Digital Control

Report

SC42095: Digital Control
Project 26

Delft University of Technology

TUDelft

Digital Control

Report

by

Project 26

Student Name	Initials	Student Number
1 Mihaly Fey	M.	5476747

Tutors:
Coaches:
Project Duration: - , 2025
Faculty: Mechanical Engineering, Delft

Contents

1. Continous time design

A high-speed steel rolling mill with variable thickness output, has a transfer function in Equation 1.1. The design task is for a continous-time controller with minimal settling time and disturbance rejecction ha to be designed.

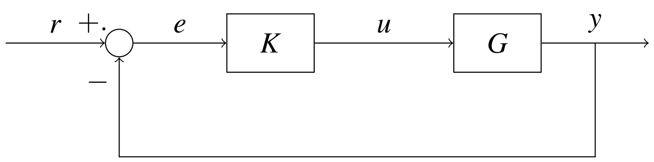
$$G(s) = \frac{1}{s(s+2)(s^2+100s+2600)} \tag{1.1}$$

1.1. Design for reference following

The requirements are for the step input reference are the following:

- Minimal settling time
- Overshoot <5%
- Steady state error = 0

The block diagram of the system is then the following:



Approximating the closed loop system as a 2nd order one, a formula for the damping ration can be written from the overshoot, in Equation 1.2. From this the two corresponding poles of the transfer function need to have $Re(pole)/Im(pole) < 0.07013$

$$\zeta = \sqrt{\frac{\ln(OS)^2}{\pi^2+\ln(OS)^2}} = \sqrt{\frac{\ln(0.05)^2}{\pi^2+\ln(0.05)^2}} = 0.69011 \tag{1.2}$$

The plant includes a term $\frac{1}{s}$, an integrator, which means it already has 0 steady state error. Therefore an integrator term is not necessary for the reference following controller. It will be however, for disturbance rejection. The proposed controller structure is therefore a PD controller, taking the form in Equation 1.3. According to this formulation there is a zero to be placed freely and a pole. To cancel out a slow zero in the plant this zero can be placed at -2. As such $\tau_D + T_d = 0.5$. τ_D is usually chosen to be around 0.1-0.2 times that of T_d [1]. For exact values, $\tau_D = 0.05, T_d = 0.45$ were chosen. This gives the pole at -20.

As only on parameter, K_p is left to tune, the root locus method can be used to achieve the overshoot target. Looking at the root-locus map in Figure 1.1, 3 significant poles can be seen, two of which have imaginary components, and one at -2, which is then cancelled out by the pole in the PD controller. This then means that the two poles left will dominate the step response and determine the overshoot. The two poles were then put to to the 0.69012 value to achieve the desired overshoot. As $\frac{\tau_D+T_d}{\tau_D} = 10$, moving the roots around gives $K_p = 2.98 \cdot 10^4$ for a damping ratio of $\zeta = 0.690$. This gives a final settling time of 0.7002 and an overshoot of 4.8532.

$$K_p(1+\frac{T_d s}{\tau_D s+1}) \tag{1.3}$$

$$K_p(1+\frac{T_d s}{\tau_D s+1}) = K_p(\frac{(\tau_D+T_d)s+1}{\tau_D s+1}) = \frac{K_p(\tau_D+T_d)}{\tau_D}\left(\frac{s+\frac{1}{\tau_D+T_d}}{s+\frac{1}{\tau_D}}\right) \tag{1.4}$$

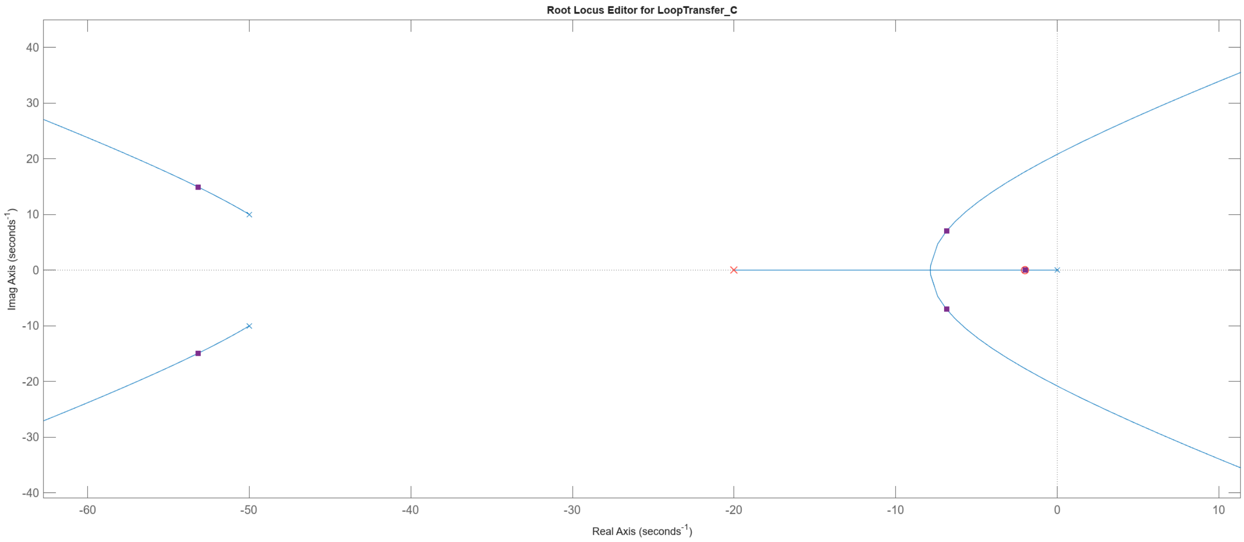


Figure 1.1: Root-locus of the closed loop with PD controller

1.2. Design for disturbance rejection

The controller in the previous section exhibits no disturbance rejection due to the lack of I-term. This can be seen on the step-response in Figure 1.2, where a steady state error exists. Adding an integrator to the PD formulation results in Equation 1.5, that is multiplying the original PD with $1+\frac{I_s}{s}$. This gives one more zero to place in the system. As the lack of the integral term is what caused the steady state error in case of disturbances increasing the gain of the integral will make the action faster. Increasing the gain in the formulation corresponds to moving the added away from 0, but increases the settling time of the system. The root-locus of the closed loop system is in Figure 1.3. It can be seen that there is a slow pole on the real axis, moving the added zero close to zeros moves this closer to the added zero, meaning an added zero close to 0 will improve reduce the magnitude of the mode corresponding to the plot. At the same time the zero moves the two imaginary poles close to the imaginary line, towards instability, so it can no be too close to 0. The balance lies somewhere around -2, so that is selected as the added zero. The tuning is then only the gain of the the controller, which is selected with on the root locus map and then rounded to 1100000. The peak of the response is $4.2400e-06$ and the settling time is 3.4789 seconds. This step response can be seen in Figure 1.4.

As this controller is tuned for disturbance rejection with not regular step response being in mind and the step response will not a very low settling time. This can be seen in Figure 1.5.

$$K_p\left(1+\frac{T_d s}{\tau_D s+1}\right)\left(1+\frac{I_s}{s}\right)=K_p\left(1+\frac{T_d s}{\tau_D s+1}\right)I_s\left(\frac{\frac{1}{I_s}s+1}{s}\right) \tag{1.5}$$

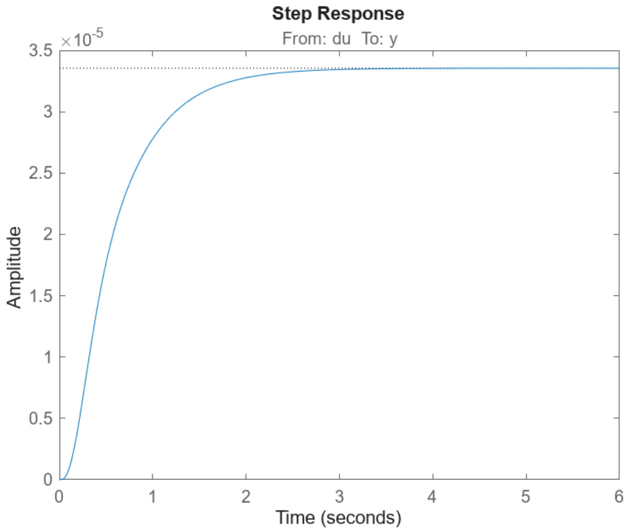


Figure 1.2: Disturbance rejection for the reference following controller

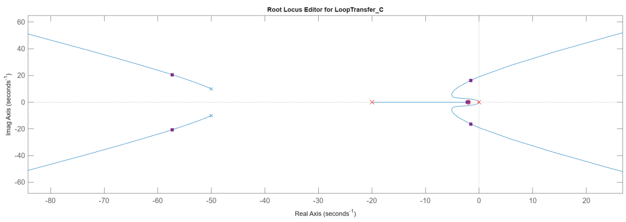


Figure 1.3: Pole-zero map of the system with disturbance rejection

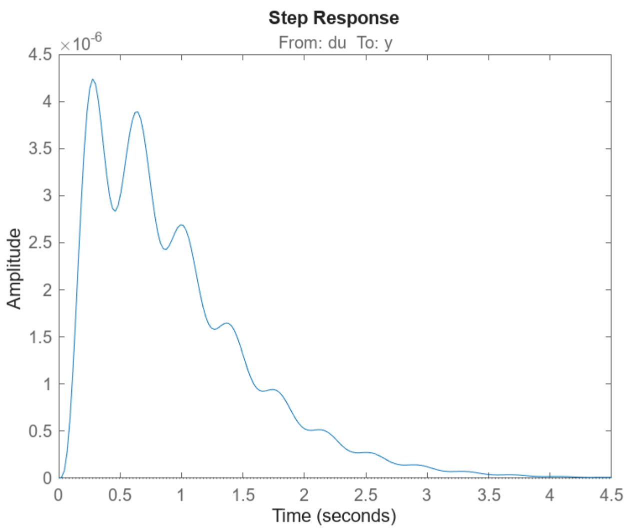


Figure 1.4: Response to a unit step disturbance input

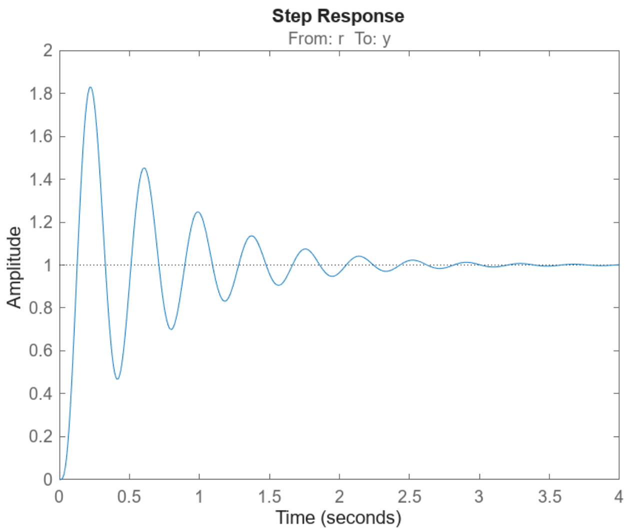


Figure 1.5: Response of the PID controller to a unit step change in the reference input

2. Discrete time modelling

The continuous time model is now discretized and all kinds of stuff are done in discrete time now.

2.1. System description

The continous time model of the high-speed steel-rolling mull in the previous section was only used in state-space from, so it is now realized in a state-space from to make the discrete model creation easier. This has to be dones such that the state-space description in Equation 2.2, which can be achieved with the controllable canonical form in Equation 2.1, as in the lecture slides of SC42015 Control Theory course. Applying this transformation to the plant transfer function in Equation 1.1 results in the matrices in Equation 2.3.

$$G(s) = \frac{\beta_{n-}s^{n-1} + \cdots + \beta_1 s + \beta_0}{s^n + \alpha_{n-1}s^{n-1} + \cdots + \alpha_1 s + \alpha_0} + d_0$$
$$A = \begin{bmatrix} -\alpha_1 & -\alpha_2 & -\alpha_3 & \dots & -\alpha_n \\ 1 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \dots & 1 & 0 \end{bmatrix} \qquad B = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{2.1}$$
$$C = \begin{bmatrix} \beta_0 & \beta_1 & \dots & \beta_{n=1} \end{bmatrix} \qquad D = \begin{bmatrix} d_0 \end{bmatrix}$$

$$\begin{aligned} \dot{x} &= Ax + bu \\ y &= Cx + Du \end{aligned} \tag{2.2}$$

$$A = \begin{bmatrix} -102 & -2800 & -5200 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \qquad B = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{2.3}$$
$$C = \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix} \qquad D = \begin{bmatrix} 0 \end{bmatrix}$$

2.2. Choice of sampling time

Before the continuous time system is made discrete a sampling time has to be selected. This was done with two methods two ensure the validity.

Sampling based on eigenvalues First the eigenvalues of the system are analyzed, which are $\lambda = 0, -50+10i, -50-10i, -2$. With four distinct eigenvalues the system trajectory will be a linear combination of the four exponential functions e^{λ}. To avoid aliasing the sampling rate must be at least twice the frequency or rate of change of the fastest mode, which will correspond to the largest magnitude eigenvalue. Take this as the nyquist frequency, multiply it by two and divide 2π by it to get the sampling frequency. Round it down for safety, which gives a sampling period of 0.0569

Sampling based on system approximation The sampling frequency can also be derived from the bandwidth of the system, as $f_s = 20 f_b$, the bandwidth frequency. As the gain of the system is always negative on the Bode plot the 180° crossing can be looked at in Figure 2.1. The crossover freqency is 7.14 rad/s, so the sampling frequency should be 142 rad/s and the period 0.0440 s. This is quite a bit lower than the previous estimate, so this is used going forward in the exercise.

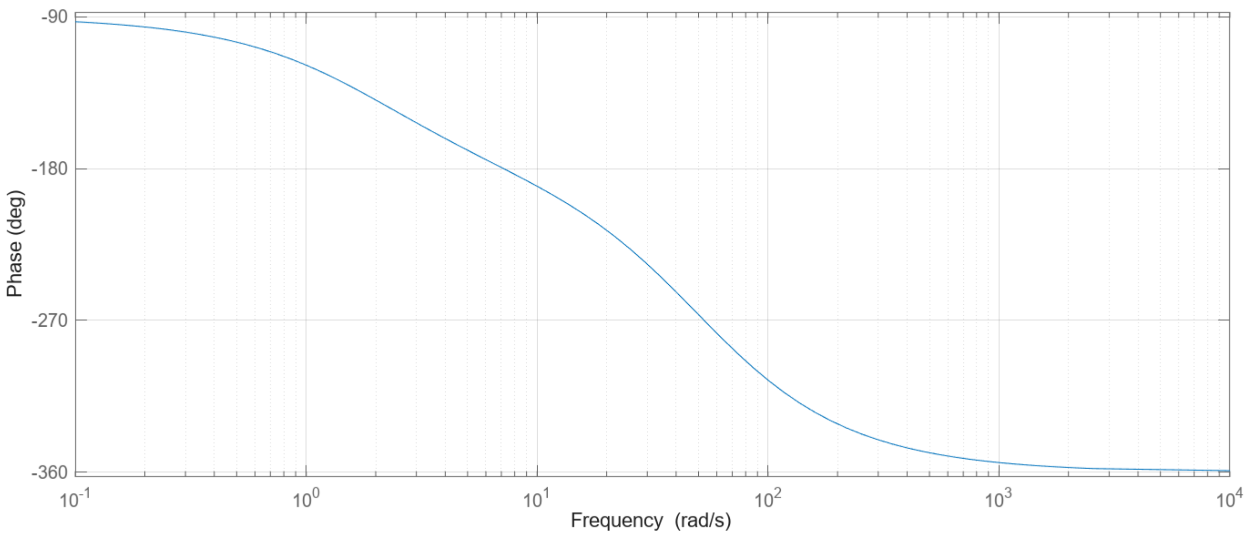


Figure 2.1: Frequency bode plot of the open-loop continuous time system

2.3. Discretization

2.3.1. Discretization method

With the sampling frequency now selected a discretization method has to be selected. The evaluation of each options are the following[2]:

- **Impulse-invariant method** The impulse invariant methods preserves the impulse response for a time step across the discrete and the continuous time systems. This makes for accurate discrete representation, when the control inputs are closer to impulse inputs than to anything else. When the inputs are more continuous in time, a step-invariant or ramp-invariant method is more accurate and better (most of the time).
- **Step-invariant methods** The step-invariant method focuses on preserving the step response of the continuous-time system in the discrete-time domain. It performs well in systems which use the zero-order-hold in the real controller as well, which are simple, easy to implement and are widely used. While it provides accurate time-domain behavior for ZOH-inputs, it has "unnecessarily bad frequency response" [2], so frequency-domain design with it is suboptimal.
- **Ramp-invariant methods** The ramp-invariant method is a less commonly used discretization approach that ensures the ramp response of the continuous-time system is preserved in the discrete-time domain. This method aligns the sampled ramp response values of the continuous system with the discrete system, making it more suitable for continuous-input systems than the step-invariant method. As with all previous methods this one also keeps the location and stability of the sampled poles.
- **Tustin's approximation** Tustin's approximation is a bilinear transform from between the s and the z domains, so it can only be applied for transfer functions and not state-space models. $z = e^{sh}$ is approximated with $\frac{1+sh/2}{1-sh/2}$. This keeps the stability of all the poles of the system and is easy to use. There due to the inexact mapping of the z to s the location and therefore the response of the poles is changes slightly, but stability is preserved for all of them. The shifted pole locations affect the frequency response of the sampled system, which do not harmonize well with filters designed in the frequency-domain. To avoid this the transfer function can be multiplied with a term containing s or z, whichever way the conversion is going, to "pre-warp" the frequencies, which the do not shift. Keeping the frequency-response reasonably accurate makes it ideal for discretization of a controller/filter designed with frequency-domain design.

For this exercise two methods are selected. For physical, actuated systems the step-invariant discretization is selected assuming zero-order hold actuation, while for computer controllers and systems that are digital

Tustin's approximation is used, to get the discrete time behavior as close as possible to the continuous time one.

2.3.2. Discretization of a state-space system

the discrete time system can now be computed. The formulation is in Equation 2.4. The C and D matrices are the same as in the continuous case. The derivation of each matrix is in Equation 2.5. The matrices A and B, are the same as in the continuous case, and the integration was done numerically to get an approximation of the matrix.

$$x(k+1) = \Phi x(k) + \Gamma u(k) y(k) = Cx(k) + Du(k) \tag{2.4}$$

$$\Phi = e^{Ah} m \quad \Psi = \int_0^h e^{As} ds\, B \tag{2.5}$$

2.3.3. Discretization of the controlled system

The state-space realization of the plant was converted to the discrete time with the above zero-order hold method, while the controller was converted with Tustin's method. State-space realizations of both the controllers made for continuous time and the plant were made, which then were discretized. Both a reference and a disturbance step were applied on the respective controllers, the responses of which are in Figure 2.2. The reference step does still track, and the settling time does not change very much, at 0.7040 seconds, but the overshoot constraints are violated at 11.28. The disturbance rejection controller on the other hand has is unstable in the closed loop, the two poles placed carefully in the root locus became unstable, which were stable for the continuous time system.

2.4. Discrete-time pole placement

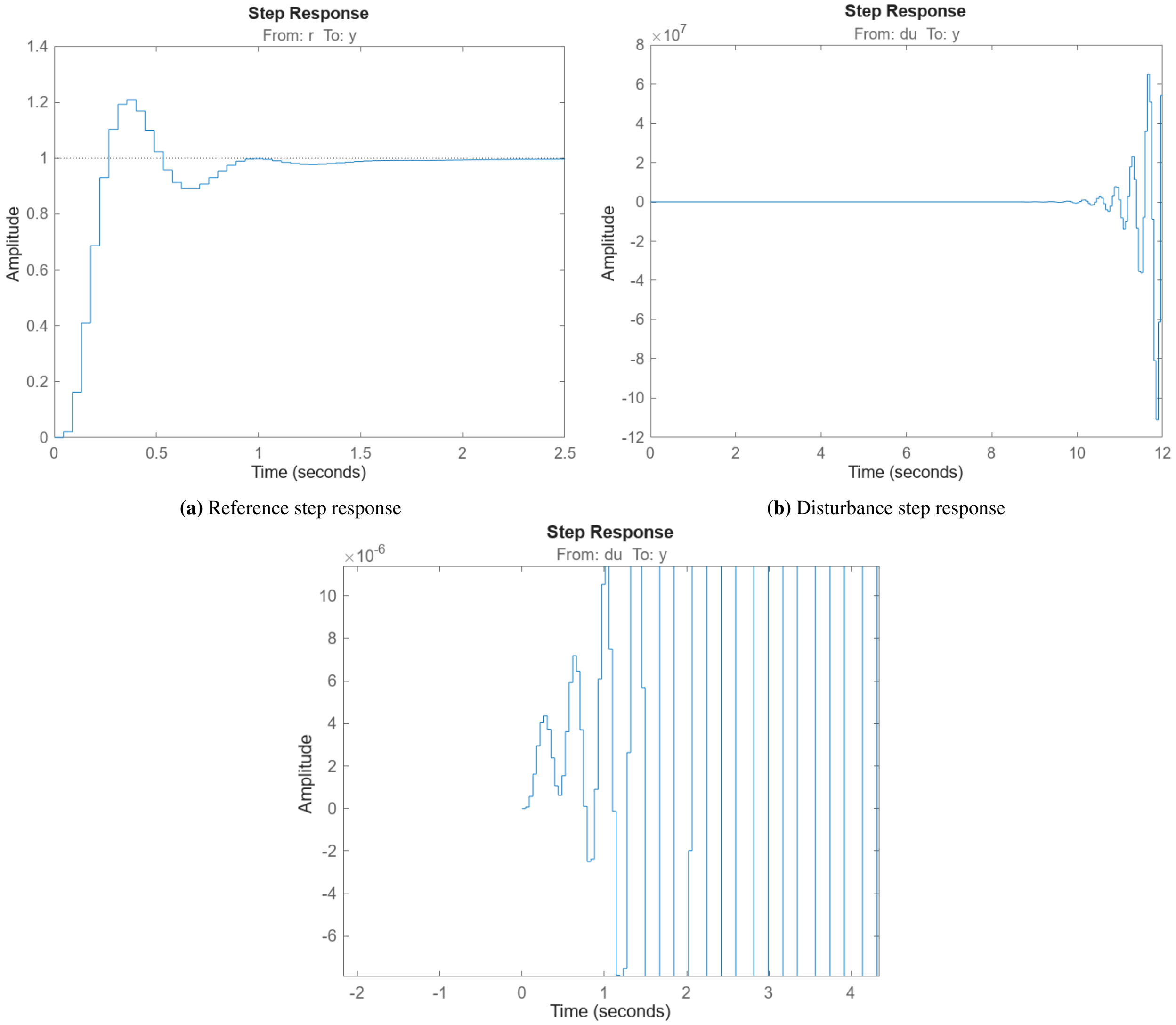
Bad performance of the controllers necessitates a different type of controller, designed for the sampled state-space system. A good method is a full state-feedback linear controller, the gains of which are determined by pole-placement.

Such a tracking controller has a linear feedback term and since the steady gain of system might not be unity a feedforward amplifier linear term also has to be included on the reference. The block diagram of such a system is shown in Figure 2.3. The plant input u in such a system is in Equation 2.6, so the whole system from r to y simplifies to the form Equation 2.7. If such a system is controllable, then the feedback gain K can be chosen such that the eigenvalues of $(\Phi - KB)$ are in arbitrary places. Such a system is controllable when the matrix $[\Phi \quad \Phi\Gamma \quad \Phi\Gamma^2 \quad \Phi\Gamma^3]$ has full row rank. In the case of high-speed steel-rolling mill this is the rank of this matrix is 4, so such a controller can be realized.

$$u = u_{ff} - u_{fb} \qquad u_{ff} = K_{ff} r \qquad u_{fb} = Kx \tag{2.6}$$

$$\begin{gathered} x(k+1) = (\Phi - \Gamma K)x(k) + BK_{ff} r \\ y(k) = Cx(k) + DK_{ff} r \end{gathered} \tag{2.7}$$

The gain for a set of poles can be calculated with matlabs `place` function accomplishes that, so only the poles have to be selected. Three sets of poles were selected with continuous time in mind, then the poles were put into discrete time with $p_z = e^{p_s T_s}$. The step response data for each of them can be found in Table 2.1. The plot of the step response is in Figure 2.4. From these the second set of poles perform best, due to the smaller time constants/faster decay rates. In theory due to no constraints being given on control inputs or anything that could limit the system performance the selected poles could be infinitely small, leading to practically instantaneous decays, which necessitate infinitely high-frequency sampling. This is not done, writer did not see any point in doing it, instead something that is closer in performance to the continuous time PID controller is selected.



(a) Reference step response

(b) Disturbance step response

(c) Zoomed in disturbance step response

Figure 2.2: Step responses of the continuous-converted systems

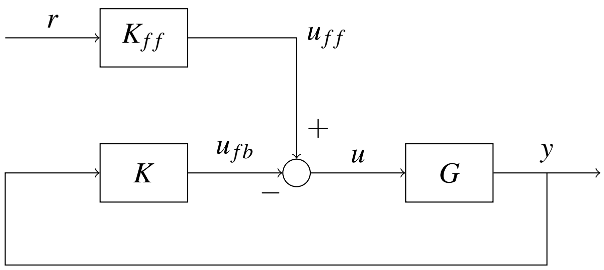


Figure 2.3: Block diagram of the full-state feedback controller

Continuous time poles	Discrete time poles	Rise time	Settling time	Overshoot	K
$-53.15 \pm 14.9i$ $-6.85 \pm 7.04i$	$0.0764 \pm 0.0588i$ $0.7046 \pm 0.2255i$	0.2200	0.6600	4.5583	[-3375.4, 153.08 6.8830, 12.0728]
-53 ± 1 -14114.2i	$0.4381 \pm 0.3159i$ 0.0929, 0.1015	0.1320	0.3520	3.5925	[-8123.7, 957.40 166.08, 19.989]
-60 ± 10 $-7.5 \pm 7.4i$	$0.6812 \pm 0.2300i$ 0.0460, 0.1108	0.2200	0.6160	3.9236	[-29012, 5266.7 31.9612, 13.267]

Table 2.1: Poles and response data for the pole-placement controller

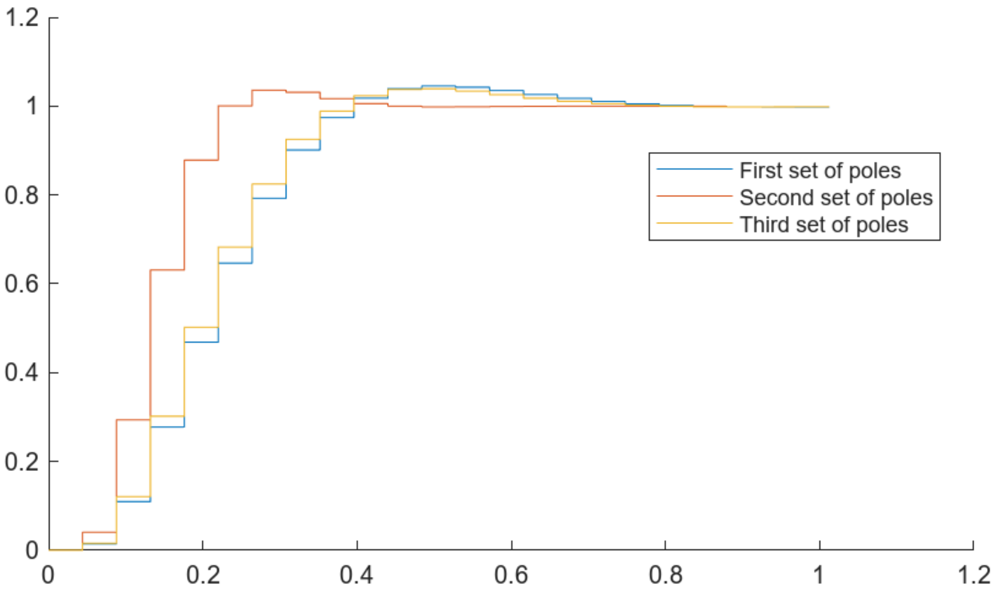


Figure 2.4: Step responses of the full-state feedback controller with the three different sets of poles

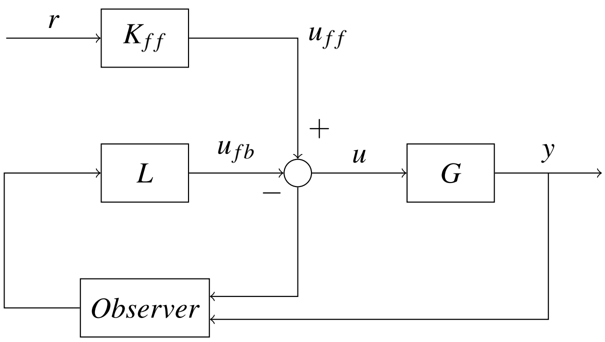


Figure 2.5: Block diagram of the full-state feedback controller

2.5. Output-feedback control

The full-state feedback controller designed assumes perfect information about the full state. The only information of the system available is however only the output. As such for a full state-feedback controller to work the states have to be reconstructed with a linear observer. The block diagram of this setup is Figure 2.5. The equation of the observer is in Equation 2.8. The complete system has now twice as many states $[x\,\hat{x}]^T$, for which one combined state-space model can be set up, Equation 2.9. After a state-coordinate change the form in Equation 2.10 is also revealead, through which it can be seen that the eigenvalues of the complete system are the union of the eigenvalues of $(\Phi-\Gamma L)$ and $(\Phi-KC)$. This means that through two separate pole placements, unrelated to each other, all 8 eigenvalues can be designed for. This equation still includes the feed-forward term H_{ff}, which is the inverse of the DC gain of the whole system without the feedforward term.

$$\begin{aligned}\hat{x}(k+1) &= \Phi x(k) + \Gamma(u(k) + K(y(k) - \hat{y}(k))) \\ \hat{y}(k) &= C\hat{x}(k)\end{aligned} \tag{2.8}$$

$$\begin{bmatrix} x(k+1) \\ \hat{x}(k+1) \end{bmatrix} = \begin{bmatrix} \Phi & -\Gamma L \\ KC & (\Phi - KC - \Gamma L) \end{bmatrix} \begin{bmatrix} x(k) \\ \hat{x}(k) \end{bmatrix} + \begin{bmatrix} \Gamma H_{ff} \\ \Gamma H_{ff} \end{bmatrix} u(k) \tag{2.9}$$

$$\begin{bmatrix} x(k+1) \\ e(k+1) \end{bmatrix} = \begin{bmatrix} (\Phi - \Gamma L) & \Gamma L \\ 0 & (\Phi - KC) \end{bmatrix} \begin{bmatrix} x(k) \\ e(k) \end{bmatrix} + \begin{bmatrix} \Gamma H_{ff} \\ 0 \end{bmatrix} u(k) \tag{2.10}$$

-	Settling time	Overshoot(%)	Undershoot(%)
Full state feedback	0.3358	4.0066	2.9238
Output feedback	0.3850	37.4555	41.6222
Full state feedback, double sampling rate	0.4720	24.3645	477.8283
Output feedback, double sampling rate	0.3644	11.7205	6.4461

Table 2.2: Poles and response data for the pole-placement controller

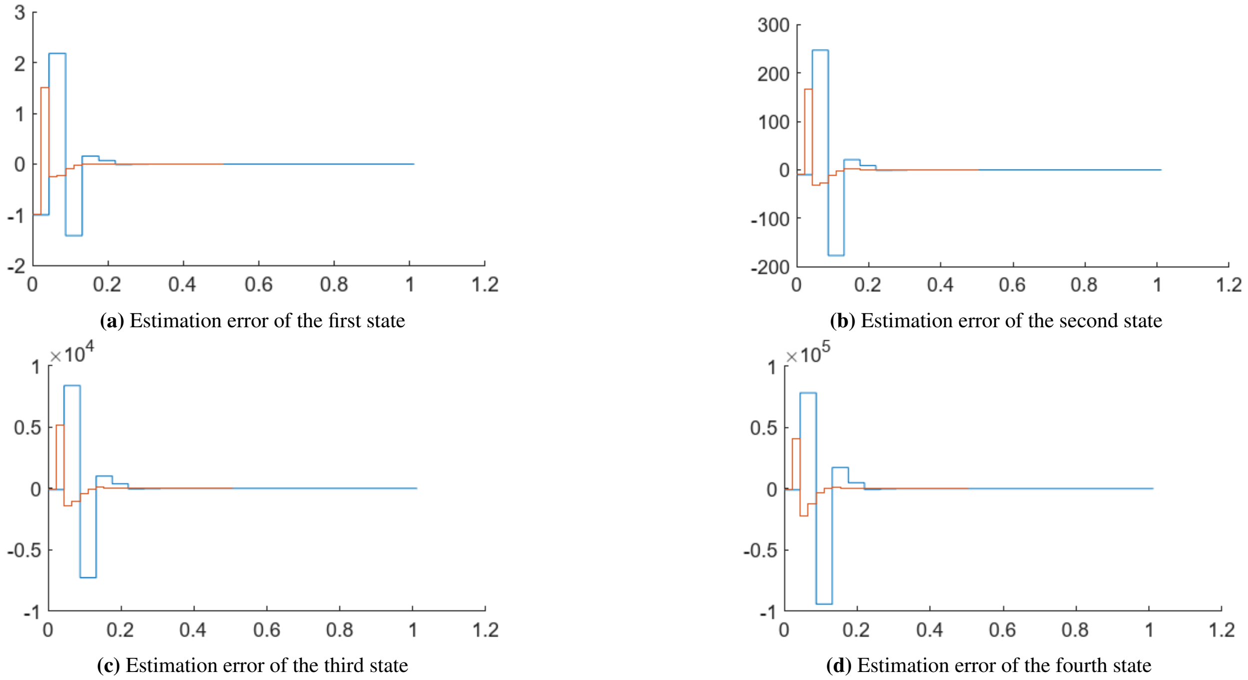
Selecting the values for K and L was again done with the matlab `place` function. Continuous time poles for L were kept the same as in the previous exercise, and for K they are usually a couple of times the magnitude of the specification was five times magnitude of L for fast convergence of the estimate. The downside of this is amplifying measurement noise, which are not present in this exercise, so this is perfectly fine.

The performance of the output-state feedback was compared against the full state feedback with an initial starting state of $[1\ 10\ 100\ 1000]^T$ with the observer being initialized at 0. At this point it was noticed that the systems converge on very quickly compared to the sampling time, in less than 10 steps, so the step responses were investigated with double sample rates, for both the full-state and the output feedback controllers at 0.0220 seconds. The plot of the responses are in Figure 2.7 and numerical data is in Table 2.2. The error of each state estimate by the observer is in Figure 2.6. Observations about the responses and the errors are the following:

- Both observers converge to 0 estimation error quickly for all states.
- The double sampling rate observer converges much faster to the true states, enabling a better tracking.
- The performance of the double rate output feedback controller is improved a lot, as the nonzero initial state decays and the the estimation error quickly decays to 0. After that, performance is very similar to the single rate full state feedback controller, suggesting that increasing the sampling rate does not actually increase tracking performance directly, but does improve the estimation accuracy, which then leads to the increased performance.
- The double rate full state feedback has awful performance, there is over 400% undershoot, which is an order of magnitude worse than any other error. The initial state is a very unstable one and the system begins moving to 0 very quickly, which is the amplified by the control inputs, which are only of this magnitude in the perfect state knowledge.

2.5.1. Output-feedback control with disturbances

Adding a disturbance to the system is detailed in section 3.2, with the resulting combined formulation in Equation 2.11. The combined state-space model encompasses the system state, the observer state, the integrator and the disturbance as well. Plotting the disturbance step response of this system with L_{int} set to zero, meaning no integrator action, is in Figure 2.8a. The steady state error can clearly be seen. Adding an integrator action with the the integrator weight is set to 0.004 for stability reasons, from section 3.2 the disturbance step response is in Figure 2.8b. The steady-state error then decays to zero, meaning the integrator action works. One side effect of the integrator action is that it adds a very slow pole. For this reason the gain was increased, until some oscillation was seen in the disturbance response and the step response still settled in reasonable time. This was tested with the double sampling rates, with the nonzero initial condition being $[1\ 10\ 100\ 1000]^T$. The responses can be seen in Figure 2.8. The data for the responses can be seen in Table 2.3.



(a) Estimation error of the first state

(b) Estimation error of the second state

(c) Estimation error of the third state

(d) Estimation error of the fourth state

Figure 2.6: Observer estimation error of four states

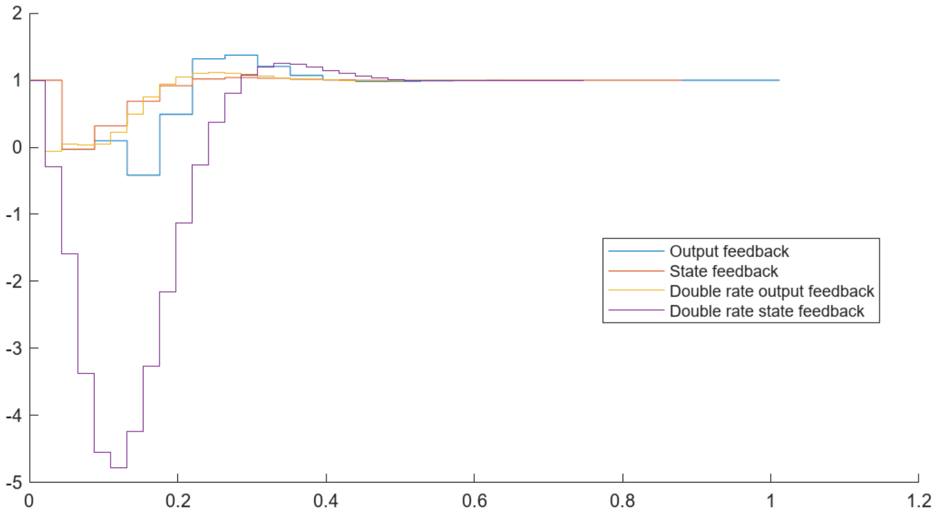
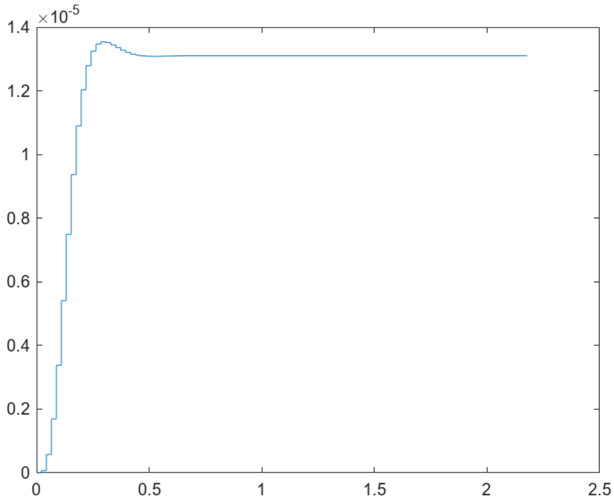
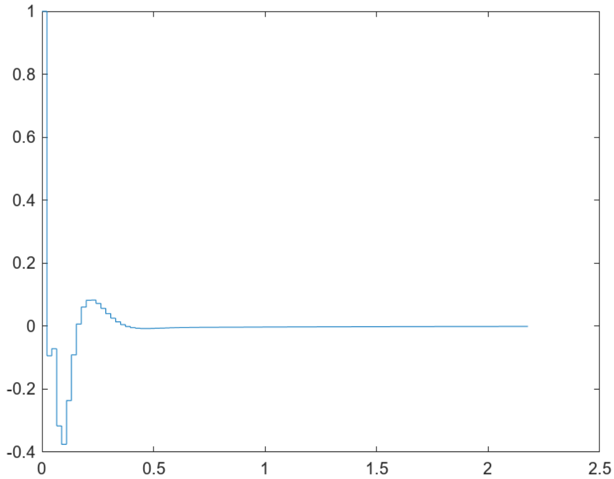


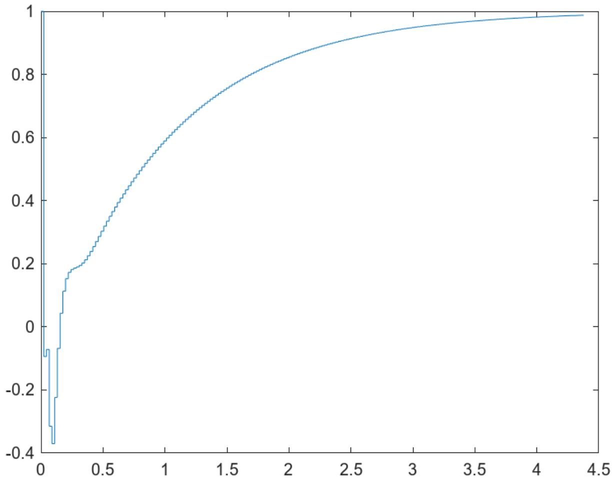
Figure 2.7: Step responses of the output and full state feedback controllers with nonzero initial conditions and single and double sampling rates



(a) Disturbance step response of the output-feedback control, double sampling rate



(b) Disturbance step response of the output-feedback control with integrator action, doubling sampling rate and nonzero initial condition



(c) Input step response of the output-feedback control with integrator action, doubling sampling rate and nonzero initial condition

Figure 2.8: Disturbance step response with and without integrator action

$$\left(\begin{array}{cccc|c} (\Phi-\Gamma K) & -\Gamma L & -\Gamma L_{int} & \Gamma & \Gamma K_{ff} \\ KC & (\Phi-KC-BL) & -\Gamma L_{int} & 0 & \Gamma K_{ff} \\ C & 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & 1 & 0 \\ \hline \multicolumn{4}{c|}{C \quad 0 \quad 0 \quad 0} & DK_{ff} \end{array}\right) \tag{2.11}$$

2.6. Linear-quadratic control

Linear-quadratic control is a form of optimal control, which minimizes a combination of states and inputs through a full-state feedback, just as in earlier sections. The block diagram is then the same, Figure 2.3. The cost function to be minimized is then in Equation 2.12, where $u(k) = -Kx(k)$. The solution K is given in Equation 2.13, where S is given by solving the Riccatti-equation, Equation 2.14. For this to be solvable for S the matrices Q_1 and $\begin{bmatrix} Q_1 & Q_{12} \\ Q_{12}^T & Q_2 \end{bmatrix}$ have to be positive semi-definite, and Q_2 has to be positive definite. The stead-state gain of such a controller is again not necessarily unity, so the a feed-forward block is used, the same as in all of the other controllers.

Continuous time poles	Discrete time poles	Rise time	Settling time	Undershoot
-52 ± 2, $-14 \pm 14.2i$ -265 ± 5, $-70 \pm 71i$	$0.7028 \pm 0.2006i$, $0.0019 \pm 0.2144i$ 0.0026, 0.0033 0.2785, 0.3606, 0.9774	0	3.4501	37.4762

Table 2.3: Poles and response data for output-feedback controller with integrator action from a nonzero initial state and unit step disturbance

$$\begin{gathered} \min_{u_{1:\inf}} \sum_{k=1}^{\inf} J_k \\ J_k = x_k^T Q_1 x_k + u_k^T Q_2 u_k + 2x^T Q_{12} u_k \end{gathered} \tag{2.12}$$

$$K = (\Gamma^T S \Gamma + Q_1)^{-1} (\Gamma^T S \Phi + Q_{12}^T) \tag{2.13}$$

$$\Phi^T S \Phi - S - (\Phi^T S B + Q_{12})(\Gamma^T S \Phi + Q_2)^{-1} (\Gamma^T S A + Q_{12}) + Q_1 = 0 \tag{2.14}$$

To tune the controller the selection of the Q matrices has to be performed. All of the data and weights tried can be found in Table 2.4. The rows of the table are in order of mentions in the text. A good initial starting point is to use Byron's rule, where the diagonal elements are the squared inverse of the maximum allowed deviation in the state or control input. Following this the first set of weights was devised, to minimize settling time. The maximum allowable deviation was 5% of the steady state value for each state, and the maximum allowable control input was 1000, taken from later parts of the exercise. This set produced a curve with no overshoot and decent settling time, 1.0016 seconds. All response plots can be found in Figure 2.9. The settling time is quite lacking compared to the both the full-state and the output feedback controller,but does settle very nicely, without any overshoot.

Next the effect of having weights only on the output state was investigated. Reducing the magnitude of Q_1 changes the relative weight of Q_1 compared to Q_2, so the maximum allowable deviation was multiplied by 0.7. This even increases the relative weight of Q_1 compared to the first set of weights. This reduced the settling time, but not by much. Still the performance is worse than the other feedback controllers, but the settling is almost perfectly damped.

For the third set of weights only the fourth state was given one, and the weight on the controller is relaxed, to again keep the relative weight Q_2 was reduced. This produced an overdamped curve with long settling time, suggesting that weighing the states other than the output affects the settling time negatively.

For the fourth set of weights Q_1 was kept the same as the first set, but Q_2 was reduced 10000 fold. This gave a fast settling time with a response slightly more underdamped than the first set of weights.

As it can be seen just using Byron's rule the resulting systems are all close-to critically damped, and it is hard to achieve the damping required for a 5% overshoot, which leads to a lower settling time. The response time is also roughly in the 1-2 seconds range, provided the weights are kept sane.

Q_1	Q_2	Q_{12}^T	Rise time	Settling time	Overshoot(%)
$\begin{bmatrix} 400 & 0 & 0 & 0 \\ 0 & 0.0384 & 0 & 0 \\ 0 & 0 & 5.917e-5 & 0 \\ 0 & 0 & 0 & 1.479e-5 \end{bmatrix}$	$\begin{bmatrix} 1e-6 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$	0.7277	1.3328	0.0439
$\begin{bmatrix} 816.33 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 1e-6 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$	0.7138	1.9849	2.2182
$\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 3.0190e-5 \end{bmatrix}$	$\begin{bmatrix} 1e-10 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$	1.0965	2.3177	0
$\begin{bmatrix} 400 & 0 & 0 & 0 \\ 0 & 0.0384 & 0 & 0 \\ 0 & 0 & 5.917e-5 & 0 \\ 0 & 0 & 0 & 1.479e-5 \end{bmatrix}$	$\begin{bmatrix} 1e-10 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$	0.5375	1.1273	0

Table 2.4: Weights and performance data for different LQR tunings

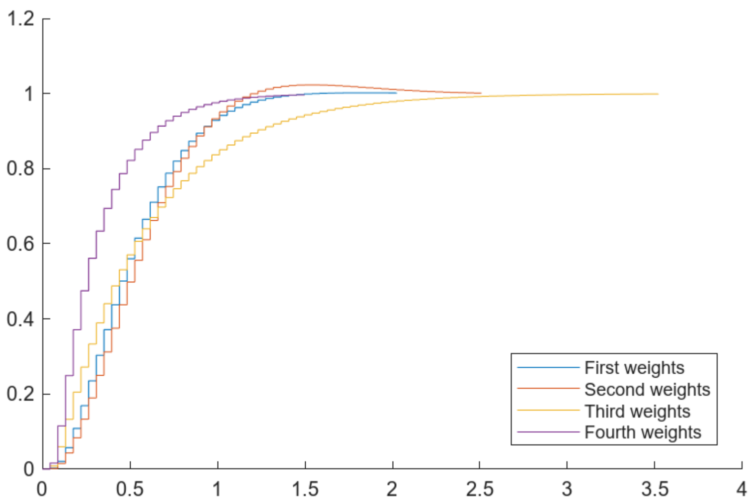


Figure 2.9: Step responses of the LQR controller with four different sets of weights

3. Actuator saturation

In a real system the magnitude of the control signals are limited, this aspect designed for in this chapter. The existing controllers were analyzed and retuned.

3.1. Existing controllers

3.1.1. Input sizes for existing controllers

Four existing control designs were analyzed for their control inputs, the PD, the full-state feedback, the output feedback and the LQR. Since the minimum settling time was the control objective, the control inputs are very large with all of them. Control inputs for state-space systems with feedback controllers can be calculated by augmenting the state space models with one extra output. The size of this extra output is in Equation 3.1 and in Equation 3.2 for full-state feedback systems and for output feedback systems respectively.

$$u_out = -Kx + H_{ff}u \tag{3.1}$$

$$u_out = \begin{bmatrix} 0 & 0 & 0 & 0 & -L \end{bmatrix} x + H_{ff}u \tag{3.2}$$

This extra output was added to the existing state-space systems and the results were plotted for a reference step. The plot of this is in Figure 3.1. It can be seen that all four controllers violate the ± 1000 limit for the control input, some even go under -1000.

3.1.2. PD controller retuning

For reference following in continuous time a PD controller was used. This was tuned with the root locus method for minimal settling time. This had to then be done in discrete time, with a ± 1000 constraint on control input magnitude. This was done again with the root-locus method, but in discrete time. The tool used was MATLAB `controlSystemDesigner`. The discrete time PD controller takes the form $u = K\frac{z-z_1}{z-p_1}$, leading to the three design variables being K, z_1, p_1. It was found that peak of the control input for a unit step reference change was related mostly to K and was much less sensitive to changes in the pole and zero locations. Then, on the root-locus plot, such as one in Figure 3.2 the following procedure was done: 1. choose a zero location 2. Choose a pole location that is larger 3. Adjust the gain so the system is

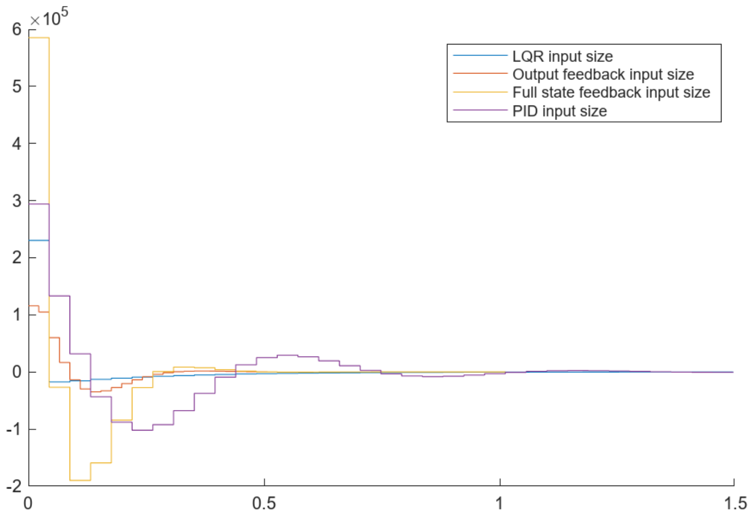


Figure 3.1: Plant input sizes for a reference step for four controllers

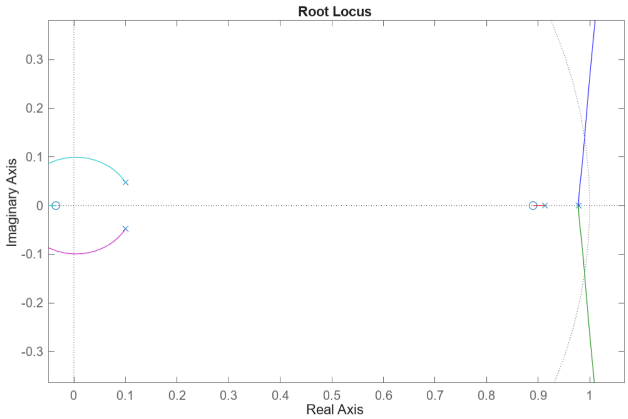


Figure 3.2: Root-locus with the PD controller

Controller	Discrete poles of system	Rise time	Settling time
$536.69\frac{z-0.89}{z-0.95321}$	$0.9780 \pm 0.0001i$, 0.9131 $0.1002 \pm 0.0473i$	6.6880	13.3320

Table 3.1: Data for the retuned PD controller

perfectly damped 4. Choose a larger/smaller pole depending on if the peak control input is above/below 1000 respectively and repeat from 2. 5. Increase the zero until the found pole locations reach a local minimum. This produced a controller with acceptable response. The final result is in Table 3.1.

3.1.3. Full-state feedback controller retuning

Tuning the full-state feedback controller here is the same procedure as without the constraints, specifying the continuous time poles first. It was found during the PD controller tuning that the closer the system is to perfectly damped the lower the maximum control input is, which means that the closer the poles are to each other the faster the smaller the peak control input needed to achieve the same step tracking performance. For this reason the four continuous time poles were put as close to each other as the precision of MATLAB `place` allows and the middle point was moved from 0 toward minus infinity, until the peak of the control input reached 1000. This controller produced an even faster response than the PD. The final result can be seen in Table 3.2.

3.1.4. LQR retuning

The baseline LQR controller was quite good but not all of its performance was utilized. For this task a new way of defining the cost was used; instead of using just the diagonal values terms consisting of $(ax_1 + bx_2)^2$ were added to it, with x_1 and x_2 being either one of the states or the input to the system. After some trial and error it was found that the best performance is if one additional term is used, using the following states and and inputs: $(100x_3 + x_4 + 280u)^2$. The performance of this weighing term can be observed in Table 3.3. For comparison a set of weights are also included where only the diagonal elements are included. These are the same weights as for the earlier LQR but the weight on the input was increased until the maximum of the control inputs was below 1000. It can be seen that including the off-diagonal elements reduce the settling time by more than 10% compared to the original.

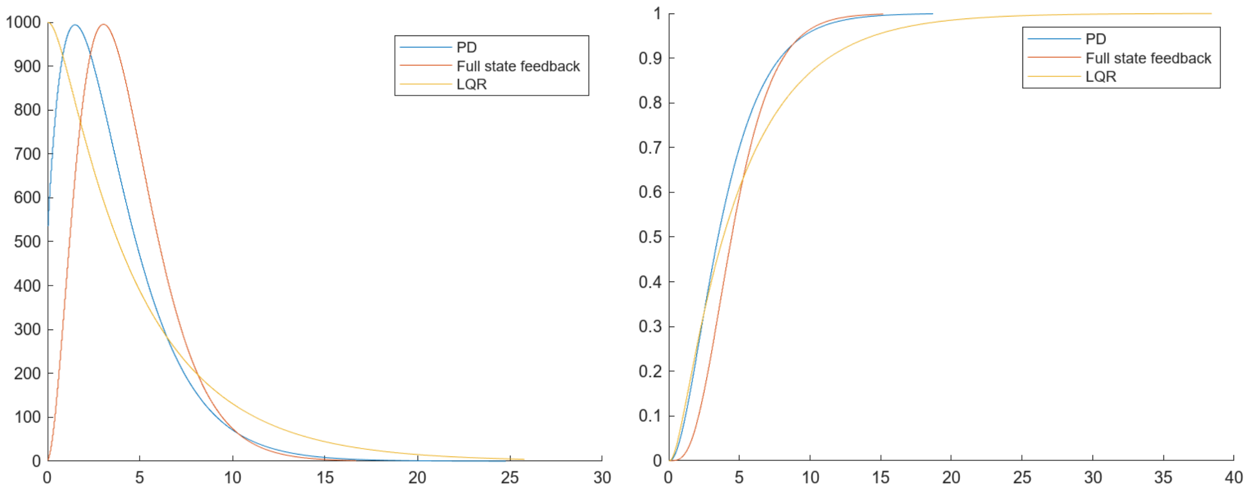
The step response and input sizes of all three retuned controllers are in Figure 3.3.

Continuous time poles	Discrete time poles	Rise time	Settling time	K
$-0.8486 \pm 0.003i$ $-0.8286 \pm 0.01i$	$0.9654 \pm 0.0006i$ $0.9631 \pm 0.0007i$	5.9840	12.1440	[9.0590e+07 -1.4331e+06 2.0276e+04 -228.6743]

Table 3.2: Poles and response data for the revised full-state feedback controller

Q_1	Q_2	Q_{12}^T	Rise time	Settling time	K
$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 9.612e-5 & 0 & 0 \\ 0 & 0 & 1.4792e-7 & 0 \\ 0 & 0 & 0 & 3.698e-8 \end{bmatrix}$	$\begin{bmatrix} 4.131e-06 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$	11.4400	24.3759	[0.0292 -0.0242 0.0123 0.1860]
$\begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 10001 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 78500 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ 28000 \\ 280 \end{bmatrix}$	10.2960	21.9119	[4.546e-5 1.051e-5 0.3567 1.539e-4]

Table 3.3: Weights and performance data for the retuned LQR



(a) Input sizes

(b) Output sizes

Figure 3.3: Input sizes and step response for a reference step with the retuned controllers

Continuous time poles	Discrete time poles	Settling time	K	L
$-0.8486 \pm 0.003i$ $-0.8286 \pm 0.01i$	$0.9654 \pm 0.0006i$ $0.9631 \pm 0.0007i$	0.583	[2.3041 267.29 9.8403e+03 1.1101e+05 3.5359e+05]	[3.8924e+06 -6.9167e+04 1.1911e+03 -10.8876 1]

Table 3.4: Poles and response data for with disturbance observer output feedback controller

3.2. Eliminating steady-state error

In the case of a disturbance input a controller might not be able eliminate all steady-state errors, for this reason additional measures have to be taken, which can be different depending on the controller used. There are two such methods discussed here. Both are just fancy ways of adding a pole at $z = 1$ to the controller.

3.2.1. Disturbance elimination with an observer

Disturbance elimination with an observer is done by assuming the a discrete LTI model for the disturbance input and using it the same way as the regular observer to estimate the disturbance state, and then supply an input with the opposite sign. This works well, when the disturbance can be modelled as an LTI system. In the case of this exercise, the disturbance is constant, which is easily modellable by Equation 3.3. For simulation purposes this is then added to the full state-space model, as in Equation 3.4. This method is straightforward to implement and simulate on linear feedback controllers, be it an output-feedback controller or a full-state feedback controller. The weights also do not play a role here.

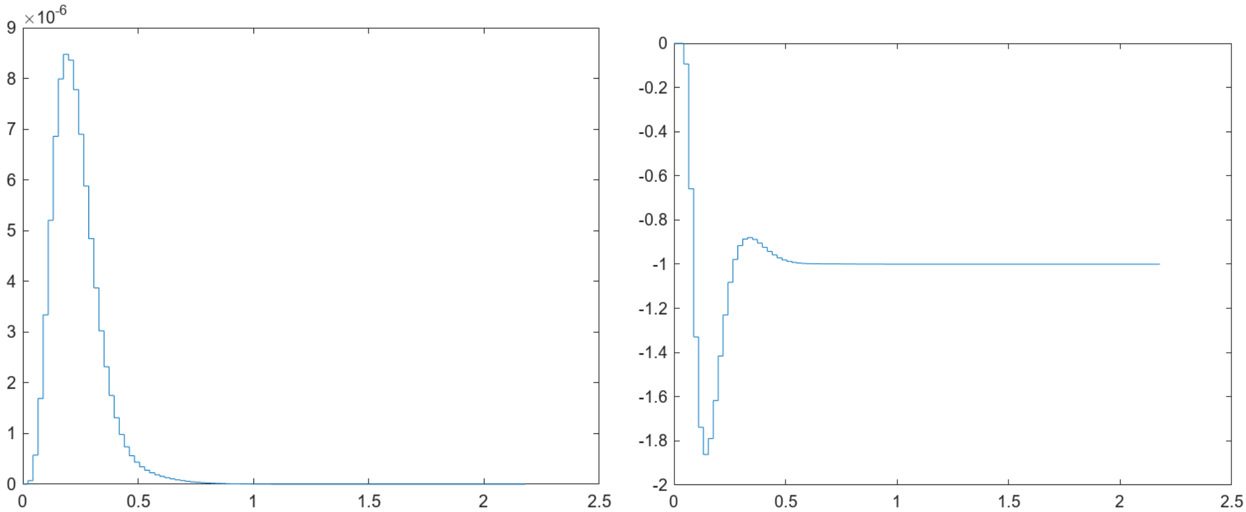
This kind of disturbance elimination was applied to the output-feedback controller, with both the state and the estimation augmented with the 5th disturbance state. For this to work a 5th, disturbance convergence pole had to be added to the set of target poles for the pole placement. This was chosen to be -4. From here the same pole placement method was used as previously. The state was initialized with the disturbance at 1, to simulate a step input. The response and the estimation of the disturbance is in Figure 3.4. It can be seen that the steady state error is eliminated and the estimated observer state converges to 1, so the observer works.

$$\begin{aligned} w(k+1) &= 1 \cdot w(k) + 0 \\ y_w &= w(K) \end{aligned} \tag{3.3}$$

$$\text{sys} = \left(\begin{array}{c|c} \Phi_{\text{sys}} & \Gamma_{\text{sys}} \\ \hline C_{\text{sys}} & D_{\text{sys}} \end{array}\right) = \left(\begin{array}{cc|c} \Phi & \Gamma & \Gamma \\ 0 & 1 & 0 \\ \hline C & 0 & 0 \end{array}\right) \tag{3.4}$$

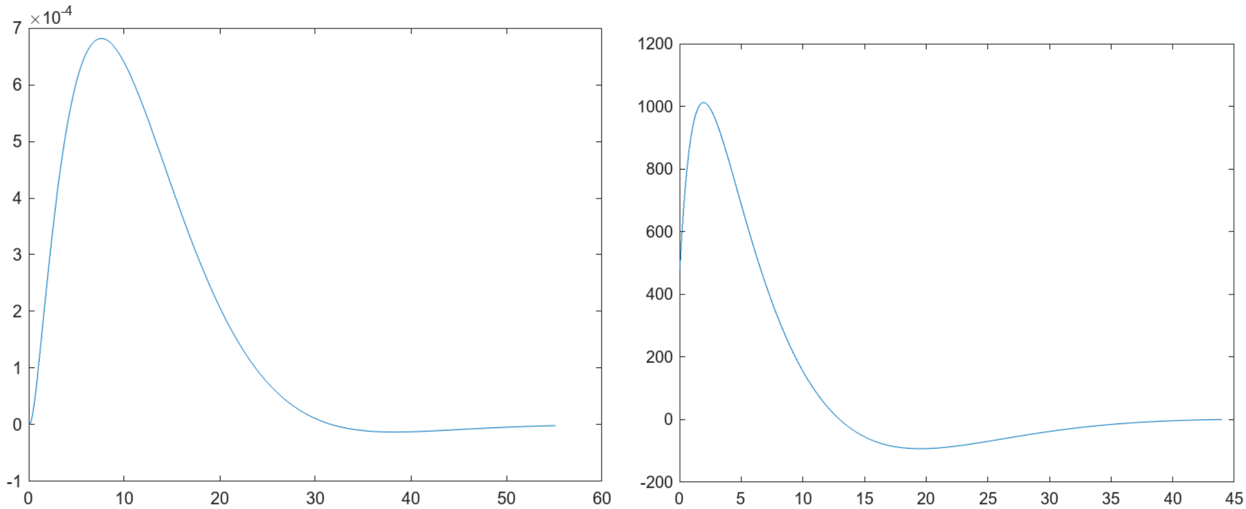
3.2.2. Disturbance elimination with integrator on the output

Integrating the output takes the same formulation as the previous model, but L_P and L_I are the scalar values instead of vectors. In the transfer function form, adding such term equates to multiplying the controller transfer function by $1 + \frac{L_I}{(z-1)}$. This second approach is used for the PD controller. Adding the term should increase to overall controller output, so the overall gain will have to be reduced. The tuning was done the same way as for the input limited controller, with MATLAB `controlSystemDesigner`. The existing pole-zero locations were not changed. It was found that the system is only stable for a very small integral gain. After some adjusting it 0.004 was selected as an approximate value. Then the overall gain was adjusted such that the actuator is not saturated, which produced the following controller transfer function: $u = 477.36\frac{(z-0.89)(y-0.996)}{(z-0.95321)(z-1)}$. The step response, disturbance rejection and control input size are in Figure 3.5.

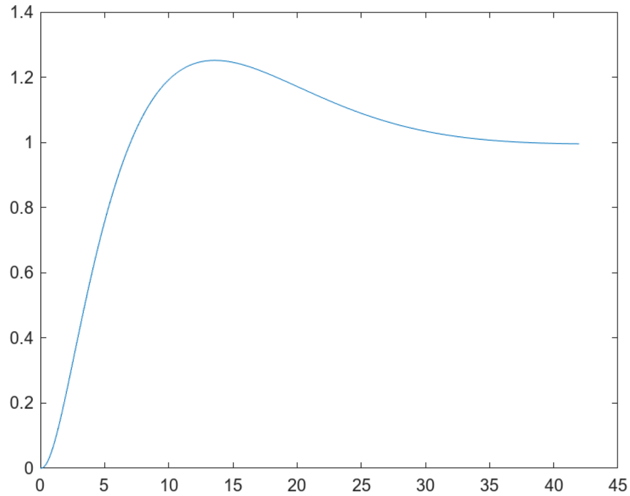


(a) Disturbance response **(b)** Disturbance estimate

Figure 3.4: Step disturbance response and disturbance estimate with Luenberger observer

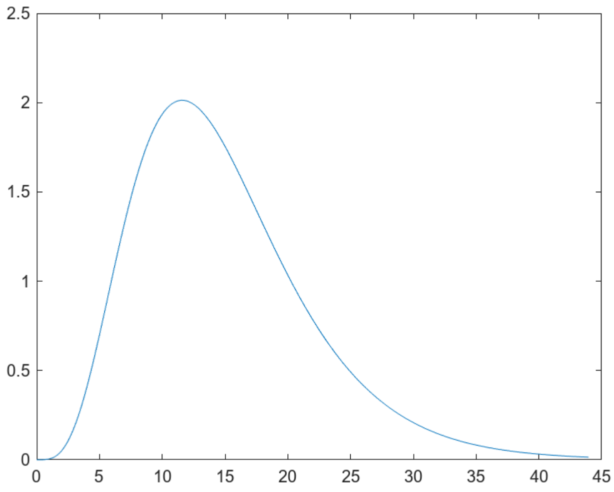


(a) Disturbance step response **(b)** Reference step control input

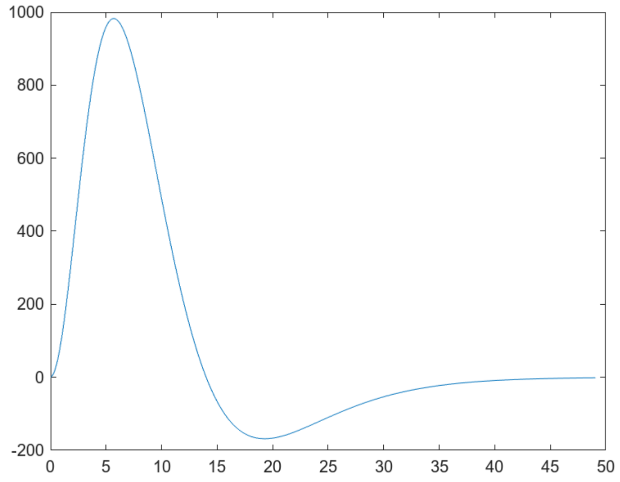


(c) Reference step response

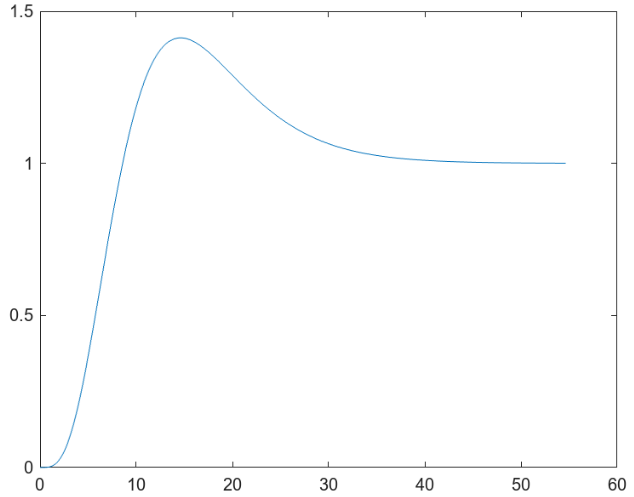
Figure 3.5: Disturbance step response and reference step response and control inputs for the disturbance-rejection PID



(a) Disturbance step response



(b) Reference step control input



(c) Reference step response

Figure 3.6: Disturbance step response and reference step response and control inputs for the full-state-feedback with integral

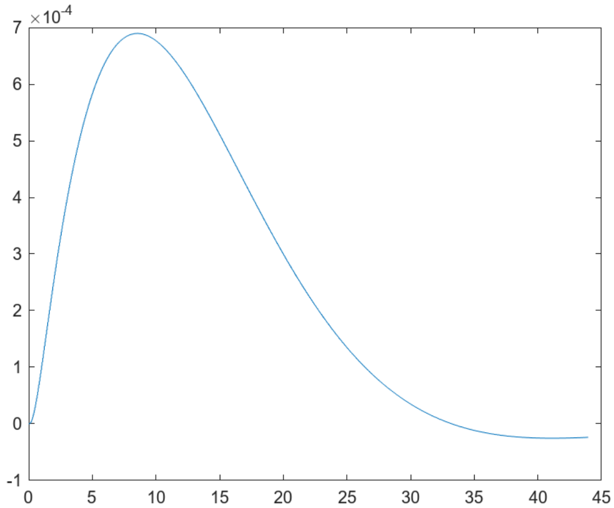
3.2.3. Disturbance elimination with integrator on the output, alternative formulation

The above method works well with PID controllers formulated with transfer functions. For full-state feedback controller a similar method has to be defined, that can be nicely expressed in state-space form. The original full-state feedback Equation 2.7, is augmented with an additional integrator state, which accumulates the error between the output and the reference. This is then further augmented by the disturbance model in Equation 3.4, to give the closed-loop form in Equation 3.5, where L_{int} is the integration weight.

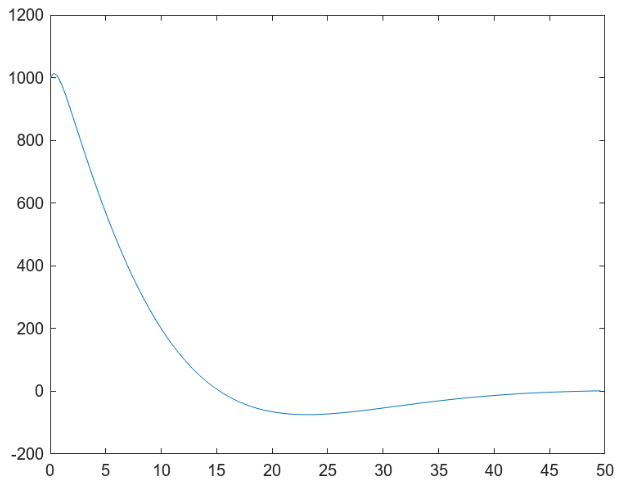
$$\left(\begin{array}{ccc|c} (\Phi-\Gamma K) & -\Gamma L_{int} & \Gamma & \Gamma K_{ff} \\ C & 1 & 0 & -1 \\ 0 & 0 & 1 & 0 \\ \hline C & 0 & 0 & DK_{ff} \end{array}\right) \tag{3.5}$$

Tuning the weights for this is done by assuming the system without the disturbance model, only five states. Then one additional pole is added and the the regular pole-placement method is used. The weight on the integral term will be the last term of the gains, so the result of the pole placement will be $[K\ L_{int}]$. For the purpose of this exercise the continuous time pole set used was $[-0.4\ \ -0.37\pm0.03i\ \ -0.43\pm0.03i]$. The step response, disturbance rejection and control input size are in Figure 3.6. It can be seen that the steady state error is eliminated and the disturbance is rejected, while still keeping within the actuator constraints.

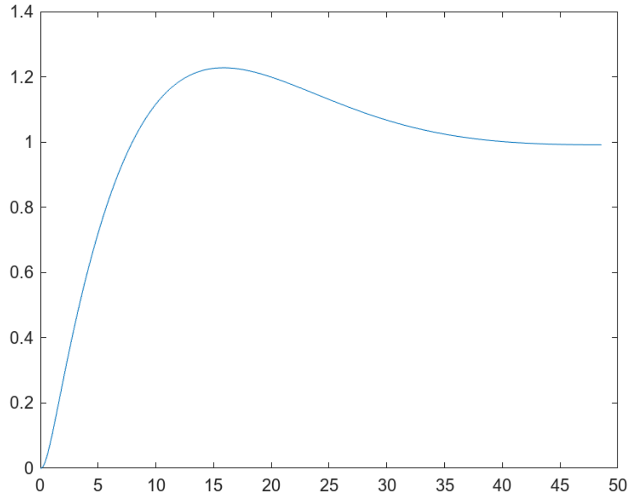
The same method also works for the LQR controller, since is a full-state feedback controller, but the gain has



(a) Disturbance step response



(b) Reference step control input



(c) Reference step response

Figure 3.7: Disturbance step response and reference step response and control inputs for the LQR with integral action

to be set manually. After the PID controller it was set at 0.004, which gave the step response, disturbance rejection and control input size that are in Figure 3.7.

4. Conclusions

The control of high-speed steel rolling mill is a tricky business. Several control methods were investigated as the inputs and outputs are of not the scale one would expect at first. For a small change a large control input is required, which either makes the system slow to respond or requires large control inputs. Several methods

References

[1] Graham C. Goodwin, Stefan F. Graebe and Mario E. Salgado. *Control System design*. 1st Sept. 2000. URL: `http://ci.nii.ac.jp/ncid/BA49548959`.

[2] Björn Wittenmark, Karl-Erik Årzén and Karl Johan Åström. *Computer Control: An Overview*. English. IFAC PROFESSIONAL BRIEF. International Federation of Automatic Control, 2002.