S13-11 Design of A Fuzzy Controller for Inverted Pendulum

Intermediate Report

Otso Mäki
Vesa Nikkilä
Sami E-Madhoun

> *"In a reporting event, the status of the project is presented by using the project plan as a basis. Do not make any changes to the original plan, but report only the modifications to the original plan. Show also what you have achieved this far in the project."*

Project Goal

The goal of the project is to design a fuzzy controller to be used in Simulink to control and stabilize two different systems. The systems examined are the inverted pendulum and the ball and beam system.

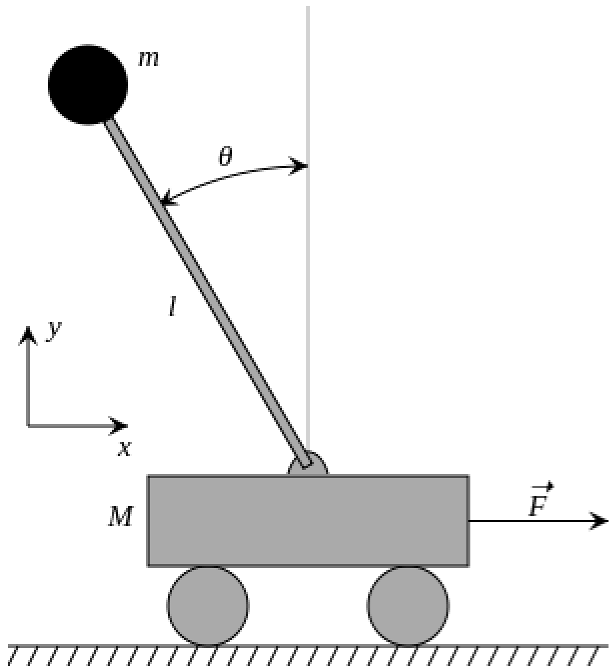


Figure 1: Inverted pendulum

The inverted pendulum model can be seen in Figure 1. The model has one one input which controls the acceleration of the cart. With right control values, it is possible to stabilize the rod which is attached to the cart. Sometimes the inverted pendulum problem contains the problem of also controlling the position of the cart while balancing the pendulum. However, it is not part of this project and only the angle of the pendulum is controlled.

Second problem is to control ball-and-beam system (Figure 2). In this task, we will control the angular acceleration $\ddot{\theta}$.

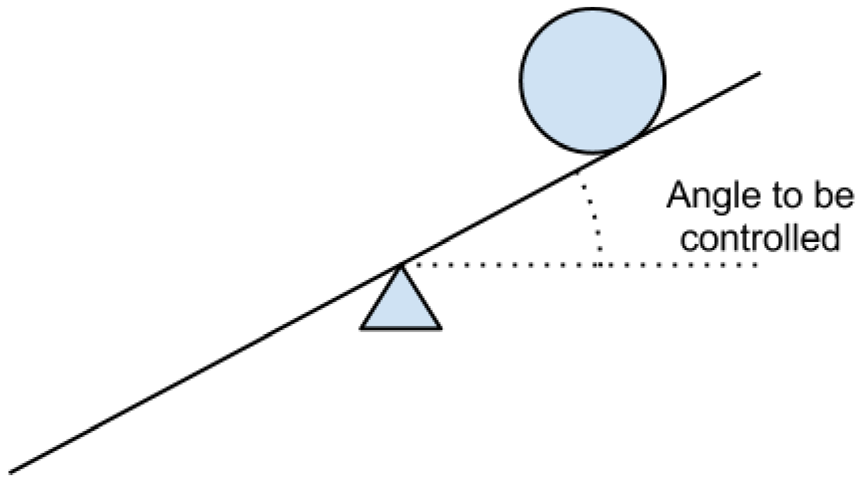


Figure 2: Ball-and-beam system

Project Structure

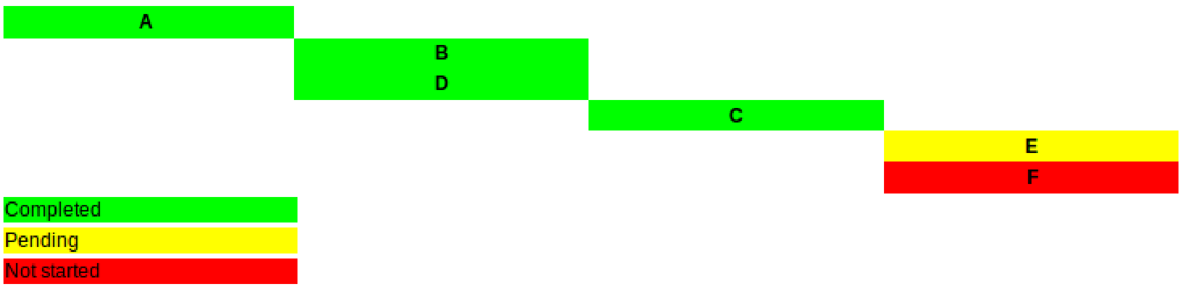
This project consists of the following work packages:

- Revision of fuzzy control theory
    - We will revise the theory of fuzzy control (Mendel M. Jerry, Fuzzy Logic Systems for Engineering: A Tutorial)
    - Workload: 5 hours/group member
    - Deadline: 8.2
    - **Completed!**
- Designing the structure of the fuzzy control algorithm
    - Separate functions are needed for fuzzification, decision making and defuzzification
    - Steps in designing:
        i. Identify the inputs and outputs of the overall function and determine their formats
        ii. Find what are the inputs from the first function to the second function etc.
        iii. Determine the main computational problems in each of the three functions
    - Workload: 20 hours/group member
    - Deadline: 15.2
    - **Completed!**
- Implementing the algorithm and verifying it with by comparing it to Matlab's own implementation of fuzzy control
    - Workload: 30 hours/group member
    - Deadline: 20.3
    - **Completed!**
- Create Simulink models for the inverted pendulum and the ball and beam system
    - A straightforward part that can be done simultaneously with the implementation
    - Workload: 5 hours/group member
    - Deadline: 20.3
    - **Completed!**
- Tuning the controller.
    - Select the input variables for both problems
    - Select the number and shape of membership functions for both inputs and output
    - Create the decision matrix governing the rules of the controller
    - Workload: 5 hours/group member
    - Deadline: 1.4
    - **Pending**

- Final documentation and creating example cases for presentation
    - Workload: 10 hours/group member
    - Deadline: 1.5

- ### Not started

A	Revision of fuzzy control theory
B	Designing the structure of the fuzzy control algorithm
C	Implementing the algorithm and verifying it with by comparing it to Matlab's own implementation of fuzzy control
D	Create Simulink models for the inverted pendulum and the ball and beam system
E	Tuning the controller
F	Final documentation and creating example cases for presentation (documenting is continuous)



State space representation

Inverted pendulum

Let $x_1 = \theta$ and $x_2 = \dot{\theta}$. Then:

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= \frac{g \sin x_1 - \dfrac{m l x_2^2 \cos x_1 \sin x_1}{m_c + m}}{l\left(\dfrac{4}{3} - \dfrac{m \cos^2 x_1}{m_c + m}\right)} + \frac{\dfrac{\cos x_1}{m_c + m}}{l\left(\dfrac{4}{3} - \dfrac{m \cos^2 x_1}{m_c + m}\right)} u \end{aligned}$$

Where

- g = 9,81 (the gravity of Earth)
- m_c = 1 (mass of the cart)
- m = 0,1 (mass of the pole)
- l = 0,5 (**half** length of the pole)

Ball-and-beam

Let x = $(r, \dot{r}, \theta, \dot{\theta})$. Here r denotes the distance between beam's axle and balls origin.

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} x_2 \\ \alpha(x_1 x_4^2 - \beta \sin x_3) \\ x_4 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} u$$

Where

- $\alpha = 0{,}7143$
- $\beta = 9{,}81$

Fuzzy controller (for inverted pendulum)

Reviewing the theory of fuzzy control

Fuzzy control logics can be divided into three phases (Figure 3):

- **Mapping inputs.** In this phase we calculate how much the input value belongs to each membership function. In the case of inverted pendulum, we have two inputs: theta and thetad. For both of these inputs, there exist 5 membership functions (10 in total).
- **Applying rules.** So far, we have calculated the belonging of input to each MF. Next thing to do is to form rule matrix and apply it to the input belongings (for each MF). After this phase, we will know how much each input belong to each output membership functions.
- **Defuzzification.** In this phase we will get a real number for the output.

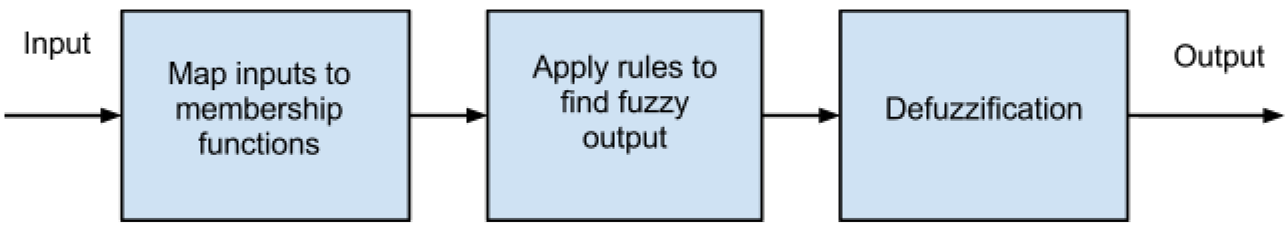


Figure 3: Fuzzy control

Implemented controller

The controller (Figure 4) has total 6 inputs and 1 output. Inputs and outputs are explained in table 1.

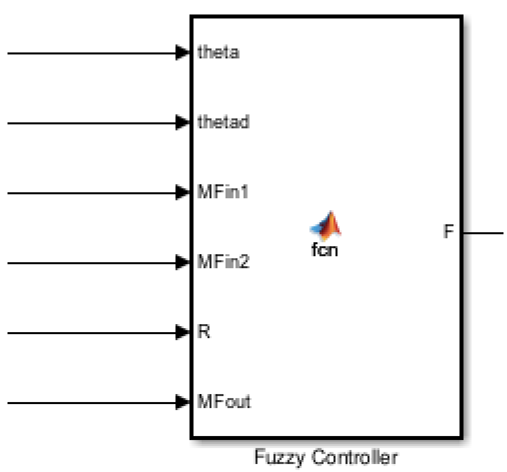


Figure 4: The fuzzy controller

Table 1: IOs for the fuzzy controller in case of inverted pendulum

Inputs		
Name	**Type**	**Description**
theta	scalar	Angle error
thetad	scalar	Angular speed error
MFin1	matrix	Membership functions for theta
MFin2	matrix	Membership functions for thetad
R	matrix	Rule matrix
MFout	matrix	Membership functions for output
Outputs		
Name	**Type**	**Description**
F	scalar	Applied force to the cart

Tuning the controller

We tuned the controller using Fuzzy toolbox. In this way, we saved time because of the graphical interface. As it can be seen from table above, we used matrix notation to represent membership function set. We used triangular membership functions. We need three points on the base of one triangle to express left, center and right point. We can encapsulate these points to one column which will now represent one membership function:

$$\begin{bmatrix} \mathrm{MF}_{1\mathrm{L}} & \mathrm{MF}_{2\mathrm{L}} & & \mathrm{MF}_{\mathrm{NL}} \\ \mathrm{MF}_{1\mathrm{C}} & \mathrm{MF}_{2\mathrm{C}} & \cdots & \mathrm{MF}_{\mathrm{NC}} \\ \mathrm{MF}_{1\mathrm{R}} & \mathrm{MF}_{2\mathrm{R}} & & \mathrm{MF}_{\mathrm{NR}} \end{bmatrix}$$

In our case, 5 membership function are adequate for theta, thetad and for output. Note that the most left and right triangles are split from the middle because we want the center point to be located on the boundary.

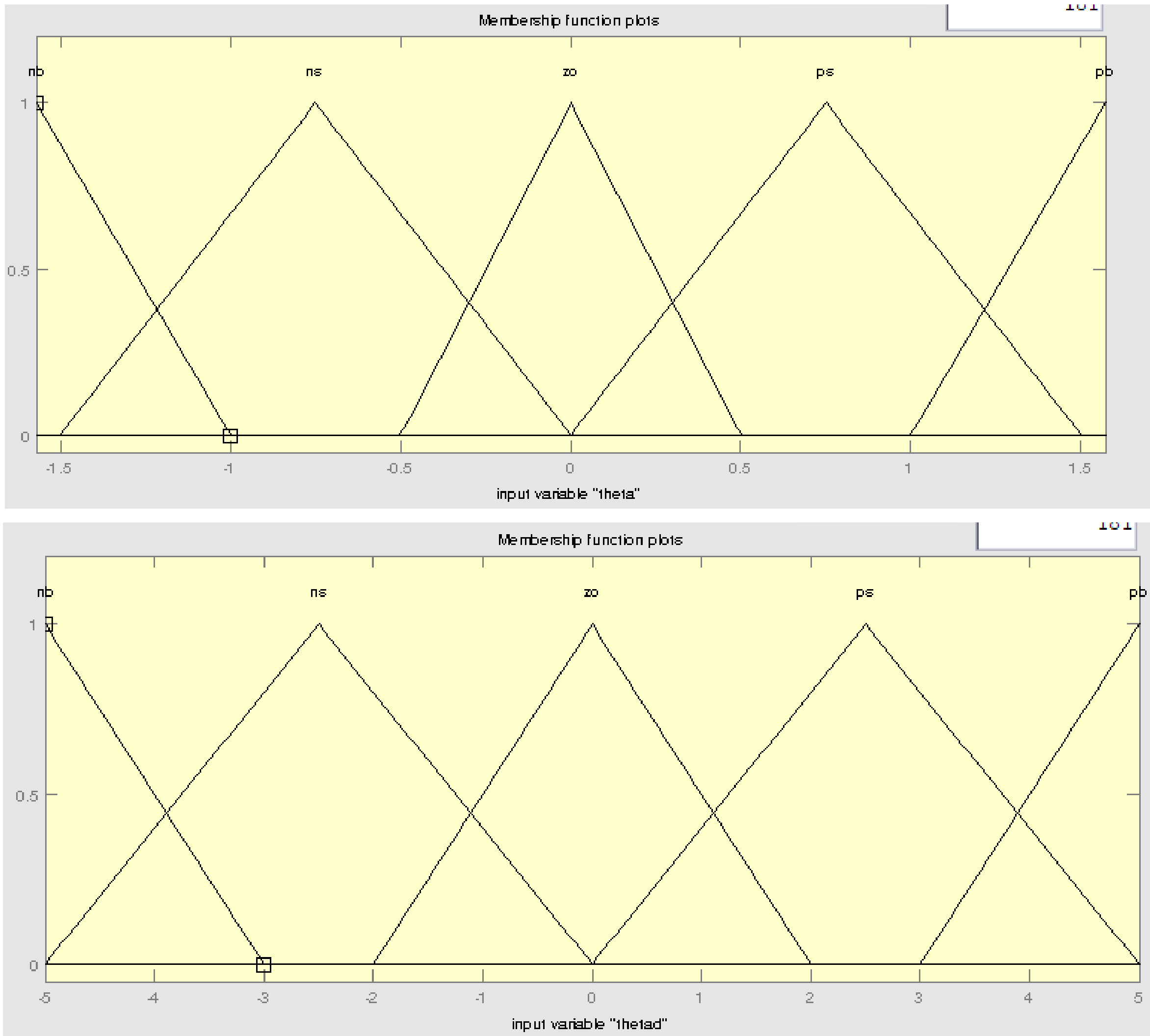


Figure 5: Membership function for theta and thetad

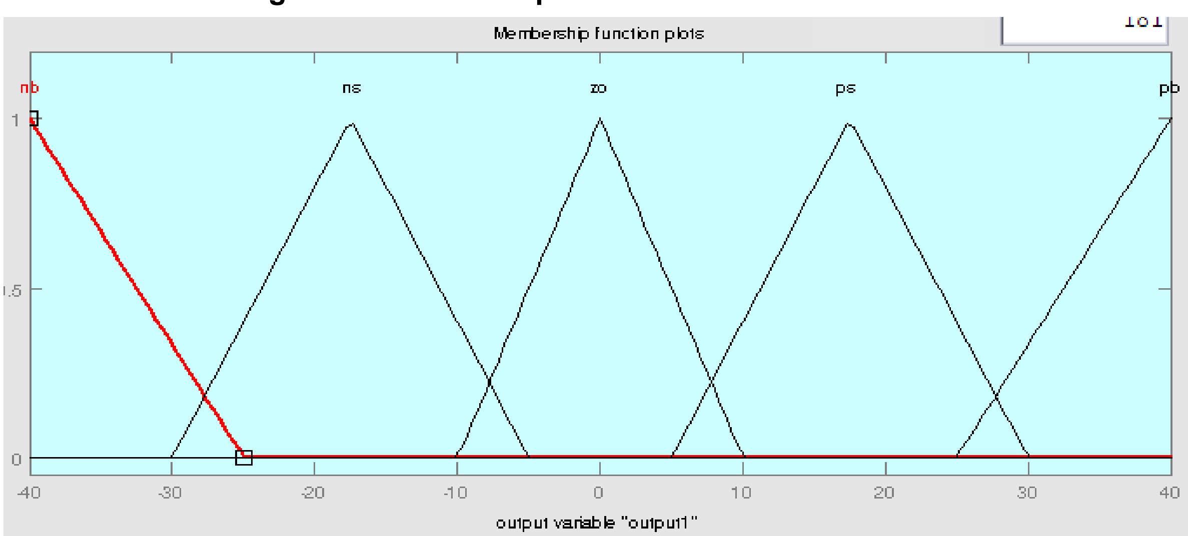


Figure 6: Output membership functions

```
MFin1 = ...
    [-3    -1.5   -0.5  0     1
     -1.57  -0.75  0     0.75  1.57
     -1     0      0.5   1.5   3];

MFin2 = ...
    [-9 -5    -2  0     3
     -5  -2.5  0   2.5  5
     -3  0     2   5     9];

MFout = ...
    [-72  -30    -10  5     25
     -40   -17.5  0    17.5  40
     -25   -5     10   30    72];

R = ...
    [1 1 1 2 3
     1 1 2 3 4
     1 2 3 4 5
     2 3 4 5 5
     3 4 5 5 5];
```

Figure 7: Controller parameters for pendulum

Rules for the controller:

		Angle θ				
		NB	**NS**	**ZO**	**PS**	**PB**
	NB	NB	NB	NB	NS	ZO
	NS	NB	NB	NS	ZO	PS
Angular speed $\dot{\theta}$	**ZO**	NB	NS	ZO	PS	PB
	PS	NS	ZO	PS	PB	PB
	PB	ZO	PS	PB	PB	PB

Initial results

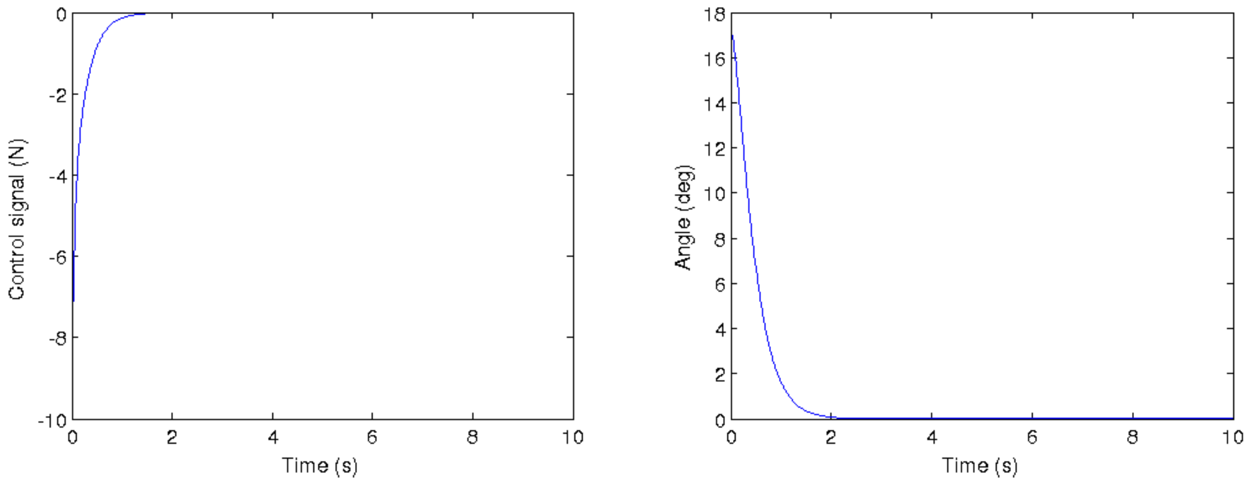


Figure 8: Controller stabilizes pendulum easily without vibration

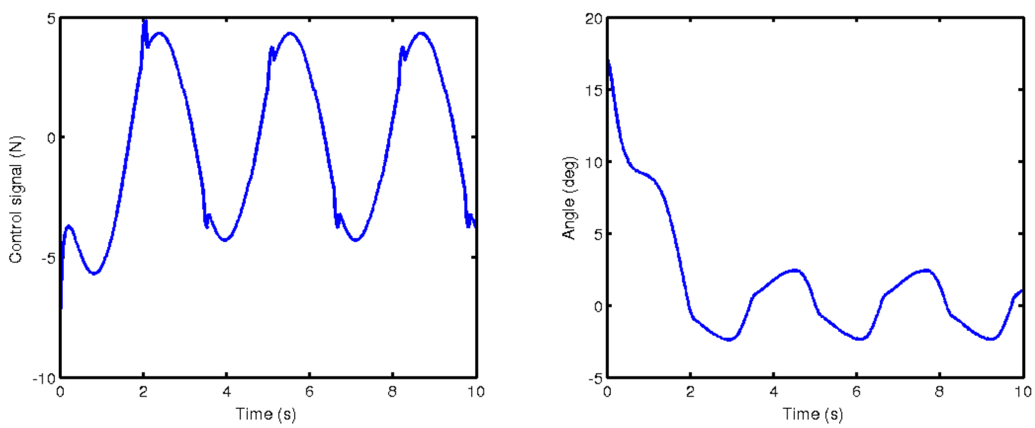


Figure 9: Results with sine interference (amplitude = 4; 1/T = 0,5)

Fun link

http://vesa.hosting.evecy.fi/inverted_pendulum/interface.html