

# ROTARY FLEXIBLE JOINT

MATTHIAS BAETEN AND MORITZ WOLTER



Project Report

Supervised by Prof. Bart de Moor

Mauricio Agudelo

December 23, 2014 – Version 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:

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

The linear state equation can be written in the standard form  $\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}$  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  $\mathbf{y} = \mathbf{C}\mathbf{x}$  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 set-points, to give a fast response, to give a good disturbance rejection. Of course the closed loop system should also be stable.

## 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 sinusoidal 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:

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

Initial but not optimal  $\mathbf{Q}$  and  $\mathbf{R}$  where given. The  $\mathbf{Q}$  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  $\mathbf{R}$  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  $\theta$  without overshoot, while keeping the movement of the arm  $\alpha$  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:

$$\mathbf{Q} = \text{diag}([350 \ 1500 \ 3 \ 0.5]) \text{ for } \mathbf{R} = 10.$$

$$\begin{pmatrix} 500 & & & \\ & 1000 & & \\ & & 3 & \\ & & & 10 \end{pmatrix} \quad (4)$$

With the input weight  $\mathbf{R} = 30$ . 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  $\alpha$  in comparison to a high value for  $\alpha$ . This is the case because we

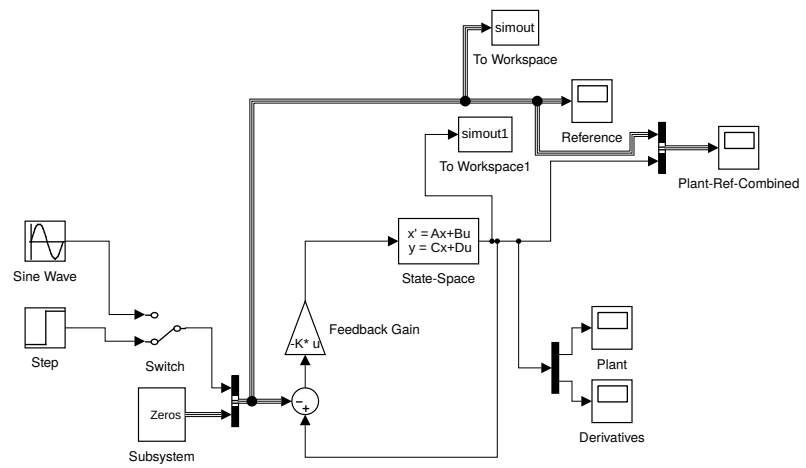


Figure 1: The simulink Layout of our Simulation knowledge of all the states is assumed

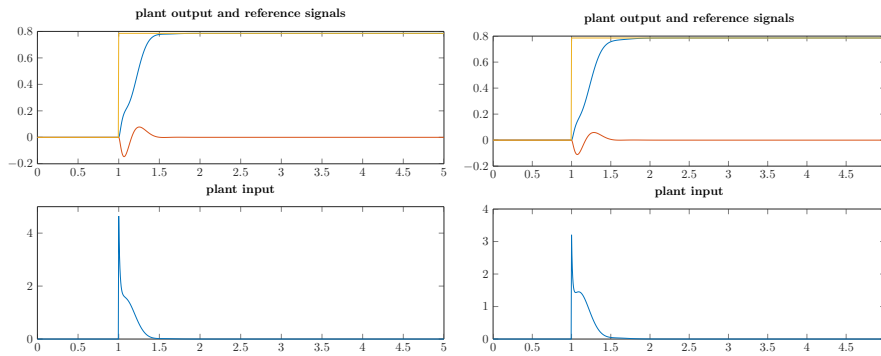


Figure 2: Simulation results for the original weighting parameters left and for the improved weighting parameters right. Plant input is given in Volt [V]. Plant output and references are given in radians [rad].

know that the reference for  $\alpha$  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

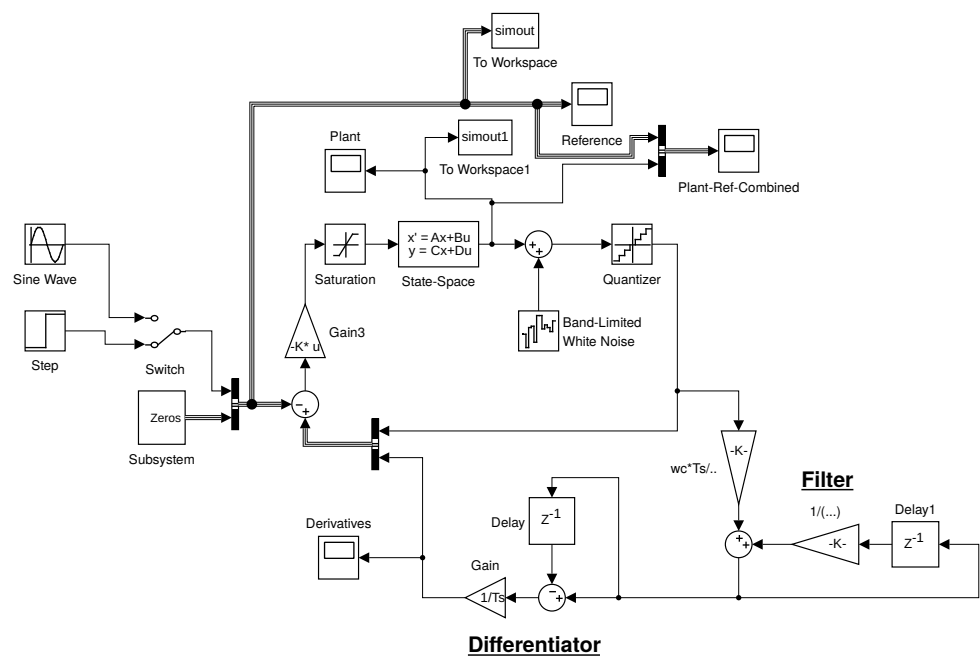


Figure 3: The simulink Layout of our Simulation

## 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  $\ddot{\alpha}$ .

### 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 5. In comparison to simulation 1 results are as expected. The arm angle  $\alpha$  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  $45^\circ$  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  $\alpha$  as close to zero as possible. Again we are facing a conflicting control objectives. We cannot perfectly track the hub angle  $\theta$  and keep the arm angle  $\alpha$  close to zero. In our experiment the arm  $\alpha$  remained within 2 and  $-2$  degrees at all times. If we tuned more towards tracking we

*Recall that we had the tuned controller:  
 $Q = \text{diag}([1500 \ 1000 \ 3 \ 10])$  with  $R = 30$ .*



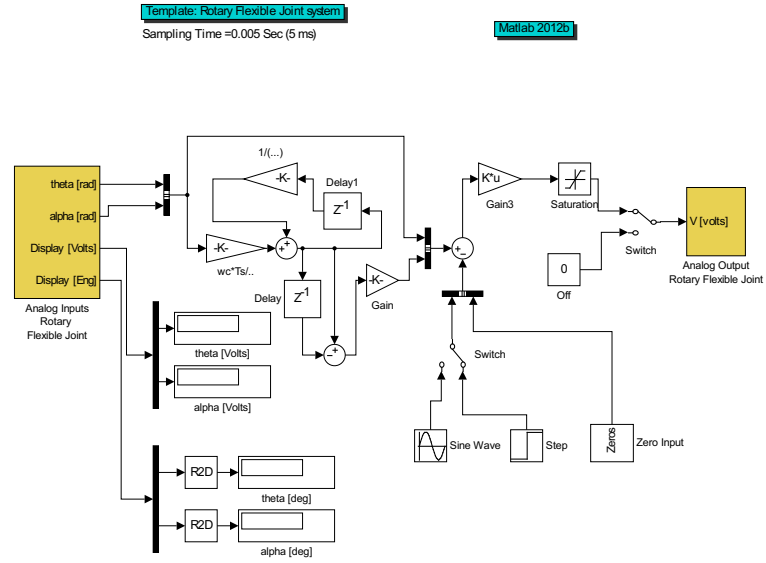
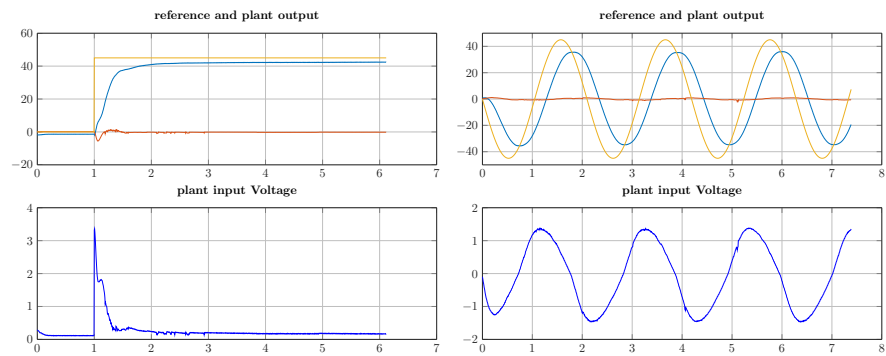


Figure 4: Real-time Simulink diagram

Figure 5: Setpoint tracking test results. References and plant outputs are given in degree  $^{\circ}$ . Plant input voltages are given in volts [V].

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