University of Birmingham School of Engineering **Department of Mechanical Engineering** **Individual Engineering Project** **FINAL REPORT** **MEng**	
Surname	Kamperis
First Name(s)	Oliver
ID number	1332412
Supervisor's Name	Dr Marco Castellani
Project Title	Non-linear systems control using Fuzzy Logic expert controllers

Non-linear systems control using Fuzzy Logic expert controllers

Oliver Kamperis
University of Birmingham

Abstract

Fuzzy logic control is commonly used in industry. Unfortunately, it is not customarily offered in academic curricula, and the availability of versatile fuzzy logic controllers for control engineering is poor.

A prototype Java application has been produced embodying the concept of experiential learning, to facilitate the education of control theory and fuzzy logic control systems. The application uses the inverted pendulum on a cart system to convey the difficulty of achieving high performance control on highly non-linear and inherently unstable systems. The design focuses on a visual representation of the system behaviour to create a user-friendly intuitive learning environment. The application allows the learner to change a large array of the physical system parameters, as well as the control and simulation parameters, to allow wide experimentation and foster deep understanding of the effects of these parameters on the system stability, and thus promote learning through reflection on doing.

In addition, a prototype C++ fuzzy logic controller API has been produced focusing on design simplicity, transparency, versatility and expandability. The implementation is based on a control engineering perspective and thus is designed to be application independent. It provides the operator with many proven effective utilities, including rule modules and dynamic importance degrees which increase the controller reconfigurability. The original contribution of Resolution Hedges is introduced to ease of the controller tuning process.

Keywords: Fuzzy logic expert control systems, Experiential learning, Fuzzy control application programming interface, Non-linear systems control, Control engineering.

Table of contents

1. Introduction 1
2. Literature review 1
3. Objectives 3
4. Educational Java application based on the inverted pendulum on a cart system 4
   4.1. Learning objectives 4
   4.2. The inverted pendulum on a cart system 4
      4.2.1. Included system properties 5
   4.3. The learning procedure 5
   4.4. Design to meet learning objectives 6
      4.4.1. The simulation view 6
      4.4.2. The results view 11
      4.4.3. The control parameter view 11
      4.4.4. The options view 11
5. Fuzzy logic controller C++ API 15
   5.1. Code design 15
   5.2. Input variables 15
   5.3. Membership functions 16
   5.4. Resolution hedges 16
   5.5. Rules 18
   5.6. Rule modules and dynamic importance degrees 18
6. Experimental tests 19
7. Discussion 21
   7.1. Java educational application 21
   7.2. Fuzzy logic controller C++ API 22
   7.3. Resolution hedges 22
   7.4. Experimental observations 23
   7.5. Comments from the author 24
8. Conclusions 24
9. References 25
10. Appendix 29
   10.1. Acronyms 29
   10.2. Definitions 29
   10.3. Equations 31
   10.4. Tables 33
   10.5. Figures 35

10.6. Java application simulation model and controller implementation............................38
10.6.1. Model assumptions...38
10.6.2. PID controller...38
10.6.3. Fuzzy logic controller...38
10.7. Model and controller setup for experimentation 40

1. Introduction

Control theory is concerned with the automatic control of real world systems[1]. The most potent motivation for its existence and continuing development is in the automatization of processes to allow the replacement of human operators with automatic controllers[1,2]. Ultimately, this provides human beings with a means to gain liberty, freedom, and quality of life[2].

All modern automated systems employ control theory in some form. Washing machines are an example of one such system. With their invention people no longer had to labour intensively over a menial task, and their commercial success is an indicator of the human desire for an easier life through automation.

In the modern world automatic control has become far more complicated than in its inception and is now used in complex systems such as chemical plants, electric power stations, rocket flight and even spacecraft manoeuvring[2-4].

The improvement of control systems is therefore of great importance in human advancement.

Fuzzy logic (FL) expert systems are an example of a well-established automatic control technique which has gained wide industrial and commercial success[1,3-8]. Fuzzy control (FC) is excellent at controlling complex, highly non-linear and ill-defined systems[3,9,10]. This is thanks to its capability of replicating human behaviour by emulating approximate, but reliable human reasoning[11-15]. It does so by allowing experts to encode their control knowledge into human-friendly linguistic rules that can be processed efficiently by a computer[3,10,16,17].

2. Literature review

Fuzzy set theory [18] and FL [19] were proposed by Zadeh in 1965 and he has done significant work in progressing the field since[11-14,20,21]. It was Mamdani [16] who was the first to test the performance of FC on a model steam engine. Mamdani's style fuzzy inference system (FIS) uses fuzzy sets to specify rule consequents. This makes the system intuitive at the cost of increased computational complexity as defuzzification is required. Takagi–Sugeno [22] proposed a FIS which specifies rule consequences as finite quantitative values, requiring no defuzzification. The fuzzy controller output is instead calculated using a weighted average method. Many stability analysis and controller synthesis techniques exist for the Takagi-Seguno FIS[23,24]. Practitioners choosing Takagi-Sugeno FIS favour computational efficiency[17,25-27]. Whereas those choosing Mamdani FIS favour the intuitiveness of the controller[28-31].

King and Mamdani [8], Sugeno [32] and Jamshidi et al. [33] present many industrial applications of FC. Several authors [34-38] show that FC preforms well for industrial applications such as cement kilns, sludge wastewater treatment and urban storm drainage. These authors state that FC is the best knowledge based expert system for the control of highly non-linear dynamic systems due to its high tolerance to uncertainties[7].

Several authors [17,25-31] proved that various fuzzy logic controller (FLC) designs can adequately control the inverted pendulum on a cart system to some degree. However, they all studied ideal systems, ignoring undesirable physical properties such as friction and drag which commonly degrade control schemes in the real world.

Mendel [5] provides a comprehensive overview of FL literature necessary to synthesise a standard FLC.

Roosea et al. [17] use the sector non-linearity approach to the FLC rule base design. This approach guarantees stability. However, it becomes less practical as more properties that increase the complexity of the system dynamics are included in the simulation.

Yubazaki [25] uses a Single Input Rule Module (SIRM) dynamically connected FIS. Each input to the controller has a SIRM and a dynamic importance degree. The fuzzy controller automatically activates and deactivates modules by varying their dynamic importance degrees. This is used to switch between pendulum angle and cart position control. This allows the control scheme to change according to the operating conditions of the system. This scheme produces a simple and understandable fuzzy controller with high generalization and a much-reduced number of rules, without compromising on control performance. Radhamohan et al. [29] use a similar concept to Yubazaki's, achieving excellent control performance using "switches" on sets of rules. These switches can vary the control scheme between the swing-up and stabilization phases of control. Due to the changing dynamics of the system, each of these phases require a very different control strategy.

Liu et al. [39] discuss linguistic hedges for use in FLC. These are operators that dynamically adjust the shape of membership functions. They can potentially help to reduce the need to laboriously tune the shape of membership functions and provide a more intuitive interface for rule specification.

Roosea et al. [17] and Nour et al. [26] compare FC with PID control. They conclude that FC provides improved stability by reducing overshooting and settling time and is therefore more robust to physical parameter changes.

Ross covers FC in detail in his book "Fuzzy logic with engineering applications"[3]. He presents a strong case for the use of FL for control applications, stating its successful use in many consumer products. He however also states that despite having been accepted in the control community for quite some time and having great success in the marketplace, FC is still poorly taught in academia.

Mamdani [7] explains that FC has only been adopted in the western world within the last few decades. He believes this to because of a "cult of analyticity within control engineering research". This may be a cause of the lack of FC education in academia.

Beceriklia and Celikb [30] create a simple educational Java application demonstrating the fuzzy control of the inverted pendulum on a cart system. The application allows a small number of physical parameters to be changed, does not consider many of the inverted pendulum system's undesirable physical properties and does not attempt to solve the problem of returning the cart's position to the track centre. This results in a limited learning scope.

The MATLAB SIMULINK fuzzy logic toolbox [40] provides a simple application programming interface (API) to allow creation of a fuzzy inference engine for general dynamic systems. It provides a great number of features that are commonly used in fuzzy control. Several authors successfully use the toolbox to create a mathematical model to simulate the fuzzy control of the inverted pendulum system[17,26,29,31]. This suggests that the toolbox is strong for mathematical verification of control schemes. MATLAB also provides an educational GUI simulating the FL and PID control of the water level in a tank. The MATLAB platform can compile to C-code on both Windows and UNIX systems, but requires a paid license.

Juan Rada-Vilela's FuzzyLite [41] is a freeware FC library available for C++ and Java. The design provides a natural language processor capable of interpreting a FL control scheme specified in English language and converting it into C++ code. FuzzyLite appears mostly geared to artificial intelligence practitioners rather than control engineers. FuzzyLite includes educational features however, this aspect of the software requires a paid license.

3. Objectives

FC has existed as a distinct discipline and has been successfully used in many engineering applications for decades. However, the education of the subject is still not a standard offering in academia and there is a distinct lack of educational tools for the discipline[1,3].

Experiential learning applies well to control engineering, since deep understanding demands significant experience[42]. Unfortunately, experimental rigs are expensive and rarely available[1]. There is therefore great advantage in testing control schemes on computer simulations. Simulations are a cheap alternative for students to experience and understand intuitively the many difficulties control engineers must overcome when developing real world control schemes.

The first aim of this project is to create an application made on a portable, free and widely available platform that facilitates the education of control theory and FC. The goal is to create a realistic simulation which accurately represents the behaviour of a highly non-linear and inherently unstable system and provides an intuitive visual learning environment. The student should be able to change a large array of the physical system's parameters as well as the control and simulation parameters. This will allow the student to study their effects on the system's stability and thus learn through reflection on doing[42].

The second aim of this project is to create a C++ FLC API for engineering applications. The reviewed literature shows excellent performance of FC for complex dynamic systems. Yet there is an absence of versatile FLCs focusing on a control engineering perspective. The creation of such controllers would aid the incorporation of fuzzy control into a wider range of engineering applications.

4. Educational Java application based on the inverted pendulum on a cart system

A prototype educational Java application has been produced to favour experiential learning of FL and PID control on a simulation of the inverted pendulum on a cart. Java was used to create the application due to its high portability. The Java Virtual Machine is available freeware for all modern operating systems[43].

The simulation runs in “real time”. Consequently, it can be viewed whilst it is running, rather than being calculated beforehand and viewed afterwards. This kind of simulation is less mathematically precise but more effective for educational purposes. It allows immediate observation and understanding of the behaviour of the system.

The application is available freeware at: https://github.com/OllieKampo/Educational-Java-application-based-on-the-inverted-pendulum-on-a-cart-system

4.1. Learning objectives

The application was developed with the following learning objectives in mind:

1. Understand that the cart and pendulum system exhibit complex dynamics that are difficult to predict precisely and requires robust control schemes.
2. Understand that the two control goals of the cart and pendulum system may demand conflicting actions and formulate some strategy to overcome this problem.
3. Understand the fundamental differences between FL and PID control systems.
4. Critically analyse the control performance and tolerance to discontinuities, non-linearities, steady state error, imprecision and changes in physical parameters of both control systems.
5. Determine the advantages and limitations of both control systems.

4.2. The inverted pendulum on a cart system

The system consists of an inverted pendulum with an end weight attached, mounted on the centre top of a wheeled cart. The cart is fixed to a track such that its range of movement is limited by the bounds of the track. The system is highly non-linear, inherently unstable, simply structured and easy to understand. It is widely regarded as a standard benchmark for testing the effectiveness of control systems[17,25].

There are two control goals that must be achieved simultaneously; 1) balance the pendulum in the upright position, 2) return the cart to the centre position of the track[17,25,26,28-31]. This is a very difficult and intellectually stimulating problem to solve. The movement required to move the cart to the track centre will destabilize the pendulum. To restabilize the pendulum the cart must have a force applied in the opposite direction to movement. Thus, the two control goals are in opposition to each other.

The system exhibits many common control scheme degrading properties including discontinuities, non-linearities, frictional and inertial losses, hysteresis, backlash and slip-stick motion[1]. Its dynamics also apply to many other real-world systems of great interest such as rocket flight and legged robot balance[17]. The system is therefore excellent for educational purposes and has a broad application scope.

4.2.1. Included system properties

To increase realism and provide a challenging learning environment, the model includes many control degrading properties. These properties are described below:

The length of the track is bounded. Collision with the end of the track creates discontinuity in the system dynamics, requiring a very sudden and large change in the control action required to return the system to a stable state. Boundedness is an inherently present feature of most real world dynamic systems, and thus is important for the student to understand[10].

Random sensor input and motor output errors are included to make the system stochastic and inherently imprecise. All real sensing and actuating devices are imprecise to some degree. Small reductions in precision required by control algorithms result in large reductions in computational complexity[3]. It is therefore important that the student understands that realistic control systems must be highly tolerant to imprecision[1].

Rolling resistance is included as a source of discontinuous steady-state error. The rolling resistance force exists only when the cart is moving and is always in the direction of the cart velocity. Rolling resistance is predictable but not necessarily easily dealt with. Similar sources of steady-state error are very common in the real world and it is important that the learner is exposed to this issue[1].

Aerodynamic drag is included as an example of non-linear a phenomenon whose magnitude varies continuously and unpredictably with time. Understanding of drag in control problems is important as it applies to almost all dynamic systems operating within an atmosphere.

4.3. The learning procedure

There is rarely an immediately obvious way to control most real world dynamic systems. This is true also for the cart and pendulum system[1]. The student is encouraged to take a trial and error approach to learning[42].

When attempting to find a set of suitable control parameters the student should:

1. Run the simulation to observe the behaviour of the system.
2. Analyse the performance of the current control scheme.
3. Interpret, based on the above analysis, what parameters might need changing to produce a more effective control scheme.
4. Change the parameters and observe the simulation again.
5. Determine how successful or otherwise the changes were.
6. Draw new conclusions on the behaviour of the system to make more informed changes to the control scheme.
7. Continue until a suitable control scheme is found.

This technique is based on the concept of unsupervised experiential learning[42]. At every iteration of this cycle, the student learns through reflection on doing. Over time they will develop a deep understanding of the control problem and ultimately achieve the learning objectives.

When a student finds a suitable control strategy for each controller, they should begin altering some of the pre-set values for the physical and simulation parameters listed in Table 2 and 3 respectively. This allows the student to gain an understanding of how changes in each parameter affect the behaviour of the system and analyse the tolerance of each controller to such changes. The student will learn that small changes in just a few parameters will likely require large changes in the current control strategy to suit the new conditions.

4.4. Design to meet learning objectives

The design of the application employs the concepts of human computer interaction (HCI) to create a user-friendly and intuitive graphical educational environment that meets the capabilities and learning requirements outlined above.

Several learning aids are provided to allow the student to understand and analyse the system, whilst hiding any irrelevant information which could confuse the student and degrade educational effectiveness.

All aspects are designed to be "fool proof" and user friendly, user error is removed by pre-checking all attempted parameter changes. The result of any change, successful or not, is printed to the console. When erroneous inputs are detected they are marked in red and default values are applied to prevent system crashes (Figure 1).

The application consists of 4 main views; the simulation view, results view, control parameters view and options view. Each of these views split the application up into a structured fashion, organising its parts based on their function, to create an orderly learning environment.

4.4.1. The simulation view

The main section of the simulation view displays a large 2D view of the cart and pendulum, providing a clear visual representation of the system. The current magnitude of the tractive force output from the selected controller is displayed as a numeric value and with an arrow, indicating the direction and magnitude (length and colour) of the force.

Two graphs are displayed alongside the cart and pendulum: one showing the pendulum angle against cart position, representing the current error of both control goals; and the other displaying the pendulum angle against time, representing the response of the controller to what is normally considered the priority goal[17].

Two separate simulations of the cart and pendulum system run concurrently, one controlled by a FLC and the other by a PID controller. The student may switch between the two shown in Figures 2 and 3 respectively by selecting the relevant tab at the top of the view.

When the FLC is selected, graphical plots show the degree of membership of each input variable within each membership function. This makes it easy to determine what rules are currently firing and whether they might need modification.

When the PID controller is selected, the forces generated by each individual term are displayed in a table, the colour of which changes according to the magnitude. This facilitates easier tuning by showing what proportion of the total output is being provided by each term, something that is normally quite difficult to determine.

The student can vary the “run time scaling” of the simulation. This allows the learner to slow down the simulation speed such that they can better study the system behaviour. The student may also restart the simulation from the last used start point, or switch to the results view to observe the system behaviour again.

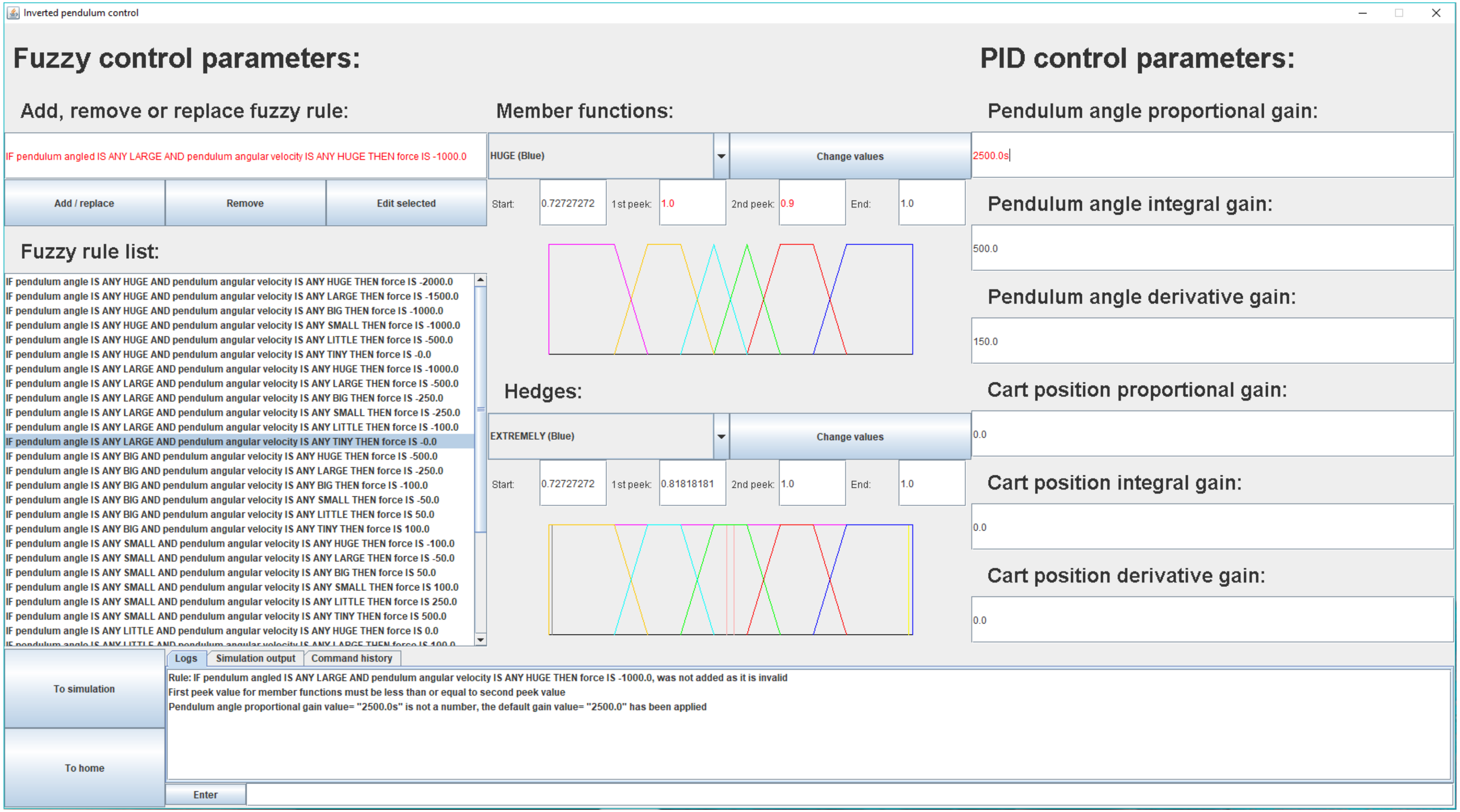

Figure 1: Example of erroneous input detection and parameter change description printing to console in the control parameter view

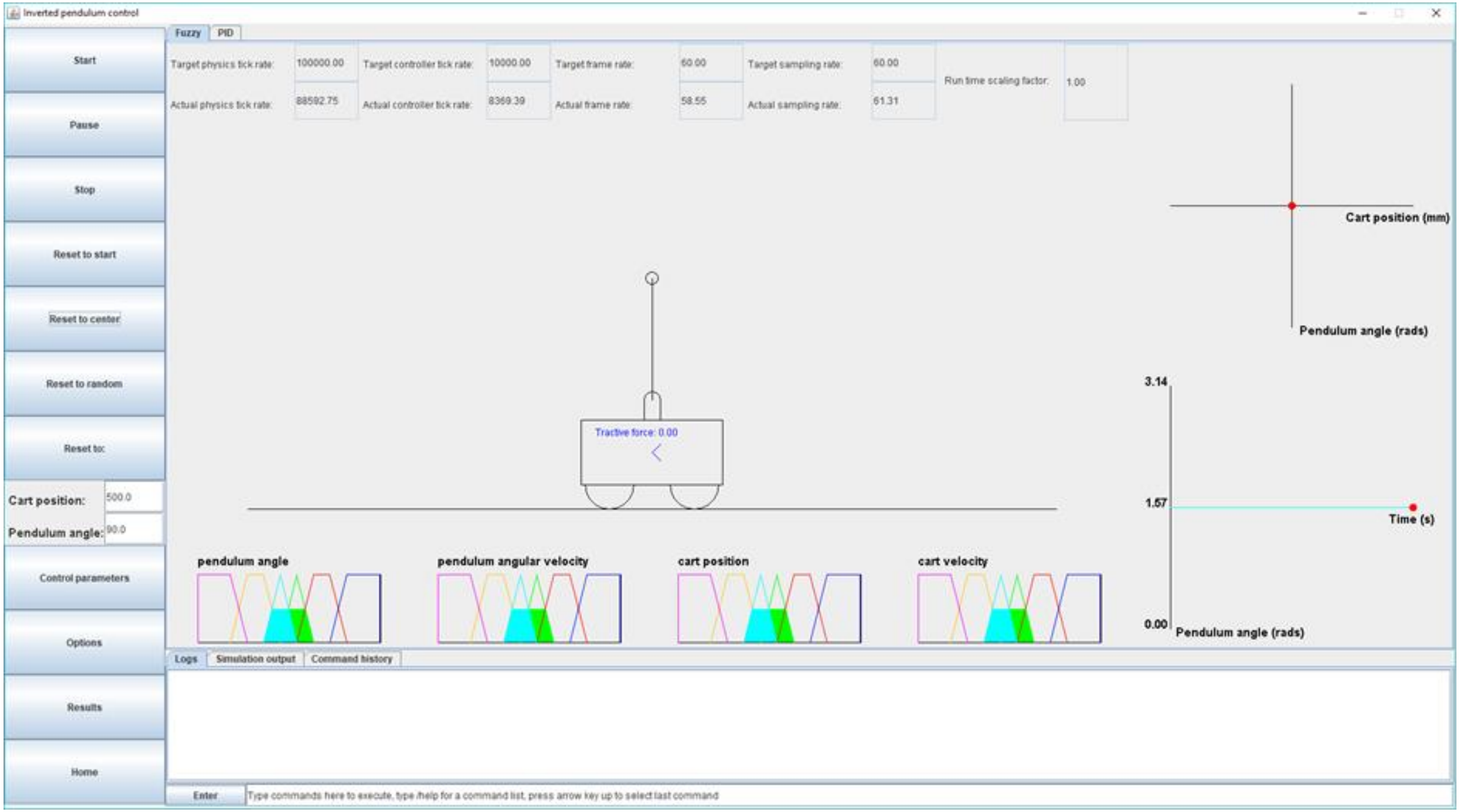

Figure 2: Fuzzy logic controller simulation view

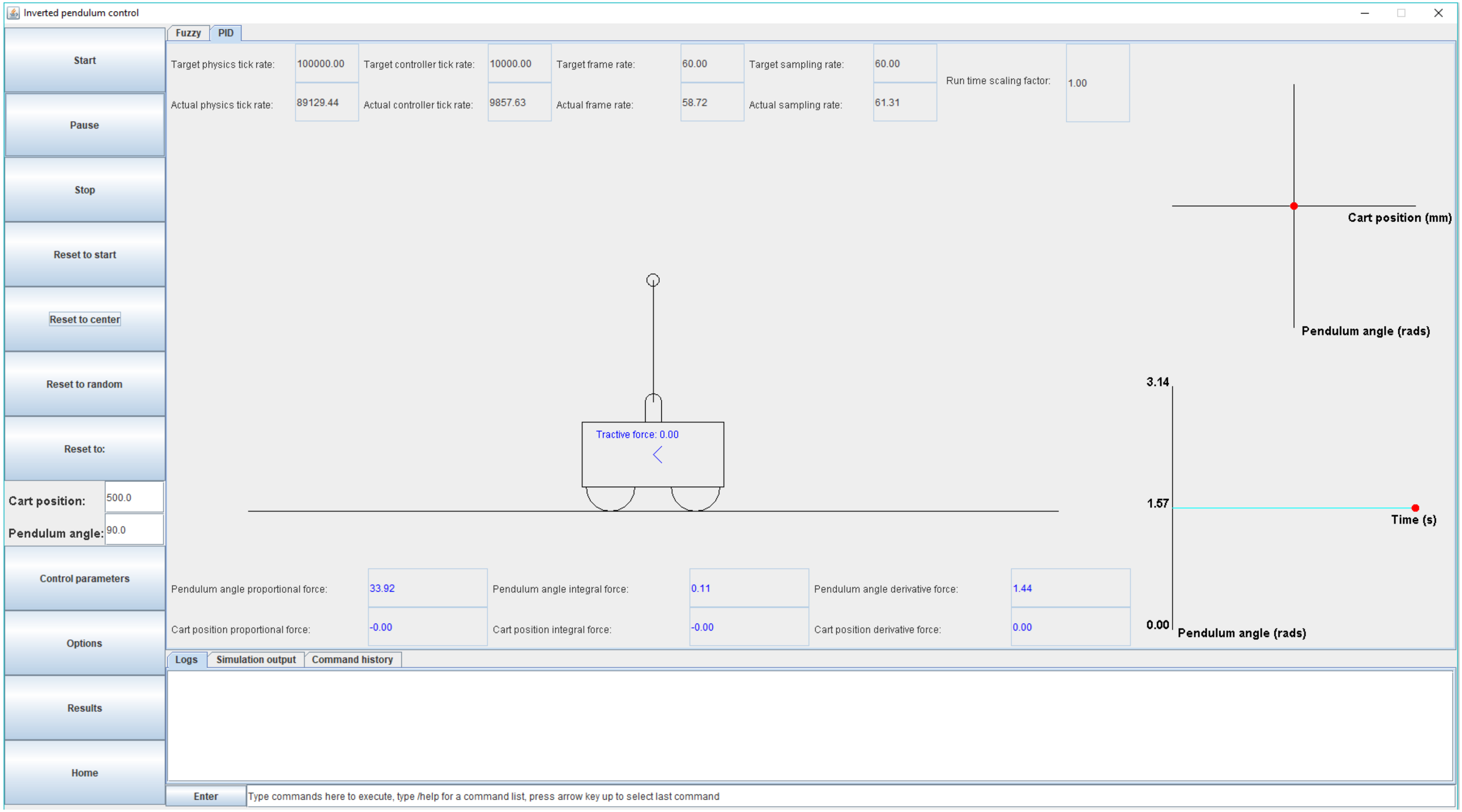

Figure 3: PID controller simulation view

4.4.2. The results view

Whilst the simulation is running results are automatically recorded. Afterwards results may be played back in the view shown in Figure 4. Four graphs allow visualisation of the difference in performance between the two controllers in terms of pendulum angle, pendulum angular velocity, cart position and cart velocity against time. Results can be played forwards, rewound, paused or the user can jump to any time point via the use of a slide bar.

4.4.3. The control parameter view

In this view (Figure 5) the student may change the control parameters. For the PID controller this involves only the specification of 6 gain values, one for each of the three PID terms for the two control goals. For the FLC more functionality is provided.

The student may edit the rules, membership functions, and hedges to study how their modification changes the behaviour of the controller. There are 6 membership functions, 9 hedges, 4 aggregate connectives and the NOT operator that can be used to specify rule antecedents, these are listed in Tables 4, 5 and 6 respectively in appendix 10.6.3. Bracketed sets of statements are not possible. A list of all fuzzy rules that are currently in the rule-base is clearly displayed to the user.

A rule-from-string interpreter is included, which makes it easy to add and edit rules by converting typed linguistic rules such as:

‘IF pendulum angle IS VERY LARGE AND pendulum angular velocity IS ANY SMALL THEN force IS 500.0’

Into a form that can be processed efficiently by the computer, thus providing an intuitive, user friendly interface abstracted from the underlying implementation. The rule interpreter notifies the user as to whether a rule is incorrect before allowing it into the list to prevent processing errors.

4.4.4. The options view

From the options view, shown in Figure 6, the student may change any of the physical and simulation parameters listed in Tables 2 and 3 respectively.

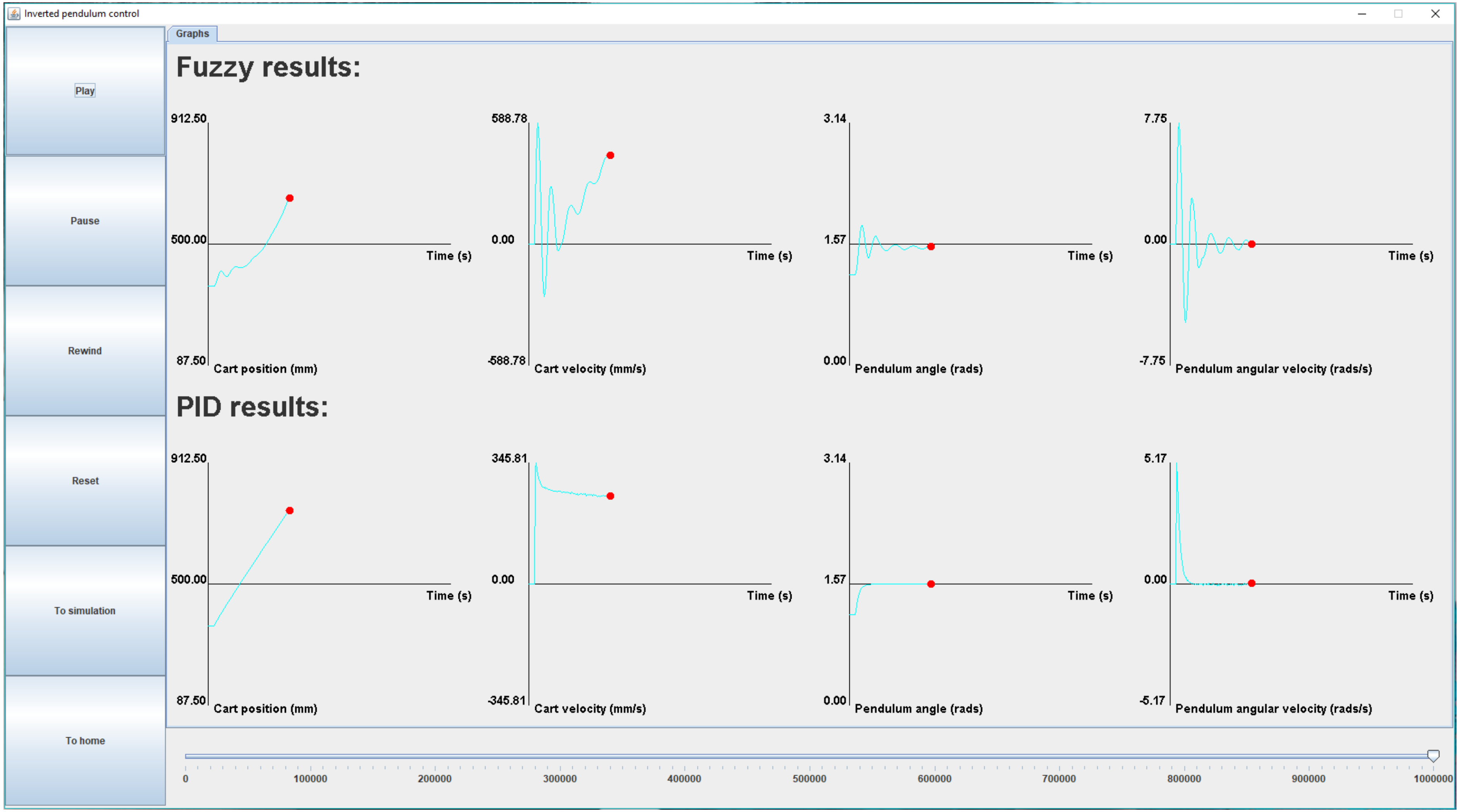

Figure 4: Results view

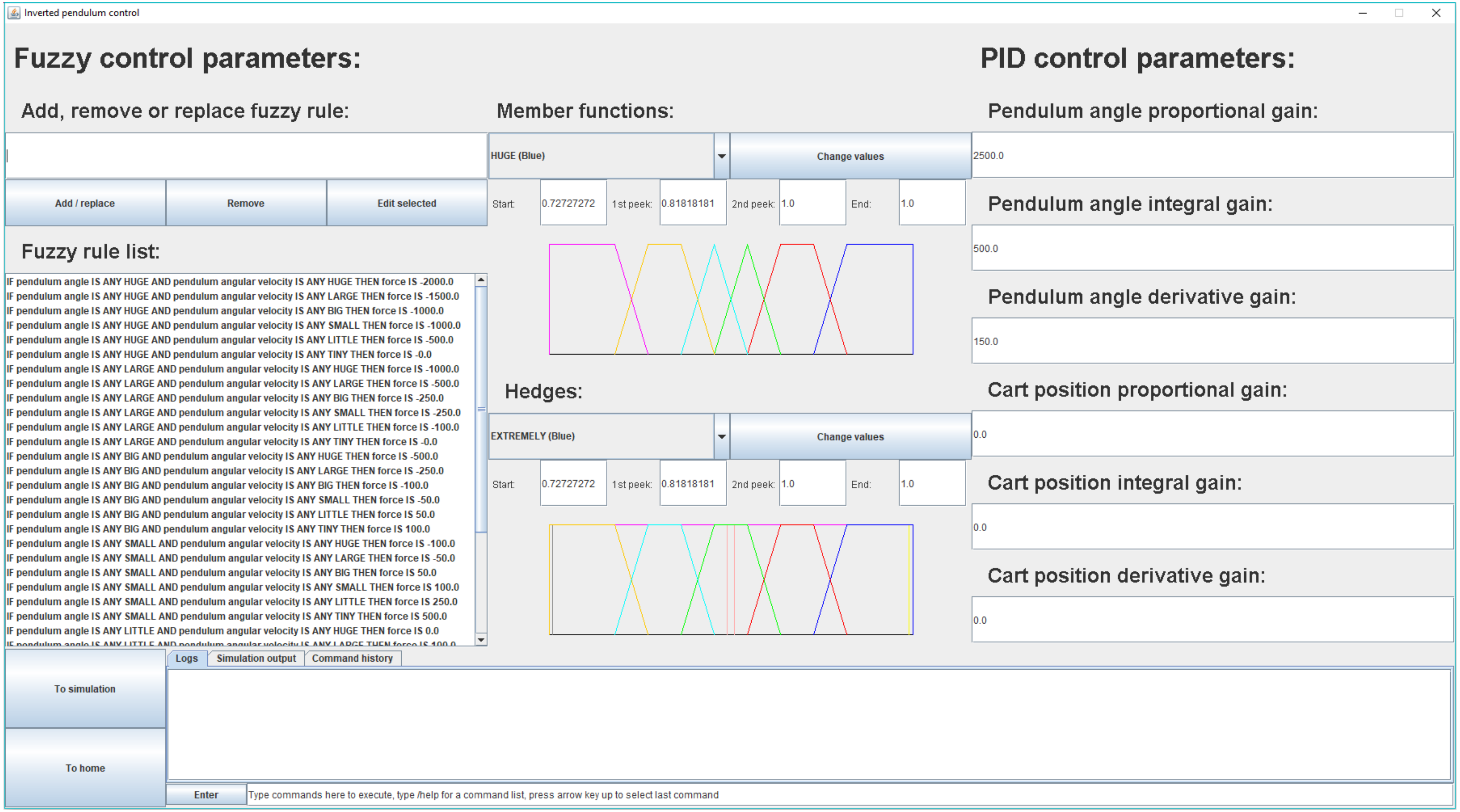

Figure 5: Control parameters view

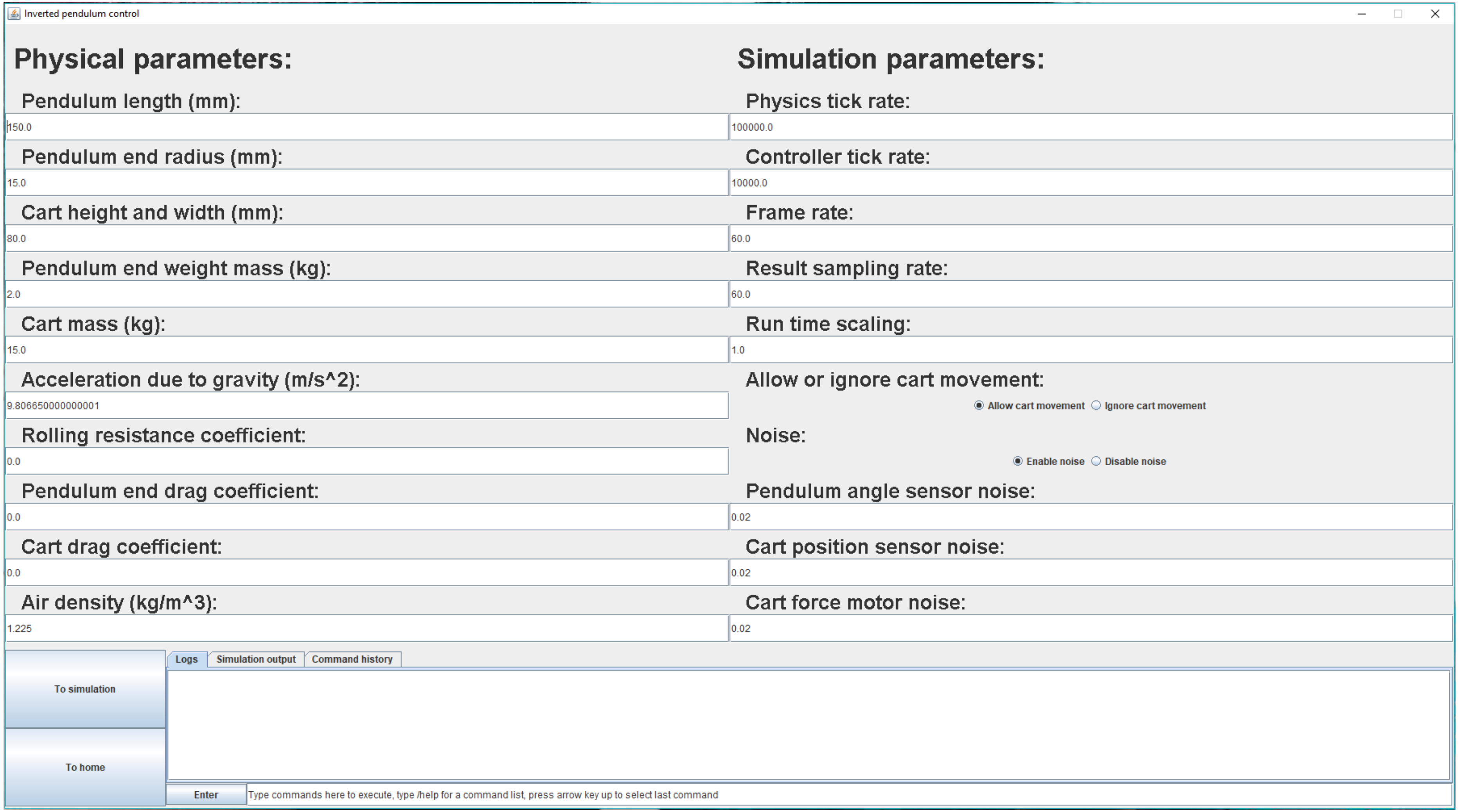

Figure 6: Options view

5. Fuzzy logic controller C++ API

A prototype FLC API has been produced based on a control engineering perspective, focusing on simplicity, transparency, versatility, expandability and application independency. It provides the operator with many features which allow the controller to be adapted to control any system with minimal limitation.

The API is available freeware at: https://github.com/OllieKampo/Fuzzy-logic-controller-C-API

Subsections 5.5. and 5.6. introduce additional non-standard features to the fuzzy controller. Subsection 5.5. introduces resolution hedges, an original contribution. Subsection 5.6. introduces rule modules and dynamic importance degrees from the literature.

5.1. Code design

The design extensively employs templating, inheritance and function virtualisation to allow the controller to be implemented with ease and provide great expandability to its functions. The controller is one single templated abstract class containing several other inner classes that define the input variables, membership functions, hedges and rules. The template requires one parameter taking the class type of the system, allowing the controller to be applied to any C++ class.

The controller design asserts that only one output variable may be specified. Allowing multiple output variables significantly increases the API's design complexity, yet a controller with two outputs is functionally identical to two controllers with one output each[3,9].

5.2. Input variables

Three standard types of input variable are available to the operator:

1. Proportional
2. Integral
3. Derivative

These variable types are based on the well-known PID controller concept and are the most commonly used in control[1,4,45]. The operator may also define their own type of input variable if required for controllers using concepts such as sector non-linearity[17]. All input variables allow an input gain to be applied and require the input bounds to be defined.

The proportional variable returns a value directly proportional to the plant error, where the error is the input itself. It generally serves as the main driver of the controller[4]. The integral variable returns a value proportional to the magnitude and duration of the error[45]. It serves to drive steady-state error to zero[4]. The integral input gain defines the integral ramping factor which determines the rate of error accumulation during windup. The derivative variable returns a value proportional to the rate of change of the error[45]. It serves to allow damping of the system[4]. The derivative input gain defines the damping factor.

Integral and derivative terms are estimated automatically for the user based on the change in input and time between each control time-step. It is therefore important to note that the shorter the time-step, the more accurate the integral and derivative estimates will be[1].

5.3. Membership functions

Six standard types of membership function that are most commonly used in fuzzy control are available to the operator: [3,5]

1. Triangular
2. Trapezoidal
3. Rectangular
4. Piecewise linear
5. Singleton
6. Gaussian

The operator may directly instantiate any of the above membership functions or may define their own custom membership functions. Figures 11 to 16 represent the above membership functions in respective order.

Two non-standard membership functions; "Max saturated" and "Min saturated" are also available to the operator. They may be used to specify rules that determine an output in the case where an input variable is saturated above or below the maxima or minima of its bounds respectively. Fuzzy controllers usually perform poorly in the presence of unboundedness[10]. When saturation occurs, it can cause a very sudden, and very large, change in the output of the controller which may lead to extreme instability. This happens particularly for integral or derivative variables when their bounds are uncertain or non-existent[10].

5.4. Resolution hedges

Fuzzy hedges are modifiers which change the shape of membership functions to allow more precise definition of rule antecedents[21,46]. Zadeh's original style of hedge allows the boundaries of a fuzzy set to be manipulated such that the degree of membership of elements that are only partially in the set can be defined more distinctly[10,18,19]. Whilst hedges of this type can be useful for some applications, they have little meaning in control theory[47] and are never used[17,25-31].

A new style of hedge is suggested for control. This hedge describes where in the range of a membership function the input lies. It defines fuzzy sub-sets of the fuzzy super-sets defined by the membership functions. This effectively splits the membership functions in many, smaller functions. Thus, it increases the total number of available fuzzy sets and therefore also the maximum number of possible unique rules. Ultimately, this allows a finer resolution in the specification of the control strategy across the system's bounds, hence this new type of hedge will be called a "Resolution Hedge". The modified shape of the fuzzy set created by the aggregation of a membership function and a resolution hedge (Figure 7) is simply the intersection of the two (Figure 8). The concept of resolution hedges is concerned with making it more feasible for humans to define control schemes by simplifying the tuning process. It provides the ability to rapidly modify the shape of membership functions without having to change the membership function parameters themselves.

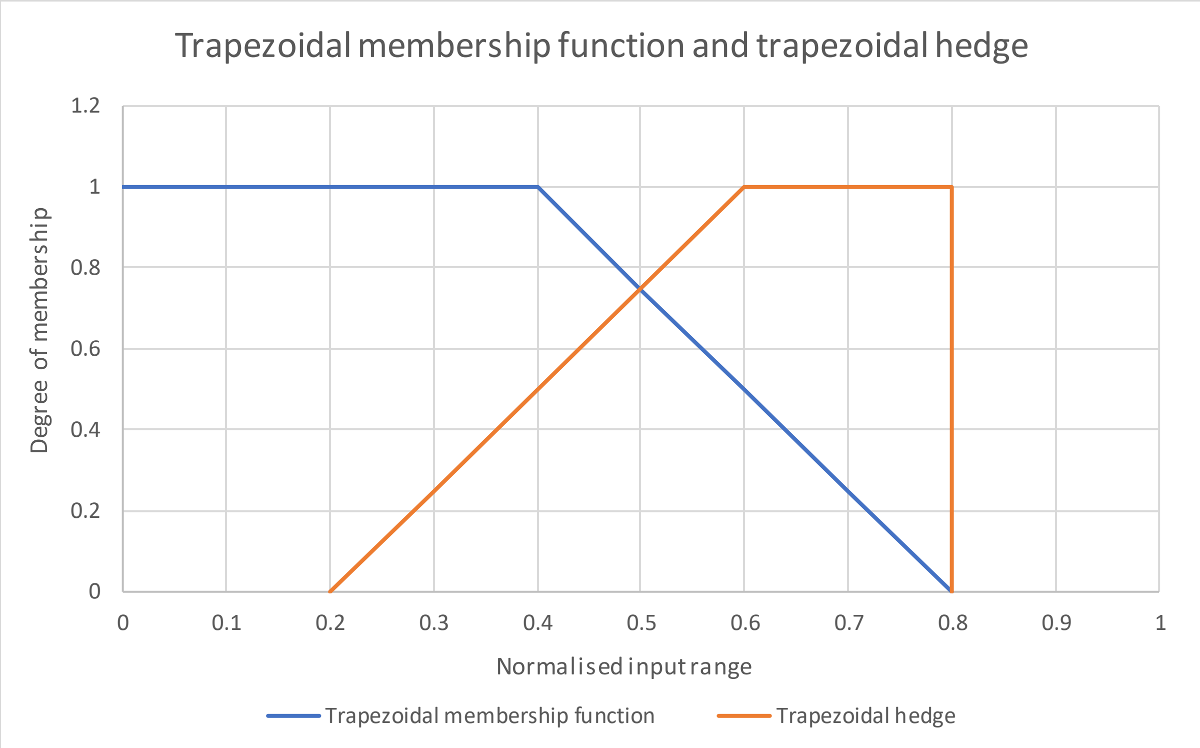

Figure 7: Trapezoidal membership function and trapezoidal hedge

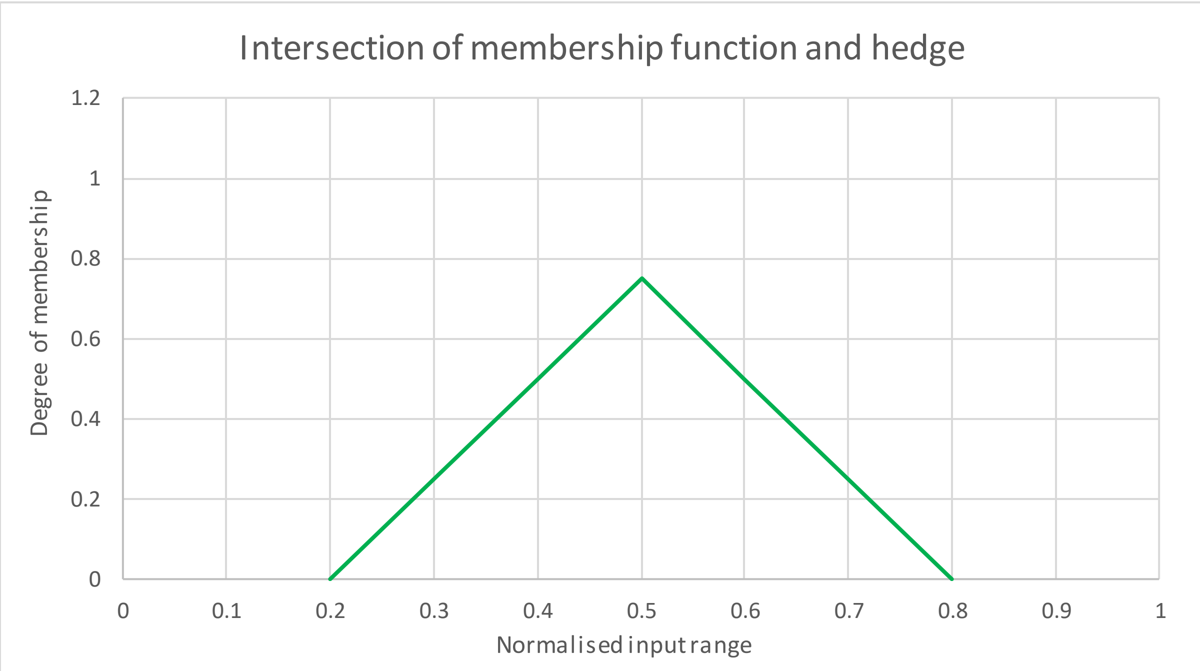

Figure 8: Intersection of trapezoidal membership function and trapezoidal hedge

Resolution hedges are universal to all membership functions, i.e. every membership function has the same set of hedges, as such they are only defined once. Consequently, many rules can be obtained from few membership functions and resolution hedges. Resolution hedges do not have to be included in every linguistic term. This means that the minimum required number of rules is not increased. Equations 1 and 2 show the minimum and maximum number of required rules respectively. To preserve computational efficiency when a hedge is not specified, it is not evaluated.

5.5. Rules

The controller uses a standard Takagi-Seguno rule-based FIS[22]. This was chosen because of its computational efficiency and design ease when compared with Mamdani FIS[9,10,16,23,44]. Rules are specified linguistically, however the inference engine preforms only numerical computation. This creates an intuitive interface to an efficient fuzzy inference engine[14]. The API does not assert that all possible rules must be defined but it does ensure that the same rule antecedent does not exist twice.

Rule antecedents allow many individual statements to be aggregated using the AND connective[44] and do not allow bracketed sets of statements. The most stringent and commonly used technique for creating a rule list is to define all possible unique rules[3]. Every rule antecedent must contain a statement for each input variable exactly once with only the AND connective between statements. All possible combinations of membership functions must then be specified. Equation 1 shows the number of rules required for this setup[9]. Stable fuzzy control of the inverted pendulum system is achieved by several authors with the above technique[17,28,30,31].

The standard rule evaluation technique requires every statement in every rule to be evaluated, the time complexity of this is shown in Equation 3. To save on computational time, the first statement in the rule antecedent is evaluated and the aggregate statements are evaluated only if the first statement has a non-zero degree of truth[9,10]. By doing this the time complexity of rule evaluation is reduced (Equation 4).

5.6. Rule modules and dynamic importance degrees

Rule modules are separated sets of rules, each rule module acts like a separate controller such that the control action of each module is calculated independently of the others[25]. The total output of a single rule module may be calculated as the weighted sum or the weighted average of the degrees of activation of the fired rules. The total controller output may then also be calculated as the weighted sum or the weighted average of the output of the modules, where the weights are defined by the module output gains. To improve computational efficiency, rule modules of zero gain are not evaluated. Modules help to reduce the number of rules required for control. They help to negate the dilution effect caused by multiple separate sets of rules firing simultaneously[25]. This is particularly useful for systems that require many input variables and membership functions, or when multiple control goals are to be achieved simultaneously. The reduction in the number of rules significantly improves transparency, reduces control scheme production times and improves computational efficiency[3].

Dynamic importance degrees are also included. These may be described as “continuously variable switches” that allow the output gain of rule modules to be dynamically varied based on the operating region of a priority variable[25]. Thus, the control scheme can be continuously changed based on the operating conditions of the system. This can help to deal with non-linearities, discontinuities and multiple control goals. To use dynamic importance degrees, an additional “dynamic importance rule module” is assigned to each individual standard rule module. The output of the dynamic module defines the gain of its standard rule module.

Rule modules with dynamic importance degrees were used to reduce the number of rules required to control the inverted pendulum on a cart system from 81 to just 21, without compromising control stability[25].

6. Experimental tests

A simulation of the inverted pendulum on a cart system with unbounded track length was used to test the effectiveness of integral and derivative input variables, saturation membership functions and resolution hedges. Equations 5 and 6 define the dynamics of the system. The following physical parameters were used for the experimentation:

- Cart mass: 1 Kilogram
- Pendulum length: 1 Meter
- Starting pendulum angle: 45 Degrees
- Pendulum bounds: 150.0 - 30.0 Degrees
- Steady state error: -10 Newtons

The list of input variables, membership functions, resolution hedges and rules used for the experimentation are detailed in appendix 10.7.

Figure 9 shows results of the fuzzy stabilisation control of the inverted pendulum to test integral and derivative input variables using control schemes incorporating P, PD and PID terms. With just a proportional term, low stability and highly oscillatory behaviour exists. The inclusion of a derivative term eliminates overshoot to provide a large improvement in stability. The inclusion of an integral term removes steady state error. The use of saturation membership functions has produced a continuous output from the integral and derivative input terms without laboriously tuning their input bounds. This verifies the effectiveness of integral and derivative input variables and saturation membership functions for the control of a simple system.

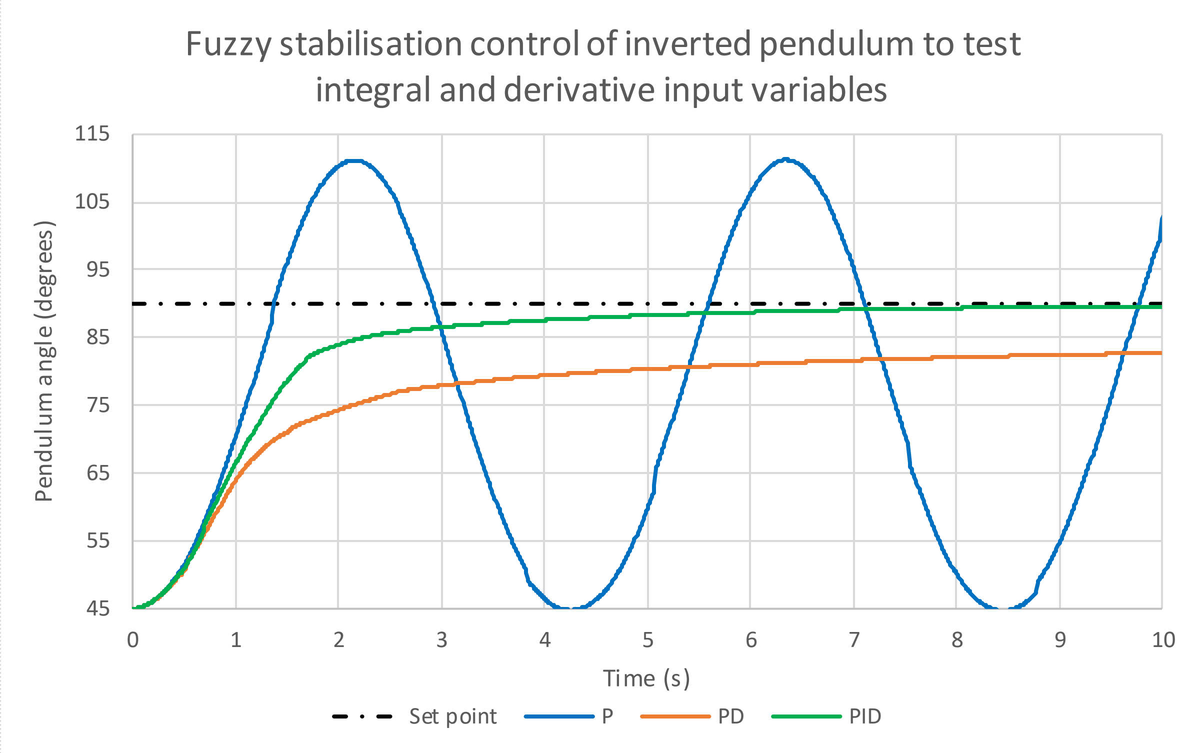

Figure 9: Fuzzy stabilisation control of inverted pendulum to test integral and derivative input variables

The rules that govern the proportional response to the pendulum angle where changed to see how the inclusion of resolution hedges affect the control performance. Six additional rules were added in this process.

Figure 10 shows results of the fuzzy stabilisation control of the inverted pendulum with and without resolution hedges. The graph shows that the scheme with resolution hedges provides a faster rise to the set point of 90 degrees, and no instability or overshoot is added. Table 1 shows the rise time of the same simulation, a noticeable decrease in rise time is evident with the inclusion of resolution hedges. The performance of the control scheme has been improved by increasing the number of rules from 22 to 28. Thus, resolution hedges improve the overall effectiveness of the control system at little cost.

Table 1: Rise time of fuzzy stabilisation control of inverted pendulum with and without resolution hedges

Percent rise time in seconds	**Without resolution hedges**	**With resolution hedges**
1.0% Rise time (s)	6.92	2.20
0.5% Rise time (s)	9.02	3.63
0.1% Rise time (s)	13.91	8.84
Absolute rise time (s)	34.91	29.31

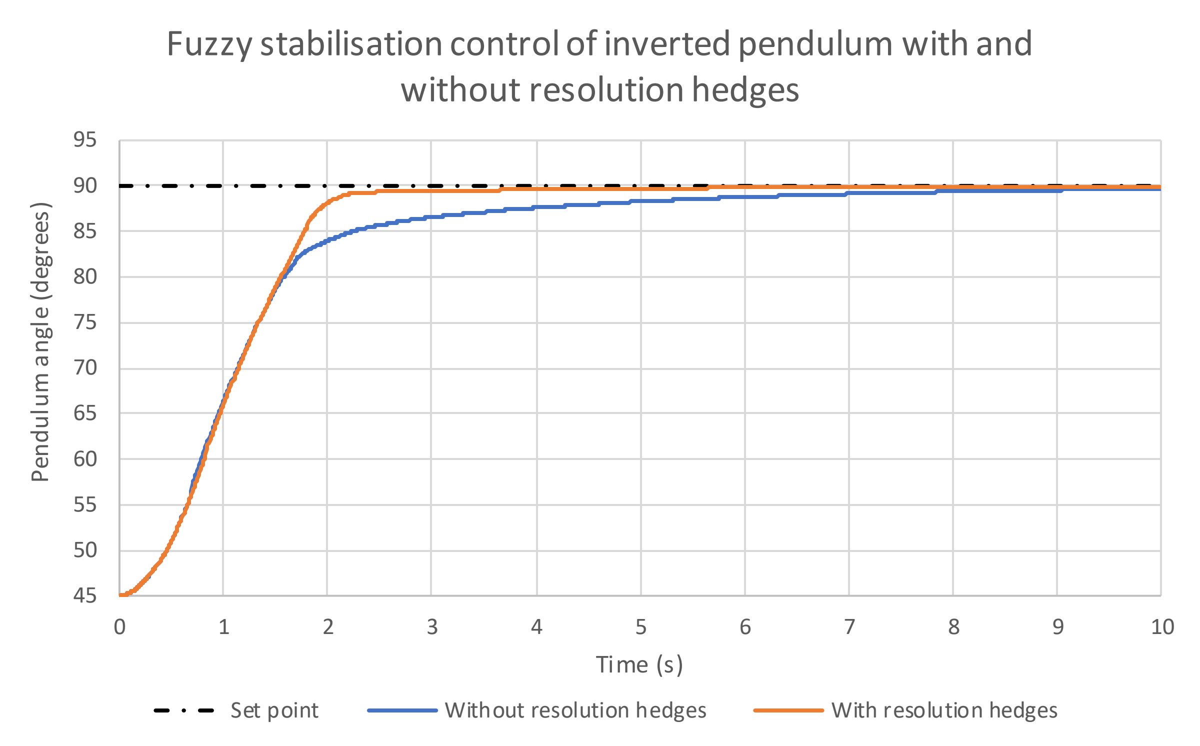

Figure 10: Fuzzy stabilisation control of inverted pendulum with and without resolution hedges

7. Discussion

The successfulness of the proposed applications and possible future improvements are discussed.

7.1. Java educational application

The design and learning scope of the proposed educational application is compared with other similar applications.

MATLAB's fuzzy control toolbox educational GUI [40] is confusing and not user friendly, it is not obvious what most buttons do or what the graphical displays are representing. Furthermore, it is not well structured, the location of functions are not obvious, resulting in poor HCI. Its learning scope is severely limited by the simplicity of the water tank system used in its simulation and the exclusion of control degrading properties.

Comparing MATLAB [40] to the proposed application, the structure has been improved by grouping functionality into ordered views. The function of buttons and meaning of displays is made more obvious to increase user-friendliness. The cart and pendulum system is better for educational purposes compared to the water tank because of its complexity created by the two conflicting control goals. This produces a far more challenging control problem and therefore a more rewarding learning experience.

The Java educational application made by Beceriklia and Celikb [30] is extremely simple. Whilst this makes it easy to use, it greatly limits the learning scope. It features a small display of the cart and pendulum, which is difficult to follow. The current error is displayed numerically rather than graphically. This is not intuitive and very difficult to read as the values change quickly. It allows only very few physical parameters and none of the control parameters to be changed. Consequently, there is little for the student to experiment with and the application is not intellectually stimulating.

Compared to Beceriklia's and Celikb's [30] application, the proposed application features a clearer display of the cart and pendulum system and several graphic displays instead of raw numeric outputs. This results in a more informative representation of the system. Furthermore, the proposed application allows more physical and control parameters to be changed. This facilitates the understanding of their effects and thus provides a much larger learning scope.

The most important improvement over the current state-of-the-art [30,40] is the inclusion in the proposed application of the cart and pendulum control degrading properties. These properties are inherently present in most real dynamic systems and are the most difficult issues that control engineers must overcome when designing control schemes. Understanding these properties is therefore of paramount importance in the learning of real world control systems. Simulations that exclude these properties trivialise the problem, giving the learner an oversimplified and unrealistic view of real world control problems.

By including control degrading properties, both FC and PID controllers and the complex cart and pendulum system, the broad learning objectives outlined in subsection 4.1. have been successfully fulfilled. Other applications are not capable of fulfilling these learning objectives.

7.2. Fuzzy logic controller C++ API

The proposed FLC API contains many advantageous design features from the literature. The effectiveness of differential and integral variables, saturation membership functions and resolution hedges in producing more stable and easy to tune control schemes have been verified in this study. Rule modules and dynamic importance degrees have been introduced based on compelling results from other studies [25,29] and should be further tested in the context of the proposed API. The use of additional features is up to the user, if they are unused, computational efficiency is preserved.

The only of the above features available in MATLAB's fuzzy logic toolbox [40] and FuzzyLite [41] is saturation membership functions. Although integral and derivative input terms are fundamental components of many control schemes, neither MATLAB, nor FuzzyLite provide inbuilt functionality to automatically estimate integral or derivative terms. Given the complexity of MATLAB's and FuzzyLite's implementations, inclusion of the other features would be problematic. The API made in this paper therefore contains a significant number of new and unique features which improve over MATLAB and FuzzyLite.

The API does not provide the ability to specify Mamdani style rule consequences. This feature is commonly used in control engineering and is to be added in a future version.

7.3. Resolution hedges

Zadeh style hedges have not commonly been used before in control. The original idea of resolution hedges is here suggested to provide a practical function for hedges in control and is an original contribution of this study. Resolution hedges employ the concept of fuzzy sub-sets, a field extensively covered in fuzzy set theory, but which is not used in fuzzy control applications[3,15]. The increase in performance achievable by preforming tuning with resolution hedges has been successfully demonstrated with an experimental simulation in which a small number of rules were used to control a simple system. This experimental simulation is however quite limited as the API is designed for real time control and cannot be considered adequate to provide stringent experimental verification. However, the results are compelling enough to encourage further investigation into the advantages of the proposed resolution hedges.

Resolution hedges have the potential to benefit fuzzy control by making control scheme design more human-friendly. Operators may perform rapid tuning by defining additional unique rules in regions where stability is particularly low, without having to change existing, or define new, membership functions. It can be argued that it is possible to replicate any control scheme that uses resolution hedges with a scheme using just standard membership functions, however, this results in a more complex fuzzy system. Therefore, resolution hedges increase the feasibility of humans designing control schemes. They provide better stability for simple systems without the need to employ optimisation algorithms.

The tendency of complex systems to require many rules results in reduced transparency[3,38]. In these situations, the use of resolution hedges would likely become less feasible. It becomes difficult for a human to interpret many rules in a meaningful enough way to successfully employ resolution hedges.

7.4. Experimental observations

Boundedness and discontinuities are particularly difficult to overcome problems that are inherently present in most real-world systems[1]. PID controllers are very poor in the presence of these as they provide no tools to solve them. FLCs can better deal with boundedness and discontinuities however in doing so the number of rules can become large. However, when system bounds are not well defined FLCs perform poorly, this is commonly the case in the real world[3].

The operating conditions of real world systems are liable to change, and two instances of the same system commonly exhibit different behaviours[1]. PID controllers do not deal well with changes in physical parameters, this is particularly bad for highly non-linear systems. FC is much more tolerant to such changes.

Most real-world systems of interest are non-linear[1]. PID controllers are based entirely on linear functions. Whilst they can deal with non-linearities to some extent it requires significant tuning. Conversely, FLCs can quite easily deal with non-linearities as a non-linear control output can be specified easily.

FC can deal with both symmetric and asymmetric systems. If a system is symmetric so should be the membership functions and rules. PID control cannot deal with asymmetry well because its functions are inherently symmetric.

The simplicity of PID controllers make them easy to use but also heavily limits their versatility. FC is superior to PID in terms of universality and versatility, it can control a much wider range of systems. FC provides a more intuitive specification of a control scheme. However, for highly complex systems, where many rules are needed the controller very quickly becomes less transparent and more difficult and time consuming to tune. Consequently, features that reduce the number of required rules are highly advantageous.

In general, FLCs are more robust than PID controllers. Only for simple systems does PID control preform adequately to warrant its use.

7.5. Comments from the author

Creating a simulation of the inverted pendulum on a cart system was greatly rewarding. It was not until I experienced trying to design a suitable control scheme for the cart and pendulum system on my own educational application that I was able to truly appreciate how difficult it was, and how important is to experience the problem first hand rather than just reading the theory. I've learnt a lot more from making and using my application than I have from the literature, therefore I believe that educational applications such as this are exceptionally useful to students.

The process of designing my own FLC has provided me with a unique perspective on FC. There is a great amount of literature discussing FC, however this provided me with little knowledge of how to code a FLC myself. I believe this is why so few fuzzy controllers are widely available. Therefore, I believe that the creation of this new publicly available freeware FLC is beneficial to the progression and learning of FC.

8. Conclusions

A Java educational application has been successfully produced which facilitates the improved education of control theory and FLCs. It provides an intuitive visual learning environment that significantly improves on other similar applications. The application includes many features that encourage trial and error based experiential learning. This provides a deep and meaningful understanding of the difficulties control engineers must overcome when designing control schemes for highly complex real-world systems. The application fulfils the specified learning objectives whose importance have been unlined in this paper.

A C++ fuzzy logic controller API has been successfully produced. Significant improvements have been made on other FLCs for control engineering purposes. The inclusion of multiple additional features such as differential and integral variables, saturation membership functions, resolution hedges, rule modules and dynamic importance degrees has increased transparency, versatility and universality. This allows the user to create a controller capable of overcoming a wider range of control problems and create higher performance control schemes with less limitations and minimal lead times. This is the first publicly available freeware fuzzy controller to include all these new features.

The study has provided the original contribution of resolution hedges to the field of FC. Resolution hedges give use to the concepts of linguistic hedges from FL and fuzzy sub-sets fuzzy set theory in FC. Compelling results have been obtained that show the effectiveness of resolution hedges. However, they should be tested further on more complex systems.

9. References

[1] J R Leigh. (2004). Control theory. 2nd edition. (IEE Control Series). Stevenage, Herts, UK: IEE, 2004. Pages 1-174 and 276-483.

[2] E. Fernandez-Cara and E. Zuazua. (2003). Control Theory: History, Mathematical Achievements and Perspectives. Available at: http://verso.mat.uam.es/web/ezuazua/documentos_public/archivos/personal/comites/ctsema.pdf (Accessed: 15 Jan 2018)

[3] Timothy J. Ross. (2010). Fuzzy logic with engineering applications. 3rd edition. Chichester: Wiley, 2010. Pages 1-29, 111-131, 167-195 and 225-285.

[4] Jacqueline Wilkie, Michael Johnson and Reza Katebi. (2002). Control engineering an introductory course. Houndmills, Basingstoke, Hampshire, England: Palgrave Publishers ltd, 2002. Pages 1-16 and 277-312.

[5] Jerry M. Mendel. (1995). Fuzzy Logic Systems for Engineering: A tutorial. Proceedings of the IEEE, Volume 83(3), pp. 345-377. Available at: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=364485 (Accessed: 21 Nov 2017)

[6] R. M. Tong. (1977). A Control Engineering Review of Fuzzy Systems. Automatica, Volume 13(6), pp. 559-569. Available at: https://www.sciencedirect.com/science/article/pii/0005109877900772 (Accessed: 21 Nov 2017)

[7] E. H. Mamdani. (1993). Twenty years of Fuzzy Control: Experiences Gained and Lessons Learnt. Second IEEE International Conference on Fuzzy Systems, San Francisco, CA, 1993, Volume 1(1), pp. 339-344. Available at: https://ieeexplore.ieee.org/document/327439/ (Accessed: 12 Nov 2017)

[8] P. J. King and E. H. Mamdani. (1977). The application of fuzzy control systems to industrial processes. Automatica, Volume 13(3), pp. 235-242. Available at: https://www.sciencedirect.com/science/article/pii/0005109877900504 (Accessed: 12 Nov 2017)

[9] Jan Jantzen. (2007). Foundations of fuzzy control. 1st edition. Antony Rowe Ltd, Chippenham, Wiltshire: Wiley, 2007. Pages 1-115 and 173-189.

[10] Kevin M. Passino and Stephen Yurkovich. (1998). Fuzzy control. 1st edition. Harlow, England: Addison Wesley Longman, Inc, 1998. Pages 1-177 and 415-451.

[11] L. A. Zadeh. (1975). The concept of a linguistic variable and its application to approximate reasoning - I. Inf. Sci., Volume 8, pp. 199-249. Available at: https://pdfs.semanticscholar.org/74d5/e9445c8154c8a77cc71487a4c5cfd9a9588c.pdf?_ga=2.251216582.1149185436.1524066693-1652775536.1524066693 (Accessed: 2 Nov 2017)

[12] L. A. Zadeh. (1984). Making computers think like people [fuzzy set theory]. IEEE Spectrum, Volume 21(8), pp. 26-32. Available at: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6370431 (Accessed: 2 Nov 2017)

[13] L. A. Zadeh. (1968). Fuzzy algorithms. Information and Control, [online] Volume 12(2), pp. 94-102. Available at:

https://www.sciencedirect.com/science/article/pii/S0019995868902118 (Accessed: 3 Nov 2017)

[14] L. A. Zadeh. (1996). Fuzzy logic = computing with words. IEEE Transactions on Fuzzy Systems, Volume 4(2), pp. 103-111. Available at: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=493904 (Accessed: 3 Nov 2017)

[15] Abdel Ghani Aissaoui and Ahmed Tahour. (2012). Application of Fuzzy Logic in Control of Electrical Machines. Fuzzy Logic - Controls, Concepts, Theories and Applications, Prof. Elmer Dadios, InTech. Available at: https://pdfs.semanticscholar.org/0d90/fc3c9d55cd448c34fc1c6b1fc1ad0932d031.pdf (Accessed: 18 Nov 2017)

[16] E.H. Mamdani. (1974). Application of fuzzy algorithms for control of simple dynamic plant. Proc. IEEE, [online] Volume 21(12), pp.1585-1588 Available at: http://ieeexplore.ieee.org/document/5250910/ (Accessed: 8 Sep 2017)

[17] Ahmad Ilyas Roosea, Samer Yahya and Hussain Al-Rizzo. (2017). Fuzzy-logic control of an inverted pendulum on a cart. Available at: http://www.sciencedirect.com/science/article/pii/S0045790617313654 (Accessed: 5 Nov 2017)

[18] L. A. Zadeh. (1965). Fuzzy sets. Information and Control, Volume 8(3), pp. 338-353. Available at: https://www.sciencedirect.com/science/article/pii/S001999586590241X (Accessed: 12 Sep 2017)

[19] L. A. Zadeh. (1988). Fuzzy Logic. Computer, Volume 21(4), pp. 83-93. Available at: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=53 (Accessed: 12 Sep 2017)

[20] L. A. Zadeh. (1973). Outline of a New Approach to the Analysis of Complex Systems and Decision Processes. IEEE Transactions on systems, man, and cybernetics, Volume 3(1), pp. 28-44. Available at: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5408575 (Accessed: 2 Nov 2017)

[21] L. A. Zadeh. (2008). A Fuzzy-Set-Theoretic Interpretation of Linguistic Hedges. Journal of Cybernetics, Volume 2(3), pp. 4-34. Available at: http://dx.doi.org/10.1080/01969727208542910 (Accessed: 5 Nov 2017)

[22] Tomohiro Takagi and Michio Sugeno. (1985). Fuzzy Identification of Systems and Its applications to Modelling and Control. Available at: http://takagiken.com/takagi-sugeno-modeling.pdf (Accessed: 8 Sep 2017)

[23] Gang Feng. (2010). Analysis and Synthesis of Fuzzy Control Systems A Model-Based Approach. 1st edition. CRC press, Broken South Parkway NW: Taylor and Francis group, 2010. Pages 1-111. Pages 1-111.

[24] Jairo Espinosa, Joos Vandewelle and Vincent Wertz. (2005). Fuzzy logic, Identification and Predictive Control. 1st edition. United States of America: Springer-Verlag London Limited, 1998. Pages 1-20, 123-151 and 215-223.

[25] J. Yi, N. Yubazaki. (2000). Stabilization fuzzy control of inverted pendulum systems. Available at: http://www.sciencedirect.com/science/article/pii/S0954181000000078 (Accessed: 2 Nov 2017)

[26] M. I. H. Nour, J. Ooi and K. Y. Chan. (no date). Fuzzy Logic Control vs. Conventional PID Control of an Inverted Pendulum Robot. Available at: https://pdfs.semanticscholar.org/7307/d52cb1a92c24184ed89d465644ef2c6c7706.pdf (Accessed: 2 Nov 2017)

[27] Hua 0. Wang, Kazuo Tanaka and Michael F. Griffin. (1996). An approach to fuzzy control of non-linear systems: Stability and design issues. IEEE Transactions on Fuzzy Systems, Volume 4(1). Available at: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=481841 (Accessed: 5 Nov 2017)

[28] Adrain-vasile Duka, Stelian Emilian Oltean and Mircea Dulau. (2005). Model Reference Adaptive Control and Fuzzy Model Reference Learning Control for the Inverted Pendulum. Comparative Analysis. Available at: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.498.8806&rep=rep1&type=pdf (Accessed: 26 Feb 2018)

[29] Singh Vivekkumar Radhamohan, Mona Subramaniam A and DR. M.J.Nigam. (2005). Fuzzy swing-up and stabilization of real inverted pendulum using single rulebase. Available at: https://pdfs.semanticscholar.org/5835/9411dab21baaffa74a2932b7a192a98ce608.pdf (Accessed: 26 Feb 2018)

[30] Yasar Beceriklia and B. Koray Celikb. (2006). Fuzzy control of inverted pendulum and concept of stability using Java application. Available at: https://www.sciencedirect.com/science/article/pii/S0895717707000076 (Accessed: 16 Sep 2017)

[31] Haresh A. Suthar and Kaushal B. Pandya. (no date). Designing fuzzy logic controller for inverted pendulum. Artificial Intelligence Applications and Innovations, Volume 187, pp. 61-68. Available at: https://link.springer.com/content/pdf/10.1007%2F0-387-29295-0_7.pdf (Accessed: 5 Nov 2017)

[32] Michio Sugeno. (1985). Industrial applications of fuzzy control. 1st edition. Amsterdam, The Netherlands: Elsevier science publishers, 1992. All pages.

[33] Mohammad Jamshidi, Nader Vadiee and Timothy J. Ross. (1993). Fuzzy logic and Control. Englewood Cliffs, New Jersey: PTR Prentice-Hall, Inc. Pages 149-393.

[34] Hapurne Tania Mariana and Pricop Andrei. (2000) Application of Fuzzy Set Theory in Controlling of Urban Storm Drainage Systems. In: International conference on applications of computer science and mathematics in architecture and construction. City: Bauhaus University, Weimar. Available at: https://www.researchgate.net/profile/Andrei_Pricop/publication/237226656_Application_of_Fuzzy_Set_Theory_in_Controlling_of_Urban_Storm_Drainage_Systems/links/5506a5f40cf24cee3a05b858/Application-of-Fuzzy-Set-Theory-in-Controlling-of-Urban-Storm-Drainage-Systems.pdf (Accessed: 15 Nov 2017)

[35] L. P. Holmblad, J. J. Ostergaard. (1981). Control of a Cement Kiln by Fuzzy Logic Techniques. IFAC Proceedings Volumes, Volume 14(2), pp. 809-814. Available at:

https://www.sciencedirect.com/science/article/pii/S1474667017635821 (Accessed: 15 Nov 2017)

[36] R. M. Tong, M. B. Beck and A. Latten. (1980). Fuzzy control of the activated sludge wastewater treatment process. Automatica, Volume 16(6), pp. 695-701. Available at: https://www.sciencedirect.com/science/article/pii/0005109880900114 (Accessed: 15 Nov 2017)

[37] W. Kickert and M. Van Naute Lemke. (1976). The application of fuzzy set theory to control a warm water process. Automatica, Volume 12(1), pp. 301-308. Available at: https://pure.tue.nl/ws/files/4436126/334294.pdf (Accessed: 15 Nov 2017)

[38] I. G. Umbers and P. J. King. (1980). An analysis of human decision-making in cement kiln control and the implications for automation. International Journal of Man-Machine Studies, Volume 12(1), pp. 11-23. Available at: https://www.sciencedirect.com/science/article/pii/S0020737380800514 (Accessed: 15 Nov 2017)

[39] Bin-Da Liu, Chuen-Yau Chen and Ju-Ying Tsao. (2001). Design of Adaptive Fuzzy Logic Controller Based on Linguistic-Hedge Concepts and Genetic Algorithms. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), Volume 31(1), pp. 32-53. Available at: https://ieeexplore.ieee.org/document/907563/ (Accessed: 7 Feb 2018)

[40] MathWorks. MATLAB Fuzzy Logic Toolbox. Available at: https://uk.mathworks.com/help/fuzzy/index.html

[41] Juan Rada-Vilela. FuzzyLite: A fuzzy logic control library, 2017. Available at: https://www.fuzzylite.com/

[42] David A. Kolb. (2015). Experiential Learning: Experience as the Source of Learning and Development. 2nd edition. Upper Saddle River, New Jersey USA: Pearson Education Inc. Pages 31-65 and 239-311.

[43] Oracle. Java. Available at: https://java.com

[44] H. J. Zimmermann. (2001). Fuzzy Set Theory-and Its Applications. 4th Edition. New York: Springer Seience+Business Media, 2001. Pages 185-265.

[45] John G. Bollinger and Neil A. Duffie. (1988). Computer Control of Machines and Processes. Reading, Massachusetts, USA: Addison-Wesley Publishing Company, 1988. Pages 1-29.

[46] Martha Lewis and Jonathan Lawry. (2016). A Label Semantics Approach to Linguistic Hedge. Available at: https://www.groundai.com/project/a-label-semantics-approach-to-linguistic-hedges/ (Accessed: 7 Feb 2018)

[47] Martine De Cock. (no date). A New Generation of the Fuzzy-set-theoretical Representation of Modified Linguistic Terms. Available at: https://pdfs.semanticscholar.org/2a6e/fe037bd57a5a6ce4447454f18a729b49b8cd.pdf (Accessed: 7 Feb 2018)

10. Appendix

10.1. Acronyms

API: Application programming interface. A set of functions which provide an abstracted interface to a set of features or data that aid the creation of computer applications.

FL: Fuzzy logic. A type of many valued logic in which the degree of truth of a statement or variable may be any value between 0 and 1.

FC: Fuzzy control. The field of study of fuzzy logic and fuzzy set theory for control engineering purposes.

FIS: Fuzzy inference system or fuzzy inference technique.

FLC: Fuzzy logic controller, fuzzy logic control system. An automatic control system which employs fuzzy control to determine a control strategy.

GUI: Graphical user interface. A human-computer interface that allows human interaction with computer applications via graphical icons and visual indicators.

HCI: Human computer interaction. The study of how people interact with computers and how well-developed computers and applications are for successful interaction with humans.

PID: Proportional plus integral plus derivative control.

PD: Proportional plus derivative control.

P: Proportional control.

SIRM: Single input rule module. A rule module containing rules which use only one input variable.

10.2. Definitions

Automatic controller: A device or computer application designed to automatically control a system.

C++: A freeware computer programming language that will run on most popular operating systems including Linux, Windows and Mac OS X.

Class (Computer science): A class is a programming template for creating objects in object-oriented programming. They provide member variables and member functions.

Control goal: A goal state for system to be controlled that the controller is to achieve.

Control scheme: A specific set of control parameters that determine a strategy or protocol for control.

Control system: An algorithm specifically designed for control applications which provides a set of parameters to the operator that allow a control strategy or protocol to be defined.

Damping: A phenomena that reduces the rate of change of a system to towards the goal state. Damping helps to minimise overshoot.

Experiential learning: The process of learning through experience or learning through reflection on doing.

Java: A freeware computer programming language designed to run on the Java virtual machine, allowing it to run on any computer system.

Java Virtual Machine: The freeware virtual machine that the Java programming language runs on. A virtual machine is a software computer which can run an operating system and applications similarly to a physical computer.

Overshoot: The amount that the system over responds to an error. Overshoot occurs when the control action is too large causing the set point to be over shot, this causes an opposite error to be created.

Plant error: The difference between the measured value and the set point of an input variable to a controller.

Priority control goal: The most important control goal to achieve.

Priority variable: The input variable which governs the response towards the priority control goal, usually a proportional variable.

Rise time: Time taken for the system to reach the set point.

System variable: A measurable variable that exists in the system to be controlled. Also termed an input variable when the system variable is used an input to a controller.

Set point: The goal value of a system variable.

Stability: The degree to which the system is stable, high stability requires little fluctuation in the state of the system.

Steady-state: The system has reaches a stable state where the error is no longer fluctuating or only fluctuating by a “small” degree.

Steady-state error: The value of the error whilst at steady-state. Steady-state error usually manifests itself symmetrically around the set point, assuming the system is symmetric.

Unix: A group of operating systems that derive from the original AT&T Unix including Linux and Mac OS X.

Windows: A graphical operating system developed and sold by Microsoft. Windows is used on approximately 90% of the world’s personal computers.

10.3. Equations

$$R = N^V \tag{1}$$

$$R = (H * N)^V \tag{2}$$

$$O(R * V) \tag{3}$$

$$O(R) \tag{4}$$

Where:

- R = Number of rules
- N = Number of membership functions
- H = Number of hedges
- V = Number of input variables

Assuming:

- Only the AND aggregate is used
- All possible rules are specified
- Membership functions sum to one across the entire input range
- Each control goal has an equal number of input variables and input term types that govern it

$$a = \frac{F}{M + m * \|\cos\theta\|} \tag{5}$$

$$\alpha = \frac{a}{l} * \sin\theta - \frac{g}{l} * \cos\theta \tag{6}$$

$$F_{dc} = \frac{1}{2} C_{dc}\, \rho\, A_c U^2 \tag{7}$$

$$F_{dp} = \frac{1}{2} C_{dp}\, \rho\, A_p (\omega * l)^2 \tag{8}$$

$$F_r = C_r * (M + m * \sin\theta) \tag{9}$$

Where:

- a = Acceleration of cart
- F = Force on cart
- M = Mass of cart
- m = Mass of pendulum end
- θ = Angle of pendulum
- α = Angular acceleration of pendulum
- l = Length of pendulum
- g = acceleration due to gravity
- F_{dc} = Aerodynamic drag force on cart
- C_{dc} = Aerodynamic drag coefficient of cart
- F_{dp} = Aerodynamic drag force on pendulum end
- C_{dp} = Aerodynamic drag coefficient of pendulum end
- A_c = Frontal cross-sectional area of cart
- A_p = Frontal cross-sectional area of pendulum end
- U = Velocity of cart
- ω = Angular velocity of pendulum
- ρ = Density of air
- F_r = Rolling resistance force on cart
- C_r = Coefficient of rolling resistance

$$F(t) = K_p e(t) + K_i \int_0^t e(\tau)\, d\tau + K_d \frac{de(t)}{dt} \quad (10)$$

Where:

- K_p = Proportional gain
- K_i = Integral gain
- K_d = Derivative gain
- t = Instantaneous time
- $e(t)$ = Error at time t
- τ = Variable of integration
- $F(t)$ = Output at time t

$$F = \frac{\sum_{i=1}^{R} b_i u_i}{\sum_{i=1}^{R} u_i} \quad (11)$$

Where:

- R = Number of rules
- b_i = Raw output of i^{th} rule
- u_i = Degree to truth of i^{th} rule
- $F(t)$ = Output

10.4. Tables

Table 2: Physical parameters for Java application

Parameter	**Description**
Pendulum length	The length of the inverted pendulum. This will influence the angular acceleration of the pendulum.
Pendulum end radius	The radius of the end weight attached to the pendulum. His will influence the magnitude of the drag force on the pendulum end.
Cart height and width	The height and width of the cart. This will influence the magnitude of the drag force on the cart.
Pendulum end weight mass	The mass of the weight attached to the pendulum. This will influence the acceleration of the pendulum towards the ground.
Cart mass	The mass of the cart. This will influence the acceleration of the cart and the magnitude of the rolling resistance force.
Acceleration due to gravity	The acceleration due to gravity. This affects the acceleration of the pendulum towards the ground and the normal reaction of the cart against the ground which will influence the magnitude of the rolling resistance force.
Rolling resistance coefficient	The coefficient of rolling resistance. The magnitude of rolling resistance is directly proportional to this coefficient.
Pendulum end drag coefficient	The aerodynamic drag coefficient of the pendulum end weight. The magnitude of the aerodynamic drag force on the pendulum end is directly proportional to this coefficient.
Cart drag coefficient	The aerodynamic drag coefficient of the cart. The magnitude of the aerodynamic drag force is directly proportional to this coefficient.
Air density	The density of the air. The magnitude of both aerodynamic drag forces is directly proportional to this value.

Table 3: Simulation parameters for Java application

Parameter	**Description**
Physics tick rate	The number of times per second that the mechanics of the simulation are calculated. Higher values will result in a more accurate simulation but will demand greater computational resources[1].
Controller tick rate	The number of times per second that the control actual is calculated. Higher values will result in a more accurate simulation but will demand greater computational and memory resources[16].
Frame rate	The number of times per second that graphical displays are updated. Higher values will make the viewing experience smoother but will demand greater computational resources.
Result sampling rate	The number of times per second that result samples are taken. Higher values will produce more accurate results but will demand greater computational and memory resources.
Run time scaling factor	The factor of simulation time to real time. The simulation speed will speed up or slow down directly proportionally to this value.
Cart movement activation	Specify the cart to be fixed or free to move.
Noise activation	Specify whether sensor/motor is to be used.
Pendulum angle sensor noise factor	The percentage noise of the pendulum angle. The noise is a random value with a maximum equal to the current error multiplied by this factor.
Cart position sensor noise factor	The percentage noise of the cart position. The noise is a random value with a maximum equal to the current error multiplied by this factor.
Cart motor force noise factor	The percentage noise of the cart motor. The noise is a random value with a maximum equal to the current error multiplied by this factor.

10.5. Figures

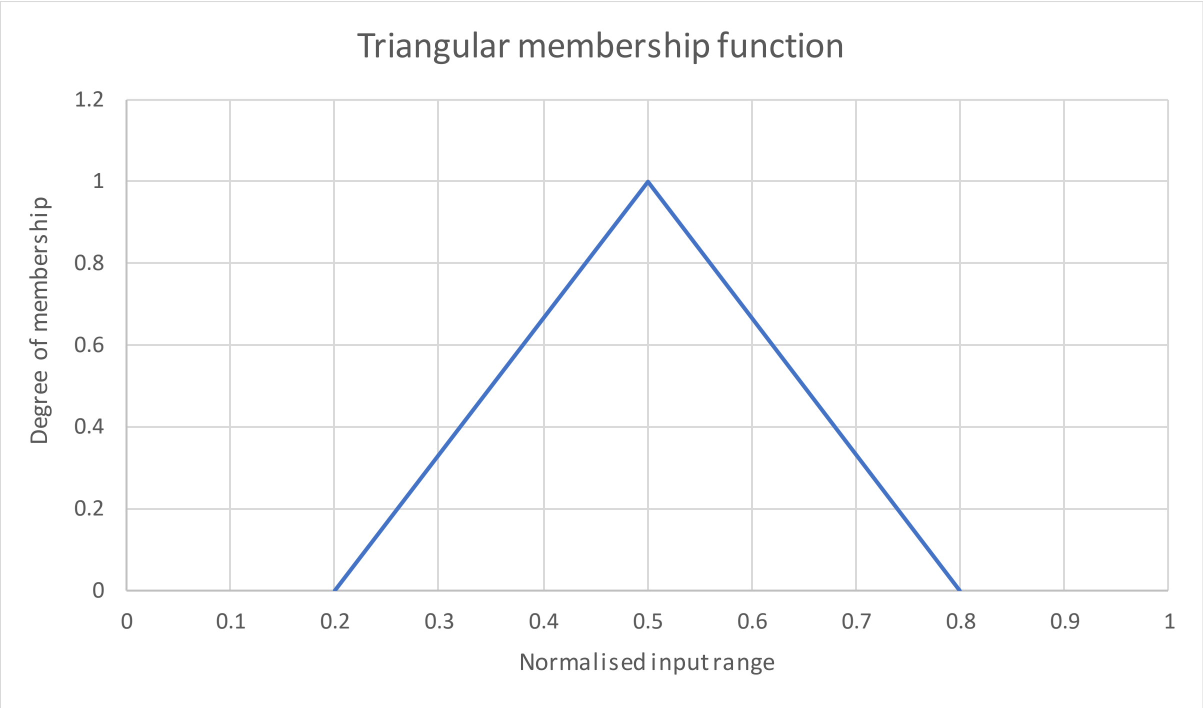

Figure 11: Triangular membership function

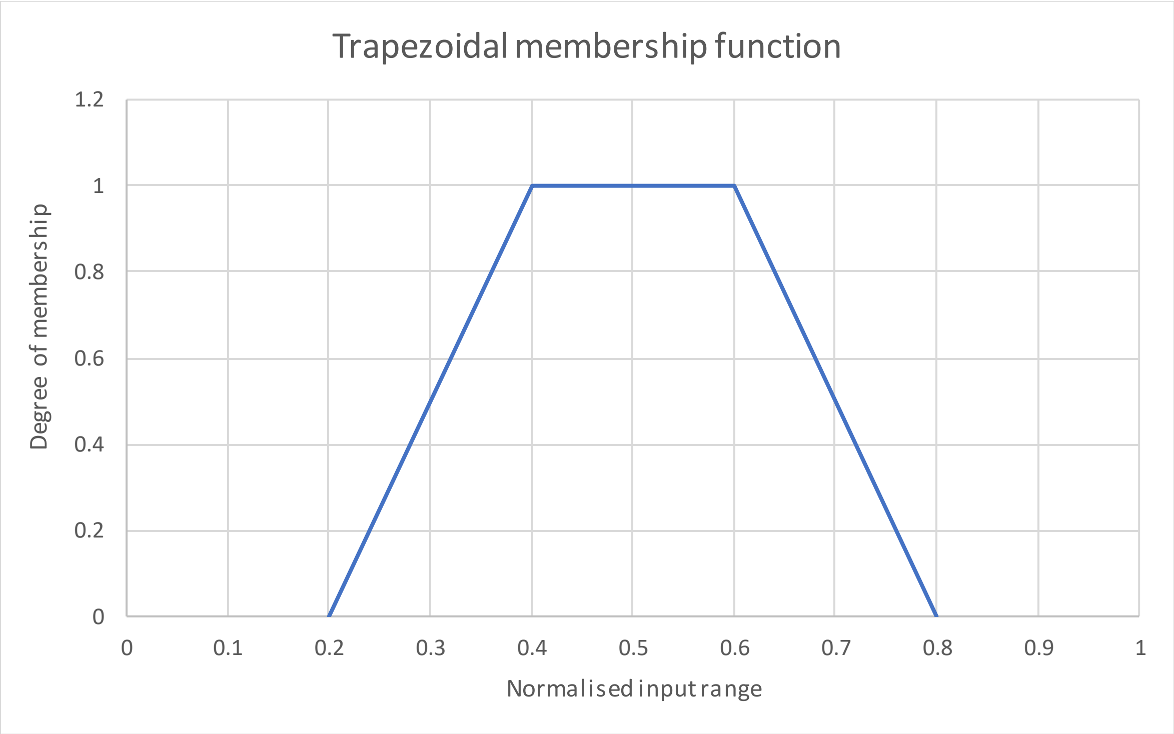

Figure 12: Trapezoidal membership function

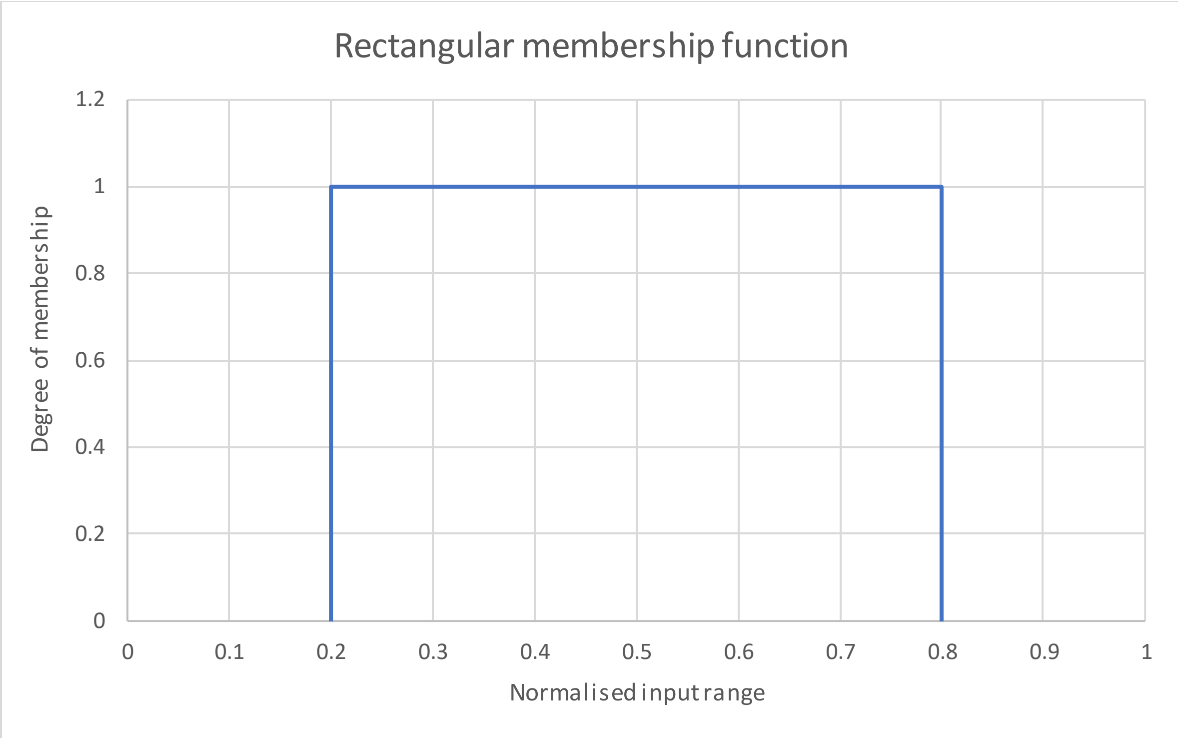

Figure 13: Rectangular membership function

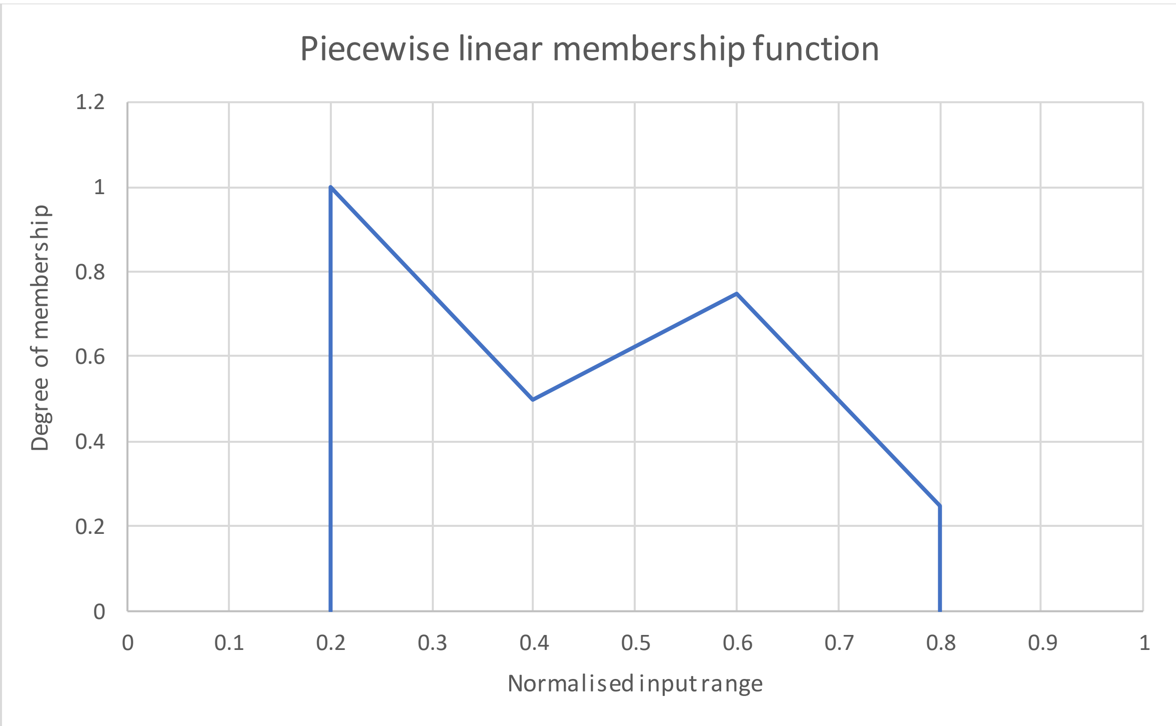

Figure 14: Piecewise linear membership function

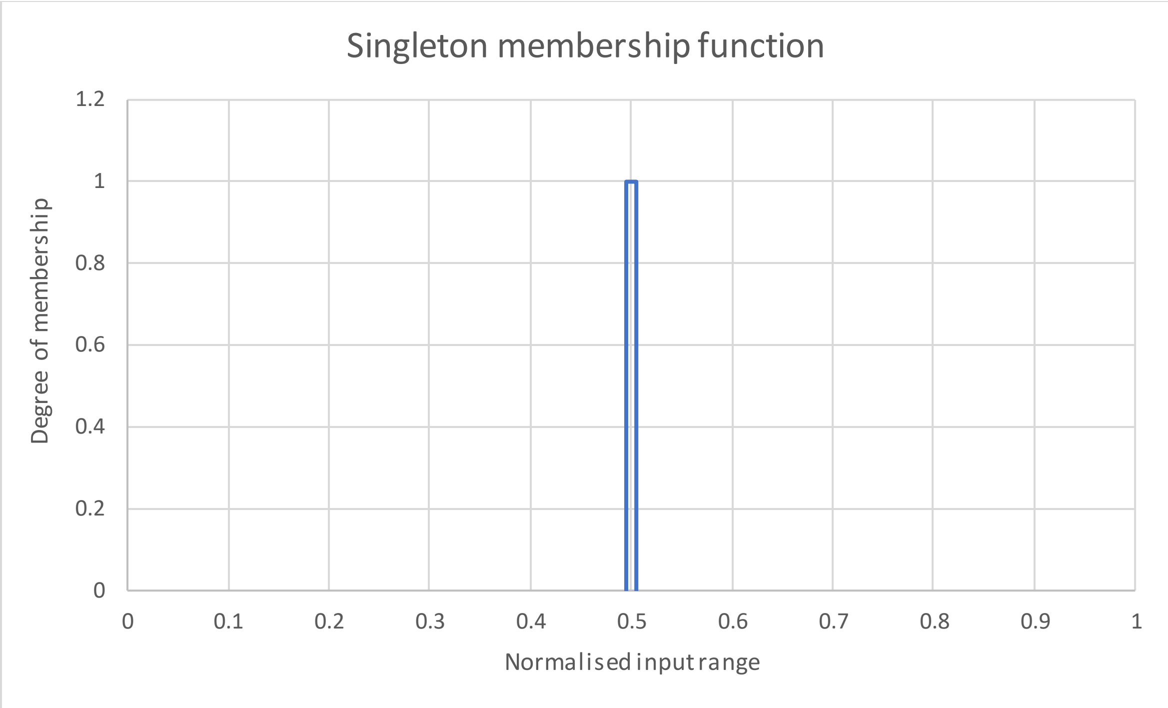

Figure 15: Singleton membership function

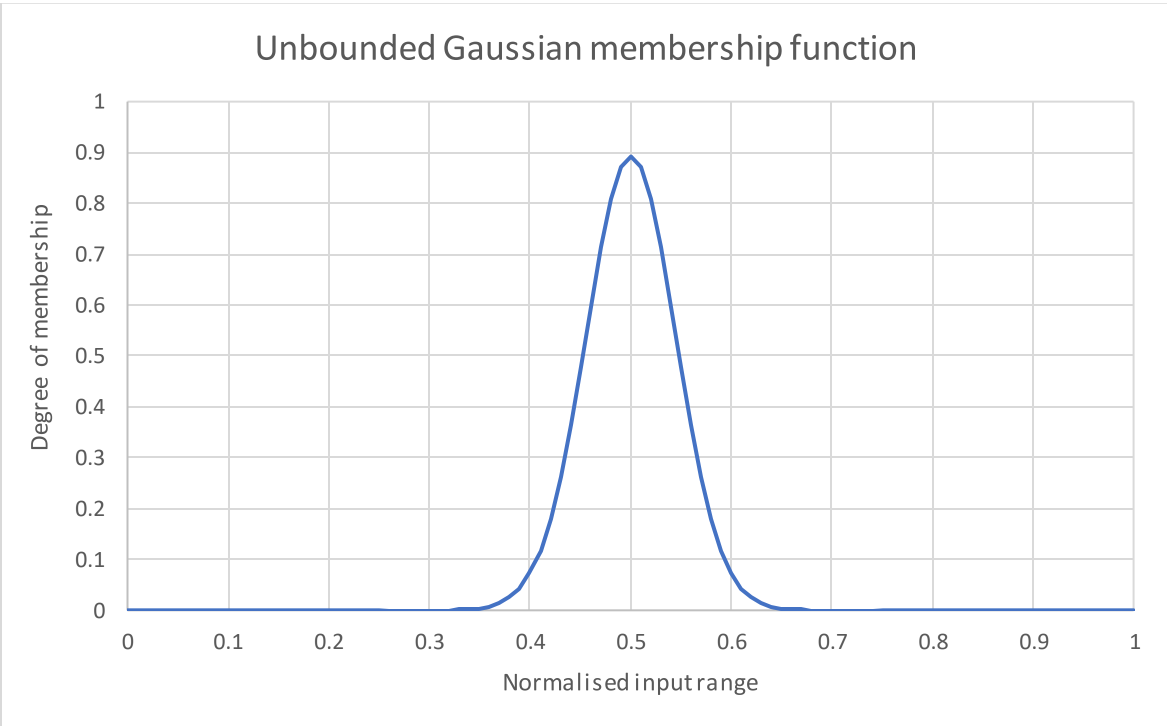

Figure 16: Unbounded Gaussian membership function

10.6. Java application simulation model and controller implementation

The dynamics of the cart and pendulum are governed by Equations 5 and 6 respectively. The magnitude of the aerodynamic drag force on the cart and pendulum are governed by Equations 7 and 8 respectively. The magnitude of the rolling resistance force on the cart is governed by Equation 9.

10.6.1. Model assumptions

The inverted pendulum on a cart system is governed by a vast array of properties, each of which are represented by equations of high arithmetic complexity. It is not realistically feasible to model the system in its entirety. Assumptions were therefore made to comprise between realism and simplicity, anything not required to achieve the learning objectives was ignored.

- The pendulum rod is assumed to be perfectly rigid, massless and has no volume
- Friction in the various bearings is negligible
- There are no inertial losses in drivetrain components
- There is no limit to the available motor force or braking force
- There is no hysteresis in the motor or brakes, the requested tractive force is applied immediately regardless of the current cart velocity
- No backlash occurs when the cart collides with the end of the track, all cart inertia is lost immediately upon impact
- No backlash occurs when the pendulum falls to its stops, all pendulum rotational inertia is lost immediately upon impact
- Friction between the interface of the cart wheels and track is infinite such that there is a no-slip condition, wheel spin can never occur

10.6.2. PID controller

There are two separate PID controllers, one for each control goal, one governing the cart's position and the other governing the pendulum's angle. The output of each PID controller is calculated using Equation 10. The total output is simply the summation of the output from each PID controller.

10.6.3. Fuzzy logic controller

The FLC uses a standard rule-based Tagaki-Sugeno FIS which is outlined in [3]. The total controller output is calculated using Equation 11.

All membership functions, resolution hedges and aggregate functions included in the FLC are shown in Tables 4, 5 and 6 respectively.

Table 4: Membership functions for Java application

Name	Start	First peek	Second peek	End
HUGE	0.72727272	0.81818181	1.0	1.0
LARGE	0.54545454	0.63636363	0.72727272	0.81818181
BIG	0.45454545	0.54545454	0.54545454	0.63636363
SMALL	0.36363636	0.45454545	0.45454545	0.54545454
LITTLE	0.18181818	0.27272727	0.36363636	0.45454545
TINY	0.0	0.0	0.18181818	0.27272727

Table 5: Resolution hedges for Java application

Name	Start	First peek	Second peek	End
ANY	0.0	0.0	1.0	1.0
EXTREMELY	0.72727272	0.81818181	1.0	1.0
VERY	0.54545454	0.63636363	0.72727272	0.81818181
MEDIUM	0.36363636	0.45454545	0.54545454	0.63636363
SOMEWHAT	0.18181818	0.27272727	0.36363636	0.45454545
SLIGHTLY	0.0	0.0	0.18181818	0.27272727
ABS_MAX	0.99	0.99	1.0	1.0
ABS_MEDIAN	0.495	0.495	0.505	0.505
ABS_MIN	0.0	0.0	0.01	0.01

Table 6: Aggregate functions for Java application

Name	Function
AND	MIN(x, y)
OR	MAX(x, y)
NAND	1.0 - MIN(x, y)
NOR	1.0 - MAX(x, y)
NOT	1.0 - x

10.7. Model and controller setup for experimentation

The following physical parameters were used for the experimentation:

- Cart mass: 1 Kilogram
- Pendulum length: 1 Meter
- Starting pendulum angle: 45 Degrees
- Pendulum bounds: 150.0 - 30.0 Degrees
- Steady state error: -10 Newtons

The following controller parameters were used for the experimentation:

- Module output aggregation mode: Weighted sum
- Gain: 1.0

Input variables used:

- Name: “pendulum angle”, Type: Proportional, Gain: 1.0, Maxima: 150.0, Minima: 30.0
- Name: “pendulum integral error”, Type: Integral, Gain: 0.25, Maxima: 150.0, Minima: 30.0
- Name: “pendulum angular velocity”, Type: Derivative, Gain: 1.0, Maxima: 5.0, Minima: -5.0

Membership functions used:

- Name: "huge", Type: Trapezoidal, Start: 0.72727272, First peek: 0.81818181, Second peek: 1.0, End: 1.0
- Name: "large", Type: Trapezoidal, Start: 0.54545454, First peek: 0.63636363, Second peek: 0.72727272, End: 0.81818181
- Name: "big", Type: Triangular, Start: 0.45454545, Peek: 0.54545454, End: 0.63636363
- Name: "small", Type: Triangular, Start: 0.36363636, Peek: 0.45454545, End: 0.54545454
- Name: "little", Type: Trapezoidal, Start: 0.18181818, First peek: 0.27272727, Second peek: 0.36363636, End: 0.45454545
- Name: "tiny", Type: Trapezoidal, Start: 0.0, First peek: 0.0, Second peek: 0.18181818, End: 0.27272727
- Name: “max-saturated”, Type: Saturation, Activate above: 1.0, Activate below: infinity
- Name: “min-saturated”, Type: Saturation, Activate above: negative infinity, Activate below: 0.0

Resolution hedges used:

- Name: "extremely", Type: Trapezoidal, Start: 0.72727272, First peek: 0.81818181, Second peek: 1.0, End: 1.0
- Name: "very", Type: Trapezoidal, Start: 0.54545454, First peek: 0.63636363, Second peek: 0.72727272, End: 0.81818181
- Name: "medium", Type: Trapezoidal, Start: 0.36363636, First peek: 0.45454545, Second peek: 0.54545454, End: 0.63636363
- Name: "somewhat", Type: Trapezoidal, Start: 0.18181818, First peek: 0.27272727, Second peek: 0.36363636, End: 0.45454545
- Name: "slightly", Type: Trapezoidal, Start: 0.0, First peek: 0.0, Second peek: 0.18181818, End: 0.27272727

Rules where split into modules to increase transparency, module 1 contains all proportional terms, module 2 the derivative terms and module 3 the integral terms. Integral and derivative modules employ the use of saturation membership functions. All modules were set to weighted average mode with a gain of 1.0.

Module 1:

- IF pendulum angle IS any huge THEN force IS -50.0
- IF pendulum angle IS any large THEN force IS -25.0
- IF pendulum angle IS any big THEN force IS -10.0
- IF pendulum angle IS any small THEN force IS 10.0
- IF pendulum angle IS any little THEN force IS 25.0
- IF pendulum angle IS any tiny THEN force IS 50.0

Module 2:

- IF pendulum angular velocity IS any max-saturated THEN force IS -25.0
- IF pendulum angular velocity IS any huge THEN force IS -17.5
- IF pendulum angular velocity IS any large THEN force IS -10.0
- IF pendulum angular velocity IS any big THEN force IS -5.0
- IF pendulum angular velocity IS any small THEN force IS 5.0
- IF pendulum angular velocity IS any little THEN force IS 10.0
- IF pendulum angular velocity IS any tiny THEN force IS 17.5
- IF pendulum angular velocity IS any min-saturated THEN force IS 25.0

Module 3:

- IF pendulum integral error IS any max-saturated THEN force IS -10.0
- IF pendulum integral error IS any huge THEN force IS -7.5
- IF pendulum integral error IS any large THEN force IS -5.0
- IF pendulum integral error IS any big THEN force IS -2.5
- IF pendulum integral error IS any small THEN force IS 2.5
- IF pendulum integral error IS any little THEN force IS 5.0
- IF pendulum integral error IS any tiny THEN force IS 7.5
- IF pendulum integral error IS any min-saturated THEN force IS 10.0

The rules in module 1 that govern the proportional response to the pendulum angle where changed to see how the inclusion of resolution hedges affect the control performance.

Changed module 1:

- IF pendulum angle IS any huge THEN force IS -50.0
- IF pendulum angle IS any large THEN force IS -25.0
- IF pendulum angle IS extremely big THEN force IS -25.0
- IF pendulum angle IS very big THEN force IS -25.0
- IF pendulum angle IS medium big THEN force IS -17.5
- IF pendulum angle IS somewhat big THEN force IS -10.0
- IF pendulum angle IS very small THEN force IS 10.0
- IF pendulum angle IS medium small THEN force IS 17.5
- IF pendulum angle IS somewhat small THEN force IS 25.0
- IF pendulum angle IS slightly small THEN force IS 25.0
- IF pendulum angle IS any little THEN force IS 25.0
- IF pendulum angle IS any tiny THEN force IS 50.0