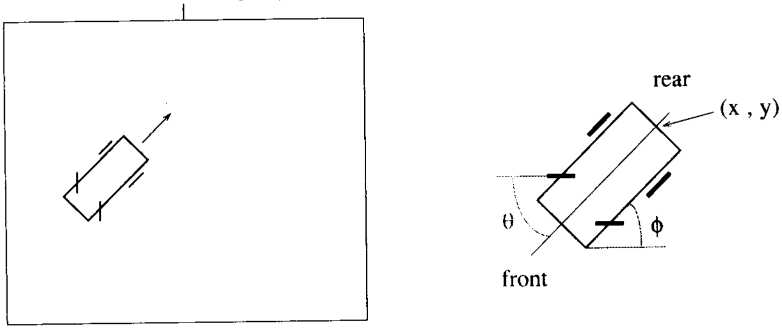


FIGURE 9.1 Diagram of simulated truck and loading zone.

BACKING UP A TRUCK

Figure 9.1 shows the simulated truck and loading zone. The truck corresponds to the cab part of the neural truck in the Nguyen-Widrow neural truck backer-upper system. The three state variables ϕ, x, and y exactly determine the truck position. ϕ specifies the angle of the truck with the horizontal. The coordinate pair (x, y) specifies the position of the rear center of the truck in the plane.

The goal was to make the truck arrive at the loading dock at a right angle ($\phi_f = 90°$) and to align the position (x, y) of the truck with the desired loading dock (x_f, y_f). We considered only backing up. The truck moved backward by some fixed distance at every stage. The loading zone corresponded to the plane $[0, 100] \times [0, 100]$, and (x_f, y_f) equaled (50, 100).

At every stage the fuzzy and neural controllers should produce the steering angle θ that backs up the truck to the loading dock from any initial position and from any angle in the loading zone.

Fuzzy Truck Backer-Upper System

We first specified each controller's input and output variables. The input variables were the truck angle ϕ and the x-position coordinate x. The output variable was the steering-angle signal θ. We assumed enough clearance between the truck and the loading dock so we could ignore the y-position coordinate. The variable ranges were as follows:

$$0 \le x \le 100$$

$$-90 \le \phi \le 270$$

$$-30 \le \theta \le 30$$

Positive values of θ represented clockwise rotations of the steering wheel. Negative values represented counterclockwise rotations. We discretized all values to reduce computation. The resolution of ϕ and θ was one degree each. The resolution of x was 0.1.

Next we specified the fuzzy-set values of the input and output fuzzy variables. The fuzzy sets numerically represented linguistic terms, the sort of linguistic terms an expert might use to describe the control system's behavior. We chose the fuzzy-set values of the fuzzy variables as follows:

Angle ϕ		x-position x		Steering-angle signal θ	
RB:	Right Below	LE:	Left	NB:	Negative Big
RU:	Right Upper	LC:	Left Center	NM:	Negative Medium
RV:	Right Vertical	CE:	Center	NS:	Negative Small
VE:	Vertical	RC:	Right Center	ZE:	Zero
LV:	Left Vertical	RI:	Right	PS:	Positive Small
LU:	Left Upper			PM:	Positive Medium
LB:	Left Below			PB:	Positive Big

Fuzzy subsets contain elements with degrees of membership. A fuzzy membership function m_A: $Z \longrightarrow [0, 1]$ assigns a real number between 0 and 1 to every element z in the universe of discourse Z. This number $m_A(z)$ indicates the degree to which the object or data z belongs to the fuzzy set A. Equivalently, $m_A(z)$ defines the *fit* (fuzzy unit) value [Kosko, 1986] of element z in A.

Fuzzy membership functions can have different shapes depending on the designer's preference or experience. In practice fuzzy engineers have found triangular and trapezoidal shapes help capture the modeler's sense of fuzzy numbers and simplify computation. Figure 9.2 shows membership-function graphs of the fuzzy subsets above. In the third graph, for example, $\theta = 20°$ is Positive Medium to degree 0.5, but only Positive Big to degree 0.3.

In Figure 9.2 the fuzzy sets CE, VE, and ZE are narrower than the other fuzzy sets. These narrow fuzzy sets permit fine control near the loading dock. We used wider fuzzy sets to describe the endpoints of the range of the fuzzy variables ϕ, x, and θ. The wider fuzzy sets permitted rough control far from the loading dock.

Next we specified the fuzzy "rulebase" or bank of *fuzzy associative memory* (FAM) rules. Fuzzy associations or "rules" (A, B) associate output fuzzy sets B of control values with input fuzzy sets A of input-variable values. We can write fuzzy associations as antecedent-consequent pairs or IF-THEN statements.

In the truck backer-upper case, the FAM bank contained the 35 FAM rules in Figure 9.3. For example, the FAM rule of the left upper block (FAM rule 1) corresponds to the following fuzzy association:

$$\text{IF } x = \text{LE AND } \phi = \text{RB}, \text{ THEN } \theta = \text{PS}$$

FAM rule 18 indicates that if the truck is in near the equilibrium position, then the controller should not produce a positive or negative steering-angle signal. The FAM rules in the FAM-bank matrix reflect the symmetry of the controlled system.

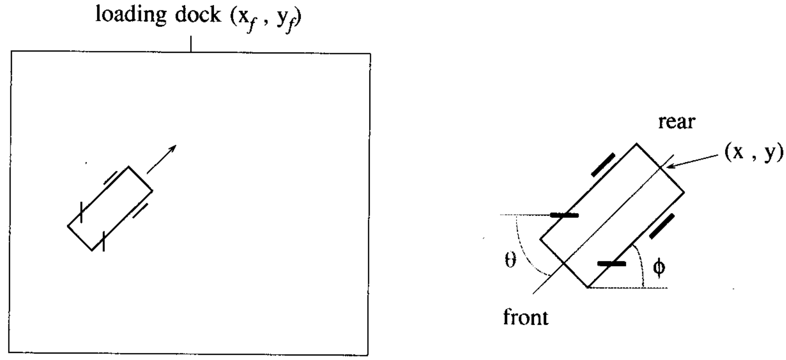


FIGURE 9.1 Diagram of simulated truck and loading zone.

BACKING UP A TRUCK

Figure 9.1 shows the simulated truck and loading zone. The truck corresponds to the cab part of the neural truck in the Nguyen-Widrow neural truck backer-upper system. The three state variables ϕ, x, and y exactly determine the truck position. ϕ specifies the angle of the truck with the horizontal. The coordinate pair (x, y) specifies the position of the rear center of the truck in the plane.

The goal was to make the truck arrive at the loading dock at a right angle ($\phi_f = 90°$) and to align the position (x, y) of the truck with the desired loading dock (x_f, y_f). We considered only backing up. The truck moved backward by some fixed distance at every stage. The loading zone corresponded to the plane $[0, 100] \times [0, 100]$, and (x_f, y_f) equaled (50, 100).

At every stage the fuzzy and neural controllers should produce the steering angle θ that backs up the truck to the loading dock from any initial position and from any angle in the loading zone.

Fuzzy Truck Backer-Upper System

We first specified each controller's input and output variables. The input variables were the truck angle ϕ and the x-position coordinate x. The output variable was the steering-angle signal θ. We assumed enough clearance between the truck and the loading dock so we could ignore the y-position coordinate. The variable ranges were as follows:

$$0 \le x \le 100$$

$$-90 \le \phi \le 270$$

$$-30 \le \theta \le 30$$

Positive values of θ represented clockwise rotations of the steering wheel. Negative values represented counterclockwise rotations. We discretized all values to reduce computation. The resolution of ϕ and θ was one degree each. The resolution of x was 0.1.

Next we specified the fuzzy-set values of the input and output fuzzy variables. The fuzzy sets numerically represented linguistic terms, the sort of linguistic terms an expert might use to describe the control system's behavior. We chose the fuzzy-set values of the fuzzy variables as follows:

Angle ϕ		x-position x		Steering-angle signal θ	
RB:	Right Below	LE:	Left	NB:	Negative Big
RU:	Right Upper	LC:	Left Center	NM:	Negative Medium
RV:	Right Vertical	CE:	Center	NS:	Negative Small
VE:	Vertical	RC:	Right Center	ZE:	Zero
LV:	Left Vertical	RI:	Right	PS:	Positive Small
LU:	Left Upper			PM:	Positive Medium
LB:	Left Below			PB:	Positive Big

Fuzzy subsets contain elements with degrees of membership. A fuzzy membership function m_A: $Z \longrightarrow [0, 1]$ assigns a real number between 0 and 1 to every element z in the universe of discourse Z. This number $m_A(z)$ indicates the degree to which the object or data z belongs to the fuzzy set A. Equivalently, $m_A(z)$ defines the *fit* (fuzzy unit) value [Kosko, 1986] of element z in A.

Fuzzy membership functions can have different shapes depending on the designer's preference or experience. In practice fuzzy engineers have found triangular and trapezoidal shapes help capture the modeler's sense of fuzzy numbers and simplify computation. Figure 9.2 shows membership-function graphs of the fuzzy subsets above. In the third graph, for example, $\theta = 20°$ is Positive Medium to degree 0.5, but only Positive Big to degree 0.3.

In Figure 9.2 the fuzzy sets CE, VE, and ZE are narrower than the other fuzzy sets. These narrow fuzzy sets permit fine control near the loading dock. We used wider fuzzy sets to describe the endpoints of the range of the fuzzy variables ϕ, x, and θ. The wider fuzzy sets permitted rough control far from the loading dock.

Next we specified the fuzzy "rulebase" or bank of *fuzzy associative memory* (FAM) rules. Fuzzy associations or "rules" (A, B) associate output fuzzy sets B of control values with input fuzzy sets A of input-variable values. We can write fuzzy associations as antecedent-consequent pairs or IF-THEN statements.

In the truck backer-upper case, the FAM bank contained the 35 FAM rules in Figure 9.3. For example, the FAM rule of the left upper block (FAM rule 1) corresponds to the following fuzzy association:

$$\text{IF } x = \text{LE AND } \phi = \text{RB}, \text{ THEN } \theta = \text{PS}$$

FAM rule 18 indicates that if the truck is in near the equilibrium position, then the controller should not produce a positive or negative steering-angle signal. The FAM rules in the FAM-bank matrix reflect the symmetry of the controlled system.

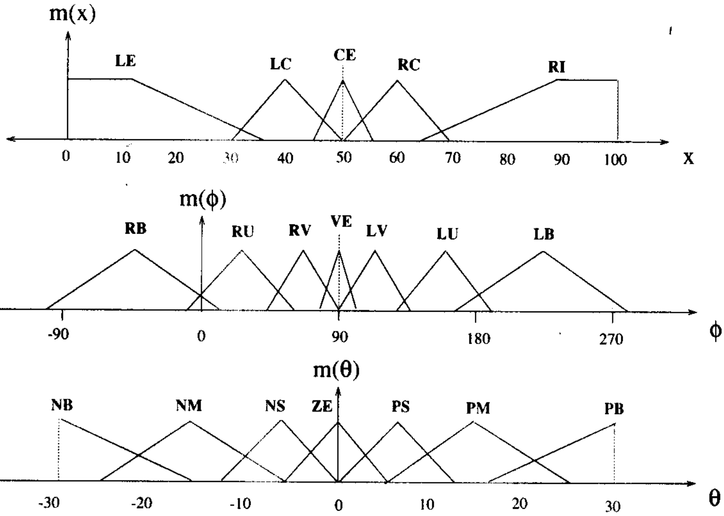


FIGURE 9.2 Fuzzy membership functions for each linguistic fuzzy-set value. To allow finer control, the fuzzy sets that correspond to near the loading dock are narrower than the fuzzy sets that correspond to far from the loading dock.

ϕ \ x	LE	LC	CE	RC	RI
RB	1 **PS**	2 **PM**	3 **PM**	4 **PB**	5 **PB**
RU	6 **NS**	7 **PS**	**PM**	**PB**	**PB**
RV	**NM**	**NS**	**PS**	**PM**	**PB**
VE	**NM**	**NM**	18 **ZE**	**PM**	**PM**
LV	**NB**	**NM**	**NS**	**PS**	**PM**
LU	**NB**	**NB**	**NM**	**NS**	**PS**
LB	**NB**	**NB**	**NM**	**NM**	35 **NS**

FIGURE 9.3 FAM-bank matrix for the fuzzy truck backer-upper controller.

For the initial condition $x = 50$ and $\phi = 270$, the fuzzy truck did not perform well. The symmetry of the FAM rules and the fuzzy sets cancelled the fuzzy controller output in a rare saddle point. For this initial condition, the neural controller (and truck-and-trailer below) also performed poorly. Any perturbation breaks the symmetry. For example, the rule (IF $x = 50$ AND $\phi = 270$, THEN $\theta = 5$) corrected the problem.

The three-dimensional control surfaces in Figure 9.4 show steering-angle signal outputs θ that correspond to all combinations of values of the two input state

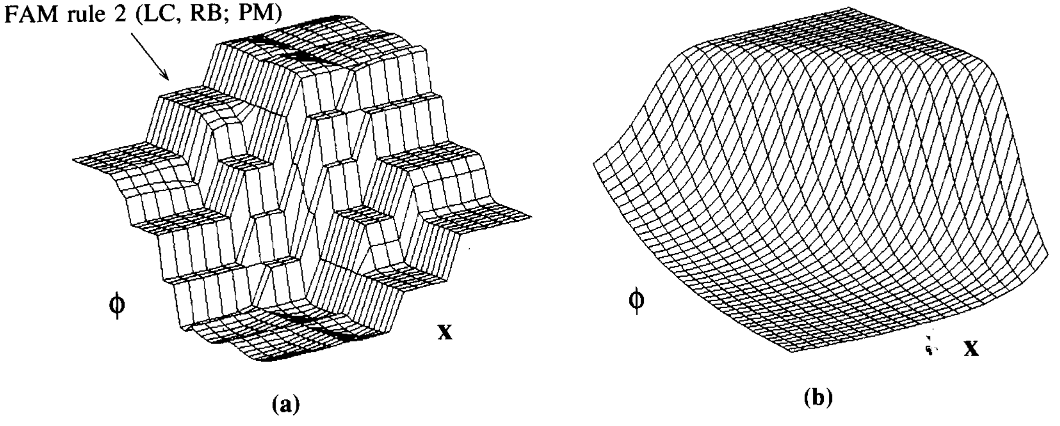


FIGURE 9.4 (a) Control surface of the fuzzy controller. Fuzzy-set values determined the input and output combination corresponding to FAM rule 2 (IF $x =$ LC AND $\phi =$ RB, THEN $\theta =$ PM). (b) Corresponding control surface of the neural controller for constant value $y = 20$.

variables ϕ and x. The control surface defines the fuzzy controller. In this simulation the correlation-minimum FAM inference procedure, discussed in Chapter 8, determined the fuzzy control surface. If the control surface changes with sampled variable values, the system behaves as an *adaptive* fuzzy controller. Below we demonstrate unsupervised adaptive control of the truck and the truck-and-trailer systems.

Finally, we determined the output action given the input conditions. We used the correlation-minimum inference method illustrated in Figure 9.5. Each FAM rule produced the output fuzzy set clipped at the degree of membership determined by the input conditions and the FAM rule. Alternatively, correlation-product inference would combine FAM rules multiplicatively. Each FAM rule emitted a fit-weighted output fuzzy set O_i at each iteration. The total output O added these weighted outputs:

$$O = \sum_i O_i \tag{9-1}$$

$$= \sum_i \min(f_i, S_i) \tag{9-2}$$

where f_i denotes the antecedent fit value and S_i represents the consequent fuzzy set of steering-angle values in the ith FAM rule. Earlier fuzzy systems combined the output sets O_i with pairwise maxima. But this tends to produce a uniform output set O as the number of FAM rules increases. Adding the output sets O_i invokes the fuzzy version of the central limit theorem. This tends to produce a symmetric, unimodal output fuzzy set O of steering-angle values.

Fuzzy systems map fuzzy sets to fuzzy sets. The fuzzy control system's

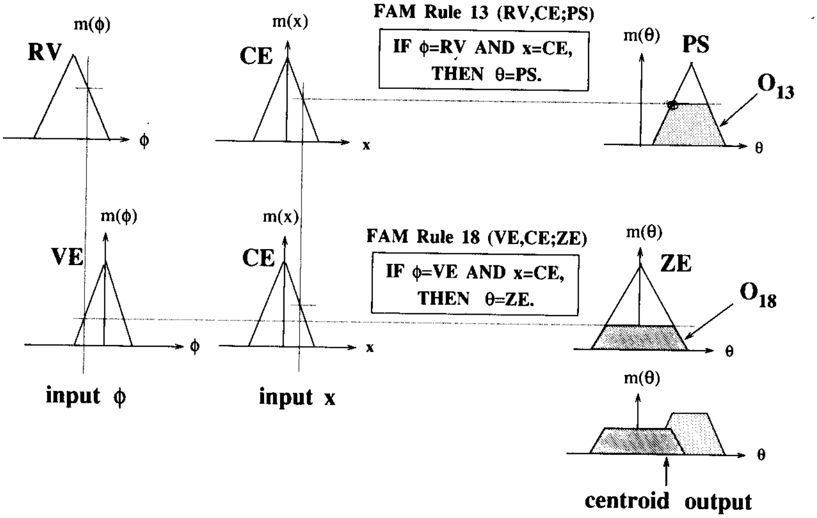


FIGURE 9.5 Correlation-minimum inference with centroid defuzzification method. Then FAM-rule antecedents combined with AND use the *minimum* fit value to activate consequents. Those combined with OR would use the *maximum* fit value.

output defines the fuzzy set O of steering-angle values at each iteration. We must "defuzzify" the fuzzy set O to produce a numerical (point-estimate) steering-angle output value θ.

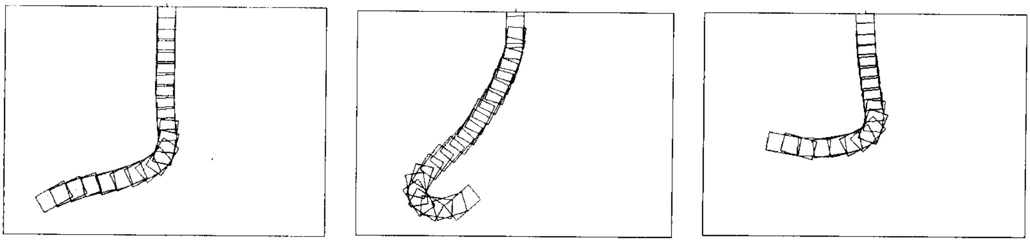
As discussed in Chapter 8, the simplest defuzzification scheme selects the value corresponding to the *maximum fit* value in the fuzzy set. This mode-selection approach ignores most of the information in the output fuzzy set and requires an additional decision algorithm when multiple modes occur.

Centroid defuzzification provides a more effective procedure. This method uses the *fuzzy centroid* $\bar{\theta}$ as output:

$$\bar{\theta} = \frac{\sum_{j=1}^{p} \theta_j m_O(\theta_j)}{\sum_{j=1}^{p} m_O(\theta_j)} \tag{9-3}$$

where O defines a fuzzy subset of the steering-angle universe of discourse $\Theta = \{\theta_1, \ldots, \theta_p\}$. The central-limit-theorem effect produced by adding output fuzzy set O_i benefits both max-mode and centroid defuzzification. Figure 9.5 shows the correlation-minimum inference and centroid defuzzification applied to FAM rules 13 and 18. We used centroid defuzzification in all simulations.

With 35 FAM rules, the fuzzy truck controller produced successful truck backing-up trajectories starting from any initial position. Figure 9.6 shows typical



(a) (b) (c)

FIGURE 9.6 Sample truck trajectories of the fuzzy controller for initial positions (x, y, ϕ): (a) (20, 20, 30), (b) (30, 10, 220), and (c) (30, 40, −10).

examples of the fuzzy-controlled truck trajectories from different initial positions. The fuzzy control system did not use ("fire") all FAM rules at each iteration. Equivalently most output consequent sets are empty. In most cases the system used only one or two FAM rules at each iteration. The system used at most 4 FAM rules at once.

Neural Truck Backer-Upper System

The neural truck backer-upper of Nguyen and Widrow [1989] consisted of multilayer feedforward neural networks trained with the backpropagation gradient-descent (stochastic-approximation) algorithm. The *neural control system* consisted of two neural networks: the controller network and the truck emulator network. The *controller network* produced an appropriate steering-angle signal output given any parking-lot coordinates (x, y), and the angle ϕ. The *emulator network* computed the next position of the truck. The emulator network took as input the previous truck position and the current steering-angle output computed by the controller network.

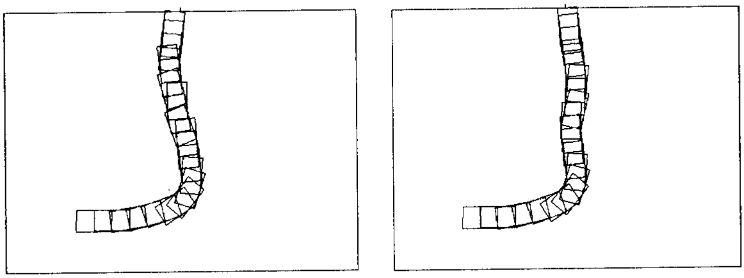
We did not train the emulator network since we could not obtain "universal" synaptic connection weights for the truck-emulator network. The backpropagation learning algorithm did not converge for some sets of training samples. The number of training samples for the emulator network might exceed 3000. For example, the combinations of training samples of a given angle ϕ, x-position, y-position, and steering-angle signal θ might correspond to 3150 ($18 \times 5 \times 5 \times 7$) samples, depending on the division of the input-output product space. Moreover, the training samples were numerically similar, since the neuronal signals assumed scaled values in [0, 1] or [−1, 1]. For example, we treated close values, such as 0.40 and 0.41, as distinct sample values.

Simple kinematic equations replaced the truck-emulator network. If the truck moved backward from (x, y) to (x', y') at an iteration, then

$$x' = x + r\cos(\phi') \tag{9-4}$$

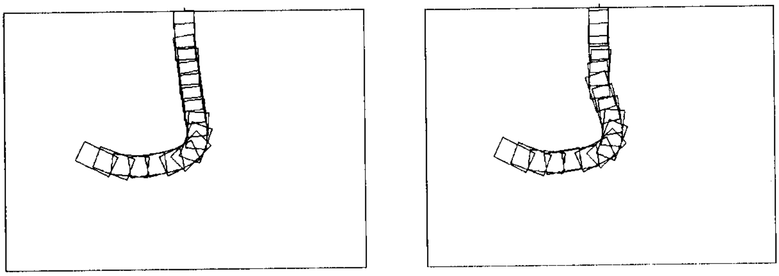
$$y' = y + r\sin(\phi') \tag{9-5}$$

$$\phi' = \phi + \theta \tag{9-6}$$



(a) (b)

FIGURE 9.9 The fuzzy truck trajectory after we replaced the key steady-state FAM rule 18 by the two worst rules: (a) IF x = CE AND ϕ = VE, THEN θ = PB; (b) IF x = CE AND ϕ = VE, THEN θ = NB.



(a) (b)

FIGURE 9.10 Fuzzy truck trajectory when (a) no FAM rules are removed and (b) FAM rules 7, 13, 18, and 23 are removed.

final position (ϕ, x, y) to the desired final position (ϕ_f, x_f, y_f):

$$\text{Docking error} \quad = \quad \sqrt{(\phi_f - \phi)^2 + (x_f - x)^2 + (y_f - y)^2} \tag{9-7}$$

In Figure 9.11(b), the trajectory error equaled the ratio of the actual trajectory length of the truck divided by the straight-line distance to the loading dock:

$$\text{Trajectory error} \quad = \quad \frac{\text{length of truck trajectory}}{\text{distance(initial position, desired final position)}} \tag{9-8}$$

Adaptive Fuzzy Truck Backer-Upper

Adaptive FAM (AFAM) systems generate FAM rules directly from training data. A one-dimensional FAM system, S: $I^n \longrightarrow I^p$, defines a FAM rule, a single association of the form (A_i, B_i). In this case the input-output product space equals $I^n \times I^p$. As discussed in Chapter 8, a FAM rule (A_i, B_i) defines a cluster or ball