Fuzzy Control: The Truck Backer-Upper

Decision Support Methodologies (G53DSM)

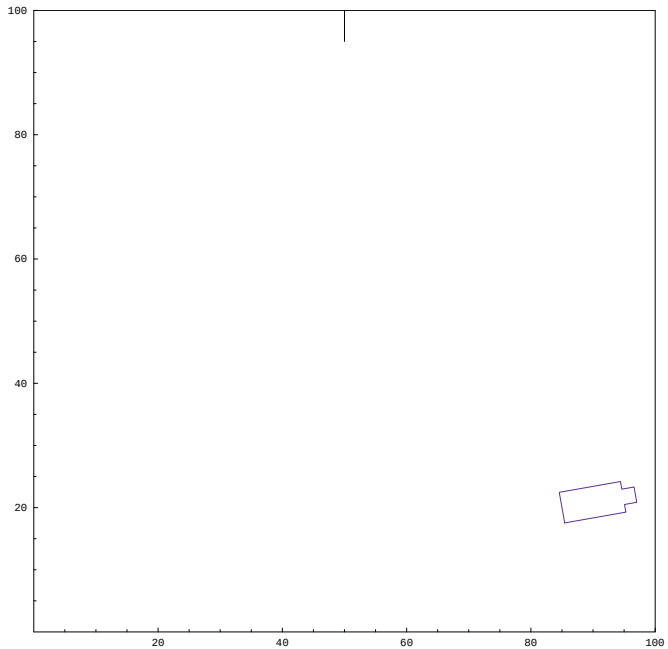
School of Computer Science
University of Nottingham

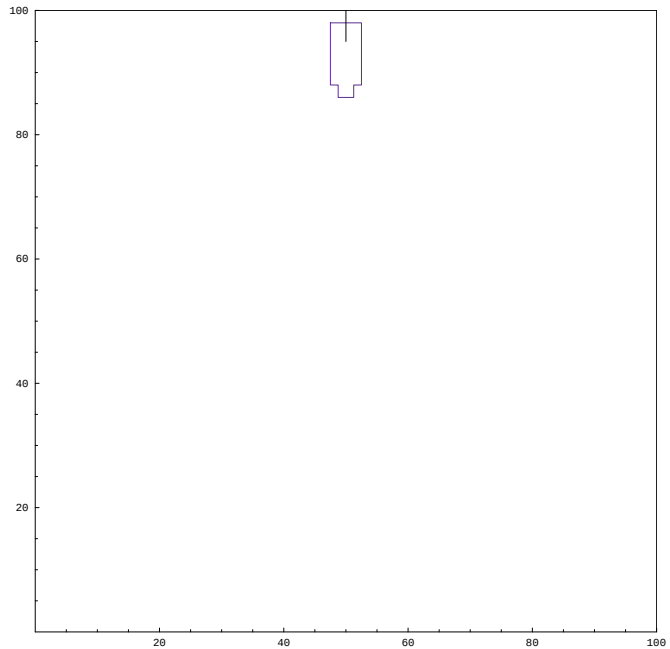
Outline

- Problem Statement
- Proposed Approach
- Testing the Controller

Problem Statement

Design a fuzzy control system that automatically back ups a truck to a given dock position in a 100 units wide and 100 units tall parking lot. The dock is located at the top-center of the parking lot.





Proposed Approach [Fre94]

Definition of inputs to the control system

Definition of outputs of the control system

Definition of Control Rules

Proposed Approach [Fre94]

Definition of inputs to the control system

1. Truck position

Definition of outputs of the control system

Definition of Control Rules

Proposed Approach [Fre94]

Definition of inputs to the control system

1. Truck position
2. Truck angle

Definition of outputs of the control system

Definition of Control Rules

Proposed Approach [Fre94]

Definition of inputs to the control system

1. Truck position
2. Truck angle

Definition of outputs of the control system

1. Steering angle

Definition of Control Rules

Proposed Approach [Fre94]

Definition of inputs to the control system

1. Truck position
2. Truck angle

Definition of outputs of the control system

1. Steering angle

Definition of Control Rules

1. Given certain linguistic values of the inputs then we compute the linguistic value for the output

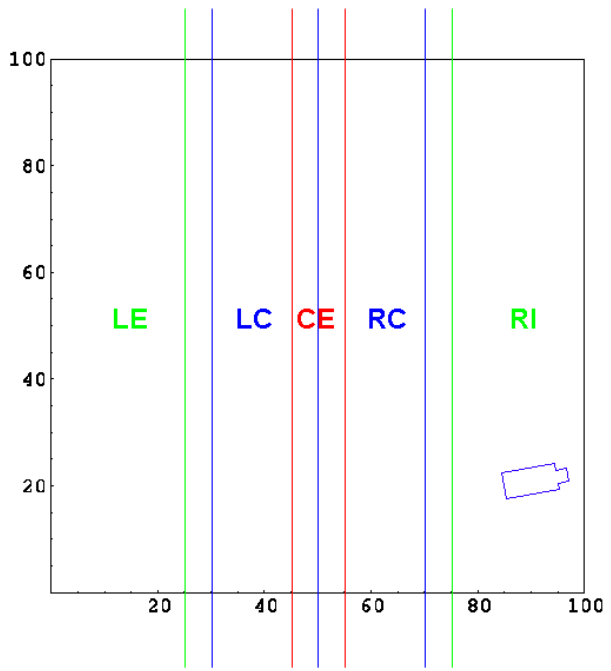
Proposed Approach

Definition of inputs to the control system

Truck position

Truck's x-position on the graph. Let's divide the parking lot in five regions along the x-axis. These regions will be the linguistic values:

1. LE: Left
2. LC: Left Center
3. CE: Center
4. RC: Right Center
5. RI: Right.



Proposed Approach

Definition of inputs to the control system

Truck position

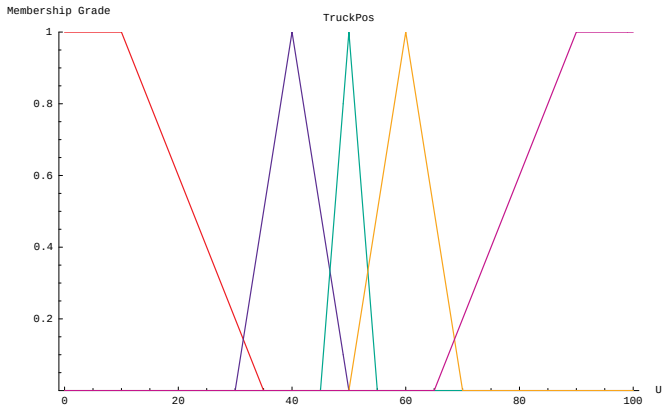
1. Change the default universal set to desired range, 0 to 100

```
SetOptions[FuzzySet,
     UniversalSpace -> {0, 100}];
```

2. Define Linguistic Variables

```
LE = FuzzyTrapezoid[0, 0, 10, 35];
LC = FuzzyTrapezoid[30, 40, 40, 50];
CE = FuzzyTrapezoid[45, 50, 50, 55];
RC = FuzzyTrapezoid[50, 60, 60, 70];
RI = FuzzyTrapezoid[65, 90, 100, 100];

TruckPos = {LE, LC, CE, RC, RI};
```



3. Plot the Membership Functions

```
FuzzyPlot[TruckPos, PlotJoined -> True,
                PlotLabel -> "TruckPos"];
```

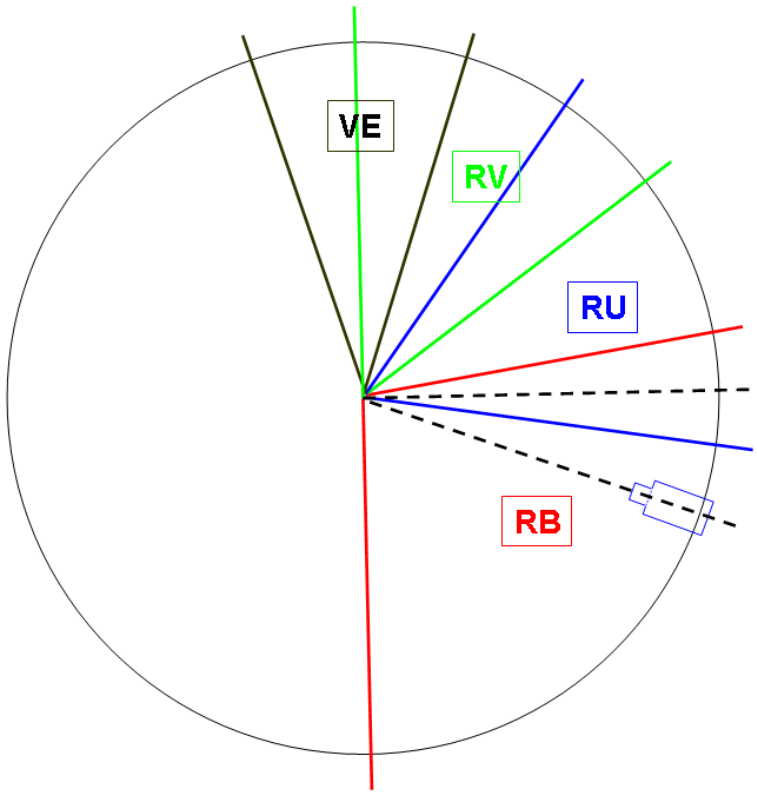
Proposed Approach

Definition of inputs to the control system

Truck angle

Angle of the truck axis to the horizontal. The angles ranges from -90 to 270 degrees.

1. RB: Right Below
2. RU: Right Upper
3. RV: Right Vertical
4. VE: Vertical
5. LV: Left Vertical
6. LU: Left Upper
7. LB: Left Below



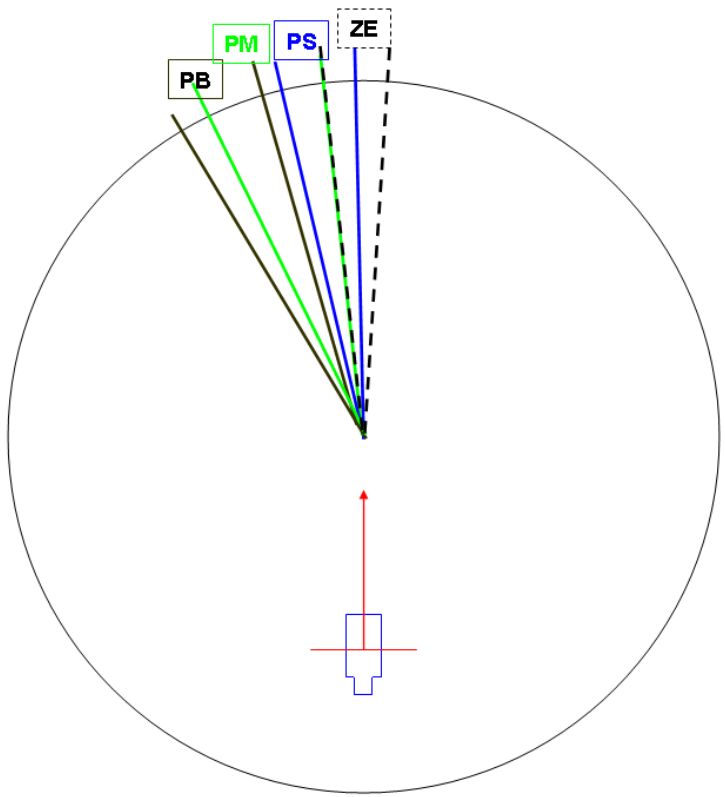
Proposed Approach

Definition of output of the control system

Steering angle

The output of our controller is a steering angle measured from the truck axis. The angle is limited to the range -30 to 30 degrees.

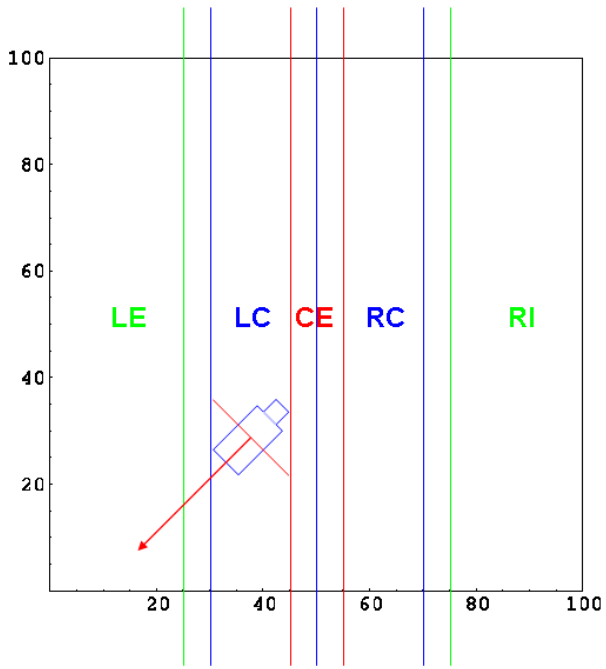
1. NB: Negative Big
2. NM: Negative Medium
3. NS: Negative Small
4. ZE: Zero
5. PS: Positive Small
6. PM: Positive Medium
7. PB: Positive Big



Proposed Approach

Definition of Control Rules

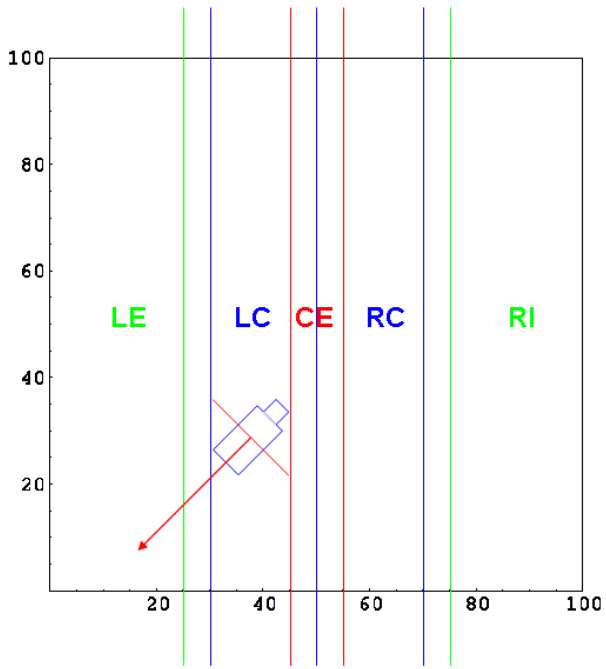
The rules are formed with linguistic terms, which follow human intuition.
IF x position is LC (Left Center) AND truck angle is LB (Left Below),
THEN steering angle is ...



Proposed Approach

Definition of Control Rules

The rules are formed with linguistic terms, which follow human intuition.
IF x position is LC (Left Center) AND truck angle is LB (Left Below),
THEN steering angle is ...



NB (Negative Big)

Proposed Approach

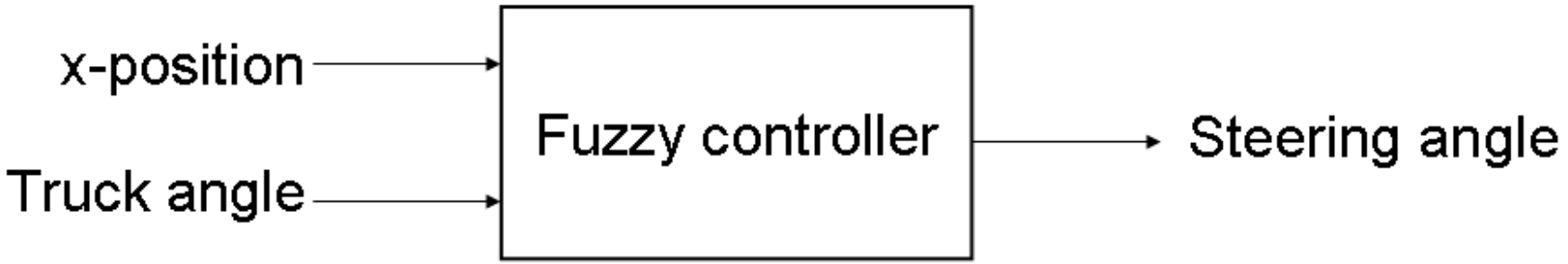
Definition of Control Rules

		Truck's x-position				
		LE-Left	**LC-LeftCenter**	**CE-Center**	**RC-RightCenter**	**RI-Right**
Truck Angle	**LB-Left Below**	NB	NB	NM	NM	NS
	LU-Left Upper	NB	NB	NM	NS	PS
	LV-Left Vertical	NB	NM	NS	PS	PM
	VE-Vertical	NM	NM	ZE	PM	PM
	RV-Right Vertical	NM	NS	PS	PM	PB
	RU-Right Upper	NS	PS	PM	PB	PB
	RB-Right Below	PS	PM	PM	PB	PB

```
ControlRules={{LE,RB,PS},{LC,RB,PM},{CE,RB,PM},{RC,RB,PB},{RI,RB,PB},{LE,RU,NS},
              {LC,RU,PS},{CE,RU,PM},{RC,RU,PB},{RI,RU,PB},{LE,RV,NM},{LC,RV,NS},
              {CE,RV,PS},{RC,RV,PM},{RI,RV,PB},{LE,VE,NM},{LC,VE,NM},{CE,VE,ZE},
              {RC,VE,PM},{RI,VE,PM},{LE,LV,NB},{LC,LV,NM},{CE,LV,NS},{RC,LV,PS},
              {RI,LV,PM},{LE,LU,NB},{LC,LU,NB},{CE,LU,NM},{RC,LU,NS},{RI,LU,PS},
              {LE,LB,NB},{LC,LB,NB},{CE,LB,NM},{RC,LB,NM},{RI,LB,NS}};
```

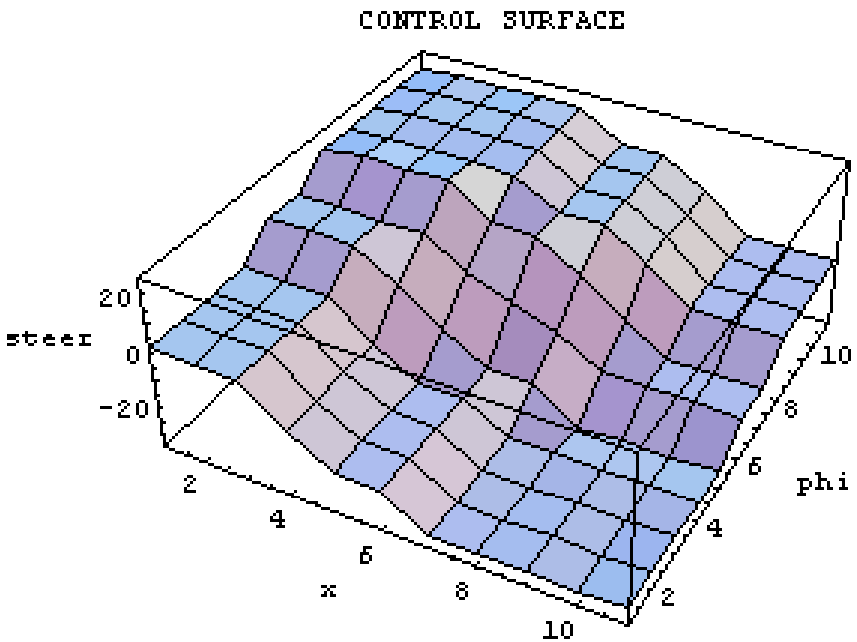
Testing the Controller

The controller



Control surface

3D graph showing the two-input/one-output relationship.



Testing the Controller

This first function models the fuzzy logic controller. phi and x represent the two inputs to the controller, the angle and truck position. The function will return a crisp output that corresponds to the steering angle.

```
Steer[phi_, x_] :=
CenterOfArea[RuleBasedInference[TruckPos, Angle, SteeringAngle, ControlRules, x, phi]]
```

Computing the trajectory

Given initial values of x, y, and angle of the truck, this function computes a list of configurations $\{x,\ y,$ truck's angle$\}$ giving a trajectory for the truck until it reaches a y value of at least 95.

```
simulateTruck[x0_, y0_, phi0_] :=
 Module[{x = x0, y = y0, phi = phi0, newPhi, result = {}},
  While[y <= 95,
   newPhi = phi + Steer[phi, x];
   AppendTo[result, {x, y, phi} = N[{x + 4 Cos[newPhi Pi / 180], y + 4 Sin[newPhi Pi / 1
  ];
 result]
```

Testing the Controller

Showing the truck

This function displays the position of the truck. It takes as inputs the triplet $\{x,\ y,$ truck's angle$\}$ and the length and width of the truck.

```
showTruck[{x_, y_, phi_}, {l_, w_}]
```

Showing the trajectory

We now map this function onto the list of configurations computed by `simulateTruck`

```
graph1 = (showTruck[#1, {10, 5}] &) /@ simlist
```

References


J. A. Freeman, *Fuzzy systems for control applications: the truck backer-upper*, The Mathematica Journal **4** (1994), 64–69.