|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| theta/ theta\_dot | in\_nl | in\_ns | in\_ze | in\_ps | in\_pl |
| in\_nl | out\_nvl | out\_nvl | out\_ze | out\_pl | out\_pl |
| in\_ns | out\_nvl | out\_nl | out\_ze | out\_pl | out\_pl |
| in\_ze | out\_nl | out\_nl | out\_ze | out\_pl | out\_pl |
| in\_ps | out\_nl | out\_nl | out\_ze | out\_pl | out\_pvl |
| in\_pl | out\_nl | out\_nl | out\_ze | out\_pvl | out\_pvl |

Table1: Output Rules for theta vs theta\_dot

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| x/x\_dot | in\_nl | in\_ns | in\_ze | in\_ps | in\_pl |
| in\_nl | out\_nm | out\_nm | out\_ze | out\_pm | out\_pm |
| in\_ns | out\_nm | out\_nm | out\_ze | out\_pm | out\_pm |
| in\_ze | out\_nm | out\_nm | out\_ze | out\_pm | out\_pm |
| in\_ps | out\_nm | out\_nm | out\_ze | out\_pm | out\_pm |
| in\_pl | out\_nm | out\_nm | out\_ze | out\_pm | out\_pm |

Table2: Output Rules for x vs x\_dot

Defuzzification of Output:

|  |  |
| --- | --- |
| Output Name/Class | Force Value |
| out\_nvl | -120 |
| out\_nl | -40 |
| out\_nm | -13 |
| out\_ns | -5 |
| out\_ze | 0 |
| out\_ps | 5 |
| out\_pm | 13 |
| out\_pl | 40 |
| out\_pvl | 120 |

Table3: Since the process used is Sugeno (Zero Order), therefore the force values needed to be constants.

Input Membership Functions Shapes:-

1. x membership function

-0.2 -0.15 0 0.15 0.2

in\_nl in\_ns in\_ze in\_ps in\_pl

2. x\_dot membership function

-1.6 -1.5 -0.2 -0.1 0.1 0.2 1.5 1.6

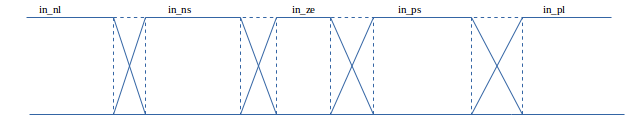
in\_nl in\_ns in\_ze in\_ps in\_pl

3. theta membership function

-0.5 -0.4 -0.05 0 0.05 0.4 0.5

in\_nl in\_ns in\_ze in\_ps in\_pl

4. theta\_dot membership function



-0.2 -0.15 -0.10 -0.08 0.08 0.10 0.15 0.2

Additional Notes:

1. Membership functions:- These explain how we take the input from crisp value to a fuzzy state. The trapezoidal diagrams are analogous to the declarations in initMembershipFunctions in fuzzyLogic.cpp.
2. Output Rules:- There are two tables denoting these rules. The first table is theta vs theta\_dot, and we can check values for output corresponding to the values of inputs.

The second table is x vs x\_dot. Although the x vs x\_dot table seems unintuitive at first, there is a reasoning to why it has been chosen and why it works. We have given more importance to theta. Theta values beyond a certain value get a very large force setting. When cart drifts to a larger x value, we deflect it towards an even bigger x; as we’ve placed emphasis on theta, theta increases in a negative x direction, and cart eventually gets pushed to the correct direction.

1. Defuzzification Values for Output:- These values are recovered from the Defuzzification of Output Table, i.e. Table2 in the Documentation.odt and Documentation.docx files.

The algorithm can balance the cart till 22 degrees.