Computational Intelligence

Assessment 1

Part A – Inverted Pendulum

Nicholas Randles – B00058026

Institute of Technology Blanchardstown
Institiúid Teicneolaíochta Baile Bhlainséir

In this part of the assessment we implemented an inverted pendulum. This is a pendulum that has its centre of mass above its pivot point. Unlike a normal pendulum an inverted pendulum is relatively unstable. Because of this we must constantly adjust the positioning of the pivot point as part of a feedback system.

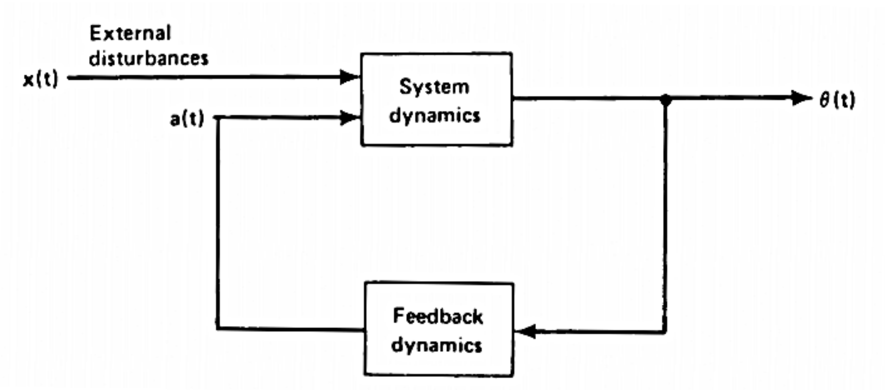

Figure 1: Feedback system

We need to create a set of linguistic variables that properly represent the input and output variables. Each variable needs a domain range of values. It is important that these values are appropriate to ensure the program runs effectively. In this program we have limited the domains of the linguistic variables as follows:

- Let x1 = !(t), x2 = x(t) and μ = a(t)
- Let -4 ≤ x1 ≤ 4 and -8 ≤ x2 ≤ 8 and -32 ≤ μ ≤ 32

After this we choose a membership function for each term and choose what values in the domain it will be assigned to.

In this program the angle made between the pole and the perpendicular ranges between -4 & 4 which represents and range of -40° ≤ !(t) ≤ 40°.

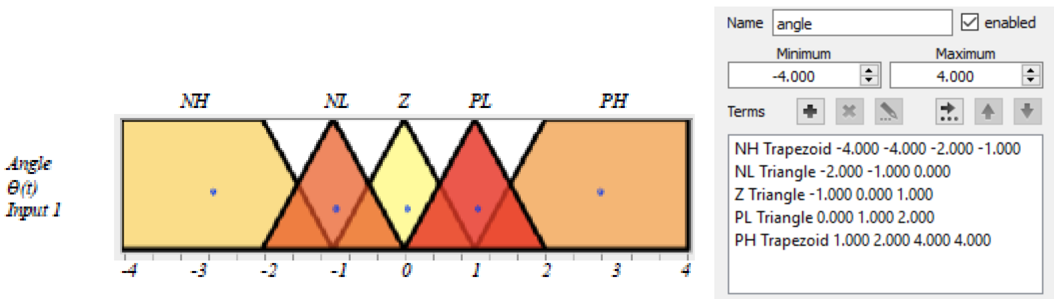

Figure 2: Angle

The angular velocity ranges between -8 & 8 which represents -8dps ≤ x(t) ≤ 8dps (degrees per second).

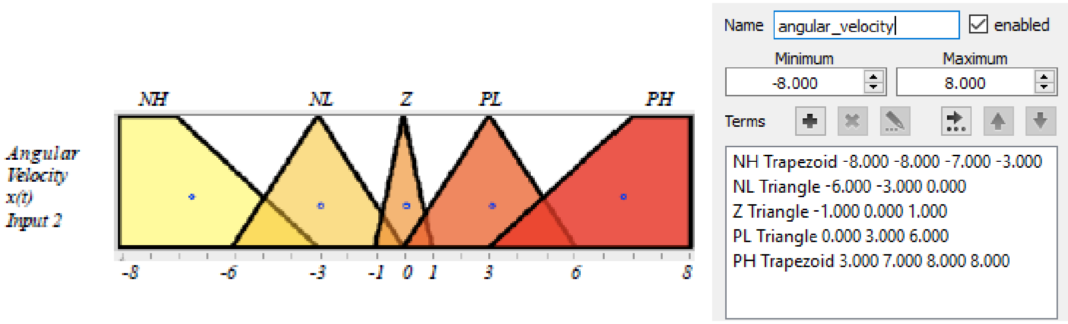

Figure 3: Angular Velocity

The force applied to the cart ranges between -32N ≤ μ ≤ 32N (newtons).

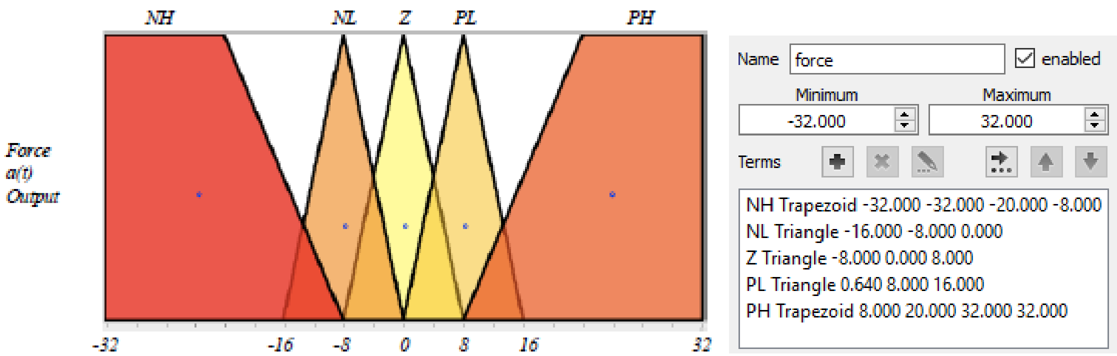

Figure 4: Force

After we set a domain range for each variable we give each variable a set of terms which are used to describe it. For this program we will use the following terms outlined in table 1 to divide the domain values into more understandable sections.

Term	**Description**
PH	Positive High
PL	Positive Low
Z	Zero
NL	Negative Low
NH	Negative High

Table 1: Terms

After we have set the terms we have to write a set of rules that will take in the angle and angular velocity and output a suitable value for the force variable based on the input. We use a decision matrix to plot out the possible values for angle and angular velocity. The centre of the grid will show the desired values for the output value force.

	Angle(x1)				
Angular Velocity(x2)	**NH**	**NL**	**Z**	**PL**	**PH**
NH	NH	NH	NH	NL	Z
NL	NH	NL	NL	Z	PH
Z	NH	NL	Z	PL	PH
PL	NH	Z	PL	PH	PH
PH	Z	PL	PH	PH	PH

Table 2: Decision Matrix

After we finish the decision matrix we can use it to write a set of rules. The decision matrix in table 2 has 25 possible outcomes but in this program we will only use 11. This is mainly because we have combined some of them using the && operator.

1. if angle is Z and angular_velocity is Z then force is Z
2. if angle is Z and angular_velocity is NH then force is NH
3. if angle is Z and angular_velocity is NL then force is NL
4. if angle is Z and angular_velocity is PL then force is PL
5. if angle is Z and angular_velocity is PH then force is PH
6. if angle is NH and angular_velocity is Z then force is NH
7. if angle is NL and angular_velocity is Z then force is NL
8. if angle is PL and angular_velocity is Z then force is PL
9. if angle is PH and angular_velocity is Z then force is PH
10. if angle is NL and angular_velocity is PL then force is Z
11. if angle is PL and angular_velocity is NL then force is Z

Now we will need to use defuzzication to produce a useful outcome from the fuzzy data. Since the rules we are using contain connective statements such as AND we have to define how they are handled. In this program we use the MAX and MIN interpretation.

Then we select an accumulation and defuzzifier function for the output. It is important we chose the most suitable accumulation function and defuzzifer function to ensure the fuzzy system produces a CRISP value which we can use as the force for the cart. For this program the accumulation function we use is MAXIMUM and for the defuzzifier function we use is CENTROID.

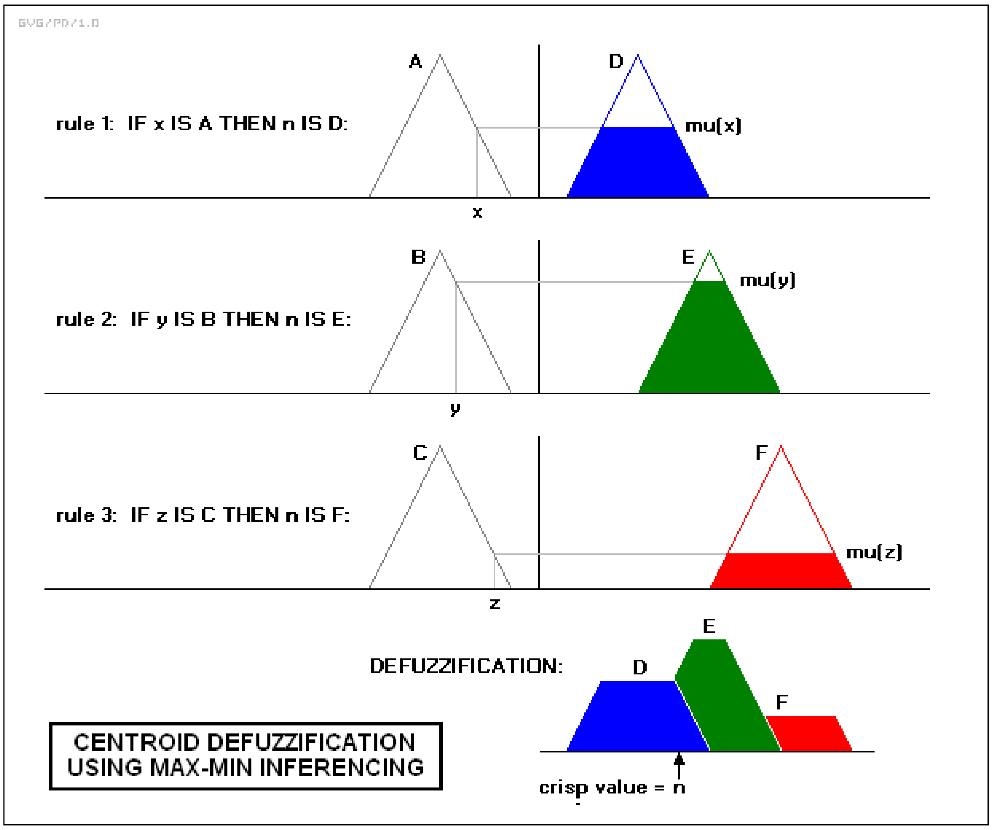

Figure 5: Defuzzification