To the Graduate Council:

I am submitting herewith a dissertation written by Renato Silvio da Frota Ribeiro entitled "Fuzzy Logic Based Automated Irrigation Control System Optimized Via Neural Networks". I have examined the final copy of this dissertation for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Doctor of Philosophy, with a major in Biosystems Engineering.

Ronald E. Yoder, Major Professor

We have read this dissertation
and recommend its acceptance:

Accepted for the Council:

Associate Vice Chancellor and
Dean of The Graduate School

FUZZY LOGIC BASED
AUTOMATED IRRIGATION CONTROL SYSTEM
OPTIMIZED VIA NEURAL NETWORKS

A Dissertation

Presented for the

Doctor of Philosophy

Degree

The University of Tennessee, Knoxville

Renato Silvio da Frota Ribeiro

August 1998

DEDICATION

I dedicate this dissertation to my wife

Ana Virginia

to my son

Rafael

and to my daughter

Carolina.

ACKNOWLEDGEMENTS

I would like to thank the "Universidade Federal do Ceará" and all the friends there for the conceded opportunity. Specifically, I would like to thank Dr. Francisco de Souza, for his encouragement towards my doctoral degree. I would also like to thank the Brazilian National Council for Scientific and Technological Development (CNPq) for the financial support for school and maintenance and its people for their attention during these years.

I would like to express my sincere gratitude to my major professor, Dr. Ronald E. Yoder, for his friendship, guidance, and support. Also, I would like to thank Dr. John B. Wilkerson, committee member, for his help and suggestions in the construction of the electronic components involved in this study. I would like to express my appreciation to Dr. D. Raj Raman, Dr. Robert E. Uhrig, and Dr. David L. Coffey, also members of my doctoral committee, for their review, comments, and assistance.

I am also grateful to The Tennessee Agricultural Experiment Station for funding part of this research. Thanks are also expressed to Dr. Arnold Saxton, Dr. Charles A. Mullins, Dr. Allan Straw, Mr. B. David Russell, Mr. John Buchanan, Mr. Chris Wheeler, and the administration of the Plateau Experiment Station, for their support during the field part of this research.

I would like to express my thanks to my fellow students, the faculty, the staff, and the administration of the Department of Agricultural and Biosystems Engineering, represented currently by Dr. C. Roland Mote, and formerly by Dr. Fred D. Tompkins, for their support during these years.

Thanks also go to the friends that we had the pleasure to meet in Knoxville.

A special gratitude to my parents-in-law Aida and Lázaro Medeiros for their unconditional support in these years away from home.

Finally, I would like to thank my wife, Ana Virginia, and our children Rafael and Carolina, for their love, patience, and support during this remarkable experience.

ABSTRACT

This research focused on application of fuzzy logic and neural networks to control an automatic irrigation system and to estimate evapotranspiration. A fuzzy logic system is able to simultaneously manage numerical data and linguistic statements, mimicking human reasoning. Fuzzy control basically refers to the control of processes through fuzzy linguistic descriptions. An artificial neural network has the ability to utilize examples taken from data and to organize the information into a model that relates the input and output variables. An automated irrigation control system was developed using the fuzzy logic approach as an alternative to improve the current management technologies applied in real-time irrigation scheduling and control. Data from an automated weather station and from soil moisture sensors were used as inputs to the system. A fuzzy logic estimation system was designed to map solar radiation and relative humidity into evapotranspiration. The fuzzy control system processes the estimated evapotranspiration and soil moisture contents at two soil depths using linguistic knowledge to produce a control output that is sent to a solenoid valve. The fuzzy irrigation control system was implemented and tested in 1997 using microirrigation under plastic mulch in bell peppers *(Capsicum annuum L.)*. The fuzzy irrigation control system presented a very good performance in response to the climatic and soil moisture variations in a fully automated manner. The soil water potential was kept between –6.5 kPa and –15.7 kPa, evidencing the feasibility of the application of fuzzy logic to automated irrigation control systems. Neural networks were then used to process the system input and output data for redefinition of membership functions, in order to optimize the system.

TABLE OF CONTENTS

CHAPTER PAGE

1. **INTRODUCTION** 1
   - Statement of the Problem 1
   - Research Objectives 2

2. **LITERATURE REVIEW** 3
   - Irrigation Scheduling 3
   - Evapotranspiration 4
   - Lysimeters 7
   - Weather Sensors 8
   - Soil Moisture Sensors 9
   - Irrigation Control 11
   - Artificial Intelligence in Irrigation 13

3. **ARTIFICIAL INTELLIGENCE** 17
   - Fuzzy Logic 18
   - Introduction to Fuzzy Logic and Fuzzy Control 18
   - Introduction to Control Using Fuzzy Logic 19
     - *System Overview* 19
   - The Fuzzy Logic Control Concept 21
   - Components of Fuzzy Control 23
   - The Expert Knowledge Base 24
   - Rule Base 24
     - *Inference Engine* 25
   - Introduction to the Mathematics of Fuzzy Logic 25
     - *The Fuzzy Concept* 25
     - *Universe of Discourse* 27
     - *Membership Functions* 28
     - *Fuzzification* 32

Fuzzy Set Operations 33
Evaluation of Linguistic Rules and Inference Techniques 34
Defuzzification 37
Neural Networks 39
Backpropagation Algorithm 42
An Example of Backpropagation Training 46
Data Processing for Neural Networks 48
Neural Methods in Fuzzy Systems 52
Fuzzy Neural Hybrids 53
Adaptive Network-Based Fuzzy Inference System 54

4. **SYSTEM DESIGN** 59
Goal of the System 59
Fuzzy System Design 59
Fuzzy Evapotranspiration Estimator (FEE) 60
Fuzzy Algorithm 64
Fuzzy Input/Output Variables 64
Membership Functions 65
Rule Base 67
Inference Engine and Defuzzifier 71
Fuzzy Irrigation Control System (FICS) 74
Fuzzy Algorithm 74
Fuzzy Input/Output Variables 74
Membership Functions 75
Rule Base 79
Inference Engine and Defuzzifier 80

5. **SYSTEM IMPLEMENTATION** 84
Conventional Irrigation Control System (CICS) 84
Inputs 84
Crop 85

Statistical Design 87
Randomization 87
Irrigation System 90
Control System 93
Hardware 93
Automated Weather Station 93
Soil Moisture Sensors 94
Thermocouple 94
Multiplexer 98
Datalogger 98
Personal Computer 98
PC Digital Input/Output Board 99
Solenoid Valves 99
Control System Operation 99
Crop Planting 99
System Installation 103

6. **RESULTS AND ANALYSIS** 104
Fuzzy Evapotranspiration Estimator 104
System Performance 108
No Irrigation 108
Conventional Irrigation Control System 112
Fuzzy Irrigation Control System 112
System Performance Analysis 112
Irrigation Requirements 117
Statistical Analysis 117
Fuzzy Irrigation Control System Output Unit Conversion 119

7. **SYSTEM OPTIMIZATION USING NEURAL NETWORKS** 122
ANFIS Procedure 122
Fuzzy Evapotranspiration Estimator 123

Training Data 123
Training Fuzzy System 124
ANFIS Training 124
Simulation 124
Evapotranspiration Estimation and Solar Radiation 130

8. **SUMMARY AND CONCLUSIONS** 133
Topics for Further Research 135

REFERENCES 136

APPENDICES 145
APPENDIX A 146
APPENDIX B 171
APPENDIX C 173

VITA 176

LIST OF FIGURES

FIGURE		PAGE
3.1	A simple block diagram illustration of a general control system	20
3.2	A block diagram illustration of an all-fuzzy control system	20
3.3	A block diagram representation of fuzzy control performed on a crisp system	21
3.4	Basic structure of a fuzzy logic controller performed on a crisp system	23
3.5	A crisp characteristic function compared to a fuzzy membership function	29
3.6	Characteristic functions of the crisp sets defined for the variable *outdoor temperature*	30
3.7	Fuzzy membership functions for the term set defined for the variable *outdoor temperature*	31
3.8	Example of Mamdani's inference using two inputs and two rules	36
3.9	Structure of a typical neuron element with several inputs and correspondent weights	43
3.10	A sigmoidal function is a typical smooth and non-linear activation function	44
3.11	Example of a neural network type 2-2-1 before backpropagation algorithm	46
3.12	Neural network type 2-2-1 after backpropagation algorithm	48
3.13	Flow chart for a neural network development with backpropagation	51
3.14	(a) Evaluation of a network of rules. (b) Illustration showing the correspondent ANFIS architecture	56

4.1 Relationship between solar radiation and ET 62

4.2 Relationship between relative humidity and ET 62

4.3 Relationship between air temperature and ET 63

4.4 Relationship between wind speed and ET 63

4.5 Membership functions defined for the input variable *radi* 68

4.6 Membership functions defined for the input variable *humi* 68

4.7 Membership functions defined for the output variable *et* 69

4.8 Example of fuzzy inference and defuzzification for the case *radi* = 950 and *humi* = 30 72

4.9 Output surface of the fuzzy evapotranspiration estimator 73

4.10 Membership functions defined for the input variables *et* 77

4.11 Membership functions defined for the input variables *soil1* and *soil2* 77

4.12 Membership functions defined for the output variable *irr* 78

4.13 Example of fuzzy inference and defuzzification for the case *et* = 0.12, *soil1* = 0.70, and *soil2* = 0.80 82

4.14 An example of an output surface of the fuzzy irrigation control system for a fixed value of *et* = 0.20 83

5.1 Flow chart for the conventional irrigation control system 86

5.2 Randomized block layout with 18 experimental units 88

5.3 Experimental unit layout showing double rows 89

5.4 Irrigation system layout showing drip lines, pipes, solenoid valves, and filtration system 91

5.5 Filtration system composed of a sand filter and a screen filter 92

5.6 Location of soil moisture sensors at each experimental unit 95

5.7 Soil moisture sensors being installed at the field 96

5.8 Location of the thermocouples installed at six experimental units 97

5.9 (a) Diagram of the fuzzy irrigation control system (FICS) including the fuzzy evapotranspiration estimator (FEE). (b) Diagram of the conventional irrigation control system (CICS) 100

5.10 Diagram of the hardware used to implement the fuzzy irrigation control system and the conventional irrigation control system.....101

5.11 PC digital input/output board used in the irrigation control system and the missing pulse detector.....102

6.1 Comparison between fuzzy ET estimations and lysimeter ET measurements 106

6.2 Linear regression between fuzzy ET estimations and lysimeter ET measurements 107

6.3 Precipitation during system operation 109

6.4 Soil water potential (average) in non-irrigated experimental units at the 15-cm depth.....110

6.5 Soil water potential (average) in non-irrigated experimental units at the 30-cm depth.....111

6.6 Soil water potential (average) at the 15-cm depth in experimental units using the conventional control.....113

6.7 Soil water potential (average) at the 30-cm depth in experimental units using the conventional control.....114

6.8 Soil water potential (average) at the 15-cm depth in experimental units using the fuzzy logic control 115

6.9 Soil water potential (average) at the 30-cm depth in experimental units using the fuzzy logic control 116

6.10 Randomized block layout showing the yields obtained from each experimental unit 118

6.11 Relationship between control value output and soil water potential at the 15-cm depth 120

6.12 Relationship between control value output and soil water potential at the 30-cm depth 121

7.1 Membership functions for the input variable *radi*, after optimization 125

7.2 Membership functions for the input variable *humi*, after optimization 125

7.3 Comparison between fuzzy ET estimations and lysimeter ET measurements, after optimization 128

7.4 Linear regression between fuzzy ET estimations and lysimeter ET measurements, after optimization 129

7.5 Linear regression between daily solar radiation sensor measurements and daily lysimeter ET measurements, using data taken during the operation of the system 131

7.6 Comparison between daily solar radiation measurements and daily lysimeter ET measurements, using data taken during the operation of the system 132

LIST OF TABLES

TABLE		PAGE
3.1	Term set that describes the linguistic variable *outdoor temperature*	27
3.2	Example of initial weights for training a neural network	47
3.3	Comparison between fuzzy and neural systems	53
4.1	Weather data and lysimeter measurements from a crop production period	61
4.2	Inputs and output description for fuzzy evapotranspiration estimator	64
4.3	Input fuzzy sets for fuzzy ET estimator	66
4.4	Output fuzzy sets defined for fuzzy ET estimator	69
4.5	FAM table that describes the fuzzy evapotranspiration estimator	70
4.6	Input and output description for fuzzy irrigation control system	75
4.7	Input fuzzy sets defined for fuzzy irrigation control system	76
4.8	Output fuzzy sets defined for fuzzy irrigation control	78
6.1	Daily lysimeter ET measurements and fuzzy ET estimations	105
7.1	*First-order Sugeno* output constants for the optimized ET estimator	126
7.2	Lysimeter ET measurements and fuzzy ET estimations, after optimization	127

CHAPTER 1

INTRODUCTION

In the past few years irrigation engineering has received enormous contributions from technologies related to sensing, communication, data processing, and actuating devices. More recently, the application of intelligent technologies, such as expert systems, fuzzy logic, and artificial neural networks, is opening a new perspective in the search for improved management and precision in irrigation.

Statement of the Problem

In many areas around the world the success of agriculture is totally dependent on irrigation. Even in humid areas, irrigation improves yields by supplying water to the crops in opportune situations. Water is a scarce and fragile resource that has to be managed wisely. Irrigation represents the major use of fresh water in nature and this need is increasing. Therefore, there is a need for more precision in irrigation technology.

Irrigation scheduling is a decision-making process to determine the timing and quantity of water applications, based on soil, crop, and weather information. Although decades of research have been devoted to determining accurate scheduling methods, the application of irrigation scheduling is still not widespread. The acceptance of the technology by farmers is limited by the increase in quantity of actions and quality of decisions to be made on a daily basis. A solution for this problem is achievable through total automation.

An automated irrigation control system is a step beyond irrigation scheduling technology. Control systems require devices for sensing, communicating, data processing, and actuating the system based on logical decision algorithms. Several different approaches for automatic irrigation control have been presented in recent years. Most of them rely on just one of the three basic components of the soil-plant-atmosphere system.

Research Objectives

The primary goal of this research was to develop a methodology to improve management of irrigation systems using artificial intelligence. The specific objectives of this study were:

1. To develop a fully automated control in real-time for microirrigation systems.
2. To minimize the number of weather input parameters, thereby simplifying the systems and increasing usage potential.
3. To reduce water and energy consumption in microirrigation while maintaining or even increasing yields.
4. To compare the new methodology to the current technology presently being applied to irrigation control.

CHAPTER 2

LITERATURE REVIEW

Irrigation Scheduling

Irrigation scheduling is a set of technical procedures used to determine the proper timing and amount of water to be applied in response to crop requirements. An adequate irrigation scheduling optimizes water use, and consequently energy consumption, without yield reduction. Irrigation scheduling can reduce irrigation water use by decreasing percolation of water beneath the root zone. In some cases, irrigation scheduling may actually increase irrigation water use, while concurrently increasing crop yield by avoiding critical soil water deficits that reduce crop yields (Howell, 1996).

Irrigation scheduling methods are based on two approaches: predictive and reactive. Predictive methods are based on soil water balance computations to estimate water depletion in the root zone. Soil water balance calculations require estimates of soil storage capacity, rooting depth, allowable moisture depletion, and crop evapotranspiration. Reactive or real-time methods are based on: (1) evapotranspiration calculations or measurements, (2) monitoring soil moisture content, or (3) monitoring plant water status.

Research and recommendation of irrigation scheduling technology started about fifty years ago (Clyma, 1996; Fereres, 1996; Shearer and Vomocil, 1981; van Bavel and Wilson, 1952), and by the 1970s, computer programs (Jensen, 1969; Jensen et al., 1970) were used for water budget calculations.

As personal computers have becoming increasingly common, the potential for computer-based decision support systems for irrigation management has also increased. Computerized irrigation scheduling systems have been developed by Cahoon et al. (1990), Phene et al. (1992), and Zazueta et al. (1994).

Evapotranspiration

Evapotranspiration represents the water loss from a combined surface of vegetation and soil through the conversion of liquid water to a gas. The process is dynamic, driven by the available energy (net radiation), and limited by the rate of energy exchange between the surface and the overlying atmosphere (sensible and latent heat fluxes), the available soil water, and the ability of the plant to conduct water from the soil to the leaf. The quantification of this water loss represents the actual evapotranspiration. The components of actual evapotranspiration are the evaporation of water from the soil surface or the free water present on plant surfaces and the evaporation of water through the plant stomates, i.e., transpiration (Hatfield and Fuchs, 1990).

Potential evapotranspiration quantifies the evaporative demand of the atmosphere. It was first defined relative to crop surfaces by Penman (1948) as "the amount of water transpired per unit ground area in a unit time by a short, green crop, completely shading the ground, of uniform height and never short of water." In this definition it is assumed that the crop occupies a sufficiently large area, so that an increase in area will not cause a change in the transpiration rate. The condition of a non-limiting supply of water is never achieved because the resistance of water flow through plants and soil has a finite value greater than zero. The theoretical value of potential evapotranspiration can be determined

by the Penman-Monteith model (Penman, 1948; Monteith, 1965), with a surface resistance set to zero.

Reference evapotranspiration is the actual water loss by a specific crop that covers a soil surface with unlimited soil water supply. Doorenbos and Pruitt (1975; 1977) introduced the reference evapotranspiration concept as "the rate at which water, if available would be removed from the soil and plant surface of a specific crop (i.e., the reference crop)." The two most common reference crops are alfalfa and short grass. An estimate for a crop other than the reference is calculated by multiplying the reference evapotranspiration by a "crop coefficient", specific for each crop (Jensen et al., 1990).

Evapotranspiration, as water vapor flux density from a vegetated surface, can be described by either convective transport, or by the surface energy balance. Neither of these approaches explains the influence of meteorological factors on the magnitude of the vapor flux density. Penman (1948) and later Monteith (1965) combined them to show that for a given vegetation surface the evapotranspiration depends only on the net radiant energy absorbed by the heat flux density into the ground, the air temperature and the corresponding water vapor pressure deficit, the wind speed, and the surface resistance.

Solar radiation, atmospheric radiation, air temperature and humidity, and wind speed are the meteorological inputs. Canopy cover and plant height determine the absorption of radiation and the transport properties of the wind. The surface resistance accounts for the path of vapor from the sites in the plant tissues or the soil where water is in liquid form, to the surface in contact with the atmosphere. The conductance of this path decreases with the availability of moisture.

A count of methods for estimating evapotranspiration has not been made, but the number must be at least fifty. They can be grouped as: (1) radiation methods, (2) temperature methods, (3) pan evaporation methods, and (4) Penman methods. Jensen et al. (1990) presented a detailed review of the methods and their use to estimate irrigation water requirements. Some of the estimation methods are used because they require few input climatic variables; however, their results often are not accurate. On the other hand, more accurate models require a large number of parameters that are only available for a few sites.

Penman methods have been through an evolutionary process. Allen et al. (1989) evaluated a number of equations using data from 11 international lysimeter sites, and found that the Penman-Monteith equation provided the best estimates of daily and monthly average reference evapotranspiration and was the most consistent across all locations evaluated. Even though the Penman-Monteith model is physically based, it contains empirical estimations and evidence of underprediction in some locations has been found (Ribeiro, 1996; Steduto et al., 1996). For hourly calculations, the FAO Penman-Monteith method is

$$ET_o = \frac{0.408\Delta(R_n - G) + \gamma \frac{37}{T+273} U_2 (e_a - e_d)}{\Delta + \gamma(1 + 0.34U_2)} \qquad (2.1)$$

where:

ET_o = reference crop evapotranspiration (mm/h)

Δ = slope of vapor pressure curve (kPa/°C)

R_n = net radiation at crop surface (MJ m^{-2} h^{-1})

G = soil heat flux (MJ m^{-2} h^{-1})

γ = psychrometric constant (kPa/°C)

T = average air temperature at 2-m height (°C)

U_2 = mean wind speed at 2-m height (m/s)

e_a = saturation vapor pressure (kPa)

e_d = actual vapor pressure (kPa)

The Food and Agricultural Organization (FAO) has recommended the Penman-Monteith equation as the best combination equation currently in use. Full details of the FAO Penman-Monteith method and procedures for determining various parameters, recommended coefficients, and units are in the ICID Bulletin Vol. 43, No. 2 (Allen et al., 1994a; 1994b).

Lysimeters

Evapotranspiration can be measured directly using a lysimeter, a tank used to isolate a soil mass containing growing plants (crops) from the surrounding soil. Its purpose is to permit the calculation of a water balance by eliminating percolation beyond the root zone and/or to prevent the addition of water from a water table below the root zone. Lysimeters range from a few centimeters in diameter to as large as 10 meters in diameter. The depth of lysimeters is also quite variable, as are the construction materials and methods. Wheeler (1998) presents a review of the current lysimeter technologies.

Lysimeters can be classified into three categories: (1) nonweighing, constant water-table type, which provides reliable data in areas where a high water table normally exists and where the water table level is maintained essentially at the same level inside as outside the lysimeter; (2) non-weighing, percolation type, in which changes in water stored in the soil are determined by sampling or by neutron scatter methods and the rainfall and percolate are measured (often used in areas of high precipitation); and (3) weighing types, in which changes in soil water are determined either by weighing the entire unit with a mechanical scale, counterbalanced scale and load cell, or by supporting the lysimeter hydraulically (Jensen et al., 1990). A detailed summary of the use of lysimeters for evapotranspiration can be found in publications by Aboukhaled et al. (1982), Howell et al. (1985), Marek et al. (1986), and Pruitt and Lourence (1985). At the present time, lysimeters, particularly sensitive weighing lysimeters, are regarded as the most reliable method of measuring evapotranspiration (Burman and Pochop, 1994).

Although lysimeters provide the best available measurements of evapotranspiration, the construction and operation of lysimeters is expensive and requires specially trained personnel. Equipment, instrumentation, and data collection require continual monitoring and adjustments as needed. Therefore, lysimeters are used most frequently for research purposes. Researchers are now focused on the possible substitution of reduced-scale size lysimeters (Wheeler, 1998) for large size lysimeters.

Weather Sensors

Traditionally, data from conventional weather stations have been used in conjunction with appropriate combination equations to provide estimates of daily reference crop

evapotranspiration. With the increasing abundance of automatic weather stations and less empirical combination equations, it is possible to calculate evapotranspiration over small time steps. Hess (1996) presented comparisons between evapotranspiration calculated by conventional weather observations and automatic weather station data and found no significant differences. However, under hot and dry conditions, the use of hourly data can result in a significantly greater calculated evapotranspiration.

Typically, the following weather variables are measured by specific sensors assembled in an automatic weather station: (1) solar radiation, measured by a pyranometer, (2) air temperature, measured by a thermistor, (3) relative humidity, measured by thin-film capacitors, (4) wind speed, measured by cup anemometers, (5) wind direction, measured by a potentiometer vane, and (6) precipitation, measured by a tipping bucket. The sensor outputs are sampled by dataloggers at time intervals sometimes less than one second, and averaged over a required time interval. Data from the datalogger memory are usually transmitted automatically via telephone lines and modems to a centrally located microcomputer that processes all the data collected (Phene et al., 1990; Campbell, 1992).

The weather station operation and instrument maintenance is critical to recording reliable data. The weather station location is also important for obtaining representative information. Several guides are available that describe the proper location and exposure of weather stations (Szeicz, 1975; Doorenbos, 1976; McGuiness et al., 1979).

Soil Moisture Sensors

Determination *in situ* of soil moisture content is of considerable interest for irrigation management. This interest has led to the development of technologies that can be

automated for determining soil moisture content, or that can be used for input to dataloggers or controllers.

Sensor development has followed the criteria of having low labor requirements, not requiring destructive sampling (after installation), and of being adaptable to electronic measurement and recording. Sensors that meet these criteria are tensiometers, neutron gages, electrical resistance sensors, electrical capacitance sensors, heat dissipation sensors, and time domain reflectometers (TDR). It should be noted that neutron gages are limited in remote applications because they cannot be left unattended and that tensiometers are limited to reading soil water contents that correspond to soil tensions of 85 kPa (12.33 psi) or less. A comparative study of eight types of sensors under carefully controlled conditions can be found in Johnson (1995) and Yoder et al. (1998).

Electrical resistance sensors are relatively inexpensive, they can be interfaced with computer controlled systems, and the installation procedures are simple. These devices require calibration to interpret their output when used with automatic data logging equipment. Armstrong and Ligon (1985), Thomson and Armstrong (1987), McCann et al. (1992), and Eldredge et al. (1993), presented calibration for the Watermark[1] model 200 soil moisture sensor (Irrometer, Inc., Riverside, CA). Thomson et al. (1996) compared the Thomson and Armstrong (1987) and the McCann et al. (1992) equations and showed that the two equations deviated significantly. Their comparison indicated that the differences between the two calibration equations may be due to differences in methods of exciting the sensors. Additional studies by Thomson et al. (1996) revealed that modified

[1] The mention of trade names is provided for the benefit of the reader and does not imply endorsement by the author or The University of Tennessee.

Watermark sensors (model 200SS) followed the same calibration curve of Thomson and Armstrong (1987). The Thomson and Armstrong (1987) equation is

$$S = - R / (0.474483 - 0.0138697 * T + 0.000147019 * T^2 - 0.01306 * R) \quad (2.2)$$

where S is soil water potential (kPa), R is sensor resistance (kΩ), and T is soil temperature (°C). Shock et al. (1996) controlled 42 miniaturized drip irrigation systems using over 1000 Watermark model 200SS sensors. They used a sensor calibration equation developed by Shock (1996). This equation is

$$S = - (2.678 + 0.003892 * R) / (1 - 0.01201 * T) \quad (2.3)$$

where the variables are as previously defined except that R has units of ohms (Ω). Bausch and Bernard (1996) compared soil water potential calculated from measured Watermark sensor resistance and soil temperature using equation 2.1 and equation 2.2 to soil water potential measured by tensiometers. They found that the Shock equation performed better than the Thomson and Armstrong equation by not computing highly erroneous values as the tensiometer lower limit was approached

Irrigation Control

The purpose of any control system is to regulate some dynamic variable or variables of a process. Irrigation control systems are expected to control the flow of water to the crops in a manner that maximizes yield while optimizing water and energy consumption.

The control system may be related to (1) hydraulic, (2) mechanical, (3) electro-mechanical, (4) electronic, or (5) computerized control (Duke et al., 1990). A reliable automated, real-time irrigation scheduling and control system has advantages that include lower labor costs, lower plant stress levels, and lower water use (Evett et al., 1996). An advanced irrigation control system should integrate devices for sensing, communicating, data processing, and actuating the system in real-time, based on logical decision algorithms and without human intervention.

Benami and Offen (1984) stated that a higher level of automation is achieved when single hydraulic valves are actuated electronically from field control units. The major irrigation control modes are (1) on-off control, (2) stepwise control, and (3) continuous control (Phene, 1986).

Several different approaches for automatic irrigation control have been presented in recent years. Fangmeier et al. (1990) developed an automatic irrigation control system using infrared thermometers, aspirated psychrometers, and electrical resistance blocks. Automated irrigation control on the basis of a threshold canopy temperature has been shown to be feasible for certain crops (Wanjura et al., 1992; Evett et al., 1996). Phene et al. (1992) have used an automated pan evaporation system to control a drip irrigation system in real-time. This control system worked automatically without manual intervention for two growing seasons. Parchomchuk et al. (1996) demonstrated the feasibility of automatically scheduling microirrigation of vineyards and orchards according to evaporative demand monitored by an electronic atmometer. Ismail and Al-Shooshan (1996) described a field application of an electrotensiometer for an automated feedback system.

Research of irrigation control using soil moisture sensors has been conducted by Phene et al. (1973), Phene and Howell (1984), and Phene (1989). Sensors were used in a feedback mode to maintain a nearly constant soil moisture content in the root zone. They concluded that the performance of the irrigation controller depended on four basic factors: (1) adequate operation of the control hardware, (2) the proper algorithm for the system, (3) a reliable soil moisture sensor, and (4) adequate operation of solenoid valves, pressure regulators, flow meters, and filters. Further studies (Phene et al., 1989) indicated that an irrigation control system based on monitoring soil water potential should have the following characteristics: (1) ability to sample the sensor data automatically, (2) means of comparing the sensor output to a threshold value, and (3) ability to control and monitor irrigation devices.

Artificial Intelligence in Irrigation

In the past few years irrigation engineering has received valuable contributions from artificial intelligence techniques. The applications involved expert systems, fuzzy logic systems, and neural networks, and were focused basically on prediction and modeling. As the knowledge and development of these techniques continue, it is expected that new applications in irrigation will deal with the ability to sense the environment and directly influence it through action.

Hawkins and Burt (1990) developed an expert system that combined the elements of soil science, plant water use, irrigation scheduling, irrigation system design constraints, field geometry, and irrigation system distribution uniformity. It was one of the first large-

scale applications of a user-friendly expert system designed for farmer irrigation management.

McClendon et al. (1991) incorporated a neural network based routine to determine optimal irrigation decisions into a crop growth simulation model. Daily environmental conditions and crop state variables associated with irrigation sequences were used to train the neural network. The resulting neural network was found to predict irrigation decisions reasonably well.

Weather data, soil moisture data, and a trend yield variable were used to train a neural network to predict yields in an application developed by Uhrig et al. (1992). The training set consisted of 25 years of data, during the period of 1960 through 1989. After the training was accomplished, the input data for 1990 was subjected to the network to predict 1990 corn yield. The results were very encouraging. A similar study developed by Koch (1993) examined the degree to which a neural network could map the relationship between the timing and volume of water deliveries during a growing season, and the resulting corn yield. He concluded that the neural network model could be used to allocate marginal increases in total water supply to various stages in the growing season.

Bardossy and Disse (1993) developed two fuzzy rule-based models for infiltration. Instead of calculating the flow with unrealistically high accuracy, these models only approximated the flow (the solution of the physical models). They were based on a few variables that could be measured and interpolated with reasonable accuracy. They stated that fuzzy infiltration models could be extended with additional components such as evapotranspiration, also considered in a fuzzy way.

A fuzzy irrigation decision support system was proposed by Xiang et al. (1994). The model included membership functions for 19 variables of three primary components: (1) variables for plants, (2) variables for water status and flow in soil, and (3) variables for weather. The system prompted the user for inputs, and after all variables were entered and processed, an output value was printed out for comparison with a previously defined threshold. The model predicted reasonable results for irrigation scheduling in peanuts.

A prototype of a real-time expert system for citrus irrigation management was developed for microirrigation water application, cold protection, and fertigation control (Xin et al., 1995). The system integrated water management and control technologies into an effective control system that could be used as a tool by irrigators. Data from a modified tensiometer and an automated weather station were used as inputs to the computer. The control system was successfully implemented for about two years.

Gowing et al. (1996) proposed a rational method for assessing irrigation performance in terms of water supply utility at farm level with the aid of fuzzy logic. In this method the farmers used linguistic expressions rather than numerical ratings to judge the appropriateness of each of the three previously defined factors (predictability, convenience, and controllability), and their importance to them under individual circumstances in the field. The method provided a reliable assessment of the satisfaction of the farmers as water customers in an irrigation district.

Clyma and Martin (1996) used fuzzy logic to develop rules and processes by which a knowledge base of water management techniques and information could be accessed and used to manage a center pivot irrigation system. Their program modeled well the uncertainty associated with irrigation scheduling decisions.

A study was conducted by Williams and Zazueta (1996) to determine the ability of a neural network to accurately predict an irrigation schedule for soybeans using a minimized number of weather input parameters. Maximum daily temperature and solar radiation had the greatest influence on evapotranspiration. The results confirmed that a neural network could be used to determine an irrigation schedule. Evapotranspiration must be determined first, and then fed back into another neural network to predict accumulated irrigation requirements.

Zhang et al. (1996) developed a fuzzy logic based irrigation control system for container production of ornamental plants in greenhouses. They used resistance-type soil moisture sensors as monitoring devices and a sensing mechanism to detect water drainage. The proposed fuzzy irrigation control system was tested successfully both in the laboratory and in the greenhouse for a few months. The ability to stop irrigation in time with the help of fuzzy logic ensured minimum leachate.

Ribeiro (1997) developed a model for evapotranspiration using fuzzy logic. Solar radiation and relative humidity values were used as input data and mapped to evapotranspiration output data obtained from a lysimeter. The fuzzy model performed slightly better when compared to evapotranspiration values estimated by the Penman-Monteith equation.

An automated real-time fuzzy irrigation control system was developed by Ribeiro and Yoder (1997). Watermark sensors and weather sensors were used to monitor soil moisture content and to estimate evapotranspiration as a fuzzy variable. The system worked as expected in response to the climatic and soil moisture variations in a fully automated manner. This study is included in this dissertation.

CHAPTER 3

ARTIFICIAL INTELLIGENCE

The term "artificial intelligence", in its broadest sense, encompasses a number of technologies that includes, but is not limited to, expert systems, fuzzy logic systems, artificial neural networks, genetic algorithms, cellular automata, chaotic systems, anticipatory systems, wavelets, and others. Interestingly, most of these technologies have their origins in biological or behavioral phenomena related to humans or animals, and many of these technologies are simple analogs of human and animal systems. More generally, fuzzy logic, neural networks, and genetic algorithms may be viewed as the principal constituents of what might be called soft computing. Fuzzy logic and neural networks individually have reached a degree of maturity where they are each being applied to real-world situations. Researchers often utilize these two technologies in series, using one as the preprocessor or postprocessor for the other. Examples include the use of fuzzy inputs and outputs for neural networks, and the use of neural networks to quantify the shape of fuzzy membership functions. A specific example of a hybrid system is the Adaptive Network-based Fuzzy Inference System (ANFIS) developed by Jang (1993). This chapter presents the fundamentals of fuzzy logic and neural networks, and relies on works from Simões (1995), Conboy (1996), Tsoukalas and Uhrig (1997), and Hines (1997). These two artificial intelligence techniques were used in this research for estimation of reference evapotranspiration and for automatic control of irrigation systems.

Fuzzy Logic

Fuzzy logic was first introduced by Zadeh (1965), and can be viewed as a generalization of conventional Boolean logic. The mathematical foundations of fuzzy logic rest in fuzzy set theory. In a "crisp", or classical set, an element either belongs or does not belong to the set. A fuzzy set possesses a distinct feature of allowing partial membership, with degrees of membership varying from 0 (non-member) to 1 (full member).

Introduction to Fuzzy Logic and Fuzzy Control

Since fuzzy logic is a relatively new concept in the field of automatic control it is appropriate here to present its fundamental elements. The basic mathematics of fuzzy sets, fuzzy logic, linguistic rules, inference, fuzzification, and deffuzification, are presented.

Definition of a few basic concepts is necessary to facilitate the following discussions. These are working definitions; formal mathematical definitions will follow in the section detailing the mathematics of fuzzy logic.

Definition 1. A *crisp set* is a set to which an object or quantity is either a full member or a non-member.

Definition 2. A *fuzzy set* is a set to which an object or quantity can be either a full member, partial member, or non-member.

Definition 3. A *crisp system* is a system in which all quantities are expressed as numbers. System theory deals with crisp systems.

Definition 4. A *fuzzy system* is a system in which quantities are expressed as fuzzy sets rather than as numbers.

Introduction to Control Using Fuzzy Logic

System Overview

Figure 3.1 presents a simple block diagram illustration of a general control system. The controller takes as inputs a set of command values and a set of process/sensor outputs and performs a nonlinear operation on them to produce a set of control inputs. The control inputs are fed to the plant and the resultant changes in its outputs are fed back to the controller. A steady-state is achieved when the plant outputs reach a state that corresponds to the commanded state.

Fuzzy controllers can operate on either fuzzy or crisp systems. Figure 3.2 illustrates fuzzy control of a fuzzy system in block diagram form. All inputs and outputs are fuzzy sets and all operations internal to the fuzzy controller are very precise "if-then" rules.

Like most systems with which engineers deal, the systems involved in irrigation are non-fuzzy systems. In other words, they supposedly can be described by laws of physics that are precise mathematical concepts.

In order to use fuzzy logic to control a crisp system, or more generally to have any type of hybrid crisp/fuzzy system, processes by which crisp values are transformed to fuzzy sets and fuzzy sets are transformed to crisp values are established.

Definition 5. *Fuzzification* is the process by which crisp values are transformed to the domain of fuzzy sets.

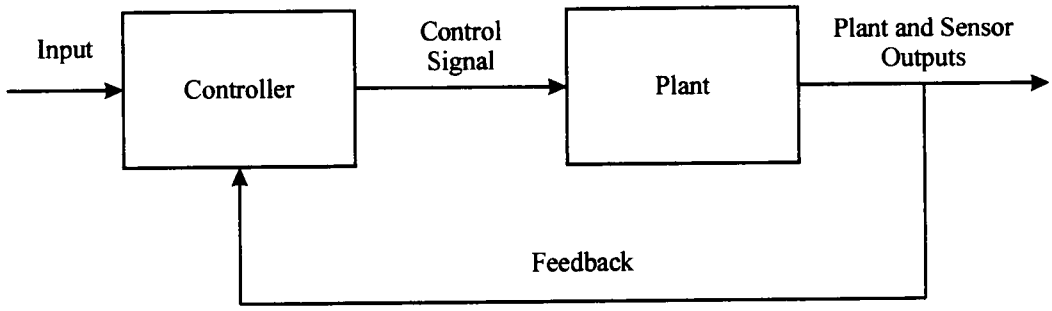


Figure 3.1 A simple block diagram illustration of a general control system.

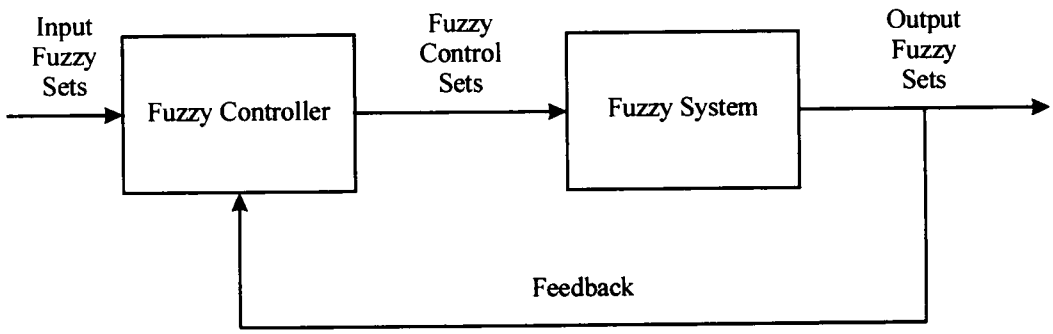


Figure 3.2 A block diagram illustration of an all-fuzzy control system.

Definition 6. *Defuzzification* is the process by which values in the domain of fuzzy sets are transformed to crisp values.

Figure 3.3 is a block diagram representation of a system where fuzzy control is performed on a crisp system. In the literature it is often the convention to include the fuzzifier and defuzzifier functions as part of the fuzzy controller. They are shown as separate blocks in Figure 3.3 for clarity of the illustration, but from this point on they will be considered part of the fuzzy controller.

The Fuzzy Logic Control Concept

There are two types of information sources: sensors that provide numerical measurements of variables, and human experts who provide linguistic instructions and descriptions about a system. Conventional engineering is able to directly incorporate only the numerical information. As a result, only part of the information available to the system designer can be utilized in classical control system design (Wang, 1994).

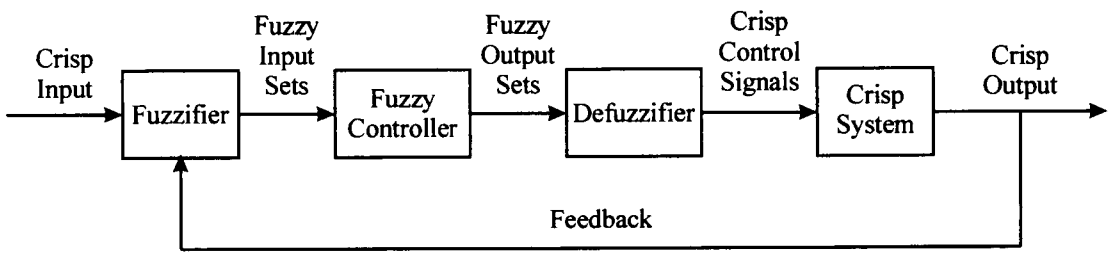


Figure 3.3 A block diagram representation of fuzzy control performed on a crisp system.

Systems that utilize the knowledge of human experts including books, reports, codes, and standards which humans prepare, are called knowledge based systems (KBS). A KBS used for control is referred to as a knowledge based controller (KBC).

Driankov et al. (1993), discuss two uses for a KBC: as a direct expert control system (DECS) and as a supervisory expert control system (SECS). The primary difference is that the DECS performs the control activity itself and the SECS oversees functions such as fault protection and monitoring, modes of operation, and other auxiliary functions. In this study only fuzzy logic control as an example of DECS will be considered. When fuzzy concepts and logic are used by a KBC, it is referred to as a fuzzy knowledge based controller (FKBC) or a fuzzy logic controller (FLC), as defined by Lee (1990). Definition 7 will serve as a working definition for a FLC.

> **Definition 7.** A *fuzzy logic controller* (FLC) is a real time expert system that uses the principles of fuzzy logic to implement part of a human operator's or design engineer's knowledge that is not readily realizable in conventional control parameters, but rather is suited for a set of situation/action pairs.

Components of Fuzzy Control

Figure 3.4 shows the basic structure of a FLC in a block diagram. It differs from Figure 3.3 in that the fuzzification and defuzzification functions have been absorbed into the controller, and the level of detail within the fuzzy controller block is increased. The role of the expert knowledge base as input to all of the fuzzy-domain function blocks is emphasized since there exist no widely accepted procedures for their design, that are historically based on expert intuition rather than an algorithmic formula. Also, the functions are highly interrelated, which means an expertise-motivated choice in one area will affect the configuration of other areas as well.

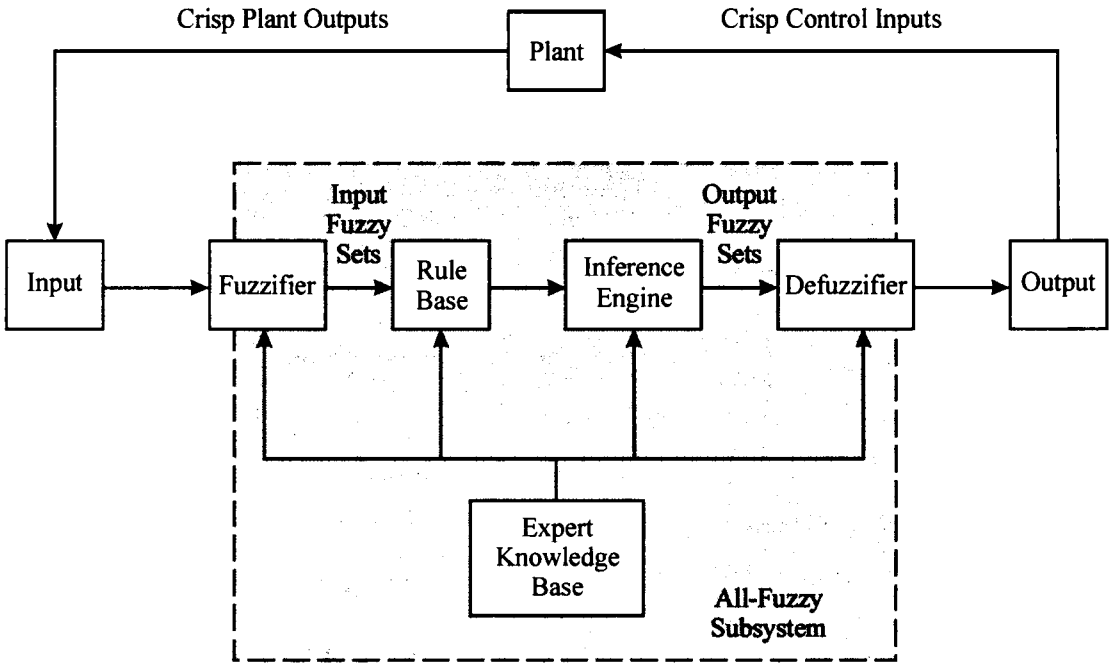


Figure 3.4 Basic structure of a fuzzy logic controller performed on a crisp system.

The Expert Knowledge Base

Human knowledge is most often expressed with language, or in a more scientific context, with *linguistic terms*. Suppose it is desired to control a robotic arm so that it can throw a basketball to a human (at a variable distance from the arm) such that the ball can be caught. The task would be to implement a control system to refine the throwing action so that the ball is catchable. Assuming the designer has the skill himself he might start by listing a set of guidelines based on his own experience:

1. *If the person is far away, the ball should be thrown hard.*
2. *If the person is very close, the ball should be thrown softly.*
3. *If the person is moving the ball must be thrown ahead of him, etc.*

If the designer were to continue until everything he knew about throwing a basketball to another person was expressed in this manner, the list would essentially constitute an expert knowledge base.

It should also be noted that personal experience is not necessarily required for the designer. It is possible the designer could acquire enough information from experts, or by observation of skilled workers, to put together an expert knowledge base. This is often the case in areas such as industrial process control where control engineers rely on experienced operators to formulate the expert knowledge base.

Rule Base

The most direct manifestation of the expert knowledge base in a FLC occurs in the rule base. It is essentially a translation of the expert knowledge base to linguistic mathematical terms of fuzzy logic. A working definition is given in Definition 8.

Definition 8. A *rule base* is a set of situation-action pairs expressed with the linguistic concepts of fuzzy logic that, taken as a whole, describe a policy to perform a task or set of tasks.

Inference Engine

The inference engine is a processor. Recall from Definition 2 that a fuzzy set is one to which an object or value may *partially* belong. In conventional logic a situation-action pair is processed as "if the condition is true then perform the action; if the condition is false then do not perform the action." In a sense this could be called a boolean inference engine, and the logic is well-suited for many applications. In a fuzzy system it is generally the case where one or more of the conditions are partially true. It is the job of the fuzzy inference engine to assemble the set of partially true conditions and construct an action policy in the form of fuzzy output sets.

Introduction to the Mathematics of Fuzzy Logic

The Fuzzy Concept

Humans are able to execute a wide variety of decisions and actions without need of numerical data. Classical control systems on the other hand cannot do anything without some form of numerical manipulation, whether it be in the form of digital computations or the natural differential equations governing analog electrical and mechanical systems. When describing action policies, humans normally use imprecise linguistic terms rather than numerical values. The following set of three linguistic rules outline a policy for decision-making typical of humans.

If it is *warm outside* then wear *no coat* else (R3.1)

If it is *mild outside* then wear a *light jacket* else (R3.2)

If it is *cold outside* then wear a *heavy coat* else (R3.3)

If it is *very cold outside* then wear a *heavy coat* else (R3.4)

If it is *extremely cold outside* then wear a *heavy coat* (R3.5)

Suppose a linguistic scale has been defined for the variable *outside temperature* with values on the scale defined by the term set (*extremely cold, very cold, cold, mild, warm*). Suppose further that each term has been assigned a set (range) of temperature values. The classical logic approach would be to say that if the temperature falls into the range of term X then the statement "the temperature is X" is true. If the temperature does not fall in the range then the statement is false. Say the temperature on a given day is 9.5 °C. Table 3.1 gives ranges for the term set of variable *outside temperature*, tests the statement "the temperature is X" and gives the corresponding boolean logic value. Notice that statement "T = 9.5 °C is X" either applies to a linguistic descriptor or it does not.

Using rules (R3.1) through (R3.5) for this temperature the heavy coat would be selected. Notice however, that if the temperature was only 0.51 °C higher, the term *mild* would apply. As a result a light jacket would be selected, yet the difference of 0.51 °C is probably too small for most people to discern. In other words, if a heavy coat is required for comfort at 9.5 °C then it is unlikely that a light jacket will be comfortable at 10.1 °C, or vice-versa.

It should be obvious that the type of logic described in Table 3.1 is an ineffective mathematical tool for global implementation of human decision-making, even though it is effective for "go-no go" types of decisions. Zadeh (1965) recognized that in the human

Table 3.1 Term set that describes the linguistic variable *outdoor temperature* (T).

Term (X)	Temperature Range	T = 9.5 °C is X	Boolean Logic Value
Extremely Cold	T <= -10 °C	False	0
Very Cold	-10 °C < T <= 0 °C	False	0
Cold	0 °C < T <= 10 °C	True	1
Mild	10 °C < T <= 20 °C	False	0
Warm	T > 20 °C	False	0

thought process a given quantity can apply to more than one descriptor with varying degrees of truth, and that the quantity need not be numeric in order for successful decisions to be made. To represent the idea mathematically he invented the notion of fuzzy sets and fuzzy logic.

Universe of Discourse

The universe of discourse U is a set of objects or values u_i associated with the outcome of some event, such as a signal measurement. In the coat-choice example the universe of discourse *outside temperature* is a range of temperatures from $-\infty$ to ∞ as the second column of Table 3.1 indicates.

Membership Functions

A set, whether crisp or fuzzy, is characterized by a membership function, *M(u),* that maps each element u_i of U to a number on the interval [0, 1].

$$M(u): U \rightarrow [0, 1]$$

Crisp membership functions (also called *characteristic* functions) are usually denoted as $\chi(u)$, and fuzzy membership functions as $\mu(u)$. Characteristic functions have the restriction that χ takes on only the values 0 or 1. In other words, $u \in U$ is either fully a member of crisp set C ($\chi_c(u) = 1$), or it is totally a non-member ($\chi_c(u) = 0$).

Fuzzy membership functions can take on any value in the interval [0,1]. When defined to be piece-wise continuous and convex, fuzzy membership functions are characterized by gradual transition from membership to non-membership. In fuzzy systems, as in crisp systems, each value (fuzzy set) on U is labeled with a term. In keeping with the spirit of fuzzy logic the terms are normally linguistic in nature, although numerics can be used. The same term set (*extremely cold, very cold. cold, mild. warm*) could be used in a fuzzy analogy to Table 3.1.

Figure 3.5 depicts a crisp characteristic function, a fuzzy membership function, and graphical representations of some of the terminology associated with fuzzy sets. The *support* of a membership function defined on U is the set of all elements of U with non-zero membership value. The *core* of the membership function is the set of elements that have full membership ($\mu(u) = 1$). The *boundaries* of μ are the segments with non-zero membership less than full membership ($0 < \mu < 1$). *Crossover points* are the points where $\mu = 0.5$. A *convex* membership function is one that is either monotonically increasing, monotonically decreasing, or can be divided into two regions, the first of which is

monotonically increasing and the second monotonically decreasing for an increasing U. There is no requirement that a fuzzy set be convex, nor is the trapezoidal shape of Figure 3.5 required. A crisp characteristic function is a special case of a fuzzy membership function (as a crisp set is of a fuzzy set).

Table 3.1 effectively defines a set of characteristic functions for the variable outdoor temperature. These are depicted in Figure 3.6. Figure 3.7 illustrates a set of fuzzy membership functions defined on the same universe of discourse.

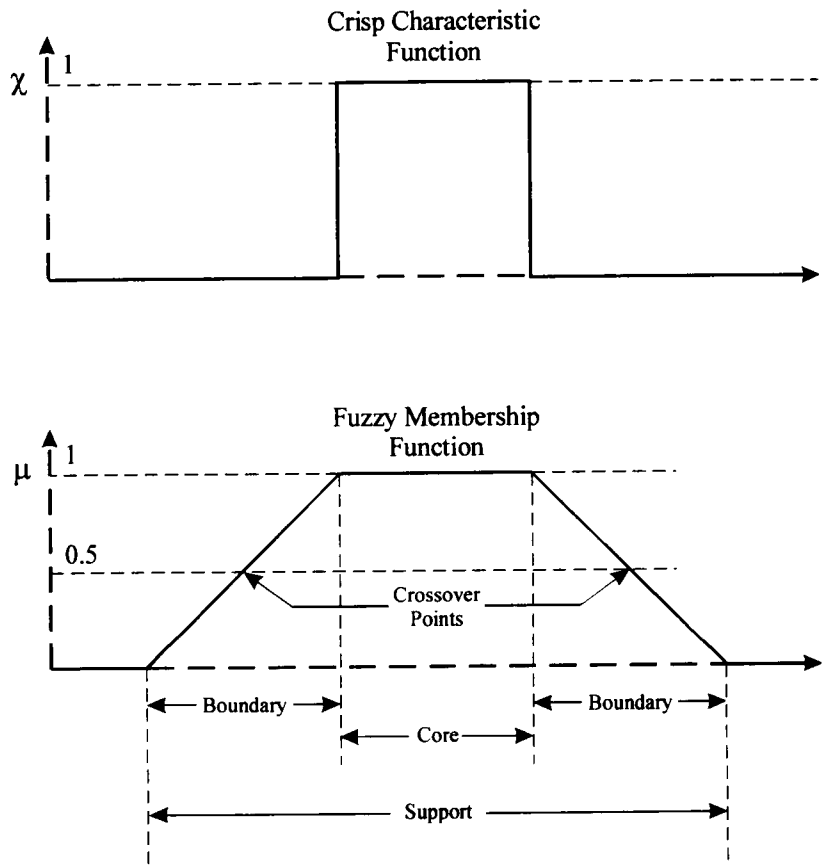


Figure 3.5 A crisp characteristic function compared to a fuzzy membership function.

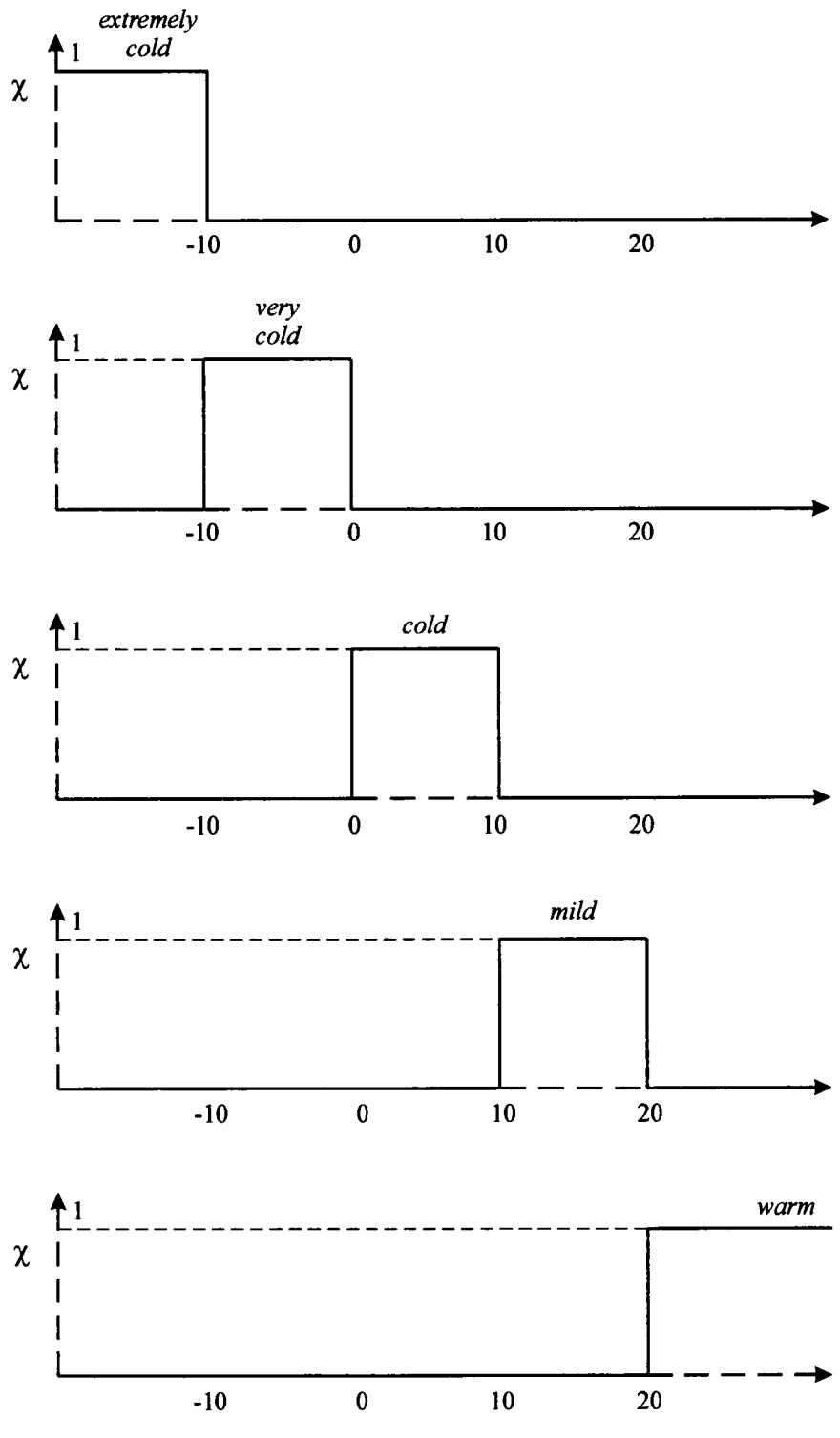


Figure 3.6 Characteristic functions of the crisp sets defined for the variable *outdoor temperature*.

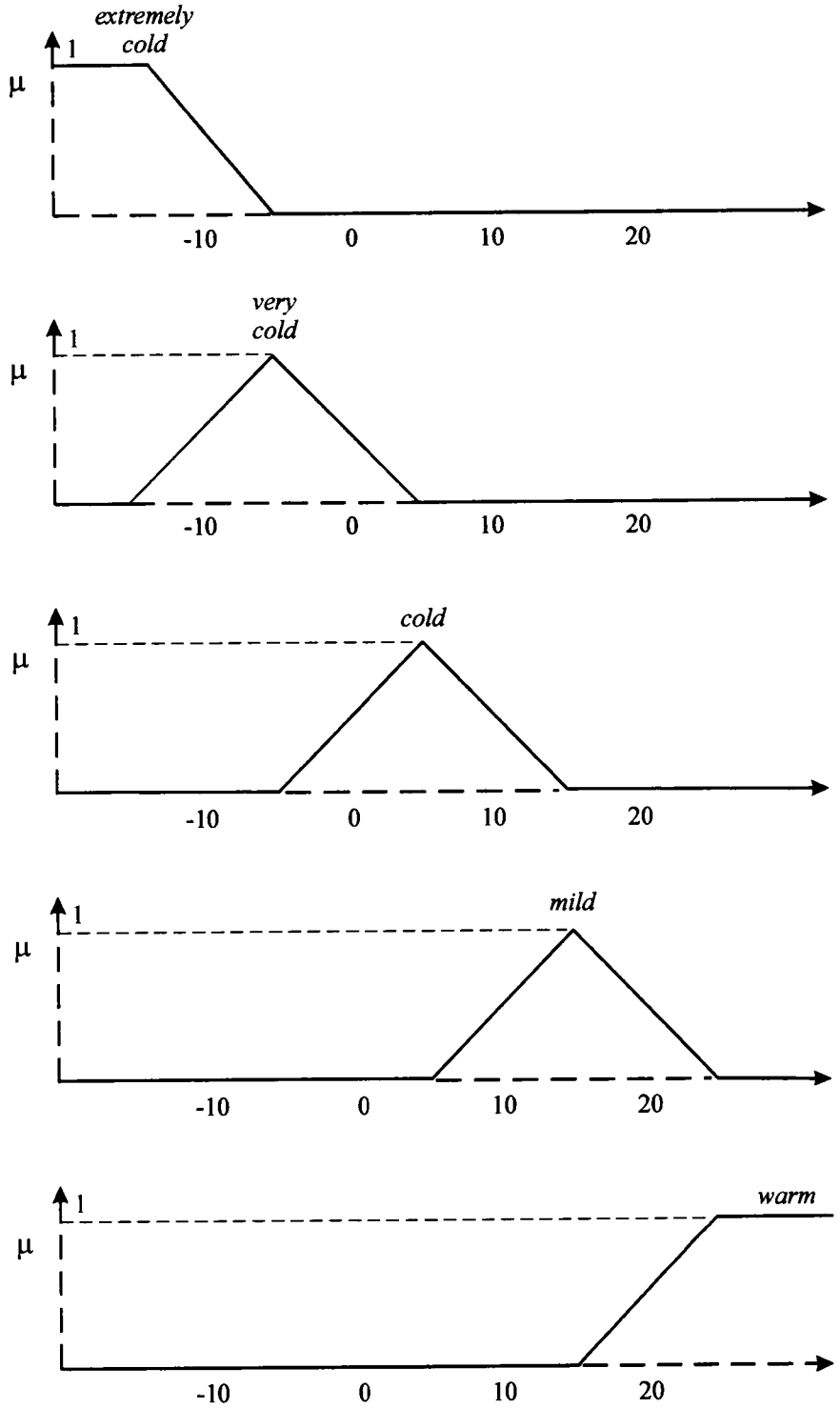


Figure 3.7 Fuzzy membership functions for the term set defined for the variable *outdoor temperature*.

There are numerous policies for defining membership functions on a universe of discourse. The most common is the use of intuition which motivated Figure 3.4 to be drawn showing the expert knowledge base as an input to the fuzzifier block. Another method is to use inference techniques on previously defined membership functions; for example if a membership for *large* is defined on some U then a membership function for *not large* can be defined using a logic complement operator. Interestingly, neural networks have been shown to be effective for implementing fuzzy functions and for deriving membership functions; an example of which is given in Ross (1995).

In control, most membership functions are normal (max $\{\mu\} = 1$) and convex. Aside from those two conventions all other parameters (size of support, size of core, shape of boundaries, degree of overlap with neighboring functions, etc.) are left to the discretion of the designer and there exist virtually no solid axiomatic guidelines on which to base the choices (Runkler and Glesner, 1993).

Fuzzification

Fuzzification was defined by Definition 5 as the process of converting crisp values to the domain of fuzzy sets. Now that membership functions have been introduced, the process can be further refined. To fuzzify a crisp value x on a universe of discourse U, where N membership functions $\{\mu_1, \mu_2, \ldots, \mu_N\}$ are defined, simply determine the membership function values of the N fuzzy sets at the value x $\{\mu_1(x), \mu_2(x), \ldots, \mu_N(x)\}$. Once these values are determined, x is said to be fuzzified.

Fuzzy Set Operations

Consider two universes of discourse A and B. Assume there are N fuzzy sets defined by membership functions $\{\mu_{A1}, \mu_{A2}, \ldots, \mu_{AN}\}$ on A and M fuzzy sets likewise described by $\{\mu_{B1}, \mu_{B2}, \ldots, \mu_{BM}\}$ defined on B. If an event occurs such that the outcome on A is the crisp value a and the outcome on B is the crisp value b, membership values can be determined on the Cartesian space $C = A \times B$ for relations defined by a number of set operators.

If R is defined as the event *a is A1 and b is B1*, $\mu_R(a,b)$ can be evaluated as

$$\mu_R(a,b) = \mu_{A1 \wedge B1}(a,b) = \min\{\mu_{A1}(a), \mu_{B1}(b)\} \tag{3.1}$$

or

$$\mu_R(a,b) = \mu_{A1 \wedge B1}(a,b) = \mu_{A1}(a) \cdot \mu_{B1}(b) \tag{3.2}$$

If R is defined as the event *a is A1 or b is B1*, $\mu_R(a,b)$ can be evaluated as

$$\mu_R(a,b) = \mu_{A1 \vee B1}(a,b) = \max\{\mu_{A1}(a), \mu_{B1}(b)\} \tag{3.3}$$

The event *a is not A1*, *A1'*, has a membership function which is the *complement* of $\mu_{A1}(a)$:

$$\mu_{A1'}(a) = 1 - \mu_{A1}(a) \tag{3.4}$$

Hedges are modifiers such as "*very*" or "*more-or-less*", and "*not*". The event that *a is very A1* has membership function $\mu_{A1}(a)^2$, and *a is more-or-less A1* has membership function $\mu_{A1}(a)^{1/2}$.

The operations described above represent only a small sample of fuzzy operations that have been proposed. The numerical evaluation methods described for the operations represent only those most commonly encountered in FLCs. Fuzzy *and* (union) and *or* (intersection) operations can be implemented by any function that meets the criterion of a

T-norm or a T-conorm (sometimes called S-norm as in Driankov et al., 1993), respectively. A T-norm defines a mapping from [0,1] x [0,1] → [0,1] that satisfies the following criteria

$$\left.\begin{array}{l} i)\ a*b = b*a \\ ii)\ (a*b)*c = a*(b*c) \\ iii)\ a \le c \text{ and } b \le d \rightarrow\ a*b \le c*d \\ iv)\ a*1 = a \end{array}\right\} \qquad (3.5)$$

where * denotes an operator. For an S-norm, conditions *i*) through *iii*) of (3.5) apply with *iv*) replaced by

$$iv)\ a*0 = a \qquad (3.6)$$

In the case of the S-norm these include fuzzy intersection (max-operator), algebraic sum, bounded sum, and drastic sum, among others (Wang, 1994).

Evaluation of Linguistic Rules and Inference Techniques

Consider three universes of discourses *A*, *B*, and *Z* with two fuzzy sets *A1* and *A2* defined on *A*, two fuzzy sets *B1* and *B2* defined on *B*, and two fuzzy sets *Z1* and *Z2* defined on *Z*. The universes of discourse are associated with crisp variables *a*, *b*, and *z*, respectively. *A* and *B* are defined as input universes of discourse and *Z* is defined as an output universe of discourse. It is desired to implement a system defined by the following rule set.

If *a* is *A1* and *b* is *B1* then z is *Z1* (R3.6)

If *a* is *A2* and b is *B2* then z is *Z2* (R3.7)

The clause associated with the if-portion of the rule is called the *antecedent*, and the clause associated with the then-portion is the *consequent*. In other words, the rule is of the form: *if-antecedent, then-consequent*. This type of if-then statement is the most commonly encountered in FLCs.

When an event occurs, the antecedents of the rule base can be evaluated. The way in which the resulting set of antecedent values affect the consequent is called *implication*. There are a number of implication techniques discussed in the literature; however, there is one used by Mamdani (1974), that exceeds the others in terms of importance in control applications. It is sometimes called *max-min* implication, or Mamdani clipping . If an event occurs such that the value of a is x and the value of b is y the implication can be expressed for the above rule set as

$$\mu_z(z) = [\alpha_1 \wedge \mu_{z1}(z)] \wedge [\alpha_2 \wedge \mu_{z2}(z)] \tag{3.7}$$

where $\alpha_1 = \mu_{A1}(x) \wedge \mu_{B1}(y)$ and $\alpha_2 = \mu_{A2}(x) \wedge \mu_{B2}(y)$ and the $\min\{\cdot\}$ and $\max\{\cdot\}$ operators are respectively used for the $\wedge$ and $\vee$ operations (Lee, 1990). This technique is more clearly illustrated using graphical techniques as shown in Figure 3.8. A variation attributed to Larson uses the algebraic product operator for $\wedge$ operations which is known as *max-product* implication. Another implication strategy is the *sum-product* method which is essentially the max-product method with the arithmetic sum operator substituted for the $\vee$ operation.

Another type of inference proposed by Takagi and Sugeno (1985), is given by

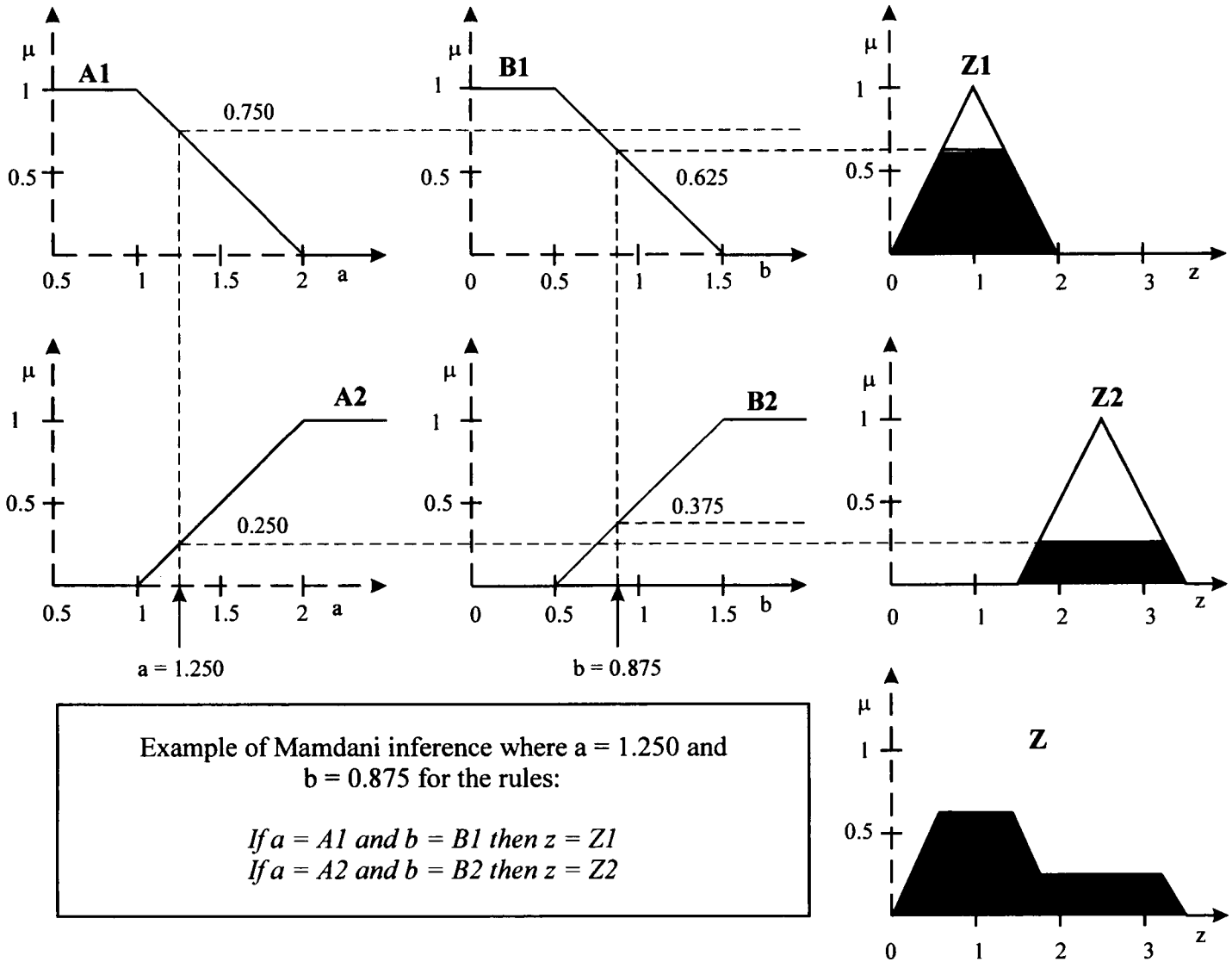


Figure 3.8 Example of Mamdani's inference using two inputs and two rules.

$$z^* = \frac{\sum_{i=1}^{N}\sum_{j=1}^{M} z_{ij}(a,b)\cdot \mu_{ij}(a,b)}{\sum_{i=1}^{N}\sum_{j=1}^{M} \mu_{ij}(a,b)} \tag{3.8}$$

where there are N fuzzy sets A_i defined on A and M fuzzy sets B_j on B and N x M rules of the form

$$\text{If } a \text{ is } A_i \text{ and } b \text{ is } B_j \text{ then } z \text{ is } z_{ij}(a,b) \tag{R3.8}$$

where $\mu_{ij}(a,b)$ is given by

$$\mu_{ij}(a,b) = \mu_{Ai}(a)\cdot\mu_{Bj}(b) \tag{3.9}$$

and

$$z_{ij}(a,b) = c_0^{ij} + c_1^{ij}\cdot a + c_2^{ij}\cdot b \tag{3.10}$$

where c_0^{ij}, c_1^{ij}, and c_2^{ij} are constants chosen by the designer. For this type of inference the output is crisp, so no defuzzification is required. Here the fuzzy *and*-operator is evaluated with the product and, in a sense, the *or*-operator is evaluated with the weighted sum of (3.8).

Defuzzification

The process of defuzzification is significantly more complex than that of fuzzification. The task is to convert a generally subnormal, nonconvex fuzzy set into a crisp value. A fuzzy set is said to be subnormal if the maximum value of its membership function is less than 1.

Defuzzification techniques can be grouped into two general classes. The first class defuzzify based on the area beneath the membership function, and will be called *area-based defuzzification* for the purposes of this study. The second class defuzzify based on the maximum, or maxima, of the membership function. These techniques will be referred to as *height-based defuzzification.*

The area based defuzzification method most often used is the *centroid*, or *center-of-gravity* method. If the crisp output of the defuzzification process is denoted z*, the technique can be expressed as

$$z^* = \frac{\int z \cdot \mu_z(z) dz}{\int \mu_z(z) dz} \tag{3.11}$$

This technique can experience problems with nonconvex fuzzy sets. An alternate technique called the *center of largest area* technique divides the nonconvex set into convex subsets and performs the centroid calculation on the subset with the largest area.

The most basic height-based defuzzification technique, called *maximum membership* or *max criteria* finds the maximum value of μ_z and assigns the corresponding value of z to z^*. This works only when the maximum is a peak. If the maximum is a plateau the corresponding value of z is not unique. Methods such as the *first-of-maxima*, *last-of-maxima*, or *mean-of-maxima* can then be used. The first-of-maxima method chooses as z^* the first z corresponding to the maximum moving along the z-axis. The last-of-maximum chooses the last z corresponding to a maxima. If the plateau begins at z_1 and ends at z_2 the mean of maxima method computes z^* as

$$z^* = (z_1 + z_2)/2 \tag{3.12}$$

The above techniques assume an output membership function U based on a logical union of n individual output fuzzy sets U_k. If each U_k is symmetric about a midpoint z_k, a technique called the weighted average method can be used. In this technique the midpoints are weighted by the value of the membership function at the midpoint.

$$z^* = \frac{\sum_{k=1}^{n} z_k \cdot \mu_{z_k}(z_k)}{\sum_{k=1}^{n} \mu_{z_k}(z_k)} \tag{3.13}$$

A similar technique known as the *center-of-sums* weights the midpoints by the area of the membership functions. This technique can be applied to asymmetric U_k by substituting the peak, mean-of-maxima, etc., for the midpoint.

Neural Networks

The field of neural networks has a history of some decades but has found solid application only in the past 15 years. The availability of computers with the power to perform simulations and the development of specialized hardware to implement neural networks have helped expand the interest and research. Neural network technology is being applied to solve a wide variety of scientific, engineering, and business problems and to perform complex functions such as noise cancellation, adaptive filtering, pattern

recognition, non-linear controls, and econometric forecasting. There are three main characteristics that make neural networks so valuable:

1. They can learn relationships between input and output data. Such learning does not depend on the programmer's prior knowledge of rules. They have the ability to infer solutions from the presented data, often capturing subtle relationships.

2. Neural networks can generalize and handle noisy, imperfect, or incomplete data. Such generalization features provide a measure of fault tolerance and are useful when examining real-world data.

3. They can capture complex, higher-order-functions, and non-linear interactions among the input variables in a system.

The development of neural networks was inspired by studies for understanding the biological nervous system. The first relevant reference are the physiology and psychology studies related to the structure and function of the brain, published by William James in 1890. His concepts foretold the notions of a neuron's activity as being a function of the sum of its inputs. Half a century later, McCulloch and Pitts (1943), published a seminal paper in which they derived theorems related to models of neuronal systems based on what was known about biological structures in the early 40s, showing that a network could represent any finite logical expression with a massively parallel architecture. In 1949, Donald O. Hebb published a book in which he defined a method to update synaptic weights that now is referred to as Hebbian learning. The landmark paper by Frank Rosenblat (1958) defined a neural network structure called perceptron. It was simulated in detail on an IBM 704 computer at the Cornell Aeronautical Laboratory and caught the attention of engineers and physicists because the paper described the perceptron as a

"learning machine." This paper laid the groundwork for both supervised and unsupervised training algorithms as they are today in back-propagation and Kohonen networks, respectively. Widrow and Hoff (1960) published a paper in which they had simulated neural networks in computers, and had also implemented their designs in hardware. They introduced a device called an *adaline* (for adaptive linear). An *adaline* consists of a single neurode with an arbitrary number of input elements that can take on values of plus or minus one and a bias element. Before being summed by the neurode summer, each input, including the bias, is modified by a gain. The Widrow-Hoff algorithm is a form of supervised learning that adjusts the weights according to the size of the error on the output of the summer. They showed that their technique for adjusting the weights could minimize the sum-squared error over all patterns in the training set.

The development of neural networks slowed at the end of the 1960s and middle 1970s, mainly after the release of a book called *Perceptrons* (Minsky and Papert, 1969). They had a comprehensive analysis on single perceptrons, without hidden layers. They showed that the two-layer perceptron was rather limited because it could only work problems with linear separable solution space. The exclusive-or (XOR) problem was used on an elementary system that the perceptron was unable to solve.

The revival in the field came with the research conducted by Hopfield (1982), leading to today's Hopfield network. However, what primarily influenced people about the capabilities of neural networks was the discovery of the backpropagation algorithm. It was first found by Werbos (1974), and Parker (1982), and independently rediscovered and popularized by Rumelhart et al., (1986). Backpropagation is a gradient descent system that tries to minimize a cost function (mean squared error) by moving down the

gradient of the error curve. This algorithm updates the weights by calculating local errors for each neuron, allowing the use of hidden layers in the neural network topology.

Backpropagation Algorithm

Despite the many descriptive names applied to the various paradigms, all neural networks perform the same fundamental function: they map one vector space onto another. An input vector is applied to the network, and in response, the network produces an output vector. Each vector consists of one or more components, each of which represents the value of some variable. Mapping may be static, in which case a feedforward neural network will suffice. Or mapping may be dynamic, involving previous network states; in this case a feedforward network may use delayed inputs or a network with feedback.

A feedforward neural network topology has one input layer, one or more hidden layers, and one output layer. The utilization of the hidden layer allows the network to develop any kind of mapping, not just linearly separable ones. A backpropagation network operates in two steps during training. First, an input pattern is presented to the network's input layer. The resulting activity flows through the network from layer to layer until the output is generated. Next, the network output is compared to the desired output for that particular input pattern. The error is passed backward through the network from the output layer back to the input layer, with the weights being modified as the error backpropagates. A typical neuron has several inputs (x_j) which are multiplied by the correspondent weights (w_i) as shown in Figure 3.9.

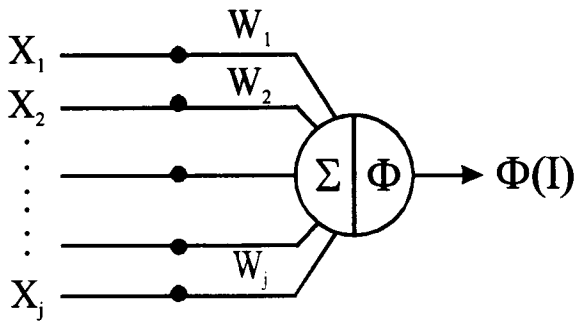


Figure 3.9 Structure of a typical neuron element with several inputs and correspondent weights.

Each neuron computes the weighted input as:

$$I = \sum_{j=1}^{n} w_i x_j \tag{3.14}$$

This summation is passed through the activation function, also called a "squashing" function. A typical activation function is usually smooth and non-linear, like the sigmoidal function given by (3.15) and depicted in the Figure 3.10.

$$\phi(I) = \frac{1}{1+e^{-1}} \tag{3.15}$$

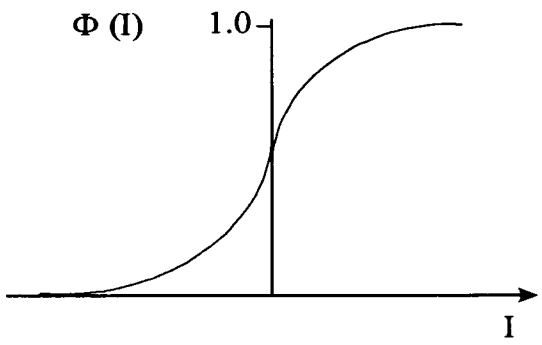


Figure 3.10 A sigmoidal function is a typical smooth and non-linear activation function.

This function has the convenient property that its derivative is a relationship of the same function:

$$\frac{d\phi(I)}{dI} = \phi(I)[1 - \phi(I)] \tag{3.16}$$

The weight change law is given by the generalized delta rule, in which the change in a given connection weight is proportional to a learning coefficient (β) and to the partial derivative of the error function (E) with respect to the weight that is being modified:

$$\Delta w_{ij} = -\beta \frac{\partial E}{\partial w_{ij}} \tag{3.17}$$

For the hidden-to-output connections the above gradient descent rule gives:

$$\Delta w_j = \beta E_j \phi(I) = \beta(y_j^{desired} - y_j^{actual})\phi(I) \tag{3.18}$$

For the output layer the error that causes the weight change is evident, since it is the difference between the network output and the desired one. But in the middle layer, a more complex procedure is followed. For the input-to-hidden connections it is necessary to differentiate with respect to the weights by applying the chain rule:

$$\Delta w_{jk} = \beta \phi_k(I)[I - \phi_k(I)] \sum_{k=1}^{n} w_{jk} E_k^{output} \phi_j(I) \tag{3.19}$$

For high-order error surfaces, the gradient descent method can be very slow if the learning coefficient β is small, and can oscillate widely if β is too large; the addition of a momentum term can overcome such problems. It gives each connection some inertia, so that it tends to change in the direction of the average "downhill force," this scheme is implemented by giving a contribution from the previous weight change to each new weight revision:

$$\Delta w_{ij} = -\beta \frac{\partial E}{\partial w_{ij}} + \alpha(\Delta w_{ij})_{previous} \tag{3.20}$$

An Example of Backprogagation Training

The procedure for the backpropagation algorithm is worked here as an example. Suppose there is a fully connected feedforward neural network. The input "i" has two neurons, the middle layer "j" has two neurons, and the output layer "k" has one neuron. Assume the learning coefficient β is 0.7 and the network of Figure 3.11 has all neurons with sigmoidal functions with unity slope gain. The initial weight values are randomized as indicated in Table 3.2.

If the actual output is 0.7 and the target is 0.45, the weight update for the output layer in this pass will be:

$\Delta w_{01} = (0.7)(0.475)(0.45)(1 - 0.45)(0.7 - 0.45) = 0.02057$

$w_{01}^{new} = -0.5 + 0.02057 = -0.479$

$\Delta w_{02} = (0.7)(0.574)(0.45)(1 - 0.45)(0.7 - 0.45) = 0.024861$

$w_{02}^{new} = 0.2 + 0.024861 = 0.2248$

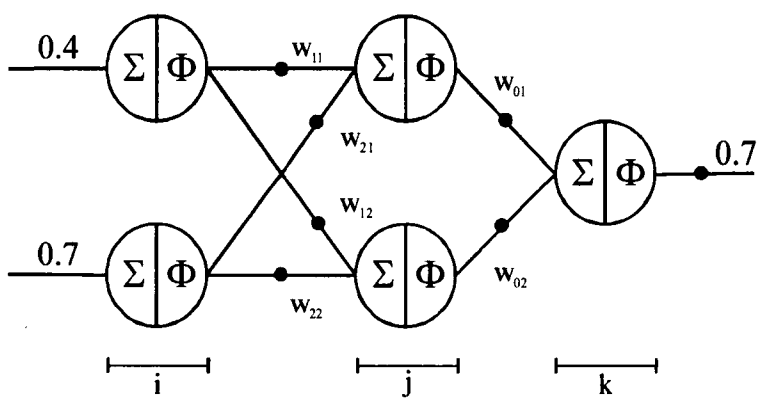


Figure 3.11 Example of a neural network type 2-2-1 before backpropagation algorithm.

Table 3.2 Example of initial weights for training a neural network.

$w_{11} = 0.1$	
$w_{21} = -0.2$	$w_{01} = 0.2$
$w_{12} = 0.4$	$w_{02} = -0.5$
$w_{22} = 0.2$	

For the weight change in the hidden layer it is necessary to compute

$$\sum_{k=1}^{n} w_{jk} E_k^{output}:$$

$\Sigma wE = (-0.5)(0.25) + (0.2)(0.25) = -0.075$

Thus,

$\Delta w_{11} = (0.7)(0.4)(0.475)(1 - 0.475)(-0.075) = -0.0052$

$w_{11} = 0.1\text{-}0.0052 = 0.095$

$\Delta w_{12} = (0.7)(0.4)(0.574)(I - 0.574)(-0.075) = -0.0051$

$w_{12} = 0.4 - 0.0052 = 0.395$

$\Delta w_{21} = (0.7)(0.7)(0.475)(1 - 0.475)(-0.075) = -0.009$

$w_{21} = -0.2 - 0.009 = -0.209$

$\Delta w_{22} = (0.7)(0.7)(0.574)(1 - 0.574)(-0.075) = -0.0089$

$w_{22} = 0.2 - 0.0089 = 0.191$

And the new configuration for the weights is given in Figure 3.12.

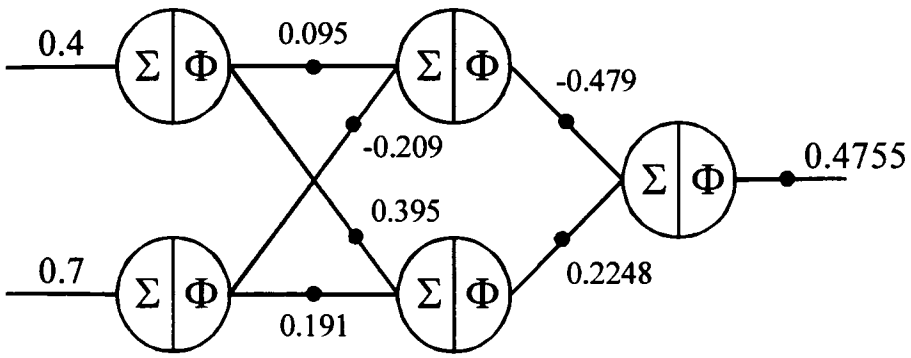


Figure 3.12 Neural network type 2-2-1 after backpropagation algorithm.

With this single pass the output has come closer to the target value (0.45). A few more training steps and the output will eventually converge to a prescribed error threshold. Of course the above training example can be further enhanced with the utilization of a bias node and a momentum term.

Data Processing for Neural Networks

A successful neural network application requires appropriate training data. The training set must span the total range of input patterns sufficiently well such that the trained network can generalize about the data and not memorize the presented patterns. The first step in the data selection is to determine which variables are used most in practice and how they are used. Combining and preprocessing data to make it more meaningful is extremely beneficial. For example, in many applications, the power spectrum density

functions are more useful as inputs to neural networks than are time series data from sampled time records.

After the data are collected, they must be examined to identify any characteristics that may indicate more complex relationships. Calculation of the correlation coefficient for two variables can give an indication of the strength of the relationship between the two. If two input variables are highly correlated it is possible to reduce the size of the model. Anomalous outliers can be disruptive in the model development; an outlier is an extreme data point which may be the result of erroneous data entry. A graph of the data is helpful in visualizing trends in data and for identifying outliers. However, outliers are not always errors, they may indicate some sort of anomaly that must be further investigated.

How data are represented and translated is important to help the network grasp the problem. Data can be continuous-valued or binary. For example, room temperature could be an input varying from 8 °C to 32 °C or it could be represented by five binary inputs: cold, cool, comfortable, warm, and hot. Such inputs could be further improved with fuzzy representations of the above temperature definitions. Another important issue is whether to use actual input amounts or changes in input amounts. In financial forecasting for example, the amount of money in a banking certificate may be less important than the variation of this amount for a certain period. Another valuable data processing step that can help the training of the network is the utilization of input ratios, especially when it is known first hand that they can influence the output. Although the network can learn to do the division by itself by effectively converting the numerator and denominator of each to logs, it is more productive to carry out the division ahead of time and let the neural network concentrate on establishing relationships from the data. Neural networks are very

sensitive to absolute scale in the values. If one input is much bigger than other, the network can erroneously assume a higher importance for such a variable. In addition to this magnitude sensitivity, the input data must correspond to the range of the activation function (from 0 to 1, or from -1 to +1) to avoid network paralysis in the training process that occurs when weights become very large, thereby forcing the neurons to operate in a region where the activation function is very flat, i.e., its derivative is very small. Since the error sent back for training in backpropagation is proportional to the derivative, very little training would take place. To avoid such problems, the numeric input data must be normalized by simply dividing all values of a set by the maximum value so that the input data is on a per-unit basis, and scaled so that the minimum and maximum values are within the linear range of the activation function.

The utilization of commercial software can help the development and training of neural networks. Figure 3.13 shows a block diagram with the necessary steps to train and develop a neural network with simulator software. In some special applications, the designer may want to have control of every detail in the training algorithm. In such cases, it is better to write the entire training code by using a high level computer language.

For processes that may have a mathematical model, a simulation study and analysis can generate the training data for a neural network. Such simulation must be further validated with actual implementation and retrained if necessary, to attain all the real process features. The initial setup of the network topology is dependent on the modeler's experience. It is best to start with a small number of nodes in the hidden layer and to gradually increase the hidden layer size with trial and error.

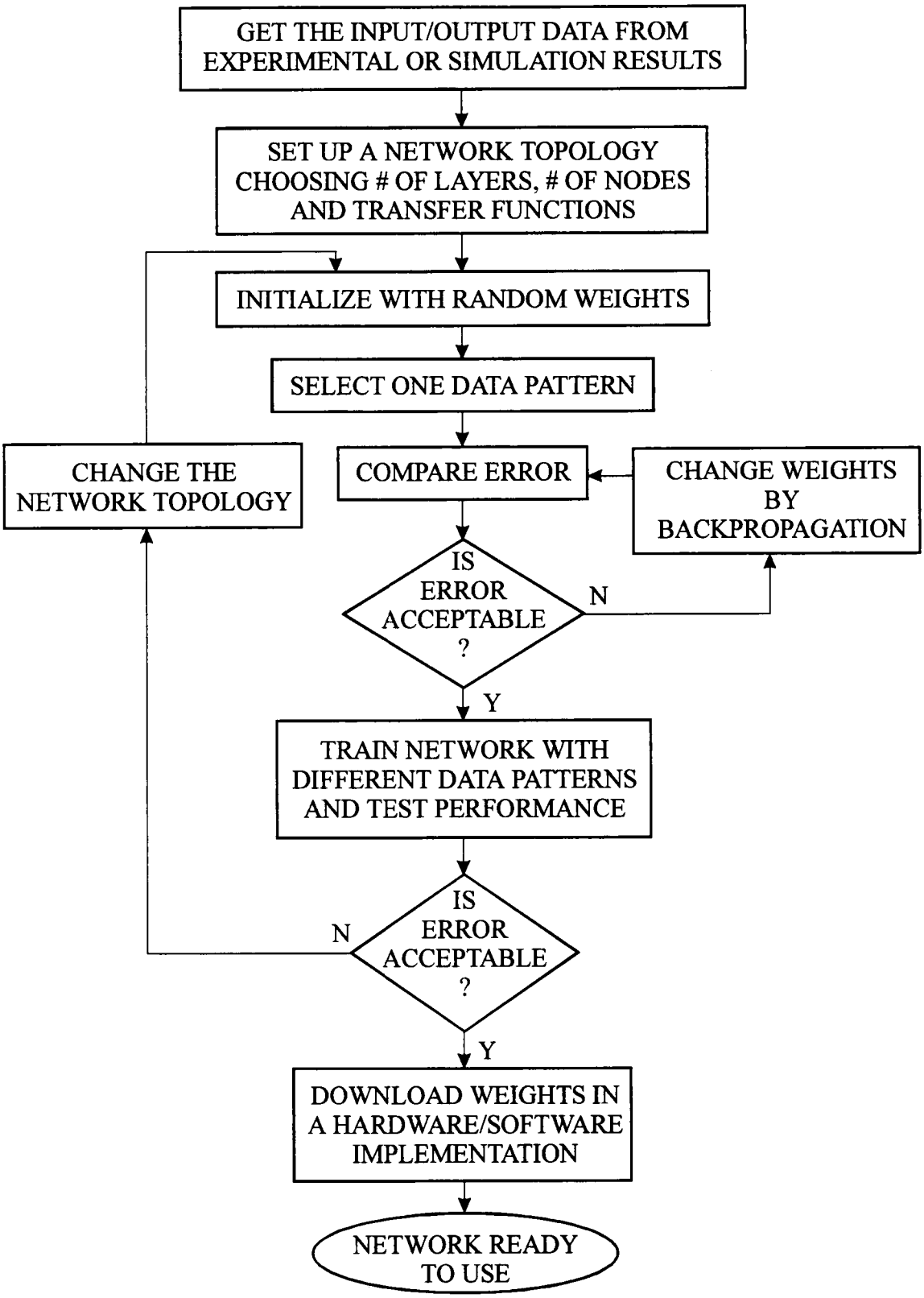


Figure 3.13 Flow chart for a neural network development with backpropagation.

After the network has been trained, it is important to test it against the training set and against examples that the network has never met before. Increasing the size of the hidden layer usually improves the network's accuracy on the training set, but decreasing the size of the hidden layer generally improves generalization, and hence the performance on new cases.

With an iterative procedure such as the backpropagation algorithm, a question that arises is how and when to stop the iteration. When the performance index (global error) has been reduced to a value close to zero or some threshold value, the solution has been found. However, during the development phase the error will never be small and one approach used to determine when to stop is the checking for minimization in error in a test set.

Neural Methods in Fuzzy Systems

The theory of fuzzy logic provides the mathematical strength to capture the uncertainties associated with human cognitive processes, such as thinking and reasoning, by attaching linguistic attributes into a rule-based framework. While fuzzy theory enables such a powerful inference mechanism, the computation with neural networks offers exciting advantages such as learning, adaptation, fault-tolerance, and generalization. Table 3.3 presents a brief comparison between fuzzy systems and neural networks.

The use of neural network training techniques allows us the ability to embed empirical information into a fuzzy system. This greatly expands the range of applications in which fuzzy systems can be used. The ability to make use of both expert and empirical information greatly enhances the utility of fuzzy systems.

Table 3.3 Comparison between fuzzy and neural systems.

	Fuzzy Systems	Neural Networks
Knowledge acquisition	Human experts	Numerical data
Training method	Interaction / induction	Algorithms / adjusting weights
Type of uncertainty	Qualitative / quantitative	Quantitative
Reasoning	Heuristic search	Parallel computations
Language interface	Explicit	Not evident
Fault tolerance	Not evident	Very high
Robustness	Very high	Very high

Fuzzy Neural Hybrids

Neural methods can be used in constructing fuzzy systems in ways other than training. They can also be used for rule selection, membership function determination and in what we can refer to as hybrid systems. Hybrid systems are systems that employ both fuzzy logic and neural networks. These hybrid systems may make use of a supervisory fuzzy system to select the best output from several neural networks or may use a neural network to intelligently combine the outputs from several fuzzy systems.

Membership function determination may be viewed as a data clustering and classification problem. Neural network architectures such as Kohonen Self Organizing Maps are well suited to finding clusters in input data. A typical use of neural networks for producing membership functions involves a two-stage process: *clustering* and

fuzzification (Adeli and Hung, 1995). The first stage is essentially a *classification* stage in which a neural network is used to classify or cluster domain data into a certain number of clusters. The second stage is a fuzzification process in which fuzzy membership values are assigned to each training instance in the set of classified clusters.

In fuzzy systems employing more than three or four fuzzy variables, it may be practically difficult to formulate fuzzy *if/then* rules, and it would be desirable if they could be extracted automatically out of data from the physical system being modeled. A comprehensive approach for the extraction and tuning of fuzzy rules was presented by Takagi and Hayashi (1991), and became known as *neural-network-driven fuzzy reasoning* or the *Takagi-Hayashi (T-H) method.*

When experiential data exists, fuzzy systems can be trained to represent an input-output relationship. By using gradient descent techniques, fuzzy system parameters, such as membership functions, and the connectives between layers in an adaptive network, can be optimized. *Adaptation* relates to the issue of *learning.* An adaptive system is one that can learn about the changes in the physical system and can modify its internal structure to improve the correspondence between the target system and itself and/or its environment.

Adaptive Network-Based Fuzzy Inference Systems

Jang (1993) introduced the adaptive network-based fuzzy inference system (ANFIS), also known as the adaptive neuro-fuzzy inference system. This system identifies a set of parameters through a hybrid learning rule combining the backpropagation gradient-descent and a least-squares method. ANFIS can construct an input-output mapping based

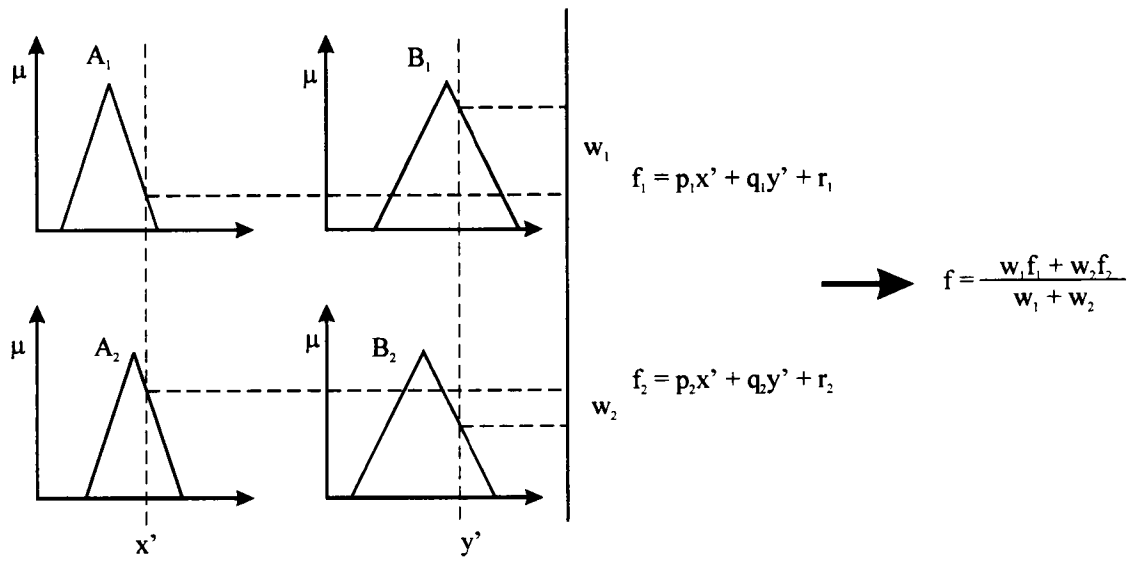
on both human knowledge (in the form of fuzzy if-then rules) and stipulated input-output data pairs.

Fundamentally, ANFIS is a graphical network representation of Sugeno-type fuzzy systems, endowed with neural learning capabilities. To illustrate its representational strength, let us consider two first-order Sugeno rules having outputs that are linear combinations of their inputs:

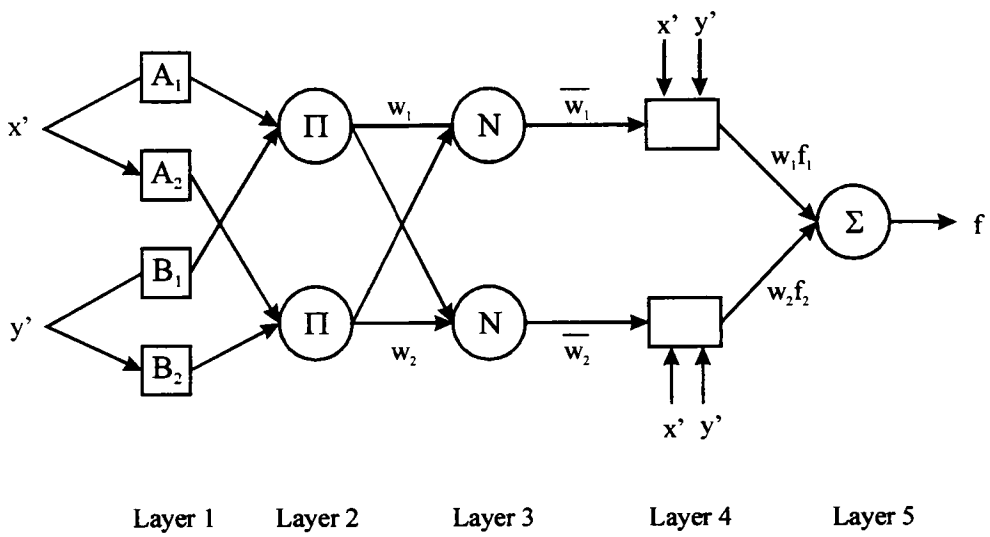
$$\text{If } x \text{ is } A_1 \text{ and } y \text{ is } B_1 \text{ then } f_1 = p_1x + q_1y + r_1 \qquad \text{(R3.9)}$$

$$\text{If } x \text{ is } A_2 \text{ and } y \text{ is } B_2 \text{ then } f_2 = p_2x + q_2y + r_2 \qquad \text{(R3.10)}$$

ANFIS can develop a network realization of these rules. Figure 3.14a illustrates the evaluation of the rules R3.9 and R3.10 and Figure 3.14b illustrates the corresponding ANFIS architecture.



(a)



(b)

Figure 3.14 (a) Evaluation of a network of rules. **(b)** Illustration showing the corresponding ANFIS architecture.

The nodes in the same layer of ANFIS are of the same function family and are arranged as follows:

Layer 1. Each node in this layer generates the membership grades of a linguistic label:

$$O_i^1 = \mu_{A_i}(x) \tag{3.21}$$

Layer 2. Each node in this layer calculates the firing strength of each rule via multiplication (or min):

$$O_i^2 = w_i = \mu_{Ai}(x) x \mu_{B_i}(y), \tag{3.22}$$

Layer 3. The *i*th node of this layer calculates the ratio of the *i*th rule's firing strength to the sum of all rules' firing strengths:

$$O_i^3 = \overline{w_i} = \frac{w_i}{w_1 + w_2}, \tag{3.23}$$

Layer 4. The *i*th node in this layer has the following function:

$$O_i^4 = \overline{w_i} f_i = \overline{w_i}(p_i x + q_i y + r_i) \tag{3.24}$$

Where w_i is the output of layer 3, and $\{p_i, q_i, r_i\}$ is the parameter set. Parameters in this layer will be referred to as the *consequent parameters*.

Layer 5. The single node in this layer aggregates the overall output as the summation of all incoming signals:

$$O_1^5 = \sum_i \overline{w}_i f_i = \frac{\sum_i w_i f_i}{\sum_i w_i} \tag{3.25}$$

Each ANFIS training epoch, using the hybrid learning rule, consists of two passes. The consequent parameters are obtained during the forward pass using a least-squares optimization algorithm and the premise parameters are updated using a gradient descent algorithm. During the forward pass all node outputs are calculated up to layer 4. At layer 4 the consequent parameters are calculated using a least-squares regression method. Next, the outputs are calculated using the new consequent parameters and the error signals are propagated back through the layers to determine the premise parameter updates.

CHAPTER 4

SYSTEM DESIGN

Goal of the System

The overall goal was to develop an automatic system to control the content of water in a soil within a small range very appropriate to plant development, i.e., close to *field capacity*. The basic inputs to the system were evapotranspiration and soil water potential. Evapotranspiration was obtained indirectly from solar radiation and relative humidity using weather sensors. Soil water potential was obtained directly using soil moisture sensors. The system should respond to current weather and soil water content changes on a real-time basis by processing the information at each fifteen-minute interval. The output of the system was a control signal sent to an external device. The system should also handle unreliable sensor data and record each control event.

Fuzzy System Design

The fuzzy system was designed as two subsystems. The first subsystem is a fuzzy evapotranspiration estimator (FEE), and the second subsystem is a fuzzy irrigation control system (FICS). The standard method of designing a fuzzy system involves identifying and naming the fuzzy inputs and outputs, creating the fuzzy membership functions for each of them, constructing the rule base, and deciding how the action will be carried out. MATLAB® with Fuzzy Logic Toolbox (Mathworks Inc., 1994) was used as the development environment.

Fuzzy Evapotranspiration Estimator (FEE)

At first, four weather variables commonly measured by automated weather stations were selected and used in the estimation of reference evapotranspiration using the Penman-Monteith model (Equation 2.1), these were solar radiation (RS), relative humidity (RH), air temperature (TA), and wind speed (WS). Since the tool being used deals with uncertainty, it is important to minimize the number of input variables to the estimator subsystem.

A simple linear regression was performed relating daily values of each of the four variables with daily values of evapotranspiration measured by a weighing lysimeter. The load cell used in the weighing mechanism in that lysimeter allowed minimum readings that corresponded to 0.13 mm of evapotranspiration, which was not appropriate for fifteen-minute intervals, or even for hourly intervals. Sensor data were previously converted to useful units after acquisition by a datalogger. The weather station and the lysimeter were installed at The University of Tennessee Plateau Experiment Station, located at Crossville, Tennessee.

The set of reliable data was chosen from months characterized as a crop production period, resulting in fifty days of weather data and lysimeter measurements from May and June of 1994, as shown in Table 4.1. Solar radiation (r^2=0.9074), and relative humidity (r^2=0.4970), presented the highest coefficients of determination among the four variables. Air temperature and wind speed presented coefficients of determination smaller than 0.1, facilitating the choice of solar radiation and relative humidity. Therefore they were the selected inputs for the fuzzy evapotranspiration estimator subsystem. These results are presented in Figures 4.1, 4.2, 4.3, and 4.4.

Table 4.1 Weather data and lysimeter measurements from a crop production period.

Day Julian	RS W/m^2/d	RH:mean %	TA:mean C	WS:mean m/s	Lysimeter ET mm/d
122	7357	65.84	10.57	1.96	4.41
123	811	89.10	10.53	1.35	0.13
124	2023	92.35	10.93	0.84	1.33
125	5870	76.55	12.16	1.05	3.16
126	5560	77.15	14.42	1.38	3.67
127	3175	84.15	14.96	2.79	1.73
128	7611	73.66	12.22	1.63	4.29
129	7931	71.54	12.67	0.91	4.92
130	6116	67.75	14.27	0.99	4.03
131	7754	67.41	13.72	0.99	5.15
132	6131	66.60	16.88	1.59	4.29
133	3113	72.19	11.09	0.53	2.02
134	6526	83.75	17.54	1.47	3.85
135	4260	86.35	19.29	1.70	2.32
136	8216	69.33	17.69	1.27	5.38
137	8442	66.11	13.95	1.94	5.26
138	8360	69.56	13.10	2.02	5.36
139	7917	69.85	10.94	2.28	4.62
140	8072	74.22	12.07	1.24	4.80
141	8024	71.11	12.92	0.95	4.77
142	8307	69.38	15.27	0.77	5.46
143	7279	70.68	17.96	0.64	4.03
144	7694	73.71	19.33	0.76	5.00
145	5874	77.99	18.97	1.18	3.72
146	1500	95.30	16.47	1.53	0.23
147	7863	75.54	13.12	1.25	3.98
148	8580	71.09	13.67	0.61	5.10
149	7927	75.91	15.85	0.85	4.90
150	7424	73.05	19.29	0.95	4.54
151	8140	73.68	18.86	0.40	5.43
152	6273	82.02	19.88	0.87	3.98
153	4822	82.51	20.68	0.80	2.86
154	6693	81.16	20.09	0.58	4.52
155	7769	75.74	20.21	0.64	5.33
156	5486	76.92	21.45	0.64	3.52
157	5594	84.38	22.72	1.16	3.21
162	5160	83.25	22.11	0.78	2.73
163	7690	88.45	23.00	0.67	4.11
164	5333	97.80	22.51	0.82	3.24
165	5076	86.75	22.99	0.72	3.11
166	3761	86.90	23.22	0.56	2.70
167	5635	82.48	22.78	0.48	3.70
169	7201	78.60	23.86	0.67	4.62
170	7321	77.20	23.65	0.48	5.36
171	3861	84.85	23.98	0.53	2.91
172	5949	79.03	24.01	0.77	4.23
173	5476	80.54	23.47	0.60	4.03
176	5853	80.39	20.32	1.44	4.11
179	3707	88.40	20.40	0.99	2.78
181	7719	77.50	22.53	0.99	5.77

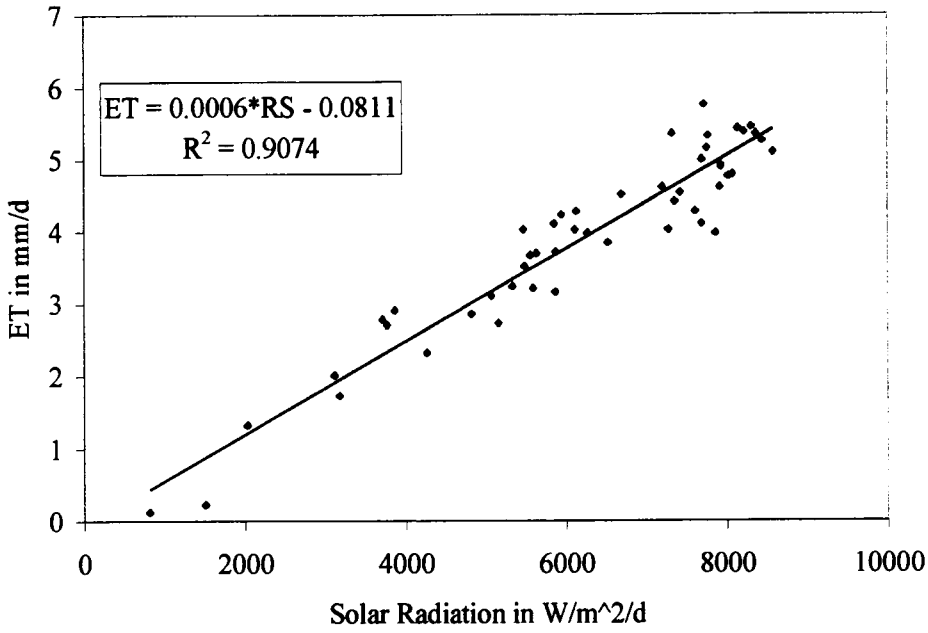


Figure 4.1 Relationship between Solar Radiation and ET.

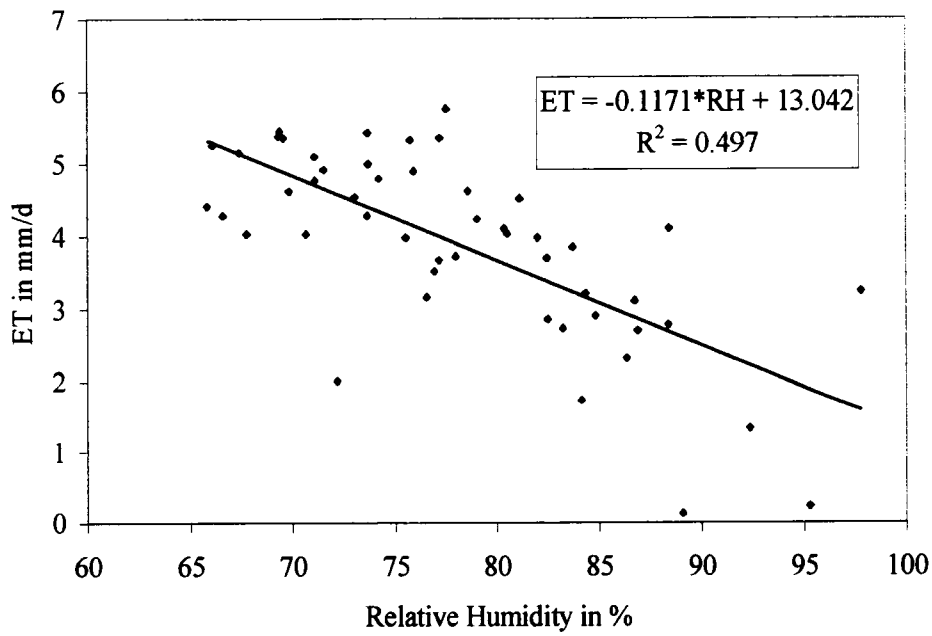


Figure 4.2 Relationship between Relative Humidity and ET.

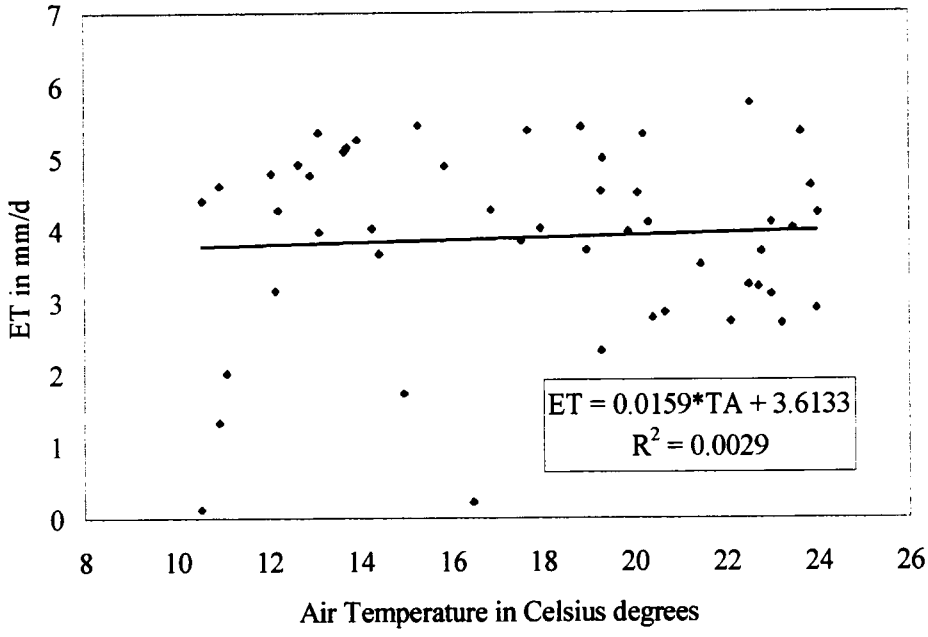


Figure 4.3 Relationship between Air Temperature and ET.

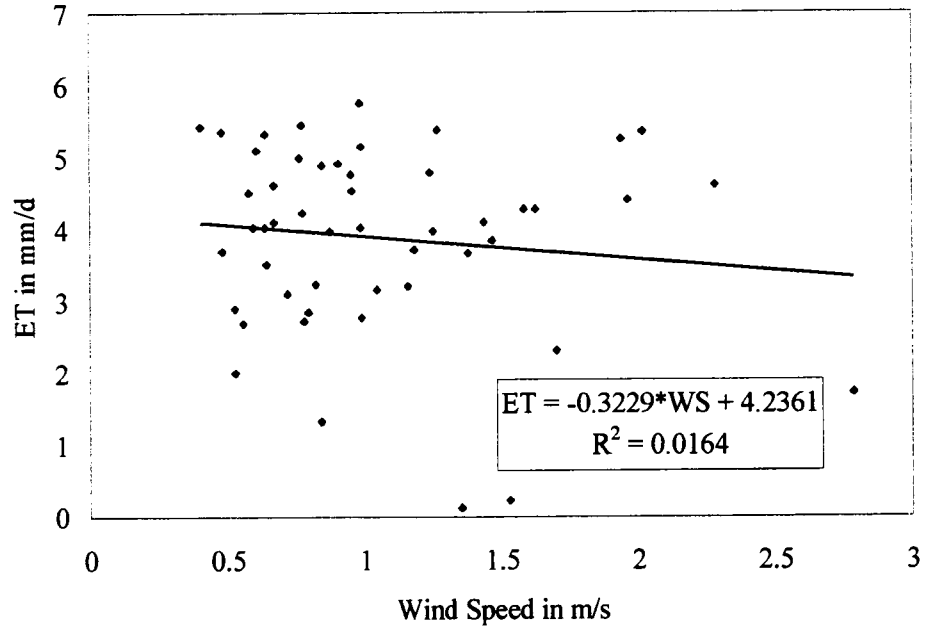


Figure 4.4 Relationship between Wind Speed and ET.

Fuzzy Algorithm

Next, an algorithm that executes the ET estimation procedure was developed and is summarized by the following steps:

1. Acquire the crisp input values for solar radiation and relative humidity.
2. Fuzzify the two inputs.
3. Determine antecedent truth values for the rule base using the fuzzy inputs.
4. Infer the consequent fuzzy output.
5. Defuzzify the output.

Fuzzy Input/Output Variables

The universe of discourse for the two inputs RS and RH were identified based on data from the weather station. The output universe of discourse was based on calculations of ET using the extreme values of each input variable. In Table 4.2 the identification of inputs and output is presented.

Table 4.2. Input and output description for fuzzy evapotranspiration estimator.

Name	Description	Units	Min	Max
Input:				
radi	Solar radiation	W/ m^2	0	1100
humi	Relative humidity	%	20	100
Output:				
et	Evapotranspiration	mm/15 min	0	0.24

Membership Functions

Design of membership functions is most often an intuitive process. There are no widely accepted procedures to systematically design membership functions. The approach taken was to assign an initial set of membership functions and modify them until satisfactory results were obtained. The design parameters are membership function shape, number of fuzzy sets on each input and output universe of discourse, and membership function location, width, and overlap.

Each input and output was assigned its own set of membership functions. Triangular membership functions are formed using straight lines. They were chosen because they have the advantage of simplicity. They can be expressed as

$$\mu(x) = \max(\min(\frac{x-a}{b-a}, \frac{c-x}{c-b}), 0) \tag{4.1}$$

where x can be any input/output value. The parameters a and c locate the base of the triangle and the parameter b locates the peak.

Initially, three sets were assigned to each universe of discourse. Simulation results indicated poor performance apparently caused by insufficient resolution on the universes of discourse. Resolution is a function of overlap and spacing between membership functions. The most common practice is to set up membership functions so that intersection with an adjacent set occurs at the crossover of each. For this design, the guideline was to force the intersection to occur between $\mu = 0.35$ and $\mu = 0.5$. In the end, five fuzzy sets were assigned to each universe of discourse. Table 4.3 presents the identification for the term sets of the two inputs.

Table 4.3 Input fuzzy sets for the fuzzy ET estimator.

Name	Description	Left	Center	Right
radi:				
VL	Very Low	0	0	250
LOW	Low	0	250	500
MED	Medium	250	500	750
HIGH	High	500	750	1000
VH	Very High	750	1100	1100
humi:				
VL	Very Low	20	20	40
LOW	Low	20	40	60
MED	Medium	40	60	80
HIGH	High	60	80	100
VH	Very High	80	100	100

Figure 4.5 shows the membership functions for input variable *radi.* Figure 4.6 shows membership functions for *humi.* Table 4.4 shows the identification for the output term sets. In Figure 4.7 the membership functions for the output variable *et* is shown.

Rule Base

The rules were assessed based on the existent knowledge about evapotranspiration and based on the relationships between each input and evapotranspiration obtained in the regression analysis. Twenty-five rules were developed for the FEE as described below:

If *radi* is VH and *humi* is VL then *et* is VH else (R4.1)

If *radi* is VH and *humi* is LOW then *et* is VH else (R4.2)

If *radi* is VH and *humi* is MED then *et* is VH else (R4.3)

If *radi* is VH and *humi* is HIGH then *et* is HIGH else (R4.4)

If *radi* is VH and *humi* is VH then *et* is HIGH else (R4.5)

If *radi* is HIGH and *humi* is VL then *et* is HIGH else (R4.6)

If *radi* is HIGH and *humi* is LOW then *et* is HIGH else (R4.7)

If *radi* is HIGH and *humi* is MED then *et* is HIGH else (R4.8)

If *radi* is HIGH and *humi* is HIGH then *et* is MED else (R4.9)

If *radi* is HIGH and *humi* is VH then *et* is MED else (R4.10)

If *radi* is MED and *humi* is VL then *et* is MED else (R4.11)

If *radi* is MED and *humi* is LOW then *et* is MED else (R4.12)

If *radi* is MED and *humi* is MED then *et* is MED else (R4.13)

If *radi* is MED and *humi* is HIGH then *et* is MED else (R4.14)

If *radi* is MED and *humi* is VH then *et* is MED else (R4.15)

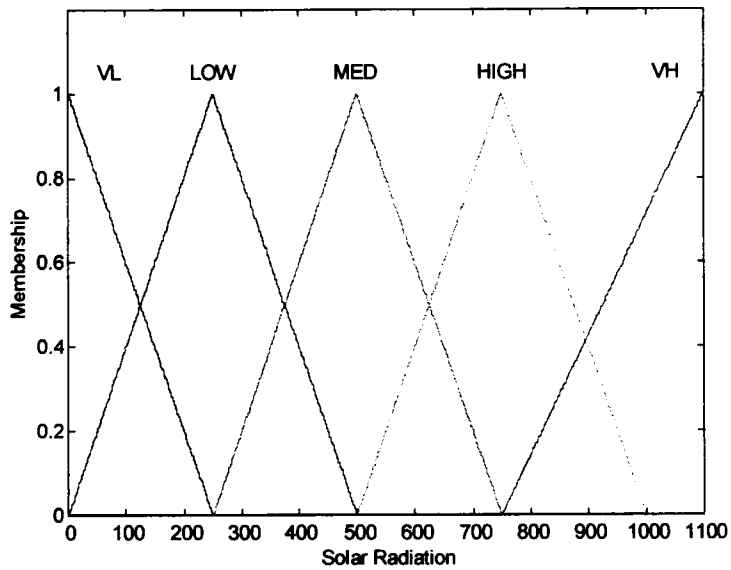


Figure 4.5 Membership functions defined for the input variable *radi.*

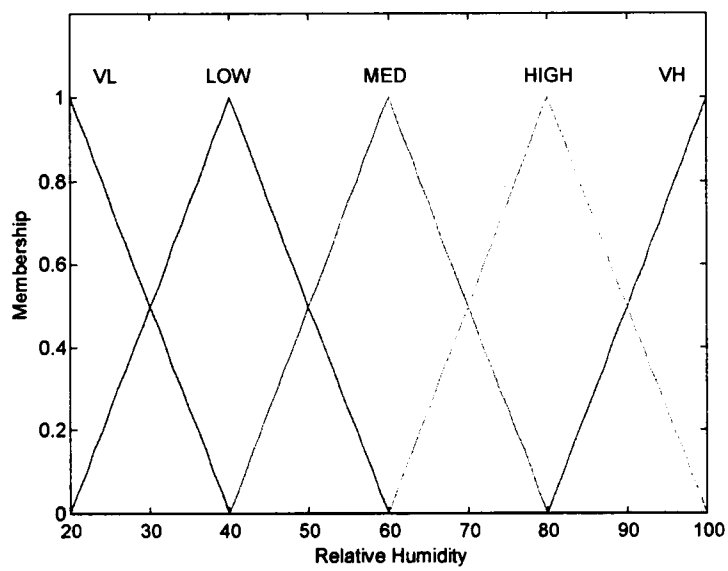


Figure 4.6 Membership functions defined for the input variable *humi.*

Table 4.4 Output fuzzy sets defined for the fuzzy ET estimator.

Name	Description	Left	Center	Right
et:				
VL	Very Low	0	0	0.06
LOW	Low	0	0.06	0.12
MED	Medium	0.06	0.12	0.18
HIGH	High	0.12	0.18	0.24
VH	Very High	0.18	0.24	0.24

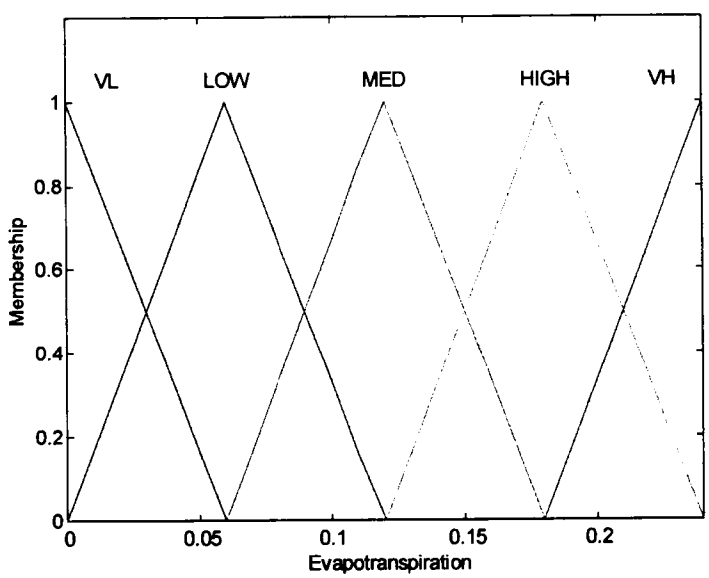


Figure 4.7 Membership functions defined for the output variable *et*.

If *radi* is LOW and *humi* is VL then *et* is MED else (R4.16)

If *radi* is LOW and *humi* is LOW then *et* is MED else (R4.17)

If *radi* is LOW and *humi* is MED then *et* is LOW else (R4.18)

If *radi* is LOW and *humi* is HIGH then *et* is LOW else (R4.19)

If *radi* is LOW and *humi* is VH then *et* is LOW else (R4.20)

If *radi* is VL and *humi* is VL then *et* is LOW else (R4.21)

If *radi* is VL and *humi* is LOW then *et* is LOW else (R4.22)

If *radi* is VL and *humi* is MED then *et* is VL else (R4.23)

If *radi* is VL and *humi* is HIGH then *et* is VL else (R4.24)

If *radi* is VL and *humi* is VH then *et* is VL (R4.25)

The set of rules that describes a system is called the *Fuzzy Associative Memory* (FAM). The linguistic rule base for the FEE is presented as a FAM table in Table 4.5.

Table 4.5 FAM table that describes the fuzzy evapotranspiration estimator.

	***radi* is VL**	***radi* is LOW**	***radi* is MED**	***radi* is HIGH**	***radi* is VH**
***humi* is VL**	LOW	MED	MED	HIGH	VH
***humi* is LOW**	LOW	MED	MED	HIGH	VH
***humi* is MED**	VL	LOW	MED	HIGH	VH
***humi* is HIGH**	VL	LOW	MED	MED	HIGH
***humi* is VH**	VL	LOW	MED	MED	HIGH

Inference Engine and Defuzzifier

Each fuzzy input is essentially a series of membership values corresponding to the fuzzy sets defined on its universe of discourse. These membership values are used to determine which rules apply to a given situation, and to what degree each applies.

The first step in the procedure is to determine how "true" is the antecedent of each rule. It consists of evaluating the fuzzy relation "*radi* is *X* and *humi* is *Y*." The *minimum* (min) operator was used to evaluate the fuzzy *and* operator

$$\mu_{XY}(radi, humi) = \min\{\mu_X(radi), \mu_Y(humi)\} \tag{4.2}$$

The second step involves the application of an implication operator to the consequent of each rule. It consists of evaluating the fuzzy implication relation "then *et* is *Z*." The *product* implication operator was applied to the consequent of the rules.

$$\mu_Z\{(radi, humi), et\} = \text{prod}\{[\min(\mu_X(radi), \mu_Y(humi))], \mu_Z(et)\} \tag{4.3}$$

The consequences of the rules were aggregated using the *maximum* (max) operator, i.e., selecting the maximum values of the consequent of each rule to generate a fuzzy output. To obtain the representative crisp value for evapotranspiration, the center of area defuzzification technique known as the centroid method, was applied.

An example of the inference procedure with the defuzzification method is considered for the case *radi* = 950 and *humi* = 30 and shown as Figure 4.8. The complete output surface of the fuzzy evapotranspiration estimator is presented as Figure 4.9.

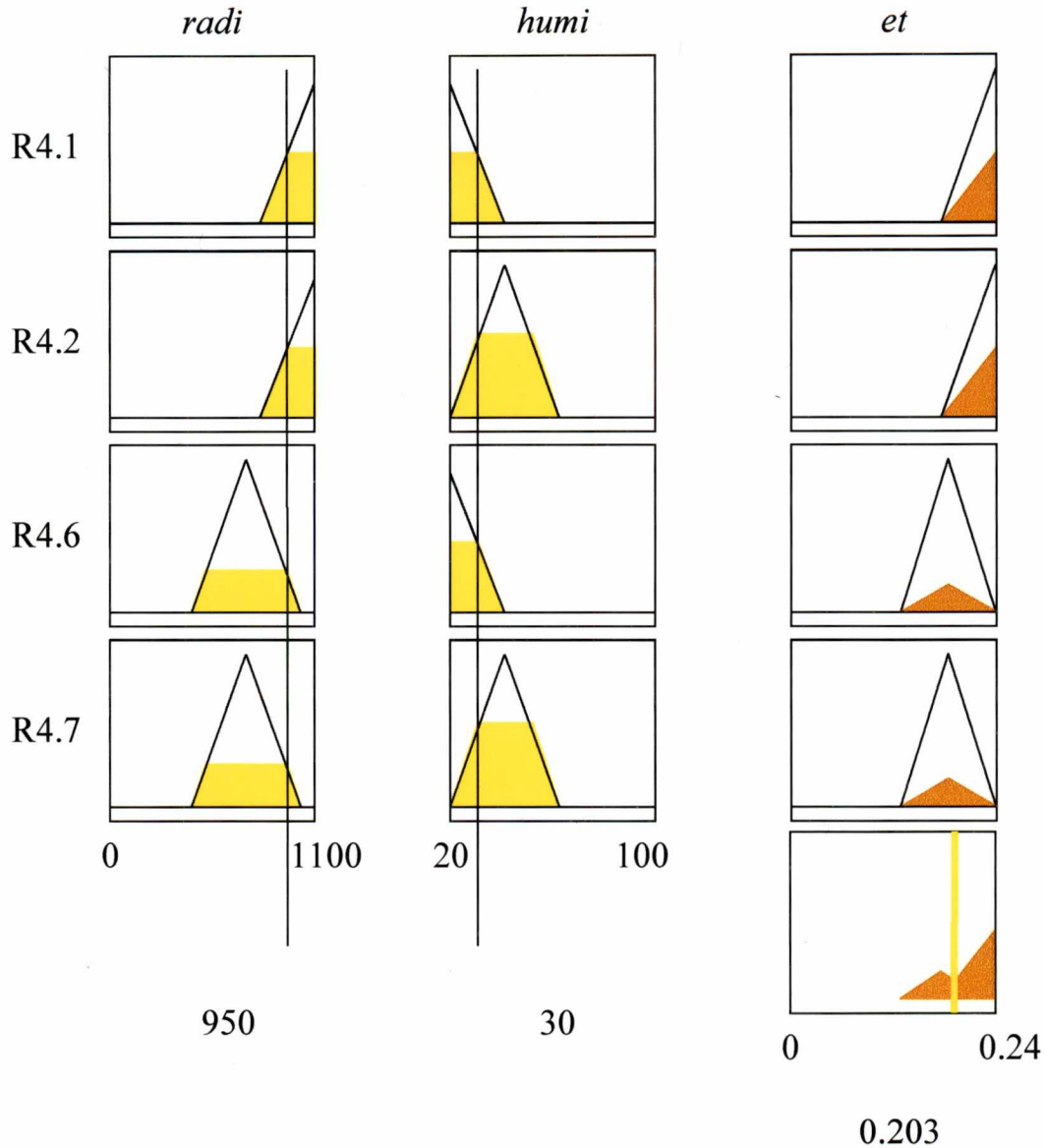


Figure 4.8 Example of fuzzy inference and defuzzification for the case *radi* = 950 and *humi* = 30.

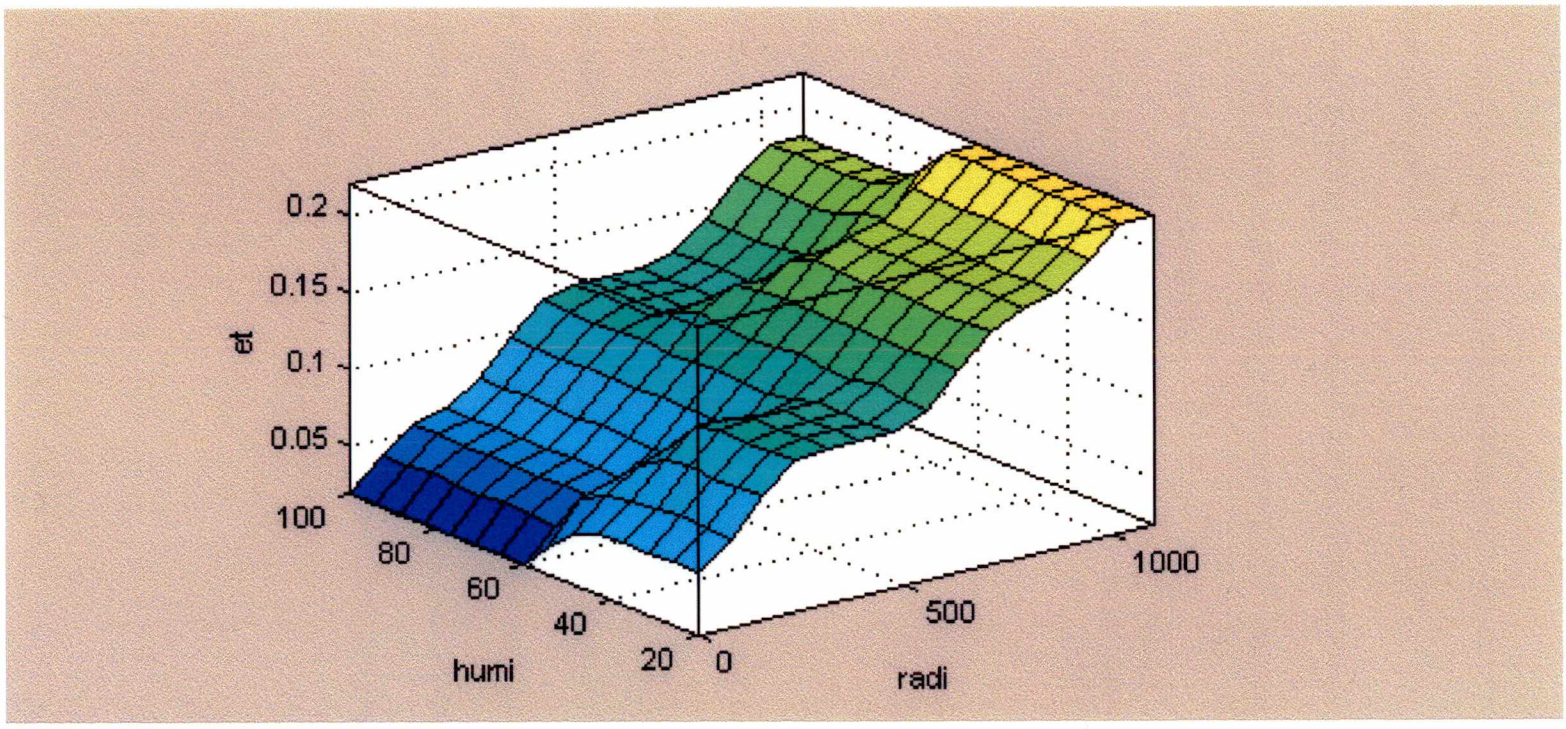


Figure 4.9 Output surface of the fuzzy evapotranspiration estimator.

Fuzzy Irrigation Control System (FICS)

Fuzzy Algorithm

An algorithm that executes the control procedure was developed and is summarized by the following steps:

1. Acquire the crisp input values for evapotranspiration, soil moisture content at 15-cm depth, and soil moisture content at 30-cm depth.
2. Fuzzify the three inputs.
3. Determine antecedent truth values for the rule base using the fuzzy inputs.
4. Infer the consequent fuzzy output.
5. Defuzzify the output.

Fuzzy Input/Output Variables

Evapotranspiration as estimated by the FEE, and soil water potential at two depths in the crop root zone (15 cm and 30 cm), were the inputs for the FICS. Soil water potential values were used here as raw data (voltage ratio), and received from the sensors on a real-time basis. The output was irrigation requirement as a control value that could be converted further into soil water potential allowing control based on an actual state variable. The input and output descriptions are presented in Table 4.6.

Table 4.6 Input and output description for fuzzy irrigation control system.

Name	Description	Units	Min	Max
Input:				
et	Evapotranspiration	mm/15 min	0	0.24
soil1	Soil water status-shallow	V_{out}/V_{in}	0	1
soil2	Soil water status-deep	V_{out}/V_{in}	0	1
Output:				
irr	Irrigation requirement	unit	0	10

Membership Functions

Following the same approach used for the fuzzy evapotranspiration estimator an initial set of membership functions was assigned and modified until satisfactory results were obtained. Triangular membership functions were also chosen for the fuzzy irrigation control subsystem. Three fuzzy sets were assigned to each input universe of discourse. Five fuzzy sets were assigned to the output universe of discourse. Table 4.7 presents the identification for the term sets of the three inputs. In Figure 4.10 the membership functions for the input variable *et* is shown. Figure 4.11 shows the membership functions for the input variables *soil1* and *soil2.* The identification for the output term sets is presented in Table 4.8. The membership functions for the output variable *irr* are shown in Figure 4.12.

Table 4.7 Input fuzzy sets defined for fuzzy irrigation control system.

Name	Description	Left	Center	Right
et:				
LOW	Low	0	0	0.12
MED	Medium	0	0.12	0.24
HIGH	High	0.12	0.24	0.24
soil1:				
WET	Wet	0	0	0.60
MED	Medium	0.30	0.60	0.90
DRY	Dry	0.60	1	1
soil2:				
WET	Wet	0	0	0.60
MED	Medium	0.30	0.60	0.90
DRY	Dry	0.60	1	1

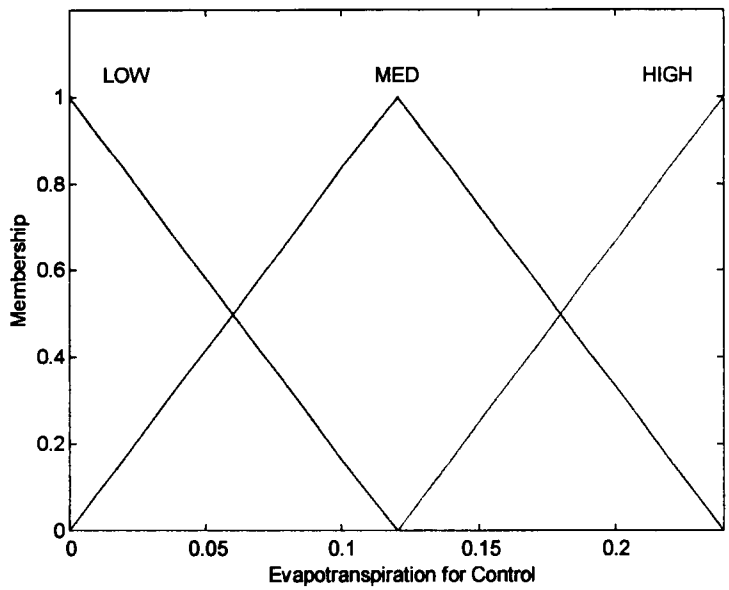


Figure 4.10 Membership functions defined for the input variable *et*.

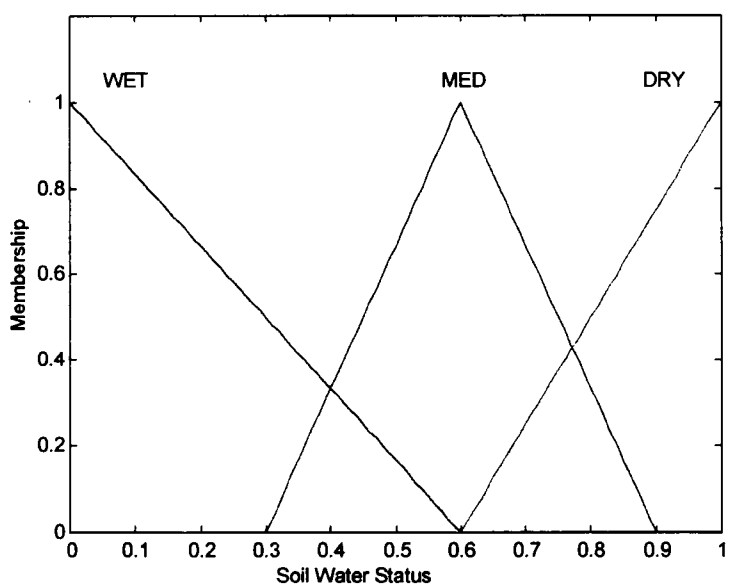


Figure 4.11 Membership functions defined for the input variables *soil1* and *soil2*.

Table 4.8 Output fuzzy sets defined for fuzzy irrigation control.

Name	Description	Left	Center	Right
irr:				
VL	Very Low	0	0	2.5
LOW	Low	0	2.5	5
MED	Medium	2.5	5	7.5
HIGH	High	5	7.5	10
VH	Very High	7.5	10	10

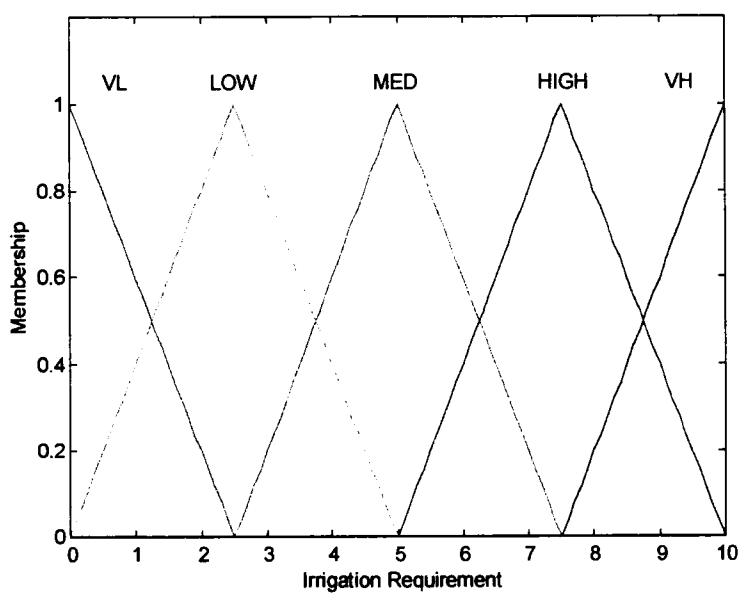


Figure 4.12 Membership functions defined for the output variable *irr*.

Rule Base

The fuzzy rules were assessed based on the knowledge of the relationships among evapotranspiration, crop water requirements, infiltration, and drainage. The basic idea was to maintain the soil water content within certain levels in the crop root zone, avoiding deep drainage, and considering the evapotranspiration rate. Twenty-seven if-then rules were assessed as presented below:

If *et* is LOW AND *soil1* is WET and *soil2* is WET then *irr* is VL else (R4.26)

If *et* is LOW AND *soil1* is WET and *soil2* is MED then *irr* is VL else (R4.27)

If *et* is LOW AND *soil1* is WET and *soil2* is DRY then *irr* is VL else (R4.28)

If *et* is LOW AND *soil1* is MED and *soil2* is WET then *irr* is VL else (R4.29)

If *et* is LOW AND *soil1* is MED and *soil2* is MED then *irr* is VL else (R4.30)

If *et* is LOW AND *soil1* is MED and *soil2* is DRY then *irr* is LOW else (R4.31)

If *et* is LOW AND *soil1* is DRY and *soil2* is WET then *irr* is MED else (R4.32)

If *et* is LOW AND *soil1* is DRY and *soil2* is MED then *irr* is HIGH else (R4.33)

If *et* is LOW AND *soil1* is DRY and *soil2* is DRY then *irr* is HIGH else (R4.34)

If *et* is MED AND *soil1* is WET and *soil2* is WET then *irr* is LOW else (R4.35)

If *et* is MED AND *soil1* is WET and *soil2* is MED then *irr* is LOW else (R4.36)

If *et* is MED AND *soil1* is WET and *soil2* is DRY then *irr* is LOW else (R4.37)

If *et* is MED AND *soil1* is MED and *soil2* is WET then *irr* is LOW else (R4.38)

If *et* is MED AND *soil1* is MED and *soil2* is MED then *irr* is MED else (R4.39)

If *et* is MED AND *soil1* is MED and *soil2* is DRY then *irr* is MED else (R4.40)

If *et* is MED AND *soil1* is DRY and *soil2* is WET then *irr* is MED else (R4.41)

If *et* is MED AND *soil1* is DRY and *soil2* is MED then *irr* is HIGH else (R4.42)

If *et* is MED AND *soil1* is DRY and *soil2* is DRY then *irr* is VH else (R4.43)

If *et* is HIGH AND *soil1* is WET and *soil2* is WET then *irr* is LOW else (R4.44)

If *et* is HIGH AND *soil1* is WET and *soil2* is MED then *irr* is MED else (R4.45)

If *et* is HIGH AND *soil1* is WET and *soil2* is DRY then *irr* is MED else (R4.46)

If *et* is HIGH AND *soil1* is MED and *soil2* is WET then *irr* is MED else (R4.47)

If *et* is HIGH AND *soil1* is MED and *soil2* is MED then *irr* is HIGH else (R4.48)

If *et* is HIGH AND *soil1* is MED and *soil2* is DRY then *irr* is HIGH else (R4.49)

If *et* is HIGH AND *soil1* is DRY and *soil2* is WET then *irr* is HIGH else (R4.50)

If *et* is HIGH AND *soil1* is DRY and *soil2* is MED then *irr* is VH else (R4.51)

If *et* is HIGH AND *soil1* is DRY and *soil2* is DRY then *irr* is VH (R4.52)

Inference Engine and Defuzzifier

The same procedures and defuzzification technique used for the fuzzy evapotranspiration estimator were used for the fuzzy irrigation control. The *minimum* (min) operator was used to evaluate the fuzzy *and* operator in fuzzy relation "*et* is *X* and *soil1* is *Y* and *soil2* is *Z*"

$$\mu_{XYZ}(et, soil1, soil2) = \min\{\mu_X(et), \mu_Y(soil1), \mu_Z(soil2\} \quad (4.4)$$

The fuzzy implication relation "then *irr* is *W*" was evaluated applying the *product* implication operator to the consequent of the rules

$$\mu_W\{(et, soil1, soil2), irr\} = \text{prod}\{[\min(\mu_X(et), \mu_Y(soil1), \mu_Z(soil2))], \mu_W(irr)\} \quad (4.5)$$

The consequences of the rules were aggregated using the *maximum* (max) operator to generate a fuzzy output. The centroid method was used to obtain the representative crisp value for irrigation requirement.

Figure 4.13 shows an example of the inference procedure and defuzzification method for the case *et* = 0.12, *soil1* = 0.70, and *soil2* = 0.80. The output surface of the fuzzy irrigation control with *et* = 0.20 is presented as Figure 4.14.

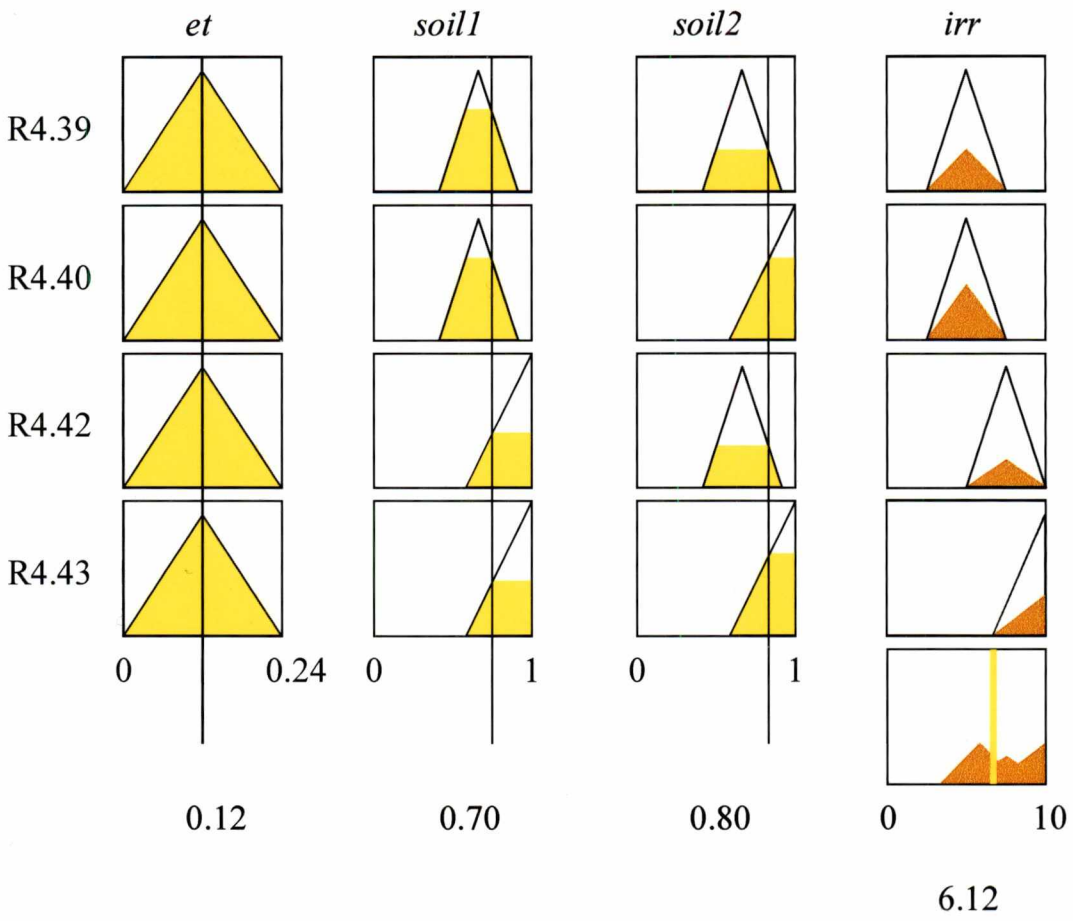


Figure 4.13 Example of fuzzy inference and defuzzification for the case *et* = 0.12, *soil1* = 0.70, and *soil2* = 0.80.

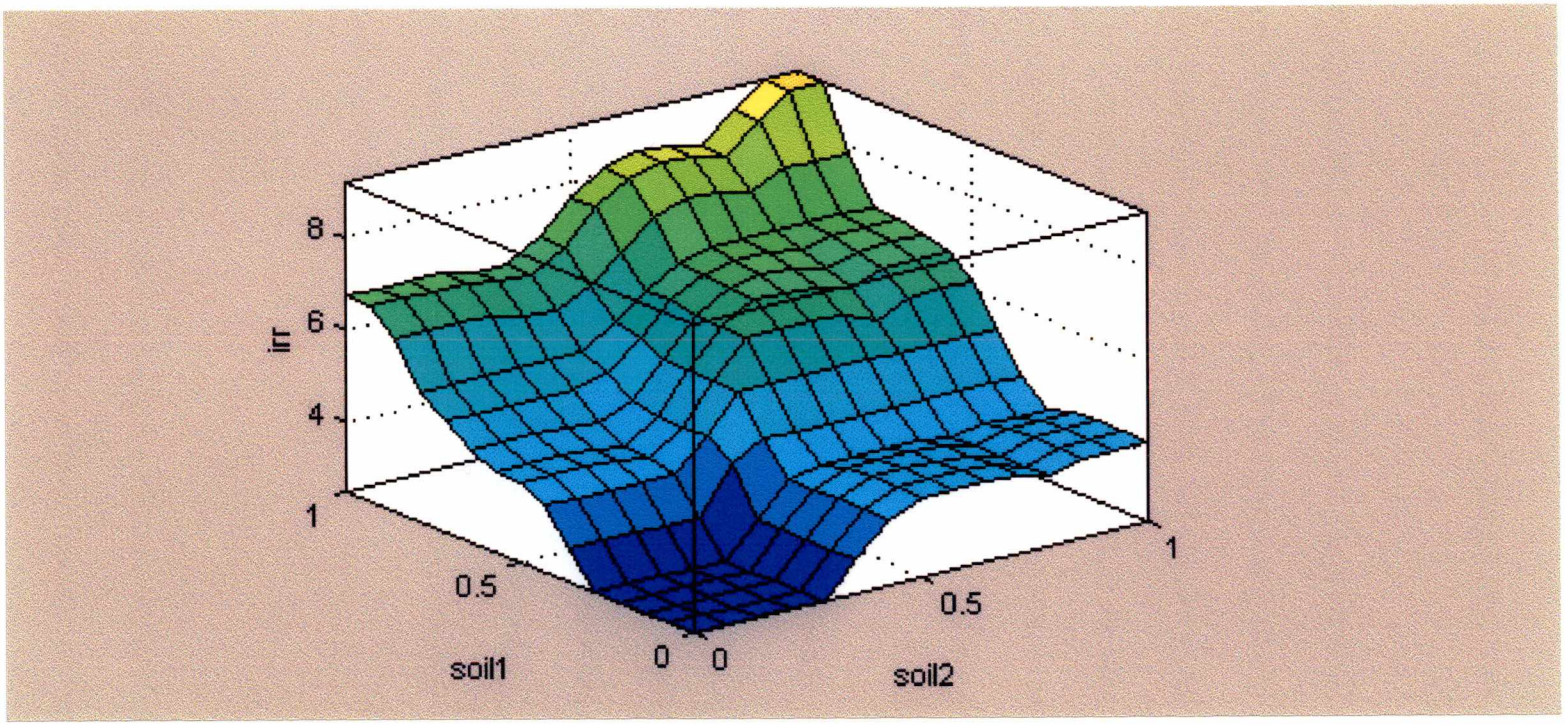


Figure 4.14 An example of an output surface of the fuzzy irrigation control system for a fixed value of *et* = 0.20.

CHAPTER 5

SYSTEM IMPLEMENTATION

A field experiment was implemented at The University of Tennessee Plateau Experiment Station, near Crossville, TN at an elevation of 573 m at 35° 55' N by 85° 07' W. The area presents sandy-loam soils of the Lily Series which is a typical cropland soil series of the Cumberland Plateau. The idea was to test the fuzzy irrigation control system and also to compare it with a high-level conventional irrigation control system. The word conventional is used here to describe a non-fuzzy control system without feedback.

Conventional Irrigation Control System (CICS)

The conventional system was designed to control irrigation based on crop evapotranspiration estimated using the reference ET Penman-Monteith method, as shown in Equation 2.1. The CICS should activate its control valve every time the cumulative crop evapotranspiration reached a certain crisp threshold value. The amount of water released to the crop area under control in 15 minutes was equivalent to 2.5 mm, and it was the chosen threshold value.

Inputs

Evapotranspiration estimations using the Penman-Monteith equation require at least four weather variables: solar radiation, relative humidity, air temperature, and wind speed. Sensors for each of these variables are common parts of any standard automatic

weather station. The system was developed to receive the inputs from the weather station, to process the information, and to send control actions to activate or deactivate a specific valve. There was no feedback in this system, which is a very common situation in current irrigation control systems. Figure 5.1 shows a block diagram of the conventional irrigation control system.

Crop

Bell pepper is a crop with considerable potential for the Cumberland Plateau region of Tennessee. Peppers have been classified as susceptible to very susceptible to water stress, with blossom stage being the most sensitive period (Bruce et al., 1980). Yields and water use have been shown to be greatest when irrigation was applied at 25 kPa as compared to 50 and 75 kPa, with soil water tensions being measured at 10 cm (Smittle et al., 1994).

According to the FAO crop coefficient method (Doorenbos and Pruitt, 1977), the crop coefficient for peppers with a frequency of irrigation less than 2 days varies from 0.95 to 1.05 for the entire season. A crop coefficient equal to 1.00 was chosen to compute the crop evapotranspiration.

Bell pepper response to polyethylene mulch varies from no response in relatively dry years with split applications of nitrogen to substantial response in wet seasons (Locascio et al., 1985). The use of polyethylene mulch is becoming a very widely used technique for bell pepper production. Coupled with trickle irrigation and well adapted varieties, this appears to be a good method to increase fruit yields and quality.

The bell pepper varieties "Enterprise" and "King Arthur" were planted in double rows with a 45-cm in-row spacing and 45 cm between rows. In this arrangement each

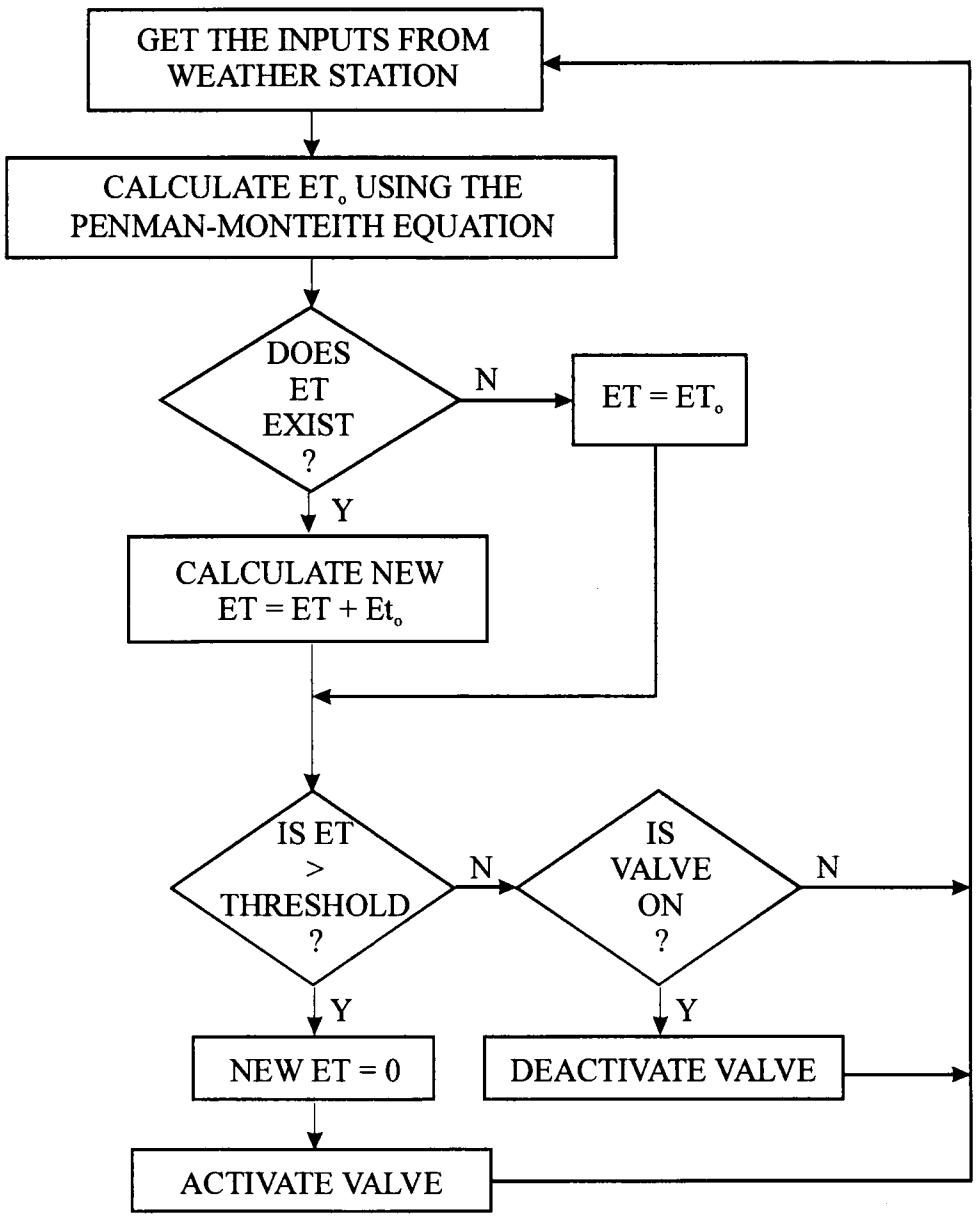


Figure 5.1 Flow chart for the conventional irrigation control system.

variety corresponded to one specific row. Polyethylene mulch was used and the crop was irrigated using a trickle irrigation system.

Statistical Design

The appropriate design to compare the systems was a randomized block. Three treatments were tested in six blocks. Each block consisted of three experimental units 12.2 m long and 4.05 m wide. Each experimental unit consisted of three plots of double rows separated by 1.8 m between the bed centers. The total area of the 18 experimental units was 1340 m^2. The treatments were specified as:

1. Fuzzy Logic based automatic irrigation control.
2. Conventional automatic irrigation control.
3. No irrigation.

Randomization

A normally distributed random matrix of three rows and six columns was generated using MATLAB®. A "seed" was applied to allow knowledge of the elements, i. e., the same values are presented every time the same seed is used (Appendix A1). The randomized block layout with 18 experimental units is shown in Figure 5.2. An experimental unit layout is shown in Figure 5.3.

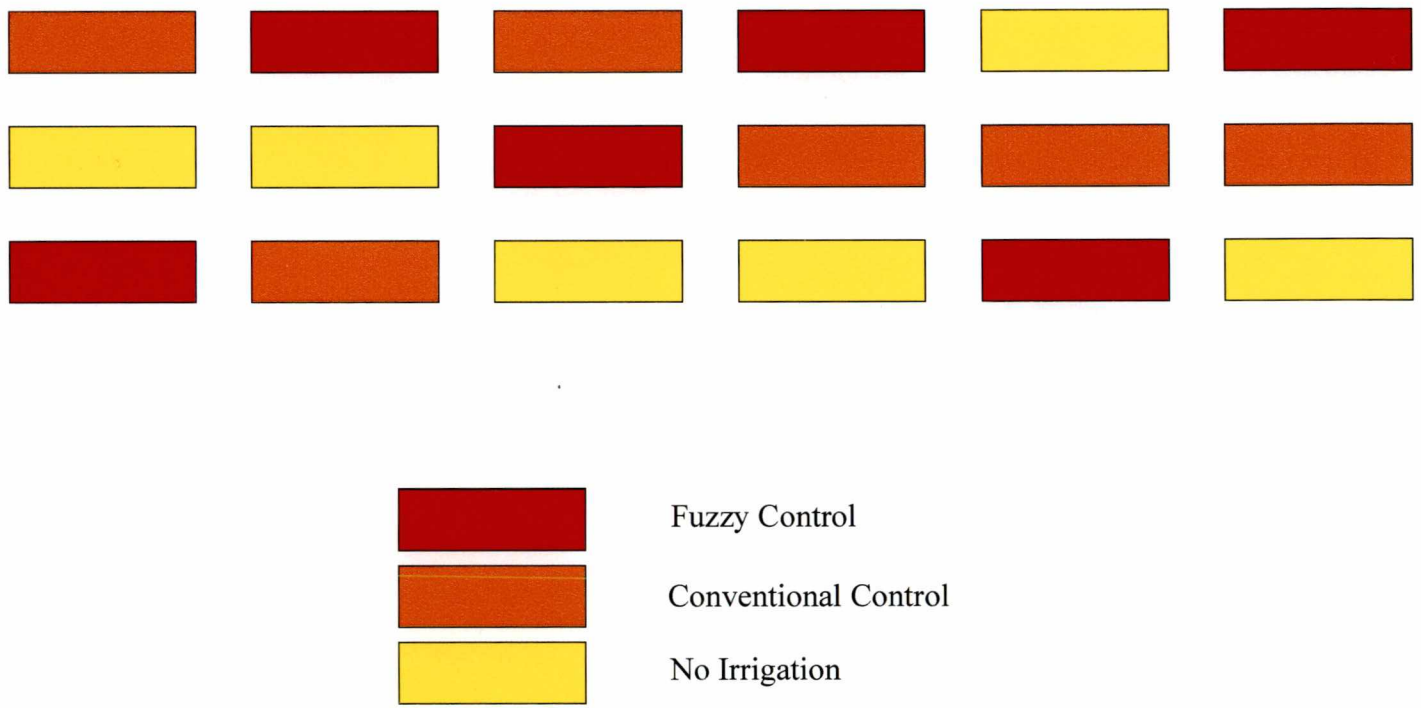


Figure 5.2 Randomized block layout with 18 experimental units.

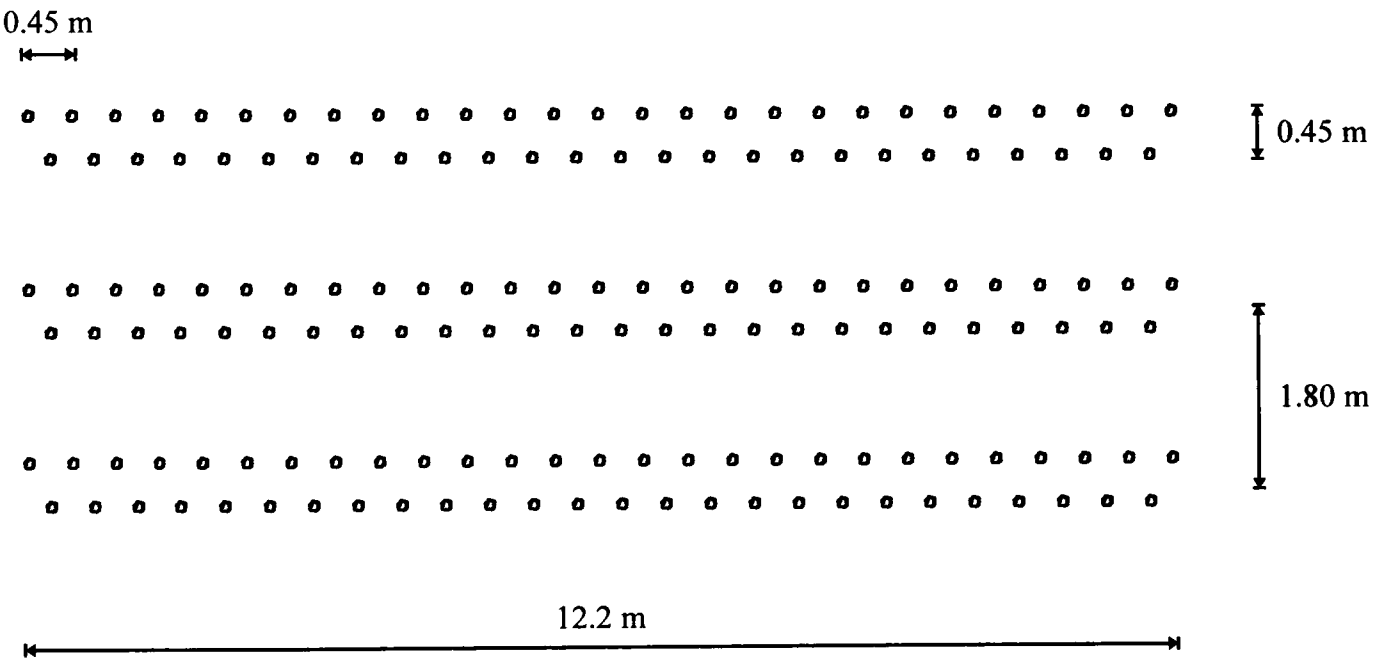


Figure 5.3 Experimental unit layout showing double rows.

Irrigation System

Microirrigation systems wet only the soil volume around the emitters. In order to minimize the effects of soil spatial variability on water distribution, these systems should apply water in small volumes, at a high frequency, and at a rate equal to plant water uptake. The soil water potential can be maintained reasonably constant under high frequency irrigation. Irrigation control systems should maintain soil water content at the "optimum" level for the crops and avoid applying excessive amounts of water. Irrigation water should be applied in precise amounts at a rate equivalent to the evapotranspiration rate.

A microirrigation system consists basically of a water supply, a pump, and a filter, followed by a network of mainlines, laterals, and emitters. A trickle irrigation system was designed for this experiment. The source of water was municipal, therefore a pump was not required. A filtration system composed of a sand filter and a screen filter (Fresno Valves and Castings, Inc., Selma, CA) was installed. Instead of isolated emitters, Typhoon drip lines (Netafim Irrigation, Inc., Altamonte Springs, FL) with emitters spaced at each 45 cm, were used for each plant row. At a pressure of 82 kPa the emitter discharge was about 1.5 l/h. Pressure regulators, pressure gauges, and flow meters were also included in the irrigation system. A layout of the irrigation system is shown in Figure 5.4. The filtration system is shown in Figure 5.5.

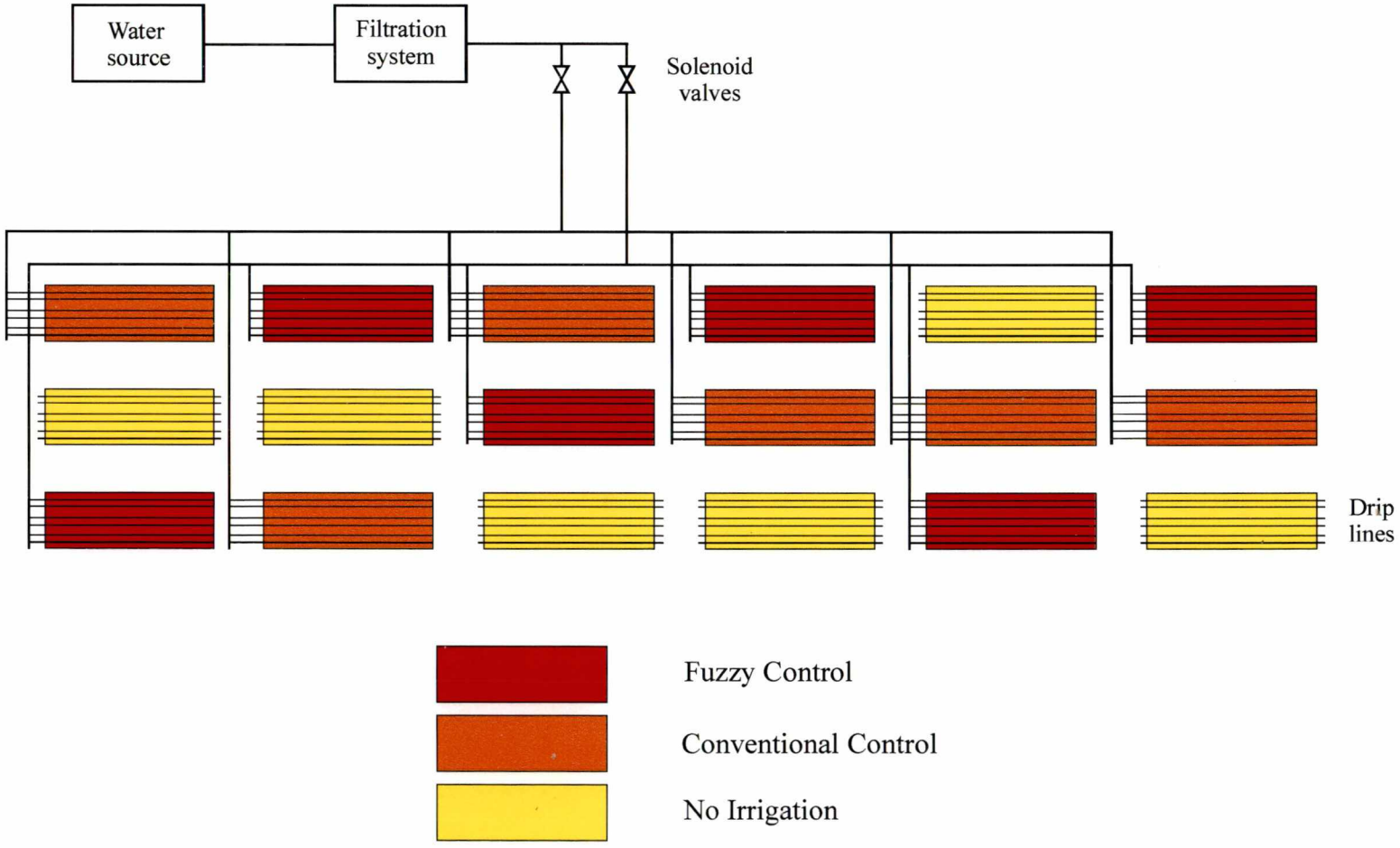


Figure 5.4 Irrigation system layout showing drip lines, pipes, solenoid valves, and filtration system.

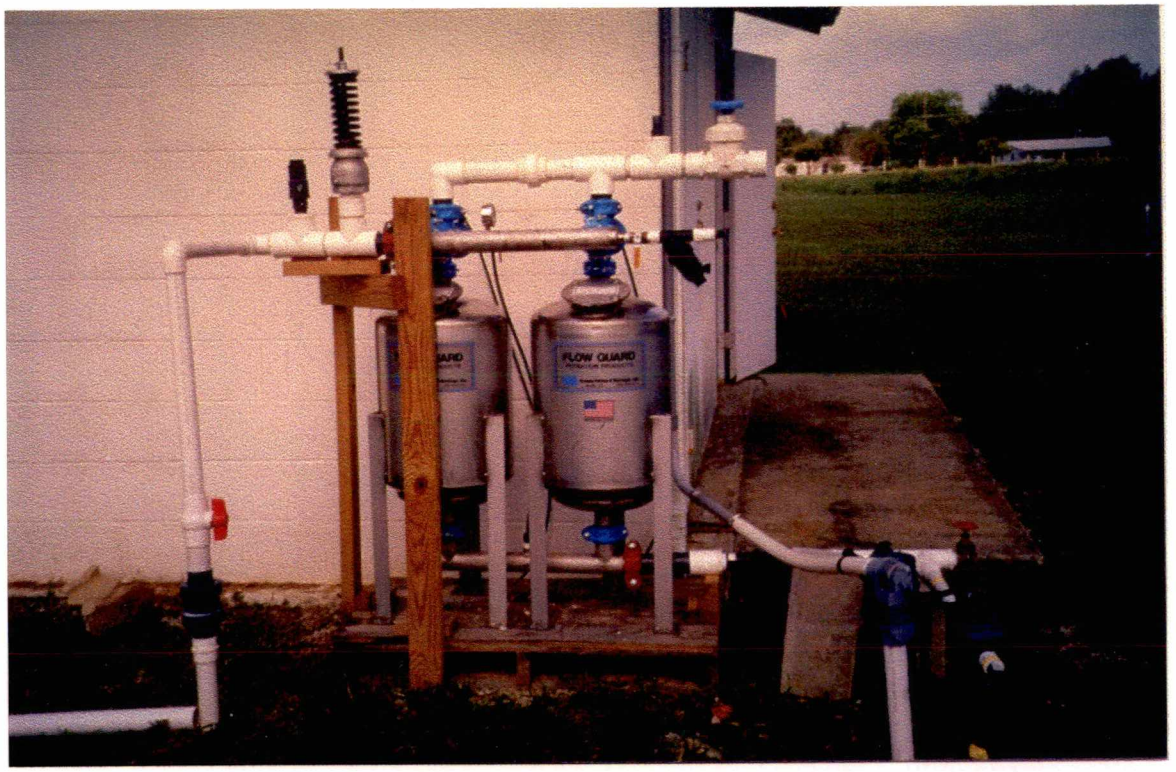


Figure 5.5 Filtration system composed of a sand filter and a screen filter.

Control System

Hardware

Since the irrigation control systems were designed for real-time control, external devices had to be accessed, either to acquire input data or to send control actions. The following hardware was used to implement the systems in the field:

- Automated weather station,
- Soil moisture sensors,
- Thermocouple,
- Multiplexer,
- Datalogger,
- Personal computer,
- Computer digital input/output board,
- Solenoid valves.

Automated Weather Station

The weather station was equipped with a Li-Cor LI200SZ pyranometer sensor, a Campbell Scientific HMP35C air temperature and relative humidity probe, a R. M. Young Wind Sentry Set including a vertical-axis anemometer and wind direction vane, and a tipping bucket rain gauge. The station provides the following data: solar radiation, relative humidity, air temperature, wind speed, and precipitation.

Soil Moisture Sensors

The Watermark® electrical resistance type sensor is relatively inexpensive and can be interfaced with computer controlled systems. In this research we used the Watermark model 200SS (Irrometer Co., Riverside, CA). According to the manufacturer they provide accurate readings over a soil water potential range of 0 to –200 kPa, which covers the entire soil moisture range required in irrigated agriculture. The measured resistance can be converted to soil water potential using Equation 2.3. The other input required in this equation is soil temperature, measured at the same sensor depth. This information was provided by thermocouples.

Two sensors were installed at the central plots of each experimental unit. One was installed 15 cm deep and the other was installed 30 cm deep. Both were located in the wetted area at 30 cm from the emitters. Each treatment was provided 12 sensors, 6 installed 15 cm deep and 6 installed 12 cm deep. A total of 36 Watermark® sensors were installed for the three treatments. Figure 5.6 shows a layout of the location of the 36 sensors. Figure 5.7 shows soil moisture sensors being installed at the field.

Thermocouple

Copper-constantan (Type-T) thermocouples were installed in six experimental units in two blocks. A total of 12 thermocouples were installed close to 12 soil moisture sensors. Six were installed 15 cm deep and the other six were installed 30 cm deep. Figure 5.8 shows the location layout of 12 thermocouples.

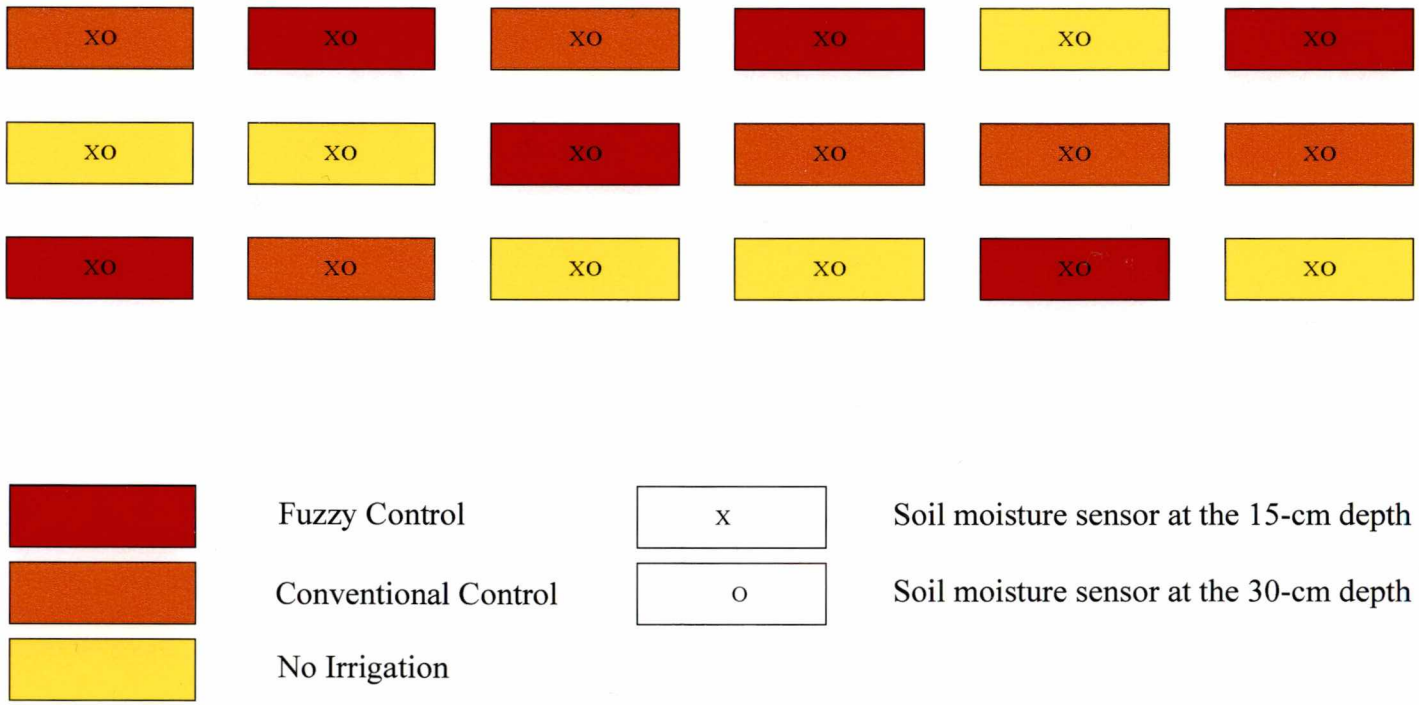


Figure 5.6 Location of the soil moisture sensors at each experimental unit.

Figure 5.7 Soil moisture sensors being installed at the field.

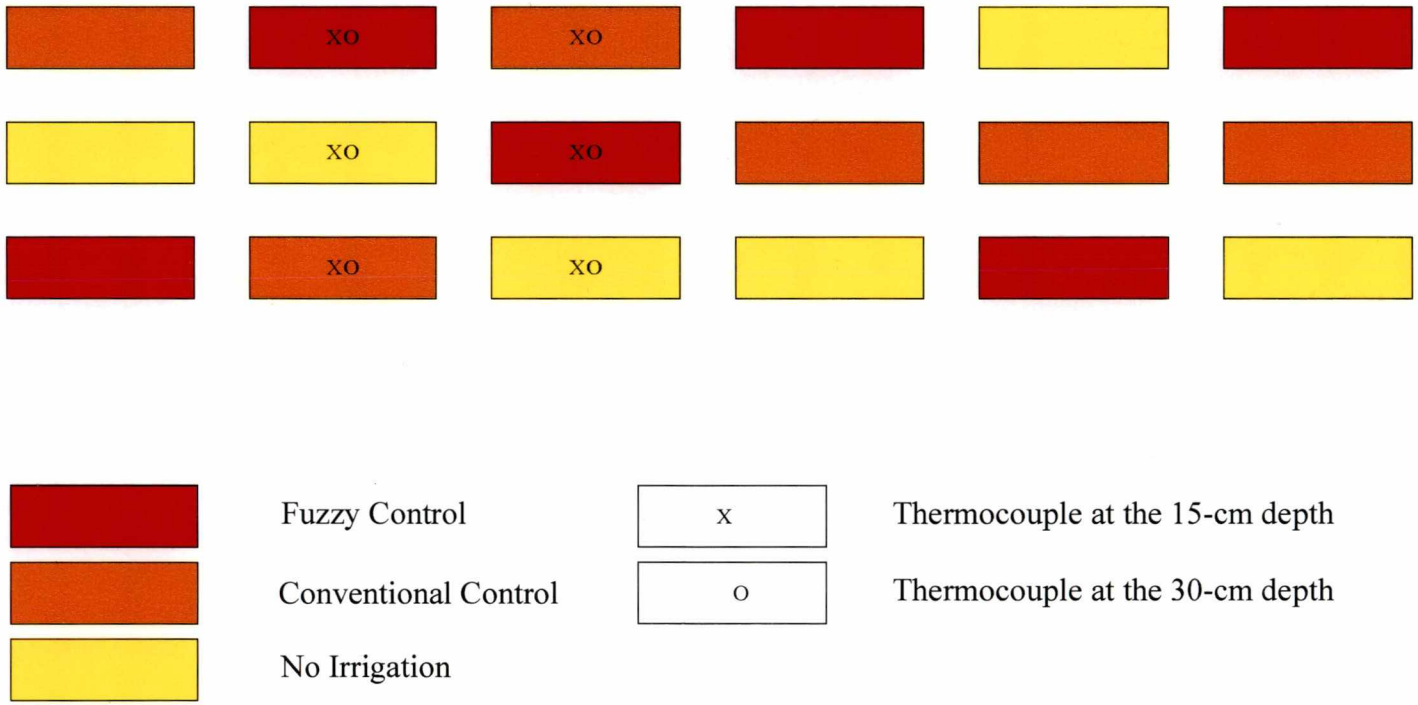


Figure 5.8 Location of the thermocouples installed at six experimental units.

Multiplexer

Two Campbell Scientific® AM416 multiplexers were used in this experiment. A multiplexer basically provides multiple channels for sequential sensor measurements. Twenty-four soil moisture sensors were connected to one multiplexer. The second multiplexer was used to switch twelve soil moisture sensors and twelve thermocouples.

Datalogger

Two sets of input data were required: weather data used to calculate evapotranspiration and soil status data used to infer control actions in the fuzzy irrigation control system. The weather data were collected from the weather station using a Campbell Scientific® CR10 datalogger (Appendix A2). The CR10 was programmed to acquire data every 60 seconds. Readings of solar radiation, relative humidity, air temperature, wind speed, and wind direction were averaged each fifteen minutes. Rain gauge readings were summed each fifteen minutes.

Personal Computer

A desktop computer with a 486 processor (Zenith Data Systems, Herndon, VA), and 16 MB of RAM, was used to run the control system (Appendices A4, A5, A6, A7, and A8). Two serial ports and one parallel port were required for communications among the computer, 2 dataloggers, and a pulse detector. The dataloggers were connected to the serial ports and the missing pulse detector was connected to the parallel port. The missing pulse detector was added to the system as a safety device. Its function was to deactivate the valves if for any reason no signals were received for about 60 seconds.

Figure 5.9a shows a diagram of the fuzzy irrigation control system (FICS), and Figure 5.9b shows a diagram of the conventional irrigation control system (CICS). Figure 5.10 shows a diagram of the hardware used to implement the two systems.

PC Digital Input/Output Board

A digital input/output (I/O) board was used as the interface between the computer and the irrigation control valves. A CIO-DIO24 (ComputerBoards, Inc., Mansfield, MA) was used for this application. The CIO-DIO24 is a single 82C55 digital I/O chip interfaced to the PCbus, with all I/O lines accessible through its 37 D type pin connector. Figure 5.11 shows the CIO-DIO24 board and the missing pulse detector.

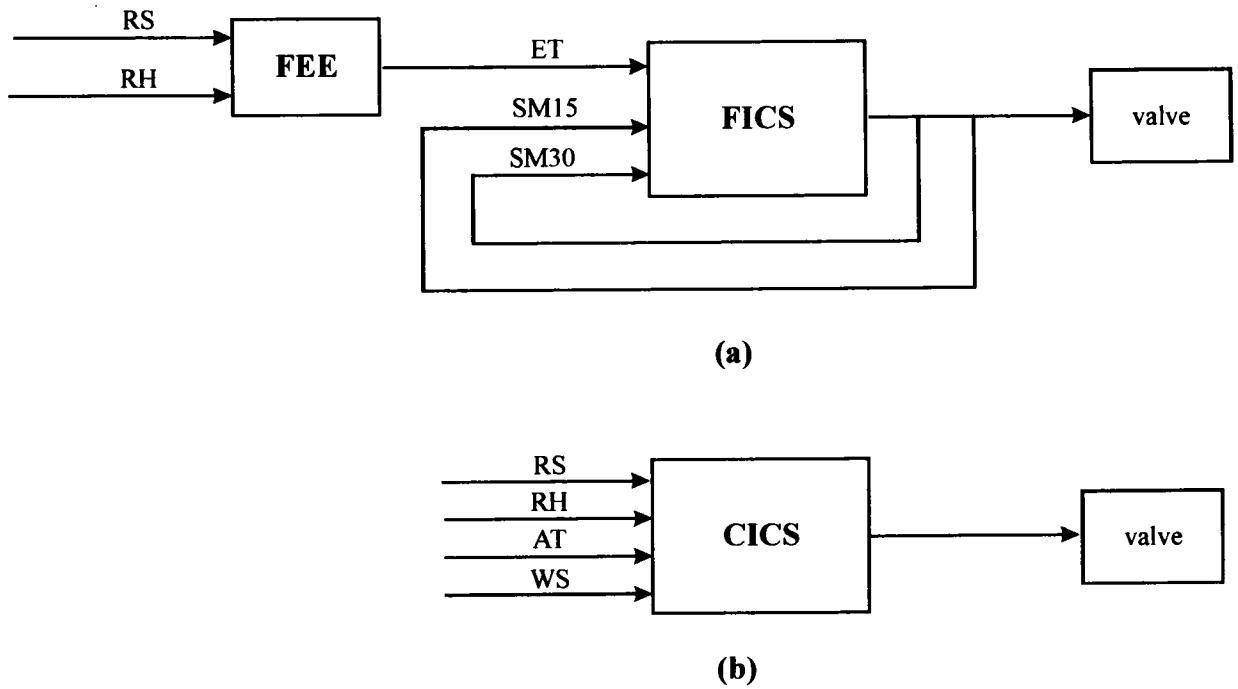
Solenoid Valves

Two Buckner model VP-15, 24 VAC solenoid valves were used for the irrigation control system. The outputs from the fuzzy irrigation control system and from the conventional irrigation control system were the control actions to activate or deactivate each valve.

Control System Operation

Crop Planting

The planting date was delayed by five weeks (June 24, 1997) due to the excessive soil moisture resulting from 218 mm of precipitation that occurred during 17 out of 31 days in May and from 145 mm that occurred during 14 out of 23 days in June.



Legend:

RS = Solar radiation
RH = Relative humidity
ET = Evapotranspiration
SM15 = Soil moisture at the 15-cm depth
SM30 = Soil moisture at the 30-cm depth
AT= Air temperature
WS = Wind speed

Figure 5.9 (a) Diagram of the fuzzy irrigation control system (FICS) including the fuzzy evapotranspiration estimator (FEE). **(b)** Diagram of the conventional irrigation control system (CICS).

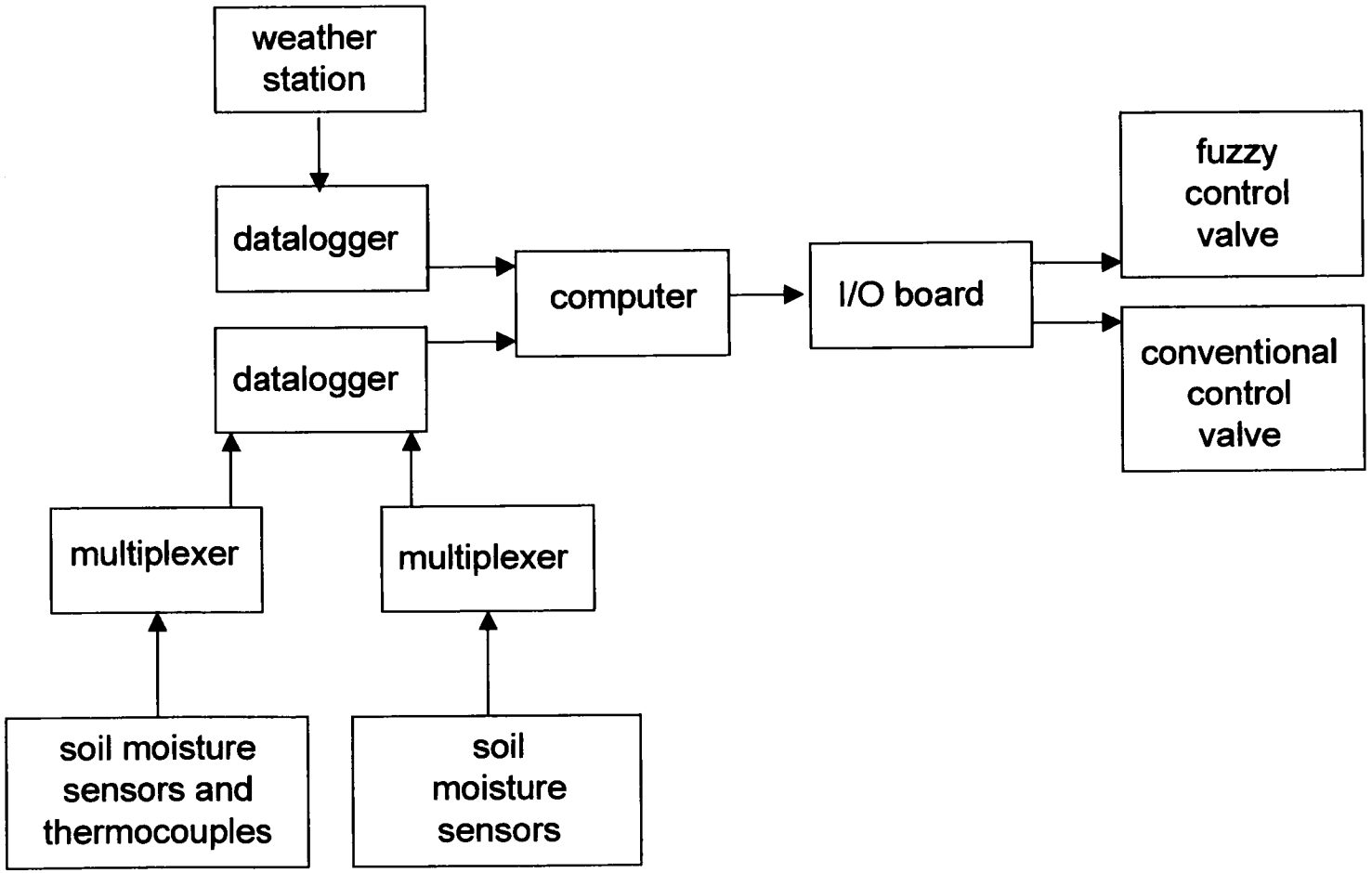


Figure 5.10 Diagram of the hardware used to implement the fuzzy irrigation control system and the conventional irrigation control system.

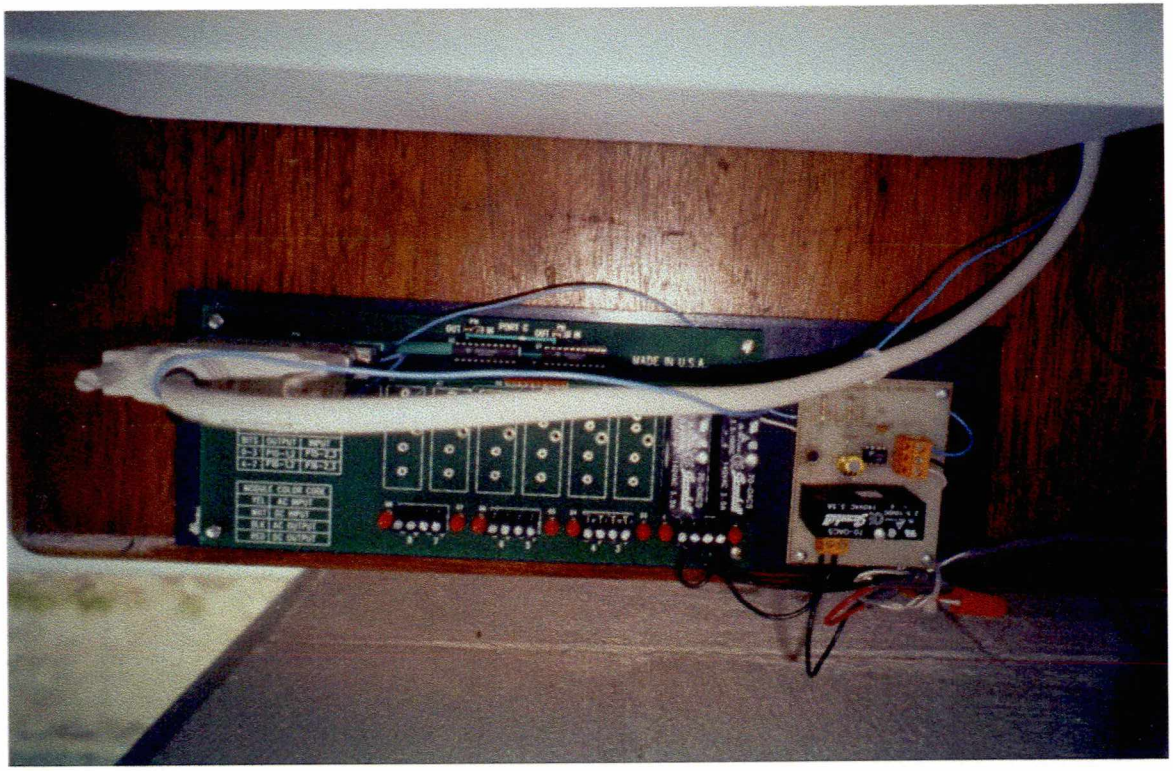


Figure 5.11 PC digital input/output board used in the irrigation control system and the missing pulse detector.

System Installation

The components were assembled during four weeks after planting. The hydraulic components were installed first. Next, the electronic portion of the system was assembled. Lastly, the programming codes were installed (Appendices A4, A5, A6, A7, and A8). Tests were done after each segment was installed. The whole system was started on July 19, 1997. The system operated continuously on a real-time basis, processing the information at fifteen-minute intervals and sending control actions to the valves when required. Input information and control outputs were saved in files for further analysis.

CHAPTER 6

RESULTS AND ANALYSIS

In Chapter 4 designs of a fuzzy evapotranspiration estimator and a fuzzy irrigation control system were presented. A field implementation based on a statistical design to compare the fuzzy system with a conventional system was presented in Chapter 5. In this chapter the results in terms of system performances and statistical analysis are presented.

Fuzzy Evapotranspiration Estimator

Daily lysimeter evapotranspiration measurements and fuzzy evapotranspiration estimations during the operation of the systems are shown in Table 6.1. A comparison between the two methods is presented in Figure 6.1. Lysimeter measurements were taken in thirty-minute intervals and globally summarized on a daily basis. The daily fuzzy estimations were obtained summing 96, fifteen-minute estimations in each day.

The fuzzy evapotranspiration estimator presented values larger than values from lysimeter measurements. On the other hand, the values presented the same trend. The similarity in trends evidences good response from the rule base. The differences in values evidence the need for a larger database and consequent redefinition of the membership functions. Figure 6.2 shows the result of a linear regression performed between the two methods.

Table 6.1 Daily lysimeter ET measurements and fuzzy ET estimations.

Day	Fuzzy ET	Lysimeter	Day	Fuzzy ET	Lysimeter
Julian	mm/d	mm/d	Julian	mm/d	mm/d
200	6.43	5.23	248	6.42	4.08
203	4.65	3.40	249	5.55	3.58
204	4.11	4.31	250	5.59	3.45
205	5.38	4.71	251	4.46	3.05
207	6.61	5.49	252	3.01	1.52
208	5.48	4.58	254	4.41	2.86
213	6.72	4.84	255	5.34	3.39
214	7.36	6.01	256	5.35	3.38
215	6.57	4.93	257	5.11	3.38
216	5.36	4.75	258	5.67	3.62
219	6.52	4.56	259	5.48	3.60
220	4.23	2.75	260	4.45	3.12
221	2.41	0.14	261	5.79	3.92
222	3.56	1.84	262	5.72	3.92
226	5.14	4.18	263	4.25	3.14
227	4.98	4.31	264	4.68	3.53
228	5.86	5.23	265	3.75	2.62
234	6.17	3.24	268	3.35	1.44
235	6.41	3.72	269	5.78	3.66
236	6.81	4.32	270	5.72	3.92
237	4.39	2.48	271	2.91	1.96
238	5.06	3.39	272	5.49	3.40
241	3.92	2.08	273	6.04	3.40
242	5.60	3.26	274	5.42	3.01
243	5.58	3.54	275	5.69	3.01
245	5.29	3.80	279	5.42	3.01
246	2.82	1.87	280	3.99	3.01
247	6.67	3.91	281	4.51	1.83

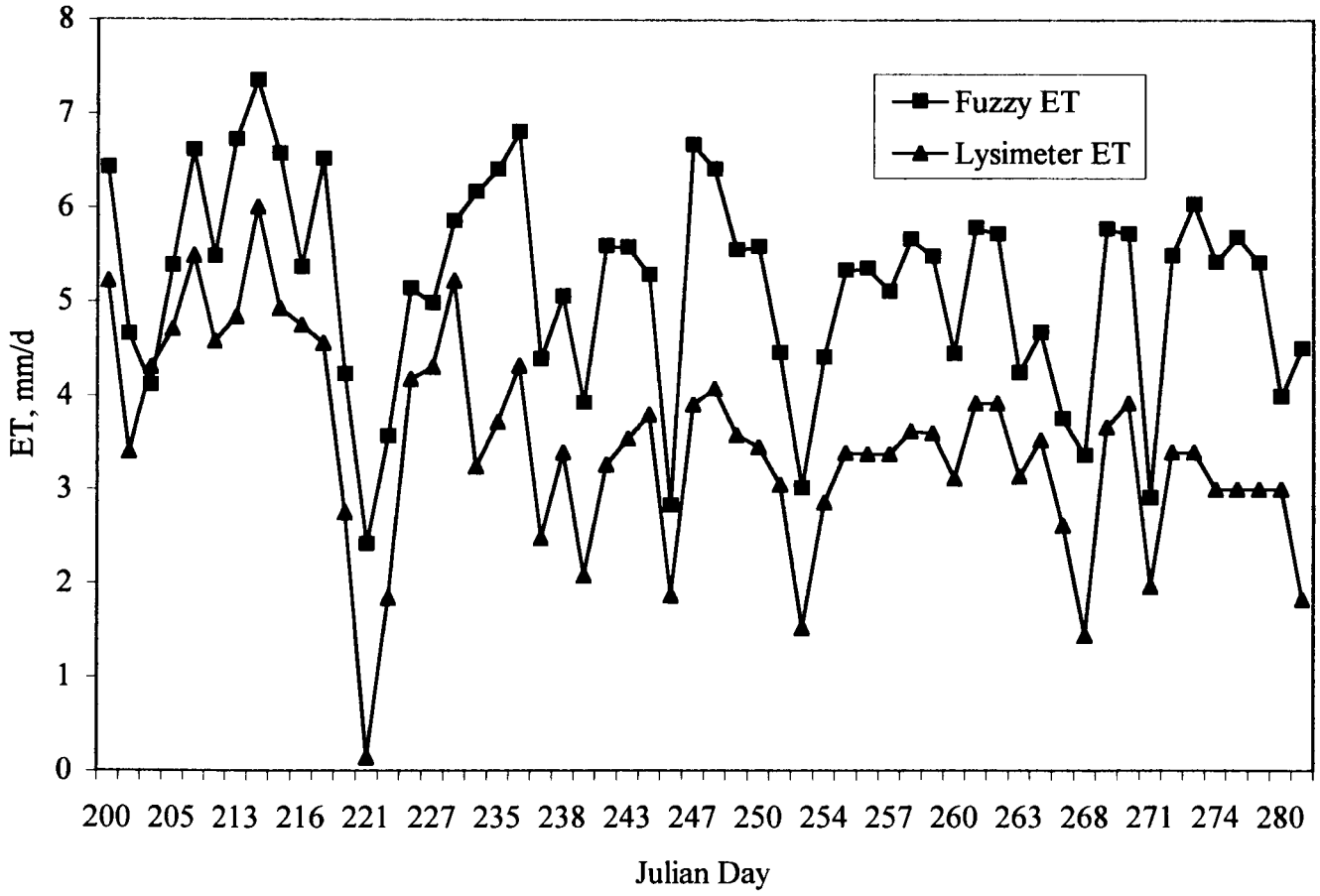


Figure 6.1 Comparison between fuzzy ET estimations and lysimeter ET measurements.

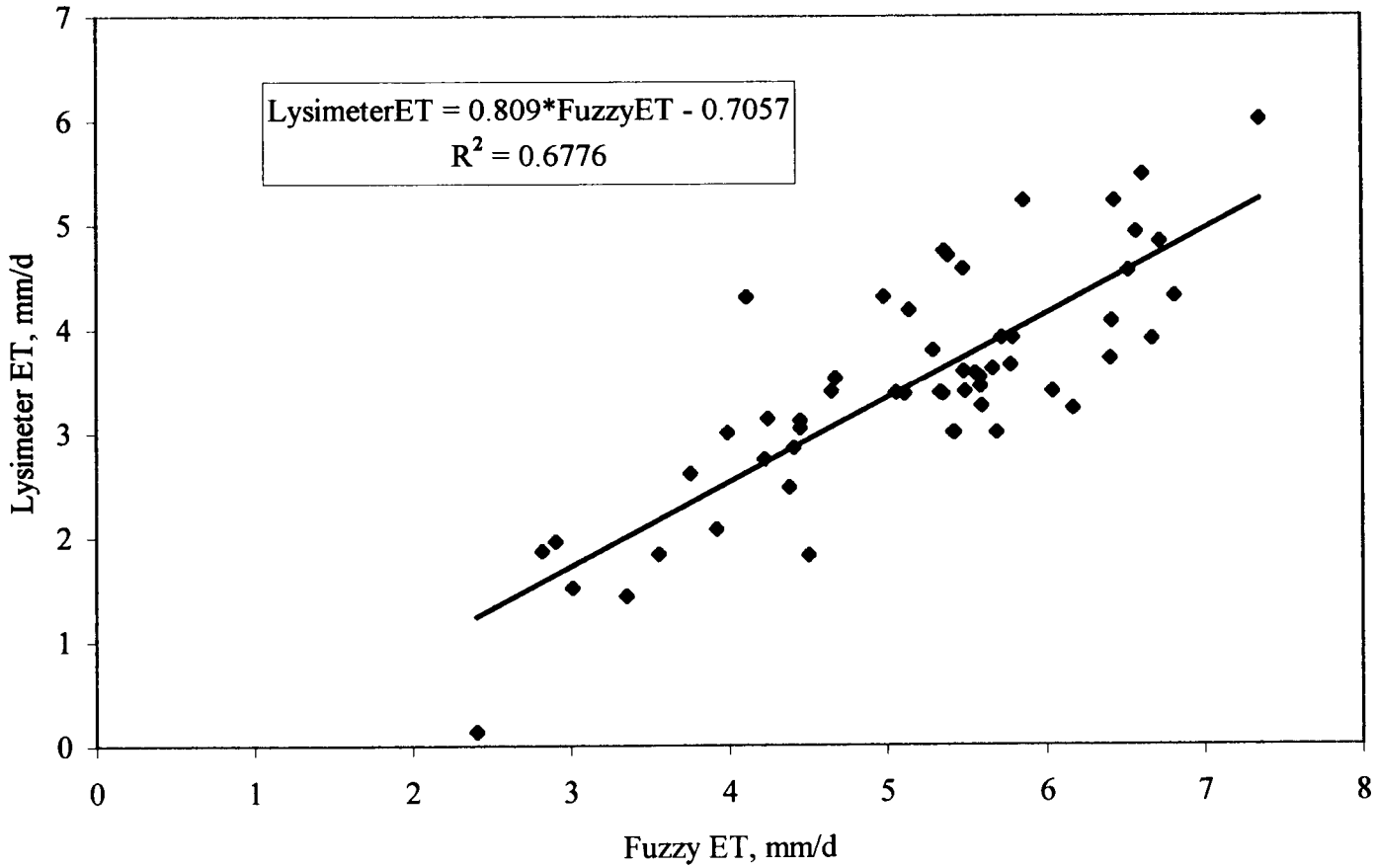


Figure 6.2 Linear regression between fuzzy ET estimations and lysimeter ET measurements.

System Performance

The performances of the systems were analyzed in terms of control of the soil water content and response to changes in weather and soil water content on a real-time basis. Three situations were analyzed:

1. No irrigation.
2. Irrigation under the conventional irrigation control system.
3. Irrigation under the fuzzy irrigation control system.

No Irrigation

Figure 6.3 shows the precipitation during the 82 days of the system operation. This figure facilitates the understanding of the results of the three mentioned situations. Figure 6.4 shows the changes of the soil water potential in the non-irrigated experimental units at the 15-cm depth. The soil water potential varied from –7.5 kPa (wet) to –77 kPa (dry). Figure 6.5 shows the changes of the soil water potential in the non-irrigated experimental units at the 30-cm depth. The soil water potential varied from –7.25 kPa to –50 kPa. In these cases the changing factor was rain. As expected, the crops were submitted to few periods of water scarcity.

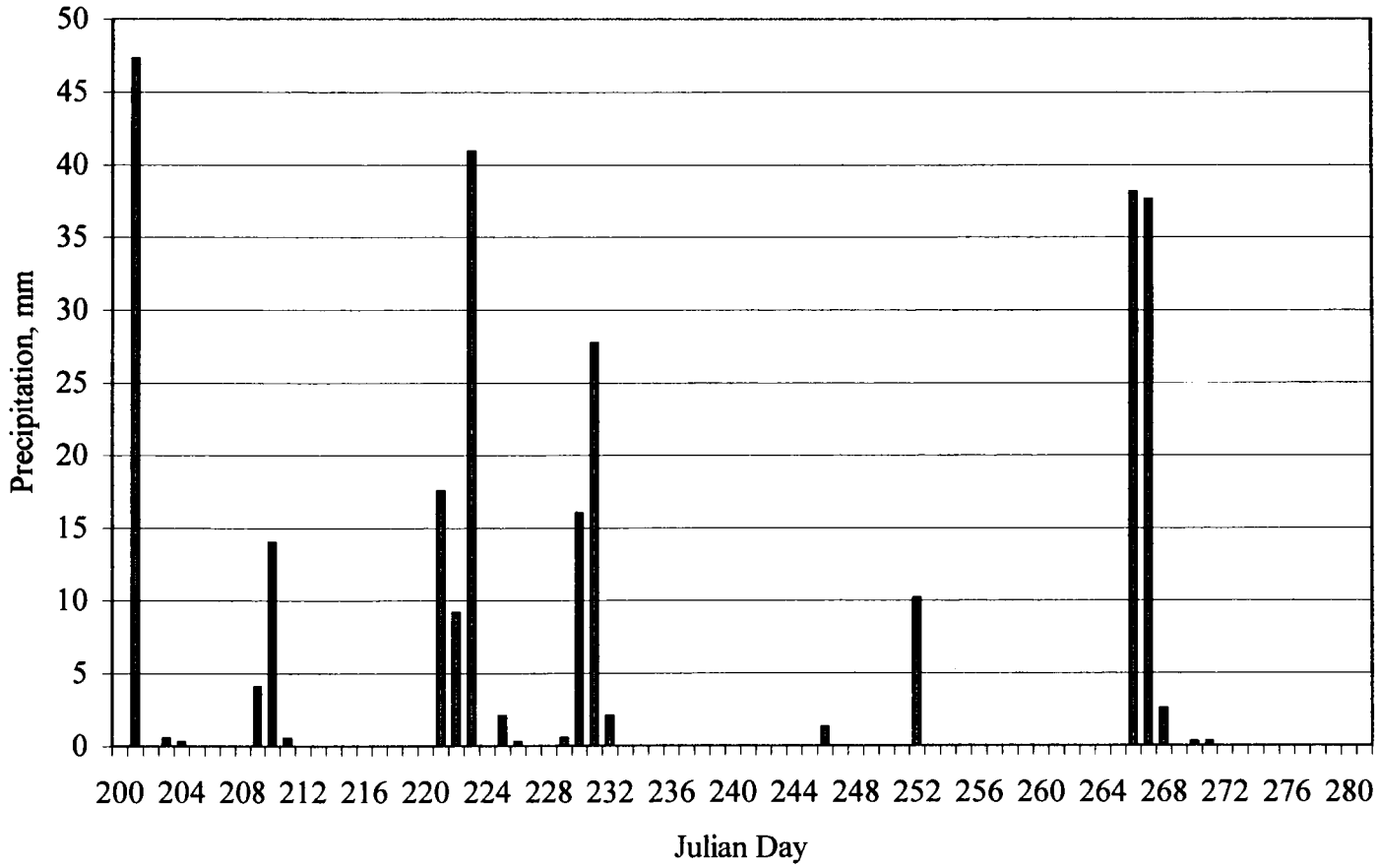


Figure 6.3 Precipitation during system operation.

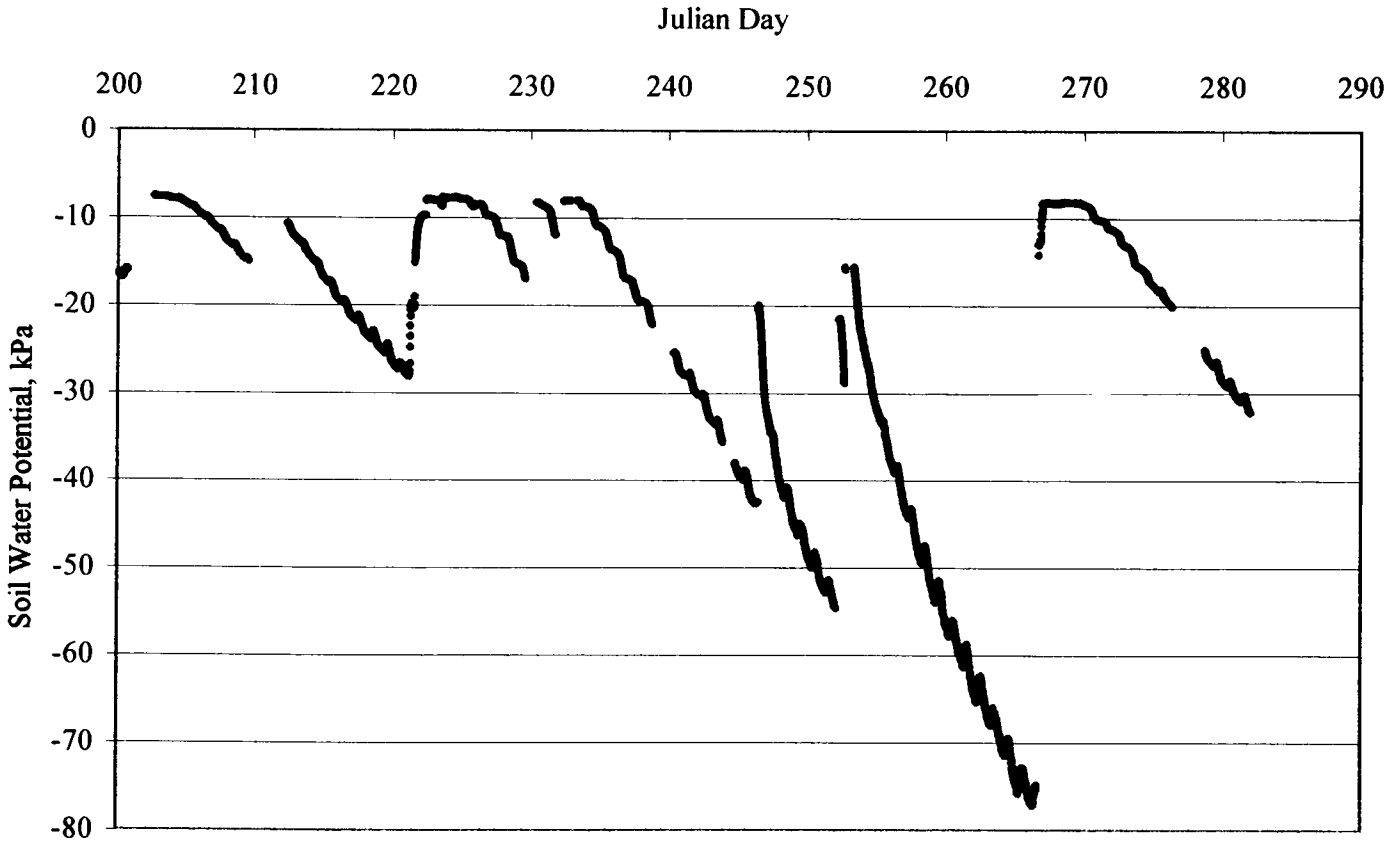


Figure 6.4 Soil water potential (average) in non-irrigated experimental units at the 15-cm depth.

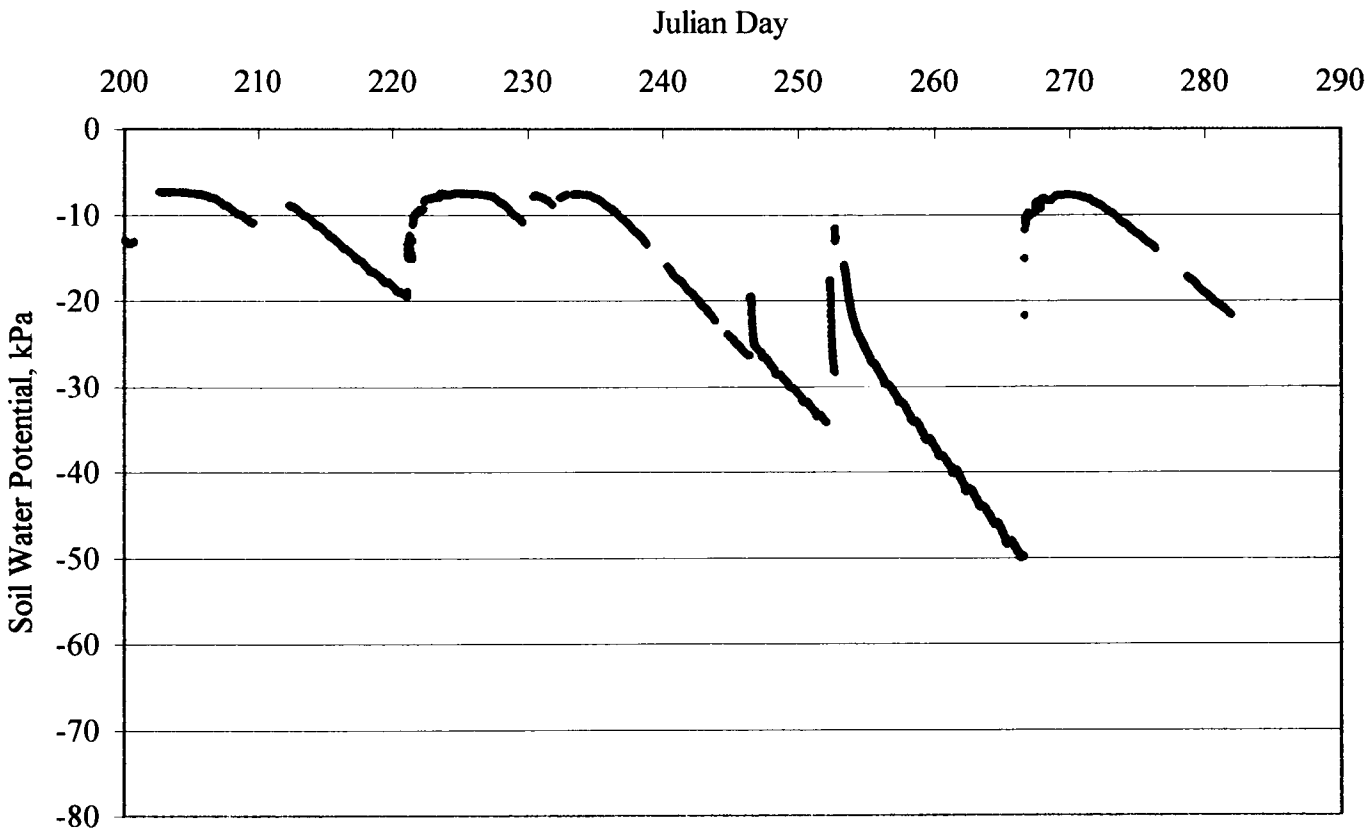


Figure 6.5 Soil water potential (average) in non-irrigated experimental units at the 30-cm depth.

Conventional Irrigation Control System

Figure 6.6 shows the changes in soil water potential at the 15-cm depth in experimental units irrigated using the conventional control. The soil water potential varied from –7 kPa to –30.9 kPa. Figure 6.7 shows the changes in soil water potential at the 30-cm depth in experimental units irrigated using the conventional control. The soil water potential varied from –6.9 kPa to –29.3 kPa.

Fuzzy Irrigation Control System

Figure 6.8 shows the changes in soil water potential at the 15-cm depth in experimental units irrigated using the fuzzy logic control. The soil water potential varied from -6.5 kPa to –14.2 kPa. Figure 6.9 shows the changes in soil water potential at the 30-cm depth in experimental units irrigated using the fuzzy control. The soil water potential varied from –7 kPa to –15.7 kPa.

System Performance Analysis

The fuzzy irrigation control system worked as expected in response to the climatic and soil moisture variations. The soil water potential was kept within a very small range (-6.5 kPa to -15.7 kPa). This range is very appropriate to plant growth and development.

The conventional irrigation control system also worked properly. The soil water potential was also kept within a small range (-6.9 kPa to –30.9 kPa), although the variation (24 kPa) was very large when compared to the variation presented by the FICS (9.2 kPa). This larger variation is probably related to the absence of feedback.

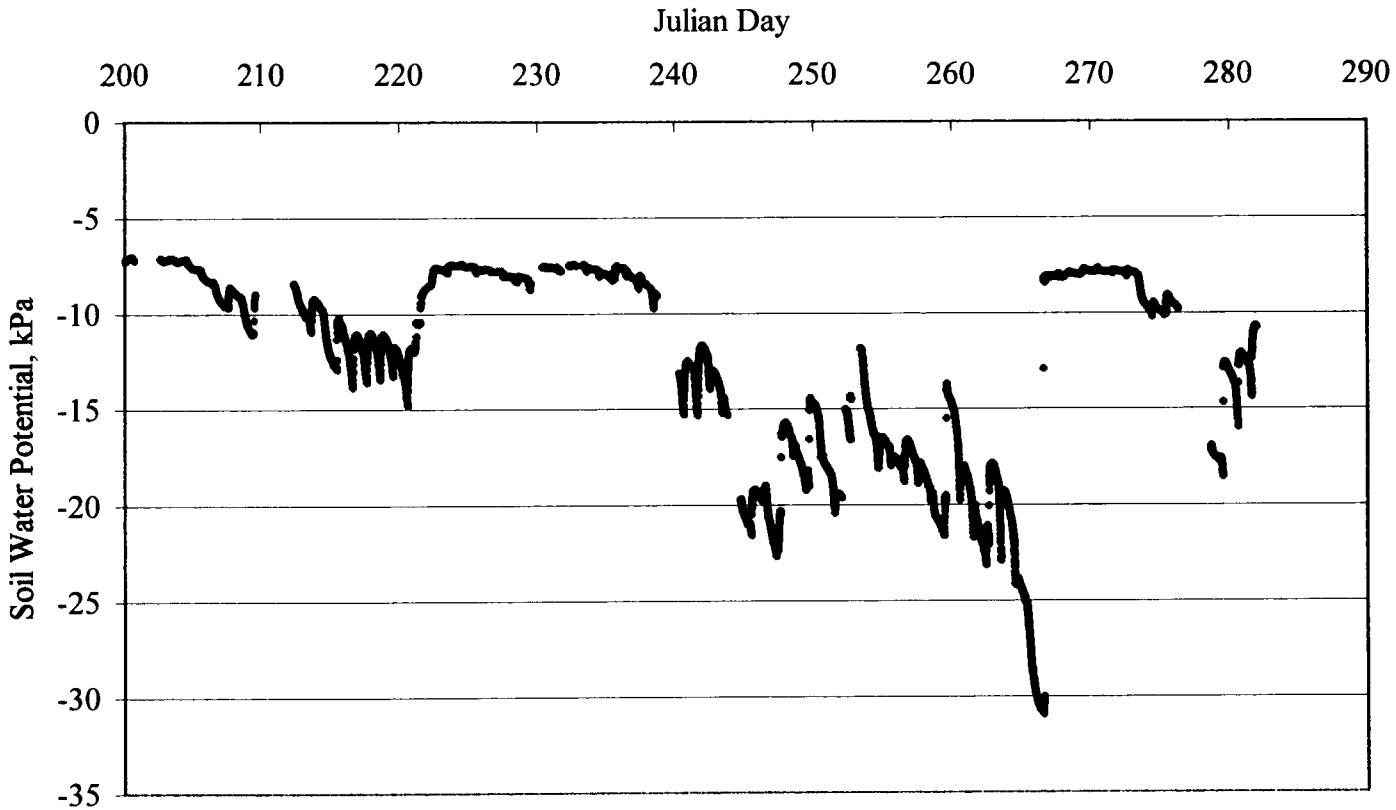


Figure 6.6 Soil water potential (average) at the 15-cm depth in experimental units using the conventional control.

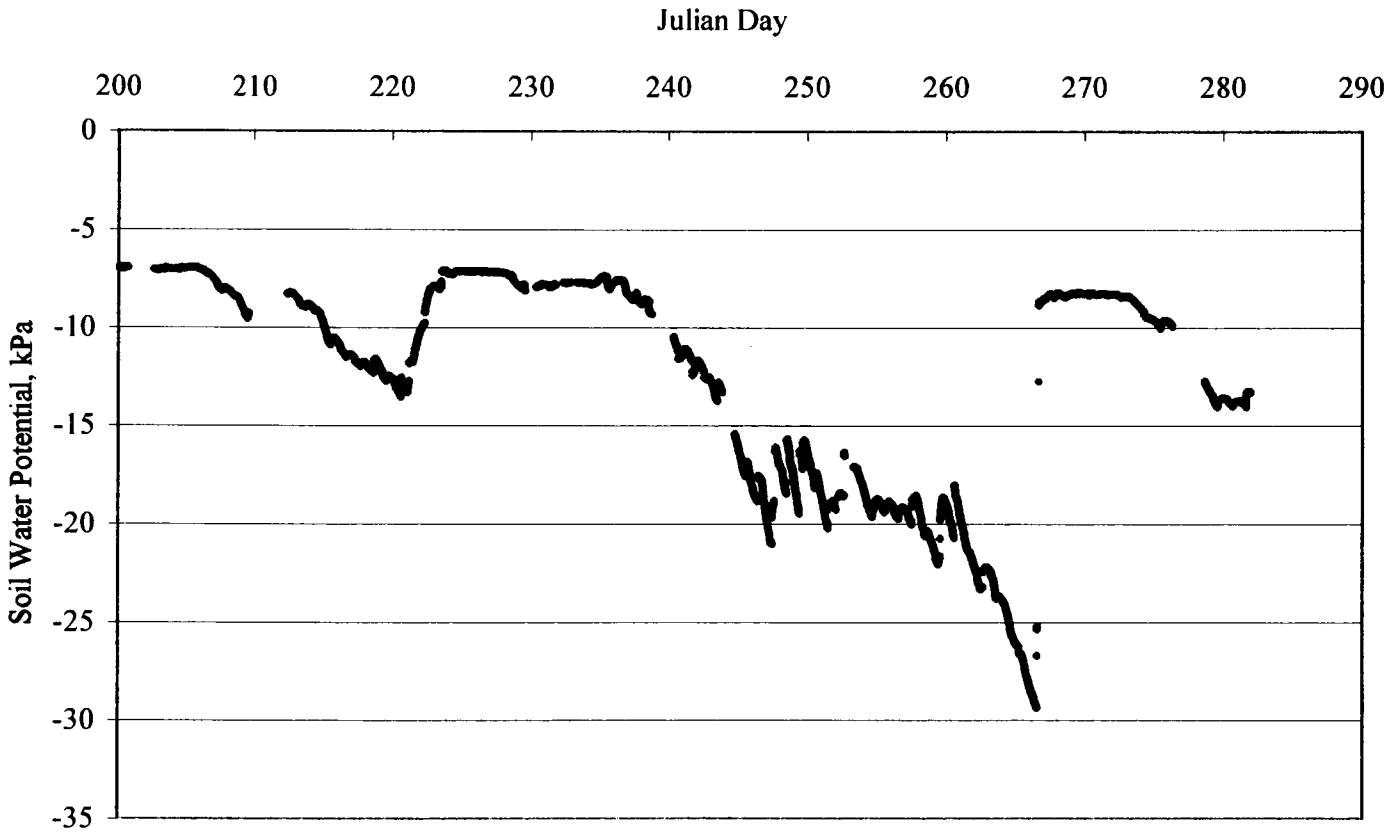


Figure 6.7 Soil water potential (average) at the 30-cm depth in experimental units using the conventional control.

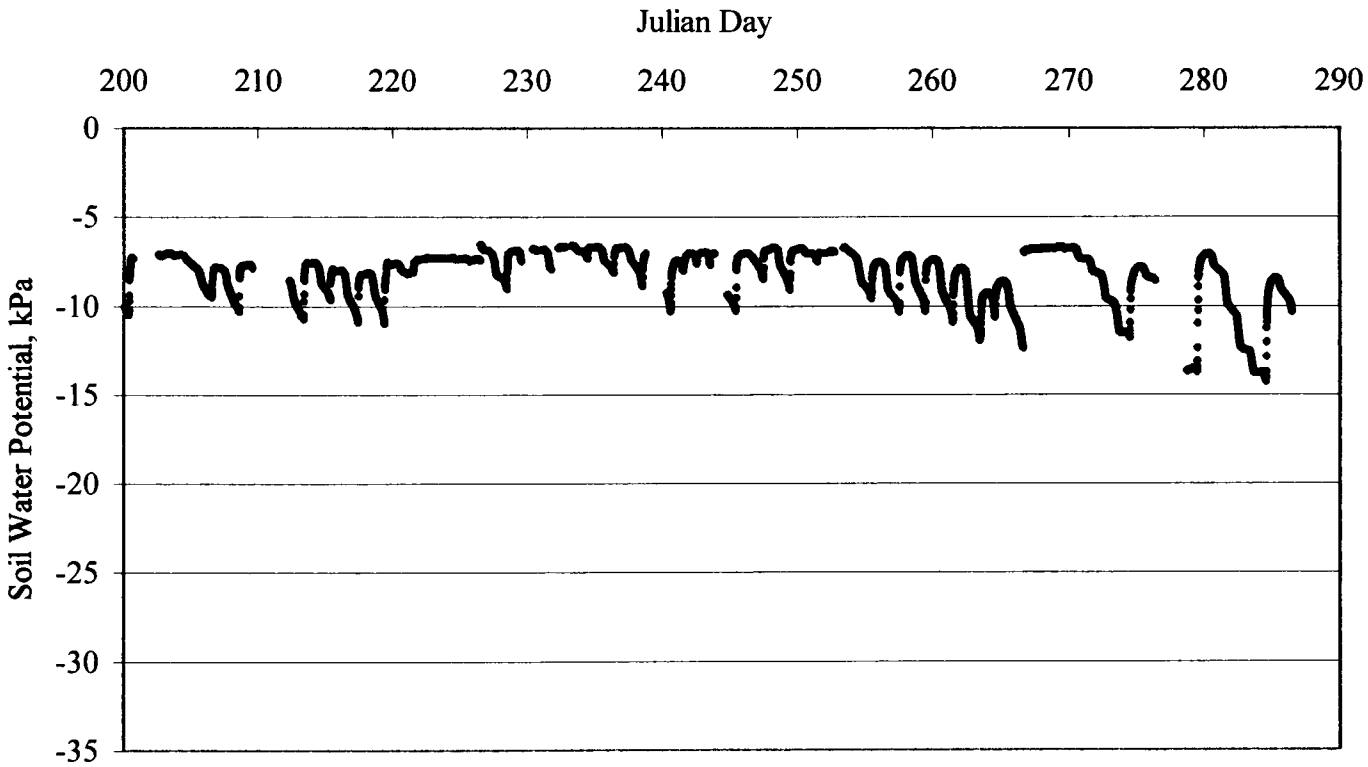


Figure 6.8 Soil water potential (average) at the 15-cm depth in experimental units using the fuzzy logic control.

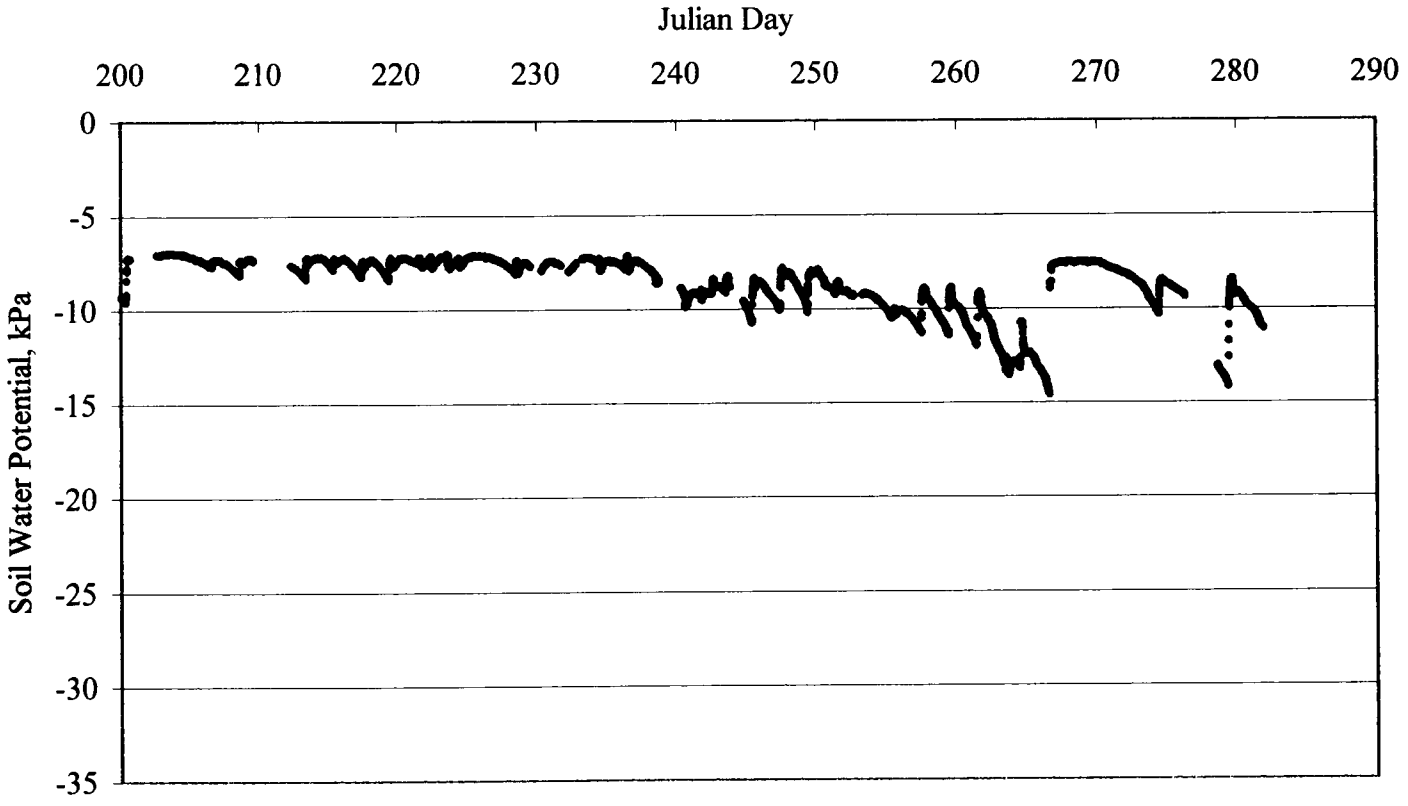


Figure 6.9 Soil water potential (average) at the 30-cm depth in experimental units using the fuzzy logic control.

Irrigation Requirements

The water requirement for peppers ranges from 380 mm to 500 mm. The total measured evapotranspiration during the crop season reached approximately 400 mm. The measured precipitation from planting to harvest summed 393 mm. Effective precipitation based on the Soil Conservation Service method was estimated as 250 mm. Therefore, there was a need for about 130 mm to 250 mm of supplementary irrigation.

The FICS added about 200 mm of irrigation water during its period of operation. The CICS released an amount of 180 mm of water in the same period. The average irrigation interval was 40 hours for both systems.

Statistical Analysis

A statistical analysis was performed to verify the performance of the systems related to crop yield. A SAS® code using the *general linear models procedure* was written to verify statistically significant differences among the three treatments (Appendix B1). Specifically, the parameter *estimate* was used to test differences between:

1. The *fuzzy irrigation control system* and *no irrigation.*
2. The *conventional irrigation control system* and *no irrigation.*
3. The *fuzzy irrigation control system* and the *conventional irrigation control system.*
4. Both *fuzzy/conventional irrigation control systems* and *no irrigation.*

The yield samples were collected from the central portion (6.1 m) of the intermediate plot of each experimental unit, and weighed on site. Figure 6.10 shows the yields obtained from each experimental unit. The *fuzzy irrigation control system* yields varied

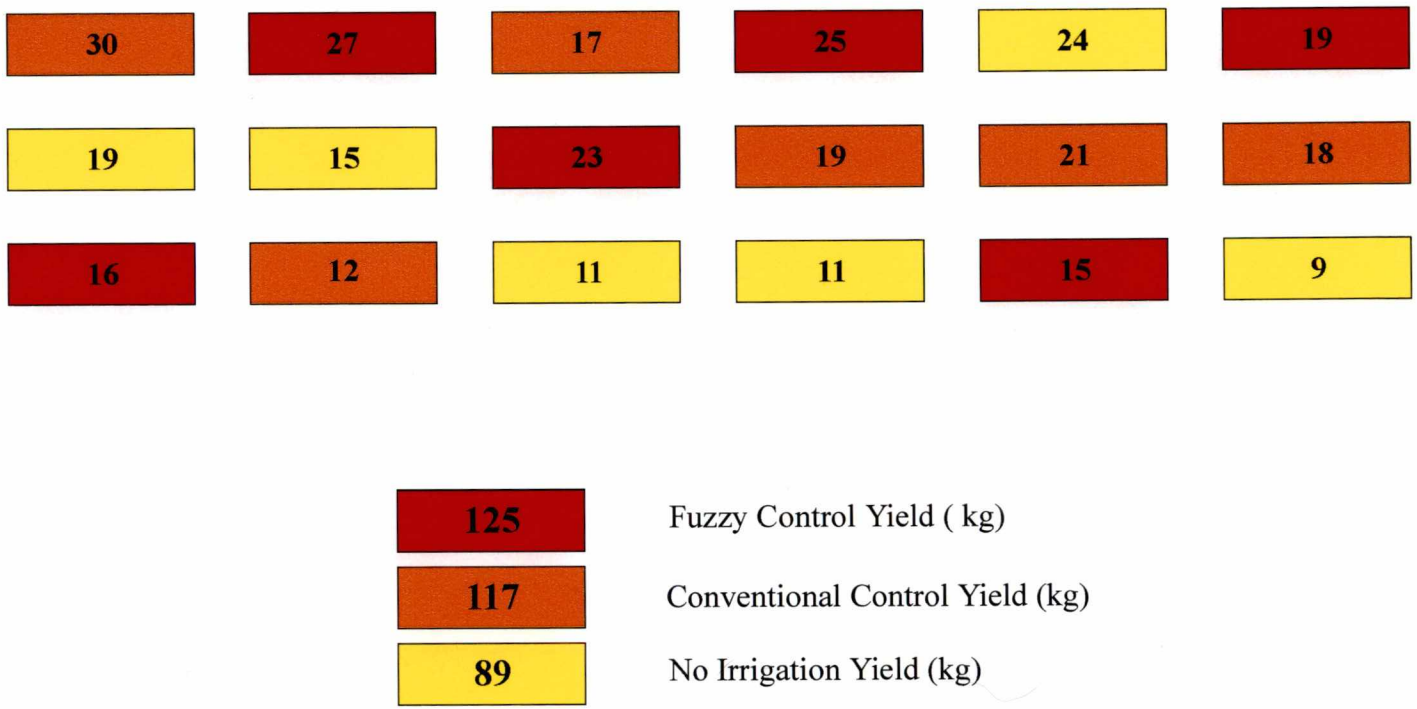


Figure 6.10 Randomized block layout showing the yields obtained from each experimental unit.

from 15 kg to 27 kg. The *conventional irrigation control system* yields varied from 12 kg to 30 kg. *No irrigation treatment* yields varied from 9 kg to 24.

The *fuzzy irrigation control system* presented the lesser spread in sample yields. Although a trend in yield from the different treatments was noticed, the statistical results showed that yields from the *fuzzy irrigation control system* and *no irrigation treatments* were not different ($p > 0.13$). Yields from the *conventional irrigation control system* and *no irrigation treatments* also did not present significant difference ($p > 0.23$). The yields from *fuzzy irrigation control system* and the *conventional irrigation control system* were not significantly different ($p > 0.70$). The result from the test comparing the *fuzzy / conventional systems* with *no irrigation* did not present significant difference ($p > 0.12$).

Fuzzy Irrigation Control System Output Unit Conversion

The conversion of control value outputs (0 to 10) to soil water potential (kPa) at the two depths of the soil allows more flexibility in control specifications, i.e., control decisions can be made based on output values related to an actual *state variable*. A regression analysis was performed relating control value outputs and soil water potential at two depths. The control value data consisted of the outputs from the FICS, the CICS, and from the *no irrigation* treatment, on a fifteen-minute basis.

The soil water potential data consisted of the actual measurements obtained at each fifteen-minute interval. Two hundred pairs were analyzed for each soil depth. Figure 6.11 shows the comparison between control value output and soil water potential at the 15-cm depth. Figure 6.12 presents the comparison between control value output and soil water potential at the 30-cm depth.

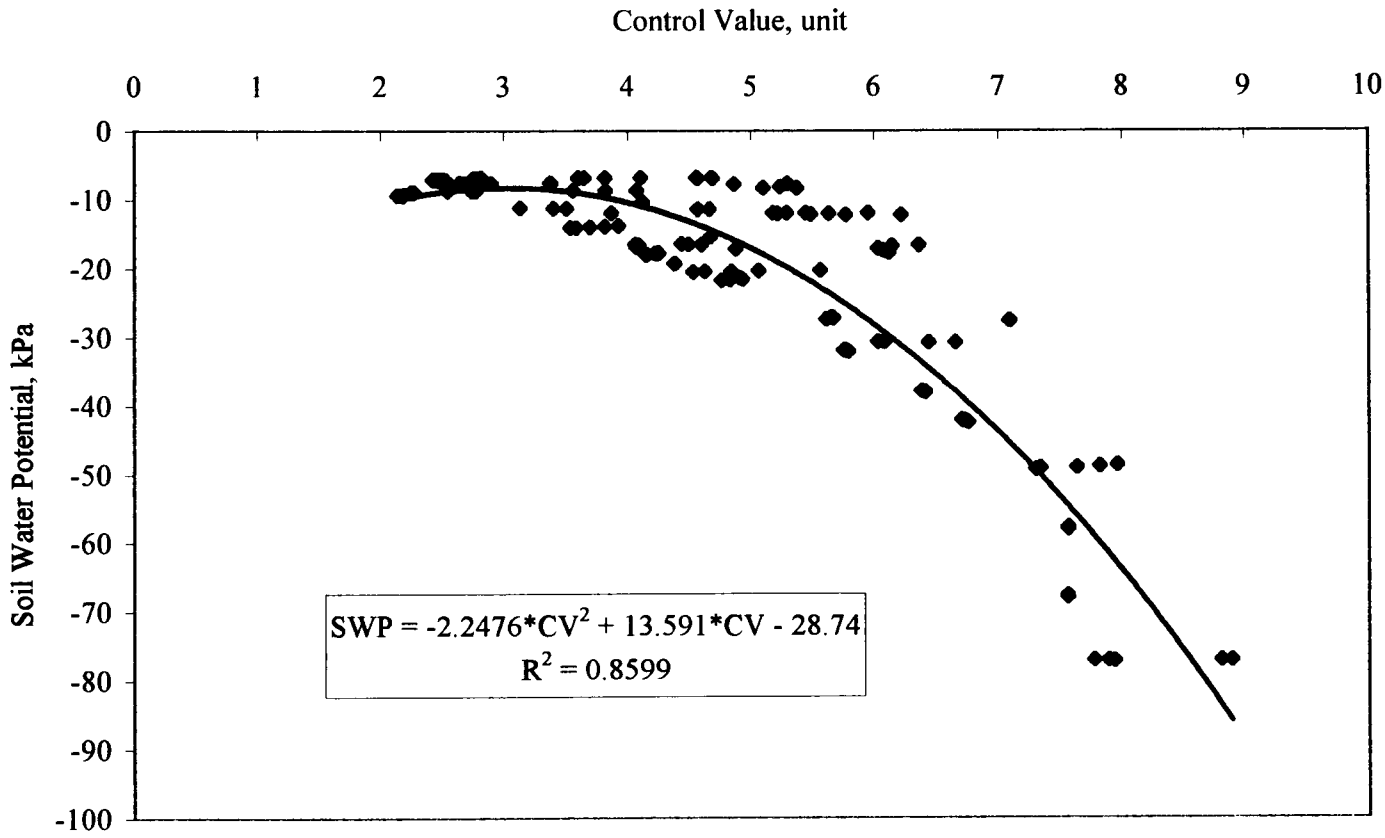


Figure 6.11 Relationship between control value output and soil water potential at the 15-cm depth.

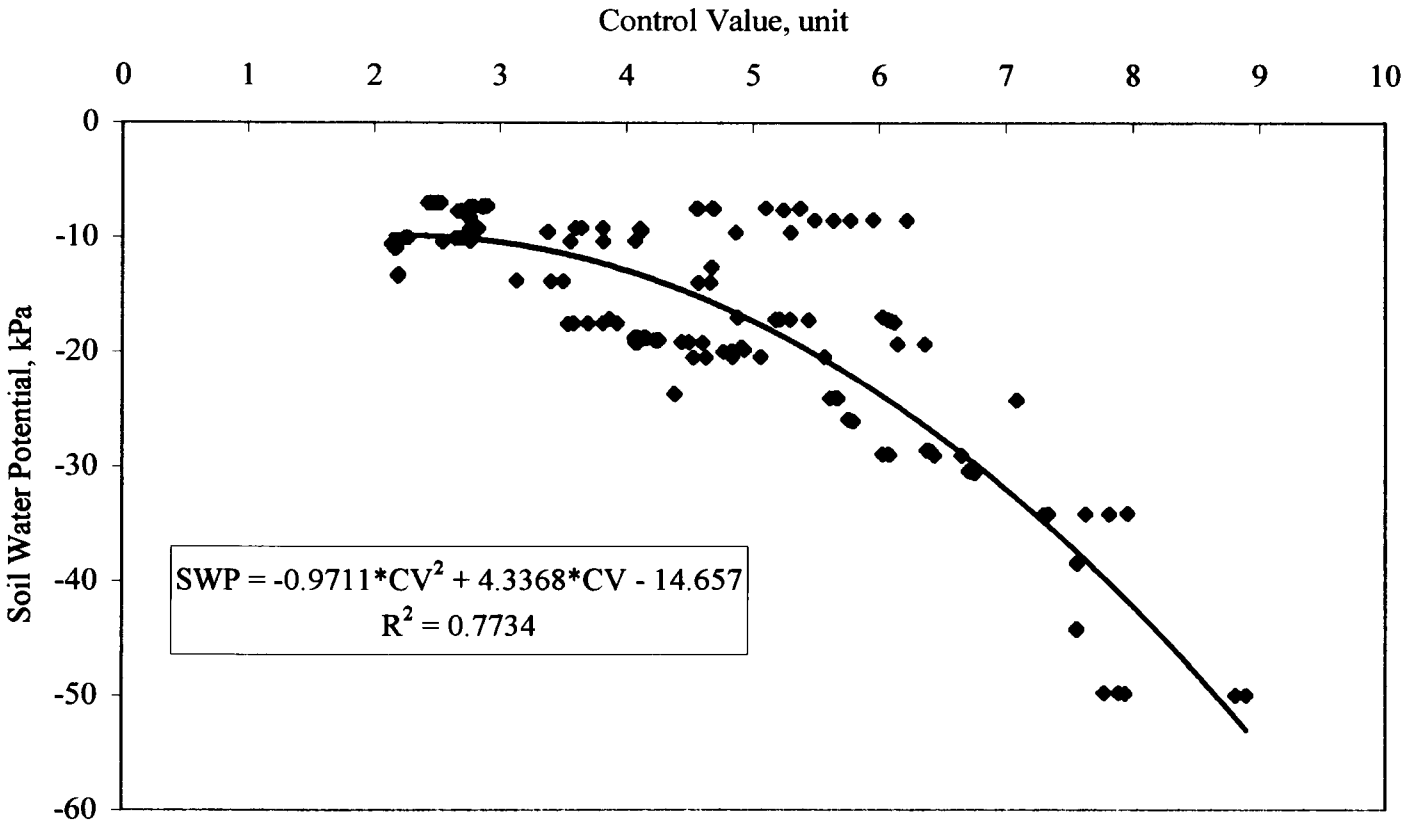


Figure 6.12 Relationship between control value output and soil water potential at the 30-cm depth.

CHAPTER 7

SYSTEM OPTIMIZATION USING NEURAL NETWORKS

In Chapter 6 it was shown that the fuzzy evapotranspiration estimator presented an offset with larger ET outputs when compared to ET lysimeter measurements. Since it presented the same trend, this difference can be eliminated by redefining the membership functions to bring output values closer to lysimeter ET while maintaining the same rule base. This enhancement can be achieved by applying the adaptive neuro-fuzzy inference system method (ANFIS) to the database obtained during the system operation. Such a procedure is presented in this chapter.

ANFIS Procedure

Fundamentally, ANFIS tunes a fuzzy inference system with a backpropagation algorithm based on collection of input-output data. This allows the fuzzy system to learn. A network structure facilitates the computation of the gradient vector for parameters in a fuzzy inference system. Once the gradient vector is obtained, an optimization routine is applied to reduce an error measure, usually defined by the sum of the squared difference between actual and desired outputs. This process is called learning by example in the artificial neural networks literature.

Since ANFIS is more complex than basic fuzzy inference systems, it only supports first order *Sugeno* systems with a single output derived by weighted average defuzzification, and unity weight for each rule. *Mamdani* systems such as the fuzzy

evapotranspiration estimator developed in this research are not supported. However, the basic difference between *Mamdani* and *Sugeno* systems is in the defuzzification process. Therefore, we can use the basic fuzzy evapotranspiration estimator design to generate a Sugeno-type system to match ANFIS requirements.

Fuzzy Evapotranspiration Estimator

Training Data

The ANFIS learning procedure requires a *training data set* that contains desired input/output data of the target system to be modeled. An optional *checking data set* can check the generalization capability of the resulting fuzzy inference system. Usually these data sets are collected based on observations of the target system.

The input data consisted of solar radiation and relative humidity measurements during the system operation on a fifteen-minute basis. The desired output data was obtained by taking the fifteen-minute ET estimations correspondent to each input set and multiplying each of them by a factor. This factor was the ratio between daily values of ET lysimeter measurements and the sum of ninety-six, fifteen-minute ET outputs from the fuzzy estimator for each day as expressed below

$$output_{desired} = output_{actual} * \frac{measured\_ET_{daily}}{estimated\_ET_{daily}} \tag{7.1}$$

A total of 5064 input/output pairs were divided for training and checking data sets. The training data set and the checking data set were each composed of 2534 non-consecutive data pairs.

Training Fuzzy System

A Sugeno-type fuzzy system was generated using the command *genfis1* from MATLAB® Fuzzy Logic Toolbox. The same parameters used in the fuzzy estimator design were used to generate this system (5 triangular membership functions were assigned for each input). The system presented the same 25 rules.

ANFIS Training

The training was started and specified to stop after 40 learning epochs (Appendix C1). The optimized system was chosen at the epoch that presented the minimal root mean squared error from the checking training data. Figure 7.1 presents the modified membership functions for the input variable *radi*. Figure 7.2 shows the modified membership functions for the input variable *humi*. Table 7.1 presents the constants (c_0, c_1, and c_2) for each rule of the *first-order Sugeno* system for the optimized fuzzy evapotranspiration estimator output as given by

$$et(radi,humi) = c_0 + c_1 * radi + c_2 * humi \tag{7.2}$$

Simulation

Using the actual inputs obtained during the system operation, a simulation was performed applying the optimized fuzzy evapotranspiration estimator. The daily sum of ET outputs resulting from simulation and daily lysimeter ET measurements are shown in Table 7.2. A comparison between the two methods is presented in Figure 7.3. A linear regression comparing the two methods was performed and is shown as Figure 7.4.

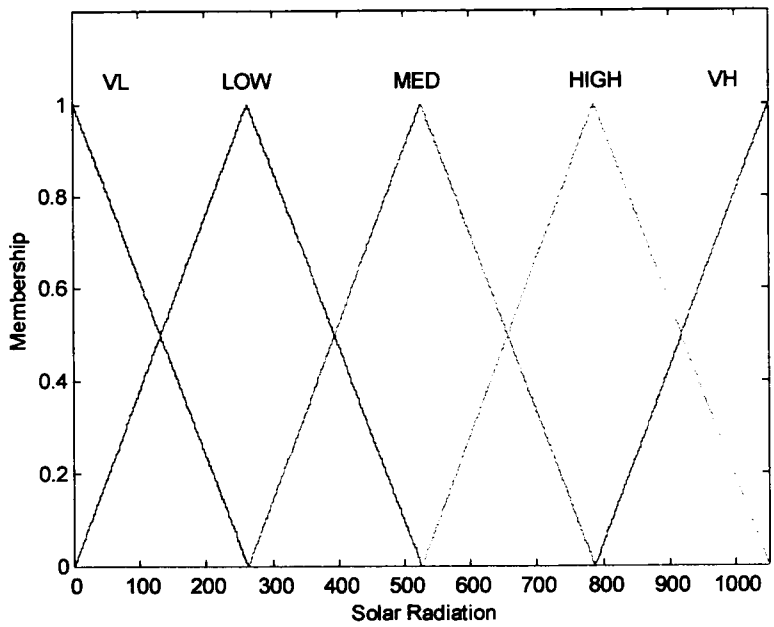


Figure 7.1 Membership functions for the input variable *radi*, after optimization.

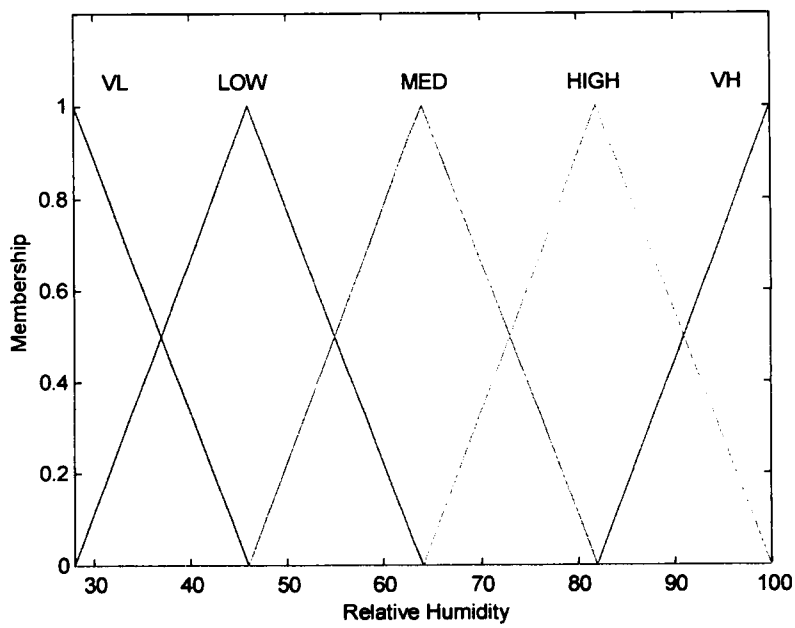


Figure 7.2 Membership functions for the input variable *humi*, after optimization.

Table 7.1 *First-order Sugeno* output constants for the optimized ET estimator.

Rule	c_0	c_1	c_2
R4.1	-0.1253	-0.0272	0.08223
R4.2	-1.255	-0.0003778	0.03921
R4.3	-1.098	-0.000593	0.02853
R4.4	-1.065	-0.0001425	0.01692
R4.5	2.195	0.008765	-0.1305
R4.6	0.1587	0.0003224	-0.01086
R4.7	1.193	-0.0006693	-0.01179
R4.8	1.182	-0.0005104	-0.01024
R4.9	0.961	-0.0004797	-0.00564
R4.10	-2.144	0.002643	0.001917
R4.11	0.007209	0.0003188	-0.003523
R4.12	0.5107	-0.0006442	-0.001759
R4.13	0.3336	-0.0004128	-0.0005669
R4.14	0.5582	-0.0005413	-0.002351
R4.15	-0.9105	0.002451	-0.003107
R4.16	0.0109	0.0003596	-0.001356
R4.17	0.4171	-0.0008261	-0.002865
R4.18	0.1795	-0.0002513	-0.001208
R4.19	0.2619	-0.000519	-0.001037
R4.20	-0.5262	0.002566	-0.001133
R4.21	0.2136	0.000832	-0.005792
R4.22	0.04481	-0.000786	-0.00001639
R4.23	-0.06721	0.00004549	0.001016
R4.24	-0.03299	-0.0002735	0.0003942
R4.25	-0.01105	0.002849	0.00009385

Table 7.2 Lysimeter ET measurements and fuzzy ET estimations, after optimization.

Julian Day	Fuzzy ET Mm/d	Lysimeter Mm/d	Julian Day	Fuzzy ET Mm/d	Lysimeter Mm/d
200	4.34	5.23	**248**	4.27	4.08
203	3.19	3.40	**249**	3.61	3.58
204	3.06	4.31	**250**	3.72	3.45
205	3.81	4.71	**251**	2.98	3.05
207	4.54	5.49	**252**	1.91	1.52
208	3.79	4.58	**254**	3.04	2.86
213	4.54	4.84	**255**	3.55	3.39
214	5.08	6.01	**256**	3.53	3.38
215	4.50	4.93	**257**	3.40	3.38
216	3.71	4.75	**258**	3.66	3.62
219	4.46	4.56	**259**	3.60	3.60
220	2.90	2.75	**260**	2.95	3.12
222	2.31	1.84	**261**	3.79	3.92
226	3.43	4.18	**262**	3.75	3.92
227	3.55	4.31	**263**	2.88	3.14
228	4.30	5.23	**264**	3.17	3.53
234	4.18	3.24	**265**	2.61	2.62
235	4.35	3.72	**268**	2.07	1.44
236	4.53	4.32	**269**	3.84	3.66
237	2.96	2.48	**270**	3.79	3.92
238	3.57	3.39	**271**	1.91	1.96
241	2.66	2.08	**272**	3.60	3.40
242	3.88	3.26	**273**	3.88	3.40
243	3.81	3.54	**274**	3.53	3.01
245	3.50	3.80	**275**	3.63	3.01
246	1.86	1.87	**279**	3.46	3.01
247	4.29	3.91	**281**	2.93	1.83

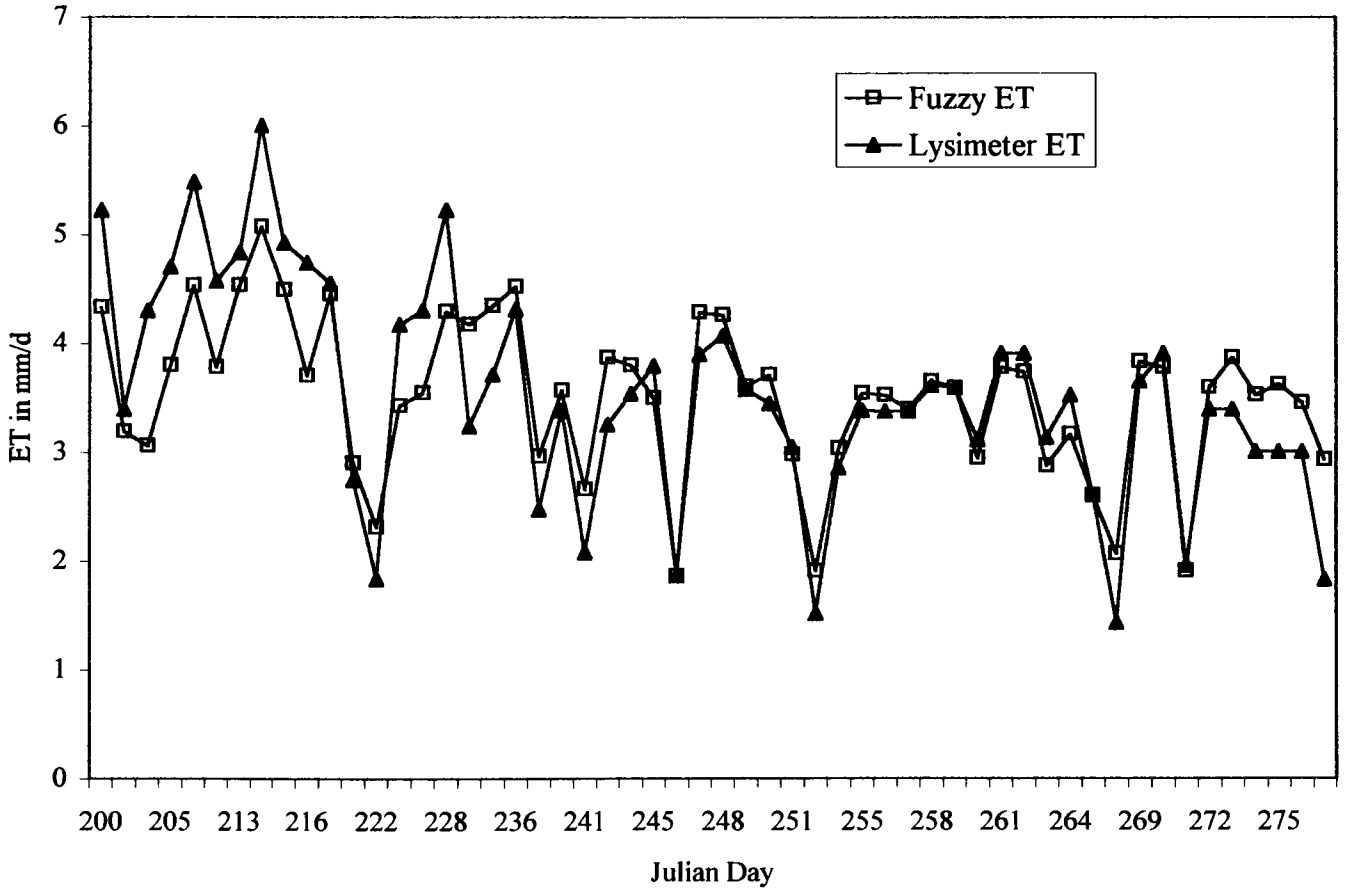


Figure 7.3 Comparison between fuzzy ET estimations and lysimeter ET measurements, after optimization.

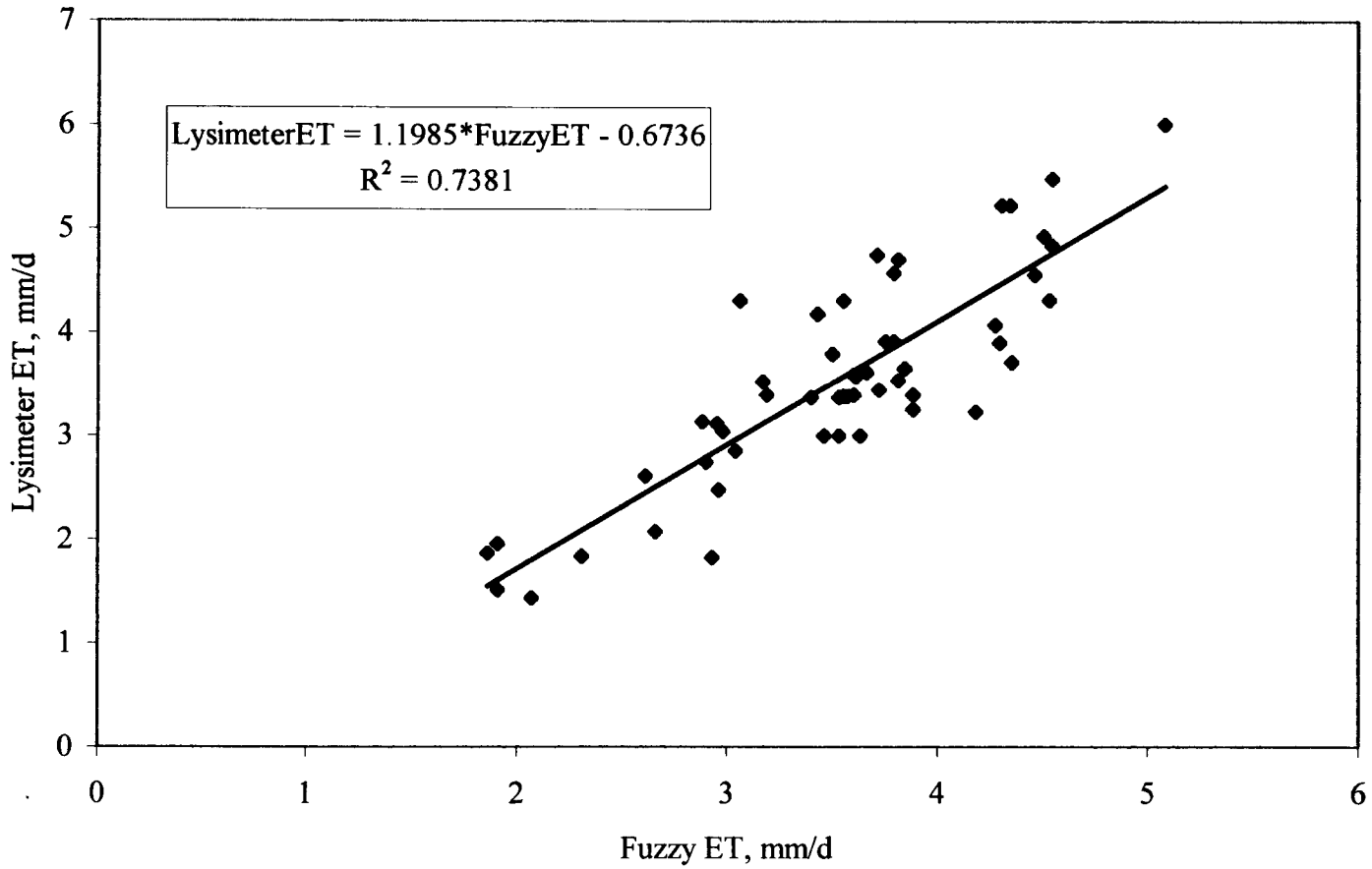


Figure 7.4 Linear regression between fuzzy ET estimations and lysimeter ET measurements, after optimization.

Optimized fuzzy ET outputs were closer ($R^2 = 0.7381$) to lysimeter ET measurements when compared to fuzzy ET outputs obtained before the optimization process ($R^2 = 0.6776$). The offset was practically eliminated, representing a significant improvement on the fuzzy ET estimator.

Evapotranspiration Estimation and Solar Radiation

A linear regression between daily solar radiation sensor measurements and daily lysimeter ET measurements ($R^2 = 0.8308$) taken during the operation of the system performed very well when compared to the FEE, as shown in Figure 7.5. This result was expected since an earlier regression analysis for data taken in 1994 (Figure 4.1), presented an even higher coefficient of determination ($R^2 = 0.9074$). This high relationship between solar radiation and evapotranspiration is probably due to the characteristic humid climate of the Cumberland Plateau associated to a relatively low wind speed during the crop season. However, it is necessary to have measurements taken on a fifteen-minute basis to construct the relationship that could be used to substitute the FEE as the ET estimator within the FICS. Figure 7.6 shows a comparison between daily solar radiation measurements and daily lysimeter ET measurements taken during the operation of the system.

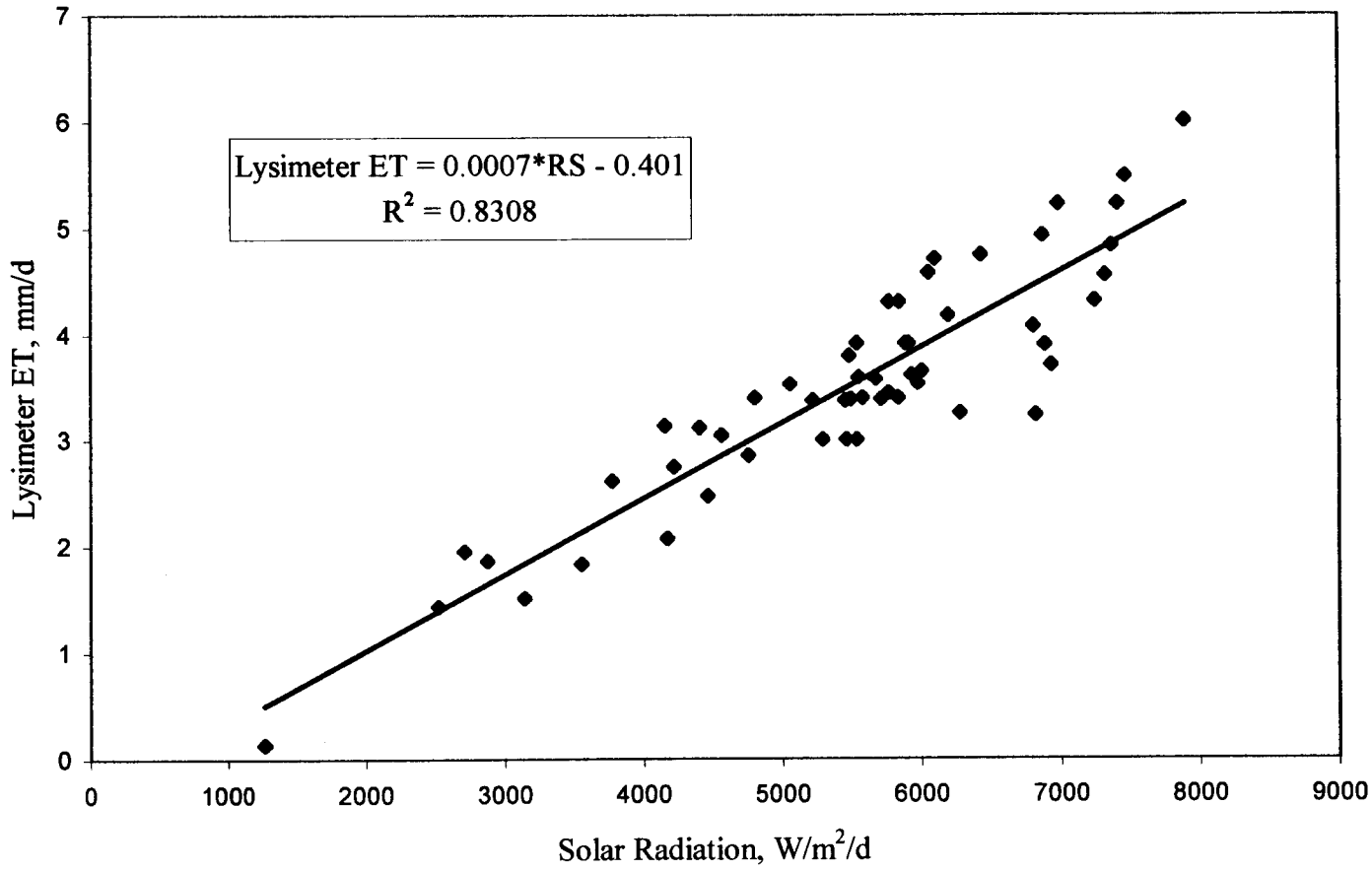


Figure 7.5 Linear regression between daily solar radiation sensor measurements and daily lysimeter ET measurements, using data taken during the operation of the system.

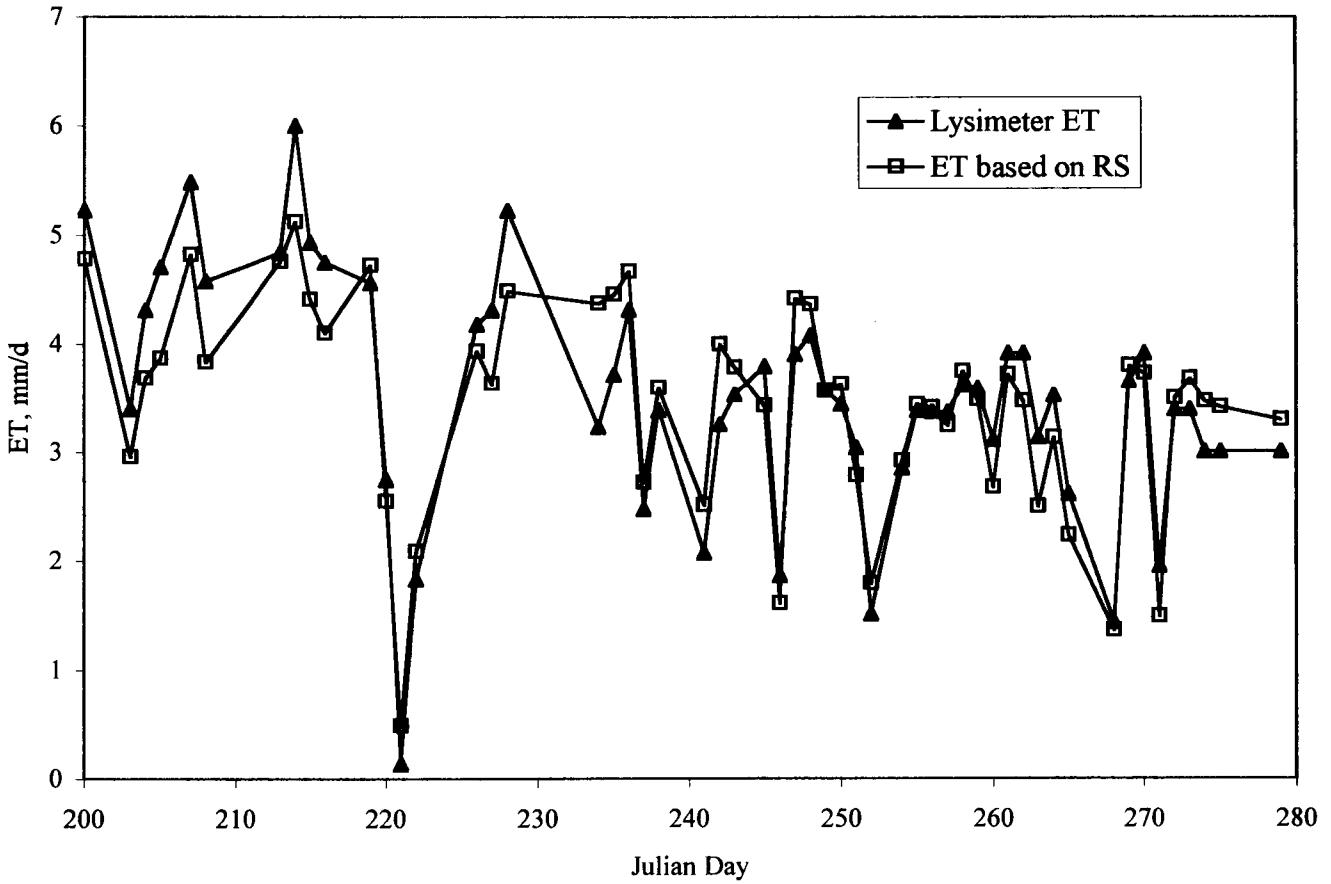


Figure 7.6 Comparison between daily solar radiation measurements and daily lysimeter ET measurements, using data taken during the operation of the system.

CHAPTER 8

SUMMARY AND CONCLUSIONS

The systematic application of fuzzy logic and neural networks for estimation and control in irrigation management has been discussed in this dissertation. Estimation of evapotranspiration and automatic control of microirrigation systems have been addressed. The study methodology involved initial system analysis, system design, experimental implementation, system optimization, and computer simulation to confirm the feasibility of the proposed strategies.

Fuzzy logic was initially applied to the estimation of evapotranspiration with a reduced number of inputs when compared to the current estimation models. Solar radiation and relative humidity sensor measurements were acquired and processed into evapotranspiration. In Chapter 6 the result of the estimation technique was compared with actual lysimeter measurements. It was demonstrated that the initial fuzzy estimator tracked the measurements very well although it presented some offset. This was expected since this system was designed using a very small input/output database to assess the membership functions.

An optimization method that integrates neural networks and fuzzy logic (ANFIS) was applied to the fuzzy estimator. In Chapter 7 it was seen by the results of a simulation that the fuzzy evapotranspiration estimator was significantly enhanced.

An automatic irrigation control system based on fuzzy logic was designed and tested in a field experiment. The control system relied on two components of the soil-plant-

atmosphere system: weather parameters and soil water status. The system made use of the fuzzy evapotranspiration estimator and a set of soil moisture sensors to achieve real-time feedback and control. Real-time was taken as fifteen-minute intervals which represents an advance in terms of real-time irrigation control, since most of the current *state-of-the-art* irrigation control technology is based on hourly intervals.

A statistical experiment design was implemented to test the system and compare it with a high-level conventional control system. In Chapter 6 the performance of the fuzzy control technique was analyzed. The fuzzy control evidenced a slightly better performance, keeping the soil moisture conditions within a smaller and very appropriate range. However, statistically no differences were evident for yield among both the fuzzy and the conventional control systems and a third treatment in which no irrigation was applied. Apparently, the excessive rainfall during the season masked the results.

Judgement on which control system is superior is not made because the systems were not completely similar, since the conventional control operated as an *open-loop* system while the fuzzy control was a *closed-loop* system. Some conclusions drawn from this study are summarized below:

- Fuzzy logic and neural networks allow closed-loop irrigation control systems to operate efficiently and reliably on a real-time basis and in a fully automated manner.
- Fuzzy irrigation control systems can be designed to require fewer input parameters than conventional control systems while keeping good operational performance.
- Fuzzy logic and neural networks technologies can be added to existing conventional irrigation control systems to improve their capabilities.

Topics for Further Research

The field of fuzzy logic and neural networks, and their applications to engineering problems are relatively new. This makes them potential subjects for future studies. There are many items particular to this research that could be explored. Alternate membership functions, inference techniques, and defuzzification techniques could be tried. Additional input variables such as wind speed and air temperature could be applied to the fuzzy evapotranspiration estimator to test for output refinement and precision level.

The real-time interval can be tested at smaller intervals reaching the minute or even second level. This allows for variable water application with the systems controlling not only *when start* but also *when stop* irrigation devices such as pumps and valves.

Studies could be conducted using fuzzy logic control with plant sensor measurements and comparing with soil and weather based systems. Deficit irrigation management is another specific topic for application of fuzzy logic and neural networks. Fuzzy logic and neural networks could be applied to irrigation control systems that rely on wireless sensing technologies. Fuzzy-neural implementation could be used for real-time controller adaptation, where the system itself can learn and modify its own fuzzification and rule base structure, allowing implementation over a wide range of soils, crops, climates, hydrologic regimens, and geographic locations. Neuro-fuzzy systems could be used for adaptive calibration of sensors reducing the effects caused by degradation. Artificial intelligence technologies could also be integrated to global positioning systems and geographic information systems to improve precision in irrigation systems management technology.

REFERENCES

REFERENCES

Aboukhaled, A., A. Alfaro, and H. Smith. 1982. Lysimeters. FAO Irrigation and Drainage Paper No. 39, FAO, Rome, 68 pp.

Adeli, H. and S. L. Hung. 1995. Machine Learning – Neural Networks, Genetic Algorithms and Fuzzy Systems. John Wiley & Sons, New York.

Allen, R. G., M. E. Jensen, J. L. Wright, and R. D. Burman. 1989. Operational Estimates of Evapotranspiration. Agronomy Journal 81:650-662.

Allen, R. G., M. Smith, A. Perrier, and L. S. Pereira. 1994a. An Update for the Definition of Reference Evapotranspiration. ICID Bulletin 43(2): 1-34.

Allen, R. G., M. Smith, A. Perrier, and L. S. Pereira. 1994b. An Update for the Calculation of Reference Evapotranspiration. ICID Bulletin 43(2): 34-92.

Armstrong, C. F. and J. T. Ligon. 1985. Calibration of Watermark Model 200 Soil Moisture Sensor. ASAE Paper No. 85-2077, St. Joseph, MI.

Bardossy, A. and M. Disse. 1993. Fuzzy Rule-Based Models for Infiltration. Water Resources Research 29(2):373-382.

Bausch, W. C. and T. M. Bernard. 1996. Validity of the Watermark Sensor as a Soil Moisture Measuring Device. Proceedings of the International Conference on Evapotranspiration and Irrigation Scheduling, San Antonio, Texas, ASAE, St. Joseph, MI, pp. 933-938.

Benami, A. and A. Offen. 1984. Irrigation Engineering. IESP, Hiafa, Israel ISBN 965-222-029-9. 257 p.

Bruce, R. R., J. L. Chesness, T. C. Keisling, J. E. Pallas, Jr., D. A. Smittle, J. R. Stansell, and A. W. Thomas. 1980. Irrigation of Crops in the Southeastern United States: Principles and Practices. U. S. Department of Agriculture, Sci. Ed. Admin. Agr. Rev. Man. ARM-S-9.

Burman, R. D. and L. O. Pochop. 1994. Evaporation, Evapotranspiration and Climatic Data. Elsevier Science B. V., Amsterdam, The Netherlands.

Cahoon, J., J. Ferguson, D. Edwards, and P. Tacker. 1990. A Microcomputer-Based Irrigation Scheduler for the Humid Mid-South Region. Applied Engineering in Agriculture 6(3):289-294.

Campbell, G. S. 1992. On-line Estimation of Grass Reference Evapotranspiration with the Campbell Scientific Automated Weather Station. Campbell Scientific Application Note, Campbell Scientific Inc., Logan, UT, 22 p.

Clyma, H. E., and D. L. Martin. 1996. Irrigation Management Using Fuzzy Logic. Proceedings of the International Conference on Evapotranspiration and Irrigation Scheduling, San Antonio, Texas, ASAE, St. Joseph, MI, pp. 1134-1139.

Conboy, D. J. 1996. Fuzzy Logic and State Variable Control Applied to a Seeker Servomotor System. M. S. Thesis, The University of Tennessee, Knoxville, TN.

Doorenbos, J. 1976. Agro-Meteorological Field Stations. FAO Irrigation and Drainage Paper No. 27, FAO, Rome.

Doorenbos, J. and W. O. Pruitt. 1975. Guidelines for Prediction of Crop Water Requirements. FAO Irrigation and Drainage Paper No. 24. FAO, Rome.

Doorenbos, J. and W. O. Pruitt. 1977. Guidelines for Predicting Crop Water Requirements. FAO Irrigation and Drainage Paper No. 24. (2^{nd} ed.) FAO, Rome.

Driankov, D., H. Hellendoorn, and M. Reinfrank. 1993. An Introduction to Fuzzy Control, Springer-Verlag, Berlin.

Duke, H. R., L. E. Stetson, and N. C. Ciancaglini. 1990. Irrigation System Controls. *In* Hoffman, G. J., T. A. Howell, and K. H. Solomon. Management of Farm Irrigation Systems. ASAE, St. Joseph, MI, pp. 265-312.

Eldredge, E. P., C. C. Shock, and T. D. Stieber. 1993. Calibration of Granular Matrix Sensors for Irrigation Management. Agronomy Journal 85:1228-1232.

Evett, S. R., T. A. Howell, A. D. Schneider, D. R. Upchurch, and D. F. Wanjura. 1996. Canopy Temperature Based Automatic Irrigation Control. Proceedings of the International Conference on Evapotranspiration and Irrigation Scheduling, San Antonio, Texas, ASAE, St. Joseph, MI, pp. 207-213.

Fangmeier, D. D., D. J. Garrot Jr., C. F. Mancino, and S. H. Husman. 1990. Automated Irrigation Systems Using Plant and Soil Sensors. *In* Visions of the Future. Proceedings of the Third National Irrigation Symposium, Phoenix, Arizona, ASAE, St. Joseph, MI, pp. 533-537.

Fereres, E. 1996. Irrigation Scheduling and its Impact on the 21^{st} Century. Proceedings of the International Conference on Evapotranspiration and Irrigation Scheduling, San Antonio, Texas, ASAE, St. Joseph, MI, pp. 547-553.

Gowing, J., A. Tarimo, and O. El-Awad. 1996. A Rational Method for Assessing Irrigation Performance at Farm Level with the Aid of Fuzzy Set Theory. Irrigation and Drainage Systems 10:319-330, Kluwer Academic Publishers, Netherlands.

Hatfield, J. L. and M. Fuchs. 1990. Evapotranspiration Models. *In* Hoffman, G. J., T. A. Howell, and K. H. Solomon. Management of Farm Irrigation Systems. ASAE, St. Joseph, MI, pp. 33-59.

Hawkins, T. and C. M. Burt. 1990. AGWATER – Irrigation Management and Planning Expert System. *In* Visions of the Future. Proceedings of the Third National Irrigation Symposium, Phoenix, Arizona, ASAE, St. Joseph, MI, pp. 64-68.

Hebb, D. O. 1949. The Organization of Behavior. Wiley, New York.

Hess, T. M. 1996. A Comparison between Evapotranspiration Calculated from an Automatic Weather Station and Conventional Weather Data – Implications for Irrigation Scheduling in the UK. Proceedings of the International Conference on Evapotranspiration and Irrigation Scheduling, San Antonio, Texas, ASAE, St. Joseph, MI, pp. 516-521.

Hines, J. W. 1997. MATLAB Suplement to Fuzzy and Neural Approaches in Engineering. John Wiley & Sons, Inc. New York, NY.

Hopfield, J. J. 1982. Neural Networks and Physical Systems with Emergent Collective Computational Abilities. Proceedings of the National Academy of Sciences of the U.S.A. 79: 2554-2558.

Howell, T. A. 1996. Irrigation Scheduling Research and its Impact on Water Use. Proceedings of the International Conference on Evapotranspiration and Irrigation Scheduling, San Antonio, Texas, ASAE, St. Joseph, MI, pp. 21-33.

Howell, T. A., R. L. McCormick, and C. J. Phene. 1985. Design and Installation of Large Weighing Lysimeters. Transactions of ASAE, 28:106-112, 117.

Ismail, E. S. E. S. and A. A. Al-Shooshan. 1996. Irrigation Feedback-Control System with Modified Tensiometer. Proceedings of the International Conference on Evapotranspiration and Irrigation Scheduling, San Antonio, Texas, ASAE, St. Joseph, MI, pp. 365-370.

James, W. 1890. Psychology (Briefer Course). Holt, New York.

Jang, J. R. 1993. ANFIS: Adaptive Network-Based Fuzzy Inference System. IEEE Transactions on Systems, Man, and Cybernetics, 23(3):665-685.

Jensen, M. E. 1969. Scheduling Irrigation with Computers. Journal of Soil and Water Conservation 24(5):193-195.

Jensen, M. E., D. C. N. Robb, and G. E. Franzoy. 1970. Scheduling Irrigations using Climate-Crop-Soil Data. Journal of Irrigation and Drainage Division. ASCE 96(IRI):25-38.

Jensen, M. E., R. D. Burman, and R. G. Allen. 1990. Evapotranspiration and Irrigation Water Requirements. Manual and Reports on Engineering Practice No. 70, ASCE, New York, NY.

Johnson, D. L. 1995. Soil Moisture Sensor Evaluation. M. S. Thesis. Department of Agricultural & Biosystems Engineering, The University of Tennessee, Knoxville, TN.

Koch, P. R. 1993. Artificial Neural Networks: New Tools for Predicting Yields. ASAE Paper No. 934513, St. Joseph, MI.

Lee, C. C. 1990. Fuzzy Logic in Control Systems: Fuzzy Logic Controller, Part I and Part II. IEEE Transactions on Systems, Man, and Cybernetics 20(2):404-418, 419-435.

Locascio, S. J., J. G. A. Fiskell, D. A. Graetz, and R. D. Hauck. 1985. Nitrogen Accumulation by Pepper as Influenced by Mulch Time of Fertilizer Application. J. Amer. Soc. Hort. Sci. 106: 628-632.

Mamdani, E. H. 1974. Application of Fuzzy Algorithms for Control of a Simple Dynamic Plant. Proceedings of the IEEE, 121(12): 1585-1588.

Marek, T. H., A. D. Schneider, T. A. Howell, and L. L. Ebeling. 1986. Design and Construction of Large Weighing Monolith Lysimeters. ASAE, Paper No. 86-2525, Saint Joseph, MI, 20 pp.

Mathworks Inc. 1994. MATLAB Fuzzy Logic Toolbox User's Guide. Natick, MA.

Mathworks Inc. 1994. MATLAB User's Guide. Prentice-Hall Inc., Upper Saddle River, NJ.

McCann, I. R., D. C. Kincaid, and D. Wang. 1992. Operational Characteristics of the Watermark Model 200 Soil Water Potential Sensor for Irrigation Management. Applied Engineering in Agriculture 8(5):603-609.

McClendon, R. W., I. Seginer, and J. W. Jones. 1991. Optimal Control Applied to Peanut Irrigation Management. ASAE Paper No. 912129, St. Joseph, MI.

McCulloch, W. S. and W. Pitts. 1943. A Logical Calculus of the Ideas Imminent in Nervous Activity. Bulletin of Mathematical Biophysics 5: 115-133.

McGuiness, J. L. W. C. Mills and P. R. Nixon. 1979. Climate. *In* Field Manual for Research in Agricultural Hydrology, USDA-SEA, Agricultural Handbook No. 224, Ch. 4:215-237.

Minsky, M. L. and S. A. Papert. 1969. Perceptrons. MIT Press, Cambridge, MA.

Monteith, J. L. 1965. Evaporation and Environment. Symp. *In* The State and Movement of Water in Living Organisms, XIXth Symposium Soc. for Exp. Biology, Swansea, Cambridge University Press, pp. 205-234.

Parchomchuk, P., R. C. Berard and T. W. Van der Gulik. 1996. Automated Irrigation Scheduling Using an Electronic Atmometer. Proceedings of the International Conference on Evapotranspiration and Irrigation Scheduling, San Antonio, Texas, ASAE, St. Joseph, MI, pp. 1099-1104.

Parker, D. 1982. Learning Logic. Stanford University, Department of Electrical Engineering. Invention Report 681-64.

Penman, H. L. 1948. Natural Evaporation from Open Water, Bare Soil and Grass. Proc. Royal Soc. London, A-193:120-146.

Phene, C. J. 1986. Operation Principles: Automation. *In* Trickle Irrigation for Crop Production: Design, Operation and Management, ed. F. S. Nakayama and D. A. Bucks, Elsevier, Tokyo, pp. 188-215.

Phene, C. J. 1989. Techniques for Computerized Irrigation Management. Computers and Electronics in Agriculture 3:189-208, Elsevier Science Publishers B. V., Amsterdam, The Netherlands.

Phene, C. J. and T. A. Howell. 1984. Soil Sensor Control of High-Frequency Irrigation Systems. Transactions of the ASAE, Paper No. 81-2013, St. Joseph, MI.

Phene, C. J., C. P. Allee, and D. J. Pierro. 1989. Soil Matric Potential Sensor Measurements in Real-Time Irrigation Scheduling. Agricultural Water Management 16(3):173-185.

Phene, C. J., G. J. Hoffman, and R. S. Austin. 1973. Controlling Automated Irrigation with Soil Matric Potential Sensor. Transactions of the ASAE, Paper No. 71-230, St. Joseph, MI.

Phene, C. J., R. J. Reginato, B. Itier, and B. R. Tanner. 1990. Sensing Irrigation Needs. *In* Hofman, G. T., T. A. Howell, and K. H. Solomon. Management of Farm Irrigation Systems. ASAE, St. Joseph, MI, pp. 207-261.

Phene, C. J., W. R. DeTar, and D. A. Clark. 1992. Real-Time Irrigation Scheduling of Cotton with an Automated Pan Evaporation System. Applied Engineering in Agriculture 8(6):787-793.

Pruitt, W. O. and F. J. Lourence. 1985. Experiences in Lysimetry for ET and Surface Drag Measurements. *In* Advances in Evapotranspiration. ASAE Publication No. 74-85:51-69.

Ribeiro, R. S. F. 1996. ET Estimation Using Penman-Monteith Equation Compared to ET Measured by Lysimetry. Internal Report of The University of Tennessee, Knoxville, TN, June 1996.

Ribeiro, R. S. F. 1997. A Fuzzy Model for Evapotranspiration. Internal Report of The University of Tennessee, Knoxville, TN, May 1997.

Ribeiro, R. S. F. and R. E. Yoder. 1997. An Automated Fuzzy Irrigation Control System. *In Proceedings*: The 18th Annual Irrigation Association Exposition and Technical Conference, November 2-4, Opryland Hotel Convention Center, Nashville, TN, pp. 171-178.

Rosenblat, F. 1958. The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain. Psychological Review 65: 386-408.

Ross, T. J. 1995. Fuzzy Logic with Engineering Applications, International Edition. McGraw-Hill, New York.

Rumelhart, D. E., G. E. Hinton, and R. J. Williams. 1986. Learning Internal Representations by Error Propagation. Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 1, Chapter 8, MIT Press, Cambridge, MA.

Runkler, T. A. and M. Glesner. 1993. A Set of Axioms for Defuzzification Strategies towards a Theory of Rational Defuzzification Operators. Second IEEE International Conference on Fuzzy Systems, pp. 1161-1166.

SAS System for Elementary Data Analysis. 1987. SAS Institute, Inc. Cary, NC.

Shearer, M. V., and J. Vomocil. 1981. Twenty Five Years of Promoting Irrigation Scheduling. *In* Irrigation Scheduling for Water and Energy Conservation in the 80's. ASAE Special Publication 23-81. St. Joseph, MI, pp. 208-212.

Shock, C. C. 1996. Personal Communication. Malheur Experiment Station, Oregon State University, Ontario, OR.

Shock, C. C., E. Fibert, and M. Saunders. 1996. Malheur Experiment Station Annual Report. Special Report 964, Oregon State University, Ontario, OR.

Simões, M. G. 1995. Fuzzy Logic and Neural Network Based Advanced Control and Estimation Techniques in Power Electronics and AC Drives. Ph.D. Dissertation, The University of Tennessee, Knoxville, TN.

Smittle, D. A., W. L. Dickens, and J. R. Stansell. 1994. Irrigation Regimes Affect Yield and Water Use by Bell Pepper. J. Amer. Soc. Hort. Sci. 119(5): 936-939.

Steduto, P., A. Caliandro, P. Rubino, N. B. Mechlia, M. Masmoudi, A. Martinez-Cob, M. J. Faci, G. Rana, M. Mastrorilli, M. El Mourid, M. Karrou, R. Kanber, C. Kirda, D. El-Quosy, K. El-Askari, M. A. Ali, D. Zareb, and R. L. Snyder. 1996. Penman-Monteith Reference Evapotranspiration Estimates in the Mediterranean Region. Proceedings of the International Conference on Evapotranspiration and Irrigation Scheduling, San Antonio, Texas, ASAE, St. Joseph, MI, pp. 357-363.

Szeicz, G. 1975. Instruments and their Exposure. *In* Vegetation and the Atmosphere, ed. J. L. Monteith, New York, Academic Press, Vol. I:229-273.

Takagi, H. and I. Hayashi. 1991. NN-Driven Fuzzy Reasoning, International Journal of Approximate Reasoning 5(3): 191-212.

Takagi, T. and M. Sugeno. 1985. Fuzzy Identification of Systems and its Applications to Modeling and Control. IEEE Transactions on Systems, Man, and Cybernetics, 15(1):116-132.

Thomson, S. J. and C. F. Armstrong. 1987. Calibration of the Watermark Model 200 Soil Moisture Sensor. Applied Engineering in Agriculture 3(2):186-189.

Thomson, S. J., T. Younos, and K. Wood. 1996. Evaluation of Calibration Equations and Application Methods for the Watermark® Granular Matrix Soil Moisture Sensor. Applied Engineering in Agriculture 12(1):99-103.

Tsoukalas, L. H. and R. E. Uhrig. 1997. Fuzzy and Neural Approaches in Engineering. John Wiley & Sons, Inc. New York, NY.

Uhrig, J. W., B. A. Engel, and W. L. Baker. 1992. An Application of Neural Networks: Predicting Corn Yields. Proceedings of the 4th International Conference on Computers in Agricultural Extension Programs, Lake Buena Vista, Florida.

van Bavel, C. H. M., and T. V. Wilson. 1952. Evapotranspiration Estimates as Criteria for Determining Time of Irrigation. Agricultural Engineering 33(7):417-418, 420.

Wang, L. X. 1994. Adaptive Fuzzy Systems and Control: Design and Stability Analysis. Prentice-Hall, Englewood Cliffs, NJ.

Wanjura, D. F., D. R. Upchurch and J. R. Mahan. 1992. Automated Irrigation Based on Threshold Canopy Temperature. Transactions of the ASAE 35:153-159.

Werbos, P.J. 1974. Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences. Ph.D. Dissertation, Harvard University, Cambridge, MA.

Wheeler, C. R. 1998. Design and Comparison of a Small, Monolithic Weighing Lysimeter to a Large Lysimeter and to Penman and Penman-Monteith Equations. M. S. Thesis, Department of Agricultural & Biosystems Engineering, The University of Tennessee, Knoxville, TN.

Widrow, B. and M. E. Hoff, Jr. 1960. Adaptive Switching Circuits. IRE WESCON Convention Record, pp. 96-104.

Williams, D. B. and F. S. Zazueta. 1996. Minimizing Irrigation Scheduling Parameters via a Neural Network. Proceedings of the International Conference on Evapotranspiration and Irrigation Scheduling, San Antonio, Texas, ASAE, St. Joseph, MI, pp. 950-955.

Xiang, H., B. P. Verma, and G. Hoogenboom. 1994. Fuzzy Irrigation Decision Support System. ASAE Paper No. 943557, St. Joseph, MI.

Xin, J. N., F. S. Zazueta, A. G. Smajstrla and T. A. Wheaton. 1995. Real-Time Expert System for Citrus Microirrigation Management. Proceedings of the 5th International Microirrigation Congress, Orlando, Florida, ASAE, St. Joseph, MI, pp. 787-791.

Yoder, R. E., D. L. Johnson, J. B. Wilkerson, and D. C. Yoder. 1998. Soil Water Sensor Performance. Applied Engineering in Agriculture 14(2):121-133.

Zadeh, L. A. 1965. Fuzzy sets. Information and Control, 8:338-353.

Zazueta, F. S., J. N. Xin, A. Wheaton, J. Jackson, and M. Almedo. 1994. Development of a Computer-Based Control System for Reclaimed Water Citrus Irrigation. Proceedings of the 5th International Conference on Computers in Agriculture, Orlando, FL.

Zhang, Q., C. J. Wu, and K. Tilt. 1996. Application of Fuzzy Logic in an Irrigation Control System. Proceedings of the IEEE International Conference on Industrial Technology, Shanghai, China, pp. 593-597.

APPENDICES

APPENDIX A

APPENDIX A1. MATLAB® Code for Randomization of Treatments.

```
% RANDEX.M
% Normally distributed random matrix for field experiment
% smaller value = conventional irrigation control (C)
% intermediate value = fuzzy irrigation control (F)
% larger value = no irrigation (N)

randn('seed', 16)
Y = randn(6, 3)
End



% Result

Y =

    -1.2101    -2.0050    -0.7860
     2.4277     0.1752     0.5544
     0.5135    -0.1876     1.8146
     0.6101     1.3284     1.4217
     0.8881     1.3111     0.2814
    -0.7870    -0.3969    -0.4094


Y =

     F     C     N
     N     C     F
     F     C     N
     C     F     N
     F     N     C
     C     N     F
```

APPENDIX A2. Campbell Scientific® CR10 Datalogger Program for Weather Station.

```
;{CR10}
*Table 1 Program
  01: 60.0      Execution Interval (seconds)

1:  If time is (P92)
 1: 0           Minutes (Seconds --) into a
 2: 15          Interval (same units as above)
 3: 10          Set Output Flag High

2:  Real Time (P77)
 1: 110         Day,Hour/Minute

3:  Batt Voltage (P10)
 1: 1           Loc [ batt_vol  ]

4:  Internal Temperature (P17)
 1: 2           Loc [ panel_tem ]

5:  Volt (Diff) (P2)
 1: 1           Reps
 2: 13          ñ 25 mV Fast Range
 3: 1           DIFF Channel
 4: 3           Loc [ solar     ]
 5: 104.38      Mult
 6: 0           Offset

6:  Excite-Delay (SE) (P4)
 1: 1           Reps
 2: 5           ñ 2500 mV Slow Range
 3: 2           SE Channel
 4: 2           Excite all reps w/Exchan 2
 5: 15          Delay (units 0.01 sec)
 6: 2500        mV Excitation
 7: 4           Loc [ RH        ]
 8: .1          Mult
 9: 0           Offset

7:  Temp (107) (P11)
 1: 1           Reps
 2: 3           SE Channel
 3: 1           Excite all reps w/Exchan 1
 4: 5           Loc [ air_temp  ]
 5: 1           Mult
 6: 0           Offset

8:  Volt (Diff) (P2)
 1: 1           Reps
 2: 15          ñ 2500 mV Fast Range
 3: 4           DIFF Channel
 4: 6           Loc [ evap_pan  ]
 5: 1           Mult
```

A2. (continued)

```
 6: 0        Offset

9:  Average (P71)
 1: 4        Reps
 2: 3        Loc [ solar     ]

10:  Pulse (P3)
 1: 1        Reps
 2: 1        Pulse Input Channel
 3: 2        Switch Closure
 4: 7        Loc [ rain      ]
 5: .254     Mult
 6: 0        Offset

11:  Totalize (P72)
 1: 1        Reps
 2: 7        Loc [ rain      ]

12:  Excite-Delay (SE) (P4)
 1: 1        Reps
 2: 15       ñ 2500 mV Fast Range
 3: 5        SE Channel
 4: 3        Excite all reps w/Exchan 3
 5: 2        Delay (units 0.01 sec)
 6: 2500     mV Excitation
 7: 8        Loc [ wind_dir  ]
 8: .142     Mult
 9: 0        Offset

13:  Pulse (P3)
 1: 1        Reps
 2: 2        Pulse Input Channel
 3: 11       Low Level AC
 4: 9        Loc [ wind_spd  ]
 5: .0125    Mult
 6: .2       Offset

14:  Wind Vector (P69)
 1: 1        Reps
 2: 3600     Samples per Sub-Interval
 3: 0        S, é1, & å(é1) Polar
 4: 9        Wind Speed/East Loc [ wind_spd  ]
 5: 8        Wind Direction/North Loc [ wind_dir  ]

15:  IF (X<=>F) (P89)
 1: 9        X Loc [ wind_spd  ]
 2: 4        <
 3: .5       F
 4: 30       Then Do

16:  Z=F (P30)
 1: 0        F
```

A2. (continued)

```
 2: 0         Exponent of 10
 3: 8         Z Loc [ wind_dir  ]

17:  End (P95)

18:  Z=X*F (P37)
 1: 9         X Loc [ wind_spd  ]
 2: 3.6       F
 3: 10        Z Loc [ wind_run  ]

19:  Average (P71)
 1: 3         Reps
 2: 8         Loc [ wind_dir  ]

20:  Sample (P70)
 1: 2         Reps
 2: 1         Loc [ batt_vol  ]


21:  Serial Out (P96)
 1: 42        Printer ASCII/9600 Baud

22:  Serial Out (P96)
 1: 71        SM192/SM716/CSM1

*Table 2 Program
  01: 0.0       Execution Interval (seconds)

*Table 3 Subroutines

End Program



-Input Locations-
1 batt_vol  1 1 1
2 panel_tem 1 1 1
3 solar     1 1 1
4 RH        1 1 1
5 air_temp  1 1 1
6 evap_pan  1 1 1
7 rain      1 1 1
8 wind_dir  1 2 2
9 wind_spd  1 4 1
10 wind_run  1 0 1
11 _________ 1 0 0
12 _________ 0 0 0
13 _________ 0 0 0
14 _________ 0 0 0
15 _________ 0 0 0
16 _________ 0 0 0
```

A2. (continued)

```
17 _________ 0 0 0
18 _________ 0 0 0
19 _________ 0 0 0
20 _________ 0 0 0
21 _________ 0 0 0
22 _________ 0 0 0
23 _________ 0 0 0
24 _________ 0 0 0
25 _________ 0 0 0
26 _________ 0 0 0
27 _________ 0 0 0
28 _________ 0 0 0
-Program Security-
0
0
0
-Mode 4-
-Final Storage Area 2-
0
```

APPENDIX A3. Campbell Scientific® 21X Datalogger Program for Soil Moisture Sensors and Thermocouples.

```
;{21X}
;
*Table 1 Program
  01: 10.0      Execution Interval (seconds)

1:  If time is (P92)
 1: 0           Minutes into a
 2: 15          Minute Interval
 3: 10          Set Output Flag High

2:  Real Time (P77)
 1: 110         Day,Hour/Minute

3:  Batt Voltage (P10)
 1: 1           Loc [ bat_vol   ]

4:  Internal Temperature (P17)
 1: 2           Loc [ pan_temp  ]

5:  If time is (P92)
 1: 0           Minutes into a
 2: 1           Minute Interval
 3: 30          Then Do

;loop for mux, to read watermarks sensors


6:  Set Port (P20)
 1: 1           Set High
 2: 1           Port Number

7:  Beginning of Loop (P87)
 1: 0           Delay
 2: 12          Loop Count

8:  Excitation with Delay (P22)
 1: 4           Ex Channel
 2: 1           Delay w/Ex (units = 0.01 sec)
 3: 1           Delay After Ex (units = 0.01 sec)
 4: 5000        mV Excitation

9:  AC Half Bridge (P5)
 1: 1           Reps
 2: 15          ñ 5000 mV Fast Range
 3: 1           SE Channel
 4: 1           Excite all reps w/Exchan 1
 5: 5000        mV Excitation
 6: 3       -- Loc [ moist1#1  ] ;
 7: 1.0         Mult
 8: 0.0         Offset
```

A3. (continued)

```
10:  AC Half Bridge (P5)
 1: 1         Reps
 2: 15        ñ 5000 mV Fast Range
 3: 2         SE Channel
 4: 2         Excite all reps w/Exchan 2
 5: 5000      mV Excitation
 6: 15     -- Loc [ moist2#1  ]
 7: 1.0       Mult
 8: 0.0       Offset

11:  AC Half Bridge (P5)
 1: 1         Reps
 2: 15        ñ 5000 mV Fast Range
 3: 3         SE Channel
 4: 3         Excite all reps w/Exchan 3
 5: 5000      mV Excitation
 6: 27     -- Loc [ moist3#1  ]
 7: 1.0       Mult
 8: 0.0       Offset

12:  Thermocouple Temp (DIFF) (P14)
 1: 1         Reps
 2: 1         ñ 5 mV Slow Range
 3: 3         DIFF Channel
 4: 1         Type T (Copper-Constantan)
 5: 2         Ref Temp Loc [ pan_temp  ]
 6: 39     -- Loc [ temp1#1   ]
 7: 1.0       Mult
 8: 0.0       Offset

13:  End (P95)

14:  Set Port (P20)
 1: 0         Set Low
 2: 1         Port Number

15:  Sample (P70)
 1: 50        Reps
 2: 1         Loc [ bat_vol   ]

16:  Serial Out (P96)
 1: 12        Printer ASCII/9600 Baud


17:  End (P95)

*Table 2 Program
  02: 0.0       Execution Interval (seconds)

*Table 3 Subroutines

End Program
```

A3. (continued)

```
-Input Locations-
1 bat_vol    1 1 1
2 pan_temp   1 2 1
3 moist1#1   7 1 1
4 moist1#2   11 1 0
5 moist1#3   11 1 0
6 moist1#4   11 1 0
7 moist1#5   11 1 0
8 moist1#6   11 1 0
9 moist1#7   11 1 0
10 moist1#8  11 1 0
11 moist1#9  11 1 0
12 moist1#10 11 1 0
13 moist1#11 11 1 0
14 moist1#12 19 1 0
15 moist2#1  7 1 1
16 moist2#2  11 1 0
17 moist2#3  11 1 0
18 moist2#4  11 1 0
19 moist2#5  11 1 0
20 moist2#6  11 1 0
21 moist2#7  11 1 0
22 moist2#8  11 1 0
23 moist2#9  11 1 0
24 moist2#10 11 1 0
25 moist2#11 11 1 0
26 moist2#12 19 1 0
27 moist3#1  7 1 1
28 moist3#2  11 1 0
29 moist3#3  11 1 0
30 moist3#4  11 1 0
31 moist3#5  11 1 0
32 moist3#6  11 1 0
33 moist3#7  11 1 0
34 moist3#8  11 1 0
35 moist3#9  10 1 0
36 moist3#10 10 1 0
37 moist3#11 10 1 0
38 moist3#12 18 1 0
39 temp1#1   7 1 1
40 temp1#2   10 1 0
41 temp1#3   10 1 0
42 temp1#4   10 1 0
43 temp1#5   10 1 0
44 temp1#6   10 1 0
45 temp1#7   10 1 0
46 temp1#8   10 1 0
47 temp1#9   10 1 0
48 temp1#10  10 1 0
49 temp1#11  10 1 0
50 temp1#12  18 1 0
-Program Security-
```

APPENDIX A4. MATLAB® Code for Reading Serial Ports and Writing to the Control Board.

```
%
% This file calls compiled files for reading the serial ports and
% writing to the control board.  The communication between the file
% takes place by writing control files and reading data files.
%

%
% place initilization commands here
%

!d:\matlab\rr\inicont

% end
% start loop here
ETi = 0;
ctr = 3;
run=1;
while run==1

%
% read serial ports
%

!d:\matlab\rr\datap1

%
%open file and read data here
%
% moisture file data
%

fid = fopen('d:\data\time.err','r+');
t=fscanf(fid,'%s',inf);
if t=='timeout'
   !d:\matlab\rr\inicont
   fprintf(fid,'       ');
   fclose(fid);
else
   fclose(fid);
   fid=fopen('d:\data\moist.dat','r+');
   f=fread(fid);
   fclose(fid);
   f=setstr(f');
   u=find(f==',');
   for x=1:12;
     if x==1;
        F(x)=str2num(f(1:u(x)-1));
     else;
        F(x)=str2num(f(u(x-1)+1:u(x)-1));
     end;
   end;
```

A4. (continued)

```
% weather file data
%
   fid=fopen('d:\data\weath.dat','r+');
   w=fread(fid);
   fclose(fid);
   w=setstr(w');
   u=find(w==',');
   for x=1:12;
      if x==1;
          W(x)=str2num(w(1:u(x)-1));
      else;
          W(x)=str2num(w(u(x-1)+1:u(x)-1));
   end;
   end;

%
%place fuzzy control code here
%
   control = 0;
   fire
% Call control code?
if control==1

   fid = fopen('c:\data\cont.dat','w');
   fprintf(fid,'%i',ctr);
   fclose(fid);
   !d:\matlab\rr\control
  end
  end
end
```

APPENDIX A5. BASIC Code for Writing a Value to Digital Output Port to Turn the Valves Off.

```
'$INCLUDE: 'CB.BI'            ' Mandatory INCLUDE file to access default
                              ' parameter values

CONST BoardNum = 0            ' Board number

'Initiate error handling
'  activating error handling will trap errors like
'  bad channel numbers and non-configured conditions.
'  Parameters:
'     PRINTALL    :all warnings and errors encountered will be printed
'     STOPALL     :if any error is encountered, the program will stop

   UDStat% = cbErrHandling%(PRINTALL, STOPALL)


CLS

'configure FIRSTPORTA for digital output
'  Parameters:
'     BoardNum    :the number used by CB.CFG to describe this board
'     PortNum%    :the output port
'     Direction%  :sets the port for input or output

   PortNum% = FIRSTPORTCL
   Direction% = DIGITALOUT

   UDStat% = cbDConfigPort%(BoardNum, PortNum%, Direction%)

'turn off the control data value

   DataValue% = 3

'write the value to FIRSTPORTA
'  Parameters:
'     BoardNum    :the number used by CB.CFG to describe this board
'     PortNum%    :the output port
'     DataValue%  :the value written to the port

      UDStat% = cbDOut%(BoardNum, PortNum%, DataValue%)

END
```

APPENDIX A6. BASIC Code for Reading Weather and Soil Moisture Sensor Data.

```
'   DEFINE ARRAYS FOR VALUES
'   WEATHER IS FOR THE WEATHER STATION DATA
'   MOISTURE IS FOR THE MOISTURE SENSOR DATA

DIM WEATHER(12) AS SINGLE
DIM MOISTURE(12) AS SINGLE

' OPEN THE COM PORTS AND DATA FILES

OPEN "com1:9600,n,8,1,cs,ds" FOR RANDOM AS #1
OPEN "com2:9600,n,8,1,cs,ds" FOR RANDOM AS #2
OPEN "D:\DATA\DATALOG.DAT" FOR APPEND AS #3
OPEN "D:\DATA\WEATH.DAT" FOR OUTPUT AS #4
OPEN "D:\DATA\MOIST.DAT" FOR OUTPUT AS #5
OPEN "D:\DATA\TIME.ERR" FOR OUTPUT AS #6

' PARALLEL PORT ASSIGNMENTS

dataport = &H378
status = &H379
control = &H37A

' WRITE TO PARALLEL PORT FOR STILL ALIVE PULSE
'   UNTIL DATA IS RECIVED AT COM1
ports = 0
c = 0
d = 0
count = 1

OUT control, 0

PRINT #6,
' main program
ti = TIMER
DO
            IF c = 0 THEN
               c = LOC(1)
               OUT dataport, &HFF
               IF c > 0 THEN
                    ports = ports + 1
                    LINE INPUT #1, x$
                    LINE INPUT #1, Y$
                    REM LINE INPUT #1, Z$
                    Y$ = " " + RIGHT$(Y$, LEN(Y$) - 1)
                    REM Z$ = " " + RIGHT$(Z$, LEN(Z$) - 1)
                    x$ = x$ + Y$
                    sp = INSTR(x$, "01+") + 32
                    sl = LEN(x$)
                    x$ = RIGHT$(x$, sl - (sp - 32))
                    sp = 32
                    PRINT #3, "weather", x$
```

A6. (continued)

```
            PRINT #3,

            FOR x% = 1 TO 12

                  WEATHER(x%) = VAL(MID$(x$, sp, 6))

                  PRINT #4, WEATHER(x%); ", ";
                  sp = sp + 10
            NEXT x%

      END IF
END IF

IF d = 0 THEN
      d = LOC(2)
      OUT dataport, &HFF
      IF d > 0 THEN
            ports = ports + 1
            FOR x% = 1 TO 7
                  LINE INPUT #2, a$
                  IF x% > 1 THEN
                        a$ = " " + RIGHT$(a$, LEN(a$) - 1)
                        b$ = b$ + a$
                  ELSE
                        b$ = a$
                  END IF
             NEXT x%

            PRINT #3, "moisture", b$
            PRINT #3,

            sp2 = INSTR(b$, "01+") + 52

            FOR x% = 1 TO 12
                MOISTURE(x%) = VAL(MID$(b$, sp2, 6))
                PRINT #5, MOISTURE(x%); ",";
                sp2 = sp2 + 10
                IF x% = 6 THEN
                   sp2 = sp2 + 180
                END IF
            NEXT x%


  END IF
END IF
PRINT c, d, ports  'PUT THE OUT STATEMENT HERE
IF count MOD 2 THEN
      OUT dataport, &HFE
ELSE
      OUT dataport, &HFF
END IF
count = count + 1
```

A6. (continued)

```
          tl = TIMER
          IF (tl - ti) > 1800 THEN
               PRINT #3, DATE$, TIME$

               PRINT #4,

               PRINT #5,

               PRINT #6, "timeout"

               OUT dataport, &HFF
               CLOSE
               EXIT DO
          END IF
LOOP UNTIL ports = 2

' close ports and files
OUT dataport, &HFF
CLOSE
END
```

APPENDIX A7. BASIC Code for Writing a Value to Digital Output Port to Control the Valves.

```
'$INCLUDE: 'CB.BI'              ' Mandatory INCLUDE file to access default
                                ' parameter values

CONST BoardNum = 0              ' Board number

   PortNum% = FIRSTPORTCL
   Direction% = DIGITALOUT
   UDStat% = cbDConfigPort%(BoardNum, PortNum%, Direction%)

CLS

'open controll file

OPEN "c:\data\cont.dat" FOR INPUT AS #1

'read the control data value

REM INPUT #1, DataValue%
DataValue% = 3
'write the value to FIRSTPORTA
'  Parameters:
'     BoardNum     :the number used by CB.CFG to describe this board
'     PortNum%     :the output port
'     DataValue%   :the value written to the port

      UDStat% = cbDOut%(BoardNum, PortNum%, DataValue%)
CLOSE #1
END
```

APPENDIX A8. MATLAB® Code for Controlling the Fuzzy and the Conventional Irrigation Systems.

```
% Inputs

radi = W(1);
humi = W(2);
mu = (F(1)+F(3)+F(5)+F(7)+F(9)+F(11))/6;
ml = (F(2)+F(4)+F(6)+F(8)+F(10)+F(12))/6;

temp = W(3);
wind = W(6);


if any(F<0 | F>1)
   u=find(F<0 | F>1)
   t=fix(clock);
   fid=fopen('d:\data\badata.m','a+');
   fprintf(fid,'Soil Moisture Sensor %4g is bad\n',u);
   fprintf(fid,'Time is %6g %3g %3g %4g %4g %4g\n',t);
   fclose(fid);
end



if radi < 0 | radi > 1050
   t=fix(clock);
   fid=fopen('d:\data\badata.m','a+');
   fprintf(fid,'Solar Radiation %6g is bad\n',radi);
   fprintf(fid,'Time is %6g %3g %3g %4g %4g %4g\n',t);
   fclose(fid);
end



if humi < 20 | humi > 100
   t=fix(clock);
   fid=fopen('d:\data\badata.m','a+');
   fprintf(fid,'Relative Humidity %6g is bad\n',humi);
   fprintf(fid,'Time is %6g %3g %3g %4g %4g %4g\n',t);
   fclose(fid);
end



if temp < 0 | temp > 50
   t=fix(clock);
   fid=fopen('d:\data\badata.m','a+');
   fprintf(fid,'Temperature %6g is bad\n',temp);
   fprintf(fid,'Time is %6g %3g %3g %4g %4g %4g\n',t);
   fclose(fid);
end
```

A8. (continued)

```
if wind < 0 | wind > 10
   t=fix(clock);
   fid=fopen('d:\data\badata.m','a+');
   fprintf(fid,'Wind Speed %6g is bad\n',wind);
   fprintf(fid,'Time is %6g %3g %3g %4g %4g %4g\n',t);
   fclose(fid);
end



% radi=input('Solar Radiation in w.(15 min)/m^2 (0 - 1050) =');
% humi=input('Relative Humidity in % (20 - 100) =');
% mu=input('Soil Moisture (upper layer) in kPa (0 - 1.00) =');
% ml=input('Soil Moisture (lower layer) in kPa (0 - 1.00) =');

% temp=input('Temperature in celsius degrees =');
% wind=input('Wind speed in m/s =');



 if radi < 0
 radi=0;
 elseif radi > 1050
 radi=1050;
 end

 if humi < 20
 humi=20;
 elseif humi > 100
 humi=100;
 end

if mu < 0
 mu = 0;
 elseif mu > 1
 mu = 1;
 end


if ml < 0
 ml = 0;
 elseif ml > 1
 ml = 1;
 end

if temp < 0
 temp=0;
 elseif temp > 50
 radi=50;
 end

if wind < 0
```

A8. (continued)

```
 wind=0;
elseif wind > 10
 wind=10;
 end



% Solar Radiation Membership Functions
rad =  [0:1:1100];
vlr = mf_tri(rad,[0 0 250],'n');
lowr = mf_tri(rad,[0 250 500],'n');
medr = mf_tri(rad,[250 500 750],'n');
highr = mf_tri(rad,[500 750 1000],'n');
vhr = mf_tri(rad,[750 1100 1100],'n');
radia = [vlr;lowr;medr;highr;vhr];


% Relative Humidity Membership Functions
hum =  [20:1:100];
vlh = mf_tri(hum,[20 20 40],'n');
lowh = mf_tri(hum,[20 40 60],'n');
medh = mf_tri(hum,[40 60 80],'n');
highh = mf_tri(hum,[60 80 100],'n');
vhh = mf_tri(hum,[80 100 100],'n');
humid = [vlh;lowh;medh;highh;vhh];


% Evapotranspiration Membership Functions
etp = [0:0.01:0.24];
vle = mf_tri(etp,[0 0 0.06],'n');
lowe = mf_tri(etp,[0 0.06 0.12],'n');
mede = mf_tri(etp,[0.06 0.12 0.18],'n');
highe = mf_tri(etp,[0.12 0.18 0.24],'n');
vhe = mf_tri(etp,[0.18 0.24 0.24],'n');
et = [vle;lowe;mede;highe;vhe];


% Degree of Fullfillments of Antecedent MFs
DOF1=interp1(rad',radia',radi')';
DOF2=interp1(hum',humid',humi')';



% Fuzzy relation operation AND
antecedent_DOF = [min(DOF1(5), DOF2(1))
min(DOF1(5), DOF2(2))
min(DOF1(5), DOF2(3))
min(DOF1(5), DOF2(4))
min(DOF1(5), DOF2(5))
min(DOF1(4), DOF2(1))
min(DOF1(4), DOF2(2))
min(DOF1(4), DOF2(3))
```

A8. (continued)

```
min(DOF1(4), DOF2(4))
min(DOF1(4), DOF2(5))
min(DOF1(3), DOF2(1))
min(DOF1(3), DOF2(2))
min(DOF1(3), DOF2(3))
min(DOF1(3), DOF2(4))
min(DOF1(3), DOF2(5))
min(DOF1(2), DOF2(1))
min(DOF1(2), DOF2(2))
min(DOF1(2), DOF2(3))
min(DOF1(2), DOF2(4))
min(DOF1(2), DOF2(5))
min(DOF1(1), DOF2(1))
min(DOF1(1), DOF2(2))
min(DOF1(1), DOF2(3))
min(DOF1(1), DOF2(4))
min(DOF1(1), DOF2(5))];


% Consequent Definitions
consequent = [et(5,:)
et(5,:)
et(4,:)
et(4,:)
et(4,:)
et(4,:)
et(4,:)
et(3,:)
et(3,:)
et(3,:)
et(3,:)
et(3,:)
et(3,:)
et(3,:)
et(3,:)
et(3,:)
et(2,:)
et(2,:)
et(2,:)
et(2,:)
et(2,:)
et(1,:)
et(1,:)
et(1,:)];

Consequent = product(consequent,antecedent_DOF);


% Aggregation
aggregation = max(Consequent);
```

A8. (continued)

```
% Defuzzification
output=centroid(etp,aggregation);


% ET Crisp Value after defuzzification
ev =output;


% Evapotranspiration Membership Functions for control
evp = [0:0.01:0.24];
lowev = mf_tri(evp,[0 0 0.12],'n');
medev = mf_tri(evp,[0 0.12 0.24],'n');
highev = mf_tri(evp,[0.12 0.24 0.24],'n');
etv = [lowev;medev;highev];


% Soil Moisture (shallow) Membership Functions
moiu =  [0:0.01:1.00];
wetu = mf_tri(moiu,[0 0 0.60],'n');
medwu = mf_tri(moiu,[0.30 0.60 0.90],'n');
dryu = mf_tri(moiu,[0.60 1 1],'n');
moisu = [wetu;medwu;dryu];


% Soil Moisture (deep) Membership Functions
moil =  [0:0.01:1.00];
wetl = mf_tri(moil,[0 0 0.60],'n');
medwl = mf_tri(moil,[0.30 0.60 0.90],'n');
dryl = mf_tri(moil,[0.60 1 1],'n');
moisl = [wetl;medwl;dryl];


% Fuzzy control Membership Functions
irr =  [0:0.1:10];
vli = mf_tri(irr,[0 0 2.5],'n');
lowi = mf_tri(irr,[0 2.5 5],'n');
medi = mf_tri(irr,[2.5 5 7.5],'n');
highi = mf_tri(irr,[5 7.5 10],'n');
vhi = mf_tri(irr,[7.5 10 10],'n');
irri = [vhi;highi;medi;lowi;vli];


% Degree of Fullfillments of Antecedent MFs
DOF3=interp1(evp',etv',ev')';
DOF4=interp1(moiu',moisu',mu')';
DOF5=interp1(moil',moisl',ml')';


% Antecedent DOF2
antecedent_DOF2 = [min(DOF3(1), min(DOF4(1), DOF5(1)))
```

A8. (continued)

```
min(DOF3(1), min(DOF4(1), DOF5(2)))
min(DOF3(1), min(DOF4(1), DOF5(3)))
min(DOF3(1), min(DOF4(2), DOF5(1)))
min(DOF3(1), min(DOF4(2), DOF5(2)))
min(DOF3(1), min(DOF4(2), DOF5(3)))
min(DOF3(1), min(DOF4(3), DOF5(1)))
min(DOF3(1), min(DOF4(3), DOF5(2)))
min(DOF3(1), min(DOF4(3), DOF5(3)))
min(DOF3(2), min(DOF4(1), DOF5(1)))
min(DOF3(2), min(DOF4(1), DOF5(2)))
min(DOF3(2), min(DOF4(1), DOF5(3)))
min(DOF3(2), min(DOF4(2), DOF5(1)))
min(DOF3(2), min(DOF4(2), DOF5(2)))
min(DOF3(2), min(DOF4(2), DOF5(3)))
min(DOF3(2), min(DOF4(3), DOF5(1)))
min(DOF3(2), min(DOF4(3), DOF5(2)))
min(DOF3(2), min(DOF4(3), DOF5(3)))
min(DOF3(3), min(DOF4(1), DOF5(1)))
min(DOF3(3), min(DOF4(1), DOF5(2)))
min(DOF3(3), min(DOF4(1), DOF5(3)))
min(DOF3(3), min(DOF4(2), DOF5(1)))
min(DOF3(3), min(DOF4(2), DOF5(2)))
min(DOF3(3), min(DOF4(2), DOF5(3)))
min(DOF3(3), min(DOF4(3), DOF5(1)))
min(DOF3(3), min(DOF4(3), DOF5(2)))
min(DOF3(3), min(DOF4(3), DOF5(3)))];


% Consequent 2
consequent2 = [irri(5,:)
irri(5,:)
irri(5,:)
irri(5,:)
irri(4,:)
irri(3,:)
irri(2,:)
irri(2,:)
irri(4,:)
irri(4,:)
irri(4,:)
irri(4,:)
irri(3,:)
irri(3,:)
irri(3,:)
irri(2,:)
irri(1,:)
irri(4,:)
irri(3,:)
irri(3,:)
irri(3,:)
irri(2,:)
```

A8. (continued)

```
irri(2,:)
irri(2,:)
irri(1,:)
irri(1,:)];


Consequent2 = product(consequent2,antecedent_DOF2);


% Aggregation
aggregation2 = max(Consequent2);


% Defuzzification
output2=centroid(irr,aggregation2);


% Fuzzy control value
ig =output2;


% Fuzzy Control
 if ig < 5
 valve1 = 0;
 else
 valve1 = 1;
 end



% ETPM Calculations

T = temp;
RS = radi;
RH = humi;
U2 = wind;


phi=0.62832;
gha=0.409*sin(0.0172*J-1.39);
Lz=90;
Lm=85;
b=(2*(pi)*(J-81))/364;
Sc=0.1645*sin(2*b)-(0.1255*cos(b)-0.025*sin(b);
w=(pi/12)*((t+0.06667*(Lz-Lm)+Sc)-12);
w1=w-((pi)*(1))/24);
w2=w+((pi)*(1))/24);
df=1+0.0033*cos(0.0172*J);
Gsc=0.082;
RA=((12*60)/pi)*Gsc*df*((w2-w1)*sin(phi)*sin(gha)+
cos(phi)*cos(gha)*(sin(w2)-sin(w1)));
```

A8. (continued)

```
D = 2504*exp(((17.27*T)/(T+273.3)))/((T+273.3)^2);
Rso = (0.75+2*10^(-5)*573)*RA*(0.0036);
E = 0.261*exp(-7.77*10^(-4)*T^2)-0.02;
si = 2.04*10^(-10);

if RS <= 10
    G = .5;
else
    G = .9;
end

P = (101.3*(((293-0.0065*573)/293)^5.25));
la = (2.501-(2.361*10^(-3))*T);
ga = 0.00163*(P/la);
ea = ((0.611*exp((17.27*T)/(T+237.3)))*(1-RH/100));
Rn = (0.77*RS*0.0036)-(1.35*((RS*0.0036)/Rso)-0.35)*E*si*(273+T)^4;


ETo =
(((0.408*D*(G*Rn))+(ga*(37/(T+273))*U2*ea)))/(D+ga*(1+0.34*U2))/4;

ETo = ETi + ETo;

% ETPM control
if ETo < 3.3
   valve2 = 0;
   ETi = ETo;
   ETo = 0;
elseif ETo >= 3.3
   valve2 = 1;
   ETi = 0;
   ETo = 0;
end


if ETi < 0
 ETi=0;
 end


if valve1==0 & valve2==0 & ctr~=3
   ctr=3;
   control=1;
   t=fix(clock);
   fid=fopen('d:\data\contdat.txt','a+');
   fprintf(fid,'ctr =%3g and ',ctr);
   fprintf(fid,'time is %6g %3g %3g %4g %4g %4g\n',t);
   fclose(fid);
elseif valve1==1 & valve2==0 & ctr~=2
   ctr=2;
   control=1;
   t=fix(clock);
```

A8. (continued)

```
   fid=fopen('d:\data\contdat.txt','a+');
   fprintf(fid,'ctr =%3g and ',ctr);
   fprintf(fid,'time is %6g %3g %3g %4g %4g %4g\n',t);
   fclose(fid);



elseif valve1==0 & valve2==1 & ctr~=1
   ctr=1;
   control=1;
   t=fix(clock);
   fid=fopen('d:\data\contdat.txt','a+');
   fprintf(fid,'ctr =%3g and ',ctr);
   fprintf(fid,'time is %6g %3g %3g %4g %4g %4g\n',t);
   fclose(fid);
elseif valve1==1 & valve2==1 & ctr~=0
   ctr=0;
   control=1;
   t=fix(clock);
   fid=fopen('d:\data\contdat.txt','a+');
   fprintf(fid,'ctr =%3g and ',ctr);
   fprintf(fid,'time is %6g %3g %3g %4g %4g %4g\n',t);
   fclose(fid);
end
```

APPENDIX B

APPENDIX B1. SAS® Code Used to Determine Differences Among the Treatments.

```
options ls=72 ps=54 pageno=1;
/* UNIX filename: res.sas
   Research Project
*/
data a;
     input bloc @;
       do i = 1 to 3;
       input treat yield @;
       output;
     end;
cards;
1 1 19 2 18 3 9
2 1 15 2 21 3 24
3 1 25 2 19 3 11
4 1 23 2 17 3 11
5 1 27 2 15 3 12
6 1 16 2 30 3 19
;
run;
proc glm;
   class treat bloc;
   model yield = treat bloc;
   means treat;
estimate '1 vs 3' treat 1 0 -1;
estimate '2 vs 3' treat 0 1 -1;
estimate '1 vs 2' treat 1 -1 0;
estimate '12 vs 3' treat .5 .5 -1;
run;
```

APPENDIX C

APPENDIX C1. MATLAB® Code for ANFIS Training of the Input Data and Target Output Data.

```
% Number of total data pairs

numPts = 5064;
load c:\matlab\disserta\etANFISdat.txt;
data = etanfisdat;
trnData = data(1:2:numPts,:); % training data set
chkData = data(2:2:numPts,:); % checking data set


% Generate a Sugeno FIS matrix

numMFs = 5; % number of MFs
mfType = 'trimf'; % MF type is triangular
fismat = genfis1(trnData,numMFs,mfType);


% Plot initial MFs
figure(1)
[x1,mf1] = plotmf(fismat,'input',1);
plot(x1,mf1)
title('Initial MFs for Solar Radiation')
figure(2)
[x2,mf2] = plotmf(fismat,'input',2);
plot(x2,mf2)
title('Initial MFs for Relative Humidity')


% Training/Learning

numEpochs = 40;
[fismat1,trnErr,ss,fismat2,chkErr] =
anfis(trnData,fismat,numEpochs,NaN,chkData);


% Verifying learning results

trnOut = evalfis(trnData(:,[1 2]),fismat2);
trnRMSE = norm(trnOut - trnData(:,3))/sqrt(length(trnOut));


% Plotting error curves
epoch = 1:numEpochs;
figure(3)
plot(epoch,trnErr,'o',epoch,chkErr,'x')
hold on; plot(epoch,[trnErr chkErr]);hold off


% Plotting step size
figure(4)
plot(epoch,ss,'-',epoch,ss,'x')
xlabel('epochs'), ylabel('ss'), title('Step Sizes')
```

C1. (continued)

```
% Plot final MFs

figure(5)
[x1,mf1] = plotmf(fismat2,'input',1);
plot(x1,mf1)
% title('Final MFs for Solar Radiation')
figure(6)
[x2,mf2] = plotmf(fismat2,'input',2);
plot(x2,mf2)
% title('Final MFs for Relative Humidity')
```

VITA

Renato Silvio da Frota Ribeiro was born in Fortaleza, Ceará, Brazil on March 6, 1959. He entered the Federal University of Ceará in February, 1978, and received the degree of Bachelor of Science in Agronomic Engineering in February, 1983. In March 1983 he started a partnership in a small consulting company, Quanta Engineering S/C. In October 1984, he took a position with the Rural Technical Assistance Company of Ceará – EMATER, as an agricultural development engineer. In March 1989, he reentered the Federal University of Ceará, and received a Master of Science degree in Irrigation and Drainage. In August 1991 he succeeded in a contest for a faculty position at the Agricultural Engineering Department of the Federal University of Ceará. In August 1994, he was granted a scholarship from the Brazilian National Council for Scientific and Technological Development – CNPq, to pursue his doctoral degree at The University of Tennessee, Knoxville. He is currently an Assistant Professor with the Department of Agricultural Engineering of the Federal University of Ceará.