## 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

#### $\mathbf{BY}$

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

#### **TABLE OF CONTENT**

1 IN	TRODUCTION	1
1.1	Objectives of the analysis	1
1.2	Report organization	1
1.3	Summary of the results	2
2 SY	STEM DESCRIPTION	3
2.1	Lake Huron primary water supply system (LHPWSS)	3
2.1.1	Intake system	5
2.1.2	Water treatment system	7
2.1.3	Conveyance and storage systems	8
2.2	Elgin area primary water supply system (EAPWSS)	9
2.2.1	Intake system	11
2.2.2	Water treatment system	11
2.2.3	Conveyance and storage systems	12
3 M	ETHODOLOGY FOR SYSTEM RELAIBILITY ANALYSIS	13
3.1	Multi-component system representation	13
3.2	Capacity and requirement of system components	15
3.2.1	System component capacity membership function	19
3.2.2	System component requirement membership function	21
3.2.3	Standardization of membership functions	22
3.3	Calculation of fuzzy performance indices	24
3.3.1	System-state membership function	25
3.3.2	Acceptable level of performance membership function	27

3.3.3	System-failure membership function	29
3.3.4	Fuzzy reliability-vulnerability index	32
3.3.5	Fuzzy robustness index	38
3.3.6	Fuzzy resiliency index	40
4 AN	NALYSIS OF THE LAKE HURON SYSTEM	43
4.1	LHPWSS system representation and data	43
4.2	Results	43
4.2.1	Assessment of the fuzzy performance Indic ices	43
4.2.2	Importance of different membership function shapes	48
4.2.3	Significance of system components	51
5 AN	NALYSIS OF THE ELGIN AREA SYSTEM	54
5.1	EAPWSS system representation and data	54
5.2	Results	54
5.2.1	Assessment of the fuzzy performance Indic ices	55
5.2.2	Importance of different membership function shapes	55
5.2.3	Significance of system components	59
6 C(	ONCLUSIONS	62
7 RI	EFERENCE	66

#### **LIST OF FIGURES**

Figure (1): The city of London regional water supply system	4
Figure (2) Schematic representation of the LHPWSS	6
Figure (3) Schematic representation of the EAPWSS	10
Figure (4) Water supply system layout	14
Figure (5) System integrated layout for the reliability analysis calculation	15
Figure (6) Support and a-cut of the fuzzy membership function	18
Figure (7) Membership function development using design capacity and overload	
capacity	21
Figure (8) Supply requirement membership function	22
Figure (9) Input data for LHPWSS	28
Figure (10) Fuzzy representation of the acceptable failure region	29
Figure (11) Typical example of a multi-component system	33
Figure (12) Calculation of the system-state membership function for the multi-com-	iponent
system	34
Figure (13) Overlap area between the system-state membership function and the	
acceptable level of performance	35
Figure (14) Flow chart for the fuzzy reliability-vulnerability index calculation	37
Figure (15) Flow chart for the fuzzy robustness index calculation.	39
Figure (16) Flow chart for calculation of the fuzzy resiliency index	42
Figure (17) LHPWSS system integrated layout- part 1	44
Figure (17) LHPWSS system integrated layout- part 2	45
Figure (17) LHPWSS system integrated layout- part 3	46
Figure (18) Acceptable levels of performance	47
Figure (19) Resulting system-state membership functions for triangular and trapezo	oidal
input membership functions	50
Figure (20) System-failure membership functions using triangular and trapezoidal	shapes.
	50

Figure (21) System-state membership function change for different system components.
52
Figure (22) EAPWSS system integrated layout- part 156
Figure (22) EAPWSS system integrated layout- part 257
Figure (23) Resulting EAPWSS system-state membership functions for triangular and
trapezoidal input membership functions59
Figure (24) System-state membership function change with introduction of system
components60
Figure (25a) Fuzzy performance indices for the LHPWSSS and EAPWSS systems for the
triangular membership function shape63
Figure (25b) Fuzzy performance indices for the LHPWSSS and EAPWSS systems for the
trapezoidal membership function shape63

#### **LIST OF TABELS**

Table (1) MF calculations for a multi-component system	-33	
Table (2) The LHPWSS system fuzzy performance indices for different membership		
function shapes	-49	
Table (3) System fuzzy performance indices change due to the improvement of PAC		
transfer pump capacity	-52	
Table (4) Change in the system fuzzy performance indices due to change in the maxim		
capacity of the pac transfer pump	-54	
Table (5) The EAPWSS system fuzzy performance indices for different membership		
function shapes	-58	
Table (6) System fuzzy performance indices change due to the change in the PAC		
maximum capacity	-61	

#### 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 m³/day). The plant's individual components are designed with a 35% overload capacity resulting in the maximum capacity of 454,600 m³/day. The current daily production, based on the annual average, is 157,000 m³/day with a maximum production value of 264,000 m³/day in 2001.

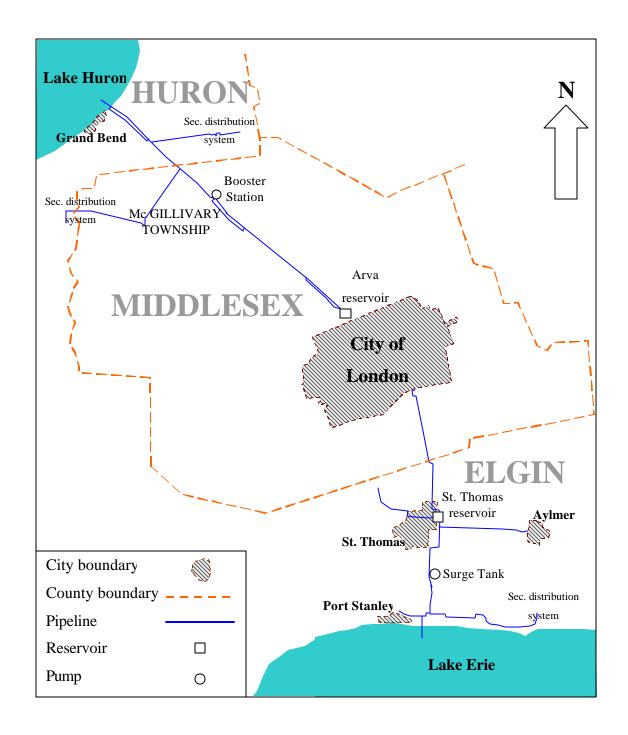


Figure (1): The City of London regional water supply system.

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 m³/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 m³/day.

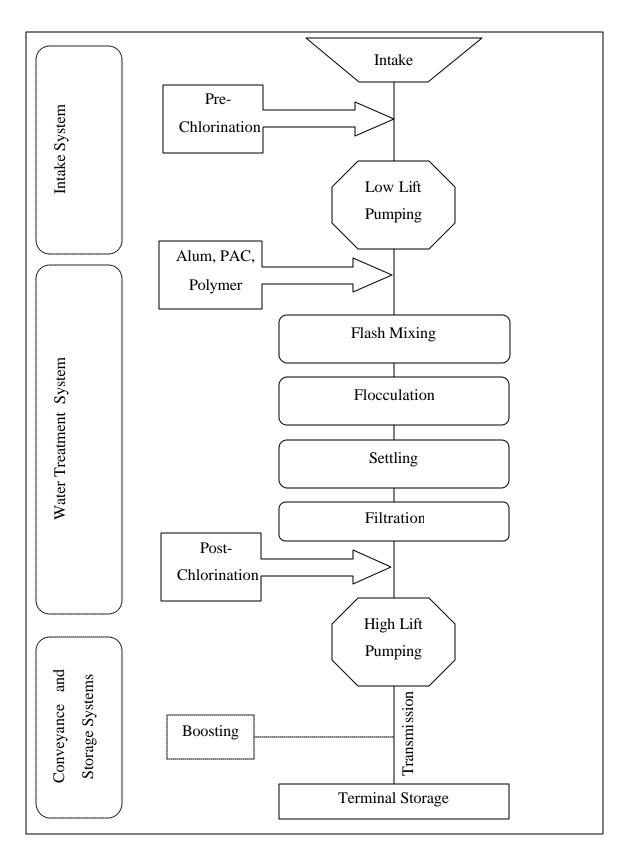


Figure (2) Schematic representation of the LHPWSS.

#### 2.1.2 Water treatment system

Water from the bw 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 m³/day and 170,000 m³/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<sup>3</sup> 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<sup>3</sup>/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<sup>3</sup>/day, with an average daily flow of 52,350 m<sup>3</sup>/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.

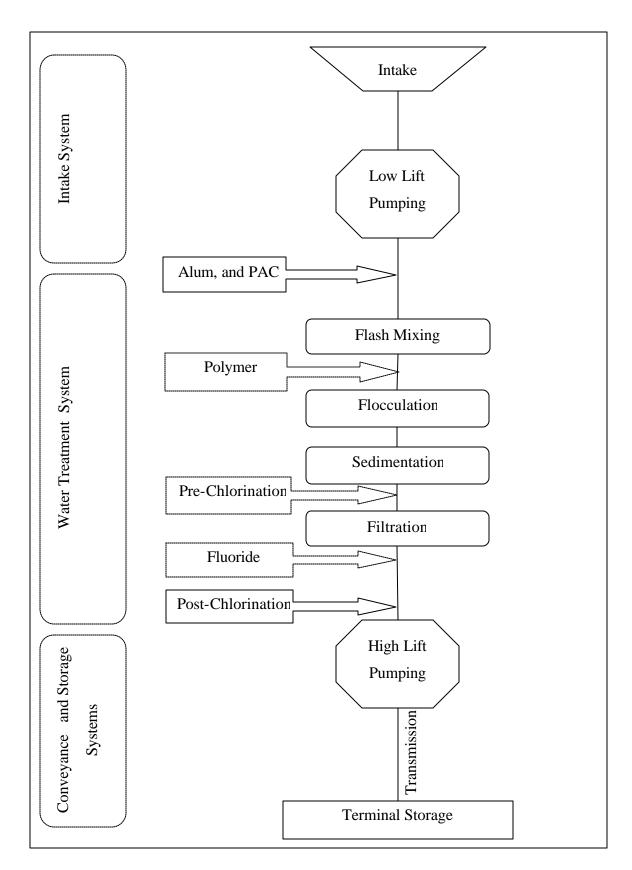


Figure (3) Schematic representation of the EAPWSS.

#### 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 m³/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 m³/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<sup>3</sup>/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 m³ 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.

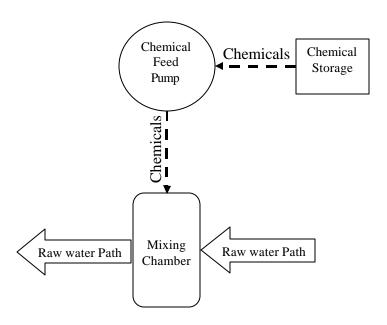


Figure (4) Water supply system layout.

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.

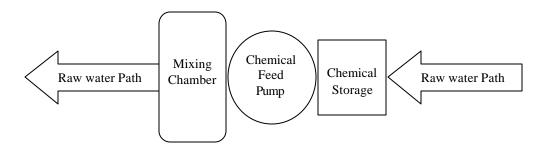


Figure (5) System integrated layout for the reliability analysis calculation.

#### 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:

$$\tilde{X} = \{(x, \mu_{\tilde{x}}(x)) : x \in R; \mu_{\tilde{x}}(x) \in [0,1]\}$$
 .....(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  $\alpha$  -cuts. The  $\alpha$  -cut is the ordinary set of all the elements belonging to the fuzzy set whose value of membership is a or higher, that is:

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

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

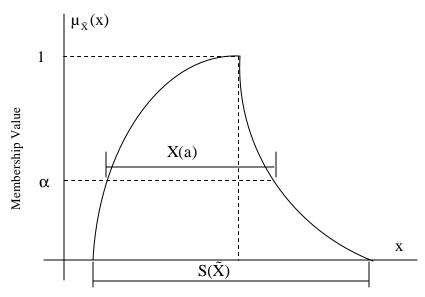


Figure (6) Support and a-cut of the fuzzy membership function (after Ganoulis, 1994).

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

- o Earth Tech Canada Inc.,2000;
- o Earth Tech Canada Inc.,2001;
- o American Water Services Canada-AWSC, 2003a;
- o American Water Services Canada-AWSC, 2003b; and
- o 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 (m³). Capacity of pumps is measured using flow units, cubic meter per day (m³/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:

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

where

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

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

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

$$\mu_{\tilde{A}}(x) = \begin{cases} 0, & \text{if } x \leq (1-2c)a \\ \frac{x - (1-2c)a}{(1-1.5c)a - (1-2c)a}, & \text{if } x \in [(1-2c)a, (1-1.5c)a] \\ 1, & \text{if } x \in [(1-1.5c)a, (1-0.5c)a] \\ \frac{a - x}{a - (1-0.5c)a}, & \text{if } x \in [(1-0.5c)a, a] \\ 0, & \text{if } x \geq a \end{cases} \qquad \dots \dots (5)$$

where

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

(1-2c)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.

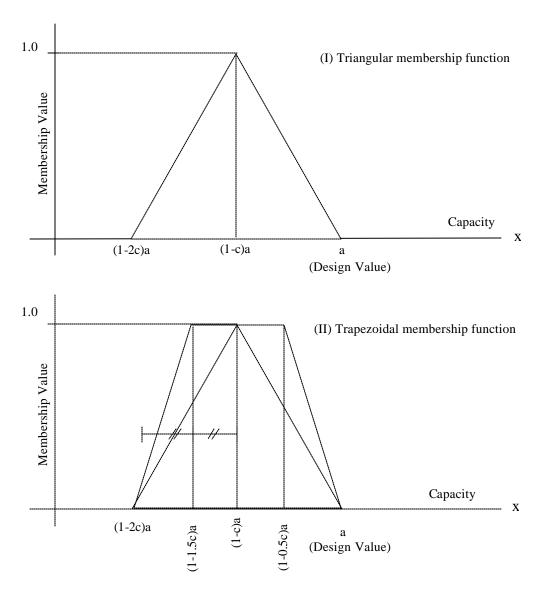


Figure (7) Membership function development using design capacity and overload capacity.

#### 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.

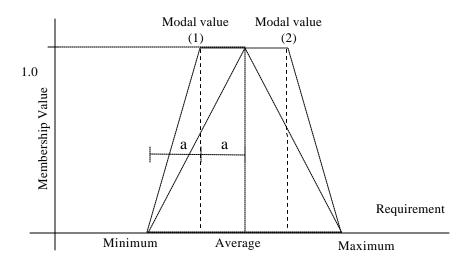


Figure (8) Supply requirement membership function.

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 (m³). 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<sup>3</sup>) is defined as follows:

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

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:

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

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 mth 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:

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

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:

$$\tilde{\mathbf{S}}(\mathbf{u}) = \min_{\mathbf{N}} \left( \tilde{\mathbf{S}}_{1}, \tilde{\mathbf{S}}_{2}, \dots, \tilde{\mathbf{S}}_{\mathbf{N}} \right) \qquad \dots \dots (9)$$

where:

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

$$\left(\tilde{S}_{l},\tilde{S}_{2},.....,\tilde{S}_{N}\right)$$
 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).

$$\tilde{\mathbf{M}}_{i} = \tilde{\mathbf{X}}_{i} \ (-) \ \tilde{\mathbf{Y}}_{i}$$
  $\forall i = 1, 2, \dots, n$  .....(10)

where:

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

 $\tilde{\boldsymbol{X}}_{i}$  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

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

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.

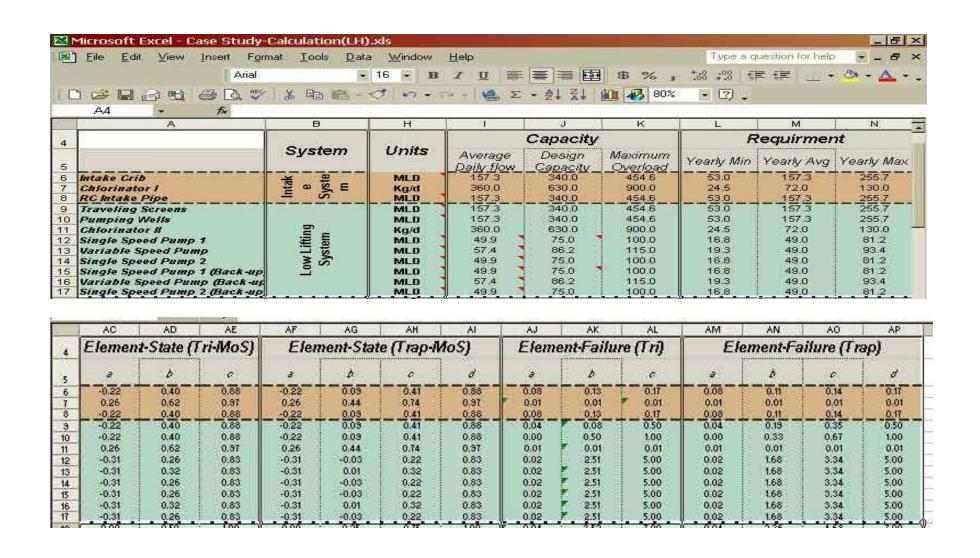


Figure (9) Input data for LHPWSS

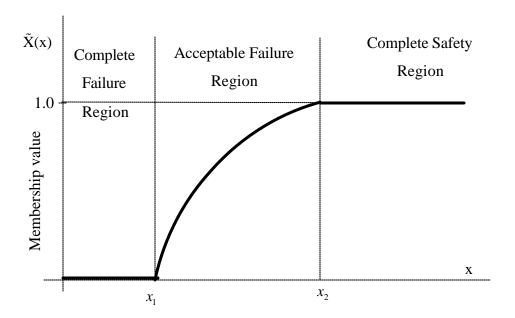


Figure (10) Fuzzy representation of the acceptable failure region.

#### 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)

where:

 $\tilde{T}(a)$  is the system fuzzy maximum recovery time at  $\alpha$  -cut

(as defined by Equation 2);

 $t_{j_1}(a)$  is the lower bound of the j-th recovery time at  $\alpha$  -cut

(as defined by Equation 2);

 $t_{\scriptscriptstyle 2_{\scriptscriptstyle 1}}(a)$  is the upper bound of the j-th recovery time at  $\alpha$  -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)

where:

 $\tilde{T}(t)$  is the whole system-failure membership function; and  $\left(\tilde{T}_{l},\tilde{T}_{2},\ldots,\tilde{T}_{M}\right)$  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)

$$\tilde{T}(t) = \tilde{T}_{c}(t)$$
 .....(14)

given

$$S(\tilde{T}_c) = m \underset{N}{a} x \left( S(\tilde{T}_1), S(\tilde{T}_2), \dots, S(\tilde{T}_N) \right)$$

and .....(15)

$$\tilde{T}_{c}(1) = \max_{N} (\tilde{T}_{l}(1), \tilde{T}_{2}(1), \dots, \tilde{T}_{N}(1))$$

where:

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

 $\tilde{T}_{\mbox{\tiny c}}(t)$  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);

 $\left(S(\tilde{T}_l),S(\tilde{T}_2),...,S(\tilde{T}_N)\right)$  are the support of the N components recovery time membership functions (as defined by Equation 3);  $\tilde{T}_c(1) \text{ is the controlling recovery time membership function at the } \alpha \text{ -cut = 1}$  (as defined by Equation 2); and  $\left(\tilde{T}_l(1),\tilde{T}_2(1),...,\tilde{T}_N(1)\right) \text{ are the recovery time membership functions}$  at  $\alpha$  -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.

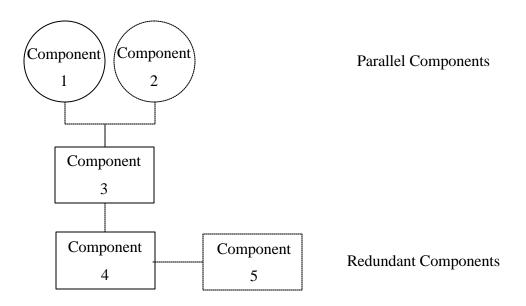


Figure (11) Typical example of a multi-component system.

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	
Component 2	(1,2,3)		NA	Min [ (2,4,6), (1,3,5),
Component 3	(1,3,5)	NA	NA	(1,2,3)]
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.

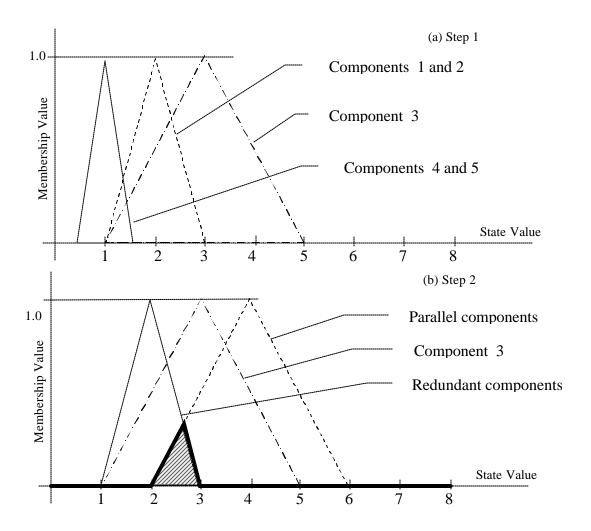


Figure (12) Calculation of the system-state membership function for the multi-component system.

The compatibility measure (CM) is calculated as:

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

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

$$FuzzyReliability-VulnerabilityIndex = \frac{\max\limits_{i \in K} \left\{ CM_{_{1}}, CM_{_{2}}, ......CM_{_{i}} \right\} \times LR_{_{max}}}{\max\limits_{i \in K} \left\{ LR_{_{p}}LR_{_{2}}, .....LR_{_{i}} \right\}}.....(17)$$

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);

 ${\rm CM_i}$  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.

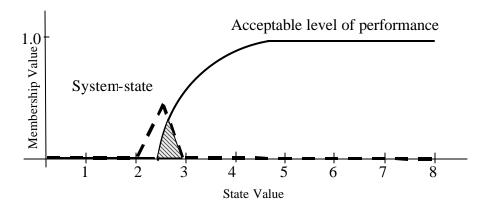


Figure (13) Overlap area between the system-state membership function and the acceptable level of performance.

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 componentstate membership functions. Both types of membership functions, triangular and trapezoidal, are constructed;
- o 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.

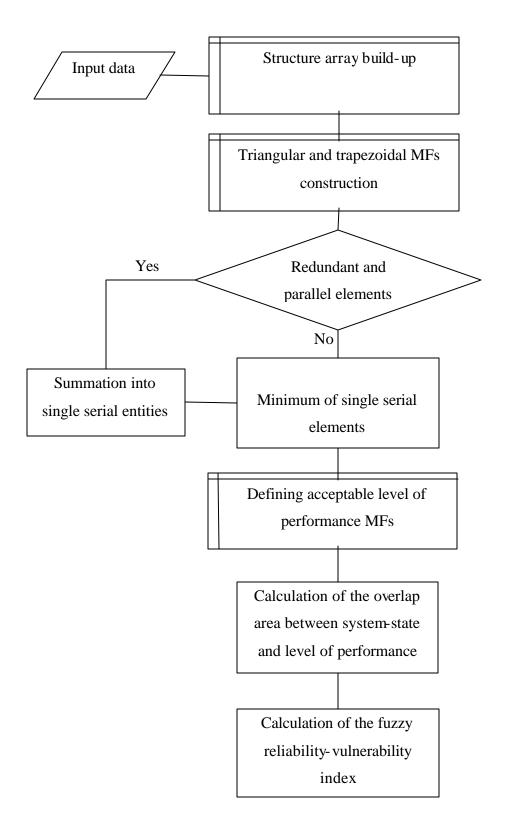


Figure (14) Flow chart for the fuzzy reliability-vulnerability index calculation.

## 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:

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

where:

CM<sub>1</sub> is the compatibility measure before the change in conditions; and

CM<sub>2</sub> 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 componentstate membership functions. Both types of membership functions, triangular and trapezoidal, are constructed;

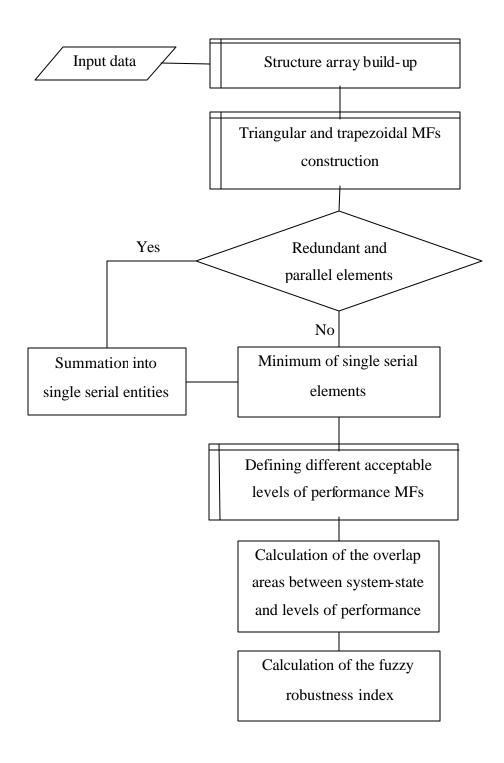


Figure (15) Flow chart for the fuzzy robustness index calculation.

- o Step (2); storing the input data in an appropriate data format (i.e., structure array).
- O 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):

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

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.

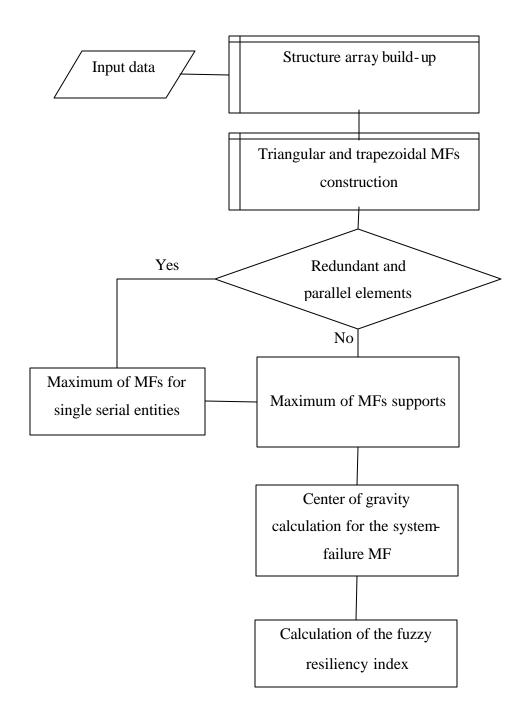


Figure (16) Flow chart for calculation of the fuzzy resiliency index.

#### 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.

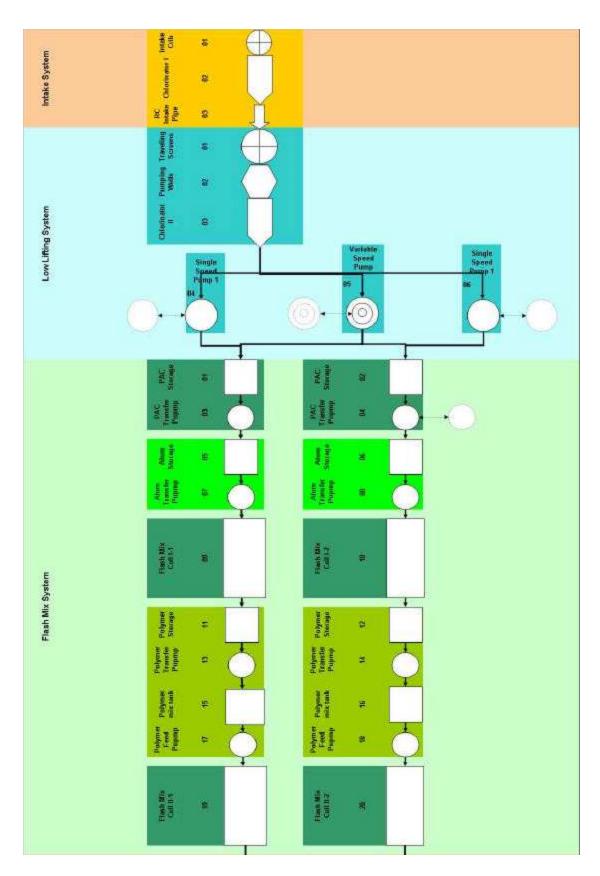


Figure (17) LHPWSS system integrated layout- Part 1

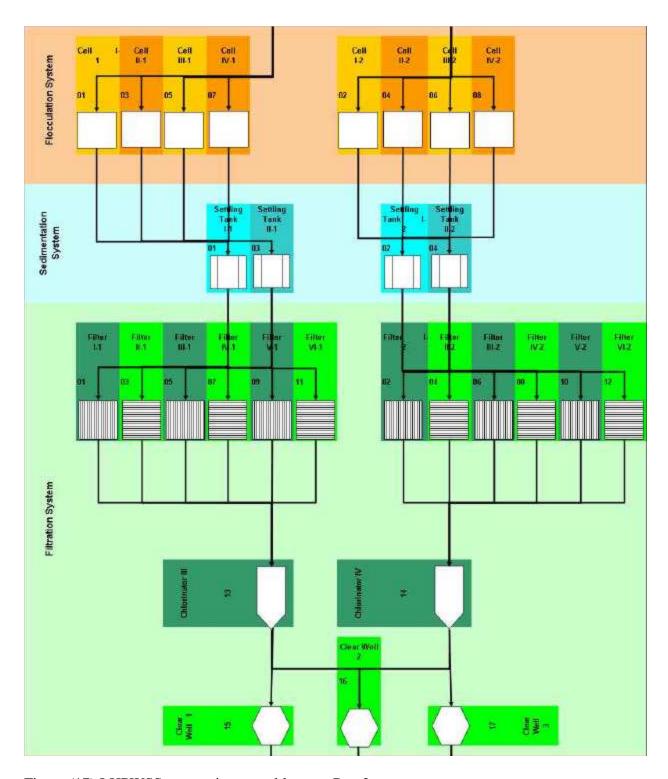


Figure (17) LHPWSS system integrated layout- Part 2

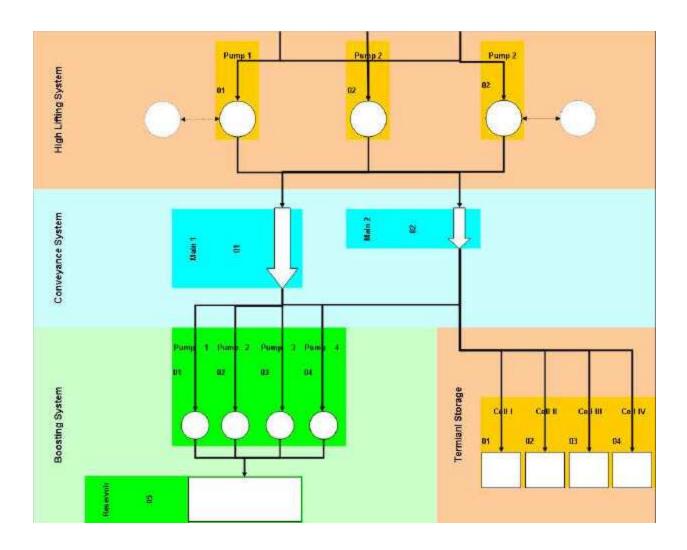


Figure (17) LHPWSS system integrated layout- Part 3

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.

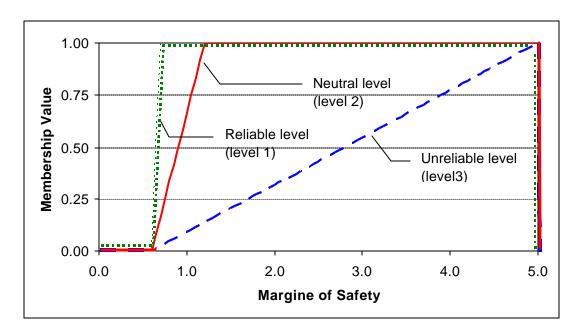


Figure (18) Acceptable levels of performance.

### **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.

Triangular MF	Trapezoidal MF
0.699	0.642
NA*	NA*
-2.120	-2.473
-2.120	-2.473
0.017	0.017
	0.699 NA* -2.120 -2.120

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

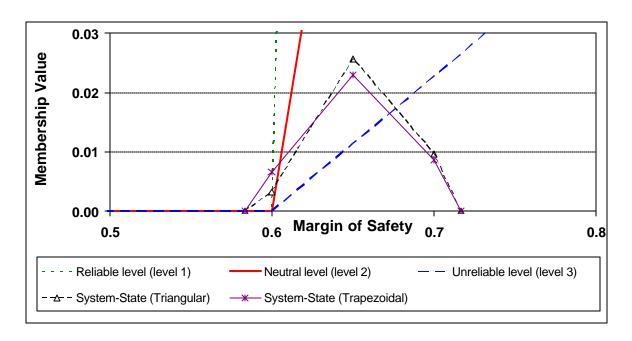


Figure (19) Resulting system-state membership functions for triangular and trapezoidal input membership functions.

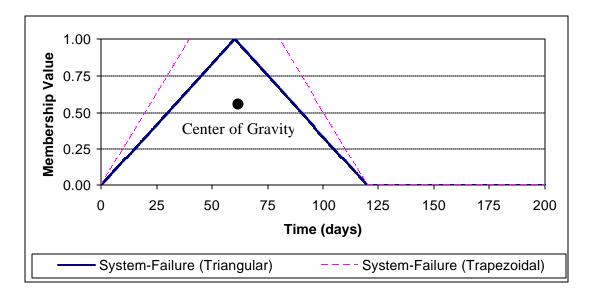


Figure (20) System-failure membership functions using triangular and trapezoidal shapes.

### **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.

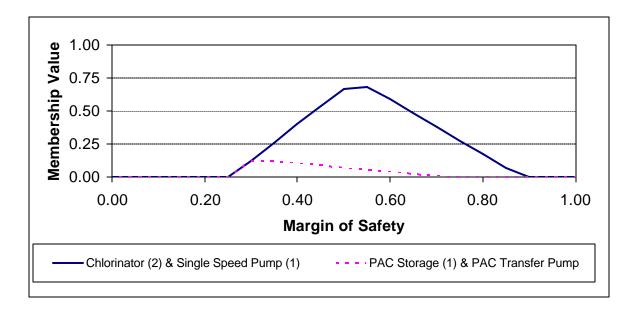


Figure (21) System-state membership function change for different system components.

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.

	Percentage change of the maximum capacity		
Fuzzy performance index	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.

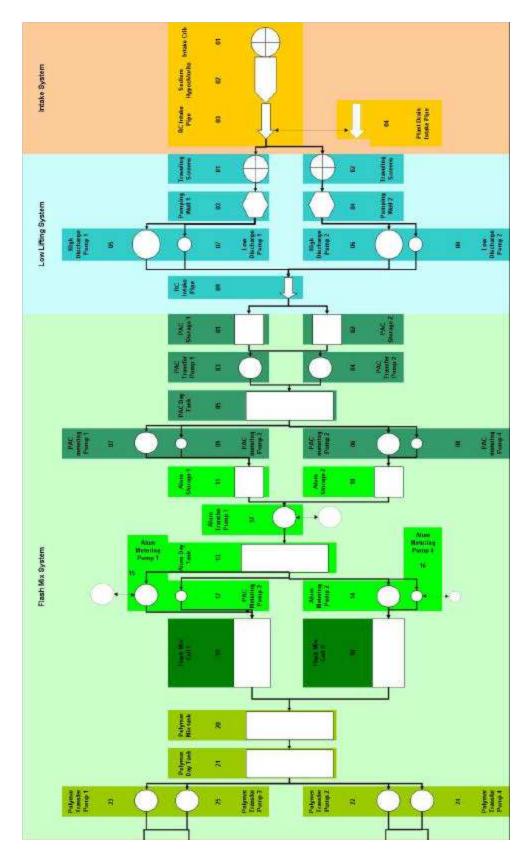


Figure (22) EAPWSS system integrated layout- Part 1.

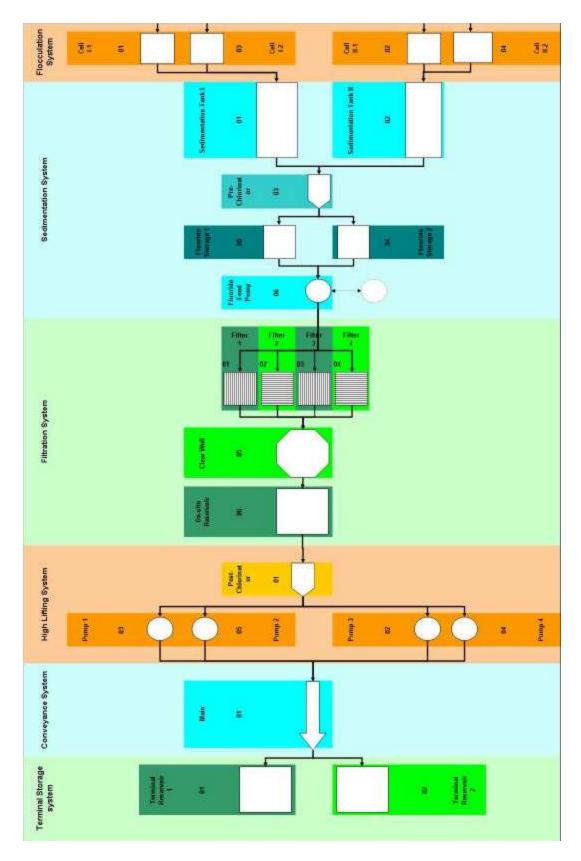


Figure (22) EAPWSS system integrated layout- Part 2

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.

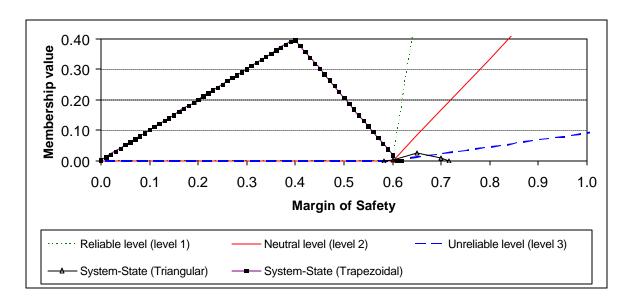


Figure (23) Resulting EAPWSS system-state membership functions for triangular and trapezoidal input membership functions.

# **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.

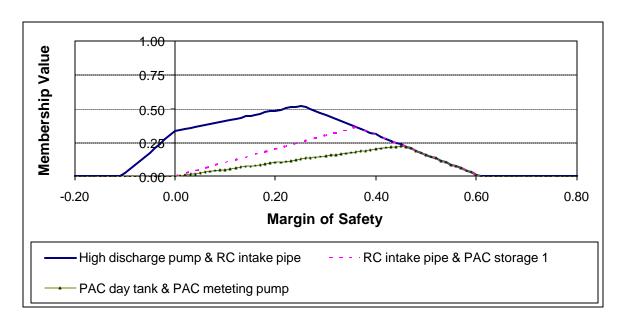


Figure (24) System-state membership function change with introduction of system components.

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.

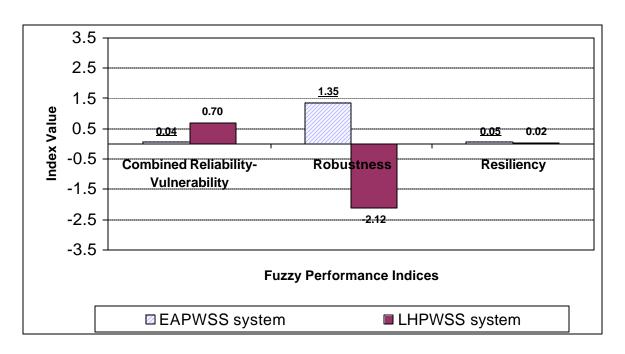


Figure (25a) Fuzzy performance indices for the LHPWSSS and EAPWSS systems for the triangular membership function shape.

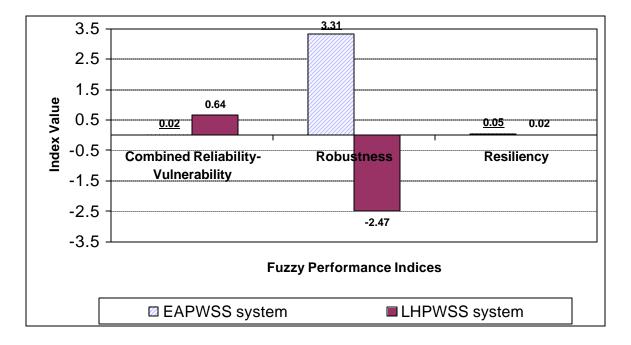


Figure (25b) Fuzzy performance indices for the LHPWSSS and EAPWSS systems for the trapezoidal membership function shape.

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)

	og eggene	1940000	1	Capacity	Č-x-	R	equirmen	t	Normal	ized Capac	ty (TRI)	Normali	zedRequiri	ment(Trl)	Non	nalized C	apacity(	Trap)
	System	Units	Avarage Dody Ana	Dange	Maximum Overload	Yearly Min	Yearly Zvg	Yearly Mor	Average Daily flow	Deagn Canada	Maximum Overlead	Yearly Min	Yearty Avg	Yearty Adam	Average Carly flow	Capacity		Maximum
selve Colly	e E	MED	Detly flow	3400	454.5	62.0	167.9	Max	_Pak.096	0.75	1.00	0.12	0.35	0条	0.86	0.55	0.87	Queologi 1.00
bioriestor)	Intake	Keyli	360.0	610.0	900.0	245	72.0	100.0	11.00	0.70	1.00	0.00	0.00	n.re	0.00	11.65	DIK	1.00
Clateke Pijse	= 6	MLD	193	340.0	404.5	53.0	873	200.7	0.35	0.75	100	0.12	0.15	0.58	0.25	0.60	0.60	1.00
ervaling Screen: Inspire Wells		MLD MLD	167.3	3400	454.5 454.6	53.0 50.0	167.3 167.3	256.7 -256.7	0.35	075 075	1.00 1.00	0.12	0 H	0.56	0.35 0.35	0.55	0.67	1.00
Montutor E	9	Keyli	3800	6300	300.0	24.5	720	130.0	0.40	0.70	1.00	0.00	0.09	0.14	0.40	0.55	0.85	1.00
Jegde Speed Pasep 1	21	MLB	49.9	76.0	100.0	68	49.0	B12	0,50	0.75	5100	0.12	0.49	0.91	0.50	0.62	0,87	1300
vriable Syxied Parqui Ingle Spend Parqui 2	7 8	MLD MLD	37.4 49.9	750	100.0	19.3	40.0 40.0	81.2	0.50	0.75	1.00	0.17	0.47	08t	0.50	0.62	0.67	1.00
(rigite Speed Passp T (Back-up)	System System	MLD:	49.9	75.0	100.0	19.8	49.0	812	0.90	0.75	7.00	0.17	0.49	0.01	0.60	0.60	0.87	1.00
ferioble Speed Pivoto (Block-up)		MLD MLD	67.4	86.2 75.0	1950	19.3	29.0	班生	0.90	0.75	1.00	0.47	0.43	0.81	0.60	0.65	0.65	1.00
Angle Speed Pamp 2 (Back-ep) NC Storage Fank		THE STREET	0.0	750	150 D	0.1	0.2	03	0.00	0.79	1.00	0.00	0.00	0.00	0.00	0.0	DUE	100
AC Sharepe Fank		100	88	750	190.0	30	0.2	0.3	0.00	0.50	1.00	0.06	0.90	000	0.09	0.25	-0.25	1,00
AC Freuder Panys AC Tavader Panys		Kg/d	80.0	1000	300.0	227.0	419.7 419.7	867.3	0.20	060	1.00	0.74	1.40	2.22	0.30	0 45 0 40	B80 D00	1.00
laro Storage Flexit		10/	0.0	33.0	68.0	17	1990	7.0	0.00	0.50	Y.cb	0.00	0.25	0.11	0.00	0.25	5.75	1.00
Ann Storege Tank		100	0.0	33.0	66.0	统	2.1	7.0	0.00	0.50	1.00	0.00	0.15	0.66	0.00	0.3	875	1.00
Sans Transfer Poung Sans Transfer Poung	E	m741 m741	33	48	67 67	17	31	70	0.90	0.73	1.00	0.25	0.46	1.05	0.50	0.61	0.66 0.66	1.00
tuch 860r GeV F-T	System	MLD	767	1/00	227.5	20.5	77.2	127.9	0.36	\$75	100	0.72	0.34	0.58	6.35	0.55	1067	1.00
had Mir GeV F2		Mt.O	70.7	1708	327,9	365	77.2	12/19	035	0.75	100	0.12	0.34	0.96	0.96	0.55	8.67 0.75	100
olymer Storage Took olymer Storage Took	78	10,	8,0 0.0	6.6	13.6	0.0	44	8.8	0.00	0.60	1.00	0.00	0.33	0.66	0.00	0.75	0.75 8.75	1.00
bilyoter Touride Purop	6	Lie	0.0	33	84	00	2.0	4.2	0.00	0.60	1.00	0.00	0.33	0.05	0.00	0.25	0.75	1.00
olymer Transfer Plyrop olymer Mile Tank	2	Lite	9.0	3.2	68	00	21	4.2	000	0.40	1.00	0.08	0.15	0.65	000	0.25	0.76	1.00
tolymer Mile Took		102	0.0	84	6.0	00	22	4.4	0.00	0.60	1.00	0.00	0.33	0.66	0.00	0.3	826	1.00
Tolyrois Feet Purop		la:	0.000	0.4	0.8	1,0	a a	0.4	CD37	0:01	1.00	0.01	0.38	0.00	0.07	0.00	D.70	1.00
Selvent Paul Purop Seuls Min Cell S-T		MI D	797	1700	0.6	265	77.2	127.9	0.95	0.58 0.75	1.00	0.17 0.72	0.32	0.65	0.17	0.38	5.79 0.67	1,00
Sich Billy Call II-2		MLD	78.7	1700	227.3	29.5	77.2	127.9	0.36	0.75	1.00	0.12	0.14	0.66	0.96	0.55	0.67	1:00
WC Traveler Plump		Kg/ti	90.0	190.0	300.0	237/0	497	923	0.20	0.60 0.60	100	971	1.41	222	0.70	(7.45)	0.80 0.80	1.00
26067 26062	3.6	MLD MLD	8.0	25.3	426 425	66	19.9	30.0	U.19	0.69	100	G16	0.45	0.76 975	0.10	0.00	0.80	1.00
140%	ê e	MLD	8.0	25.3	42.6	9.6	19.3	33.0	0.19	0.49	1.00	0.96	0.45	8.75	8.09	0.99	0.80	1.00
MAZ	System	90,0	90 80	25.3	42.5	5.6	193	32.0 32.0	0.49	0.93	1.00	0.48	0.45	0.75 0.75	0.19	0.08	080	1.00
CeO 8-7	occulatio Bystem	MLD MLD	8.0	25.3	42.5	5.6	19.3	30 D 50 D	0.12	0.50 0.50	100	0.16	0.45 0.45	0.75	0.19	0.50	DED	1.00
SHIBET	<b>E</b>	MLD	8.0	25.3	428	9.6	接筆	32.0	0.39	0.69	1,00	0.18	0.45	0.76	0.19	0.39	D80	7.00
Felting Tank (-)		MLD	BD 78.7	1700	20 5 20 7	66	19.0	20:0	0.98	DS9	1.00	0.04	0.45	0.75-1	0.10	U 10	889	1.00
Settling Tank (-2	Sedims modon System	M.D	787	1900	227.5	133	30.6	63.0	0.35	D.FS	1.00	0.00	0.17	0.26	0.98	0.65	Det	1.00
etting Fank 9-1	252	MLD.	78.7	1700	227.5	1)3	38.6	619	0.35	075	1.00	0.06	0.17	0.29	035	0.66	DBJ	1:00
Settling Tank 8-2 War I-1		MLD.	- 37	28.3	27.9	133	12.2	- 8.9	0.36	0.76	100	0.05	0.17	0.00	035	0.55	0.87	100
War 12		MID	111	28.3	37.0	64	129	25.3	0.26	0.75	1.00	0.12	0.34	0.68	0.05	0.50	Der	1.00
the 81 Yor 82		MLD MLD	13.1	28.3	37.9	88	129	213 213	0.35	0.75	1.00	6:12	0.14	0.98	0.35 0.35	0.66	B 67 D 67	1.00
War K-f	102	MLD.	131	28.3	37.9	24	129	21.3	1036	0.76	100	0.12	0.14	0.96	6.35	0.99	0.67	100
Day #-2	E	MLO	133.1	26.7	37.9	2.4	129	213	0.36	0.75	1.00	0.13	0.15	0.50	0.36	0.55	0.07	1.00
The MT	16/6	MLD	19.1	28/3	37.9 37.9	4.8 4.9	12.9	213	0.36	0.75	1.00	0.12	034	0.66	0.35	0.66	0.67	1.00
When V-2	6	MID	13:1	29.7	37.9	4.4	12.8	7213	0.36	0.79	1.00	0.12	0.34	0.65	9.36	0.55	0.67	1.00
West U-f	thration	MOD	1111	20,7	373	11	12.9	2213	10.00	0.75	1.00	0.01	0.14	0.60	0.05	0.60	D.EUT	1,00
Mar W-7 Mer W-7	age of the same of	MLD MLD	131	26.5	37.9	4.8	129	21.5 21.3	0.35	0.75	100 100	0.12	0.34 0.34	0.66	0.35 9.35	0.95	0.67 0.67	1.00
Chiloriniator N	14.	Kekt	360.0	630.0	900.0	34.0	545	91.0	0.40	0.70	1.00	0.04	0.36	0.10	0.40	0.65	D85	1.00
Morkstor N		KgN	580.0	630.0 105.9	100 D	34.0	655	Din	0.46	0.70	1.00	0.0s	0.86	0.10	0.00	0.55	D85	1.00
Clear Well 1: Clear Well 2		MLD	524	105.9	109.4	11	129	21.3	0.33	0.68	1.00	0.00	0.00	0.13	0.33	0.50	D83	100
Skar Wed S	30,00000	MI.D	424	1069	399.4		129	- Carlon	0.39	0.85	100	0.00	0.19	0.13	0.83	0.50	D.83	1.00
Sings I	- 08	MLD	787	89.4 89.4	100.1	25.9	76.9	127.9	0.79	0.00	1.00	0.05	0.76	1.20 1.20	0.79	0.64	0.96 0.95	1.00
Pump 2 Pump 3	High Lifting System	MLD MLD	767	89.4	100.1	253	75.5	127.9	0.79	0.03	1.00	0.75	0.76	1.20	0.79 0.79	VI.59	0.96	1.00
lengo f (Eleck ap)	129	MLD	787	68.4	100-1	25.3	75.9	127.9	0.79	0.99	1.00	0.25	0.76	1.29	0.79	0.94	0.96	1.00
Vege 2 (Such as)		MLD		- B0.4	391	63	759	32/3	0.78	081	3.00	134	0.75	1200	0.70	0.62	0.85	100
AND THE PROPERTY OF THE PROPER	ě s	0100 HC212	167.3	1923	227.3	25.3	75.9	127.9	0.69	0.86	1.00	0.41	0.33	0.66	0.69	0.77	0.82	1.00
Nate of	Conveyan	ML0			1000			100000				4,000		1	3.574.0			1
	2 6		197.0	1927	227.9	35.3	75.9	127.9	0.69	1085	1.00	¢n.	0.22	0.66	0.69	0.77	0.92	1.00
Sale 2	ß	MLD	350		2011	40.07	1000	10.3			100	277	0.00	Loon	U.S.		0,00	1,00
ting f	9 -	10.0	45	523	100.1	4.6	62.3	1001	0.06	0.62	100	0.06	0.62	100	006	0.09	076	1.00
Vote 2	20 20	MLD.	1.6	57.3	100.1	26	623	100.1	0.05	0.63	100	0.06	0.12	100	0.05	0.20	0.76	1.00
http: 3 http://d	Boosting System	MLD MLD	4.6	50.3 50.3	100 1	46 46	52.5 52.3	100.1	0.05	952 952	1.00	0.05	0.32	100	1805 0.05	0.06	12.76 13.76	100
Securior	B. W.	MLD	0.0	91	18.2	00	81	192	0.00	0.60	1.00	0.00	0.90	100	0.00	0.35	876	1.00
WII 0	8 A.F	MLD.	DD	157	27.2	0.0	4.6	15.5	0.00	0.60	1.00	0.00	0.14	0.40	0.00	0.25	0.75	1,00
e0.2	Fermina Storoge System	MI.O	DD	197	37.1	0.0	95	133	0.00	0.50	1.00	0.00	0.14	0.46	0.00	0.85	076	1.00
ed 2	Part Stor	MLD.	0.0	13.7	27,1	0.0	66	13:3	0.00	0.50	1.00	0.00	0.24	0.46	0.00	0.25	875	1.00
60.4		Mr.th	-0.0	137	77.3	0.0	8.8	13.9	800	0.60	1.00	0.00	0.04	0.48:-	960	0.25	8.25	1.00

1	-	100000	Normalized Requirment(Trap)				Elemen	nt-State (	Tri-MoS)	Eler	nent-Sta	te (Trap-li	NoS)	Eleme	ent-Failu	re (Tri)	EI	ement-Fa	illure (Tra	(QF
-24 8 24 2 8 2 1 8 2	System	Units	Attarage Cally flow	Capacity		Maximum		Ь	e e	4	Ď	e	e	-	b.	0		ò	4	ď
lehe Colh	2 E	MLD.	0.12	0.46	0.45	0.56	0.22	0.40	0.69	4.22	0.09	QA)	0.89	0.0e	0.13	71.0	0.00	0.16	0.14	0.17
Morington F	old in the state of the state o	KeNi	10.00	II.41	241	D14	0.35	0.60	0.07	0.06	0.64	11.74	0.97	0.01	0.01	0.01	bot	(10)	0.01	0.01
Saturker Pipes	= 6	MLD	0.12	0.65	0.45	0.56	4141	0.40	0.08	-0.22	0.09	0.41	1166	0.08	010	0.17	0.06	0.11	0.14	un.
resiting Screens: repring Wells	147407.0	MLD MLD	0.12 8.12	0.46	0.45	0.56 0.56	-0,22 -0.22	0.40	0.80	41.22 -0.22	0.09	0.41	0.86	0.04	0.08	0.50 1.00	DDM 0000	0.20	0.35	1.00
donnator f	2	Keyli	0.08	0.51	0.11	0.14	0.36	0.62	0.97	0.26	0.44	0.74	0.97	0.01	0.01	0.01	0.01	0.01	0.01	8.01
kyde Speed Pasys F	LowLifting System	MLD	0.97	0.99	0.65	0.81	0.31	0.76	0.98	-0.34	-0.03	0.32	0.89	0.02	260	9.00	0.02	1.98	3.34	5.00
vilable Systed Pursp ingle Spend Pamp 2	2 2	MLD.	0.17	0.55 0.65	0.65	Dát	-031	0.30	0.03	431	-003	0.32	089	0.00	2.61	5.00	0.02	1.00	334	5.00
ingle Speed Passp T (Back-up)	3 "	Mt.B	0.17	0.66	0.66	0.81	0.31	9,26	0.83	-0.31	0.03	0.22	6.63	0.02	2.61	5.00	0.02	1.88	334	6,00
erioble Speed Pirosp (Good-up)		MLD MLD	0.57	0.96	0.62	0.81	0.31	0.32 0.26	0.00	431	0.01	8.30	0.83	0.00	261 261	6.00	0.005	1.69	834	500
imple Speed Pamp 2 (Black-up) AC Storage Fact	TOTAL CONTRACTOR	THE STATE OF THE S	0.00	0.00	0.00	0.00	0.00	050	1.00	0.00	0.25	0.75	100	0.04	342	7.00	0.04	2.30	4.66	7.00
AC Shrepe Fant		100	0.00	0.06	0.00	9:00	6.00	0.50	1.00	0.00	0.25	0.75	1.00	0.04	342	7:80	0.04	236	4.00	7.00
NC Fresider Flyings NC Frender Flyings		Kghi	0.76 0.76	1.72	181	2.22	2.02	0.00	0.24	510	-taf	0 92 -0 92	0.34	0.84 0.04	3,62	7.00 7.00	0.04 0.04	236	4.68	7.00 7.00
laro Storage Tank		Kg/d	0.08	Dide-	0.00	D.(1	40.71	0.46	0.97	(4.1)	0.77	0.60	0.97	0.04	342	1.00	7004	2.56	4.00	7.00
lani Storege Tank		(8)	0.08	0.06	0.08	0.11	0.11	0.46	0.97	40.00	0.17	0.69	78.9	0.04	342	7.10	0.04	236	4.60	2.08
tani Transhe Panin tani Transhe Panin	E	m7/d m2/dr	0.26	0.57	0.76 0.76	7.06	-0.66 -0.66	0.26 0.26	0.75	47.55 47.65	014	0.29	0.75	0.04	3.62	7,00	D/04 D/04	2.96	4.68	7,00
tuch filler Co.W.F.T	System System	MLD	0.12	0.48	0.45	0.58	-0.22	0.41	0.58	422	0.70	0.92	0.88	0.50	7.25	14.00	0.50	5/00	550	14.00
had Mir GeV i 2		Ut.0	0.12	0.45	0.48	9.98	JU22	0.41 0.18	0.88	30.00	0.90	0.42	889	6.50	COPE.	14:00	0.90	500	950	14 00
olymer Storago Tosh olymer Storago Tosh	2 X	16,	000	0.49	0.49	0.66	-0.99	0.18	1.00	4.66 4.66	0.21	0.26	1.00	0.50	7.25	14:00	0.90	500 600	9.90	14:00
Hyother Touarder Parcep	List I	Line:	0.00	0.49	0.45	0.65	-0.65	0.10	1.00	-0.85	-0.78	0.26	1.00	0.50	7.75	16.00	0.50	500	0.50	14 00.
olymer Examine Purcy	문	Lin	0.00	0.42	0.45	0.65	5106	0.18	1.00	J185	-034	0.06	1.00	0.50	7.8	14.00	0.50	600	9.50	14 00
olymor Min Tank olymer Min Tank		102	0.00	0.49	0.49	0.66	-0.96	0.18	1.00	-0.66	024	0.26 0.26	1.00	0.60	7.25	14:00	050	5.90	950	14.10
blycoir Feed Plutp		1ac	0.11	0.01	0.69	Diff	-0.40	0.20	0.69	41.48	0.18	0.26	0.00	0.50	2.22	14:00	0.00	5.00	0.50	14.00
Signar Paul Plump Sech Min Cell 9-7		Mit Di	0.11	0.45	0.51	0.66 0.56	-0.48 -0.22	0.20	0.80	47.88	0.14	0.42	0.89	0.50	135	14.00	0.50	500	950	14.00
Sich Billy Call II-2		MLD.	0.12	0.45	0.45	0.66	0.22	0.41	0.86	40.22 40.22	0.10	0.42	0.88	0.50	7.25	14.00	0.50	500	9.50	14.00
AC Traveler Planta		KgN	0.76	1.72	7.81	2.22	202	0.80	0.24	212	and die	0.02	0.74	0.04	15.262	7.00	0.04	2.39	4.F8 D67	700
e0.62		MLD MLD	0.16	0.60	0.60	8.76 8.76	-0.96 -0.96	D.14 D.14	0.64 6.64	-0.56 -0.56	-0.21	Ø 19	084	0.00	0.60	1.00	0.00	0.13	0.67	100
1409	8 :	MLD	10.1B	0.90	0.60	0.76	0.99	0.14	0.64	2.58	0.21	0.19	0.84	0.00	0.60	1.00	-9.00	0.99	0.67	1.00
AU D-Z	58	90,0	0.16	0.50	0.60	0.75 0.75	0.96	0.14	8.94	0.66	0.21	0.19	0.64	0.00	0.60	1.00	0.00	0.33	0.62 0.67	100
203FT	occulatio System	MLD.	0.16 0.18	0.60	080	0.76	-0.56	D.14	0.04	42.55	-0.21	G 19 G 19	0.84 0.84	0.00	D.60	1 00	0,00	0.33	0.67	100
eli 8F1	Ø-	MLO	0.16	0.90	0.60	0.76	0.96	0.14	0.64	0.56	-0.21	0.19	0.84	6.00	0.60	1:00	0.00	0.99	0.67	1.00
eDN-2 esting Tank (-1		MUD	0.45	0.80	0.60	0.76	0.96	1014	0.54	0.50	0.57	010	094	0.00	0.90	100	0.00	0.33	0.00	1,00
cotting Tank (-2	adless rates	W.0	0.08	0.25	6.28	0.26	G DE	0.56	12.54	0.08	0.02	0.65	134	0.08	054	1.00	0.06	0.22	0.60	1.00
etting Fank 9-7	Sedio masso Syste	MLD.	0.06	0.29	0.33	8.28	0.06	0.98	fl.94	0.06	0,32	0.68	094	0.00	0.64	0.00	0.09	0.39	0.69	1,00
etaling Tank 6-2 Vew 5-5		MLD.		0.6	0.45	0.28	0.06	Date	0.50	61.22	0.32	0.65	0.94	0.04	0.64	100	0.04	136	6725	1.00
Mar 52		MLD	0.12	0.45	0.45	0.68	4.21	DAIL	0.96	40.22	0.92	0.42	086	0.04	502	10.00	11.5%	3.30	5.66	70.00
the 81		MLD MLD	0.12	0.45	0.45	0.66	0.22	D.41 B.41	0.08	40.20 40.20	0.90	0.42	0.89	0.04	502 502	10.00	0.04	3.36	6.68 6.68	10.00
War #2 Day #-1	162	M.O.	0.12	0.45	8.46	0.95	0.22	0.45	8.88	6.22	0.30	842	0.96	0.04	6002	10.00	5104	336	5.58	10.00
Dav #-2	E	MLO	0.12	0.65	0.45	0.55	-0.22	0.81	0.00	-0.22	0.10	0.42	Date	0.04	5.00	97.00	0.02	336	. n.co	10 00
The M-T The M-2	6	MID	0.12	0.45	0.45	0.58	-0.22	0.41 0.41	0.89	41.22 41.22	0.90	0.42	0.86	0.04	5 (E) 5 (E)	10.00	0.04	3.36	5,66 5,68	10.00
War V-2		MID	0.12	0.45	0.45	8.66	0.22	Dat	0.98	0.22	0.10	0.42	6.69	0.84	5.02	17.00	0.04	3.36	6.68	10.00
New U-1	£	Milit	0.12	0.41	0.45	D/fills	30.00	0.49	0.00	41.02	0.10	0.42	0.00	0.04	5.03	10,00	0.04	1.30	1,05	10.00
ther UF7	elfretton	MLD MLD	012	0.46 0.45	0.45	0.56 0.56	0.22	D41	0.88	40.22	0.90	GA2 GA2	0.00	G 84 0 84	582	10.00	D04	330	5.08 6.08	10.00
Workstor W	14.	Key'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
Monkator N		Kq\d MLIS	0.04	D.GF U.St	0.08	D10 D13	0.30	0.68	0.00	0.30	0.27	0.75	0.98	0.00	0.01	0.01	0.00	0.53	0.01	1.00
Year Well 1: Year Well 2		MI ID	0.00	0.11	U.31 G.11	819	0.00	D.50	0.97	0.20	0.29	075 075	0.97	0.00	0.50	1.00	0.00	0.53	067	1.00
Year Mire() 5		MI.D	0.03	0.1	40	0.13	9.20	1_09_	0.90	0.30	0.39	0.78	0.97	0.00	0.60	1.00	0.00	0.33	0.67	1.00
NAME 2		MLD MLD	0.25	1.01	102	1.28	-0.49	0.13	0.75	40.49	0.10	-0.07 -0.07	0.76 6.76	0.04	60:02 mmz	120.00	00A 500	40.08	99:01 20:01	120.00
Nosp-2 Nosp-3	High Lifting System	M.O	0.055	101	1.00	1.25	4149	0.13	0.75	41.45	-0.18	407	0.75	0.04	60.02	120.00	0.04	40.03	00.01	120 00
brosp f (Back up)	129	MLD	0.25	1/01	7.02	1:28	0.49	0.19	0.79	0.49	-016	0.07	0.76	0.04	60,02	120.00	0.04	40.08	50.01	120,00
vege 2 (Swolvag)		ME.D	0.8	101	7180	139	20,49		0.25	0548	0.10	97.0	8.76	0.94	50.02	120000	0.54	40:00	90.01	129,00
COLUMN TERROR TO THE SECOND	Š E	out the state of t	011	0.45	0.45	0.66	0.13	-0.91	0.89	0.13	8.82	0.48	0.09	0.00	191	300	-0.03	101	201	3.00
tale f	System	10.0			Janes				75000				1		S					
	100		Uni	0.46	0.45	0.56	0.12	050	0.9/	100	0.32	0.48	0.09	0.02	1.61	3.00	0.00	1.01	209	3.00
tale 2	S	MLD	18000		1130	0.00	13.7	100	0.50	2130	0.02	W	0.00	9.06			000	100		
PROP F	9 -	M.0	0.06	0.76	0年	100	-0.95	0.00	0.96	- 0.86	-0.49	0.00	0.95	0.00	60.04	120:00	0.08	4) (6	80.03	120,00
vop 2	2 2	60 D	0.05	0.70	0.76	1.00	-0.95	0,00	0.00	42.95	0.01	.010	0.95	0.00	10.01	120.00	0.06	40.00	190.03	120,00
http: 3 http://d	Bystem Bystem	MLD MLD	0.06	0.76	0.76	100	0.95	0.00	0.00	47.96	-0.40	000	0.95	0.08	60.04	120.00	0.00	40 00 40 06	80.03	120 00
SOCIATOR	E	WLD	0.00	8.76	0.75	100	4.00	0.00	100	180	-0.60	0.80	100	9.08	t engs	170.00	0.08	40.06	80.03	120,00
MI 0	B & F	MLD.	0.00	0.32	0.38	0.45	-0.48	0.26	1.00	42.40	0.11	0.30	1.00	0.000	054	1.00	1108	0.29	980	1.00
e0.2	Spatem System	MI.0	0.00	0.96	0.36	0.46	4148	0.26	1.00	440	-0.11	0.39	1.00	.0.08	9.64	1.00	0.00	0.39	0.69	1300
ed 2	and Storage	MLD.	0.00	0,96	0.96	0.48	-0.48	0.26	+06	40.48	011	0.00	100	0.06	051	1.00	0.00	0.39	0.69	1.00
e0.4		Mr th	0.00	0.06	0.36	D-49	0.49	0.76	1.08	-VE48	011	0.39	100	0.08-	0.64	1.00	5000	0.39	D69	1.00

# APPENDIX I: INPUT DATA I-B ELGIN AREA PRIMARY WATER SUPPLY SYSTEM (EAPWSS)

System		Elem	ent ID	l	Theorem.	Capacity			R	equirmer	nt	Normali	zed Capac	ity (TRI)	Normali	zedRequin	ment(Tri)	Normalized Capacity(Trap)			
System		53:019-01	Element	No.	Units	Average Only for	Design Capacity	Maximum Overload	Yearly Min	Yearly Avg	Felority: Max	Airerage Owly flat	Design Choecily	Maximum Operions	Yearn Ann	Yearly Aug	Yearly Max	Average Dark flow	Capacity	Capacity	Alexandra Cherthan
elektir Criti Loditum Pipocchionite IC Interior Pipoc	Intake	0 1 0 1 0 1	0 1 2 0 3	2 3	MED MED	52.6 52.6 52.4	91.0 91.0	190.0 180.0 180.0	363 363	524 524 524	71.8 71.8	0.39 0.29 0.29	0.56 0.50 0.50	1,00 1,00 1,00	0.19 0.19 0.19	0.39 0.29 0.29	0.40 0.40 0.41	0.39 0.29 0.39	0 40 0 40 0.40	0.75 0.75 0.75	1.00
Nort Drain Nambe Pipe Towelling Screens Investing Screens Investing World 2 North Oricharge Plump 1 Investigating 1 Investigating Plump 2 Injectionage Plump 2 Injectionage Plump 2	LowLifting System	0 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 1 0 2 0 3 0 4 0 5 0 6	5 6 7 5 5 10	PALD PALD PALD PALD PALD PALD PALD	26.2 26.2 26.2 26.2 26.2 29.3 8.2 19.8	45.5 45.5 45.5 45.5 34.5 10.9 14.6	90.5 90.5 90.5 90.5 90.6 97.0 21.0	353 353 363 355 134 42 334	52.4 52.4 52.4 52.1 95.5 52.1 98.9	71.6 71.8 71.8 71.6 71.5 86 97.3	0.20 0.29 0.29 0.20 0.20 0.30	050 050 050 050 050 052 052	1:00 1:00 1:00 1:00 1:00 1:00	0.39 0.39 0.39 0.59 0.59 0.30 0.20	0.50 0.50 0.50 0.50 0.50 0.30 0.30	0.79 0.79 0.79 0.79 0.41 0.41 0.41	0.29 0.29 0.29 0.29 0.30 0.30	0.40 0.40 0.40 0.40 0.41 0.41	0.75 0.75 0.75 0.76 0.76 0.76	1 00 1 00 1 00 1 00 1 00 1 00 1 00
ow Dückonge Planty 2 C Vision Pipe NC Stonge 1 NC Stonge 2		0 2 0 3	0 8	u u u	MLD MCD MC	52.4 0.0 0.0	91.0 91.0 95.5 95.5	21.0 161.0 171.0 171.0	4.2 35.3 0.0 0.0	62 4 01 01	86 71.8 02 03	0.30 0.39 0.00 0.00	050 050 050	100 100 100	021 035 000 000	0.00 0.00 0.00	0.41 (r.4) 800 000	0.39 0.39 0.00	0.40 0.26 0.25	0.76 0.75 0.75	1.00
NC Francisc Pump 3 NC Francisc Pump 2 NC Pum Tanis NC methodog Pump 3 NC methodog Pump 3 NC methodog Pump 3 NC methodog Pump 3 NC methodog Pump 4 Unio Stanisg 2		0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 4 5 6 7 6 6 7 6 6 7 6 7 6 7 6 7 6 7 6 7 6	16 17 18 19 20 21 22 22	15 15 15 15 15 15 15 15 15 15 15 15 15 1	00 00 00 00 00 00 00	32 32 15 36.6 36.6 77.5 77.5 77.5	63 50 635 635 635 635 635 635 635 635 635 635	00 00 00 00 00 00	0.0 0.0 0.0 0.0 0.0 0.0 0.0	00 00 03 01 01 01 01	0.00 0.00 0.00 0.00 0.00 0.00	050 850 050 050 050 050 050	1:00 1:00 1:00 1:00 1:00 1:00 1:00 1:00	0.00 0.00 0.00 0.00 0.00 0.00	0.00 0.00 0.04 0.00 0.00 0.00 0.00	000 000 000 000 000 000 000	0.00 0.00 0.00 0.00 0.00 0.00 0.00	0.5 0.5 0.5 0.5 0.5 0.5 0.5	0.75 0.75 0.75 0.75 0.75 0.75 0.75	100
User Strong P.  User Tengraf Purep  User Day Turk  User Strong Purep. 3  User sedenting Purep. 7  User sedenting Purep. 7  User Sedenting Purep. 3  Each Select Call S	Flash Mx System	0 3 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	7 2 3 4 7 9 7 9 7 9 7 9 7 9 7 9 7 9 7 9 7 9 7	26 25 26 27 28 29 29 21	EC EC EC EC EC EC EC EC EC EC EC EC EC E	0.0 0.0 0.0 0.0 0.0 0.0 0.0	75,0 0.5 60.0 90.0 30.0 30.0 30.0	180 18 120 120 120 120 120 120 120 120 120 120	0.0 0.0 0.0 0.7 0.7 0.7 0.7	1.7 0.0 0.0 14.7 14.7 14.7 14.7 26.2	25 0.1 0.1 0.26 0.26 1.26 1.26 1.26 1.26 1.26	0.00 0.00 0.00 0.00 0.00 0.00 0.00	050 050 050 050 050 050 050 050	1:00 1:00 1:00 1:00 1:00 1:00 1:00	000 000 000 000 000 000 000	0.08 0.00 0.02 0.29 0.28 0.58 0.58	013 000 008 1,00 1,03 2,06 2,06 2,06 0,07	0.00 0.00 0.00 0.00 0.00 0.00	20 20 20 20 20 20 20 20 20 20 20 20 20 2	0.75 0.75 0.75 0.75 0.75 0.75 0.75	1.00
Such Mit Cod / holywork Mit Such holywork Mit Yank holywork makering Pamp I holywork makering Pamp 2 holywork makering Pamp 3 holywork makering Pamp 3 holywork makering Pamp 4 holem Templer Pamp (black ng) home Mamering Pamp (black ng) home Mitmoling Pamp (black ng)		000000000000000000000000000000000000000	1212233567	32 33 36 35 36 37 38 39	MLD L L/min/Hear!	00 08 00 00 00 00 00	26.9 142.6 200.0 0.6 1.0 0.6 1.0 76.0 26.6	59.7 286.0 570.0 1.2 1.9 1.2 1.9 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0	17 6 0.0 0.0 0.0 0.0 0.0 0.0 0.0	36.2 38.9 38.0 0.2 0.3 0.2 0.3 0.0 0.0	98.8 98.8 98.0 0.9 0.5 0.5 0.5 0.5	0.00 0.00 0.00 0.00 0.00 0.00 0.00	050 050 050 050 050 050 050 050	1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00	0.00 0.00 0.00 0.00 0.00 0.00 0.00	0.49 0.14 0.07 0.14 0.14 0.14 0.16 0.00 0.00	0.67 0.39 0.16 0.20 0.39 0.30 0.30 0.30 0.00	0.00 0.00 0.00 0.00 0.00 0.00	025 025 025 025 025 025	0.75 0.75 0.75 0.75 0.75 0.75 0.75	100
ha Abrola Com. Bholasi 4041 4041	System	0 3 0 4	0 1 0 2	41 43	MLD MLD	00 00	45.5	80.0 91.0	nn B8	13.1 13.1	0.1 17.0 17.0	0.00 0.00	0 SD	1,00 1,00	9 10 9 10	0.14 0.14	0.30 0.30	0.00 0.00	0.25 0.26	0.75 0.75	1 M 1 M
e0.62 e0.63	Floct Syr	0 4	9 1	45	MLD MLD	0.0 0.0	45.5 45.5	91.0 91.0	6.0 5.5	111	17.5 17.0	0.00 0.00	050 050	1/00 1:00	0.00	0.14 0.14	0.20 0.20	000 000	0.25	075 076	1.00
redinantitation Touch II  euConstitution Touch II  to Calculation  booklo Stronge II  booklo Stronge II  booklo Part Part II  booklo France (book-op)  booklo France (book-op)  booklo France (book-op)  booklo Touch	Sedmentation System	0 5 5 6 5 6 5 6	0 3 3 0 4 0 5 0 6 0 7	47 48 49 50 51 52	MLD Ng/d m² m² kr² L/a	8.5 0.0 0.0 0.0 0.0 0.0	360 1250 45 45 78	90 8 250 B 9.0 9.0 9.0 9.0 14.5	0.0 0.0 0.0 0.0 0.0 0.0	96.2 96.2 98.2 98.0 98.0 90.0 25.25	79.0 79.0 0.0 0.0 5.0 5.0	0 10 0 20 0 00 0 00 0 20 0 20 0 00	055 050 050 050 050 050	1:00 1:00 1:00 1:00 1:00	0.00 0.00 0.00 0.00 0.00	0.56 0.58 0.15 0.00 0.00 0.00 0.07	0.79 0.79 0.32 0.00 0.00 0.34 0.34	0.30 0.30 0.30 0.30 0.30 0.30	035 035 035 035 035	0.76 0.75 0.75 0.75	180
Mer 3 ther 1 ther 4 her Well trade Separant	Fibration System	0 0 0 0 0 0	0 3 9 0 9 0 9	51 55 56 57 58	HE D HE D HE D HE D HE D	00 00 00 00 00	74 50 150 150 150 150 457	30.0 30.0 30.0 30.0 30.0 91.5	88 88 350	111 121 121 121 124	77 B 17 B 17 B 17 B 17 B 11 B	0.00 0.00 0.00 0.00 0.00	050 050 050 050 050 050	1.00 1.00 1.00 1.00 1.00 1.00	0.26 0.29 0.25 0.39 0.39	0.44 0.44 0.44 0.44 0.57 0.59	060 060 060 060 070 070	0.00 0.00 0.00 0.00 0.00	0.5 0.5 0.5 0.5 0.5	0.75 0.75 0.75 0.75 0.75 0.75	1 00
Son-Citientrasser Susp. 3 Nexp. 7 Susp. 6 Susp. 2	High Lifting System	0 7 0 7 0 7	0 3 0 4 0 4	93 93 93 92 93	HELD MELD MELD MELD	20.1 26.1 26.1 26.1	40.0 40.0 40.0	620 620 620 620	50 55 55 68 88	12.1 12.1 12.1 12.1	17.0 17.0 17.0 17.0	054 054 054 054	0.97 0.97 0.97 0.97	100 100 100 100 100	0.17 0.17 0.17 0.17	0.75 0.25 0.25 0.26	0.32 0.34 0.34 0.34	050 050 050 050	065 065 065 956	0.66 0.66 0.86 0.86	100
Ser.	Conveyance System	0 8	Q 1	51	MLD.	522	82:0	1910	353	324	71.8	0.29	0.45	1.00	0.19	0.29	0.40	0.39	a ar	0.73	1.00
anninal Reservoir 1  anninal Reservoir 2	System	0 9	g t	65	MED	D0 00	137	27.3	no no	uı uı	26.2 26.2	0.00	050 050	1.00 1.00	am am	D.45 D.48	0.96	0 00 0 00	0.25	0.75	1.00

280,075,03		Elei	nent ID				Normalized Requirment(Trap)				Element-State (Tri-MoS)			Element-State (Trap-MoS)				Elem	ent-Fallu	re (Trl)	EI	Element-Failure (Trap)			
System		System	Elevne	nt A	No.	Units	Average Dayly Sox	Capacity	Capacity	Maiamun Overload	rive Charles		4		8		d		ь	e.			must on	ø	
icale Criti Cocken Hypocalisada IC Insake Pipo Seet Darin Inthin / Sees	Printer	0 0	0000	2 2 2	2 3	MLD MLD MLD MLD	019 019 019	0.34 0.34 0.34	034 024 034 034	0.40 0.40 0.40 9.40	4 H 4 H 4 H 4 H	021 021 021 021	0.81 0.81 0.81 0.81	011 011 011 301	0.65 0.85 0.85	0.42 0.42 0.42 0.42	0.81 0.61 0.61 0.61	0.08 0.04 0.08 0.08	015 362 019 019	0.0 r.ob 0.0 0.0	0.18 0.14 0.19 0.90	0.11 2.30 0.11 9.10	D14 428 D14 D14	0.17 2.00 0.17 P.32	
ameling Science consider Science ampling Well 1 ampling Well 1 ampling Well 1 am Discharge Pamp 7 am Discharge Pamp 2 to Mischarge Pamp 2 to Mischarge Pamp 2 to Science Pamp 2	LowLifting System	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	00000000	123256788	5 6 7 8 9 40 11 12 13	MID MID MID MID MID MID MID MID MID	0.29 0.39 0.39 0.39 0.30 0.20 0.23 0.23 0.20	007 007 007 007 005 005 005	0.99 0.99 0.99 0.95 0.95 0.95 0.95	0.79 0.79 0.79 0.41 0.41 0.41 0.41 0.45	050 050 050 000 011 011 011	400 400 400 622 623 623 623 623 623 623	881 861 861 880 880 980 980 980	0.60 0.60 0.60 0.11 -0.11 -0.11	029 029 029 036 036 036	0.08 0.08 0.09 0.41 0.41 0.41 0.41	9.67 0.67 0.67 0.69 0.66 0.66 0.66 0.66 0.66	004 004 004 004 000 004 004	0.25 0.27 0.27 0.42 0.42 0.42 0.42	0.50 0.50 0.50 0.50 0.50 27.80 27.80 27.80 27.80	0.14 0.14 0.14 0.14 0.14 0.15 0.14 0.14	0.19 0.19 0.19 7.09 7.00 7.00 7.00	0.36 0.36 0.36 0.36 14.01 14.01 14.01 14.01	050 050 050 050 2100 2100 2100 2100	
66. Shanger T. 66. Shanger T. 66. Shanger Z. 66. Taesolar Plange 2. 66. Taesolar Plange 2. 66. Days Taesolar Plange 2. 66. Days Taesolar Plange 2. 66. Shanger Plange 3. 66. Shanger Planger 3. 66. Shanger 3. 66. Shanger 3. 66. Shanger	Finan Mx System		000000000000000000000000000000000000000	00170345670001734	有自场打电电池出现沿线沿流下海沿面从建设场的路沿面的最后	m' UA Ub	000 000 000 000 000 000 000 000 000 00	000 000 000 000 000 000 000 000 000 00	0.000 0.000	000 0000 0000 0000 0000 0000 0000 0000 0000	000 000 000 000 000 000 000 011 000 011 000 010 010 00	050 050 050 050 050 050 050 060 061 041 041 041 050 061 061 061 063 063 063 063 063 063 063 063 063 063	100 100 100 100 100 100 100 100 100 100	000 000 000 000 000 000 000 000 000 00	0.25 0.25 0.25 0.25 0.25 0.25 0.25 0.25	0.000 0.000	100 100 100 100 100 100 100 100 100 100	008 008 008 008 008 008 008 008 008 008	244 244 244 244 244 244 244 244 244 244	\$00 \$00 \$00 \$00 \$00 \$00 \$00 \$00 \$00 \$00	0 100 0 100	177 177 177 177 177 177 177 177 177 177	3 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2	5.00 5.00 5.00 5.00 5.00 5.00 5.00 3.00	
स्म । सम्म । सम्म ।	Systom	0 0 0	0 0	2 3	63 44	MED MED	019 019 019	0.17 0.17 0.17	0.17 0.17 0.17	0.30 0.20 0.20	420 420 420	0.96 0.96 0.96	090 090 090	4020 4020 4020 4020	0.90 0.90 0.90	0.56 0.58 0.50	096 096 096 046	0.04 0.04 0.04	352 352 352 350	700 700 700 700	014 014 014	236 236 236	455 458 458 458	7.00 2.00 7.00 7.00	
AC 63 coloriosterios Fank F coloriosterios Fank M a-Calintautar laccride Stonege 2 Laccride Stonege 3 Laccride Fank Fank (Soch-Left Laccride Fank (Soch-Left Laccride Fank Pank (Soch-Left Laccride Fank Pank (Soch-Left Laccride Fank (Soch-Left Laccride Fank (Soch-Left Laccride Fank (Laccride Laccride Laccride Fank (Laccride Laccride Laccride Fank (Laccride Laccride Laccride Fank (Laccride Laccride	Sedimentation F System	0 0 0 0 0 0	9 9 9	2 2	65 67 69 59 51 51	MLD MLD MLD Keps m' m' Lin Lin Lin	0.39 0.39 0.00 0.00 0.00 0.00	0.67 0.67 0.19 0.00 0.00 0.25 0.25	0 50 0 50 0 27 0 90 0 90 0 90	0.79 0.79 0.00 0.00 0.00	0.09 0.09 0.00 0.00 0.00 0.00 0.00 0.00	-4 02 -2 03 0.37 0.50 0.50 0.50 0.50	0.61 0.81 1.00 1.00 1.00	058 -069 -032 -000 -034 -034	036 036 035 035 036 030 030	0.10 0.10 0.50 0.75 0.75	0.61 0.61 1.00 1.00 1.00 1.00	0.04 0.04 0.04 0.04 0.04 0.04 0.04	3.52 3.52 3.52 3.52 3.52 3.52 3.52 3.52	7.00 7.00 7.00 7.00 7.00 7.00 7.00	0 12 0 14 0 14 0 14 0 14 0 14	2.8 2.8 2.8 2.8 2.8 2.8 2.8 2.8	468 468 468 468 468 468 468 468	700 700 300 700 700 700	
Wer I Wer 2 Wer 3 Wer 6 Saar Wed weden Kansanach	Filtration System	D 8	0 0 0 0	0.00	51 58 59 56 57 58	MLD MLD MLD ML ML	0.29 0.29 0.29 0.29 0.29	0.51 0.51 0.51 0.51 0.07 0.67	0.52 0.52 0.52 0.52 0.52 0.52 0.59	0.80 0.80 0.80 0.90 0.70 0.73	0.80 0.80 42.80 0.80 4.78 4.29	0.06 0.06 0.08 0.06 -8 07 -8 08	0.71 0.71 0.71 0.71 0.71 0.61	060 060 060 076 078	0.27 0.27 0.27 0.27 0.45 0.43	0.00 0.04 0.04 0.04 0.04 0.06 0.06	071 071 071 071 061 061	0.04 0.04 0.04 0.04 0.04 0.08	5.02 5.02 5.02 5.02 3.52 3.52	10.80 10.90 10.90 10.90 7.00 7.00	0.04 0.02 0.04 0.04 0.04 0.04	1.00 0.06 3.00 0.06 2.36 2.36	6.68 9.68 8.88 6.68 4.88 4.68	7 (00 10:00 10:00 10:00 10:00 7 (00 7 (00 7 (00	
ua Chlorivator 1993 I 1995 I 1995 I	High Lifting System	0 0 0 0 0	0 0 0	0546	85 80 80 80	MUD MLD MLD MLD	017 017 017 017 017	0.39 0.39 0.39 0.39 0.39	0.30 0.30 0.30	0.34 0.34 0.34 0.34 0.34	0.72 0.18 0.18 0.19 0.19	0.50 0.62 0.62 0.62 0.62	0.90 0.03 0.03 0.03 0.03	0.12 0.19 0.19 0.19 0.19	0.55 0.36 0.36 0.36	0.50 0.59 0.59 0.59 0.59	080 080 081 081 081	7.00 7.00 7.00 7.00 7.00 2.00	9,52 9,50 9,50 16,50 10,60	7.00 30.90 20.00 30.00 20.00	0.54 7 80 7 80 7 90 7 90 7 90	2 30 14 60 14 60 14 60	20:35 20:35 20:35 20:35 20:35	30 00 30 00 30 00 30 00 30 00	
ser	Comreyance System	0 8	0		u	MTD	0.19	034	0,04	0.40	611	0:16	091	(6)11	0.89	0.38	Der	0.08	0.42	6.75	0.10	0.81	0.51	0.78	
www.elfamorer /	System System	D 5	a		65	MID	0.00	0.72 0.72	0.72 0.72	0.96	4196	0.00	1.00	0% 0%	0.0°	0.09 0.09	1.01 1.01	0.04	3.62	7.00	0.04	2.96	4.60	7.00	

# APPENDIX II: MATLAB SOURCE CODE II-A LAKE HURON PRIMARY WATER SUPPLY SYSTEM (LHPWSS)

```
December 2, 2004
                                                                              4:41:32 PM
    % Last Modified: Thursday, October 07, 2004 @5:45pm
     x=-10:0.05:5;
 3
    time=0:0.1:200;
    alpha=0:0.05:1;
 5 %%%.....Step ONB......Building Structure Array
 6 $%%it is an array that containes fields under each fieldthere are values or
 7
    ***strings
 B
 9
         %determines the number of total elements in the system
10
         [No_of_Elements,DummyO1]=size(rowheaders);
11
1.2
         % Build the Structure array from the imported excel file.
13
         for n=1:No of Elements
1.4
                 System(n) = struct('Element_Name', rowheaders(n,1),'Element_ID',data(n,1 &
     :4) ....
15
                     'Element_global_No',data(n,5),'Element_Capacity',data(n,6:8),...
16
                     "Element Requirment', data(n,9:11), "Element Norm Cap Tri', data(n,12:1 &
     4), 'Element_Norm_Req_Tri', data(n, 15:17),...
17
                     "Blement Norm Cap Trap", data (n. 18:21), 'Element Norm Reg Trap', data (n.
     ,22:25),...
18
                     "Element State TriMos',data(n,26:28), "Element State TrapMos',data(n, 🗸
     29:32)....
19
                     "Element Failure Tri',data(n,33:35), "Element Failure Trap',data(n,36 &
     139)1/
20
         and
21
22
     %%% ..........Step TWO.....Deterimine the No. of Bub-Systems
     $8% by caclualting the maximum No. Value in the ID first two digits && Deterimine
24
     *** the No. of elements in each Sub-System by caclualting the maximum No. Value in
25
     *** the ID last two digits
26
27
             for sse=1:No of Elements
                                                 * sse stands for Sub-System Element
28
                  % Calculation of the Number of Sub-Systems
29
                 SubSystem (sse, 1) = [ [System (sse) .Element ID(1, 1) System (sse) .Element ID(1, 	€
     2)]];
30
                 SubSystm Tenth(sse, 1) = rium2str(SubSystem{sse, 1}(1,1));
31
                 SubSystm Units(sse, 1) = num2str(SubSystem{sse, 1](1,2));
32
                 SubSystem_String_Code=streat(SubSystm_Tenth,SubSystm_Units);
33
                 SubSystem Numerical Code=str2num(SubSystem String Code);
```

Total SubSystems-max(SubSystem Numerical Code);

\* Calculation of the Number of Elements in each Sub-System

34

```
37
              Element(sse,1)={(System(sse).Element_ID(1,3) System(sse).Element_ID(1,4) &
    1);
38
              SubSystm_Element_Tenth(sse, 1) = num2str(Element(sse, 1)(I,1));
39
              SubSystm Element Units(sse, 1) = num2str(Element(sse, 1)(1,2));
40
              Element_String_Code=strcat(SubSyetm_Element_Tenth,SubSyetm_Element_Units &
    11
41
              Element_Numerical_Code=str2num(Element_String_Code);
42
            end
43
44
    System_Numeric_ID_Array=[SubSystem_Numerical_Code_Element_Numerical_Code];
45
46
    * Creating zero matrix to store information on number of elements in each subsystem
47
    Information Array=reros (Total SubSystems, 2);
48
49
    counter-1;
50
        for sseZ=1: (No_of_Elements-1)
51
           Information Array(counter, 1) -counter;
52
           Information Array(counter, 2) = System Numeric ID Array(sse2, 2);
53
54
               if (System Numeric ID Array((sse2+1),2))<(System Numeric ID Array(sse2,2 e
    33
55
                  counter-counter+1;
56
               end
57
58
              if counter -- Total SubSystems
59
                 Information Array (counter, 2) - System Numeric ID Array (sse2+1, 2) :
60
              and
51
        end
52
63
    **************************
64
    *** ..........Step THREE.....Make Puzzy MF usable in Augmentation *
    ₹₹₹ Using the Triangula MP in the system Matrix
65
66
           for i=1:No of Elements
              A-System(i) .Element State TriMos(1,1);
57
              B-System(i) .Element State TriMos(1,2);
68
69
              CaSystem(i) .Element State TriMos(1,3);
70
              Element MF(1,1)={[triangular(slpha,A,B,C)]};
71
75 *** Using the Trapizoidal MF in the system Matrix
```

December 2, 2004 4:41:32 PM 76 %for i=1:No\_of\_Elements 77 A=System(i). Element State TrapMos(1,1); 78 B-System(1) Element\_State\_TrapMos(1,2); 79 C=System(i) .Blement State TrapMos(1,3); 80 % D=System(i) Blement\_State\_TrapMos(1,4); 81 # Blement MP(1,1)=[[trapizoidal(alpha,A,B,C,D)]]; 22 kend 83 84 86 \*\*\* using the SUMMATION operator for both 87 \*\* NOTE: all redundant elements or parallel elelemtns will be added to the 89 %% value of the first element) 90 91 92 %1. Intake Sub-System 93 ♦ there is no redundancy in any of the elements 94 95 \$2. Low Lifting Sub-System 96 & J Redundant elements 97 04----07. 05----08 & 06----09 98 Element MF{7,1}=fuzzymath(alpha, Element MF{7,1}, Element MF{10,1}, 'sum'); 99 Element MF(8,1)=fuzzymath(alpha,Element MF(8,1),Element MF(11,1),'sum'); 100 Element MF(9,1)-fuzzymath(alpha,Element MF(9,1),Element MF(12,1),'sum'); 101 102 %3. Palch Mixing Sub-System 103 \*PAC, Alum, and Polymer Sub system 104 ₹ 1 Redundant element 105 03----21 106 Element\_MF{15,1}=fuzzymath(alpha,Element\_MF{15,1},Element\_MF{33,1},'sum'); 107 108 %4. Plocculation Sub-System ₹ there is no redundancy in any of the elements 109 110 111 \$5: Sedimentation Sub-System

%6. Piltering Sub-System

# there is no redundancy in any of the elements

\* there is no redundancy in any of the elements

112

113 114

```
152
       Element MF{34,1}=fuzzymath(alpha, Blement MF{34,1}, Blement MF{36,1}, 'sum');
153
      Element MF{34,1}=fuzzymath(alpha,Element MF{34,1},Element MF{38,1},'sum');
154
       Element_MP{34,1}-fuzzymath(alpha,Element_MP{34,1},Element_MP{40,1},'sum');
155
      Element MF{35,1}=fuzzymath(alpha,Element MF{35,1},Element MF{37,1},'sum');
156
      Element_MF{35,1}=fuzzymath(alpha,Element_MF{35,1},Element_MF{39,1},'sum');
157
      Element MP{35,1}=fuzzymath(alpha,Element MP{35,1},Element MP{41,1}, 'sum');
158
159
160
                            %5. Sedimentation Sub-System
161
                               $ 2 Paralel groups of elements
162
                                8 (01,03) - (02,04)
163
      Element MF{42,1}-fuzzymath(alpha,Element MF{42,1},Element MF{44,1},'sum');
164
165
      Element_MP{43,1}=fuzzymath(alpha,Element_MP{43,1},Element_MP{45,1},'sum');
166
167
                            ₹6. Filtering Sub-System
158
                                   ₹ 3 Paralel groups of elements
159
                                   ♦ (01,03,05,07,09,11) & (02,04,06,08,10,12) &
170
                                   8 (15, 16, 17)
171
      Element MF{46,1}=fuzzymath(alpha,Element NF{46,1},Element MF{48,1}, 'sum');
172
      Element MP{46,1}-fuzzymath(alpha,Element MP{46,1},Element MP{50,1},*sum');
173
      Element MF{46,1}-fuzzymath(alpha, Element MF{46,1}, Element MF{52,1}, "sum");
174
      Element MF{46,1}-fuzzymath(alpha, Element MF{46,1}, Element MF{54,1}, 'sum');
175
      Element MF(46,1)-fuzzymath(alpha,Element MF(46,1),Element MF(56,1),'sum')/
176
       Element MF{47,1}-fuzzymath(alpha, Element MF{47,1}, Element MF{49,1}, "sum")/
177
       Element MP{47,1}-fuzzymath(alpha, Element MP{47,1}, Element MP{51,1}, 'sum');
178
      Element MF{47,1}-fuzzymath(alpha, Element MF{47,1}, Element MF{53,1}, *sum');
179
       Element MP{47,1}-fuzzymath(alpha,Element MP{47,1},Element MP{55,1}, 'sum');
180
      Element MF{47,1}-fuzzymath(alpha,Element MF{47,1},Element MF{57,1}, "sum");
181
      Element MF(60,1)=fuzzymath(alpha,Element MF(60,1),Element MF(61,1),*sum');
182
      Element_MF{60,1}-fuzzymath(alpha,Element_MF{60,1},Element_MF{62,1},'sum');
183
184
                            %7. High Lifting Sub-System
185
                                   $ 1 Paralel group
186
                                   8 (01,02,03)
187
      Element MF(63,1)=fuzzymath(alpha, Blement MF(63,1), Blement MF(64,1), 'sum');
      Element MP{63,1}-fuzzymath(alpha,Element MP{63,1},Element MP{65,1},'sum');
189
190
                            %8. Conveyance Sub-System
191
                                   ₹ I ParaleI group
192
                                   $ (01,02)
193
      Element_MF{68,1}=fuzzymath(alpha,Element_MF{68,1},Element_MF{69,1},'sum');
```

```
194
195

    Boosting Sub-System

196
                                 % I Paralel group
197
                                * (01,02,03,04)
198
      Element MF{70,1}=fuzzymath(alpha,Element MF{70,1},Element MF{71,1},'sum');
199
      Element_MP{70,1}=fuzzymath(alpha,Element_MF{70,1},Element_MF{72,1},'sum');
200
      Element_MF{70,1}=fuzzymath(alpha,Element_MF{70,1},Element_NF{73,1},'sum');
201
202
                          %10. Storage Sub-System
203
                                * I Paralel group
204
                                & (01,02,03,04)
205
       Element MF{75,1}=fuzzymath(alpha,Element MF{75,1},Element MF{76,1},'sum');
206
       Element_MF{75,1}=fuzzymath(alpha,Element_MF{75,1},Element_MF{77,1}, 'sum');
207
       Element MF{75,1}=fuzzymath(alpha,Element MF{75,1},Element MF{78,1},'sum');
208
     \star
209
     33333333333333333333333333 Serial Elementstassassassassassassassassassassassas
      2222
210
211
     # Build the sequence of Augmentation Sub-System step wise
212
     ₹ Connection between sub_system is considered serial connection
213
214
                          %1. Intake Sub-System
215
                             ₹ 1 serial 2 serial 3
216
      System State-fuzzylogmath(x,alpha,Element MF{1,1},Element MF{2,1},'minimum');
217
      System State-fuzzylogmath(x,alpha,System State,Element MF(3,1),'minimum');
218
219
                         $2. Low Lifting Sub-System
220
                             * 1 serial 2 serial 3 serial (4* parallel 5* parrallel 6*)
221
                             % star element means it has a idundant element.
222
     System State-fuzzylogmath(x,alpha,System State,Element MF(4,1),'minimum');
223
      System State=fuzzylogmath(x,alpha,System State,Element MP(5,1),'minimum');
224
      System State-fuzzylogmath(x,alpha,System State,Element MF(6,1),'minimum');
      System State-fuzzylogmath(x,alpha,System_State,Element_MP{7,1},'minimum');
226
227
                         23. Falsh Mixing Sub-System
228
                             %PAC, Alum, and Polymer Sub_system
229
                             % (1 serial 3* serial 5 serial 7 serial 9 serial 11
230
                                  serial 13 serial 15 serial 17 serial 19)
231
                              & PARALLEL TO
232
                              $ (2 serial 4 serial 6 serial 8 serial 10 serial
```

December 2, 2004 4:41:32 PM 233 12 serial 14 serial 16 serial 18 serial 20) 234 \* this summation to avoid producing MPs that can not be summed using the 235 \* summation function (as minmising will produce MFs in different froms 236 & than that needed by the summation function) 237 Blement MF{13,1}=fuzzymath(alpha,Element MF{13,1},Element MF{14,1},'sum'); 238 Element MP{15,1}=fuzzymath(alpha,Element MP{15,1},Element MP{16,1}, 'sum'); 239 Element MF{17,1}=fuzzymath(alpha,Element MF{17,1},Element MF{18,1}, 'sum'); 240 Element\_MF{19,1}=fuzzymath(alpha,Element\_MF{19,1},Element\_MF{20,1},'sum'); 241 Element MF{21,1}=fuzzymath(alpha,Element MF{21,1},Element MF{22,1}, "sun"); 242 Blement\_MF{23,1}=fuzzymath(alpha,Element\_MF{23,1},Element\_MF{24,1}, "sum"); 242 Element\_MF{25,1}-fuzzymath(alpha,Element\_MF{25,1},Element\_MF{26,1}, \*sum\*); Element MF{27,1}=fuzzymsth(alpha,Element MF{27,1},Element MF{28,1}, 'sum'); 244 245 Element MF(29.1)=fuzzymath(alpha, Element MF(29.1), Element MF(30.1), 'sum'); 246 Element\_MF{31,1}=fuzzymath(alpha,Element\_MF{31,1},Element\_MF{32,1},'sum'); 247 Element MF(13,1}=fuzzylogmath(x,alpha,Element MF(13,1],Element MF(15,1],'minimum'); 248 Blement\_MF{13,1}=fuzzylogmath(x,alpha,Element\_MF{13,1],Element\_MF{17,1},'minimum'); 249 Element MF{13,1}-fuzzylogmath(x,alpha,Element MF{13,1},Element MF{19,1},'minimum'); 250 Element MF(13,1}-fuzzylogmath(x,alpha,Element MF(13,1],Element MF(21,1],'minimum'); 251 Blement MF{13,1}=fussylogmath(x,alpha,Element MF{13,1},Element MF{23,1},'minimum'); 252 Element MF{13,1}=fuzzylogmath(x,alpha,Element MF{13,1},Element MF{25,1},'minimum'); 253 Element MF{13,1}=fuzzylogmath(x,alpha,Element MF{13,1},Element MF{27,1},'minimum'); 254 Rlement\_MF{13,1}=fuzzylogmath(x,alpha,Element\_MF{13,1},Element\_MF{29,1},'minimum'); 255 Element MF(13,1)=fuzzylogmath(x,alpha,Element MF(13,1),Element MF(31,1), 'minimum'); 256 257 System State-fuzzylogmath(x,alpha,System State,Element MF{13,1},'minimum'); 258 259 \$4. Plocculation Sub-System 260 % no serials, only previous system-state and 261 % sum of tow parrallel groups of elements 262 Element MP{34,1}=fuzzymath(alpha,Element MP{34,1},Element MF{35,1}, 'sum'); 263 System\_State=fuzzylogmath(x,alpha,System\_State,Element\_MF{34,1},'minimum'); 264 265 %5. Sedimentation Sub-System 266 \$ no serials, only previous system-state and 267 % sum of tow parrallel groups of elements 268 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'); 270 271 %6. Filtering Sub-System

% (01,03,05,07,09,12) serial 13

% (02,04,06,08,10,12) serial 14

& PARALLEL TO

272

273

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);
347
348
        Weighted Area-Calculation
349
        weighted strip Area System State=0;
350
        for ttt=1: (step_no_System_State-1)
351
           weighted strip Area System State-weighted strip Area System State+((0.5*((Sy ✔
     stem_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));
352
        end
353
        Weighted_Area_System_State=weighted_strip_Area_System_State+((0.5*(System_State_w
     R((step no System State+1),1)-System State L((step no System State+1),1))*step Syste &
     m State) *System State L(step no System State, 2));
354
        355
356
    ********************************
357
    358 *** using trapisoidal MF as (a,b,c,d) where c-d
359
    No of Levels=3;
                                                             7
360 m1-0.6; m2-0.7; m3-5; m4-5;
                                                             3
361 Level 1y=trapmf(x,[m1 m2 m3 m4]);
362 Level 1=(x' Level 1y');
363 LR Level I=(m1*m2)/(m2-m1);
364
365 n1=0.6; n2=1.2; n3=5; n4=5;
366 Level 2y-trapmf(x,[n1 n2 n3 n4]);
367 Level 2-[x' Level 2y'];
368 LR Level 2=(n1*n2)/(n2-n1);
369
370 q1=0.6; q2=5; q3=5; q4=5;
371 Level 3y=trapmf(x,[q1 q2 q3 q4]);
372 Level 3=[x' Level 3y'];
373 LR Level 3=(q1*q2)/(q2-q1);
374
375 IR-[LR Level 1; LR Level 2; LR Level 3];
376
377
380 *** of the state function and each level
381 OverLap Level 1y=min(Level 1(:,2),System State xwise(:,2));
382 OverLap_Level_i=[x' OverLap_Level_iy];
```

```
383
384
      OverLap Level 2y=min(Level 2(:,2),System State xwise(:,2));
385
      OverLap_Level_2-[x' OverLap_Level_2y];
386
387
     OverLap_Level_3y=min(Level_3(:,2),System_State_xwise(:,2));
388 OverLap Level 3-[x' OverLap Level 3y];
389
390
         * take-off the zeros from both ends of the overlap area
391
392 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] #
394 Area_Level_1_end=[max(OverLap_Level_1((find(OverLap_Level_1(:,2)>0)),1))+x_step 0];
395 Area Level 1= [Area Level 1 start; Area Level 1; Area Level 1 end];
396
397
     Area_Level_Z=OverLap_Level_Z((find(OverLap_Level_Z(:,Z)>0)),:);
398
     Area Level 2 start=[min(GverLap Level 2((find(GverLap Level 2(:,2)x0)),1)]-x step 0] v
399
      Area Level 2 end-[max(OverLap Level 2((find(OverLap Level 2(:,2)>0)),1))+x step 0];
400
      Area Level 2-[Area Level 2 start; Area Level 2; Area Level 2 end];
401
402
403
     Area Level 3-OverLap Level 3((find(OverLap Level 3(:,2)>0)),:);
404
      Area Level 3 start=[min(OverLap Level 3((find(OverLap Level 3(:,2)>0)),1))-x step 0] #
405
      Area Level 3 end=(max(OverLap Level 3((find(OverLap Level 3(:,2)>0)),1))+x step 0];
406
      Area Level 3-[Area Level 3 start; Area Level 3; Area Level 3 end];
407
408
      409
      *** ........... Step SEVEN..... Calculation of the overlap area & weighted area by inte &
      prolation over alpha cuts
410 %%% for each overlap matrix
411
412
         % Determine the maximum alpha value in teh ovelap matrix.
414
         Maximum_alpha_Area_Level_1=max(Area_Level_1(:,2));
415
416
         % based on the No. of rows, a loop will search for the row index of the
417
         * maximum alpha value
418
         [row Area Level 1,column Area Level 1]=size(Area Level 1);
419
         max index Area Level 1=0;
420
         for ii=1:row_Area_Level_1
```

```
421
               max_index_Area_Level_1=max_index_Area_Level_1+1;
422
               if Area Level 1(ii,2)==Maximum alpha Area Level 1
423
                   bresk:
424
               end
425
          end
426
427
          * the range from the maximum value to Zero will be divided for high
428
          * resolution alpha.
429
          alpha range Area Level 1-Maximum alpha Area Level 1;
430
          step_no_Area_Level_1=100;
431
          step_Area_Level_1-Maximum_alpha_Area_Level_1/step_no_Area_Level_1/
432
433
          %Interpolation
434
              %Left-Hand Interpolation
435
                  Left Area Level 1-(interpl(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) ≰
       1/1
436
                  Area Level 1 L=[Left Area Level 1 (0:step Area Level 1:Maximum alpha Are 4
      a Level 1)');
437
               %Right-Hand Interpolation
438
                  Right Area Level 1=(interpl(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 Leve ⊀
      1 1:Maximum alpha Area Level 1) *);
439
                  Area Level 1 R-[Right Area Level 1 (0:step Area Level 1:Maximum alpha Ar 🛩
      ea Level 1)'|/
440
441
          %Area-Calculation
442
          strip Area Area Level 1-0;
443
           for jj=1: (step no Area Level 1-1)
444
               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);
445
          end
446
          Area Area Level 1-strip Area Area Level 1+(0.5*(Area Level 1 R((step no Area Lev ✔
      el 1+1),1)-Area Level 1 L((step no Area Level 1+1),1))*step Area Level 1);
447
448
          Weighted Area-Calculation
449
          weighted_strip_Area_Area_Level_1=0;
450
          for kk=1: (step no Area Level 1-1)
451
              weighted str:p Area Area Level 1-weighted str:p Area Area Level 1+((0.5*((Ar €
      ea Level 1 R(kk,1)-Area Level 1 L(kk,1))+(Area Level 1 R((kk+1),1)-Area Level 1 L((k w
      k+1),1)) *step_Area_Level_1) *((Area_Level_1_L(kk,2)+Area_Level_1_L((kk+1),2)) *0.5));
```

```
452
          end
453
          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));
          454
455
456
          457
          * Determine the maximum alpha value in teh ovelap matrix.
458
          Maximum alpha Area Level 2-max(Area Level 2(1,2));
459
460
          * based on the No. of rows, a loop will search for the row index of the
461
         * maximum alpha value
462
         [row Area Level 2,column Area Level 2]=size(Area Level 2);
463
          max_index_Area_Level_2=0;
464
          for iii-1:row Area Level 2
465
             max_index_Area_Level_2=max_index_Area_Level_2+1;
455
              if Area Level 2(111,2) -- Maximum alpha Area Level 2
467
                  break;
              end
468
469
          end
470
471
          * the range from the maximum value to Zero will be divided for high
472
          * resolution alpha.
473
          alpha range Area Level 2-Maximum alpha Area Level 2;
474
          step no Area Level 2=100;
475
          step Area Level 2-Maximum alpha Area Level 2/step no Area Level 2;
476
477
          %Interpolation
478
              *Left-Hand Interpolation
479
                 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) &
480
                 Area Level 2 L=[Left Area Level 2 (0:step Area Level 2:Maximum alpha Are &
      a Level 2) '];
481
              %Right-Hand Interpolation
482
                 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 ≰
      1_2:Maximum_alpha_Area_Level_2) ');
483
                 Area_Level_2_R=[Right_Area_Level_2 (0:step_Area_Level_2:Maximum_alpha_Ar &
      ea Level 2) 1/
484
485
          %Area-Calculation
```

```
486
          strip_Area_Area_Level_2=0;
487
          for jjj=1:(step_no_Area_Level_2-1)
488
              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);
489
490
          Area_Area_Level_2=strip_Area_Area_Level_2+(0.5*(Area_Level_2_R((step_no_Area_Lev &
      el_2+1),1)-Area_Level_2_L((step_no_Area_Level_2+1),1))*step_Area_Level_2);
491
492
           *Weighted Area-Calculation
493
          weighted_strip_Area_Area_Level_2-0;
494
          for kkk-1: (step no Area Level 2-1)
495
              weighted strip Area Area Level 2=weighted strip Area Area Level 2+(10.5*({Ar €
      ea_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))* #
      0.571:
496
          end
497
          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));
498
          $$$$$$$$$$$....,Level TWO...$$$$$$$$$$$$$$$$$$$$$$$$$$$
499
500
          $855232555....Level THREE...$8233555235553533355533355533
501
           * Determine the maximum alpha value in teh ovelap matrix.
502
          Maximum alpha Area Level 3-max(Area Level 3(:,2));
503
504
          * based on the No. of rows, a loop will search for the row index of the
505
          ₹ maximum alpha value
506
          [row Area Level 3,column Area Level 3]-size(Area Level 3);
507
          max index Area Level 3=0;
508
          for iiii-1:row Area Level 3
509
              max index Area Level 3-max index Area Level 3+1;
510
              if Area Level 3(iiii,2) == Maximum alpha Area Level 3
511
                   break:
512
              end
513
          end
514
515
          * the range from the maximum value to Zero will be divided for high
516
          % resolution alpha.
517
          alpha range Area Level 3-Maximum alpha Area Level 3;
518
          step no Area Level 3=100;
519
          step_Area_Level_3=Maximum_alpha_Area_Level_3/step_nc_Area_Level_3;
```

\*\*\* ...... Step EIGHT .... Reliability measure Calculation

```
548
   CM Level 1=Weighted Area Area Level 1/Weighted Area System State;
549
   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];
552
553 Reliability Index=(max(CN)*LR(Max Level, 1))/max(LR)
   554
555 *** ..... Step MINE.... Roubustness measure Calculation
556 Robustness Index 1 2-1/(CM Level 1-CM Level 2)
557 Robustness Index 1 3=1/(CM_Level_1-CM_Level_3)
558 Robustness Index 2 3-1/(CM Level 2-CM Level 3)
562 ****************************
563 *******************
564 **********
565 ******
566
   332
567
   23
568 ₹
569 ₹₹
570 ***
   ******
571
572 ***********
574 **************************
577 *** ...... Step TEN.... Reslience measure Calculation
579 *** ......Step TEN-ONE....Make Puzzy MP usable in augmentation *
580 *** Using the Triangula MP in the system Matrix
        for i=1:No of Elements
581 *
582 8
         a-System(i).Element Pailure Tri(1,1);
583 8
         b=System(i).Element Failure Tri(1,2);
          c=System(i) .Element Pailure Tri(1,3);
505
    8
          Blement_Pail(i,1)={(triangular(alpha,a,b,c))};
586
     & and
588
   ***********************
```

C:\Documents and Settings\Ibrahim\Desktop\Daily w...\rmLH.m Page 17
December 2, 2004 4:41:32 PM

```
590
     *** Using the Trapizoidal MP in the system Matrix
                                                            *
591
             for i=1:No of Elements
                                                            8
592
                a-System(i).Element_Failure_Trap(1,1);
593
               b=System(i).Element Failure Trap(1,2);
594
               c=System(i).Element_Failure_Trap(1,3);
595
               d-System (1) .Element Failure Trap (1,4):
                                                            3
596
                Element_Fail(i,1)={[trapizoidal(alpha,a,b,c,d)]};
597
            end
600 *** using the MAXIMUM operator for both
601 $% NOTE: all redundant elements or parallel elelemtns will be added to the
603 ** value of the first element)
604
605
     606
                      %1. Intake Sub-System
607
                            * there is no redundancy in any of the elements
608
609
                      $2. Low Lifting Sub-System
610
                            & J Redundant elements
611
                                04----07. 05----09 & 06----09
612
     Element_Fail{7,1}-fuzzylogmath(time,alpha,Element_Fail{7,1},Element_Fail{10,1},*maxi &
     miam')/
613
     Element Fail(8,1)-fuzzylogmath(time,alpha,Element Fail(8,1),Element Fail(11,1), 'maxi &
614
     Element Fail (9,1)-fuzzylogmath (time,alpha,Element Fail (9,1),Element Fail (12,1), 'maxi &
     mam');
615
616
                      $3. Palsh Mixing Sub-System
617
                         *PAC, Alum, and Polymer Sub system
618
                            % I Redundant element
619
                               03----21
520 Element Fail{15,1}-fuzzylogmath(time,alpha,Element Fail{15,1},Element Fail{33,1},'ma &
     mum *);
621
622
                      $4. Plocculation Sub-System
623
                            * there is no redundancy in any of the elements
624
625

    SedImentation Sub-System

626
                            * there is no redundancy in any of the elements
```

662 % elements 663 664 \$4. Plocculation Sub-System 665 § 2 Paralel groups of elements 666 \$ (01,03,05,07) \$ (02,04,06,08) 667 Element Fail{34,1}-fuzzylogmath(time,alpha,Element Fail{34,1},Element Fail{36,1}, 'ma & ximum'); 668 Element\_Fail{34,1}=fuzzylogmath(time,alpha,Element\_Fail{34,1},Element\_Fail{38,1}, 'ma & ximum \* 5 669 Element\_Fail{34,1}=fuzzylogmath(time,alpha,Element\_Fail{34,1},Element\_Fail{40,1},'ma & mimum'); 670 Element Fail [35,1]=fuzzylogmath(time,alpha,Element Fail [35,1],Element Fail [37,1], 'ma & ximum\*); 671 Element\_Fail(35,1)=fuzzylogmath(time,alpha,Element\_Fail(35,1),Element\_Fail(39,1), 'ma & Element\_Fail{35,1}=fuzzylogmath(time,alpha,Element\_Fail{35,1},Blement\_Fail{41,1},'ma < 672 minum\*); 673 674 \$5. Sedimentation Sub-System 675 ₹ 2 Paralel groups of elements 676 8 (01,03) - (02,04) 677 Element\_Fail{42,1}-fuzzylogmath(time,alpha,Element\_Fail{42,1},Element\_Fail{44,1}, 'ma \* ximum\*); 678 Element Fail{43,1}=fuzsylogmath(time,alpha,Element Fail{43,1},Element Fail{45,1}, 'ma & mimum\*); 679 680 681 %6. Piltering Sub-System 682 % 3 Paralel groups of elements 683 \* (01,03,05,07,09,11) & (02,04,06,08,10,12) & 684 \* (15, 16, 17) 685 Element\_Fail{46,1}=fuszylogmath(time,alpha,Element\_Fail{46,1},Element\_Fail{48,1},'ma & ximum'): 686 Element Fail{46,1}=fuzzylogmath(time,alpha,Element Fail{46,1},Element Fail{50,1},'ma 

✓ 687 Element\_Fail{46,1}=fuzzylogmath{time,alpha,Element\_Fail{46,1},Element\_Fail{52,1},'ma & 688 Element\_Fail[46,1]=fuzzylogmath(time,alpha,Element\_Fail{46,1},Element\_Fail{54,1}, 'ma ' 689 Element Fail[46,1]-fussylogmath(time,alpha,Element\_Fail[46,1],Element\_Fail[56,1],'ma & 690 Element\_Fail{49,1}=fuzzylogmath(time,alpha,Element\_Fail{49,1},Element\_Fail{51,1},'ma w

```
C:\Documents and Settings\Ibrahim\Desktop\Daily w...\rmLH.m
                                                                                  Page 20
 December 2, 2004
                                                                              4:41:32 PM
       ximum*);
691
      Element_Fail{49,1}=fuzzylogmath(time,alpha,Element_Fail{49,1},Blement_Fail{53,1},'ma 🗸
692
      Element Fail{49,1}=fuzzylogmath(time,alpha,Element Fail{49,1},Element Fail{55,1},'ma &
      ximum');
693
      Element Fail{49,1}-fuzzylogmath(time,alpha,Element Fail{49,1},Element Fail{57,1}, 'ma &
      ximum');
694
      Element_Fail{60,1}=fuzzylogmath(time,alpha,Element_Fail{60,1},Blement_Fail{61,1},'ma &
      ximum * 5
      Element_Fail[60,1]=fuzzylogmath(time,alpha,Element_Fail[60,1],Element_Fail[62,1],'ma &
695
      mimum');
696
697
                            $7. High Lifting Sub-System
698
                                  & I Paralel group
699
                                   8 (01,02,03)
700
      Element_Fail [63,1]=fuzzylogmath(time,alpha,Element_Fail [63,1],Element_Fail [64,1], 'ma &
701
      Element Fail (63,1)-fuzzylogmath (time, alpha, Element Fail (63,1), Element Fail (65,1), 'ma &
      cimum*);
702
703
                            $8. Conveyance Sub-System
704
                                  & I Paralel group
705
                                   * (01,02)
706
      Element Fail [68,1]=fuzsylogmath(time,alpha,Element Fail [68,1],Element Fail [69,1],'ma w
      mimum*);
707
708
709
                            $9. Boosting Sub-System
710
                                   % 1 Paralel group
711
                                   $ (01,02,03,04)
712
      Element_Fail{70,1}-fuzzylogmath(time,alpha,Element_Fail{70,1},Element_Fail{71,1}, 'ma &
      ximum*);
713
      Element_Fail{70,1}=fuzzylogmath(time,alpha,Element_Fail{70,1},Element_Fail{72,1},'ma &
      ximum*);
      Element Fail{70,1}=fuzzylogmath(time,alpha,Element_Fail{70,1},Element_Fail{73,1},'ma w
      ximum*);
715
716
                            %10. Storage Sub-System
717
                                  $ 1 Paralel group
718
                                  % (01,02,03,04)
719
      Element Fail (75,1)=fuzzylogmath (time, alpha, Element Fail (75,1), Element Fail (76,1), 'ma .
      ximum*);
```

```
C:\Documents and Settings\Ibrahim\Desktop\Daily w...\rmLH.m
                                                                      Page 21
 December 2, 2004
                                                                   4:41:32 PM
720
     Element_Fail{75,1}=fuzzylogmath(time,alpha,Element_Fail{75,1},Blement_Fail{77,1},'ma &
     Element_Fail{75,1}-fuzzylogmath(time,alpha,Element_Fail{75,1},Element_Fail{73,1},'ma &
721
     .
722
      ************
************
724
     [Row time, Col time] = size(time);
725
726 * Rendering all Element-Pailure MPs a full length MP on time domain.
727 for e1-1:No of Elements
728
         Element Fail{e1,1}=fuzzylogmath(time,alpha,Element Fail{e1,1},Element Fail{e1,1} &
     , 'maximum');
729
730
731 % Then....getting the start and the end of the support of each Element-Pailure MP
732 % 1. Start of Support
733
     for 82-1:No of Elements
734
         for f=1: (Col time-1)
735
            if (Element Fail{s2,1}(f,2)==0) & (Element Fail{s2,1}((f+1),2)>0)
736
                System(s2).Element_Failure_Support(1,1)=Element_Fail[s2,1](f,1);
737
            end
738
         end
739
     end
740
     # 2. End of Support
741
     for s3=1:No of Elements
742
         for f=1:(Col time-1)
743
            if (Element Fail{s3,1}(f,2)>0) & (Element Fail{s3,1}((f+1),2)==0)
744
                System(s3).Element_Failure_Support(1,2)=Element_Fail{s3,1}((f+1),1);
745
            end
746
         end
747 end
748 % 3. Calculation of support length of all MPs
749 for s4=1:No of Elements
750
         System(s4).Element_Failure_Support_Length(1,1)=(System(s4).Element_Failure_Suppo &
     rt(1,2)-System(s4).Element Failure Support(1,1));
751 end
752
```

753 - ↑ Then....getting the start and the end of the modal values of each Element-Pailure ⊀

754 % 1. Start of modal

```
755
       for m2=1:No of Elements
756
           for ff=1: (Col time-1)
               if (Element_Fail{m2,1}(ff,2)<max(Element_Fail{m2,1}(:,2))) S (Element_Fail{m 	€
757
       2,1 ((ff+1),2) == max(Element_Fail(m2,1)(;,2)))
758
                   System(m2) .Element_Failure_Modal(1,1)=Element_Fail{m2,1}(ff,1);
759
               end
760
           end
761
      end
762
      $ 2. End of Support
763
      for m3=1:No_of_Elements
784
           for ff=1: (Col time-1)
               if (Element Fail{m3,1}(ff,2)=-max(Element Fail{m3,1}(c,2))) & (Element Fail{ &
765
      m3,1}((ff+1),2)<max(Element_Fail[m3,1](:,2)))
766
                   System(m3) .Element_Failure_Modal(1,2)=Element_Fail{m3,1}((ff+1),1);
767
               end
768
           end
769
770

३ 3. Calculation of support length of all MFs.

771
       for m4-1:No of Elements
772
           System(m4).Element Failure Modal Length(1,1) = (System(m4).Element Failure Modal(1 &
       ,2)-System(m4). Element Failure Modal(1,1));
773
      end
774
775
776
       # Build the sequence of Augmentation Sub-System step wise
777
       * Connection between sub system is considered serial connection
778
779
      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);...
780
                               System(2).Element global_No(1,1) System(2).Element Failure 8 €
      upport_Length(1,1) System(2).Element_Failure_Modal_Length(1,1);...
781
                               System (3) .Element global No(1,1) System (3) .Element Failure S 🗸
      upport_Length(1,1) System(3).Element_Failure_Modal_Length(1,1);...
782
                               System (4) .Element global No (1,1) System (4) .Element Failure S &
      upport Length (1,1) System (4) . Element Failure Modal Length (1,1) . . .
783
                               System (5) .Element global No (1,1) System (5) .Element Failure S 🗸
      upport Length (1,1) System (5) . Element Failure Modal Length (1,1) . . .
784
                               System(6).Element_global_No(1,1) System(6).Element_Failure_S 🗸
      upport_Length(1,1) System(6) Element_Failure_Modal_Length(1,1);...
785
                               System(7).Element global No(1,1) System(7).Element Failure S ≰
      upport Length (1,1) System (7) Element Failure Modal Length (1,1);...
786
                               System (13) .Element_global_No(1,1) System (13) .Element_Failure *
```

```
Support_Length(1,1) System(13) .Element_Failure_Modal_Length(1,1);...
787
                                System (34) .Element global No (1,1) System (34) .Element Failure 🗸
       _Support_Length(1,1) System(34) .Element_Failure_Modal_Length(1,1) ....
788
                                System (42) .Element global No(1,1) System (42) .Element Failure &
       _Support_Length(1,1) System(42).Element_Failure_Modal_Length(1,1);...
789
                               System (46) .Element global No(1,1) System (46) .Element Pailure ♥
       _Support_Length(1,1) System(46) .Element_Failure_Modal_Length(1,1);...
790
                               System (58) .Element_global_No(1,1) System (58) .Element_Failure &
       _Support_Length(1,1) System(58).Element_Failure_Modal_Length(1,1);...
791
                                System (60) . Element global No (1,1) System (60) . Element Failure 🗸
       _Support_Length(1,1) System(60) Element_Failure_Modal_Length(1,1);...
792
                               System (63) . Element global No (1,1) System (63) . Element Failure 🗸
       _Support_Length(1,1) System(63).Element_Failure_Modal_Length(1,1);...
793
                               System (68) .Element_global_No(1,1) System (68) .Element_Failure v
       _Support_Length(1,1) System(68) .Element_Failure Modal Length(1,1);...
794
                               System (70) .Element_global_No(1,1) System (70) .Element_Failure ≰
       Support Length (1,1) System (70) . Element Failure Modal Length (1,1) / ...
795
                               System (75) .Element global No(1,1) System (75) .Element Failure &
       _Support_Length(1,1) System(75).Element_Failure_Modal_Length(1,1)],
796
797
       * Determine the controling MF, (first MF with max support, then MF with max
798
       a modal)
799
800
       [Row Fail, Col Fail] - size (System Fail Information) /
801
802
       max support-max(System Fail Information(:,2));
903
       max support elements-find(System Fail Information(:,2) -- max support);
804
       [Row max support elements, Col max support elements] = size (max support elements);
805
806
       if (Row max support elements>1)
807
           max modal-max(System Fail Information(:,3));
808
           max modal elements-find(System Pail Information(:,3) == max modal);
809
           [Row max modal elements, Col max modal elements] = size (max modal elements);
810
           if (Row max modal elements>1)
911
               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)];
912
813
           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)];
814
       end
```

```
C:\Documents and Settings\Ibrahim\Desktop\Daily w...\rmLH.m Page 24
December 2, 2004 4:41:32 PM
```

```
815
816
     System_Fail=[System_Fail_Information(max_support_elements(1,1),1) System_Fail_Inform &
     ,311;
817
819
     *** .....Step TEN-POUR....use the MP of the controling Element
82.0
     System_Fail_MF=Element_Fail{System_Fail(1,1),1};
821
822
    [row_t,col_t]=size(System_Fail_MF);
833
824 Resilience_Index=0;
825 nominator=0:
826 denominator=0;
827 for t=1:(row t-1)
828
        nominator=nominator+(0.5*(System_Fail_MF(t+1,1)+System_Fail_MF(t,1)));
829
        denominator=denominator+(System_Fail_MF(t,2)*(0.5*(System_Fail_MF(t+1,1)+System_ e
     Fail MF(t,1))));
830
    end
     Resilience_Index-nominator/denominator
831
83.2
```

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

December 2, 2004 4:54:54 PM

```
% Last Modified: Thursday, October 07, 2004 @5:45pm
     x=-1:0.01:4:
 3
    time=0:0.1:200;
    alpha=0:0.05:1;
 5 ***.....Step ONE......Building Structure Array
 6 $%%it is an array that containes fields under each fieldthere are values or
 7
     ***strings
 B
 9
         *determines the number of total elements in the system
10
         [No_of_Elements,DummyO1]=size(rowheaders);
11
1.2
         % Build the Structure array from the imported excel file.
13
         for n=1:No of Elements
1.4
                 System(n) = struct('Element_Name', rowheaders(n,1), 'Element_ID', data(n,1 &
     :4) ....
15
                      'Element_global_No',data(n,5),'Element_Capacity',data(n,6:8),...
16
                     "Element Requirment', data(n, 9:11), "Element Norm Cap Tri', data(n, 12:1 &
     4), 'Element Norm Req Tri', data(n, 15:17) ....
17
                      "Blement Norm Cap Trap", data (n. 18:21), 'Element Norm Reg Trap', data (n.
     ,22:25),...
18
                      'Element State TriMos', data(n, 26:28), 'Element State TrapMos', data(n, 🗸
     29:32)....
19
                      "Element Failure Tri", data(n,33:35), "Element Failure Trap", data(n,36 &
     139)1/
20
         and
21
22
     %%% ..........Step TWO.....Deterimine the No. of Sub-Systems
      $8% by caclualting the maximum No. Value in the ID first two digits 44 Deterimine
24
     *** the No. of elements in each Sub-System by caclualting the maximum No. Value in
25
     *** the ID last two digits
26
27
             for sse=1:No of Elements
                                                  * sse stands for Sub-System Element
28
                  % Calculation of the Number of Sub-Systems
29
                 SubSystem (sse, 1) = [ [System (sse) .Element ID(1, 1) System (sse) .Element ID(1, 	€
     2) 1 17
30
                 SubSystm Tenth(sse, 1) = num2str(SubSystem(sse, 1)(1,1));
31
                 SubSystm Units(sse, 1) = num2str(SubSystem(sse, 1)(1,2));
32
                 SubSystem_String_Code=strcat(SubSyetm_Tenth,SubSyetm_Units);
33
                 SubSystem Numerical Code=str2num(SubSystem String Code);
34
                 Total SubSystems-max(SubSystem Numerical Code);
35
36
                 * Calculation of the Number of Elements in each Sub-System
```

```
December 2, 2004
                                                                              4:54:54 PM
37
                 Element(sse,1)={(System(sse).Element_ID(1,3) System(sse).Element_ID(1,4) &
     1);
38
                 SubSystm_Element_Tenth(sse, 1) = num2str(Element(sse, 1)(I,1));
39
                 SubSystm Element Units(sse, 1) = num2str(Element(sse, 1)(1,2));
40
                 Element_String_Code=strcat(SubSyetm_Element_Tenth,SubSyetm_Element_Units &
     11
41
                 Element_Numerical_Code=str2num(Element_String_Code);
42
              end
43
44
     System_Numeric_ID_Array=[SubSystem_Numerical_Code_Element_Numerical_Code];
45
46
     * Creating zero matrix to store information on number of elements in each subsystem
47
     Information Array=reros (Total SubSystems, 2);
48
49
     counter-1;
```

C:\Documents and Settings\Ibrahim\Desktop\Daily w...\rmAE.m

```
50
      for sseZ=1: (No_of_Elements-1)
51
          Information Array(counter, 1) -counter;
52
          Information Array(counter, 2) = System Numeric ID Array(sse2, 2);
53
54
             if (System Numeric ID Array((sse2+1),2))<(System Numeric ID Array(sse2,2 e
   33
55
               counter-counter+1;
56
             end
57
58
            if counter -- Total SubSystems
59
               Information Array (counter, 2) - System Numeric ID Array (sse2+1, 2) :
60
            and
51
       end
52
63
    64
   *** ...........Step THREE.....Make Puzzy MP usable in Augmentation *
   ₹₹₹ Using the Triangula MP in the system Matrix
65
66
          for i=1:No of Blements
                                                     3
             a-System(i).Element State TriMos(1,1);
67
                                                     5
             b-System(i).Element State TriMos(1,2);
69
     180
             c=System(i).Element State TriMos(1,3);
                                                      8
      *
             Element MF(1,1)={(triangular(alpha,a,b,c))};
                                                     8
71
      8
75 *** Using the Trapizoidal MF in the system Matrix
```

```
117
118
                        *7. High Lifting Sub-System
119
                               * there is no redundancy in any of the slements
120
121
                        $8. Conveyance Sub-System
122
                               * there is no redundancy in any of the elements
123
124

    Storage Sub-System

125
                               % there is no redundancy in any of the elements
126
     127
128
     129
                        %1. Intake Sub-System
130
                              * there is no parallel elements
131
132
                        $2. Low Lifting Sub-System
133
                               & 4 Paralel elements
134
                               8 09-10-11-12
135
      Element MF(9,1)=fuzzymath(alpha,Element MF(9,1),Element MF(10,1),'sum');
136
      Element MF(9,1)-fuzzymath(alpha, Element MF(9,1), Element MF(11,1), 'sum');
137
      Element MF(9,1)=fuzzymath(alpha,Element MF(9,1),Element MF(12,1),'sum');
138
139
                        %3. Palsh Mixing Sub-System
140
                            *PAC, Alum, and Polymer Sub system
141
                              * 14-15
142
                              $ 16-17
143
                              $ 19-20-21-22
144
                              § 23-24
145
                              $ 27-28-29-30
146
                              8 31-32
147
                              ¥ 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, Blement MF{19,1}, Blement MF{20,1}, 'sum');
151
     Element MF{19,1}=fuzzymath(alpha, Element MF{19,1}, Element MF{21,1}, 'sum');
152
      Element MF(19,1)=fuzzymath(alpha,Element MF(19,1),Element MF(22,1),'sum');
153 Element_MF{23,1}=fuzzymath(alpha,Element_MF{23,1},Element_MF{24,1},'sum');
```

% 2 Paralel elements

195

```
196
                                $ 65-66
197
       Element MF{65,1}=fuzzymath(alpha,Element MF{65,1},Element MF{66,1}, *sum*);
198
      199
      2552
200
201
     # Build the sequence of Augmentation Sub-System step wise
      Connection between sub system is considered serial connection
202
203
204
                         #1: Intake Sub-System
205
                            $ 2 serial 2 serial 3#
206
                            % star element means it has a rdundant element
207
     System_State=fuzzylogmath(x,alpha,Element_MF{1,1},Element_MF{2,1},.minimum*);
208
     System_State=fuzzylogmath(x,alpha,System_State,Element MF(3,1),'minimum');
209
210
                        $2. Low Lifting Sub-System
                            ₹ 5 serial 7 serial (9#) serial 13
211
212
      System State-fussylogmath(x,alpha,System State,Element MF(5,1),'minimum'),
213
     System State-fuzzylogmath(x,alpha,System State,Element MF(7,1),'minimum');
214
     System State-fuzzylogmath(x,alpha,System State,Element MF(9,1),'minimum');
215
     System_State-fuzzylogmath(x,alpha,System_State,Element_MF(13,1),'minimum');
216
217
                        83. Palsh Mixing Sub-System
218
                             *PAC, Alum, and Polymer Sub system
219
                             * (14#) serial (16#) serial 18 serial (19#) serial (23) ser 🗷
     ial
220

§ 25* serial 26 serial (27#) serial (31#) serial 33 serial 
≰
      34 serial (35#)
221
     System State=fuzzylogmath(x,alpha,System State,Element MF(14,1), 'minimum');
222
     System State-fuzzylogmath(x,alpha,System State,Element MF{16,1},'minimum');
223
     System State-fuzzylogmath(x,alpha,System State,Element MF(i8,1),'minimum');
224
     System State=fuzzylogmath(x,alpha,System State,Element MP{19,1},'minimum');
225
     System State-fuzzylogmath(x,alpha,System State,Element MF(23,1), 'minimum');
     System State-fuzzylogmath(x,alpha,System State,Element MF(25,1),'minimum');
227
     System State=fuzzylogmath(x,alpha,System State,Element MF{26,1},'minimum');
     System State-fuzzylogmath(x,alpha,System State,Element MF(27,1),'minimum');
229
     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');
232
     System State=fuzzyloqmath(x,alpha,System State,Element MF(35,1), 'minimum');
233
```

December 2, 2004 4:54:54 PM

```
234
                         #4. Plocculation Sub-System
235
                               % no serials, only previous system-state
236
     System_State-fuzzylogmath(x,alpha,System_State,Element_MF{42,1},'minimum');
237
23.8
                         %5. Sedimentation Sub-System
239
                               % (46#) serial 49 serial (50#) serial 51
240
     System_State=fuzzylogmath(x,alpha,System_State,Element_MF{46,1},'minimum');
241
     System_State=fuzzylogmath(x,alpha,System_State,Element_MF{48,1},'minimum');
     System State-fuzzylogmath(x,alpha,System State,Element MF(50,1),'minimum');
243
     System_State=fuzzylogmath(x,alpha,System_State,Element_MF{51,1},'minimum');
244
245
                         $6. Piltering Sub-System
246
                            % (52#) serial 56 serial 57
247
     System_State=fuzzylogmath(x,alpha,System_State,Element_MF(52.1),'minimum');
248
     System State-fuzzylogmath(x,alpha,System State,Element MF(56,1), 'minimum');
     System_State=fuzzylogmath(x,alpha,System_State,Element_MF(57,1),'minimum');
250
251
252
                         $7. High Lifting Sub-System
253
                            $ 59 serial (60#)
254
     System State-fuzzylogmath(x,alpha,System State,Element MF(59,1),'minimum');
255
     System_State-fuzzylogmath(x,alpha,System_State,Element_MF(60,1),'minimum');
256
257
                         %8. Conveyance Sub-System
258
                            * no serials, only previous system-state
259
     System State-fuzzylogmath(x,alpha,System State,Element MF{64,1},'minimum');
260
261
                         %9. Storage Sub-System
                            % no serials, only previous system-state
262
263
     System State=fuzzylogmath(x,alpha,System State,Element MF(65,1),'minimum');
264
265
      [Row x,Col x]=size(x); * basic information about x-values (x-axis)
266 No_x_steps=Col_x-1;
267 x step=max(x)/No x steps;
269 * keeping the original System State MF with zeros for farther use in
270 % calcuation of the overlap area
271 System_State_xwise=System_State;
273
274 System State=[System State(find(System State(7,2)>0),1) System State(find(System State
     te(:,2)>0),2)1:
```

```
275
       [dumm_row,dumm_col]=size(System_State);
276
      System State=[(System State(1,1)-x step) 0;System State;(System State(dumm row,1)+x 🗸
      step) 0];
277
278
      $$$$$$$$$$$$,....System State Areas...$$$$$$$$$$$$$$$$$$$$$$$$$$$
279
          * Determine the maximum alpha value in teh system-state matrix.
280
          Maximum_alpha_System_State=max(System_State(:,2));
281
282
          % based on the No. of rows, a loop will search for the row index of the
283
          * maximum alpha value
284
          [row_System_State,column_System_State]=size(System_State);
285
          max index System State=0;
286
          for t=1:row System State
287
              max index System State-max index System State+1;
288
              if System State(t,2) -- Maximum alpha System State
289
                   break:
290
              end
291
          end
293
293
          * the range from the maximum value to Zero will be divided for high
294
          * resolution alpha.
          alpha_range_System_State=Maximum_alpha_System_State;
295
296
          step no System State=100;
297
          step System State-Maximum alpha System State/step no System State;
298
299
          *Interpolation
300
              %Left-Hand Interpolation
301
                  Left System State=(interpl(System State(1:max index System State,2),Syst &
      em State(1:max index System State,1),0:step System State:Maximum alpha System State) 🗸
       .51
302
                  System_State_L=[Left_System_State (0:step_System_State:Maximum_alpha_Sys &
      tem State) '];
303
               &Right-Hand Interpolation
304
                  Right System State=(interpl(System State(max index System State:row Syst &
      em State,2),System State(max index System State:row System State,1),O:step System St &
      ate: Maximum alpha System State) ');
305
                  System State R=[Right System State (0:step System State:Maximum alpha Sy €
      stem_State)'];
306
307
          MArea-Calculation
308
          strip Area System State=0;
309
          for tt=1: (step_no_System_State-1)
```

```
********************************
344
     *** of the state function and each level
347 OverLap Level 1y=min(Level 1(:,2), System State xwise(:,2));
348 OverLap_Level_1=[x' OverLap_Level_1y];
349
350 OverLap_Level_2y=min(Level_2(;,2),System_State_xwise(;,2));
351 OverLap_Level_2=[x' OverLap_Level_2y];
352
353 OverLap_Level_3y=min(Level_3(:,2),System_State_xwise(:,2));
354
    OverLap_Level_3=[x' OverLap_Level_3y];
355
356
        % take-off the seros from both ends of the overlap area
357
358 Area Level 1=OverLap Level 1((find(OverLap Level 1(:,2)>0)),:);
     Area_Level_1_start=[min(OverLap_Level_1((find(OverLap_Level_1(x,2) > 0)),1)) -x_step 0] &
360
     Area Level 1 end-[max(OverLap Level 1((find(OverLap Level 1(:,2)>0)),1))+x step 0|r
361
     Area Level 1-[Area Level 1 start; Area Level 1; Area Level 1 end];
362
363 Area Level 2=OverLap Level 2((find(OverLap Level 2(:,2)>0)),:);
364 Area_Level_2_start=[min(OverLap_Level_2((find(OverLap_Level_2(:,2)>0)),1))-x_step 0] ✔
365
    Area Level 2 end-[max(OverLap Level 2((find(OverLap Level 2(:,2)>0)),1))+x step 0];
366
     Area Level 2-[Area Level 2 start; Area Level 2; Area Level 2 end];
367
369
369
     Area Level 3=OverLap Level 3((find(OverLap Level 3(:,2)>0)),:);
370
    Area Level 3 start=[min(OverLap Level 3((find(OverLap Level 3(:,2)>0)),1))-x step 0] v
371 Area_Level_3_end=[max(OverLap_Level_3((find(OverLap_Level_3(:,2)>0)),1))+x_step 0];
372 Area_Level_3=[Area_Level_3_start;Area_Level_3;Area_Level_3_end];
373
prolation over alpha cuts
376 $%% for each overlap matrix
377
378
        379
        % Determine the maximum alpha value in teh ovelap matrix.
380
        Maximum alpha Area Level 1=max(Area Level 1(:,2)):
381
```

```
C:\Documents and Settings\Ibrahim\Desktop\Daily w...\rmAE.m Page 11
December 2, 2004 4:54:54 PM
```

```
382
           * based on the No. of rows, a loop will search for the row index of the
383
           * maximum alpha value
384
          [row_Area_Level_1,column_Area_Level_1]-size(Area_Level_1);
385
          max index Area Level 1=0;
386
          for ii=1:row Area Level 1
387
              max index Area Level 1-max index Area Level 1+1;
388
              if Area_Level_1(ii,2)==Maximum_alpha_Area_Level_1
389
                   break:
390
              end
391
          end
392
          % the range from the maximum value to Zero will be divided for high
393
394
          % resolution alpha.
395
          alpha_range_Area_Level_1=Maximum_alpha_Area_Level_1;
396
          step no Area Level 1-100;
397
          step_Area_Level_1=Maximum_alpha_Area_Level_1/step_no_Area_Level_1;
398
399
          &Interpolation
400
              *Left-Hand Interpolation
401
                  Left Area Level 1-(interpl(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) &
      1/1
402
                  Area Level 1 L-[Left Area Level 1 (0:step Area Level 1:Maximum alpha Are 🗸
      a Level 1)'];
403
              ₹Right-Hand Interpolation
404
                  Right Area Level 1-(interp) (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 Leve
      1 1:Maximum alpha Area Level 1) *);
405
                  Area Level 1 R-[Right Area Level 1 (0:step Area Level 1:Maximum alpha Ar 🖝
      ea Level 1)'1;
406
407
          %Area-Calculation
408
          strip Area Area Level 1=0;
409
          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)- w
410
      Area_Level_1_L(jj,1))+(Area_Level_1_R((jj+1),1)-Area_Level_1_L((jj+1),1))*step_Area &
      Level 1);
411
412
          Area_Area_Level_1=strip_Area_Area_Level_1+(0.5*(Area_Level_1_R((step_no_Area_Lev w
      el 1+1),1)-Area Level 1 L((step no Area Level 1+1),1))*step Area Level 1);
413
414
          Weighted Area-Calculation
```

```
December 2, 2004
                                                                             4:54:54 PM
415
          weighted strip Area Area Level 1=0;
416
           for kk=1: (step no Area Level 1-1)
417
              weighted_strip_Area_Area_Level_1-weighted_strip_Area_Area_Level_1+((0.5*((Ar 🗸
      ea Level 1 R(kk,1)-Area Level 1 L(kk,1))+(Area Level 1 R((kk+1),1)-Area Level 1 L((k .
      k+1),1)))*step_Area_Level_1)*((Area_Level_1_L(kk,2)+Area_Level_1_L((kk+1),2))*0.5));
418
419
          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));
420
           $$$$$$$$$$.....Level ONE...$$$$$$$$$$$$$$$$$$$$$$$$$$$
421
          $$$$$$$$$$$....Level TWO...$$$$$$$$$$$$$$$$$$$$$$$$
422
423
          * Determine the maximum alpha value in teh ovelap matrix.
424
          Maximum_alpha_Area_Level_2=max(Area_Level_2(:,2));
425
426
          * based on the No. of rows, a loop will search for the row index of the
427
           * maximum alpha value
428
          [row Area Level 2,column Area Level 2]=size(Area Level 2);
439
          max index Area Level 240;
430
          for iii-1:row Area Level 2
431
              max index Area Level 2-max index Area Level 2+1;
432
              if Area Level 2(iii,2) -- Maximum alpha Area Level 2
433
                   break:
434
              end
435
           and
436
437
          * the range from the maximum value to Zero will be divided for high
438
           * resolution alpha.
439
          alpha range Area Level 2-Maximum alpha Area Level 2;
440
          step no Area Level 2=100;
441
          step Area Level 2-Maximum alpha Area Level 2/step no Area Level 2;
442
443
          *Interpolation
444
              ₹Left-Hand Interpolation
445
                  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) &
       118
446
                  Area_Level_2_L=[Left_Area_Level_2 (0:step_Area_Level_2:Maximum_alpha_Are &
      a Level 2)'];
447
              %Right-Hand Interpolation
448
                  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 &
```

C:\Documents and Settings\Ibrahim\Desktop\Daily w...\rmAE.m

```
1 2:Maximum alpha Area Level 2) ');
449
                  Area Level 2 R=[Right Area Level 2 (0:step Area Level 2:Maximum alpha Ar 🗸
      ea_Level_2) '] /
450
451
          *Area-Calculation
452
          strip Area Area Level 2-0;
453
          for jjj=1:(step_no_Area_Level_2-1)
454
              strip Area Area Level 2=strip Area Area Level 2+(0.5*((Area Level 2 R(j)j,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);
455
          end
456
          Area Area Level 2-strip Area Area Level 2+(0.5*(Area Level 2 R((step no Area Lev €
      el 2+1),1)-Area Level 2 L((step no Area Level 2+1),1))*step Area Level 2);
457
458
          aWeighted Area-Calculation
459
          weighted_strip_Area_Area_Level_2=0;
460
          for kkk=1: (step no Area Level 2-1)
461
              weighted strip Area Area Level 2-weighted strip Area Area Level 2+((0.5*((Ar 	€
       ea 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));
462
463
          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));
464
           $$$$$$$$$$$....Level TWO...$$$$$$$$$$$$$$$$$$$$$$$$$
465
466
           anasatanan....Level THREE... attanhasatanhananananananan
467
          & Determine the maximum alpha value in teh ovelap matrix.
468
          Maximum alpha Area Level 3=max(Area Level 3(;,2));
469
470
          % based on the No. of rows, a loop will search for the row index of the
471
          * maximum alpha value
472
          [row Area Level 3,column Area Level 3]=size(Area Level 3);
473
          max index Area Level 3-0;
474
          for iiii=1:row Area Level 3
475
              max index Area Level 3-max index Area Level 3+1;
476
              if Area_Level_3(iiii,2) == Maximum_alpha_Area_Level_3
477
                   break;
478
              end
479
          end
480
```

C:\Documents and Settings\Ibrahim\Desktop\Daily w ...\rmAE.m

```
509
510 Area=[Area Area Level 1; Area Area Level 2; Area Area Level 3];
511 Max_Level-find(Area--max(Area));
514 OM Level 1-Weighted Area Area Level 1/Weighted Area System State;
515 CM Level 2=Weighted Area Area Level 2/Weighted Area System State;
516 CM_Level_3=Weighted_Area_Area_Level_3/Weighted_Area_System_State;
517 CM-[CM Level 1;CM Level 2;CM Level 3];
518
519 Reliability Index-(max(CM)*LR(Max Level,1))/max(LR)
522 Robustness_Index_1_2=1/(CM_Level_1-CM_Level_2)
523 Robustness Index 1 3=1/(CM Level 1-CM Level 3)
524 Robustness_Index_2_3=1/(CM_Level_2-CM_Level_3)
527
   *************************
528
   **************************
529 *******************
530 ***********
531 ******
532
   223
533 38
534 *
535 **
536 ###
537 ******
538 **********
539 *******************
540 **************************
544
547 *** Using the Triangula MP in the system Matrix
548 *
       for i=1:No of Elements
                                     3
549
         aa=System(i).Element Pailure Tri(1,1);
550
   8
        bb=System(i).Element Pailure Tri(1,2);
```

Element Fail{27,1}~fuzzylogmath(time,alpha,Element\_Fail{27,1},Element\_Fail{29,1},'ma v

Element\_Fail{27,1}-fuzzylogmath(time,alpha,Element\_Fail{27,1},Element\_Fail{29,1},'ma &

Element Fail{27,1}-fuzsylogmath(time,alpha,Element Fail{27,1},Element Fail{30,1},'ma w

Element Fail{31,1}-fuzsylogmath(time,alpha,Element Fail{31,1},Element Fail{32,1}, 'ma w

Element Fail(35,1)-fuzzylogmath(time,alpha,Element Fail(35,1),Element Fail(36,1), wa &

638

639

640

641

ximum\*);

mimum\*);

mimum \* );

wimmer's

C:\Documents and Settings\Ibrahim\Desktop\Daily w...\rmAE.m

	XIIIdili ()
643	Element_Fail{35,1}=fuzzylogmath(time,alpha,Element_Fail{35,1},Element_Fail{37,1},'ma 🗸
	ximum*);
644	Element_Fai1{35,1}=fuszylogmath(time,alpha,Element_Fai1{35,1},Element_Fai1{38,1},'ma &
	ximum*);
645	
646	<ol> <li>Plocculation Sub-System</li> </ol>
647	<pre>% 4 Paralel elements</pre>
648	3 42-43-44-45
649	$Element\_Fail\{42,1\} = fuzzy logmath(time,alpha,Element\_Fail\{42,1\},Element\_Fail\{43,1\}, 'ma \   \textbf{\textit{w}} = fuzzy logmath(time,alpha,Element\_Fail\{43,1\},Element\_Fail\{43,1\}, 'ma \   \textbf{\textit{w}} = fuzzy logmath(time,alpha,Element\_Fail\{43,1\},Element\_Fail\{43,1\}, 'ma \   \textbf{\textit{w}} = fuzzy logmath(time,alpha,Element\_Fail\{43,1\},Element\_F$
	ximum*);
650	Element_Fail(42,1}-fuzzylogmath(time,alpha,Element_Fail(42,1),Element_Fail(44,1),'ma x ximum');
651	${\tt Element\_Fail\{42,1\}=fuzzylogmath(time,alpha,Element\_Fail\{42,1\},Element\_Fail\{45,1\},`ma*}$

ximum\*); 652 653 ≱5. Sedimentation Sub-System 654 % 2 Paralel groups of elements 655 \$ 46-47 656 \$ 49-50 657 Element\_Fail{46,1}=fuzzylogmath(time,alpha,Element\_Fail{46,1},Element\_Fail{47,1},'ma & 658 Element\_Fail{49,1}=fuzzylogmath(time,alpha,Element\_Fail{49,1},Element\_Fail{50,1},'ma & ximum\*); 659 660 %6. Piltering Sub-System 661 % 4 Paralel elements 662 ¥ 52-53-54-55 663 Element Fail [52,1]-fuzzylogmath(time,alpha,Element Fail [52,1],Element Fail [52,1], 'ma w ximum'); 664 Element Fail [52,1]-fuzsylogmath (time, alpha, Element Fail [52,1], Element Fail [54,1], 'ma v 665 Element Fail [52,1]-fuzzylogmath (time, alpha, Element Fail [52,1], Element Fail [55,1], 'ma & ximum\*52 666 667 \$7. High Lifting Sub-System 668 \* 4 Paralel elements 669 8 60-61-62-63 670 Element Fail [60,1]-fuzzylogmath (time,alpha,Element Fail [60,1], Element Fail [61,1], 'ma & mimum \* ); 671 Element Fail{60,1}-fuzzylogmath(time,alpha,Element Fail{60,1},Element Fail{62,1},'ma & 672 Element Fail [60,1]-fuzzylogmath (time, alpha, Element Fail [60,1], Element Fail [63,1], 'ma v ximum'); 673 674 \$8. Conveyance Sub-System 675 \* there is no parallel elements 676 677 %9. Storage Sub-System 678 \$ 65-66 679 Element Fail [65,1]-fuzzylogmath(time,alpha,Element Fail [65,1],Element Fail [66,1], 'ma & 680  $\bullet \\$ \*\*\*\*\*\*\*\*\*\*\*\* \*\*\*\*\*\*\*\*\*\*\*

```
682
683
       [Row_time, Col_time] = size(time);
684
685 * Rendering all Slement-Pailure MPs a full length MF on time domain.
686 for e1=1:No of Elements
687
          Element Fail(e1,1)-fuzzylogmath(time,alpha,Element Fail(e1,1),Element Fail(e1,1) #
      , 'maximum');
688
      end
689
690 % Then....getting the start and the end of the support of each Element-Failure MF
691 # 1. Start of Support
692 for s2-1:No of Elements
693
          for f=1: (Col time-1)
694
              if (Element_Fail{s2,1}(f,2)==0) & (Element_Fail{s2,1}((f+1),2)>0)
695
                  System(s2).Element Failure Support(1,1)-Element Fail(s2,1)(f,1);
696
              end
697
          end
698
      end
699
      * 2. End of Support
700
      for 83-1:No of Elements
701
          for f=1:(Col time-1)
702
              if (Element_Fail{s3,1}(f,2)>0) & (Element_Fail{s3,1}((f+1),2)==0)
703
                  System(s3).Element_Failure_Support(1,2)=Element_Fail(s3,1)((f+1),1);
704
              end
705
          and
706
707
      # 3. Calculation of support length of all MPs
708
      for 84-1:No of Elements
709
          System(s4).Element Failure Support Length(1,1)=(System(s4).Element Failure Suppor
      rt(1,2)-System(s4). Element Failure Support(1,1));
710
711
712
      * Then.....getting the start and the end of the modal values of each Element-Failure ♥
713 % 1. Start of modal
714
     for m2=1:No of Elements
715
          for ff=1:(Col time-1)
716
              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)))
717
                  System(m2) .Element Failure Modal(1,1)=Element Fail(m2,1)(ff,1);
718
              end
719
          end
```

```
720
      end
721
      * 2. End of Support
722
      for m3-1:No_of_Elements
723
           for ff=1: (Col time-1)
724
               if (Element_Fail{m3,1}(ff,2)==max(Element_Fail{m3,1}(:,2))) & (Element_Fail{ *
      m3,1}((ff+1),2) <max(Element Pail(m3,1)(:,2)))
725
                   System(m3) .Element_Failure_Modal(1,2)=Element_Fail{m3,1}((ff+1),1);
726
               end
727
           end.
728
      end
729 % 3. Calculation of support length of all MPs
730 for m4=1:No of Elements
           System (m4) Element Failure Modal Length (1,1) = (System (m4) Element Failure Modal (1 4
       ,2)-System(m4).Element_Failure_Modal(1,1));
73.2
733
734
735
      ₹ Build the sequence of Augmentation Sub-System step wise
73.6
       * Connection between sub system is considered serial connection
737
      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):...
738
                               System (2) .Element_global_No(1,1) System (2) .Element_Failure_S 🗸
      upport Length(1,1) System(2). Element Failure Modal Length(1,1):...
739
                               System(3).Element global No(1,1) System(3).Element Failure S 🗷
      upport Length(1,1) System(3). Element Failure Modal Length(1,1):...
740
                               System (5) .Element global No (1,1) System (5) .Element Failure S &
      upport Length (1,1) System (5) .Element Failure Modal Length (1,1):...
741
                               System (7) .Element global No (1,1) System (7) .Element Failure S &
      upport Length (1,1) System (7) . Element Failure Modal Length (1,1) . . .
742
                               System(9).Element global_No(1,1) System(9).Element Failure 8 €
      upport_Length(1,1) System(9).Element_Failure_Modal_Length(1,1);...
743
                               System (13) .Element global No (1,1) System (13) .Element Failure v
      _Support_Length(1,1) System(13).Element_Failure_Modal_Length(1,1);...
744
                               System (14) . Element global No (1.1) System (14) . Element Failure &
       Support Length(1,1) System(14) . Element Failure Medal Length(1,1) ; ...
745
                               System (16) . Element global No (1,1) System (16) . Element Failure 🗸
       _Support_Length(1,1) System(16) .Element_Failure Modal Length(1,1);...
746
                               System(18).Element_global_No(1,1) System(18).Element_Failure &
       _Support_Length(1,1) System(18) .Element_Failure_Modal_Length(1,1);...
747
                               System(19).Element global No(1,1) System(19).Element Failure ✔
      _Support_Length(1,1) System(19) .Element_Failure_Modal_Length(1,1);...
748
                               System (23) .Element_global_No(1,1) System (23) .Element_Failure *
```

C:\Documents and Settings\Ibrahim\Desktop\Daily w...\rmAE.m

December 2, 2004

759 System (50) .Element global No(1,1) System (50) .Element Failure 🗸 Support Length (1,1) System (50) Element Failure Modal Length (1,1) ... 760 System (S1) .Element global No (1,1) System (S1) .Element Failure & Support Length(1,1) System(51). Element Failure Modal Length(1,1);... 761 System (52) .Element global No (1,1) System (52) .Element Failure & \_Support\_Length(1,1) System(52) .Element\_Failure\_Modal\_Length(1,1);... 762 System (56) . Element global No (1,1) System (56) . Element Pailure 🗸 Support Length(1,1) System(56) .Element Failure Modal Length(1,1) / ... 763 System (57) .Element global No(1,1) System (57) .Element Failure & Support Length(1,1) System(57) Element Failure Modal Length(1,1);... 764 System (63) . Element global No (1,1) System (63) . Element Failure & \_Support\_Length(1,1) System(63) .Element\_Failure\_Modal\_Length(1,1);... 765 System (59) . Element global No (1,1) System (59) . Element Failure 🗸 \_Support\_Length(1,1) System(59) .Element\_Failure\_Modal\_Length(1,1);... 766 System (60) . Element global No (1,1) System (60) . Element Failure \* Support Length(1,1) System(60) Element Failure Modal Length(1,1);... 767 System (65) . Element global No (1,1) System (65) . Element Failure & \_Support\_Length(1,1) System(65).Element\_Failure\_Modal\_Length(1,1)];

Page 22

4:54:54 PM

```
768
769
      * Determine the controling MF, (first MF with max support, then MF with max
770
      * modal)
771
772
      [Row_Fail,Col_Fail] = size (System_Fail_Information);
773
774
      max support=max(System Fail Information(:,2));
775
      max_support_elements=find(System_Fail_Information(:,2)==max_support);
776
     [Row max support elements, Col max support elements] = size (max support elements);
777
778 if (Row max support elements>1)
          max modal-max(System Fail Information(:,3));
779
780
          max modal elements=find(System Fail Information(:,3) == max modal);
781
          [Row max modal elements, Col max modal elements] = size (max modal elements);
782
          if (Row max modal elements:1)
783
              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, v
      1),3111
784
          end
785
          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));
786
787
788
      System Fail-|System Fail Information(max support elements(1,1),1) System Fail Inform &
      ation(max support elements(1,1),2) System Fail Information(max support elements(1,1) &
      ,3)1:
789
      *************************
790
791
      *** ..... Step TEN-POUR..... use the MP of the controling Element
792
      System_Fail_MF=Element_Fail{System_Fail(1,1),1};
793
794
      [row_t,col_t]=size(System_Fail_MF);
795
796 Resilience Index=0;
797 nominator=0;
798 denominator=0;
799 for t=1:(row_t-1)
800
          nominator=nominator+(0.5*(System Fail MF(t+1,1)+System Fail MF(t,1)));
801
          denominator=denominator+(System Fail MF(t,2)*(0.5*(System Fail MF(t+1,1)+System ✓
      Fail MF(t,1))));
802
     end
803
      Resilience Index-nominator/denominator
804
```

## II-C CUSTOM MATLAB FUZZY SCRIPTS

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

```
1 function A=triangular(alpha,a,b,c)
     *TRIANGULAR Fuzzy Number
 3
    * Last Modified: Sunday, July 25, 2004 @6:15pm
 5 if (b<a),
 6
       disp('Not valid input - b must be >= a');
 7
        break;
 B elseif (c<b),
 9
       disp('Not valid input - c must be >= b');
10
       break:
11 end
12
13
14
    [Row_al,Col_al]=size(alpha);
15
16 No_alpha_step=Col_al-1;
17
    alpha_step=max(alpha)/No_alpha_step;
18
19
         for 1=1:((2*No alpha step)+1)
20
                 if (i <= No alpha step)
21
                    MFN(1,1)=a+((1-1)*alpha step*(b-a));
22
                    MFN(1,2)=min(alpha)+((i-1)*alpha_step);
23
                 else
24
                    MPN(1,1)=b+((1-1-No_alpha_step)*alpha_step*(c-b));
25
                    MFN(1,2)=max(alpha)-((1-1-No alpha step)*alpha step);
26
                 and
27
         end
28 A=[MFN(:,1) MFN(:,2)];
```

```
1 function A=trapizoidal (alpha,a,b,c,d)
2 %TRAPIZOIDAL Puzzy Number
 3 % Last Modified: Sunday, July 25, 2004 @7:00pm
 5 if (b<a),
 6
       disp('Not valid input - b must be >= a');
 7
       break:
 8 elseif (c-a),
 9
       disp('Not valid input - c must be >= a');
10
        break;
11 elseif (dea),
       disp('Not valid input - d must be >= a');
12
13
        break;
14 elseif (c<b),
15
       disp('Not valid input - c must be >= b') r
16
        break;
17 elseif (dch).
18
       disp("Not valid input - d must be >= b');
19
        break;
20
    elseif (dec),
21
        disp('Not valid input - d must be >= c');
22
        break;
23 end
24
25
26
    [Row al, Col al]=size(alpha);
27
28 No alpha step=Col al-1;
29 alpha_step=max(alpha)/No_alpha_step:
30
31
         for 1=1:((2*No_alpha_step)+2)
32
                if (i<=No_alpha_step)
33
                    MFN(i,1)=a+((i-1)*alpha_step*(b-a));
34
                    MFN(i,2)=min(alpha)+((i-1)*alpha step):
35
                end
36
37
                if (i==(No_alpha_step+1))
38
                    MFN(1,1)=b;
39
                    MFN(1,2)=1;
40
                end
41
42
                if (i==(No_alpha_step+2))
```

```
C:\Documents and Settings\Ibrahim\Desktop\...\trapizoidal.m
                                                                           Page 2
                                                                      5:00:03 PM
December 2, 2004
43
                   MPN(1,1)=c;
44
                   MFN(1,2)=1;
45
               end
46
47
               if (i>(No_alpha_step+2))
48
                   MFN(i,1)=c+((i-2-No_alpha_step)*alpha_step*(d-c));
49
                   MFN(i,2)=max(alpha)-((i-2-No_alpha_step)*alpha_step);
50
                end
51
        end
52 A=[MFN(:,1) MFN(:,2)];
```

December 2, 2004

```
function result = fuzzymath(alpha,A,B,operator)
     % Last Modified: Monday, July 26, 2004 @9:00am
 3
 4
 5 [Row_al,Col_al]=size(alpha); * basic information about membership values (alpha)
 6 No alpha step=Col al-1;
 7
    alpha_step=max(alpha)/No_alpha_step;
 B
 9 $ Basic information about the two membership functions
10 [Row_A,Col_A]=size(A);
11 [Row_B,Col_B] = size(B) /
12
13 % a loop to make one matrix with left and right value to perform the interval calcul &
     ations (this is done for the
14
    %% two membership functions)
15
         * the LEFT values
16
    for i=1:Col al
         AA(1,2)-A(1,1);
17
18
         BB(1,2)-B(1,1),
19
     end
20
21
         ₹ the RIGHT values
22 counter=0;
23
    for j=0:No_alpha_step
24
         counter-counter+1;
25
         AA(counter,3)=A((Row A-j),1);
26
         BB(counter, 3) = B((Row_A-j), 1);
27
     end
28
         ₹ corresponding alpha values
29
    for k=1:Col al
30
         AA(k,1)-A(k,2);
31
         BB(k,1)=B(k,2);
32
    end
33
34 % Interval arithmatic
35 for L=1:Col al
36
         if stromp(operator, 'sum'),
37
                 result_interval(L,1)=A(L,2);
38
                 result interval(L,2) = AA(L,2) +BB(L,2);
39
                 result interval(L,3)=AA(L,3)+BB(L,3);
40
41
         elseif stromp(operator, 'sub'),
```

```
C:\Documents and Settings\Ibrahim\Desktop\Da...\fuzzymath.m Page 2
December 2, 2004 5:00:25 PM
```

```
42
                 result_interval=[(AA(L,1)-BB(L,2)) (AA(L,2)-BB(L,1))];
43
                 result_interval(L,1)=A(L,2);
44
                 result_interval(L,2)=AA(L,2)-BB(L,3);
45
                 result_interval(L,3)=AA(L,3)-BB(L,2);
46
47
         end
48
     end
49
50
51 % Rebuilding the resultant Membership function in the same way as in the
    * input membership functions
52
53
         % the Triangular case
54 if (A(Col_al,2)==1) & (A((Col_al+1),2)==1)
55
         for y1=1:Col_al
56
             result (y1,2)-A(y1,2);
57
             result(y1,1)=result_interval(y1,2);
58
59
         for y2-1:No alpha step
60
             result ((Col al+y2),2)=A((Col al-y2),2);
61
             result((Col_al+y2),1)=result_interval((Col_al-y2),3);
6.2
         end
63
         * the Trapizoidal case
64
     elseif (A(Col_al,2)==1) & (A((Col_al+1),2)==1)
65
         for 21=1:Col al
66
             result(s1,2)-A(s1,2);
57
             result(z1,1)=result interval(z1,2);
68
         end
69
         for 22-0:No alpha step
70
             result ((Col_al+z2+1),2)=A((Col_al-z2),2);
71
             result((Col_al+z2+1),1)=result_interval((Col_al-z2),3);
72
         end
73
     end
74
75
76
```

```
function C = fuzzylogmath(Domain,alpha,A,B,operator)
     % Last Modified: Priday, August 19, 2004 08:00pm
 3
 4
 5 [Row_al,Col_al]=size(alpha); * basic information about membership values (alpha)
 6 No alpha steps=Col al-1;
 7
    alpha_step=max(alpha)/No_alpha_steps;
 B
 9
10 [Row_Domain,Col_Domain]=size(Domain); * basic information about Domain-values (Domai &
     n-axis)
11 No_Domain_steps=Col_Domain-1;
12 Domain step=max(Domain)/No Domain steps;
13
14
15 ***-----Step--ONB----- (Modifying MYs)------
17
         % Add end values for A and B to gover the whole range of Domain-values....to avo ⊌
     id giving negative values in case
18
         ३% of not defining the end Domain-values of A and B.
19
20
         if min(Bomain) <A(1,1)
21
            AA(1,1)=min(Domain);
            AA(1,2)-D;
23
            A=[AA(1,1) AA(1,2);A];
24
25
    [Row A,Col A]=size(A); * Basic information about the MP of A, it is here to take in ✔
     to account any change
26
                            ₹ due to the change in dimensions after the first condition
27
28
         if max(Domain) >A(Row A, 1)
29
            AA((Row_A+1),1)=max(Domain);
30
            AA((Row A+1),2)=0;
31
            A=[A;AA((Row A+1),1) AA((Row A+1),2)];
32
33
34
         if min(Domain) <B(1,1)
35
            BB(1,1)=min(Domain);
36
            BB(1,2)=0;
37
            B=[BB(1,1) BB(1,2);B];
38
    [Row_B,Col_B]=size(B): * Basic information about the MF of B, it is here to take in ★
```

```
to account any change
40
                   * due to the change in dimensions after the first condition
41
42
      if max(Domain) >B(Row B,1)
43
        BB((Row B+1),1)=max(Domain);
44
        BB((Row B+1),2)=0;
45
        B=[B;BB((Row_B+1),1) BB((Row_B+1),2)];
46
48 88
50 ***------Step---TWO------(Interpolate for Domain steps)------
51 888------
52 A=interpl(A(:,1),A(:,2),Domain);
53 A-[Domain;A];
54 A=A1:
55
56 B=interpl(B(:,1),B(:,2),Domain),
57
   B=[Domain,B];
58 B-B':
60 88-----
61
62
   ***-----Step---THREE----- (Identify the operator and perform operation)
63
64
65
   if stromp(operator, 'minimum'),
66
      C=min(A(:,2),B(:,2))
67
68
69
   if stromp(operator, 'maximum'),
70
      C=max(A(:,2),B(:,2))
71
   end
72
73 C-[(Domain') C];
74
75
   [Row C,Col C] = size(C);  # Basic information about the MF of C
76 for w=1:Row_C
77
      if C(w, 2) <0
78
        C(W, 2) = 0;
79
      end
80
   end
```