Compiled: 2024-01-10

Continuous- & Discrete-Time Control of Fast HDD Laser Alignment

Delft Centre for System and Control
Laurens Hoogenboom - 4609638

Contents

Introduction

The control of hard drive disks (HDDs) is a crucial implementation for the functioning of the modern world. While it is true that consumer electronics mostly use solid-state storage these days, a majority of servers running the internet or corporate infrastructure still use optical storage for its cost effectiveness and data recoverability after hardware faillure. The optimal control of HDDs is therefore an important and interesting topic from both an technological and economic viewpoint.

$$\begin{aligned} P(s) &:= \frac{2\zeta\omega s + \omega^2}{s^2(s^2 + 2\zeta\omega s + \omega^2)} \\ \zeta := .1, &\quad \omega := 1000\frac{\text{rad}}{s} \\ P(s) &= \frac{200s + 10^6}{s^4 + 200s^3 + 10^6 s^2} \end{aligned} \tag{1}$$

In Equation 1 we see a transferfunction that will serve as a generalized HDD (the plant) throughout this paper. In this assignment the aim is to find feedback control methods that optimize the controller's performance in both reference tracking and disturbance rejection. In order to do so a frequency domain analysis of the plant will be made. Following which, seperate controllers will be formulated to do reference tracking and input-disturbance rejection.

First, PID-type controllers will be tuned in continuous-time. These controllers will afterwards be discretised to obtain the discrete-time controllers. The transferfunction will also be mapped to the time domain as a state-space system. This state-space system will also be discretised. The discrete state-space system can then be used to design a pole-placement and LQR controller.

When all these controllers are formulated an additional copntraint on the designs will be implemented, where the input into the plant gets limited. Methods to remove steady-state errors will also be discussed, but the methods will already be implemented whenever relevant. Lastly, single-step deleys will be implemented in the discrete control systems to see their effects. This assignment will be closed by some of my final thoughts.

Continuous-Time Control

As per the project assignment the model-free control section of this assignment will be performed using the controller configuration seen in Figure 1. Here, r refers to the reference signal, while y is the plant output. Hence, e is the error of the output and reference. u is the controller output/plant input. q is a additive constant distubance on u.

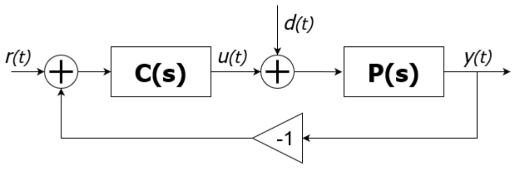


Figure 1: Block diagram of the continuous time controller configuration

By equating the denominator of the plant in Equation 1 to zero: $s^2\left(s^2 + 200s + 10^6\right) = 0$, the open-loop poles can be found, as seen in Table 1. Moreover, in order to find the closed-loop poles the denominator of the sensitivity function in Equation 2 can be used for the uncontrolled case $C(s) = 1$. i.e. One needs to solve $1 + P(s) = 0$

Looking at the poles in Table 1, we see that the plant has two poles in the origin these are there becausethere is a double integrator $\frac{1}{s^2}$ in $P(s)$. Because of this the plant is guaranteed to be unstable in an open-loop configuration. However, the closed loop poles have a complex conjugate pole pair that is purely imaginary. Because of this the closed loop system is marginally stable. Both the unstable open-loop and marginally stable closed-loop step responses are illustrated in Figure 2.

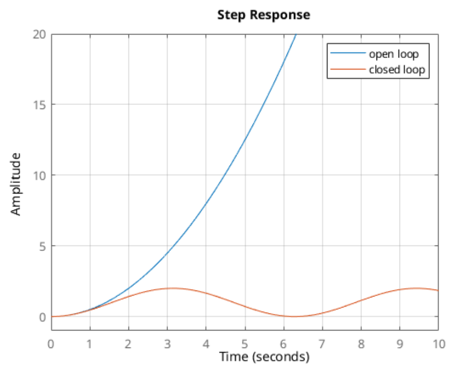


Figure 2: Open- and closed-loop unit-step responses of the plant

$$T(s) := \underbrace{\frac{P(s)C(s)}{1+P(s)C(s)}}_{\text{reference part}} R(s) + \underbrace{\frac{P(s)}{1+P(s)C(s)}}_{\substack{\text{load disturbance}\\ \text{part}}} Q(s) \tag{2}$$

Needless to say the plant will not be operable without a decent controller. We will now start looking at the design of a reference tracking controller, folowed by the design of a seperate disturbance rejection controller using PID control.

Table 1: Table 1: Poles of Uncontrolled plant

open-loop	0±	−100± 995i
closed loop	0± 1i	−100± 994.99i

Q1 | Reference Tracking Controller

For the reference tracking controller the assumption will be made that there is no disturbance ($q(t) = 0$). Because of this the sensitivity function of the system will reduce to the reference part of Equation 2 function, shown in Equation 3. Additionionally, the controlled system will have the following requirements:

Settling Time The controler needs to be optimised for the output converge to the reference signal as fast as possible with a tolerance of 1%.
Overshoot The ouput is not allowed to exceed an overshoot of 5% with respect to the reference signal.
Steady State The system has no steady state error on the output with respect to the reference signal.
Model Free Control The controller in the system is some combination of proportional, integral, and differential action on the error of the output with respect to the reference signal.

$$T_r(s) = \frac{P(s)C(s)}{1+P(s)C(s)} \tag{3}$$

The zeros of the open- and closed-loop transferfunction have not yet been discussed. However, this will not be necessary as looking at the root locus plot of our closed loop system in Figure 3 will provide enough insight. The most important take away of the plot is to see that whatever gain is added in a closed-loop configuration, the poles on the imaginary axis will never get a negative real part. Instead, they will become positive, rendering the system unstable.

From this we can conclude that the system will never be stabilised simply by adding gain. Hence, a P-controller will not be of any use.

In order to design the following contollers one could attempt to tune the PID controller by expressing the closed loop poles as functions of the tuning paramters, and placing the controlled poles that way. However, this proved to be very laborious for a 4th order plant. Therefore, loopshaping will instead be done, using bode plots. The benefit of loopshaping is that it allows for a more heuristic approach. This is quicker, but less precise. As a result it is not obvious where the controleld poles will be placed and how they relate to oneother untill the tuning paramters can be substituted in the sensitivity function. Finding the tuning parameters' values will therefore be more of an educated guess than willfull placement.

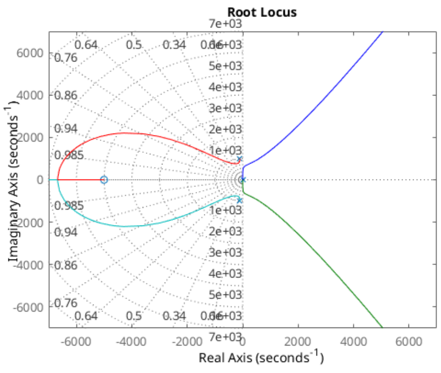


Figure 3: Root locus plot of plant in negative feedback loop configuration

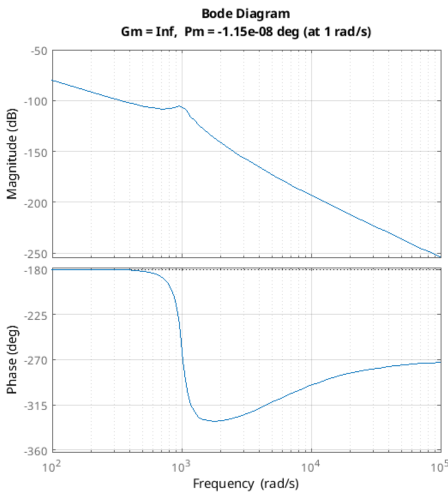


Figure 4: Bode diagram of the plant in an uncontrolled feedback configuration

When looking at bode plots it is common practice to look at the open-loop configurations, because the effects of an open-loop frequency response in the closed-loop configuration is a well understood topic. In Figure 4 it can be seen that the uncontrolled step response gain is low for all frequencies. What would be more preferable to see here is a high gain for low frequencies and a negative gain for high frequencies. Similarly, it is desirable to have phase above $-180°$ for low frequencies, and below for higher frequencies. The gain is in decibels, which means it is an absolute number, and negative decibells indicate small gain, rather than negative gain. The sign of gain is instead indicated by the phase. The phase gets shifted between the open- and closed loop configurations. Hence, the phase crossover line being at $-180°$.

Because the frequency scale is also logarithmic, a frequency of zero is not defined on the horizontal axis. This is however not a problem, as we can see what the zero-frequency tends to, and it is much more important to see the behaviour for generally low frequencies.

Now, before we can start finding a controller, that last thing we need to know is what kind of controller we need to assure there is no steady state error. If our system were stable in the closed-loop configuration we could have used the finite value theorem to see if this system converges to the reference signal value, which is 1. Moreover, if the derivative converges to zero in the limit of s to zero. This is not the case. Instead we will take a more heuristic approach of designing different stabilising controllers and perform the

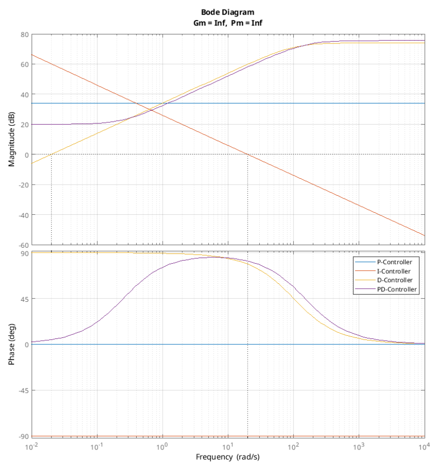


Figure 5: Arbitrarily tuned P,I,D,PD Controllers to illustrate their general frequency responses

finite value on the controlled feedback system to determine
if the controller is enough à posteriori.

D controller

Having shown that a P controller will not suffice, the starting point will insteaed be with a D-controller. Figure 5 shows that a generic D-controller adds $90°$ to the zero-frequency, while adding no phase in the limit to infinity. Moreover, a D controller adds relatively little gain to the lower frequencies, while adding more to the higherfrequencies. The lowest gain can be in the negative dBs, but is not nesciarily so depending on how the controller is tuned.

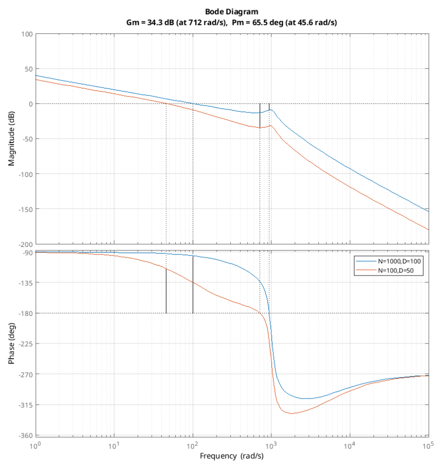


Figure 6: Bode diagrams of the D-controllers with very different step responses

Figure 6 shows two options for D controllers that were found. The blue and red gain curves diverge towards higher frequencies. Comparing to Figure 4 we see the gain dropoff was reduced, compared to the plant. In this case it is important to the tuning of the D-controller that the higher frequencies are sufficiently diminished. The blue bode gain line shows a notch that comes close to the 0dB line, with the left part of the notch still receiving positive phase. As a consequence we see in Figure 7 that the step response with this controller shows unwanted oscilatory behaviour. Therefore the D-controller will be adjusted to have its phase crossover frequency lower than that of the notch and also changed the controller to add less gain to the higher frequencies, such that the notch receives less gain. This slowed down the controlled system while assuring a nicer, smooth, step response. The relevant characteristics of the step response can be seen in Table 2.

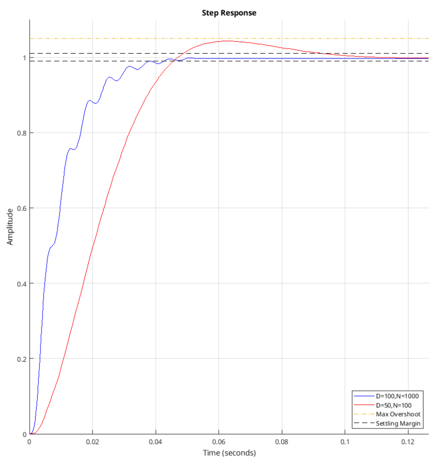


Figure 7: Step resonses of both D-Controllers seen in Figure 6.

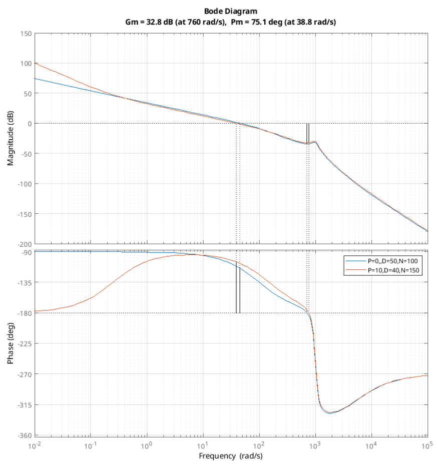


Figure 8: Bode diagrams of the D- and PD-controllers that were considered

PD-Controller

The D-controller from the previous section will be used as a starting point for the PD-controller. It is very unlikely that for an arbitrary plant the optimal D and PD controllers wil have identical D terms. Therefore, the D term was tuned whilst tuning the P term too. However, it cannot claimed that this will yield the best PD controller.

In Figure 5 it becomes apparent that the PD controller does not add phase to the zero frequency and also adds more gain to the lower frequencies. As a result a PD controller can have a very similar frequency response around 10^3rad/s, where the notch is, while mainting the faster response of the D-controller with unwanted oscilations. This is reflected in Figure 8. As a result this yields a faster step response that remains smooth, while also reducing the overshoot. This can be seen in Figure 9. In Table 2 we can see that, although the rise time is slightly longer, the PD controller settles faster. However, but the question remains whether the controllers yield systems with a steady state error.

Table 2: Table 2: Continuous-time step response characteristics of D-controller vs. PD-controller

	C(s)	Overshoot	Rise Time	Settling Time
D-Controller	$\frac{50s}{\frac{s}{100}+1}$	4.2738%	0.030294 s	0.084023 s
PD-Contrtoller	$10+\frac{50s}{\frac{s}{100}+1}$	0.79977%	0.031279 s	0.049196 s

Steady State errors

Now that the tuning paramters have been selected and the controlled systems are stabilised, the finite value theorem can be used to see if the system converges to the reference value for the limit of time to infinity.

Let $f(t)$ be a step response of a stable LTI system. Then,

$$\lim(t,\infty)f(t) = \lim(s,0)s\mathcal{L}\{f(t)\}(s) \tag{4}$$

can be used to determine if the steady state value of the response exists, and what it is.

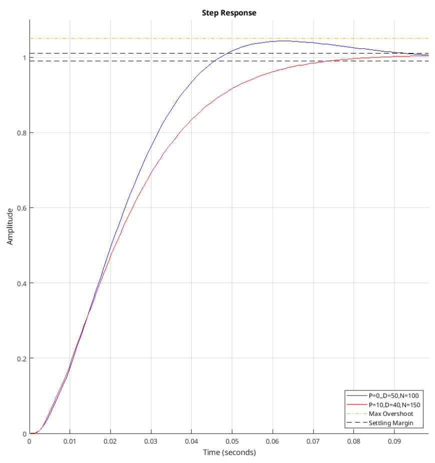


Figure 9:

Using the finite value theorem as stated in Equation 4 we can formulate Equation 5, which in both the cases of the D- and the PD-controller yields $L = 0$. This is good because this means that the error $e(t)$ with respect to $r(t)$ is driven to zero. i.e. no steady-state error. See Figure 1 for an overview of the signals.

$$L = \lim_{s \to 0} s \frac{E(s)}{R(s)} = \lim_{s \to 0} s \frac{1}{1 + P(s)C(s)} \tag{5}$$

It can therefore be concluded that either structure is sufficient for convergence, but the PD controller is faster. Therefore the PD-controller from Equation 6 will be be selected.

$$C(s) = 10 + \frac{50s}{\frac{s}{100} + 1} \tag{6}$$

Q2 | Disturbance Rejection

For finding the controller that rejects input load disturbance $q(t)$, it will be assumed that there is no applied reference signal ($r(t) = 0$). The load disturbance sensitivity function therefore reduces to the load disturbance part of Equation 2, which is displayed in Equation 7. In the uncontrolled case this sensitivity function reduces the the same as that of Equation 3. Therefore, using the pole and root locus analysis it is clear that a P-controller still will not suffice.

$$T_l(s) = \frac{P(s)}{1 + P(s)C(s)} \tag{7}$$

Because the loopshaping is performed using the open-loop bode plots we can also still use the bode plot from Figure 4 to analise the system. Tuning with the PD-controller from the reference tracking as a starting point with this sensitivity function yields a bode plot and step response as shown in Figure 10 and Figure 11 respectively.

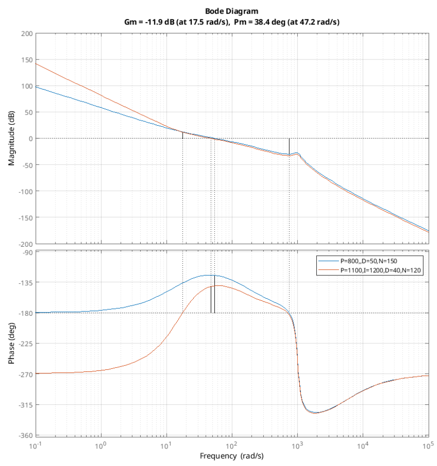


Figure 10: Atlered PD-controller from reference tracking and PID-controller

Despite having different tuning parameter values than the reference tracking controller, the PD-controller in Figure 10 displays a very similar frequency response. These tuning paramters, along with the response characteristics, can be seen in Table 3. The PID-controller in red shows an even more desirable frequency response, with similar gains and phase for the high frequency range, while adding even more gain to the lower frequencies. This controller also shows negative phase for these low frequencies, which is desirable, as the effect of the step input will in this case need to be diminished. Looking at the $-180°$ phase that the PD-controller yields at low frequencies, it is not against expectations that this controller leaves a steady-state error. It is important to note that this is different from the desired frequency response for reference tracking. This is confirmed in Figure 11. Although the PD-controller is decently fast, this is not acceptable. Therefore an integral term was added to the controller and it was re-tuned, yielding the PID-controller in these figures, and Table 3.

Table 3: Table 3: Discrete-time step response characteristics of D-controller vs. PD-controller

	C(s)	Peak	Peak Time	Transient Time
PD-Controller	$1000 + \frac{40s}{\frac{s}{100}+1}$	0.0010141	0.1074 s	0.083335 s
PID-Contrtoller	$1100 + \frac{1200}{s} + \frac{40s}{\frac{s}{120}+1}$	0.00078885	0.075674 s	0.22295 s

Steady State Error

In order to check if the PID-controller really has no steady-state error the finite value theorem as in Equation 4 will again be used. The desired limit L still yields $L = 0$, because the aim is to reject effects of $q(t)$ on $y(t)$.

$$\lim_{s\to 0} sF(s) = \lim_{s\to 0} s\frac{Y(s)}{Q(s)} = \frac{P(s)}{1 + P(s)C(s)} = L \qquad (8)$$

Again, the limit tends to $L = 0$, so it can be concluded that the PID controller is sufficient. We will therefore conclude the continuous time control section with the disturbance rejection controller in Equation 9.

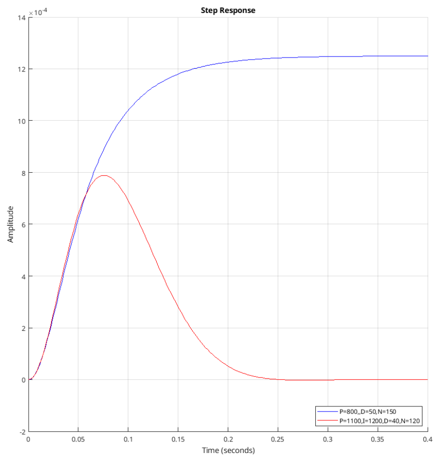


Figure 11: Step responses of the PD- and PID-controller in Figure 10

$$C(s) = 1100 + \frac{1200}{s} + \frac{40s}{\frac{s}{120} + 1} \qquad (9)$$

Discrete-Time Control

In this section the provided plant will be transformed in its state-space representation and will be subsequently discretised. Then, model-based and mode-free control methods will be used to stabilise the system.

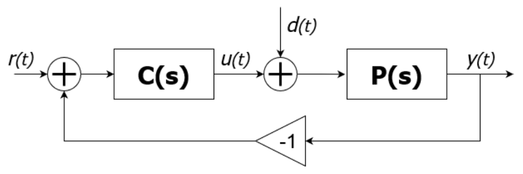


Figure 12: Block diagram of the continuous time controller configuration

Q3 | Discretisation

In order to perform discrtisation the state space model will be brought to its state-space form. It will then be discretised using matlab, while applying the rule of thum that having ten samples in the rise time of the continuous time response in generally enough.

Using the expanded notation in Equation 1 we can find a state space solution from the numerator $0s^3 + 0s^2 + 200s + 10^6 = \beta_1 s^3 + \beta_2 s^3 + \beta_3 s^3 + \beta_4 s^3$, and $s^4 + 200s^3 + 10^6 s^2 + 0s + 0 = s^4 + \alpha_1 s^3 + \alpha_2 s^2 + \alpha_3 s + \alpha_4$. Hence, the state space realisation yields Equation 10, which thus equates to Equation 11.

$$\dot{x} = \begin{bmatrix} -\alpha_1 & -\alpha_2 & -\alpha_3 & -\alpha_4 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} x + \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} u \tag{10}$$
$$y = [0 \;\; 0 \;\; \beta_1 \;\; \beta_2] x + [0] u$$

$$\dot{x} = \begin{bmatrix} -200 & -10^6 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} x + \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} u \tag{11}$$
$$y = [0 \;\; 0 \;\; 200 \;\; 10^6] x + [0] u$$

Now for the sampling rate we will choose the frequency such that there are 10 samples in the rise time of the step response. This was performed using the c2d() function in matlab, which uses a zero-order-hold discretisation.

Table 4:

$$\begin{bmatrix} 0.3337 & 524.1 & 0 & 0 \\ -0.0005241 & 0.2289 & 0 & 0 \\ 7.711 * 10^{-7} & -0.0003699 & 1 & 0 \\ 5.603 * 10^{-9} & 1.892 * 10^{-6} & 0.005233 & 1 \end{bmatrix} x + \begin{bmatrix} -0.0005241 \\ 7.711 * 10^{-7} \\ 5.603 * 10^{-9} \\ 1.18 * 10^{-11} \end{bmatrix} u \tag{12}$$
$$[0 \;\; 0 \;\; 200 \;\; 10^6] x + [0] u$$

Q4 | Discretising PID controllers

Table 5: Table 5: Characteristics of Both CT controllers after discretisation

	Reference Tracking	Disturbance Rejection
C(s)	$10 + \frac{50s}{\frac{s}{100}+1}$	$1100 + \frac{1200}{s} + \frac{40s}{\frac{s}{120}+1}$
C(z)	$\frac{7510z-7505}{z-0.4561}$	$\frac{5900z^2-1.154*10^4z+5640}{z^2-1.765z+0.7653}$
DT Overshoot/Peak	12.582%	0.00087114
DT Rise-Time/Peak-Time	0.0157 s	0.098098 s
DT Settling-Time/Transient-Time	0.0628 s	3.1726 s

The controllers were discretised using their own rise-time or transient-time for the reference tracking and disturbance rejection controllers respectively from the continuous-time domain. The rule of tumb that 10 samples per rise time is generally a suffiecient sampling rate, so both the before mentioned times were divided by 10, and used as sampling times. The state-space system used the rise-time of the reference tracking step response.

In table Table 5 we see that the discretisation actually has a lower rise time, but a larger settling time. This is due to the larger over- and undershoot compared toi the continuous-time controller. The overshoot also far exceeds the required maximum of 5%. Therefore, we can conclude that for the reference tracking the discretisation of the continuous-time controller will not be sufficient.

For the disturbance rejection controller we can see that the peak height and the peak-time are not that much larger than with the continuous-time controller. However, the transient time is significantly larger. For this controller too, I would argue that it is likely a bettor option to find a different controller in discrete-time.

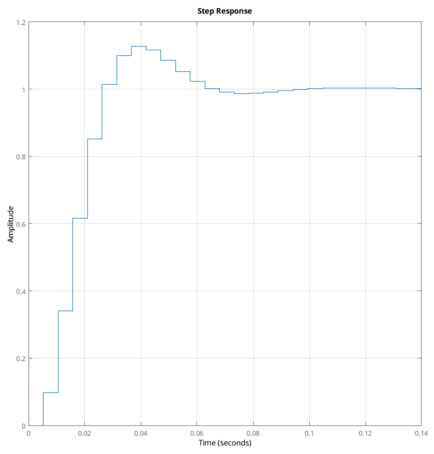


Figure 13: Discretisation of the continuous-time reference tracking controller

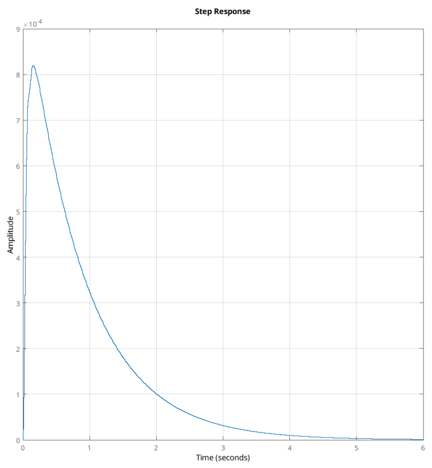


Figure 14: Discretisation of the continuous-time disturbance rejection controller

Q5 | Full-Information Pole Placement Control

This section will be dedicated to formulating a pole-placement controller for the reference tracking control of the discretised system. For intuition's sake, the poles will first be placed in the laplace domain, after which the pole locations will be mapped to the z-domain, such that the discrete poles can be placed appropriately. Although the same information can be deduced from the z-domain locations as from the laplace-domain locations, I personaly am more comforable in the laplace domain.

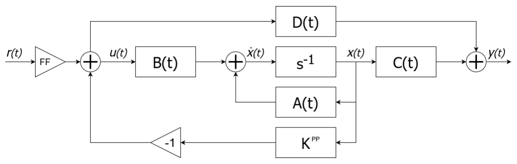
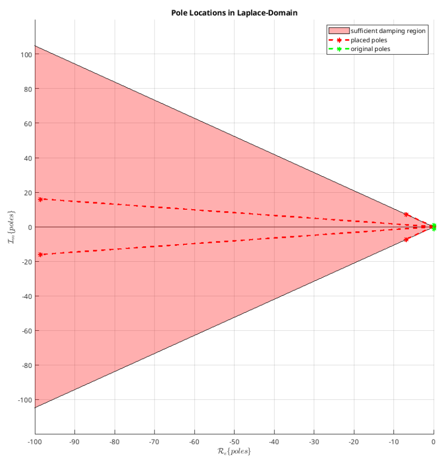


Figure 15: Block diagram of the continuous time controller configuration



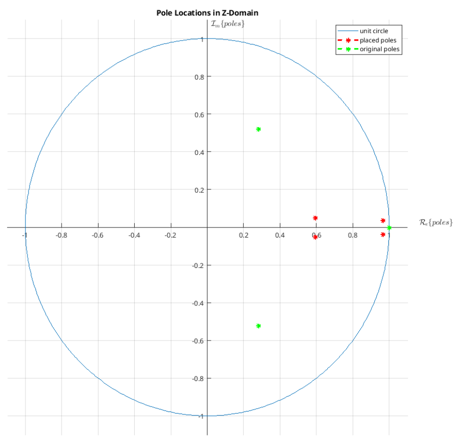


Figure 16: Original and placed poles of the SS-system in the s-plane

Figure 17: Original and placed poles of the SS-system mapped to the z-plane

In Figure 16 we see the pole placement locations indicated with red asterisks. THe exactlocations of the polos can also be seen in Equation 13. The original poles are shown in green. One of the original pole pairs is shown, as the other has such a large magnitude and damping ratio that it is out of bounds of the plot. Because the shown original poles are on the imaginary axis, they have a damping ratio of 1. That is to say there is no damping.

$$-69.011 \pm 72.371i \quad \& \quad -986.93 \pm 161.13i \tag{13}$$

The shaded area indicates the damping ratio range of a 2-dimensional system with a 5% oversoot, or less. In choosing the pole locations the dominant pole pair, which is the one closer to the origin, is placed on the edge of this range, such that the system will have an overshoot near 5%. The non-dominant pole pair is placed with a lower damping ratio, and has a magnitude 10 times larger than the dominant pole pair. This magnitude was chosen so large because this causes the pole pair to be very recessive, meaning the system approximates the behaviour of a 2-dimensional system with only the dominant pole pair. However, as we will later see this has a significant effect on the gain of the controller output.

After the placement of the poles in the laplace-domain, the determined locations for the poles can be mapped to their z-domain equivalent. The mapping to the z-domain is partially dependent on the sampling frequency. Therefore, the locations of these discrete-time poles will be different if any other sampling frequency is used. *This link* provides a convenient tool to play around with poles in both domains, and see the effects of all relevant discretisation factors.

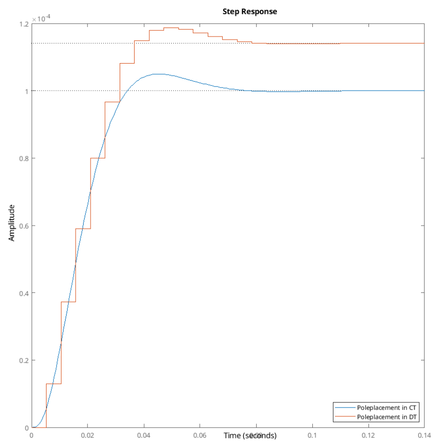


Figure 18: Step response of the placed poles in the s- and z-plane from Figure 16 and Figure 17 respectively

In Figure 18 we see the continuous- and discrete- time step responses of the poles in their placed locations. Both responses show a steadystate error, since the vertical axis is scaled with 10^{-4}, but the shape of the curve is very similar, apart from smooth and jagged curve. These plots show the necessity of a feed-forward gain, which is implemented in the system the way seen in Figure 15. Here, the FF-gain is determined by the ratio between 200 and the DC-gain. By multiplying the B matrix with the number it is added to the system.

Figure 19 shows both the continuous- and discrete-time implementations of the Feed-Forward gain. Ofcourse, both required a different gain. In both cases the steady-state error was brought to zero, and the overshoot remains at 5%. Moreover, we see that the implementation of feed-forward gains has had no effect on the rise time, nor the settling time.

This fastness of the response can be arbitrarily chosen at this point, because we do not (yet) implement limitations on the gain of the controller output. This means we can have our controller add infinitely much energy to the system, which is not realistic. In a later section we will implement a constraint on this, so that we actually have an upperbound on how fast the response is, but first we will implement this controller in an observer-based configuration.

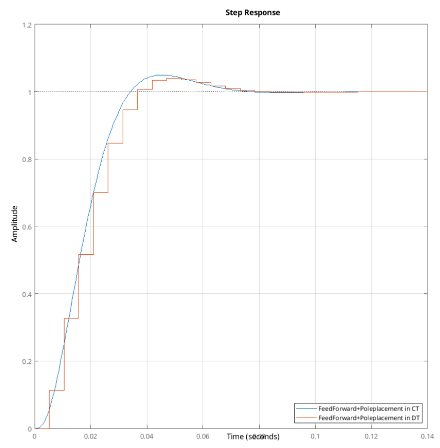


Figure 19: Step responses from Figure 18 with added feed-forward terms

Q6 | Observer-Based Pole Placement Control

In the previous section we assumed full-information, which implies that the states are exactly knowable at any time. In pole placement control this simplifies the control design, because this causes the controller to work in the most simple negative-feedback configuration. However, in this section we will work with partial-information feedback, which necesiates an observer to estimate for the hidden states.

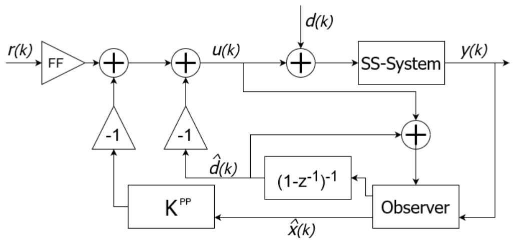


Figure 20: Block diagram of the continuous time controller configuration

Because state estimates are exactly that, estimates, they will inherently be wrong. In order to facilitate the pole placement controller the best way, the poles of the observer will be placed faster that those of the controller.

The idea behind this is that as long as the estimates converge to asymptotically to the real states and the real states converge asymptotically to the right values, the observer based controller can be stable, as long as the estimates converge faster than the states. I.e. the estimates can always keep up with whatever the control action is doing to the states and even reduce the estimation error over time while said control action is still at play.

In order to acchieve this the observer poles will be placed at the same damping ratios as the observer poles, but with 4 times the magnitude in the continuous-time phase plane; Making them faster than the system poles.

$$\begin{aligned} x_{k+1} &= \Phi x_k + \Gamma u_k \\ y_k &= C x_k + D u_k \end{aligned} \tag{14}$$

$$\begin{aligned} \begin{bmatrix} x_{k+1} \\ \hat{x}_{k+1} \\ \hat{d}_{k+1} \end{bmatrix} &= \begin{bmatrix} \Phi & -\Gamma * F & -\Gamma \\ LC & \Phi - LC - \Gamma F & 0 \\ L_i C & -L_i C & 1 \end{bmatrix} \begin{bmatrix} x_k \\ \hat{x}_k \\ \hat{d}_k \end{bmatrix} + \begin{bmatrix} \Gamma G & \Gamma \\ \Gamma G & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} r_k \\ \hat{d}_k \end{bmatrix} \\ y_k &= [C \;\; 0 \;\; 0] \begin{bmatrix} x_k \\ \hat{x}_k \\ \hat{d}_k \end{bmatrix} + [0 \;\; 0] \begin{bmatrix} r_k \\ d_k \end{bmatrix} \end{aligned} \tag{15}$$

In Equation 15 we see the closed-loop state space representation of the observer-based control configuaration. The Φ and Γ matrices are the same system matrices as in Equation 14, but as we can see the system has been augmented with the state and distrubance estimates as new states, and the reference and disturbance as input to the system. With this MISO system the effects of either, or both, of the reference or disturbance can be simulated.

We see in Figure 21 from the simulation of the closed loop system that it is performant with and without discturbance. This is nice, because in the continuous time case we used seperate controllers for this. What we do not know is if the previous controller performed as well for both tasks. In this particular case it was necessary to find an estimate of the states and the disturbance simultaneously anyway. It was a minor sidestep to output the step response on either at the same time.

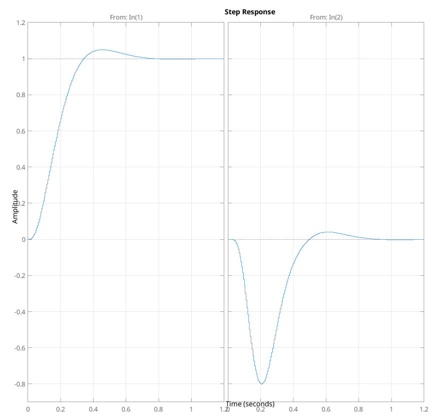


Figure 21: Step resonses to - Left: The reference unit step - Right: Disturbance unit step

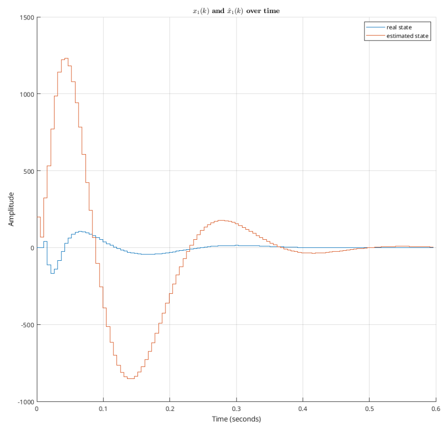


Figure 22: Step response of the 1st state and the estimate of the 1st state under both reference and disturbance steps simultaneously

In Figure 22 we see that the estimate of the observer starts off with a a different initial state than the real state. This causes the estimate to have a very different step response than the real state, but only during the transient period. We see that both values converge to the same value. This means that the observer does its work correctly, although perhaps the poles could have been placed such that the transient responses would diverge less.

Q7 | LQ-Control

$$J = \min_{u_{0,\dots,N}} \sum_{k=0}^{N} x_k^\top Q x_k + u_k^\top R u_k \tag{16}$$

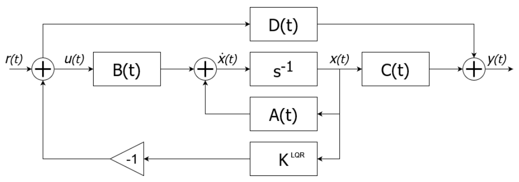


Figure 23: Block diagram of the continuous time controller configuration

In order to find the LQ-controller for our specified Q and R values the algraic riccatti equation was solved with the idare() function in matlab. This function returns the closed-loop state feedback gain for the optimisation of the cost function J. These gains can be used as an optimal full-information controller for the specified Q and R matrices.

Hence, with only tuning of the Q and R matrices we can heuristically determine the behaviour of the controller by setting the costs on the states and input. Looking at the A matrix of the discrete system, we see the states are not decoupled in the used representation. This means that a cost put explicitly on one state also puts a cost on the other states implicitly. Moreover, we see that the output of the discrete system solely depends on the 3rd and 4th states.

$$Q = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 2*10^3 & 0 \\ 0 & 0 & 0 & 10^8 \end{bmatrix}, \qquad R = [0] \tag{17}$$

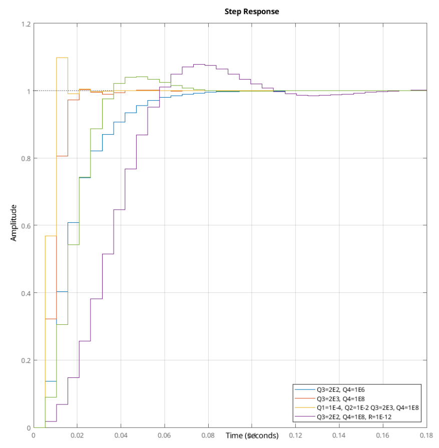


Figure 24: Reference step response of a variety of LQ controller tunings

We will therefore put corresponding costs on these states respectively. We will construct the Q matrix to contain the C matrix, but augmented with zeros. This causes the states that disrupt the output most to also be prioritised the most in the minimisation of the cost. Moreover, R is set to be 0, because in this implementation we are not (yet) interested in limiting our input. We see in Figure 24 that introducing non-zero costs for the other states and the input, or changing the costs on the 3rd and 4th states slows down the system response, or causes an overshoot that is too large. Here the best response was obtained using shown in red and spelled out in Equation 17.

Q8 | Input Saturation

In real-world systems there must be limitations on the gains that are added to the signal. There can be multiple reasons for this, which could be anything from power limitations on the controller, to the plant being unable to operate with input gains that are to high, to the conduit of the signal not being able to conduct such energy, etc. In a hard drive disk it is logical that the motor that drives the laser positioning has limits on the forces it can excert; Meaning that we can not get arbitrarily fast position responses in the real world. If the motor gets these inputs it will not operate properly and might even get damaged.

Therefore in this assignment an input gain of 200 has been set as the maximum allowable gain on the plant's input; i.e. controller output. In the PID-type controllers the transferfunctions for the u with respect to the reference signal can be found. These equations will be used to find the gain of the u signal for a unit-step reference signal.

$$\begin{aligned} \text{Reference: } \frac{u(s)}{r(s)} &= \frac{P(s)}{1+C(s)P(s)} \\ \text{Disturbance: } \frac{u(s)}{r(s)} &= \frac{C(s)P(s)}{1+C(s)P(s)} \end{aligned} \tag{18}$$

For the model based controllers we can modify the C and D matrices to equate y to u. Here it is important to realise that this does not simply mean that and $C = 0$ $D = I$ for direct feedthrough, because this will return the r signal. This is because r is the input of the closed loop system. Taking a look at Figure 23 we can deduce that the required C and D matrices are $C = -F$ and $D = I$ respectively. Because this is a SISO system one could also consider D to be 1.

Q9 | Input-Saturated PID

Reference Tracking

The PID-type controller that was tuned earlier for the reference trakcing does not satify the criterion of a gain maximum of 200 for the u signal. Intially the overall gain of the controller was lowered, but this rednered the system unstable. Instead the controller needed to be re-tuned.

The input saturation constraint complicated the tuning of the dicrete controller. The main reason for this is that the magnitude of the step response does not match between the different time domains. It is therefore easier to approximate the discrete-time response with the continuous-time controller. This was done using Equation 19, where $\hat{P}(s)$ is an approximation of $P(z)$ in the laplace domain.

$$\hat{P}(s) := P(s)e^{-\frac{T_s}{2}} \tag{19}$$

Using this approximation the D-controller in Equation 20 was found. It is clearly visible in Figure 25 that this controller is significantly slower than the CT controller. However, we also see in Figure 26 that the input saturation constraint is not met. In order to assure the controller does this, an additional step was taken to add overall gain to the controller. The added gain was the ratio between 200 and the DC-gain of the step response, such that the step response maxes out exactly at 200.

$$C(s) = \frac{200 * 60D}{\text{DC-gain} * \left(\frac{s}{120} + 1\right)} \tag{20}$$

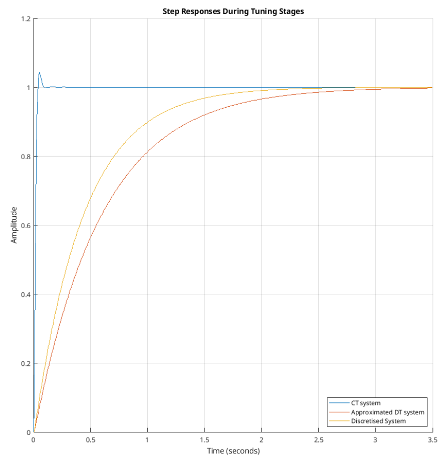


Figure 25: Step response of the reference tracking PID-type controller retuned for the input saturation constrained

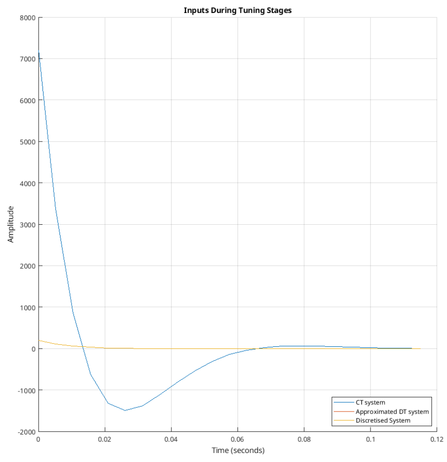


Figure 26: Step response of the reference tracking PID-type controller retuned for the input saturation constrained

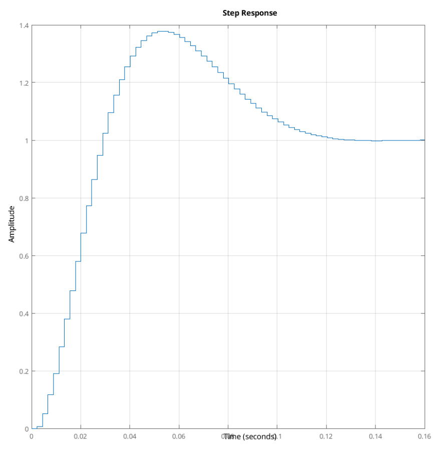


Figure 27: Disturance step response, which already satisfied the input saturation constraint

Disturbance Rejection

Interestingly, in the case of disturbance rejection, the already tuned controller was sufficient. The input gain of this controller stays below 1.4, which is surprisingly low, considering the maximum is 200. During tuning of the disturbance rejection controller in CT, I found that there was a tradeoff to be made between settling time and peak magnitude of the response. Since both fastness of the rejection and insignificance of the disturbance were tuning goals, I opted to lean more into insignificancew of the disturbance. As a result the system is quite slow, but does not require a high input gain. It is therefore not necessary to tune the controller again.

Q10 | Input-Saturated Pole Placement

The initial pole placement controller goes far above the saturations limit. To combat this, the poles will need to be placed slower (i.e. lower magnitude). Hence, it is not necessary to change the sampling rate. Without changing the sampling rate, we will have an oversampled system. Because of this there is little difference in the characteristics of the discrete- and continuous-time systems.

$$-20.703 \pm 21.711i \quad \& \quad -54.281 \pm 8.862i \qquad (21)$$

After re-tuning, the poles were determined to be those in Equation 21. These poles maintain the same phase as the poles in thr before tuned pole placement controller. The only alteration is the magnitude of both poles.

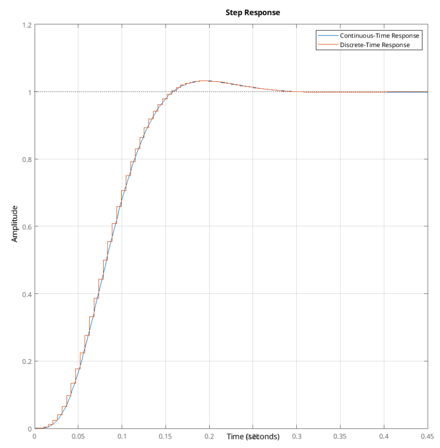


Figure 28: Reference step response after re-tuning for input saturation constraint of pole placement controller

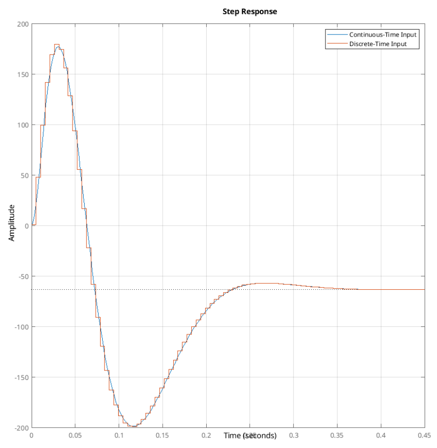


Figure 29: Input of the pole placement controller after re-tuning for input saturation constraint of pole placement controller

Before, in order to keep the dominance of the slower pole sufficient, the faster pole was placed at 10 times the magnitude of the dominant pole. In this case the factor is lower than 2. Yet, there is no noticable effect of the non-dominant pole in the step response. This was in all honesty a bit of a surprise. In contrast, the effect of the non-dominant pole on the input gain is quite significant. Because of this it was important to find the right balance in the pole magnitudes. A lower non-dominant pole magnitude yields a lower input gain, but can introduce a secondary mode in the step response if it gets to close to the dominant pole pair. Because of the the dominant pole pair can also not get a too high magnitude. This cause the step response to slow down. A higher dominant pole pair magnitude would require a higher non-dominant pole pair magnitude. This makes the pole placement quite constrictive, which required a bit of searching between the right dominant magnitude and the ratio between the magnitudes.

Q11 | Input-Saturated LQR Controller

In order to incorporate the constraint on the input gain the R matrix was adjusted. The Q matrix did not need to be adjusted. The relations between the states did not change. Neither did the effects of the states on the step-response. It does therefore make sense to leave the R matrix as-is, and instead inroduce a cost on the input. Because the input cost is a scalar this is simply a matter of increasing it until the criterion is met.

$$Q = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 2*10^3 & 0 \\ 0 & 0 & 0 & 10^8 \end{bmatrix}, \quad R = [3.4*10^{-9}] \tag{22}$$

The value in R is rather small. This means that it is likely the case that the signal going into the plant is not directly driving the plant; i.e. The energy of u is not the energy that powers the motor, but some other power source is likely uitilised.

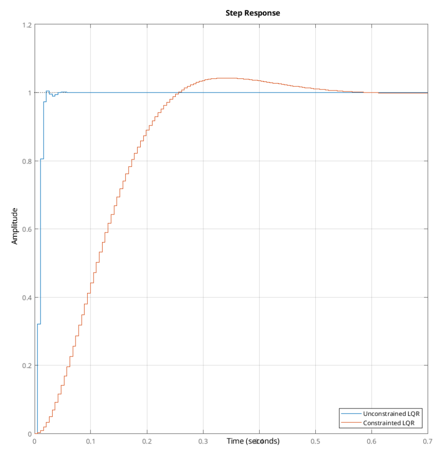


Figure 30: Reference step response after re-tuning for input saturation constraint of LQ controller

In LQR control cases this is relevant information, because the order of magnitude between the values of the states and the u signal needs to be reflected in the costs. Besides the order of magnitude being different, it is also highly likly that in an arbitrarty control system the states and inpuit signal will have a different units, which further argues why one should not raise their eyebrows at these vastly different costs.

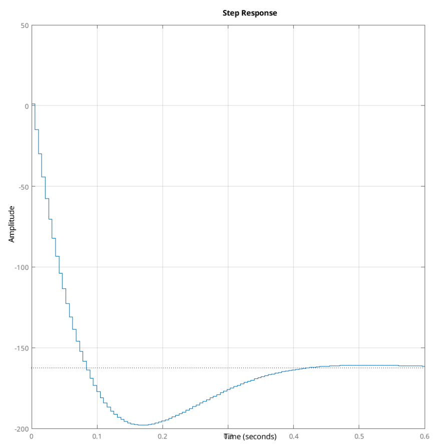


Figure 31: Input of the pole placement controller after re-tuning for input saturation constraint of LQ controller

we see in Figure 31 that the initial control signal is zero,shoots down to nearly −200, and eventually setlles at . . . It is good that the control signal settles with a margin from the saturation point. This makes it such that a controller could also deal with some pertubations in the system, without saturating the control signal. This can cause the disturbance rejection of the system to not only be slower, the clipping that occurs might even destabilise the controlled system alltogether.

Appropriate tuning to combine disturbance rejection with the reference tracking controller that we have here will likely change each of the costs on the states and input differently. We will therefore likely need to find completely different LQR matrices to implement this. However, this is outside of the scope of question 10 in the assignment.

Q12 | Steady State Errors

Throughout this assignment the steady state errors that have been occuring have been dealt with whenever they arose. However, this section will provide an overview of how these errors were dealt with.

PID-type controllers

In PID-type controllers it is common practice to apply the finite value theorem to see if there will be steady state errors with respect to the reference signal. Non-convergence can arise due to sustained residual oscilations, or a constant steady-state offset. Residual oscilations can generally be dealt with through the addition of a D-term in the controller. Intuitively, this will look bring the derivative of the signal to zero, meaning that the D-term will cause the steady-state response to flatline if the applied controller is stabilising in the first place. Not every system must have residual oscilations and therefore a D-term is not always required.

If the controlled converges, but not towards the reference value it has a constant steady-state offset from the reference signal then an I-term will be usefull. The intuition behind this term is that that it will bring the integral of the response with respect to the reference to zero, and will therefore cause the stady-state offset to disappear over time.

Pole placement controllers

With pole placement controllers one can simply look at what the characteristics are of the poles in the phase plane of the closed loop system. As we have seen in Figure 2, the closed loop uncontrolled system displayed residual osccilations in the step-response; i.e. marginal stability. This can be explained by a complex-conjugte pole pair on the imaginary axis. The angle of poles with respect to the origin is directly related to the damping ratio of the step response of those poles. Therefore, when one places the poles in a system on oughts to understand that the placement of these poles will determine the damping ratio, and hence the persistence of oscilations around reference, if the controller stabilises.

With regard to the steady-state value of the system; the most straight forward way to combat this is to add feed-forward gain to the reference, taking into account the steady-state offset from the reference, and the reference itself.

LQR Controllers

The tuning of LQR controllers works more heuristically, which makes it difficult to explicitly tune them to be damped-stable. One's best course of action here is to use an understanding of the workings of the plant, such that the state- and input-consts can be appropriately tunes while ensuring that $Q \geq 0$ and $R > 0$, such that the Algebraic Riccatti equation will yield a unique optimal solution to the control problem.

In order to deal with steady-state errors, the exact same can be done as with pole placement controllers, where a feed-forward term will adjust the reference, such that the steady-state value of the step response will be that of the originally intended reference.

Q13 | Input Delays

Lastly, before closing with some final thoughts on this assignment, we will look at the effects of a single-step delay on all types of controllers that were used in this assignment.

Implementation of the Delays

In this section delays will be implemented in the systems' control signals, such that its effect on the controllers found in the previous sections can be studied. Because the PID-type controllers use the transferfunction of the system, while the pole placement and LQR controllers use the state-space representation of the system, the exact implementation will be different for both cases.

Transfer function

Ther implementation of a delay in a discrete-time transferfunction is rather simple. In order to do this we will use the tranferfunctions for $\frac{y(z)}{r(z)}$ that were found in Equation 3 and Equation 7. In the discrete time domain, we don't only need to specify the amount of steps delay added to the system, rahter than the exact time. This boils down to simply multiplying the tranferfunction with z^{-1}, as shown in Equation 23.

$$T(z)^{\text{delayed}} = \frac{1}{z}T(z) \tag{23}$$

This increases the order of ther denominator by 1 in the transferfunction, which is equal to adding an aditional state. We will see in that the implementation of a delay in the state-space system also involves the addition of a new state.

State-Space System

The incorporation of a delay into the state-space representation requires the control input to become part of the states. This "stores" the state for one time step and updates the stored input every new timestep. This causes a retention time of the input in the system, i.e. a delay. The required state-space augmentation can be seen in Equation 24.

$$\begin{bmatrix} x(k+1) \\ u(k) \end{bmatrix} = \begin{bmatrix} \Phi & \Gamma \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x(k) \\ u(k-1) \end{bmatrix} + \begin{bmatrix} 0 \\ I \end{bmatrix} u(k)$$
$$y(k) = [C \;\; 0] \begin{bmatrix} x(k) \\ u(k-1) \end{bmatrix} + D(u(k)) \tag{24}$$

Delayed Reference Tracking PID control

In Figure 32 we see the effect of the single-step delay on the PD-controller that was found for the reference tracking problem. Here we see that the delay shifts the initial effort by one time step. For the first four timesteps this does not seem to have a very significant effect, other than delaying the system, and slighyl changing the output vlaues for each step. However, we see that the system has a much larger overshoot than the original system. The following oscilations also have a much larger amplitude, causeing the settling time to be pushed back by much more time steps than just the delay.

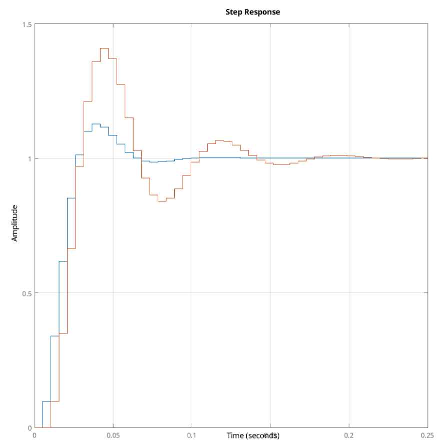


Figure 32: Step response of the reference traking PID-type controller and its response with a delay

Delayed Disturbance Rejection PID control

Contrarily, to what we have seen in Figure 32, the sreponses in Figure 33 show a lot of resemblence. Not much is changed in the characterisics of the step resonse, other than the peak of the disturbance response being higher. The effect of the delay is visible in the first couple steps, where we see a similar effect to what happened for the reference tracking; A similar, but delayed response. However, where the effect fo the delay grew for reference tracking, here is seems to diminish over time. This could be because of over-sampling, but changes in the sampling rate seem to be significantly detrimental to the performance of the discretised controller, even before adding a delay. Since this is not necissarily a bad thing, I therefore opted to keep the sampling rate as-is.

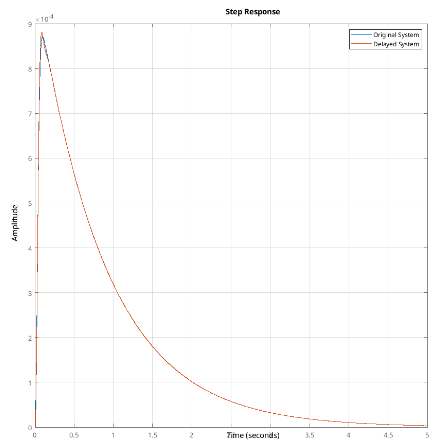


Figure 33: Step response of the disturbance rejection PID-type controller and its response with a delay

Delayed Reference Tracking Pole Placement Control

Pole placement control Shows the best performance for reference trakcing so far. Where the PD-controller for reference trakcing did not just delay the output values, but also altered them, this controller yields the same output as the original controller, but simply delayed by one timestep. This makes sense, because the implementation of the dfelay in the PID controller effectively changed the tranferfunction, and thus the system dynamics, while the pole placement controller still has the same system dynamics. Without the reference, the system does not have any response. So in this case the delayed response is simply that of the original controller, but the controller starts acting one timestep later.

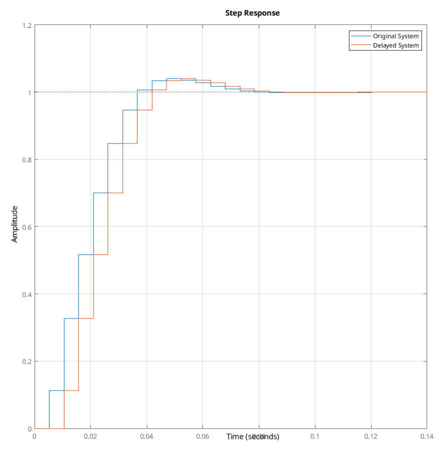


Figure 34: Step response of the pole placement controller and its response with a delay

Delayed Reference Tracking LQR Control

Here too, it can be said that the response is the same as that of the original controller, but just delayed by one time step. This makes sense, since the implementation of the delay is is identical between the pole placement, and the the LQR controller.

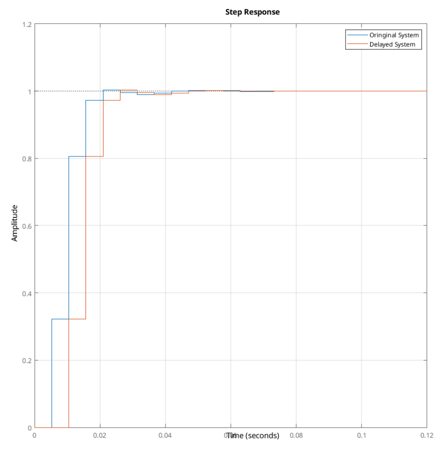


Figure 35: Step response of the LQ controller and its response with a delay

Final Thoughts

Having implemented several controllers, it has become clear than not all controllers are created equally. The heuristic aspect of model free control isboth it's strenght and its weakness. The PID-type controllers can be implemented regerdless if the system is controllable, observible or some weaker forms of these. The step response and bode diagrams provide significant insight in how the controllers can be tuned, but we have little idea of the optimality of the controllers.

With pole placement we can place the poles intentionally to give it desired characteristics that are quanitfiable. However, the viability of this control method relies on the controllability and observability of this system. If certain modes are unstable and not controllable, or we cannot apply full-information control while being unobservible, everyhring goes out the window.

LQR is in my eyes somewhere in between. The tuning of costs on the states and the inputs are performed somewhat heuristically, while it is a model-based control method. This implementation also assumed full-information, but this is not nescicarily an option in practical implementations.

For both pole-placement and LQR, the implementation of the observer introduces a possible point of failure. While a full-information pole placement controller can be exact in what the system characteristics ought to be, the quality of the observer can be at the detriment of the controller and can even cause a well performing full-information controller to fail.

Another interesting realisation that arose during this assignment is that constraints, such as input saturation, does not translate one-to-one from a continuous to a discretised controller. This was true for both the model-based and model-free controllers. For the model based controllers the the magnitude of the poles and the input costs had to be adjusted, while the the PID-type controllers had to be retuned completely. It can occur that that the overall gain of the model-free controllers could just be lowered, but in this case this interfered with the overshoot constraint. Because of this the controllers need to be re-tuned. This was a more labourious task than with the model-based controllers.

When deciding on what controller I would actually implement on this system I am quite certain I would opt for a model-based controller. For me personally, with PID-type controllers my experience is that the tuning parameters are found through a laborious process; And even then, we cannot say much about the optimality of this controller. I have seen instances where people were able to express the pole locations in terms of the tuning paramters, but any of these cases I have seen among my peers have sofar been with a lower order plant. I have spent several days attempting to do this myself too and I found that even matlab was not recourceful enough to find these poles. While expressing the poles symbolically I came to the point rather quickly where matlab concluded that the ‘root of said equation equals the root of said equation’. When even matlab had nothing insightful to say I switched to loopshaping.

Hence, a model-based controller has my preference. Moreover, I found it to be rather simple to account for input saturation, while the controllers were also performant with a time delay on the control. I don’t personally have a lot of insight into pole placement in the z-domain, but I found it rather straight forward to work with the different domains simultaneously.

I was able to work with only the LQ controller in the z-domain directly. While the costs themselves might need to be different between the continuous and discrete controllers, there is no difference in intuition between the tuning in continuous and discrete time. There is therefore nothing holding me from tuning in discrete time directly. Even though the values for the costs remain an educated guess, insight into the system’s dynamics comes a long way.

In the end I was able to tune the best controller in the DT domain by using pole placement, including the input saturation constraint. Hence why I would go for this controller if I had to implement one if this plant were a real one. However, I did not attempt to see how the pole placement controller hold up agains the LQR controller using an observer. So I can only speak for a a full-information scenario.