

Fuzzy Truck Parking Demo

This demonstration of Fuzzy Logic has been created using the FuzzyJ Toolkit for the Java(tm) Platform that was created at the [National Research Council of Canada](#). The code for this demo is entirely Java(tm) code, although there is a capability to create more sophisticated rule based expert systems in concert with [Jess \(Expert System Shell\)](#) from Sandia National Laboratories (see the demo of the hybrid Java/Jess Fuzzy Shower).

This is a reworking of an example originally created by Christopher Britton. See *Neural Networks and Fuzzy Systems: A Dynamical Systems Approach to Machine Intelligence* by Bart Kosko (ISBN 0136114350) for more information on the problem and its solution.

This demonstration shows an application of fuzzy logic to park a truck by backing it up to a loading platform. The left side of the java applet has an area in which the truck is displayed. The truck has a yellow body (rear portion of the truck) and a red cab (front portion of the truck). It can be positioned anywhere in this area simply by clicking in the area. At the top of this area is a pink region in which the coordinates of the mouse are displayed (from 0,0 to 100,100) as the mouse moves while in this area. At the bottom of there is another pink region in which various status messages are displayed. The loading platform is at the top of the truck's display area and the center of the platform is shown by a small vertical line. The objective of the demonstration is for the truck to back up to the platform so that it is close to being centered on this line and at an angle of less than 5 degrees to the platform.

The truck can be positioned, as mentioned above, by clicking the mouse anywhere within the truck's driving area. The orientation (angle) of the truck can also be set using a slider on the lower right. This allows the truck to have an angle between -90 degrees and 270 degrees. Initially it is set at 90 degrees. In this position the rear of the truck is facing the top of the driving area. With a setting of -90 degrees the truck will have its rear end pointing directly down and with an angle of 0 degrees it will be pointing directly right.

To start the truck moving simply select the **Go** button on the right. The truck will move, backing up to the platform. If the truck goes out of the bounds of the driving area the simulation will stop. If it ends up with its back facing the platform, almost centered on the vertical line at the top, and within 5 degrees of vertical, then the attempt is successful.

The simulation can be controlled in various ways with several buttons and sliders on the right side. These are summarized below.

Go Button - As mentioned above this will start the simulation and the truck will attempt to back up to the loading platform. If it is already at the platform or off screen you will need to move it or use the Reset button to position it before selecting Go.

Step Button - This is similar to the Go button except that instead of continuously moving the truck, a single iteration of the fuzzy rule set is executed and the truck moves a small amount. One iteration of the rule set calculates the amount the truck should turn (angle change). Repeated presses of the Step button will move the truck through the back up procedure in small increments (depending on the speed of the truck).

Pause Button - A press of this button will halt the movement of the truck where it is. This might be useful if you want to stop the truck at some point to observe some things about the rule firings or to single Step through a part of the simulation.

Reset Button - This will move the truck to its previous starting point. This is very useful if you want to repeat the simulation from exactly the same starting position.

Truck Angle Slider - As detailed above, this slider changes the orientation of the truck. Initially the truck is at 90 degrees (rear facing the top). It is interesting to note how the truck behaves at different

positions and angles with the given set of rules.

Truck Speed Slider - The truck can be set to move at 3 different speeds. This just identifies how far the truck will travel on each iteration of the fuzzy rule firings. At a faster setting the truck will move further between decisions.

Simulation Speed Slider - This slider has 10 positions that determine how long the system will pause between rule firings. The longer the pause the slower the simulation will run, giving the user more time to catch the changes in the simulation as it proceeds. The Step button gives the user full control over the speed of the simulation.

A **check box**, above and just to the left of the sliders is labeled **Trace Truck Path**. It determines whether the truck leaves a trace of where it has been as it moves during the simulation. A trace can be useful to see the path that the truck took during the parking attempt.

A second **check box** labelled **Show Rule Firings** is used to control the display of colors in the rule matrix as described below.

The area in the upper right has a set of radio buttons, an array of buttons with 7 rows and 5 columns, and a button labeled **Reset Rules**. The array is a representation of the 35 rules used to control the simulation. The rules are of the form:

```
If    the truck is left of the platform's x position target by a medium
amount (LeftMedium)
and   the truck angle is below 90 degrees by a small amount
      (SmallBelow90)
then  change the angle of the truck by a negative small amount (NS)
```

Each row in the table represents a rule with a condition related to the truck angle. The individual rows correspond to:

```
LargeBelow90
MediumBelow90
SmallBelow90
At90
SmallAbove90
MediumAbove90
LargeAbove90
```

and each column from left to right represents a condition related to the x position of the truck relative to the target position marked on the loading platform. The columns correspond to:

```
LeftBig
LeftMedium
Centred
RightMedium
RightBig
```

The array positions are marked with 2 letter acronyms that identify the amount the angle of the truck should be changed if the rule fires (in fact these are the conclusions for the rules).

```
NB  - negative big change
NM  - negative medium change
NS  - negative small change
ZE  - zero (no change)
PS  - positive small change
PM  - positive medium change
PB  - positive big change
```

As the simulation runs, all of the rules represented at the array positions are considered for firing. If

the inputs match well enough then the rule will fire. The contribution of each of the rules that fires is aggregated to determine a total output for the rule set. This is an inference cycle. On each cycle, the array positions will have their background color changed if the rule fires (and the [Show Rule Firings CheckBox](#) is set). The colors will vary from white to deep red. White means the rule fired but the match of the conditions with the inputs was very low. Red means the rule fired and the match of the conditions with the inputs was very high. Various shades of red (pink) show matches between very low and very high. As well, the actual change in angle for each iteration will be displayed in the pink status line below the truck's parking area.

The user can experiment with the rules by changing the conclusions in the array positions. To do this, select one of the radio buttons to the left of the 7x5 array. This will select one of the rule outputs (NB, NM, etc.). Then click on one of the array locations and that location's label will be replaced by the label in the selected radio button. The [kill radio button](#) is used to place an empty label in an array position. This effectively removes the rule from the set of rules. An interesting experiment is to [kill](#) the rule at position 4,3 where a ZE is initially found and observe what it does to the stability of the truck's movements. To reset the rules to their original conclusions, select the [Reset Rules button](#).

The simulation applet should appear below. Note that your browser **MUST** be set up to use the JFC/Swing classes (version 1.1 or later). This may require that your classpath environment variable be set to include the required jar file (swingall.jar).

Sorry, No Java appears to be available for the browser!!!