Fuzzy Motion Control of the Ball on Beam

Alexandar Djordjevich
Department of Manufacturing Engineering and Engineering Management
City University of Hong Kong
83 Tat Chee Avenue, Kowloon
HONG KONG SAR, CHINA

Abstract: - Application of fuzzy logic to the well known "ball on beam" motion control problem is reported. Comparison results indicate clearly superior performance of the fuzzy controller to that of the traditional PID controller. The process of control synthesis is summarised.

Key-Words: - Ball on Beam Experiment, Fuzzy Control

1 Introduction

As illustrated in Figure 1, the objective is to control position *X* of the ball along a (grooved) beam by adjusting the beam tilt α. This primary task is achieved through the motor shaft position control that delivers the appropriate angle θ for the required tilt α – as shown in the Appendix I, the two angles are directly proportional when θ is "small". The subdivision into two independent tasks or subsystems implied hereby is possible because the motor dynamics can be made to be much faster then the ball dynamics and the transients of the former are then not seen by the latter.

The conventional wisdom would be to derive a mathematical model for both sub-systems and then design two PID controllers to satisfy some design requirements specified in advance. Indeed, such mathematical modelling has been carried out (in Appendix I). However, not only that it assumes incorrectly that the angle θ is "small" (hence $\theta=\sin\theta$) despite its 160° range (±80°), but nonlinear vibrations of the beam as well as ball hopping on the beam are difficult to account. In order to avoid (minimise) the occurrence of these effects, the actually calculated controller parameters had to be replaced by those adjusted based on the engineering insight and intuition. It was then decided to build a fuzzy controller for the ball-beam subsystem that may be better suited to such insight and intuition.

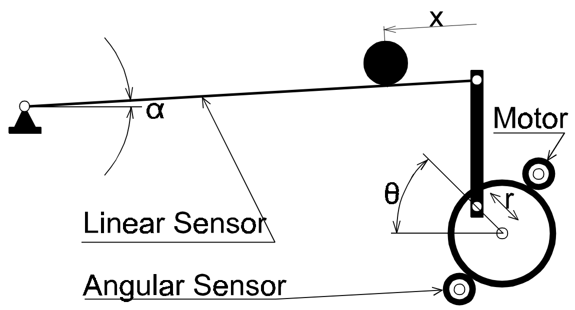


Figure 1: System kinematics

2 Motion System Description and Implementation of Fuzzy Logic

The motion system includes a beam on which a stainless steel ball can roll (Figure 1). The beam tilt α depends on the motor shaft position. The beam consists of two rails: one is a nickel-chromium wire-wound resistor and the other rail is a steel rod. As the ball rolls on such tracks, it acts like a wiper of a potentiometer, providing the voltage feedback on its position along the track (and forming the "linear sensor" in Figure 1).

Fuzzy rules used are of the form:

if

the ball is far to the right (large positive position error),

and if

the high ball velocity is away from the desired destination (high negative velocity),

then

the beam should be inclined severely by raising its driven end significantly.

Velocity information is obtained by numeric differentiation of the position feedback.

A total of 36 such rules and input conditions have been identified for the ball velocity and position error as there are two fuzzy variables, ball position and velocity, with six partitions each (high, medium and small, positive and negative with respect to the set point). Beam tilt is the output variable from the ball-beam fuzzy subsystem. It represents the input (set) point for the PID control of the motor dynamics that is to deliver the required beam-tilt based on the feedback from the sensor labelled “angular sensor” in Figure 1. As the PID motor position control is a rather routine task, only the ball-beam dynamics is discussed further.

It is apparent that the problem is anti-symmetric with respect to the zero position error point: changing the sign of both fuzzy variables (position and velocity) requires the exactly opposite beam tilt. Hence, only half of the fuzzy rules (18=36/2) had to be used explicitly: those for the position error positive were chosen. Any input condition with the negative position error was mapped into its counterpart with this error positive and with altered velocity direction, while flagging that the sign of the resulting beam tilt must then be altered. Such crisp reasoning is carried out by the following simple code upon obtaining each new ball position and velocity feedback and by multiplying the output tilt by the variable called “`MultiplyOutput`” that equals +1 or −1:

```
IF PositionError < 0 THEN
  PositionError = -PositionError
  Velocity = -Velocity
  MultiplyOutput = -1
ELSE
  MultiplyOutput = 1
END IF
```

Full list of the eighteen fuzzy rules is shown in Table 1 (next page).

Membership functions for the input variables, and consequence levels are determined using intuition and test trials. A rough guide used in this process is summarised next:

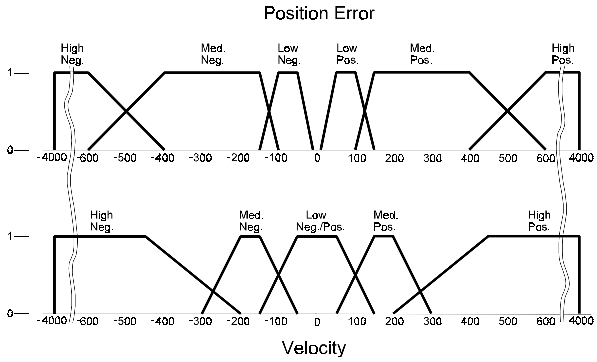


Figure : Fuzzy Partitions for Position and Velocity

a) The partition for "low velocity" reflects total unreliability of the calculated velocity in the ±50 region - fuzzy rules then do not utilise the motion direction information. Magnitudes above 50, with gradual increase in confidence, indicate the motion direction ("Medium velocity").

b) The "med." and "high" velocity partitions reflect the need for smoothness of the defuzzified command signal (output) with the gradual increase in velocity magnitude (the tilts are functions of the consequence levels).

c) For the "small" and "medium" position error, as well as the "down low" and "down low medium" consequence levels, final settling of the ball was observed. Consequence number 9 (horizontal track) is used implicitly when none of the rules are called upon : fuzzification of the position error did not include its entire domain as the region from −10 to +10 does not fall under any fuzzy partition. This region is further increased by the dead band of the drive system which does not respond to small inputs. The objective was to establish a zone within which the ball would stop by itself along the horizontal track when the ball enters this zone with the "medium" velocity. The control continues only if the ball overshoots to the other end. This called for the "med" velocity partition to be narrow around such ideal velocity for entering the zone (± velocity noise).

d) For the "med"/"big" position interface, system dumping was studied : by classifying the position error as "big" sooner, more agile control is obtained.

Table 1: Fuzzy Rules

INPUT CONDITION			**CONSEQUENCE**(required output level)	
Index	Position Error	Velocity	Required Move for the Driven End of the Track	Index of Output
1	High +	High +	Down Low Medium	3
2	High +	Med. +	Down Low Medium	3
3	High +	Low +	Down High	1
4	High +	Low -	Down High	1
5	High +	Med. -	Down High	1
6	High +	High -	Down High	1
7	Med. +	High +	Up High	8
8	Med. +	Med. +	Up Medium	7
9	Med. +	Low +	Down Low Medium	3
10	Med. +	Low -	Down Low Medium	3
11	Med. +	Med. -	Down High	1
12	Med. +	High -	Down High	1
13	Low +	High +	Up High	8
14	Low +	Med. +	Up Low Medium	6
15	Low +	Low +	Down Low	4
16	Low +	Low -	Down Low	4
17	Low +	Med. -	Down Medium	2
18	Low +	High -	Down High	1

It appears that the output state number 5 (up low) is not used as none of the 18 fuzzy rules calls for it. It should be noted, however, that this state is used with **negative** position error for input conditions corresponding to those positive ones which call for consequence 4 (down low). This is in connection to the switch from 36 to 18 rules as explained previously. An additional “do nothing” output state is used implicitly when none of the other rules apply, as indicated by the deliberately built-in dead band apparent in the middle of the membership function for position-error in Figure 2.

3 System Response and Comparison to That with a PID Controller

System response with the fuzzy controller is shown in Figure 3a and 3b. The large deviations observed are the manually introduced disturbances. The superimposed graphs in both pertain to the motor shaft position feedback from the “angular sensor” in Figure 1.

Significantly shorter settlement time can be observed than with the PID controller shown in Figure 4. While a predictably oscillatory character with the PID controller is easily noticeable, each run with the fuzzy controller is different, with the transient becoming unpredictable and subject to a seemingly chance outcome once the deviation from the set-point has dropped in magnitude to a minor value.

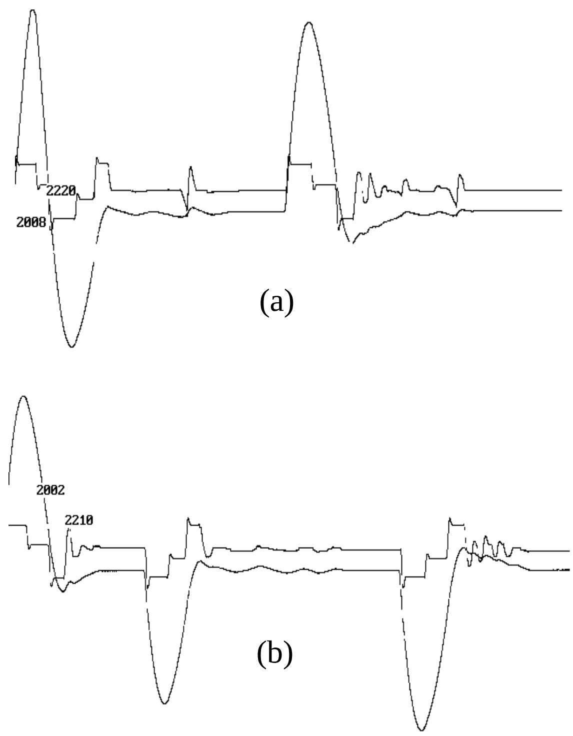


Figure 3: Sample System Response with Fuzzy Controller. Disturbances Were Introduced Manually. Shown Superimposed to the Ball-position Signal Is the Motor-shaft Position Feedback.

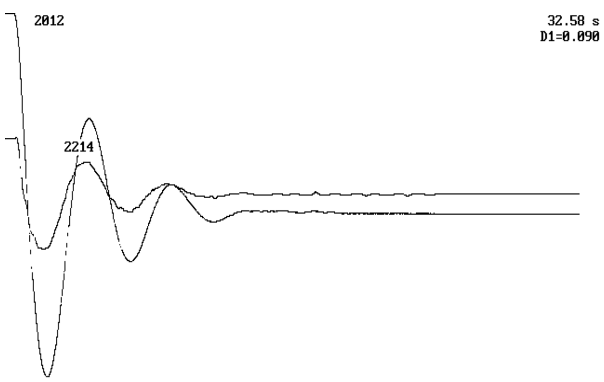


Figure 4: Response with the PID Controller.

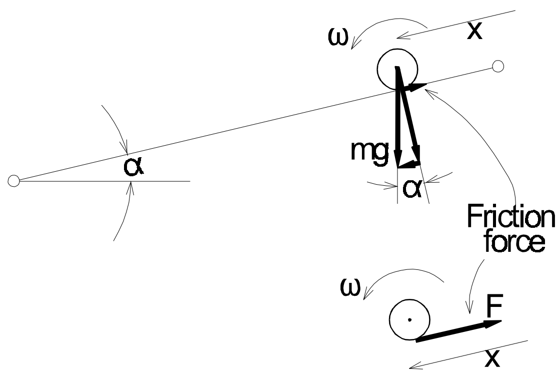
4 Conclusions

Two separate algorithms, for PID and fuzzy control, are implemented on the same "ball on beam" motion system. The original motivation was to illustrates to the students one implementation of a PID control. However, because the mathematical model does not accurately represent the system, ball hopping off the vibrating beam in particular, the rigorously calculated controller parameters could not be applied. Intuitively tuned values were used instead. This lack of rigour motivated the author to develop a fuzzy controller for the same system as it can better capture the intuitive insight involved. Application of the two controllers in a class of engineering students is shown in Appendix II.

Appendix I: PID Controller

The mathematical model is derived in three parts: 1) x-α ; 2) α-θ ; 3) θ-motor voltage.

1) x-α relationship



For the ball with mass m, moment of inertia J and radius R, it is :

$$mg\ \sin\alpha\ =\ m\ddot{x}\ +\ F$$

$$RF\ =\ \dot{\omega}J\ =\ \ddot{x}\frac{J}{R}$$

From the above, one gets :

$$mg\ \sin\alpha\ =\ m\ddot{x}\ +\ \ddot{x}\frac{J}{R^2}$$

As for the ball $r = \frac{2}{5} mR$, it follows :

$$mg \sin\alpha = \frac{7}{5} m\ddot{x}$$

and for small α : $\ddot{x} = \frac{5}{7} g\alpha$.

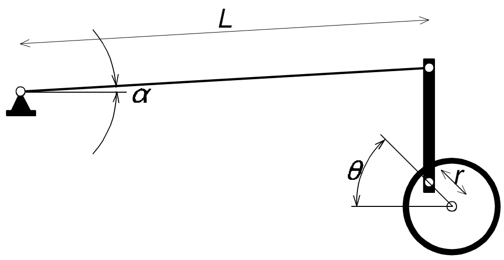
The corresponding transfer function is:

$\frac{X(s)}{\alpha(s)} = \frac{5}{7}\frac{g}{s^2}$ where x is the position of the ball along the track tilted α radians.

2) α-θ relationship

Assumption : with change in θ within the range of ±60° or so, the rotation of the rod connecting the track and the gear controlling it is small (since the rod is $\gg$ r).

Implication: both ends of the rod move approximately equal amounts in the vertical direction - since ***cos****(small rod rotation)≈1.*



Hence, $r \sin\theta = L \sin\alpha$ and, for small θ (although it is not small), $\theta = \frac{L}{r}\alpha$.

The third part of the mathematical model is the well known mathematical model of a permanent magnet DC motor.

Two PD controllers are then designed. The output of one controller is the required angle θ that is needed to affect the ball, and the other determines the output voltage for the motor so that this θ can be delivered. The P and D constants for two controllers, are (respectively): 0.312, 0299, 5.79 and 0.22 (for the design requirement of 0.707 for the damping ratio and peak time of 3 s for the ball and 0.2 for the motor.

Appendix II: Class Demonstration

The author brings the actual hardware (and computer) to the classroom for the demonstration. A number of students are then invited to control the ball position manually by rotating a wheel that affects the slope of the beam. This manual approach by a student volunteer is then named “intelligent control”. Students quickly rename it “unintelligent control” (or even worse) because no student has yet succeeded to stabilize the ball. Students who make the loudest remarks are then invited to try their own skills. By realising their own inability to make any headway, students gain respect for the subsequent quick and seemingly flawless performance by the PID control algorithm – no amount of talking could achieve the same effect. Weeks later, the merit of automation as being much more than merely a means of labour replacement, even if inexpensive labour force is abundant, sinks in far more easily.

After the wave of **“PID is good”** excitement has calmed down, the author points to the students the undesirable aspects of the system performance. He also points to the difference between the controller parameters used and those predicted by calculations. They then jointly try to evaluate (i) how accurately they know some of the “constants” in the mathematical model, (ii) how inaccurate that model actually is despite its complexity, (iii) whether they had an alternative to the approximations introduced and (iv) how variable the system “constants” are during the motion. They actually set the controller parameters as determined by calculation, and observe very poor performance due to the cause not included in the “sound” model. They then question the merit of doing the seemingly precise modelling and PID system synthesis only to plug into it at the end quite rough estimates of the system constants and to adjust the final results based on engineering insight. While at this stage PID is still not considered “bad”, this has set a stage for the fuzzy control approach that starts with such engineering insight in the first place and does not maintain the notion of being very exact.

Another wave of excitement follows when fuzzy control is demonstrated on the same problem. As the ball goes directly to where it is supposed to, "no**, fuzzy is good**" is heard. Once this second wave of excitement has calmed down, they try to identify what the key factors were in making the whole approach work in this particular case and how this relatively complex problem was partitioned into simpler subsets. They then move to the negative aspects. The author brings to the students' attention the shimming under the leg of the table and where the need for the table to be horizontal came from – something not a problem for the PID algorithm. He points that, while seemingly alike due to a dominant characteristic, each run is different, unpredictable and influenced by an one chance outcome. He reveals that the control is essentially limited to a set of conditions it had been fine-tuned for, which is far more restrictive than what the PID algorithm was able to accommodate. They then go back to PID.

With PID running for the second time around, the author varies controller parameters on-line trying to identify the "optimal" values for different stages of the ball transient on its way to the final destination and for the steady state thereafter. While the traditional design wisdom would be to finally select a single value for each parameter and thereby strike a compromise between contradictory design requirements, they instead verbally articulate rules how those parameters should ideally vary during the process that they observe repeatedly. In essence, these are fuzzy rules, that bring them back to the fuzzy control, but this time fuzzy is used to adjust parameters of the PID on-line to suit each stage of the process – rather than to firmly fix them based on some compromise.

The author and his students conclude by realising that they made two full circles across different methods and ended up with a hybrid that combines useful features of them all.