Project title: “Traffic light controller”
Project description:

Traffic congestion is one of the serious problems we face daily and it waste our time which effects on our productivity and that problem solved by traffic lights but due to the increasing of number of vehicles. Fixed time controlled traffic light can’t solve the high traffic congestion.

The goal of the traffic congestion problem is to reduce road delays by making efficient use of existing traffic signaling systems without creating new roads.

Traffic systems have many parameters such day, time, weather and season and many other situations. So we can solve the problem of traffic jam by using intelligent traffic control system based on fuzzy logic that can adjust the timings of traffic lights continuously depending on the current variables and traffic jam.

Fuzzy if-then rules can implement the rules that police man use to manage traffic lights.

Our fuzzy traffic control system aim to adjust the cycle time of green light signal based on the number of vehicles to improve traffic protection at the intersection (junctions), usage of junction at its maximum level and minimize the delays.

Linguistic variables and terms:

We have 2 inputs in our fuzzy traffic signals controller:

Passing vehicles (PV): number of vehicles passing during green light, there are detectors at the fix point of each lane of the intersection that detects the vehicles passing through it.

Waiting vehicles (WP): number of vehicles waiting during red light, which is detected by the detector installed in the intersection.

PV Linguistic variables {Low, Medium, High}

WV Linguistic variables {Low, Medium, High}

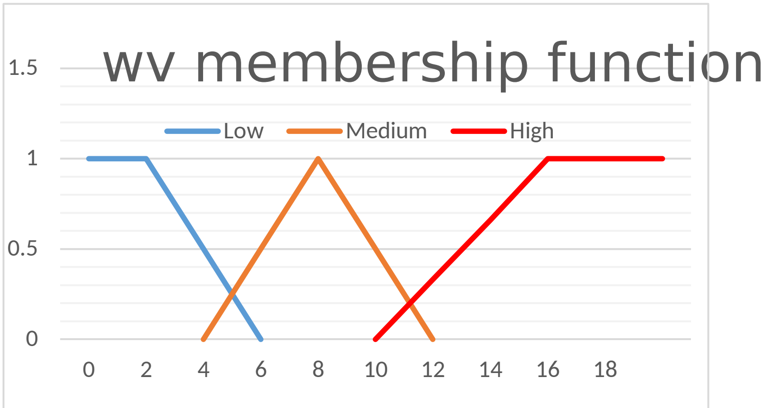
The output will be the **Green light duration (GLD)** which is the time duration of the green traffic light on the basis of the current congestion situation of the intersection.

GLD Linguistic variables {short, medium, long}

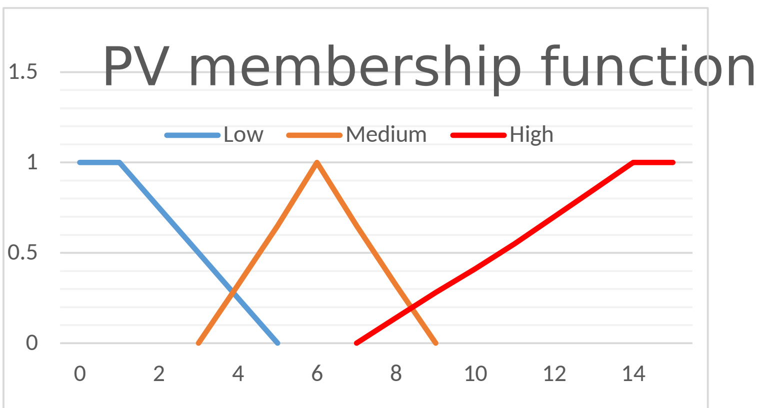
Knowledge base layout:

Input		**Output**
Waiting Vehicles	Passing Vehicles	Green light Duration
L	**L**	**S**
L	**M**	**M**
L	**H**	**L**
M	**L**	**S**
M	**M**	**M**
M	**H**	**M**
H	**L**	**S**
H	**M**	**S**
H	**H**	**S**

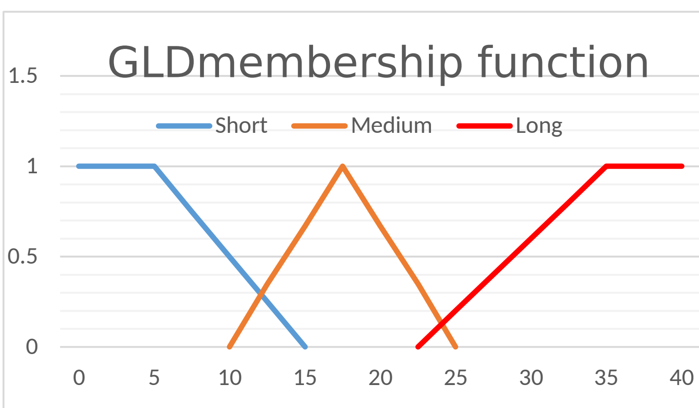
Membership function for waiting vehicle variables:



Membership function for passing vehicle variables:



Membership function for Green light duration variables:



Knowledge base rules:

IF waiting vehicles = Low **AND** passing vehicles = Low **THEN** green light duration is **Short**

IF waiting vehicles = Low **AND** passing vehicles = Medium **THEN** green light duration is **Medium**

IF waiting vehicles = Low **AND** passing vehicles = High **THEN** green light duration is **Long**

IF waiting vehicles = Medium **AND** passing vehicles = Low **THEN** green light duration is **Short**

IF waiting vehicles = Medium **AND** passing vehicles = Medium **THEN** green light duration is **Medium**

IF waiting vehicles = Medium **AND** passing vehicles = High **THEN** green light duration is **Medium**

IF waiting vehicles = High **AND** passing vehicles = Low **THEN** green light duration is **Short**

IF waiting vehicles = High **AND** passing vehicles = Medium **THEN** green light duration is **Short**

IF waiting vehicles = High **AND** passing vehicles = High **THEN** green light duration is **Short**

Description of the fuzzification process:

In this process crisp inputs from the domain are transformed into fuzzy inputs with help of membership function. Input variables are converted into suitable values from the domain of discourse. There are three linguistic variables assigned to each input factor.

Waiting vehicles membership functions:

$$F_{low}(x) = \begin{cases} 0 & , \quad x > 6 \\ \frac{6-x}{4} & , \quad 2 \leq x \leq 6 \\ 1 & , \quad x < 2 \end{cases}$$

$$F_{medium}(x) = \begin{cases} 0 & , \quad x \geq 12 \\ \frac{x-4}{4} & , \quad 4 < x \leq 8 \\ \frac{12-x}{4} & , \quad 8 < x < 12 \\ 0 & , \quad x \leq 4 \end{cases}$$

$$F_{High}(x) = \begin{cases} \frac{x-10}{6}, & 10 \leq x \leq 16 \\ 1, & x > 16 \\ 0, & x < 10 \end{cases}$$

Passing vehicles membership functions:

$$F_{low}(x) = \begin{cases} 0 & , \quad x > 5 \\ \frac{5-x}{4} & , \quad 1 \le x \le 5 \\ 1 & , \quad x < 1 \end{cases}$$

$$F_{medium}(x) = \begin{cases} 0 & , \quad x \ge 9 \\ \frac{x-3}{3} & , \quad 3 < x \le 6 \\ \frac{9-x}{3} & , \quad 6 < x < 9 \\ 0 & , \quad x \le 3 \end{cases}$$

$$F_{High}(x) = \begin{cases} \frac{x-7}{7}, & 7 \le x \le 14 \\ 1, & x > 14 \\ 0, & x < 7 \end{cases}$$

Green light duration membership functions:

$$F_{low}(x) = \begin{cases} 0 & , \quad x > 15 \\ \frac{15-x}{10} & , \quad 5 \le x \le 15 \\ 1 & , \quad x < 5 \end{cases}$$

$$F_{medium}(x) = \begin{cases} 0 & , \quad x \ge 25 \\ \frac{x-10}{6} & , \quad 10 < x \le 16 \\ \frac{25-x}{9} & , \quad 16 < x < 25 \\ 0 & , \quad x \le 10 \end{cases}$$

$$F_{High}(x) = \begin{cases} \frac{x-22}{13}, & 22 \le x \le 35 \\ 1, & x > 35 \\ 0, & x < 22 \end{cases}$$

Description of the inference algorithm:

Mamdani Fuzzy Inference Systems:

Mamdani fuzzy inference was first introduced as a method to create a control system by synthesizing a set of linguistic control rules obtained from experienced human operators. In a Mamdani system, the output of each rule is a fuzzy set. Since Mamdani systems have more intuitive and easier to understand rule bases, they are well-suited to expert system applications where the rules are created from human expert knowledge.

The output of each rule is a fuzzy set derived from the output membership function, these output fuzzy sets are combined into a single fuzzy set. to compute a final crisp output value, the combined output fuzzy set is defuzzified.

Here we have two input each input have 3 linguistic variables. In mamdani we have 5 layers:

Layer 1: Outputs are same with the inputs.

Layer 2 : Fuzzification.

Layer 3: Perform logical AND operations.

Layer 4 : Carries out the task of fuzzy inference.

Layer 5 : Defuzzification.

There is the following steps to calculate the outputs in the mamdani:

Step 1: Set of fuzzy rules need to be determined in this step.

Step 2: In this step, by using input membership function, the input would be made fuzzy.

Step 3: Now establish the rule strength by combining the fuzzified inputs according to fuzzy rules.

Step 4: In this step, determine the consequent of rule by combining the rule strength and the output membership function.

Step 5: For getting output distribution combine all the consequents.

Step 6: Finally, a defuzzified output distribution is obtained

Cases needed to combine rules (aggregation):

First case GLD is short:

IF waiting vehicles = Low **AND** passing vehicles = Low **THEN** green light duration is **Short**

IF waiting vehicles = Medium **AND** passing vehicles = Low **THEN** green light duration is **Short**

IF waiting vehicles = High **AND** passing vehicles = Low **THEN** green light duration is **Short**

IF waiting vehicles = High **AND** passing vehicles = Medium **THEN** green light duration is **Short**

IF waiting vehicles = High **AND** passing vehicles = High **THEN** green light duration is **Short**

Second case GLD is medium:

IF waiting vehicles = Low **AND** passing vehicles = Medium **THEN** green light duration is **Medium**

IF waiting vehicles = Medium **AND** passing vehicles = Medium **THEN** green light duration is **Medium**

IF waiting vehicles = Medium **AND** passing vehicles = High **THEN** green light duration is **Medium**

Description of the defuzzification:

The defuzzification is the process of converting the fuzzy values to crisp values.
We will use the weighted mean method to do the defuzzification step.

The weighted mean method is the most frequently used in fuzzy applications as it is one of the most computationally efficient methods. It is given by the algebraic expression

$$z = \frac{\sum \mu(z) z}{\sum \mu\,(z)}$$

Where, ∑denotes the algebraic sum and where z is the centroid of each membership function. The weighted average method is formed by weighting each membership function in the output by its respective maximum membership value.

Architecture diagram illustrating the main components of the designed project: