FINAL PROJECT REPORT

CSE 2009-SOFT COMPUTING

SLOT – H+TH

TOPIC-ANTILOCK BREAKING SYSTEM USING FUZZY LOGIC

TEAM

1) B PRATYUSH (19BCN7114)

2) KVPS GANESH (19BCN7073)

GUIDED BY PROF.SHRUTI MISHRA

TABLE OF CONTENTS

PROBLEM STATEMENT

To implement the Antilock braking system using Fuzzy logic technology.

We aim to reduce the accidents caused by implementing the ABS system by considering certain approximations and rules where we increase the efficiency of decision making and responding in critical situations.

It can be observed many vehicles today are provided with Air bags which prevents the main parts of a person like brain from critical damage. The air bags reach to the output of the inference engine. They open when sudden break is applied to the vehicle and depending break impact the air bags open. So for determining the break pressure required to stop the accelerating vehicle we use ABS technology.

It determines the braking pressure to be applied on the break by giving the wheel slip ratio and wheel acceleration as input to the inference engine.

This system makes use of Fuzzy logic technology by which it makes the correct judgements and increases the judgement accuracy and response time of a vehicle by making some approximations and considering various factors devised. Using the fuzzy logic we will provide a reliable judgement making capability to the vehicle during these critical situations.

MOTIVATION

The prime aspect of Antilock Braking System is to improve the driver's ability to control the direction of vehicle under tough road conditions. It decreases the accident rate under hard conditions. The system tries to understand the underlying conditions which is responsible for wheel braking on pressure application.

With Fuzzy ABS, steering ability is not only retained during wheel braking maneuver, slowing down of vehicle is also observed. Vehicle sensors are configured accordingly which react to the workflow of the fuzzy logic system and act accordingly to react to the particular situation.

The reason for the development of antilock brakes is in essence very simple. Under braking, if one or more of a vehicle's wheels lock (begins to skid) then this has number of consequences:

a) Braking distance increases.

b) Steering control is lost

c) Tire wear will be abnormal.

The obvious consequence is that an accident is far more likely to occur. The application of brakes generates a force that impedes a vehicles motion by applying a force in the opposite direction. During severe braking scenarios, a point is obtained in which the tangential velocity of the tire surface and the velocity road surface are not the same. This term is referred to as slip ratio and is the ratio between the tangential velocity of the tire and the true ground speed of the vehicle and is expressed as: Vg is circumferential velocity of braked wheel and Vt is vehicle road speed. 0% is slip of free rolling wheel and 100% for locked wheel. The braking force or the adhesion coefficient of braking force (µf) measured in the direction the wheel is turning is function of slip. µf depends on a number Vg is circumferential velocity of braked wheel and Vt is vehicle road speed. 0% is slip of free rolling wheel and 100% for locked wheel. The braking force or the adhesion coefficient of braking force (µf) measured in the direction the wheel is turning is function of slip. µf depends on a number of factors, and the main ones are:

a) Road surface material condition

b) Tire material, inflation pressure, tread depth, tread pattern and construction.

SURVEY ANALYSIS

- **Enhanced Antilock Braking System using Fuzzy Logic Road Detector(ISSN Online: 2347-5439), a research journal by Electrical Engineering professors of NITTTR, CHANDIGARH have come up with the road detector API which detects the road conditions based on the adhesion coefficient and slip ratio. They have used fuzzy logic technology to implement the Detector. They proposed this to be added to the existing ABS configuration**

- **Antilock-Braking System Using Fuzzy Logic (ISSN 1990-9233), a journal published by DR K. SUBBULAKSHMI, ECE DEPARTMENT, BHARAT UNIVERSITY deals with the ABS system where the wheel slip ratio and wheel acceleration are taken as input to Inference engine and the Break pressure is calculated. From this study we can observe the difference between a vehicle with ABS and without ABS. The journal consists of the braking behavior and mechanics of the wheel. The algorithm analysis is provided with the simulation outputs observed during the process.**

- **FUZZY LOGIC ANTI-LOCK BRAKE SYSTEM(ISSN 2229-5518), a journal published by Darshan Modi, Zarana Padia and Kartik Patel deals with the ABS system and an adaption of the previous journal framed above. In this they also added the engine behavior to the existing inputs and made an attempt to improve the vehicle performance. Analysis was done using CASE TOOL with the simulated outputs present in the journal.**

METHODOLOGY

FUZZY ABS SYSTEM

The Fuzzy Controller System uses two input values: Wheel slip ratio and Wheel acceleration. To showcase the fuzzy logic application the code takes calculated values of slip ratio and wheel acceleration and does not deal with the calculation part.

The calculations can be done as follows:

$$s_B = \frac{v_{Fzg} - \omega R}{v_{Fzg}} = \frac{v_{Fzg} - v_{Wheel}}{v_{Fzg}} \quad \text{and the wheel acceleration:} \quad a_{Wheel} = \frac{\partial v_{Wheel}}{\partial t} \approx \frac{\Delta v_{Wheel}}{\Delta t},$$

The slip ratio and wheel acceleration is passed into the Fuzzy System as crisp data. The data is changed to linguistic data via fuzzy membership functions. The fuzzified outputs go to the inference where certain fuzzy rules are applied on the data. The output from inference engine is defuzzified back to crisp data using centroid method.

The inference system used in this case is the MAMDANI Fuzzy Inference Engine.

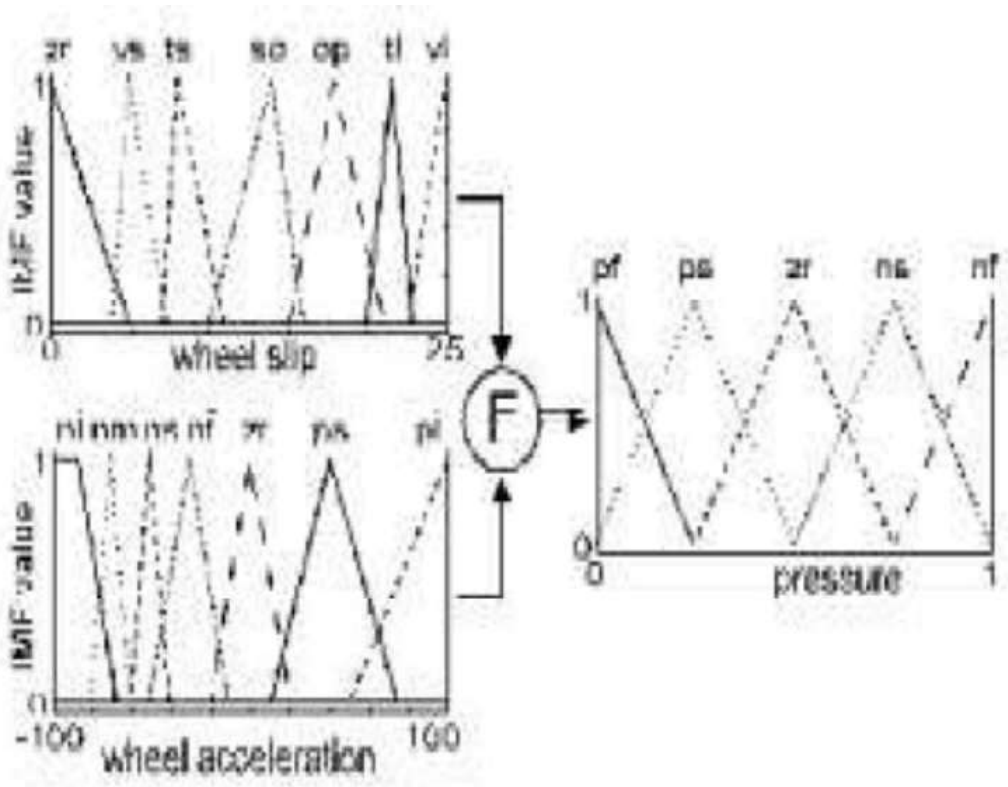
FUZZIFICATION PROCESS:

Wheel slip Range (0,25)

Wheel acceleration Range (-100,100)

Brake Pressure Range (0,1)

We established the membership functions for the variables as given below



We have taken triangle membership for all the variables.

Universe of Discourse

Slip ratio – {zero, very small, too small, smaller than optimum, optimum, too large, very large}

Wheel acceleration – {negative large, negative medium, negative small, negative few, zero, positive small, positive large}

Brake Pressure - {positive fast, positive slow, zero, negative slow, negative fast}

Antecedent – {Slip ratio, Wheel acceleration}

Consequent – {Brake pressure}

Rule Table

Slip ratio

	Zero	VS	TS	STO	OPT	TL	VL
NL	NS	NS	NS	NS	NF	NF	NF
NM	NS	NS	NS	NS	NF	NF	NF
NSM	NS	NS	NS	NS	NS	NS	NS
NF	Zero	Zero	Zero	Zero	Zero	Zero	Zero
Zero	PS	PS	PS	PS	PS	PS	PS
PS	PF	PF	PF	PS	PS	PS	PS
PL	PF	PF	PF	PS	PS	PS	PS

Total rules – 49

INDEX

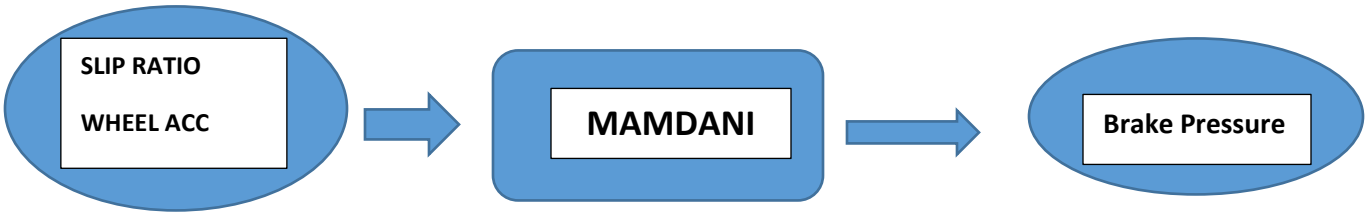
Slip ratio – VS (VERY SMALL), TS (TOO SMALL), STO (SMALLER THAN OPTIMUM), OPT (OPTIMUM), TL (TOO LARGE), VL (VERY LARGE)

Wheel acceleration – NL (NEGATIVE LARGE), NM (NEGATIVE MEDIUM), NSM (NEGATIVE SMALL), NF (NEGATIVE FEW), PS (POSITIVE SMALL), PL (POSITIVE LARGE)

Brake pressure – NS (NEGATIVE SLOW), NF (NEGATIVE FAST), PS (POSITIVE SLOW), PF (POSITIVE FAST)

Defuzzification

Crisp output generated using MAMDANI Inference Engine using the centroid method



Result Analysis

```
import numpy as np
import skfuzzy as fuzz
from skfuzzy import control as ctrl
import matplotlib.pyplot as plt

# Define input and output variables
wheelslip= ctrl.Antecedent(np.arange(0, 26, 1),'wheelslip')
wheelacc = ctrl.Antecedent(np.arange(-100, 120, 1),'wheelacc')
pressure= ctrl.Consequent(np.arange(0, 1.25, 0.25),'pressure')

# Wheelslip membership function
wheelslip['zero'] = fuzz.trapmf(wheelslip.universe, [0, 0, 3, 6])
wheelslip['very small'] = fuzz.trimf(wheelslip.universe, [4, 5, 7.5])
wheelslip['too small'] = fuzz.trimf(wheelslip.universe, [7, 8, 11])
wheelslip['smaller than optimum'] = fuzz.trimf(wheelslip.universe, [10, 13, 16])
wheelslip['optimum'] = fuzz.trimf(wheelslip.universe, [15, 18, 21])
wheelslip['too large'] = fuzz.trimf(wheelslip.universe, [20, 21, 23])
wheelslip['very large'] = fuzz.trimf(wheelslip.universe, [23, 25, 25])
wheelslip.view()

# Wheel acceleration membership function
wheelacc['negative large'] = fuzz.trimf(wheelacc.universe, [-100, -80, -70])
wheelacc['negative medium'] = fuzz.trimf(wheelacc.universe, [-80, -70, -60])
wheelacc['negative small'] = fuzz.trimf(wheelacc.universe, [-60, -50, -40])
wheelacc['negative few'] = fuzz.trimf(wheelacc.universe, [-50, -30, -15])
wheelacc['zero'] = fuzz.trimf(wheelacc.universe, [-20, 0, 20])
wheelacc['positive small'] = fuzz.trimf(wheelacc.universe, [10, 40, 75])
wheelacc['positive large'] = fuzz.trimf(wheelacc.universe, [50, 100, 100])
wheelacc.view()

# Pressure membership function
pressure['positive fast'] = fuzz.trimf(pressure.universe, [0, 0, 0.25])
pressure['positive slow'] = fuzz.trimf(pressure.universe, [0, 0.25, 0.50])
pressure['zero'] = fuzz.trimf(pressure.universe, [0.25, 0.50, 0.75])
pressure['negative slow'] = fuzz.trimf(pressure.universe,[0.50, 0.75, 1])
pressure['negative fast'] = fuzz.trimf(pressure.universe,[0.75, 1, 1])
pressure.view()

# Inference Engine
rule1 = ctrl.Rule(antecedent=(wheelslip['zero'] & wheelacc['negative
large']),consequent=pressure['negative slow'])
rule2 = ctrl.Rule(antecedent=(wheelslip['zero'] & wheelacc['negative
medium']),consequent=pressure['negative slow'])
```

```
rule3 = ctrl.Rule(antecedent=(wheelslip['zero'] & wheelacc['negative
small']),consequent=pressure['negative slow'])
rule4 = ctrl.Rule(antecedent=(wheelslip['zero'] & wheelacc['negative
few']),consequent=pressure['zero'])
rule5 = ctrl.Rule(antecedent=(wheelslip['zero'] & wheelacc['zero']),consequent=pressure['positive
slow'])
rule6 = ctrl.Rule(antecedent=(wheelslip['zero'] & wheelacc['positive
small']),consequent=pressure['positive fast'])
rule7 = ctrl.Rule(antecedent=(wheelslip['zero'] & wheelacc['positive
large']),consequent=pressure['positive fast'])
rule8 = ctrl.Rule(antecedent=(wheelslip['very small'] & wheelacc['negative
large']),consequent=pressure['negative slow'])
rule9 = ctrl.Rule(antecedent=(wheelslip['very small'] & wheelacc['negative
medium']),consequent=pressure['negative slow'])
rule10 = ctrl.Rule(antecedent=(wheelslip['very small'] & wheelacc['negative
small']),consequent=pressure['negative slow'])
rule11 = ctrl.Rule(antecedent=(wheelslip['very small'] & wheelacc['negative
few']),consequent=pressure['zero'])
rule12 = ctrl.Rule(antecedent=(wheelslip['very small'] &
wheelacc['zero']),consequent=pressure['positive slow'])
rule13 = ctrl.Rule(antecedent=(wheelslip['very small'] & wheelacc['positive
small']),consequent=pressure['positive fast'])
rule14 = ctrl.Rule(antecedent=(wheelslip['very small'] & wheelacc['positive
large']),consequent=pressure['positive fast'])
rule15 = ctrl.Rule(antecedent=(wheelslip['too small'] & wheelacc['negative
large']),consequent=pressure['negative slow'])
rule16 = ctrl.Rule(antecedent=(wheelslip['too small'] & wheelacc['negative
medium']),consequent=pressure['negative slow'])
rule17 = ctrl.Rule(antecedent=(wheelslip['too small'] & wheelacc['negative
small']),consequent=pressure['negative slow'])
rule18 = ctrl.Rule(antecedent=(wheelslip['too small'] & wheelacc['negative
few']),consequent=pressure['zero'])
rule19 = ctrl.Rule(antecedent=(wheelslip['too small'] &
wheelacc['zero']),consequent=pressure['positive slow'])
rule20 = ctrl.Rule(antecedent=(wheelslip['too small'] & wheelacc['positive
small']),consequent=pressure['positive fast'])
rule21 = ctrl.Rule(antecedent=(wheelslip['too small'] & wheelacc['positive
large']),consequent=pressure['positive fast'])
rule22 = ctrl.Rule(antecedent=(wheelslip['smaller than optimum'] & wheelacc['negative large']),
                   consequent=pressure['negative slow'])
rule23 = ctrl.Rule(antecedent=(wheelslip['smaller than optimum'] & wheelacc['negative medium']),
                   consequent=pressure['negative slow'])
rule24 = ctrl.Rule(antecedent=(wheelslip['smaller than optimum'] & wheelacc['negative small']),
                   consequent=pressure['negative slow'])
rule25 = ctrl.Rule(antecedent=(wheelslip['smaller than optimum'] & wheelacc['negative few']),
                   consequent=pressure['zero'])
```

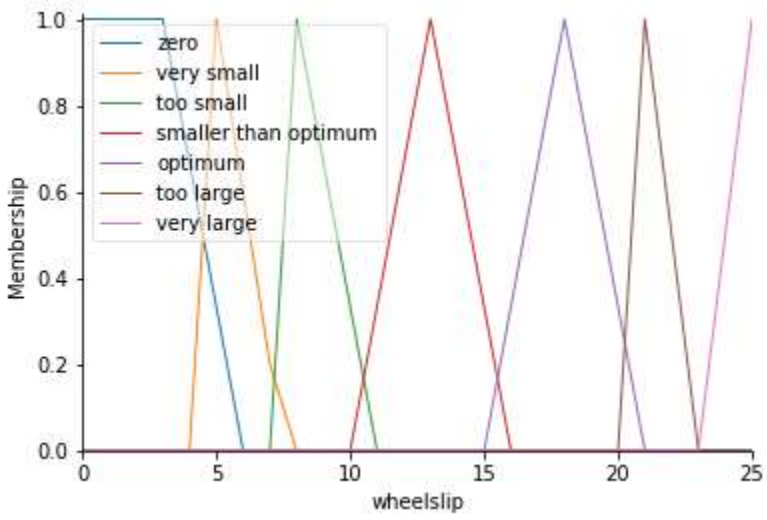
```
rule26 = ctrl.Rule(antecedent=(wheelslip['smaller than optimum'] & wheelacc['zero']),
                   consequent=pressure['positive slow'])
rule27 = ctrl.Rule(antecedent=(wheelslip['smaller than optimum'] & wheelacc['positive small']),
                   consequent=pressure['positive slow'])
rule28 = ctrl.Rule(antecedent=(wheelslip['smaller than optimum'] & wheelacc['positive large']),
                   consequent=pressure['positive slow'])
rule29 = ctrl.Rule(antecedent=(wheelslip['optimum'] & wheelacc['negative
large']),consequent=pressure['negative fast'])
rule30 = ctrl.Rule(antecedent=(wheelslip['optimum'] & wheelacc['negative
medium']),consequent=pressure['negative fast'])
rule31 = ctrl.Rule(antecedent=(wheelslip['optimum'] & wheelacc['negative
small']),consequent=pressure['negative slow'])
rule32 = ctrl.Rule(antecedent=(wheelslip['optimum'] & wheelacc['negative
few']),consequent=pressure['zero'])
rule33 = ctrl.Rule(antecedent=(wheelslip['optimum'] &
wheelacc['zero']),consequent=pressure['positive slow'])
rule34 = ctrl.Rule(antecedent=(wheelslip['optimum'] & wheelacc['positive
small']),consequent=pressure['positive slow'])
rule35 = ctrl.Rule(antecedent=(wheelslip['optimum'] & wheelacc['positive
large']),consequent=pressure['positive slow'])
rule36 = ctrl.Rule(antecedent=(wheelslip['too large'] & wheelacc['negative
large']),consequent=pressure['negative fast'])
rule37 = ctrl.Rule(antecedent=(wheelslip['too large'] & wheelacc['negative
medium']),consequent=pressure['negative fast'])
rule38 = ctrl.Rule(antecedent=(wheelslip['too large'] & wheelacc['negative
small']),consequent=pressure['negative slow'])
rule39 = ctrl.Rule(antecedent=(wheelslip['too large'] & wheelacc['negative
few']),consequent=pressure['zero'])
rule40 = ctrl.Rule(antecedent=(wheelslip['too large'] &
wheelacc['zero']),consequent=pressure['positive slow'])
rule41 = ctrl.Rule(antecedent=(wheelslip['too large'] & wheelacc['positive
small']),consequent=pressure['positive slow'])
rule42 = ctrl.Rule(antecedent=(wheelslip['too large'] & wheelacc['positive
large']),consequent=pressure['positive slow'])
rule43 = ctrl.Rule(antecedent=(wheelslip['very large'] & wheelacc['negative
large']),consequent=pressure['negative fast'])
rule44 = ctrl.Rule(antecedent=(wheelslip['very large'] & wheelacc['negative
medium']),consequent=pressure['negative fast'])
rule45 = ctrl.Rule(antecedent=(wheelslip['very large'] & wheelacc['negative
small']),consequent=pressure['negative slow'])
rule46 = ctrl.Rule(antecedent=(wheelslip['very large'] & wheelacc['negative
few']),consequent=pressure['zero'])
rule47 = ctrl.Rule(antecedent=(wheelslip['very large'] &
wheelacc['zero']),consequent=pressure['positive slow'])
rule48 = ctrl.Rule(antecedent=(wheelslip['very large'] & wheelacc['positive
small']),consequent=pressure['positive slow'])
```

```
rule49 = ctrl.Rule(antecedent=(wheelslip['very large'] & wheelacc['positive
large']),consequent=pressure['positive slow'])
# Controller simulation
rules = [rule1,
 rule2,
 rule3,
 rule4,
 rule5,
 rule6,
 rule7,
 rule8,
 rule9,
 rule10,
 rule11,
 rule12,
 rule13,
 rule14,
 rule15,
 rule16,
 rule17,
 rule18,
 rule19,
 rule20,
 rule21,
 rule22,
 rule23,
 rule24,
 rule25,
 rule26,
 rule27,
 rule28,
 rule29,
 rule30,
 rule31,
 rule32,
 rule33,
 rule34,
 rule35,
 rule36,
 rule37,
 rule38,
 rule39,
 rule40,
 rule41,
 rule42,
 rule43,
```

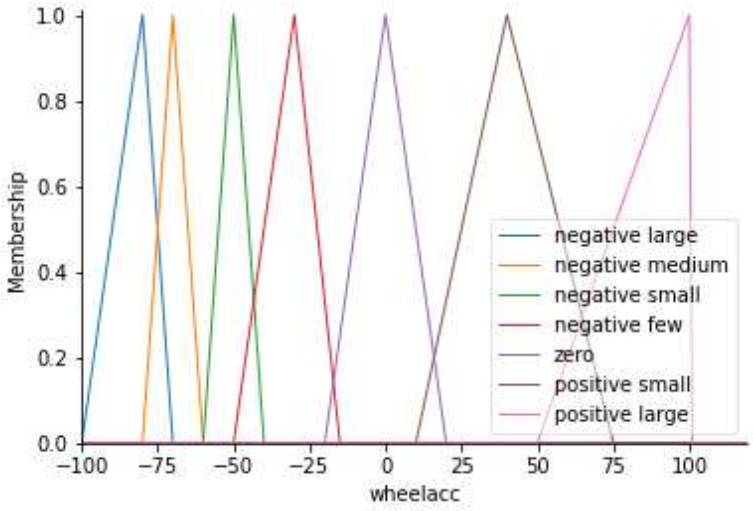
```
 rule44,
 rule45,
 rule46,
 rule47,
 rule48,
 rule49]

ABS_ctrl = ctrl.ControlSystem(rules)
ABS = ctrl.ControlSystemSimulation(ABS_ctrl)

# Automatic brake processing
ABS.input['wheelslip'] = 15
ABS.input['wheelacc'] = 40
ABS.compute()
pressure.view(sim = ABS)
print("Output for breaking pressure : " + str(ABS.output['pressure']))

if __name__ == "__main__":
      slip = int(input("Enter Slip ratio : "))
      acc = int(input("Enter Wheel acceleration in rad/s^2: "))
      ABS.input['wheelslip'] = slip
      ABS.input['wheelacc'] = acc
      ABS.compute()
      print("Output for breaking pressure : " + str(ABS.output['pressure'])+"kBar")
      pressure.view(sim = ABS)
```

OUTPUTS

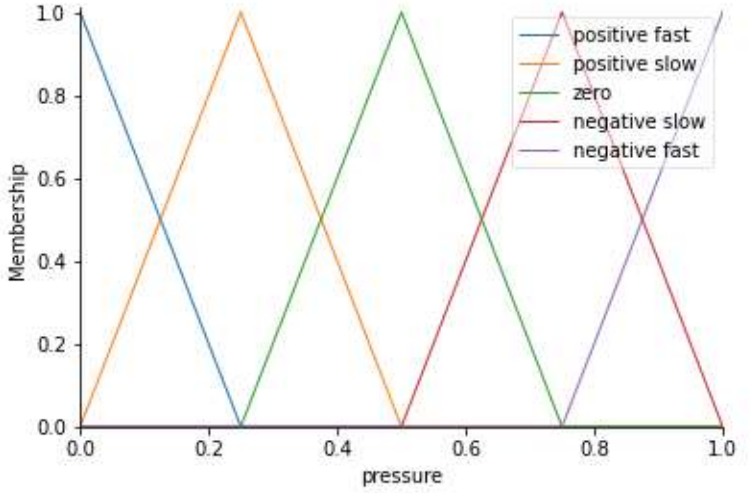
Slip ratio membership graph



Wheel acceleration membership graph



Brake Pressure Membership graph

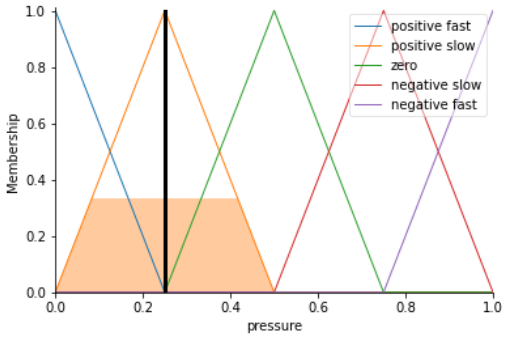


Testing output for predefined inputs

In [8]:
```
ABS_ctrl = ctrl.ControlSystem(rules)
ABS = ctrl.ControlSystemSimulation(ABS_ctrl)

# Automatic brake processing
ABS.input['wheelslip'] = 15
ABS.input['wheelacc'] = 40
ABS.compute()
pressure.view(sim = ABS)
print("Output for breaking pressure : " + str(ABS.output['pressure']))
```

```
C:\Users\Lenovo\anaconda3\lib\site-packages\skfuzzy\control\fuzzyvariable.py:122: UserWarning: Matplotlib is currently using mo
dule://ipykernel.pylab.backend_inline, which is a non-GUI backend, so cannot show the figure.
  fig.show()
```

```
Output for breaking pressure : 0.25
```

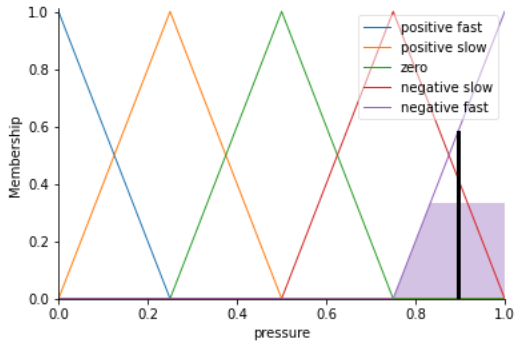


Testing with user defined inputs

In [9]:
```
if __name__ == "__main__":
        slip = int(input("Enter Slip ratio : "))
        acc = int(input("Enter Wheel acceleration in rad/s^2: "))
        ABS.input['wheelslip'] = slip
        ABS.input['wheelacc'] = acc
        ABS.compute()
        print("Output for breaking pressure : " + str(ABS.output['pressure'])+"kBar")
        pressure.view(sim = ABS)
```

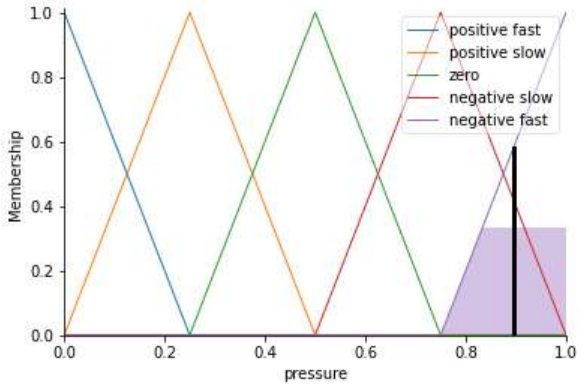
```
Enter Slip ratio : 20
Enter Wheel acceleration in rad/s^2: -65
Output for breaking pressure : 0.8944444444444446kBar
```

```
C:\Users\Lenovo\anaconda3\lib\site-packages\skfuzzy\control\fuzzyvariable.py:122: UserWarning: Matplotlib is currently using mo
dule://ipykernel.pylab.backend_inline, which is a non-GUI backend, so cannot show the figure.
  fig.show()
```



```
Enter Slip ratio : 20
Enter Wheel acceleration in rad/s^2: -65
Output for breaking pressure : 0.8944444444444446kBar
```

```
C:\Users\Lenovo\anaconda3\lib\site-packages\skfuzzy\control\fuzzyva
dule://ipykernel.pylab.backend_inline, which is a non-GUI backend,
  fig.show()
```



Conclusion

The basis of the controlling algorithm consists of a nonlinear characteristic surface, which was created by fuzzy logic. The convincing advantage of fuzzy logic is the ability to modify and tune certain parts of this characteristic surface easily and carefully. Just the linguistic rules or variables need to be varied. This simplifies the development and shortens the development time considerable. Implementation of the fuzzy ABS leads to excellent results of braking behavior of the test vehicle. The deceleration level and steer ability is comparable to commercially available systems. The use of fuzzy-logic in conjunction with microcontrollers is a fairly new development in automotive applications. Intel is not currently aware of any projects in production for ABS applications, Fuzzy Sets and Systems is an excellent journal devoted to fuzzy logic and control systems based on fuzzy logic.

REFERENCES

- https://www.ijser.org/paper/FUZZY-LOGIC-ANTI-LOCK-BRAKE-SYSTEM.html
- https://www.idosi.org/mejsr/mejsr20(10)14/19.pdf
- http://www.iaster.com/uploadfolder/8EnhancedAntilockBrakingSystemusingFuzzyLogicRoadDetectorGM24Sept13Copy/8Enhanced%20Antilock%20Braking%20System%20using%20Fuzzy%20Logic%20Road%20Detector%20GM24Sept13%20Copy.pdf
- https://github.com/XiangyuDing/Auto-Brake-System
- https://github.com/regisfaria/abs-fuzzy-system/blob/master/abs-fuzzy.py