ROTARY FLEXIBLE JOINT

MATTHIAS BAETEN AND MORITZ WOLTER

Project Report

Supervised by Prof. Bart de Moor
Mauricio Agudelo
December 18, 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:

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

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 exceeds 5 Volts, which will lead to actuator saturation if applied to the real plant. We chose the following weighting matrix to fix this problem:

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

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

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 know that the reference for α will always be zero. Putting a higher

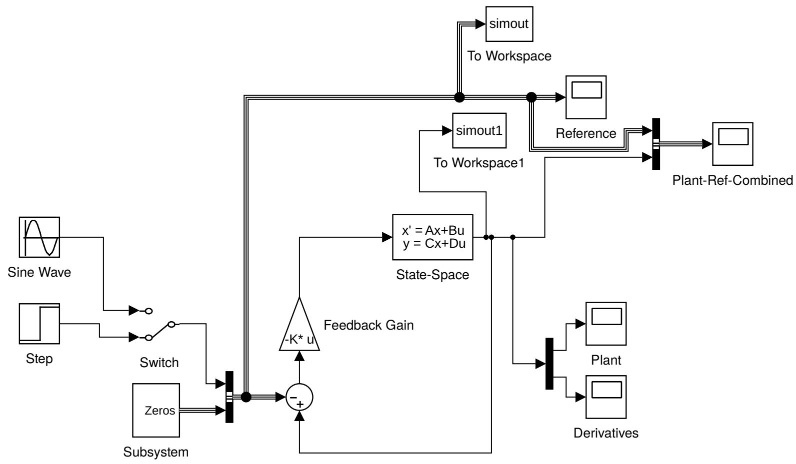

Figure 1: The simulink Layout of our Simulation knowledge of all the states is assumed

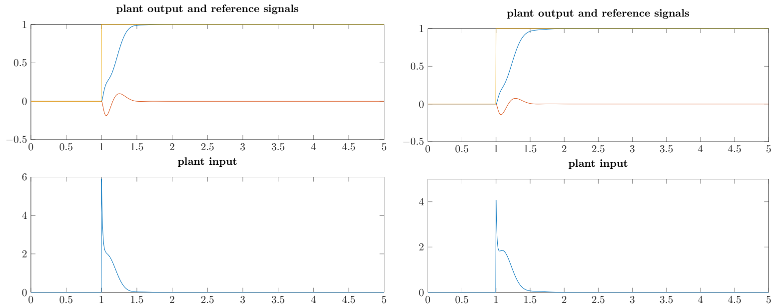

Figure 2: Simulation results for the original weighting parameters left and for the improved weighting parameters right.

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

2.3 TESTS OF THE CONTROLLER ON THE REAL SETUP

In figure (**??**) we see the real-time Simulink diagram that we used in the lab for the real setup. The "Analog Inputs" block gives the measurements of the sensors (they are updated each 5 ms) in radians for the angles of the hub and the 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 you to switch on or switch off the motor. We included a saturator in our diagram just before the signal goes in to the "Analog Output" block because the output voltage is limited from $-5V$ to $5V$. The saturator settings in our diagram are set accordingly. You can also see a sin wave, a step input and a zero input. These are used to represent a reference signal for the output. The sine wave or the step input is the reference for $\theta_{desired}$ and the zero block are 3 signals equals to zero that represent the reference for $\alpha, \dot{\theta}$ and $\dot{\alpha}$.

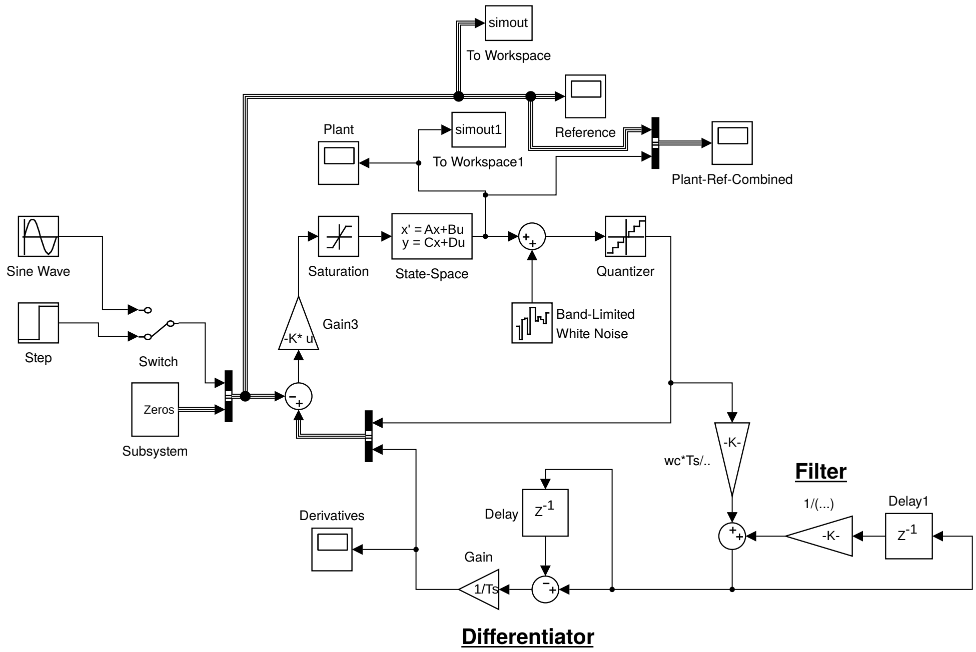

Figure 3: The simulink Layout of our Simulation

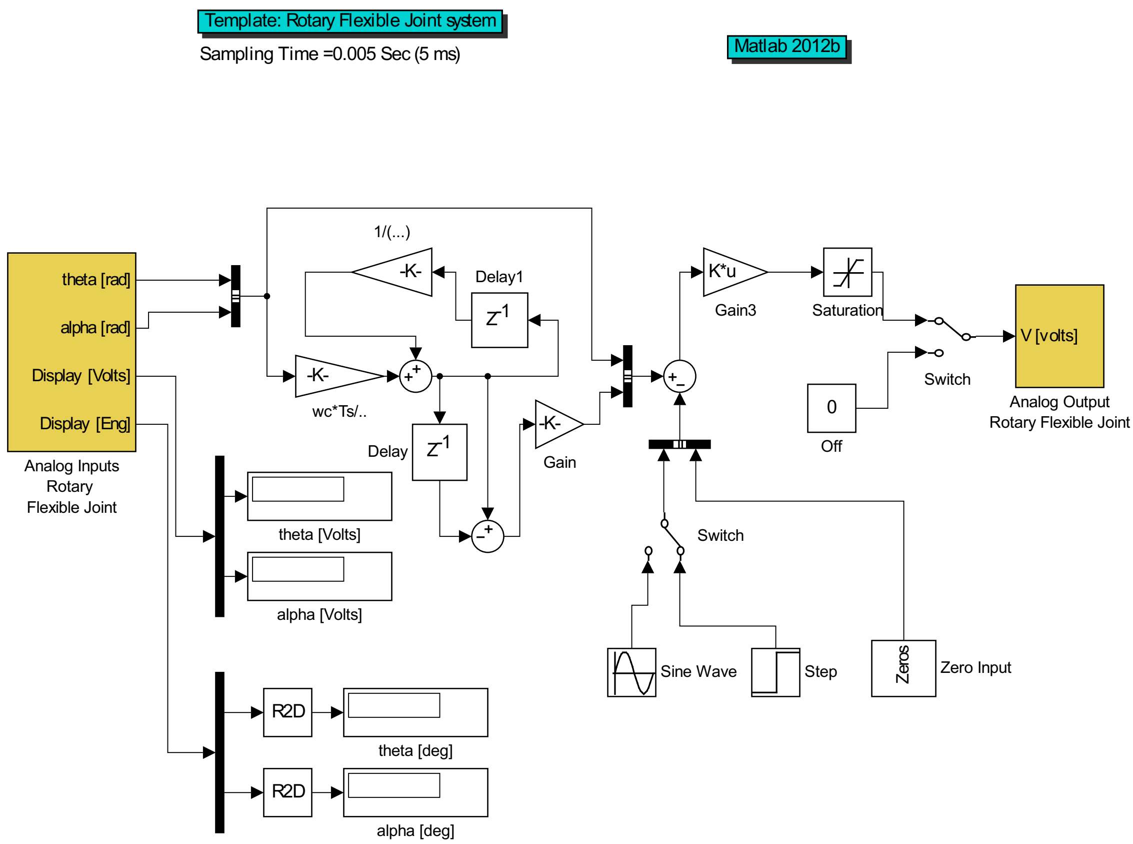

Figure 4: Real-time Simulink diagram