EE7207 MINI PROJECT
DUE : 12 NOV 2021 by softcopy submission via email to eekpoh@ntu.edu.sg
Complete and Submit All 2 Questions.

Note: You can use MatLab/Simulink or any other software package to perform your simulation.

1. FUZZY CONTROL. We first consider a truck-parking control problem, as shown in Figure 1.1. The objective is to design a controller, without assuming a mathematical model of the truck, to park the truck anywhere on the *x*-axis.

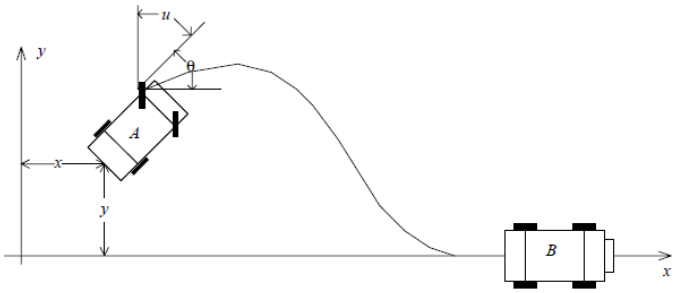

Figure 1.1

(a) Suppose that the truck can move forward at a constant speed of v = 0.5 *m/s* and it is assumed that the truck is equipped with sensors that can measure location (x,y) and orientation θ (angle) at all times. The fuzzy logic controller is to provide an input, u, to rotate the steering wheels and, consequently, to maneuver the truck from any given initial condition to reach y=θ=0 at the parked position. In the simulation, the input variables are the truck angle and the vertical position coordinate, y, while the output variable is the steering angle (signal), u. The variable ranges are pre-assigned as

$$-100 \le y \le 100,\ -180° \le \theta \le 180°,\ -30° \le u \le 30°.$$

Here, clockwise rotations are considered positive in θ, and, therefore, counter-clockwise are negative.

A simplified model to generate the values of (x,y) and orientation θ (angle) can be obtained from the geometry of the truck shown in Figure 1.1 as follows:

$$\begin{aligned} \theta(k+1) &= \theta(k) + v\,T \tan(\,u(k)\,) / L, \\ x(k+1) &= x(k) + v\,T \cos(\,\theta(k)\,), \\ y(k+1) &= y(k) + v\,T \sin(\,\theta(k)\,), \end{aligned} \tag{1.1}$$

where

$\theta(k)$	—	angle of the truck at time k,
$x(k)$	—	horizontal position of the truck rear-end at time k,
$y(k)$	—	vertical position of the truck rear-end at time k,
$u(k)$	—	steering angle as control input to the truck at time k,
L	—	length of the truck (L=2.5m),
T	—	sampling time of the discrete model (T=0.1s),
v	—	constant speed of the truck (v=0.5m/s).

The linguistic terms used in this design are given as follows:

y-position	Angle θ	Steering angle *u*
AB: Above	AO: Above much	NB: Negative big
AC: Above center	AR: Above	NM: Negative medium
CE: Center	AH: Above horizontal	NS: Negative small
BC: Below center	HZ: Horizontal	ZE: Zero
BE: Below	BH: Below horizontal	PS: Positive small
	BR: Below	PM: Positive medium
	BO: Below much	PB: Positive big

Table 1.1

The associated membership functions for the variables *y* and θ are shown in Figure 1.2 and 1.3 respectively.

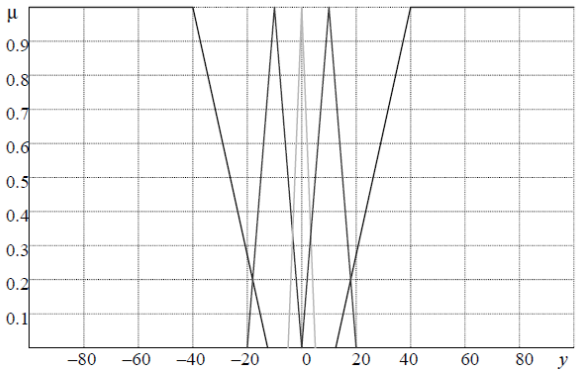

Figure 1.2

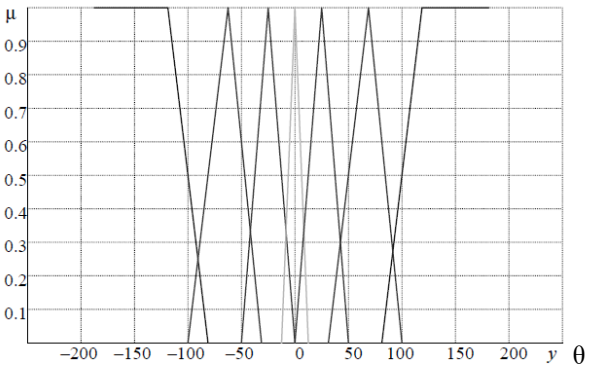

Figure 1.3

The rule base used in simulation is summarized in Table 1.2. Each rule has the form IF *y* is *Y* AND θ is Θ THEN *u* is *U*, as usual. A glance of these rules reveals the symmetry, an intrinsic property of this controller, which is reasonable for this parking application since the truck can move in any direction.

	BE	BC	CE	AC	AB
BO	*PB*	*PB*	*PM*	*PM*	*PS*
BR	*PB*	*PB*	*PM*	*PS*	*NS*
BH	*PB*	*PM*	*PS*	*NS*	*NM*
HZ	*PM*	*PM*	*ZE*	*NM*	*NM*
AH	*PM*	*PS*	*NS*	*NM*	*NB*
AR	*PS*	*NS*	*NM*	*NB*	*NB*
AO	*NS*	*NM*	*NM*	*NB*	*NB*

Table 1.2

The membership function of the control variable u is shown in Figure 1.4.

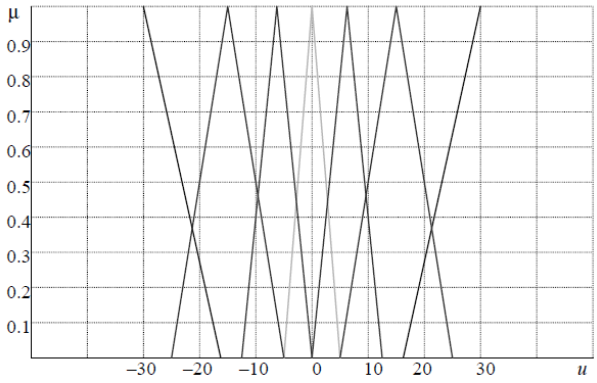

Figure 1.4

Finally, we need to obtain the output action u under the given input conditions. The following standard Centre of Gravity (COG) defuzzification formula is to be used in the simulation:

$$u_{COG} = \frac{\int_U u_i \ \mu_{U_i}(u_i) du_i}{\int_U \mu_{U_i}(u_i) du_i}$$

Design a fuzzy logic controller and perform simulations for the following 4 initial conditions and plot out the corresponding $u(t)$, $y(t)$ and $\theta(t)$ to show that y and θ reaches 0 at the end of the parked position.

Case	1	2	3	4
θ	0	90	220	-10
y	40	-30	30	10
x	20	10	40	50

Table 1.3

(b) For the given model in Equation (1.1), design a classical non-fuzzy logic controller to reach y=θ=0 given the same 4 initial conditions listed in Table 1.3 and plot out the results.

2. CLUSTERING. Clustering is one of the most fundamental issues in pattern recognition. It plays a key role in searching for structures in data. Given a finite set of data, **X**, the problem of clustering in **X** is to find several cluster centres that can properly characterize relevant classes of **X**. In classical cluster analysis, these classes are required to form a partition of **X** such that the degree of association is strong for data within blocks of the partition and weak for data in different blocks. However, this requirement is too strong in many practical applications, and it is thus desirable to replace it with a weaker requirement. When the requirement of a crisp partition of **X** is replaced with a weaker requirement of a fuzzy partition or a fuzzy pseudopartition on **X**, we refer to the emerging problem area as fuzzy clustering. Fuzzy pseudopartitions are often called fuzzy c partitions, where c designates the number of fuzzy classes in the partition. There are two basic methods of fuzzy clustering. One of them, which is based on fuzzy c-partitions, is called a fuzzy c-means (FCM) clustering method. However, the fuzzy c-means method requires that the desired number of clusters be specified. This is a disadvantage whenever the clustering problem does not specify any desired number of clusters. In such problems, the number of clusters should reflect, in a natural way, the structure of given data. The other method, based on fuzzy equivalence relations, is called a fuzzy equivalence relation-based hierarchical clustering method, works in this way.

2.1 Cosine Amplitude

Given a set of data array **X**,

$$\mathbf{X} = \{x_1, x_2, \ldots, x_n\}$$

where each of the elements x_i in the data array **X** is a vector of length m, that is,

$$x_i = \begin{bmatrix} x_{i_1} & x_{i_2} & \ldots & x_{i_m} \end{bmatrix}^T$$

Fuzzy relations map elements from Cartesian product of 2 universes to [0,1] where the strength of the relation is expressed by the membership function of the relation. Let **R** defined on $X \times X$ be a binary fuzzy relation from single universe X to the same universe X. Each element of a relation $\boldsymbol{R}$, r_{ij}, results from a pairwise comparison of 2 data samples, say x_i and x_j, where the strength of the relationship between data samples x_i and x_j is given by the membership value expressing that strength, that is $r_{ij} = \mu_{\boldsymbol{R}}(x_i, x_j)$. This relation matrix $\boldsymbol{R}$ will be of size n x n and relates the similarity of the data samples.

The cosine amplitude method is a useful similarity method to calculate and ensures that the resultant relation $\boldsymbol{R}$ will be symmetric and reflexive and is in fact a tolerance relation (see Section 2.2). It calculates r_{ij} in the following manner:

$$r_{ij} = \frac{\left| \sum_{k=1}^{m} x_{ik} x_{jk} \right|}{\sqrt{\left(\sum_{k=1}^{m} x_{ik}^2 \right) \left(\sum_{k=1}^{m} x_{jk}^2 \right)}} \quad , \qquad \text{where } i, j = 1,\ 2,\ \ldots,\ n \tag{2.1}$$

Close inspection of Equation (2.1) reveals that this method is related to the dot product for the cosine function. When 2 vectors are collinear (most similar), their dot product is unity; when the 2 vectors are at right angles to one another (most dissimilar), their dot product is zero.

2.2 Fuzzy Equivalence Relation

R is a **fuzzy equivalence relation** (also known as similarity relation) if **all** of the following 3 properties are satisfied:

(a) Reflexive : $\mu_R(x_i, x_i) = 1$ for all $x_i \in X$.

(b) Symmetric : $\mu_R(x_i, x_j) = \mu_R(x_j, x_i)$ for all $x_i, x_j \in X$.

(c) Transitive : $\mu_R(x_i, x_j) = \lambda_1$ and $\mu_R(x_i, x_j) = \lambda_2 \rightarrow \mu_R(x_i, x_k) \geq \min(\lambda_1, \lambda_2)$ for all $x_i, x_j, x_k \in X$.

A **fuzzy tolerance relation**, R_1 defined on $X \times X$, has properties of reflexivity and symmetry. R_1 can be transformed into a fuzzy equivalence relation R by ***at most*** (n-1) compositions where n is the cardinality of X. That is

$$R_1^m = R_1 \circ \overset{m-2}{\cdots} \circ R_1 = R \text{ where } m \le n-1$$

Example 2.1: The following fuzzy tolerance relation R_1

$$R_1 = \begin{bmatrix} 1 & 0.8 & 0.4 & 0.5 & 0.2 \\ 0.8 & 1 & 0.4 & 0.5 & 0.9 \\ 0.4 & 0.4 & 1 & 0.4 & 0.4 \\ 0.5 & 0.5 & 0.4 & 1 & 0.5 \\ 0.2 & 0.9 & 0.4 & 0.5 & 1 \end{bmatrix}$$

is easily seen to be both reflexive and symmetric. However, it is not transitive, for example,

$\mu_R(x_1, x_2) = 0.8,\ \mu_R(x_2, x_5) = 0.9$ but $\mu_R(x_1, x_5) = 0.2 < \min(0.8, 0.9)$.

A single composition operation results in

$$R_1^2 = R_1 \circ R_1 = \begin{bmatrix} 1 & 0.8 & 0.4 & 0.5 & 0.8 \\ 0.8 & 1 & 0.4 & 0.5 & 0.9 \\ 0.4 & 0.4 & 1 & 0.4 & 0.4 \\ 0.5 & 0.5 & 0.4 & 1 & 0.5 \\ 0.8 & 0.9 & 0.4 & 0.5 & 1 \end{bmatrix} = R \quad ,$$

is transitive since $\mu_R(x_i, x_j) = \lambda_1$ and $\mu_R(x_i, x_j) = \lambda_2 \rightarrow \mu_R(x_i, x_k) \ge \min(\lambda_1, \lambda_2)$ for all $x_i, x_j, x_k \in X$.
Furthermore,

$$R_1^3 = R_1 \circ R_1^2 = \begin{bmatrix} 1 & 0.8 & 0.4 & 0.5 & 0.8 \\ 0.8 & 1 & 0.4 & 0.5 & 0.9 \\ 0.4 & 0.4 & 1 & 0.4 & 0.4 \\ 0.5 & 0.5 & 0.4 & 1 & 0.5 \\ 0.8 & 0.9 & 0.4 & 0.5 & 1 \end{bmatrix} = R_1^2 = R \quad .$$

Hence, a simple algorithm to transform a fuzzy tolerance relation R_1 into a fuzzy equivalence relation R is as follows:

Algorithm 2.1

1. $R' = R_1 \circ R_1$
2. If $R' \neq R_1$, assign $R_1 = R'$ and goto Step 1.
3. Else Stop: $R = R'$

2.3 α-Compatibility Class

It can be shown that if R is a fuzzy equivalence relation (also known as similarity relation) then each α-cut R_α is a crisp equivalence relation. Effectively, then, we may use any fuzzy equivalence relation R and by taking any α-

cut R_α for any value $\alpha \in (0,1]$, create a crisp equivalence relation that represents the presence of similarity between the elements to the degree α. Each of these equivalence relations forms a partition of X.

Example 2.2: Let us illustrate this clustering approach with $X = \{x_1, x_2, x_3, x_4, x_5\}$ and using α-cut R_α on the fuzzy equivalence relation R in Example 2.1.

$$R = \begin{bmatrix} 1 & 0.8 & 0.4 & 0.5 & 0.8 \\ 0.8 & 1 & 0.4 & 0.5 & 0.9 \\ 0.4 & 0.4 & 1 & 0.4 & 0.4 \\ 0.5 & 0.5 & 0.4 & 1 & 0.5 \\ 0.8 & 0.9 & 0.4 & 0.5 & 1 \end{bmatrix}$$

By taking α-cut R_α of fuzzy equivalence relation R at values of α = 1, 09, 0.8, 0.5 and 0.4, we get the following:

$$R_1 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad R_{0.9} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \end{bmatrix}, \quad R_{0.8} = \begin{bmatrix} 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 \end{bmatrix},$$

$$R_{0.5} = \begin{bmatrix} 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 \end{bmatrix}, \quad R_{0.4} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

We can see that the clustering of the five data points according to the α level is as shown in Table 2.1

α-cut level	Classification Results
1.0	$\{x_1\},\{x_2\},\{x_3\},\{x_4\},\{x_5\}$
0.9	$\{x_1\},\{x_2, x_5\},\{x_3\},\{x_4\}$
0.8	$\{x_1, x_2, x_5\},\{x_3\},\{x_4\}$
0.5	$\{x_1, x_2, x_4, x_5\},\{x_3\}$
0.4	$\{x_1, x_2, x_3, x_4, x_5\}$

Table 2.1 Classification of 5 data points according to α-cut level

We can also express the classification results described in Table 2.1 with a tree classification diagram as shown in Figure 2.1. In the figure, it can be observed that the higher the value of α, the finer is the classification. As α gets larger, the results of fuzzy classification approach the trivial case when each data point is assigned to its own class.

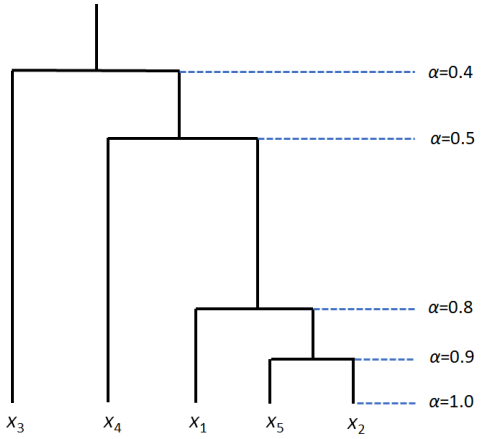

Figure 2.1 Classification Tree

2.4 Problem. 16 districts in a country have suffered flooding from a recent thunderstorm. The flooding in each district are characterized according to three flooding levels : mild (<0.1 m), moderate (0.1 to 0.2 m), severe (0.2 to 0.3 m), catastrophic (> 0.3 m). The percentage of areas within a given district with each of the flooding levels is given in Table 2.2.

Districts	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}	x_{13}	x_{14}	x_{15}	x_{16}
x_{i1} - Ratio of Mild	0.1	0.0	0.2	0.8	0.3	0.0	0.5	0.6	0.0	0.1	0.3	0.1	0.2	0.2	0.1	0.2
x_{i2} - Ratio of Moderate	0.7	0.5	0.2	0.1	0.0	0.4	0.0	0.3	0.5	0.6	0.2	0.5	0.0	0.6	0.7	0.4
x_{i3} - Ratio of Severe	0.2	0.5	0.2	0.0	0.4	0.0	0.4	0.0	0.1	0.0	0.1	0.4	0.2	0.1	0.1	0.2
x_{i4} - Ratio of Catastrophic	0.0	0.0	0.4	0.1	0.3	0.6	0.1	0.1	0.4	0.3	0.4	0.0	0.6	0.1	0.1	0.2

Table 2.2 Flooding Levels in 16 Districts

(a) Use the cosine method in Equation (2.1) of Section 2.1 to generate the fuzzy tolerance relation R_1.
(b) Use Algorithm 2.1 in Section 2.2 to transform R_1 into a fuzzy equivalence relation R.
(c) Generate the α-cut R_α in Section 2.3 to provide classification classes for α-cut level = 0.4 and α-cut level = 0.8?
(d) If we want to classify the flooding for the whole county into 3 classes - **Red**, **Yellow** **and** **Green**. What should be the appropriate the α-cut value?