THE UNIVERSITY OF WESTERN ONTARIO
DEPARTMENT OF CIVIL AND
ENVIRONMENTAL ENGINEERING

Water Resources Research Report

Application of the Fuzzy Performance Indices to the City of London Water Supply System

By:
Ibrahim El-Baroudy
and
Slobodan P. Simonovic

Report No: 050
Date: January 2005

ISSN: (print) 1913-3200; (online) 1913-3219;
ISBN: (print) 978-0-7714-2630-8; (online) 978-0-7714-2631-5;

APPLICATION OF THE FUZZY PERFORMANCE INDICES TO THE CITY OF LONDON WATER SUPPLY SYSTEM

BY

Ibrahim El-Baroudy, Ph.D. Candidate
Department of Civil and Environmental Engineering
University of Western Ontario, London, Ontario

AND

Slobodan S. Simonovic, Professor and Research Chair
Department of Civil and Environmental Engineering
Institute for Catastrophic Loss Reduction
University of Western Ontario, London, Ontario

January, 2005

TABLE OF CONTENT

LIST OF FIGURES

LIST OF TABELS

1 INTRODUCTION

1.1 Objectives of the analysis

This study explores the utility of fuzzy performance indices: (i) combined reliability-vulnerability index, (ii) robustness index, and (iii) resiliency index, for evaluating the performance of a complex water supply system. Regional water supply system for the City of London is used as the case study. The two main components being investigated in this case study are; (i) the Lake Huron Primary Water Supply System (LHPWSS), and (ii) the Elgin Area Primary Water Supply system (EAPWSS).

Computational requirements for the implementation of the fuzzy performance indices are investigated together with the sensitivity of these criteria to different shapes of fuzzy membership functions.

1.2 Report organization

Chapter 2 briefly introduces the Lake Huron Primary Water Supply System (LHPWSS), and the Elgin Area Primary Water Supply system (EAPWSS). Chapter 3 presents the methodology used for the analysis of both systems. The chapter starts by describing the procedure for system representation. The description of the method used to construct membership functions for different system components follows. The calculation process of the fuzzy performance indices is presented in details at the end.

Chapters 4 and 5 present the fuzzy performance indices for LHPWSS and EAPWSS systems, respectively. In both chapters, the sensitivity of fuzzy indices to the different shapes of fuzzy membership functions is explored first. The utility of these measures in identifying critical system components is demonstrated afterwards. Finally, the conclusions of the analysis performed in the previous two chapters are presented in Chapter 6.

1.3 Summary of the results

The analysis of the results revealed that LHPWSS system is reliable and not too vulnerable to disruption in service. On the contrary, EAPWSS system is found to be highly unreliable and vulnerable to disruption in service. The results show that LHPWSS system is more robust than EAPWSS system, and therefore LHPWSS system can accommodate possible change in requirement conditions.

Combined reliability-vulnerability index and robustness index are sensitive to change in the shape of the membership function. The value of the resiliency index does not depend on the shape of membership function.

The fuzzy performance indices are capable of identifying weak system components that require attention in order to achieve future improvement in system performance.

2 SYSTEM DESCRIPTION

The City of London regional water supply system consists of two main components; (i) the Lake Huron Primary Water Supply System (LHPWSS), and (ii) the Elgin Area Primary Water Supply system (EAPWSS). The LHPWSS system obtains raw water from the Lake Huron. Water is treated and pumped from the lake to the terminal reservoir in Arva, as shown in Figure 1. Water from the Arva reservoir is pumped to the north of the City of London where it enters the municipal distribution system. The system provides water for the City of London as well as a number of smaller neighboring municipalities (through a secondary system).

The EAPWSS system treats raw water from the Lake Erie and pumps the treated water to the terminal reservoir located in St. Thomas. Water from the reservoir is pumped to the south of the City of London where it enters the municipal distribution system, as shown in Figure 1. In the case of emergency, the City of London can obtain additional water from a number of wells located inside the City and in the surrounding areas.

2.1 Lake Huron primary water supply system (LHPWSS)

The Lake Huron treatment facility has a treatment capacity of about 336 million liters per day (336,400 m^3/day). The plant's individual components are designed with a 35% overload capacity resulting in the maximum capacity of 454,600 m^3/day. The current daily production, based on the annual average, is 157,000 m^3/day with a maximum production value of 264,000 m^3/day in 2001.

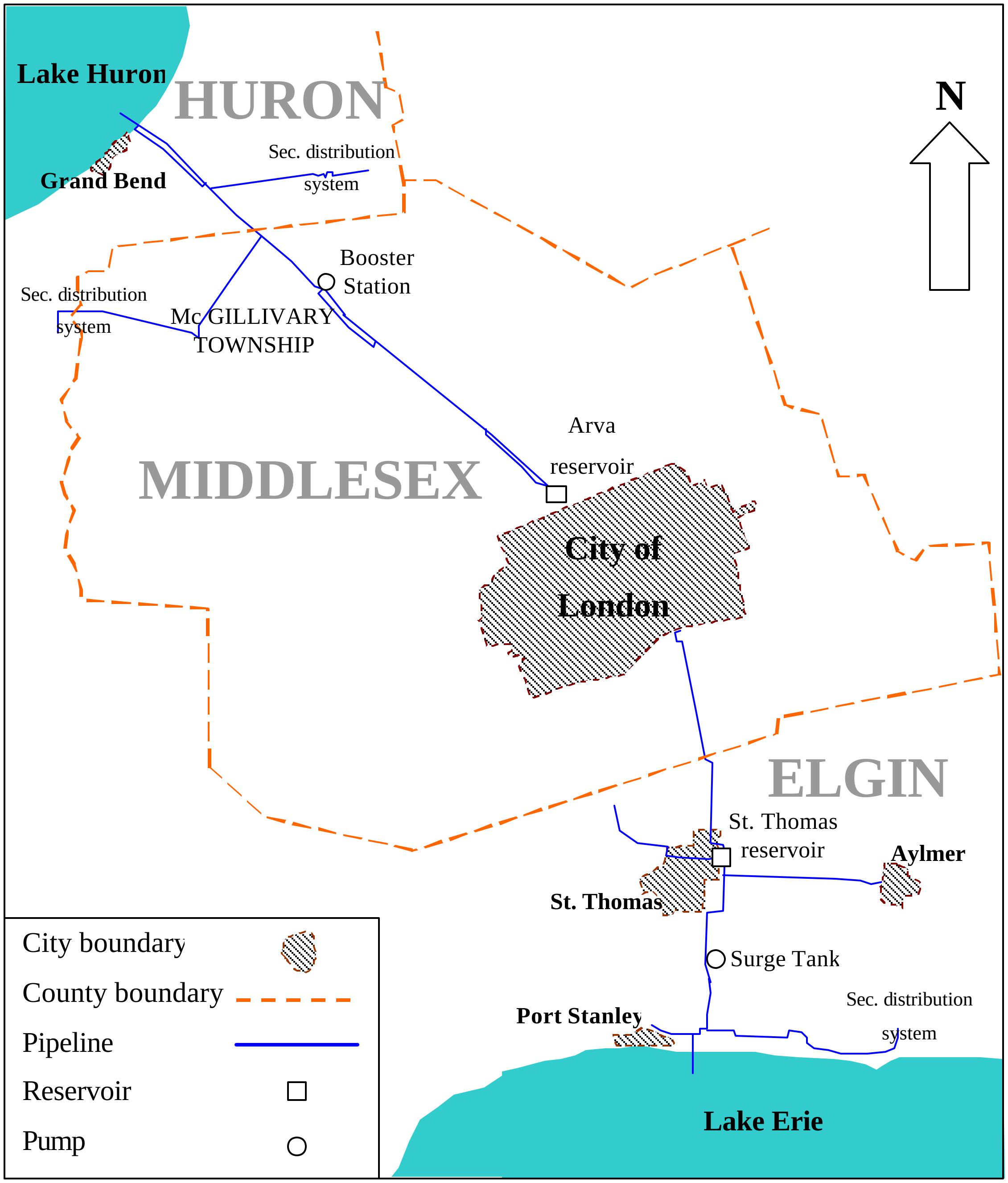

Figure (1): The City of London regional water supply system.

The water treatment system employs conventional and chemically assisted flocculation and sedimentation systems, dual-media filtration, and chlorination as the primary disinfection. Both, the treatment system and the water quality are continuously monitored using computerized Supervisor Control and Data Acquisition (SCADA) system.

A brief description of the system's works, from the intake through the treatment plant to the terminal reservoir at Arva is provided in the following section. A schematic representation of the system is depicted in Figure 2.

2.1.1 Intake system

Raw water flows by gravity from Lake Huron through a reinforced concrete intake pipe to the low lift pumping station. The intake pipe discharges raw water through mechanically cleaned screens into the pump-well of the low lift pumping station. The intake crib and the intake pipe are designed for the maximum capacity of 454,600 m^3/day. Chlorine can be injected in the intake crib through the screens or to the low lift pumping station for zebra mussel control (pre-chlorination). The low lift pumping station is located on the shore of Lake Huron at the treatment plant site. The low lift pumping station consists of six pumps with rated capacity between 115,000 and 100,000 m^3/day.

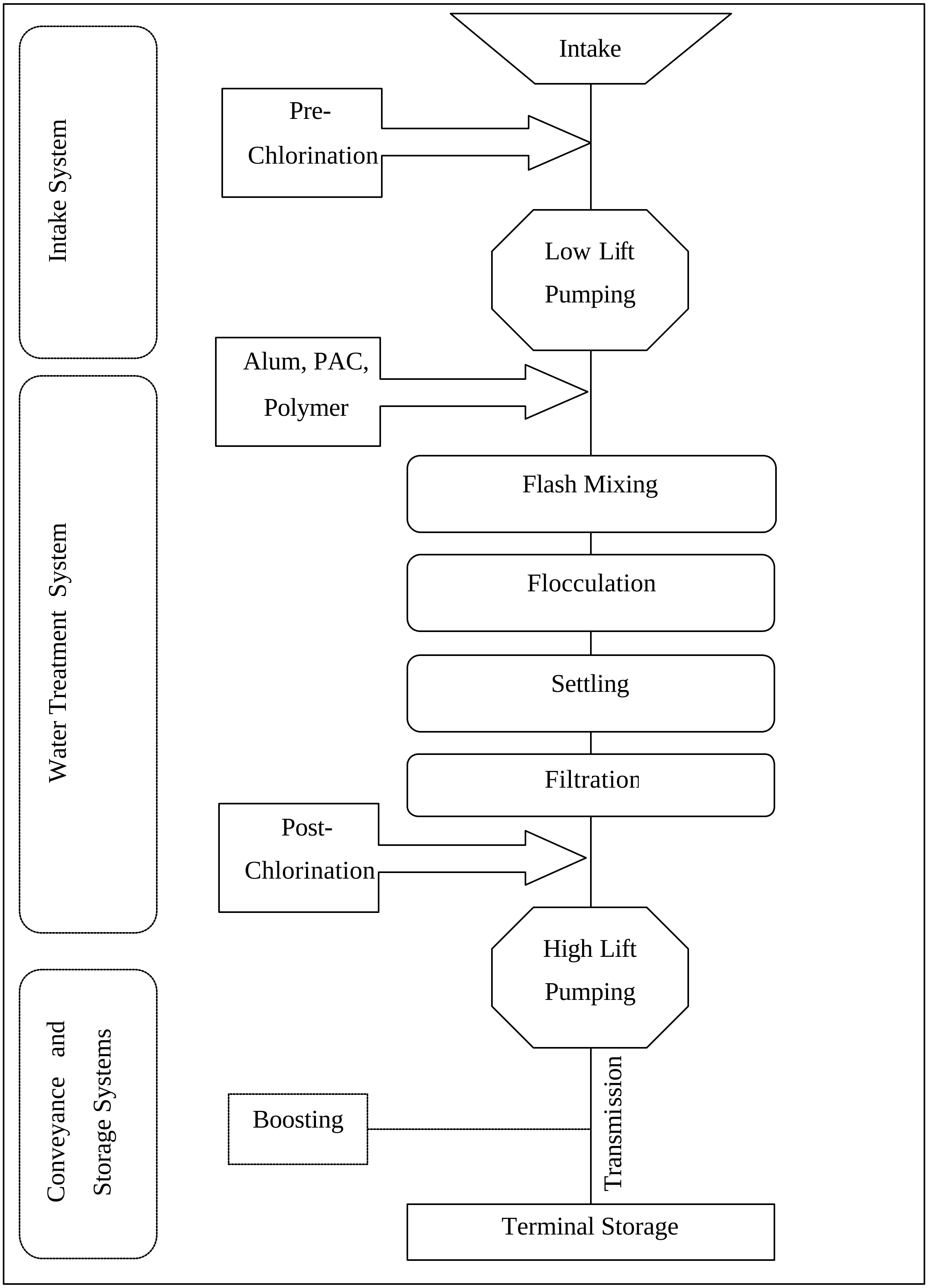

Figure (2) Schematic representation of the LHPWSS.

2.1.2 Water treatment system

Water from the low lift pumping station is discharged into the treatment plant where it bifurcates into two parallel streams designated as the North and the South. Two flash mix chambers, one in each stream, consist of two cells and one mixer per cell. The water flows by gravity from the flash mix chambers to the flocculation tanks.

In the first treatment step, which takes place in the flash mix chambers, Alum is added (for coagulation) together with Powdered Activated Carbon (PAC) (seasonally added for taste and odor control) and Polymer (as coagulant aid). Chlorine, which is used for disinfection, is added upstream of the flash mixers.

Mechanical flocculation process takes place in both, North and South treatment lines. Each flocculation tank is divided into two zones, primary and secondary, with the capacity ranging between 32,000 m^3/day and 170,000 m^3/day. Water flows through the two zones where walking beams (or paddle mixers) perform the mixing, to the darifiers/settlers. Water flows into the settlers from one end, flows up through the parallel plate clarifiers and is discharged at the opposite end. A scraper, at the bottom of the tank, thickens the settled solids and moves them to the central hopper.

Waste sludge pumps transfer settled solids to the solid bowl centrifuges for dewatering. The solid wastes are stored into a container for off-site disposal while the concentrate is returned to the lake through the main plant drain.

Twelve high rate gravity filters perform the removal of particulate matter from water flowing from the clarifiers. Water flows to any of the twelve filters from both treatment lines. Filtered water is then discharged into the three clear-wells where Chlorine is added for post-chlorination.

2.1.3 Conveyance and storage systems

Finished water is pumped from the clear-wells through the transmission main to the terminal reservoir at Arva by the high lift pumps. The high lift pumping station consists of five high lift pumps rated at 1,158 L/s. Water flows through the primary transmission main, a 1220 mm diameter concrete pipe, under pressure for about 47 km. A total of 21 km of the primary transmission main is twined to maintain the capacity and increase the redundancy in case of emergency. The primary transmission main is surge-protected during power failure or transit pressure conditions (due to cycling of the high lift pumps). The terminal reservoir at Arva consists of four individual cells, each of 27,000m^3 storage capacity.

An intermediate reservoir and booster station are constructed in the McGillivary township. The intermediate reservoir serves the users in the McGillivary township. Water from the reservoir can be withdrawn back into the primary transmission main during the high demand periods, by four high lift pumps at the booster station.

2.2 Elgin area primary water supply system (EAPWSS)

The Elgin water treatment facility was constructed in 1969 to supply water from the Lake Erie to the City of London, St. Thomas and a number of smaller municipalities. In 1994, the facility has been expanded to double its throughput to its current 91,000m^3/day capacity. A series of upgrades took place from 1994 to 2003 to add surge protection and introduce fluoridation treatment. The design capacity of the treatment facility is 91,000 m^3/day, with an average daily flow of 52,350 m^3/day, which serves about 94,400 persons.

The water treatment in EAPWSS employs almost the same conventional treatment methods used in LHPWSS. The only exception is that the facility uses the fluoridation treatment system to provide dental cavity control to the users. As in LHPWSS, the treatment system and water quality are continuously monitored using computerized Supervisor Control and Data Acquisition (SCADA) system. The finished treated water is pumped to the terminal reservoir located in St. Thomas. A short description of the EAPWSS is given in the following section. A schematic of the system is shown in Figure 3.

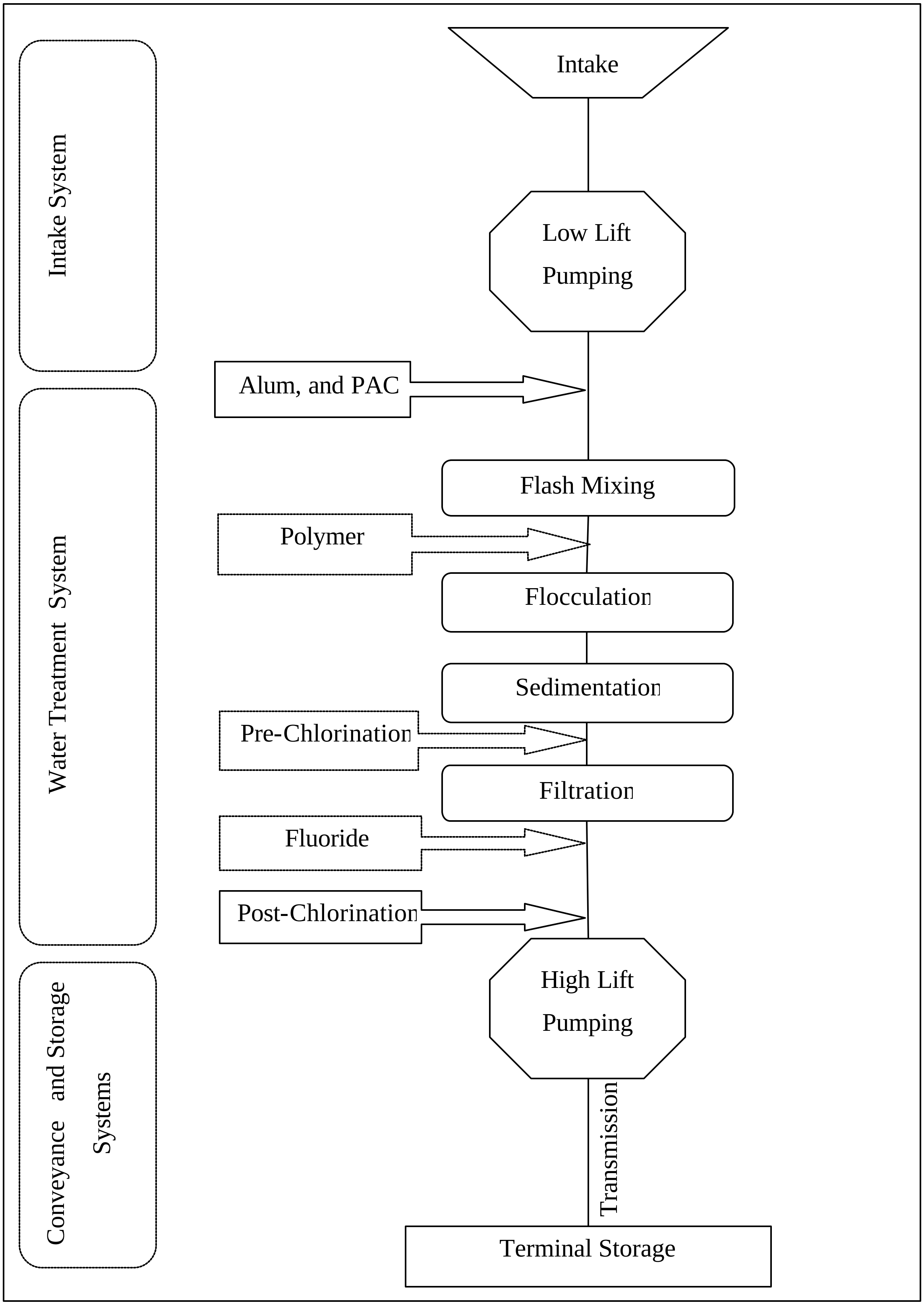

Figure (3) Schematic representation of the EAPWSS.

2.2.1 Intake system

Raw water, drawn from the Lake Erie, is pumped through a 1500 mm diameter intake conduit to the low lift pumping station at the shore of the lake. The ultimate capacity of the intake conduit is 182,000 m^3/day; in case of an emergency the plant drain serves as an alternative intake, with almost the same maximum capacity. The low lift pumping station houses two clear-wells. Each well has two independent vertical turbine pumps that discharge into a 750 mm transmission main to the water treatment plant.

2.2.2 Water treatment system

The raw water discharged from the low lift pumping station is metered and split evenly into two parallel streams, as in the LHPWSS. The split continues from the head-works to the filtration process. The first treatment process is the flash mixing where Alum is added as a coagulation agent together with PAC. There is one flash mixing chamber with two cells and one mixer per cell in each treatment line. Water flows by gravity from the flash mix chamber to the flocculation tanks.

The flocculation system consists of two banks, North and South, of flocculation tanks, each with a capacity of 91,000 m^3/day. Each bank has two tanks that make a total of eight flocculation tanks. Polymer can be added at any point in the series of flocculation tanks. Water flows directly from the flocculation tanks into the sedimentation system. There is one gravity sedimentation tank in each process stream. Pre-chlorination takes place after the sedimentation process and before the filtration.

Finally, the particulate matter is removed using four gravity filters during the filtration process. The treatment is no longer split into two parallel streams as the water can be directed to any of the four filters. The filtered water is collected in the filtered water conduit underlying the filters and flows into a clear well and the on-site reservoir. Post-chlorination takes place in the conduit leading from the on-site reservoir to the high lift pumping station.

2.2.3 Conveyance and storage systems

The high lift pumping station delivers finished water through the transmission main to the terminal reservoir in St. Thomas. It also delivers water to the secondary distribution system. The high lift pumping station houses four high lift pumps, each with a rated capacity of 52,000 m^3/day. The treated water is discharged through the primary transmission main (14 km long 750 mm diameter concrete pressure pipe).

The surge facility was constructed in 1994 to protect the transmission main from damage due to the system transit pressure conditions during cycling of the high lift pumps. Through the valve chamber, upstream of the terminal reservoir, water from the transmission main is directed to one, or both, reservoirs at the Elgin-Middlesex facility. Both reservoirs have equal capacity of 27,300 m^3 and store water supply for Aylmer, St. Thomas and the Elgin-Middlesex (serving London) pumping system. Water can by-pass the reservoirs and flow directly to each of the secondary pumping stations.

3 METHODOLOGY FOR SYSTEM RELAIBILITY ANALYSIS

3.1 Multi-component system representation

Water supply system is a typical example of a multi-component system that includes a collection of conveyance, treatment, and storage components. These components are at risk of failure due to a wide range of causes. In the same time, these elements are connected in complicated networks that affect the overall performance of the water supply system.

The key step in the evaluation of system performance is the appropriate representation of different relationships between system components. This representation should reflect the effect of the performance of each component on the overall system performance. For example, the chemical treatment of raw water in a water supply system depends on adding different chemicals at certain locations in the treatment process. This process requires the availability of chemicals in the storage facility and the ability to transfer them to the required location on time. Storage and conveyance facilities, responsible for delivering these chemicals to the mixing chambers, are not part of the raw water path. The failure of these facilities directly affects the water treatment process and might cause a total failure of the water treatment system. As a result, it is important to consider these facilities when performing a system reliability analysis.

Figure 4 shows the layout of one part of the water treatment plant, where the stored chemicals are conveyed to the mixing location via the feed pump. It is evident that taking these components into consideration in the system reliability analysis is difficult becuase of the need to identify the functional relationships between them and the other system components. Similar relationships are required for all non-carrying water components. If these components are not taken into consideration the chance of improper estimation of system reliability may increase.

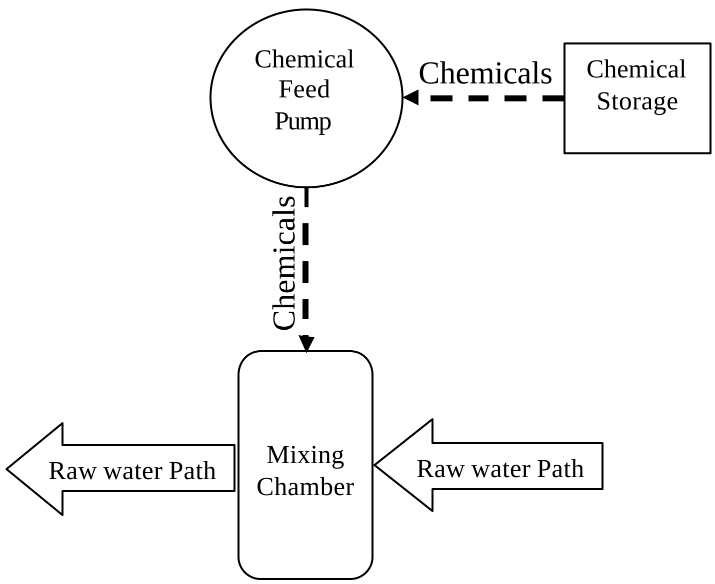

Figure (4) Water supply system layout.

Representing a multi-component system as a system of components having different failure relationships can be used as an effective mean to integrate water-carrying and non-water carrying components into one system. For example, any two components are considered serially connected if the failure of one component leads to the failure of the other. Two components are considered to have a parallel connection if the failure of one component does not lead to the failure of the other. A clear identification of the failure relationship between different components facilitates the calculation of the performance indices. Figure

5 shows the integrated layout for the previous example. In this figure, the system representation integrates components carrying chemicals into the path of raw water.

Calculation of the system's performance indices based on the integrated layout will be fairly difficult as there is no clear link between the failure of the components carrying chemicals and the components carrying raw water. Note that operational components having redundancy are treated as components with parallel connection. This reflects the fact that redundant elements reduce the possibility of system failure.

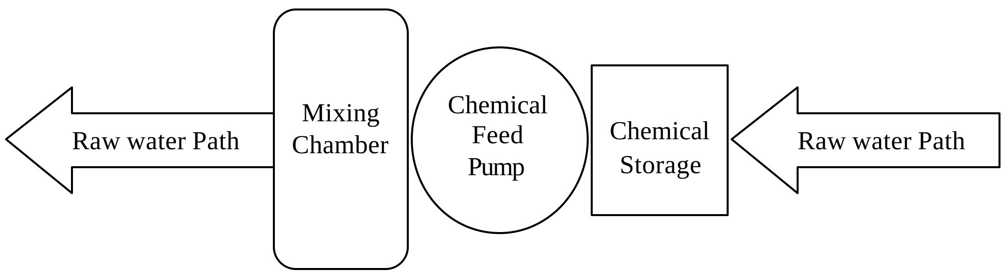

Figure (5) System integrated layout for the reliability analysis calculation.

3.2 Capacity and requirement of system components

System reliability analysis uses load and resistance as the fundamental concepts to define the risk of system failure, (Simonovic, 1997). These two concepts are used in structural engineering to reflect the characteristic behavior of the system under external loading conditions. In water supply systems, load and resistance are replaced by requirement and capacity, respectively, to reflect the specific domain variables of the water supply system. Hence, system requirement is defined as the variable that reflects different water demand requirements that may be imposed over the useful life of the system (Ang and Tang, 1984). System capacity, on the other hand, is defined as the system characteristic variable which describes the capacity of the system to satisfy demand requirements.

The fuzzy reliability analysis uses membership functions (MFs) to express uncertainty in both capacity and requirement of each system component. The general representation of membership function is:

$$\tilde{X} = \{(x, \mu_{\tilde{X}}(x)) : x \in R;\ \mu_{\tilde{X}}(x) \in [0,1]\} \qquad \text{……….(1)}$$

where:

$\tilde{X}$ is the fuzzy membership function;

$\mu_{\tilde{X}}(x)$ is the membership value of element *x* to $\tilde{X}$; and

R is the set of real numbers.

Membership functions are usually defined by their α-cuts. The α-cut is the ordinary set of all the elements belonging to the fuzzy set whose value of membership is a or higher, that is:

$$X(a) = \{x : \mu_{\tilde{X}}(x) \geq a;\ x \in R;\ a \in [0,1]\} \qquad \text{……….(2)}$$

where

$X(a)$ is the ordinary set at the a-cut; and a is the membership value.

Another characteristic property of the fuzzy membership function is its support. The support of the fuzzy membership function can be defined as the ordinary set that is:

$$S(\tilde{X})=\tilde{X}(0)=\{x:\mu_{\tilde{X}}(x)>0\} \qquad \ldots\ldots\ldots.(3)$$

where

$S(\tilde{X})$ is the ordinary set at the a-cut=0.

The fuzzy membership function support is the 0-cut set and includes all the elements with the membership value higher than 0, as shown in Figure 6. Construction of membership function is based on the system design data and choice of the suitable shape. There are many shapes of membership functions. However, the application context dictates the choice of the suitable shape. For the problem domain addressed in this study, system components have maximum and minimum capacity that cannot be exceeded. Therefore, any candidate membership function shape should have two extreme bounds with zero membership values. Triangular and trapezoidal shapes are the simplest MF shapes that meet this requirement.

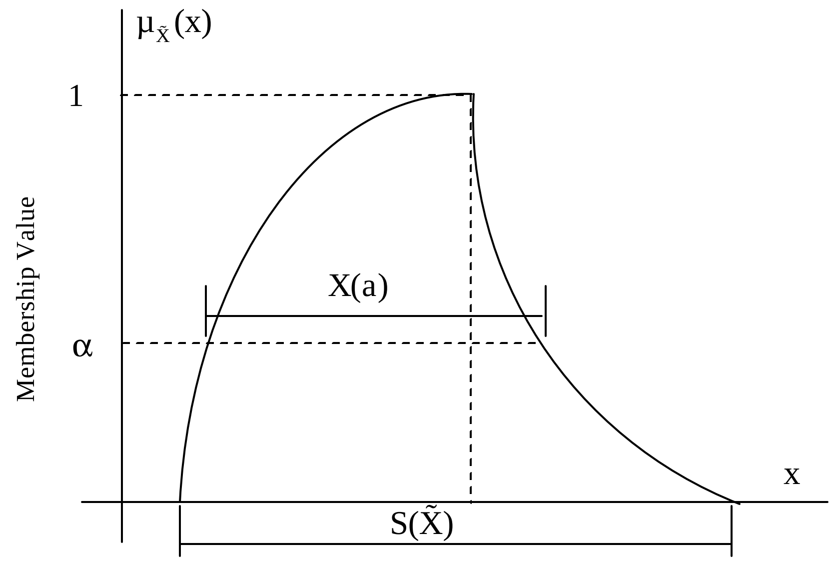

Figure (6) Support and a-cut of the fuzzy membership function (after Ganoulis, 1994).

In the presented case study, the following reports are used as the source of data for determining capacity and requirement for each component:

- Earth Tech Canada Inc.,2000;
- Earth Tech Canada Inc.,2001;
- American Water Services Canada-AWSC, 2003a;
- American Water Services Canada-AWSC, 2003b; and
- DeSousa and Simonovic, 2003.

Some problems are experienced with the available data. First, many components have single design capacity that creates a problem in the development of a membership function. The second problem is the use of different units for capacity of different components. For instance, capacity of storage facilities is expressed in volumetric units, cubic meters (m^3). Capacity of pumps is measured using flow units, cubic meter per day (m^3/day). Thus, their direct comparison may not be possible. The third problem is the identification of the requirement for each system component. Most of the available information corresponds to the system requirement (i.e., the requirement of the chlorination system not the capacity of individual chlorinator).

3.2.1 System component capacity membership function

A triangular membership function, representing the capacity of a system component, is constructed using three design values (i.e., the minimum, modal, and the maximum value). In many cases only one value is available. For example in the case of reservoirs, only the maximum capacity is available. If there is no other source of information, the minimum capacity is set to zero. The modal value can be subjectively selected within the range from minimum to maximum capacity. In case of trapezoidal membership function, two modal points are subjectively selected.

In cases when the components are designed with an overload capacity (i.e. maximum design capacity higher than the rated capacity) this value is used to build the membership function. Figure 7 depicts a component with a maximum capacity of *a* units with *c* (%) overload capacity. In case (I), a triangular membership function is defined as follows:

$$\mu_{\tilde{A}}(x) = \begin{cases} 0, & \text{if } x \le (1-2c)a \\ \dfrac{x-(1-2c)a}{(1-c)a-(1-2c)a}, & \text{if } x \in [(1-2c)a,(1-c)a] \\ \dfrac{a-x}{a-(1-c)a}, & \text{if } x \in [(1-c)a,a] \\ 0, & \text{if } x \ge a \end{cases} \qquad \ldots\ldots\ldots.(4)$$

where

(1-*c*)*a* is the modal value; and

(1-2*c*)*a* and *a* are the lower and upper bounds of the membership function.

In case (II), a trapezoidal membership function is defined as follows:

$$\mu_{\tilde{A}}(x)=\begin{cases} 0, & \text{if } x \le (1-2c)a \\ \dfrac{x-(1-2c)a}{(1-1.5c)a-(1-2c)a}, & \text{if } x \in [(1-2c)a,(1-1.5c)a] \\ 1, & \text{if } x \in [(1-1.5c)a,(1-0.5c)a] \\ \dfrac{a-x}{a-(1-0.5c)a}, & \text{if } x \in [(1-0.5c)a,a] \\ 0, & \text{if } x \ge a \end{cases} \qquad \text{……….(5)}$$

where

(1-1.5*c*)*a* and (1-0.5*c*)*a* are the modal values; and

(1-2*c*)*a* and a are the lower and upper bounds of the membership function.

The modal values in case (II) (i.e. trapezoidal membership function) equally divide the distance from the modal value (in the triangular membership function) to the lower and upper bounds, respectively. In both cases the maximum value corresponds to the design capacity.

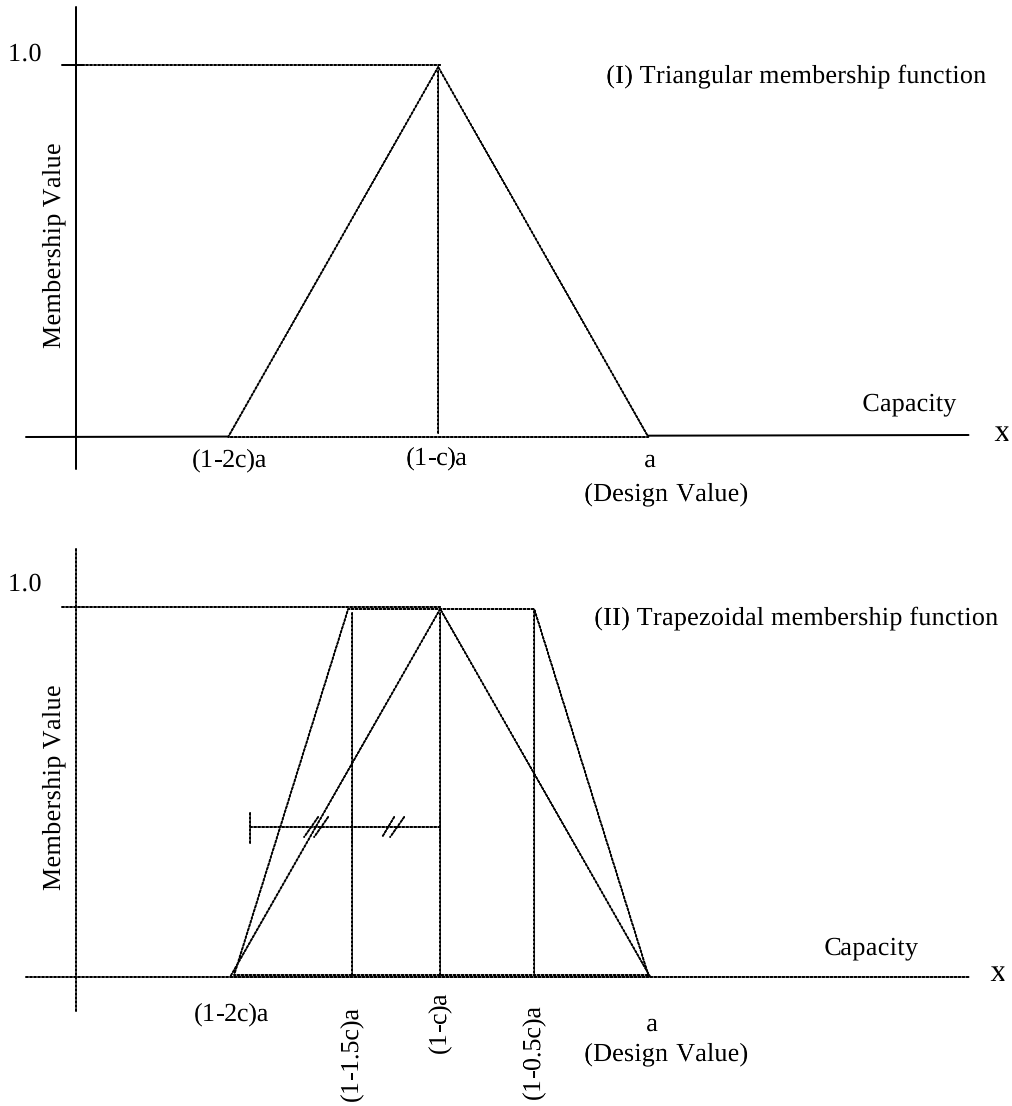

Figure (7) Membership function development using design capacity and overload capacity.

3.2.2 System component requirement membership function

The requirement membership function of a group of components performing the same function is based on the assumption of equal role for every unit. For example, if a collection of four chlorinators supply Y [kg/day] of Chlorine, the maximum supply requirement of each chlorinator is (Y/4) [kg/day]. The yearly average and minimum requirements are used to develop the requirement membership function of each component, as shown in Figure 8.

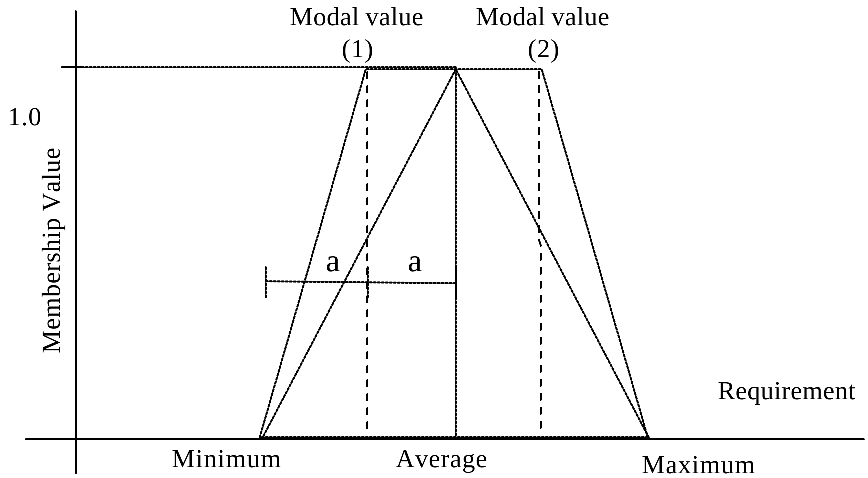

Figure (8) Supply requirement membership function.

The two modal values of the trapezoidal membership function in Figure 8 are the middle points between the maximum, or minimum, supply and the average requirement value. In case when yearly average data is not available, the modal value is considered to be the average value of the maximum and minimum supply.

Proxy conversions are used to overcome the problem of using different units for expressing capacity and requirement. For example, the supply requirement of certain chemical is usually expressed in kilograms (kg) while the storage facility capacity is expressed in cubic meters (m^3). In this case, the corresponding chemical bulk density is used to convert the supply requirement using volumetric units.

3.2.3 Standardization of membership functions

In the process of calculating system fuzzy reliability indices, membership functions of system components are aggregated using fuzzy operators. Therefore, all membership functions must be expressed in the same units. This can be achieved only through standardization of the membership functions (i.e., division by the unit maximum capacity value).

The membership function of each system component will have a maximum value of one. For example, a triangular membership function representing a reservoir capacity (m^3) is defined as follows:

$$\mu_{\tilde{A}}(x) = \begin{cases} 0, & \text{if } x \le a \\ \dfrac{x-a}{m-a}, & \text{if } x \in [a,m] \\ \dfrac{b-x}{b-m}, & \text{if } x \in [m,b] \\ 0, & \text{if } x \ge b \end{cases} \qquad \text{……….(6)}$$

where

m is the modal value; and

a and b are the lower and upper bounds of the non-zero values of the membership.

This membership function is standardized to the following (dimensionless) membership function:

$$\mu_{\tilde{A}}(x)=\begin{cases} 0, & \text{if } x \le (a/b) \\ \dfrac{x-(a/b)}{(m/b)-(a/b)}, & \text{if } x \in [(a/b),(m/b)] \\ \dfrac{1-x}{1-(m/b)}, & \text{if } x \in [(m/b),1] \\ 0, & \text{if } x \ge 1 \end{cases} \qquad \text{……….(7)}$$

where

(m/b) is the modal value; and

(a/b) and 1 are the lower and upper bounds of the non-zero values of the membership.

The capacity and requirement membership functions are processed together as one membership function representing the component-state membership function. The same standardization method is applied to the requirement membership functions. The membership function values are divided by the maximum capacity of a system component.

3.3 Calculation of fuzzy performance indices

The membership functions representing system-state and acceptable levels of performance are used in the calculation of the fuzzy reliability-vulnerability and robustness indices.

3.3.1 System-state membership function

Multi-component systems have several component-state membership functions describing each component of the system. Aggregation of these membership functions results in the system-state membership function for the whole-system (El-Baroudy and Simonovic, 2003 and 2004).

First, all parallel and redundant components are aggregated into a number of serially connected components. For a group of M parallel (or redundant) components, the m-th component has a component-state membership function $\tilde{S}_m(u)$ defined on the universe of discourse U. All the components states contribute to the whole group system-state membership function. Failure of the group occurs if all components fail. Hence, the system-state is calculated as follows:

$$\tilde{S}(u) = \sum_{m=1}^{M} \tilde{S}_m(u) \qquad \text{……….(8)}$$

where:

$\tilde{S}_m(u)$ is the m-th component-state membership function; and

M is the total number of parallel (or redundant) components.

For the system of N serially connected groups, where the *n*-th group has a state membership function $\tilde{S}_n(u)$, the weakest component controls the whole system-state or causes the failure of the whole system. Therefore, the system-state is calculated as follows:

$$\tilde{S}(u) = \min_{N}\left(\tilde{S}_1, \tilde{S}_2, \ldots\ldots\ldots, \tilde{S}_N\right) \qquad \ldots\ldots\ldots(9)$$

where:

$\tilde{S}(u)$ is the system-state membership function; and

$\left(\tilde{S}_1, \tilde{S}_2, \ldots\ldots\ldots, \tilde{S}_N\right)$ component-state membership functions.

In the present case study, all component-state membership functions are formulated in terms of fuzzy margin of safety using the fuzzy subtraction operator (El-Baroudy and Simonovic, 2003 and 2004).

$$\tilde{M}_i = \tilde{X}_i \ (-)\ \tilde{Y}_i \qquad \forall i = 1,2,\ldots..n \qquad \ldots\ldots\ldots(10)$$

where:

$\tilde{M}_i$ is the fuzzy margin of safety of the i-th component;

$\tilde{X}_i$ is the fuzzy capacity of the i-th component;

$\tilde{Y}_i$ is the fuzzy requirement of the i-th component; and

n is the number of system components.

Capacity and requirement membership functions are stored in the spreadsheet, where all the necessary calculations are performed to obtain the final component-state and component-failure membership functions. Figure 9 shows a part of the spreadsheet for LHPWSS, while Appendix (I) contains the full-length spreadsheet files for both systems under investigation (LHWPSS and EAPWSS). The fuzzy performance indices are then calculated using the calculation script that is developed to perform different calculation steps. Appendix (II) includes the source code of the script files for both LHPWSS and EAPWSS.

3.3.2 Acceptable level of performance membership function

The acceptable level of performance is a fuzzy membership function that is used to reflect the decision-makers ambiguous and imprecise perception of risk, (El-Baroudy and Simonovic, 2003 and 2004). The reliability reflected by the acceptable level of performance is quantified by

$$LR = \frac{x_1 \times x_2}{x_2 - x_1} \qquad \text{……….(11)}$$

where:

LR is the reliability measure of the acceptable level of performance; and

x_1 and x_2 are the bounds of the acceptable failure region, as shown in Figure 10.

The calculation of the fuzzy reliability-vulnerability and fuzzy robustness indices depends on the calculation of the overlap area between the membership functions of both the system-state and the acceptable level of performance.

Microsoft Excel - Case Study-Calculation(LH).xls

	A	B	H	I	J	K	L	M	N
4		System	Units	Capacity			Requirment		
5				Average Daily flow	Design Capacity	Maximum Overload	Yearly Min	Yearly Avg	Yearly Max
6	***Intake Crib***	Intake System	**MLD**	157.3	340.0	454.6	53.0	157.3	255.7
7	***Chlorinator I***		**Kg/d**	360.0	630.0	900.0	24.5	72.0	130.0
8	***RC Intake Pipe***		**MLD**	157.3	340.0	454.6	53.0	157.3	255.7
9	***Traveling Screens***	Low Lifting System	**MLD**	157.3	340.0	454.6	53.0	157.3	255.7
10	***Pumping Wells***		**MLD**	157.3	340.0	454.6	53.0	157.3	255.7
11	***Chlorinator II***		**Kg/d**	360.0	630.0	900.0	24.5	72.0	130.0
12	***Single Speed Pump 1***		**MLD**	49.9	75.0	100.0	16.8	49.0	81.2
13	***Variable Speed Pump***		**MLD**	57.4	86.2	115.0	19.3	49.0	93.4
14	***Single Speed Pump 2***		**MLD**	49.9	75.0	100.0	16.8	49.0	81.2
15	***Single Speed Pump 1 (Back-up***		**MLD**	49.9	75.0	100.0	16.8	49.0	81.2
16	***Variable Speed Pump (Back-u***		**MLD**	57.4	86.2	115.0	19.3	49.0	93.4
17	***Single Speed Pump 2 (Back-up***		**MLD**	49.9	75.0	100.0	16.8	49.0	81.2

	AC	AD	AE	AF	AG	AH	AI	AJ	AK	AL	AM	AN	AO	AP
4	Element-State (Tri-MoS)			Element-State (Trap-MoS)				Element-Failure (Tri)			Element-Failure (Trap)			
5	a	b	c	a	b	c	d	a	b	c	a	b	c	d
6	-0.22	0.40	0.88	-0.22	0.09	0.41	0.88	0.08	0.13	0.17	0.08	0.11	0.14	0.17
7	0.26	0.62	0.97	0.26	0.44	0.74	0.97	0.01	0.01	0.01	0.01	0.01	0.01	0.01
8	-0.22	0.40	0.88	-0.22	0.09	0.41	0.88	0.08	0.13	0.17	0.08	0.11	0.14	0.17
9	-0.22	0.40	0.88	-0.22	0.09	0.41	0.88	0.04	0.08	0.50	0.04	0.19	0.35	0.50
10	-0.22	0.40	0.88	-0.22	0.09	0.41	0.88	0.00	0.50	1.00	0.00	0.33	0.67	1.00
11	0.26	0.62	0.97	0.26	0.44	0.74	0.97	0.01	0.01	0.01	0.01	0.01	0.01	0.01
12	-0.31	0.26	0.83	-0.31	-0.03	0.22	0.83	0.02	2.51	5.00	0.02	1.68	3.34	5.00
13	-0.31	0.32	0.83	-0.31	0.01	0.32	0.83	0.02	2.51	5.00	0.02	1.68	3.34	5.00
14	-0.31	0.26	0.83	-0.31	-0.03	0.22	0.83	0.02	2.51	5.00	0.02	1.68	3.34	5.00
15	-0.31	0.26	0.83	-0.31	-0.03	0.22	0.83	0.02	2.51	5.00	0.02	1.68	3.34	5.00
16	-0.31	0.32	0.83	-0.31	0.01	0.32	0.83	0.02	2.51	5.00	0.02	1.68	3.34	5.00
17	-0.31	0.26	0.83	-0.31	-0.03	0.22	0.83	0.02	2.51	5.00	0.02	1.68	3.34	5.00

Figure (9) Input data for LHPWSS

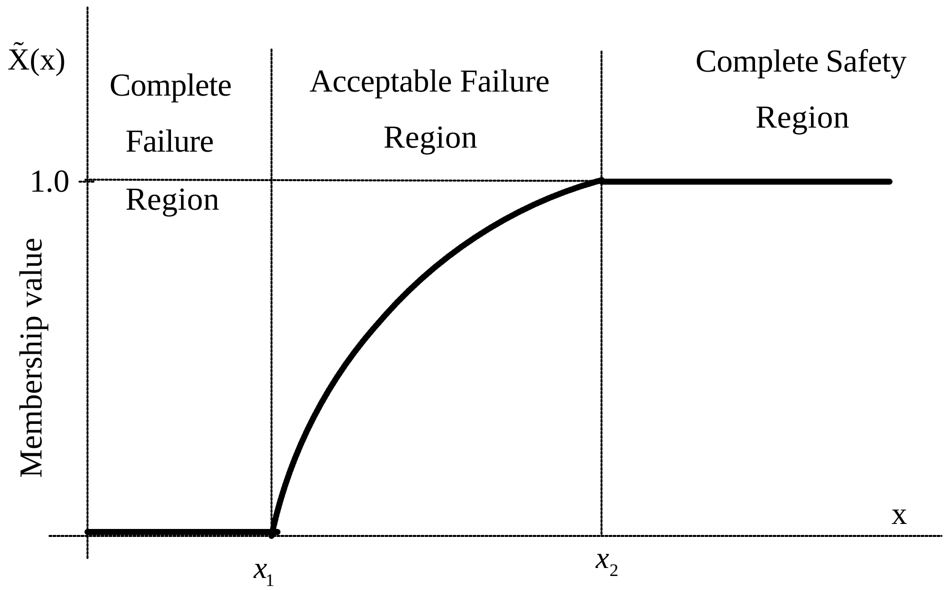

Figure (10) Fuzzy representation of the acceptable failure region.

3.3.3 System-failure membership function

The system-failure membership function is used in the calculation of the fuzzy resiliency index. This membership function represents the system's time of recovery from the failure state. For each type of failure the system might have a different recovery time. Therefore, a series of fuzzy sets, each for different type of failure, are developed for the system under consideration (El-Baroudy and Simonovic, 2003 and 2004). Then the maximum recovery time is used to represent the system-characteristic recovery time as follows, (Kaufmann and Gupta, 1985)

$$\tilde{T}(a) = \left(\max_{j \in J}[t_{1_1}(a), t_{1_2}(a), \ldots\ldots, t_{1_J}(a)], \max_{j \in J}[t_{2_1}(a), t_{2_2}(a), \ldots\ldots, t_{2_J}(a)] \right) \qquad \ldots\ldots\ldots(12)$$

where:

$\tilde{T}(a)$ is the system fuzzy maximum recovery time at α -cut (as defined by Equation 2);

$t_{1_J}(a)$ is the lower bound of the j-th recovery time at α -cut (as defined by Equation 2);

$t_{2_J}(a)$ is the upper bound of the j-th recovery time at α -cut (as defined by Equation 2); and

J is total number of fuzzy recovery times.

Multi-component systems have several system-failure membership functions representing the system-failure for each component. Aggregation of these membership functions results in a system-failure membership function for the whole-system.

Parallel and redundant components are aggregated into serial groups using the fuzzy maximum operator. For parallel system configuration composed of M components, the m-th component has a maximum recovery time membership function $\tilde{T}_m(t)$, defined on the universe of discourse T. Therefore, the system-failure membership function (i.e. the membership function that represents the system recovery time) can be calculated as follows, (El-Baroudy and Simonovic, 2003 and 2004)

$$\tilde{T}(t) = \max_{M}\left(\tilde{T}_1, \tilde{T}_2, \ldots\ldots\ldots, \tilde{T}_M\right) \qquad \ldots\ldots\ldots(13)$$

where:

$\tilde{T}(t)$ is the whole system-failure membership function; and

$\left(\tilde{T}_1, \tilde{T}_2, \ldots\ldots\ldots, \tilde{T}_M\right)$ are recovery time membership functions

for different components.

The system-failure membership function is then calculated for the N serially connected components using, (El-Baroudy and Simonovic, 2003 and 2004)

$$\tilde{T}(t) = \tilde{T}_c(t) \qquad \ldots\ldots\ldots.(14)$$

given

$$S(\tilde{T}_c) = \max_N \left(S(\tilde{T}_1), S(\tilde{T}_2), \ldots\ldots\ldots, S(\tilde{T}_N)\right)$$

and ……….(15)

$$\tilde{T}_c(1) = \max_N \left(\tilde{T}_1(1), \tilde{T}_2(1), \ldots\ldots\ldots, \tilde{T}_N(1)\right)$$

where:

$\tilde{T}(t)$ is the whole system recovery time membership function;

$\tilde{T}_c(t)$ is the controlling recovery time membership function;

$S(\tilde{T}_c)$ is the support of the controlling recovery time membership function (as defined by Equation 3);

$\left(S(\tilde{T}_1),S(\tilde{T}_2),.........,S(\tilde{T}_N)\right)$ are the support of the N components recovery time membership functions (as defined by Equation 3);

$\tilde{T}_c(1)$ is the controlling recovery time membership function at the α -cut =1 (as defined by Equation 2); and

$\left(\tilde{T}_1(1),\tilde{T}_2(1),.........,\tilde{T}_N(1)\right)$ are the recovery time membership functions at α -cut =1 of the N components (as defined by Equation 2).

3.3.4 Fuzzy reliability-vulnerability index

Figure 11 shows an example of a multi-component system. The system has two parallel components connected serially to a third component that has a redundant component. The component-state membership functions for all five components are listed in Table 1, together with the system-state membership functions for the parallel and redundant components.

Figure 12 illustrates the process of calculating the system-state membership function for the given example. The membership functions of parallel and redundant components are summed to obtain three system-state membership functions for the serial components. The resulting membership function is then calculated using the fuzzy minimum operator, represented by the shaded area in Figure 12.

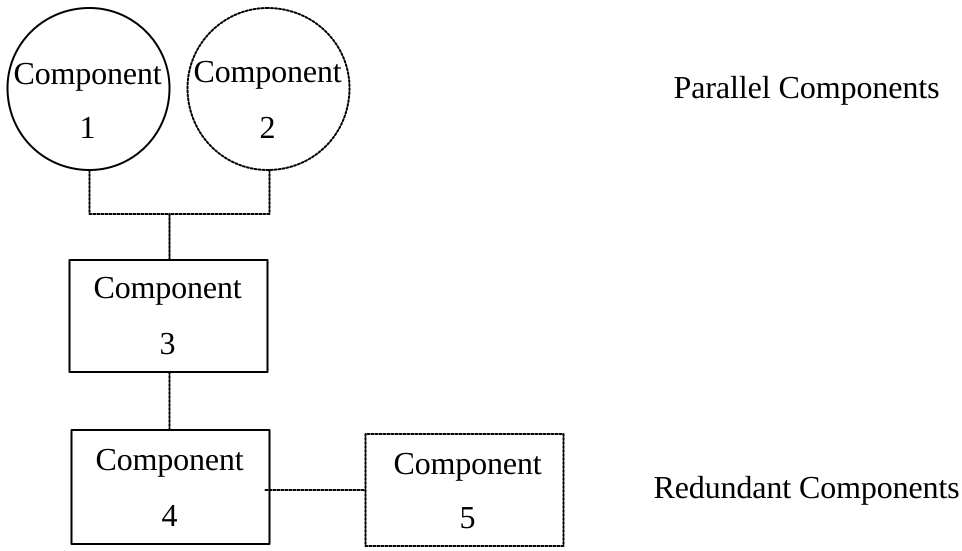

Figure (11) Typical example of a multi-component system.

Table (1) MF calculations for a multi-component system.

Component	**MF**	**Parallel Summation**	**Redundant Summation**	**System-State MF**
Component 1	(1,2,3)	(2,4,6)	NA	Min [(2,4,6), (1,3,5), (1,2,3)]
Component 2	(1,2,3)		NA	
Component 3	(1,3,5)	NA	NA	
Component 4	(0.5,1,1.5)	NA	(1,2,3)	
Component 5	(0.5,1,1.5)	NA		

The compatibility between the system-state and the acceptable level of performance is the basis for the calculation of the fuzzy combined reliability-vulnerability performance index, as shown in Figure 13.

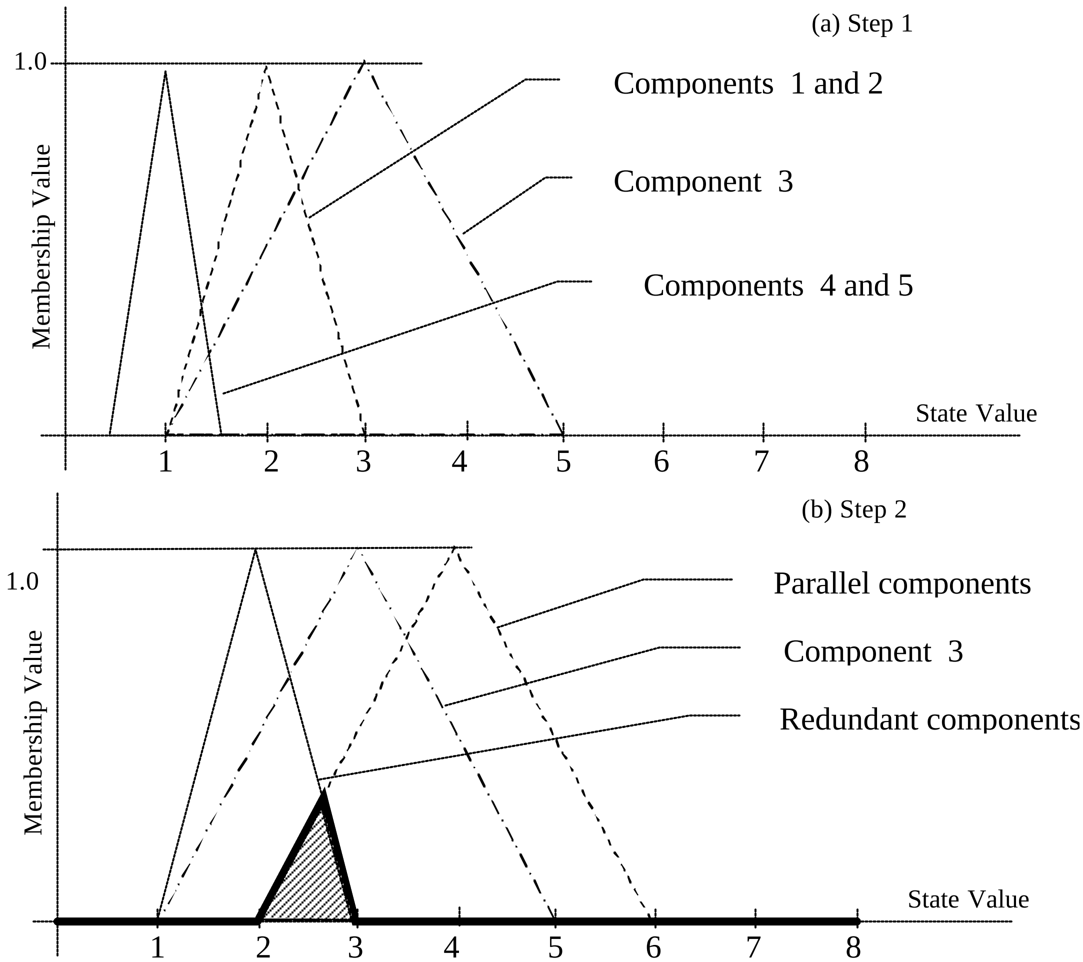

Figure (12) Calculation of the system-state membership function for the multi-component system.

The compatibility measure (CM) is calculated as:

$$\text{Compatibility Measure (CM)} = \frac{\text{Weighted overlap area}}{\text{Weighted area of system-state function}} \qquad \text{……….(16)}$$

and then used to calculate the combined fuzzy reliability-vulnerability performance index

$$\text{FuzzyReliability-VulnerabilityIndex} = \frac{\max_{i \in K}\{CM_1, CM_2, ...CM_i\} \times LR_{max}}{\max_{i \in K}\{LR_1, LR_2, ...LR_i\}} \quad ....(17)$$

where:

LR_{max} is the reliability measure of acceptable level of performance for which the system-state has the maximum compatibility value(CM);

LR_i is the reliability measure of the i-th acceptable level of performance (as defined by Equation 11) ;

CM_i is the compatibility measure for system-state with the i-th acceptable level of performance; and

K is the total number of defined acceptable levels of performance.

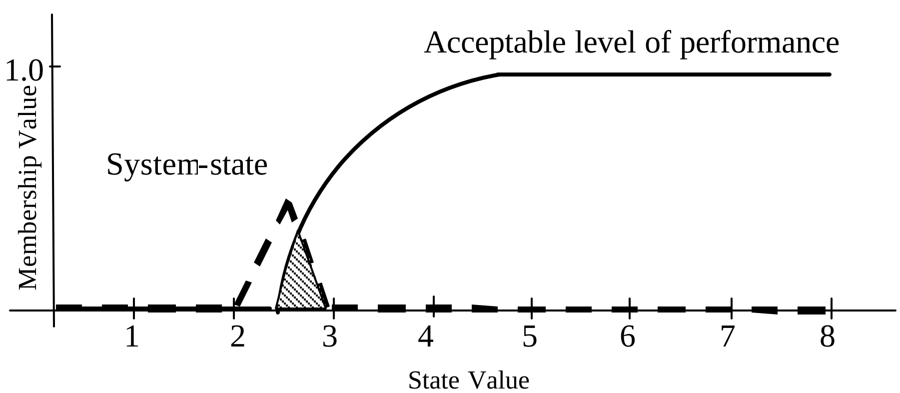

Figure (13) Overlap area between the system-state membership function and the acceptable level of performance.

Figure 14 shows the flow chart for the calculation of the fuzzy combined reliability-vulnerability index;

- Step (1); reading input data from the spreadsheet file containing the component-state membership functions. Both types of membership functions, triangular and trapezoidal, are constructed;
- Step (2); storing the input data in an appropriate data format (i.e., structure array).
- Step (3); transforming input data into both, triangular and trapezoidal membership function shapes. Appendix (II) contains source code for transformation into triangular and trapezoidal shapes;
- Step (4); all parallel and redundant components are augmented using the fuzzy summation operator to calculate the membership functions representing the parallel and the redundant groups, respectively. The system is turned into a group of serially connected components, and then the maximum operator is used to calculate the system-state membership function. Appendix (II) contains the source code for the fuzzy operator, specially designed for this case study; and
- Step (5); calculating the fuzzy combined reliability-vulnerability index based on the overlap area between the system-state and the acceptable level of performance.

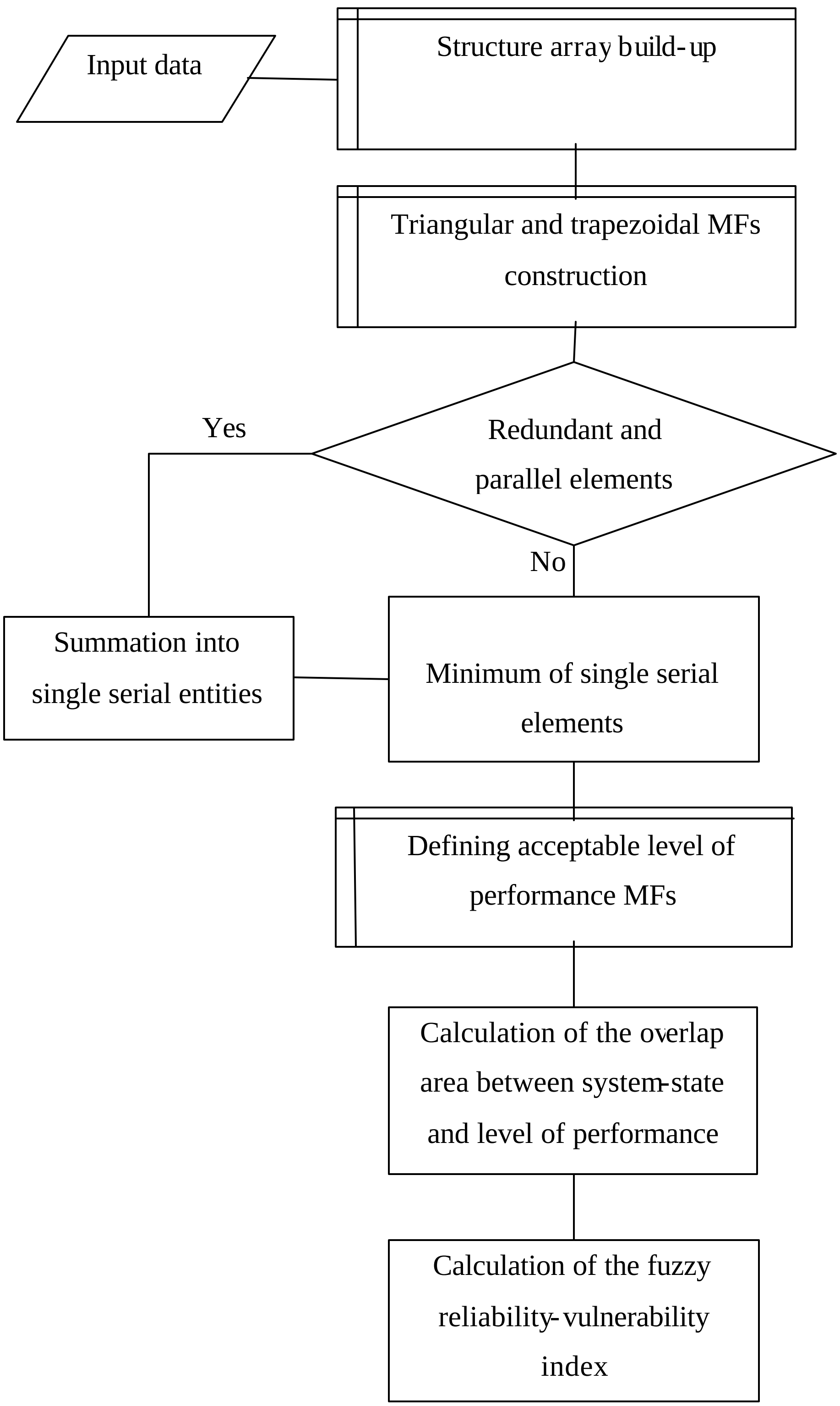

Figure (14) Flow chart for the fuzzy reliability-vulnerability index calculation.

3.3.5 Fuzzy robustness index

Robustness is a measure of system performance that is concerned with the ability of the system to adapt to a wide range of possible demand conditions, in the future, at little additional cost (Hashimoto et al, 1982b). The fuzzy form of change in future conditions can be reflected through the change in the acceptable level of performance and, also, in the change of the system-state membership function (El-Baroudy and Simonovic, 2003 and 2004). The change in overlap area is used to calculate system fuzzy robustness index as follows:

$$FuzzyRobustnessIndex = \frac{1}{CM_1 - CM_2} \qquad \text{.........(18)}$$

where:

CM_1 is the compatibility measure before the change in conditions; and

CM_2 is the reliability after the change in conditions.

Figure 15 shows the flow chart for the calculation of the fuzzy robustness index;

- Step (1); reading input data from the spreadsheet file containing the component-state membership functions. Both types of membership functions, triangular and trapezoidal, are constructed;

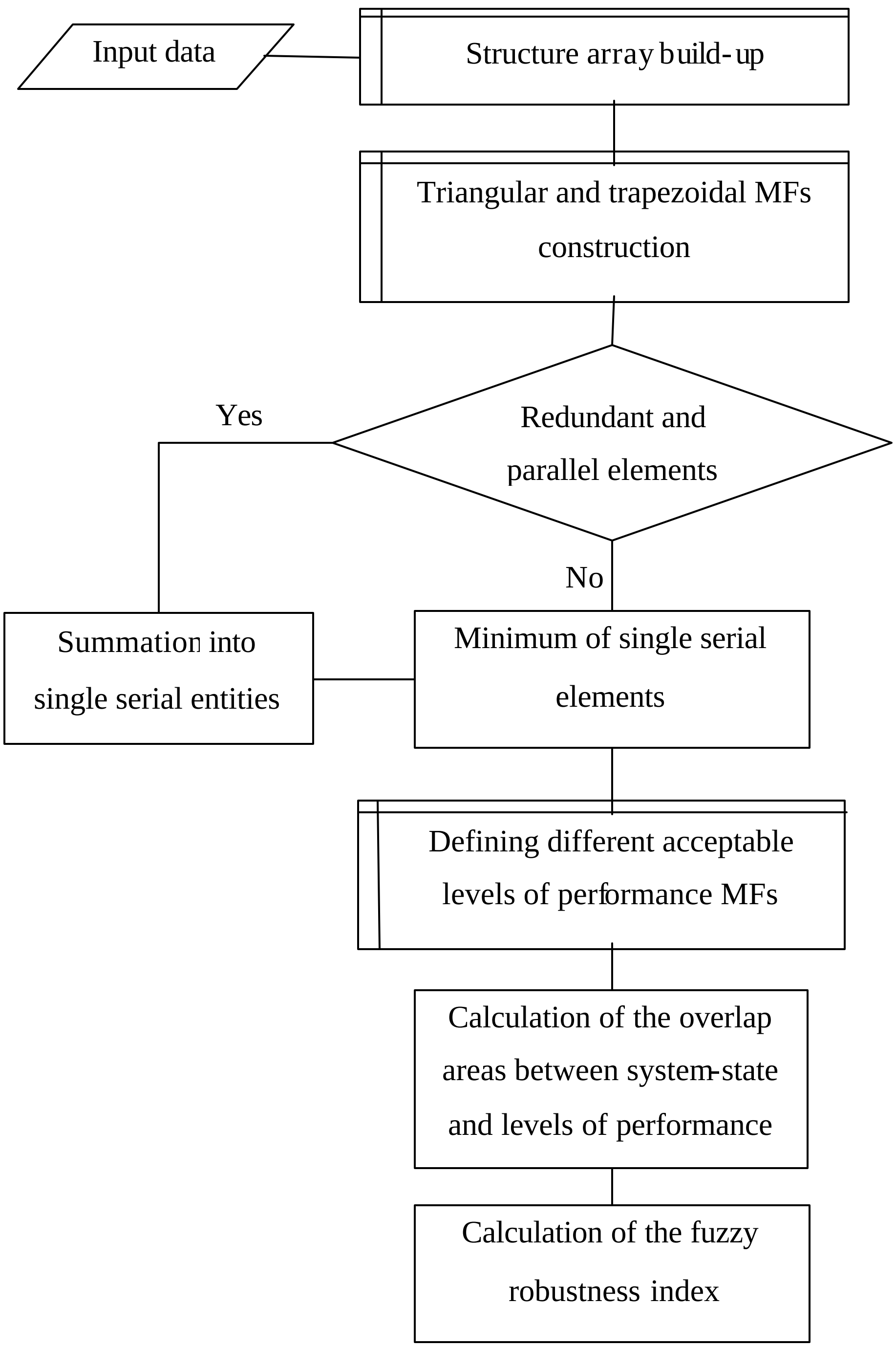

Figure (15) Flow chart for the fuzzy robustness index calculation.

- Step (2); storing the input data in an appropriate data format (i.e., structure array).
- Step (3); transforming input data into both, triangular and trapezoidal shapes. Appendix (II) contains source code for transformation into triangular and trapezoidal shapes;
- Step (4); all parallel and redundant components are augmented using the fuzzy summation operator to calculate the membership functions representing the parallel and the redundant groups, respectively. The system is transformed into a group of serially connected components, and then the maximum operator is used to calculate the system-state membership function. Appendix (II) contains the source code for the fuzzy operator, specially designed for this case study; and
- Step (5); calculating the fuzzy robustness index based on the overlap area between the system-state and predefined acceptable levels of performance.

3.3.6 Fuzzy resiliency index

Resiliency is a measure of system's time for recovery from the failure state (Hashimoto et al, 1982a). The fuzzy resiliency index is calculated using the value of the center of gravity of the system-failure membership function (El-Baroudy and Simonovic, 2003 and 2004):

$$\text{FuzzyResilienceIndex} = \left[\frac{\int_{t_1}^{t_2} t\, \tilde{T}(t)dt}{\int_{t_t}^{t_2} \tilde{T}(t)dt} \right]^{-1} \qquad \text{……….(19)}$$

where;

$\tilde{T}(t)$ is the system fuzzy maximum recovery time membership function;

t_1 is the lower bound of the support of the system recovery time membership function (as defined by Equation 3); and

t_2 is the upper bound of the support of the system recovery time membership function (as defined by Equation 3).

The calculation script allows the use of both triangular and trapezoidal shapes, as shown in Figure 16.

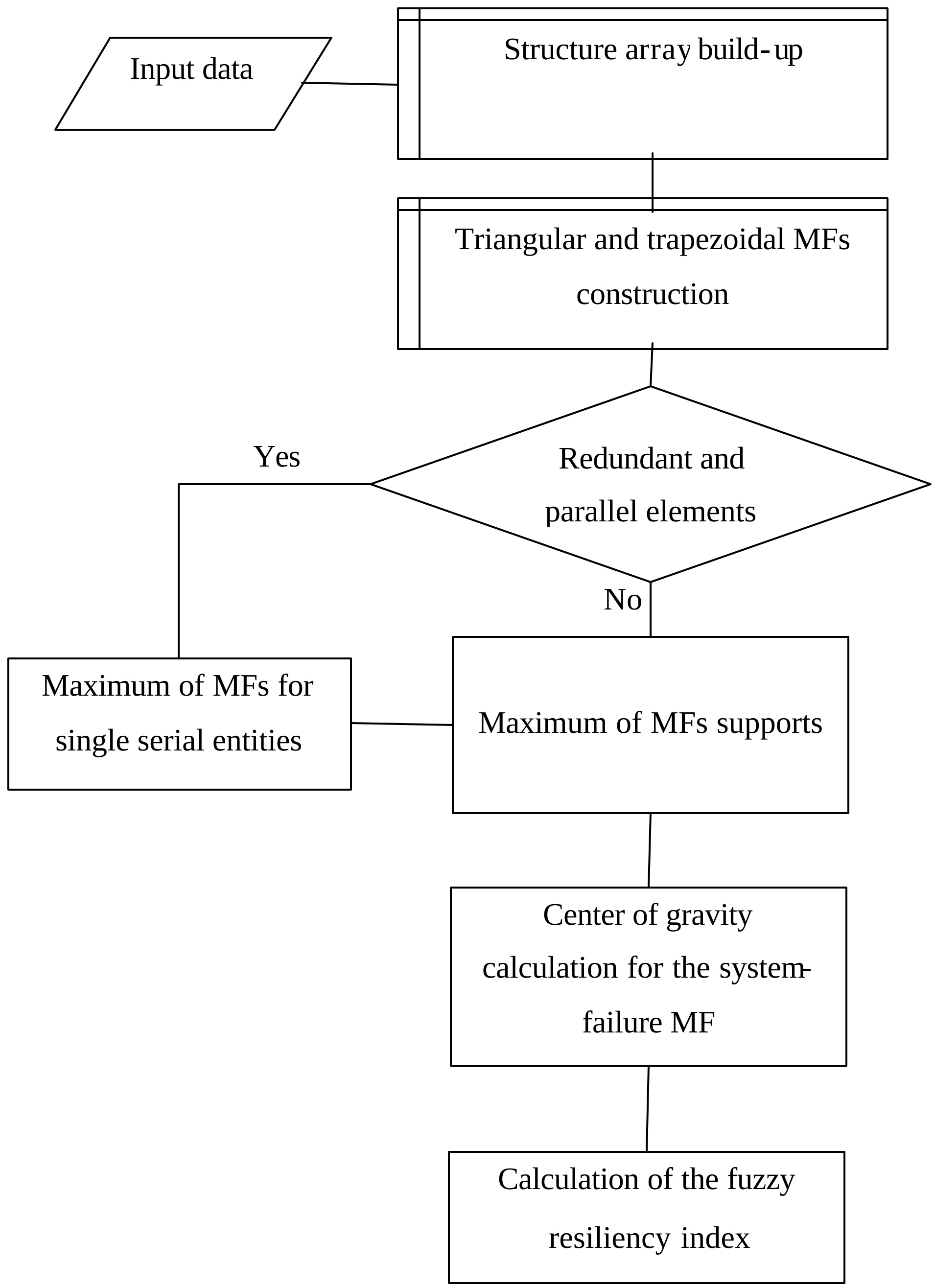

Figure (16) Flow chart for calculation of the fuzzy resiliency index.

4 ANALYSIS OF THE LAKE HURON SYSTEM

4.1 LHPWSS system representation and data

The system representation provides the integrated layout that reflects the failure-driven relationships among different components. Figure 17 shows LHPWSS with all major components combined in an integrated layout. Component-state and component-failure membership functions are constructed based on the data from (Earth Tech Canada Inc.,2000), (Earth Tech Canada Inc.,2001), (American Water Services Canada-AWSC, 2003a), (American Water Services Canada-AWSC, 2003b), and (DeSousa and Simonovic, 2003) for the LHPWSS. Appendix (I) includes all the input data used in the calculation of the triangular and trapezoidal membership functions.

4.2 Results

4.2.1 Assessment of the fuzzy performance Indic ices

This section presents an assessment of the three fuzzy performance indices for the LHPWSS. Three acceptable levels of performance are arbitrary defined on the universe of the margin of safety; as (0.6,0.7,5.0,5.0), (0.6,1.2,5.0,5.0), and (0.6,5.0,5.0,5.0). They are selected to reflect three different views of decision-makers as shown by the reliability measure in Equation 11. Their reliability measures are 4.20, 1.20 and 0.68, respectively. Further, they are referred to as reliable level (level 1), neutral level (level 2), and unreliable level (level 3), as shown in Figure 18.

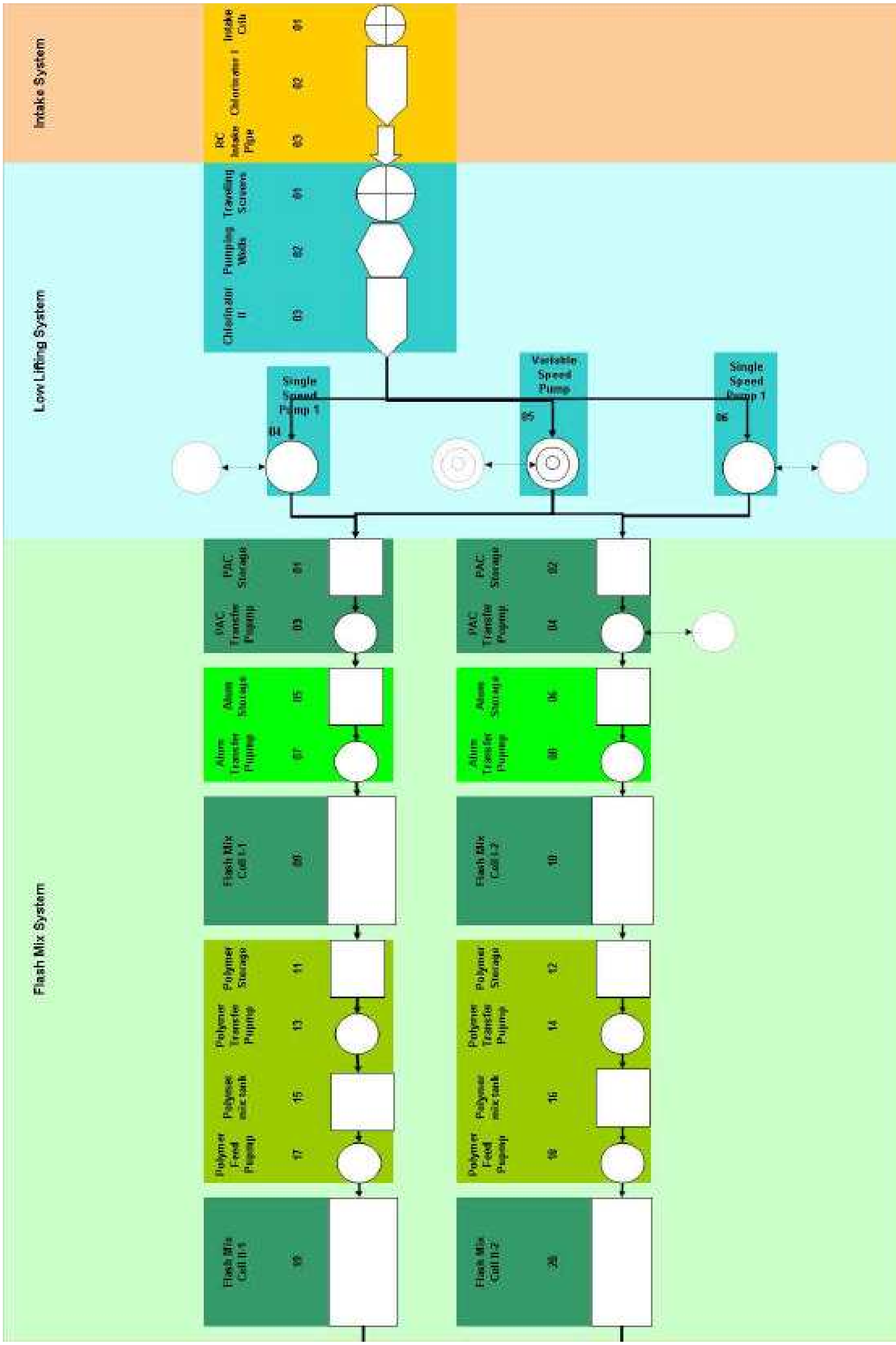

Figure (17) LHPWSS system integrated layout- Part 1

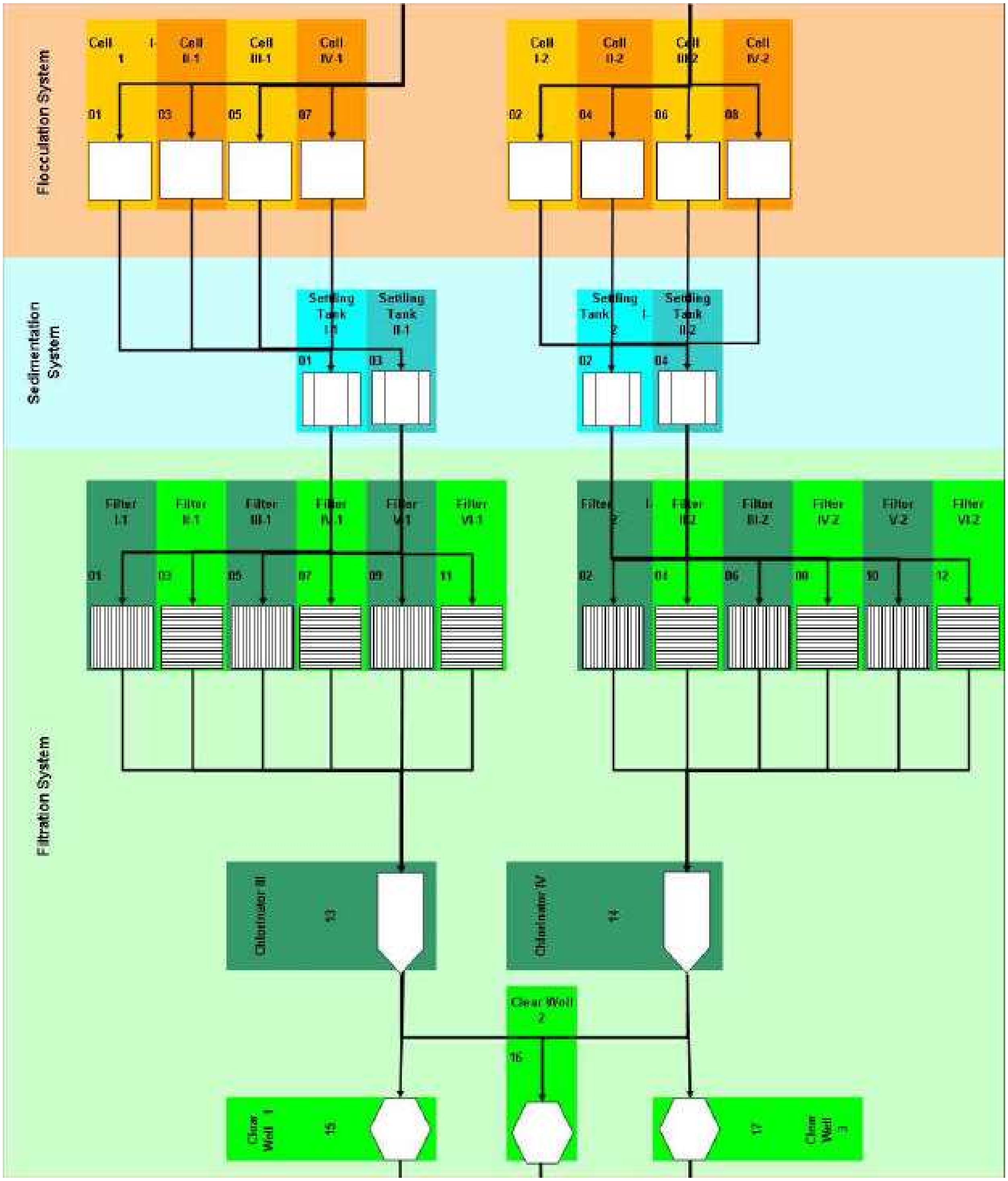

Figure (17) LHPWSS system integrated layout- Part 2

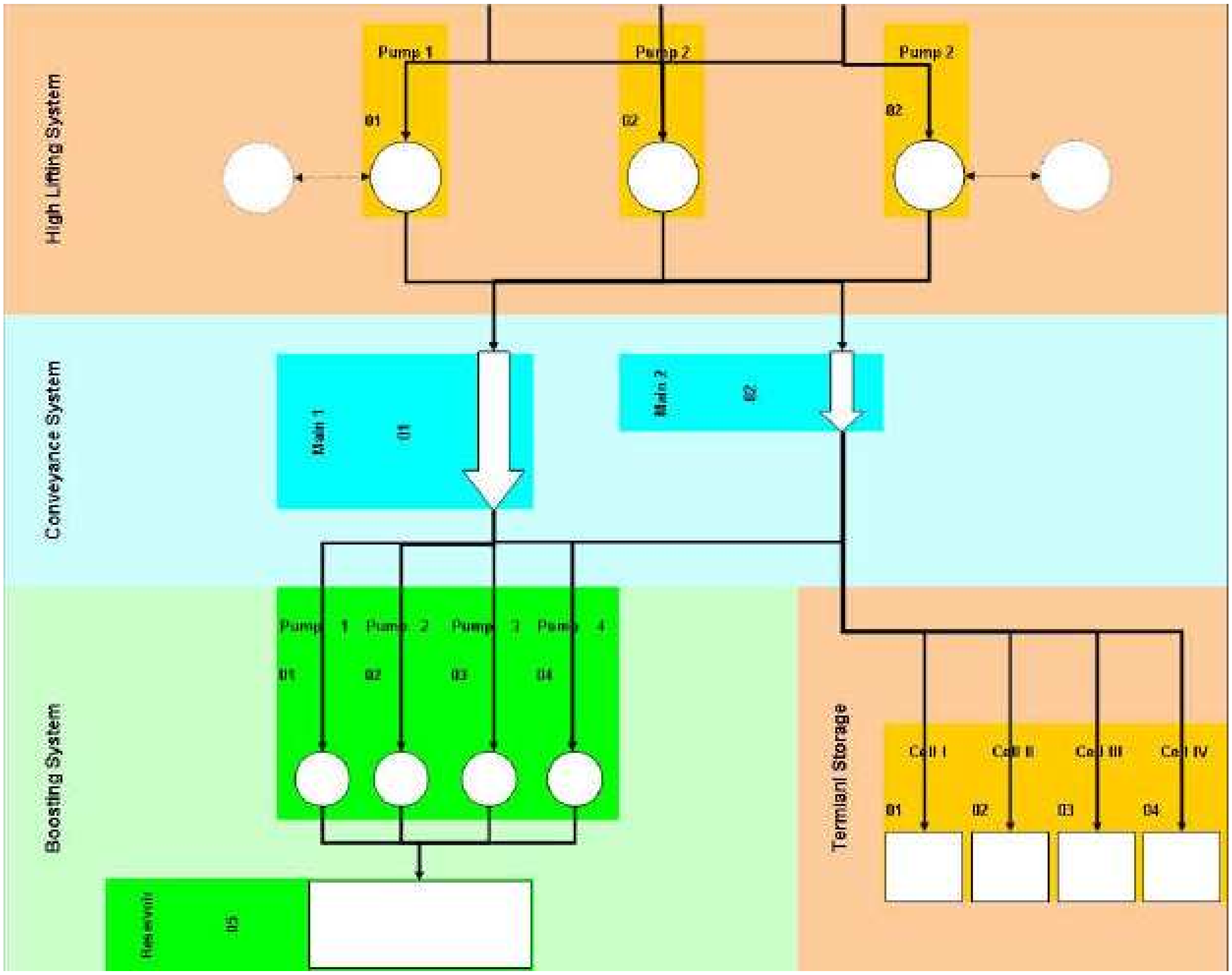

Figure (17) LHPWSS system integrated layout- Part 3

The results show that the combined reliability-vulnerability index for LHPWSS is 0.699. This value reflects the compatibility of the system with one of the three predefined levels of performance, as defined in Equation 17; in this case it is the reliable level (level 1). Therefore, the reliability of the system is relatively high, taking into account that the system is almost 70% compatible with the highest level of performance. The fuzzy robustness index for the LHPWSS is -2.12. Taking into consideration, that this value is the inverse of change in the overlap area, as defined in Equation 18, LHPWSS is considered to be highly robust as the overlap area increase by more than 47%. The fuzzy resiliency index value for the LHPWSS is 0.017, which means that it takes the system more than 58 days to return to the full operation mode, as defined by Equation 19. This value is relatively high as it means the system service is disrupted for about 2 months and large portion of the population served by this system (estimated to be about 325 000 person) will be affected by this disruption.

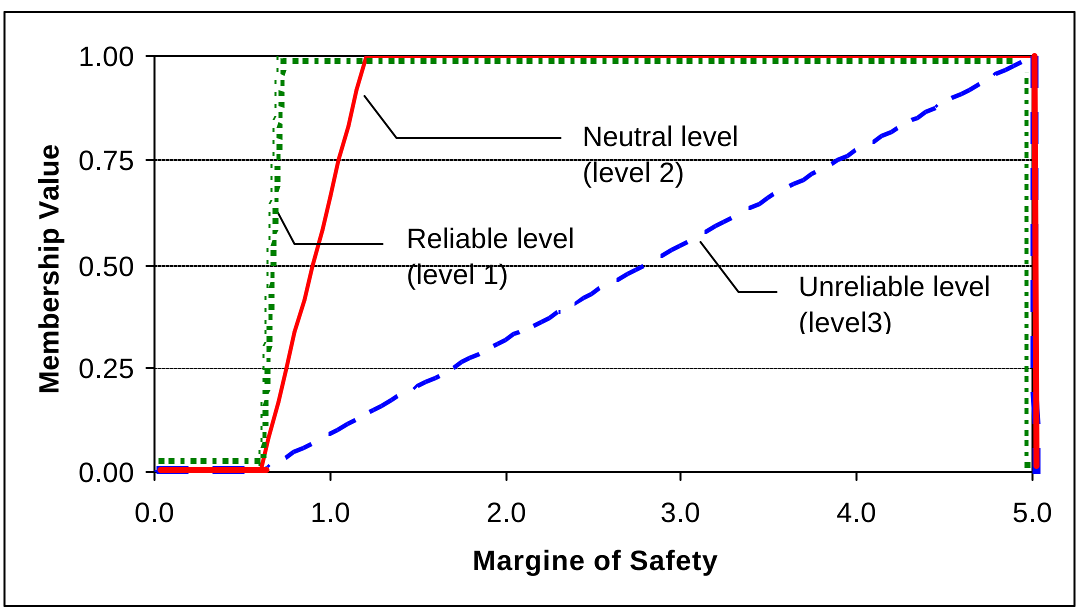

Figure (18) Acceptable levels of performance.

4.2.2 Importance of different membership function shapes

The effect of the membership function shape is investigated by calculating the three fuzzy performance indices using triangular and trapezoidal shapes. Table 2 shows the calculated fuzzy performance indices for triangular and trapezoidal membership function shapes, respectively. For the two shapes, the values of the reliability-vulnerability index are relatively high (i.e. over 0.60), taking into consideration that the maximum value of the index is 1. As shown in Figure 19, most of the system-state membership function overlaps with the reliable level of performance (level 1). This indicates that LHWPSS is highly reliable and less vulnerable to disruption in service. This is expected because; (i) the LHPWSS system has over 20 parallel groups of components and 6 redundant groups as shown in Figure 18, and (ii) many individual components are designed with a 35 % overload capacity (Earth Tech Canada Inc.,2001). This positively increases the capacity and consequently the reliability of the whole system.

There is no significant difference in the fuzzy reliability-vulnerability index values for the triangular and trapezoidal membership function shapes (i.e. the index value for the trapezoidal shape is less than 9% of the index value for the triangular shape), as shown in Table 2. This is because the change in the area of the system-state membership function is not significant and consequently the overlap area, as shown in Figure 19. Generally, it can be concluded that use of the trapezoidal shape leads to relatively lower reliability-vulnerability index than the triangular shape.

The robustness index value for LHPWSS system decreases from -2.120 (triangular shape) to -2.473 (trapezoidal shape). This corresponds to the deterioration in the system reliability-vulnerability index vale from 0.699 to 0.642. This is clear for the first case where the LHPWSS system is required to satisfy higher reliability conditions (represented by the transition from the neutral level to the reliable level). The NA values in table 2 indicate that there is no change in the overlap area and consequently the value of the robustness index will approach infinity.

The resiliency index is not affected by the shape of the membership function, since the center of gravity for both system-failure membership functions coincide, as shown in Figure 20.

Table (2) The LHPWSS system fuzzy performance indices for different membership function shapes.

Fuzzy Performance Index	**Triangular MF**	**Trapezoidal MF**
Combined Reliability-Vulnerability	0.699	0.642
Robustness (level 2 – level 1)	NA*	NA*
Robustness (level 2 – level 1)	-2.120	-2.473
Robustness (level 3 – level 2)	-2.120	-2.473
Resiliency	0.017	0.017

NA* Not-available value as there is no change in overlap area.

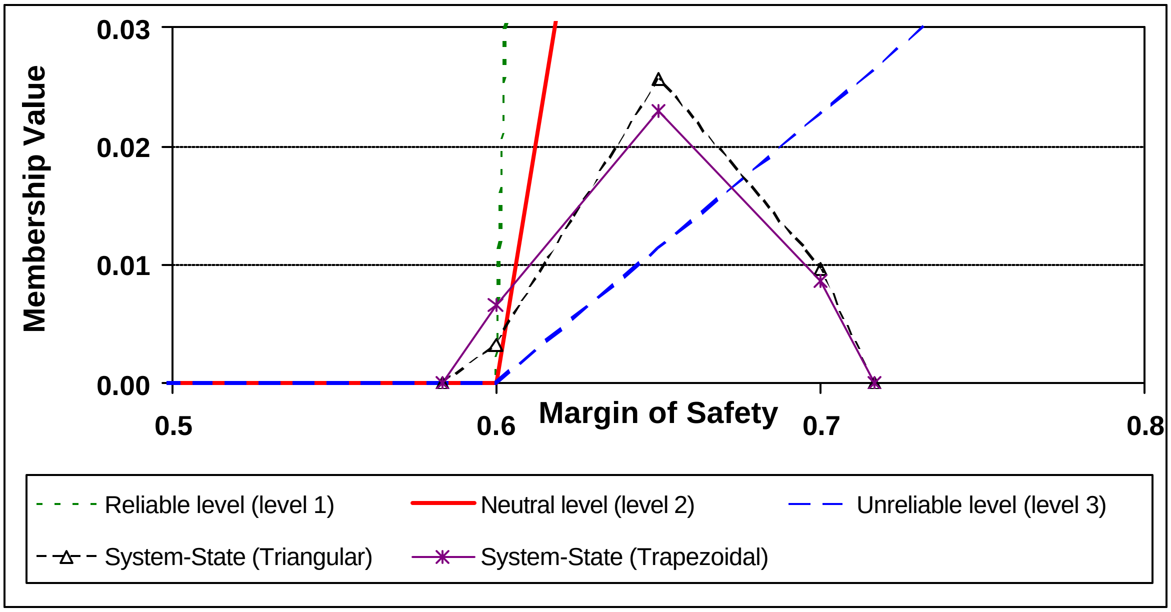

Figure (19) Resulting system-state membership functions for triangular and trapezoidal input membership functions.

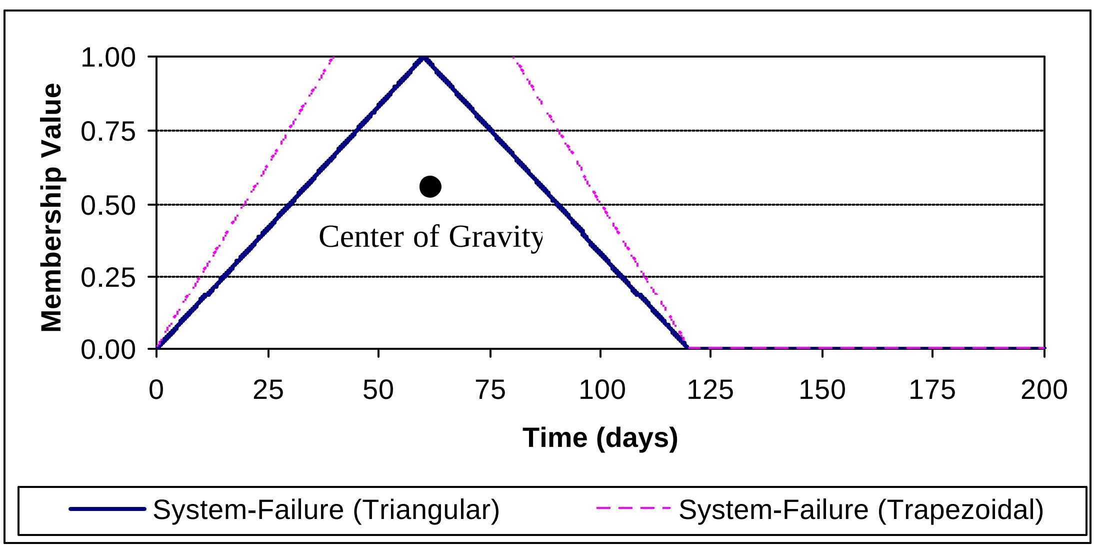

Figure (20) System-failure membership functions using triangular and trapezoidal shapes.

4.2.3 Significance of system components

System reliability depends on the reliability of its components. However, not all components are of equal importance (different location; different rate;…etc). For example, serial components have a more significant effect on the overall system reliability than parallel components, because the failure of any serial component leads to the failure of the whole system. Therefore, system's performance can only be enhanced by improving the performance of critical components. Critical component is the component that significantly reduces the area of the system-state membership function and accordingly the fuzzy performance indices of the system.

The developed computational procedure can be used to identify the critical components of the system. As mentioned in Chapter 3, the calculation transforms the multi-component system into a system of serially connected components. The fuzzy summation operator is used to turn parallel and redundant components into single entities with equivalent component-state membership functions. Then, the fuzzy minimum operator is used to sum up all serial components and entities into the system-state membership function. Observing the change in the system-state membership function can be used to identify critical system.

For the triangular membership function shape, the change resulting in the system-state membership function is shown in Figure 21. The system-state membership function changes significantly with the addition of the PAC transfer pump. This is the point where the flash mix is introduced into the system. The enhancement of flash mix system components will lead to the enhancement of the overall system performance. Looking into the components of the flash mix system, it is found that the PAC transfer pump has the smallest component-state membership function relative to other flash mix components.

If the capacity of the PAC transfer pump is increased, the area of the component-state membership function will increase. This will lead to a direct improvement of the overall system performance. Table 3 summarizes the fuzzy performance indices for both cases (i.e., before and after changing the PAC transfer pump's component-state membership function value). The combined reliability-vulnerability index has increased from 0.699 to 0.988, which means an increase of 41% of the original value. On the other hand, the fuzzy robustness index has increased from -2.120 to -1.127 indicating an improvement of the system robustness.

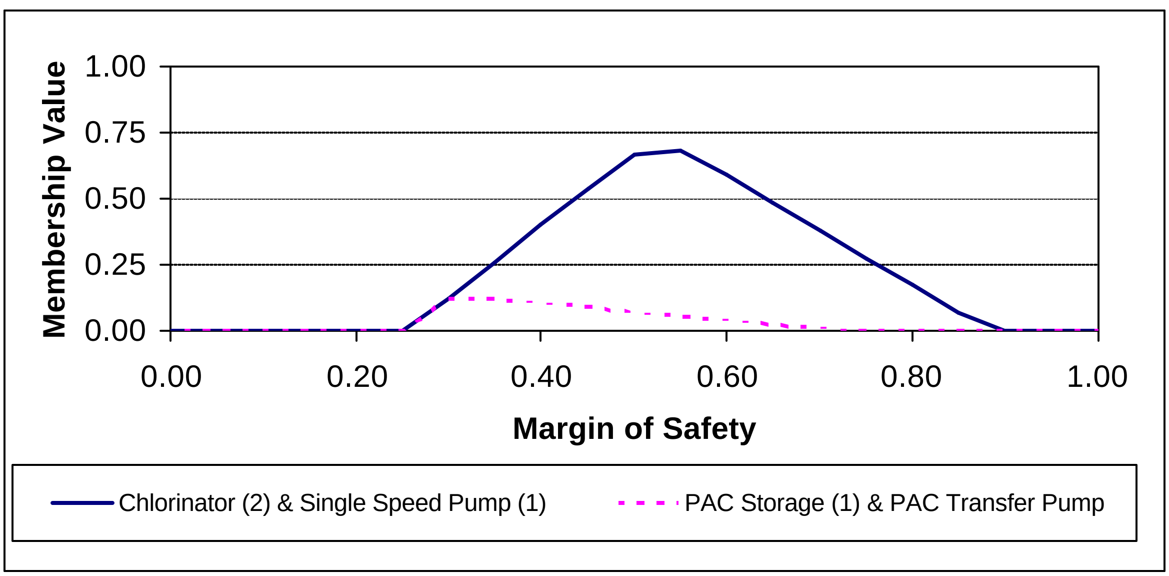

Figure (21) System-state membership function change for different system components.

Table (3) System fuzzy performance indices change due to the improvement of PAC transfer pump capacity.

Fuzzy performance index	**Before change**	**After change**

Combined Reliability-Vulnerability	0.699	0.988
Robustness (level 2 – level 1)	NA*	NA*
Robustness (level 3 – level 1)	-2.120	-1.127
Robustness (level 3 – level 2)	-2.120	-1.127

NA* Not-available value as there is no change in overlap area.

Table 4 shows three different changes in the maximum capacity of the PAC transfer pump and their impact on the system fuzzy performance indices. A 5% increase in the maximum capacity of the PAC transfer pump resulted in a more than 7% increase in the combined reliability-vulnerability index with almost no significant increase in the robustness index.

Change in the maximum capacity of the critical component and consequently its membership function results in the appearance of new critical components that control the overall system performance. Therefore, the optimum improvement of system performance can be achieved by an iterative procedure for analysis of the system fuzzy performance indices.

Table (4) Change in the system fuzzy performance indices due to change in the maximum capacity of the PAC transfer pump.

Fuzzy performance index	**Percentage change of the maximum capacity**		
	300%	**20%**	**5%**
Reliability-Vulnerability	0.988	0.921	0.749
Robustness (level 2 – level 1)	NA*	NA*	NA*
Robustness (level 3 – level 1)	-1.127	-1.607	-2.100
Robustness (level 3 – level 2)	-1.127	-1.607	-2.100

NA* Not-available value as there is no change in overlap area.

5 ANALYSIS OF THE ELGIN AREA SYSTEM

5.1 EAPWSS system representation and data

The system representation provides the integrated layout that reflects the failure-driven relationship among different components. Figure 22 shows EAPWSS with all major components combined in an integrated layout. Component-state and component-failure membership functions are constructed based on the data from the (Earth Tech Canada Inc.,2000), (Earth Tech Canada Inc.,2001), (American Water Services Canada-AWSC, 2003a), (American Water Services Canada-AWSC, 2003b), and (DeSousa and Simonovic, 2003). Appendix (I) includes all the input data used in the calculation of the triangular and trapezoidal membership functions representing component-state and component-failure.

5.2 Results

5.2.1 Assessment of the fuzzy performance Indic ices

The same acceptable levels of performance are used in the assessment process (i.e., (0.6,0.7,5.0,5.0), (0.6,1.2,5.0,5.0), and (0.6,5.0,5.0,5.0)). The combined reliability-vulnerability index for EAPWSS is 0.042. This value is extremely low, taking into account that the system is only 4% compatible with the highest level of performance. The fuzzy robustness index for the system is 1.347. Taking into consideration, that this value is the inverse of change in the overlap area, as defined in Equation 18, EAPWSS has low robustness as the overlap area is reduced by more than 74%.

The fuzzy resiliency index value for the EAPWSS is 0.054, which means that it takes the system more than 18 days to return to the full operation mode, as defined by Equation 19. This value is relatively low as it means the system service is disrupted for less than 3 weeks.

5.2.2 Importance of different membership function shapes

As performed in LHPWSS analysis, the effect of the system-state membership function shape is investigated using triangular and trapezoidal shapes. Table 5 shows values of fuzzy performance indices for EAPWSS system.

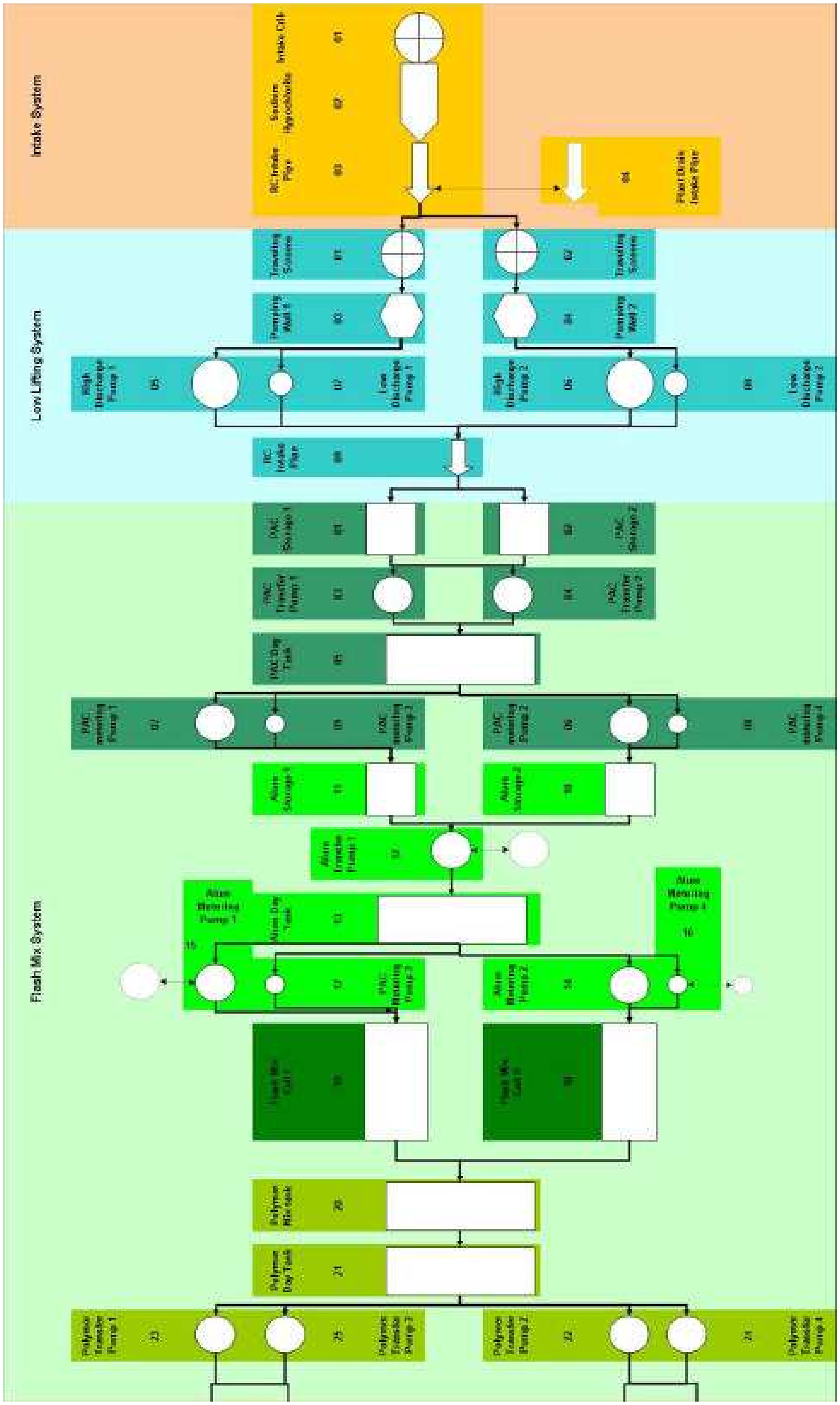

Figure (22) EAPWSS system integrated layout- Part 1.

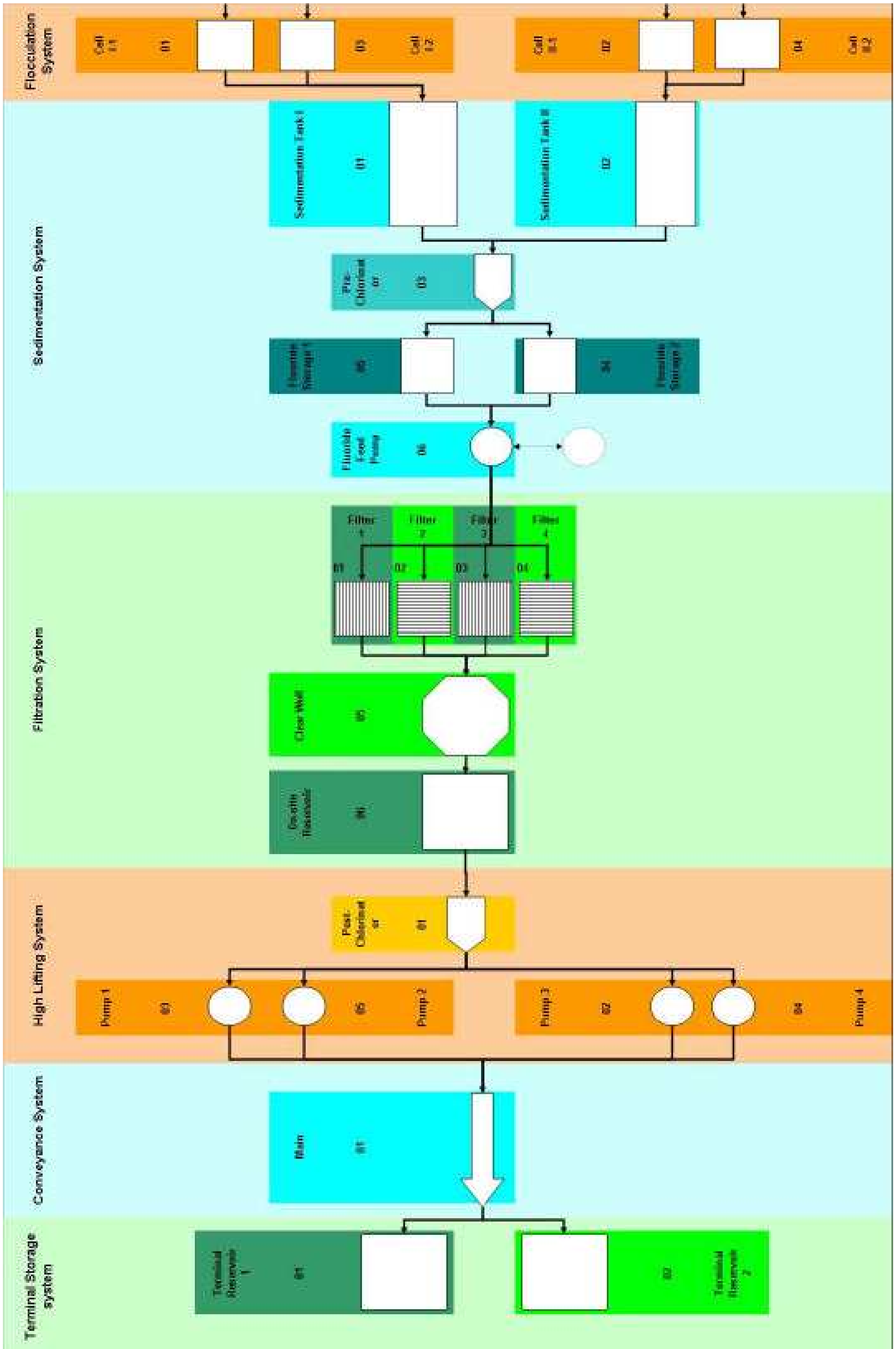

Figure (22) EAPWSS system integrated layout- Part 2

Table (5) The EAPWSS system fuzzy performance indices for different membership function shapes.

Fuzzy performance index	**Triangular MF**	**Trapezoidal MF**
Combined Reliability-Vulnerability	0.042	0.017
Robustness (level 2 – level 1)	1.347	3.314
Robustness (level 3 – level 1)	NA*	NA*
Robustness (level 3 – level 2)	-1.347	-3.314
Resiliency	0.054	0.054

NA* Not-available value as there is no change in overlap area.

As shown in Table 5, the reliability-vulnerability index value has decreased from 0.042 for the triangular shape to 0.017 for the trapezoidal shape (i.e. more than 50 % decrease of the value for the triangular shape). This is similar to the behavior for LHPWSS system as shown in Figure 23. The robustness index values, also, changes with different shapes of membership functions.

For the triangular shape, the robustness index value is 1.347, while it is 3.314 for the trapezoidal shape. It has to be noted that the sign of the fuzzy robustness index indicates the type of change in the overlap area with the corresponding acceptable levels of performance. Therefore, it is more important to observe the absolute value of the fuzzy robustness index rather than its sign.

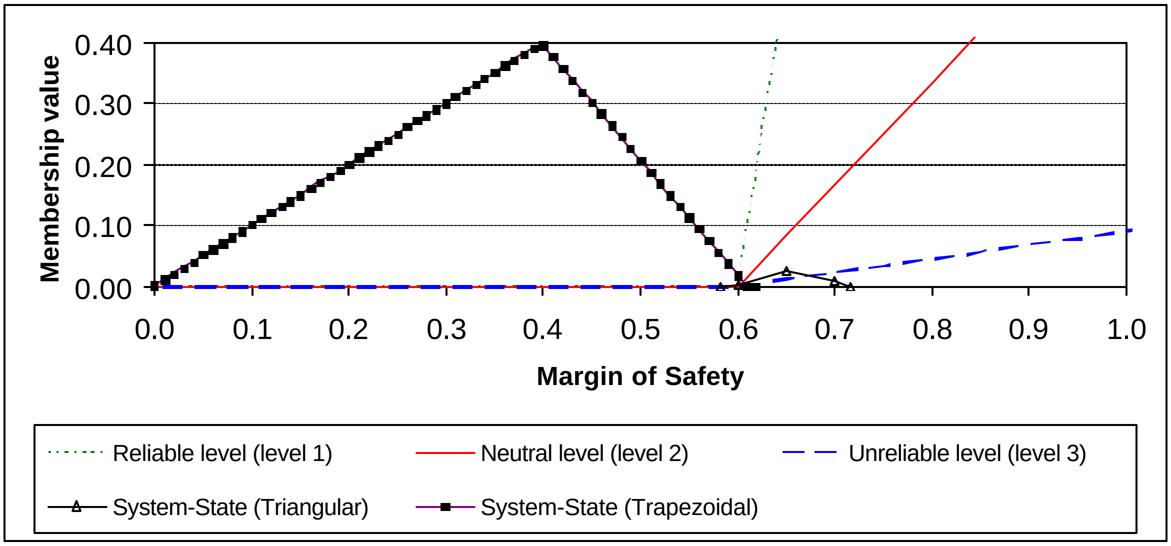

Figure (23) Resulting EAPWSS system-state membership functions for triangular and trapezoidal input membership functions.

5.2.3 Significance of system components

The change of the system-state membership function is observed to identify the critical system components. Triangular membership functions are used and the resulting system-state membership function progress is shown in Figure 24. Figure 24 shows that the system-state membership function significantly changes twice, after including the PAC storage and after including the PAC metering pump. Similar to LHPWSS system, this is the point where the flash mix system is introduced into the system. Therefore, improvement of the performance of these components will result in the improvement of the overall system performance.

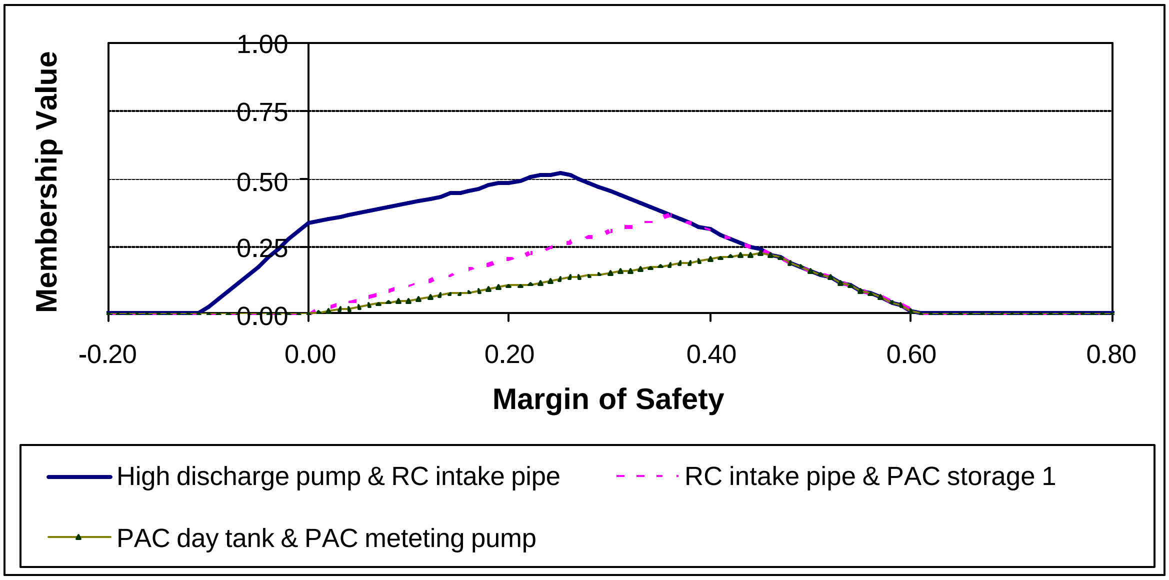

Figure (24) System-state membership function change with introduction of system components.

The PAC components have almost similar component-state membership functions (i.e. for the triangular shape they are (0.00,0.50,1.00)). As a result, the change of maximum capacity of all PAC components is mandatory to significantly change the system-state membership function and consequently the system fuzzy performance indices.

Increasing the maximum capacity of the PAC system to cause a change of the component-state membership functions by 20% is used to investigate the effect of the change on the fuzzy performance indices. This change will be applied to the modal and the end values of the membership function (i.e., the component-state membership function will be (0.00,0.60,1.20)). Table 6 summarizes the fuzzy performance indices for both cases (i.e., before and after changing the PAC component-state membership function value).

Table (6) System fuzzy performance indices change due to the change in the PAC maximum capacity.

Fuzzy performance index	**Before change**	**After change**
Combined Reliability-Vulnerability	0.042	0.047
Robustness (level 1 – level 2)	-1.347	-1.210
Robustness (level 1 – level 3)	NA*	NA*
Robustness (level 2 – level 3)	1.347	1.210

NA* Not-available value as there is no change in overlap area.

The combined reliability-vulnerability index increased by only 12 % (i.e., from 0.042 to 0.047), while the robustness index decreased by 10%. Changing the critical component maximum capacity results in the appearance of new critical components that control the system performance.

6 CONCLUSIONS

The combined fuzzy reliability-vulnerability index, robustness index, and resiliency index are used to asses the performance of the Lake Huron Primary Water Supply System (LHPWSS) and the Elgin Area Primary Water Supply system (EAPWSS). Triangular and trapezoidal membership function shapes are used to examine the sensitivity of these performance indices. They are calculated for arbitrary selected acceptable levels of performance. Three different views of decision-makers are assumed and referred to as reliable, neutral and unreliable levels of performance. The same levels of performance are used for both LHPWSS and EAPWSS systems to facilitate the comparison of the fuzzy performance indices.

Figures 25 (a) and (b) show the three fuzzy performance indices for the two systems. It can be concluded that LHPWSS system is more reliable and less vulnerable than EAPWSS system. The combined reliability-vulnerability index for the LHPWSS system is higher than that of the EAPWSS system for the both triangular and trapezoidal shapes of membership functions (i.e. at least 10 times higher). This is supported by the fact that increasing the system redundancy, by adding parallel and standby components, increases the capacity of the overall system. The LHPWSS system has more than 20 parallel groups and 7 redundant components, while the EAPWSS system has less than 16 parallel groups and 4 redundant elements. This increases the reliability of the LHPWSS system over that of the EAPWSS.

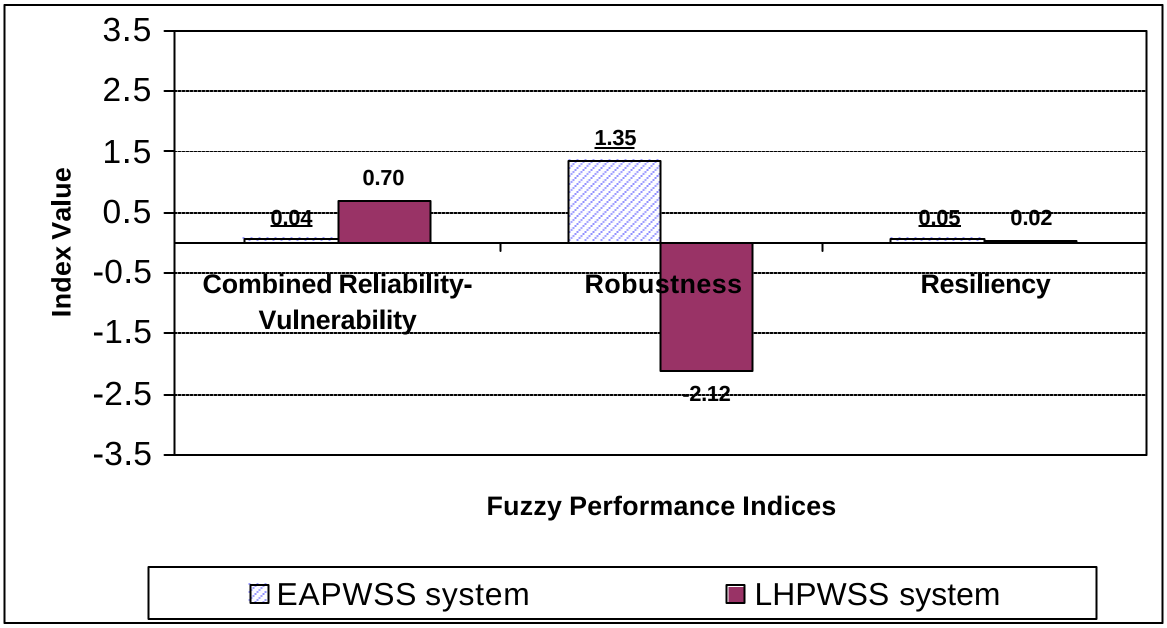

Figure (25a) Fuzzy performance indices for the LHPWSSS and EAPWSS systems for the triangular membership function shape.

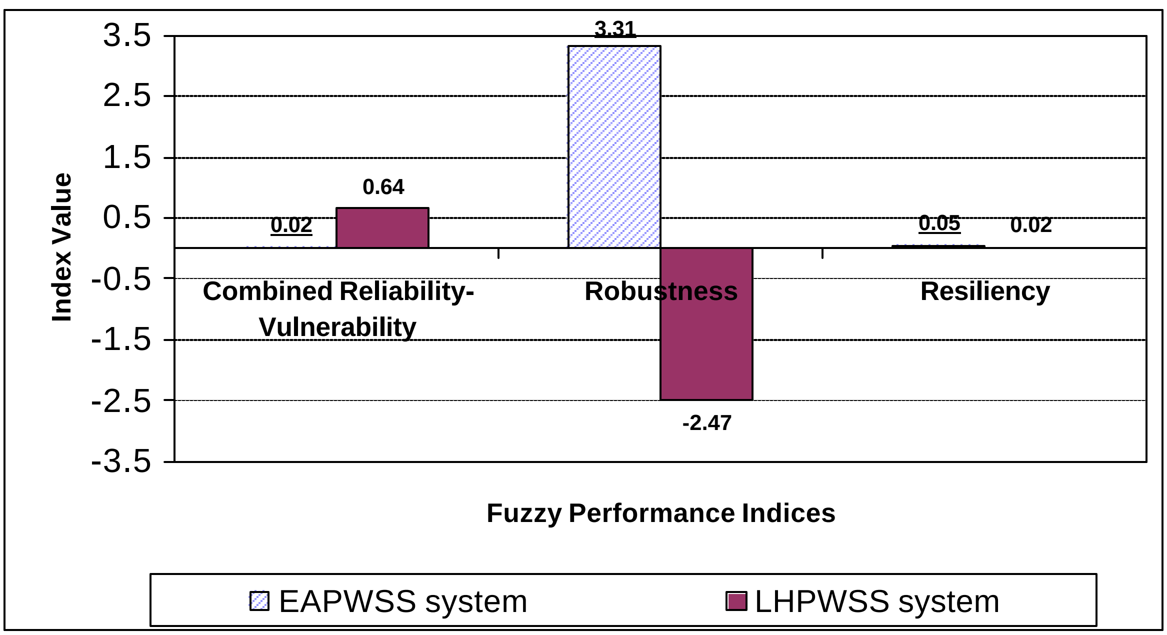

Figure (25b) Fuzzy performance indices for the LHPWSSS and EAPWSS systems for the trapezoidal membership function shape.

Additionally, the components of the LHPWSS system are designed with an overload capacity of 35% that positively affects the reliability of the system. As a general conclusion, the LHPWSS system is more reliable and less vulnerable to disruption in service than the EAPWSS system.

Robustness index value shows similar behavior for the triangular membership function shape. The difference in the robustness index values between the two systems is not as high as the difference in the combined reliability-vulnerability index. LHPWSS is more robust than EAPWSS for the two used shapes of the membership function. Therefore, EAPWSS system is more sensitive to any possible change in demand conditions than LHPWSS system as evident form the values of the robustness index of both systems.

The combined reliability-vulnerability index is highly sensitive to the shape of the membership function. Changing the membership function shape from the triangular to the trapezoidal, in both systems, results in a significant decrease in reliability. As an example in EAPWSS, the value of the combined reliability-vulnerability index decreases from 0.042 to 0.017 for trapezoidal shape. In case of robustness index, the change in the value is not as significant as in the case of the combined reliability-vulnerability index.

The recovery time for EAPWSS system components does not exceed 30 days. Some of the components in the LHPWSS system have a recovery time of more than 120 days. Therefore, the fuzzy resiliency index for the EAPWSS system is 4 times higher than for the LHPWSS system. However, the resiliency index is not sensitive to the shape of the membership function. This is due to the fact that the resiliency index value uses the center of gravity (COG) of the system-failure membership function, and the change in shapes does not affect the value of the COG and consequently the index value.

The developed calculation script can be used to identify critical system components. For example, the PAC components are found to be the critical components for both systems. Slight changes in their maximum capacity significantly affect system performance indices.

7 REFERENCE

American Water Services Canada-AWSC (2003a), Elgin Area Water Treatment Plant 2003 Compliance Report. Technical report, Joint Board of Management for the Elgin Area Primary Water Supply System, London, Ontario, Canada.

(http://www.watersupply.london.ca/Compliance_Reports/Elgin_Area_2003_Compliance_Report.pdf)
(accessed November, 2004)

American Water Services Canada-AWSC (2003b), Lake Huron Water Treatment Plant 2003 Compliance Report. Technical report, Joint Board of Management for the Lake Huron Primary Water Supply System, London, Ontario, Canada.

(http://www.watersupply.london.ca/Compliance_Reports/Huron_2003_Compliance_Report.pdf)
(accessed November, 2004)

Ang, H-S and H. Tang (1984), *Probability Concepts in Engineering Planning and Design*, John Wiley & Sons, Inc, USA.

DeSousa, L. and S. Simonovic (2003). Risk Assessment Study: Lake Huron and Elgin Primary Water Supply Systems. Technical report, University of Western Ontario, Facility for Intelligent Decision Support, London, Ontario, Canada.

El-Baroudy, I. and S. Simonovic (2003), New Fuzzy Performance Indices for Reliability Analysis of Water Supply Systems. Technical report, University of Western Ontario, Facility for Intelligent Decision Support, London, Ontario, Canada. (http://www.engga.uwo.ca/research/iclr/fids/Documents/Fuzzy_Preformance_Indices.pdf) (accessed November, 2004)

El-Baroudy, I. and S. Simonovic (2004), Fuzzy criteria for the evaluation of water resource systems performance. *Water Resource Research*, Vol. 40, No. 10.

Earth Tech Canada Inc. (2000), Engineers' Report: Elgin Area Primary Water Supply System. Earth Tech Canada Inc., London, Ontario, Canada.

Earth Tech Canada Inc. (2001), Engineers' Report: Lake Huron Primary Water Supply system. Earth Tech Canada Inc., London, Ontario, Canada.

Ganoulis, J. G. (1994), *Engineering Risk Analysis of Water Pollution: Probabilities*, VCH, Weinheim, The Netherlands.

Hashimoto, T., J. R. Stedinger and D. P. Loucks (1982a), "Reliability, Resiliency, and Vulnerability Criteria for Water Resources System Performance Evaluation", *Water Resources Research*, Vol. 18, No. 1, pp. 14-20.

Hashimoto, T., D. P. Loucks, and J. R. Stedinger (1982b), “Robustness of Water Resources Systems”, *Water Resources Research*, Vol. 18, No. 1, pp. 21-26.

Kaufmann, A. and M. Gupta (1985), *Introduction to Fuzzy Arithmetic: Theory and Applications*, Van Nostrand Reinhold Company Inc, New York, USA.

Simonovic, S. P. (1997), “Risk in sustainable Water Resources Management”, *Sustainability of Water Resources Under Increasing Uncertainty*, Proceedings of the Rabat Symposium S1, IAHS Publication, No. 240, pp.3-17.

APPENDICES

APPENDIX I: INPUT DATA

I-A LAKE HURON PRIMARY WATER SUPPLY SYSTEM

(LHPWSS)

	System	Units	Capacity			Requirment			Normalized Capacity (TRI)			NormalizedRequirment(Tri)			Normalized Capacity(Trap)			
			Average Daily flow	Design Capacity	Maximum Overload	Yearly Min	Yearly Avg	Yearly Max	Average Daily flow	Design Capacity	Maximum Overload	Yearly Min	Yearly Avg	Yearly Max	Average Daily flow	Capacity (1)	Capacity (2)	Maximum Overload
Intake Crib	Intake System	MLD	157.3	340.0	454.5	53.0	157.3	255.7	0.35	0.75	1.00	0.12	0.35	0.56	0.35	0.55	0.87	1.00
Chlorinator I		Kg/d	360.0	630.0	900.0	24.5	72.0	130.0	0.40	0.70	1.00	0.03	0.08	0.14	0.40	0.55	0.85	1.00
RC Intake Pipe		MLD	157.3	340.0	454.5	53.0	157.3	255.7	0.35	0.75	1.00	0.12	0.35	0.56	0.35	0.55	0.87	1.00
Travelling Screens	Low Lifting System	MLD	157.3	340.0	454.5	53.0	157.3	255.7	0.35	0.75	1.00	0.12	0.35	0.56	0.35	0.55	0.87	1.00
Pumping Wells		MLD	157.3	340.0	454.5	53.0	157.3	255.7	0.35	0.75	1.00	0.12	0.35	0.56	0.35	0.55	0.87	1.00
Chlorinator II		Kg/d	360.0	630.0	900.0	24.5	72.0	130.0	0.40	0.70	1.00	0.03	0.08	0.14	0.40	0.55	0.85	1.00
Single Speed Pump 1		MLD	49.9	75.0	100.0	16.8	49.0	81.2	0.50	0.75	1.00	0.17	0.49	0.81	0.50	0.62	0.87	1.00
Variable Speed Pump		MLD	57.4	86.2	115.0	19.3	49.0	93.4	0.50	0.75	1.00	0.17	0.43	0.81	0.50	0.62	0.87	1.00
Single Speed Pump 2		MLD	49.9	75.0	100.0	16.8	49.0	81.2	0.50	0.75	1.00	0.17	0.49	0.81	0.50	0.62	0.87	1.00
Single Speed Pump 1 (Back-up)		MLD	49.9	75.0	100.0	16.8	49.0	81.2	0.50	0.75	1.00	0.17	0.49	0.81	0.50	0.62	0.87	1.00
Variable Speed Pump (Back-up)		MLD	57.4	86.2	115.0	19.3	49.0	93.4	0.50	0.75	1.00	0.17	0.43	0.81	0.50	0.62	0.87	1.00
Single Speed Pump 2 (Back-up)		MLD	49.9	75.0	100.0	16.8	49.0	81.2	0.50	0.75	1.00	0.17	0.49	0.81	0.50	0.62	0.87	1.00
PAC Storage Tank	Flash Mix System	m^3	0.0	75.0	150.0	0.1	0.2	0.3	0.00	0.50	1.00	0.00	0.00	0.00	0.00	0.25	0.75	1.00
PAC Storage Tank		m^3	0.0	75.0	150.0	0.1	0.2	0.3	0.00	0.50	1.00	0.00	0.00	0.00	0.00	0.25	0.75	1.00
PAC Transfer Pump		Kg/d	60.0	180.0	300.0	227.0	419.7	667.3	0.20	0.60	1.00	0.76	1.40	2.22	0.20	0.40	0.80	1.00
PAC Transfer Pump		Kg/d	60.0	180.0	300.0	227.0	419.7	667.3	0.20	0.60	1.00	0.76	1.40	2.22	0.20	0.40	0.80	1.00
Alum Storage Tank		m^3	0.0	33.0	66.0	1.7	3.1	7.0	0.00	0.50	1.00	0.03	0.05	0.11	0.00	0.25	0.75	1.00
Alum Storage Tank		m^3	0.0	33.0	66.0	1.7	3.1	7.0	0.00	0.50	1.00	0.03	0.05	0.11	0.00	0.25	0.75	1.00
Alum Transfer Pump		m^3/d	3.3	4.8	6.7	1.7	3.1	7.0	0.50	0.73	1.00	0.25	0.46	1.05	0.50	0.61	0.86	1.00
Alum Transfer Pump		m^3/d	3.3	4.8	6.7	1.7	3.1	7.0	0.50	0.73	1.00	0.25	0.46	1.05	0.50	0.61	0.86	1.00
Flash Mix Cell I-1		MLD	78.7	170.0	227.3	26.5	77.2	127.9	0.35	0.75	1.00	0.12	0.34	0.56	0.35	0.55	0.87	1.00
Flash Mix Cell I-2		MLD	78.7	170.0	227.3	26.5	77.2	127.9	0.35	0.75	1.00	0.12	0.34	0.56	0.35	0.55	0.87	1.00
Polymer Storage Tank		m^3	0.0	6.6	13.6	0.0	4.4	8.8	0.00	0.50	1.00	0.00	0.33	0.65	0.00	0.25	0.75	1.00
Polymer Storage Tank		m^3	0.0	6.6	13.6	0.0	4.4	8.8	0.00	0.50	1.00	0.00	0.33	0.65	0.00	0.25	0.75	1.00
Polymer Transfer Pump		L/s	0.0	3.2	6.4	0.0	2.1	4.2	0.00	0.50	1.00	0.00	0.33	0.65	0.00	0.25	0.75	1.00
Polymer Transfer Pump		L/s	0.0	3.2	6.4	0.0	2.1	4.2	0.00	0.50	1.00	0.00	0.33	0.65	0.00	0.25	0.75	1.00
Polymer Mix Tank		m^3	0.0	3.4	6.8	0.0	2.2	4.4	0.00	0.50	1.00	0.00	0.33	0.65	0.00	0.25	0.75	1.00
Polymer Mix Tank		m^3	0.0	3.4	6.8	0.0	2.2	4.4	0.00	0.50	1.00	0.00	0.33	0.65	0.00	0.25	0.75	1.00
Polymer Feed Pump		L/s	0.1	0.4	0.6	0.1	0.2	0.4	0.17	0.58	1.00	0.11	0.38	0.65	0.17	0.38	0.79	1.00
Polymer Feed Pump		L/s	0.1	0.4	0.6	0.1	0.2	0.4	0.17	0.58	1.00	0.11	0.38	0.65	0.17	0.38	0.79	1.00
Flash Mix Cell II-1		MLD	78.7	170.0	227.3	26.5	77.2	127.9	0.35	0.75	1.00	0.12	0.34	0.56	0.35	0.55	0.87	1.00
Flash Mix Cell II-2		MLD	78.7	170.0	227.3	26.5	77.2	127.9	0.35	0.75	1.00	0.12	0.34	0.56	0.35	0.55	0.87	1.00
PAC Transfer Pump		Kg/d	60.0	180.0	300.0	227.0	419.7	667.3	0.20	0.60	1.00	0.76	1.40	2.22	0.20	0.40	0.80	1.00
Cell I-1	Flocculation System	MLD	8.0	25.3	42.5	6.6	19.3	32.0	0.19	0.59	1.00	0.16	0.45	0.75	0.19	0.39	0.80	1.00
Cell I-2		MLD	8.0	25.3	42.5	6.6	19.3	32.0	0.19	0.59	1.00	0.16	0.45	0.75	0.19	0.39	0.80	1.00
Cell II-1		MLD	8.0	25.3	42.5	6.6	19.3	32.0	0.19	0.59	1.00	0.16	0.45	0.75	0.19	0.39	0.80	1.00
Cell II-2		MLD	8.0	25.3	42.5	6.6	19.3	32.0	0.19	0.59	1.00	0.16	0.45	0.75	0.19	0.39	0.80	1.00
Cell III-1		MLD	8.0	25.3	42.5	6.6	19.3	32.0	0.19	0.59	1.00	0.16	0.45	0.75	0.19	0.39	0.80	1.00
Cell III-2		MLD	8.0	25.3	42.5	6.6	19.3	32.0	0.19	0.59	1.00	0.16	0.45	0.75	0.19	0.39	0.80	1.00
Cell IV-1		MLD	8.0	25.3	42.5	6.6	19.3	32.0	0.19	0.59	1.00	0.16	0.45	0.75	0.19	0.39	0.80	1.00
Cell IV-2		MLD	8.0	25.3	42.5	6.6	19.3	32.0	0.19	0.59	1.00	0.16	0.45	0.75	0.19	0.39	0.80	1.00
Settling Tank I-1	Sedimentation System	MLD	78.7	170.0	227.3	13.3	38.6	63.9	0.35	0.75	1.00	0.06	0.17	0.28	0.35	0.55	0.87	1.00
Settling Tank I-2		MLD	78.7	170.0	227.3	13.3	38.6	63.9	0.35	0.75	1.00	0.06	0.17	0.28	0.35	0.55	0.87	1.00
Settling Tank II-1		MLD	78.7	170.0	227.3	13.3	38.6	63.9	0.35	0.75	1.00	0.06	0.17	0.28	0.35	0.55	0.87	1.00
Settling Tank II-2		MLD	78.7	170.0	227.3	13.3	38.6	63.9	0.35	0.75	1.00	0.06	0.17	0.28	0.35	0.55	0.87	1.00
Filter I-1	Filtration System	MLD	13.1	28.3	37.9	4.4	12.9	21.3	0.35	0.75	1.00	0.12	0.34	0.56	0.35	0.55	0.87	1.00
Filter I-2		MLD	13.1	28.3	37.9	4.4	12.9	21.3	0.35	0.75	1.00	0.12	0.34	0.56	0.35	0.55	0.87	1.00
Filter II-1		MLD	13.1	28.3	37.9	4.4	12.9	21.3	0.35	0.75	1.00	0.12	0.34	0.56	0.35	0.55	0.87	1.00
Filter II-2		MLD	13.1	28.3	37.9	4.4	12.9	21.3	0.35	0.75	1.00	0.12	0.34	0.56	0.35	0.55	0.87	1.00
Filter III-1		MLD	13.1	28.3	37.9	4.4	12.9	21.3	0.35	0.75	1.00	0.12	0.34	0.56	0.35	0.55	0.87	1.00
Filter III-2		MLD	13.1	28.3	37.9	4.4	12.9	21.3	0.35	0.75	1.00	0.12	0.34	0.56	0.35	0.55	0.87	1.00
Filter IV-1		MLD	13.1	28.3	37.9	4.4	12.9	21.3	0.35	0.75	1.00	0.12	0.34	0.56	0.35	0.55	0.87	1.00
Filter IV-2		MLD	13.1	28.3	37.9	4.4	12.9	21.3	0.35	0.75	1.00	0.12	0.34	0.56	0.35	0.55	0.87	1.00
Filter V-2		MLD	13.1	28.3	37.9	4.4	12.9	21.3	0.35	0.75	1.00	0.12	0.34	0.56	0.35	0.55	0.87	1.00
Filter V-1		MLD	13.1	28.3	37.9	4.4	12.9	21.3	0.35	0.75	1.00	0.12	0.34	0.56	0.35	0.55	0.87	1.00
Filter VI-1		MLD	13.1	28.3	37.9	4.4	12.9	21.3	0.35	0.75	1.00	0.12	0.34	0.56	0.35	0.55	0.87	1.00
Filter VI-2		MLD	13.1	28.3	37.9	4.4	12.9	21.3	0.35	0.75	1.00	0.12	0.34	0.56	0.35	0.55	0.87	1.00
Chlorinator III		Kg/d	360.0	630.0	900.0	34.0	54.5	91.0	0.40	0.70	1.00	0.04	0.06	0.10	0.40	0.55	0.85	1.00
Chlorinator IV		Kg/d	360.0	630.0	900.0	34.0	54.5	91.0	0.40	0.70	1.00	0.04	0.06	0.10	0.40	0.55	0.85	1.00
Clear Well 1		MLD	52.4	105.9	159.4	4.4	12.9	21.3	0.33	0.66	1.00	0.03	0.08	0.13	0.33	0.50	0.83	1.00
Clear Well 2		MLD	52.4	105.9	159.4	4.4	12.9	21.3	0.33	0.66	1.00	0.03	0.08	0.13	0.33	0.50	0.83	1.00
Clear Well 3		MLD	52.4	105.9	159.4	4.4	12.9	21.3	0.33	0.66	1.00	0.03	0.08	0.13	0.33	0.50	0.83	1.00
Pump 1	High Lifting System	MLD	78.7	88.4	100.1	25.3	75.9	127.9	0.79	0.88	1.00	0.25	0.76	1.28	0.79	0.84	0.96	1.00
Pump 2		MLD	78.7	88.4	100.1	25.3	75.9	127.9	0.79	0.88	1.00	0.25	0.76	1.28	0.79	0.84	0.96	1.00
Pump 3		MLD	78.7	88.4	100.1	25.3	75.9	127.9	0.79	0.88	1.00	0.25	0.76	1.28	0.79	0.84	0.96	1.00
Pump 1 (Back-up)		MLD	78.7	88.4	100.1	25.3	75.9	127.9	0.79	0.88	1.00	0.25	0.76	1.28	0.79	0.84	0.96	1.00
Pump 2 (Back-up)		MLD	78.7	88.4	100.1	25.3	75.9	127.9	0.79	0.88	1.00	0.25	0.76	1.28	0.79	0.84	0.96	1.00
Main 1	Conveyance System	MLD	157.3	192.3	227.3	25.3	75.9	127.9	0.69	0.85	1.00	0.11	0.33	0.56	0.69	0.77	0.92	1.00
Main 2		MLD	157.3	192.3	227.3	25.3	75.9	127.9	0.69	0.85	1.00	0.11	0.33	0.56	0.69	0.77	0.92	1.00
Pump 1	Boosting System	MLD	4.6	52.3	100.1	4.6	52.3	100.1	0.05	0.52	1.00	0.05	0.52	1.00	0.05	0.28	0.76	1.00
Pump 2		MLD	4.6	52.3	100.1	4.6	52.3	100.1	0.05	0.52	1.00	0.05	0.52	1.00	0.05	0.28	0.76	1.00
Pump 3		MLD	4.6	52.3	100.1	4.6	52.3	100.1	0.05	0.52	1.00	0.05	0.52	1.00	0.05	0.28	0.76	1.00
Pump 4		MLD	4.6	52.3	100.1	4.6	52.3	100.1	0.05	0.52	1.00	0.05	0.52	1.00	0.05	0.28	0.76	1.00
Reservoir		MLD	0.0	9.1	18.2	0.0	9.1	18.2	0.00	0.50	1.00	0.00	0.50	1.00	0.00	0.25	0.75	1.00
Cell 1	Terminal Storage System	MLD	0.0	13.7	27.3	0.0	6.6	13.3	0.00	0.50	1.00	0.00	0.24	0.49	0.00	0.25	0.75	1.00
Cell 2		MLD	0.0	13.7	27.3	0.0	6.6	13.3	0.00	0.50	1.00	0.00	0.24	0.49	0.00	0.25	0.75	1.00
Cell 3		MLD	0.0	13.7	27.3	0.0	6.6	13.3	0.00	0.50	1.00	0.00	0.24	0.49	0.00	0.25	0.75	1.00
Cell 4		MLD	0.0	13.7	27.3	0.0	6.6	13.3	0.00	0.50	1.00	0.00	0.24	0.49	0.00	0.25	0.75	1.00

	System	Units	Normalized Requirment(Trap)				Element-State (Tri-MoS)			Element-State (Trap-MoS)				Element-Failure (Tri)			Element-Failure (Trap)			
			Average Daily flow	Capacity (1)	Capacity (2)	Maximum Overload	a	b	c	a	b	c	d	a	b	c	a	b	c	d
Intake Crib	Intake System	MLD	0.12	0.46	0.45	0.56	-0.22	0.40	0.88	-0.22	0.09	0.41	0.88	0.08	0.13	0.17	0.08	0.11	0.14	0.17
Chlorinator I		Kg/d	0.03	0.11	0.11	0.14	0.26	0.62	0.97	0.26	0.44	0.74	0.97	0.01	0.01	0.01	0.01	0.01	0.01	0.01
RC Intake Pipe		MLD	0.12	0.46	0.45	0.56	-0.22	0.40	0.88	-0.22	0.09	0.41	0.88	0.08	0.13	0.17	0.08	0.11	0.14	0.17
Travelling Screens	Low Lifting System	MLD	0.12	0.46	0.45	0.56	-0.22	0.40	0.88	-0.22	0.09	0.41	0.88	0.04	0.08	0.50	0.04	0.19	0.35	0.50
Pumping Wells		MLD	0.12	0.40	0.45	0.56	-0.22	0.40	0.88	-0.22	0.09	0.41	0.88	0.00	0.50	1.00	0.00	0.33	0.67	1.00
Chlorinator II		Kg/d	0.03	0.11	0.11	0.14	0.26	0.62	0.97	0.26	0.44	0.74	0.97	0.01	0.01	0.01	0.01	0.01	0.01	0.01
Single Speed Pump 1		MLD	0.17	0.65	0.65	0.81	-0.31	0.26	0.83	-0.31	-0.03	0.22	0.83	0.02	2.51	5.00	0.02	1.68	3.34	5.00
Variable Speed Pump		MLD	0.17	0.55	0.62	0.81	-0.31	0.32	0.83	-0.31	0.01	0.32	0.83	0.02	2.51	5.00	0.02	1.68	3.34	5.00
Single Speed Pump 2		MLD	0.17	0.65	0.65	0.81	-0.31	0.26	0.83	-0.31	-0.03	0.22	0.83	0.02	2.51	5.00	0.02	1.68	3.34	5.00
Single Speed Pump 1 (Back-up)		MLD	0.17	0.65	0.66	0.81	-0.31	0.26	0.83	-0.31	-0.03	0.22	0.83	0.02	2.51	5.00	0.02	1.68	3.34	5.00
Variable Speed Pump (Back-up)		MLD	0.17	0.56	0.62	0.81	-0.31	0.32	0.83	-0.31	0.01	0.32	0.83	0.02	2.51	5.00	0.02	1.69	3.34	5.00
Single Speed Pump 2 (Back-up)		MLD	0.17	0.65	0.65	0.81	-0.31	0.26	0.83	-0.31	-0.03	0.22	0.83	0.02	2.51	5.00	0.02	1.68	3.34	5.00
PAC Storage Tank	Flash Mix System	m^3	0.00	0.00	0.00	0.00	0.00	0.50	1.00	0.00	0.25	0.75	1.00	0.04	3.52	7.00	0.04	2.36	4.68	7.00
PAC Storage Tank		m^3	0.00	0.00	0.00	0.00	0.00	0.50	1.00	0.00	0.25	0.75	1.00	0.04	3.52	7.00	0.04	2.36	4.68	7.00
PAC Transfer Pump		Kg/d	0.75	1.72	1.81	2.22	-2.02	-0.80	0.24	-2.02	-1.41	-0.92	0.24	0.04	3.52	7.00	0.04	2.36	4.68	7.00
PAC Transfer Pump		Kg/d	0.76	1.73	1.81	2.22	-2.02	-0.80	0.24	-2.02	-1.41	-0.93	0.24	0.04	3.52	7.00	0.04	2.36	4.68	7.00
Alum Storage Tank		m^3	0.03	0.08	0.08	0.11	-0.11	0.46	0.97	-0.11	0.17	0.69	0.97	0.04	3.52	7.00	0.04	2.36	4.68	7.00
Alum Storage Tank		m^3	0.03	0.08	0.08	0.11	-0.11	0.46	0.97	-0.11	0.17	0.69	0.97	0.04	3.52	7.00	0.04	2.36	4.68	7.00
Alum Transfer Pump		m^3/d	0.25	0.57	0.76	1.06	-0.56	0.26	0.75	-0.56	-0.14	0.29	0.75	0.04	3.52	7.00	0.04	2.36	4.68	7.00
Alum Transfer Pump		m^3/d	0.25	0.57	0.76	1.06	-0.56	0.26	0.75	-0.56	-0.14	0.29	0.75	0.04	3.52	7.00	0.04	2.36	4.68	7.00
Flash Mix Cell I-1		MLD	0.12	0.45	0.45	0.56	-0.22	0.41	0.88	-0.22	0.10	0.42	0.88	0.50	7.25	14.00	0.50	5.00	9.50	14.00
Flash Mix Cell I-2		MLD	0.12	0.45	0.45	0.56	-0.22	0.41	0.88	-0.22	0.10	0.42	0.88	0.50	7.25	14.00	0.50	5.00	9.50	14.00
Polymer Storage Tank		m^3	0.00	0.49	0.49	0.66	-0.66	0.18	1.00	-0.66	-0.24	0.26	1.00	0.50	7.25	14.00	0.50	5.00	9.50	14.00
Polymer Storage Tank		m^3	0.00	0.49	0.49	0.66	-0.66	0.18	1.00	-0.66	-0.24	0.26	1.00	0.50	7.25	14.00	0.50	5.00	9.50	14.00
Polymer Transfer Pump		L/s	0.00	0.49	0.49	0.65	-0.65	0.18	1.00	-0.65	-0.24	0.26	1.00	0.50	7.25	14.00	0.50	5.00	9.50	14.00
Polymer Transfer Pump		L/s	0.00	0.49	0.49	0.65	-0.66	0.18	1.00	-0.66	-0.24	0.26	1.00	0.50	7.25	14.00	0.50	5.00	9.50	14.00
Polymer Mix Tank		m^3	0.00	0.49	0.49	0.66	-0.66	0.18	1.00	-0.66	-0.24	0.26	1.00	0.50	7.25	14.00	0.50	5.00	9.50	14.00
Polymer Mix Tank		m^3	0.00	0.49	0.49	0.66	-0.66	0.18	1.00	-0.66	-0.24	0.26	1.00	0.50	7.25	14.00	0.50	5.00	9.50	14.00
Polymer Feed Pump		L/s	0.11	0.51	0.51	0.65	-0.48	0.20	0.89	-0.48	-0.14	0.26	0.89	0.50	7.25	14.00	0.50	5.00	9.50	14.00
Polymer Feed Pump		L/s	0.11	0.51	0.51	0.65	-0.48	0.20	0.89	-0.48	-0.14	0.26	0.89	0.50	7.25	14.00	0.50	5.00	9.50	14.00
Flash Mix Cell II-1		MLD	0.12	0.45	0.45	0.56	-0.22	0.41	0.88	-0.22	0.10	0.42	0.88	0.50	7.25	14.00	0.50	5.00	9.50	14.00
Flash Mix Cell II-2		MLD	0.12	0.45	0.45	0.56	-0.22	0.41	0.88	-0.22	0.10	0.42	0.88	0.50	7.25	14.00	0.50	5.00	9.50	14.00
PAC Transfer Pump		Kg/d	0.75	1.72	1.81	2.22	-2.02	-0.80	0.24	-2.02	-1.41	-0.92	0.24	0.04	3.52	7.00	0.04	2.36	4.68	7.00
Cell I-1	Flocculation System	MLD	0.16	0.60	0.60	0.75	-0.56	0.14	0.64	-0.56	-0.21	0.19	0.84	0.00	0.50	1.00	0.00	0.33	0.67	1.00
Cell I-2		MLD	0.16	0.60	0.60	0.75	-0.56	0.14	0.64	-0.56	-0.21	0.19	0.84	0.00	0.50	1.00	0.00	0.33	0.67	1.00
Cell II-1		MLD	0.16	0.60	0.60	0.75	-0.56	0.14	0.64	-0.56	-0.21	0.19	0.84	0.00	0.50	1.00	0.00	0.33	0.67	1.00
Cell II-2		MLD	0.15	0.60	0.60	0.75	-0.56	0.14	0.64	-0.56	-0.21	0.19	0.84	0.00	0.50	1.00	0.00	0.33	0.67	1.00
Cell III-1		MLD	0.16	0.60	0.60	0.75	-0.56	0.14	0.64	-0.56	-0.21	0.19	0.84	0.00	0.50	1.00	0.00	0.33	0.67	1.00
Cell III-2		MLD	0.16	0.60	0.60	0.75	-0.56	0.14	0.64	-0.56	-0.21	0.19	0.84	0.00	0.50	1.00	0.00	0.33	0.67	1.00
Cell IV-1		MLD	0.16	0.60	0.60	0.75	-0.56	0.14	0.64	-0.56	-0.21	0.19	0.84	0.00	0.50	1.00	0.00	0.33	0.67	1.00
Cell IV-2		MLD	0.16	0.60	0.60	0.75	-0.56	0.14	0.64	-0.56	-0.21	0.19	0.84	0.00	0.50	1.00	0.00	0.33	0.67	1.00
Settling Tank I-1	Sedimentation System	MLD	0.06	0.23	0.23	0.28	0.06	0.58	0.94	0.06	0.32	0.65	0.94	0.08	0.54	1.00	0.08	0.39	0.69	1.00
Settling Tank I-2		MLD	0.06	0.23	0.23	0.28	0.06	0.58	0.94	0.06	0.32	0.65	0.94	0.08	0.54	1.00	0.08	0.39	0.69	1.00
Settling Tank II-1		MLD	0.06	0.23	0.23	0.28	0.06	0.58	0.94	0.06	0.32	0.65	0.94	0.08	0.54	1.00	0.08	0.39	0.69	1.00
Settling Tank II-2		MLD	0.06	0.23	0.23	0.28	0.06	0.58	0.94	0.06	0.32	0.65	0.94	0.08	0.54	1.00	0.08	0.39	0.69	1.00
Filter I-1	Filtration System	MLD	0.12	0.45	0.45	0.56	-0.22	0.41	0.88	-0.22	0.10	0.42	0.88	0.04	5.02	10.00	0.04	3.36	6.68	10.00
Filter I-2		MLD	0.12	0.45	0.45	0.56	-0.22	0.41	0.88	-0.22	0.10	0.42	0.88	0.04	5.02	10.00	0.04	3.36	6.68	10.00
Filter II-1		MLD	0.12	0.45	0.45	0.56	-0.22	0.41	0.88	-0.22	0.10	0.42	0.88	0.04	5.02	10.00	0.04	3.36	6.68	10.00
Filter II-2		MLD	0.12	0.45	0.45	0.56	-0.22	0.41	0.88	-0.22	0.10	0.42	0.88	0.04	5.02	10.00	0.04	3.36	6.68	10.00
Filter III-1		MLD	0.12	0.45	0.45	0.56	-0.22	0.41	0.88	-0.22	0.10	0.42	0.88	0.04	5.02	10.00	0.04	3.36	6.68	10.00
Filter III-2		MLD	0.12	0.45	0.45	0.56	-0.22	0.41	0.88	-0.22	0.10	0.42	0.88	0.04	5.02	10.00	0.04	3.36	6.68	10.00
Filter IV-1		MLD	0.12	0.45	0.45	0.56	-0.22	0.41	0.88	-0.22	0.10	0.42	0.88	0.04	5.02	10.00	0.04	3.36	6.68	10.00
Filter IV-2		MLD	0.12	0.45	0.45	0.56	-0.22	0.41	0.88	-0.22	0.10	0.42	0.88	0.04	5.02	10.00	0.04	3.36	6.68	10.00
Filter V-2		MLD	0.12	0.45	0.45	0.56	-0.22	0.41	0.88	-0.22	0.10	0.42	0.88	0.04	5.02	10.00	0.04	3.36	6.68	10.00
Filter V-1		MLD	0.12	0.45	0.45	0.56	-0.22	0.41	0.88	-0.22	0.10	0.42	0.88	0.04	5.02	10.00	0.04	3.36	6.68	10.00
Filter VI-1		MLD	0.12	0.45	0.45	0.56	-0.22	0.41	0.88	-0.22	0.10	0.42	0.88	0.04	5.02	10.00	0.04	3.36	6.68	10.00
Filter VI-2		MLD	0.12	0.45	0.45	0.56	-0.22	0.41	0.88	-0.22	0.10	0.42	0.88	0.04	5.02	10.00	0.04	3.36	6.68	10.00
Chlorinator III		Kg/d	0.04	0.07	0.08	0.10	0.30	0.64	0.96	0.30	0.47	0.78	0.96	0.01	0.01	0.01	0.01	0.01	0.01	0.01
Chlorinator IV		Kg/d	0.04	0.07	0.08	0.10	0.30	0.64	0.96	0.30	0.47	0.78	0.96	0.01	0.01	0.01	0.01	0.01	0.01	0.01
Clear Well 1		MLD	0.03	0.11	0.11	0.13	0.20	0.58	0.97	0.20	0.39	0.73	0.97	0.00	0.50	1.00	0.00	0.33	0.67	1.00
Clear Well 2		MLD	0.03	0.11	0.11	0.13	0.20	0.58	0.97	0.20	0.39	0.73	0.97	0.00	0.50	1.00	0.00	0.33	0.67	1.00
Clear Well 3		MLD	0.03	0.11	0.11	0.13	0.20	0.58	0.97	0.20	0.39	0.73	0.97	0.00	0.50	1.00	0.00	0.33	0.67	1.00
Pump 1	High Lifting System	MLD	0.25	1.01	1.02	1.28	-0.49	0.13	0.75	-0.49	-0.18	-0.07	0.75	0.04	60.02	120.00	0.04	40.03	80.01	120.00
Pump 2		MLD	0.25	1.01	1.02	1.28	-0.49	0.13	0.75	-0.49	-0.18	-0.07	0.75	0.04	60.02	120.00	0.04	40.03	80.01	120.00
Pump 3		MLD	0.25	1.01	1.02	1.28	-0.49	0.13	0.75	-0.49	-0.18	-0.07	0.75	0.04	60.02	120.00	0.04	40.03	80.01	120.00
Pump 1 (Back-up)		MLD	0.25	1.01	1.02	1.28	-0.49	0.13	0.75	-0.49	-0.18	-0.07	0.75	0.04	60.02	120.00	0.04	40.03	80.01	120.00
Pump 2 (Back-up)		MLD	0.25	1.01	1.02	1.28	-0.49	0.13	0.75	-0.49	-0.18	-0.07	0.75	0.04	60.02	120.00	0.04	40.03	80.01	120.00
Main 1	Conveyance System	MLD	0.11	0.45	0.45	0.56	0.13	0.51	0.89	0.13	0.32	0.48	0.89	0.02	1.51	3.00	0.02	1.01	2.01	3.00
Main 2		MLD	0.11	0.46	0.45	0.56	0.13	0.51	0.91	0.13	0.32	0.48	0.89	0.02	1.51	3.01	0.02	1.01	2.01	3.00
Pump 1	Boosting System	MLD	0.05	0.76	0.76	1.00	-0.95	0.00	0.95	-0.95	-0.48	0.00	0.95	0.08	60.04	120.00	0.08	40.06	80.03	120.00
Pump 2		MLD	0.05	0.76	0.76	1.00	-0.95	0.00	0.95	-0.95	-0.48	0.00	0.95	0.08	60.04	120.00	0.08	40.06	80.03	120.00
Pump 3		MLD	0.05	0.76	0.76	1.00	-0.95	0.00	0.95	-0.95	-0.48	0.00	0.95	0.08	60.04	120.00	0.08	40.06	80.03	120.00
Pump 4		MLD	0.05	0.76	0.76	1.00	-0.95	0.00	0.95	-0.95	-0.48	0.00	0.95	0.08	60.04	120.00	0.08	40.06	80.03	120.00
Reservoir		MLD	0.00	0.75	0.75	1.00	-1.00	0.00	1.00	-1.00	-0.50	0.00	1.00	0.08	60.04	120.00	0.08	40.06	80.03	120.00
Cell 1	Terminal Storage System	MLD	0.00	0.36	0.36	0.48	-0.48	0.26	1.00	-0.48	-0.11	0.39	1.00	0.08	0.54	1.00	0.08	0.39	0.69	1.00
Cell 2		MLD	0.00	0.36	0.36	0.48	-0.48	0.26	1.00	-0.48	-0.11	0.39	1.00	0.08	0.54	1.00	0.08	0.39	0.69	1.00
Cell 3		MLD	0.00	0.36	0.36	0.48	-0.48	0.26	1.00	-0.48	-0.11	0.39	1.00	0.08	0.54	1.00	0.08	0.39	0.69	1.00
Cell 4		MLD	0.00	0.36	0.36	0.48	-0.48	0.26	1.00	-0.48	-0.11	0.39	1.00	0.08	0.54	1.00	0.08	0.39	0.69	1.00

APPENDIX I: INPUT DATA

I-B ELGIN AREA PRIMARY WATER SUPPLY SYSTEM

(EAPWSS)

System		Element ID				No.	Units	Capacity			Requirment			Normalized Capacity (TRI)			NormalizedRequirment(Tri)			Normalized Capacity(Trap)			
		System		Element				Average Daily flow	Design Capacity	Maximum Overload	Yearly Min	Yearly Avg	Yearly Max	Average Daily flow	Design Capacity	Maximum Overload	Yearly Min	Yearly Aug	Yearly Max	Average Daily flow	Capacity (1)	Capacity (2)	Maximum Overload
Intake Crib	Intake System	0	1	0	1	1	MLD	52.4	91.0	181.0	35.3	52.4	71.8	0.29	0.50	1.00	0.19	0.29	0.40	0.29	0.40	0.75	1.00
Sodium Hypochlorite		0	1	0	2	2	MLD	52.4	91.0	181.0	35.3	52.4	71.8	0.29	0.50	1.00	0.19	0.29	0.40	0.29	0.40	0.75	1.00
RC Intake Pipe		0	1	0	3	3	MLD	52.4	91.0	181.0	35.3	52.4	71.8	0.29	0.50	1.00	0.19	0.29	0.40	0.29	0.40	0.75	1.00
Plant Drain Intake Pipe		0	1	0	4	4	MLD	52.4	91.0	181.0	35.3	52.4	71.8	0.29	0.50	1.00	0.19	0.29	0.40	0.29	0.40	0.75	1.00
Traveling Screens	Low Lifting System	0	2	0	1	5	MLD	36.2	45.5	90.5	35.3	52.4	71.8	0.29	0.50	1.00	0.39	0.58	0.79	0.29	0.40	0.75	1.00
Traveling Screens		0	2	0	2	6	MLD	36.2	45.5	90.5	35.3	52.4	71.8	0.29	0.50	1.00	0.39	0.58	0.79	0.29	0.40	0.75	1.00
Pumping Well 1		0	2	0	3	7	MLD	36.2	45.5	90.5	35.3	52.4	71.8	0.29	0.50	1.00	0.39	0.58	0.79	0.29	0.40	0.75	1.00
Pumping Well 2		0	2	0	4	8	MLD	36.2	45.5	90.5	35.3	52.4	71.8	0.29	0.50	1.00	0.39	0.58	0.79	0.29	0.40	0.75	1.00
High Discharge Pump 1		0	2	0	5	9	MLD	19.9	34.6	67.0	13.4	19.9	27.3	0.30	0.52	1.00	0.20	0.30	0.41	0.30	0.41	0.76	1.00
Low Discharge Pump 1		0	2	0	6	10	MLD	6.2	10.9	21.0	4.2	6.2	8.6	0.30	0.52	1.00	0.20	0.30	0.41	0.30	0.41	0.76	1.00
High Discharge Pump 2		0	2	0	7	11	MLD	19.9	34.6	67.0	13.4	19.9	27.3	0.30	0.52	1.00	0.20	0.30	0.41	0.30	0.41	0.76	1.00
Low Discharge Pump 2		0	2	0	8	12	MLD	6.2	10.9	21.0	4.2	6.2	8.6	0.30	0.52	1.00	0.20	0.30	0.41	0.30	0.41	0.76	1.00
RC Intake Pipe		0	2	0	9	13	MLD	52.4	91.0	181.0	35.3	52.4	71.8	0.29	0.50	1.00	0.19	0.29	0.40	0.29	0.40	0.75	1.00
PAC Storage 1	Flash Mix System	0	3	0	1	14	m^3	0.0	85.5	171.0	0.0	0.1	0.2	0.00	0.50	1.00	0.00	0.00	0.00	0.00	0.25	0.75	1.00
PAC Storage 2		0	3	0	2	15	m^3	0.0	85.5	171.0	0.0	0.1	0.2	0.00	0.50	1.00	0.00	0.00	0.00	0.00	0.25	0.75	1.00
PAC Transfer Pump 1		0	3	0	3	16	L/s	0.0	3.2	6.3	0.0	0.0	0.0	0.00	0.50	1.00	0.00	0.00	0.00	0.00	0.25	0.75	1.00
PAC Transfer Pump 2		0	3	0	4	17	L/s	0.0	3.2	6.3	0.0	0.0	0.0	0.00	0.50	1.00	0.00	0.00	0.00	0.00	0.25	0.75	1.00
PAC Day Tank		0	3	0	5	18	m^3	0.0	1.5	3.0	0.0	0.1	0.2	0.00	0.50	1.00	0.00	0.04	0.07	0.00	0.25	0.75	1.00
PAC metering Pump 2		0	3	0	6	19	L/s	0.0	316.5	633.0	0.0	0.0	0.1	0.00	0.50	1.00	0.00	0.00	0.00	0.00	0.25	0.75	1.00
PAC metering Pump 1		0	3	0	7	20	L/s	0.0	316.5	633.0	0.0	0.0	0.1	0.00	0.50	1.00	0.00	0.00	0.00	0.00	0.25	0.75	1.00
PAC metering Pump 3		0	3	0	8	21	L/s	0.0	77.5	155.0	0.0	0.0	0.1	0.00	0.50	1.00	0.00	0.00	0.00	0.00	0.25	0.75	1.00
PAC metering Pump 4		0	3	0	9	22	L/s	0.0	77.5	155.0	0.0	0.0	0.1	0.00	0.50	1.00	0.00	0.00	0.00	0.00	0.25	0.75	1.00
Alum Storage 2		0	3	1	0	23	m^3	0.0	9.5	18.9	0.8	1.7	2.6	0.00	0.50	1.00	0.05	0.09	0.13	0.00	0.25	0.75	1.00
Alum Storage 1		0	3	1	1	24	m^3	0.0	9.5	18.9	0.8	1.7	2.6	0.00	0.50	1.00	0.05	0.09	0.13	0.00	0.25	0.75	1.00
Alum Transfer Pump		0	3	1	2	25	L/s	0.0	75.0	150.0	0.0	0.0	0.1	0.00	0.50	1.00	0.00	0.00	0.00	0.00	0.25	0.75	1.00
Alum Day Tank		0	3	1	3	26	m^3	0.0	0.9	1.8	0.0	0.0	0.1	0.00	0.50	1.00	0.01	0.02	0.08	0.00	0.25	0.75	1.00
Alum metering Pump 2		0	2	1	4	27	L/s	0.0	60.0	120.0	9.7	34.7	123.6	0.00	0.50	1.00	0.08	0.29	1.03	0.00	0.25	0.75	1.00
Alum metering Pump 1		0	3	1	5	28	L/s	0.0	60.0	120.0	9.7	34.7	123.6	0.00	0.50	1.00	0.08	0.29	1.03	0.00	0.25	0.75	1.00
Alum metering Pump 4		0	3	1	6	29	L/s	0.0	30.0	60.0	9.7	34.7	123.6	0.00	0.50	1.00	0.16	0.58	2.06	0.00	0.25	0.75	1.00
Alum metering Pump 3		0	3	1	7	30	L/s	0.0	30.0	60.0	9.7	34.7	123.6	0.00	0.50	1.00	0.16	0.58	2.06	0.00	0.25	0.75	1.00
Flash Mix Cell II		0	3	1	8	31	MLD	0.0	26.9	53.7	17.6	26.2	35.9	0.00	0.50	1.00	0.33	0.49	0.67	0.00	0.25	0.75	1.00
Flash Mix Cell I		0	3	1	9	32	MLD	0.0	26.9	53.7	17.6	26.2	35.9	0.00	0.50	1.00	0.33	0.49	0.67	0.00	0.25	0.75	1.00
Polymer Mix tank		0	3	2	0	33	L	0.0	142.6	285.0	0.0	39.9	79.8	0.00	0.50	1.00	0.00	0.14	0.28	0.00	0.25	0.75	1.00
Polymer Day Tank		0	3	2	1	34	L	0.0	285.0	570.0	0.0	39.9	79.8	0.00	0.50	1.00	0.00	0.07	0.14	0.00	0.25	0.75	1.00
Polymer metering Pump 1		0	3	2	2	35	L/min/Head	0.0	0.6	1.2	0.0	0.2	0.3	0.00	0.50	1.00	0.00	0.14	0.28	0.00	0.25	0.75	1.00
Polymer metering Pump 2		0	3	2	3	36	L/min/Head	0.0	1.0	1.9	0.0	0.3	0.5	0.00	0.50	1.00	0.00	0.14	0.28	0.00	0.25	0.75	1.00
Polymer metering Pump 3		0	3	2	4	37	L/min/Head	0.0	0.6	1.2	0.0	0.2	0.3	0.00	0.50	1.00	0.00	0.14	0.28	0.00	0.25	0.75	1.00
Polymer metering Pump 4		0	3	2	5	38	L/min/Head	0.0	1.0	1.9	0.0	0.3	0.5	0.00	0.50	1.00	0.00	0.14	0.28	0.00	0.25	0.75	1.00
Alum Transfer Pump (back-up)		0	3	2	6	39	L/s	0.0	75.0	150.0	0.0	0.0	0.1	0.00	0.50	1.00	0.00	0.00	0.00	0.00	0.25	0.75	1.00
Alum Metering Pump 1(back-up)		0	2	2	7	40	L/s	0.0	316.5	633.0	0.0	0.0	0.1	0.00	0.50	1.00	0.00	0.00	0.00	0.00	0.25	0.75	1.00
Alum Metering Pump 2(back-up)		0	2	2	8	41	L/s	0.0	316.5	633.0	0.0	0.0	0.1	0.00	0.50	1.00	0.00	0.00	0.00	0.00	0.25	0.75	1.00
Cell#1	Flocculation System	0	4	0	1	42	MLD	0.0	45.5	91.0	8.8	13.1	17.9	0.00	0.50	1.00	0.10	0.14	0.20	0.00	0.25	0.75	1.00
Cell#-1		0	4	0	2	43	MLD	0.0	45.5	91.0	8.8	13.1	17.9	0.00	0.50	1.00	0.10	0.14	0.20	0.00	0.25	0.75	1.00
Cell#2		0	4	0	3	44	MLD	0.0	45.5	91.0	8.8	13.1	17.9	0.00	0.50	1.00	0.10	0.14	0.20	0.00	0.25	0.75	1.00
Cell#2		0	4	0	4	45	MLD	0.0	45.5	91.0	8.8	13.1	17.9	0.00	0.50	1.00	0.10	0.14	0.20	0.00	0.25	0.75	1.00
Sedimentation Tank I	Sedimentation System	0	5	0	1	46	MLD	4.5	26.0	45.4	17.6	26.2	35.9	0.10	0.55	1.00	0.39	0.58	0.79	0.10	0.33	0.78	1.00
Sedimentation Tank II		0	5	0	2	47	MLD	4.5	26.0	45.4	17.6	26.2	35.9	0.10	0.55	1.00	0.39	0.58	0.79	0.10	0.33	0.78	1.00
Pre-Chlorinator		0	5	0	3	48	Kg/d	0.0	125.0	250.0	0.0	32.0	79.0	0.00	0.50	1.00	0.00	0.13	0.32	0.00	0.25	0.75	1.00
Fluoride Storage 2		0	5	0	4	49	m^3	0.0	4.5	9.0	0.0	0.0	0.0	0.00	0.50	1.00	0.00	0.00	0.00	0.00	0.25	0.75	1.00
Fluoride Storage 1		0	5	0	5	50	m^3	0.0	4.5	9.0	0.0	0.0	0.0	0.00	0.50	1.00	0.00	0.00	0.00	0.00	0.25	0.75	1.00
Fluoride Feed Pump		0	5	0	6	51	L/s	0.0	7.4	14.8	0.0	2.5	5.0	0.00	0.50	1.00	0.00	0.17	0.34	0.00	0.25	0.75	1.00
Fluoride Feed Pump (back-up)		0	5	0	7	52	L/s	0.0	7.4	14.8	0.0	2.5	5.0	0.00	0.50	1.00	0.00	0.17	0.34	0.00	0.25	0.75	1.00
Filter 1	Filtration System	0	6	0	1	53	MLD	0.0	15.0	30.0	8.8	13.1	17.9	0.00	0.50	1.00	0.29	0.44	0.60	0.00	0.25	0.75	1.00
Filter 2		0	6	0	2	54	MLD	0.0	15.0	30.0	8.8	13.1	17.9	0.00	0.50	1.00	0.29	0.44	0.60	0.00	0.25	0.75	1.00
Filter 3		0	6	0	3	55	MLD	0.0	15.0	30.0	8.8	13.1	17.9	0.00	0.50	1.00	0.29	0.44	0.60	0.00	0.25	0.75	1.00
Filter 4		0	6	0	4	56	MLD	0.0	15.0	30.0	8.8	13.1	17.9	0.00	0.50	1.00	0.29	0.44	0.60	0.00	0.25	0.75	1.00
Clear Well		0	6	0	5	57	ML	0.0	45.7	91.5	35.3	52.4	71.8	0.00	0.50	1.00	0.39	0.57	0.79	0.00	0.25	0.75	1.00
Onsite Reservoir		0	6	0	6	58	ML	0.0	45.4	90.8	35.3	52.4	71.8	0.00	0.50	1.00	0.39	0.58	0.79	0.00	0.25	0.75	1.00
Post-Chlorinator	High Lifting System	0	7	0	1	59	Kg/d	0.0	125.0	250.0	5.0	30.0	79.0	0.00	0.50	1.00	0.04	0.12	0.32	0.00	0.25	0.75	1.00
Pump 3		0	7	0	2	60	MLD	28.0	40.0	52.0	8.8	13.1	17.9	0.54	0.77	1.00	0.17	0.25	0.34	0.54	0.66	0.88	1.00
Pump 1		0	7	0	3	61	MLD	28.0	40.0	52.0	8.8	13.1	17.9	0.54	0.77	1.00	0.17	0.25	0.34	0.54	0.66	0.88	1.00
Pump 4		0	7	0	4	62	MLD	28.0	40.0	52.0	8.8	13.1	17.9	0.54	0.77	1.00	0.17	0.25	0.34	0.54	0.66	0.88	1.00
Pump 2		0	7	0	5	63	MLD	28.0	40.0	52.0	8.8	13.1	17.9	0.54	0.77	1.00	0.17	0.25	0.34	0.54	0.66	0.88	1.00
	Conveyance System	0	8					52.4	82.0	181.0	35.3	52.4	71.8	0.29	0.46	1.00	0.19	0.29	0.40	0.29	0.37	0.73	1.00
Main				0	1	64	MLD																
Terminal Reservoir 1	Boosting System	0	9	0	1	65	MLD	0.0	13.7	27.3	0.0	13.1	26.2	0.00	0.50	1.00	0.00	0.48	0.96	0.00	0.25	0.75	1.00
Terminal Reservoir 2		0	9	0	2	66	MLD	0.0	13.7	27.3	0.0	13.1	26.2	0.00	0.50	1.00	0.00	0.48	0.96	0.00	0.25	0.75	1.00

System		Element ID				No.	Units	Normalized Requirment(Trap)				Element-State (Tri-MoS)			Element-State (Trap-MoS)				Element-Failure (Tri)			Element-Failure (Trap)			
		System		Element				Average Daily Flow	Capacity (1)	Capacity (2)	Maximum Overload	a	b	c	a	b	c	d	a	b	c	a	b	c	d
Intake Crib	Intake System	0	1	0	1	1	MLD	0.19	0.34	0.34	0.40	-0.11	0.21	0.81	-0.11	0.05	0.42	0.81	0.08	0.13	0.17	0.08	0.11	0.14	0.17
Sodium Hypochlorite		0	1	0	2	2	MLD	0.19	0.34	0.34	0.40	-0.11	0.21	0.81	-0.11	0.05	0.42	0.81	0.04	3.52	7.00	0.04	2.36	4.68	7.00
RC Intake Pipe		0	1	0	3	3	MLD	0.19	0.34	0.34	0.40	-0.11	0.21	0.81	-0.11	0.05	0.42	0.81	0.08	0.13	0.17	0.08	0.11	0.14	0.17
Plant Davis Intake Pipe		0	1	0	4	4	MLD	0.19	0.34	0.34	0.40	-0.11	0.21	0.81	-0.11	0.05	0.42	0.81	0.08	0.13	0.17	0.08	0.11	0.14	0.17
Traveling Screens	Low Lifting System	0	2	0	1	5	MLD	0.39	0.67	0.69	0.79	-0.50	-0.08	0.61	-0.50	-0.29	0.08	0.61	0.04	0.27	0.50	0.04	0.19	0.36	0.50
Traveling Screens		0	2	0	2	6	MLD	0.39	0.67	0.69	0.79	-0.50	-0.08	0.61	-0.50	-0.29	0.08	0.61	0.04	0.27	0.50	0.04	0.19	0.36	0.50
Pumping Well 1		0	2	0	3	7	MLD	0.39	0.67	0.69	0.79	-0.50	-0.08	0.61	-0.50	-0.29	0.08	0.61	0.04	0.27	0.50	0.04	0.19	0.36	0.50
Pumping Well 2		0	2	0	4	8	MLD	0.39	0.67	0.69	0.79	-0.50	-0.08	0.61	-0.50	-0.29	0.08	0.61	0.04	0.27	0.50	0.04	0.19	0.36	0.50
High Discharge Pump 1		0	2	0	5	9	MLD	0.30	0.35	0.35	0.41	-0.11	0.22	0.80	-0.11	0.05	0.41	0.80	0.04	10.52	21.00	0.04	7.03	14.01	21.00
Low Discharge Pump 1		0	2	0	6	10	MLD	0.30	0.35	0.35	0.41	-0.11	0.22	0.80	-0.11	0.05	0.41	0.80	0.04	10.52	21.00	0.04	7.03	14.01	21.00
High Discharge Pump 2		0	2	0	7	11	MLD	0.30	0.35	0.35	0.41	-0.11	0.22	0.80	-0.11	0.05	0.41	0.80	0.04	10.52	21.00	0.04	7.03	14.01	21.00
Low Discharge Pump 2		0	2	0	8	12	MLD	0.30	0.35	0.35	0.41	-0.11	0.22	0.80	-0.11	0.05	0.41	0.80	0.04	10.52	21.00	0.04	7.03	14.01	21.00
RC Intake Pipe		0	2	0	9	13	MLD	0.19	0.34	0.34	0.40	-0.11	0.21	0.81	-0.11	0.05	0.42	0.81	0.04	10.52	21.00	0.04	7.03	14.01	21.00
PAC Storage 1	Flash Mix System	0	3	0	1	14	m^3	0.00	0.00	0.00	0.00	0.00	0.50	1.00	0.00	0.25	0.75	1.00	0.08	2.54	6.00	0.08	1.72	3.36	6.00
PAC Storage 2		0	3	0	2	15	m^3	0.00	0.00	0.00	0.00	0.00	0.50	1.00	0.00	0.25	0.75	1.00	0.08	2.54	6.00	0.08	1.72	3.36	6.00
PAC Transfer Pump 1		0	3	0	3	16	L/s	0.00	0.00	0.00	0.00	0.00	0.50	1.00	0.00	0.25	0.75	1.00	0.08	2.54	6.00	0.08	1.72	3.36	6.00
PAC Transfer Pump 2		0	3	0	4	17	L/s	0.00	0.00	0.00	0.00	0.00	0.50	1.00	0.00	0.25	0.75	1.00	0.08	2.54	6.00	0.08	1.72	3.36	6.00
PAC Day Tank		0	3	0	5	18	m^3	0.00	0.06	0.06	0.07	-0.07	0.46	1.00	-0.07	0.19	0.69	1.00	0.08	2.54	6.00	0.08	1.72	3.36	6.00
PAC metering Pump 2		0	3	0	6	19	L/h	0.00	0.00	0.00	0.00	0.00	0.50	1.00	0.00	0.25	0.75	1.00	0.08	2.54	6.00	0.08	1.72	3.36	6.00
PAC metering Pump 1		0	3	0	7	20	L/h	0.00	0.00	0.00	0.00	0.00	0.50	1.00	0.00	0.25	0.75	1.00	0.08	2.54	6.00	0.08	1.72	3.36	6.00
PAC metering Pump 3		0	3	0	8	21	L/h	0.00	0.00	0.00	0.00	0.00	0.50	1.00	0.00	0.25	0.75	1.00	0.08	2.54	6.00	0.08	1.72	3.36	6.00
PAC metering Pump 4		0	3	0	9	22	L/h	0.00	0.00	0.00	0.00	0.00	0.50	1.00	0.00	0.25	0.75	1.00	0.08	2.54	6.00	0.08	1.72	3.36	6.00
Alum Storage 2		0	3	1	0	23	m^3	0.05	0.11	0.11	0.13	-0.13	0.41	0.95	-0.13	0.14	0.64	0.95	0.08	1.54	3.00	0.08	1.06	2.00	3.00
Alum Storage 1		0	3	1	1	24	m^3	0.05	0.11	0.11	0.13	-0.13	0.41	0.95	-0.13	0.14	0.64	0.95	0.08	1.54	3.00	0.08	1.06	2.00	3.00
Alum Transfer Pump		0	3	1	2	25	L/s	0.00	0.00	0.00	0.00	0.00	0.50	1.00	0.00	0.25	0.75	1.00	0.08	1.54	3.00	0.08	1.06	2.00	3.00
Alum Day Tank		0	3	1	3	26	m^3	0.01	0.03	0.05	0.08	-0.08	0.48	0.99	-0.08	0.30	0.72	0.99	0.08	1.54	3.00	0.08	1.06	2.00	3.00
Alum metering Pump 2		0	3	1	4	27	L/h	0.08	0.39	0.86	1.03	-1.03	0.21	0.92	-1.03	-0.41	0.36	0.92	0.08	1.54	3.00	0.08	1.06	2.00	3.00
Alum metering Pump 1		0	3	1	5	28	L/h	0.08	0.39	0.86	1.03	-1.03	0.21	0.92	-1.03	-0.41	0.36	0.92	0.08	1.54	3.00	0.08	1.06	2.00	3.00
Alum metering Pump 4		0	3	1	6	29	L/h	0.16	0.79	1.32	2.06	-2.06	-0.08	0.84	-2.06	-1.07	-0.04	0.84	0.08	1.54	3.00	0.08	1.06	2.00	3.00
Alum metering Pump 3		0	3	1	7	30	L/h	0.16	0.79	1.32	2.06	-2.06	-0.08	0.84	-2.06	-1.07	-0.04	0.84	0.08	1.54	3.00	0.08	1.06	2.00	3.00
Flash Mix Cell 2		0	3	1	8	31	MLD	0.33	0.67	0.58	0.67	-0.67	0.01	0.67	-0.67	-0.33	0.18	0.67	1.00	4.00	7.00	1.00	3.00	5.00	7.00
Flash Mix Cell 1		0	3	1	9	32	MLD	0.33	0.67	0.58	0.67	-0.67	0.01	0.67	-0.67	-0.33	0.18	0.67	1.00	4.00	7.00	1.00	3.00	5.00	7.00
Polymer Mix tank		0	3	2	0	33	L	0.00	0.21	0.21	0.28	-0.28	0.36	1.00	-0.28	0.04	0.54	1.00	0.04	3.52	7.00	0.04	2.36	4.68	7.00
Polymer Day Tank		0	3	2	1	34	L	0.00	0.10	0.10	0.14	-0.14	0.43	1.00	-0.14	0.16	0.66	1.00	0.04	3.52	7.00	0.04	2.36	4.68	7.00
Polymer metering Pump 1		0	3	2	2	35	L/min/head	0.00	0.21	0.21	0.28	-0.28	0.36	1.00	-0.28	0.04	0.54	1.00	0.04	3.52	7.00	0.04	2.36	4.68	7.00
Polymer metering Pump 2		0	3	2	3	36	L/min/head	0.00	0.21	0.21	0.28	-0.28	0.36	1.00	-0.28	0.04	0.54	1.00	0.04	3.52	7.00	0.04	2.36	4.68	7.00
Polymer metering Pump 3		0	3	2	4	37	L/min/head	0.00	0.21	0.21	0.28	-0.28	0.36	1.00	-0.28	0.04	0.54	1.00	0.04	3.52	7.00	0.04	2.36	4.68	7.00
Polymer metering Pump 4		0	3	2	5	38	L/min/head	0.00	0.21	0.21	0.28	-0.28	0.36	1.00	-0.28	0.04	0.54	1.00	0.04	3.52	7.00	0.04	2.36	4.68	7.00
Alum Transfer Pump (back-up)		0	3	2	6	39	L/s	0.00	0.00	0.00	0.00	0.00	0.50	1.00	0.00	0.25	0.75	1.00	0.08	1.54	3.00	0.08	1.06	2.00	3.00
Alum Metering Pump 5(back-up)		0	3	2	7	40	L/h	0.00	0.00	0.00	0.00	0.00	0.50	1.00	0.00	0.25	0.75	1.00	0.08	1.54	3.00	0.08	1.06	2.00	3.00
Alum Metering Pump 3(back-up)		0	3	2	8	41	L/h	0.00	0.00	0.00	0.00	0.00	0.50	1.00	0.00	0.25	0.75	1.00	0.08	1.54	3.00	0.08	1.06	2.00	3.00
Cell I-1	Flocculation System	0	4	0	1	42	MLD	0.10	0.17	0.17	0.20	-0.20	0.36	0.90	-0.20	0.08	0.58	0.90	0.04	3.52	7.00	0.04	2.36	4.68	7.00
Cell II-1		0	4	0	2	43	MLD	0.10	0.17	0.17	0.20	-0.20	0.36	0.90	-0.20	0.08	0.58	0.90	0.04	3.52	7.00	0.04	2.36	4.68	7.00
Cell I-2		0	4	0	3	44	MLD	0.10	0.17	0.17	0.20	-0.20	0.36	0.90	-0.20	0.08	0.58	0.90	0.04	3.52	7.00	0.04	2.36	4.68	7.00
Cell II-2		0	4	0	4	45	MLD	0.10	0.17	0.17	0.20	-0.20	0.36	0.90	-0.20	0.08	0.58	0.90	0.04	3.52	7.00	0.04	2.36	4.68	7.00
Sedimentation Tank I	Sedimentation System	0	5	0	1	46	MLD	0.39	0.67	0.80	0.79	-0.69	-0.03	0.61	-0.69	-0.36	0.10	0.61	0.04	3.52	7.00	0.04	2.36	4.68	7.00
Sedimentation Tank II		0	5	0	2	47	MLD	0.39	0.67	0.80	0.79	-0.69	-0.03	0.61	-0.69	-0.36	0.10	0.61	0.04	3.52	7.00	0.04	2.36	4.68	7.00
Pre-Chlorinator		0	5	0	3	48	Kg/d	0.00	0.19	0.32	0.32	-0.32	0.37	1.00	-0.32	0.03	0.56	1.00	0.04	3.52	7.00	0.04	2.36	4.68	7.00
Fluoride Storage 2		0	5	0	4	49	m^3	0.00	0.00	0.00	0.00	0.00	0.50	1.00	0.00	0.25	0.75	1.00	0.04	3.52	7.00	0.04	2.36	4.68	7.00
Fluoride Storage 1		0	5	0	5	50	m^3	0.00	0.00	0.00	0.00	0.00	0.50	1.00	0.00	0.25	0.75	1.00	0.04	3.52	7.00	0.04	2.36	4.68	7.00
Fluoride Feed Pump		0	5	0	6	51	L/h	0.00	0.25	0.25	0.34	-0.34	0.33	1.00	-0.34	0.00	0.50	1.00	0.04	3.52	7.00	0.04	2.36	4.68	7.00
Fluoride Feed Pump (back-up)		0	5	0	7	52	L/h	0.00	0.25	0.25	0.34	-0.34	0.33	1.00	-0.34	0.00	0.50	1.00	0.04	3.52	7.00	0.04	2.36	4.68	7.00
Filter 1	Filtration System	0	6	0	1	53	MLD	0.39	0.51	0.62	0.80	-0.60	0.06	0.71	-0.60	-0.27	0.24	0.71	0.04	5.02	10.00	0.04	3.36	6.68	10.00
Filter 2		0	6	0	2	54	MLD	0.39	0.51	0.62	0.80	-0.60	0.06	0.71	-0.60	-0.27	0.24	0.71	0.04	5.02	10.00	0.04	3.36	6.68	10.00
Filter 3		0	6	0	3	55	MLD	0.39	0.51	0.62	0.80	-0.60	0.06	0.71	-0.60	-0.27	0.24	0.71	0.04	5.02	10.00	0.04	3.36	6.68	10.00
Filter 4		0	6	0	4	56	MLD	0.39	0.51	0.62	0.80	-0.60	0.06	0.71	-0.60	-0.27	0.24	0.71	0.04	5.02	10.00	0.04	3.36	6.68	10.00
Clear Well		0	6	0	5	57	ML	0.39	0.67	0.88	0.78	-0.78	-0.07	0.61	-0.78	-0.45	0.08	0.61	0.04	3.52	7.00	0.04	2.36	4.68	7.00
On-site Reservoir		0	6	0	6	58	ML	0.39	0.67	0.88	0.78	-0.79	-0.08	0.61	-0.78	-0.43	0.08	0.61	0.04	3.52	7.00	0.04	2.36	4.68	7.00
Post-Chlorinator	High Lifting System	0	7	0	1	59	Kg/d	0.04	0.16	0.22	0.32	-0.32	0.38	0.90	-0.32	0.03	0.50	0.96	0.04	3.52	7.00	0.04	2.36	4.68	7.00
Pump 2		0	7	0	2	60	MLD	0.17	0.28	0.30	0.34	0.19	0.62	0.83	0.19	0.36	0.59	0.83	7.00	18.50	30.00	7.00	14.67	22.33	30.00
Pump 1		0	7	0	3	61	MLD	0.17	0.28	0.30	0.34	0.19	0.62	0.83	0.19	0.36	0.59	0.83	7.00	18.50	30.00	7.00	14.67	22.33	30.00
Pump 4		0	7	0	4	62	MLD	0.17	0.28	0.30	0.34	0.19	0.62	0.83	0.19	0.36	0.59	0.83	7.00	18.50	30.00	7.00	14.67	22.33	30.00
Pump 3		0	7	0	5	63	MLD	0.17	0.28	0.30	0.34	0.19	0.62	0.83	0.19	0.36	0.59	0.83	7.00	18.50	30.00	7.00	14.67	22.33	30.00
Main	Conveyance System	0	8	0	1	64	MLD	0.19	0.34	0.34	0.40	-0.11	0.16	0.81	-0.11	0.03	0.38	0.81	0.08	0.42	0.75	0.08	0.31	0.53	0.75
Terminal Reservoir 1	Boosting System	0	9	0	1	65	MLD	0.00	0.72	0.72	0.96	-0.96	0.02	1.00	-0.96	-0.47	0.03	1.00	0.04	3.52	7.00	0.04	2.36	4.68	7.00
Terminal Reservoir 2		0	9	0	2	66	MLD	0.00	0.72	0.72	0.96	-0.96	0.02	1.00	-0.96	-0.47	0.03	1.00	0.04	3.52	7.00	0.04	2.36	4.68	7.00

APPENDIX II: MATLAB SOURCE CODE

II-A LAKE HURON PRIMARY WATER SUPPLY SYSTEM (LHPWSS)

```
% Last Modified: Thursday, October 07, 2004 @5:45pm
x=-10:0.05:5;
time=0:0.1:200;
alpha=0:0.05:1;
%%%..........Step ONE........Building Structure Array
%%%it is an array that containes fields under each fieldthere are values or
%%%strinngs

    %determines the number of total elements in the system
    [No_of_Elements,Dummy01]=size(rowheaders);

    % Build the Structure array from the imported excel file.
    for n=1:No_of_Elements
            System(n) = struct('Element_Name', rowheaders(n,1),'Element_ID',data(n,1:4),...
                'Element_global_No',data(n,5),'Element_Capacity',data(n,6:8),...
                'Element_Requirment',data(n,9:11),'Element_Norm_Cap_Tri',data(n,12:14),'Element_Norm_Req_Tri',data(n,15:17),...
                'Element_Norm_Cap_Trap',data(n,18:21),'Element_Norm_Req_Trap',data(n,22:25),...
                'Element_State_TriMos',data(n,26:28),'Element_State_TrapMos',data(n,29:32),...
                'Element_Failure_Tri',data(n,33:35),'Element_Failure_Trap',data(n,36:39));
    end

%%% ..........Step TWO....Deterimine the No. of Sub-Systems
%%% by caclualting the maximum No. Value in the ID first two digits && Deterimine
%%%  the No. of elements in each Sub-System by caclualting the maximum No. Value in
%%%  the ID last two digits

        for sse=1:No_of_Elements            % sse stands for Sub-System Element
            % Calculation of the Number of Sub-Systems
            SubSystem(sse,1)={[System(sse).Element_ID(1,1) System(sse).Element_ID(1,2)]};
            SubSyetm_Tenth(sse,1)=num2str(SubSystem{sse,1}(1,1));
            SubSyetm_Units(sse,1)=num2str(SubSystem{sse,1}(1,2));
            SubSystem_String_Code=strcat(SubSyetm_Tenth,SubSyetm_Units);
            SubSystem_Numerical_Code=str2num(SubSystem_String_Code);
            Total_SubSystems=max(SubSystem_Numerical_Code);

            % Calculation of the Number of Elements in each Sub-System
```

```
            Element(sse,1)={[System(sse).Element_ID(1,3) System(sse).Element_ID(1,4) ↙
]};
            SubSyetm_Element_Tenth(sse,1)=num2str(Element{sse,1}(1,1));
            SubSyetm_Element_Units(sse,1)=num2str(Element{sse,1}(1,2));
            Element_String_Code=strcat(SubSyetm_Element_Tenth,SubSyetm_Element_Units ↙
);
            Element_Numerical_Code=str2num(Element_String_Code);
         end

System_Numeric_ID_Array=[SubSystem_Numerical_Code Element_Numerical_Code];

% Creating zero matrix to store information on number of elements in each subsystem
Information_Array=zeros(Total_SubSystems,2);

counter=1;
    for sse2=1:(No_of_Elements-1)
        Information_Array(counter,1)=counter;
        Information_Array(counter,2)=System_Numeric_ID_Array(sse2,2);

            if (System_Numeric_ID_Array((sse2+1),2))<(System_Numeric_ID_Array(sse2,2 ↙
))
                counter=counter+1;
            end

            if counter==Total_SubSystems
               Information_Array(counter,2)=System_Numeric_ID_Array(sse2+1,2);
           end
    end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% ..........Step THREE.....Make Fuzzy MF usable in Augmentation %
%%% Using the Triangula MF in the system Matrix                   %
        for i=1:No_of_Elements                                    %
            A=System(i).Element_State_TriMos(1,1);                %
            B=System(i).Element_State_TriMos(1,2);                %
            C=System(i).Element_State_TriMos(1,3);                %
            Element_MF(i,1)={[triangular(alpha,A,B,C)]};          %
        end                                                       %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% ..........Step THREE.....Make Fuzzy MF usable in Augmentation %
%%% Using the Trapisoidal MF in the system Matrix                 %
```

```
%for i=1:No_of_Elements                                               %
 %       A=System(i).Element_State_TrapMos(1,1);                     %
  %      B=System(i).Element_State_TrapMos(1,2);                     %
   %     C=System(i).Element_State_TrapMos(1,3);                     %
    %    D=System(i).Element_State_TrapMos(1,4);                     %
     %   Element_MF(1,1)={[trapizoidal(alpha,A,B,C,D)]};             %
     %end                                                            %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%% ..........Step FOUR.....Augmenting the paralel and redundant elements
%%% using the SUMMATION operator for both
%% NOTE: all redundant elements or parallel elelemtns will be added to the
%% first element in the group (all values will be incorporated in the MF
%% value of the first element)

%%%%%%%%%%%%%%%%%%%%%%%%%% Redundant Elements%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
                      %1.  Intake Sub-System
                             % there is no redundancy in any of the elements

                      %2.  Low Lifting Sub-System
                             % 3 Redundant elements
                             %    04----07, 05----08 & 06----09
Element_MF{7,1}=fuzzymath(alpha,Element_MF{7,1},Element_MF{10,1},'sum');
Element_MF{8,1}=fuzzymath(alpha,Element_MF{8,1},Element_MF{11,1},'sum');
Element_MF{9,1}=fuzzymath(alpha,Element_MF{9,1},Element_MF{12,1},'sum');

                      %3.  Palsh Mixing Sub-System
                          %PAC, Alum, and Polymer Sub_system
                             % 1 Redundant element
                             %    03----21
Element_MF{15,1}=fuzzymath(alpha,Element_MF{15,1},Element_MF{33,1},'sum');

                      %4.  Flocculation Sub-System
                              % there is no redundancy in any of the elements

                      %5.  Sedimentation Sub-System
                              % there is no redundancy in any of the elements


                      %6.  Filtering Sub-System
                              % there is no redundancy in any of the elements
```

```
                        %7.  High Lifting Sub-System
                               % 2 Redundant elements
                               %    01----04 & 02----05
Element_MF{63,1}=fuzzymath(alpha,Element_MF{63,1},Element_MF{66,1},'sum');
Element_MF{64,1}=fuzzymath(alpha,Element_MF{64,1},Element_MF{67,1},'sum');

                        %8.  Conveyance Sub-System
                                % there is no redundancy in any of the elements

                        %9.  Boosting Sub-System
                               % there is no redundancy in any of the elements

                        %10. Storage Sub-System
                               % there is no redundancy in any of the elements

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Parallel Elements%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
                        %1.  Intake Sub-System
                                % there is no parallel elements

                        %2.  Low Lifting Sub-System
                               % 3 Paralel elements
                               % 04-05-06
Element_MF{7,1}=fuzzymath(alpha,Element_MF{7,1},Element_MF{8,1},'sum');
Element_MF{7,1}=fuzzymath(alpha,Element_MF{7,1},Element_MF{9,1},'sum');

                        %3.  Falsh Mixing Sub-System
                            %DAC, Alum, and Polymer Sub_system
                               % there is no parallel elements
                               % Parallelsmis between the whole two lines not
                               % elements

                        %4.  Flocculation Sub-System
                               % 2 Paralel groups of elements
                               % (01,03,05,07) $ (02,04,06,08)
```

```
Element_MF{34,1}=fuzzymath(alpha,Element_MF{34,1},Element_MF{36,1},'sum');
Element_MF{34,1}=fuzzymath(alpha,Element_MF{34,1},Element_MF{38,1},'sum');
Element_MF{34,1}=fuzzymath(alpha,Element_MF{34,1},Element_MF{40,1},'sum');
Element_MF{35,1}=fuzzymath(alpha,Element_MF{35,1},Element_MF{37,1},'sum');
Element_MF{35,1}=fuzzymath(alpha,Element_MF{35,1},Element_MF{39,1},'sum');
Element_MF{35,1}=fuzzymath(alpha,Element_MF{35,1},Element_MF{41,1},'sum');


                    %5.  Sedimentation Sub-System
                           % 2 Paralel groups of elements
                           % (01,03)-(02,04)
Element_MF{42,1}=fuzzymath(alpha,Element_MF{42,1},Element_MF{44,1},'sum');

Element_MF{43,1}=fuzzymath(alpha,Element_MF{43,1},Element_MF{45,1},'sum');

                    %6.  Filtering Sub-System
                              % 3 Paralel groups of elements
                              % (01,03,05,07,09,11) & (02,04,06,08,10,12) &
                              % (15,16,17)
Element_MF{46,1}=fuzzymath(alpha,Element_MF{46,1},Element_MF{48,1},'sum');
Element_MF{46,1}=fuzzymath(alpha,Element_MF{46,1},Element_MF{50,1},'sum');
Element_MF{46,1}=fuzzymath(alpha,Element_MF{46,1},Element_MF{52,1},'sum');
Element_MF{46,1}=fuzzymath(alpha,Element_MF{46,1},Element_MF{54,1},'sum');
Element_MF{46,1}=fuzzymath(alpha,Element_MF{46,1},Element_MF{56,1},'sum');
Element_MF{47,1}=fuzzymath(alpha,Element_MF{47,1},Element_MF{49,1},'sum');
Element_MF{47,1}=fuzzymath(alpha,Element_MF{47,1},Element_MF{51,1},'sum');
Element_MF{47,1}=fuzzymath(alpha,Element_MF{47,1},Element_MF{53,1},'sum');
Element_MF{47,1}=fuzzymath(alpha,Element_MF{47,1},Element_MF{55,1},'sum');
Element_MF{47,1}=fuzzymath(alpha,Element_MF{47,1},Element_MF{57,1},'sum');
Element_MF{60,1}=fuzzymath(alpha,Element_MF{60,1},Element_MF{61,1},'sum');
Element_MF{60,1}=fuzzymath(alpha,Element_MF{60,1},Element_MF{62,1},'sum');

                    %7.  High Lifting Sub-System
                              % 1 Paralel group
                              % (01,02,03)
Element_MF{63,1}=fuzzymath(alpha,Element_MF{63,1},Element_MF{64,1},'sum');
Element_MF{63,1}=fuzzymath(alpha,Element_MF{63,1},Element_MF{65,1},'sum');

                    %8.  Conveyance Sub-System
                              % 1 Paralel group
                              % (01,02)
Element_MF{68,1}=fuzzymath(alpha,Element_MF{68,1},Element_MF{69,1},'sum');
```

```

                        %9.  Boosting Sub-System
                               % 1 Paralel group
                               % (01,02,03,04)
Element_MF{70,1}=fuzzymath(alpha,Element_MF{70,1},Element_MF{71,1},'sum');
Element_MF{70,1}=fuzzymath(alpha,Element_MF{70,1},Element_MF{72,1},'sum');
Element_MF{70,1}=fuzzymath(alpha,Element_MF{70,1},Element_MF{73,1},'sum');

                        %10. Storage Sub-System
                               % 1 Paralel group
                               % (01,02,03,04)
 Element_MF{75,1}=fuzzymath(alpha,Element_MF{75,1},Element_MF{76,1},'sum');
 Element_MF{75,1}=fuzzymath(alpha,Element_MF{75,1},Element_MF{77,1},'sum');
 Element_MF{75,1}=fuzzymath(alpha,Element_MF{75,1},Element_MF{78,1},'sum');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%% Serial Elements%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Build the sequence of Augmentation Sub-System step wise
% Connection between sub_system is considered serial connection

                        %1.  Intake Sub-System
                          % 1 serial 2 serial 3
System_State=fuzzylogmath(x,alpha,Element_MF{1,1},Element_MF{2,1},'minimum');
System_State=fuzzylogmath(x,alpha,System_State,Element_MF{3,1},'minimum');

                       %2.  Low Lifting Sub-System
                          % 1 serial 2 serial 3 serial (4* parallel 5* parralel 6*)
                          % star element means it has a rdundant element.
System_State=fuzzylogmath(x,alpha,System_State,Element_MF{4,1},'minimum');
System_State=fuzzylogmath(x,alpha,System_State,Element_MF{5,1},'minimum');
System_State=fuzzylogmath(x,alpha,System_State,Element_MF{6,1},'minimum');
System_State=fuzzylogmath(x,alpha,System_State,Element_MF{7,1},'minimum');

                       %3.  Falsh Mixing Sub-System
                           %PAC, Alum, and Polymer Sub_system
                           % (1 serial 3* serial 5 serial 7 serial 9 serial 11
                           %    serial 13 serial 15 serial 17 serial 19)
                           % PARALLEL TO
                           % (2 serial 4 serial 6 serial 8 serial 10 serial
```

```
                          %    12 serial 14 serial 16 serial 18 serial 20)
 % this summation to avoid producing MFs that can not be summed using the
 % summation function (as minmising will produce MFs in different froms
 % than that needed by the summation function)
 Element_MF{13,1}=fuzzymath(alpha,Element_MF{13,1},Element_MF{14,1},'sum');
 Element_MF{15,1}=fuzzymath(alpha,Element_MF{15,1},Element_MF{16,1},'sum');
 Element_MF{17,1}=fuzzymath(alpha,Element_MF{17,1},Element_MF{18,1},'sum');
 Element_MF{19,1}=fuzzymath(alpha,Element_MF{19,1},Element_MF{20,1},'sum');
 Element_MF{21,1}=fuzzymath(alpha,Element_MF{21,1},Element_MF{22,1},'sum');
 Element_MF{23,1}=fuzzymath(alpha,Element_MF{23,1},Element_MF{24,1},'sum');
 Element_MF{25,1}=fuzzymath(alpha,Element_MF{25,1},Element_MF{26,1},'sum');
 Element_MF{27,1}=fuzzymath(alpha,Element_MF{27,1},Element_MF{28,1},'sum');
 Element_MF{29,1}=fuzzymath(alpha,Element_MF{29,1},Element_MF{30,1},'sum');
 Element_MF{31,1}=fuzzymath(alpha,Element_MF{31,1},Element_MF{32,1},'sum');
 Element_MF{13,1}=fuzzylogmath(x,alpha,Element_MF{13,1},Element_MF{15,1},'minimum');
 Element_MF{13,1}=fuzzylogmath(x,alpha,Element_MF{13,1},Element_MF{17,1},'minimum');
 Element_MF{13,1}=fuzzylogmath(x,alpha,Element_MF{13,1},Element_MF{19,1},'minimum');
 Element_MF{13,1}=fuzzylogmath(x,alpha,Element_MF{13,1},Element_MF{21,1},'minimum');
 Element_MF{13,1}=fuzzylogmath(x,alpha,Element_MF{13,1},Element_MF{23,1},'minimum');
 Element_MF{13,1}=fuzzylogmath(x,alpha,Element_MF{13,1},Element_MF{25,1},'minimum');
 Element_MF{13,1}=fuzzylogmath(x,alpha,Element_MF{13,1},Element_MF{27,1},'minimum');
 Element_MF{13,1}=fuzzylogmath(x,alpha,Element_MF{13,1},Element_MF{29,1},'minimum');
 Element_MF{13,1}=fuzzylogmath(x,alpha,Element_MF{13,1},Element_MF{31,1},'minimum');

System_State=fuzzylogmath(x,alpha,System_State,Element_MF{13,1},'minimum');

                      %4.  Flocculation Sub-System
                              % no serials, only previous system-state and
                              % sum of tow parrallel groups of elements
Element_MF{34,1}=fuzzymath(alpha,Element_MF{34,1},Element_MF{35,1},'sum');
System_State=fuzzylogmath(x,alpha,System_State,Element_MF{34,1},'minimum');

                      %5.  Sedimentation Sub-System
                              % no serials, only previous system-state and
                              % sum of tow parrallel groups of elements
Element_MF{42,1}=fuzzymath(alpha,Element_MF{42,1},Element_MF{43,1},'sum');
System_State=fuzzylogmath(x,alpha,System_State,Element_MF{42,1},'minimum');

                      %6.  Filtering Sub-System
                          % (01,03,05,07,09,11) serial 13
                          % PARALLEL TO
                          % (02,04,06,08,10,12) serial 14
```

```
                             % SERIAL TO    (15*,16,17)
Element_MF{46,1}=fuzzymath(alpha,Element_MF{46,1},Element_MF{47,1},'sum');
System_State=fuzzylogmath(x,alpha,System_State,Element_MF{46,1},'minimum');
Element_MF{58,1}=fuzzymath(alpha,Element_MF{58,1},Element_MF{59,1},'sum');
System_State=fuzzylogmath(x,alpha,System_State,Element_MF{58,1},'minimum');
System_State=fuzzylogmath(x,alpha,System_State,Element_MF{60,1},'minimum');

                         %7.  High Lifting Sub-System
                            % no serials, only previous system-state
 System_State=fuzzylogmath(x,alpha,System_State,Element_MF{63,1},'minimum');

                         %8.  Conveyance Sub-System
                            % no serials, only previous system-state
System_State=fuzzylogmath(x,alpha,System_State,Element_MF{68,1},'minimum');

                         %9.  Boosting Sub-System
                            % (01,02,03,04) serial 5
System_State=fuzzylogmath(x,alpha,System_State,Element_MF{70,1},'minimum');
System_State=fuzzylogmath(x,alpha,System_State,Element_MF{74,1},'minimum');

                         %10. Storage Sub-System
                            % this sub-system is paralle to the previous
                            % sub-system
System_State=fuzzylogmath(x,alpha,System_State,Element_MF{75,1},'minimum');

[Row_x,Col_x]=size(x); % basic information about x-values (x-axis)
No_x_steps=Col_x-1;
x_step=max(x)/No_x_steps;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% keeping the original System State MF with zeros for farther use in
% calcuation of the overlap area
System_State_xwise=System_State;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
System_State=[System_State(find(System_State(:,2)>0),1) System_State(find(System_State(:,2)>0),2)];
[dumm_row,dumm_col]=size(System_State);
System_State=[(System_State(1,1)-x_step) 0;System_State;(System_State(dumm_row,1)+x_step) 0];

%%%%%%%%%%%.....System State Areas...%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
    % Determine the maximum alpha value in teh system-state matrix.
    Maximum_alpha_System_State=max(System_State(:,2));

    % based on the No. of rows, a loop will search for the row index of the
    % maximum alpha value
    [row_System_State,column_System_State]=size(System_State);
    max_index_System_State=0;
    for t=1:row_System_State
        max_index_System_State=max_index_System_State+1;
        if System_State(t,2)==Maximum_alpha_System_State
             break;
        end
    end

    % the range from the maximum value to Zero will be divided for high
    % resolution alpha.
    alpha_range_System_State=Maximum_alpha_System_State;
    step_no_System_State=100;
    step_System_State=Maximum_alpha_System_State/step_no_System_State;

    %Interpolation
        %Left-Hand Interpolation
            Left_System_State=(interp1(System_State(1:max_index_System_State,2),System_State(1:max_index_System_State,1),0:step_System_State:Maximum_alpha_System_State)');
            System_State_L=[Left_System_State (0:step_System_State:Maximum_alpha_System_State)'];
        %Right-Hand Interpolation
            Right_System_State=(interp1(System_State(max_index_System_State:row_System_State,2),System_State(max_index_System_State:row_System_State,1),0:step_System_State:Maximum_alpha_System_State)');
            System_State_R=[Right_System_State (0:step_System_State:Maximum_alpha_System_State)'];

    %Area-Calculation
    strip_Area_System_State=0;
    for tt=1:(step_no_System_State-1)
        strip_Area_System_State=strip_Area_System_State+(0.5*((System_State_R(tt,1)-System_State_L(tt,1))+(System_State_R((tt+1),1)-System_State_L((tt+1),1)))*step_System_State);
    end
    Area_System_State=strip_Area_System_State+(0.5*(System_State_R((step_no_System_S
```

```
tate+1),1)-System_State_L((step_no_System_State+1),1))*step_System_State);

    %Weighted Area-Calculation
    weighted_strip_Area_System_State=0;
    for ttt=1:(step_no_System_State-1)
        weighted_strip_Area_System_State=weighted_strip_Area_System_State+((0.5*((System_State_R(ttt,1)-System_State_L(ttt,1))+(System_State_R((ttt+1),1)-System_State_L((ttt+1),1)))*step_System_State)*((System_State_L(ttt,2)+System_State_L((ttt+1),2))*0.5));
    end
    Weighted_Area_System_State=weighted_strip_Area_System_State+((0.5*(System_State_R((step_no_System_State+1),1)-System_State_L((step_no_System_State+1),1))*step_System_State)*System_State_L(step_no_System_State,2));
    %%%%%%%%%%%%.....System State Areas...%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% ..........Step FIVE.....Define the Acceptable level(s) of perfoemance    %
%%% using trapizoidal MF as (a,b,c,d) where c=d                              %
No_of_Levels=3;                                                              %
m1=0.6; m2=0.7; m3=5; m4=5;                                                  %
Level_1y=trapmf(x,[m1 m2 m3 m4]);
Level_1=[x' Level_1y'];
LR_Level_1=(m1*m2)/(m2-m1);

n1=0.6; n2=1.2; n3=5; n4=5;                                                  %
Level_2y=trapmf(x,[n1 n2 n3 n4]);
Level_2=[x' Level_2y'];
LR_Level_2=(n1*n2)/(n2-n1);

q1=0.6; q2=5; q3=5; q4=5;                                                    %
Level_3y=trapmf(x,[q1 q2 q3 q4]);
Level_3=[x' Level_3y'];
LR_Level_3=(q1*q2)/(q2-q1);

LR=[LR_Level_1;LR_Level_2;LR_Level_3];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% ..........Step SIX.....Calculation of the overlap Matrix by taking the minimum
%%% of the state function and each level
OverLap_Level_1y=min(Level_1(:,2),System_State_xwise(:,2));
OverLap_Level_1=[x' OverLap_Level_1y];
```

```
OverLap_Level_2y=min(Level_2(:,2),System_State_xwise(:,2));
OverLap_Level_2=[x' OverLap_Level_2y];

OverLap_Level_3y=min(Level_3(:,2),System_State_xwise(:,2));
OverLap_Level_3=[x' OverLap_Level_3y];

    % take-off the zeros from both ends of the overlap area

Area_Level_1=OverLap_Level_1((find(OverLap_Level_1(:,2)>0)),:);
Area_Level_1_start=[min(OverLap_Level_1((find(OverLap_Level_1(:,2)>0)),1))-x_step 0];
Area_Level_1_end=[max(OverLap_Level_1((find(OverLap_Level_1(:,2)>0)),1))+x_step 0];
Area_Level_1=[Area_Level_1_start;Area_Level_1;Area_Level_1_end];

Area_Level_2=OverLap_Level_2((find(OverLap_Level_2(:,2)>0)),:);
Area_Level_2_start=[min(OverLap_Level_2((find(OverLap_Level_2(:,2)>0)),1))-x_step 0];
Area_Level_2_end=[max(OverLap_Level_2((find(OverLap_Level_2(:,2)>0)),1))+x_step 0];
Area_Level_2=[Area_Level_2_start;Area_Level_2;Area_Level_2_end];


Area_Level_3=OverLap_Level_3((find(OverLap_Level_3(:,2)>0)),:);
Area_Level_3_start=[min(OverLap_Level_3((find(OverLap_Level_3(:,2)>0)),1))-x_step 0];
Area_Level_3_end=[max(OverLap_Level_3((find(OverLap_Level_3(:,2)>0)),1))+x_step 0];
Area_Level_3=[Area_Level_3_start;Area_Level_3;Area_Level_3_end];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% ..........Step SEVEN.....Calculation of the overlap area & weighted area by inteprolation over alpha cuts
%%% for each overlap matrix

    %%%%%%%%%%%%.....Level ONE...%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    % Determine the maximum alpha value in teh ovelap matrix.
    Maximum_alpha_Area_Level_1=max(Area_Level_1(:,2));

    % based on the No. of rows, a loop will search for the row index of the
    % maximum alpha value
    [row_Area_Level_1,column_Area_Level_1]=size(Area_Level_1);
    max_index_Area_Level_1=0;
    for ii=1:row_Area_Level_1
```

```
        max_index_Area_Level_1=max_index_Area_Level_1+1;
        if Area_Level_1(ii,2)==Maximum_alpha_Area_Level_1
            break;
        end
    end

    % the range from the maximum value to Zero will be divided for high
    % resolution alpha.
    alpha_range_Area_Level_1=Maximum_alpha_Area_Level_1;
    step_no_Area_Level_1=100;
    step_Area_Level_1=Maximum_alpha_Area_Level_1/step_no_Area_Level_1;

    %Interpolation
        %Left-Hand Interpolation
            Left_Area_Level_1=(interp1(Area_Level_1(1:max_index_Area_Level_1,2),Area_Level_1(1:max_index_Area_Level_1,1),0:step_Area_Level_1:Maximum_alpha_Area_Level_1)');
            Area_Level_1_L=[Left_Area_Level_1 (0:step_Area_Level_1:Maximum_alpha_Area_Level_1)'];
        %Right-Hand Interpolation
            Right_Area_Level_1=(interp1(Area_Level_1(max_index_Area_Level_1:row_Area_Level_1,2),Area_Level_1(max_index_Area_Level_1:row_Area_Level_1,1),0:step_Area_Level_1:Maximum_alpha_Area_Level_1)');
            Area_Level_1_R=[Right_Area_Level_1 (0:step_Area_Level_1:Maximum_alpha_Area_Level_1)'];

    %Area-Calculation
    strip_Area_Area_Level_1=0;
    for jj=1:(step_no_Area_Level_1-1)
        strip_Area_Area_Level_1=strip_Area_Area_Level_1+(0.5*((Area_Level_1_R(jj,1)-Area_Level_1_L(jj,1))+(Area_Level_1_R((jj+1),1)-Area_Level_1_L((jj+1),1)))*step_Area_Level_1);
    end
    Area_Area_Level_1=strip_Area_Area_Level_1+(0.5*(Area_Level_1_R((step_no_Area_Level_1+1),1)-Area_Level_1_L((step_no_Area_Level_1+1),1))*step_Area_Level_1);

    %Weighted Area-Calculation
    weighted_strip_Area_Area_Level_1=0;
    for kk=1:(step_no_Area_Level_1-1)
        weighted_strip_Area_Area_Level_1=weighted_strip_Area_Area_Level_1+((0.5*((Area_Level_1_R(kk,1)-Area_Level_1_L(kk,1))+(Area_Level_1_R((kk+1),1)-Area_Level_1_L((kk+1),1)))*step_Area_Level_1)*((Area_Level_1_L(kk,2)+Area_Level_1_L((kk+1),2))*0.5));
```

```
    end
    Weighted_Area_Area_Level_1=weighted_strip_Area_Area_Level_1+((0.5*(Area_Level_1_ ↙
R((step_no_Area_Level_1+1),1)-Area_Level_1_L((step_no_Area_Level_1+1),1))*step_Area_ ↙
Level_1)*Area_Level_1_L(step_no_Area_Level_1,2));
    %%%%%%%%%%%%.....Level ONE...%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

    %%%%%%%%%%%%.....Level TWO...%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    % Determine the maximum alpha value in teh ovelap matrix.
    Maximum_alpha_Area_Level_2=max(Area_Level_2(:,2));

    % based on the No. of rows, a loop will search for the row index of the
    % maximum alpha value
    [row_Area_Level_2,column_Area_Level_2]=size(Area_Level_2);
    max_index_Area_Level_2=0;
    for iii=1:row_Area_Level_2
        max_index_Area_Level_2=max_index_Area_Level_2+1;
        if Area_Level_2(iii,2)==Maximum_alpha_Area_Level_2
             break;
        end
    end

    % the range from the maximum value to Zero will be divided for high
    % resolution alpha.
    alpha_range_Area_Level_2=Maximum_alpha_Area_Level_2;
    step_no_Area_Level_2=100;
    step_Area_Level_2=Maximum_alpha_Area_Level_2/step_no_Area_Level_2;

    %Interpolation
        %Left-Hand Interpolation
            Left_Area_Level_2=(interp1(Area_Level_2(1:max_index_Area_Level_2,2),Area ↙
_Level_2(1:max_index_Area_Level_2,1),0:step_Area_Level_2:Maximum_alpha_Area_Level_2) ↙
');
            Area_Level_2_L=[Left_Area_Level_2 (0:step_Area_Level_2:Maximum_alpha_Are ↙
a_Level_2)'];
        %Right-Hand Interpolation
            Right_Area_Level_2=(interp1(Area_Level_2(max_index_Area_Level_2:row_Area ↙
_Level_2,2),Area_Level_2(max_index_Area_Level_2:row_Area_Level_2,1),0:step_Area_Leve ↙
l_2:Maximum_alpha_Area_Level_2)');
            Area_Level_2_R=[Right_Area_Level_2 (0:step_Area_Level_2:Maximum_alpha_Ar ↙
ea_Level_2)'];

    %Area-Calculation
```

```
    strip_Area_Area_Level_2=0;
    for jjj=1:(step_no_Area_Level_2-1)
        strip_Area_Area_Level_2=strip_Area_Area_Level_2+(0.5*((Area_Level_2_R(jjj,1)-Area_Level_2_L(jjj,1))+(Area_Level_2_R((jjj+1),1)-Area_Level_2_L((jjj+1),1)))*step_Area_Level_2);
    end
    Area_Area_Level_2=strip_Area_Area_Level_2+(0.5*(Area_Level_2_R((step_no_Area_Level_2+1),1)-Area_Level_2_L((step_no_Area_Level_2+1),1))*step_Area_Level_2);

    %Weighted Area-Calculation
    weighted_strip_Area_Area_Level_2=0;
    for kkk=1:(step_no_Area_Level_2-1)
        weighted_strip_Area_Area_Level_2=weighted_strip_Area_Area_Level_2+((0.5*((Area_Level_2_R(kkk,1)-Area_Level_2_L(kkk,1))+(Area_Level_2_R((kkk+1),1)-Area_Level_2_L((kkk+1),1)))*step_Area_Level_2)*((Area_Level_2_L(kkk,2)+Area_Level_2_L((kkk+1),2))*0.5));
    end
    Weighted_Area_Area_Level_2=weighted_strip_Area_Area_Level_2+((0.5*(Area_Level_2_R((step_no_Area_Level_2+1),1)-Area_Level_2_L((step_no_Area_Level_2+1),1))*step_Area_Level_2)*Area_Level_2_L(step_no_Area_Level_2,2));
    %%%%%%%%%%%.....Level TWO...%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

    %%%%%%%%%%%.....Level THREE...%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    % Determine the maximum alpha value in teh ovelap matrix.
    Maximum_alpha_Area_Level_3=max(Area_Level_3(:,2));

    % based on the No. of rows, a loop will search for the row index of the
    % maximum alpha value
    [row_Area_Level_3,column_Area_Level_3]=size(Area_Level_3);
    max_index_Area_Level_3=0;
    for iiii=1:row_Area_Level_3
        max_index_Area_Level_3=max_index_Area_Level_3+1;
        if Area_Level_3(iiii,2)==Maximum_alpha_Area_Level_3
            break;
        end
    end

    % the range from the maximum value to Zero will be divided for high
    % resolution alpha.
    alpha_range_Area_Level_3=Maximum_alpha_Area_Level_3;
    step_no_Area_Level_3=100;
    step_Area_Level_3=Maximum_alpha_Area_Level_3/step_no_Area_Level_3;
```

```

    %Interpolation
        %Left-Hand Interpolation
            Left_Area_Level_3=(interp1(Area_Level_3(1:max_index_Area_Level_3,2),Area_Level_3(1:max_index_Area_Level_3,1),0:step_Area_Level_3:Maximum_alpha_Area_Level_3)');
            Area_Level_3_L=[Left_Area_Level_3 (0:step_Area_Level_3:Maximum_alpha_Area_Level_3)'];
        %Right-Hand Interpolation
            Right_Area_Level_3=(interp1(Area_Level_3(max_index_Area_Level_3:row_Area_Level_3,2),Area_Level_3(max_index_Area_Level_3:row_Area_Level_3,1),0:step_Area_Level_3:Maximum_alpha_Area_Level_3)');
            Area_Level_3_R=[Right_Area_Level_3 (0:step_Area_Level_3:Maximum_alpha_Area_Level_3)'];

    %Area-Calculation
    strip_Area_Area_Level_3=0;
    for jjjj=1:(step_no_Area_Level_3-1)
        strip_Area_Area_Level_3=strip_Area_Area_Level_3+(0.5*((Area_Level_3_R(jjjj,1)-Area_Level_3_L(jjjj,1))+(Area_Level_3_R((jjjj+1),1)-Area_Level_3_L((jjjj+1),1)))*step_Area_Level_3);
    end
    Area_Area_Level_3=strip_Area_Area_Level_3+(0.5*(Area_Level_3_R((step_no_Area_Level_3+1),1)-Area_Level_3_L((step_no_Area_Level_3+1),1))*step_Area_Level_3);

    %Weighted Area-Calculation
    weighted_strip_Area_Area_Level_3=0;
    for kkkk=1:(step_no_Area_Level_3-1)
        weighted_strip_Area_Area_Level_3=weighted_strip_Area_Area_Level_3+((0.5*((Area_Level_3_R(kkkk,1)-Area_Level_3_L(kkkk,1))+(Area_Level_3_R((kkkk+1),1)-Area_Level_3_L((kkkk+1),1)))*step_Area_Level_3)*((Area_Level_3_L(kkkk,2)+Area_Level_3_L((kkkk+1),2))*0.5));
    end
    Weighted_Area_Area_Level_3=weighted_strip_Area_Area_Level_3+((0.5*(Area_Level_3_R((step_no_Area_Level_3+1),1)-Area_Level_3_L((step_no_Area_Level_3+1),1))*step_Area_Level_3)*Area_Level_3_L(step_no_Area_Level_3,2));
    %%%%%%%%%%%%.....Level THREE...%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

Area=[Area_Area_Level_1;Area_Area_Level_2;Area_Area_Level_3];
Max_Level=find(Area==max(Area));
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% ..........Step EIGHT.....Reliability measure Calculation
```

```
CM_Level_1=Weighted_Area_Area_Level_1/Weighted_Area_System_State;
CM_Level_2=Weighted_Area_Area_Level_2/Weighted_Area_System_State;
CM_Level_3=Weighted_Area_Area_Level_3/Weighted_Area_System_State;
CM=[CM_Level_1;CM_Level_2;CM_Level_3];

Reliability_Index=(max(CM)*LR(Max_Level,1))/max(LR)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% ..........Step NINE.....Roubustness measure Calculation
Robustness_Index_1_2=1/(CM_Level_1-CM_Level_2)
Robustness_Index_1_3=1/(CM_Level_1-CM_Level_3)
Robustness_Index_2_3=1/(CM_Level_2-CM_Level_3)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%
%%%%%%%
%%%
%%
%
%%
%%%
%%%%%%%
%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% ..........Step TEN.....Reslience measure Calculation
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% ..........Step TEN-ONE.....Make Fuzzy MF usable in augmentation  %
%%% Using the Triangula MF in the system Matrix                      %
%        for i=1:No_of_Elements                                       %
 %           a=System(i).Element_Failure_Tri(1,1);                    %
  %          b=System(i).Element_Failure_Tri(1,2);                    %
   %         c=System(i).Element_Failure_Tri(1,3);                    %
    %        Element_Fail(i,1)={(triangular(alpha,a,b,c))};           %
    %   end                                                          %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% ..........Step TEN-TWO.....Make Fuzzy MF usable in Augmentation  %
```

```
%%% Using the Trapizoidal MF in the system Matrix                    %
          for i=1:No_of_Elements                                     %
             a=System(i).Element_Failure_Trap(1,1);                  %
             b=System(i).Element_Failure_Trap(1,2);                  %
             c=System(i).Element_Failure_Trap(1,3);                  %
             d=System(i).Element_Failure_Trap(1,4);                  %
             Element_Fail(i,1)={[trapizoidal(alpha,a,b,c,d)]};       %
         end                                                         %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% ..........Step TEN-THREE.....Augmenting the paralel and redundant elements
%%% using the MAXIMUM operator for both
%% NOTE: all redundant elements or parallel elelemtns will be added to the
%% first element in the group (all values will be incorporated in the MF
%% value of the first element)

%%%%%%%%%%%%%%%%%%%%%%%%% Redundant Elements%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
                    %1.  Intake Sub-System
                           % there is no redundancy in any of the elements

                    %2.  Low Lifting Sub-System
                           % 3 Redundant elements
                           %    04----07, 05----08 & 06----09
Element_Fail{7,1}=fuzzylogmath(time,alpha,Element_Fail{7,1},Element_Fail{10,1},'maximum');
Element_Fail{8,1}=fuzzylogmath(time,alpha,Element_Fail{8,1},Element_Fail{11,1},'maximum');
Element_Fail{9,1}=fuzzylogmath(time,alpha,Element_Fail{9,1},Element_Fail{12,1},'maximum');

                    %3.  Falsh Mixing Sub-System
                        %PAC, Alum, and Polymer Sub_system
                           % 1 Redundant element
                           %    03----21
Element_Fail{15,1}=fuzzylogmath(time,alpha,Element_Fail{15,1},Element_Fail{23,1},'maximum');

                    %4.  Flocculation Sub-System
                            % there is no redundancy in any of the elements

                    %5.  Sedimentation Sub-System
                            % there is no redundancy in any of the elements
```

```

                        %6.  Filtering Sub-System
                                % there is no redundancy in any of the elements

                        %7.  High Lifting Sub-System
                               % 2 Redundant elements
                               %    01----04 & 02----05
Element_Fail{63,1}=fuzzylogmath(time,alpha,Element_Fail{63,1},Element_Fail{66,1},'maximum');
Element_Fail{64,1}=fuzzylogmath(time,alpha,Element_Fail{64,1},Element_Fail{67,1},'maximum');

                        %8.  Conveyance Sub-System
                                % there is no redundancy in any of the elements

                        %9.  Boosting Sub-System
                               % there is no redundancy in any of the elements

                        %10. Storage Sub-System
                               % there is no redundancy in any of the elements

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%% Parallel Elements%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
                        %1.  Intake Sub-System
                                % there is no parallel elements

                        %2.  Low Lifting Sub-System
                               %  Paralel elements
                               % 04-05-06
Element_Fail{7,1}=fuzzylogmath(time,alpha,Element_Fail{7,1},Element_Fail{8,1},'maximum');
Element_Fail{7,1}=fuzzylogmath(time,alpha,Element_Fail{7,1},Element_Fail{9,1},'maximum');


                       %3.  Falsh Mixing Sub-System
                            %PAC, Alum, and Polymer Sub_system
                               % there is no parallel elements
                               % Parallelsmis between the whole two lines not
```

```
                                  % elements

                          %4.  Flocculation Sub-System
                                  % 2 Paralel groups of elements
                                  % (01,03,05,07) $ (02,04,06,08)
Element_Fail{34,1}=fuzzylogmath(time,alpha,Element_Fail{34,1},Element_Fail{36,1},'maximum');
Element_Fail{34,1}=fuzzylogmath(time,alpha,Element_Fail{34,1},Element_Fail{38,1},'maximum');
Element_Fail{34,1}=fuzzylogmath(time,alpha,Element_Fail{34,1},Element_Fail{40,1},'maximum');
Element_Fail{35,1}=fuzzylogmath(time,alpha,Element_Fail{35,1},Element_Fail{37,1},'maximum');
Element_Fail{35,1}=fuzzylogmath(time,alpha,Element_Fail{35,1},Element_Fail{39,1},'maximum');
Element_Fail{35,1}=fuzzylogmath(time,alpha,Element_Fail{35,1},Element_Fail{41,1},'maximum');

                          %5.  Sedimentation Sub-System
                              % 2 Paralel groups of elements
                              % (01,03)-(02,04)
Element_Fail{42,1}=fuzzylogmath(time,alpha,Element_Fail{42,1},Element_Fail{44,1},'maximum');
Element_Fail{43,1}=fuzzylogmath(time,alpha,Element_Fail{43,1},Element_Fail{45,1},'maximum');


                          %6.  Filtering Sub-System
                                  % 3 Paralel groups of elements
                                  % (01,03,05,07,09,11) & (02,04,06,08,10,12) &
                                  % (15,16,17)
Element_Fail{46,1}=fuzzylogmath(time,alpha,Element_Fail{46,1},Element_Fail{48,1},'maximum');
Element_Fail{46,1}=fuzzylogmath(time,alpha,Element_Fail{46,1},Element_Fail{50,1},'maximum');
Element_Fail{46,1}=fuzzylogmath(time,alpha,Element_Fail{46,1},Element_Fail{52,1},'maximum');
Element_Fail{46,1}=fuzzylogmath(time,alpha,Element_Fail{46,1},Element_Fail{54,1},'maximum');
Element_Fail{46,1}=fuzzylogmath(time,alpha,Element_Fail{46,1},Element_Fail{56,1},'maximum');
Element_Fail{49,1}=fuzzylogmath(time,alpha,Element_Fail{49,1},Element_Fail{51,1},'ma
```

```
ximum');
Element_Fail{49,1}=fuzzylogmath(time,alpha,Element_Fail{49,1},Element_Fail{53,1},'maximum');
Element_Fail{49,1}=fuzzylogmath(time,alpha,Element_Fail{49,1},Element_Fail{55,1},'maximum');
Element_Fail{49,1}=fuzzylogmath(time,alpha,Element_Fail{49,1},Element_Fail{57,1},'maximum');
Element_Fail{60,1}=fuzzylogmath(time,alpha,Element_Fail{60,1},Element_Fail{61,1},'maximum');
Element_Fail{60,1}=fuzzylogmath(time,alpha,Element_Fail{60,1},Element_Fail{62,1},'maximum');

                          %7.  High Lifting Sub-System
                                 % I Paralel group
                                 % (01,02,03)
Element_Fail{63,1}=fuzzylogmath(time,alpha,Element_Fail{63,1},Element_Fail{64,1},'maximum');
Element_Fail{63,1}=fuzzylogmath(time,alpha,Element_Fail{63,1},Element_Fail{65,1},'maximum');

                          %8.  Conveyance Sub-System
                                 % I Paralel group
                                 % (01,02)
Element_Fail{68,1}=fuzzylogmath(time,alpha,Element_Fail{68,1},Element_Fail{69,1},'maximum');


                          %9.  Boosting Sub-System
                                 % I Paralel group
                                 % (01,02,03,04)
Element_Fail{70,1}=fuzzylogmath(time,alpha,Element_Fail{70,1},Element_Fail{71,1},'maximum');
Element_Fail{70,1}=fuzzylogmath(time,alpha,Element_Fail{70,1},Element_Fail{72,1},'maximum');
Element_Fail{70,1}=fuzzylogmath(time,alpha,Element_Fail{70,1},Element_Fail{73,1},'maximum');

                          %10. Storage Sub-System
                                 % I Paralel group
                                 % (01,02,03,04)
Element_Fail{75,1}=fuzzylogmath(time,alpha,Element_Fail{75,1},Element_Fail{76,1},'maximum');
```

```
Element_Fail{75,1}=fuzzylogmath(time,alpha,Element_Fail{75,1},Element_Fail{77,1},'maximum');
Element_Fail{75,1}=fuzzylogmath(time,alpha,Element_Fail{75,1},Element_Fail{78,1},'maximum');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%% Serial Elements%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
[Row_time,Col_time]=size(time);

% Rendering all Element-Failure MFs a full length MF on time domain.
for e1=1:No_of_Elements
    Element_Fail{e1,1}=fuzzylogmath(time,alpha,Element_Fail{e1,1},Element_Fail{e1,1},'maximum');
end

% Then.....getting the start and the end of the support of each Element-Failure MF
% 1. Start of Support
for s2=1:No_of_Elements
    for f=1:(Col_time-1)
        if (Element_Fail{s2,1}(f,2)==0) & (Element_Fail{s2,1}((f+1),2)>0)
            System(s2).Element_Failure_Support(1,1)=Element_Fail{s2,1}(f,1);
        end
    end
end
% 2. End of Support
for s3=1:No_of_Elements
    for f=1:(Col_time-1)
        if (Element_Fail{s3,1}(f,2)>0) & (Element_Fail{s3,1}((f+1),2)==0)
            System(s3).Element_Failure_Support(1,2)=Element_Fail{s3,1}((f+1),1);
        end
    end
end
% 3. Calculation of support length of all MFs
for s4=1:No_of_Elements
    System(s4).Element_Failure_Support_Length(1,1)=(System(s4).Element_Failure_Support(1,2)-System(s4).Element_Failure_Support(1,1));
end

% Then.....getting the start and the end of the modal values of each Element-Failure MF
% 1. Start of modal
```

```
for m2=1:No_of_Elements
    for ff=1:(Col_time-1)
        if (Element_Fail{m2,1}(ff,2)<max(Element_Fail{m2,1}(:,2))) & (Element_Fail{m ↙
2,1}((ff+1),2)==max(Element_Fail{m2,1}(:,2)))
            System(m2).Element_Failure_Modal(1,1)=Element_Fail{m2,1}(ff,1);
        end
    end
end
% 2. End of Support
for m3=1:No_of_Elements
    for ff=1:(Col_time-1)
        if (Element_Fail{m3,1}(ff,2)==max(Element_Fail{m3,1}(:,2))) & (Element_Fail{ ↙
m3,1}((ff+1),2)<max(Element_Fail{m3,1}(:,2)))
            System(m3).Element_Failure_Modal(1,2)=Element_Fail{m3,1}((ff+1),1);
        end
    end
end
% 3. Calculation of support length of all MFs
for m4=1:No_of_Elements
    System(m4).Element_Failure_Modal_Length(1,1)=(System(m4).Element_Failure_Modal(1 ↙
,2)-System(m4).Element_Failure_Modal(1,1));
end


% Build the sequence of Augmentation Sub-System step wise
% Connection between sub_system is considered serial connection

System_Fail_Information=[System(1).Element_global_No(1,1) System(1).Element_Failure_ ↙
Support_Length(1,1) System(1).Element_Failure_Modal_Length(1,1);...
                        System(2).Element_global_No(1,1) System(2).Element_Failure_S ↙
upport_Length(1,1) System(2).Element_Failure_Modal_Length(1,1);...
                        System(3).Element_global_No(1,1) System(3).Element_Failure_S ↙
upport_Length(1,1) System(3).Element_Failure_Modal_Length(1,1);...
                        System(4).Element_global_No(1,1) System(4).Element_Failure_S ↙
upport_Length(1,1) System(4).Element_Failure_Modal_Length(1,1);...
                        System(5).Element_global_No(1,1) System(5).Element_Failure_S ↙
upport_Length(1,1) System(5).Element_Failure_Modal_Length(1,1);...
                        System(6).Element_global_No(1,1) System(6).Element_Failure_S ↙
upport_Length(1,1) System(6).Element_Failure_Modal_Length(1,1);...
                        System(7).Element_global_No(1,1) System(7).Element_Failure_S ↙
upport_Length(1,1) System(7).Element_Failure_Modal_Length(1,1);...
                        System(13).Element_global_No(1,1) System(13).Element_Failure ↙
```

```
_Support_Length(1,1) System(13).Element_Failure_Modal_Length(1,1);...
                              System(34).Element_global_No(1,1) System(34).Element_Failure ↙
_Support_Length(1,1) System(34).Element_Failure_Modal_Length(1,1);...
                              System(42).Element_global_No(1,1) System(42).Element_Failure ↙
_Support_Length(1,1) System(42).Element_Failure_Modal_Length(1,1);...
                              System(46).Element_global_No(1,1) System(46).Element_Failure ↙
_Support_Length(1,1) System(46).Element_Failure_Modal_Length(1,1);...
                              System(58).Element_global_No(1,1) System(58).Element_Failure ↙
_Support_Length(1,1) System(58).Element_Failure_Modal_Length(1,1);...
                              System(60).Element_global_No(1,1) System(60).Element_Failure ↙
_Support_Length(1,1) System(60).Element_Failure_Modal_Length(1,1);...
                              System(63).Element_global_No(1,1) System(63).Element_Failure ↙
_Support_Length(1,1) System(63).Element_Failure_Modal_Length(1,1);...
                              System(68).Element_global_No(1,1) System(68).Element_Failure ↙
_Support_Length(1,1) System(68).Element_Failure_Modal_Length(1,1);...
                              System(70).Element_global_No(1,1) System(70).Element_Failure ↙
_Support_Length(1,1) System(70).Element_Failure_Modal_Length(1,1);...
                              System(75).Element_global_No(1,1) System(75).Element_Failure ↙
_Support_Length(1,1) System(75).Element_Failure_Modal_Length(1,1)];

% Determine the controling MF, (first MF with max support, then MF with max
% modal)

[Row_Fail,Col_Fail]=size(System_Fail_Information);

max_support=max(System_Fail_Information(:,2));
max_support_elements=find(System_Fail_Information(:,2)==max_support);
[Row_max_support_elements,Col_max_support_elements]=size(max_support_elements);

if (Row_max_support_elements>1)
    max_modal=max(System_Fail_Information(:,3));
    max_modal_elements=find(System_Fail_Information(:,3)==max_modal);
    [Row_max_modal_elements,Col_max_modal_elements]=size(max_modal_elements);
    if (Row_max_modal_elements>1)
        System_Fail=[System_Fail_Information(max_modal_elements(1,1),1) System_Fail_ ↙
Information(max_modal_elements(1,1),2) System_Fail_Information(max_modal_elements(1, ↙
1),3)];
    end
    System_Fail=[System_Fail_Information((max_modal_elements(1,1)==max_modal),1) Sys ↙
tem_Fail_Information((max_modal_elements(1,1)==max_modal),2) System_Fail_Information ↙
((max_modal_elements(1,1)==max_modal),3)];
end
```

```
System_Fail=[System_Fail_Information(max_support_elements(1,1),1) System_Fail_Information(max_support_elements(1,1),2) System_Fail_Information(max_support_elements(1,1),3)];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% ..........Step TEN-FOUR.....use the MF of the controling Element
System_Fail_MF=Element_Fail{System_Fail(1,1),1};

[row_t,col_t]=size(System_Fail_MF);

Resilience_Index=0;
nominator=0;
denominator=0;
for t=1:(row_t-1)
    nominator=nominator+(0.5*(System_Fail_MF(t+1,1)+System_Fail_MF(t,1)));
    denominator=denominator+(System_Fail_MF(t,2)*(0.5*(System_Fail_MF(t+1,1)+System_Fail_MF(t,1))));
end
Resilience_Index=nominator/denominator
```

II-B ELGIN AREA PRIMARY WATER SUPPLY SYSTEM (LHPWSS)

```
% Last Modified: Thursday,  October 07, 2004 @5:45pm
x=-1:0.01:4;
time=0:0.1:200;
alpha=0:0.05:1;
%%%...........Step ONE........Building Structure Array
%%%it is an array that containes fields under each fieldthere are values or
%%%strinngs

    %determines the number of total elements in the system
    [No_of_Elements,Dummy01]=size(rowheaders);

    % Build the Structure array from the imported excel file.
    for n=1:No_of_Elements
            System(n) = struct('Element_Name', rowheaders(n,1),'Element_ID',data(n,1:4),...
                'Element_global_No',data(n,5),'Element_Capacity',data(n,6:8),...
                'Element_Requirment',data(n,9:11),'Element_Norm_Cap_Tri',data(n,12:14),'Element_Norm_Req_Tri',data(n,15:17),...
                'Element_Norm_Cap_Trap',data(n,18:21),'Element_Norm_Req_Trap',data(n,22:25),...
                'Element_State_TriMos',data(n,26:28),'Element_State_TrapMos',data(n,29:32),...
                'Element_Failure_Tri',data(n,33:35),'Element_Failure_Trap',data(n,36:39));
    end

%%% ..........Step TWO.....Deterimine the No. of Sub-Systems
%%% by caclualting the maximum No. Value in the ID first two digits && Deterimine
%%%  the No. of elements in each Sub-System by caclualting the maximum No. Value in
%%%  the ID last two digits

        for sse=1:No_of_Elements            % sse stands for Sub-System Element
            % Calculation of the Number of Sub-Systems
            SubSystem(sse,1)={[System(sse).Element_ID(1,1) System(sse).Element_ID(1,2)]};
            SubSyetm_Tenth(sse,1)=num2str(SubSystem{sse,1}(1,1));
            SubSyetm_Units(sse,1)=num2str(SubSystem{sse,1}(1,2));
            SubSystem_String_Code=strcat(SubSyetm_Tenth,SubSyetm_Units);
            SubSystem_Numerical_Code=str2num(SubSystem_String_Code);
            Total_SubSystems=max(SubSystem_Numerical_Code);

            % Calculation of the Number of Elements in each Sub-System
```

```
            Element(sse,1)={[System(sse).Element_ID(1,3) System(sse).Element_ID(1,4)]};
            SubSyetm_Element_Tenth(sse,1)=num2str(Element{sse,1}(1,1));
            SubSyetm_Element_Units(sse,1)=num2str(Element{sse,1}(1,2));
            Element_String_Code=strcat(SubSyetm_Element_Tenth,SubSyetm_Element_Units);
            Element_Numerical_Code=str2num(Element_String_Code);
        end

System_Numeric_ID_Array=[SubSystem_Numerical_Code Element_Numerical_Code];

% Creating zero matrix to store information on number of elements in each subsystem
Information_Array=zeros(Total_SubSystems,2);

counter=1;
    for sse2=1:(No_of_Elements-1)
        Information_Array(counter,1)=counter;
        Information_Array(counter,2)=System_Numeric_ID_Array(sse2,2);

            if (System_Numeric_ID_Array((sse2+1),2))<(System_Numeric_ID_Array(sse2,2))
                counter=counter+1;
            end

            if counter==Total_SubSystems
               Information_Array(counter,2)=System_Numeric_ID_Array(sse2+1,2);
           end
    end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% ..........Step THREE.....Make Fuzzy MF usable in Augmentation %
%%% Using the Triangula MF in the system Matrix                   %
%        for i=1:No_of_Elements                                    %
 %           a=System(i).Element_State_TriMos(1,1);                %
  %          b=System(i).Element_State_TriMos(1,2);                %
   %         c=System(i).Element_State_TriMos(1,3);                %
    %        Element_MF(i,1)={[triangular(alpha,a,b,c)]};          %
    %   end                                                       %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% ..........Step THREE.....Make Fuzzy MF usable in Augmentation %
%%% Using the Trapizoidal MF in the system Matrix                 %
```

```
        for i=1:No_of_Elements                                      %
            a=System(i).Element_State_TrapMos(1,1);                 %
            b=System(i).Element_State_TrapMos(1,2);                 %
            c=System(i).Element_State_TrapMos(1,3);                 %
            d=System(i).Element_State_TrapMos(1,4);                 %
            Element_MF(i,1)={[trapizoidal(alpha,a,b,c,d)]};         %
        end                                                          %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%% ..........Step FOUR.....Augmenting the paralel and redundant elements
%%% using the SUMMATION operator for both
%% NOTE: all redundant elements or parallel elelemtns will be added to the
%% first element in the group (all values will be incorporated in the MF
%% value of the first element)

%%%%%%%%%%%%%%%%%%%%%%%%%%% Redundant Elements%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
                      %1.  Intake Sub-System
                             % 1 Redundant elements
                             % 03----04
Element_MF{3,1}=fuzzymath(alpha,Element_MF{3,1},Element_MF{4,1},'sum');

                      %2.  Low Lifting Sub-System
                             % 3 Redundant elements
                             %    25----39, 27----40 & 29----41
Element_MF{25,1}=fuzzymath(alpha,Element_MF{25,1},Element_MF{39,1},'sum');
Element_MF{27,1}=fuzzymath(alpha,Element_MF{27,1},Element_MF{40,1},'sum');
Element_MF{29,1}=fuzzymath(alpha,Element_MF{29,1},Element_MF{41,1},'sum');

                      %3.  Falsh Mixing Sub-System
                          %PAC, Alum, and Polymer Sub_system
                              % there is no redundancy in any of the elements

                      %4.  Flocculation Sub-System
                              % there is no redundancy in any of the elements

                      %5.  Sedimentation Sub-System
                              % 1 Redundant elements
                              % 51----58
Element_MF{51,1}=fuzzymath(alpha,Element_MF{51,1},Element_MF{58,1},'sum');

                      %6.  Filtering Sub-System
                              % there is no redundancy in any of the elements
```

```

                     %7.  High Lifting Sub-System
                            % there is no redundancy in any of the elements

                     %8.  Conveyance Sub-System
                             % there is no redundancy in any of the elements

                     %9. Storage Sub-System
                            % there is no redundancy in any of the elements

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%% Parallel Elements%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
                     %1.  Intake Sub-System
                             % there is no parallel elements

                     %2.  Low Lifting Sub-System
                            % 4 Paralel elements
                            % 09-10-11-12

Element_MF{9,1}=fuzzymath(alpha,Element_MF{9,1},Element_MF{10,1},'sum');
Element_MF{9,1}=fuzzymath(alpha,Element_MF{9,1},Element_MF{11,1},'sum');
Element_MF{9,1}=fuzzymath(alpha,Element_MF{9,1},Element_MF{12,1},'sum');

                     %3.  Falsh Mixing Sub-System
                         %PAC, Alum, and Polymer Sub_system
                            % 14-15
                            % 16-17
                            % 19-20-21-22
                            % 23-24
                            % 27-28-29-30
                            % 31-32
                            % 35-36-37-38
Element_MF{14,1}=fuzzymath(alpha,Element_MF{14,1},Element_MF{15,1},'sum');
Element_MF{16,1}=fuzzymath(alpha,Element_MF{16,1},Element_MF{17,1},'sum');
Element_MF{19,1}=fuzzymath(alpha,Element_MF{19,1},Element_MF{20,1},'sum');
Element_MF{19,1}=fuzzymath(alpha,Element_MF{19,1},Element_MF{21,1},'sum');
Element_MF{19,1}=fuzzymath(alpha,Element_MF{19,1},Element_MF{22,1},'sum');
Element_MF{23,1}=fuzzymath(alpha,Element_MF{23,1},Element_MF{24,1},'sum');
```

```
Element_MF{27,1}=fuzzymath(alpha,Element_MF{27,1},Element_MF{28,1},'sum');
Element_MF{27,1}=fuzzymath(alpha,Element_MF{27,1},Element_MF{29,1},'sum');
Element_MF{27,1}=fuzzymath(alpha,Element_MF{27,1},Element_MF{30,1},'sum');
Element_MF{31,1}=fuzzymath(alpha,Element_MF{31,1},Element_MF{32,1},'sum');
Element_MF{35,1}=fuzzymath(alpha,Element_MF{35,1},Element_MF{36,1},'sum');
Element_MF{35,1}=fuzzymath(alpha,Element_MF{35,1},Element_MF{37,1},'sum');
Element_MF{35,1}=fuzzymath(alpha,Element_MF{35,1},Element_MF{38,1},'sum');

                    %4.  Flocculation Sub-System
                           % 4 Paralel elements
                           % 42-43-44-45
Element_MF{42,1}=fuzzymath(alpha,Element_MF{42,1},Element_MF{43,1},'sum');
Element_MF{42,1}=fuzzymath(alpha,Element_MF{42,1},Element_MF{44,1},'sum');
Element_MF{42,1}=fuzzymath(alpha,Element_MF{42,1},Element_MF{45,1},'sum');

                    %5.  Sedimentation Sub-System
                        % 2 Paralel groups of elements
                        % 46-47
                        % 49-50

Element_MF{46,1}=fuzzymath(alpha,Element_MF{46,1},Element_MF{47,1},'sum');
Element_MF{49,1}=fuzzymath(alpha,Element_MF{49,1},Element_MF{50,1},'sum');

                    %6.  Filtering Sub-System
                           % 4 Paralel elements
                           % 52-53-54-55
Element_MF{52,1}=fuzzymath(alpha,Element_MF{52,1},Element_MF{53,1},'sum');
Element_MF{52,1}=fuzzymath(alpha,Element_MF{52,1},Element_MF{54,1},'sum');
Element_MF{52,1}=fuzzymath(alpha,Element_MF{52,1},Element_MF{55,1},'sum');

                    %7.  High Lifting Sub-System
                           % 4 Paralel elements
                           % 60-61-62-63
Element_MF{60,1}=fuzzymath(alpha,Element_MF{60,1},Element_MF{61,1},'sum');
Element_MF{60,1}=fuzzymath(alpha,Element_MF{60,1},Element_MF{62,1},'sum');
Element_MF{60,1}=fuzzymath(alpha,Element_MF{60,1},Element_MF{63,1},'sum');

                    %8.  Conveyance Sub-System
                           % there is no parallel elements

                    %9. Storage Sub-System
                           % 2 Paralel elements
```

```
                          % 65-66
 Element_MF{65,1}=fuzzymath(alpha,Element_MF{65,1},Element_MF{66,1},'sum');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%% Serial Elements%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Build the sequence of Augmentation Sub-System step wise
% Connection between sub_system is considered serial connection

                     %1.  Intake Sub-System
                        % 1 serial 2 serial 3*
                         % star element means it has a rdundant element
System_State=fuzzylogmath(x,alpha,Element_MF{1,1},Element_MF{2,1},'minimum');
System_State=fuzzylogmath(x,alpha,System_State,Element_MF{3,1},'minimum');

                     %2.  Low Lifting Sub-System
                        % 5 serial 7 serial (9#) serial 13
System_State=fuzzylogmath(x,alpha,System_State,Element_MF{5,1},'minimum');
System_State=fuzzylogmath(x,alpha,System_State,Element_MF{7,1},'minimum');
System_State=fuzzylogmath(x,alpha,System_State,Element_MF{9,1},'minimum');
System_State=fuzzylogmath(x,alpha,System_State,Element_MF{13,1},'minimum');

                     %3.  Falsh Mixing Sub-System
                         %PAC, Alum, and Polymer Sub_system
                         % (14#) serial (16#) serial 18 serial (19#) serial (23) serial
                         % 25* serial 26 serial (27#) serial (31#) serial 33 serial 34 serial (35#)
System_State=fuzzylogmath(x,alpha,System_State,Element_MF{14,1},'minimum');
System_State=fuzzylogmath(x,alpha,System_State,Element_MF{16,1},'minimum');
System_State=fuzzylogmath(x,alpha,System_State,Element_MF{18,1},'minimum');
System_State=fuzzylogmath(x,alpha,System_State,Element_MF{19,1},'minimum');
System_State=fuzzylogmath(x,alpha,System_State,Element_MF{23,1},'minimum');
System_State=fuzzylogmath(x,alpha,System_State,Element_MF{25,1},'minimum');
System_State=fuzzylogmath(x,alpha,System_State,Element_MF{26,1},'minimum');
System_State=fuzzylogmath(x,alpha,System_State,Element_MF{27,1},'minimum');
System_State=fuzzylogmath(x,alpha,System_State,Element_MF{31,1},'minimum');
System_State=fuzzylogmath(x,alpha,System_State,Element_MF{33,1},'minimum');
System_State=fuzzylogmath(x,alpha,System_State,Element_MF{34,1},'minimum');
System_State=fuzzylogmath(x,alpha,System_State,Element_MF{35,1},'minimum');
```

```
                  %4.  Flocculation Sub-System
                       % no serials, only previous system-state
System_State=fuzzylogmath(x,alpha,System_State,Element_MF{42,1},'minimum');

                  %5.  Sedimentation Sub-System
                       % (46#) serial 48 serial (50#) serial 51
System_State=fuzzylogmath(x,alpha,System_State,Element_MF{46,1},'minimum');
System_State=fuzzylogmath(x,alpha,System_State,Element_MF{48,1},'minimum');
System_State=fuzzylogmath(x,alpha,System_State,Element_MF{50,1},'minimum');
System_State=fuzzylogmath(x,alpha,System_State,Element_MF{51,1},'minimum');

                  %6.  Filtering Sub-System
                    % (52#) serial 56 serial 57
System_State=fuzzylogmath(x,alpha,System_State,Element_MF{52,1},'minimum');
System_State=fuzzylogmath(x,alpha,System_State,Element_MF{56,1},'minimum');
System_State=fuzzylogmath(x,alpha,System_State,Element_MF{57,1},'minimum');


                  %7.  High Lifting Sub-System
                     % 59 serial (60#)
System_State=fuzzylogmath(x,alpha,System_State,Element_MF{59,1},'minimum');
System_State=fuzzylogmath(x,alpha,System_State,Element_MF{60,1},'minimum');

                  %8.  Conveyance Sub-System
                     % no serials, only previous system-state
System_State=fuzzylogmath(x,alpha,System_State,Element_MF{64,1},'minimum');

                  %9. Storage Sub-System
                     % no serials, only previous system-state
System_State=fuzzylogmath(x,alpha,System_State,Element_MF{65,1},'minimum');

[Row_x,Col_x]=size(x); % basic information about x-values (x-axis)
No_x_steps=Col_x-1;
x_step=max(x)/No_x_steps;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% keeping the original System State MF with zeros for farther use in
% calcuation of the overlap area
System_State_xwise=System_State;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

System_State=[System_State(find(System_State(:,2)>0),1) System_State(find(System_Sta
te(:,2)>0),2)];
```

```
[dumm_row,dumm_col]=size(System_State);
System_State=[(System_State(1,1)-x_step) 0;System_State;(System_State(dumm_row,1)+x_step) 0];

%%%%%%%%%%%%.....System State Areas...%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    % Determine the maximum alpha value in teh system-state matrix.
    Maximum_alpha_System_State=max(System_State(:,2));

    % based on the No. of rows, a loop will search for the row index of the
    % maximum alpha value
    [row_System_State,column_System_State]=size(System_State);
    max_index_System_State=0;
    for t=1:row_System_State
        max_index_System_State=max_index_System_State+1;
        if System_State(t,2)==Maximum_alpha_System_State
             break;
        end
    end

    % the range from the maximum value to Zero will be divided for high
    % resolution alpha.
    alpha_range_System_State=Maximum_alpha_System_State;
    step_no_System_State=100;
    step_System_State=Maximum_alpha_System_State/step_no_System_State;

    %Interpolation
        %Left-Hand Interpolation
            Left_System_State=(interp1(System_State(1:max_index_System_State,2),System_State(1:max_index_System_State,1),0:step_System_State:Maximum_alpha_System_State)');
            System_State_L=[Left_System_State (0:step_System_State:Maximum_alpha_System_State)'];
        %Right-Hand Interpolation
            Right_System_State=(interp1(System_State(max_index_System_State:row_System_State,2),System_State(max_index_System_State:row_System_State,1),0:step_System_State:Maximum_alpha_System_State)');
            System_State_R=[Right_System_State (0:step_System_State:Maximum_alpha_System_State)'];

    %Area-Calculation
    strip_Area_System_State=0;
    for tt=1:(step_no_System_State-1)
```

```
            strip_Area_System_State=strip_Area_System_State+(0.5*((System_State_R(tt,1)-System_State_L(tt,1))+(System_State_R((tt+1),1)-System_State_L((tt+1),1)))*step_System_State);
        end
        Area_System_State=strip_Area_System_State+(0.5*(System_State_R((step_no_System_State+1),1)-System_State_L((step_no_System_State+1),1))*step_System_State);

        %Weighted Area-Calculation
        weighted_strip_Area_System_State=0;
        for ttt=1:(step_no_System_State-1)
            weighted_strip_Area_System_State=weighted_strip_Area_System_State+((0.5*((System_State_R(ttt,1)-System_State_L(ttt,1))+(System_State_R((ttt+1),1)-System_State_L((ttt+1),1)))*step_System_State)*((System_State_L(ttt,2)+System_State_L((ttt+1),2))*0.5));
        end
        Weighted_Area_System_State=weighted_strip_Area_System_State+((0.5*(System_State_R((step_no_System_State+1),1)-System_State_L((step_no_System_State+1),1))*step_System_State)*System_State_L(step_no_System_State,2));
        %%%%%%%%%%%.....System State Areas...%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% ..........Step FIVE.....Define the Acceptable level(s) of perfoemance    %
%%% using trapizoidal MF as (a,b,c,d) where c=d                             %
No_of_Levels=3;                                                            %
m1=-0.5; m2=4.5; m3=5; m4=5;                                                %
Level_1y=trapmf(x,[m1 m2 m3 m4]);
Level_1=[x' Level_1y'];
LR_Level_1=(m1*m2)/(m2-m1);

n1=0.2; n2=1; n3=5; n4=5;                                                 %
Level_2y=trapmf(x,[n1 n2 n3 n4]);
Level_2=[x' Level_2y'];
LR_Level_2=(n1*n2)/(n2-n1);

q1=-1.0; q2=5; q3=5; q4=5;                                                %
Level_3y=trapmf(x,[q1 q2 q3 q4]);
Level_3=[x' Level_3y'];
LR_Level_3=(q1*q2)/(q2-q1);

LR=[LR_Level_1;LR_Level_2;LR_Level_3];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% ..........Step SIX.....Calculation of the overlap Matrix by taking the minimum
%%% of the state function and each level
OverLap_Level_1y=min(Level_1(:,2),System_State_xwise(:,2));
OverLap_Level_1=[x' OverLap_Level_1y];

OverLap_Level_2y=min(Level_2(:,2),System_State_xwise(:,2));
OverLap_Level_2=[x' OverLap_Level_2y];

OverLap_Level_3y=min(Level_3(:,2),System_State_xwise(:,2));
OverLap_Level_3=[x' OverLap_Level_3y];

    % take-off the seros from both ends of the overlap area

Area_Level_1=OverLap_Level_1((find(OverLap_Level_1(:,2)>0)),:);
Area_Level_1_start=[min(OverLap_Level_1((find(OverLap_Level_1(:,2)>0)),1))-x_step 0];
Area_Level_1_end=[max(OverLap_Level_1((find(OverLap_Level_1(:,2)>0)),1))+x_step 0];
Area_Level_1=[Area_Level_1_start;Area_Level_1;Area_Level_1_end];

Area_Level_2=OverLap_Level_2((find(OverLap_Level_2(:,2)>0)),:);
Area_Level_2_start=[min(OverLap_Level_2((find(OverLap_Level_2(:,2)>0)),1))-x_step 0];
Area_Level_2_end=[max(OverLap_Level_2((find(OverLap_Level_2(:,2)>0)),1))+x_step 0];
Area_Level_2=[Area_Level_2_start;Area_Level_2;Area_Level_2_end];


Area_Level_3=OverLap_Level_3((find(OverLap_Level_3(:,2)>0)),:);
Area_Level_3_start=[min(OverLap_Level_3((find(OverLap_Level_3(:,2)>0)),1))-x_step 0];
Area_Level_3_end=[max(OverLap_Level_3((find(OverLap_Level_3(:,2)>0)),1))+x_step 0];
Area_Level_3=[Area_Level_3_start;Area_Level_3;Area_Level_3_end];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% ..........Step SEVEN.....Calculation of the overlap area & weighted area by intoprolation over alpha cuts
%%% for each overlap matrix

    %%%%%%%%%%%%.....Level ONE...%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    % Determine the maximum alpha value in teh ovelap matrix.
    Maximum_alpha_Area_Level_1=max(Area_Level_1(:,2));

```

```matlab
    % based on the No. of rows, a loop will search for the row index of the
    % maximum alpha value
    [row_Area_Level_1,column_Area_Level_1]=size(Area_Level_1);
    max_index_Area_Level_1=0;
    for ii=1:row_Area_Level_1
        max_index_Area_Level_1=max_index_Area_Level_1+1;
        if Area_Level_1(ii,2)==Maximum_alpha_Area_Level_1
             break;
        end
    end

    % the range from the maximum value to Zero will be divided for high
    % resolution alpha.
    alpha_range_Area_Level_1=Maximum_alpha_Area_Level_1;
    step_no_Area_Level_1=100;
    step_Area_Level_1=Maximum_alpha_Area_Level_1/step_no_Area_Level_1;

    %Interpolation
        %Left-Hand Interpolation
            Left_Area_Level_1=(interp1(Area_Level_1(1:max_index_Area_Level_1,2),Area_Level_1(1:max_index_Area_Level_1,1),0:step_Area_Level_1:Maximum_alpha_Area_Level_1)');
            Area_Level_1_L=[Left_Area_Level_1 (0:step_Area_Level_1:Maximum_alpha_Area_Level_1)'];
        %Right-Hand Interpolation
            Right_Area_Level_1=(interp1(Area_Level_1(max_index_Area_Level_1:row_Area_Level_1,2),Area_Level_1(max_index_Area_Level_1:row_Area_Level_1,1),0:step_Area_Level_1:Maximum_alpha_Area_Level_1)');
            Area_Level_1_R=[Right_Area_Level_1 (0:step_Area_Level_1:Maximum_alpha_Area_Level_1)'];

    %Area-Calculation
    strip_Area_Area_Level_1=0;
    for jj=1:(step_no_Area_Level_1-1)
        strip_Area_Area_Level_1=strip_Area_Area_Level_1+(0.5*((Area_Level_1_R(jj,1)-Area_Level_1_L(jj,1))+(Area_Level_1_R((jj+1),1)-Area_Level_1_L((jj+1),1)))*step_Area_Level_1);
    end
    Area_Area_Level_1=strip_Area_Area_Level_1+(0.5*(Area_Level_1_R((step_no_Area_Level_1+1),1)-Area_Level_1_L((step_no_Area_Level_1+1),1))*step_Area_Level_1);

    %Weighted Area-Calculation
```

```
    weighted_strip_Area_Area_Level_1=0;
    for kk=1:(step_no_Area_Level_1-1)
        weighted_strip_Area_Area_Level_1=weighted_strip_Area_Area_Level_1+((0.5*((Area_Level_1_R(kk,1)-Area_Level_1_L(kk,1))+(Area_Level_1_R((kk+1),1)-Area_Level_1_L((kk+1),1))))*step_Area_Level_1)*((Area_Level_1_L(kk,2)+Area_Level_1_L((kk+1),2))*0.5));
    end
    Weighted_Area_Area_Level_1=weighted_strip_Area_Area_Level_1+((0.5*(Area_Level_1_R((step_no_Area_Level_1+1),1)-Area_Level_1_L((step_no_Area_Level_1+1),1))*step_Area_Level_1)*Area_Level_1_L(step_no_Area_Level_1,2));
    %%%%%%%%%%%.....Level ONE...%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

    %%%%%%%%%%%.....Level TWO...%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    % Determine the maximum alpha value in teh ovelap matrix.
    Maximum_alpha_Area_Level_2=max(Area_Level_2(:,2));

    % based on the No. of rows, a loop will search for the row index of the
    % maximum alpha value
    [row_Area_Level_2,column_Area_Level_2]=size(Area_Level_2);
    max_index_Area_Level_2=0;
    for iii=1:row_Area_Level_2
        max_index_Area_Level_2=max_index_Area_Level_2+1;
        if Area_Level_2(iii,2)==Maximum_alpha_Area_Level_2
            break;
        end
    end

    % the range from the maximum value to Zero will be divided for high
    % resolution alpha.
    alpha_range_Area_Level_2=Maximum_alpha_Area_Level_2;
    step_no_Area_Level_2=100;
    step_Area_Level_2=Maximum_alpha_Area_Level_2/step_no_Area_Level_2;

    %Interpolation
        %Left-Hand Interpolation
            Left_Area_Level_2=(interp1(Area_Level_2(1:max_index_Area_Level_2,2),Area_Level_2(1:max_index_Area_Level_2,1),0:step_Area_Level_2:Maximum_alpha_Area_Level_2)');
            Area_Level_2_L=[Left_Area_Level_2 (0:step_Area_Level_2:Maximum_alpha_Area_Level_2)'];
        %Right-Hand Interpolation
            Right_Area_Level_2=(interp1(Area_Level_2(max_index_Area_Level_2:row_Area_Level_2,2),Area_Level_2(max_index_Area_Level_2:row_Area_Level_2,1),0:step_Area_Leve
```

```
l_2:Maximum_alpha_Area_Level_2)');
            Area_Level_2_R=[Right_Area_Level_2 (0:step_Area_Level_2:Maximum_alpha_Area_Level_2)'];

    %Area-Calculation
    strip_Area_Area_Level_2=0;
    for jjj=1:(step_no_Area_Level_2-1)
        strip_Area_Area_Level_2=strip_Area_Area_Level_2+(0.5*((Area_Level_2_R(jjj,1)-Area_Level_2_L(jjj,1))+(Area_Level_2_R((jjj+1),1)-Area_Level_2_L((jjj+1),1)))*step_Area_Level_2);
    end
    Area_Area_Level_2=strip_Area_Area_Level_2+(0.5*(Area_Level_2_R((step_no_Area_Level_2+1),1)-Area_Level_2_L((step_no_Area_Level_2+1),1))*step_Area_Level_2);

    %Weighted Area-Calculation
    weighted_strip_Area_Area_Level_2=0;
    for kkk=1:(step_no_Area_Level_2-1)
        weighted_strip_Area_Area_Level_2=weighted_strip_Area_Area_Level_2+((0.5*((Area_Level_2_R(kkk,1)-Area_Level_2_L(kkk,1))+(Area_Level_2_R((kkk+1),1)-Area_Level_2_L((kkk+1),1)))*step_Area_Level_2)*((Area_Level_2_L(kkk,2)+Area_Level_2_L((kkk+1),2))*0.5));
    end
    Weighted_Area_Area_Level_2=weighted_strip_Area_Area_Level_2+((0.5*(Area_Level_2_R((step_no_Area_Level_2+1),1)-Area_Level_2_L((step_no_Area_Level_2+1),1))*step_Area_Level_2)*Area_Level_2_L(step_no_Area_Level_2,2));
    %%%%%%%%%%%.....Level TWO...%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

    %%%%%%%%%%%.....Level THREE...%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    % Determine the maximum alpha value in teh ovelap matrix.
    Maximum_alpha_Area_Level_3=max(Area_Level_3(:,2));

    % based on the No. of rows, a loop will search for the row index of the
    % maximum alpha value
    [row_Area_Level_3,column_Area_Level_3]=size(Area_Level_3);
    max_index_Area_Level_3=0;
    for iiii=1:row_Area_Level_3
        max_index_Area_Level_3=max_index_Area_Level_3+1;
        if Area_Level_3(iiii,2)==Maximum_alpha_Area_Level_3
            break;
        end
    end

```

```
    % the range from the maximum value to Zero will be divided for high
    % resolution alpha.
    alpha_range_Area_Level_3=Maximum_alpha_Area_Level_3;
    step_no_Area_Level_3=100;
    step_Area_Level_3=Maximum_alpha_Area_Level_3/step_no_Area_Level_3;

    %Interpolation
        %Left-Hand Interpolation
            Left_Area_Level_3=(interp1(Area_Level_3(1:max_index_Area_Level_3,2),Area_Level_3(1:max_index_Area_Level_3,1),0:step_Area_Level_3:Maximum_alpha_Area_Level_3)');
            Area_Level_3_L=[Left_Area_Level_3 (0:step_Area_Level_3:Maximum_alpha_Area_Level_3)'];
        %Right-Hand Interpolation
            Right_Area_Level_3=(interp1(Area_Level_3(max_index_Area_Level_3:row_Area_Level_3,2),Area_Level_3(max_index_Area_Level_3:row_Area_Level_3,1),0:step_Area_Level_3:Maximum_alpha_Area_Level_3)');
            Area_Level_3_R=[Right_Area_Level_3 (0:step_Area_Level_3:Maximum_alpha_Area_Level_3)'];

    %Area-Calculation
    strip_Area_Area_Level_3=0;
    for jjjj=1:(step_no_Area_Level_3-1)
        strip_Area_Area_Level_3=strip_Area_Area_Level_3+(0.5*((Area_Level_3_R(jjjj,1)-Area_Level_3_L(jjjj,1))+(Area_Level_3_R((jjjj+1),1)-Area_Level_3_L((jjjj+1),1)))*step_Area_Level_3);
    end
    Area_Area_Level_3=strip_Area_Area_Level_3+(0.5*(Area_Level_3_R((step_no_Area_Level_3+1),1)-Area_Level_3_L((step_no_Area_Level_3+1),1))*step_Area_Level_3);

    %Weighted Area-Calculation
    weighted_strip_Area_Area_Level_3=0;
    for kkkk=1:(step_no_Area_Level_3-1)
        weighted_strip_Area_Area_Level_3=weighted_strip_Area_Area_Level_3+((0.5*((Area_Level_3_R(kkkk,1)-Area_Level_3_L(kkkk,1))+(Area_Level_3_R((kkkk+1),1)-Area_Level_3_L((kkkk+1),1)))*step_Area_Level_3)*((Area_Level_3_L(kkkk,2)+Area_Level_3_L((kkkk+1),2))*0.5));
    end
    Weighted_Area_Area_Level_3=weighted_strip_Area_Area_Level_3+((0.5*(Area_Level_3_R((step_no_Area_Level_3+1),1)-Area_Level_3_L((step_no_Area_Level_3+1),1))*step_Area_Level_3)*Area_Level_3_L(step_no_Area_Level_3,2));
    %%%%%%%%%%%%.....Level THREE...%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```

Area=[Area_Area_Level_1;Area_Area_Level_2;Area_Area_Level_3];
Max_Level=find(Area==max(Area));
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% ..........Step EIGHT.....Reliability measure Calculation
CM_Level_1=Weighted_Area_Area_Level_1/Weighted_Area_System_State;
CM_Level_2=Weighted_Area_Area_Level_2/Weighted_Area_System_State;
CM_Level_3=Weighted_Area_Area_Level_3/Weighted_Area_System_State;
CM=[CM_Level_1;CM_Level_2;CM_Level_3];

Reliability_Index=(max(CM)*LR(Max_Level,1))/max(LR)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% ..........Step NINE.....Roubustness measure Calculation
Robustness_Index_1_2=1/(CM_Level_1-CM_Level_2)
Robustness_Index_1_3=1/(CM_Level_1-CM_Level_3)
Robustness_Index_2_3=1/(CM_Level_2-CM_Level_3)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%
%%%%%%%
%%%
%%
%
%%
%%%
%%%%%%%
%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% ..........Step TEN.....Reslience measure Calculation

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% ..........Step TEN-ONE.....Make Fuzzy MF usable in augmentation  %
%%% Using the Triangula MF in the system Matrix                      %
%        for i=1:No_of_Elements                                       %
 %           aa=System(i).Element_Failure_Tri(1,1);                   %
  %          bb=System(i).Element_Failure_Tri(1,2);                   %
```

```
    %         cc=System(i).Element_Failure_Tri(1,3);                    %
     %        Element_Fail(i,1)={[triangular(alpha,aa,bb,cc)]};            %
     %   end                                                           %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% ..........Step TEN-TWO.....Make Fuzzy MF usable in Augmentation  %
%%% Using the Trapizoidal MF in the system Matrix                      %
          for i=1:No_of_Elements                                        %
              aa=System(i).Element_Failure_Trap(1,1);                   %
              bb=System(i).Element_Failure_Trap(1,2);                   %
              cc=System(i).Element_Failure_Trap(1,3);                   %
              dd=System(i).Element_Failure_Trap(1,4);                   %
              Element_Fail(i,1)={[trapizoidal(alpha,aa,bb,cc,dd)]};     %
          end                                                            %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%% ..........Step TEN-THREE.....Augmenting the paralel and redundant elements
%%% using the MAXIMUM operator for both
%% NOTE: all redundant elements or parallel elelemtns will be added to the
%% first element in the group (all values will be incorporated in the MF
%% value of the first element)

%%%%%%%%%%%%%%%%%%%%%%%%%% Redundant Elements%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
                      %1.  Intake Sub-System
                             % 1 Redundant element
                             %    03----04
Element_Fail{3,1}=fuzzylogmath(time,alpha,Element_Fail{3,1},Element_Fail{4,1},'maximum');


                      %2.  Low Lifting Sub-System
                             % 3 Redundant element
                             %    25---39, 27---40, 29----41
Element_Fail{25,1}=fuzzylogmath(time,alpha,Element_Fail{25,1},Element_Fail{39,1},'maximum');
Element_Fail{27,1}=fuzzylogmath(time,alpha,Element_Fail{27,1},Element_Fail{40,1},'maximum');
Element_Fail{29,1}=fuzzylogmath(time,alpha,Element_Fail{29,1},Element_Fail{41,1},'maximum');

                      %3.  Falsh Mixing Sub-System
                          %PAC, Alum, and Polymer Sub_system
```

```
                              % there is no redundancy in any of the elements

                         %4.  Floooulation Sub-System
                                 % there is no redundancy in any of the elements

                         %5.  Sedimentation Sub-System
                                 % 3 Redundant element
                                 %  51---58
Element_Fail{51,1}=fuzzylogmath(time,alpha,Element_Fail{51,1},Element_Fail{58,1},'ma
ximum');

                         %6.  Filtering Sub-System
                                 % there is no redundancy in any of the elements

                         %7.  High Lifting Sub-System
                                % there is no redundancy in any of the elements

                         %8.  Conveyance Sub-System
                                 % there is no redundancy in any of the elements

                         %9.  Storage Sub-System
                                % there is no redundancy in any of the elements

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%% Parallel Elements%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%
                         %1.  Intake Sub-System
                                 % there is no parallel elements

                         %2.  Low Lifting Sub-System
                                % 4 Paralel elements
                                % 09-10-11-12
Element_Fail{9,1}=fuzzylogmath(time,alpha,Element_Fail{9,1},Element_Fail{10,1},'maxi
mum');
Element_Fail{9,1}=fuzzylogmath(time,alpha,Element_Fail{9,1},Element_Fail{11,1},'maxi
mum');
Element_Fail{9,1}=fuzzylogmath(time,alpha,Element_Fail{9,1},Element_Fail{12,1},'maxi
mum');

                        %3.  Falsh Mixing Sub-System
                             %PAC, Alum, and Polymer Sub_system
```

```
                                        % 14-15
                                        % 16-17
                                        % 19-20-21-22
                                        % 23-24
                                        % 27-28-29-30
                                        % 31-32
                                        % 35-36-37-38
Element_Fail{14,1}=fuzzylogmath(time,alpha,Element_Fail{14,1},Element_Fail{15,1},'ma ↙
ximum');
Element_Fail{16,1}=fuzzylogmath(time,alpha,Element_Fail{16,1},Element_Fail{17,1},'ma ↙
ximum');
Element_Fail{19,1}=fuzzylogmath(time,alpha,Element_Fail{19,1},Element_Fail{20,1},'ma ↙
ximum');
Element_Fail{19,1}=fuzzylogmath(time,alpha,Element_Fail{19,1},Element_Fail{21,1},'ma ↙
ximum');
Element_Fail{19,1}=fuzzylogmath(time,alpha,Element_Fail{19,1},Element_Fail{22,1},'ma ↙
ximum');
Element_Fail{23,1}=fuzzylogmath(time,alpha,Element_Fail{23,1},Element_Fail{24,1},'ma ↙
ximum');
Element_Fail{27,1}=fuzzylogmath(time,alpha,Element_Fail{27,1},Element_Fail{28,1},'ma ↙
ximum');
Element_Fail{27,1}=fuzzylogmath(time,alpha,Element_Fail{27,1},Element_Fail{29,1},'ma ↙
ximum');
Element_Fail{27,1}=fuzzylogmath(time,alpha,Element_Fail{27,1},Element_Fail{30,1},'ma ↙
ximum');
Element_Fail{31,1}=fuzzylogmath(time,alpha,Element_Fail{31,1},Element_Fail{32,1},'ma ↙
ximum');
Element_Fail{35,1}=fuzzylogmath(time,alpha,Element_Fail{35,1},Element_Fail{36,1},'ma ↙
ximum');
Element_Fail{35,1}=fuzzylogmath(time,alpha,Element_Fail{35,1},Element_Fail{37,1},'ma ↙
ximum');
Element_Fail{35,1}=fuzzylogmath(time,alpha,Element_Fail{35,1},Element_Fail{38,1},'ma ↙
ximum');

                              %4.  Flocculation Sub-System
                                        % 4 Paralel elements
                                        % 42-43-44-45
Element_Fail{42,1}=fuzzylogmath(time,alpha,Element_Fail{42,1},Element_Fail{43,1},'ma ↙
ximum');
Element_Fail{42,1}=fuzzylogmath(time,alpha,Element_Fail{42,1},Element_Fail{44,1},'ma ↙
ximum');
Element_Fail{42,1}=fuzzylogmath(time,alpha,Element_Fail{42,1},Element_Fail{45,1},'ma ↙
```

```
ximum');

                    %5.  Sedimentation Sub-System
                        % 2 Paralel groups of elements
                        % 46-47
                        % 49-50
Element_Fail{46,1}=fuzzylogmath(time,alpha,Element_Fail{46,1},Element_Fail{47,1},'maximum');
Element_Fail{49,1}=fuzzylogmath(time,alpha,Element_Fail{49,1},Element_Fail{50,1},'maximum');

                    %6.  Filtering Sub-System
                            % 4 Paralel elements
                            % 52-53-54-55
Element_Fail{52,1}=fuzzylogmath(time,alpha,Element_Fail{52,1},Element_Fail{52,1},'maximum');
Element_Fail{52,1}=fuzzylogmath(time,alpha,Element_Fail{52,1},Element_Fail{54,1},'maximum');
Element_Fail{52,1}=fuzzylogmath(time,alpha,Element_Fail{52,1},Element_Fail{55,1},'maximum');

                     %7.  High Lifting Sub-System
                            % 4 Paralel elements
                            % 60-61-62-63
Element_Fail{60,1}=fuzzylogmath(time,alpha,Element_Fail{60,1},Element_Fail{61,1},'maximum');
Element_Fail{60,1}=fuzzylogmath(time,alpha,Element_Fail{60,1},Element_Fail{62,1},'maximum');
Element_Fail{60,1}=fuzzylogmath(time,alpha,Element_Fail{60,1},Element_Fail{63,1},'maximum');

                    %8.  Conveyance Sub-System
                            % there is no parallel elements

                     %9. Storage Sub-System
                            % 65-66
Element_Fail{65,1}=fuzzylogmath(time,alpha,Element_Fail{65,1},Element_Fail{66,1},'maximum');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%% Serial Elements%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```

[Row_time,Col_time]=size(time);

% Rendering all Element-Failure MFs a full length MF on time domain.
for e1=1:No_of_Elements
    Element_Fail{e1,1}=fuzzylogmath(time,alpha,Element_Fail{e1,1},Element_Fail{e1,1},'maximum');
end

% Then.....getting the start and the end of the support of each Element-Failure MF
% 1. Start of Support
for s2=1:No_of_Elements
    for f=1:(Col_time-1)
        if (Element_Fail{s2,1}(f,2)==0) & (Element_Fail{s2,1}((f+1),2)>0)
            System(s2).Element_Failure_Support(1,1)=Element_Fail{s2,1}(f,1);
        end
    end
end
% 2. End of Support
for s3=1:No_of_Elements
    for f=1:(Col_time-1)
        if (Element_Fail{s3,1}(f,2)>0) & (Element_Fail{s3,1}((f+1),2)==0)
            System(s3).Element_Failure_Support(1,2)=Element_Fail{s3,1}((f+1),1);
        end
    end
end
% 3. Calculation of support length of all MFs
for s4=1:No_of_Elements
    System(s4).Element_Failure_Support_Length(1,1)=(System(s4).Element_Failure_Support(1,2)-System(s4).Element_Failure_Support(1,1));
end

% Then.....getting the start and the end of the modal values of each Element-Failure MF
% 1. Start of modal
for m2=1:No_of_Elements
    for ff=1:(Col_time-1)
        if (Element_Fail{m2,1}(ff,2)<max(Element_Fail{m2,1}(:,2))) & (Element_Fail{m2,1}((ff+1),2)==max(Element_Fail{m2,1}(:,2)))
            System(m2).Element_Failure_Modal(1,1)=Element_Fail{m2,1}(ff,1);
        end
    end
```

```
end
% 2. End of Support
for m3=1:No_of_Elements
    for ff=1:(Col_time-1)
        if (Element_Fail{m3,1}(ff,2)==max(Element_Fail{m3,1}(:,2))) & (Element_Fail{m3,1}((ff+1),2)<max(Element_Fail{m3,1}(:,2)))
            System(m3).Element_Failure_Modal(1,2)=Element_Fail{m3,1}((ff+1),1);
        end
    end
end
% 3. Calculation of support length of all MFs
for m4=1:No_of_Elements
    System(m4).Element_Failure_Modal_Length(1,1)=(System(m4).Element_Failure_Modal(1,2)-System(m4).Element_Failure_Modal(1,1));
end


% Build the sequence of Augmentation Sub-System step wise
% Connection between sub_system is considered serial connection
System_Fail_Information=[System(1).Element_global_No(1,1) System(1).Element_Failure_Support_Length(1,1) System(1).Element_Failure_Modal_Length(1,1);...
                         System(2).Element_global_No(1,1) System(2).Element_Failure_Support_Length(1,1) System(2).Element_Failure_Modal_Length(1,1);...
                         System(3).Element_global_No(1,1) System(3).Element_Failure_Support_Length(1,1) System(3).Element_Failure_Modal_Length(1,1);...
                         System(5).Element_global_No(1,1) System(5).Element_Failure_Support_Length(1,1) System(5).Element_Failure_Modal_Length(1,1);...
                         System(7).Element_global_No(1,1) System(7).Element_Failure_Support_Length(1,1) System(7).Element_Failure_Modal_Length(1,1);...
                         System(9).Element_global_No(1,1) System(9).Element_Failure_Support_Length(1,1) System(9).Element_Failure_Modal_Length(1,1);...
                         System(13).Element_global_No(1,1) System(13).Element_Failure_Support_Length(1,1) System(13).Element_Failure_Modal_Length(1,1);...
                         System(14).Element_global_No(1,1) System(14).Element_Failure_Support_Length(1,1) System(14).Element_Failure_Modal_Length(1,1);...
                         System(16).Element_global_No(1,1) System(16).Element_Failure_Support_Length(1,1) System(16).Element_Failure_Modal_Length(1,1);...
                         System(18).Element_global_No(1,1) System(18).Element_Failure_Support_Length(1,1) System(18).Element_Failure_Modal_Length(1,1);...
                         System(19).Element_global_No(1,1) System(19).Element_Failure_Support_Length(1,1) System(19).Element_Failure_Modal_Length(1,1);...
                         System(23).Element_global_No(1,1) System(23).Element_Failure
```

```
_Support_Length(1,1) System(23).Element_Failure_Modal_Length(1,1);...
                    System(25).Element_global_No(1,1) System(25).Element_Failure_Support_Length(1,1) System(25).Element_Failure_Modal_Length(1,1);...
                    System(26).Element_global_No(1,1) System(26).Element_Failure_Support_Length(1,1) System(26).Element_Failure_Modal_Length(1,1);...
                    System(27).Element_global_No(1,1) System(27).Element_Failure_Support_Length(1,1) System(27).Element_Failure_Modal_Length(1,1);...
                    System(31).Element_global_No(1,1) System(31).Element_Failure_Support_Length(1,1) System(31).Element_Failure_Modal_Length(1,1);...
                    System(33).Element_global_No(1,1) System(33).Element_Failure_Support_Length(1,1) System(33).Element_Failure_Modal_Length(1,1);...
                    System(34).Element_global_No(1,1) System(34).Element_Failure_Support_Length(1,1) System(34).Element_Failure_Modal_Length(1,1);...
                    System(35).Element_global_No(1,1) System(35).Element_Failure_Support_Length(1,1) System(35).Element_Failure_Modal_Length(1,1);...
                    System(42).Element_global_No(1,1) System(42).Element_Failure_Support_Length(1,1) System(42).Element_Failure_Modal_Length(1,1);...
                    System(46).Element_global_No(1,1) System(46).Element_Failure_Support_Length(1,1) System(46).Element_Failure_Modal_Length(1,1);...

                    System(48).Element_global_No(1,1) System(48).Element_Failure_Support_Length(1,1) System(48).Element_Failure_Modal_Length(1,1);...

                    System(50).Element_global_No(1,1) System(50).Element_Failure_Support_Length(1,1) System(50).Element_Failure_Modal_Length(1,1);...

                    System(51).Element_global_No(1,1) System(51).Element_Failure_Support_Length(1,1) System(51).Element_Failure_Modal_Length(1,1);...
                    System(52).Element_global_No(1,1) System(52).Element_Failure_Support_Length(1,1) System(52).Element_Failure_Modal_Length(1,1);...
                    System(56).Element_global_No(1,1) System(56).Element_Failure_Support_Length(1,1) System(56).Element_Failure_Modal_Length(1,1);...
                    System(57).Element_global_No(1,1) System(57).Element_Failure_Support_Length(1,1) System(57).Element_Failure_Modal_Length(1,1);...
                    System(63).Element_global_No(1,1) System(63).Element_Failure_Support_Length(1,1) System(63).Element_Failure_Modal_Length(1,1);...
                    System(59).Element_global_No(1,1) System(59).Element_Failure_Support_Length(1,1) System(59).Element_Failure_Modal_Length(1,1);...
                    System(60).Element_global_No(1,1) System(60).Element_Failure_Support_Length(1,1) System(60).Element_Failure_Modal_Length(1,1);...
                    System(65).Element_global_No(1,1) System(65).Element_Failure_Support_Length(1,1) System(65).Element_Failure_Modal_Length(1,1)];
```

```
% Determine the controling MF, (first MF with max support, then MF with max
% modal)

[Row_Fail,Col_Fail]=size(System_Fail_Information);

max_support=max(System_Fail_Information(:,2));
max_support_elements=find(System_Fail_Information(:,2)==max_support);
[Row_max_support_elements,Col_max_support_elements]=size(max_support_elements);

if (Row_max_support_elements>1)
    max_modal=max(System_Fail_Information(:,3));
    max_modal_elements=find(System_Fail_Information(:,3)==max_modal);
    [Row_max_modal_elements,Col_max_modal_elements]=size(max_modal_elements);
    if (Row_max_modal_elements>1)
        System_Fail=[System_Fail_Information(max_modal_elements(1,1),1) System_Fail_Information(max_modal_elements(1,1),2) System_Fail_Information(max_modal_elements(1,1),3)];
    end
    System_Fail=[System_Fail_Information((max_modal_elements(1,1)==max_modal),1) System_Fail_Information((max_modal_elements(1,1)==max_modal),2) System_Fail_Information((max_modal_elements(1,1)==max_modal),3)];
end

System_Fail=[System_Fail_Information(max_support_elements(1,1),1) System_Fail_Information(max_support_elements(1,1),2) System_Fail_Information(max_support_elements(1,1),3)];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% ..........Step TEN-FOUR.....use the MF of the controling Element
System_Fail_MF=Element_Fail{System_Fail(1,1),1};

[row_t,col_t]=size(System_Fail_MF);

Resilience_Index=0;
nominator=0;
denominator=0;
for t=1:(row_t-1)
    nominator=nominator+(0.5*(System_Fail_MF(t+1,1)+System_Fail_MF(t,1)));
    denominator=denominator+(System_Fail_MF(t,2)*(0.5*(System_Fail_MF(t+1,1)+System_Fail_MF(t,1))));
end


Resilience_Index=nominator/denominator
```

II-C CUSTOM MATLAB FUZZY SCRIPTS

1. FUZZY TRIANGULAR MEMEBRSHIP FUNCTION
2. FUZZY TRAPIZOIDAL MEMEBRSHIP FUNCTION
3. FUZZY ARTITHEMTATIC OPERATIONS
4. FUZZY LOGIC OPERATIONS

```
function A=triangular(alpha,a,b,c)
%TRIANGULAR Fuzzy Number
% Last Modified: Sunday, July 25, 2004 @6:15pm

if (b<a),
   disp('Not valid input - b must be >= a');
   break;
elseif (c<b),
   disp('Not valid input - c must be >= b');
   break;
end


[Row_al,Col_al]=size(alpha);

No_alpha_step=Col_al-1;
alpha_step=max(alpha)/No_alpha_step;

    for i=1:((2*No_alpha_step)+1)
            if  (i<=No_alpha_step)
                MFN(i,1)=a+((i-1)*alpha_step*(b-a));
                MFN(i,2)=min(alpha)+((i-1)*alpha_step);
            else
                MFN(i,1)=b+((i-1-No_alpha_step)*alpha_step*(c-b));
                MFN(i,2)=max(alpha)-((i-1-No_alpha_step)*alpha_step);
            end
    end
A=[MFN(:,1) MFN(:,2)];
```

```
function A=trapizoidal(alpha,a,b,c,d)
%TRAPIZOIDAL Fuzzy Number
% Last Modified: Sunday, July 25, 2004 @7:00pm

if (b<a),
   disp('Not valid input - b must be >= a');
   break;
elseif (c<a),
   disp('Not valid input - c must be >= a');
   break;
elseif (d<a),
   disp('Not valid input - d must be >= a');
   break;
elseif (c<b),
   disp('Not valid input - c must be >= b');
   break;
elseif (d<b),
   disp('Not valid input - d must be >= b');
   break;
elseif (d<c),
   disp('Not valid input - d must be >= c');
   break;
end


[Row_al,Col_al]=size(alpha);

No_alpha_step=Col_al-1;
alpha_step=max(alpha)/No_alpha_step;

    for i=1:((2*No_alpha_step)+2)
            if  (i<=No_alpha_step)
                MFN(i,1)=a+((i-1)*alpha_step*(b-a));
                MFN(i,2)=min(alpha)+((i-1)*alpha_step);
            end

            if  (i==(No_alpha_step+1))
                MFN(i,1)=b;
                MFN(i,2)=1;
            end

            if  (i==(No_alpha_step+2))
```

```
                MFN(i,1)=c;
                MFN(i,2)=1;
            end

            if  (i>(No_alpha_step+2))
                MFN(i,1)=c+((i-2-No_alpha_step)*alpha_step*(d-c));
                MFN(i,2)=max(alpha)-((i-2-No_alpha_step)*alpha_step);
            end
    end
A=[MFN(:,1) MFN(:,2)];
```

```
function result = fuzzymath(alpha,A,B,operator)
% Last Modified: Monday, July 26, 2004 @9:00am


[Row_al,Col_al]=size(alpha); % basic information about membership values (alpha)
No_alpha_step=Col_al-1;
alpha_step=max(alpha)/No_alpha_step;

% Basic information about the two membership functions
[Row_A,Col_A]=size(A);
[Row_B,Col_B]=size(B);

% a loop to make one matrix with left and right value to perform the interval calculations (this is done for the
%% two membership functions)
    % the LEFT values
for i=1:Col_al
    AA(i,2)=A(i,1);
    BB(i,2)=B(i,1);
end

    % the RIGHT values
counter=0;
for j=0:No_alpha_step
    counter=counter+1;
    AA(counter,3)=A((Row_A-j),1);
    BB(counter,3)=B((Row_A-j),1);
end
    % corresponding alpha values
for k=1:Col_al
    AA(k,1)=A(k,2);
    BB(k,1)=B(k,2);
end

% Interval arithmatic
for L=1:Col_al
    if strcmp(operator, 'sum'),
            result_interval(L,1)=A(L,2);
            result_interval(L,2)=AA(L,2)+BB(L,2);
            result_interval(L,3)=AA(L,3)+BB(L,3);

    elseif strcmp(operator, 'sub'),
```

```
            result_interval=[(AA(L,1)-BB(L,2)) (AA(L,2)-BB(L,1))];
            result_interval(L,1)=A(L,2);
            result_interval(L,2)=AA(L,2)-BB(L,3);
            result_interval(L,3)=AA(L,3)-BB(L,2);

    end
end


% Rebuilding the resultant Membership function in the same way as in the
% input membership funotions
    % the Triangular case
if (A(Col_al,2)==1) & (A((Col_al+1),2)~=1)
    for y1=1:Col_al
        result(y1,2)=A(y1,2);
        result(y1,1)=result_interval(y1,2);
    end
    for y2=1:No_alpha_step
        result((Col_al+y2),2)=A((Col_al-y2),2);
        result((Col_al+y2),1)=result_interval((Col_al-y2),3);
    end
    % the Trapisoidal case
elseif (A(Col_al,2)==1) & (A((Col_al+1),2)==1)
    for z1=1:Col_al
        result(z1,2)=A(z1,2);
        result(z1,1)=result_interval(z1,2);
    end
    for z2=0:No_alpha_step
        result((Col_al+z2+1),2)=A((Col_al-z2),2);
        result((Col_al+z2+1),1)=result_interval((Col_al-z2),3);
    end
end
```

```
function C = fuzzylogmath(Domain,alpha,A,B,operator)
% Last Modified: Friday, August 19, 2004 @8:00pm


[Row_al,Col_al]=size(alpha); % basic information about membership values (alpha)
No_alpha_steps=Col_al-1;
alpha_step=max(alpha)/No_alpha_steps;


[Row_Domain,Col_Domain]=size(Domain); % basic information about Domain-values (Domain-axis)
No_Domain_steps=Col_Domain-1;
Domain_step=max(Domain)/No_Domain_steps;


%%%--------Step---ONE---------(Modifying MFs)-------------------------------
%%%-----------------------------------------------------------------------
    % Add end values for A and B to cover the whole range of Domain-values....to avoid giving negative values in case
    %% of not defining the end Domain-values of A and B.

    if min(Domain)<A(1,1)
        AA(1,1)=min(Domain);
        AA(1,2)=0;
        A=[AA(1,1) AA(1,2);A];
    end
[Row_A,Col_A]=size(A);   % Basic information about the MF of A, it is here to take into account any change
                         % due to the change in dimensions after the first condition

    if max(Domain)>A(Row_A,1)
        AA((Row_A+1),1)=max(Domain);
        AA((Row_A+1),2)=0;
        A=[A;AA((Row_A+1),1) AA((Row_A+1),2)];
    end

    if min(Domain)<B(1,1)
        BB(1,1)=min(Domain);
        BB(1,2)=0;
        B=[BB(1,1) BB(1,2);B];
    end
[Row_B,Col_B]=size(B);   % Basic information about the MF of B, it is here to take in
```

```
to account any change
                        % due to the change in dimensions after the first condition

    if max(Domain)>B(Row_B,1)
        BB((Row_B+1),1)=max(Domain);
        BB((Row_B+1),2)=0;
        B=[B;BB((Row_B+1),1) BB((Row_B+1),2)];
    end
%%-------------------------------------------------------------------------
%%-------------------------------------------------------------------------

%%%--------Step---TWO---------(Interpolate for Domain steps)--------------------
%%%------------------------------------------------------------------------
A=interp1(A(:,1),A(:,2),Domain);
A=[Domain;A];
A=A';

B=interp1(B(:,1),B(:,2),Domain);
B=[Domain;B];
B=B';
%%-------------------------------------------------------------------------
%%-------------------------------------------------------------------------

%%%--------Step---THREE--------(Identfy the operator and perform operation)
%%%------------------------------------------------------------------------

if strcmp(operator, 'minimum'),
    C=min(A(:,2),B(:,2));
end

if strcmp(operator, 'maximum'),
    C=max(A(:,2),B(:,2));
end

C=[(Domain') C];

[Row_C,Col_C]=size(C);  % Basic information about the MF of C
for w=1:Row_C
    if C(w,2)<0
        C(w,2)=0;
    end
end
```