ADAPTIVE CONTROL OF REDUNDANT MULTILINK
ROBOT USING FUZZY LOGIC

by

XIONGMIN SU, B.E.

A THESIS

IN

ELECTRICAL ENGINEERING

Submitted to the Graduate Faculty
of Texas Tech University in
Partial Fulfillment of
the Requirements for
the Degree of

MASTER OF SCIENCE

IN

ELECTRICAL ENGINEERING

December, 1993

ACKNOWLEDGMENTS

I would like to express my sincere appreciation to my graduate advisor, Dr. Sunanda Mitra for her financial support, encouragement and guidance. I would also like to extend my gratitude to Dr. Thomas F. Krile and Dr. William J.B. Oldham for their guidance and for serving in my graduate committee. Finally, I am grateful for the support and encouragement offered by my family and friends.

TABLE OF CONTENTS

ACKNOWLEDGMENTS.....	ii
ABSTRACT.....	v
LIST OF TABLES.....	vi
LIST OF FIGURES.....	vii
CHAPTER	
I. INTRODUCTION OF ROBOTICS.....	1
1.1 Background.....	1
1.2 Robot Arm Kinematics.....	2
1.2.1 Direct Kinematics.....	3
1.2.2 Inverse Kinematics.....	3
1.3 Redundancy.....	4
II. CONTROL OF KINEMATICALLY REDUNDANT ROBOT.....	6
2.1 Jacobian Matrix.....	7
2.2 Generalized Inverse and Pseudo inverse.....	8
2.3 A General Solution of Inverse Kinematics.....	9
III. FUZZY SETS AND FUZZY LOGIC CONTROL.....	12
3.1 Fuzzy Sets.....	13
3.1.1 Terminology.....	14
3.1.2 Fuzzy Set Operations.....	15
3.2 Linguistic Variables.....	16
3.3 Fuzzy Logic and Approximate Reasoning.....	17
3.4 Fuzzy Logic Control	19
3.5 Example.....	21

3.5.1 System Description.....	21
3.5.2 Fuzzy Logic Controller Design.....	23
3.5.3 Simulation Result.....	24
IV. FUZZY RULE BASED ROBOT CONTROLLER.....	27
4.1 Design of Fuzzy Logic Controller.....	27
4.2 A Modified FLC with regard to Internal and External Restrictions..	32
V. SIMULATION RESULTS.....	35
VI. CONCLUSION.....	38
REFERENCES.....	39

ABSTRACT

The kinematics model of a multilink robot manipulator involves the solution of an inverse problem. The use of fuzzy logic based linguistic rules demonstrates superior performance over classical, computationally intensive and yet approximate solutions for the target position control and tracking control of a multilink robot arm. A new approach to fuzzy distance and restriction measures is used to obtain the appropriate orientations of the links for avoiding obstacles in the robot trajectories. This approach eliminates the classical task of solving highly coupled, nonlinear equations describing the ill-posed inverse problems of multilink robot motion at a much less demanding computational time. Such clear advantages of fuzzy logic based adaptive controllers are illustrated by simulation results of guidance of a multilink robot in target positioning and trajectory tracking. The simulation results involve a three-link robot arm with the capability of moving from one position to any desired position and tracking defined trajectories accurately. A modified fuzzy rule based distance measure allows the robot to follow trajectories without hitting the obstacles in the path. The simulation results indicate the advantage of fuzzy logic based adaptive controllers in multiple criteria decision-making tasks.

LIST OF TABLES

3.1 The rule base for inverted pendulum system.....	23
4.1 Fuzzy rule base table.....	30
4.2 Fuzzy rule base table for avoiding obstacle.....	33

LIST OF FIGURES

1.1 Direct and inverse kinematics.....	3
3.1 Diagrammatic representation of linguistic variable speed.....	17
3.2 Basic fuzzy logic control model.....	20
3.3 Inverted pendulum system.....	21
3.4 Control variable membership functions.....	24
3.5 Simulation of the linearized system with feedback controller.....	25
3.6 Simulation of the linearized system with fuzzy logic controller.....	25
3.7 Simulation of the nonlinear system with feedback controller.....	26
3.8 Simulation of the nonlinear system with fuzzy logic controller.....	26
4.1 Block diagram of the fuzzy rule based controller.....	28
4.2 A 3-link redundant robot arm configuration.....	28
4.3 Membership functions.....	30
4.4 Internal restrictions.....	32
4.5 External restrictions.....	33
5.1 Trajectory tracking using conventional method.....	36
5.2 Trajectory tracking using fuzzy logic.....	36
5.3 Obstacle avoidance using conventional method.....	37
5.4 Obstacle avoidance using fuzzy logic.....	37
5.5 Target position control using fuzzy logic.....	37

CHAPTER I

INTRODUCTION OF ROBOTICS

1.1 Background

With a pressing demand for increased productivity and delivery of end products of uniform quality, industry is turning more and more toward computer-based automation. Capable of performing a variety of manufacturing functions in a flexible working environment and at low production costs, robots have been widely used in manufacturing and assembly tasks, such as material handling, spot/arc welding, parts assembly, paint spraying, space and undersea exploration and in handling hazardous materials.

The Robot Institute of America defines robot as " a re-programmable multi-functional manipulator designed to move materials, parts, tools, or specialized devices, through variable programmed motions for the performances of a variety of tasks." With this definition, a robot must have intelligence, which is usually due to computer algorithms associated with its control and sensing systems [Fu 1987].

An industrial robot is a general-purpose, computer-controlled manipulator comprising several rigid links connected in series by revolute or prismatic joints. One end of the chain is fixed to a supporting base, while the other end is free and equipped with a tool, or end effector, to manipulate objects or perform an assembly task. The motion of the joints results in relative motion of the links.

Most of today's industrial robots, though controlled by mini- and micro-computers, are basically simple positional machines. They perform a defined

task by playing back prerecorded or preprogrammed sequences of motions that have been previously guided or taught by a user with a hand-held control-teach box. They are equipped with little or no external sensors for obtaining the information about their working environment. As a result, robots are used mostly in relatively simple, repetitive tasks. To improve the overall performance of the manipulator system, more research effort is being directed in many interdisciplinary areas, including kinematics, dynamics, planning systems, control, sensing, programming languages, and machine intelligence.

1.2 Robot Arm Kinematics

Robot arm kinematics deals with the analytical study of the geometry of the motion of a robot arm with respect to a fixed reference coordinate system without considering the forces and moments that generate the motion [Fu 1987]. Thus, kinematics deals with the analytical description of the spatial displacement of the links of a robot arm as a function of time, in particular the relations between the joint-variable space and the position and orientation of the end-effector of a robot arm.

There are two basic problems in robot arm kinematics: direct (or forward) kinematics and inverse kinematics. Solving the direct kinematics problem is equivalent to finding the mapping from joint space to Cartesian space, while solving the inverse kinematics problem is equivalent to finding an inverse mapping from tool-configuration space back to joint space. A summary of the relationship between the two forms of the kinematics problem is shown in Fig. 1-1.

Figure 1-1 Direct and Inverse Kinematics.

1.2.1 Direct kinematics

The direct kinematics problem can be formally addressed as: given the vector of joint variable q of a robot manipulator, find out the position p and orientation R of the end effector with respect to a coordinate frame attached to the robot base [Schilling 1990].

Denavit and Hartenberg proposed a systematic and generalized method (D-H algorithm) of using matrix algebra to represent and describe the spatial geometry of the links of a robot arm with respect to a fixed reference frame [Denavit and Hartenberg 1955]. Successive transformations between adjoining coordinate frames, starting at the end effector to the base of the robot, then led to the arm matrix. The arm matrix represents the position and orientation of the end effector in the base frame as a function of the joint variables.

1.2.2 Inverse Kinematics

The inverse kinematics problem can be characterized as: given a desired position p and orientation R for the end effector and the geometric link parameters with respect to a reference coordinate system, determine values for the joint variables q which can realize the predefined robot end effector position and orientation.

The inverse kinematics problem is more difficult than the direct kinematics problem because a systematic closed-form solution for general robots is not available. In addition, the closed-form solutions are not unique. Since the independent variables in a robot arm are the joint variables, and a task is usually given in the Cartesian space, the inverse kinematics problem is used more frequently. Indeed, a key to make robots more versatile lies in using feed-back from external sensors such as tactile arrays and vision. External sensors provide information about part location and part orientation directly in terms of Cartesian space variables. It is then necessary to use this information to determine appropriate values for the joint variables to properly configure the end effector. A number of investigators have proposed various methods such as the matrix algebraic [Denavit 1956], iterative [Uicker 1964], or geometric approaches [Lee 1984]. Conventionally, the matrix algebraic method has been studied by many investigators. A well-known matrix algebraic method suggested by Lovass-Nagy will be discussed in Chapter II.

1.3 Redundancy

Redundancy has been introduced into various systems that need high reliability, such as aircraft, spacecraft and industry robots.

A kinematically redundant manipulator is a robot arm which has more axes or degrees of freedom (d.o.f.) than are necessary to establish an arbitrary position and orientation of the end effector. For example, a rigid body has three d.o.f. for translation and three d.o.f. for rotation. The arbitrary adjustment of 6 d.o.f. inside a well defined working space can be achieved with a robot arm with 6 joints only. The set of tasks achievable by the manipulator with a

smaller number of joints is limited. In the case of a higher number of d.o.f. than necessary a robot manipulator can execute the task in different ways.

Redundant manipulators have several potential advantages over non redundant robot arms. In a work space cluttered with obstacles, the extra degrees of freedom can be utilized to reach around or between obstacles and thereby manipulate otherwise unobtainable parts. In addition, redundant joints might also be employed to avoid internal joint limits and degenerate or singular configurations of the arm which are associated with very high joint velocities.

Chapter II discusses the control problems of a kinematically redundant multilink robot arm and presents a conventional matrix algebraic control model. Chapter III introduces fuzzy sets and advantages of fuzzy rule based control. Chapter IV develops a fuzzy rule based controller for a 3-link planar robot arm. Chapter V presents the simulation results of trajectory tracking and obstacle avoidance by a 3-link planar robot arm using the conventional matrix algebraic method and a fuzzy rule based controller. Finally, Chapter VI discusses the advantages of the proposed fuzzy logic controller for a multilink robot arm and future work.

CHAPTER II

CONTROL OF KINEMATICALLY REDUNDANT ROBOT

There are two kinds of tasks in robot arm control: target positioning and tracking. The goal of tracking control is to guide the end effector along a defined trajectory given in the Cartesian space. The goal of target position control is to lead the end effector from an initial position to a desired position within the working space. For target position control, generally, a two-point boundary problem needs to be solved to obtain a path in the Cartesian space. Since there are an infinite number of solutions, an optimization procedure is needed to obtain a unique solution and it is very computationally expensive. After the path is defined, the problem of target position control becomes the problem of trajectory tracking control. Obviously, both control tasks involve the inverse kinematics problem which require converting the Cartesian space to the Joint space. The problem of the control of a redundant robot arm, the so called inverse kinematics problem, becomes the determination of internal joint values if a task is predefined in operational space. It is very complex because of the high nonlinearity and coupling effect of a robot mechanism and there is no analytical closed-form solution for a general robot manipulator.

Differential kinematics of robot manipulators were introduced by Whitney . He proposed that we use differential relationships to solve for the joint motion from a given Cartesian motion of the end-effector [Whitney 1969]. The relationship between the end-effector velocity and the joint velocity is described by a linear algebraic equation. The coefficient of the linear equation

is the Jacobian matrix and a nonlinear function of the joint angles. This method is well known as resolved motion rate control.

The inverse can be solved by including additional conditions such as external restrictions (e.g., obstacles) and internal restrictions (boundary positions of the links). An approach proposed by Lovass-Nagy which is based on the use of the generalized inverse is useful for optimizing with respect to some motion criteria [Lovass-Nagy 1987].

2.1 Jacobian Matrix

The control of the motion of a manipulator end effector is accomplished indirectly by controlling the rates at which the individual joints are driven. The kinematics model of an n -axis robot arm can be described as a system of m nonlinear algebraic equations in the following general form:

$$x = f(q). \quad (2.1)$$

Here the vector $x \in R^m$ denotes the position and orientation of the end effector within the workspace and represents a point in Cartesian space. Similarly, $q \in R^n$ is the vector of joint variables, and it represents a point in the joint space of the manipulator. For an articulated coordinate robot the components of q are simply the joint angles. More generally, q consists of a combination of joint angles for the revolute joints and joint distances for the prismatic joints.

Given a desired trajectory $x(t)$ in Cartesian space R^m , we need to find a corresponding trajectory $q(t)$ in joint space R^n that is consistent with the kinematics model. One approach to this problem is to differentiate the

kinematics model with respect to time and thereby reformulate the problem in terms of configuration space velocity and joint space velocity as follows:

$$\dot{x} = J(q)\dot{q} \quad (2.2)$$

where $J(q) = \partial f(q) / \partial q$ is the $m \times n$ Jacobian matrix of partial derivatives of f evaluated at the operating point q . It relates the velocity of the joints in joint space to the velocity of the end effector in Cartesian space. The inverse Jacobian relates the velocity of the end effector in Cartesian space to the joint velocities.

The number of rows in the Jacobian depends on the number of degrees of freedom in Cartesian space required by the task which is in turn determined by the degrees of mobility of the manipulator and task constraints. The number of columns in the Jacobian is determined by the number of joints in the manipulator, that is the degrees of mobility in joint space. If there are more columns than rows, then the manipulator is redundant. That is, the number of degrees of mobility of the manipulator is greater than the number of degrees of freedom required by the task, and we have multiple inverse solutions.

2.2 Generalized Inverse and Pseudo inverse

A generalized inverse of a matrix $A \in R^{m \times n}$ is a matrix $X = A^{-1} \in R^{n \times m}$ satisfying:

$$AXA = A \quad (2.3)$$

A pseudo inverse of a matrix $A \in R^{m \times n}$ is a matrix $X = A^\# \in R^{n \times m}$ satisfying the following conditions:

$$AXA = A \quad (2.4)$$

$$XAX = X \quad (2.5)$$

$$(AX)^T = AX \quad (2.6)$$

$$(XA)^T = XA \quad (2.7)$$

A pseudo inverse is sometimes called the Moore-Penrose inverse after the pioneering works by Moore [Moore 1935] and Penrose [Penrose 1955].

2.3 A General Solution of Inverse Kinematics

A redundant manipulator yields $n-m$ redundant d.o.f which can be used freely with regard to various criteria for the joint position. Corresponding investigations have been done by Liegeois [Liegeois 1977] and Lovass-Nagy [Lovass-Nagy 1987]. The main idea is to solve the inverse problem as follows:

$$dq = J1 \cdot dx + (J2 \cdot J - I) \cdot y \quad (2.8)$$

$J1$ and $J2$ are generalized inverse matrices and I is the $n \times n$ identity matrix and y is an $n \times 1$ vector.

The choice of y has to be based on the basis of corresponding criteria of optimization.

The validity of (2.8) can be proved with the attributes of the generalized inverse by

$$J \cdot dq = J \cdot J1 \cdot J \cdot dq + (J \cdot J2 \cdot J - J) \cdot y = dx \quad (2.9)$$

The $n \times m$ matrix $J1$ can be obtained by inversion of the $m \times m$ sub matrix J_{subi} with

$$|\det(J1_{subi})| \rightarrow \max, \quad i = 1, \dots, \binom{n}{m} \quad (2.10)$$

With this, the next point in the trajectory x is completed by a minimal change of the corresponding coordinates. The locations of the remaining $n-m$ joints are kept constant. As a result, the matrix

$$J1 = \begin{pmatrix} g_{11} & g_{12} & \cdots & g_{1m} \\ g_{21} & g_{22} & \cdots & g_{2m} \\ \vdots & \vdots & & \vdots \\ g_{n1} & g_{n2} & \cdots & g_{nm} \end{pmatrix} \quad (2.11)$$

has m linearly independent rows. The remaining $n-m$ rows contain null elements. If, because of internal restrictions, the motion computed cannot be performed, another $m \times m$ sub matrix of J is selected for inversion. The solution of the inverse $J1$ is, in a hierarchical sense, a local one with a local criterion [Palm 1992].

The second term of (2.8) is the projection of a vector z onto the null-space of J . The matrix $J2$ is usually selected as the pseudo inverse with the attributes given by (2.4)-(2.7).

This selection is, although very computationally time consuming, favored because $J2$ guarantees the solution of (2.8) with minimum norm for $dx=0$. Considering (2.6) and (2.8), we have

$$J2 = J^T \cdot (J \cdot J^T)^{-1}. \quad (2.12)$$

The vector y can be determined according to certain restrictions. For example: Let

$$H = \frac{1}{N} \sum_{i=1}^N \frac{(q_i - a_i)^2}{(a_i - q_{iM})^2} \quad (2.13)$$

be the criterion for which minimization guarantees a large distance of the joints from their internal boundaries with q_{im} the minimum boundary, q_{iM} the maximum boundary, N the number of joints and $a_i = \frac{1}{2}(q_{im} + q_{iM})$. An appropriate control vector y can be obtained by taking the derivative of H as a Lyapunov function

$$\dot{H} = (\nabla H_q)^T \cdot \dot{q}. \quad (2.14)$$

In the particular case $dx=0$, $\dot{H}$ is negative definite with the control law

$$\dot{q} = -K \cdot \nabla H_q \quad (2.15)$$

where K is an $(n \times n)$ -matrix with positive eigenvalues.

From this we have

$$\dot{H} = -(\nabla H_q)^T \cdot K \cdot (\nabla H_q). \quad (2.16)$$

A matrix with the above-mentioned attributes which fulfills (2.8) is

$$K = \alpha \cdot (I - J^2 \cdot J) \quad (2.17)$$

where α is scalar, positive. The elements of y are denoted as

$$y_i = \alpha \cdot \nabla H_{q_i} = \alpha \frac{\partial H}{\partial q_i} = \alpha \cdot \frac{2}{N} \frac{q_i - a_i}{(a_i - q_{iM})^2}. \quad (2.18)$$

With the above procedures, the inverse problem of a redundant robot arm using a conventional matrix algebraic method can be solved.

CHAPTER III

FUZZY SETS AND FUZZY LOGIC CONTROL

As general tools for dealing with complexity independent of the particular content of concern, the theory of fuzzy sets and the various mathematical descriptions and measurements of uncertainty and information have a virtually unlimited applicability [Klir 1988]. Successful applications have been made in various areas as numerous and diverse as engineering, psychology, artificial intelligence, medicine, ecology, decision theory, pattern recognition, information retrieval, sociology, and meteorology. The theory of fuzzy sets has been receiving more and more recognition as a valid and useful extension of classical mathematics and will undoubtedly continue to constitute an important framework for further investigations into proper representations of uncertainty, information and complexity.

During the past several years, fuzzy control has emerged as one of the most active and fruitful areas for research in the application of fuzzy set theory. The pioneering research of Mamdani and his colleagues on fuzzy logic control was motivated by Zadeh's seminal papers on the linguistic approach and system analysis based on the theory of fuzzy sets [Mamdani 1975]. Recent applications of fuzzy control in water quality control, automatic container crane operation systems, elevator control, nuclear reactor control, automobile transmission control and fuzzy logic controller hardware systems have pointed the way for an effective utilization of fuzzy control in the context of complex ill-defined processes that can be controlled by a skilled human operator without the knowledge of their underlying dynamics.

3.1 Fuzzy Sets

The crisp set forms the cornerstone of any mathematical tool used. The crisp set is defined in such a way as to categorize the individuals in some given universe of discourse into two groups: members (those that certainly belong in the set) and nonmembers (those that do not). A sharp, definite distinction exists between the members and nonmembers of the class or the category described by a crisp set.

Sometimes it happens that this distinction may easily be made since the categories we are working with are precise and well-defined. However, in many engineering tasks, we confront ill-defined classes such as tall man, high temperature, significant, etc. The transition from member to nonmember is gradual rather than abrupt. We need to construct a convenient, flexible and easy-to understand tool for handling fuzziness mentioned above. An idea of fuzzy sets originally formulated by Zadeh formed a necessary step in this direction [Zadeh 1965].

The fuzzy set introduces fuzziness with the goal of reducing complexity by eliminating the abrupt boundary dividing members of the class from nonmembers. It changes the original basic notion of a set, allowing the grades of belonging to vary from complete exclusion (0) up to complete belonging (1), while in set theory only two border values are discussed. The higher the value of the membership of a certain object in the fuzzy set, the stronger the link of the object to the class is reported.

3.1.1 Terminology

In set theory, one can often meet a so-called characteristic function of the set A , $\chi_A(x)$, defined as $\chi_A : X \rightarrow \{0, 1\}$ taking its value in the $\{0, 1\}$ set such that

$$\chi_A = \begin{cases} 1, & \text{if } x \in A \\ 0, & \text{otherwise} \end{cases} \quad (3.1)$$

This function can be generalized so that the values assigned to the elements of the universal set fall within a specified range and indicate the membership grade of these elements in the set.

Let U be a collection of objects denoted generically by $\{u\}$, which could be discrete or continuous. U is called the universe of discourse and u represents the generic element of U .

A fuzzy set F in a universe of discourse U is characterized by a membership function μ_F which takes values in the interval $[0,1]$ namely, $\mu_F : U \rightarrow [0,1]$. A fuzzy set may be viewed as a generalization of the concept of a crisp set whose membership function only takes two values $\{0, 1\}$. Thus a fuzzy set F in U may be represented as a set of ordered pairs of a generic element u and its grade of membership function:

$$F = \{(u, \mu_F(u)) | u \in U\}. \quad (3.2)$$

When U is continuous, a fuzzy set F can be written as

$$F = \int_U \mu_F(u) / u. \quad (3.3)$$

When U is discrete, a fuzzy set F is represented as

$$F = \sum_{i=1}^n \mu_F(u_i) / u_i. \quad (3.4)$$

The support of a fuzzy set F is the crisp set of all points u in U such that $\mu_F(u)$ is positive. In particular, the element u at which $\mu_F = .5$ is called the

crossover point and a fuzzy set whose support is a single point in U with $\mu_F = 1.0$ is called fuzzy singleton. A is a subset of B , denoted $A \subset B$, if and only if $\mu_A(u) \leq \mu_B(u)$, for all u in U .

3.1.2 Fuzzy Set Operations

Let A and B be two fuzzy sets in U with membership functions μ_A and μ_B , respectively. The set operations of union, intersection and complement for fuzzy sets are defined via their membership functions as follows.

The membership function $\mu_{A \cup B}$ of the union $A \cup B$ is pointwise defined for all $u \in U$ by

$$\mu_{A \cup B}(u) = \max \{ \mu_A(u), \mu_B(u) \}. \quad (3.5)$$

The membership function $\mu_{A \cap B}$ of the intersection $A \cap B$ is pointwise defined for all $u \in U$ by

$$\mu_{A \cap B}(u) = \min \{ \mu_A(u), \mu_B(u) \}. \quad (3.6)$$

The membership function μ_A of the complement of a fuzzy set A is pointwise defined for all $u \in U$ by

$$\mu_A(u) = 1 - \mu_A(u). \quad (3.7)$$

If $A_1, \dots, A_n$ are fuzzy sets in $U_1, \dots, U_n$, respectively, the Cartesian product of $A_1, \dots, A_n$ is a fuzzy set in the product space $U_1 \times \dots \times U_n$ with the membership function

$$\mu_{A_1 \times \dots \times A_n}(u_1, u_2, \dots, u_n) = \min \{ \mu_{A_1}(u_1), \dots, \mu_{A_n}(u_n) \} \quad (3.8)$$

or

$$\mu_{A_1 \times \dots \times A_n}(u_1, u_2, \dots, u_n) = \mu_{A_1}(u_1) \cdot \mu_{A_2}(u_2) \cdot \dots \cdot \mu_{A_n}(u_n). \quad (3.9)$$

A fuzzy relation R from a set A to a set B is a fuzzy subset of the Cartesian product $A \times B$. ($A \times B$ is the collection of ordered pairs (a, b) , $a \in A$, $b \in$

B). R is characterized by a bivariate membership function $\mu_R(a, b)$ and is expressed

$$R = \int_{A \times B} \mu_R(a, b) / (a, b). \quad (3.10)$$

More generally, for an n -ary fuzzy relation R which is a fuzzy subset of $X_1 \times X_2 \times \dots \times X_n$, we have

$$R = \int_{X_1 \times \dots \times X_n} \mu_R(x_1, \dots, x_n) / (x_1, \dots, x_n), \quad x_i \in X_i, \quad i = 1, \dots, n. \quad (3.11)$$

If R and S are fuzzy relations in $U \times V$ and $V \times W$, respectively, then the sup-star composition of R and S is a fuzzy relation denoted by $R \circ S$ and defined by

$$R \circ S = \left\{ [(u, w), \sup(\mu_R(u, v) * \mu_S(v, w))] \mid u \in U, v \in V, w \in W \right\} \quad (3.12)$$

where $*$ could be any operator in the class of triangular norms, namely, minimum, algebraic product, bounded product, or drastic product.

3.2 Linguistic Variables

A fuzzy number F in a continuous universe U , is a fuzzy set F in U which is normal and convex, i.e.,

$$\begin{aligned} \max_{u \in U} \mu_F(u) &= 1, \quad (\text{normal}) \\ \mu_F(\lambda u_1 + (1 - \lambda)u_2) &\geq \min(\mu_F(u_1), \mu_F(u_2)), \lambda \in [0, 1]. \quad (\text{convex}) \end{aligned} \quad (3.13)$$

The use of fuzzy sets provides a basis for a systematic way for the manipulation of vague and imprecise concepts. In particular, we can use fuzzy sets to describe linguistic variables. A linguistic variable can be considered either as a variable whose value is a fuzzy number or as a variable whose values are labels of sentences in a natural or artificial language.

More specifically, a linguistic variable is characterized by a quintuple $(x, T(x), U, G, M)$ in which x is the name of variable; $T(x)$ is the term set of x ,

that is , the set of names of linguistic values of x with each value for generating the names of values of x ; and M is a semantic rule for associating with each value its meaning. For example, if speed is interpreted as a linguistic variable, then its term set $T(\text{speed})$ could be $T(\text{speed})=\{\text{slow, medium, fast, very slow, less fast,...}\}$ where each term in $T(\text{speed})$ is characterized by a fuzzy set in a universe of discourse $U=[0,100]$. We might interpret "slow" as "a speed below about 30mph," "medium" as "a speed close to 45mph," and "fast" as "a speed above about 60mph." These terms can be characterized as fuzzy sets whose membership functions are shown in Figure 3-1.

Figure 3-1 Diagrammatic representation of linguistic variable speed.

By using the type of linguistic descriptions employed by humans instead of quantified variables, we acquire a capability to deal with systems which are much too complex to be susceptible to analysis in conventional mathematical terms.

3.3 Fuzzy logic and Approximate Reasoning

Fuzzy logic is concerned with the formal principles of approximate reasoning, with precise reasoning viewed as a limiting case. Unlike conventional logical systems, fuzzy logic aims at modeling the inexact modes of reasoning that play an essential role in the remarkable human ability to make rational decisions in an environment of uncertainty and imprecision. This

ability depends, in turn, on our ability to infer an approximate answer to a question based on a store of knowledge that is imprecise, incomplete, or not totally reliable. The chains in fuzzy logic are short in length, and rigor does not play as important a role as it does in classical logical systems. Everything in fuzzy logic, including truth, is a matter of degree.

In fuzzy logic and approximate reasoning, there are two important fuzzy implication inference rules named the generalized modus ponens (GMP) and the generalized modus tollens (GMT):

premise 1: x is A' ,

premise 2: if x is A then y is B .

consequence: y is B' (GMP)

premise 1: y is B' ,

premise 2: if x is A then y is B .

consequence: x is A' . (GMT)

The fuzzy implication inference is based on the compositional rule of inference for approximate reasoning proposed by Zadeh [Zadeh 1973]. Here we introduce fuzzy sets A , A' , B , B' via linguistic variables x , y instead of crisp sets in the conventional logic. The GMP, which reduces to "modus ponens" when $A'=A$ and $B'=B$, is closely related to the forward data-driven inference which is particularly useful in fuzzy logic control. The GMT, which reduces to "modus tollens" when $B'=\text{not } B$ and $A'=\text{not } A$, is closely related to the backward goal-driven inference which is commonly used in expert systems, especially in the field of medical diagnosis.

If R is a fuzzy relation in $U \times V$, and x is a fuzzy set in U , then the "sup-star compositional rule of inference" asserts that the fuzzy set y in V induced by x is given by

$$y = x \circ R . \quad (3.14)$$

where $x \circ R$ is the sup-star composition of x and R . If the star represents the minimum operator, then this definition reduces to Zadeh's composition rule of inference.

3.4 Fuzzy Logic Control

Fuzzy logic provides an effective method of capturing the approximate, inexact nature of the real world. From this point of view, the essential part of the fuzzy logic controller (FLC) is a set of linguistic control rules related by the dual concepts of fuzzy implication and the compositional rule of inference. In essence, the FLC provides an algorithm which can convert the linguistic control strategy based on expert knowledge into an automatic control strategy. The methodology of the FLC is very useful when the processes are too complex for analysis by traditional qualitative techniques or when the available sources of information are interpreted qualitatively, imprecisely, or uncertainly [Lee 1990].

Many different types of FLCs have been proposed since Dr. Mamdani wrote the first algorithm for a dynamic plant in 1974 [Mamdani 1974]. They are variations of the basic model shown in Figure 3-2.

The FLC consist of four principal components: a fuzzification interface, a knowledge base, decision-making logic, and a defuzzification interface.

The fuzzification interface comprises the following functions:

- (i) measures the physical values of the input variables,

(ii) performs a scale mapping that converts the range of values of input variables into corresponding universes of discourse,

Figure 3-2 Basic Fuzzy Logic Control Model.

(iii) performs the function of fuzzification that transfers input data into suitable linguistic values which may be viewed as labels of fuzzy sets.

The knowledge base consists of a knowledge of the application domain and the attendant control goals. It comprises a "data base" and a "linguistic (fuzzy) control rule base":

(i) the data base supplies necessary definitions, which are used to define linguistic control rules and fuzzy data manipulation in an FLC,

(ii) the rule base characterizes the control goals and control policy of the domain experts by the use of a set of linguistic control rules.

The decision making logic is the kernel of an FLC; it is capable of simulating human decision-making based on fuzzy concepts and inferring fuzzy control actions by using fuzzy implication and the rules of inference in fuzzy logic.

The defuzzification interface involves the following functions:

(i) a scale mapping, which transfers the range of values of output variables into the corresponding universe of discourse,

(ii) defuzzification, which generates a nonfuzzy control action from an inferred fuzzy control action.

Application of such a fuzzy controller to a multilink robot arm is considered in Chapter IV. An example of the simplicity of the design of a fuzzy controller for a typical nonlinear system is shown below.

3.5 Example

3.5.1 System Description

An inverted pendulum is shown as follows (Figure 3-3) :

Figure 3-3 Inverted Pendulum System.

The goal of the controller is to keep the pole, i.e., an inverted pendulum, in the vertical position, i.e., $\theta=0$. The pole is a narrow, uniform rod. There is no friction at the pivot and the cart moves only in one direction along a frictionless track. The dynamic equations of the system are as follows:

$$\ddot{x} = \frac{2u}{2M+m} - \frac{2mg}{2M+m} \tan(\theta) \quad (3.15)$$

$$\ddot{\theta} = \tan(\theta) \cdot \dot{\theta} + \frac{2(M+m) \cdot g \cdot \sin(\theta)}{(2M+m) \cdot l \cdot \cos^2(\theta)} - \frac{u}{(2M+m) \cdot l \cdot \cos(\theta)}. \quad (3.16)$$

We define the state variables of the system as the cart's position, the cart's velocity, the pole's angle and angular velocity, respectively.

$$x_2 = \dot{x} \quad (3.17a)$$

$$x_1 = x \quad (3.17b)$$

$$x_3 = \theta \quad (3.17c)$$

$$x_4 = \dot{\theta}. \quad (3.17d)$$

Assume that all the state variables are available from suitable sensors.

We can derive the state vector differential equation as follows:

$$\dot{x}_1 = x_2 \quad (3.18a)$$

$$\dot{x}_2 = \frac{2u}{2M+m} - \frac{2mg}{2M+m} \cdot \tan(x_3) \quad (3.18b)$$

$$\dot{x}_3 = x_4 \quad (3.18c)$$

$$\dot{x}_4 = \tan(x_3) \cdot x_4 + \frac{2(M+m) \cdot g \cdot \sin(x_3)}{(2M+m) \cdot l \cdot \cos^2(x_3)} - \frac{u}{(2M+m) \cdot l \cdot \cos(x_3)}. \quad (3.18d)$$

If θ is small, we have the following approximation:

$$\begin{aligned} \sin(\theta) &\approx \theta \\ \cos(\theta) &\approx 1 \end{aligned} \quad (3.19)$$

When the angle deviation is small, the system can be linearized and the state equations become

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{2mg}{2M+m} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & \frac{2(M+m)g}{(2M+m)l} & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{2}{2M+m} \\ 0 \\ \frac{1}{(2M+m)l} \end{bmatrix} u. \quad (3.20)$$

We can use a linear controller for the linearized system.

3.5.2 Fuzzy Logic Controller Design

In order to balance the pole in the vertical position, we apply an appropriate horizontal force to the cart. The relationship between the control force u and the state variables can be determined by human operator's knowledge described in natural language. For example, if the pole is leaning slightly to the right in which case the angle deviation is positive small and the angular velocity is about zero, then we need to apply a small force to the right. Let θ and $\dot{\theta}$ be the input and u be the output of the controller, we can design the FLC with the rule base shown in Table 3-1 and the corresponding membership function shown in Figure 3-4.

Table 3-1 The Rule Base for Inverted Pendulum System.

		θ					
		u	NB	NS	ZE	PS	PB
$\dot{\theta}$	NB	NB	NB	NB	NB	NS	PS
	NS	NB	NS	NS	NS	ZE	PS
	ZE	NB	NS	ZE	ZE	PS	PB
	PS	NS	ZE	PS	PS	PS	PB
	PB	NS	PS	PB	PB	PB	PB

NB: negative big
NS: negative small
ZE: zero
PS: positive small
PB: positive big

Figure 3-4 Control Variable Membership Functions.

3.5.3 Simulation Result

The fuzzy logic controller was verified and compared with the linear feedback controller by simulating an inverted pendulum specified by the following parameters:

- $M = 1$ (kg) the mass of the cart,
- $m = .1$ (kg) the mass of the pendulum,
- $L = .25$ (m) the length of the pendulum,
- $g = 9.8$ (m/s²) the gravity constant,
- $T = .01$ s the sample rate.

Figure 3-5 and Figure 3-6 show the simulation result of the linearized system with a linear feedback controller and the fuzzy logic controller respectively. Figure 3-7 and Figure 3-8 show the simulation result of the nonlinear system. Under the same initial conditions, these two comparisons

demonstrate that the fuzzy logic controller needs less settling time and requires less control force (energy) than the linear feedback controller does.

Figure 3-5 Simulation of the linearized system with feedback controller.

Figure 3-6 Simulation of the linearized system with fuzzy logic controller.

Figure 3-7 Simulation of the nonlinear system with feedback controller.

Figure 3-8 Simulation of the nonlinear system with fuzzy logic controller.

CHAPTER IV

FUZZY RULE BASED ROBOT CONTROLLER

Obviously, the matrix algebraic method is very time consuming and expensive because of the generalized and pseudo inverse matrices involved. In case of simultaneous consideration of both types of restrictions, contradictions between them might arise which can't be solved easily. An approach for robot motion control which can eliminate the computational complexity of the inverse kinematics problem and avoid the difficulties of multi-criteria decisions is the use of fuzzy logic. The Cartesian distances, the angular change and the internal and external restrictions are interpreted as linguistic variables. A fuzzy rule base is formulated which can resolve the appropriate joint angles for multi-criteria considerations.

4.1 Design of Fuzzy Logic Controller

A fuzzy logic based robot controller is proposed to solve the inverse kinematics problem using fuzzy reasoning as shown in Figure 4-1. The error between the desired and the actual Cartesian position is the input of the fuzzy rule based controller. The *fuzzifier* maps physical variables measured by external sensors to fuzzy linguistic variables. A linguistic rule base provides the logical connection of various process conditions by using IF... THEN implications within *rule inference*. The *defuzzifier* resolves the crisp physical variables from the outputs of the fuzzy inference. The fuzzy controller gives the joint angle change command to the robot which will lead the robot arm to the desired Cartesian position.

Figure 4-1 Block Diagram of the Fuzzy Rule Based Controller

Consider a planar redundant robot arm with 3 d.o.f. (Fig.4-2).

Figure 4-2 A 3-link Redundant Robot Arm Configuration

We get the end-effector position $x(t)$ and $y(t)$ from the forward kinematics:

$$\begin{aligned} x &= l_1 \cos(\theta_1) + l_2 \cos(\theta_1 + \theta_2) + l_3 \cos(\theta_1 + \theta_2 + \theta_3) \\ y &= l_1 \sin(\theta_1) + l_2 \sin(\theta_1 + \theta_2) + l_3 \sin(\theta_1 + \theta_2 + \theta_3) \end{aligned} \quad (4.1)$$

The differential relationship is

$$\begin{aligned} dx &= A_1 d\theta_1 + A_2 d\theta_2 + A_3 d\theta_3 \\ dy &= B_1 d\theta_1 + B_2 d\theta_2 + B_3 d\theta_3 \end{aligned} \quad (4.2)$$

where

$$\begin{aligned}
A_1 &= -l_1 \sin \theta_1 - l_2 \sin(\theta_1 + \theta_2) - l_3 \sin(\theta_1 + \theta_2 + \theta_3) \\
A_2 &= -l_2 \sin(\theta_1 + \theta_2) - l_3 \sin(\theta_1 + \theta_2 + \theta_3) \\
A_3 &= -l_3 \sin(\theta_1 + \theta_2 + \theta_3) \\
B_1 &= l_1 \cos(\theta_1) + l_2 \cos(\theta_1 + \theta_2) + l_3 \cos(\theta_1 + \theta_2 + \theta_3) \\
B_2 &= l_2 \cos(\theta_1 + \theta_2) + l_3 \cos(\theta_1 + \theta_2 + \theta_3) \\
B_3 &= l_3 \cos(\theta_1 + \theta_2 + \theta_3)
\end{aligned} \tag{4.3}$$

In order to move the end effector from the initial position to the desired position, we need to find out the angle change $d\theta_i$ ($i=1,2,3$). From the above equations, we know that there is a linear relationship between the joint angle change and the Cartesian position change for small joint angle deviation ($dq \ll 1$). So each angle change $d\theta_i$ can be determined independently. First, consider $d\theta_1$ from the following equation:

$$\begin{aligned}
dx &= A_1 d\theta_1 \\
dy &= B_1 d\theta_1
\end{aligned} \tag{4.4}$$

dx and dy are the desired moves of the end effector along the X and Y axes. We can see that if dx (dy) is positive then $d\theta_1$ must be positive for positive A_1 (B_1) and negative for negative A_1 (B_1). Since A_i and B_i can be known from joint angle measurement via sensors and given robot link lengths, we let

$$\begin{aligned}
dx' &= dx \cdot \text{sgn}(A_1) \\
dy' &= dy \cdot \text{sgn}(B_1)
\end{aligned} \tag{4.5}$$

where $\text{sgn}(x)$ is the sign of the variable x . The above relationships can be simply interpreted as that if dx' (dy') is positive then $d\theta_1$ is positive, and if dx' (dy') is negative then $d\theta_1$ is negative. These relationships can be formulated

by introducing the linguistic variables and fuzzy logic rule base. Define dx' , dy' , $d\theta_1$ as the following linguistic variables:

$$dx', dy', d\theta_1 = \{NB, NS, ZE, PS, PB\} \quad (4.6)$$

where NB, NS, ZE, PS and PB denote Negative Big, Negative Small, Zero, Positive Small, and Positive Big respectively. Their corresponding membership functions are shown in Figure 4-3. $d\theta_1$ is needed to be determined from dx' and dy' . By inspecting the fact that a positive $d\theta_1$ will cause a positive dx' as well as a positive dy' and a negative $d\theta_1$ will cause negative dx' and dy' , we can infer

Figure 4-3 Membership functions

such fuzzy statements about $d\theta_1$ from desired dx' and dy' as: if dx' is positive big and dy' is positive big then $d\theta_1$ should be positive big, if dx' is positive

Table 4-1: Fuzzy rule base table

		dx'				
		NB	NS	ZE	PS	PB
dy'	NB	NB	NS	ZE	PS	PB
	NS	NB	NS	ZE	PS	PB
	ZE	NB	NS	ZE	PS	PB
	PS	NS	ZE	PS	PS	PB
	PB	NS	PS	PB	PB	PB

small and dy' is negative small then $d\theta_1$ should be zero. Table 4-1 summarizes all the possible fuzzy statements.

With $d\theta_1$ determined, $d\theta_2$ can be calculated from the following equation:

$$\begin{aligned} dx - A_1 d\theta_1 &= A_2 d\theta_2 \\ dy - B_1 d\theta_1 &= B_2 d\theta_2 \end{aligned} \quad (4.7)$$

Let

$$\begin{aligned} dx'' &= \text{sgn}(A_2) \cdot (dx - A_1 d\theta_1) \\ dy'' &= \text{sgn}(B_2) \cdot (dy - B_1 d\theta_1) \end{aligned} \quad (4.8)$$

Similarly, we can define dx'' , dy'' and $d\theta_2$ as linguistic variables with the same membership functions shown in Figure 4-3. Replacing dx' and dy' with dx'' and dy'' respectively in the fuzzy rule base, we can calculate $d\theta_2$ using the same fuzzy reasoning method.

Finally, $d\theta_2$ is determined according to

$$\begin{aligned} dx''' &= \text{sgn}(A_3) \cdot (dx - A_1 d\theta_1 - A_2 d\theta_2) \\ dy''' &= \text{sgn}(B_3) \cdot (dy - B_1 d\theta_1 - B_2 d\theta_2) \end{aligned} \quad (4-9)$$

With only 25 rules, we can easily solve the inverse kinematics problem. The same membership functions and fuzzy rules can be applied to an n-d.o.f redundant robot arm. When high d.o.f are considered, the computational effort can be dramatically reduced.

4.2 A Modified FLC with regard to Internal and External Restrictions

For an actual industrial robot, there are some internal restrictions. Each joint angle cannot exceed certain minimum boundaries q_{im} and maximum boundaries q_{iM} as shown in Figure 4-4.

Figure 4-4 Internal Restrictions

Let β_i denote the distance from the boundaries:

$$\beta_i = \min(\theta_{iM} - \theta_i, \theta_i - \theta_{im}) . \quad (4-10)$$

We want to keep each joint angle away from those boundaries. So a move toward the boundaries should be restricted somehow while a move toward the middle positions should not be so. A scaling factor h_i is introduced to modify $d\theta_i$ resolved from the fuzzy controller:

$$h_i = 1 - \exp^{-\frac{\beta_i^2}{2\sigma^2}} (1 + \operatorname{sgn}(d\theta_i / \theta_i)) . \quad (17)$$

Obviously, when $d\theta_i$ and θ_i have the same sign, the angle movement is toward the boundaries. So $d\theta_i$ should be scaled by h_i ($h_i < 1$) which is an increasing function of β_i . When $d\theta_i$ and θ_i have different signs, the angle movement is away from the boundaries. Since $h_i = 1$, there will be no restriction on $d\theta_i$.

With regard to external restrictions a rectangular obstacle shown in Figure 4-5 is considered [Palm 1992]. Let

Figure 4-5 External restrictions

s be the distance between the end-effector and the upper boundary of the obstacle. We can define s as a linguistic variable: $s=\{B, S\}$ where B and S

Table 4-2 Fuzzy rule base for avoiding obstacle
if s is big:

		dx'				
$d\theta_1$		NB	NS	ZE	PS	PB
dy'	NB	NB	NB	NB	NS	PS
	NS	NB	NS	NS	ZE	PS
	ZE	NB	NS	ZE	PS	PB
	PS	NS	ZE	PS	PS	PB
	PB	NS	PS	PB	PB	PB

if s is small:

		dx'				
$d\theta_1$		NB	NS	ZE	PS	PB
dy'	NB	ZE	PS	PB	PB	PB
	NS	NS	ZE	PS	PS	PB
	ZE	NB	NS	ZE	PS	PB
	PS	NB	NS	NS	ZE	PS
	PB	NB	NB	NB	NS	ZE

denote Big, and Small respectively. When s is big, the robot can be guided using the previous rule base. When the end-effector is close to the boundary of the external obstacle, i.e., s is small, and any movement along the negative y direction should be restricted until the end-effector passes the right boundary.

Based on these facts, the fuzzy rules shown in Table 4.2 are defined to guide the robot arm to reach the target while avoiding the obstacle.

CHAPTER V

SIMULATION RESULTS

The fuzzy logic approach mentioned above was verified and compared with the matrix algebraic method discussed in Chapter II by simulating a 3-link planar robot. The lengths of the links are all .6m. The internal restrictions q_{1m} and q_{1M} are

$$q_{1m}=0. , \quad q_{2m}=-2.7, \quad q_{3m}=-2.7, \quad q_{1M}=2.9 , \quad q_{2M}=2.1, \quad q_{3M}=2.9.$$

Figure 5-1 and Figure 5-2 show the motion of the robot arm while tracking a user defined straight line ($x=1.3$) using the matrix algebraic method and fuzzy logic approach, respectively. It takes 600 iterations using the conventional method but only about 200 iterations using fuzzy logic.

For obstacle avoidance using the conventional method, we have to predefine a trajectory by specifying some intermediate points to make the robot arm avoid the obstacle. But we cannot guarantee that none of the links will hit the obstacle while following the desired trajectory. Figure 5-3 shows that the robot arm fails to avoid the obstacle while trying to reach the target. The first link hits the obstacle. Figure 5-4 shows the simulation of obstacle avoidance using fuzzy logic. The robot arm is able to reach the target without hitting the rectangle obstacle.

Figure 5-5 shows the target position control simulation using fuzzy logic. The robot arm was able to move from the initial position at $X_0=1.1m$, $Y_0=1.3m$, to the desired position at $X_0=-.5m$, $Y_0=.2m$ exactly without

predefining the trajectory and solving the inverse kinematics problem. Also the angle movements are smoother than those in the conventional method.

Figure 5-1 Trajectory tracking using conventional method.

Figure 5-2 Trajectory tracking using fuzzy logic.

Figure 5-3 Obstacle avoidance using conventional method.

Figure 5-4 Obstacle avoidance using fuzzy logic.

Figure 5-5 Target position control using fuzzy logic.

CHAPTER VI

CONCLUSION

We have proposed a model independent approach using fuzzy logic to resolve the joint angle motion which results in user defined end effector trajectories of the redundant robot arm. The complexity and computation burdens of the pseudo-inverse of the Jacobian matrix in the traditional method are avoided. The Cartesian distances, the angular change, internal and external restrictions are denoted as fuzzy terms. A fuzzy rule base is formulated which enables the robot arm to avoid the obstacles while following a desired path or trying to reach a target. This fuzzy logic approach can also be extended easily to a high d.o.f. robot arm.

For further studies, we are going to implement other conventional methods such as iterative and geometric methods for multilink robot arm control and compare them with the fuzzy logic controller. Also, we are going to apply the fuzzy logic approach to robot arm dynamic control.

REFERENCES

- Denavit 55 J. Denavit and R.S. Hartenberg, " A Kinematic Notation for Lower-Pair Mechanisms Based on Matrices," Journal of Applied Mechanics, ASME, 22, pp. 215-221, 1955.
- Denavit 56 J. Denavit, "Description and Displacement Analysis of Mechanism Based on 2×2 Dual Matrices," Ph.D. Thesis, Mechanical Engineering, Northwestern U., Evanston, Ill, 1956.
- Fu 87 K.S.Fu, R.C Gonzalez and C.S.G. Lee, Robotics: Control, Sensing, Vision, and Intelligence, McGraw-Hill, Highstown, NJ, 1987.
- Kim 93 S. W. Kim and J. J. Lee, " Resolved Motion Rate Control of Redundant Robots using Fuzzy Logic," IEEE International Conf. on Fuzzy Sys., pp. 333-338, 1993.
- Klir 88 G.J. Klir and T.A. Folger, Fuzzy Sets, Uncertainty, and Information, Prentics Hall, Englewood Cliffs, NJ, 1988.
- Lee 84 C.S.G. Lee and M. Ziegler, "A Geometric Approach in Solving the Inverse Kinematics of PUMA Robots," IEEE Trans. Aerospace and Electronic Systems, vol. 6, pp. 695-706, 1984.
- Lee 90 C.C. Lee, "Fuzzy Logic in Control Systems: Fuzzy Logic Controller - Part I and Part II," IEEE Trans. on Sys., Man Cyb. vol. 20, No. 2, pp. 404-435, 1990.
- Liegeois 77 A. Liegeois, "Automatic Supervisory Control of the Configuration and Behavior of Multibody Mechanisms," IEEE Trans. on Sys., Man Cyb., vol. 7, pp. 868-871, 1977.
- Lovass-Nagy 87 V. Lovass-Nagy and R.J. Schilling, "Control of Kinematically Redundant Robots," IEEE Trans. Systems, Man Cybernet. 17(4), pp. 644-649, 1987.

- Mamdani 75 E.H. Mamdani and S. Assilian, "An Experiment in Linguistic Synthesis with a Fuzzy Logic Controller," *International Journal of Man-Machine Studies*, pp. 1-13, 1975.
- Moore 35 E.H. More, "Generalized Analysis," *Mem. Amer. Philos. Soc.*, vol. 1, pp. 1-231, 1935.
- Nakamura 91 Y. Nakamura, *Advanced Robotics-Redundancy and Optimization*, Addison-Wesley, Reading, MA, 1991.
- Nedungadi 92 A. Nedungadi, "A Fuzzy Robot Controller - Hardware Implementation", *IEEE International Conf. on Fuzzy Sys.*, pp. 1325-1331, 1992.
- Palm 92 R. Palm, "Control of a Redundant Manipulator using Fuzzy Rules," *Fuzzy Sets and Systems*, vol. 45, pp. 279-298, 1992.
- Penrose 55 R. Penrose, "A Generalized Inverse for Matrices," *Proc. Cambridge Phil. Soc.*, vol. 51, pp. 406-413, 1955.
- Schilling 90 R.J. Schilling, *Fundamentals of Robotics Analysis & Control*, Prentice Hall, Englewood Cliffs, NJ, 1990.
- Uicker 64 J.J. Uicker, J. Denavit, and R.S. Hartenberg, "An Iterative Method for the Displacement Analysis of Spatial Mechanisms," *Trans. ASME, J. Appl. Mech.*, vol. 31, pp. 309-314, 1964.
- Whitney 69 D.E. Whitney, "Resolved Motion Rate Control of Manipulators and Human Prostheses," *IEEE Trans. Man-Machine Sys.*, vol. 2, pp. 47-53, 1969.
- Zadeh 65 L. A. Zadeh, "Fuzzy Sets," *Information and Control*, vol. 8, pp. 338-353, 1965.
- Zadeh 73 L. A. Zadeh, "Outline of a New Approach to the Analysis of Complex Systems and Decision Processes," *IEEE Trans. Syst., Man, Cybern.* 3, pp. 28-44, 1973.