Report Project MECA482 - Furuta Pendulum Full State Feedback

Taylor - Sanchez - Tissel - Fragasso - Karam

May 2020

1 Introduction

The goal of this project was to model and control the Furuta Pendulum with a full state feedback, to keep it at the unstable equilibrium. Moreover a simple tracking position system has been integrated.

2 Modeling and Linearization

The analytical model of the Furuta Pendulum has been retrievied on the Quanser website [?]. The symbolic manipulation has been performed on Maple, which is is a powerful symbolic manipulation program, and offers libraries to solve non-linear ODEs. It has also Code Generation command to easily export the epressions in Matlab in a very fast way to use all the computing power of MathWorks.
The system is a 2 D.O.F. system with input the voltage of the motor of the rotating arm. The system is an underactuated manipulator since it is possible to actuate only the rotating arm.

In Maple, the differential equations of motion of the system have been defined and they have been solved within it the software with a specific set of parameters to validate the model. Then the system has been linearized about the up position and the matrixes A,B,C have been computed and exported in Matlab.

The system is a 4th order system with the state vector [θ, α,θ_{dot},α_{dot}]. The Matrixes A and B are given by:

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 34.69 & -17.54 & 0 \\ 0 & 62.27 & -13.60 & 0 \end{bmatrix} \quad B = \begin{bmatrix} 0 \\ 0 \\ 32.98 \\ 25.56 \end{bmatrix}$$

The matric A has poles $[0, -19.10, 6.53, -4.97]$, which "proves" the linearized point is unstable. Nevertheless the couple (A,B) is controllable and it is possible to apply a Full State Feedback Control.

3 Full State Feedback Control

Once the matrixes A and B were available, and after veryfing the couple is controllable, it has been possible to apply a full state feedback controller and compute the matrix K to set the desired behavior of the system. The 4 poles of the system has been chosen in the following way.

- A couple of complex poles to set the settling time T_s and the percentage overshoot $\%OS$;
- The other two with real part at least 10 times the real part of the complex poles so that they are negligible with respect to the dominant couple of complex poles;

The settling time chosen was of 3s with an overshoot of 12%. The poles we obtain are $[-14.66, -13..33, -1.33 + 1.97j, -1.33 - 1.97j]$ which require a gain controller K given by:

$$K = \begin{bmatrix} -0.9524 & 14.4577 & -1.1155 & 1.9523 \end{bmatrix}$$

The system with dynamic matrix A-B*K has an effective settling time and an overshoot of 2.55s and 13.65% respectively.

4 Tracking Input

Furthermore the input has been scaled with a gain N such that the output would follow a reference input signal. The reference is the angle theta of the rotating arm which enables us to control the angle position theta of the rotating arm.
In order to do that it has been made sure that the DC gain between the reference R(s) and the output Y(s) was 1. The gain G between R(s) and Y(s) has been computed:

$$DC_gain = -1.0499$$

so that the scaling input costant is given by:

$$N = 1/G = -0.9524$$

The overall architecture of our system is the following:

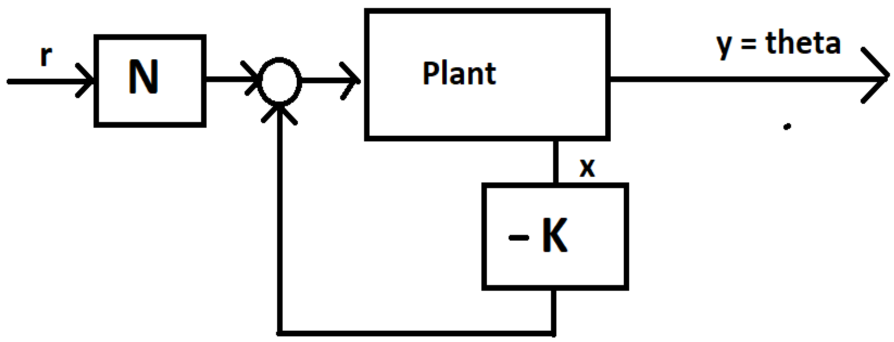

Figure 1: Architecture of the closed loop

5 Connection Matlab and Coppelia

Once the block K and N are computed, it was necessary to verify how the controller would behave with the actual system. Since it is expensive and time consuming building and testing the controller on a real system, the simulation environment Coppelia has been adopted. Coppelia is a software which allows to build a physical system in a simulated environment. Code can be integrated to add the actuation and the sensing code as well the interface with Matlab.
The interface between Matlab and Coppelia is realized with the software BlueZero which creates a communication channel between Coppelia (the server) and Matlab (the client). During the simulation Matlab will be asking Coppelia to measure the state of the furuta pendulum, and will send to the server the control voltage input computed using the control feedback. The voltage will be converted in a torque by the Coppelia actuation code.

6 Analysis of Results

For the analysis of the results, Matlab controls the Furuta Pendulum to follow three subsequent references to test the ability of the controller to follow them one after the other. In particular the Matlab controller;

1. It stabilizes the Rod, to keep it still
2. The reference r goes to 1 rad;
3. The reference input goes to -2 rad;

What it has been verified was:

1. If the feedback can stabilize the non linear system

2. The measured settling time and overshoot is close or same with the theoretical ones.

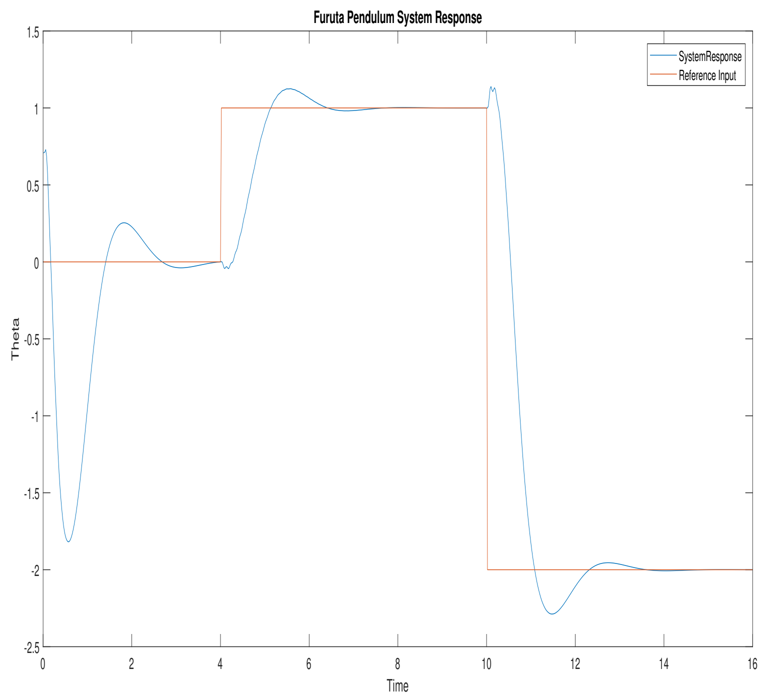

Figure 2: System Response of the system

After sampling the system response from Coppelia, the settling time and overshoot are measured. The time window when we pass from r=0 to r=1 has been considered, which basically the step response of the system. To measured them the definition have been applied:

$$\%OS = (Peak - Final_V alue)/Final_v alue * 100$$

Settling time: time such that the output reach and stay within the 2

So we extracted the window related to the reference signal r=1 and made the measurements. We got OS = 12.48 and T_s = 1.98, which are both pretty close to what we wanted.

References

[1] MIT Lecture Topic 11 16.31 Feedback Control Systems

https://ocw.mit.edu/courses/aeronautics-and-astronautics/16-30-feedback-control-systems-fall-2010/lecture-notes/MIT16_30F10_lec11.pdf

[2] Quanser: Student Workbook Jacob Apkarian, Ph.D., Quanser Paul Karam, B.A.SC., Quanser Michel Lévis, M.A.SC., Quanser Inverted Pendulum Experiment for LabVIEWTM Users