Design of A Fuzzy Controller for Inverted Pendulum

Project Plan

Automaatio- ja systeemitekniikan projektityöt
AS-0.3200
Otso Mäki
Vesa Nikkilä
Sami El-Madhoun

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. The challenge in the fuzzy control design is to select adequate amount of control variables and create fuzzy control rules for them. Moreover, it will be challenging to design the membership functions. We will first create our own fuzzy controller without using Matlab’s fuzzy control toolbox.

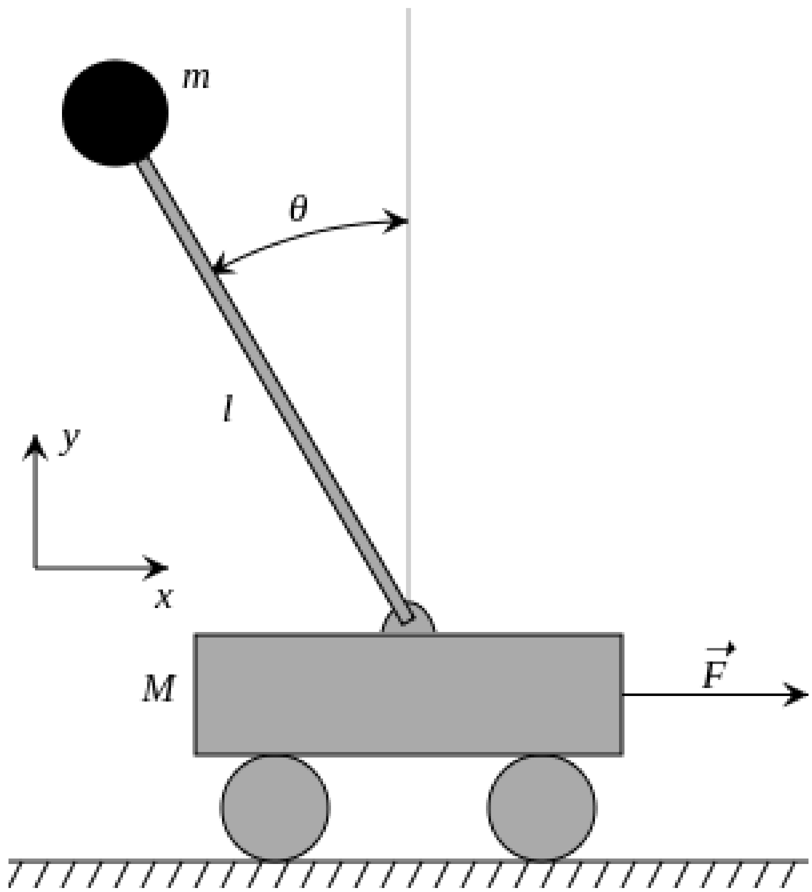

Figure 1. The inverted pendulum.

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.

Secondly, we will design controller for a ball-and-beam system (Figure 2) by manipulating the angle of the beam. The goal is to keep the ball in the origin. The system has one input: angular acceleration of theta. The general structure of fuzzy controller is introduced in Figure 3

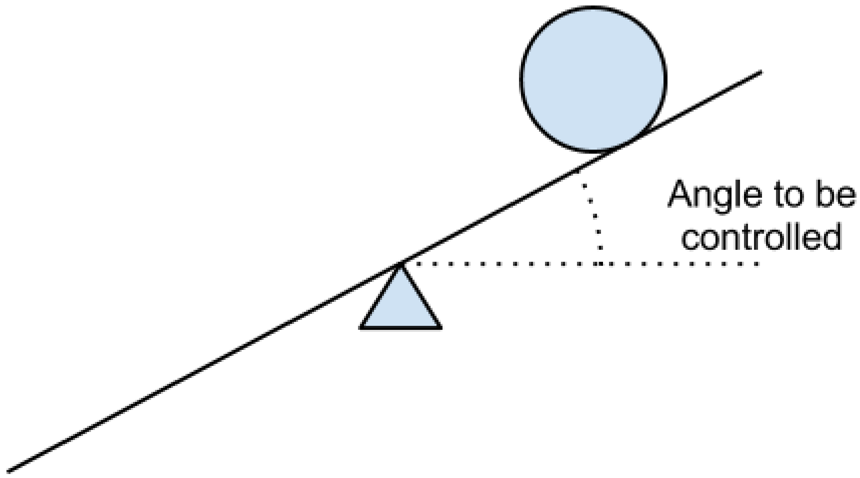

Figure 2. Ball-and-beam system

After finishing the first task, it should be straightforward to complete the second task. This work will be worth of three credits.

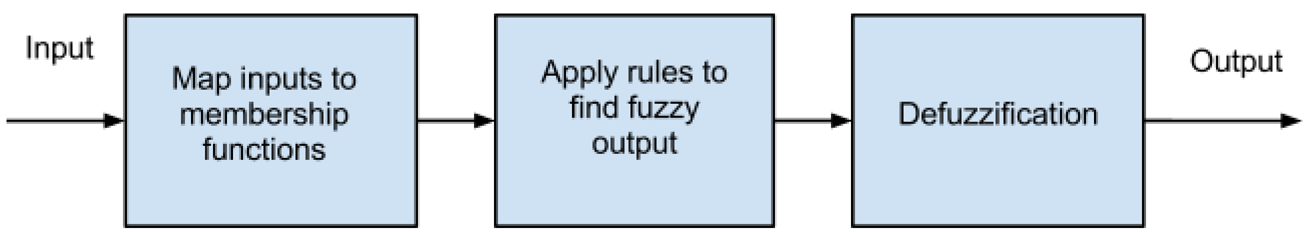

Figure 3. The general structure of a fuzzy controller.

Some material

http://www.ece.drexel.edu/ctm/examples/pend/invpen.html
http://www.library.cmu.edu/ctms/ctms/simulink/examples/pend/pendsim.htm
http://www.iiisci.org/journal/CV$/sci/pdfs/S406IQ.pdf

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
- Designing the structure of the fuzzy control algorithm
  - Separate functions are needed for fuzzification, decision making and defuzzification
  - Steps in designing:
    1. Identify the inputs and outputs of the overall function and determine their formats
    2. Find what are the inputs from the first function to the second function etc.
    3. Determine the main computational problems in each of the three functions
  - Workload: 20 hours/group member
  - Deadline: 15.2
- 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
- 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
- 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
- Final documentation and creating example cases for presentation (documenting is continuous)
  - Workload: 10 hours/group member
  - Deadline: 1.5

These work packages will be done in the exact same order as they are introduced here. Data shown in figure 4 represents the projects workload. Red dot indicates a deadline for a given task. It can be seen that design work and implementing are the most time consuming part of this project.

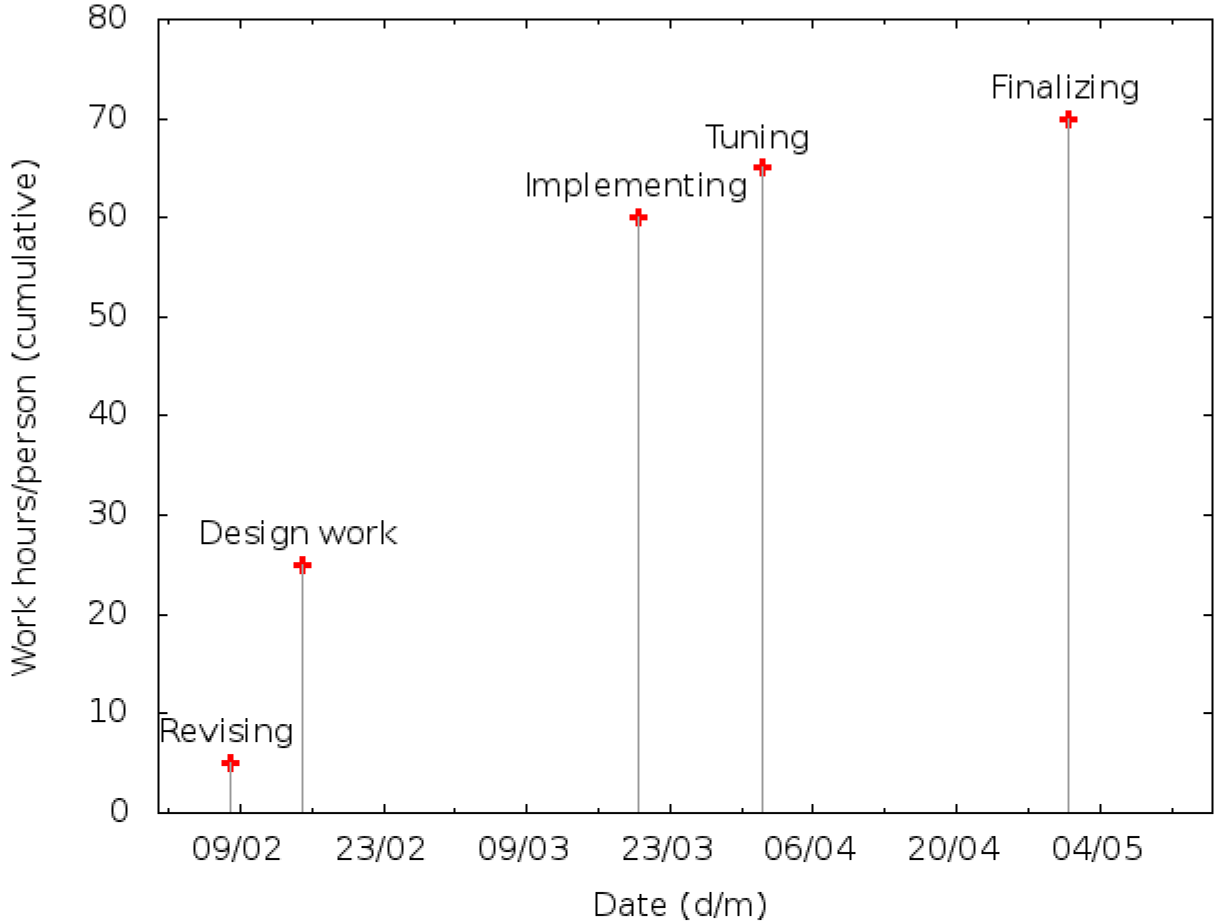

Figure 4. Workload

Time and teamwork management

We will organize group meetings at least weekly, if not more often. In group meetings we will keep each other updated of what have been done. We will also divide the work in group meetings. In addition, meetings can be used for pair programming. This project is not extremely big nor is the size of the group. Therefore too strict planning may cause problems. We will divide the work tasks for every group member in the beginning of every work package.

Risk Management

Risk	**How it can be avoided**
Insufficient time to complete the project	Everyone makes sure that other members of the group keep to the agreed timetable
Getting stuck at creating some part of the algorithm	Keep the code modular so that multiple sections of the code can be implemented simultaneously
Group member becomes sick or is absent	Flexible planning. All group members know each others tasks and can fill in.