Fuzzy System Model for Management of Driver Distractions in Motor Vehicles

Adnan Shaout and Dominic Colella
The Electrical and Computer Engineering Department
The College of Engineering and Computer Science
The University of Michigan - Dearborn
Dearborn, MI
shaout@umich.edu; dcolella@umich.edu

ABSTRACT

In this paper a low cost and driver's environment friendly design of a Fuzzy Logic software system to manage driver distractions in a motor vehicle is presented. The system uses four inputs; vehicle speed, radio volume setting, frequency of left or right hand turns per minute and brightness conditions external to the vehicle. The system provides a single output in the form of a Driver Attention Load rating. This rating is used as a parameter to determine the degree to which the driver's environment needs to be adjusted in terms of radio volume level, brightness of instrument cluster display and reducing the amount of connected phone interruptions per minute. The Fuzzy Inference Software System is modeled and simulated using MATLAB. After simulation, the final system and associated graphical user interface are designed as a standalone application written in Java. An open source Java library called jFuzzyLogic is used to model the Fuzzy Inference System and the Java Swing toolkit is used for the design of the graphical user interface.

Keywords: **Driver Attention Load Rating; Driver Distractions; Fuzzy Inference System; Java Swing; jFuzzyLogic; MATLAB**

Date of Submission: March 18, 2015 Date of Acceptance: May 13, 2015

1. Introduction

The concept of "distracted driving" may seem to have emerged relatively recently with the onset of mobile technologies, but distracted drivers have been a danger to others since the mass production of the automobile. Many factors impact a driver's ability to maintain concentration while operating a vehicle such as radio volume, cellular phone use, vehicle speed, or daylight brightness conditions. Recent studies have shown that individuals operating motor vehicles require a similar level of concentration when compared with other activities such as airline pilots or surgeons operating on patients [7, 11]. Despite staggering findings such as these, the number of motor vehicle accidents in the united states continues to grow with each passing year. As of 2012, 421,000 individuals were injured in vehicle accidents involving distracted drivers, an increase of nine percent from the previous year [6]. The proposed system in this paper will be designed to reduce the amount of distractions on drivers as their driving situation changes to require further attention.

1.1 Objective

Driver monitoring systems have been engineered by automobile OEMs since the early 21st century. Toyota has had a system in production since 2006 that monitors eye movements with cameras and directs the driver's attention to the windshield when it determines their attention has lapsed [5]. Many other OEMs have systems similar to these but they are problematic for two reasons. Frist, these systems are typically only found in luxury brands of vehicles and, even in these cases, they are optional content requiring $1000s in extra costs. Second, these systems only monitor the driver's attention to the windshield but do nothing to modify the state of the vehicle's environment that they are operating in. Shaout and Tonshal [12] presented a Driver Activity Index (DAI) using fuzzy logic to measure how busy the driver is. They used two key driving parameters to determine the driver activity index; Acceleration Pedal (ACC Pedal) and steering wheel angle measure.

1.2 Proposed Solution

This paper will present a fuzzy system that consumes data from the vehicle's infotainment, powertrain, and steering systems to assess a factor labeled as the Driver's Attention Load. The Driver Attention Load will be used as a modifier to directly adjust other factors within the vehicle environment that impact the ability of the driver to maintain concentration. Two problems this system will attempt to solve are the issues with common OEM driver management systems that are currently in production: cost and impacting the driver's environment. Low cost will be achieved by designing a system that can be implemented as a background software process within an existing vehicle ECU such as the vehicle's instrument cluster or body controller. The use of existing hardware on the automobile for measuring the driver attention load will reduce the high hardware cost required in other driver monitoring systems where cameras and/or sensors are required. The system will also reduce the amount of distractions on the driver as opposed to just forcing the driver's attention back to the task of vehicle operation as in current production OEM systems. This paper will cover the fuzzy inference system design, modeling of the system for proof of concept in MATLAB, and implementation in a portable and efficient programming language such as Java that can be easily implemented on commonly used automotive microcontrollers. The paper is organized as follows: section 2 presents the fuzzy system design, section 3 introduces the Matlab system simulation, section 4 presents the Java application design, section 5 presents the comparison between the fuzzy system and non-fuzzy system (classical) and section 6 presents concluding remarks.

2. Fuzzy System Design

The Fuzzy System proposed in the project will be designed using a five step process: define system I/O, define crisp to fuzzy relations of I/O, create the fuzzy rule base, define fuzzy inference technique, and define method to generate crisp output from fuzzy output of system.

2.1 Design of System I/O

The main goal of this proposed system is to adjust the vehicle environment to ensure that the driver can devote the highest level of attention to driving the vehicle by reducing distractions. The proposal is to reduce the driver's distractions in relation to the amount of concentration required to operate the vehicle under certain conditions. Four inputs are defined that, when increased, would require more of the driver's attention:

- Vehicle speed
- Frequency of turns
- Radio volume
- Sunlight conditions

Vehicle speed was chosen due to a 2009 National Highway Traffic Safety Administration (NHTSA) study that stated that speeding was noted as a critical contributing factor in 99% of fatal vehicle accidents [1]. Frequency of turns was chosen due to a 2009 NHTSA study that found that 36% of vehicle collisions involved vehicles that had recently made turns at intersections [9]. Radio volume was chosen due to a 2000 study that found that higher volume levels negatively impacted a driver's reaction times [10]. Sunlight conditions were chosen due to a 2009 National Safety Council study that found that vehicle collisions were 300% more likely to occur during nighttime or low light conditions when compared to daylight conditions [4]. These four pieces of data were also chosen due to the fact that common motor vehicle electrical systems already contained monitoring equipment to capture the information. This reduces cost of implementing a system like this by not requiring any special sensor hardware for data collection. Powertrain systems will already be able to provide vehicle speed, most modern suspension system controllers already capture turning rates for power steering systems, modern automatic climate control systems consume sunlight conditions for thermostat control, and the vehicle's infotainment system can easily provide the current radio volume level. All of this information can be transmitted via serial data networks like CAN or FlexRay to a centralized control that implements the fuzzy system, easily implemented in existing vehicle modules such as instrument clusters or body controllers.

Volume input data is specified on a scale between 0 and 63 steps based on a common number of volume steps found in many vehicle infotainment systems. The volume input is the actual customer set volume step, not the actual power level of the radio's amplifier output. Frequency of turn data is specified in turns per minute between, 0 and 5, in which a vehicle makes a maneuver requiring more than a 45 degree left or right adjustment from center to the vehicle's heading position. Vehicle speed input data is specified in miles per hour between 0 and 100. Sunlight is specified by readings from the vehicle's sun load sensor, measure in units called Lux, on a scale from 0 to 10,000 lx where 0 is a dark, moonless night and 10,000 is a bright, sunny day [8].

The output variable of our fuzzy system is a parameter labeled as the Driver Attention Load (DAL). The DAL is specified as a parameter between 0 and 100% that defines that amount of concentration required by the driver to correctly operate the motor vehicle. The DAL is used as a multiplier to adjust three factors that impact a driver's ability to concentrate: radio volume, cellular phone interruptions, and brightness of vehicle information displays. The DAL will impact radio volume by modifying an attenuation level that will be applied to the amplifier's output level, this will be a volume level adjustment measured in decibels between 0 and -10 dB and will cause the radio volume to be slightly lowered as vehicle conditions require increasingly more of the driver's attention. The DAL will impact cellular phone interruptions by defining a maximum number of allowed interruptions to the driver by a factor between 0 and 5 interruptions per minute. As the vehicle conditions require more concentration, the amount of cellular phone interruptions allowed will be decreased until, when optimal concentration is required, all calls, SMS messages, or notifications will not be provided to the driver. The brightness conditions of information displays is affected by slightly lowing the brightness of all non-vehicle critical information on displays until only the speedometer is visible. Figure 1 shows the proposed system block diagram.

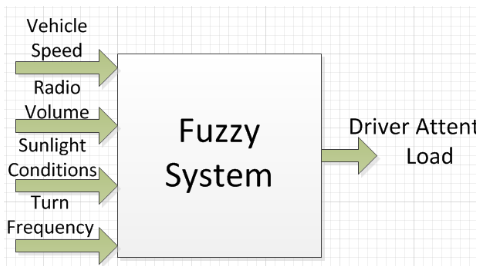


Figure 1: Fuzzy System I/O

2.2. Define Membership Functions

The membership functions are defined based on intuition using existing knowledge gained from research for this project. For the radio volume input, three linguistic variables are defined:

- Low
- Medium
- High

Membership functions for radio volume linguistic variables are defined in figure 2.

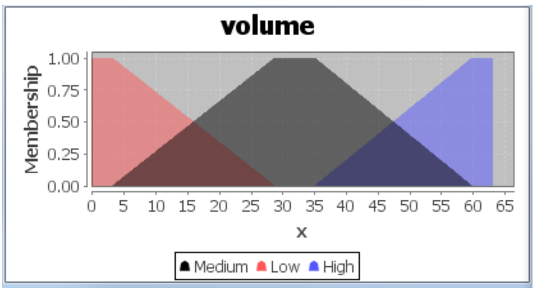


Figure 2: Radio Volume Membership Functions

For the frequency of turns input, there are two linguistic variables defined:

- High
- Low

The membership functions for frequency of turns are defined in figure 3.

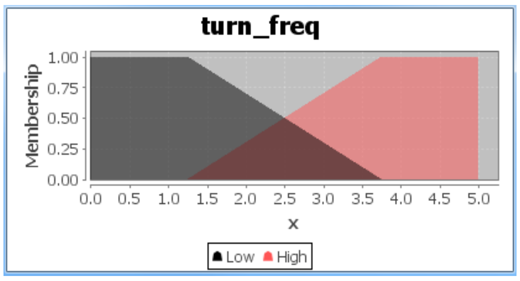


Figure 3: Frequency of Turns Membership Functions

For the sunlight input, there are two linguistic variables defined:

- Dark
- Light

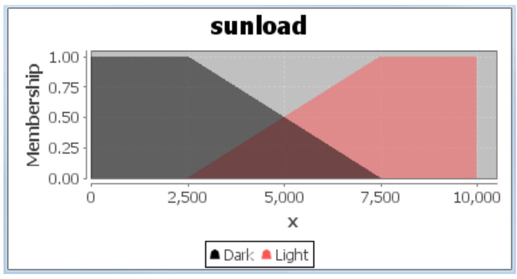


Figure 4: Sunlight Condition Membership Functions

The membership functions for sunlight conditions are defined in figure 4.

For the Vehicle Speed input, there are three linguistic variables defined:

- Dark
- Light

The membership functions for Vehicle Speed are defined in figure 5.

For the Driver Attention Load output, there are five linguistic variables defined in figure 6.

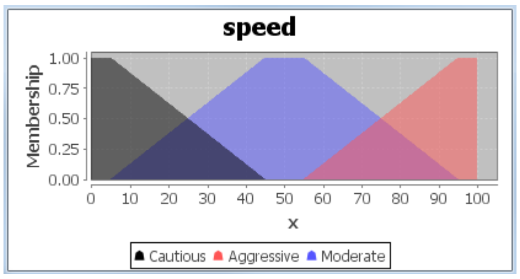


Figure 5: Vehicle Speed Membership Functions

- Low
- Medium Low
- Medium
- Medium High
- High

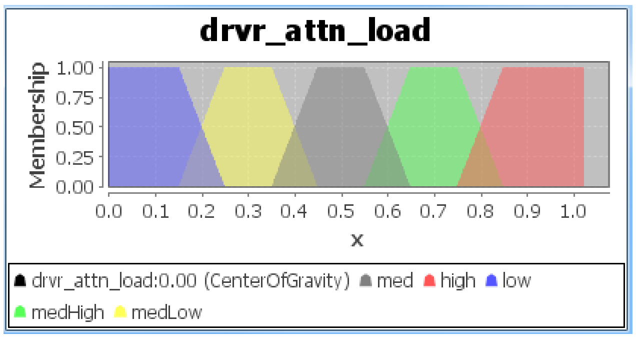


Figure 6: Driver Attention Load Membership Functions

2.3. Define Rule Base

For the design of the system's rule base, a set of fuzzy mapping rules will be created based on knowledge inferred from research on this project. The rule base will be composed of conditional rules that will map the linguistic input variables of the rule antecedents to the output variables of the rule's consequents. Our rules will use the Mamdani form of:

$$\text{IF } x_1 \text{ is } \underset{\sim}{A}_1^k \text{ and } x_2 \text{ is } \underset{\sim}{A}_2^k \text{ THEN } y^k \text{ is } \underset{\sim}{B}^k, \qquad \text{for } k = 1, 2, \ldots, r,$$

Where A represents our fuzzy antecedents and B represents our output or consequent.

From the research the following inferences can be made:

- Driving at high speed requires more concentration to drive the vehicle safely.

- A very high radio volume impacts a driver's ability to concentrate on driving.
- A high frequency of turns increases risk of collisions.
- Driving under dark conditions requires more concentration to drive safely.

Based on the inferences above, we can create a set of rules for our fuzzy system. The number of rules required is calculated by taking into account that there are two inputs with three fuzzy variables and two inputs with two fuzzy variables. From this we can see that there are 36 unique combinations of inputs since: 2 x 3 x 2 x 3 = 36. Table 1 depicts mappings of the linguistic variables for each of the input combinations to the linguistic variable for the outputs.

Table 1. Fuzzy Rule Base.

	Volume Level	Sunlight	Speed	Turn Frequency	DAL
Rule 1	Low	Light	Cautious	Low	Low
Rule 2	Low	Light	Cautious	High	Low
Rule 3	Low	Light	Moderate	Low	Low
Rule 4	Low	Light	Moderate	High	Med_Low
Rule 5	Low	Light	Aggressive	Low	Med_Low
Rule 6	Low	Light	Aggressive	High	Med
Rule 7	Low	Dark	Cautious	Low	Low
Rule 8	Low	Dark	Cautious	High	Med_Low
Rule 9	Low	Dark	Moderate	Low	Med_Low
Rule 10	Low	Dark	Moderate	High	Med
Rule 11	Low	Dark	Aggressive	Low	Med
Rule 12	Low	Dark	Aggressive	High	Med_High
Rule 13	Med	Light	Cautious	Low	Low
Rule 14	Med	Light	Cautious	High	Med_Low
Rule 15	Med	Light	Moderate	Low	Med_Low
Rule 16	Med	Light	Moderate	High	Med
Rule 17	Med	Light	Aggressive	Low	Med
Rule 18	Med	Light	Aggressive	High	Med_High
Rule 19	Med	Dark	Cautious	Low	Med_Low
Rule 20	Med	Dark	Cautious	High	Med
Rule 21	Med	Dark	Moderate	Low	Med
Rule 22	Med	Dark	Moderate	High	Med_High
Rule 23	Med	Dark	Aggressive	Low	Med_High
Rule 24	Med	Dark	Aggressive	High	High
Rule 25	High	Light	Cautious	Low	Med_Low
Rule 26	High	Light	Cautious	High	Med
Rule 27	High	Light	Moderate	Low	Med
Rule 28	High	Light	Moderate	High	Med_High
Rule 29	High	Light	Aggressive	Low	Med_High
Rule 30	High	Light	Aggressive	High	High
Rule 31	High	Dark	Cautious	Low	Med
Rule 32	High	Dark	Cautious	High	Med_High
Rule 33	High	Dark	Moderate	Low	Med_High
Rule 34	High	Dark	Moderate	High	High
Rule 35	High	Dark	Aggressive	Low	High
Rule 36	High	Dark	Aggressive	High	High

2.4. METHOD OF INFERENCE

The method of inference for a fuzzy system define the process for combining the inputs of our rule-based system to generate our fuzzy outputs. There are two major inference techniques used in industry today: Mamdani and TSK. Mamdani is the most widely used inference method in use today but is very heavy computationally. The TSK method is less complex but the output of the model is a linear function of the inputs. Mamdani consists of using If-Then rules to map linguistic input variables to output variables and applies a weighting to each rule. For the purposes of this proposed design, Mamdani is chosen and a weighting of 1 will be used for each rule.

2.5. DEFUZZIFICATION METHOD

The process of converting the fuzzy output variable into a crisp variable is known as the defuzzification method. There are several methods that are widely used throughout the industry today such as the Mean of Maximum method, Centroid method, or the Height method. While computationally intensive, the Centroid method is chosen for the proposed design due to proliferation in the industry and ease of use. The equation for the Centroid method is given below:

$$z^* = \frac{\int \mu_{\underset{\sim}{C}}(z) \cdot z\, dz}{\int \mu_{\underset{\sim}{C}}(z)\, dz},$$

Where C is the fuzzy output value, z is the value along the x-axis of the output membership function, and z* is the final, crisp output.

2.6. PROOF OF DESIGN

To prove the design, a sample scenario will be ran through to determine what the crisp end output would be. Consider a driver traveling at the following parameters:

- Vehicle Speed: 50
- Frequency of Turns: 3
- Volume Step: 31
- Sunload: 5000

This combination of discrete inputs relates to the four following rules from our subsystem:

```
RULE 15: IF volume IS med AND sunload IS light AND
speed IS moderate AND turn_freq IS low THEN
drvr_attn_load IS medLow;
RULE 16 : IF volume IS med AND sunload IS light AND
speed IS moderate AND turn_freq IS high THEN
drvr_attn_load IS med;
RULE 21: IF volume IS med AND sunload IS dark AND
speed IS moderate AND turn_freq IS low THEN
drvr_attn_load IS med;
RULE 22 : IF volume IS med AND sunload IS dark AND
speed IS moderate AND turn_freq IS high THEN
drvr_attn_load IS medHigh;
```

From these four rules we can use the Mamdani implication and aggregation equation to create our fuzzy output:

$$\mu_{\underset{\sim}{B}^k}(y) = \max_k[\min[\mu_{\underset{\sim}{A}_1^k}(\text{input}(i)), \mu_{\underset{\sim}{A}_2^k}(\text{input}(j))]], \quad k = 1, 2, \ldots, r.$$

This leads to the output membership function shown in figure 7.

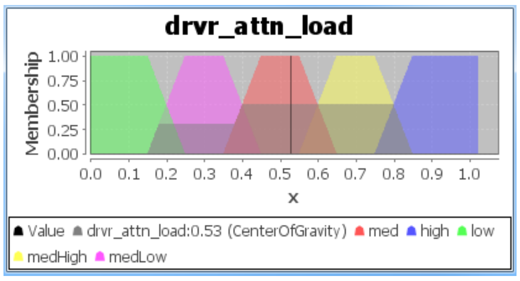
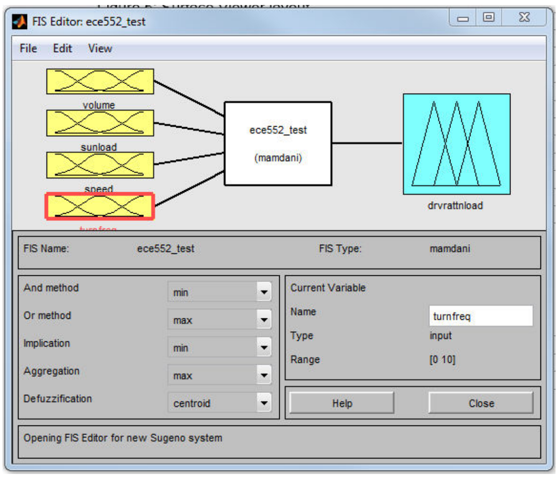


Figure 7: Output Membership Function Based on Crisp Inputs

With this fuzzy output membership function the Centroid defuzzification method can be employed using the equation stated above:



FIS Editor: ece552_test

FIS Name: ece552_test FIS Type: mamdani

And method: min
Or method: max
Implication: min
Aggregation: max
Defuzzification: centroid

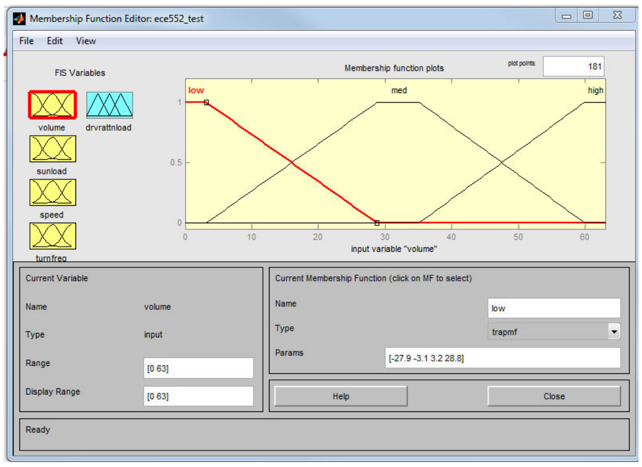
Current Variable
Name: turnfreq
Type: input
Range: [0 10]

Opening FIS Editor for new Sugeno system

Figure 8: MATLAB FIS Editor

$$\frac{\int_{0.15}^{0.175}(12x-1.8)x\,dx+\int_{0.175}^{0.3}0.3x\,dx+\int_{0.3}^{0.5}(8x-2.7)x\,dx+\int_{0.4}^{0.8}0.5x\,dx+\int_{0.8}^{0.85}(-10x+8.5)x}{\int_{0.15}^{0.175}(12x-1.8)\,dx+\int_{0.175}^{0.3}0.3\,dx+\int_{0.3}^{0.5}(8x-2.7)\,dx+\int_{0.4}^{0.8}0.5\,dx+\int_{0.8}^{0.85}(-10x+8.5)\,dx}$$

The output value from the Centroid defuzzification equation provides a Driver Attention Load value of 0.53. This value will be saved for comparison to software



Membership Function Editor: ece552_test

plot points: 181

Current Variable
Name: volume
Type: input
Range: [0 63]
Display Range: [0 63]

Current Membership Function (click on MF to select)
Name: low
Type: trapmf
Params: [-27.9 -3.1 3.2 28.8]

Ready

Figure 9: Membership Function Editor

simulations of the system to ensure design accuracy.

3. MATLAB Simulation

Once the design parameters of the system are defined then modeling/simulation can begin. Be

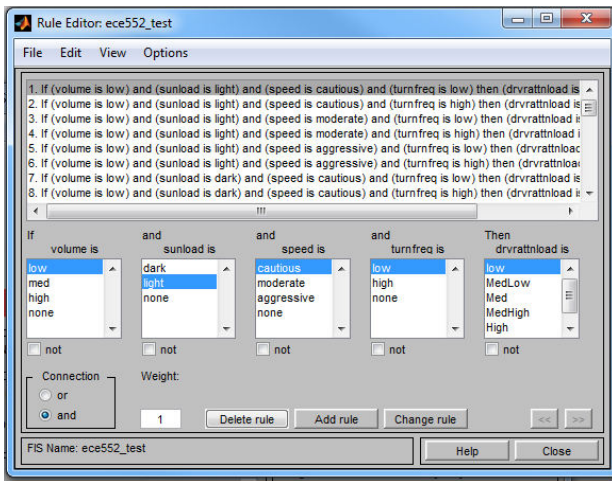
fore moving to the code writing stage of the project MATLAB will be used to validate the design. The Fuzzy Logic Toolbox within the software provides a simple point and click GUI for development of Fuzzy Inference Systems (FIS). Using the editor, a new Mamdani method FIS is created. The system is given four inputs: volume, speed, turn frequency, and sunload. The FIS is given a single output: driver attention load. The configuration for the FIS is based on our earlier choices of using Mamdani inference method, max-min implication and aggregation, and the Centroid defuzzification method as shown in figure 8.

Once the FIS initial settings are created, the Membership Function Editor is used to create the crisp to fuzzy relations for the system as shown in figure 9.
Once the membership functions are completed, the Rule Editor is used in accordance with the table of rules created in the previous section to input all 36 fuzzy rules of the system as shown in figure 10.

Once all 36 rules are input into the Rule Editor then validation on the design can begin using the Rule Viewer.

The Rule Viewer allows for quick and seamless validation of the FIS by allowing a user to adjust the four input values and receive a crisp output, the output fuzzy membership functions, and the rules that are currently active based on crisp input values. Each of the four leftmost vertical columns are the inputs of the FIS and the rightmost column is the output as shown in figure 11. The vertical red bars can be dragged over the range of the particular input to test the FIS's output. The Rule Viewer also gives a view of the output membership function, based on your implication and aggregation settings, at the bottom of the right hand column.

Now that the FIS is completed in MATLAB, validation can begin. Each of the input values from the rule table in the previous section is tested for all 36 rules to determine if the corresponding crisp output agrees with the



Rule Editor: ece552_test

1. If (volume is low) and (sunload is light) and (speed is cautious) and (turnfreq is low) then (drvrattnload is
2. If (volume is low) and (sunload is light) and (speed is cautious) and (turnfreq is high) then (drvrattnload is
3. If (volume is low) and (sunload is light) and (speed is moderate) and (turnfreq is low) then (drvrattnload is
4. If (volume is low) and (sunload is light) and (speed is moderate) and (turnfreq is high) then (drvrattnload i
5. If (volume is low) and (sunload is light) and (speed is aggressive) and (turnfreq is low) then (drvrattnloac
6. If (volume is low) and (sunload is light) and (speed is aggressive) and (turnfreq is high) then (drvrattnloa
7. If (volume is low) and (sunload is dark) and (speed is cautious) and (turnfreq is low) then (drvrattnload is
8. If (volume is low) and (sunload is dark) and (speed is cautious) and (turnfreq is high) then (drvrattnload i

If volume is	and sunload is	and speed is	and turnfreq is	Then drvrattnload is
low	dark	cautious	low	low
med	light	moderate	high	MedLow
high	none	aggressive	none	Med
none		none		MedHigh
				High

Connection: and

Weight: 1

FIS Name: ece552_test

Figure 10: Rule Editor in MATLAB

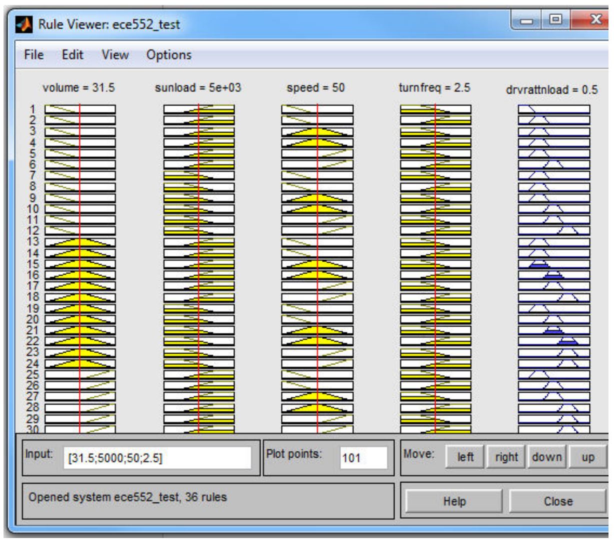


Figure 11: MATLAB Rule Viewer

fuzzy linguistic output variable stated in the rules. For this FIS, all validation proves that the design and the implementation of the system have been done correctly. Also, from the proof of design section previously, the four crisp input values are testing to validate if the equations were implemented properly.

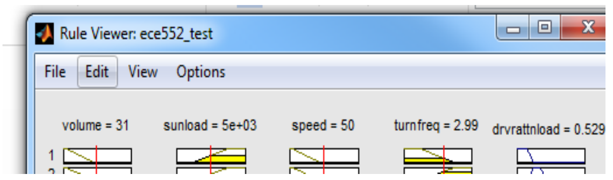


Figure 12: Output of Rule Viewer with Testing Inputs from Proof of Design Equations

The output Driver Attention Load value is approximately the same as the testing value of 0.53 from the previous section as shown in figure 12. This proves that the design and simulation have been implemented properly.

4. JAVA APPLICATION DESIGN

Once the system is modeled and validated in MATLAB, the application can be implemented in a more common programming language. The Java programming language was chosen for several reasons: efficiency, security, wide usage in industry, and portability. The last attribute is one of the biggest reasons for choosing Java, it can easily be ported to another programming language for use in other applications. For automotive applications, this is an attractive quality since many embedded ECUs use a more common programming language like C/C++. Since Java and C/C++ are very similar, one can be easily converted to the other and thus the code written for this project could theoretically be implemented in an automotive application with a fairly low development cost. The standard Java library, Java Standard Edition 8, offers no native support for programming of Fuzzy Logic applications but an open source Java library known as jFuzzyLogic is available for download. Aside from the addition of the Fuzzy Logic library, the remainder of the code uses the standard libraries in JavaSE 8 such as the Abstract Window Toolkit (AWT) and Swing Toolkits for GUI design. [Java references from wikipedia]

4.1.JFUZZYLOGIC

This library was created by two graduate students, Pablo Cingolani and Jesus Alcala-Fdez, to create a standardized development environment in Java for fuzzy logic applications. jFuzzyLogic defines Fuzzy Logic Controllers (FCL) using a control language based on the standard defined in the IEC 61131 part 7 specification[2][3]. This specification attempts to standardize a control language for the programming of programmable controllers implementing FCLs. jFuzzyLogic builds the FCL based on the fuzzy control parameters define in the “.fcl” files. “.fcl” files are written by the programmer and use a C programming language syntax to allow the user to define fuzzy inputs, fuzzy outputs, membership functions, and fuzzy rule bases. The programming of these parameters is very similar to the process used to define the FIS parameters in MATLAB. Using the “.fcl” syntax, the inputs, outputs, and membership functions from the file are imported below:

```
FUZZIFY volume    // Fuzzify input variable 'volume':
{'low', 'med' , 'high'}
        TERM low := (0, 1) (3.2, 1) (28.8,0) ;
        TERM med := (3.2, 0) (28.8,1) (35.1,1) (59.9,0);
        TERM high := (35.1, 0) (59.9, 1) (63.2, 1);
END_FUZZIFY

FUZZIFY sunload    // Fuzzify input variable 'sunload': {
'dark', 'light' }
        TERM dark := (0, 1) (2500, 1) (7500,0) ;
        TERM light := (2500,0) (7500,1) (10000, 1);
END_FUZZIFY

FUZZIFY speed     // Fuzzify input variable 'speed': {
'cautious', 'moderate', 'aggressive' }
        TERM cautious := (0, 1) (5, 1) (45,0) ;
        TERM moderate := (5,0) (45,1) (55, 1) (95 , 0);
        TERM aggressive := (55, 0) (95,1) (100,1);
END_FUZZIFY

FUZZIFY turn_freq  // Fuzzify input variable 'turn_freq': {
'low', 'high' }
        TERM low := (0, 1) (1.25, 1) (3.75,0) ;
        TERM high := (1.25, 0) (3.75,1) (5,1);
END_FUZZIFY

DEFUZZIFY drvr_attn_load     // Defuzzify output variable
'drvr_attn_load' : {'low', 'medLow', 'med', 'medHigh',
'high' }
        TERM low := (0,1) (0.15,1) (0.25,0);
        TERM medLow := (0.15,0) (0.25,1) (0.35,1) (0.45,0);
        TERM med := (0.35,0) (0.45,1) (0.55,1)(0.65,0);
        TERM medHigh := (0.55,0) (0.65,1) (0.75,1)(0.85,0);
        TERM high := (0.75,0) (0.85,1) (1,1);
        METHOD : COG;// Use 'Center Of Gravity'
defuzzification method
        DEFAULT := 0;// Default value is 0 (if no rule
activates defuzzifier)
END_DEFUZZIFY
```

FCL rules are defined within the “.fcl” files using structures called “Ruleblocks”. The following section displays a sample structure of the first four rules of this project:

```
RULEBLOCK No1
        AND : MIN;         // Use 'min' for 'and' (also
implicit use 'max' for 'or' to fulfill DeMorgan's Law)
        ACT : MIN;         // Use 'min' activation method
        ACCU : MAX;        // Use 'max' accumulation method

        RULE 1 : IF volume IS low AND sunload IS light AND
speed IS cautious AND turn_freq IS low THEN drvr_attn_load
IS low;
        RULE 2 : IF volume IS low AND sunload IS light AND
speed IS cautious AND turn_freq IS high THEN drvr_attn_load
IS medLow;
        RULE 3 : IF volume IS low AND sunload IS light AND
speed IS moderate AND turn_freq IS low THEN drvr_attn_load
IS medLow;
        RULE 4 : IF volume IS low AND sunload IS light AND
speed IS moderate AND turn_freq IS high THEN drvr_attn_load
IS medLow;

END_RULEBLOCK
```

Once the ".fcl" file is completed, the Java code can be written to access the FCL that has been defined. Using the APIs defined by jFuzzyLogic, the FCL is accessed by first creating a Fuzzy Inference System object and then loading the newly created ".fcl" file into it. From this point, fuzzy variables and membership functions can be accessed. Sample code for an FCL is displayed below:

```
String filename = "dal.fcl"; //Load filename into
string object

FIS fis = FIS.load(filename, true); //Load ".fcl"
file into newly created FIS object

FunctionBlock fb = fis.getFunctionBlock(null);
//Create FunctionBlock object from newly created FIS
object

fb.setVariable("volume", volume); //Access variables
of FunctionBlock as defined in ".fcl" file
fb.setVariable("sunload", sunLoad); //Access
variables of FunctionBlock as defined in ".fcl" file
fb.setVariable("turn_freq", turnFreq); //Access
variables of FunctionBlock as defined in ".fcl" file
fb.setVariable("speed", speed); //Access variables of
FunctionBlock as defined in ".fcl" file

fb.evaluate(); //Evaluate Fuzzy Control System using
currently set parameters
```

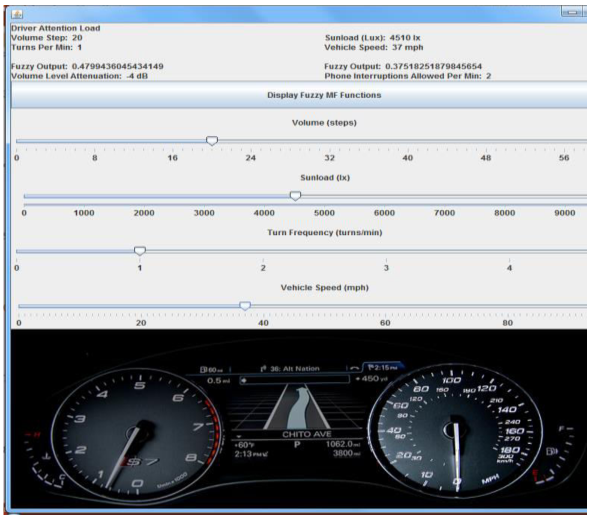

Driver Attention Load
Volume Step: 20
Turns Per Min: 1
Sunload (Lux): 4510 lx
Vehicle Speed: 37 mph
Fuzzy Output: 0.4799436045434149
Volume Level Attenuation: -4 dB
Fuzzy Output: 0.37518251879845654
Phone Interruptions Allowed Per Min: 2

Figure 13: Image of the Project UI Container

```
fb.getVariable("drvr_attn_load").defuzzify(); //Get
defuzzified fuzzy output variable
```

Once the FCL has been designed, the GUI code can be written using the JavaSE 8 standard libraries.[2][3]

4.2. Graphical User Interface Code

To design the GUI for this project, standard elements of the Java libraries were used such as the Abstract Window Toolkit and Swing Toolkits. Four major elements of the Java UI library are used for this project:

- JLabels – An input area for images or text that is not editable.
- JPanels – A container for UI elements that allows flexibility for items such as different types of text or images.
- JSlider – A sliding menu input that allows a user to select values between two bounded inputs.
- ImageIcon – A UI element that allows images to be placed and positioned easily.

The UI window is designed to as a 850px x 850px container as shown in figure 13. Within the container, there are three JPanel objects used to divide the panel into three areas:

- textPanel
- bottomPanel
- iconPanel

The textPanel object is a JPanel that is designed to present the user with all the current information on the model in a textual format. The textPanel acts as a container to hold a total of nine JLabel objects as shown in figure 14. JLabel objects are added to a Java project using the following syntax example:

```
JLabel myText = new JLabel("Driver Attention Load
Monitoring System");
```

The JLabels are divided into two sections: one section for the input information status and one section for the output information. As the user adjusts the inputs using the slider bar UI elements, the input values listed on top update to display the current crisp values. As the input values are

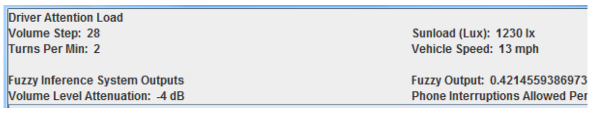
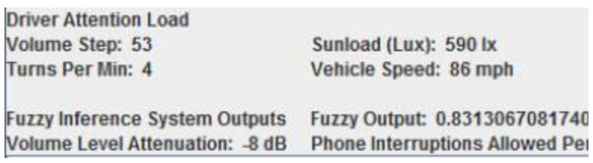

Driver Attention Load
Volume Step: 28
Turns Per Min: 2
Sunload (Lux): 1230 lx
Vehicle Speed: 13 mph
Fuzzy Inference System Outputs
Volume Level Attenuation: -4 dB
Fuzzy Output: 0.4214559386973
Phone Interruptions Allowed Per

Figure 14: TextPanel JPanel area

adjusted, the output values of "Fuzzy Output", "Volume Level Attenuation", and "Phone Interruptions Allowed" update with feedback from the Fuzzy Inference System. This panel is the primary interface used to provide information to the model user.

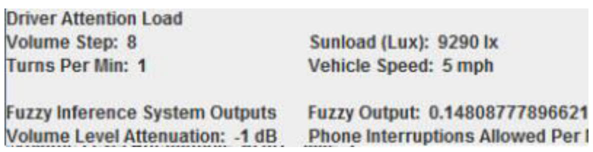
Due to adjustments to the input sliders, the Driver Attention Load Fuzzy Output value increases and the values of the Volume Level Attenuation and the Phone Interruptions Allowed will slowly increase to adjust the vehicle environment and ensure the driver can maintain concentration. As an example, the textPanel output below shows a dangerous driving environment with high speed, high volume, low sunload, and high turns per minute. Based on these inputs, the Driver Attention Load value is at 0.83 or 83%, this correlates to a reduction in the amplifier volume level by -8dB and a limit in the amount of connected phone interruptions to 1 per minute as shown in figure 15.


```
Driver Attention Load
Volume Step: 53                       Sunload (Lux): 590 lx
Turns Per Min: 4                      Vehicle Speed: 86 mph

Fuzzy Inference System Outputs        Fuzzy Output: 0.8313067081740
Volume Level Attenuation: -8 dB       Phone Interruptions Allowed Pe
```

Figure 15: Fuzzy Output Values from Example

For another example, the textPanel output below shows a cautious driving environment with low speed, low volume, high sunload, and low turns per minute. Based on these inputs, the Driver Attention Load value is at 0.148 or about 15%, this correlates to a reduction in the amplifier volume level by only -1dB and an increase in the limit in the amount of connected phone interruptions to 3 per minute


```
Driver Attention Load
Volume Step: 8                        Sunload (Lux): 9290 lx
Turns Per Min: 1                      Vehicle Speed: 5 mph

Fuzzy Inference System Outputs        Fuzzy Output: 0.14808777896621
Volume Level Attenuation: -1 dB       Phone Interruptions Allowed Per I
```

Figure 16: Fuzzy Output Values from Second Example

as shown in figure 16.

The bottomPanel is another JPanel container that is used to hold four JSlider objects that act as the user inputs for the model. The four sliders are for Volume, Sunload, Vehicle Speed, and Turn Frequency as shown in figure 17.

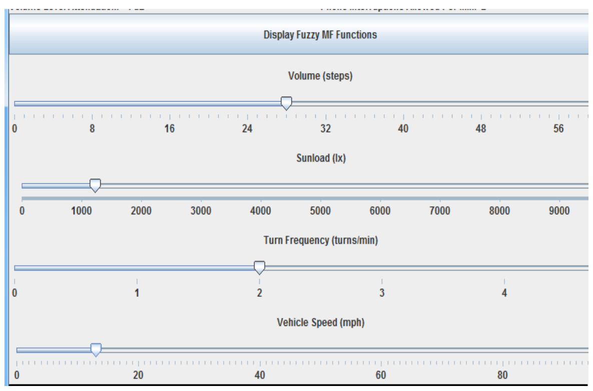


Figure17: BottomPanel Displaying the Sliders and the Button Object

JSlider objects are constructed using the following sample syntax:

```
JSlider volumeSlider = new
JSlider(JSlider.HORIZONTAL,volume_MIN, volume_MAX,
volume_INIT);
```

Also the panel holds a button labeled as “Display Fuzzy MF Functions” that, when pressed, will use the current input value settings and then evaluate the state of the fuzzy system and provide the fuzzy output value as well as plot

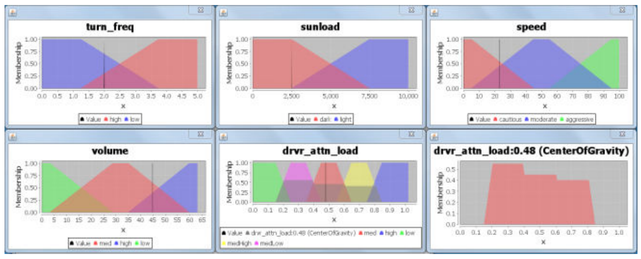


Figure18: Output Membership functions that are Displayed When n Button is Selected

charts of membership functions for the inputs and outputs. When the “Display Fuzzy MF Functions” button is selected, the jFuzzyLogic library plots the membership functions as displayed in figure 18.

The iconPanel is the container used to hold the ImageIcon object. The ImagIcon is used to display a sample Instrument Cluster image to the user. As the Driver Attention Load fuzzy output value increases, the instrument cluster display will slow begin to dim all non-critical information until only the speedometer is visible to

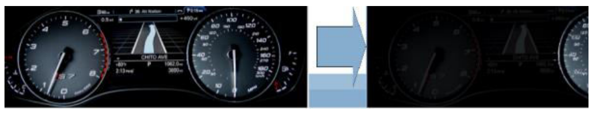


Figure19: Simulated Instrument Cluster Display

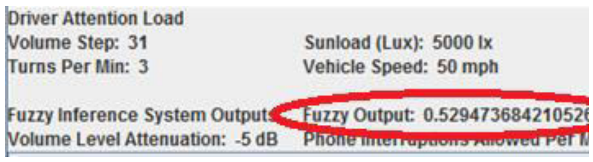
ensure that driver attention is only given to critical information content. As an example, the two instrument clusters shown in figure 19 present different Driver Attention Loads. On the left is the dimming level used with a cautious driving environment and a Driver Attention Load of 15%. On the right is a more dangerous driver with a Driver Attention Load of 83%.

4.3.JAVA APPLICATION VALIDATION

Once the Java simulation of the Driver Attention Load Fuzzy System is completed, the software can be tested against the previously created MATLAB module using the testing inputs that were determined in the design phase:

- Vehicle Speed: 50 mph.
- Frequency of Turns: 3 turns per minute.
- Volume Step: 31 steps.
- Sunload: 5000 lx.

The Java simulation output value of 0.53 for Driver Attention Load matches the value determined during the design phase and the value that was generated from the MATLAB FIS model as shown in figure 20.



```
Driver Attention Load
Volume Step: 31                      Sunload (Lux): 5000 lx
Turns Per Min: 3                     Vehicle Speed: 50 mph

Fuzzy Inference System Output        Fuzzy Output: 0.52947368421052
Volume Level Attenuation: -5 dB      Phone Interruptions Allowed Per M
```

Figure 20: Validation Results from Java Simulation

5. Comparison of Classical to Fuzzy System

Through the validation of the Fuzzy Inference System created in this project, it is determined that a Fuzzy Logic solution is more desirable than a Classical Logic solution. This is because the Fuzzy Logic solution provides a smooth and consistent transition between fuzzy subspaces. A Fuzzy Logic system is able to provide this smooth transition because of the standard requirement of systems of this type: overlapping fuzzy subspaces. Overlapping areas in the membership functions, which is critical to a proper Fuzzy System, allow for a higher degree of granularity in your fuzzy output values which leads to smoother transitions. If this function was implemented using a Classical Logical system there transitions between different levels of Driver Attention Load values would be too drastic. This would be undesirable in an automobile environment where smooth transitions are valued. If the transitions were rigid, then jumps between instrument cluster brightness and radio volume would be very choppy and would project an unrefined feel.

6. Conclusion

The Fuzzy Inference System designed and modeled for this project leads to two major conclusions. The first conclusion is that a driver monitoring system can be designed completely in software with relatively low development costs. This system would require no hardware costs and would incorporate information that is already provided in a vehicle environment. Using the Open Source library jFuzzyLogic provides a specialized software function with no further costs as well. With these low software development and hardware costs, the system provides an extremely low cost solution for driver monitoring systems. The secondary conclusion is that the Fuzzy Inference System provides a better system model that if a classical model would have been selected since this system provides smoother transition between fuzzy subspaces which equates to a more desirable customer experience.

References

[1]. Chen, Chou-Lin; Liu, Cejun; " An Analysis of Speeding-Related Crashes: 6. Performing Organization Code Definitions and the Effects of Road Environments,"; Internet: http://www-nrd.nhtsa.dot.gov/Pubs/811090.PDF

[2]. Cingolani, Pablo, and Jesus Alcala-Fdez. "jFuzzyLogic: a robust and flexible Fuzzy-Logic inference system language implementation." Fuzzy Systems (FUZZ-IEEE), 2012 IEEE International Conference on. IEEE, 2012.

[3]. Cingolani, Pablo, and Jesús Alcalá-Fdez. "jFuzzyLogic: a Java Library to Design Fuzzy Logic Controllers According to the Standard for Fuzzy Control Programming"

[4]. "Driving At Night," Internet: http://www.nsc.org/news_resources/Resources/Documents/Driving_at_Night.pdf

[5]. "Driver Monitoring System," Internet: http://en.wikipedia.org/wiki/Driver_Monitoring_System

[6]. "Facts and Consequences of Distracted Driving," Internet: http://www.learnstuff.com/facts-and-consequences-of-distracted-driving/

[7]. Lee, John D.; "Driving Attention: Cognitive Engineering in Designing Attractions and Distractions," Frontiers of Engineering; Volume 34; Number 4; Winter 2008.

[8]. "Lux,"; Internet: http://en.wikipedia.org/wiki/Lux

[9]. "National Motor Vehicle Crash Causation Survey Report to Congress," Internet: http://www-nrd.nhtsa.dot.gov/Pubs/811059.PDF

[10]. Strick, Susan; "Music Effects on Drivers' Reaction Times,"; Internet: http://www.drdriving.org/misc/music_strick_report.html

[11]. "What is Distracted Driving?" Internet: http://www.distraction.gov/content/get-the-facts/facts-and-statistics.html

[12]. Adnan Shaout and Raj Tonshal, "Real Time Driver Activity Index Detection Using Fuzzy Logic", The International Journal Of Advanced Research In Electrical, Electronics And Instrumentation Engineering (IJAREEIE) (Volume 3, Issue 9, September, 2014. Impact Factor Is 1.686.

Dr. Adnan Shaout is a full professor in the Electrical and Computer Engineering Department at the University of Michigan –Dearborn. At present, he teaches courses in fuzzy logic and engineering applications and computer engineering (hardware and software). His current research is in applications of fuzzy set theory, embedded systems, software engineering, artificial intelligence and cloud computing. Dr. Shaout has more than 34 years of experience in teaching and conducting research in the electrical and computer engineering fields at Syracuse University and the University of Michigan -Dearborn. Dr. Shaout has published over 170 papers in topics related to electrical and computer engineering fields. Dr. Shaout has obtained his B.Sc., M .S. and Ph.D. in Computer Engineering from Syracuse University, Syracuse, NY, USA, in 1982, 1983, 1987, respectively.

Dominic Colella is a graduate student in the College of Engineering and Computer Science at the University of Michigan –Dearborn.