Aeropendulum control using soft computing methods

Josep Famadas jfamadas95@gmail.com
Jordi Riu jordiriu10@gmail.com
Emer Rodriguez-Formisano emer@emer.es

Abstract

This project explores two different control approaches based on CI techniques applied to the aeropendulum problem. Firstly, a state of the art fuzzy logic PID controller is studied and taken as reference to challenge a Neural Network Model Predictive Control. After a parameter selection and optimization phase is completed, both methods are compared in terms of performance and computational cost. Strengths and weaknesses of each of them are highlighted.

Keywords: control system, aeropendulum, fuzzy logic control, control system analysis, predictive control, neural network control, PID control, Simulink, Newton-Raphson method, Multi-layer neural network

Contents

1. Introduction

Control engineering has played a vital role in the development of modern industrial society. Applications of control methods appear practically everywhere in consumer electronics, homes devices, in all types of industry, communications systems, modern types of vehicles, mechatronics, robotics, etc. Some of the controllers drive critical systems such as ensuring the stability of a nuclear power plant reactor, the precise autopilot maneuvers of an aircraft (fly-by-wire) or the stability of critical infrastructure like electrical power grids. Although their crucial role, most ordinary people do not perceive it and takes the correct functionality of the systems for guaranteed.
As summarised in [3], *controllers are therefore important for the running of a technical plant when the process to be managed is subject to influences that, without control, would cause the process variables to deviate from the set values.* In a worst case, a system may became unstable an incur to physical damage or be dangerous.

System control field presents significant unresolved research problems and challenging applications. When the progress of the field is studied in [6], a clear pattern is observed. There has been an increasing interest in applying Soft computing methods (SCM) into the control field. In particular, the resource states that *the process control research has been largely concerned with three SCM methods: knowledge-based systems, neural networks, fuzzy logic, and various combinations of these techniques.* It also adds that a survey conducted in Japan showed that fuzzy control has been used in 45% of the plants and Model Predictive Control (MPC) in 42% of the surveyed plants, being both the most common approaches. The researcher also points out that most of the fuzzy control literature focuses on fuzzy rules in combination with proportional integral derivative (PID) control while neural network techniques seems to be combined with model-based predictive control methods.

Given the importance of the field and the fact that various Computational Intelligent (CI) techniques can be applied to control problems, the present work is a catalyst for learning how the techniques are applied and obtaining first-hand experience on the challenges encountered during the process. The aim is to apply Fuzzy PID control and Neural Network Predictive Control to a reasonable control problem and compare the performance and limitations.

The report is divided in the following parts: sections 2 and 3 set the goals and provides the context. The details of the control problem are describe in 4, the methods proposed are described in section 5 while their results are presented section 6 and their comparison in 7. Section 8 concludes the work listing the achievements.

2. Problem statement and goals

Section 1 provided some hints about the aim of the work which are formally listed next. The objectives are:

- To understand the main ideas of some basic and advanced controller methods, PID and MPC.

- To understand how to apply Fuzzy Logic techniques to a control problem.
- To understand how to apply Neural Network techniques to a control problem.
- To implement approaches to the proposed control problem and compare the performance and limitations.
- To master the tools (Matlab Simulink and toolboxes) used during the learning process.

The approaches proposed in order to achieve the goals are two CI control methods called *Fuzzy PID controller* and *Neural Network Predictive Controller*. Both approaches are commonly used in the industry and both combines Computer Intelligence techniques.

The definition of the control problem and the understanding of its physics are fundamental in order to achieve the objectives.

3. Previous work

In order to apply CI methods in control system, a control problem is required. Task 6 of the Fuzzy Laboratory[5] presented the challenge of controlling the angle of a pendulum with a propeller attached (aeropendulum) and it was used as a starting point. The aeropendulum was original proposed by [2] as an affordable control system project especially suited for academia.

The Task 6 Lab definition contained a brief explanation of the physics and the Transfer function used to model the plant. However, the simplifications done in the plant raised serious limitations which were resolved by proposing a more realistic model detailed in section 4.

Once the plant was correctly modeled, a more challenging scenario was proposed in order to benchmark the performance of the controllers. The scenario consisted of repeating sequence of target angles (20, 25, 10 and 15 degrees) with a sample time of 5 seconds. The controller is required to achieve the target angle within the 5 second interval.

The Lab solution also provided a simplistic Fuzzy Controllers which used the error and its derivative with a set of rules in order to conclude a useful thrust level to achieve the target angle. However, the performance of the controller was terrible and too far from meeting the requirements. Figure 1 shows the target angles of the test scenario together with the system response without control and the control provided by the T6 Fuzzy Controller.

It was obvious that the a more challenging opponent was required. Fortunately, Taskin recently presented [8] an enhanced Fuzzy Controller for the same control problem. The proposed solution is based in PID controllers, extensively used in the industry due to its functional simplicity and robustness in a wide range of operations conditions.

Given the challenge, the Neural Network Predictive control will focus most of the attention in order to find a good settings to outperform the state-of-the-art Fuzzy PID approach.

4. Aeropendulum system

4.1. Plant

The control system presented in Task 6 used a linearized version of Equation (7). This setup leads to unrealistic behaviours with values of θ for which the approximation $\sin(\theta) \approx \theta$ is not valid. For this reason, in this project the plant uses the exact model of the aeropendulum. Specifically, its implementation in Simulink can be seen in Appendix B - Figure 11.

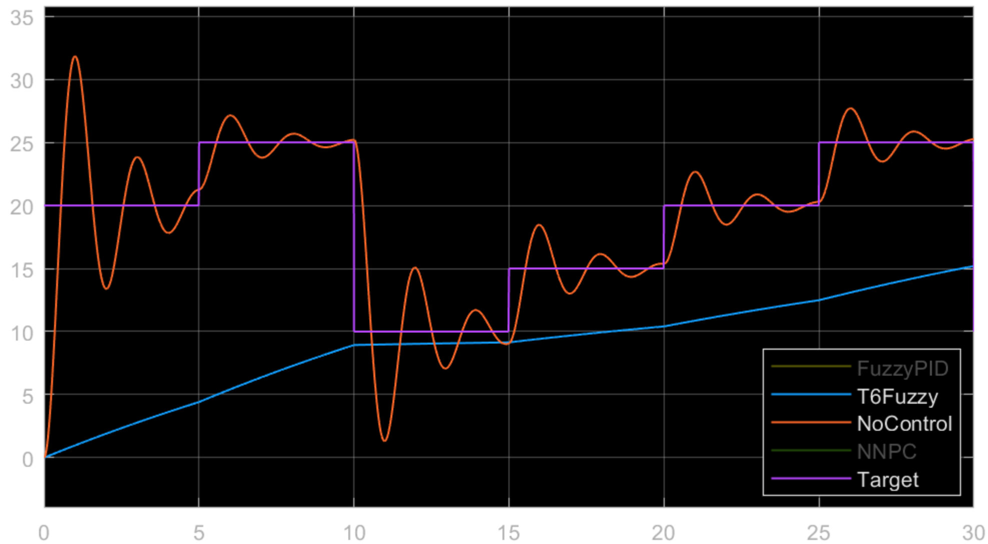

Figure 1: Test scenario with T6 Lab Fuzzy Controller

4.2. Experiment parameters

Since the stabilization time of the system without control is around 300s, generating unbiased simulated data (used to train the Neural Network plant model) both for the stationary and transient states of the system would require a high computational load. For this reason, the viscous damping (c) has been incremented from 0.00035 Nms/rad (Task 6) to 0.01 Nms/rad which results in a stabilization time close to 10s. The reponse with the new parameter is aligned with the results obtained in real world experiment showed in [2] which the pendulum stabilise in 4 seconds.
Moreover, modifying this value also allows the design of a control system for a higher range of θ values as one of the problems faced by the NN controller is that the system is highly unstable and therefore it is difficult to generate usable training data.

In Figures 2 and 3 the plant responses for the same fixed thrust value are displayed. The rest of parameters used are equal to the ones in Task 6.

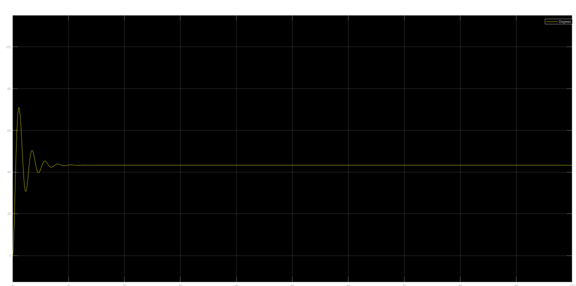

Figure 2: Plant response with $c = 0.01 Nms/rad$

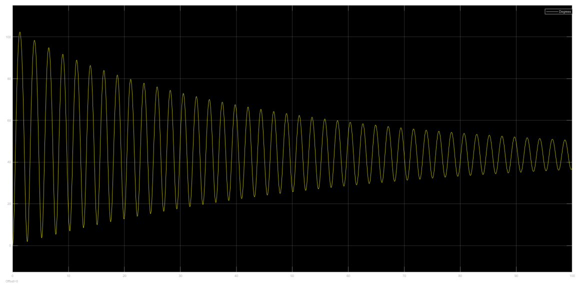

Figure 3: Plant response with $c = 0.00035 Nms/rad$

5. Control and CI methods

Two approaches are explored in order to perform the control: a Fuzzy PID controller and a MPC (which uses Neural Networks).

The former has already been proven to yield great results for the aeropendulum problem both in terms of computation time and its output controlled function. For this reason, it will be used as a reference to evaluate the performance of the designed NN-MPC.

5.1. PID with Fuzzy Logic

The Fuzzy PID controller used as reference is described in [8]. The thrust force T is computed taking into account the past (Integral), present and future (derivative) error e of the plant value. The error is defined as the difference between the desired reference angle and the angular position of the pendulum. Therefore,

$$e = \theta_{ref} - \theta, \tag{1}$$

$$T_{PID} = K_P e + K_I \int e dt + K_D \frac{de}{dt} \tag{2}$$

Each gain value K_P, K_I, K_D is treated as a fuzzy variable inferred from its corresponding error related measure using a Mamdani inference approach.

The membership functions used for the error, derivative error and integral error are shown in Figure 4 and the ones for K_P, K_I, K_D can be seen in Figure 5.

The set of fuzzy rules used by each Fuzzy Logic unit is very simple. Each fuzzy set of the input is mapped to its corresponding output fuzzy set as follows,

$$\begin{gathered} if \text{ input } is \text{ S } then \text{ output } is \text{ SG} \\ if \text{ input } is \text{ M } then \text{ output } is \text{ MG} \\ if \text{ input } is \text{ B } then \text{ output } is \text{ BG} \end{gathered}$$

The implementation of the fuzzy PID controller in Simulink is shown in Appendix B - Figure 12.

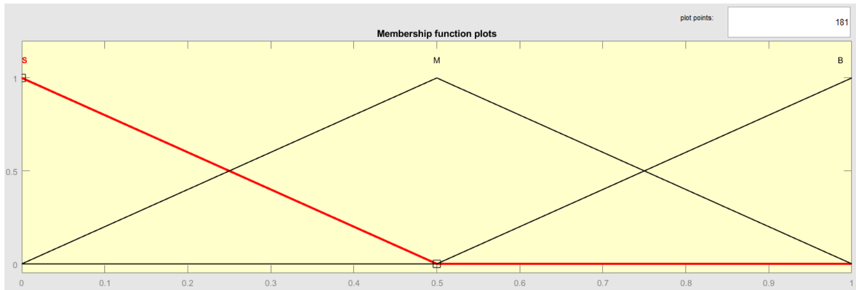

Figure 4: Fuzzy Sets with its corresponding membership functions for $e, \int edt, \frac{de}{dt}$

.

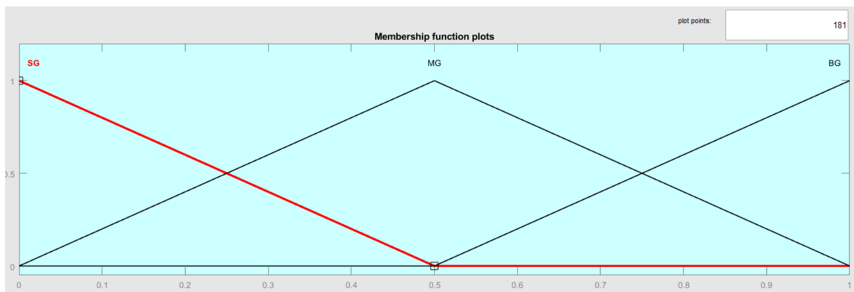

Figure 5: Fuzzy Sets with its corresponding membership functions for the gain values K_P, K_I, K_D

.

5.2. MPC with Neural Networks

The Neural Network Model Predictive Control proposed is a special case inside the Model Predictive Controls presented by the NASA Langley Research Center [7]. It is a powerful method composed by two main components, a neural network that models the plant and Cost Function Minimization algorithm. The prediction of the plant is used to derive the an optimal control signal. The optimization algorithm chosen is Newton-Raphson method. Although the overhead generated by the cost of computing the Hessian, the reduced number of iterations to convergence makes it a faster algorithm suitable for control. The objective function[1] to be minimized is presented in (3).

$$J = \sum_{j=N_1}^{N_2} (y_r(t+j) - y_m(t+j))^2 + \rho \sum_{j=1}^{N_u} (u'(t+j-1) - u'(t+j-2))^2 \tag{3}$$

The important parameters to be tuned are the Cost Horizon (N_1 and N_2) and the Control Horizon N_u. The predictions of the plant will run from N_1 to N_2 future time steps while N_u limits the control horizon.

The accuracy of the predictions impacts the performance of the approach. Neural Networks are known for capturing nonlinear dynamics which is a useful property for system identification [4]. The literature specifies that Multi-layer Perceptron with one hidden layer provides good results. The best set parameters obtained during the tuning phase are listed next:

- Size of the Hidden Layer: 7. As seen in Annex B - Figure 14 this is the number of hidden units for which the MLP performs better.
- Number of delayed plant inputs: 1.
- Number of delayed plant outputs: 3. This combination is the one that present better results (Appendix B - Figure 15)
- Activation functions: Given that the network will act as if it was facing a regression problem the selected function for the output layer has been the Linear. In order to break the linearity, for the hidden units it has been selected the Sigmoid.
- Cost function: Mean squared error. For any regression problem it is the most commonly used.
- Training function: Levenberg-Marquardt.

The training results of the Neural Network can be seen in Appendix B - Figures 16, 17 and 18.
Apart of the Neural Network, the optimizer also had some hyperparameters to tune:

- Cost Horizon: 12.
- Control Horizon: 4.

This values were choosen from different simulations (Appendix B - Figures 19 and 20) but taking into account that if they were too big, the computational cost would increase a lot.

6. Results and Discussion

In this section the results obtained by both control approaches are presented and compared for different scenarios.
The first scenario under study is the one proposed in Section 3. The target function consists of repeating a sequence of angles (20, 25, 10 and 15 degrees) with a sample time of 5 seconds. The results are shown in Figure 6.

Although the evolution of the target function was not too abrupt it can be observed that the MPC reaches stability slightly before the PID as it was expected.

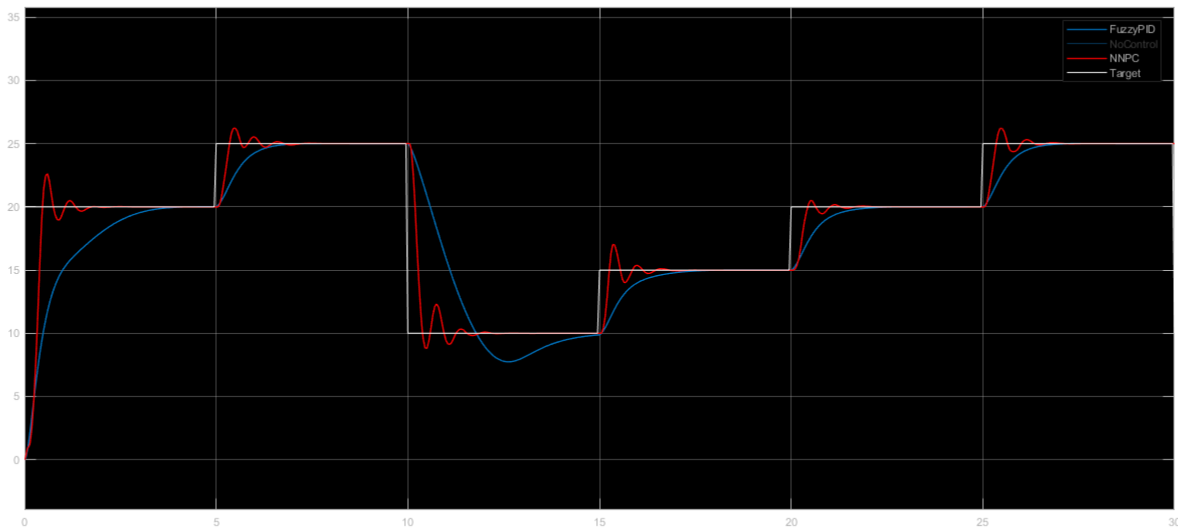

Figure 6: Performance of the MPC (red) and PID (blue) controllers for scenario 1 target (white).

For the second scenario the intention was to challenge the controllers by making a target function with an abrupt change (from 40 to -40 degrees). Again, the MPC outperforms by far the PID controller (Figure 7).

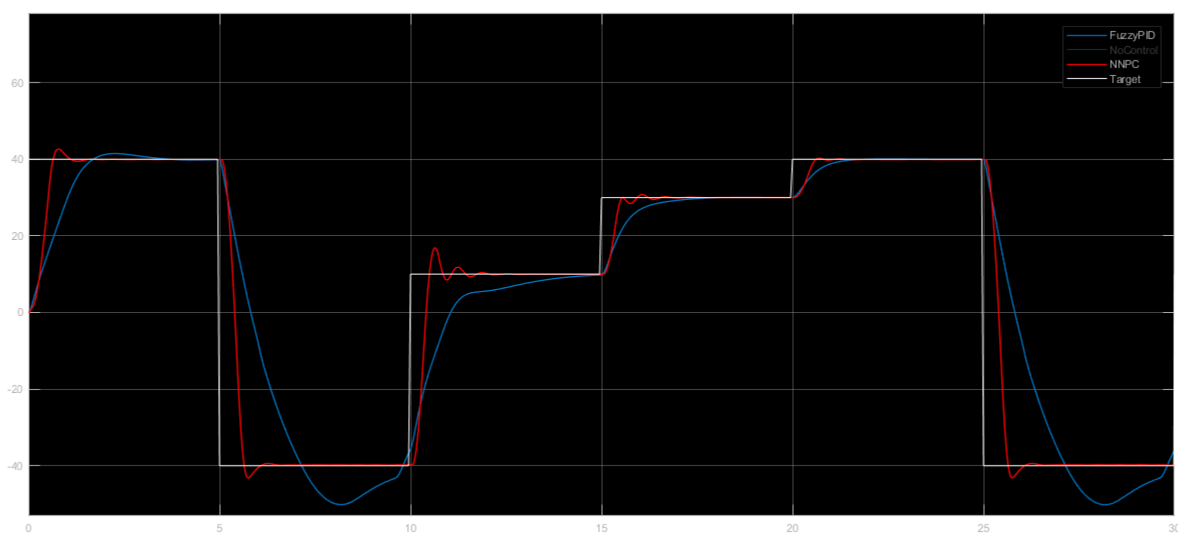

Figure 7: Performance of the MPC (red) and PID (blue) controllers for screnario 2 target (white).

The control system is also expected to suffer for high frequency target functions. For this reason, the third scenario uses the target from the original one but with a sampling time of 1 second. There is a clear distinction between the oscillatory but highly reactive nature of the MPC approach and the steady but slower response of the PID approach (Figure 8).

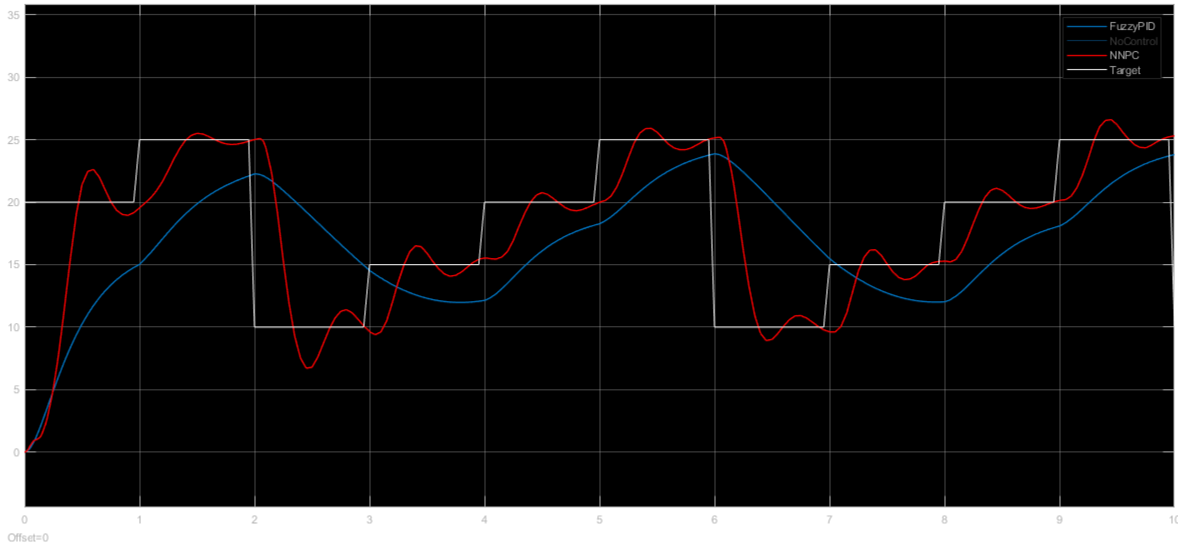

Figure 8: Performance of the MPC (red) and PID (blue) controllers for scenario 3 target (white).

7. Strengths and weaknesses

Some of the strengths and weaknesses were already highlighted in the previous sections but they are all summarized here.

The MPC controller has a better performance than PID when it comes to the early stability and response to fast changes of the target. However, this better performance has its costs. As seen in the previous figures the MPC oscillates much more than the PID, which may cause mechanical problems in a hypothetical real implementation of the system. Moreover, the computation time of the MPC is far higher than the PID one. In fact, it might be challenging to develop an integrated hardware able to make control in real time.

Another drawback of the MPC is that due to the fact that the open-loop plant is unstable for target angles higher than ± 45 degrees the neural network could not be trained in those cases. For this reason, the operation of the system is limited to the range, and the control cannot operate outside it. (Figure 9).

8. Conclusions and future work

A good understanding of the main ideas of the control approaches proposed was acquired. The application of Computational Intelligence techniques for the control of real systems presents a series of challenges that can be overcome by the use of domain knowledge of the problem (for this specific case, modifying the c parameter, viscous damping). For this

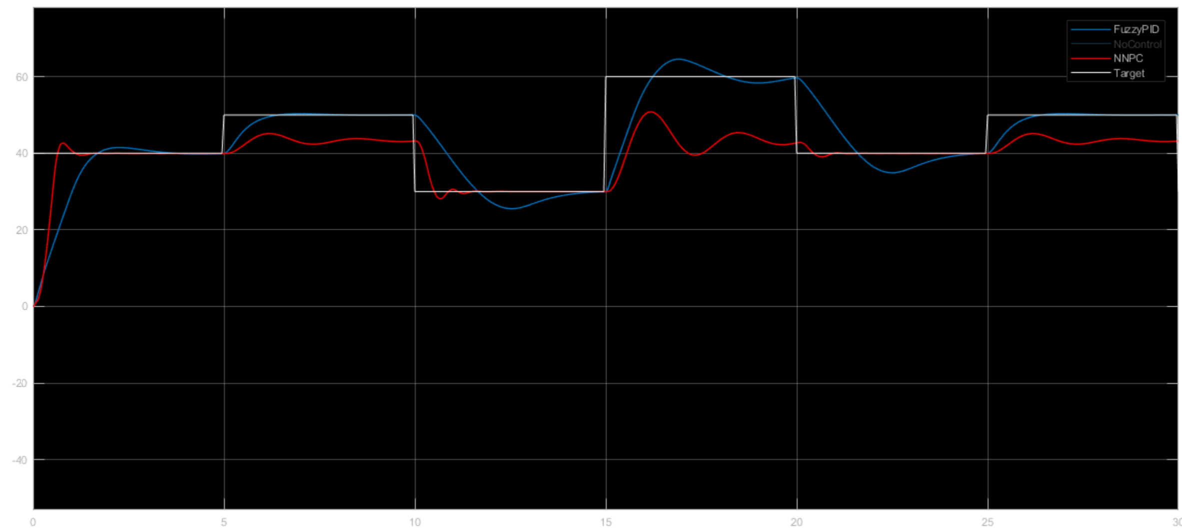

Figure 9: Performance of the MPC (red) and PID (blue) controllers for scenario 4 target (white).

reason, solving this types of problem or more complex ones may require interdisciplinary teams containing both, experts in the domain and the computational system.

Moreover, advanced Simulink features such as Simulation Data Inspector and Logic analyzer were learned and used in order to debug the signals during the development process. This software has proven to be a solid and comfortable environment to work with Neural Network based techniques and the fact of having acquired the previously mentioned knowledge on its use will make it easy to work again with it.

The results obtained and discussed clearly identified the strength and weaknesses of the approaches considered. It has been proven that MPC has a great potential but there are also some aspects that could be enhanced. For instance, when it comes to the system identification, it should be found a method to generate training data for the unstable range of angles during the open loop simulation.

The hyperparamter selection in the training phase of the Network and parameter tuning of the optimization techniques was mainly performed manually during the learning process and discussion. Now that the fundamental ideas are consolidated, it is natural step to implement an algorithm in order to automate the process of exploring the parameters space. However, good comparison function and criteria must be defined so that all the aspects of the controller are considered. Some settings perform better in specific scenarios than others.

Another extra phase that could be done following this project is the implementation on embedded hardware and trying to make it work in real time. This could be done by generating optimized binaries of the model or even develop the solution on HDL and deploy it on a FPGA.

References

[1] Design neural network predictive controller in simulink. URL https://es.mathworks.com/help/nnet/ug/design-neural-network-predictive-controller-in-simulink.html.

[2] Eniko T. Enikov, Vasco Polyzoev, and Joshua Gill. Low-cost take-home experiment on classical control using matlab/simulink real-time windows target, 2010.

[3] Edmund Linzenkirchner. The importance of control engineering in automation. URL https://www.plantautomation.com/doc/the-importance-of-control-engineering-in-auto-0001.

[4] K. S. Narendra and K. Parthasarathy. Identification and control of dynamical systems using neural networks. *IEEE Transactions on Neural Networks*, 1(1):4–27, Mar 1990. ISSN 1045-9227. doi: 10.1109/72.80202.

[5] Maria Angela Nebot Castells. Fuzzy logic systems lab class guide. *Computational Intelligence course of MSc in Artificial Intelligence*, pages 15–19, 2017-2018.

[6] Š. Kozák. Advanced control engineering methods in modern technological applications. In *Proceedings of the 13th International Carpathian Control Conference (ICCC)*, pages 392–397, May 2012. doi: 10.1109/CarpathianCC.2012.6228674.

[7] D. Soloway and P. J. Haley. Neural generalized predictive control. In *Proceedings of the 1996 IEEE International Symposium on Intelligent Control*, pages 277–282, Sep 1996. doi: 10.1109/ISIC.1996.556214.

[8] Y. Taskin. Fuzzy pid controller for propeller pendulum. *IU-JEEE*, 17(1):3175–3180, 2017.

Appendix A. Proof of theoretical results

A.1. Description and physics

The derivation of the governing equation of the aeropendulum system can be done by using Lagrangian Mecanics.

Note that the calculations below are done taking into account the setup described in Figure 10.

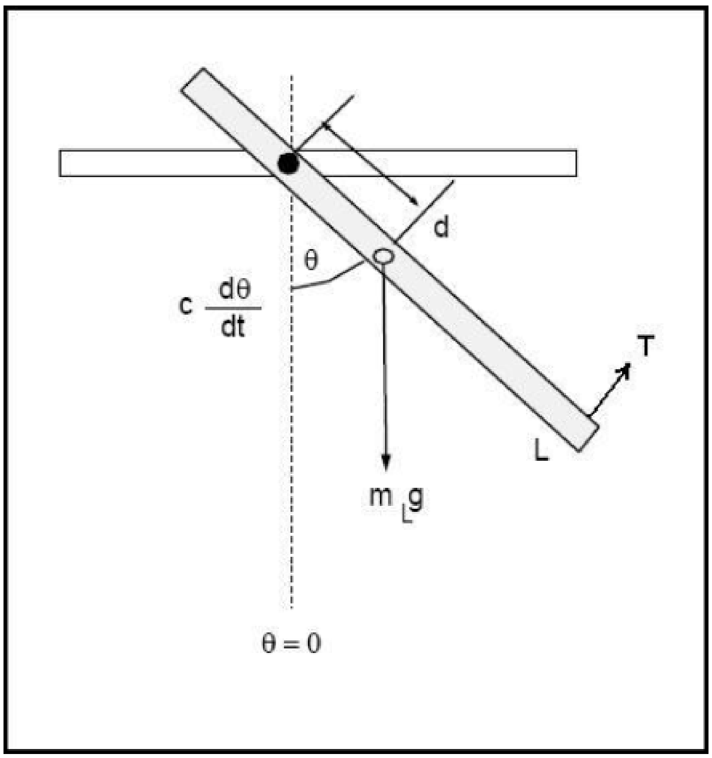

Figure 10: Aeropendulum Set Up.

Let L be the lagrangian of the system, T the kinetic energy of the pendulum and U its potential energy. Then,

$$L = T - U = \frac{1}{2}md^2\dot{\theta}^2 - mgd(1 - \cos\theta) \tag{4}$$

The application of Langrage's Equation to the Lagrangian of the system (in the absence of non-conservative forces) yields,

$$\frac{\partial L}{\partial \theta} - \frac{d}{dt}\frac{\partial L}{\partial \dot{\theta}} = -mgd\sin\theta - md^2\ddot{\theta} = 0 \tag{5}$$

Since the problem under consideration has two non-conservative forces (the artificial thrust T and a damping force D), such terms have to be introduced to the equation of the system by the means of its virtual work W. Specifically, the generalized force value is obtained as $\frac{\delta W}{\delta \theta}$.

Assuming that the thrust force is applied at the edge of the pendulum bar which is at distance $(d + \frac{l}{2})$ from the pivot point, its work is equal to the thrust force multiplied by the distance traveled by the edge. Therefore,

$$\delta W = \delta W_D + \delta W_T = c\dot{\theta}\delta\theta - T(d + \frac{l}{2})\delta\theta \tag{6}$$

Finally, by adding the generalized forces to Equation (5) the plant model for the system is reached,

$$J\ddot{\theta} + c\dot{\theta} + mgd\sin\theta = (d + \frac{l}{2})T \tag{7}$$

Where $J = md^2$ is the moment of inertia with respect to the pendulum pivot and c is the viscous friction coefficient.

Appendix B. Implementation details

B.1. Simulink implementation

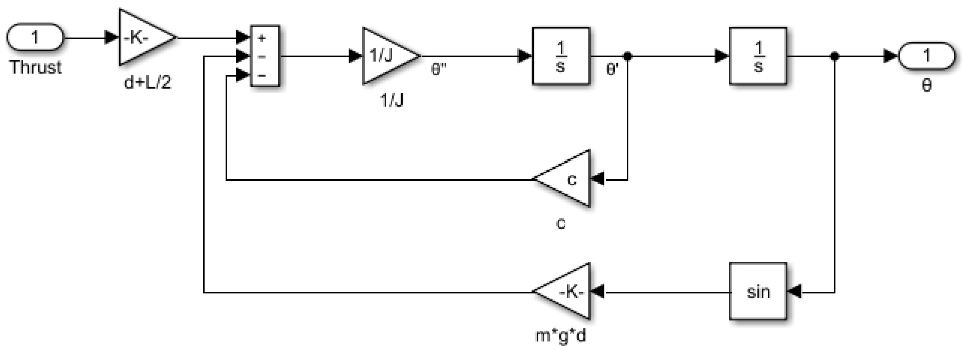

Figure 11: Simulink representation of the Aeropendulum model

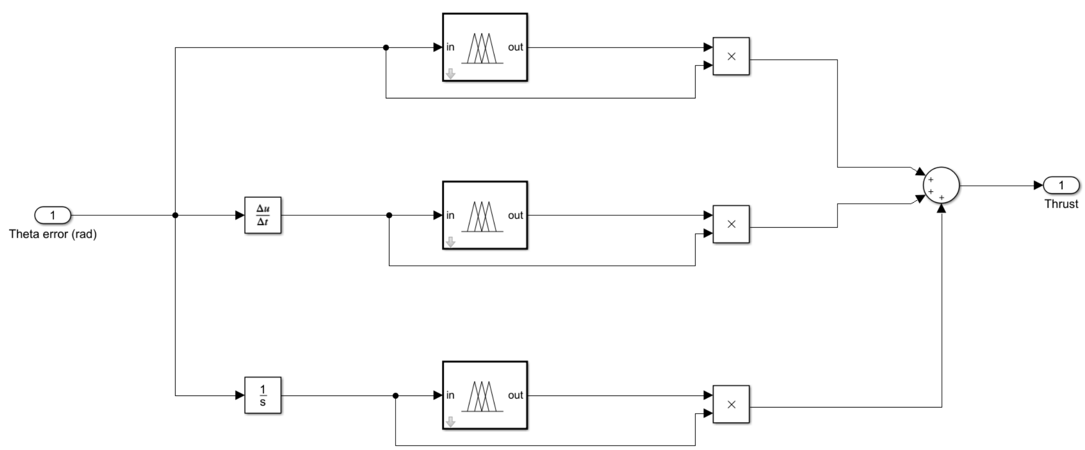

Figure 12: Simulink implementation of the Fuzzy PID controller.

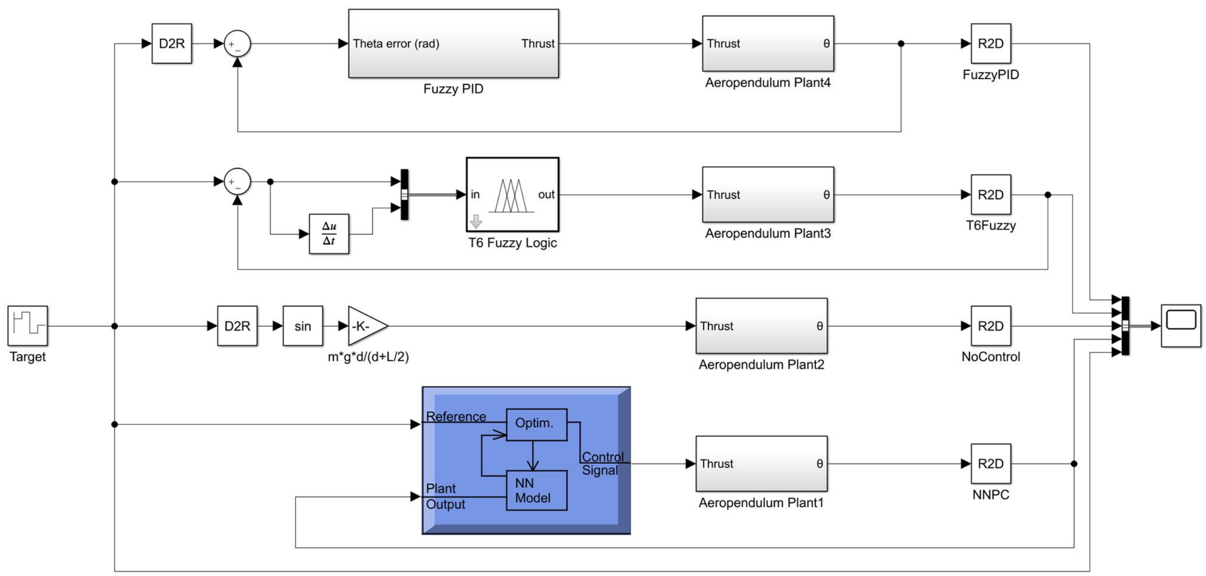

Figure 13: Simulink implementation of the full system.

B.2. MPC hyperparameters selection

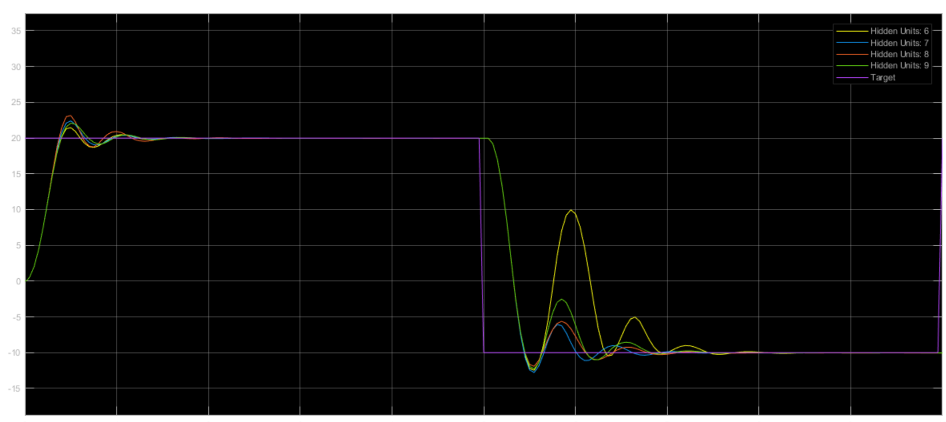

Figure 14: MLP performance with different number of hidden units: 6 (yellow), 7 (blue), 8 (orange), 9 (green).

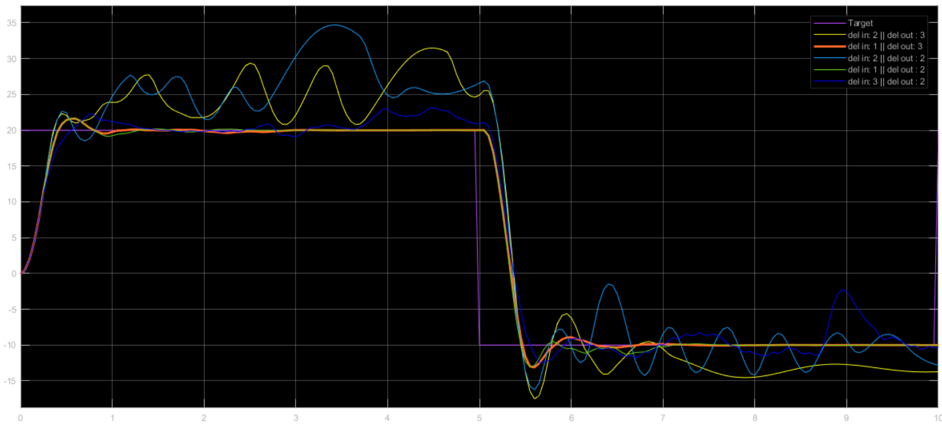

Figure 15: MLP performance with different combinations of delayed plant inputs and outputs. The best one is 1-3 (orange)

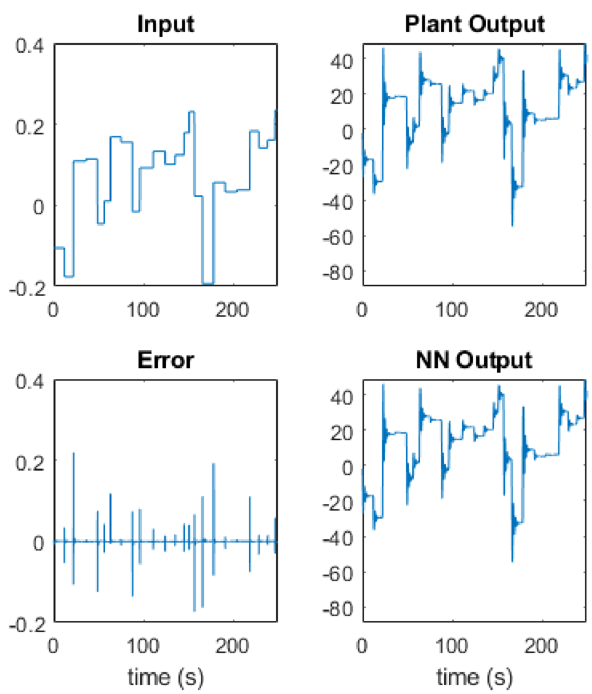

Figure 16: Training phase of the Neural Network.

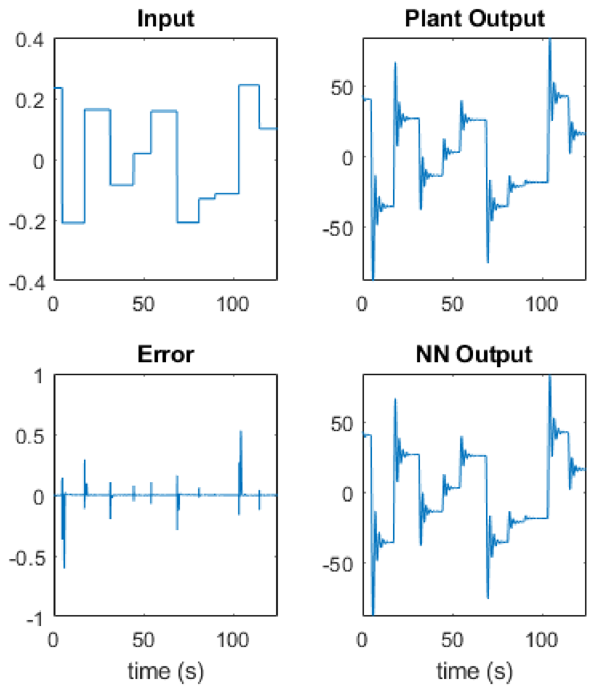

Figure 17: Validation phase of the Neural Network.

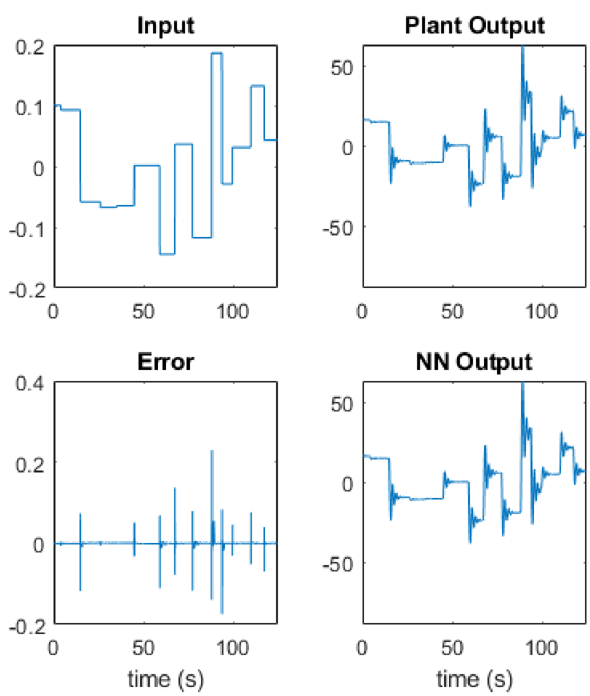

Figure 18: Test phase of the Neural Network.

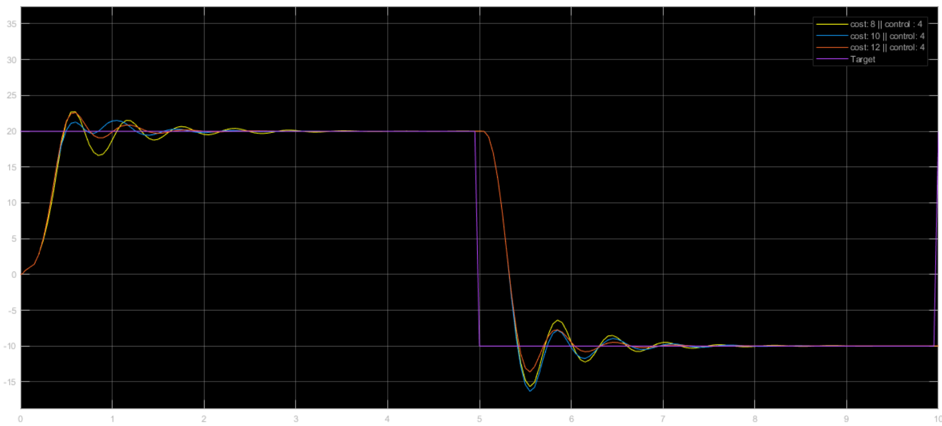

Figure 19: MLP performance changing the cost horizon: 8 (yellow), 10 (blue), 12 (orange).

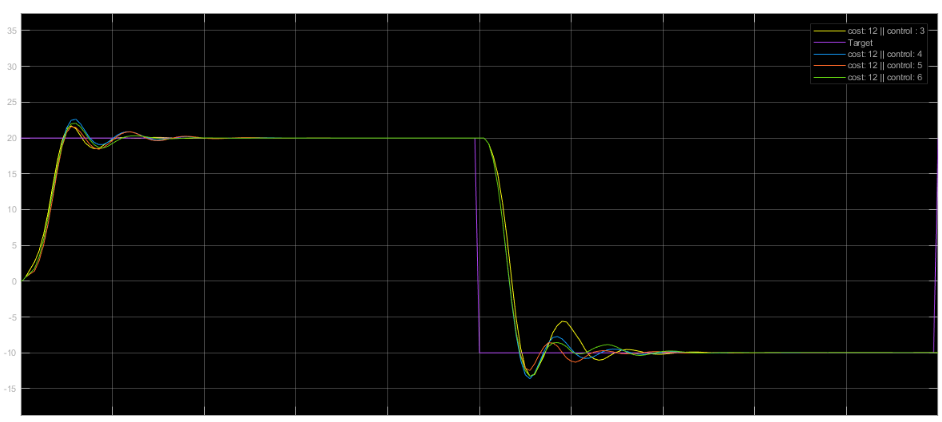

Figure 20: MLP performance changing the control horizon: 3 (yellow), 4 (blue), 5 (orange), 6 (green).