B00060572 David Kelly

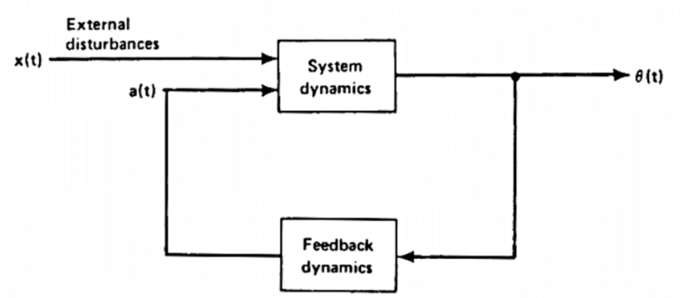
Introduction

The task given is to develop an inverted pole system from the sample application provided through FuzzyLite and implement the system in Java.

I Implemented to original version shown in the lecture notes and decided to develop my own version. While developing I discovered that the parameter values for the terms are imperative to the functioning of the system. I was consistently getting out of bound errors with input values from the cart exceeding my set values. The solution was to maintain the values shown in the sample but reduce the terms for input and output variables.

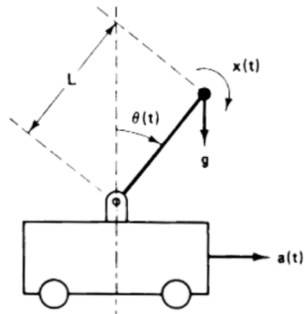
Although my version did not seem as measured as the sample it was never my intention to have a system that did not show movement. I wanted to show the near absolute values the system could take before output action was needed to balance the pole. Because of this approach I had a cart that was in constant motion but magnifies the actions of the system.

Feedback system



In order to calculate the output force of the cart the system must take inputs from the angle of the pole and the angular velocity of the pole. The system is in a cycle so feedback is needed to asses what the next output force must be to maintain the pole in an upright position.

Linguistic variables



θ(t) = angle pole makes with perpendicular
x(t) = Angular velocity of pole
a(t) = Force applied to cart
L = length of pole
g = gravity

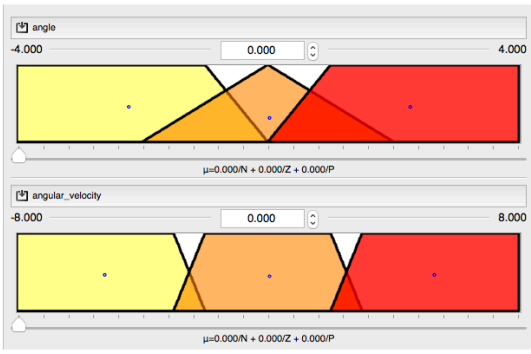
As shown in the image above there are 3 linguistic variables and 2 constants. Using the linguistic variables *angle*, *angular_velocity* and *force* it is possible to create fuzzy controller that dynamically changes the force applied to the cart depending on the values for the *angle* and *angular_velocity*. It is important to set a domain range for these variables to operate within. For the inverted pendulum I set the values as shown below for each variable.

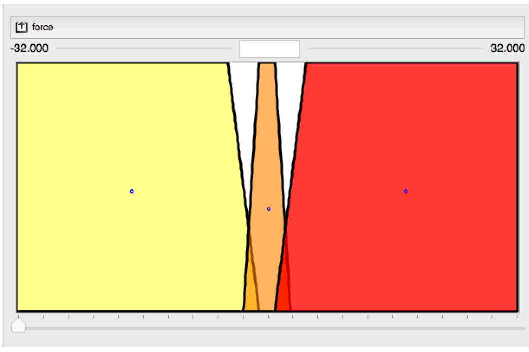
Domain range $-4 \leq x1 \leq 4$ and $-8 \leq x2 \leq 8$ and $-32 \leq \mu \leq 32$

Terms
Input Variables
Angle Terms: N = Negative, Z = Zero, P = Positive :
N(-4,-4,1,0) Z(-2,0,2) P(0,1,4,4)
Angular Velocity Terms: N = Negative, Z = Zero, P = Positive :
N(-8,-8,3,2) Z(-3,-2,2,3) N(2,3,8,8)

Output Variable
Force Terms: N = Negative, Z = Zero, P = Positive :
N(-32,-32,-5,-1) Z(-3,-1,1,3) P(1,5,32,32)

The minimum number of terms possible to maintain the balance of the pole is nine. Three terms for each variable. The terms are created with border values for minimum and maximum. Using triangles and trapezoids to show a discrete values membership of a term, can help decide the final defuzzified output measured by the center of gravity.





I found that making the angular velocity a trapezoid I was able to reduce the amount of movement in the cart. This was due to the angular velocity having a larger area at zero which in the decision matrix had an output of zero.

Decision matrix
The decision matrix I created is a simple 3x3. After initially implementing a 5x5 as in the sample example provided, I felt to show the cycle in it's purest functioning form I could simplify the rules. N= Negative, Z=Zero, P=Positive. These labels all relate to value of angle, angular velocity and the matrix shows the output force.

	ANGLE		
ANGULAR_VELOCITY	N	Z	P
N	N	N	Z
Z	N	Z	P
P	Z	P	P

Inference rules

The rules determine the actions taken to create the output variable value. These rules are created via the decision matrix and must match the syntax of the FuzzyLite system. Potentially if the system has many rules with the same output then it is not necessary to create a unique rule for each condition. In my system each rule is necessary to achieve a successful implementation.

if angle is N and angular_velocity is N then force is N
if angle is Z and angular_velocity is N then force is N
if angle is P and angular_velocity is N then force is Z
if angle is N and angular_velocity is Z then force is N
if angle is Z and angular_velocity is Z then force is Z
if angle is P and angular_velocity is Z then force is P
if angle is N and angular_velocity is P then force is Z
if angle is Z and angular_velocity is P then force is P
if angle is P and angular_velocity is P then force is P

Conjunction function

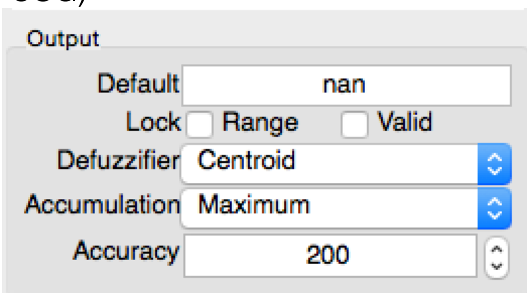
Since our rule base contains connective statements such as AND we will need to define how these Boolean operators are handled by the Fuzzy system. For the purpose of this example we will use the MAX and MIN interpretation for OR and AND respectively.

Accumulation function

The choice of accumulation function and defuzzifier is vitally important as this is how our Fuzzy system produces a CRISP value which we can use as the force (µ) on the cart to keep the pole balanced

Defuzzifier function.

Centroid (center of gravity COG)



To convert the fuzzy logic value back to an output value a defuzzifier must be set. The centroid defuzzifier finds the centre of gravity for the output terms selected. Depending on the membership of a term the centroid may be an output value between two fuzzy output values.

Sample

Input(angle = 2, angular_velocity = 0) Output force = 17.4
Input(angle = 1.3, angular_velocity = -0.7) Output force = 16.5
Input(angle = -3, angular_velocity = 4) Output force = 0