ROTARY FLEXIBLE JOINT

MATTHIAS BAETEN AND MORITZ WOLTER

Project Report

Supervised by Prof. Bart de Moor
Mauricio Agudelo
December 28, 2014 – Version 1

1 INTRODUCTION

In this project we tried to design a controller for the Rotary flexible joint. This setup consists of an arm, connected with springs and a joint to a hub, which can rotate in a horizontal plane, driven by a motor. To get an extensive view of the set up we refer to the assignment [**?**, assignment]chapter2.1) for this project. The controller must let the arm track a particular input-function (e.g. a step, or a sine wave) as good as possible. The measured signals are the angle of the hub and the angle of the arm.

The controller we build is based on model-based control system design. So we had to find a state-space model of the system. To do this a lot of mechanical and electrical study has to be done. For a extensive study of the mechanical and electrical part of the Rotary flexible joint we refer again to the assignment [**?**, assignment]chapter2.1). In the assignment the following state-space model is derived:

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

The linear state equation can be written in the standard form $\dot{\mathbf{x}} = A\mathbf{x} + B\mathbf{u}$ with our input $\mathbf{u} = V$ and our states $\mathbf{x} = [\theta, \alpha, \dot{\theta}, \dot{\alpha}]^T$. The output equation has the following standard form $\mathbf{y} = C\mathbf{x}$ with $\mathbf{y} = [\theta, \alpha]^T$. The numerical values for the parameters that we used are stated in the assignment [**?**, assignment]chapter6). The numerical values of the matrices A, B, C and D are stated below.

$$A = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 765.9810 & -52.7952 & 0 \\ 0 & -1038.618 & 52.7952 & 0 \end{pmatrix} \quad B = \begin{pmatrix} 0 \\ 0 \\ 98.3333 \\ -98.3333 \end{pmatrix} \quad C = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \quad (2)$$

We performed a short open loop analysis of the system. The poles of the open loop system are $0, -8.9761 + 18.2356i, -8.9761 - 18.2356i$ and -34.8429. Because the poles are all in complex left half plane, we can say that our open loop system is stable. Therefore our system is also stabilizable. The open loop system has no transmission zeros. The ranks of the controllability and observability matrices both equals

4 where 4 is the number of states. Hence our system is controllable and observable. From this only we can already conclude that the system is minimal. When we checked it in Matlab, it was indeed so.

The control goals for our controller are to accurately track the setpoints, to give a fast response, to give a good disturbance rejection. Of course the closed loop system should also be stable.

2 SIMULATION

2.1 LQR CONTROL WITH KNOWLEDGE OF THE FULL STATE VECTOR

Figure 1 shows our initial simulink diagram. The most important block in the control gain, that does the actual LQR-Control. The state space block in the center contains the continuous time system we showed in the previous section. In the left we have a selector that we use to choose between either a step or a sinusodial input. Below it we use a block of constant zeros to full up remaining entries of the input vector. As stated earlier we are using LQR-Control in this project. To compute the gain matrix for an LQR-Controller determines an state-feedback gain which minimizes the following cost function:

$$J = \int (\mathbf{x}^T Q \mathbf{x} + \mathbf{u}^T R \mathbf{u}) dt. \tag{3}$$

Q = diag([350 1500 3 0.5]) for R = 10.

Initial but not optimal Q and R where given. The Q matrix gives weight to the different elements of the state vector. In our case hub position, arm position, angular hub and angular arm velocity. The R matrix, in our case a scalar makes the input more expensive. Our goals for tuning the controller are to achieve a satisfactory tracking for the hub angle θ without overshoot, while keeping the movement of the arm α as low as possible. At the same time we want to avoid actuator saturation. These are obviously conflicting objectives. So a good compromise is required. Figure 2 shows the performance of the initially provided weighting parameters. From the tracking point of view the performance is not bad, however the input signal almost exceeds 5 Volts, which will might lead to actuator saturation if applied to the real plant. We chose the following weighting matrix to fix this problem:

$$\begin{pmatrix} 500 & & & \\ & 1000 & & \\ & & 3 & \\ & & & 10 \end{pmatrix} \tag{4}$$

With the input weight $R = 30$. We increased the weight on the inputs to prevent saturation. As a higher input weight will make high inputs more expensive we expected this to fix our saturation problem. Furthermore it turns out more effective to penalize the acceleration of $\dot{\alpha}$ in comparison to a high value for α. This is the case because we

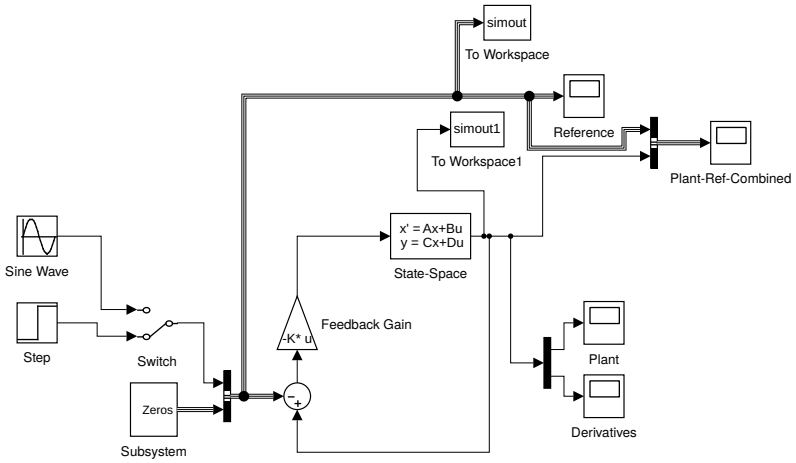

Figure 1: The simulink Layout of our Simulation knowledge of all the states is assumed

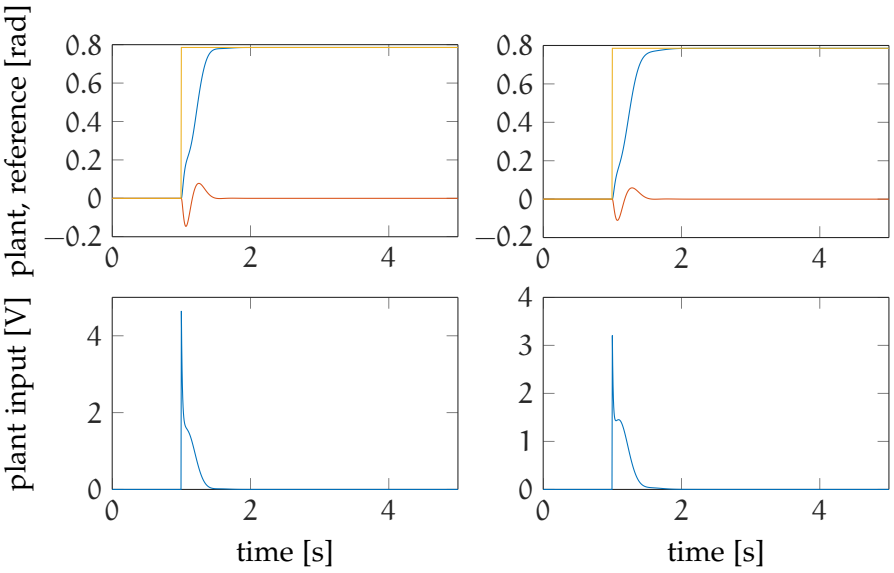

Figure 2: Simulation results for the original weighting parameters left and for the improved weighting parameters right.

know that the reference for α will always be zero. Putting a higher weight on arm-acceleration prevents the arm from moving too fast away from its zero reference. In figure 2 on the right side we show the simulated behavior with our adapted weights, where the tracking remains comparably fast.

2.2 CONCLUDING SIMULATIONS WITH FILTER AND DIFFERENTIATOR

In order to have a more realistic simulation, you have to create a Simulink diagram where the outputs of the linear model are the variables that are measured in the real setup (θ and α). So from now on we change back to the state-space model with the matrix A,B,C and D described in chapter one. Because we want to apply state feedback, we need the know the state vector. Therefore in order to derive the entire state vector, we will compute the derivatives of the measured outputs by using the backward difference equation and a low pass filter. To use the backward difference technique, it's necessary to low-pass filter the raw measurements before calculating the difference, because these are very fluctuation- and noise-sensitive. The low pass filter we will use obeys the following law:

$$y_n^f = \frac{\omega_c T_s}{1+\omega_c T_s} y_n + \frac{1}{1+\omega_c T_s} y_{n-1}^f \tag{5}$$

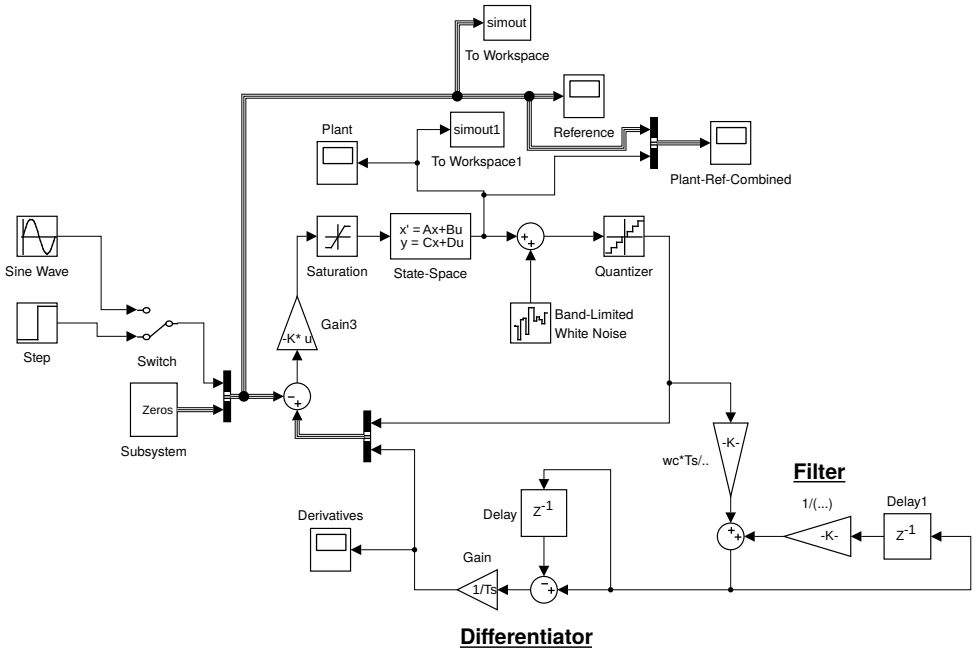

Figure 3: The simulink Layout of our Simulation

where we take the cut-off frequency $\omega_c = 4\pi$ rad/s. y_n^f is the filtered signal. The states $\dot{\theta}$ and $\dot{\alpha}$ are then calculated using the following backward difference equation:

$$\dot{y} = \frac{y^f(t) - y^f(t - T_s)}{T_s} \tag{6}$$

in Figure 3 we see the Simulink diagram where differentiator and filter are added. They are implemented in such a way that the equalities (5) and (6) are fulfilled. We added a saturation block such that the input voltage is limited between -5V and 5V. Also a quantizer is added with a quantization interval of length $1e-6$. To approximate the reality as much as possible, we also included a white noise source with a noise power of 10^{-4}. In figure (4) we applied the same step change as in figure 2 but simulated with the simulink diagram from figure 3. We see that the results are similar but the effects of the noise source are visible. This is the way it will always be in reality.

We will now investigate the role of the cut-off frequency of the low-pass filters. In figure (5) we see the step response as before but now we increased the cut-off frequency to $\omega_c = 10\pi$. We notice that when the cut-off frequency is to large, to much high frequency noise goes into the system are hence we get really noisy results. We will

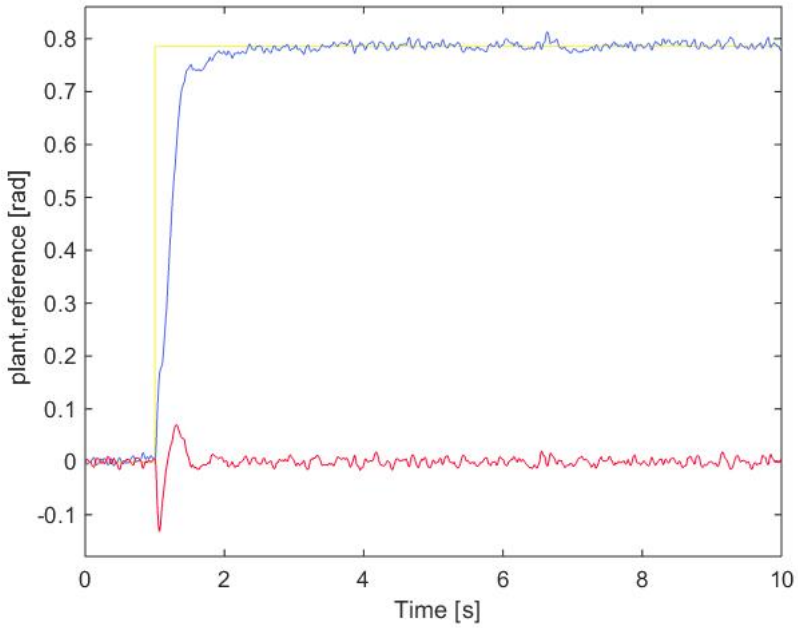

Figure 4: Simulation results with the simulink diagram from Figure 3. The yellow line is the reference signal for θ, the blue line is the value of θ and the red line is de value of α.

now investigate the effect when we decrease the cut-off frequency. In figure (6) we see the step response with a cut-off frequency $\omega_c = \pi$. In this case we see that we get overshoot and slow convergence to the step reference. Although there is no noise, we also don't want overshoot and slow convergence. So we see that it is important to pick the right cut-off frequency.

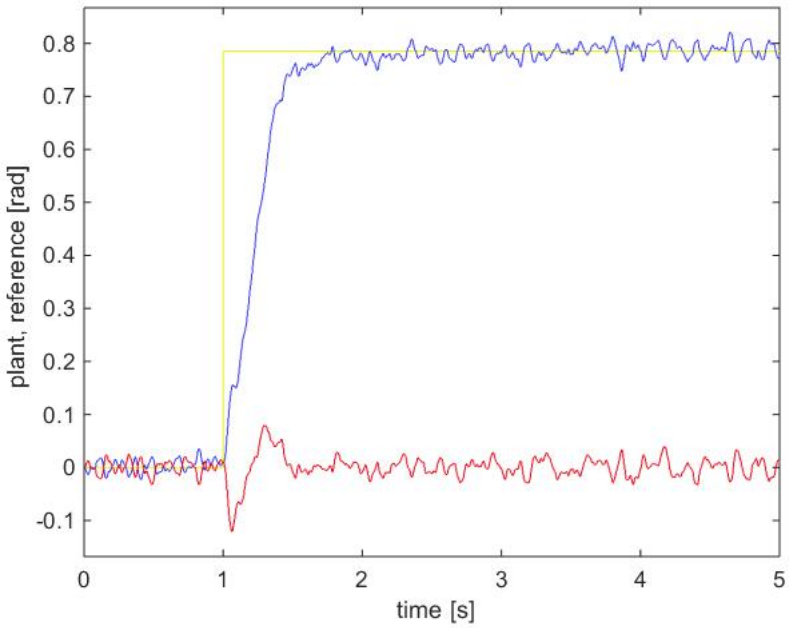

Figure 5: Simulation results with the simulink diagram from Figure 3. The yellow line is the reference signal for θ, the blue line is the value of θ and the red line is de value of α. The cut-off frequency of the low pass filter is set to $\omega_c = 10\pi$.

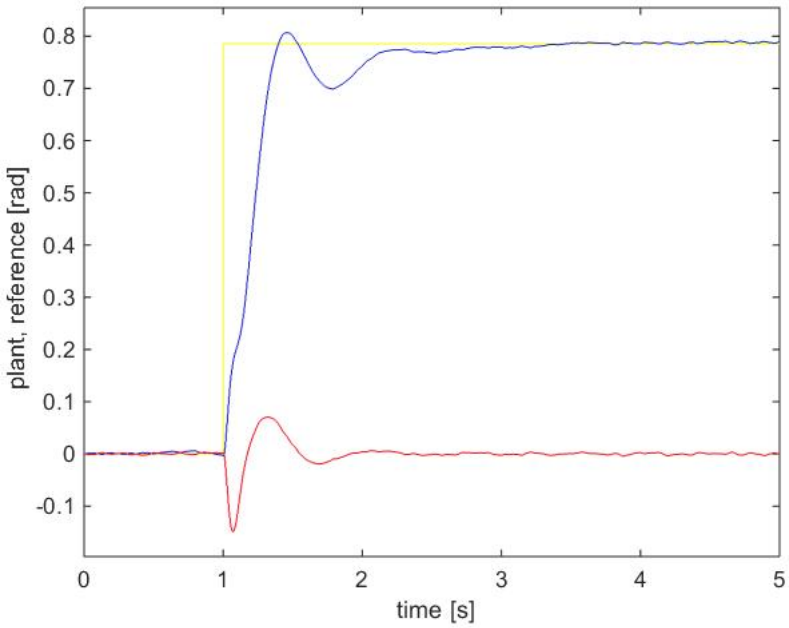

Figure 6: Simulation results with the simulink diagram from Figure 3. The yellow line is the reference signal for θ, the blue line is the value of θ and the red line is de value of α. The cut-off frequency of the low pass filter is set to $\omega_c = \pi$.

3 TESTS OF THE CONTROLLER ON THE REAL SETUP

In figure (??) we see the real-time Simulink diagram that we used in the lab to run our experiments on the real setup. The "Analog Input" block sends measurements of the sensors into the template. Each sensor reading is updated with a sampling time of 5 ms, in radians for the angles of the hub and arm. Additionally this block has two special outputs, "Display[volts]" and "Display[Eng]", which are used for visualization purposes. These two outputs are connected to a set of displays in order to provide a local visual measure of the process variables.

The "Analog Output" block allows the controller to send a voltage signal to the motor of the hub. At the input of this block there is an on-off switch. This switch allows the user to switch the motor on or off. We included a saturator in our diagram just before the signal goes in to the "Analog Output" to limit the control input within $-5V$ and $5V$. The saturation settings of the block in our diagram are set accordingly. We also included a sine wave, a step input and a zero input. These two block are used as a reference for $\theta_{desired}$. The zero block contains 3 signals equal to zero that represent the references for $\alpha, \dot{\theta}$ and $\dot{\alpha}$.

3.1 SETPOINT TRACKING

Using the controller we tuned during the simulation of our plant we started our experiments with some simple reference tracking experiments. The results are shown in figure 8. In comparison to simulation 1 results are as expected. The arm angle α oscillates a little less then expected. We observe a steady state offset that we did not see in our simulations, however this offset is within one or two degrees. Given that we have a small systematic error due to the potentiometer calibration that we did not fix completely this offset seems reasonable. We also applied a sinusoidal reference input. Tracking results are shown in figure 1 on the right side. In comparison to the reference sine, which has an amplitude of 45° our system output is damped. However in the beginning we stated that apart from good tracking results we also want our controller to keep the arm angle α as close to zero as possible. Again we are facing a conflicting control objectives. We cannot perfectly track the hub angle θ and keep the arm angle α close to zero. In our experiment the arm angle α remained within 2 and -2 degrees at all times. If we tuned more towards tracking we

Recall that we had the tuned controller: Q = diag([500 1000 3 10]) with R = 30.

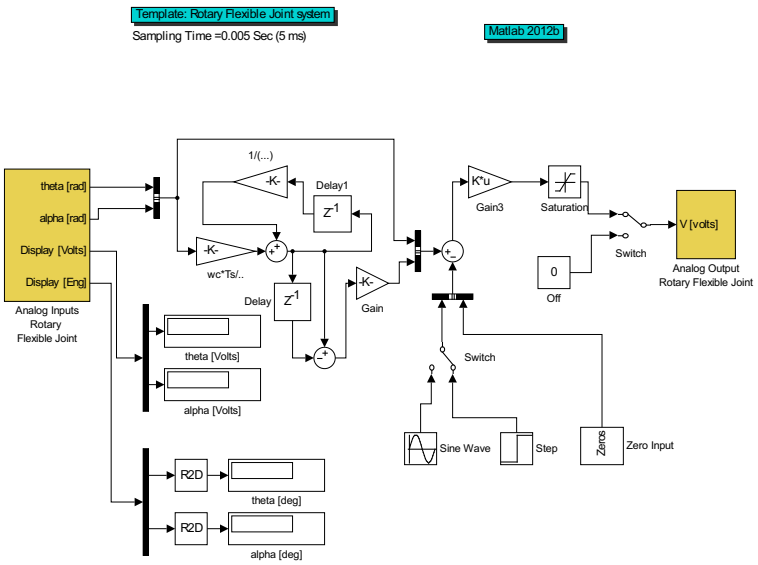

Figure 7: Real-time Simulink diagram

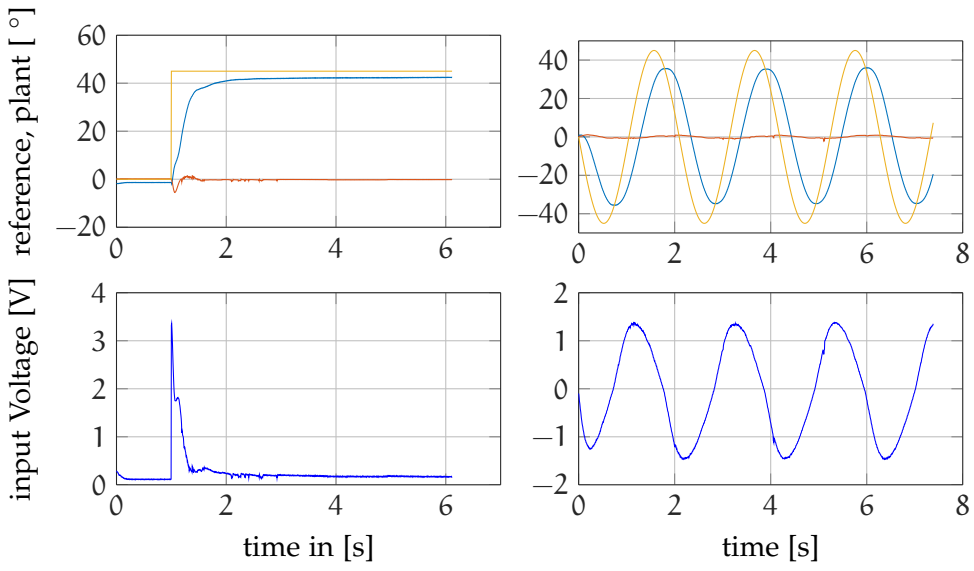

Figure 8: Setpoint tracking test results.

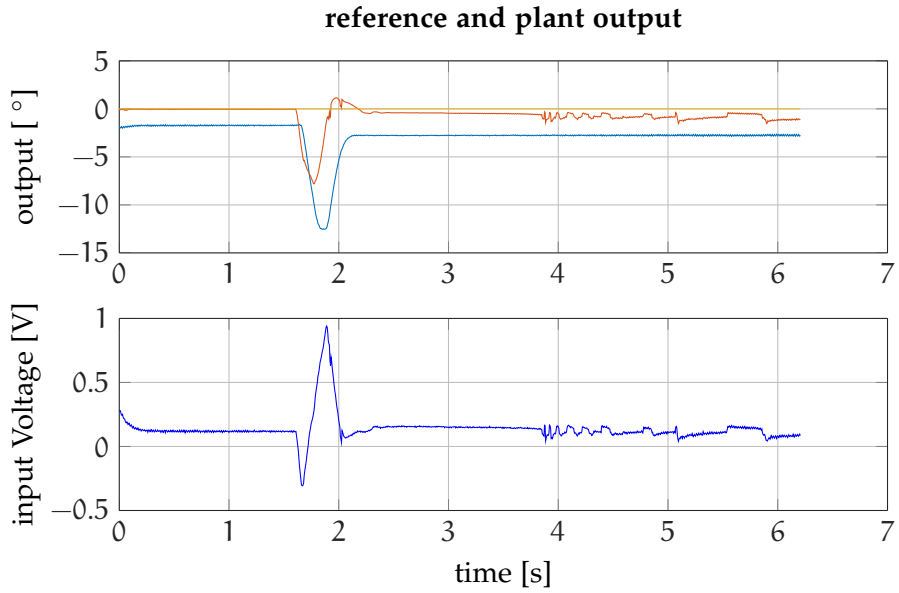

Figure 9: Plant and controller reaction to an external disturbance.

would loose performance here. So we are proposing this controller as a compromise.

3.2 DISTURBANCE REJECTION

Next we are going to describe the results of some disturbance rejection tests. During these test we applied a small kick of $\approx 10°$. Results are shown in figure 9. After the disturbance the plant returns to its original position, with a small offset of about one degree. The input voltages remain withing the required bounds. It is important to note, that the reference remains zero at all times, since we are applying an external disturbance.

3.3 THE EFFECT OF REMOVED ARM ANGLE FEEDBACK

We proceeded by removing the arm angle feedback and measured the effect it would have on the control results using the adapted controller from the simulation. Our measurements are shown in figure 10. With arm angle feedback removed, the controller has no information about the position and acceleration of the plant's arm. It therefore does far worse in regulation it in comparison to a setting where that information is available.

3.4 EXPERIMENTAL CLOSED LOOP FREQUENCY RESPSONSE

In this section we are going to present our measurement of the closed loop frequency response. To begin our measurements we placed a

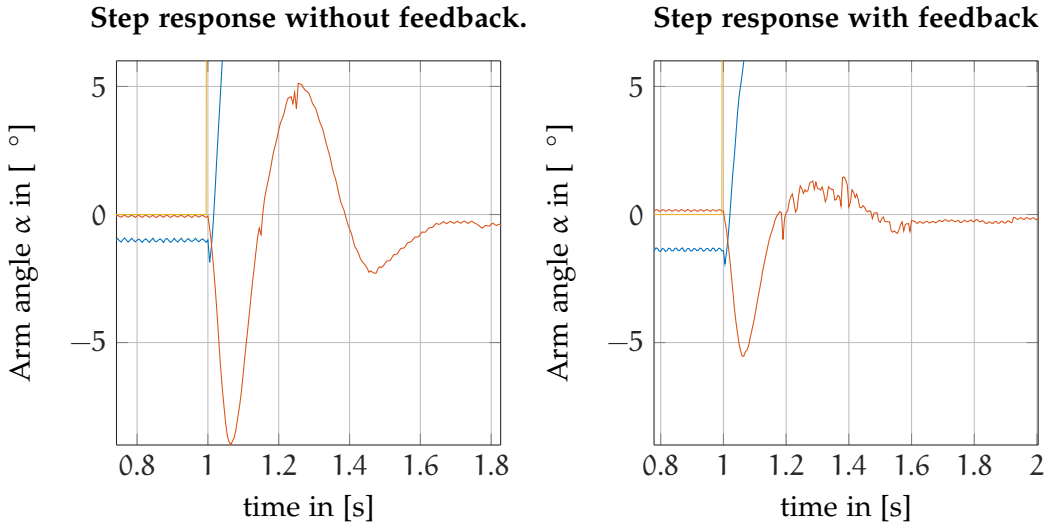

Figure 10: Plant step response with and without arm angle feedback.

"form workspace-input block" as reference in our template. This input was followed by a gain block which multiplies the input by `pi/8`. Before compilation we generated a sine wave with an increasing frequency as a function of time. Using the matlab command: `chirp(time,f0,t1,f1,'logarit` We set the measurement time to 100 seconds. We started at initial frequency $f_0 = 10^{-2}$ and stopped at $f_1 = 25$. By setting the `'logarithmic'` option of the chirp command the input waves frequencies will be placed at a logarithmic scale throughout the time available for measurement. The top row of Figure 11 shows the reference signal Voltage we applied. Below we see the response of the plant over time. In order to compute the bode plot representation we fitted transfer functions to the measured data. With four poles and three zeros we got 90.85% fit for the transfer function corresponding to the hub angle θ. The same parameters give us a 73.22% fit for the transfer function estimation that models the arm angle α. However looking at the plots in 11 where the measured response is shown together with the simulated model outputs to the same input signal. We observe, that our models capture the physically relevant behavior. The magnitudes of simulated and measured plant angles correspond very nicely. Leading us to the assumption that at least the magnitude part of the bode plot will be accurate. Model mismatches probably stem from failure to predict the phase lag properly, especially at high frequencies, where the plant completely fails to track the input signal.

The bode plots we computed from the identified transfer functions are provided in 12. The magnitude plot depicts very accurately what we saw when measuring the frequency response [1]. At low frequencies

1 A video of the plant during this measurement is also available at: http://youtu.be/b4xiNnIXjxU. When watching the video keep in mind that the frequency is a function of time. The video shows a bode plot in the making.

the hub can track the reference signal very accurately, the magnitude plot is basically zero. This situation changes dramatically for higher frequencies when it becomes harder to track the signal. Now the hub fails to reach the peak of the sine input in time and has to start moving into the reverse direction early. This leads to a decrease in the magnitude of the output wave, which is depicted in the bode plot by the falling magnitude line for higher frequencies. However theta has one resonance frequency during the measurement we could actually hear the plant oscillating. This is why the transfer function has a complex pole pair at approximately 10 Herz. When looking at the bode plot for alpha it is important to note, that this is the plot of the transfer function from the theta input to the alpha output. It is important to choose this function, as the alpha reference is zero at all times. During low frequency excitation the arm does not move when observed from the hub, leading to a zero output. This desired good tracking performance translates into extreme damping in relation to the theta input signal. In the magnitude plot of figure 12 this explains the very low values for the orange line at low frequencies. However with increasing frequencies we also rising oscillations of the arm in relation to the hub. In the time domain plot in figure 11 we observe to resonance peaks at high frequencies, with arm oscillation in between. In the bode plot the orange line rises in the area corresponding to the sine frequency at that time. The quality of the phase plot is not as good as the magnitude plot. What happens in reality for both plant output angles is that they fall behind the input farther and farther for higher frequencies as the plant fails to keep up with the input. For the blue line the phase angle plot works up until 10 Herz. The plot remains constant in the beginning as the plant tracks the input properly. And it falls when we are reaching higher frequencies. However frequencies higher then 25 Herz where not applied during our experiment. From the bode plot we would predict, that when the plant damps the input waves it becomes able to track them again at a constant delay. However this prediction remains speculative.

3.5 FILTER EFFECT

The first thing we noticed when removing the filter, was the we could hear the noise, as the engine exited be the high frequency noise started to cause the system to oscillate. The input voltages with and without low pass filtering are given in figure 13. We observe that with the filter removed high frequency noise makes it into the input. If the system was left unfiltered over a long time this would certainly lead to material exhaustion.

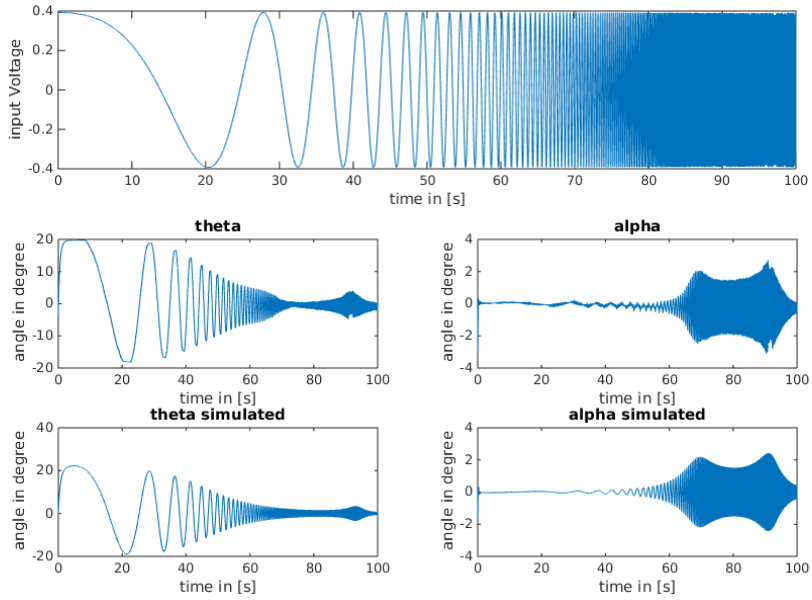

Figure 11: The input signal we used is shown in the top row plot. In the middle row we show the measured plant response to the top input signal. In the bottom the simulated output values are shown.

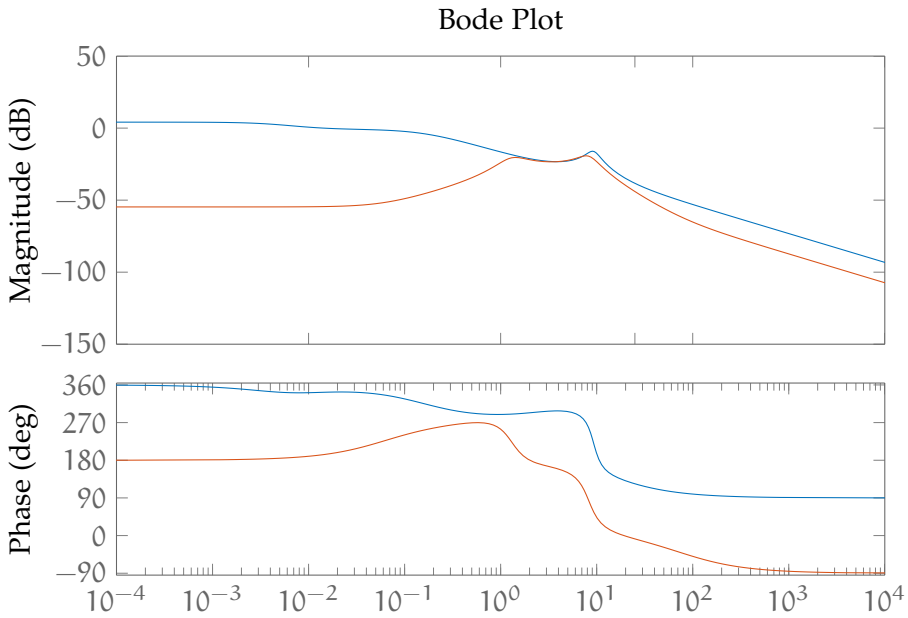

Figure 12: Bode Plot representation of our closed loop system. The blue lines show the frequency response of θ the orange lines refer to α.

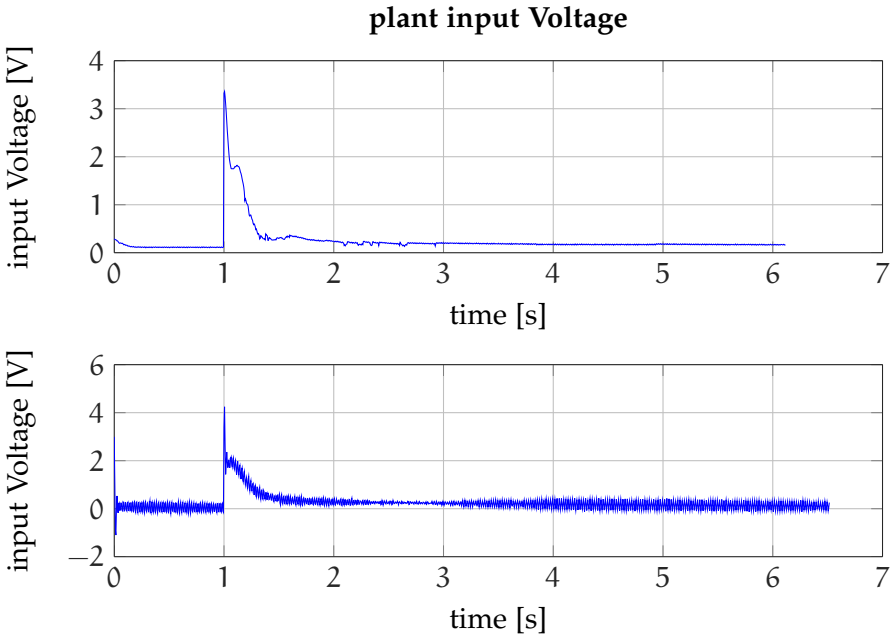

Figure 13: Plant input without (top) and with low pass filtering (bottom).

4

CONCLUSION

We conclude from this project that our controllers work good enough to control the rotary flexible joint in reality. But we have also noticed that the our controllers are not the same in simulation as in a lab. If you need to design a controller for real applications, you have to take into account so much more things such as saturation, quantization, noise, ... We also realized that we can use a continuous model even when in reality it will be executed in a discrete way. As long as the sampling time is small enough it wont cause any problems.