Delft University of Technology

Digital Control

Assignment

# Control of a high-speed steel-rolling mill

*Author:*
Máté Bakos

*Lecturer:*
Dr. Tamas Keviczky

March 30, 2018

# Contents

**1 Introduction** **2**

**2 Description of the modelled steel-rolling mill** **3**

**3 Continuous-time** **4**
3.1 Step response . . . . . 4
3.2 Input disturbance rejection . . . . . 6

**4 Discrete-time control** **8**
4.1 State-space representations . . . . . 8
4.1.1 Continuous-time . . . . . 8
4.1.2 Discrete-time . . . . . 8
4.2 Discrete step-response . . . . . 9
4.3 Pole-placement . . . . . 10
4.4 Dynamic observer . . . . . 12
4.5 Plant input disturbance . . . . . 16
4.6 LQ optimal control . . . . . 16

**5 Actuator saturation for discrete controllers** **17**
5.1 Redesign of the discrete PID controller . . . . . 20
5.2 Redesign of the pole placement controller . . . . . 22
5.3 Redesign of the LQR controller . . . . . 24

**6 Steady-state error elimination** **25**
6.1 PID controller steady-state error elimination . . . . . 25
6.2 Disturbance observer . . . . . 25

**7 Delayed control action** **26**
7.1 Effect on transfer function . . . . . 29
7.2 Effect on state-space representation . . . . . 29

**8 Conclusions** **29**

# 1 Introduction

In rolling mills, intermediate steel products are given their final shape and dimension in a series of shaping and finishing operations. Most of the slabs are heated in reheating furnaces and rolled into final shape in hot– or cold–rolling or finishing mills. While some products (e.g. reinforcement bars, steel plates) only require hot-rolling, some others may require both hot– and cold–rolling (steel for cars and white-goods). Mechanical forces for cold rolling will create much more force and energy needs, while hot rolling happens much faster with less forces.

## 2 Description of the modelled steel-rolling mill

The rolling of steel is a complex process, as seen in Figure 1 the system is driven by a servo motor, which through a series of gears, chains, pinions and racks, controls the vertical position of a roller, which in conjunction with a fixed roller determines the distance between them, and thus the thickness of the steel processed.

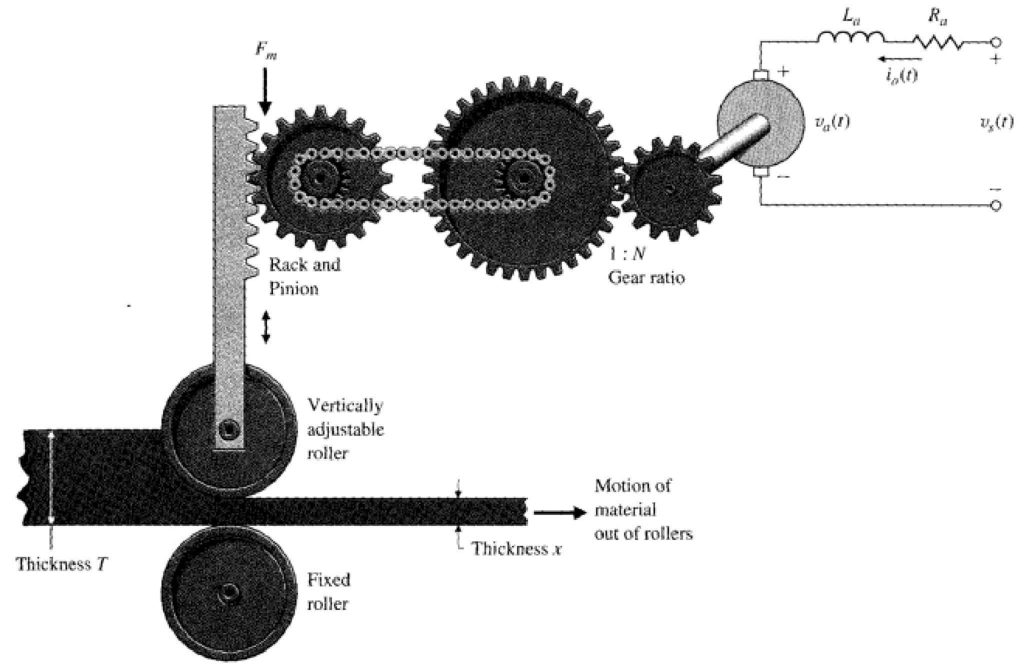

Figure 1: Schematic model of a rolling-mill

Such a mill has to be accurately controlled, and readily adjustable to fulfil its purpose, without significant waste.

The fourth order transfer function that describes the system is as follows:

$$G(s) = \frac{1}{s(s+2)(s^2+100s+2600)} \tag{1}$$

# 3 Continuous-time

In this section, the goal is first to design a controller in order to achieve the fastest response possible to a step input, then redesign said controller for input disturbance rejection.

Looking at the poles of the plant:

- 0
- -50+10i
- -50-10i
- -2

The last three poles are stable, but the first is on the edge of stability.

## 3.1 Step response

The control objectives are as follows:

- overshoot under 5%
- zero steady-state error
- minimal settling-time

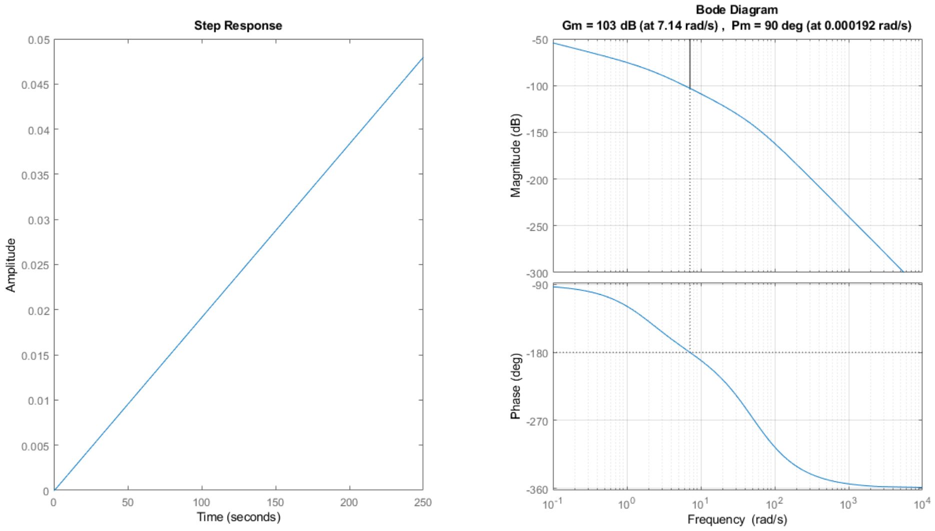

Figure 2: Step response of the uncontrolled plant, and Bode diagram of the same system.

From Figure 2 we can see, the plant is incredibly slow for the steel shaping process. Increasing just the gain of the system would increase the response speed, but based on the phase portrait, it would destabilize the system at 7.14 $\frac{rad}{s}$ and would give unsatisfactory results much before.

To counteract this, a lead compensation has to be used in order to speed up the system without destabilizing it. In order to check if we satisfy the zero steady-state error objective, we can use the final value theorem.

$$e = \lim_{s\to 0} s \, \frac{1}{1+G(s)} \, \frac{1}{s} = \lim_{s\to 0} \frac{1}{1+\frac{1}{s(s+2)(s^2+100s+2600)}} = 0 \qquad (2)$$

This suggests, that a PD controller should be satisfactory for the objectives. The controller will be formulated in the following way:

$$C(s) = K\frac{T_d\ s+1}{\alpha\ T_d\ s+1}, \quad \alpha < 1 \tag{3}$$

This controller will add phase to the system at frequency $\frac{1}{T_d}$, the magnitude of this maximum phase lead ($\phi$) of the controller can be calculated according to the following equations

$$\alpha = \frac{1-\sin(\phi)}{1+\sin(\phi)} \tag{4}$$

$$T_d = \frac{1}{\omega_c\sqrt{\alpha}} \tag{5}$$

To increase the phase in the proximity of the original crossover frequency, the crossover frequency of $\omega_c = 10\frac{rad}{s}$ was chosen for the controller. Using a high $\phi$ value of $85° - 80° - 75°$ renders the settling time to be slow, the fastest response was achieved with $\phi = 65°$, with an adequately chosen $K$ gain.

The final design is as follows:

$$C(s) = 45400\frac{0.6318\ s+1}{0.0491\cdot 0.6318\ s+1} \tag{6}$$

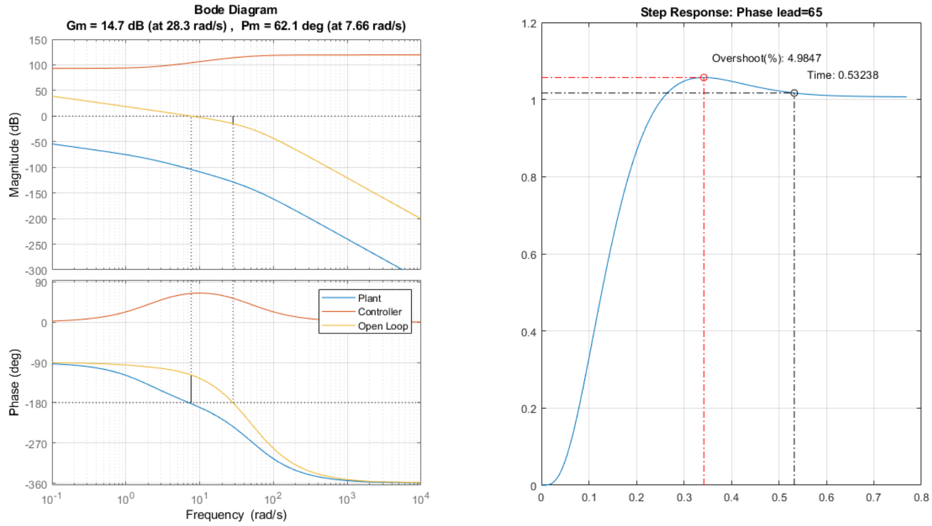

Figure 3: Bode diagram, step response and control input of the final design.

The results with this controller can be seen on Figure 3. The settling time is approximately $0.5s$, with an overshot of almost 5% and a steady-state error of 0.

To verify the steady state error, we now can use the final value theorem for the closed loop as all poles are in the open left-half plane.

$$e = \lim_{s\to 0} s\ \frac{1}{1+C(s)G(s)}\ \frac{1}{s} = \lim_{s\to 0} \frac{1}{1+(45400\frac{0.6318\ s+1}{0.0491\cdot 0.6318\ s+1})(\frac{1}{s(s+2)(s^2+100s+2600)})} = 0 \tag{7}$$

## 3.2 Input disturbance rejection

In this section, the reference signal is zero, but an input step disturbance is added to the system, a schematic model of this can be seen on Figure 4, with the measurement block being a gain of 1.

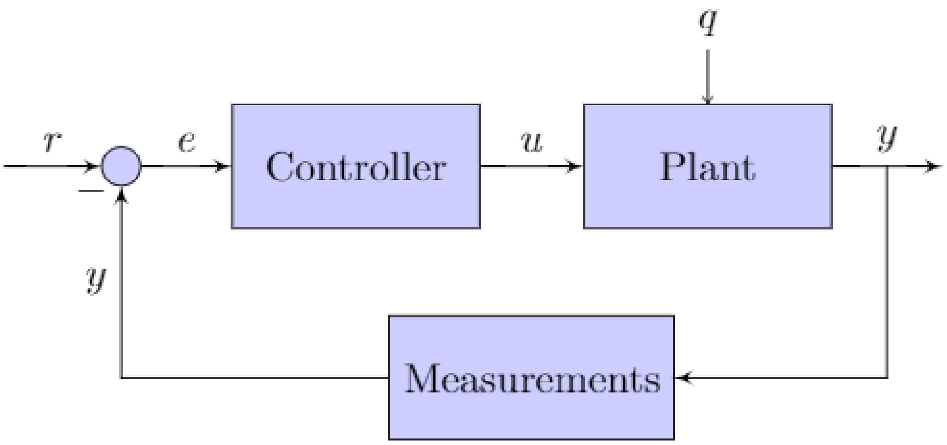

Figure 4: Generic model of the system with disturbance.

The objectives for this design are:

- minimal amplitude in the system output in $y$ caused by $q$
- minimal duration of the disturbance, under 1% of the initial value
- 0 offset in the output

Using the previous design we get the following results:

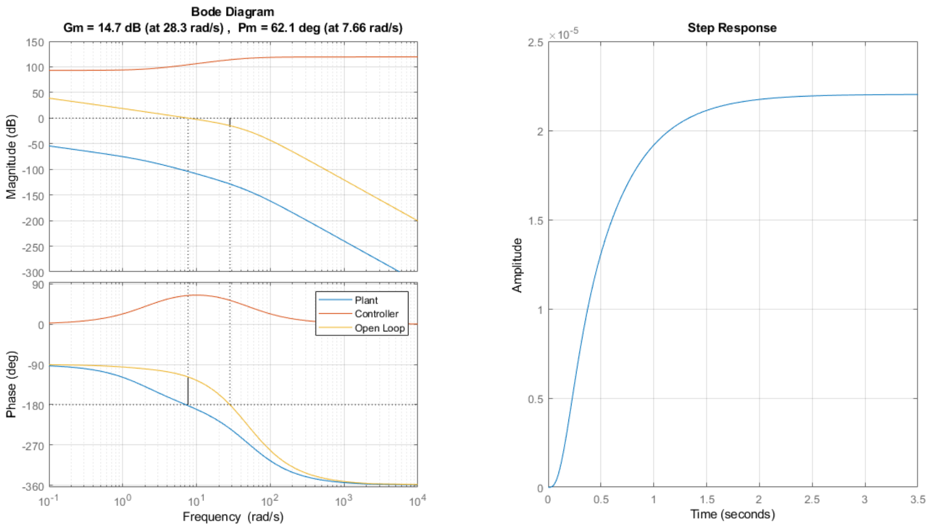

Figure 5: Disturbance rejection with PD controller.

However the disturbance is rejected fairly well, a marginal offset can be seen in the response. This indicates that we need to redesign the controller and include integral action.

The new equation used for the PID controller is as follows:

$$C(s) = K\frac{(1+\frac{s}{\omega_i})\cdot(1+\frac{s}{\omega_d})}{\frac{s}{\omega_i}\cdot(1+\frac{s}{\omega})} \tag{8}$$

With this design, we have to choose $\omega$ to be at least one decade higher, than the crssover frequency of the plant, in this case approximately $100\frac{rad}{s}$, while choosing $\omega_i$ and $\omega_d$ to be close to the crossover frequency of the plant.

To satisfy the minimal duration criteria the phase margin can't be very high as it slows down the response. Once the frequencies are tuned properly, the final controller parameters are as follows:

- $\omega = 100\ \frac{rad}{s}$
- $\omega_i = \omega_d = 3\ \frac{rad}{s}$
- $K = 50000$

The transfer function of this controller is:

$$C(s) = 50000\frac{(1+\frac{s}{3})\cdot(1+\frac{s}{3})}{\frac{s}{3}\cdot(1+\frac{s}{100})} \tag{9}$$

With this design the achieved response is:

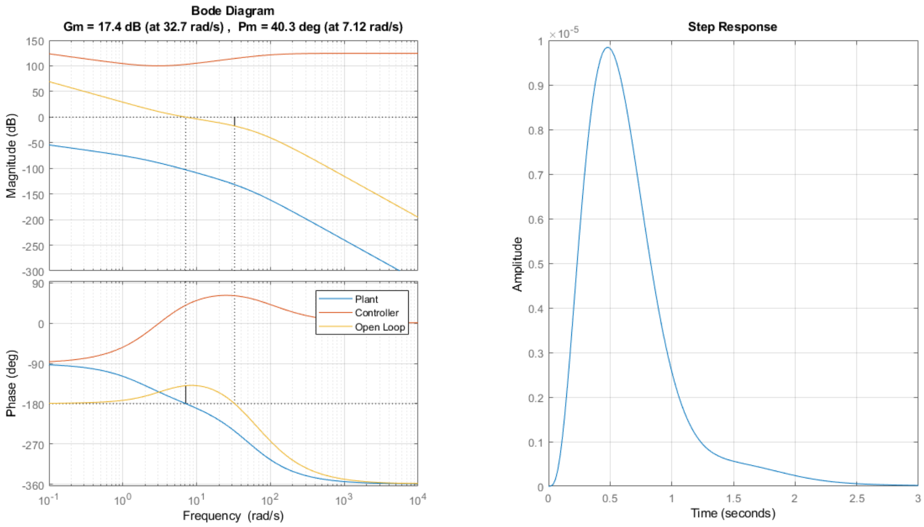

Figure 6: Disturbance rejection with PID controller.

The criteria are satisfied, the phase margin is 40.3°, the rejection is fast, and the maximal amplitude is less than 0.001% of the initial disturbance.

# 4 Discrete-time control

## 4.1 State-space representations

### 4.1.1 Continuous-time

Since the transfer function of the system is strictly proper, the controllable canonical form is a very straightforward representation of the system, it also sheds light on the inner workings of the system.

$$\dot{x}(t) = A\ x(t) + B\ u(t) = \begin{bmatrix} -102 & -2800 & -5200 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} x(t) + \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} u(t) \tag{10}$$

$$y(t) = C\ x(t) \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix} x(t) \tag{11}$$

Since we know, that the output of the system is the position of the moving roller, in this representation, the $4th$ state, and carefully observing the structure of the $A$ matrix, we can see, that the derivative of the $4th$ state, is the $3rd$ state, the derivative of the $3rd$ state is the $2nd$ state, and so on, while the $1st$ state has a much complicated form.

Thus we can conclude, that the states of the system are the jerk, acceleration, velocity, and position of the moving roller, respectively.

### 4.1.2 Discrete-time

Transforming the aforementioned representation to discrete time, can be given in the following form:

$$x_{k+1} = \Phi\ x_k + \Gamma\ u_k \tag{12}$$

$$y_k = C\ x_k + D\ u_k \tag{13}$$

where,

$$\Phi = e^{Ah}, \quad \Gamma = \int_0^h e^{As} ds\ B \tag{14}$$

Regarding the sampling time, the choice has to be appropriate, as if it is too high, we are getting close to a deadbeat controller, which will reach the desired state in one step, but requires very high control action, and will only be able to act on the sampling instance after the change, which depending on the choice, can be rather slow. When the sampling time is going towards 0, based on Equation 14, $\Phi$ will converge towards identity for every system, thus not representing the specific system anymore.

A good measure for an appropriate choice is based on the rise-time of the continuous system, for the system described in Section 3.1 choosing 10 samples per rise time proved to be satisfactory.

With Equations 14 and a sampling time of $h = 0.0153\ s$ the discretized system can be formulated as:

$$x_{k+1} = \begin{bmatrix} 0.0913 & -0.3098 & -0.0706 & 0 \\ 0.4449 & 0.8003 & -0.0459 & 0 \\ 0.0361 & 0.1132 & 0.9979 & 0 \\ 0.0000 & 0.0000 & 0.0002 & 1 \end{bmatrix} x_k + \begin{bmatrix} 0.0017 \\ 0.0011 \\ 0.0001 \\ 0.0000 \end{bmatrix} u_k \tag{15}$$

$$y_k = \begin{bmatrix} 0 & 0 & 0 & 0.5 \end{bmatrix}\ x_k \tag{16}$$

## 4.2 Discrete step-response

For transforming a system from continuous-time to discrete-time, we need a transformation from the Laplace-domain to the z-domain. For this step, different discretization methods are available, the most commonly used are $zero-order-hold$, $first-order-hold$, and $Tustinapproximation$.

The characteristics of each are briefly the following:

- ZOH: provides an exact discretization for staircase inputs
- FOH: provides an exact discretization for piecewise linear inputs
- Tustin: provides a good match between continuous- and discrete-time phases, thus important dynamics of the system are preserved

For the exact discretization of the system in Equation 15 10 samples per rise-time was adequate, but when transforming the whole system using zero-order-hold, we get the following results:

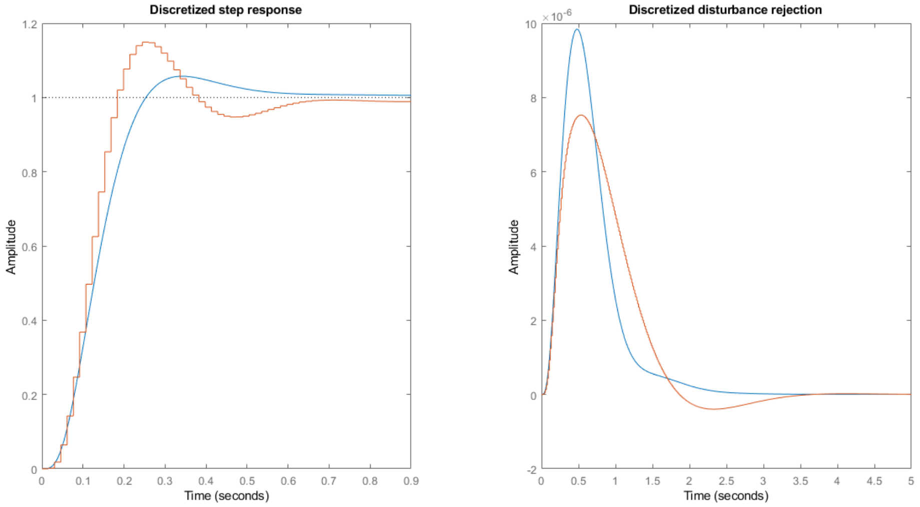

Figure 7: Discretization using zero-order-hold.

As we can see, the responses are far from having a close match. Using the Tustin approximation for both the plant and the controller however yields the following results:

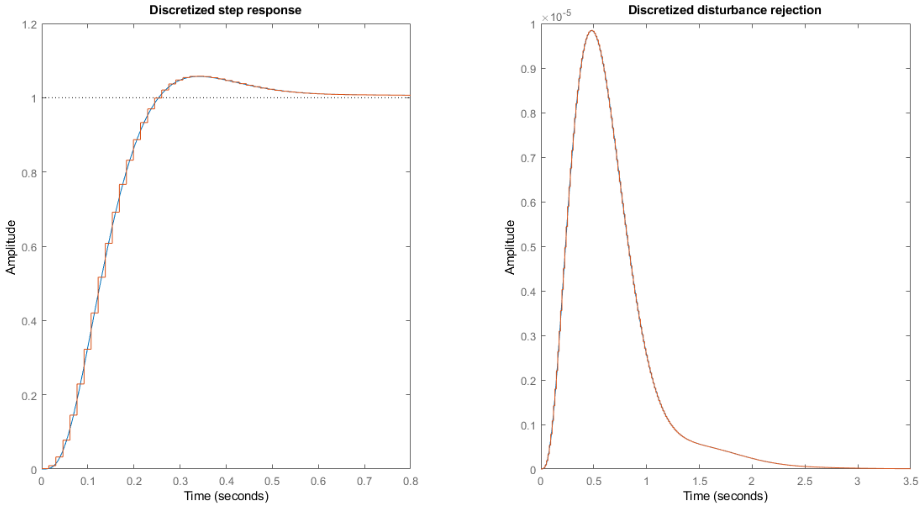

Figure 8: Discretization using Tustin approximation.

It is very important to note, that the sampling times were not changed, and similar results could have been achieved by using zero-order-hold, and decreasing the sampling time drastically. In Figure 9 the sampling time has been decreased to $h = 0.001s$, while still using zero-order-hold.

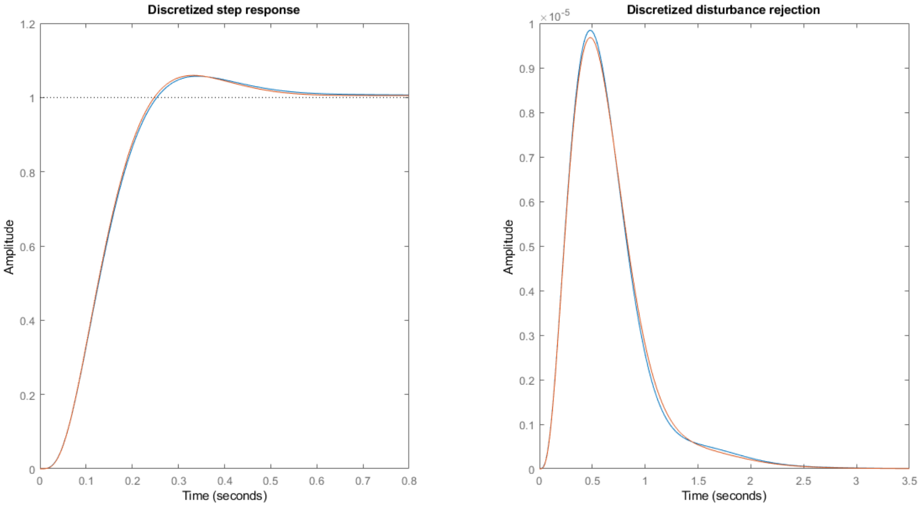

Figure 9: Discretization using low sampling time zero-order-hold.

It is apparent, that if the goal is to get a good match between the continuous- and discrete-time systems, the choice of the proper discretization method is crucial, and the Tustin approximation performs far superior, with greater sampling times, compared to zero-order-hold, when discretizing complex systems.

## 4.3 Pole-placement

This part will be solved using Simulink, with state feedback, while scaling the input to account for the steady-state error. The Simulink model used can be seen on Figure 10.

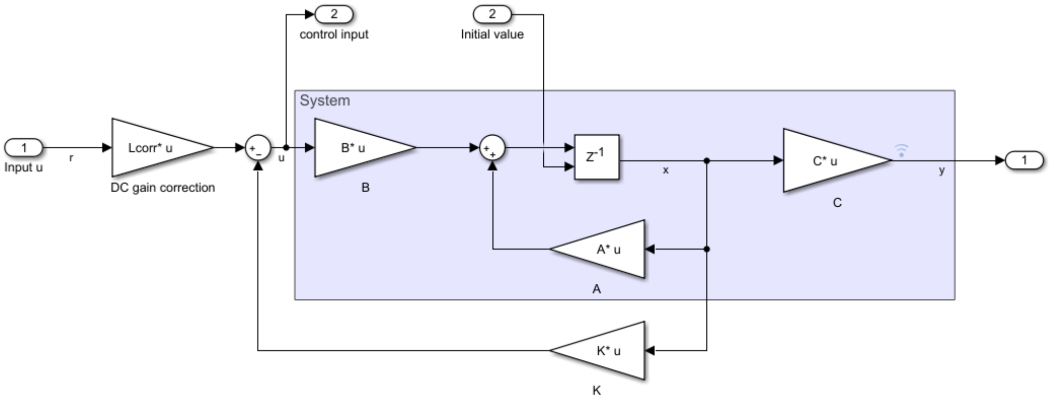

Figure 10: Simulink model of the state feedback.

Characteristics, such as rise-time, time to peak amplitude, peak overshoot of a second-order system can be directly related to its poles. We can mimic the behaviour of a second order system, and use its relations between system characteristics and its poles for our fourth-order system by using the dominant poles for our second order system, while choosing the remaining two poles to be significantly faster. Choosing these non-dominant poles to be 10 times faster than the natural frequency of the second-order system is sufficient.

Relations between the natural frequency ($\omega_n$), the damping ratio ($\xi$) and the system's peak overshoot ($M_p$), time to peak amplitude ($t_p$) is the following:

$$t_p = \frac{\pi}{\omega_n\sqrt{1-\xi^2}} \tag{17}$$

$$M_p = e^{-\xi\omega_n t_p} \tag{18}$$

With $t_p$ and $M_p$ known from the continuous time response, the set of equations is solvable, thus $\omega_n = 12.6978\frac{rad}{s}$ and $\xi = 0.7$.

The dominant poles of the continuous system are obtained from Equation **??**, the non-dominant poles are chosen to be at least 10 times faster, and on the real left half-plane.

$$s^2 + 2\xi\omega_n s + \omega_n^2 = 0 \tag{19}$$

These poles can be discretized by the following equations, using a sampling time, that gives approximately 7 samples per rise-time ($h = 0.0326s$)

$$\lambda_{disc} = e^{h\lambda_{cont}} \tag{20}$$

The poles are placed, and the corresponding feedback gain is then calculated.(The difference in the order of magnitudes of the elements is so great, that writing them out would have little use, results with proper numerics can be found in the code.)

$$K = 10^6\ [0.0004\ 0.0007\ 0.0035\ 3.3748] \tag{21}$$

In order to achieve the desired output, we need to scale the input using a feed forward gain.

$$L_{corr} = 6.7010 \cdot 10^6 \tag{22}$$

The response achieved by the pole placement method is the following:

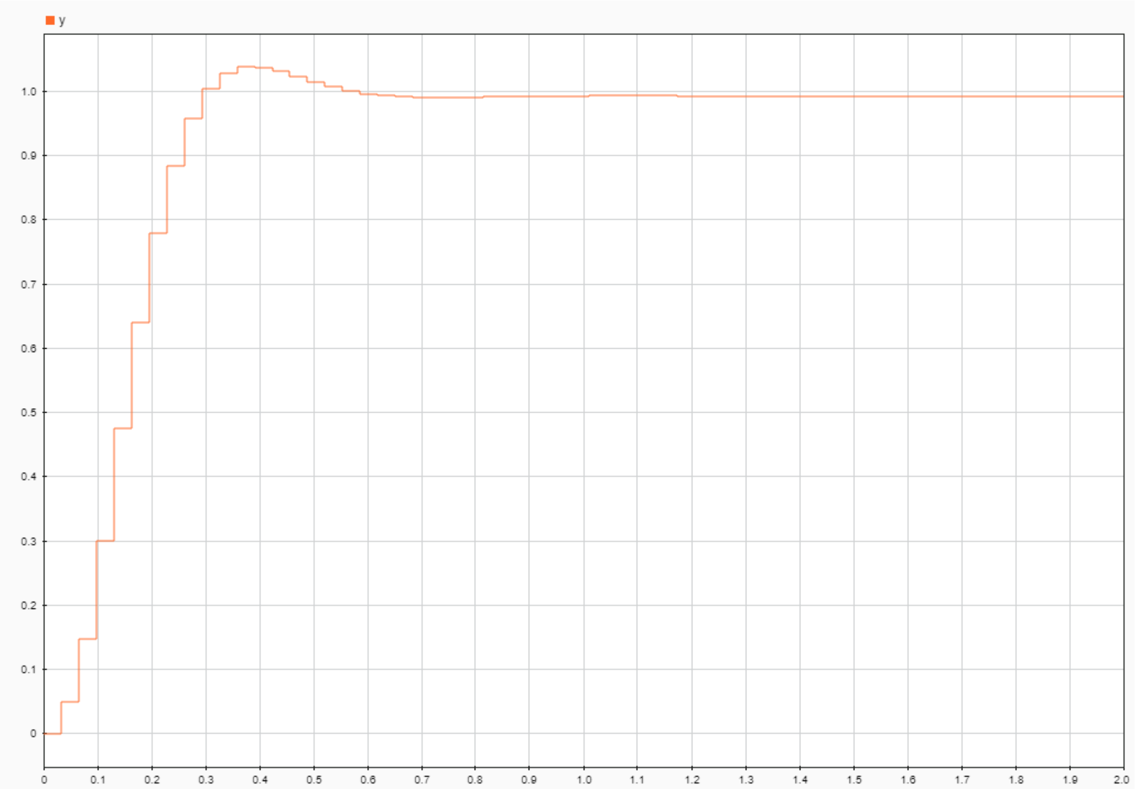

Figure 11: Set point response using pole placement.

The overshoot is under 5% and the settling-time of the system is approximately 0.6 $s$, which closely matches the continuous time response of the system designed in Section 3.1.

## 4.4 Dynamic observer

Realistically, supposing that we have full information of the states is uncanny, more often than not, we do not know from which initial state the system starts from, and thus we don't know the exact evolution of them. In these cases, if the system is observable, it is possible to build a dynamic observer, which estimates the states from an arbitrary initial state, and based on the input and output measurements, provides an estimate of the states and their evolution. Based on the difference between the real measurements and the estimated system output ($e_k = y_k - \hat{y}_k$), we can update the model with an observer gain $L$, that steers this difference towards zero, and thus match the real, and estimated system, while giving us full information on the estimated state evolution, which after this difference reaches 0, is the real state evolution.

Expanding the existing feedback and feedforward system design, described in Section 4.3, with an observer, we can reconstruct the original system, while only measuring the input and output signals of the system. The Simulink model used can be seen on Figure 12

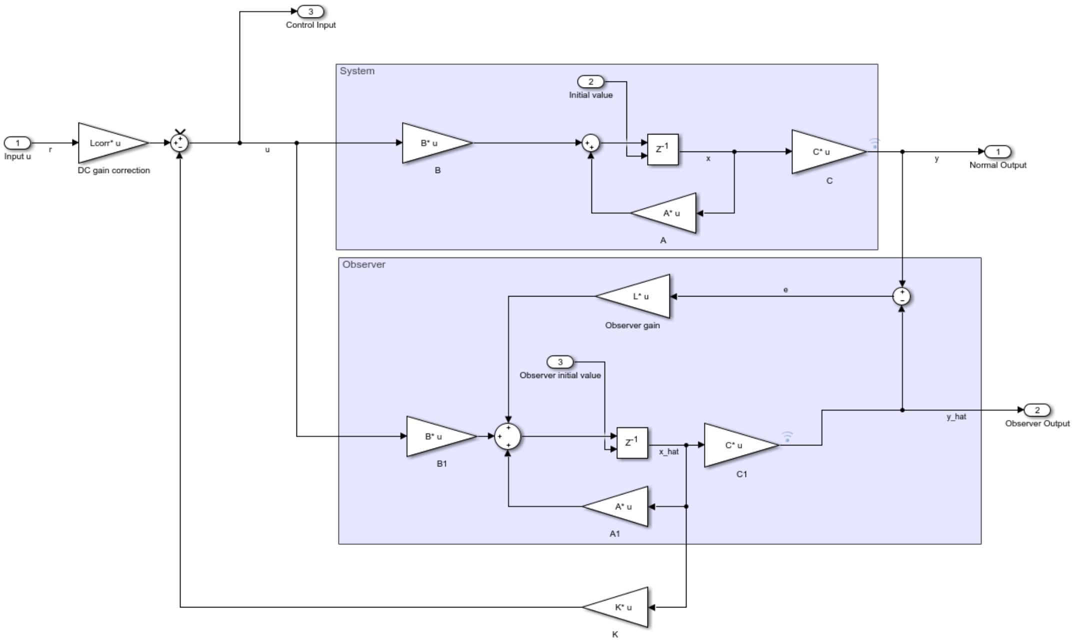

Figure 12: Simulink model of the dynamic observer.

Since we have no information on the evolution of $x_{k+1}$, we will construct a copy of that system, but choosing the initial conditions to be 0.

$$\hat{x}_{k+1} = \Phi\ \hat{x}_k + \Gamma\ u_k \tag{23}$$

$$\hat{y}_k = C\ \hat{x}_k + D\ u_k \tag{24}$$

This equation can be formulated differently, using only the input and output of the system, by including the $L$ observer gain, the $K$ feedback gain, and the $L_{corr}$ feedforward gain.

$$\hat{x}_{k+1} = (\Phi - \Gamma K - LC)\ \hat{x}_k + \Gamma L_{corr}\ r_k + L\ y_k \tag{25}$$

Using the duality principle of controllability and observability placing the observer poles can be achieved with a very similar toolset as in Section 4.3.

The initial conditions of the observer system are all 0, while the real system initializes from:

$$x_{init} = \begin{bmatrix} -17 \\ 5 \\ -8 \\ 0.5 \end{bmatrix} \tag{26}$$

The observer poles should be faster than the poles placed for the feedback gain, as we want our observer to converge to zero error fast. Figure 13 depicts two different observer pole placements, $L_{1.5,green}$ is 1.5 times faster than the feedback poles, while $L_{5,blue}$ is 5 times faster. Since the feedback poles are already quite fast, and faster observer poles have a higher offset before settling I would advise against choosing faster poles.

The corresponding observer gains are as follows:

$$L_{1.5,green} = 10^3 \begin{bmatrix} -1.3084 \\ 1.3913 \\ 9.1625 \\ 0.0029 \end{bmatrix}, \quad L_{5,blue} = 10^4 \begin{bmatrix} -2.8535 \\ 4.6586 \\ 1.6680 \\ 0.0003 \end{bmatrix} \tag{27}$$

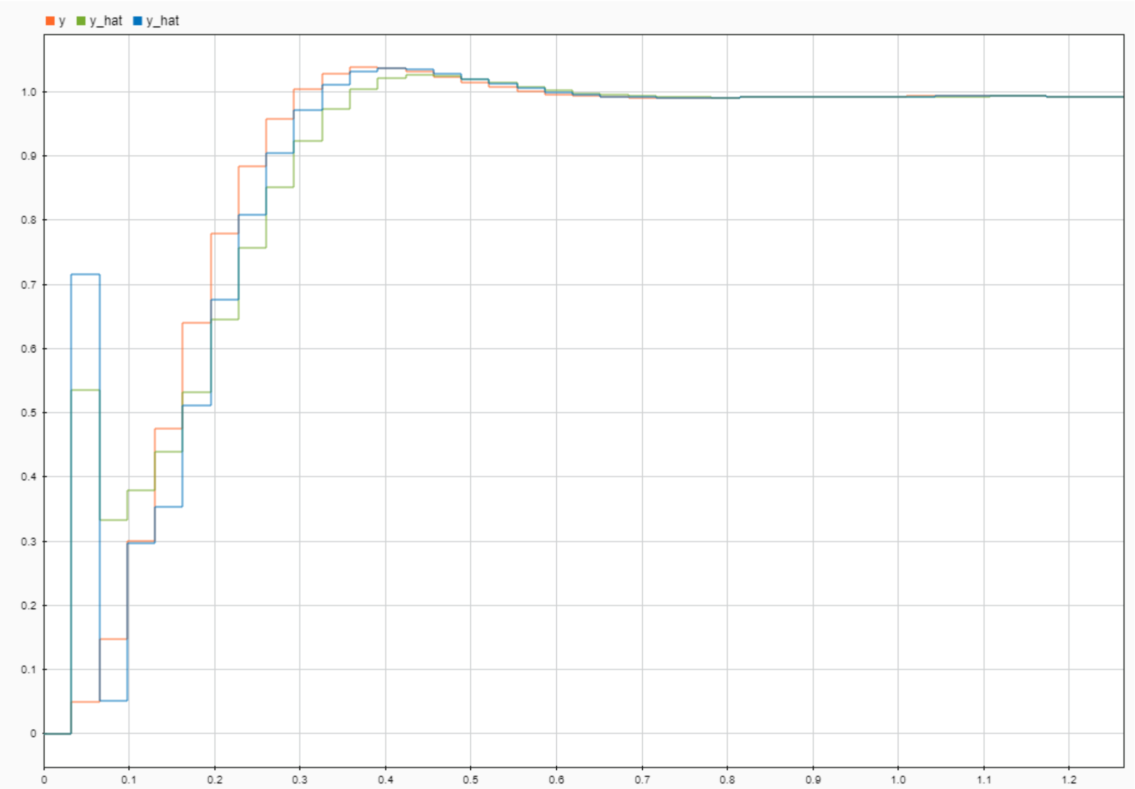

Figure 13: Comparison of the observers and the real system.

However the initial offset and oscillation can cause problems in production, having a less severe initial difference would mitigate their effect. The observer settles fast and after the initial overshoot estimates the real system behaviour accurately. The error dynamics of the observed and real states can be seen on Figures 14-17, after a large initial difference, the state errors converge to zero fast.

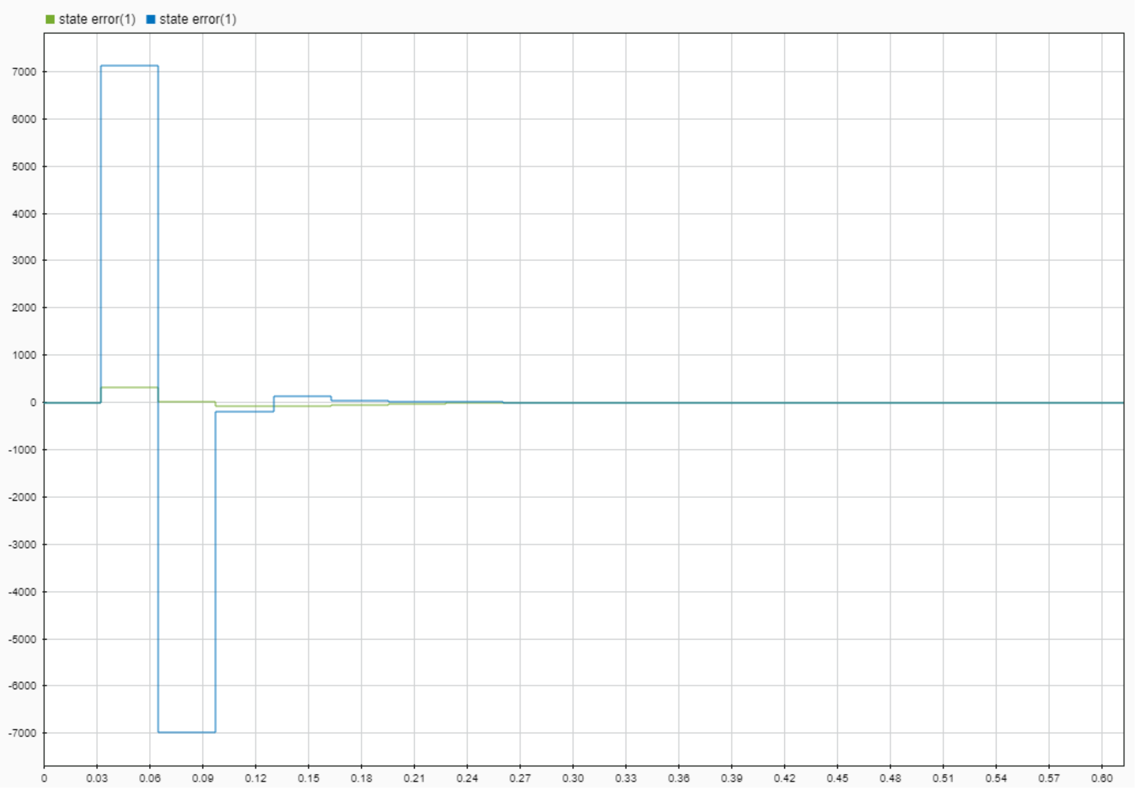

Figure 14: Comparison of the different observer error dynamics for the first state, the jerk.

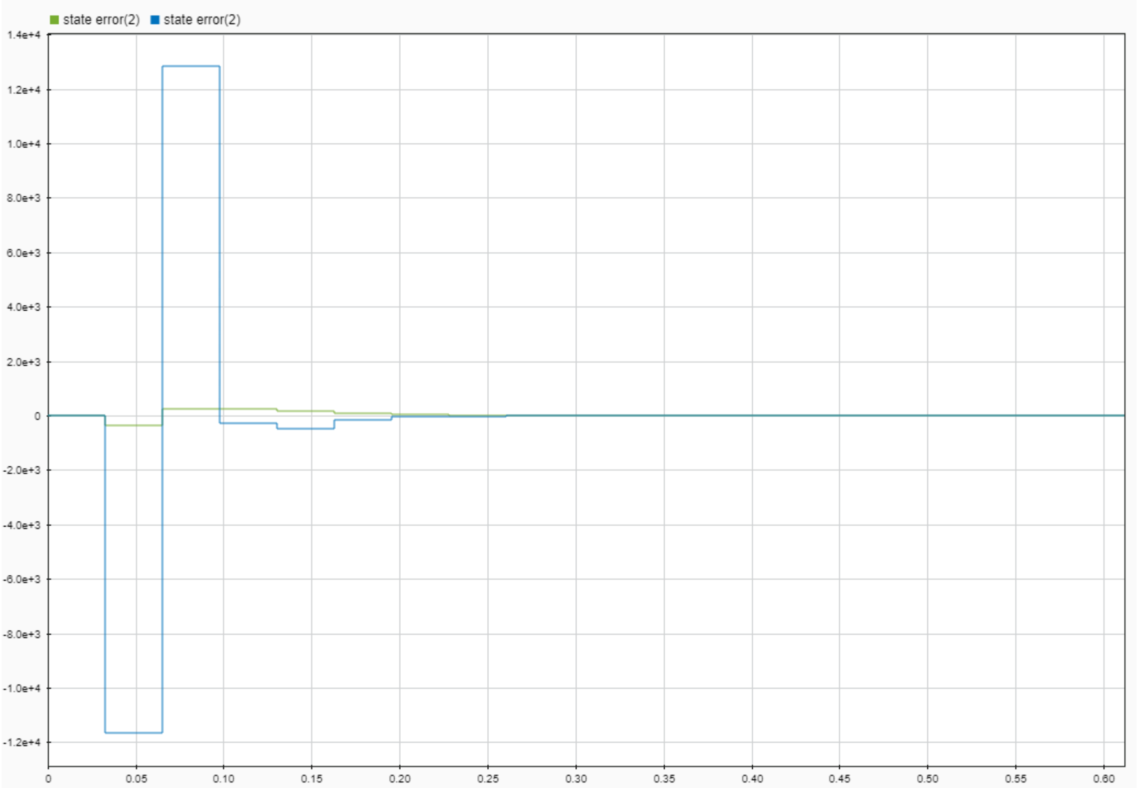

Figure 15: Comparison of the different observer error dynamics for the second state, the acceleration.

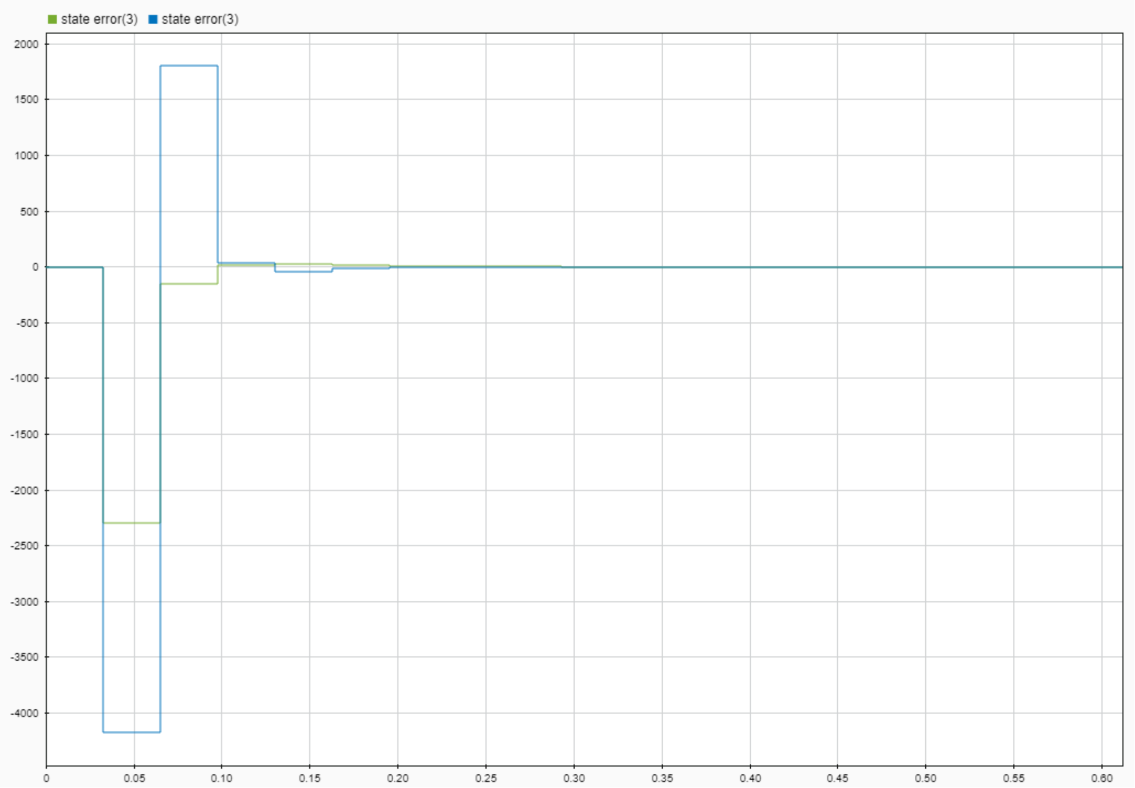

Figure 16: Comparison of the different observer error dynamics for the third state, the velocity.

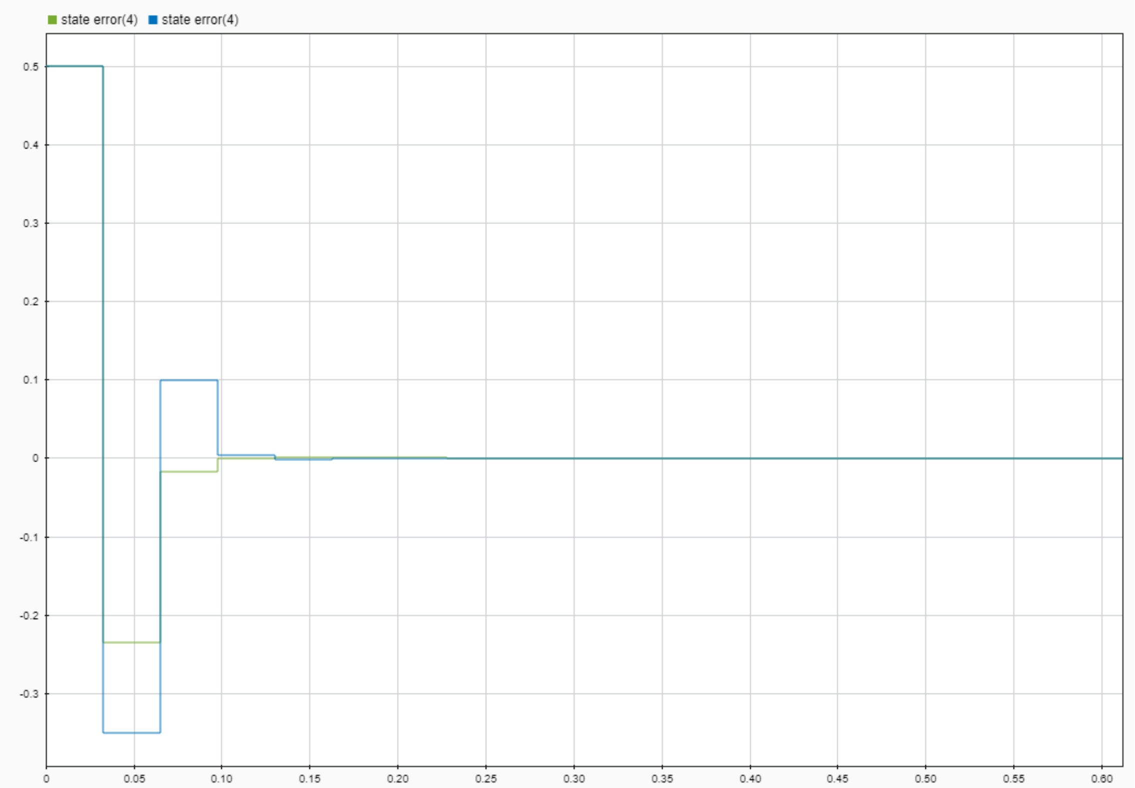

Figure 17: Comparison of the different observer error dynamics for the fourth state, the position.

## 4.5 Plant input disturbance

Assuming that the plant input has an additional step disturbance, we need to augment the system matrices, so the observer can estimate, and reduce its effect to 0. The augmented observer equations are as follows:

$$\begin{bmatrix}\hat{x}_{k+1}\\ \hat{w}_{k+1}\end{bmatrix} = \begin{bmatrix}\Phi & \Gamma\\ 0 & 1\end{bmatrix}\begin{bmatrix}\hat{x}_k\\ \hat{w}_k\end{bmatrix} + \begin{bmatrix}\Gamma\\ 0\end{bmatrix} u_k + \begin{bmatrix}L\\ L_w\end{bmatrix}(y_k - C\hat{x}_k) \tag{28}$$

$$u_k = -\begin{bmatrix}K & 1\end{bmatrix}\begin{bmatrix}\hat{x}_k\\ \hat{w}_k\end{bmatrix} \tag{29}$$

Since we want to cancel out the step disturbance completely, we augment the feedback gain K with 1. With that said, we choose the last pole of the observer to be much faster than the dominant poles of the system, resulting in the following observer gain:

$$L_{disturbance} = 10^7 \begin{bmatrix}0.0001\\ 0.0081\\ 0.0018\\ 0.0000\\ 1.4877\end{bmatrix} \tag{30}$$

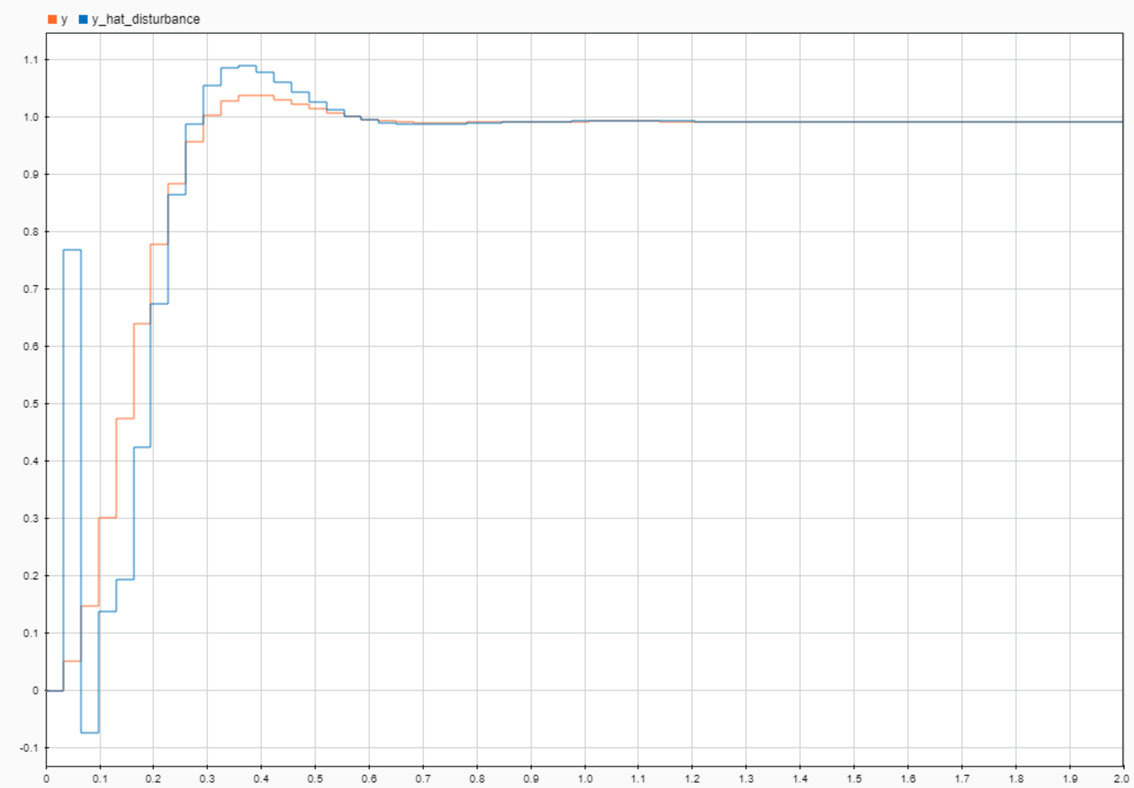

Figure 18: Comparison of the observer responses, with and without the input disturbance.

## 4.6 LQ optimal control

In this section, the control structure is the same as in Section 4.3, but instead of directly placing the closed loop poles, we are optimizing a quadratic cost function:

$$J = \sum_{k=0}^{n} (x_k^T Q x_k + u_k^T R u_k) \tag{31}$$

In this equation weighting matrix $Q$ determines the importance of the states, while $R$ concerns the importance of the inputs. Since this is a cost minimization process the actual values of $R$ are not important, only its ratio compared to the elements of $Q$.

The matrix $Q$ has to be positive semi definite. A good approach would be to determine the cost, based on the output. for this we can write $Q$ in the following form:

$$x_k^T\ Q\ x_k = \rho\ x_k^T\ C^T\ C\ x_k = \rho\ y_k^T\ y_k \tag{32}$$

Increasing the constant $\rho$ will speed up the response, but will require shorter sampling intervals, as well as increase the overshoot, introduce oscillations and eventually destabilize the system.

Different $\rho$ and $h$ combinations can be seen on Figure 19 from slowest to fastest:

$$\rho_{green} = 10^{10}, \quad \rho_{purple} = 10^{12} \quad \rho_{red} = 10^{15} \tag{33}$$

$$h_{green} = 0.03\ s, \quad h_{purple} = 0.015\ s, \quad h_{red} = 0.005\ s \tag{34}$$

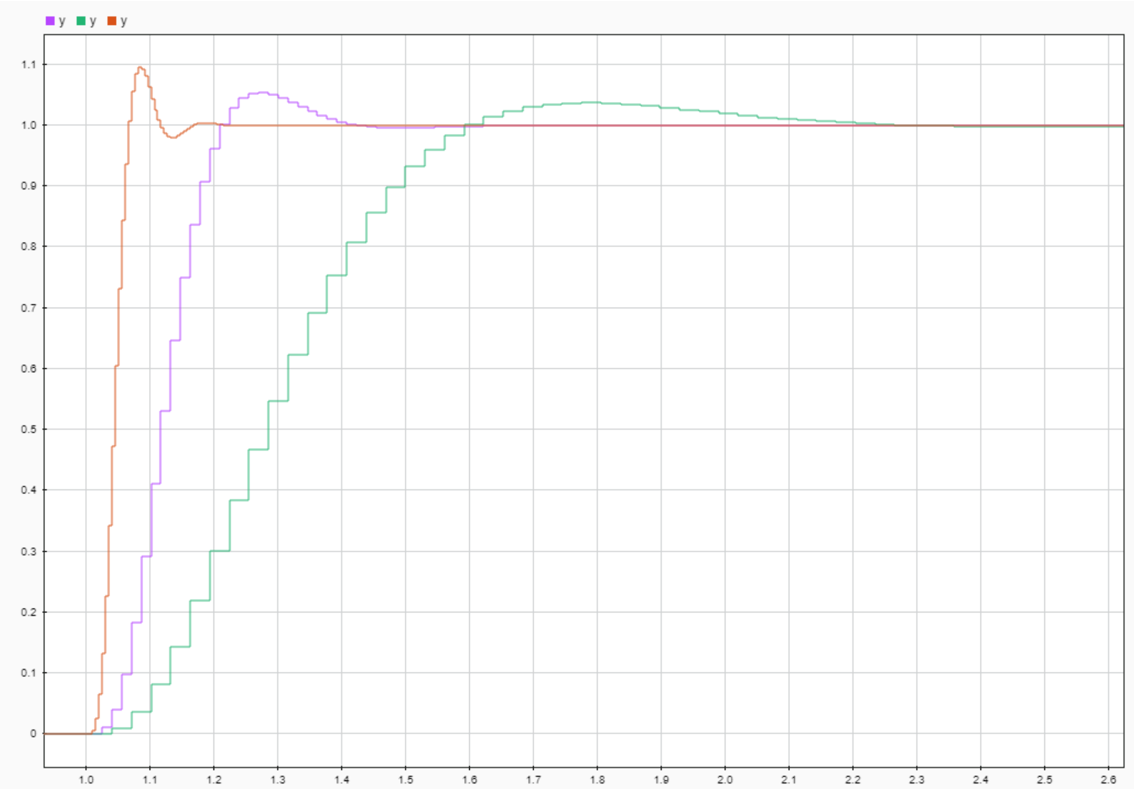

Figure 19: Comparison of different weighting matrices and sampling times.

While the low weigh creates a very slow response, and high weigh introduces increased overshoot, the purple trajectory closely mathes the continuous-time response. For these parameters, the feedback gain was the following:

$$K = 10^5\ [0.0009\ 0.0016\ 0.0070\ 4.1406] \tag{35}$$

## 5 Actuator saturation for discrete controllers

In real-world applications creating theoretically good controllers that require significantly more control effort, than that is possible with the system's actuators, is very dangerous, as these actuators will saturate, which is not their expected behaviour. This phenomena can easily destabilize systems even if the theoretical controller is nearly perfect for the process, thus we need to create controllers that give the best performance, taking the actuators control limits into account.

Since in the previous sections, didn't include any kind of limitations on the control effort, in most cases the required control inputs by our controllers are far larger, than that of any rational actuator can handle.

Limit of the controllers is $u = \pm 1000$.

It is important to note, that in case of only step disturbance, saturation of the controllers is not possible in our case, as the controller only acts against this disturbance, thus looking at the transfer function between the disturbance and the input can only be a maximum of 1.

Continuous- and discrete -time controllers:

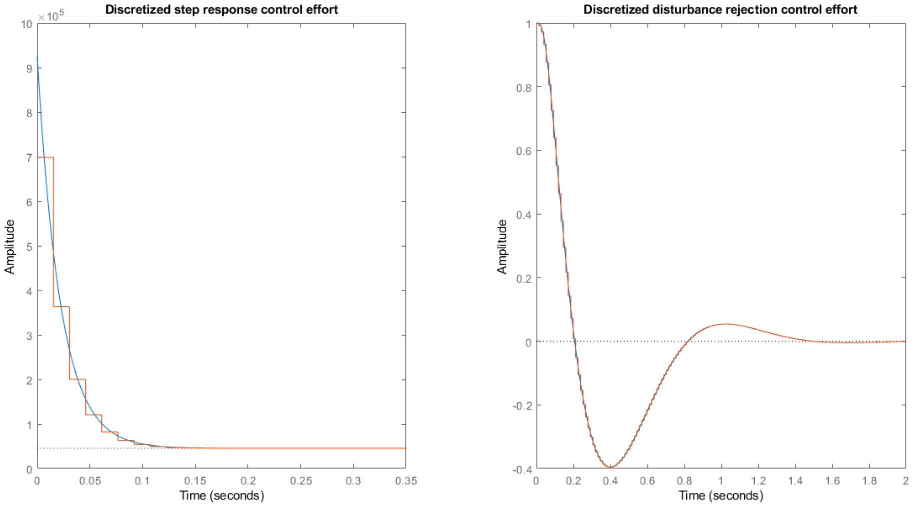

Figure 20: Set-point response and disturbance rejection control effort.

Pole placement method:

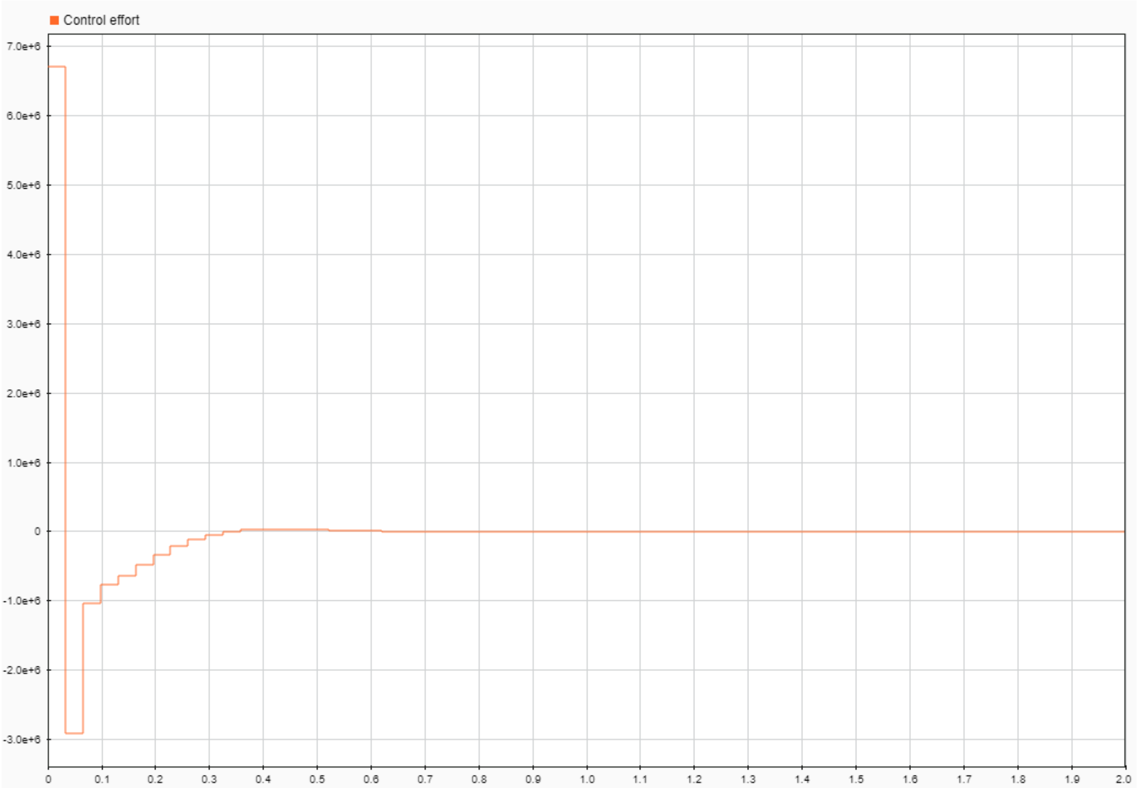

Figure 21: Pole placement control effort.

Observer pole placement scheme:

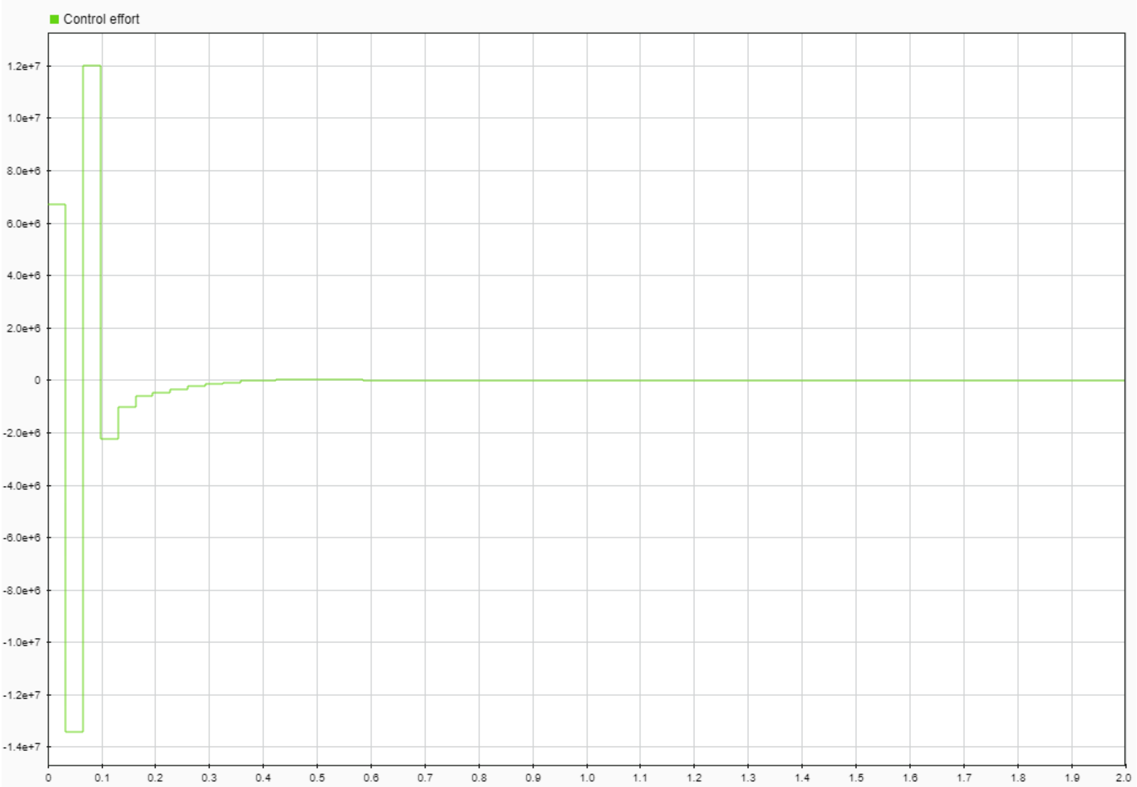

Figure 22: Observer control effort.

Linear-Quadratic Regulator:

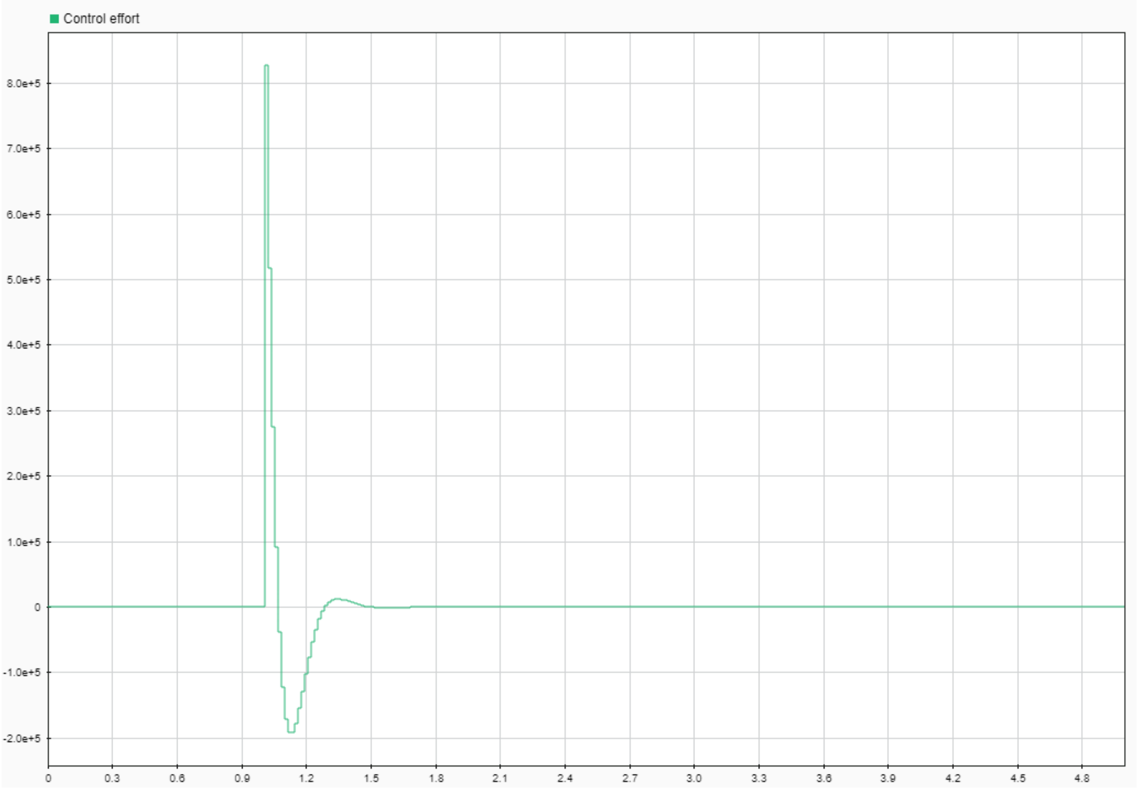

Figure 23: LQR control effort.

## 5.1 Redesign of the discrete PID controller

To take actuator saturation into account, the design of these controllers need to be revisited.

The least control effort is required by increasing the proportional gain, while still speeding up the response. Using only a P controller with gain of 1000, will result in an initial control effort of 1000, which will diminish over time.

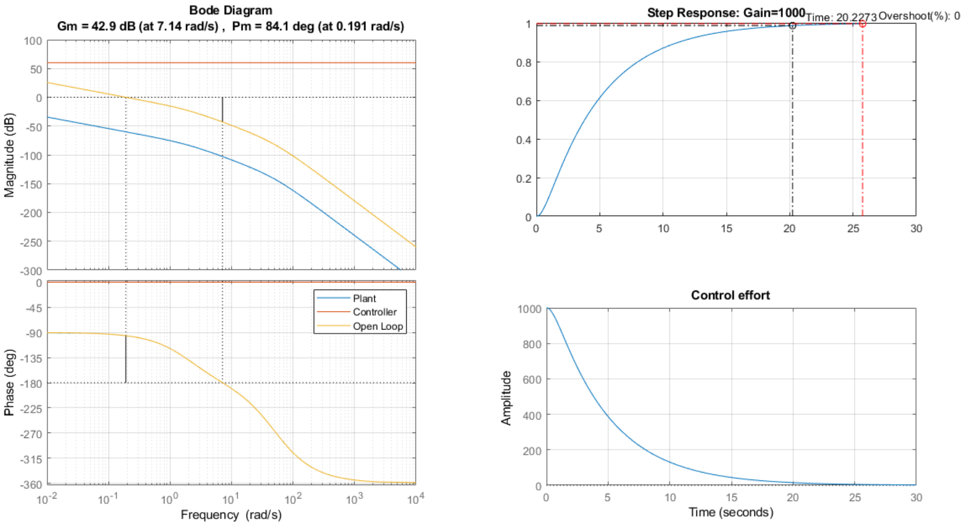

Figure 24: Proportional controller, Bode, response and control effort.

Regarding settling time, 20.3 $s$ is the best we can get without violating the control limits. However since the system only contains proportional gain, it cannot reject the disturbance completely, a marginal 0.1% remains, which can still be very significant for the steel making process.

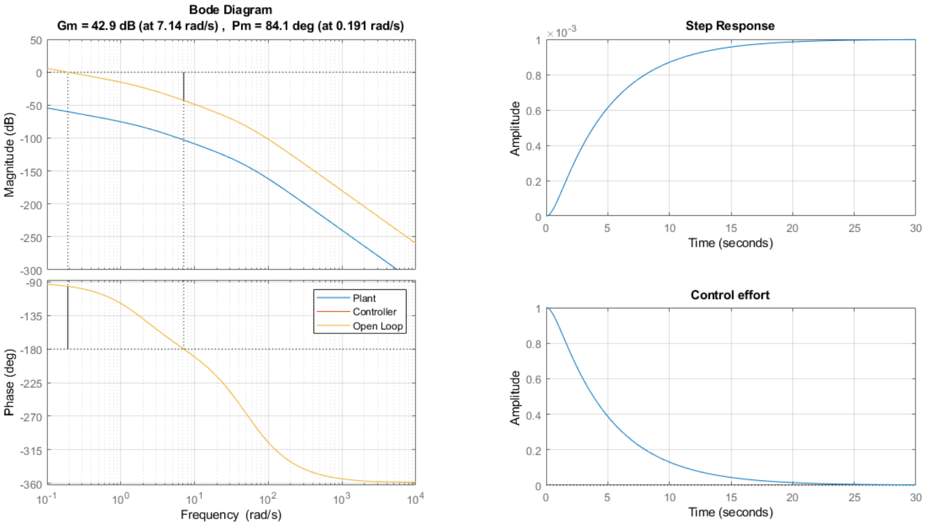

Figure 25: Proportional controller, disturbance rejection.

For this reason a revision of the controller is apparent, we need to include integral action, to cancel out the disturbance, this does slow down the system significantly, doubling the settling-time, while also increasing the initial overshoot from 0, to 23% so the company has to make a trade-off between fast settling time and zero steady state error.

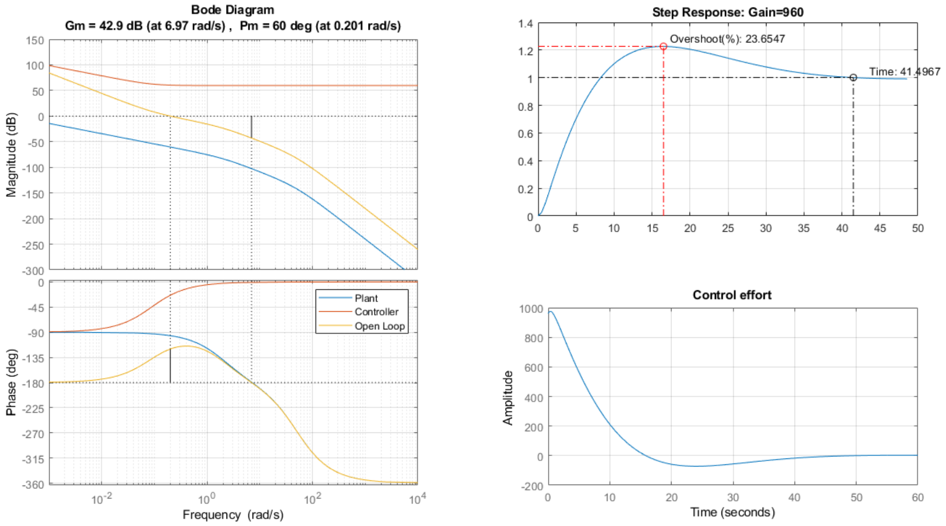

Figure 26: PI controller, step-response.

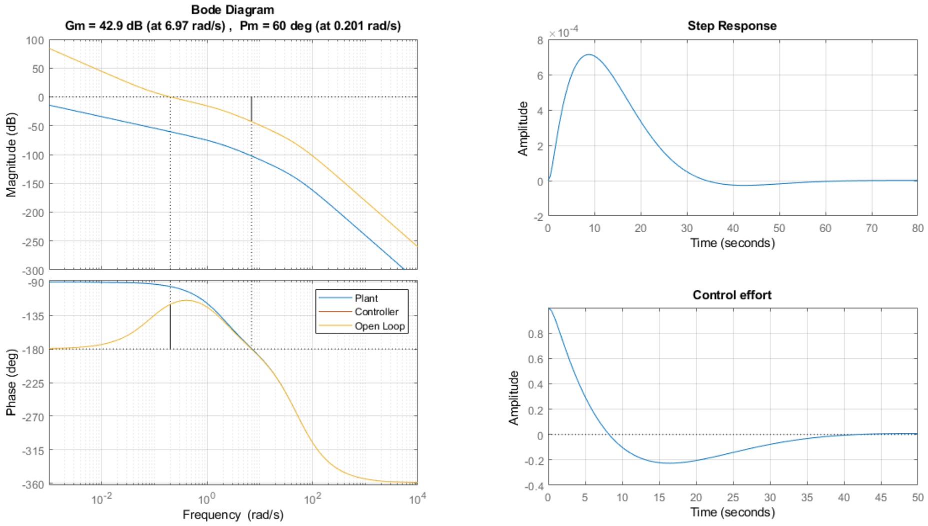

Figure 27: PI controller, disturbance rejection.

As the zero steady state error is supposedly very important for the steel shaping process, we will discretize the PI controller. The method is the same as in Section 4.2, this resulted in the sampling time being $h = 0.53\ s$.

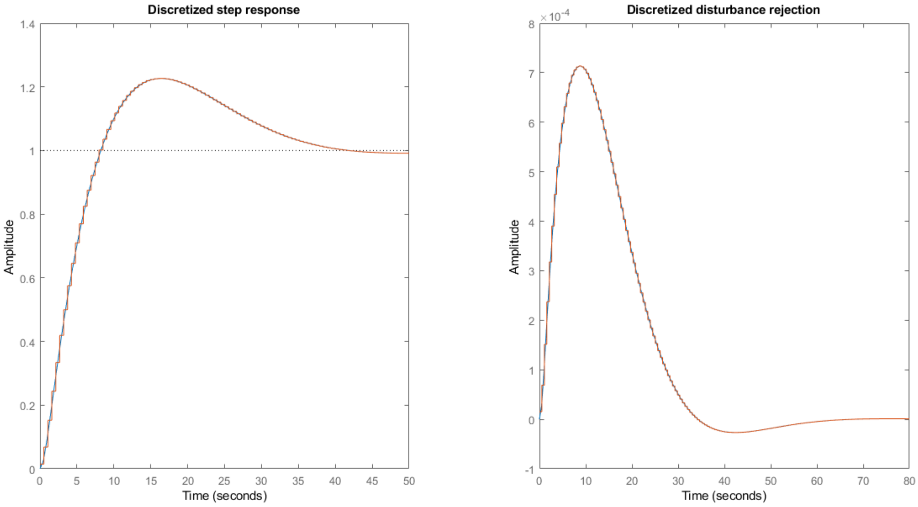

Figure 28: Discretized controller step response and disturbance rejection.

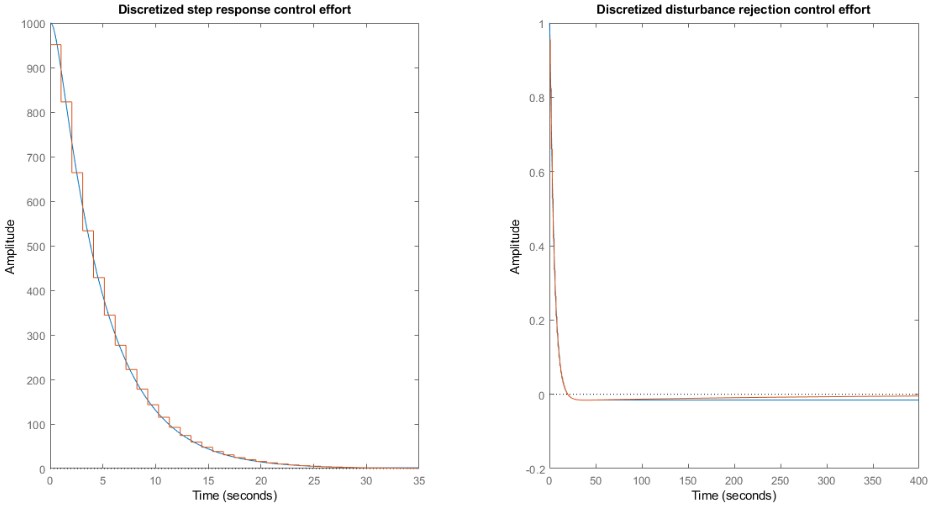

Figure 29: Discretized controller controller outputs.

## 5.2 Redesign of the pole placement controller

As the continuous system response slowed down, the characteristics and dominant poles of the second-order system changed accordingly. Thus we can use the same design methodology as in Section 4.3, as the control effort required is in the same order of magnitude as the limit. Furthermore, non-dominant poles at over 11 times the natural frequency of the second-order system proved to be faster and require less control input. Results can be seen on Figures 30 and 31.

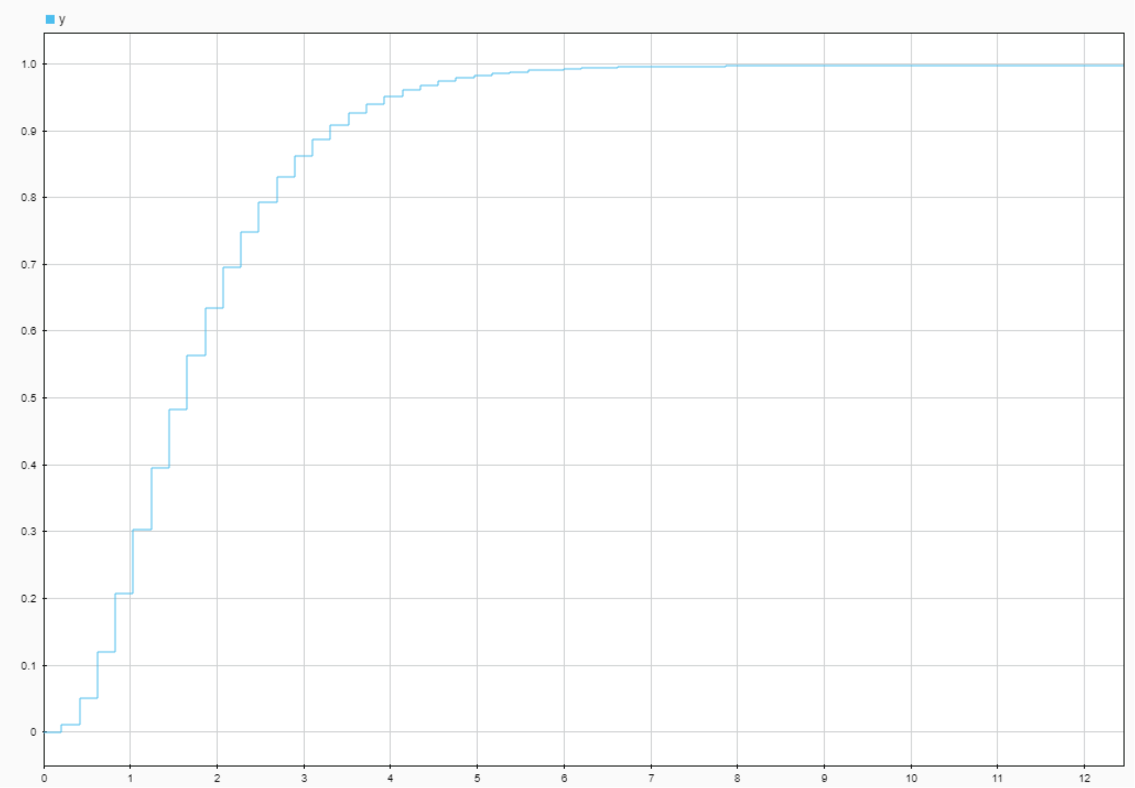

Figure 30: Pole placement step response.

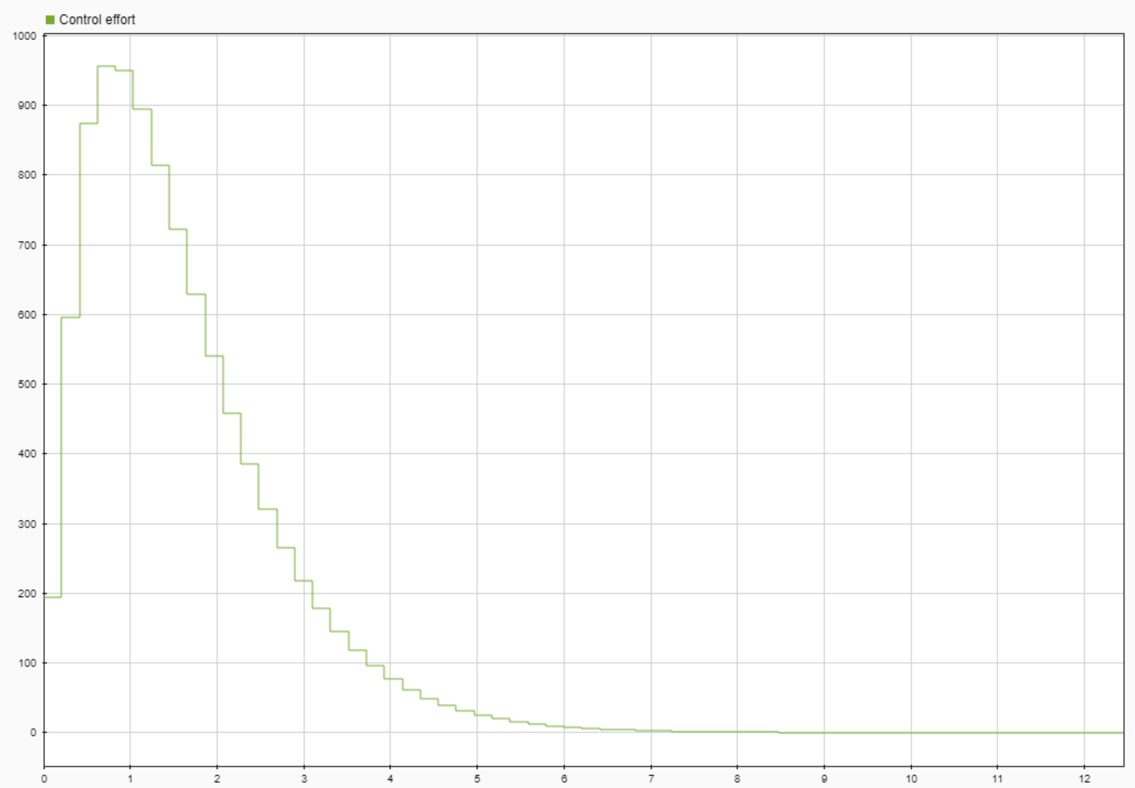

Figure 31: Control effort for the pole placement scheme.

The response is faster than in the P and PI controller case, and the overshoot is 0%.

## 5.3 Redesign of the LQR controller

For the LQR case a much stricter balance between the weighting matrices is required as we need to keep the importance of the input relatively high in order to keep the control action boundaries. $\rho = 10^6$ was used for the weights, with a sampling time of $h = 0.57\ s$.

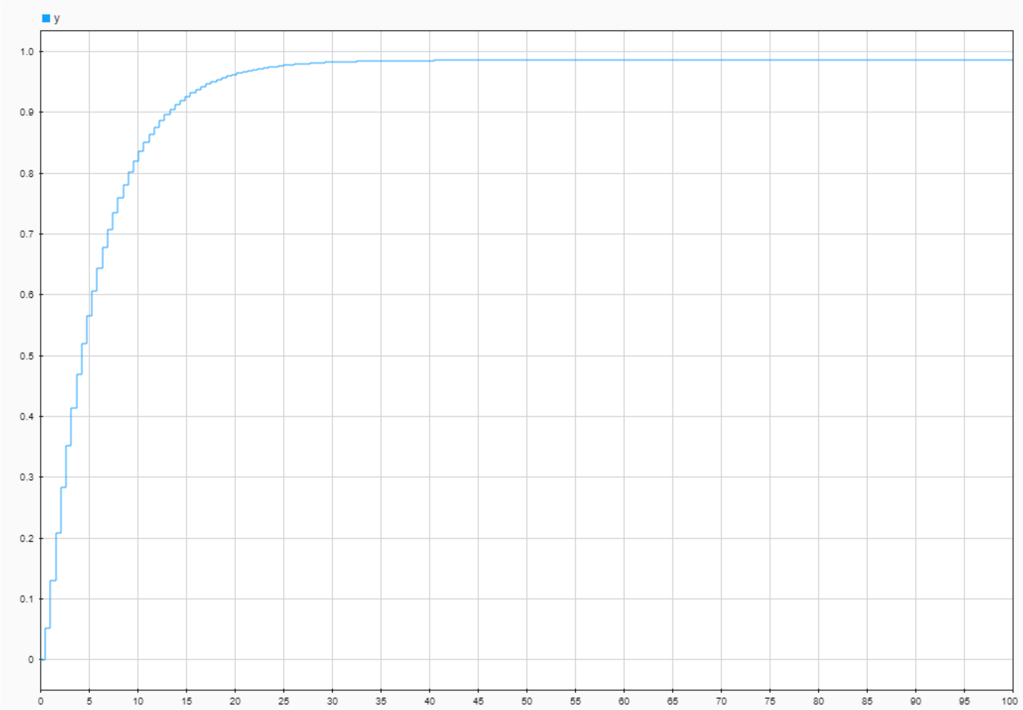

Figure 32: LQR step response.

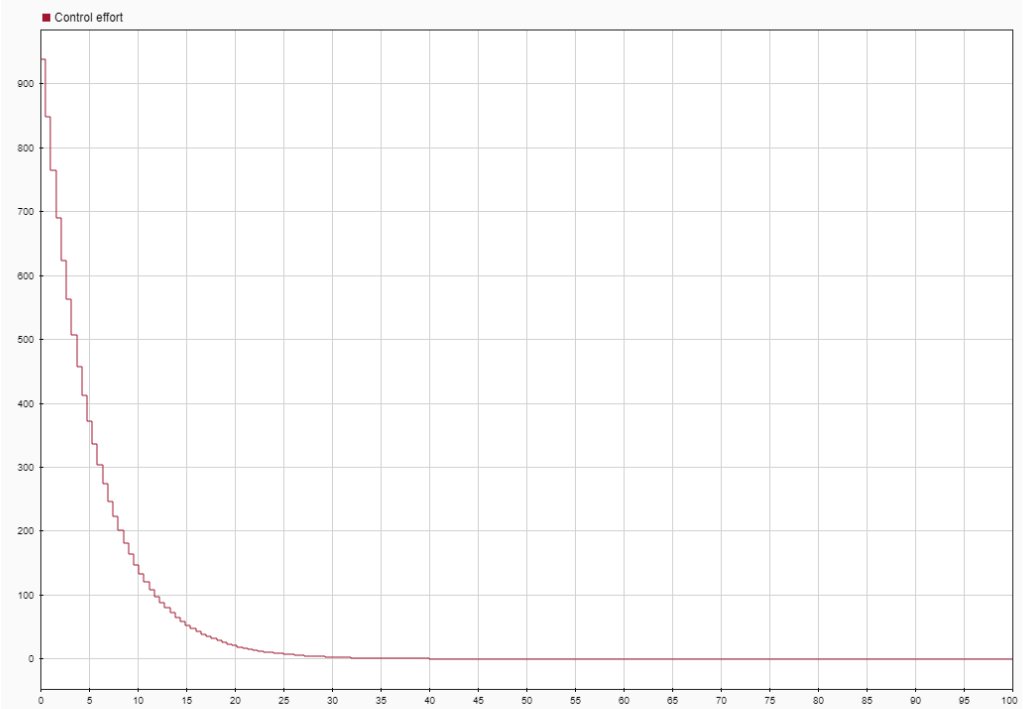

Figure 33: Control effort for the LQR controller.

This resulted in a very similar control scheme to the discretized PI controller.

# 6 Steady-state error elimination

## 6.1 PID controller steady-state error elimination

As shown in Section 5.1 introducing integral action can reduce the effect of an error to 0.

## 6.2 Disturbance observer

When a disturbance acts on the input of the plant, a disturbance observer scheme has to be used, and an appropriate $K_d$ gain has to be chosen. The Simulink model of a disturbance observer is as follows:

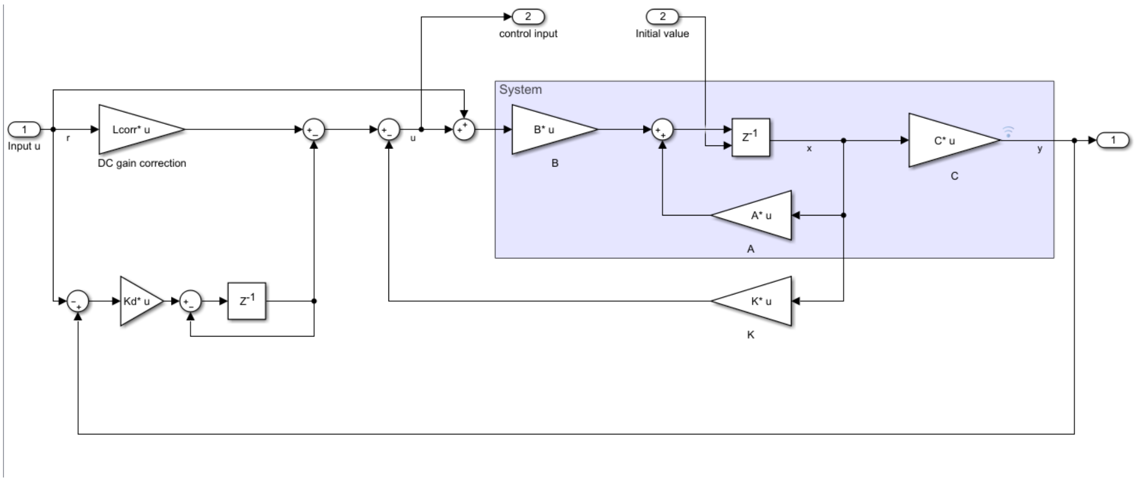

Figure 34: Simulink model of a disturbance observer.

$$\hat{d}_{k+1} = \hat{d}_k + K_d(y_k - r_k) \tag{36}$$

For the pole placement scheme, the following results were obtained, with $K_d = 50$ and small adjustments to the poles to keep the control effort within bounds.

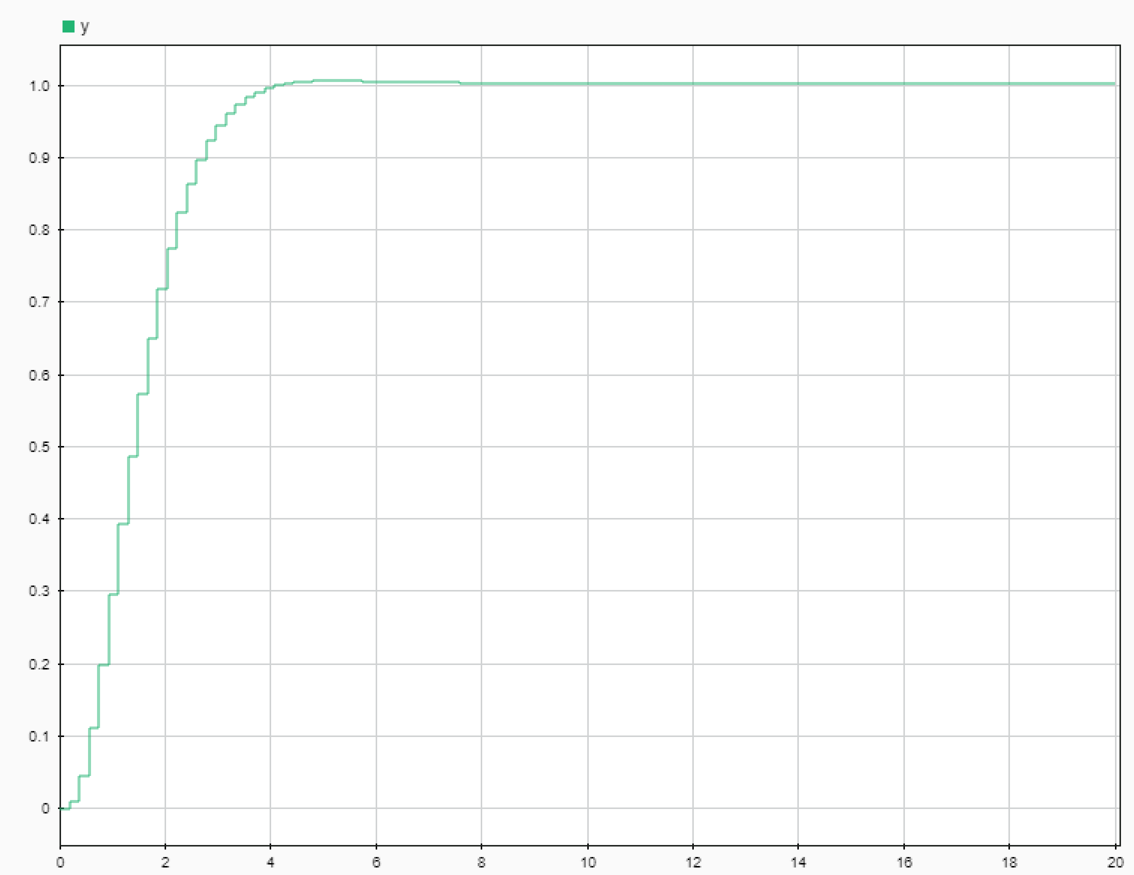

Figure 35: Pole placement method with disturbance observer.

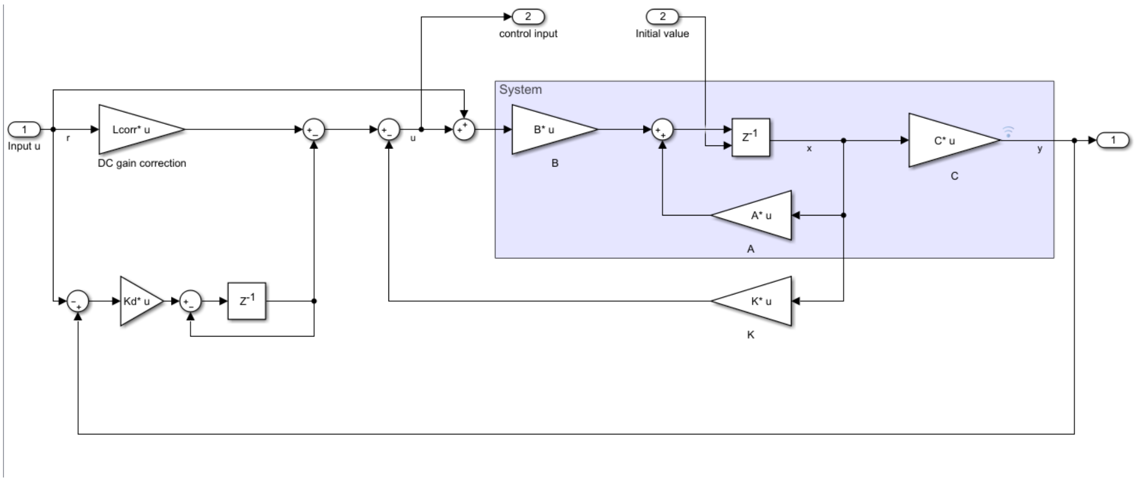

Figure 36: Simulink model of a disturbance observer.

# 7 Delayed control action

Introducing one step delay to the control action will increase the effect of unwanted phenomena, as overshoot, slower response, slower disturbance rejection.

for the PI case, the following results obtained, with $h = 0.53\ s$ :

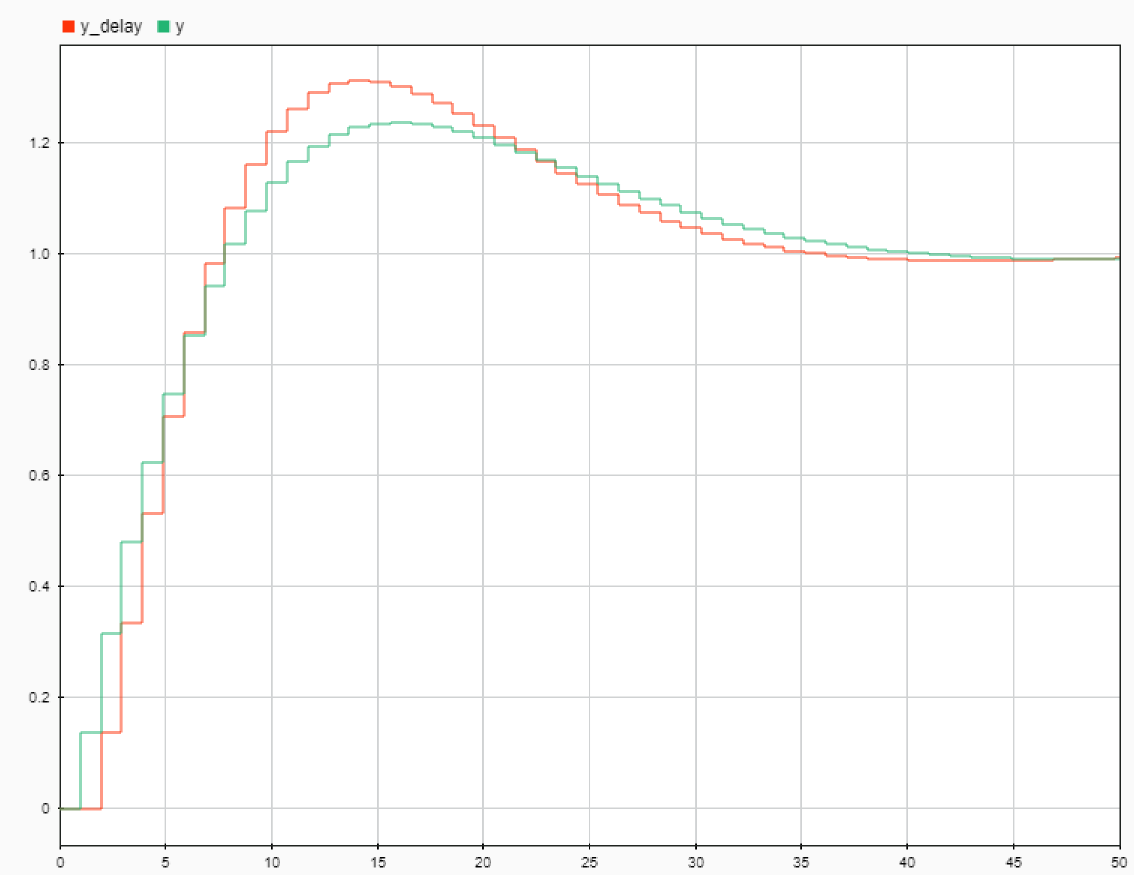

Figure 37: Effect of delay for the original sampling time.

Since the effect of the delay is low, it indicates, that the sampling time of the system is quite low, increasing the sampling time to $h = 1.46\ s$ increases the effect of the delay.

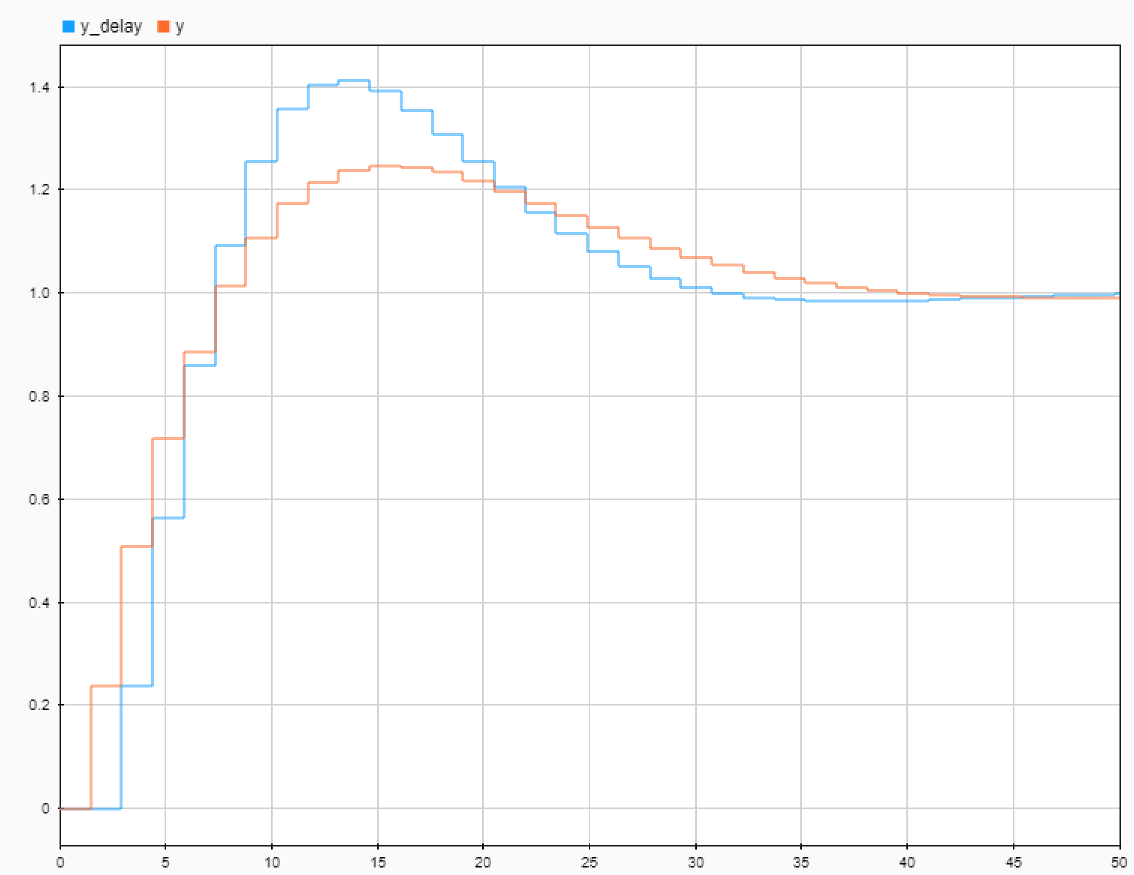

Figure 38: Effect of delay for lower sampling frequency.

In the disturbance rejection case with low sampling frequency, the following response was obtained.

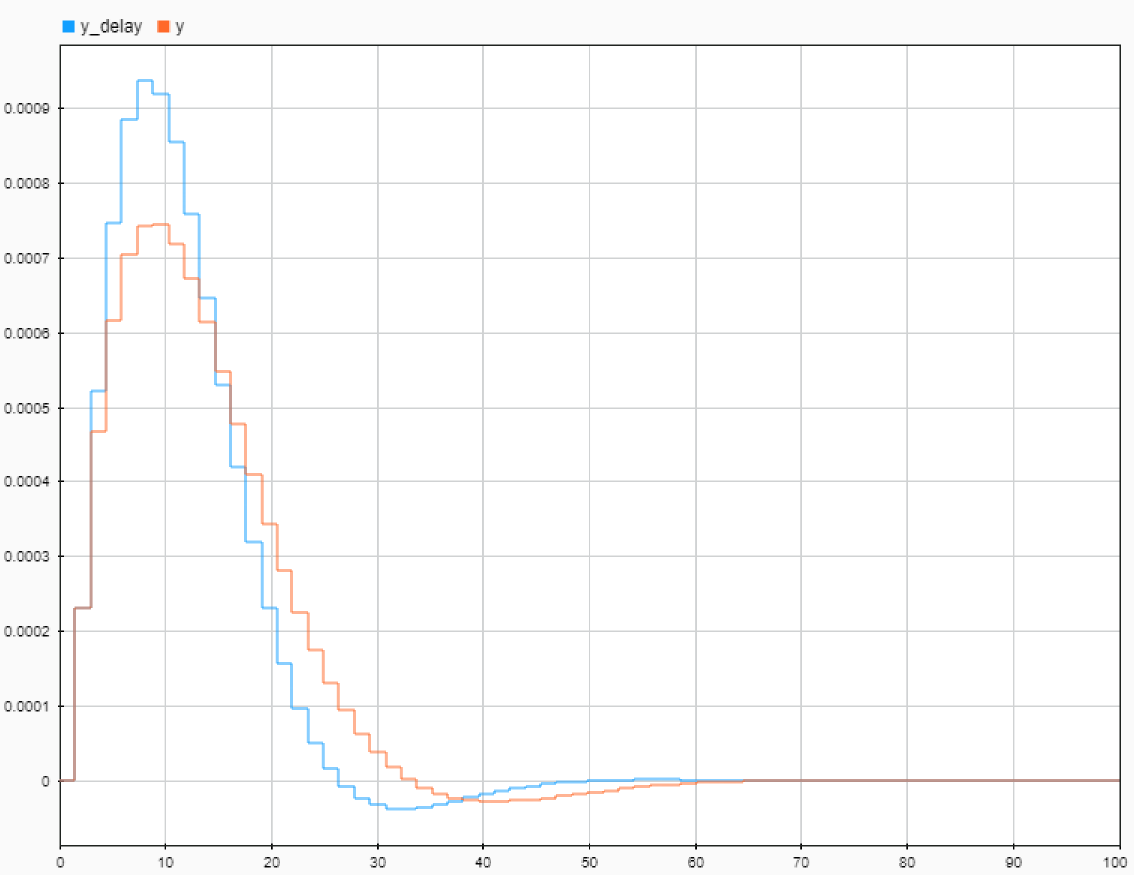

Figure 39: Effect of delay on disturbance rejection for lower sampling frequency.

As we have seen, increasing the sampling frequency mitigates the effect of such delays, in case we have a limit on how high the sampling frequency can be, but still need to tackle a delayed system, the only option is slowing the system down, where the current, or minimum sampling interval is sufficient.

### 7.1 Effect on transfer function

The transfer function of the discrete PI controller is the following:

$$C(z) = \frac{1022\ z - 897.7}{z - 1} \tag{37}$$

Taking the one step delay into account, it changes to the following:

$$C(z) = \frac{1022 - 897.7\ z^1}{z - 1} \tag{38}$$

### 7.2 Effect on state-space representation

In order to model the effect of delayed control by $h$, we need to augment the update equation of the system, and make the next state step (k+1) depend only on $u_{k-1}$ and not $u_k$.

$$\begin{bmatrix} x_{k+1} \\ u_k \end{bmatrix} = \begin{bmatrix} \Phi & \Gamma \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_k \\ u_{k-1} \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u_k \tag{39}$$

## 8 Conclusions

The control of a steel-rolling mill is an interesting and challenging task, as we moved through the exercises, from nearly ideal case, we arrived to heavily limited, but much more realistic controller designs, we described and solved different goals, and tested which approaches are best for solving said problem, while still making the necessary trade-offs.

In continuous-time a balance between fast settling-time and overshoot had to be found, which is an interesting design procedure in itself, disturbances, and their cancellation was also introduced here.

Discretizing the continuous controllers showed that the choice of discretization method has significant effect on the achieved system response. Another important aspect was the importance of sampling time, as a crucial part of discrete systems.

The pole placement method showed, that even complex systems can be simplified to much easier dynamics, where desired characteristics can be achieved, supposing proper choice of non-dominant poles.

The observer scheme showed a more realistic approach, where we don't have full information of the states, which is often the case. But still able to obtain information, and eventually calculate the state evolutions correctly.

The LQR problem gave an opportunity to specifically choose the importance of and balance between the state aggression and control effort.

Actuator saturation is a very important aspect of systems in real-world applications, the heavy limitations on the control effort showed how hard it actually would be to control such a manufacturing device with maximum efficiency, and also how important trade-offs are in such applications.

For such a process, elimination of the steady state error is very important, for any and every controller, otherwise manufactured steel won't fulfil the requirements towards customers.

Another important aspect is the phenomena of control delay, which further steered the design of controllers towards a more realistic path.

In conclusion, the design concepts and procedures were interesting, more and more realistic and provided the author with a better understanding of digital control design.