Fuzzy Control of Inverted Pendulum on a Cart

Adeel Nafis
Institute of Industrial Electronics Engineering Pakistan
With Prof. ***Ashab Mirza*** *as Technical Advisor*

Abstract:

Inverted Pendulum type mechanical arrangements and configurations are commonly encountered in mechanical engineering, Missiles and rockets with a heavy payload in the head may be loosely linked to an Inverted Pendulum. This has led to extensive studies of the Inverted Pendulum in control engineering so much so that it has become a classic control problem.

The conventional approach is to develop a PID controller by approximating the non-linear transfer function to a linear one. This enables the system to be controlled in a specific range. We are trying to develop a fuzzy controller that will enable us to control the Inverted Pendulum in a much better way then possible from a conventional controller. This is possible because of the ability of the Fuzzy system to approximate any non- linear function. Our approach however is to develop the fuzzy controller without reference to the transfer function of the IP.

The paper is divided into two parts. The first one covers the necessary introduction on fuzzy and the second part pertains to our implementation of the system.

PART 1:

Defining Fuzzy

Humans are able to handle complex problems in everyday life without having an exact mathematical understanding of the system [1]. For example when we ride a bicycle we are unaware of the value of parameters that are influencing our ride. These parameters may include such things as the cycle's weight velocity and friction. Even then a rider is usually able to handle a bicycle quite precisely in the most extreme of circumstances.

From the above example we can see that the capability of humans to solve problem is based on a heuristic approach. The accumulated experience is used in such a way as to provide an approximate solution to the problem at hand. Fuzzy logic is all about developing a system based on heuristic knowledge.

Fuzzy logic is a superset of conventional Boolean logic [2]. Boolean logic is based on the rule that every input and output of a system can have a maximum of two discrete levels of value, usually called '1' or '0'. All processing is carried out on the same bases. Boolean logic naturally assumes that every property of a system can have only fixed discrete levels of information. On the other hand Fuzzy assumes that discrete levels of information is only a special case of information being continuous in nature.

In everyday life we use many words; which have no fixed discrete value but the value is dependent on a number of factors. For example it is difficult to say what is ‘clean’ or ‘not clean’, ‘tall’ or ‘not tall’, etc. In many cases the *Law of Excluded Middle 1 fails to hold and it is easy to show that the exclusion is not the exception but the general case [10].*

Fuzzy provides a way of developing a system capable of working on these non- discrete values or aptly called fuzzy values of a system. This if applied correctly leads to a much better behaved system, capable of a much more robust control then usually possible through conventional means.

Through its ability to work on *non- crisp (discrete) values, fuzzy systems make it possible to develop systems on the heuristic knowledge of a human operator as opposed to the mathematical model based systems required for conventional control.*

Fuzzy control provides a formal methodology for representing, manipulating, and implementing a human's heuristic knowledge about how to control a system [7].

Fuzzy's History:

Most of the work started in the late 19th century, when some mathematicians and philosophers started their investigations into the world of ‘vague’. Bertrand Russell formally defined a concept or statement as vague if and only if it fails to hold the law of excluded middle (A is not equal to *not A).*

Finally in 1965, Lotfi. A. Zadeh described the mathematics of fuzzy logic in his seminal work. The theory showed how membership functions could be developed having values between true ‘1’ and false ‘0’. This provided the stimulus to Fuzzy Logic as calculus operations on fuzzy numbers were also developed. The proposed system is what the world has come to identify as fuzzy logic.

Implementation of Fuzzy*:*

Fuzzy Sets:

Any fuzzy system has first to be able to define what it holds as fuzzy. This automatically leads to the concept of *Fuzzy Sets*. In conventional logic a set can have a particular member or not have a particular member, fuzzy sets on the other hand are those which contain members to different degrees of membership, which varies from ‘1’ (completely true) to ‘0’ (completely false).

The heart of a fuzzy system is the *Rule Base* this contains information of what the systems output should be for a particular set of input conditions. Basic to the rule base are the fuzzy sets. An easy example is of a fan- speed controller, whose speed is dependent on the ambient temperature. Our requirement is to set the speed at a higher level when it is hot. Problem is we don't know what hot actually implies. That is, does it refer to 30^{O}C or 40°C.

DOT(Degree of Truth)

1

0

30^{O} (temperature)

Now as it is evident from the above graph that the transition from 'HOT' to 'not Hot' is abrupt. But our perception of temperature is not like that. We might say that although 30OC is HOT but 29OC is certainly not COLD but is instead *less HOT. The graph representing human perception of temperature in a much better way is shown below.*

HOT

20^{O}C 30^{O}C 40^{O}C

Now say that at 30^{O}C we are completely sure that it is HOT. This means at 30^{O}C we are also sure that it is also completely false that it is NOT HOT. This is shown by the dashed lines. This implies that HOT does not follow the law of excluded middle, and hence the set is fuzzy. (at 25^{O}C and 35^{O}C NOT HOT = HOT).

The shape of the set can be anything but due to their simplicity trapezoidal and triangular shapes are most commonly used.

Linguistic Terms and Variables:

After this designation of sets to each Input and Output variable (called the *Linguistic Variable* as they may not be mathematically defined). These Linguistic Variables are then divided into *Linguistic Terms*. This is the number of terms that are sufficient to define the Variable. For example it may be sufficient to define the temperature as COLD, NOMINAL and HOT. The more Terms present the better the system performs, rarely, however are thee terms more then five for each variable. The lines that define the set are called membership functions, and shape the Set. These Membership functions show the mapping of a value of the input to a Degree of Truth (Membership) of a term.

Rule Base:

Fuzzy logic works on a set of rules. The set of rules that are applicable for a particular set of inputs generate an output, which is then processed to produce a particular solution. That is the Rule- base provides a mapping of the input terms to the output terms.

This mapping is defined in the form of *If- Then* rules. The rules may be developed from intuition, experience or through learning algorithms.

Developing these rules is the most time consuming part of fuzzy engineering.

A simple rule for our speed controller maybe

If Temperature is HOT *Then* Speed is HIGH

Every rule follows the pattern of

If *antecedent* Then *Consequent*

There are at least three basic processes involved[5] for any fuzzy system

1. Fuzzification
2. Inference
3. Defuzzification.

Fuzzification:

The input to the system, say from a sensor, will be a crisp numeric value and the systems first job is to convert it to a non crisp value. This is done through *fuzzification.*

The inputs value can be anything on its *Universe of Discourse* –The range of a particular input. Fuzzification leads to the linguistic terms acquiring a particular truth-value for each input condition.

Say for a three term, temperature variable as shown below the input temperature is at $30^{O}C$.

0.65

0.30

30^{O}C

After fuzzification

COLD = 0

NOMINAL= 0.30

HOT = 0.65

After fuzzification of each input value the *fuzzified value is passed through an Inference- Engine. This inference engine using the rule- base, assigns to each output Linguistic Term a certain value.*

These terms are then defuzzified to produce a crisp output, for the particular set of inputs. A fuzzy system can have more then one output, which leads to different Inference and defuzzification for each output. Fuzzy systems are essentially feed forward system.

Inference:

After fuzzification there are different methods followed for both inference and defuzzification, we have stuck to the simplest one the SUP - STAR Rule. This is much easier to implement and conceptually easier to understand, however better systems are developed by using SAM models[10].

There are different operators used in the rule- base. The actual operation they perform depends on the model being followed. For the sup -star method

(DOM = Degree of membership / truth)

Not A = 1- DOM (A)

AND is represented by '*' and means the 'minimum of'.

$\mathbf{UAa^{j} * UA_{b}^{k} *} \ldots \mathbf{* UAn^{l}} = \min (\mathbf{UAa^{j}, UA_{b}^{k}}, \ldots, \mathbf{UAn^{l}}$

$\mathbf{UAa^{j}}$ means the Jth term of the input Aa.

This means the consequent becomes equal to the minimum of the terms in the antecedent.

OR is represented by '⊕' and means the 'maximum of'.

$\mathbf{UA_a^j} \oplus \mathbf{UA_b^k} = \max (\mathbf{UA_a^j}, \mathbf{UA_b^k})$

In this case the consequent equals the maximum of the antecedent terms.

If a rule contains only one input term then the value of the antecedent term becomes the value of the consequent term.

For example if a rule is

***If temperature is NOMINAL then speed is LOW*,**

and the Degree of Membership (DOM) of NOMINAL is 0.34 then 0.34 is assigned to the consequent in this case LOW.

In case there are more then two terms in the antecedent part then the consequent depends on the type of operator(s).

Now a output term maybe referenced more then once. This means that the consequent term maybe the same in different rules. For example we may want the same response for different conditions.

Rule #1 If Input1 is TERM(A) * Input2 is TERM(B) then Output is TERM(A)

Rule #2 If Input1 is TERM(B) * Input2 is TERM(B) then Output is TERM(A)

However for defuzzification only one value for each output term is possible. This is resolved by assigning the maximum of the possible values to the term or 'Oring' the values assigned to a term.

Let us say three rules *fire*2 the same term giving the result

0.3, 0.2, 0.5

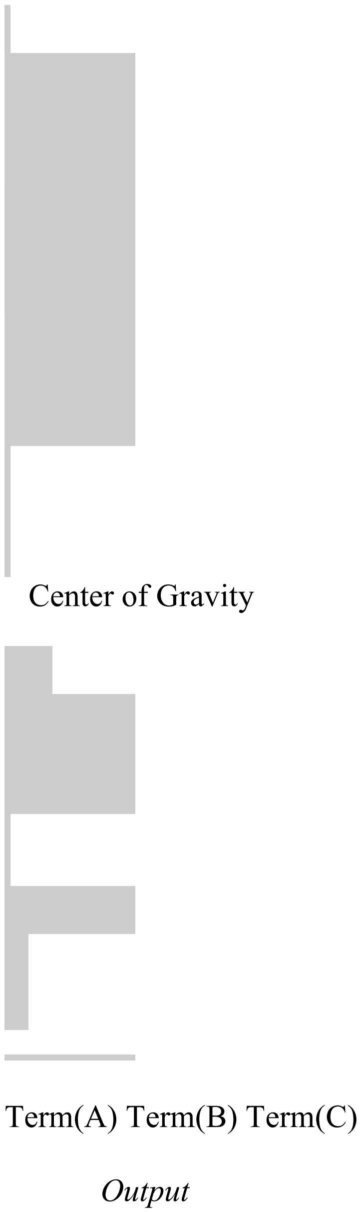
now as 0.5 is the highest and we are using the sup-star rule the maximum(0.5) gets assigned to the term.

Deffuzification:

After getting the fuzzified values from inference for each term of t he output variable, all that remains is to defuzzify.

Again there are different ways but the Center of Area method is easy to use.

The fuzzy set for each term is cut at the final term value after inference. Say for a three term output with Term(A), Term(B) and Term(C) at 0.3, 1, 0.7 respectively the final graph becomes as shown. Then a suitable formula for the Center of Area is applied to get the final crisp value of the output.



.

PART 2:

Although the PID controllers developed at the Institute of Industrial Electronics Engineering (IIEE) are capable of good control of the Inverted

Pendulum on a cart, there performance degrades when the parameters of the system (weight, rigidity, friction, etc) are changed.

The controllers also tend not to be robust. With this in mind we are trying to develop a controller that will enable much better control then presently possible.

The aim for the controller will be to control the pendulum with a bottle of sloshing liquid on top, which has so far not been achieved with the PID controllers developed at the institute.

Overview:

The system is based on pure intuitive and heuristic approach and is devoid of any mathematical model. This places the major emphasis on the rule base. The work is still continuing and the results will be available in around two weeks.

There are two inputs for effective control. One is the angular position (T*heta*) and the other the angular velocity (*Omega*). The angle is measured by a potentiometer mounted on the cart. These provide the voltage of the actuating motor as the output..

The Hardware:

A high speed ISA bus card has been built for this purpose with 16 - bit I/O capability. This makes it possible to simultaneously sample the two inputs with two 8- bit *ADC (Analog to Digital Converters). The software is run on a 333MHz Pc, to which the card is attached directly.*

Although it is possible to get the angular velocity by taking the change in angle at every sample of the angular position and dividing by the sampling time, it reduces the resolution of the angular velocity, as the angular position will not vary widely when appropriate control is there. This is why the angular velocity was taken as a totally separate input.

A general-purpose error detector provides the error in the relative angular position. This input is also then passed through a '*Differentiator'* so to provide the angular velocity, which are then passed to the high- speed ADCs interfaced to the ISA bus, for simultaneous sampling.

The output is through a *DAC* (Digital to Analog Converter) interfaced to the ISA bus, which is then passed through a power amplifier.

The block diagram is shown in fig[1].

Rule- Base:

The inputs and output are resolved into 5 terms each. This has led to 25 different rules with the AND operator as shown in fig[3]

The Membership function for each variable follow the basic shape shown below. The membership functions are narrower at the Equilibrium position for better resolution and spread out at the extremes.

VN N Z P VP

The software uses the sup-star rule for calculating the output. The major hurdle in designing this type of systems is to maintain high speed processing and high- speed communication with the external devises, while changing parameters that are best understood in graphic form.

This has been achieved by using a DLL made in Visual C++, which carries out the required processing and communication and an interface that allows for tuning of the parameters in the DLL (essentially the Membership Functions and the Rule- Base).

The results from its test runs are still not available, as yet, to provide for comparison with conventional control techniques.

Rule Base for The IP control

	Angular velocity	VN	N	Z	P	VP
Angle	VP	Z	N	VN	VN	VN
	P	Z	Z	N	N	VN
	Z	VP	P	Z	N	VN
	N	VP	P	P	Z	Z
	VN	VP	VP	VP	P	Z

Fig [3]

V= very, N= negative, P= positive, Z= zero

Block Diagram:

1 Simply stated it means that a statement can have only one of two values and no intermediate value

2 A rule is set to be 'fired' when its consequent part gets a value other then zero.

Conclusion:

Fuzzy logic from the last decade has firmly established itself as an engineering field. Although it is also used for control engineering applications but a still greater use is in decision-making. This has led to use of fuzzy in medicine, process control, and business, DSP etc.

Although the results are still not available so as to show the efficacy of using fuzzy for inverted pendulum on a cart. Simulation results on Matlab have shown high performance over conventional control. work on this system is to act as an intermediate step for Fuzzy based adaptive control of structural vibrations in a flexible structure.

Acknowledgements:
This paper would not have been possible without the help of Professor Ashab Mirza and Ahmad Hasan. They both took time out to explain concepts and basics, and provide books and material essential for this paper.

I am also greatly indebted to my colleague Rana Anees , who helped me out in every step.

References:

[1] IEEE Transactions on FUZZY SYSTEMS. February 2000 V8 No. 1.

[2] http://www.austinlinks.com/Fuzzy/overview.html

[3] Fuzzy Systems a Tutorial James F. Brule

[4] C. Lejewski "Jan Lukasiewicz", Encyclopedia of Philosophy,

Vol 5 ,Macmillan, NY:1967, pp 104- 107.

[5] http://www.austinlinks.com/Fuzzy/expert-systems.html

[6] http://www.geocities.com/CapeCanaveral/Launchpad/6000/ttfuzzy.html.

[7]Handbook of Control Engineering.© 1999 CRC Press, INC ISBN 81-7224-785-0

[8] Fuzzy Logic & NeuroFuzzy Applications in Business & Finance, Constantin Von Altrock, © 1995, Prentice Hall PTR, ISBN – 0-13-368465-2

[9] MultStage Fuzzy Control, Janusz Kacprzyk, © 1997, John Wiley & Sons Ltd.

[10] Fuzzy Engineering – Bart Kosko © Prentice Hall 1997- International Edition