Catholic
University
Leuven

**Department of
Electrical Engineering**

COMPUTERGESTUURDE REGELTECHNIEK: ROTARY FLEXIBLE JOINT

De Weer Sofie - r0802671

Pollet Max - r0757781

Academic Year 2022–2023

Contents

List of Figures

List of Tables

Introduction

This report will analyse and simulate in real-time a rotary flexible joint with an LQR controller. Before making the controller will an open-loop analysis be conducted to ensure that the system is observable and controllable. Afterwards will the controller be made with LQR strategy. The choice of the best controller is made by applying a step change to the state which needs to be controlled. The results from that are important to ensure that the controller is not too fast and doesn't cause the actuators to saturate too much.

Then will the system with controller with a Simulink setup of the real-time setup to simulate the real-time simulation as best as possible. This setup is then experimented with different step changes and a frequency response. These results are then used to assess the difference between simulation and the real-time setup.

Lastly will the controller be tested with a real-time setup. This will give a result on how good the controller is and if the simulation is close to real life.

All the simulations are performed in MATLAB and Simulink. Appendix A explains what the function of every file provided with the report is.

Chapter 1

Open-loop analysis

1.1 Linear state space model

The model given in the exercise is shown in equation 1.1. The model has four states θ, α, $\dot{\theta}$ and $\dot{\alpha}$. The angle θ describes the angle of the hub wrt the origin and the angle α describes the angle of the joint wrt to the hub. Since there are only two sensors for the angles of the hub θ and the arm α, will the system only have those as outputs.

$$\begin{bmatrix} \dot{\theta} \\ \dot{\alpha} \\ \ddot{\theta} \\ \ddot{\alpha} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & \frac{K_{stiff}}{J_h} & -\frac{K_g^2 K_m K_b}{J_h R_m} & 0 \\ 0 & -\frac{(J_l+J_h)K_{stiff}}{J_l J_h} & \frac{K_g^2 K_m K_b}{J_h R_m} & 0 \end{bmatrix} \begin{bmatrix} \theta \\ \alpha \\ \dot{\theta} \\ \dot{\alpha} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \frac{K_m K_g}{R_m J_h} \\ -\frac{K_m K_g}{R_m J_h} \end{bmatrix} V \tag{1.1}$$

Filling in the values of the parameters, which can be found in table 1.1 will lead to the following system matrices.

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 765.98 & -52.80 & 0 \\ 0 & -1038.62 & 52.80 & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 \\ 0 \\ 98.33 \\ -98.33 \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

$$D = \mathbf{0}_{2\times 1}$$

Parameter	Symbol	Value	Unit
Motor armature resistance	R_m	2.6	Ω
Motor torque constant	K_m	0.00767	Nm/A
Motor back EMF constant	K_b	0.00767	$V/rad/s$
Motor gear ratio	K_g	70:1	-
Arm inertia	J_l	0.0059	kgm^2
Hub inertia (includes motor and gears)	J_h	0.0021	kgm^2
Body anchor point (y-coordinate)	d	0.00318	m
Arm anchor point	R	0.076	m
Body anchor point (x-coordinate)	r	0.0318	m
Spring stiffness	K_{stiff}	1.60856	N/m
Spring restoring force	F_r	1.33	N
Spring length at rest	L	0.0318	m

Table 1.1: Parameters of the rotary flexible joint setup

1.2 Open Loop Analysis

1.2.1 Stability

The eigenvalues of the matrix A are shown in table 1.2. This table also shows whether or not the eigenvalue is also a pole. Since the four eigenvalues are also poles does this mean that the system is minimal.

λ_i	value	pole?
λ_1	0	yes
λ_2	-34.84	yes
λ_3	$-8.98 + 18.24j$	yes
λ_4	$-8.98 - 18.24j$	yes

Table 1.2: Eigenvalues of A

Continuous time systems are only internally stable when the real part of all eigenvalues are negative. This is not the case here, since $\Re(\lambda_1) = 0$. And because the system is minimal, is it also not input-output stable. The system is only marginally stable.

The system has no transmission zeros. A plot of the poles and zeros is shown in figure 1.1.

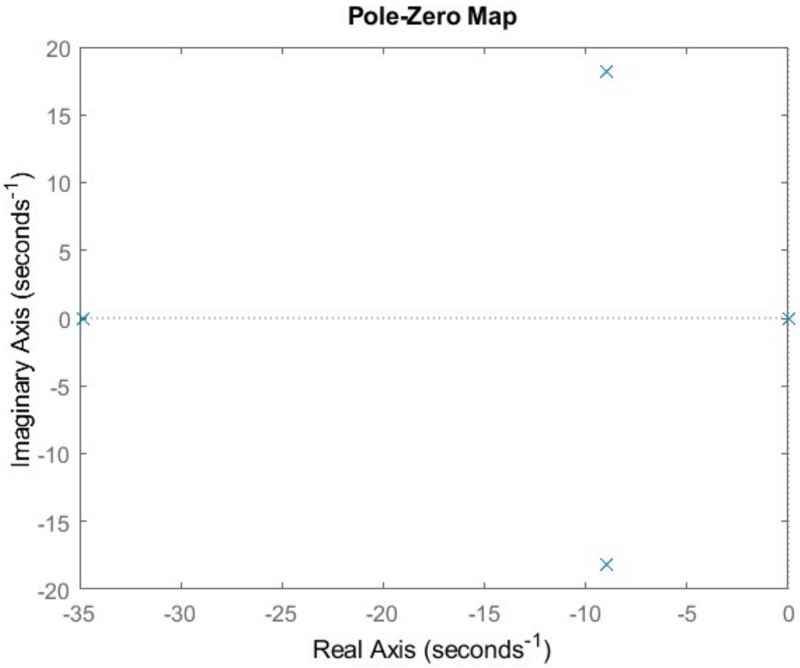


Figure 1.1: Poles and zeros of the system

1.2.2 Controllability and Observability

A quick check of the rank of both the observability and controllability matrices, given in equation 1.2, shows that the system is both observable and controllable. The ranks are both 4, which is the order of the system.

$$\begin{gathered}
Co = \begin{bmatrix} 0 & 98.33 & -5.19\times 10^3 & 1.99\times 10^5 \\ 0 & -98.33 & 5.19\times 10^3 & --1.72\times 10^5 \\ 98.33 & -5.19\times 10^3 & 1.99\times 10^5 & -6.52\times 10^6 \\ -98.33 & 5.19\times 10^3 & -1.72\times 10^5 & 5.10\times 10^6 \end{bmatrix} \\
Ob = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 765.98 & -52.80 & 0 \\ 0 & -1.04\times 10^3 & 52.80 & 0 \\ 0 & -4.04\times 10^4 & 2.79\times 10^3 & 765.98 \\ 0 & 4.04\times 10^4 & -2.79\times 10^3 & -1.04\times 10^3 \end{bmatrix}
\end{gathered} \tag{1.2}$$

Because of these two properties being applicable will the system also be stabilizable and detectable.

1.3 Control goals

The controller must be designed in such a way that the state θ follows a certain reference trajectory. The goals of this controller are thus more tracking based. Since the system is not stable but stabilizable shall the controller also be able to stabilize the system which in turn makes disturbance rejection possible. A fast response of the system is not really desirable since the controller must try to not saturate the actuators. A less nervous controller is thus more preferable.

Chapter 2

Design of the LQR controller and creation of the first closed-loop simulation diagram

2.1 Simulink diagram

The Simulink diagram is pictured in figure 2.1. The continuous system is loaded in with the system matrices A and B and an eye matrix for C and a zero matrix for D. That way will the states be available from the state space block. The states are then subtracted with the desired reference. This reference is a step input on θ. The error is then multiplied by the control gain $-K$ to obtain the new control input for the state space. For evaluation purposes are some outputs added which are later used to plot the results.

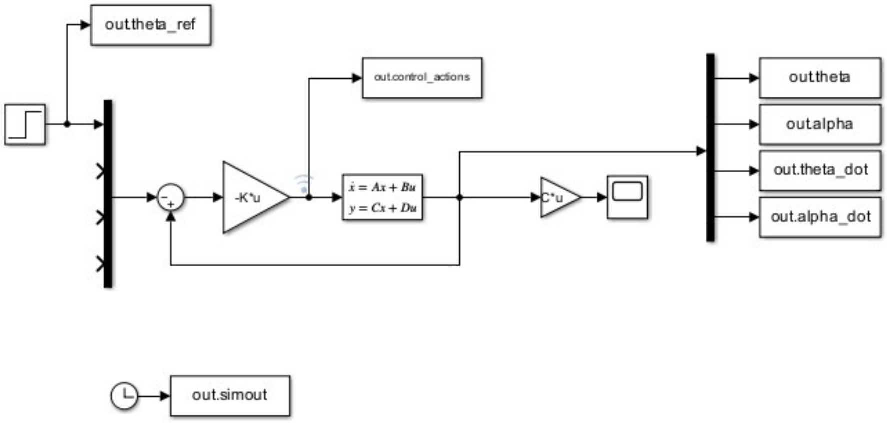
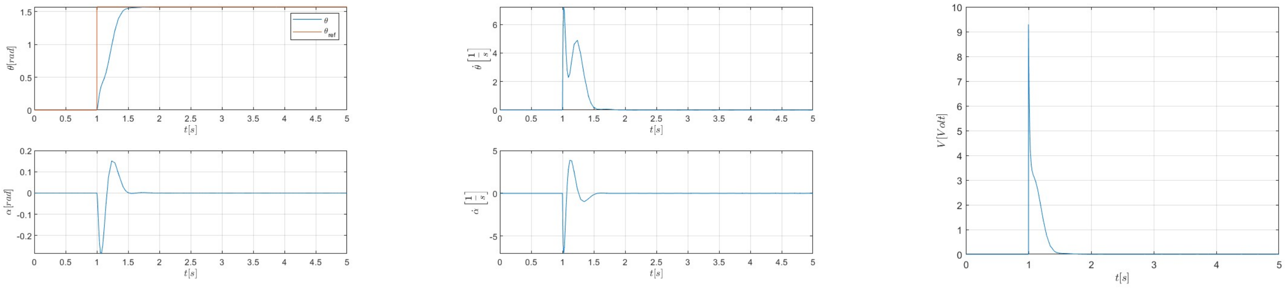


Figure 2.1: Simulink diagram of the closed loop system

2.2 LQR controller

The given matrices Q and R are given in equation 2.1. These matrices already give a very satisfied result which are shown in figure 2.2 when a step of $\pi/2$ is made. When increasing the values of Q will the controller be faster in minimizing the errors of that state. Thus when $Q(1,1)$ is increased will the angle θ more closely follow the step reference θ_{ref}. Decreasing the values of the Q-matrix will have the opposite effect. When the values of the R-matrix are changed will this have an effect on the control actions. Increasing will lower the amount of control action since its product wants to stay as close to zero, while decreasing these values will cause the control actions to increase.

$$
\begin{aligned}
Q &= \begin{bmatrix} 350 & 0 & 0 & 0 \\ 0 & 1500 & 0 & 0 \\ 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0.5 \end{bmatrix} \\
R &= 10
\end{aligned}
\tag{2.1}
$$



(a) The output of the given controller

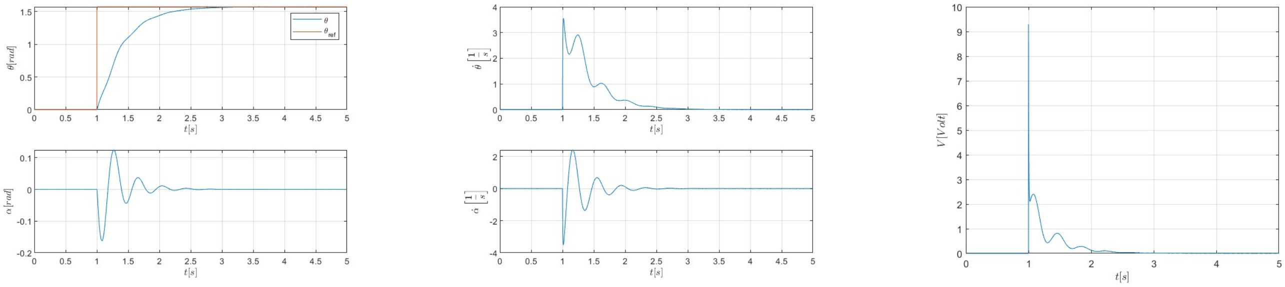
(b) The velocities of the given controller

(c) The control actions of the given controller

Figure 2.2: Step response of the given controller

Trying to find the correct values of the Q and R-matrices is a case of trial and error. Thus a few different configurations are tried out, for which a few will be analyzed here.

When the value of $Q(3,3)$ is set to 50 will the controller try to minimize the state $\dot{\theta}$. This means that changes in the state θ will always be very small and thus the outcome is a slow controller. The same result could also be obtained by increasing the value of R. Figures 2.3a and 2.3c show the output and control actions respectively when having the reference do a step of $\pi/2$. In total are the control actions roughly the same as the initial controller, only the states and the velocities are different. As expected, is the maximum value of $\dot{\theta}$ lower than the initial controller, due to the increase of the value of $Q(3,3)$. This results in the other states to be slower or to oscillate more.



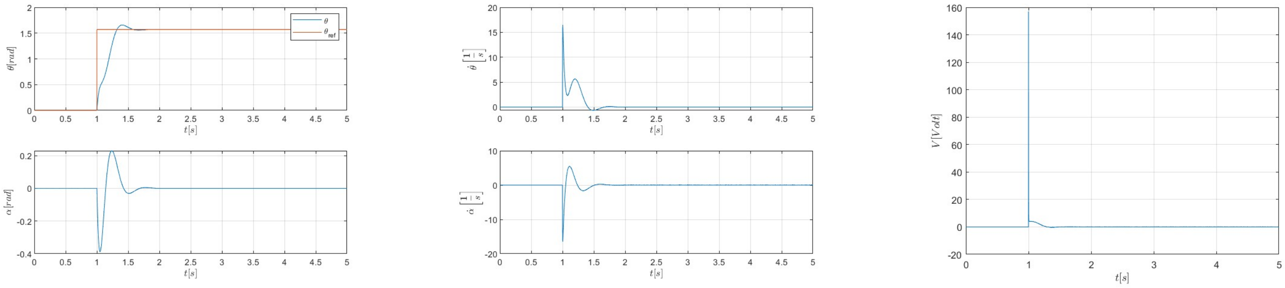
(a) The output of the slow controller
.

(b) The velocities of the slow controller

(c) The control actions of the slow controller

Figure 2.3: Step response of the slow controller

When the value of R is changed to 0.05 will the controller be able to amplify the control actions to very high numbers. This is not possible in the real application, meaning that this controller would cause the actuator to saturate for a long time. This is also seen in the results obtained from the simulation as seen in figures 2.4a for the output and 2.4c for the control actions. The control actions reach a value of around 160 Volts, which is too high for the real life plant. The outputs also show an overshoot when the step is made. Due to the controller being much faster will the velocities in figure 2.4b reach much higher values than before.



(a) The output of a nervous controller

(b) The velocities of a nervous controller

(c) The control actions of a nervous controller

Figure 2.4: Step response of a nervous controller

After trial and error was the optimal controller found. This controller changed $Q(1,1)$ to 400. This causes the reaction of θ to be faster than before. The closed-loop poles are shown in table 2.1 and are all stable. The step response is shown in 2.5, with the output shown in figure 2.5a, the velocities in 2.5b and the control actions shown in 2.5c. As expected is the rise time shorter than the original controller. The overall control actions stay the same and the highest the voltage goes is 10V. While the control actions do become higher than the actuator can give, is this for such a short amount of time that it won't cause the closed-loop system to become uncontrollable.

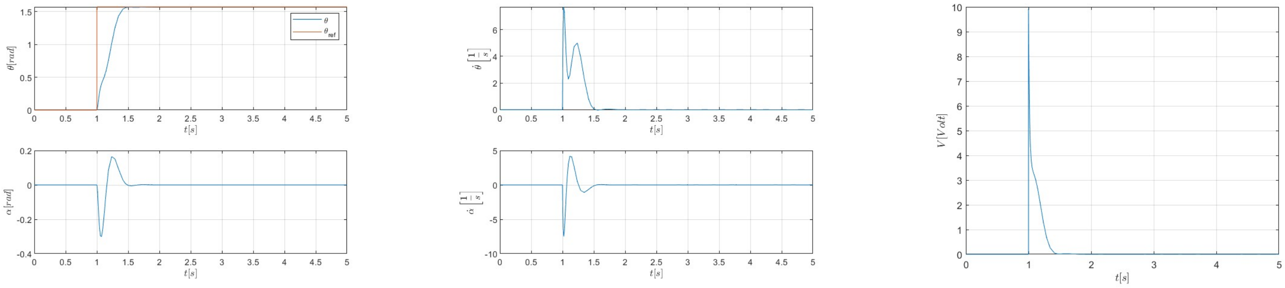
λ_i	value
λ_1	$-9.4012 + i12.9992$
λ_2	$-9.4012 - i12.9992$
λ_3	-10.2257
λ_4	-64.4285

Table 2.1: Closed-loop poles of the system

The obtained gain matrix K is:

$$K = \begin{bmatrix} 6.3246 & -6.7051 & 0.6418 & 0.2283 \end{bmatrix}$$

These values also show that the states θ and α have the most influence on the control actions as their corresponding values in the Q-matrix are the highest.



(a) The output of the optimal controller

(b) The velocities of the optimal controller

(c) The control actions of the optimal controller

Figure 2.5: Step response of the optimal controller

Chapter 3

Second closed-loop simulation diagram

3.1 Simulink diagram

The simulink diagram seen in figure 3.2 must be able to simulate the controller that will be used in the experiments. Instead of the actual rotary flexible joint is the continuous time state space block used, where the matrices are the ones used in equation 1.1. Since the outputs are analog signals must they be discretized with the quantizer seen in figure 3.3. The quantizer will first transform the radians to volts. Then, according to the quantization will the voltage be given the right value. The given information about the quantization is that the A/D converter had a voltage range of -10V to 10V with a resolution of 16 bits. This can be translated as the converter having 2^{16} possible numbers to describe each voltage. Thus, the converter has a difference of $\Delta V = \frac{20}{2^{16}} = 3.0518 \times 10^{-4}$ between each voltage. The *quantizer.m* code will find the right value for each of the analog signals to make it digital. Lastly will the voltages be retransformed to radians. Since there will always be noise on the sensors will the sensor noise be added to the output of the digital signals. The noise is random and Gaussian and can thus not be exactly estimated as seen clearly in figure 3.1. The only parts that can be estimated are the median and variance. These are estimated via the experiments and give

$$\begin{cases} var[\mathcal{N}_\theta] = 5.2511 \times 10^{-7} \\ var[\mathcal{N}_\alpha] = 1.0068 \times 10^{-7} \end{cases}$$

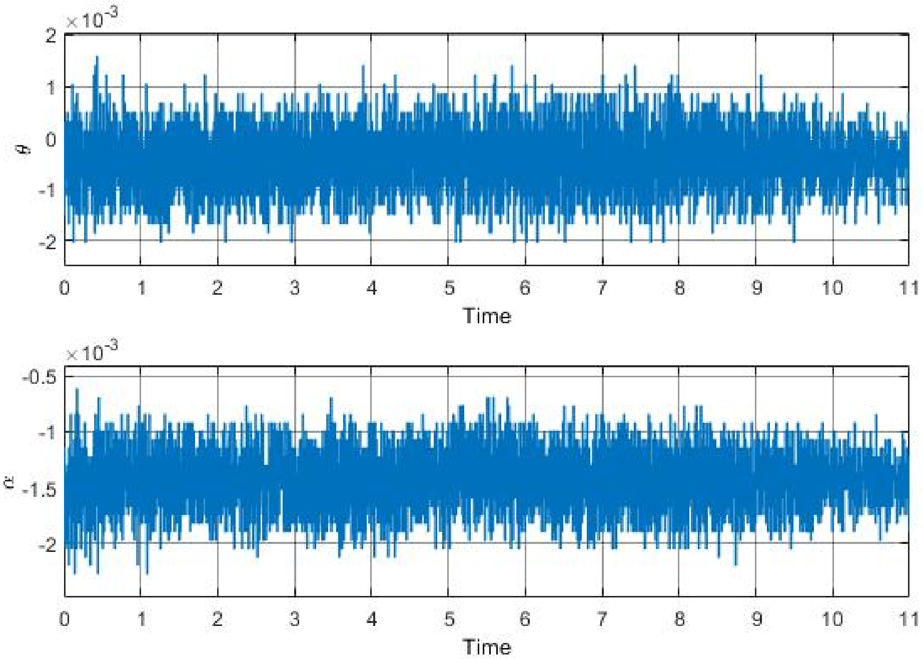


Figure 3.1: Measurements from the setup when the reference is set to 0

To retrieve the derivatives of the output signals will they first have to go through the low-pass filter seen in equation 3.1. The cut-off frequency ω_c is set to 2Hz and the time-step T_s is set to 5ms. This filter will filter out high-frequency signals, while passing low-frequency signals. The filtered signal will then be integrated with the forward Euler method to retrieve $\dot{\theta}$ and $\dot{\alpha}$.

$$F(z) = \frac{\omega_c T_s z}{(1 + \omega_c T_s)z - 1} \tag{3.1}$$

All the signals are then multiplied with their respective value of the gain K. The state θ is first subtracted with the reference θ_{ref} in order to follow the reference. The sum of the signals will result in the control action V. Since the actuator can only have values between $-5V$ and $5V$ is a saturator needed to saturate the control actions.

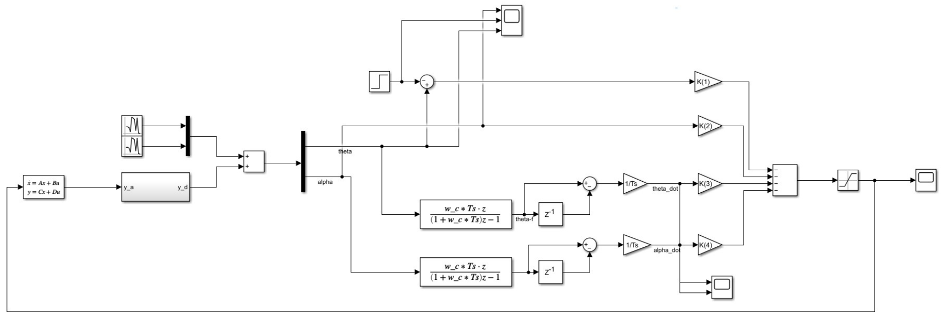


Figure 3.2: Simulink diagram of the second closed-loop system

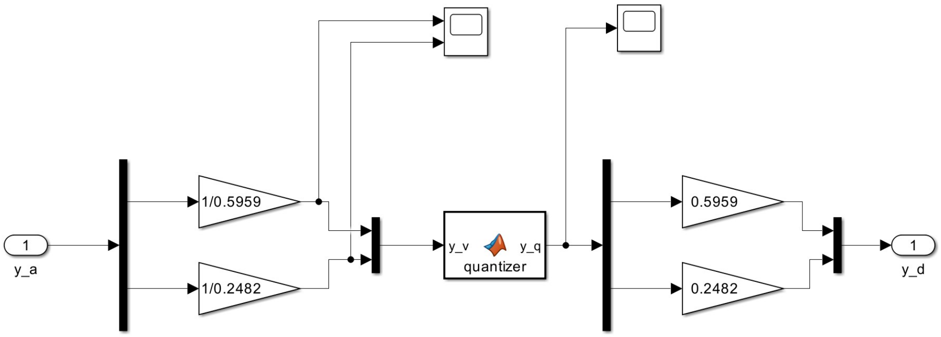


Figure 3.3: Quantization part of the second closed-loop system

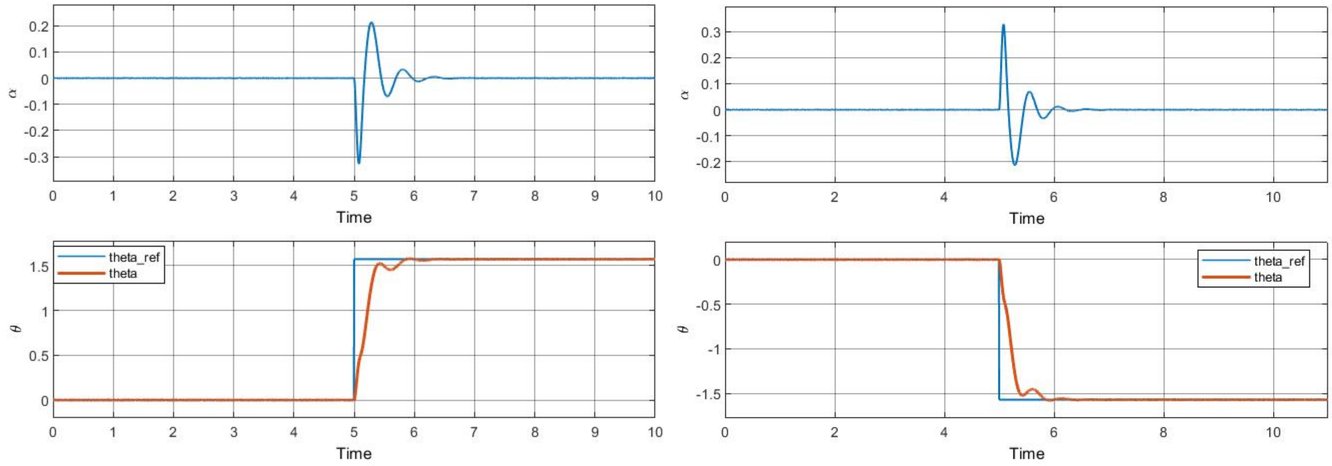
3.2 Step response

The system is simulated for different type of step changes. This makes it easier to compare the results with the previous controller and the real-time simulations.

3.2.1 Step of $\pm\pi/2$

When applying a step of $\pm\frac{\pi}{2}$ to θ_{ref} will the controller be successful in bringing the system to these states. Figure 3.4 shows the step response of the system when applying a step of $\frac{\pi}{2}$ in figure 3.4a and a step of $-\frac{\pi}{2}$ in figure 3.4b. Both figures show that the system needs longer to reach the reference, which can be explained with the control actions seen in figure 3.5. The control actions reach the limit and thus saturate the actuator for a while. This will lead to the system having a limited response and thus taking longer to reach the reference value. The outputs also clearly show that there is some noise present. This noise is almost not notable in the velocities $\dot{\theta}$ and $\dot{\alpha}$ (see figure 3.6) due to the low-pass filter filtering out the noise frequency.

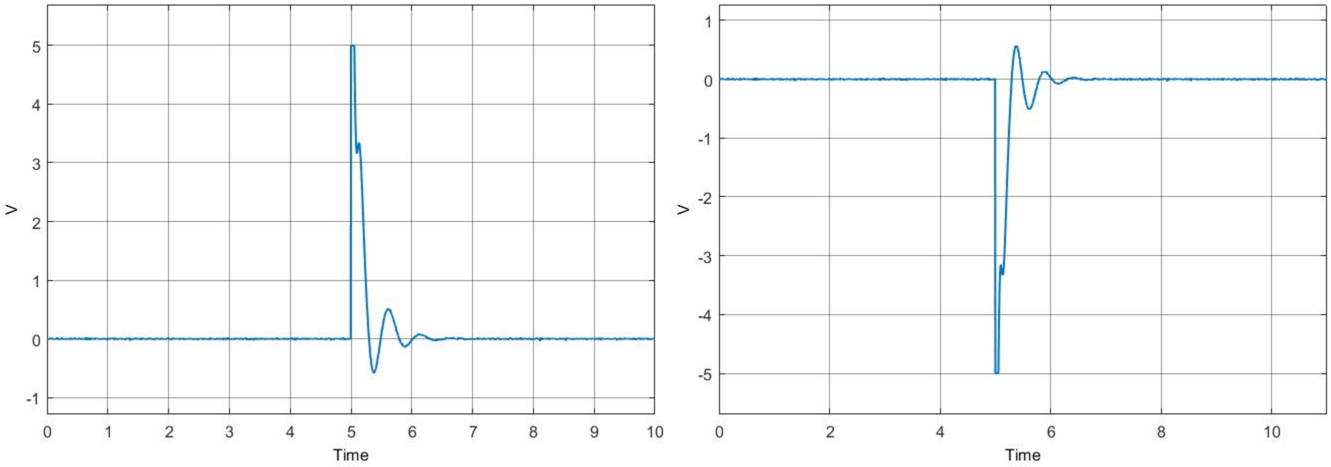
Due to the controller being able to reach the reference in a content time is changing the controller not needed.



(a) $\theta_{ref} = \frac{\pi}{2}$

(b) $\theta_{ref} = -\frac{\pi}{2}$

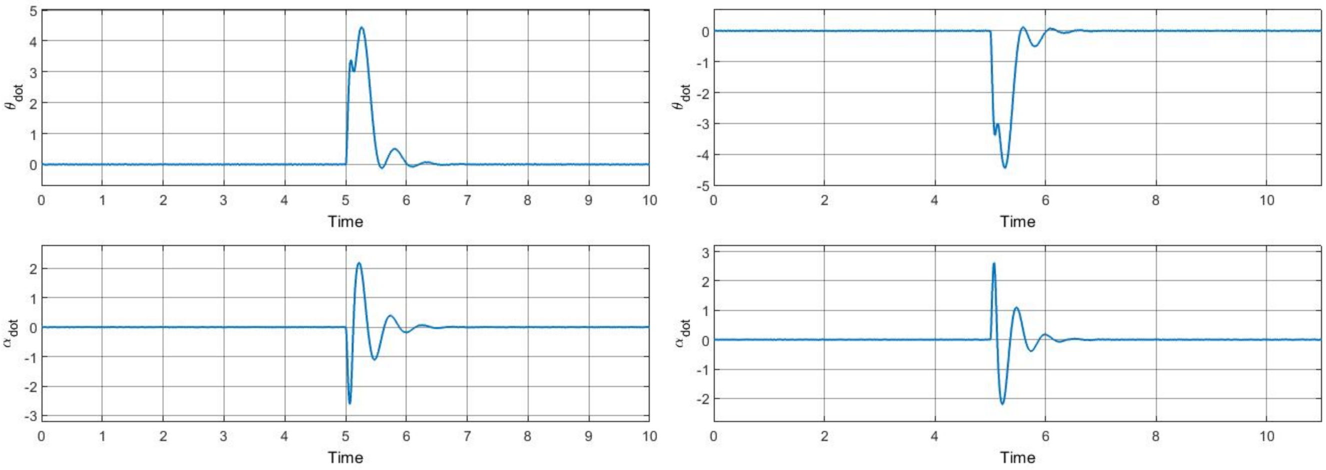
Figure 3.4: θ and α of the simulated step response



(a) $\theta_{ref} = \frac{\pi}{2}$

(b) $\theta_{ref} = -\frac{\pi}{2}$

Figure 3.5: Control actions of the simulated step response



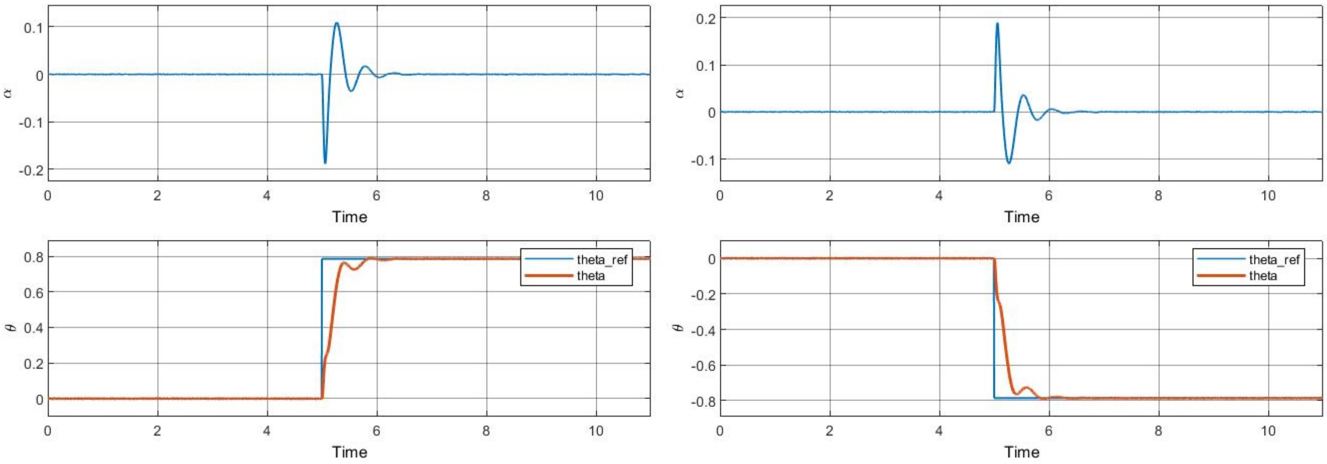
(a) $\theta_{ref} = \frac{\pi}{2}$

(b) $\theta_{ref} = -\frac{\pi}{2}$

Figure 3.6: $\dot{\theta}$ and $\dot{\alpha}$ of the simulated step response

3.2.2 Step of $\pm\pi/4$

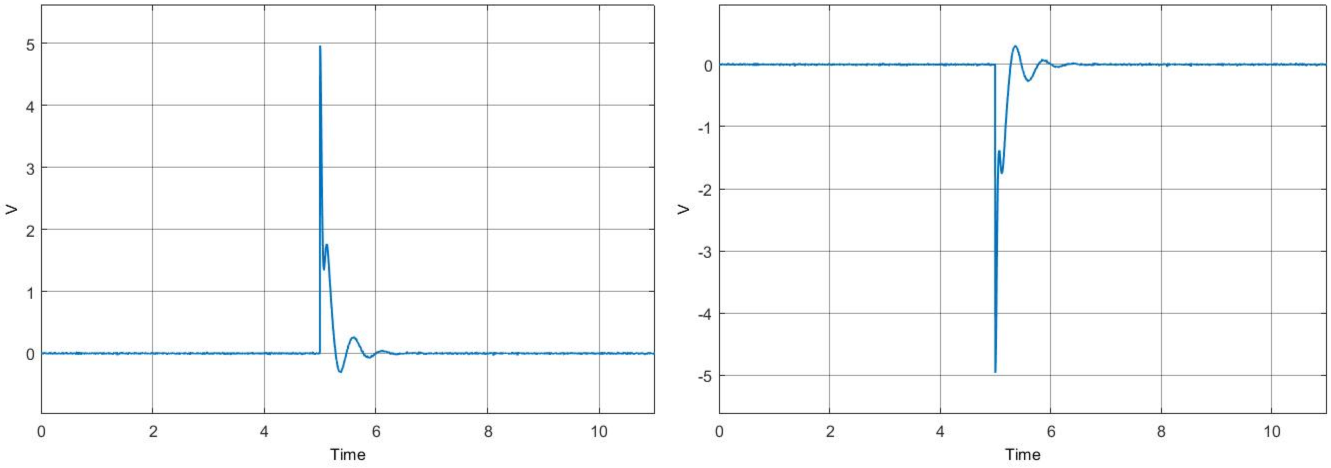
When a step of $\pm\frac{\pi}{4}$ is applied to θ_{ref} will the controller also be able to reach the setpoint with zero steady state error. Figure 3.7 shows the angles θ and α which proves that the controller is successful in both cases. Since the reference point is smaller than before will this lead to the control actions, seen in figure 3.8, to not saturate since they stay in the interval of $\pm 5V$.



(a) $\theta_{ref} = \frac{\pi}{4}$

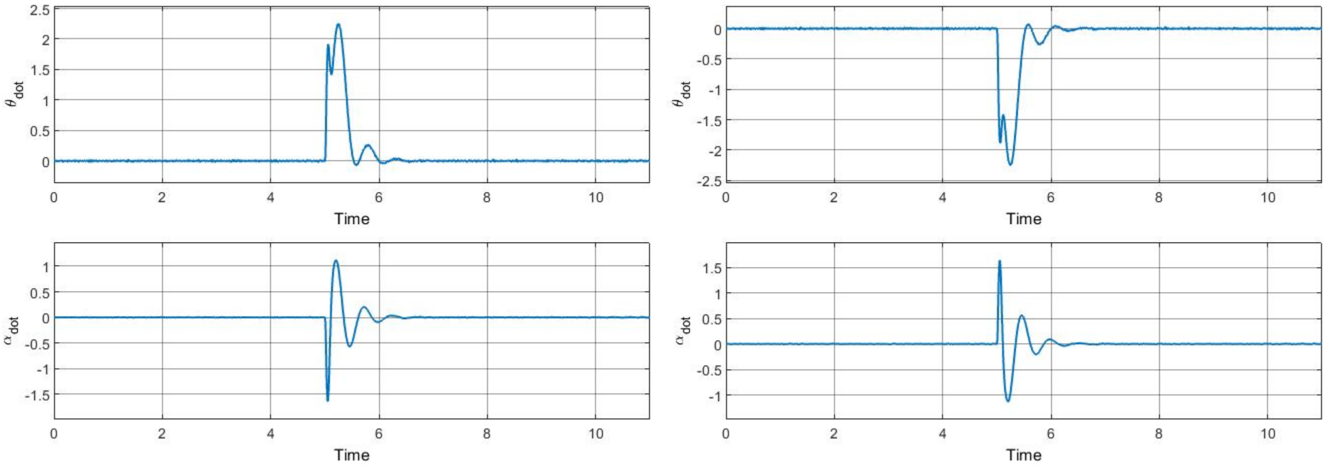
(b) $\theta_{ref} = -\frac{\pi}{4}$

Figure 3.7: θ and α of the simulated step response



(a) $\theta_{ref} = \frac{\pi}{4}$ (b) $\theta_{ref} = -\frac{\pi}{4}$

Figure 3.8: Control actions of the simulated step response

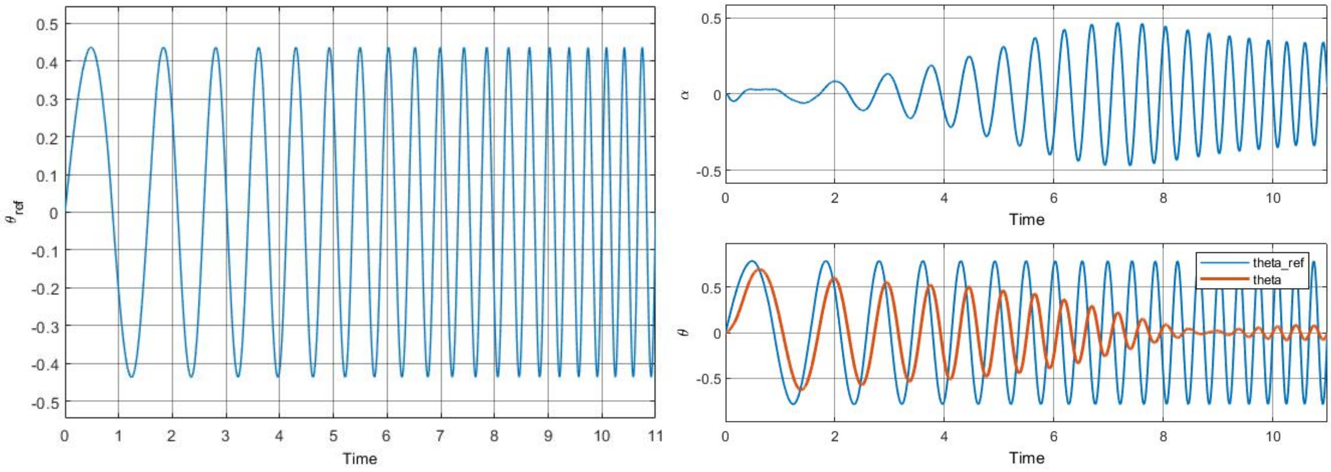


(a) $\theta_{ref} = \frac{\pi}{4}$ (b) $\theta_{ref} = -\frac{\pi}{4}$

Figure 3.9: $\dot{\theta}$ and $\dot{\alpha}$ of the simulated step response

3.3 Role of cut-off frequency ω_c

The cut-off frequency ω_c has influence on the velocities $\dot{\theta}$ and $\dot{\alpha}$. These signals are calculated using the backward Euler method, which uses the difference in the values of the signals at two different time instances. When the cut-off frequency of the filter is not chosen correctly will this lead to the passed signals to be more noise-heavy or very smooth. Both are not satisfactory since a noise-heavy signal will lead to the derivative to be noise heavy and by omission also the control actions. While a smooth signal, i.e. too few frequencies passed through, will lead to a loss of information. Figure 3.11 shows the velocities when the reference is a sinusoidal wave where the frequency is increased from 0.5 to 3Hz. The cutoff frequency is set to 2Hz and it is clear that the velocities are not noisy or lose too much information. The reference is shown in figure 3.10, together with the simulated angles. The evolution of the outputs θ and α can be explained by looking at the Bode plot of the closed-loop system shown in figure 3.13. The plot shows that when $\omega \approx 2.7$ is the gain of θ almost zero, which is why the amplitude of θ drops and rises again and simultaneously becomes in phase with the reference. The same analysis can be done for α.



(a) θ_{ref}

(b) output of the system when $\omega_c = 2Hz$

Figure 3.10: Simulated output of the system when applying a sine reference

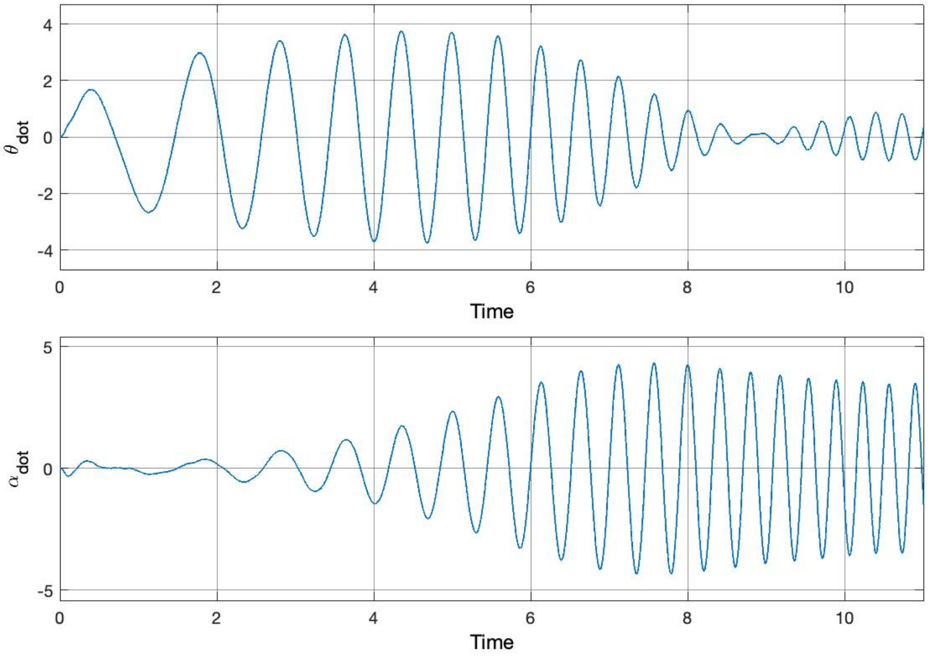


Figure 3.11: $\dot{\theta}$ and $\dot{\alpha}$ when $\omega_c = 2Hz$

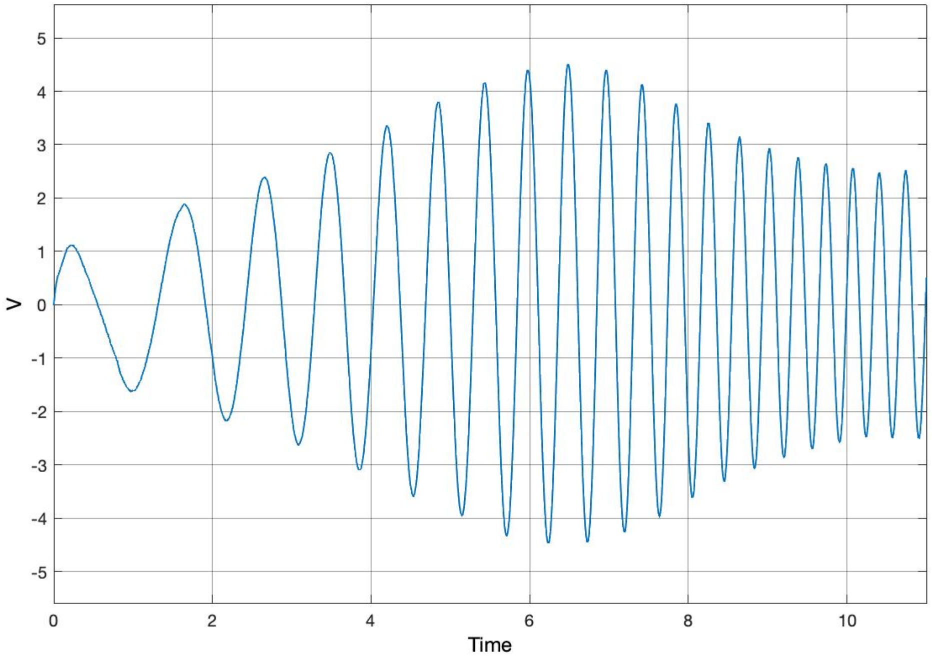


Figure 3.12: Control actions when $\omega_c = 2Hz$

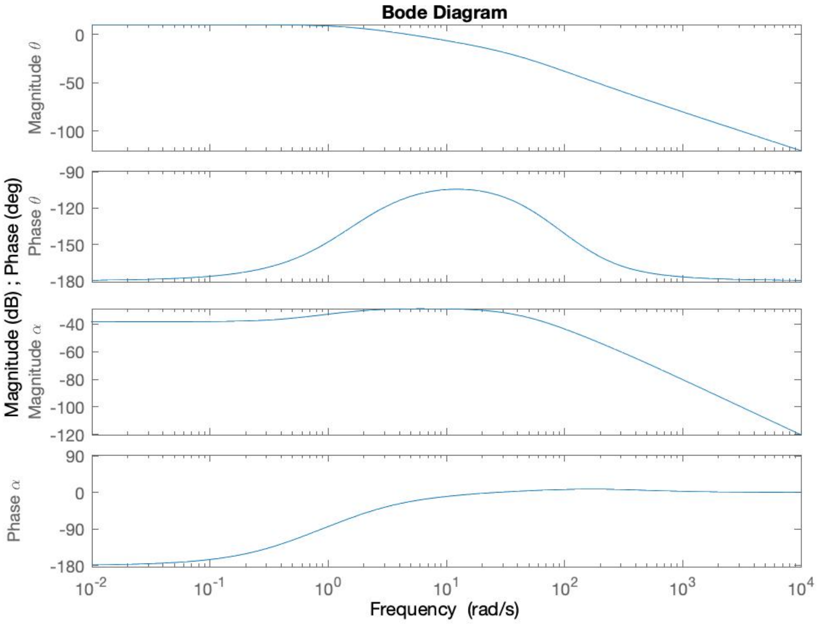


Figure 3.13: Bode diagram of the controller

When the cut-off frequency is too high will very high-frequency signals be able to go through the filter making the calculations noise-heavy, as can be seen in figure 3.14. Here the cut-off frequency is taken as 100Hz, which leads to the velocities to be more noisy than before in figure 3.11. Another impact this will have is on the control actions being more noisy than previously as seen in figure 3.15 contrary to the control actions in figure 3.12.

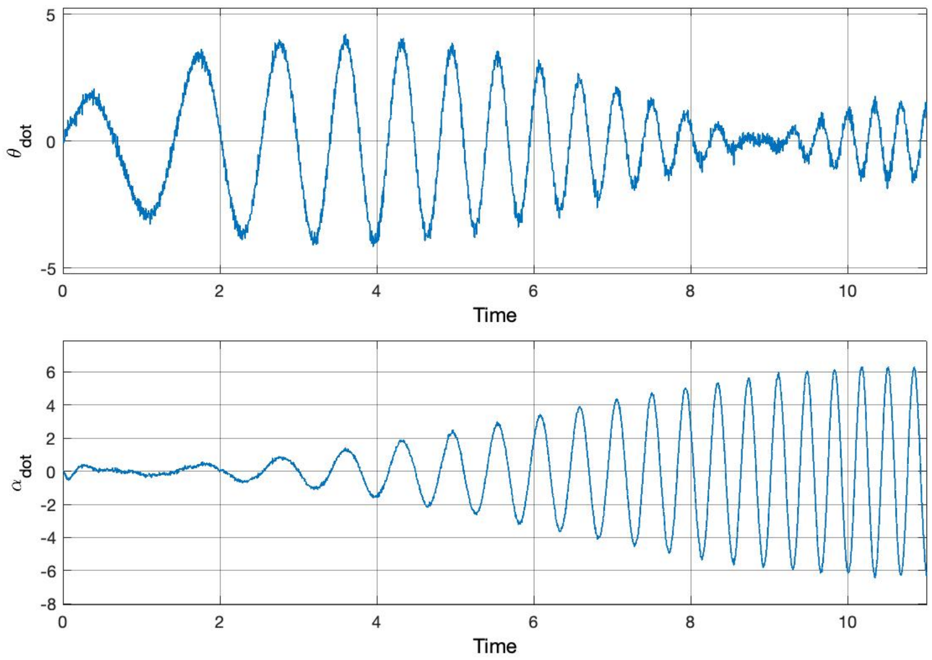


Figure 3.14: Velocities when $\omega_c = 100Hz$

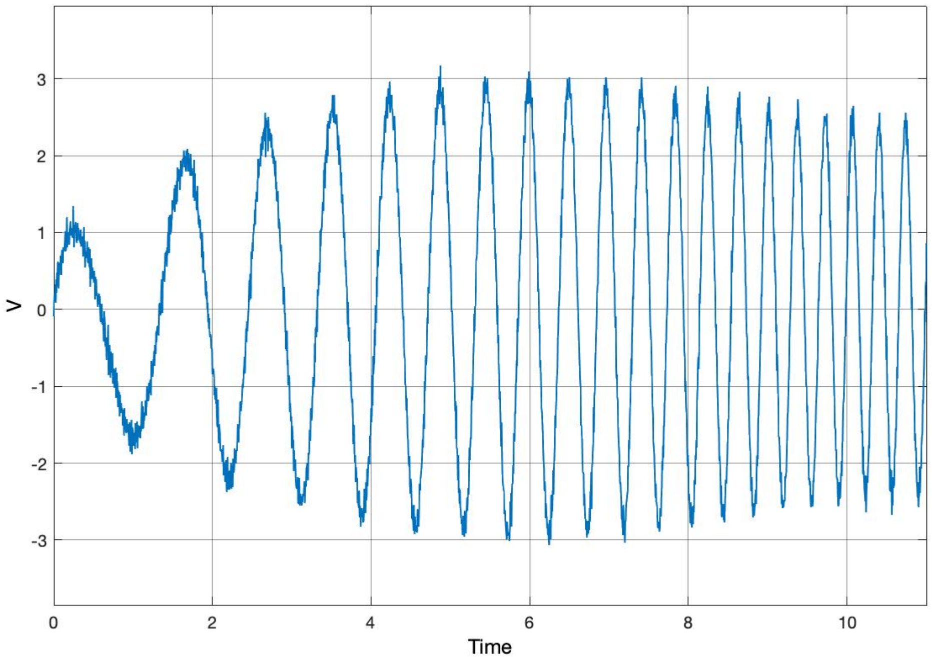


Figure 3.15: Control actions when $\omega_c = 100Hz$

When the cut-off frequency is lowered will this have the effect on the velocities that now the filter won't pass the actual signal, which will lead to the velocities having not enough information to correctly predict the correct value. Figure 3.16 shows the velocities when the cut-off frequency is lowered to $0.5Hz$. There is no noise visible since they have a high frequency and are thus cut off. The figure also shows that the values for the velocities are also lower than previously when θ_{ref} has a high frequency. This is not ideal since now will the signal lose information that is important to recreate the signal and for the controller to take action. Figure 3.17 also proves the fact that the controller continuously needs to take more action to push the system to the right state.

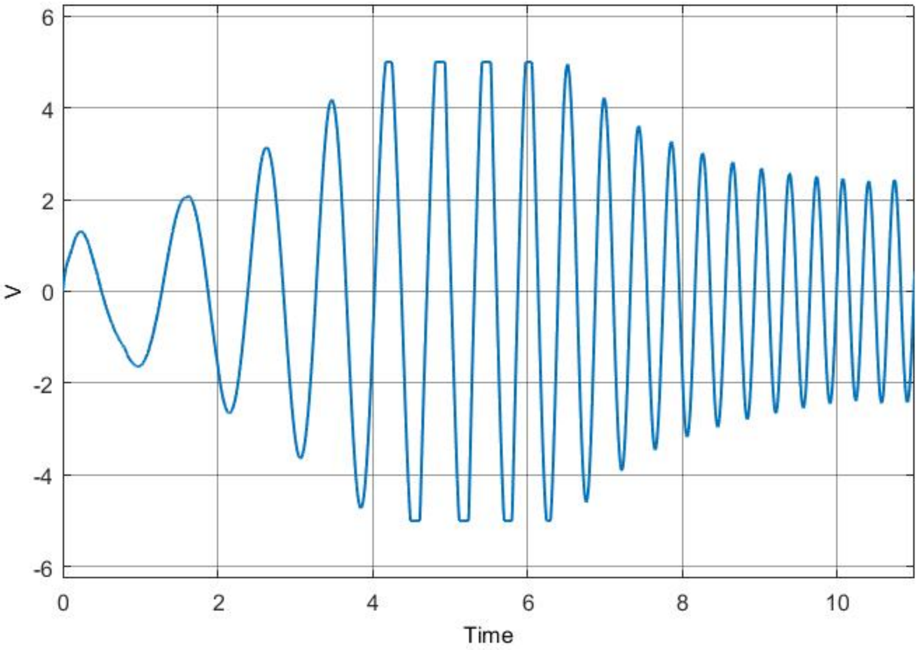


Figure 3.17: Control actions when $\omega_c = 0.5Hz$

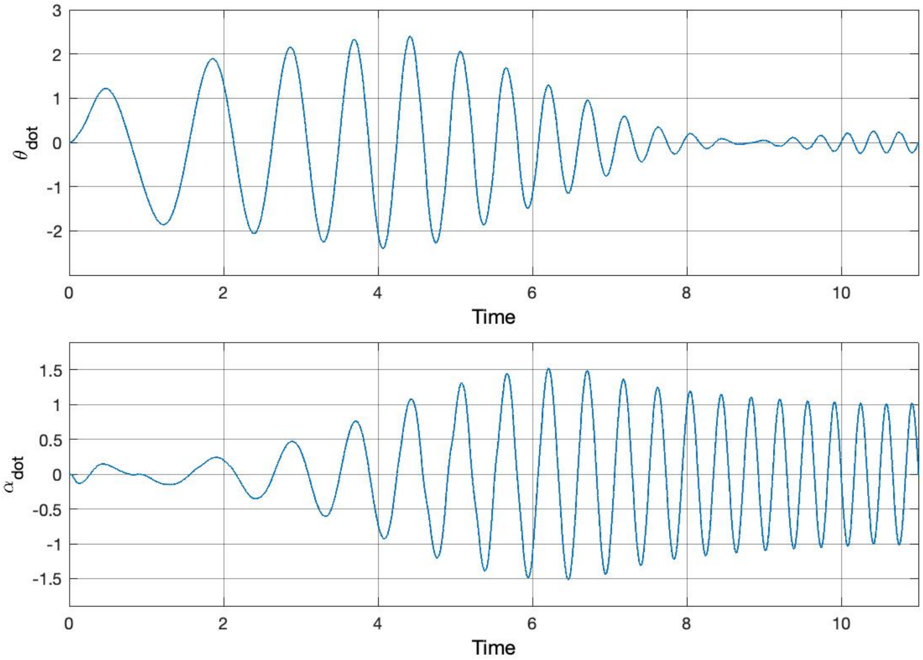


Figure 3.16: Velocities when $\omega_c = 0.5Hz$

Chapter 4

Experimental results

4.1 Real-time Simulink diagram

The real-time Simulink diagram is not much different from the simulated RFJ. The controller and filter, to filter out the noise, are exactly the same. During simulation is the Simulink file the RFJ connected to the computer. The measured angels of the systems are send to the PC to determine the voltage with the controller. The calculated voltage is then send back to the DC motor. Due to the real-time blocks doing the quantization of the analog inputs and saturation of the outputs themselves must those parts not be taken into account in the Simulink diagram.

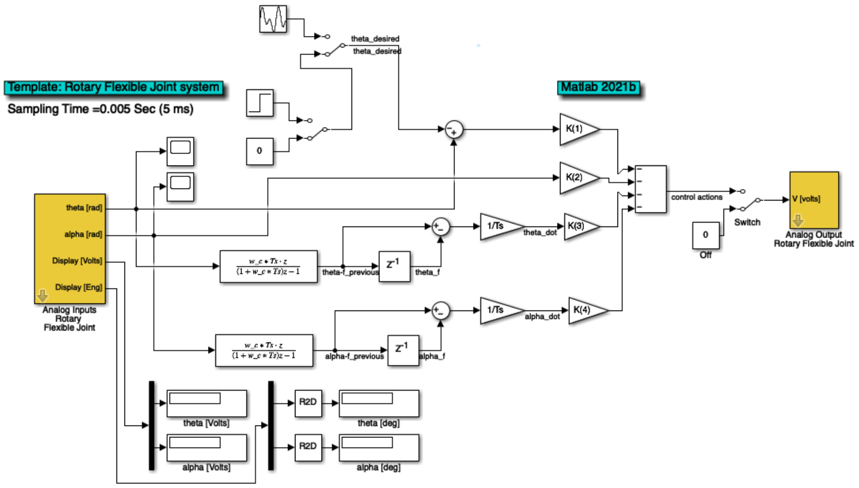


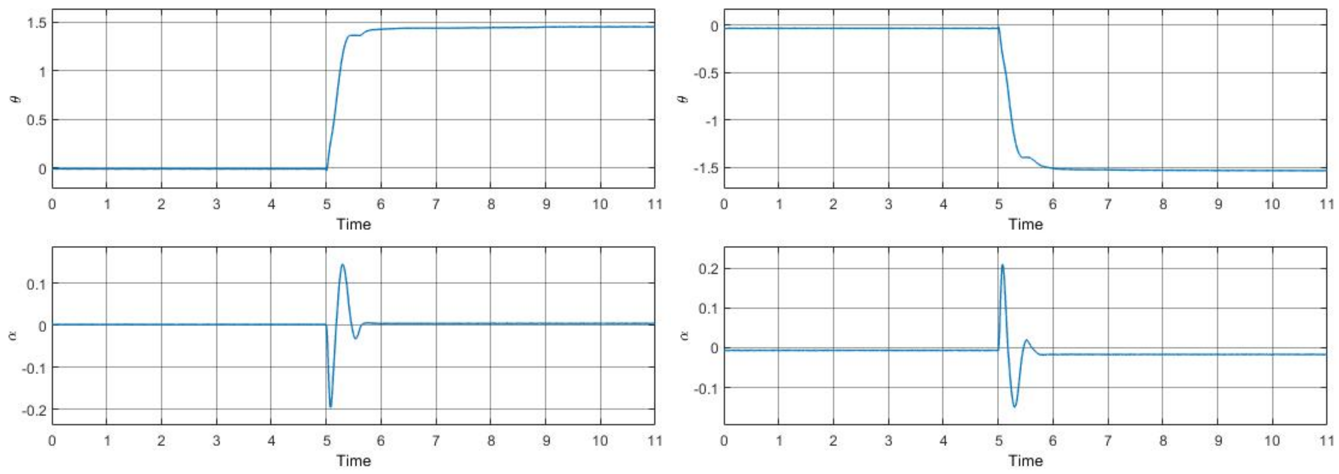
Figure 4.1: Real-time Simulink diagram

4.2 Setpoint tracking

4.2.1 Step of $\pm\pi/2$

As with the simulation will the system be tested with different setpoints. In this simulation is the setpoint set to $\pm\frac{\pi}{2}$. In the simulation in section 3.2.1 was the controller able to bring the system to the right state without a steady state error. When simulating this in real-time will this not be the case as seen in figure 4.2. Both simulations will not reach the value of $\frac{\pi}{2} \approx 1.57$ and thus a steady state error is present. The angle α also has a visible steady state error in figure 4.2b which is due to the offset Off_alpha having the wrong value. The error in the angle θ can best be described by the control actions seen in figure 4.3. They show that when states are stagnant the controller still gives a nonzero voltage to the motor due to $\theta - \theta_{ref} \neq 0$. The motor is thus not able to move the rotary flexible joint with the given voltage due to friction. The system also has two inputs and one input making it overdetermined, i.e. there are too few degrees of freedom to steer every output to the right point. Implementing an integral component could help this problem be solved.

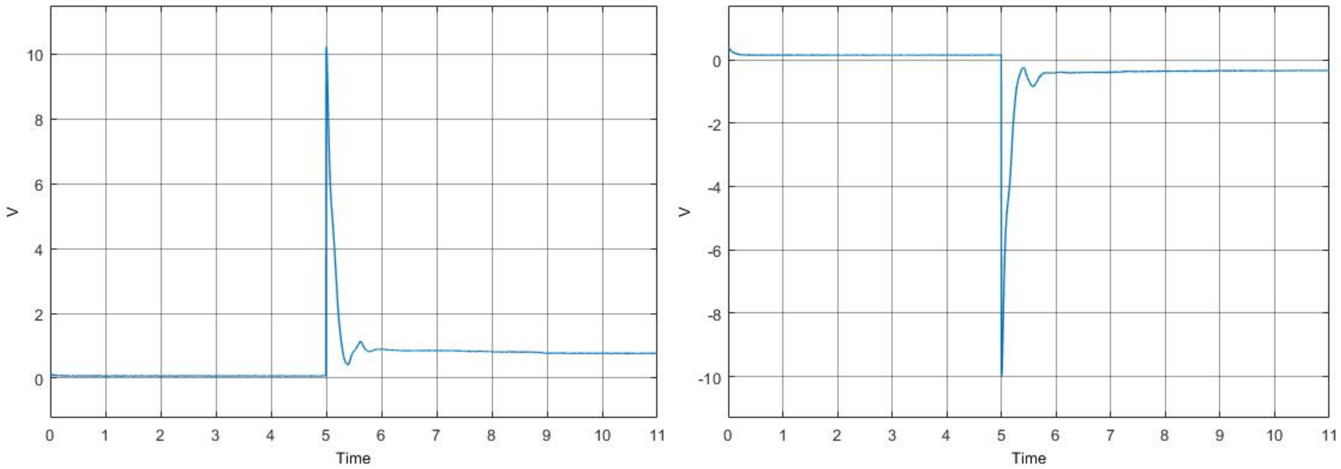
Comparing the results with the ones in section 3.2.1 is there not a big difference except the steady state error visible in the real-time simulation. The simulation also needs more oscillations to reach the setpoints, which is not seen here.



(a) $\theta_{ref} = \frac{\pi}{2}$

(b) $\theta_{ref} = -\frac{\pi}{2}$

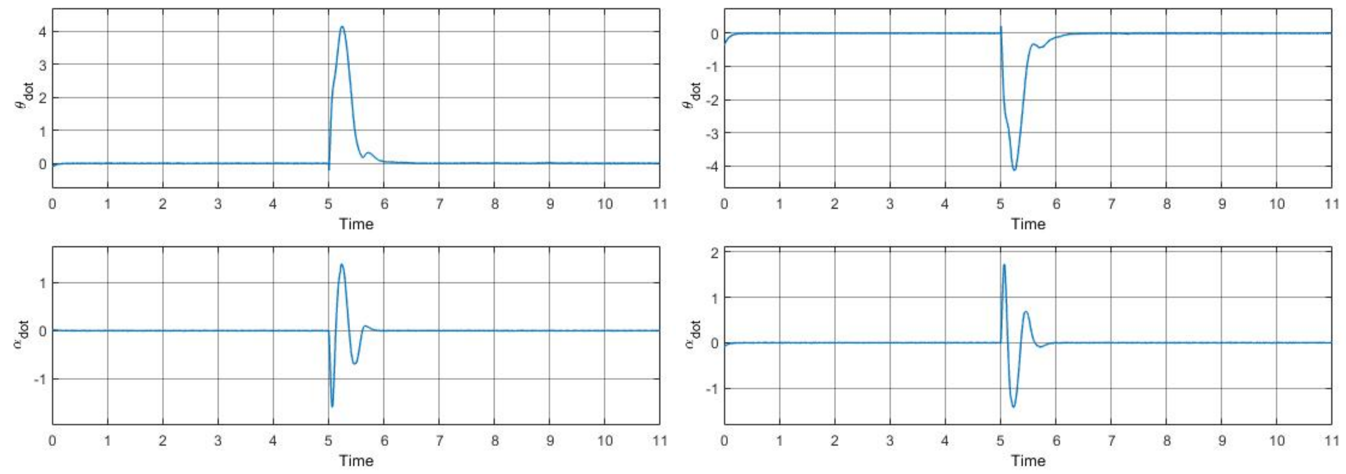
Figure 4.2: θ and α of the real-time simulation



(a) $\theta_{ref} = \frac{\pi}{2}$

(b) $\theta_{ref} = -\frac{\pi}{2}$

Figure 4.3: Control actions of the real-time simulation



(a) $\theta_{ref} = \frac{\pi}{2}$

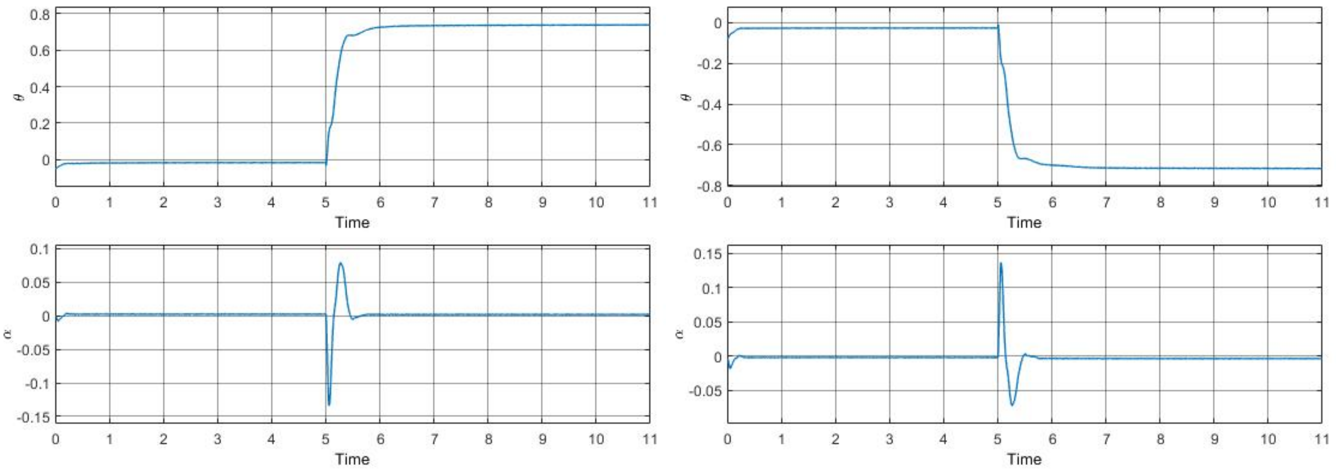
(b) $\theta_{ref} = -\frac{\pi}{2}$

Figure 4.4: $\dot{\theta}$ and $\dot{\alpha}$ of the real-time simulation

4.2.2 step of $\pm\pi/4$

When the system is now given a step of $\pm\frac{\pi}{4}$ will this give the results in figure 4.5. The system should reach a value of ~ 0.707 which is clearly not the case when looking at the control actions in figure 4.6. They are not zero at the end of the simulation. The system thus has a steady state error for the same reason as before, i.e. friction. Implementing an integral component could overcome this problem and make the error go to zero.

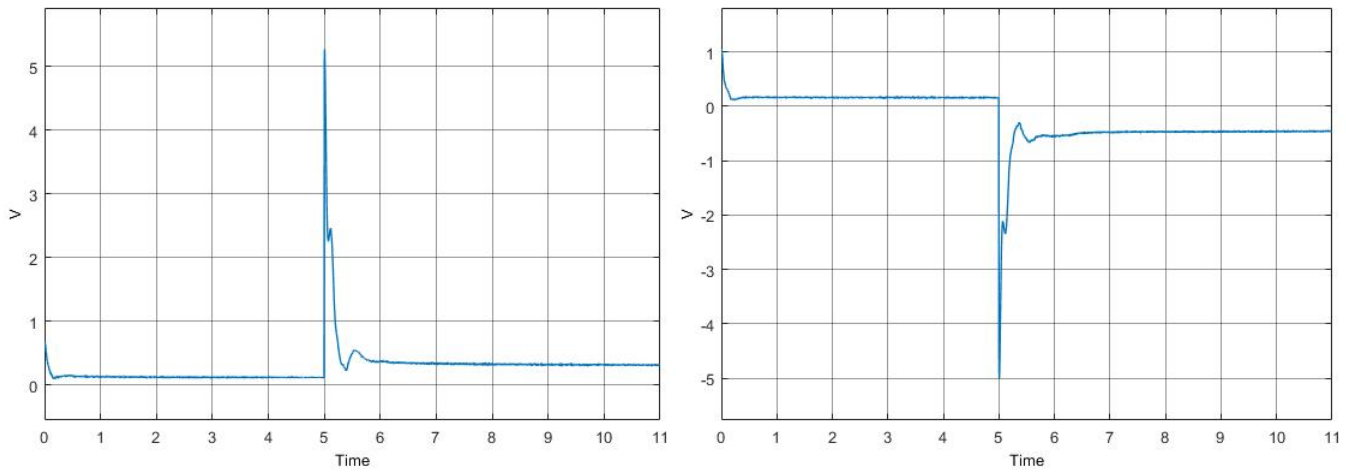
Comparing the real-time Simulink with the simulation from chapter 3.2.2 is the difference small. The biggest difference is that there is less of a ripple at the end of the impulse for both α and θ. The simulation has also no steady state error while the real-time system has some as discussed previously. The offset in the beginning is mainly due to manually having to set the voltages when the system should be in the zero position.



(a) $\theta_{ref} = \frac{\pi}{4}$

(b) $\theta_{ref} = -\frac{\pi}{4}$

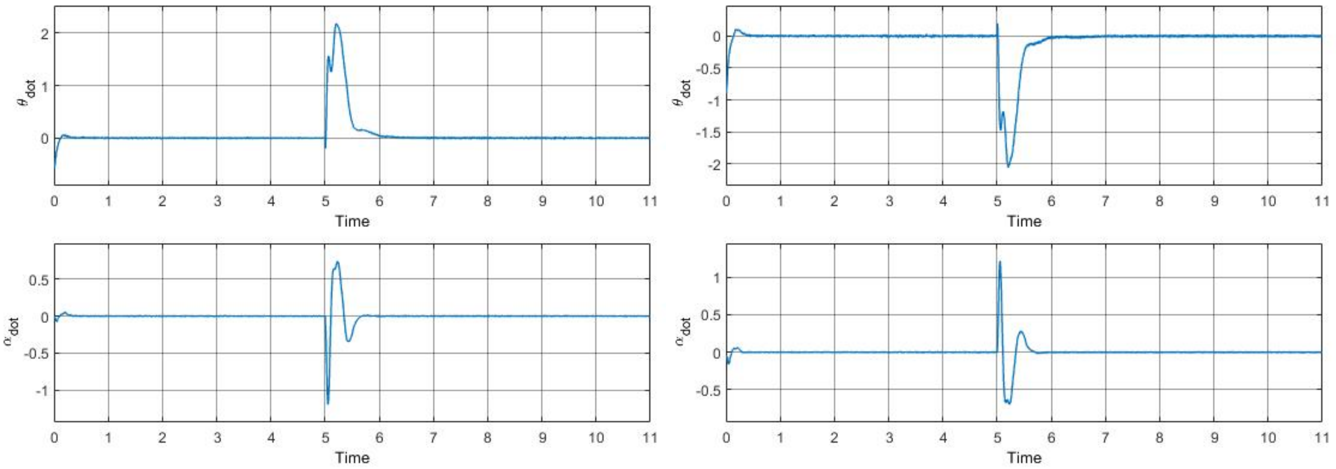
Figure 4.5: θ and α of the real-time simulation



(a) $\theta_{ref} = \frac{\pi}{4}$

(b) $\theta_{ref} = -\frac{\pi}{4}$

Figure 4.6: Control actions of the real-time simulation



(a) $\theta_{ref} = \frac{\pi}{4}$

(b) $\theta_{ref} = -\frac{\pi}{4}$

Figure 4.7: $\dot{\theta}$ and $\dot{\alpha}$ of the real-time simulation

4.3 disturbance rejection

Figures 4.8, 4.9 and 4.10 show the angels velocities and control actions when the systems with the full controller is disturbed. The system counteracts the disturbance and tries to get back to the reference angles which are both zero. After around 0.5 seconds is the RFJ back in the desired position. The controller is thus able to battle small disturbances. As can be seen will the state θ not go fully to the zero position, which can again be explained by the friction the motor cannot overcome as was seen in previous experiments.

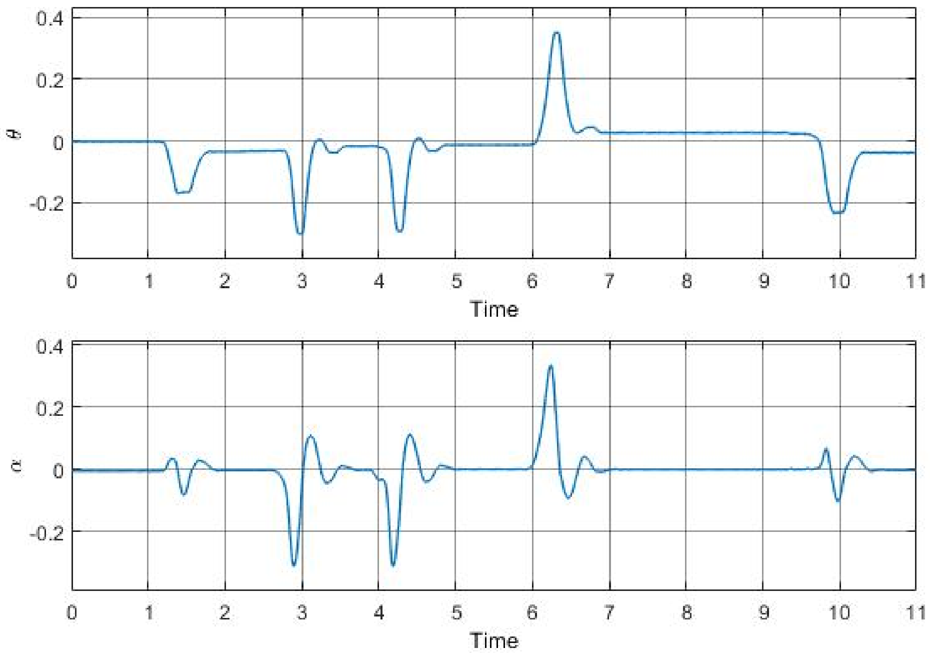


Figure 4.8: Angels during disturbance with the full controller

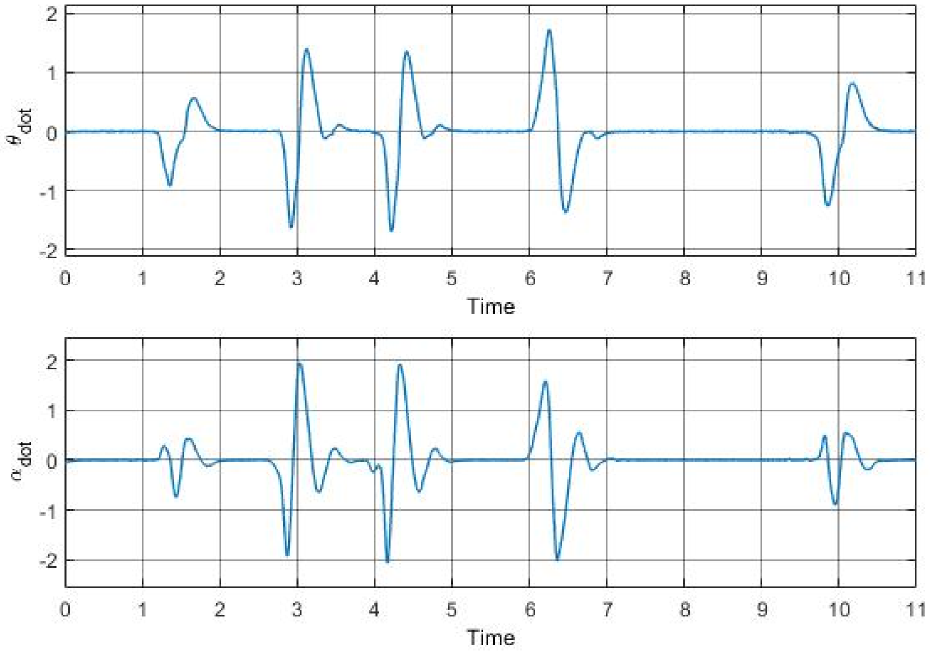


Figure 4.9: Velocity during disturbance with the full controller

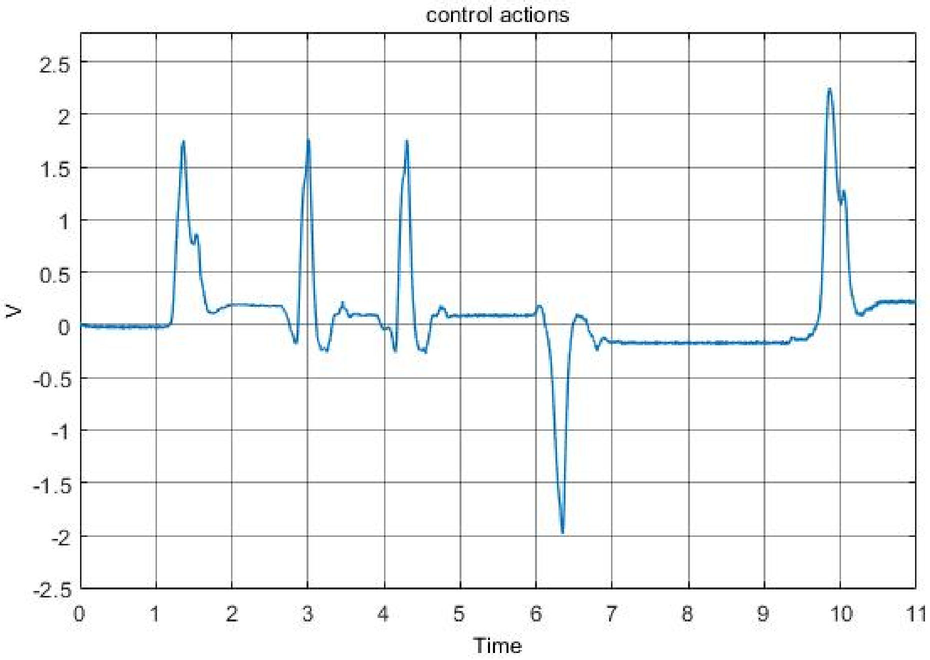
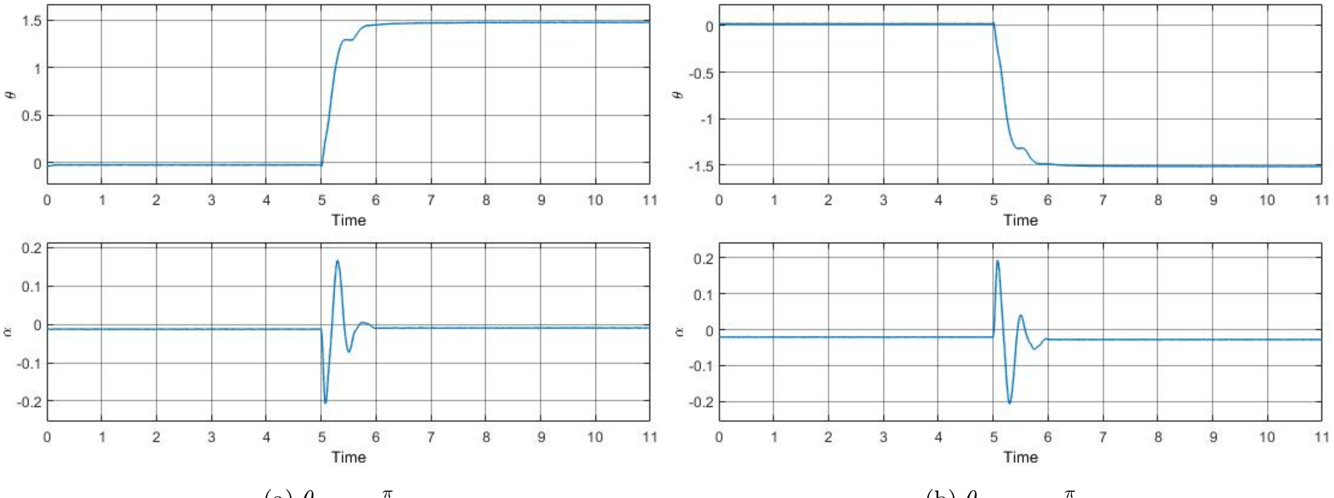


Figure 4.10: Control actions during disturbance with the full controller

4.4 half state feedback

4.4.1 step of $\pm\pi/2$

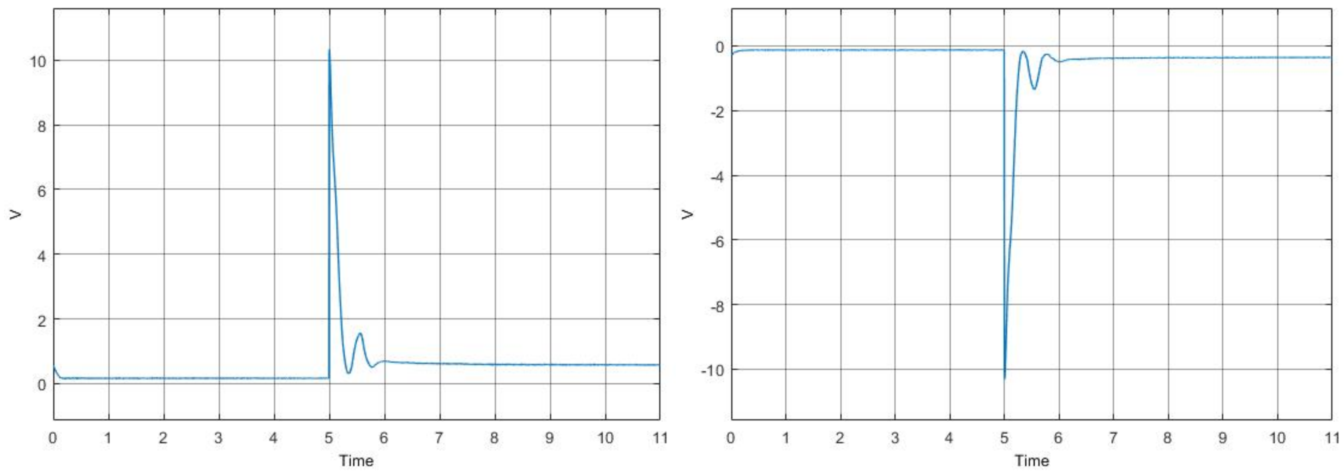
When only half of the state is fed back will this have the result that it will take the system longer to reach the steady state, due to the output α having more oscillations. In figure 4.12 can also be seen that the controller needs higher controller actions contrary to the control actions seen in figure 4.3. As before is the controller not able to fully erase the steady state error, even if half the states are not fed back.



(a) $\theta_{ref} = \frac{\pi}{2}$

(b) $\theta_{ref} = -\frac{\pi}{2}$

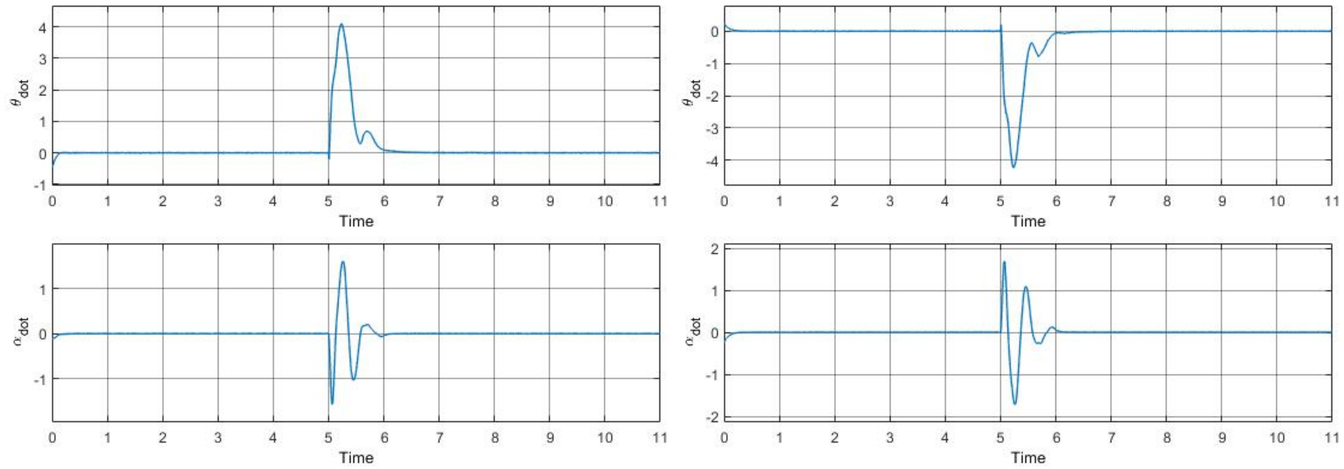
Figure 4.11: θ and α of the half state feedback controller



(a) $\theta_{ref} = \frac{\pi}{2}$

(b) $\theta_{ref} = -\frac{\pi}{2}$

Figure 4.12: Control actions of the half state feedback controller



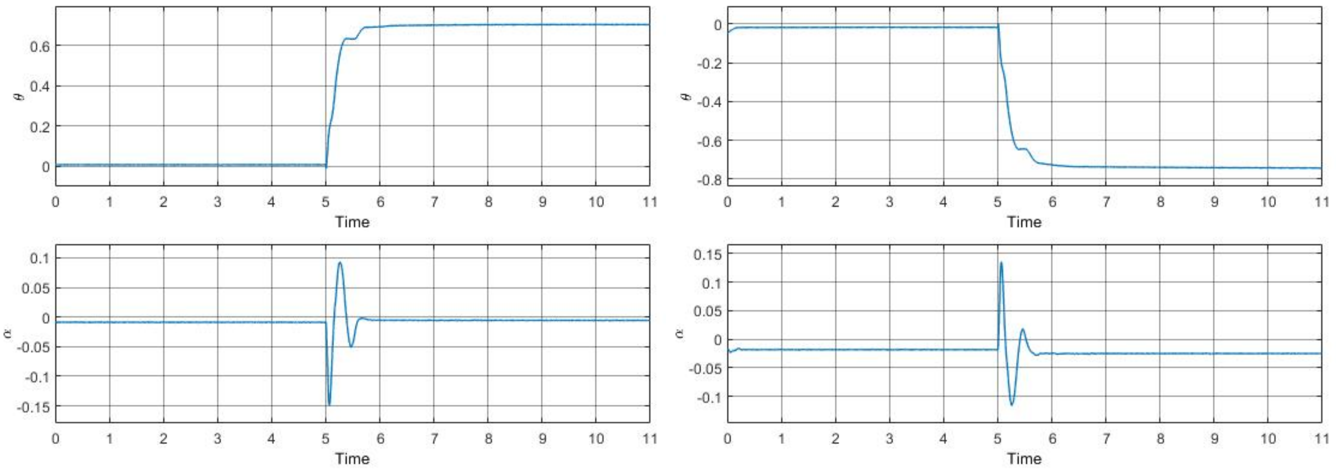
(a) $\theta_{ref} = \frac{\pi}{2}$

(b) $\theta_{ref} = -\frac{\pi}{2}$

Figure 4.13: $\dot{\theta}$ and $\dot{\alpha}$ of the half state feedback controller

4.4.2 step of $\pm\pi/4$

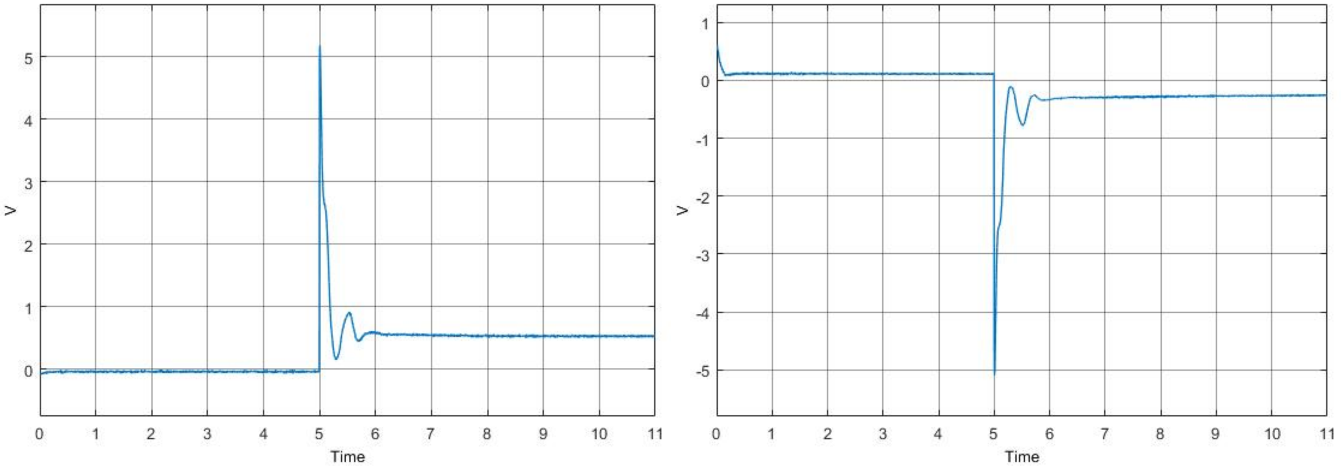
Figure 4.14 show the simulation when a step of $\pm\frac{\pi}{4}$ is applied to the half state feedback controller. As can be seen will the state α in figure 4.14b have a high steady state error, due to the wrong offset given to the system. From the control actions in figure 4.15 can be seen that the controller does not reach its setpoint and thus a steady state error is present. The actions are also higher than in figure 4.6 and will need more oscillations before stagnating due to half of the states not being fed back.



(a) $\theta_{ref} = \frac{\pi}{4}$

(b) $\theta_{ref} = -\frac{\pi}{4}$

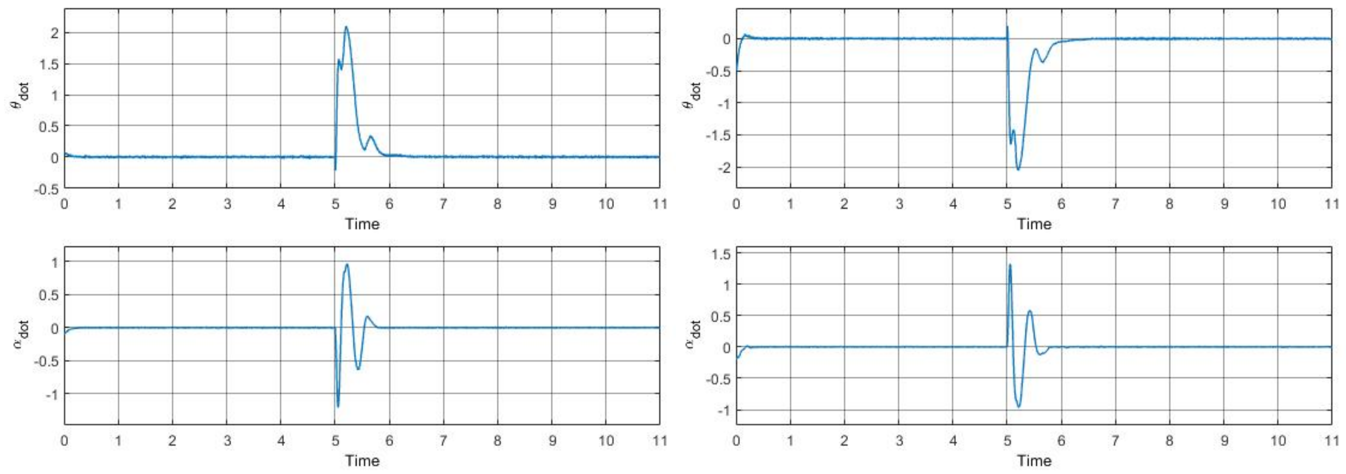
Figure 4.14: θ and α of the half state feedback controller



(a) $\theta_{ref} = \frac{\pi}{4}$

(b) $\theta_{ref} = -\frac{\pi}{4}$

Figure 4.15: Control actions of the half state feedback controller



(a) $\theta_{ref} = \frac{\pi}{4}$

(b) $\theta_{ref} = -\frac{\pi}{4}$

Figure 4.16: $\dot{\theta}$ and $\dot{\alpha}$ of the half state feedback controller

4.4.3 Disturbance rejection

Figures 4.17, 4.18 and 4.19 show the angels velocities and control actions when the system with half of the controller active is disturbed. The controller α angle and its angular velocity component are disabled and it is apparent that the system overshoots with some oscillations. The controller also needs more time to reach the desired state than before. The time to reach steady state is double the time as when the full controller is equipped. This is expected due to the controller having less info about the system states than before.

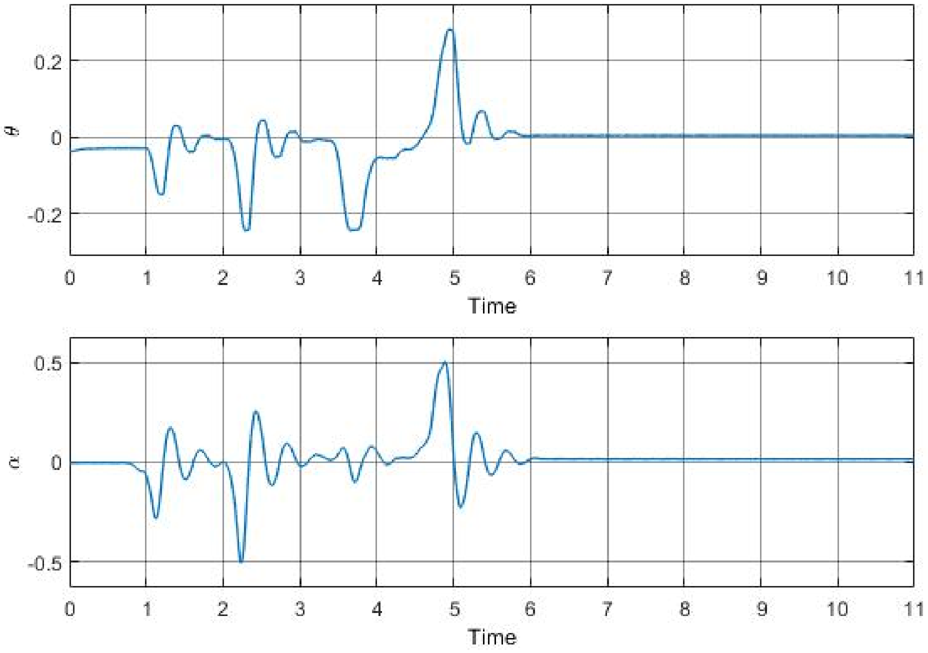


Figure 4.17: Angels during disturbance with half of the controller

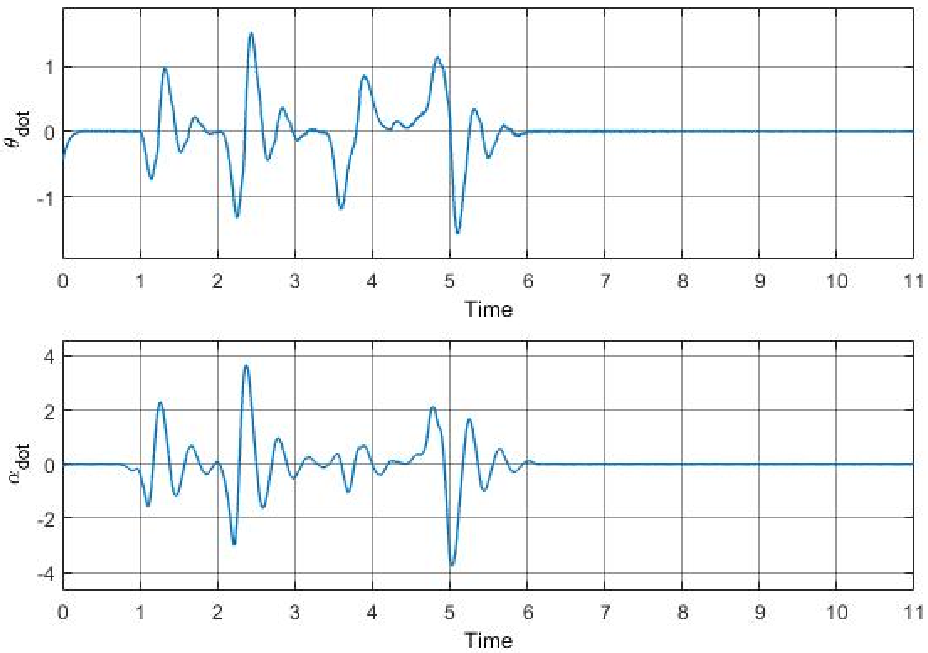


Figure 4.18: Velocity during disturbance with half of the controller

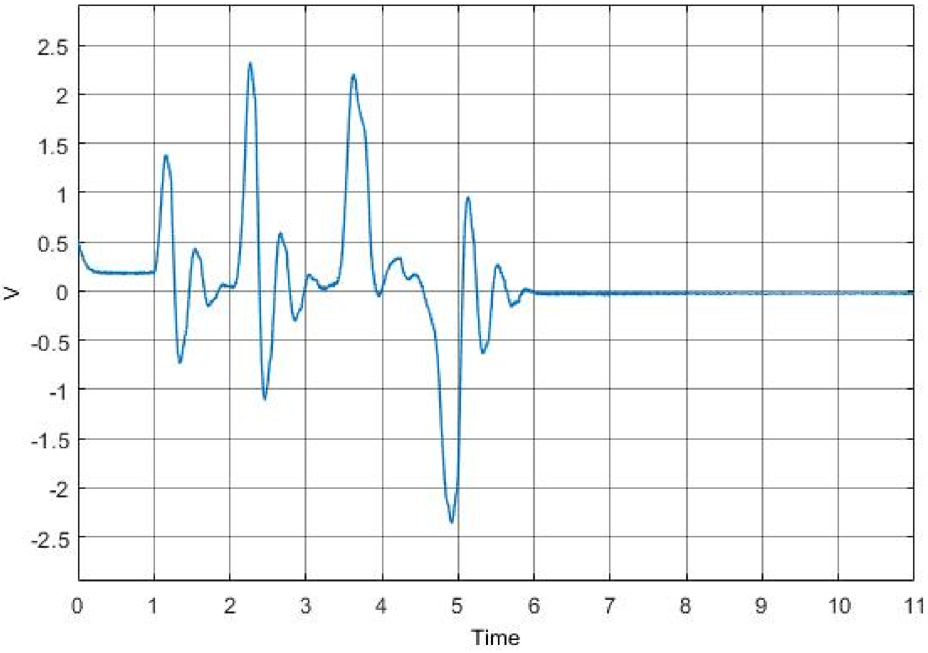


Figure 4.19: Control actions during disturbance with half of controller

4.5 frequency response

The frequency response, seen in figure 4.21 is found by applying a sinewave where the frequency increases from 0.5 Hz to 3 Hz with 0.1 Hz steps. The reference can be seen in figure 4.20. For the lower frequencies is the RFJ able to follow but when the higher frequencies are applied is the RFJ not able to keep up anymore. Meanwhile will the arm itself start to resonate. The system not being able to follow the reference can be partly explained by the bode plot found in figure 4.24. There will a frequency of around 2.5Hz not be passed through and thus suppressed. Another reason is the DC motor not being able to output enough torque. By making the controller more aggressive could this error be reduced but it will introduce overshoots during set point tracking or the control limits are hit and won't improve, i.e. causing the system to go into open loop. From this can be concluded that the frequency response in real-time and in simulation do not have the same kind of behavior. The simulation angular velocity decreases with the frequency until a point the speed is zero and then it starts to speed up again. The real-time frequency response also shows the decreasing speed until a certain point it doesn't decrease anymore unlike the simulation.

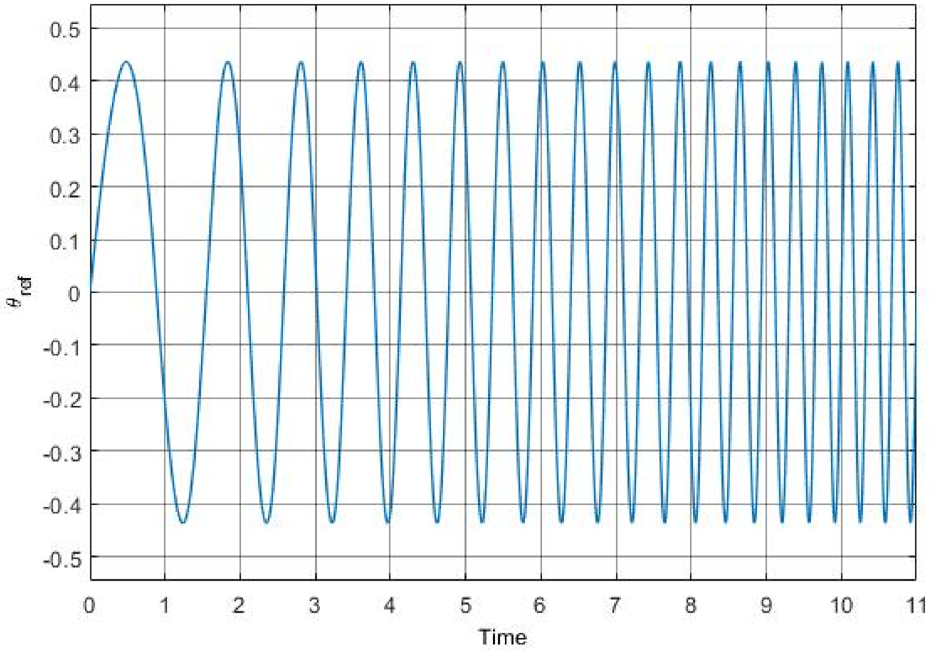


Figure 4.20: Frequency response reference angle

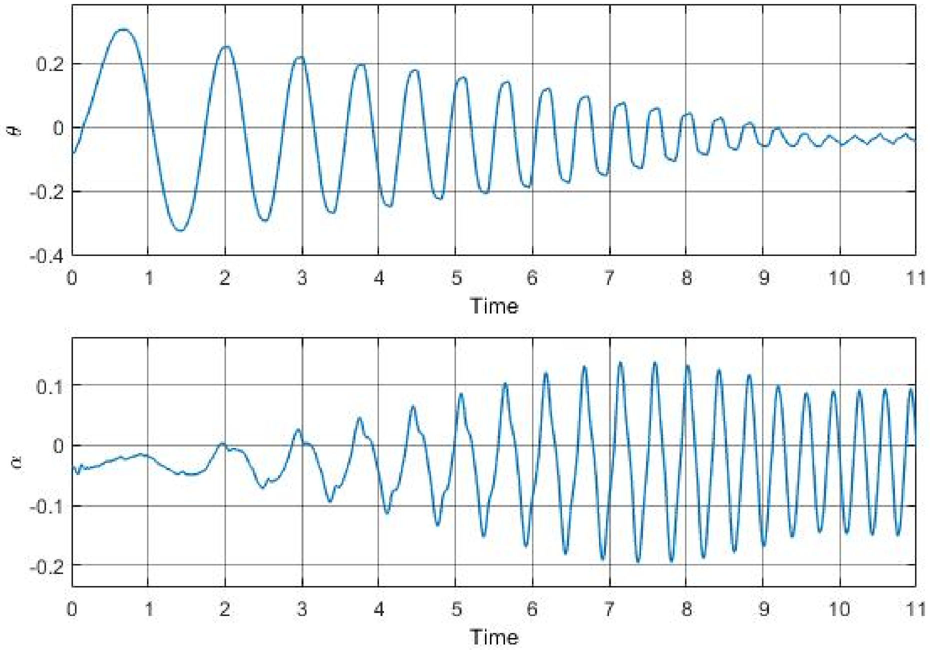


Figure 4.21: Angles during the frequency response

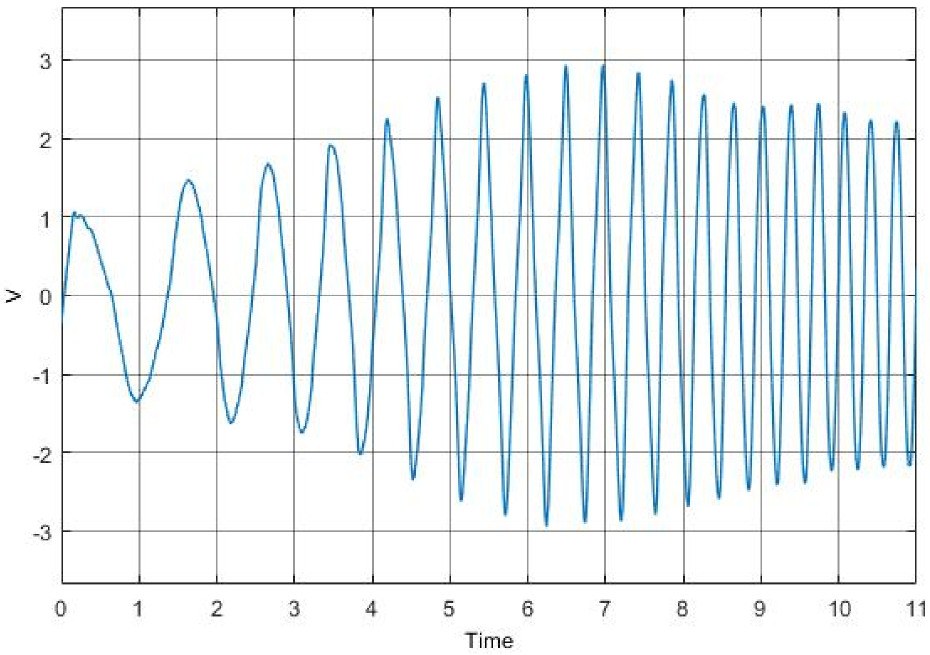


Figure 4.22: Control actions during the frequency response

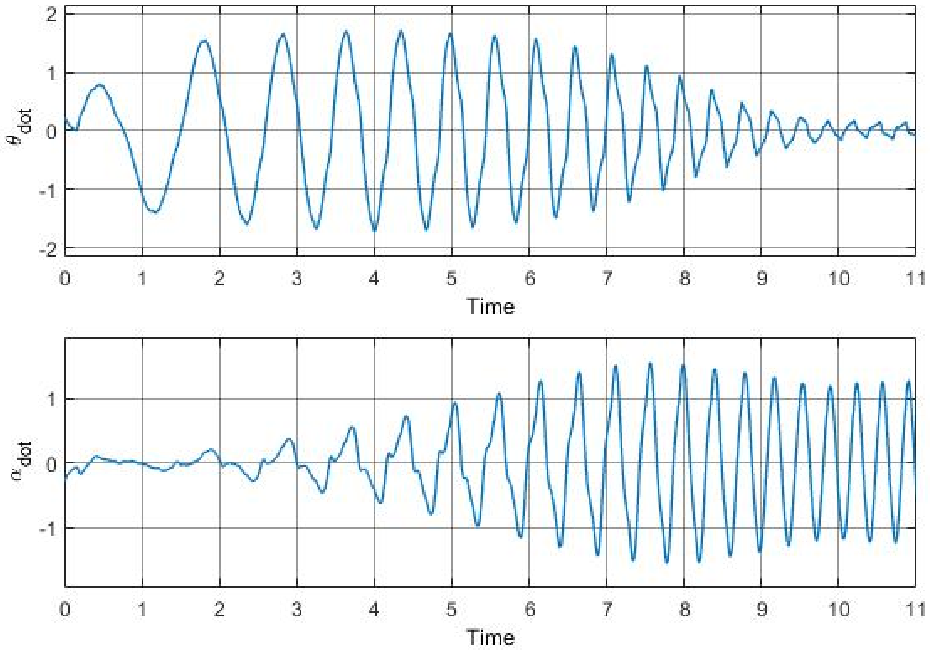


Figure 4.23: Velocities during the frequency response

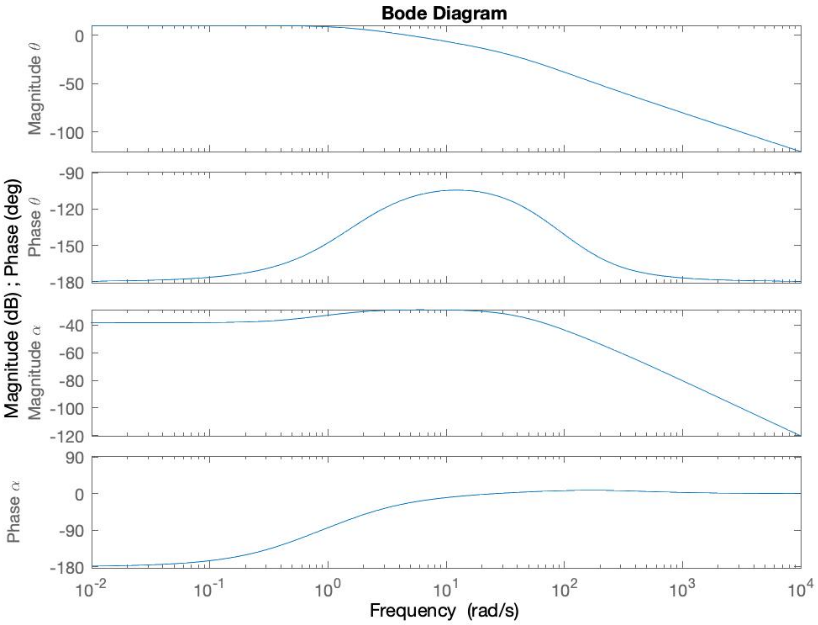


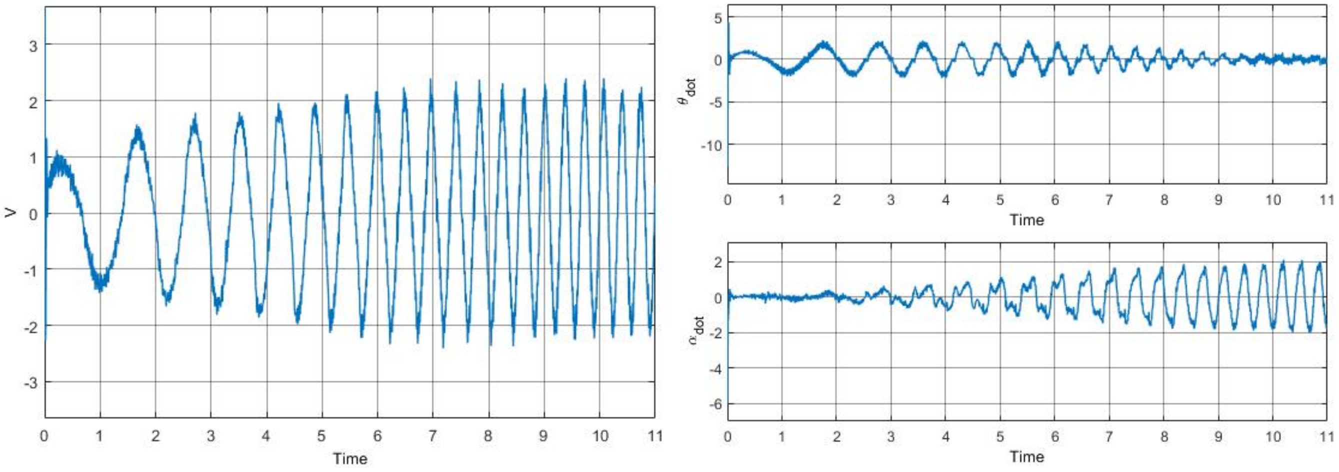
Figure 4.24: Bode diagram of the controller

4.6 Influence of cut-off frequency

As explained in section 3.3 will the cutoff frequency have the effect of passing through frequencies until that point. A high cutoff frequency has the effect of passing through noise while a low frequency has the effect of not passing through every part of the signal.

The frequency response is simulated with a cutoff frequency of 100Hz in figure 4.25. Both the control actions and velocities are more noisy due to the high cutoff frequency. This will lead to the controller to be more nervous by responding to these noises, which is not needed.

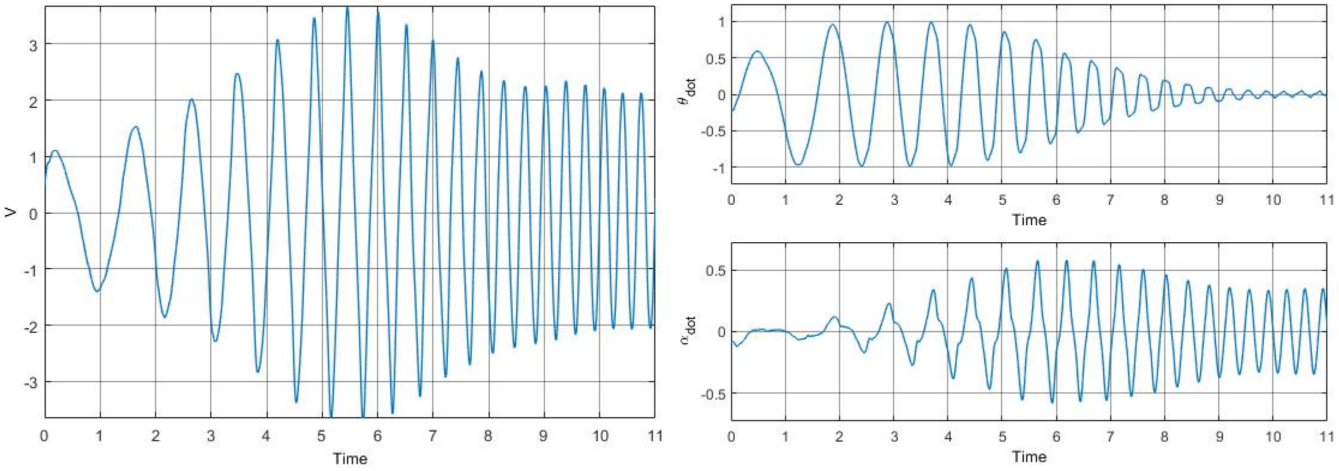
Figure 4.26 shows the case when the cutoff frequency is set to $\omega_c = 0.5Hz$. From the control actions can be seen that the controller needs to take more action in comparison to the control actions in the original controller in figure 4.22. A low cutoff frequency has the effect that the velocities will be smooth but loose information when the frequency becomes high. That is why the controller needs to take more action than before to reach the desired state.



(a) Control actions

(b) Velocities $\dot{\theta}$ and $\dot{\alpha}$

Figure 4.25: System with $\omega_c = 100Hz$



(a) Control actions

(b) Velocities $\dot{\theta}$ and $\dot{\alpha}$

Figure 4.26: System with $\omega_c = 0.5Hz$

Conclusion

This report analysed an optimal controller on a real-life setup of a rotary flexible joint (RFJ). An open-loop analysis, done in chapter 1, gave insight in the dynamics of the system with no controller. The conclusion that could be made was that the system was not stable but stabilizable and detectable.

When designing the controller must the controller thus be able to make the system stable. Chapter 2 showed that the obtained controller was indeed able to make the system stable. Moreover was the controller successful in making the system reach a certain setpoint in a satisfactory time.

The system was then simulated to simulate the real-time setup. To do this was a quantization added to the outputs of the system and was the input saturated to $[-5, 5]V$. When doing a setpoint analysis with this simulation setup, was the controller able to bring the system to each setpoint without a steady state error present. A frequency response analysis was also done which gave a result that could be explained by the bode plot of the closed-loop system. Furthermore, was in section 3.3 the influence of the cutoff frequency analysed. The analysis resulted in higher frequencies letting more noise through, while low frequencies lead in an information loss.

Lastly was everything experimented with the real-life setup in chapter 4. The experiments resulted in most setpoints tracking to have steady state error, which could be explained by the DC motor not having enough torque to overcome the friction. The same problem could be seen in the disturbance rejection tests and the tests with a half state feedback. The controller was although able to overcome the disturbances. The half state feedback test were also successful, but needed more time to reach steady state due to having only information of half of the states and thus causing more oscillations in the response. When doing the frequency response test could the effect of friction also be noted due to the high frequencies not being passed through fully as in the simulation. When doing an analysis of the cutoff frequency could the same conclusion be made as before.

Appendix A

MATLAB code

This appendix explains the function of every file provided with the report.

- *main.m*: the initialization file that provides all the variables to the Simulink files. This is why it needs to be run before all other Simulink files are used.
- *controller_check.slx*: Simulink file that checks the different controllers on a step change of the state θ.
- *controller_simulation.slx*: Simulink file that simulates the real-time setup. It is provided with a quantizer and saturation of the input.
- *real_time_controller.slx*: Simulink file used to obtain the experiments of the actual setup.