

Laboratorium 5 - FuzzyLogic

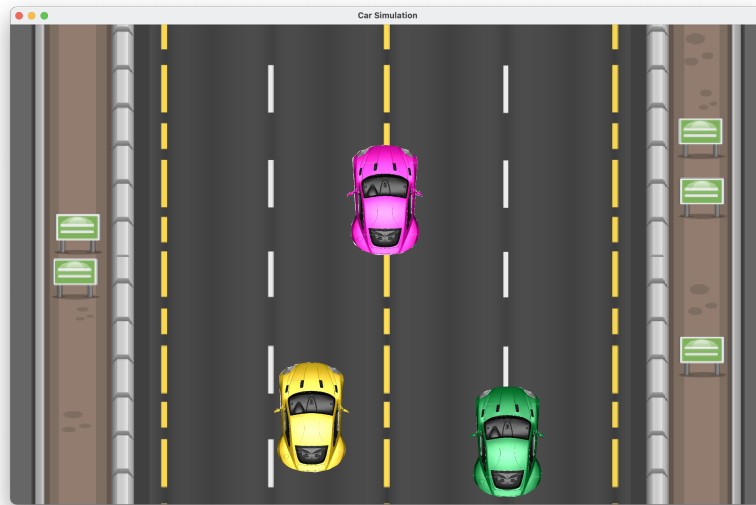
Bartosz Włodarski

4 stycznia 2022

1 Struktura katalogu

- Pliki .class - folder /com/company
- Pliki .java oraz plik .fcl - folder /src
- Pliki obrazów wykorzystanych w symulacji - folder /resources

2 Opis problemu



Rysunek 1: Symulacja

Opis sterownika znajduje się w pliku `src/fuzzy_move.fcl`. Sterownik rozmyty powstał specjalnie dla wcześniej stworzonego symulatora omijania pojazdów na drodze. Symulacja wykorzystuje Java Swing i nie wymaga innych zewnętrznych

bibliotek. Sterownik odpowiada za wyznaczenie kierunku jazdy samochodu (lub pozostaniu) tak aby samochodu nie zderzał się z innymi (co oczywiście nie zawsze się udaje). Tor jazdy jest ograniczony (od 155 do 895 pikseli).

3 Parametry wejściowe

Sterownik przyjmuje jako parametry

- obecne położenie samochodu (155-855) - `far_left`, `middle`, `far_right`
- odległość (w poziomie) od najbliższego pojazdu - `far_left`, `close_left`, `close_right`, `far_right`
- odległość od drugiego najbliższego pojazdu analogicznie do pierwszego

Jeśli chodzi o odległość od pojazdów to wartość ujemna oznacza, że samochód jest na prawo od naszego, a dodatnia oznacza, że jest na prawo od sterowanego samochodu. Wartości `close` oznaczają, że pojazdy są na torze kolizyjnym ze sobą.

```

FUNCTION_BLOCK MOVE

VAR_INPUT
car_position : REAL;
first_car_position : REAL;
second_car_position : REAL;
END_VAR

VAR_OUTPUT
move : REAL;
END_VAR

FUZZIFY car_position
TERM very_left := (155, 1) (280, 1) (280, 0);
TERM middle := (280,0) (280, 1) (470, 1) (730, 1) (730, 0);
TERM very_right := (730,0) (730, 1) (895, 1);
END_FUZZIFY

FUZZIFY first_car_position
TERM far_left := (-740, 1) (-300, 1) (-200, 0);
TERM close_left := (-300, 0) (-200, 1) (0, 1) (0, 0);
TERM close_right := (0, 0) (0, 1) (200, 1) (300, 0);
TERM far_right := (200, 0) (300, 1) (740, 1);
END_FUZZIFY

FUZZIFY second_car_position
TERM far_left := (-740, 1) (-300, 1) (-200, 0);
TERM close_left := (-300, 0) (-200, 1) (0, 1) (0, 0);
TERM close_right := (0, 0) (0, 1) (200, 1) (300, 0);
TERM far_right := (200, 0) (300, 1) (740, 1);
END_FUZZIFY

```

Rysunek 2: Definicja inputu w pliku fuzz_move.fcl

4 Parametr wyjściowy

Parametrem wyjściowym jest ruch, który powinien wykonać sterowany pojazd. Może przyjąć 5 wartości

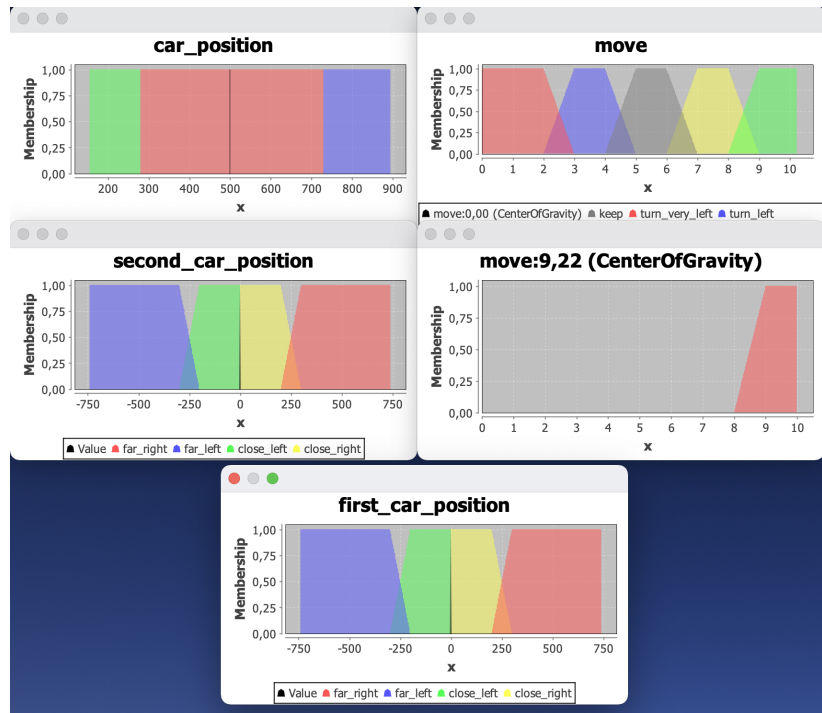
- turn_very_left - mocny skręt w lewo
- turn_left - skręt w lewo

- keep - brak zmian
- turn_right - skręt w prawo
- turn_very_right - mocny skręt w prawo

```
DEFUZZIFY move  
TERM turn_very_left := (0, 1) (2, 1) (3, 0);  
TERM turn_left := (2,0) (3,1) (4,1) (5,0);  
TERM keep := (4,0) (5,1) (6,1) (7,0);  
TERM turn_right := (6,0) (7,1) (8,1) (9,0);  
TERM turn_very_right := (8,0) (9,1) (10,1);  
METHOD : COG;  
DEFAULT := 0;  
END_DEFUZZIFY
```

Rysunek 3: Definicja outputu w pliku fuzz_move.fcl

5 Wykresy



Rysunek 4: Wykresy dla zdefiniowanego sterownika

6 Rules

Stworzone reguły można pogrupować na 2 kategorie

- gdy pojazd jest blisko krawędzi (very_left i very_right), reguły są analogiczne, wartości kierunku ruchu dla jednej strony są odbiciem lustrzanym ruchu dla drugiej strony (turn_left -i turn_right, turn_very_left -i turn_very_right, keep -i keep)
- gdy pojazd nie znajduje się blisko krawędzi (jest na środku)

7 Uruchomienie

1. Rozpakowanie pliku .zip
2. Otworzenie rozpakowanego katalogu w terminalu
3. Wykonanie komendy: `set CLASSPATH = %CLASSPATH%;(adres do pliku .jar z jFuzzyLogic)`

4. Uruchomienie samego sterownika poprzez komendę: `java com.company.FuzzyLogic (int: położenie sterowanego samochodu) (int: odległość od najbliższego samochodu) (int: odległość od drugiego najbliższego samochodu)`
5. Uruchomienie symulacji poprzez komendę: `java com.company.Main` (po zderzeniu samochodu można zrestartować symulację klikając dowolny przycisk)

7.1 Pojazd na lewej stronie (analogicznie dla prawej)

```

RULEBLOCK first
AND: MIN;
ACT: MIN;

RULE 0: IF (car_position IS very_left) AND (first_car_position IS far_left) AND (second_car_position IS far_left) THEN move IS turn_right;
RULE 1: IF (car_position IS very_left) AND (first_car_position IS far_left) AND (second_car_position IS close_left) THEN move IS turn_very_right;
RULE 3: IF (car_position IS very_left) AND (first_car_position IS far_left) AND (second_car_position IS close_right) THEN move IS turn_very_right;
RULE 4: IF (car_position IS very_left) AND (first_car_position IS far_left) AND (second_car_position IS far_right) THEN move IS turn_right;
RULE 5: IF (car_position IS very_left) AND (first_car_position IS close_left) THEN move IS turn_very_right;
RULE 6: IF (car_position IS very_left) AND (first_car_position IS close_right) THEN move IS turn_very_left;
RULE 7: IF (car_position IS very_left) AND (first_car_position IS far_right) AND (second_car_position IS far_left) THEN move IS turn_right;
RULE 8: IF (car_position IS very_left) AND (first_car_position IS far_right) AND (second_car_position IS close_left) THEN move IS turn_very_right;
RULE 9: IF (car_position IS very_left) AND (first_car_position IS far_right) AND (second_car_position IS close_right) THEN move IS turn_very_right;
RULE 10: IF (car_position IS very_left) AND (first_car_position IS far_right) AND (second_car_position IS far_right) THEN move IS turn_right;

```

Rysunek 5: Reguły dla pojazdu po lewej stronie

- Będąc blisko lewej krawędzi pojazd stara się zbliżyć ku środkowi, dlatego z jednym wyjątkiem pojazd powinien wykonać ruch w prawo (`turn_very_right` lub `turn_right`)
- Jeśli pojazd jest na torze kolizyjnym, z którymś z dwóch pojazdów wykonuje mocny skręt w prawo, w przeciwnym przypadku jedynie lekki
- Pojazd skręca mocno w lewo jedynie gdy najbliższy pojazd jest po jego prawej stronie i jest z nim na torze kolizyjnym.

7.2 Pojazd na środku

```

RULE 11: IF (car_position IS middle) AND (first_car_position IS far_left) AND (second_car_position IS far_left) THEN move IS keep;
RULE 12: IF (car_position IS middle) AND (first_car_position IS far_left) AND (second_car_position IS close_left) THEN move IS turn_very_right;
RULE 13: IF (car_position IS middle) AND (first_car_position IS far_left) AND (second_car_position IS close_right) THEN move IS turn_very_left;
RULE 14: IF (car_position IS middle) AND (first_car_position IS far_left) AND (second_car_position IS far_right) THEN move IS keep;
RULE 15: IF (car_position IS middle) AND (first_car_position IS close_left) THEN move IS turn_very_right;
RULE 16: IF (car_position IS middle) AND (first_car_position IS close_right) THEN move IS turn_very_left;
RULE 17: IF (car_position IS middle) AND (first_car_position IS far_right) AND (second_car_position IS far_left) THEN move IS keep;
RULE 18: IF (car_position IS middle) AND (first_car_position IS far_right) AND (second_car_position IS close_left) THEN move IS turn_very_right;
RULE 19: IF (car_position IS middle) AND (first_car_position IS far_right) AND (second_car_position IS close_right) THEN move IS turn_very_left;
RULE 20: IF (car_position IS middle) AND (first_car_position IS far_right) AND (second_car_position IS far_right) THEN move IS keep;

```

Rysunek 6: Reguły dla pojazdu na środku

Reguły dla pojazdu na środku są o wiele bardziej różnorodne.

- Jeśli dwa najbliższe pojazdy nie kolidują ze sterowanym, wtedy pojazd nie powinien skręcać (`keep`).

- Jeśli najbliższy pojazd koliduje z naszym pojazdem powinien mocno skrócić w stronę przeciwną do tej, po której jest ten pojazd.
- Ten sam manewr powinien być zastosowany dla drugiego najbliższego pojazdu, jednak tylko wtedy gdy najbliższy pojazd jest oddalony od sterowanego (najbliższy pojazd ma poprostu priorytet nad drugim).

8 Przykładowe uruchomienie sterownika

Parametry wejściowe:

- car_position: 500
- first_car_position: -50
- second_car_position: 0

Parametry wyjściowe

- Term keep: 0.0
- Term turn_very_left: 0.0
- Term turn_left: 0.0
- Term turn_very_right: 1.0
- Term turn_right: 0.0

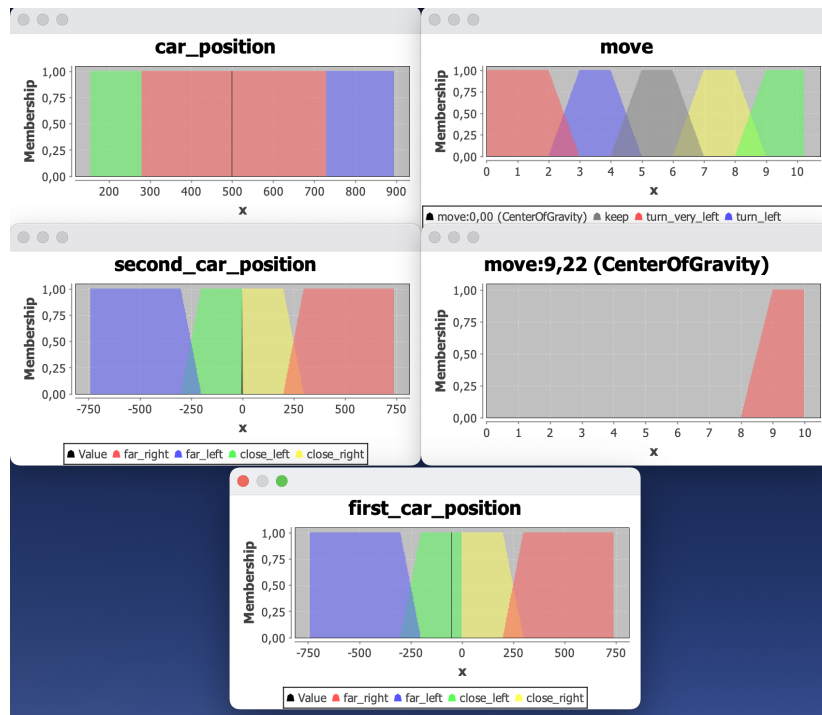
```

move :
Defuzzifier : CenterOfGravity
Latest defuzzified value: 9.219632107023326
Default value: 0.0
Term: keep      0.0   PiecewiseLinear : (4.0, 0.0) , (5.0, 1.0) , (6.0, 1.0) , (7.0, 0.0) ;
Term: turn_very_left 0.0   PiecewiseLinear : (0.0, 1.0) , (2.0, 1.0) , (3.0, 0.0) ;
Term: turn_left 0.0   PiecewiseLinear : (2.0, 0.0) , (3.0, 1.0) , (4.0, 1.0) , (5.0, 0.0) ;
Term: turn_very_right 1.0   PiecewiseLinear : (8.0, 0.0) , (9.0, 1.0) , (10.0, 1.0) ;
Term: turn_right 0.0   PiecewiseLinear : (6.0, 0.0) , (7.0, 1.0) , (8.0, 1.0) , (9.0, 0.0) ;

0.0

```

Rysunek 7: Przykładowe uruchomienie sterownika



Rysunek 8: Przykładowe uruchomienie sterownika - wykresy

Sterownik otrzymał informację o stanie, w którym nasz pojazd jest na środku a dwaj najbliższe pojazdy kolidują z nim (drugi centralnie, pierwszy od lewej strony). Sterownik zwraca rozkaz mocnego skrętu w prawo co jest prawidłową reakcją na ten stan