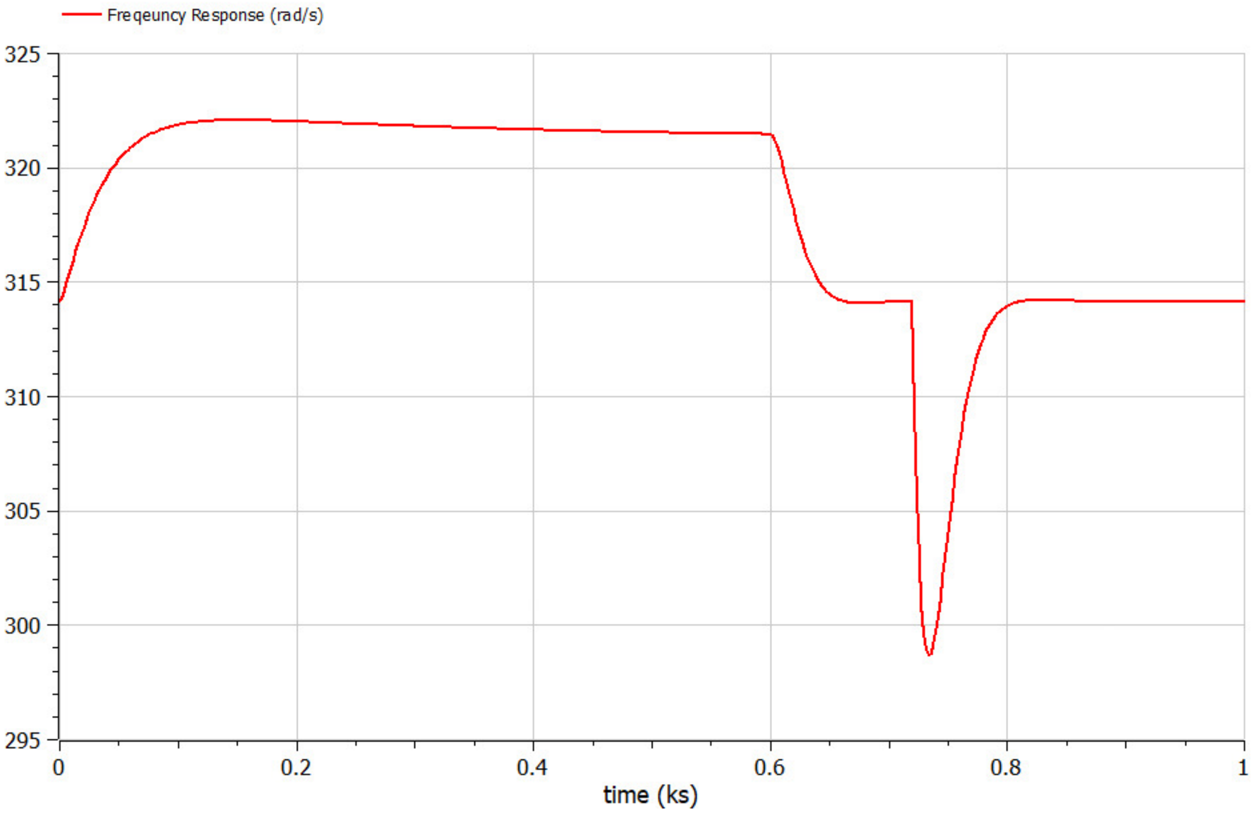


Figure 4:Modelica response to a ramp reduction of the electrical power consumed by the loads from 100% to 20% in 10 minutes, followed by a constant request for 2 minutes, and finally by a sudden increase from 20% to 30%.

The most important remark regarding this simulation is the difference between the step response applied at $t = 720\,s$ and the step response of point 5. The mismatch is mainly caused by the operating points that are different in these 2 cases: the system is strictly dependent on its operating point. More specifically, the variables dependent on the requested power are the following:

$$T \propto \overline{w} \propto \overline{P_{el}} \quad T_A \propto \frac{1}{P_{el}} \qquad \overline{A_n} \propto \overline{w} \propto \overline{P_{el}} \qquad \alpha \propto \frac{1}{P_{el}}$$

T -> parte idraulica
Ta -> turbina

w = nominal flow rate

w proporzionale a Pel è dato per assodato

T proporzionale perché l / Rn = w / 2glp = k Pel

Ta = J om^2 / Pel

An = w sqrt(pwe)

Pel è un disturbo, come alpha influenza il disturbo???

alpha = 0 --> wcs for disturbance rejection

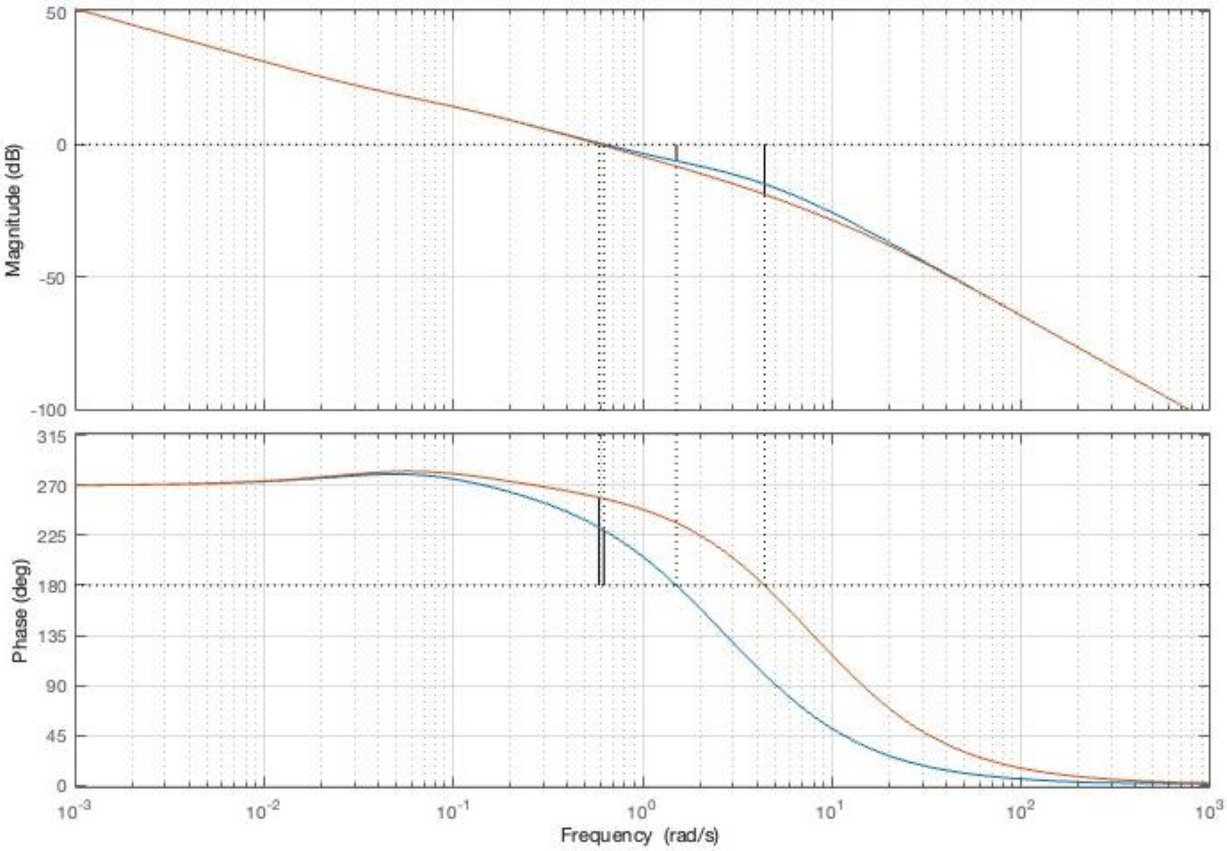


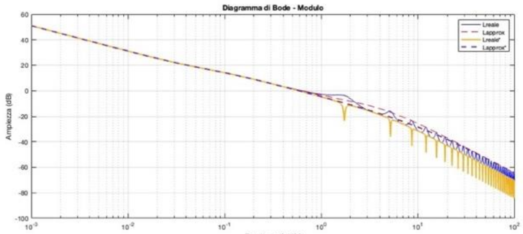
*Figure 5:comparison between $L(s)$(in blue) and $L_{new}(s)$(in orange) margins ($\varphi_m =\ 50.1161$, $\omega_c = 0.6213$, $\varphi_{m_{new}} = 79.5227$, $\omega_{c_{new}} = 0.5845$). Script in Appendix 5**Errore. L'origine riferimento non è stata trovata.***

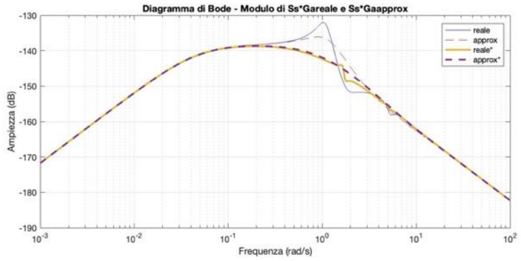
The improvement in phase margin is caused by the decrease of the time constant T, which further distances the unstable zero and the pole of the Transfer Function $\frac{1-2Ts}{1+Ts}$ from the cutoff frequency.

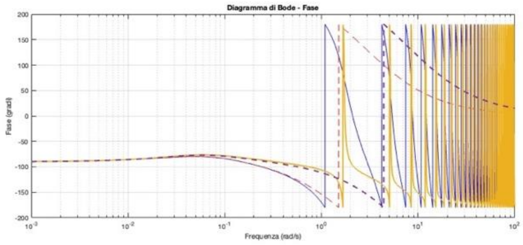
Please note that the changes in A_n, α, and T_A have no effect on the gain (they elide each other). This is consistent with the structure of the open loop Transfer Function

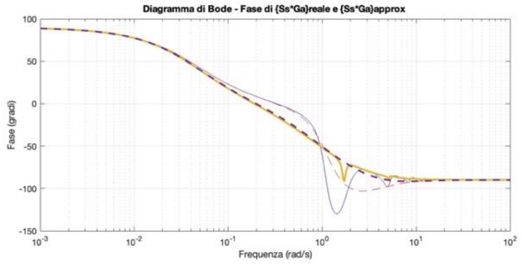
$$L(s) = \frac{\overline{\omega}}{\frac{A_n}{5}} K_p \frac{1+sT_i}{sT_i} \frac{1}{1+sT_a} \frac{1-2s\frac{T}{5}}{1+s\frac{T}{5}} \frac{1}{5\alpha + s\cdot 5T_A}$$

If alpha = 0 surely the phase margin gets worse







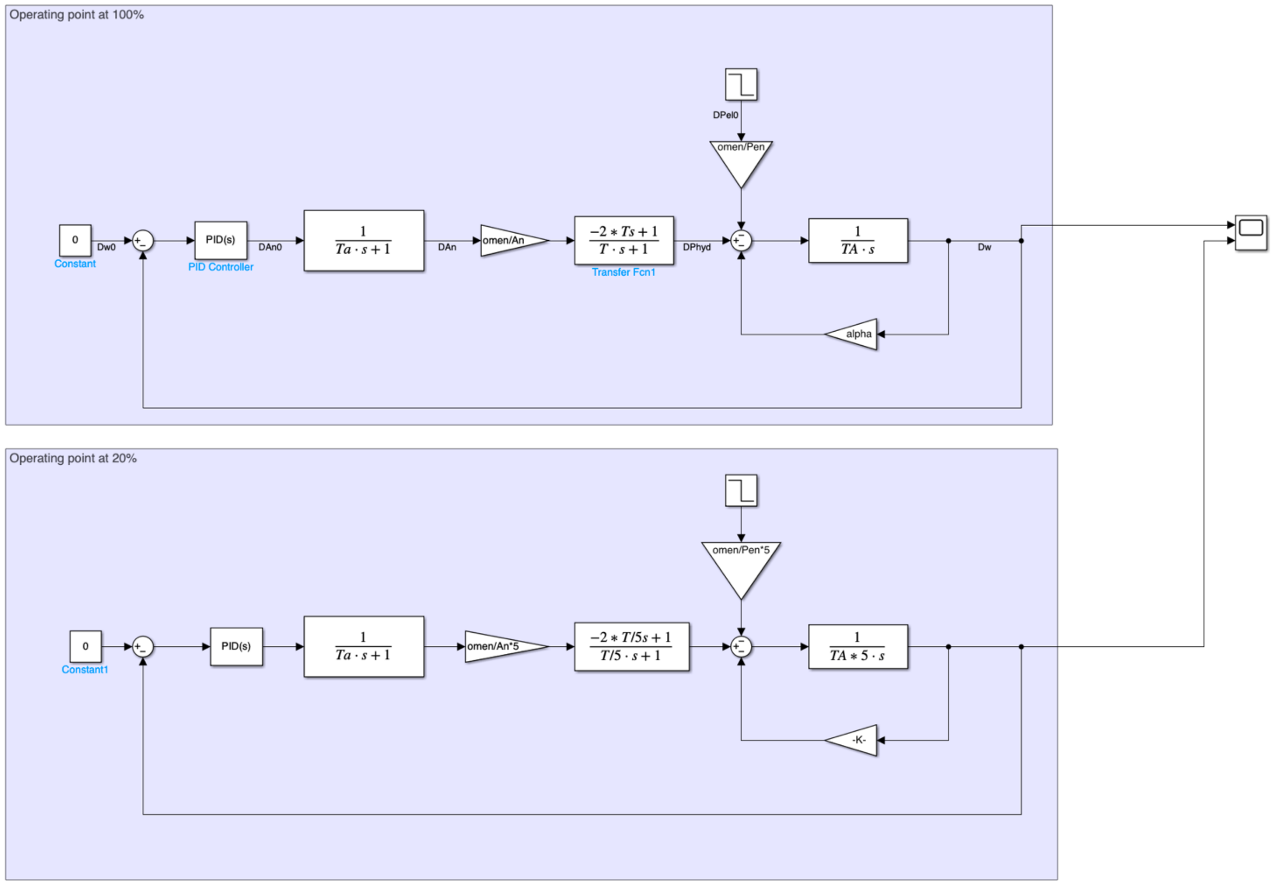


A similar study has been carried out for the closed-loop transfer function

$S(s) = \frac{1}{1+L(s)} \qquad \frac{\Delta\omega}{\Delta P_{el}} = S(s) \cdot \frac{\overline{\omega}/\overline{P_{el}}}{\alpha + T_A s}$

As one can observe in the bode diagrams, the main differences between the 2 systems lie within a limited bandwidth, further proving that a change in the set point will cause a limited change in the behavior of the system. The Script is in Appendix 6

Using the following Simulink scheme, the step responses of the approximated systems with different operating points have been compared.



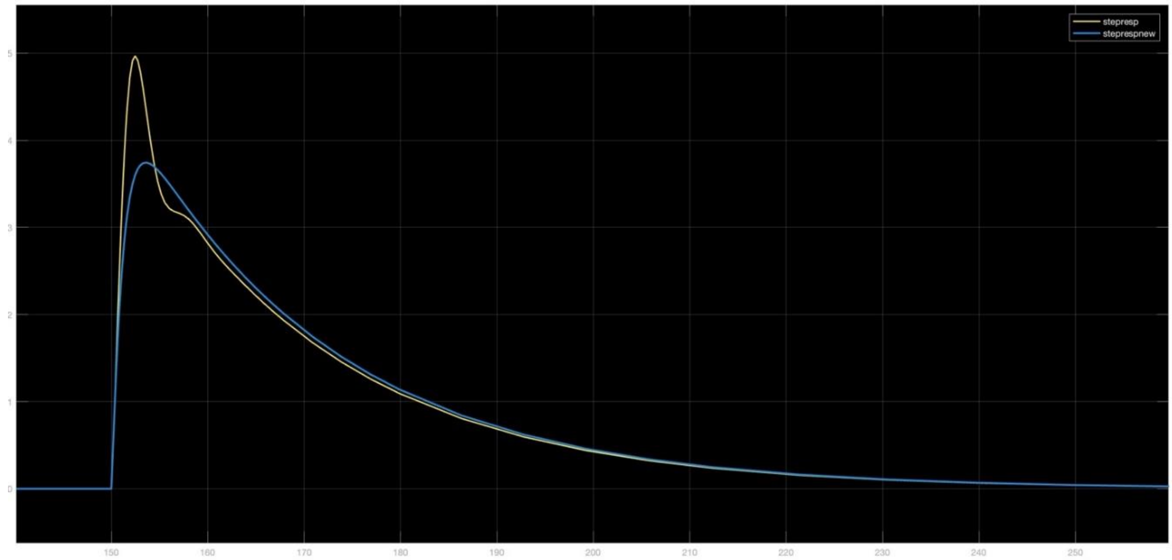


Figure 6: step response with different operating point, notice that we change the sign of the second step to facilitate the comparison with the previous one.

The resulting difference may be attributed to the improvement in phase margin, which removes the effect of the oscillation in the step response. Once again, the obtained results are in line with the Open Modelica simulation, albeit not precisely identical due to the approximation of the system.

Appendices

Appendix 1.

```
t=0:0.001:14;
x=120;
syms s
s1 = t - x/1321;
s2 = t + x/1321;
K1=-(4002.5./s).*(1./(0.614.*exp(-
0.908.*s)+1.386.*exp(0.908.*s)));
K2=-(4002.5./s).*(1./(0.614.*exp(-
0.908.*s)+1.386.*exp(0.908.*s)));
K1_t=ilaplace(K1);
K2_t=ilaplace(K2);
K2_real=(8005*heaviside(s2 + 227/250).*((-
307/693).^floor((125*s2)/227 + 1/2) - 1))/4;
K1_real=(8005*heaviside(s1 + 227/250).*((-
307/693).^floor((125*s1)/227 + 1/2) - 1))/4;
DeltaW=K1_real+K2_real;
DeltaP=1321/7.07*K1_real-1321/7.07*K2_real;
figure;
plot(t,DeltaW);
title('Ideal Delta-w')
figure;
plot(t,K1_real);
hold on
plot(t,K2_real);
hold off
mycolors = [1 0 0; 0 0 1];
ax=gca;
ax.ColorOrder = mycolors;
title('K1 & K2 at L/10')
figure;
plot(t,K2_real);
figure;
plot(t,K1_real);
figure;
plot(t,DeltaP);
title('Ideal Delta-p')
% grid on;
% figure;
% plot(t,DeltaP);
% grid on;
```

Appendix 2.

```
Ti=20;
syms x;
[solx, param, cond] = solve(atan(x*Ti)-atan(x*12)-
atan(x*0.35)-atan(x*0.7)-atan(x*0.2)+(180-90-
50)*pi/180==0,x,'ReturnConditions', true);
xvalues = vpa(solx)
t=1;
while imag(xvalues(t))~=0 || real(xvalues(t))<0
t=t+1;
end
while real(xvalues(t))<0
t=t+1;
end
ome=xvalues(t)
i=sqrt(-1); %massima frequenza ottenibile per Ti=20s e
phi_margine=50 deg
T=0.35;
Ta=0.2;
TA=12;
alfa=1;
omeN=2*pi*50; % valore nominale della frequenza
An=0.29337; % An allo steady state
I=(1+i*ome*Ti)/(i*ome*Ti); %contributo dell'integratore
Ga=(1/(i*ome*Ta+1)); %TF dell'attuatore
Ghyd=(1-2*T*i*ome)/(1+i*ome*T); %TF della diga
Galfa=(1/(i*ome*TA+alfa)); %TF della "parte eletrica"
LR=I*Ga*Ghyd*Galfa;
L=((Ti*ome*i+1)/(ome*i*Ti))*((1-
2*ome*i*T)/(1+ome*i*T))*(1/(ome*i*Ta+1))*(1/(alfa+ome*i*TA
));
L_denorm=((omeN)/(An))*L; %L denormalizzata per progettare
il controllore denormalizzato
L_approx=((Ti*ome*i+1)/(ome*i*Ti))*(1/(ome*i*TA+1));
abs(L)
abs(L_denorm)
Kp_d=1/abs(L)
KpR=1/abs(LR)
Kp_ddd=Kp_d*omeN/An
Kp_denorm=1/abs(L_denorm) % questo dovrebbe essere il
valore da usare in modelica
Kp_approx=1/abs(L_approx)
Kp_brutta=Ti*ome % Kp calcolata con Kp/Ti=omega_max
Kp_bellina=TA*ome % Kp calcolata con l'approx
Kp/TA=omega_max
```

Appendix 3.

```
% Intervallo di frequenze (omega) desiderato, con passo 0.01
omega = 0:0.001:100;

% Inizializzare i vettori per il modulo e la fase per entrambe le funzioni
magnitude_real = zeros(size(omega));
phase_real = zeros(size(omega));
magnitude_approx = zeros(size(omega));
phase_approx = zeros(size(omega));
magnitude_Lreal = zeros(size(omega));
phase_Lreal = zeros(size(omega));
magnitude_Lapprox = zeros(size(omega));
phase_Lapprox = zeros(size(omega));

% Calcolare il modulo e la fase per ciascuna frequenza per entrambe le
funzioni
for k = 1:length(omega)
  s = 1i * omega(k);
  kp = 0.00662;
  omen = 2*pi*50;
  An = 0.29337;
  kstatic = omen/An;
  T = 0.3507;
  TA = 12;
  Beta = 0.3811;
  mu=omen/(340e6);

  % Calcolo per Greale
  H_real = (1 - 2* Beta * tanh(0.91 * s)) / (1 + Beta * tanh(0.91 * s));
  magnitude_Hreal(k) = abs(H_real);
  phase_Hreal(k) = angle(H_real) * (180 / pi);

  % Calcolo per Gapprox
  H_approx = (1 - 2*T * s) / (1 + T * s);
  magnitude_Happrox(k) = abs(H_approx);
  phase_Happrox(k) = angle(H_approx) * (180 / pi);

  % Calcolo per Lreale
  factor = kp * kstatic * ((20*s+1)/(20*s)) * (1/(TA*s+1)) * (1/(0.2*s+1));
  L_real = H_real * factor;
  magnitude_Lreal(k) = abs(L_real);
  phase_Lreal(k) = angle(L_real) * (180 / pi); % Convertire la fase in gradi

  % Calcolo per Lapprox
  L_approx = H_approx * factor;
  magnitude_Lapprox(k) = abs(L_approx);
  phase_Lapprox(k) = angle(L_approx) * (180 / pi); % Convertire la fase in
gradi

   % Calcolo Ssapprox
  Ss_approx = mu*(1/(TA*s+1))*(1/(1+L_approx));
```

```
   magnitude_Ssapprox(k) = abs(Ss_approx);
   phase_Ssapprox(k) = angle(Ss_approx) * (180 / pi);

   % Calcolo Ssreale
   Ss_reale = mu*(1/(TA*s+1))*(1/(1+L_real));
   magnitude_Ssreale(k) = abs(Ss_reale);
   phase_Ssreale(k) = angle(Ss_reale) * (180 / pi);
end

% Plottare il diagramma di modulo (in dB) per Lreale e Lapprox
figure (1);
subplot(2, 1, 1);
semilogx(omega, 20 * log10(magnitude_Lreal), 'b', omega, 20 *
log10(magnitude_Lapprox), 'r--');
grid on;
title('Diagramma di Bode - Modulo di Lreale e Lapprox');
xlabel('Frequenza (rad/s)');
ylabel('Ampiezza (dB)');
legend('Lreale', 'Lapprox');

% Plottare il diagramma di fase (in gradi) per Lreale e Lapprox
subplot(2, 1, 2);
semilogx(omega, phase_Lreal, 'b', omega, phase_Lapprox, 'r--');
grid on;
title('Diagramma di Bode - Fase di Lreale e Lapprox');
xlabel('Frequenza (rad/s)');
ylabel('Fase (gradi)');

% Plottare il diagramma di modulo (in dB) per {Ss*Ga}reale e
{Ss*Ga}approx
figure (2);
subplot(2, 1, 1);
semilogx(omega, 20 * log10(magnitude_Ssreale), 'b', omega, 20 *
log10(magnitude_Ssapprox), 'r--');
grid on;
title('Diagramma di Bode - Modulo di {Ss*Ga}reale e {Ss*Ga}approx');
xlabel('Frequenza (rad/s)');
ylabel('Ampiezza (dB)');
legend('{Ss*Ga}reale', '{Ss*Ga}approx');

% Plottare il diagramma di fase (in gradi) per {Ss*Ga}reale e {Ss*Ga}approx
subplot(2, 1, 2);
semilogx(omega, phase_Ssreale, 'b', omega, phase_Ssapprox, 'r--');
grid on;
title('Diagramma di Bode - Fase di {Ss*Ga}reale e {Ss*Ga}approx');
xlabel('Frequenza (rad/s)');
ylabel('Fase (gradi)');

% Plottare il diagramma di modulo (in dB) per Hreale e Happrox
figure (3);
subplot(2, 1, 1);
semilogx(omega, 20 * log10(magnitude_Hreal), 'b', omega, 20 *
log10(magnitude_Happrox), 'r--');
grid on;
title('Diagramma di Bode - Modulo di Hreale e Happrox');
xlabel('Frequenza (rad/s)');
ylabel('Ampiezza (dB)');
legend('Hreale', 'Happrox');

% Plottare il diagramma di fase (in gradi) per Hreale e Happrox
subplot(2, 1, 2);
semilogx(omega, phase_Hreal, 'b', omega, phase_Happrox, 'r--');
grid on;
title('Diagramma di Bode - Fase di Hreale e Happrox');
xlabel('Frequenza (rad/s)');
ylabel('Fase (gradi)');
```

Appendix 4.

```
Ti=20;
ome_c=0.62394;
TA=12;
omen=2*50*pi;
Pen=340e6;
BBB=-(Ti/(ome_c*TA)/(-1*ome_c+Ti))*omen/Pen*-0.1*Pen

S_s = tf([TA 1 0], [Ti/ome_c Ti+1/ome_c 1])*Ti/ome_c/TA
dome_dPel = - S_s*tf([omen/Pen], [TA 1])*(-36e6)

HS_slow = tf([-BBB],[20 1])
HS_fast = tf([BBB],[1.6 1])

figure(2)
step(HS_slow)
hold
step(HS_fast)

figure(1)
step(HS_slow+HS_fast)
```

Appendix 5.

```
%% Confronto Ls margin
Ti=20;
Ta = 0.2;
T = 0.3507;
ome_c=0.62394;
TA=12;
omen=2*50*pi;
Pen=340e6;
mu=omen/(340e6)*-36e6;
An = 0.293366;
kp = 0.00662;
alpha = 1;

Ls = omen/An*kp * tf([Ti 1], [Ti 0]) * tf([1], [Ta 1]) * ...
   tf([-2*T 1], [T 1]) * tf([1], [TA alpha])

figure(1)
margin(Ls)
hold on
grid on
[Gm, Pm, Wcg, Wcp] = margin(Ls)

T_ = 0.3507/5;
TA_ = 5*12;
kp = 0.00662;
An_ = An/5;
alpha_ = 5*alpha;

Ls_ = omen/An_*kp * tf([Ti 1], [Ti 0]) * tf([1], [Ta 1]) * ...
   tf([-2*T_ 1], [T_ 1]) * tf([1], [TA_ alpha_])

margin(Ls_)

[Gm, Pm, Wcg, Wcp] = margin(Ls_)
```

Appendix 6.

```
%% Confronto pto7
clc
clear all
close all

% Intervallo di frequenze (omega) desiderato, con passo 0.01
omega = 0:0.001:100;

% Inizializzare i vettori per il modulo e la fase per entrambe le funzioni
magnitude_real = zeros(size(omega));
phase_real = zeros(size(omega));
magnitude_approx = zeros(size(omega));
phase_approx = zeros(size(omega));
magnitude_Lreal = zeros(size(omega));
phase_Lreal = zeros(size(omega));
magnitude_Lapprox = zeros(size(omega));
phase_Lapprox = zeros(size(omega));

% Calcolare il modulo e la fase per ciascuna frequenza per entrambe le funzioni
for k = 1:length(omega)
  s = 1i * omega(k);
  kp = 0.00662;
  omen = 2*pi*50;
  An = 0.29337;
  kstatic = omen/An;
  T = 0.3507;
  TA = 12;
  Beta = 0.3811;
  mu=omen/(340e6);

  % Calcolo per Greale
  H_real = (1 - 2* Beta * tanh(0.91 * s)) / (1 + Beta * tanh(0.91 * s));
  magnitude_Hreal(k) = abs(H_real);
  phase_Hreal(k) = angle(H_real) * (180 / pi);

  % Calcolo per Gapprox
  H_approx = (1 - 2*T * s) / (1 + T * s);
  magnitude_Happrox(k) = abs(H_approx);
  phase_Happrox(k) = angle(H_approx) * (180 / pi);

  % Calcolo per Lreale
  factor = kp * kstatic * ((20*s+1)/(20*s)) * (1/(TA*s+1)) * (1/(0.2*s+1));
  L_real = H_real * factor;
  magnitude_Lreal(k) = abs(L_real);
  phase_Lreal(k) = angle(L_real) * (180 / pi); % Convertire la fase in gradi

  % Calcolo per Lapprox
  L_approx = H_approx * factor;
  magnitude_Lapprox(k) = abs(L_approx);
  phase_Lapprox(k) = angle(L_approx) * (180 / pi); % Convertire la fase in gradi

   % Calcolo Ssapprox
  Ss_approx = mu*(1/(TA*s+1))*(1/(1+L_approx));
  magnitude_Ssapprox(k) = abs(Ss_approx);
  phase_Ssapprox(k) = angle(Ss_approx) * (180 / pi);

  % Calcolo Ssreale
  Ss_reale = mu*(1/(TA*s+1))*(1/(1+L_real));
  magnitude_Ssreale(k) = abs(Ss_reale);
  phase_Ssreale(k) = angle(Ss_reale) * (180 / pi);
end


% Plottare il diagramma di modulo (in dB) per Lreale e Lapprox
figure (1);
subplot(2, 1, 1);
semilogx(omega, 20 * log10(magnitude_Lreal), 'b', omega, 20 * log10(magnitude_Lapprox), 'r--');
grid on;
title('Diagramma di Bode - Modulo di Lreale e Lapprox');
xlabel('Frequenza (rad/s)');
ylabel('Ampiezza (dB)');

hold
% Plottare il diagramma di fase (in gradi) per Lreale e Lapprox
subplot(2, 1, 2);
semilogx(omega, phase_Lreal, 'b', omega, phase_Lapprox, 'r--');
grid on;
title('Diagramma di Bode - Fase di Lreale e Lapprox');
xlabel('Frequenza (rad/s)');
ylabel('Fase (gradi)');

hold

% Plottare il diagramma di modulo (in dB) per {Ss*Ga}reale e {Ss*Ga}approx
figure (2);
subplot(2, 1, 1);
semilogx(omega, 20 * log10(magnitude_Ssreale), 'b', omega, 20 * log10(magnitude_Ssapprox), 'r--');
grid on;
title('Diagramma di Bode - Modulo di {Ss*Ga}reale e {Ss*Ga}approx');
xlabel('Frequenza (rad/s)');
ylabel('Ampiezza (dB)');

hold

% Plottare il diagramma di fase (in gradi) per {Ss*Ga}reale e {Ss*Ga}approx
subplot(2, 1, 2);
semilogx(omega, phase_Ssreale, 'b', omega, phase_Ssapprox, 'r--');
grid on;
title('Diagramma di Bode - Fase di {Ss*Ga}reale e {Ss*Ga}approx');
xlabel('Frequenza (rad/s)');
ylabel('Fase (gradi)');

hold

% Cambio del punto operativo
% Calcolare il modulo e la fase per ciascuna frequenza per entrambe le funzioni
for k = 1:length(omega)
  s = 1i * omega(k);
  kp = 0.00662;
  omen = 2*pi*50;
  An = 0.29337/5;
  kstatic = omen/An;
  T = 0.3507/5;
  TA = 12*5;
  Beta = 0.3861/5;
  alpha = 1*5;
  mu=omen/(340e6)*5;

  % Calcolo per Greale
  H_real = (1 - 0.2*Beta * tanh(0.91 * s)) / (1 + Beta* tanh(0.91 * s));

  % Calcolo per Gapprox
  H_approx = (1 - 2*T * s) / (1 + T * s);

  % Calcolo per Lreale
  factor = kp * kstatic * ((20*s+1)/(20*s)) * (1/(TA*s+alpha)) * (1/(0.2*s+1));
  L_real = H_real * factor;
  magnitude_Lreal(k) = abs(L_real);
  phase_Lreal(k) = angle(L_real) * (180 / pi); % Convertire la fase in gradi

  % Calcolo per Lapprox
  L_approx = H_approx * factor;
  magnitude_Lapprox(k) = abs(L_approx);
  phase_Lapprox(k) = angle(L_approx) * (180 / pi); % Convertire la fase in gradi

  % Calcolo Ssapprox
  Ss_approx = mu*(1/(TA*s+alpha))*(1/(1+L_approx));
  magnitude_Ssapprox(k) = abs(Ss_approx);
  phase_Ssapprox(k) = angle(Ss_approx) * (180 / pi);

  % Calcolo Ssreale
  Ss_reale = mu*(1/(TA*s+alpha))*(1/(1+L_real));
  magnitude_Ssreale(k) = abs(Ss_reale);
  phase_Ssreale(k) = angle(Ss_reale) * (180 / pi);
end

% Plottare il diagramma di modulo (in dB) per Lreale e Lapprox
figure (1);
subplot(2, 1, 1);
semilogx(omega, 20 * log10(magnitude_Lreal), omega, 20 * log10(magnitude_Lapprox), ...
  '--', 'LineWidth', 2, 'LineWidth', 2);
grid on;
```

```
legend('Lreale', 'Lapprox', 'Lreale*', 'Lapprox*');

% Plottare il diagramma di fase (in gradi) per Lreale e Lapprox
subplot(2, 1, 2);
semilogx(omega, phase_Lreal, omega, phase_Lapprox, '--', 'LineWidth', 2, 'LineWidth', 2);
grid on;

% Plottare il diagramma di modulo (in dB) per {Ss*Ga}reale e {Ss*Ga}approx
figure (2);
subplot(2, 1, 1);
semilogx(omega, 20 * log10(magnitude_Ssreale), omega, 20 * log10(magnitude_Ssapprox), '--' ...
   , 'LineWidth', 2, 'LineWidth', 2);
grid on;
legend('reale', 'approx', 'reale*', 'approx*');

% Plottare il diagramma di fase (in gradi) per {Ss*Ga}reale e {Ss*Ga}approx
subplot(2, 1, 2);
semilogx(omega, phase_Ssreale, omega, phase_Ssapprox, '--', 'LineWidth', 2, 'LineWidth', 2);
grid on;
```

Appendix 7.

```
% Symulink data

kp = 0.00662;
Ti = 20;
TA=12;
omen=2*50*pi;
Pen=340e6;
Ta = 0.2;
T = 0.3507;
An=0.293366;
alpha = 1;
```