Special Issue Published in International Journal of Trend in Research and Development (IJTRD),
ISSN: 2394-9333, www.ijtrd.com

The Application of Fuzzy Logic Methods to Control the Robots

[1]Zindhu.S and [2]Swapna. N,
[1]Assistant professor,[2]Student,
[1,2]Department of Computer Science, St.Joseph's College of Arts and Science for Women, Hosur, Tamilnadu, India

Abstract **-** Fuzzy controllers exhibit a simple and robust framework for specifying control laws that accommodate uncertainty and imprecision. Autonomous robot systems require complex control systems. The results showed that Fuzzy Logic uses less memory than binary logic and is much easier to design, although more difficult to program initially. Fuzzy Logic can control more functions of the robot and has greater processing capabilities [1]. The power, ease of use, and small size of Fuzzy Logic instructions make Fuzzy Logic a practical solution to autonomous robotic control systems. The three applications of fuzzy logic for robotic systems are presented. The basic behaviors developed in this research are designed based on fuzzy control techniques and are integrated and coordinated to from complex robotics system. A robot task can be defined by the user and executed by the intelligent robot control system.

Keywords *- Fuzzy Logic method; Robot, Manipulator, Controller, Mobile Robot Controls.*

I. INTRODUCTION

The expansion of robotics and microcontrollers into the facts of everyday life increase the needs to develop efficient control systems. Fuzzy logic is a logical system that aims at a formalization of approximate reasoning.

The research analyzed the two systems using four criteria:

- The size of memory required to develop the control system,
- The ease of writing the control software.
- how well the control system managed the functions of the robot and
- The overall processing power of the system.

Fuzzy Logic is a multivalve logic that allows intermediate values to be defined between conventional evaluations like yes/no or true/false. Fuzzy Systems are an alternative to traditional notions of set membership and logic that has its origins in ancient Greek philosophy [3]. Fuzzy Logic makes it possible to solve complex, ill defined problems where there is a large degree of expert knowledge or the solution is easy to describe linguistically. The goal of this project is to design and built an autonomous line following robot based on fuzzy logic techniques. The robot uses the Motorola HC68HC12 microcontroller.

The instruction set contains the fuzzy logic operations of trapezoidal membership, rule evaluation and weighted average defuzzification. Fuzzy control provides a mechanism for incorporating human-like reasoning capabilities and computationally in control systems. The linguistic variables are used to mimic the human action into a system more closely than traditional control.

II. FUZZY LOGIC ON THE HC12

Motorola designed the HC12 with advanced capabilities to handle fuzzy logic calculations. The HC12 contains 4 instructions that are specific to fuzzy logic. These instructions are:

- MEM-Evaluates the trapezoidal membership functions.
- REV/REVW–Performs unweighted/weighted.
- WAV-Performs weighted average defuzzification.

The following section describes the basics of fuzzy logic on the Motorola HC12.

A. Fuzzy Logic Basics

The design of a fuzzy logic interface for the HC12 consists of two parts [2]. First the user must design a knowledge base that contains the membership functions and the rule set.

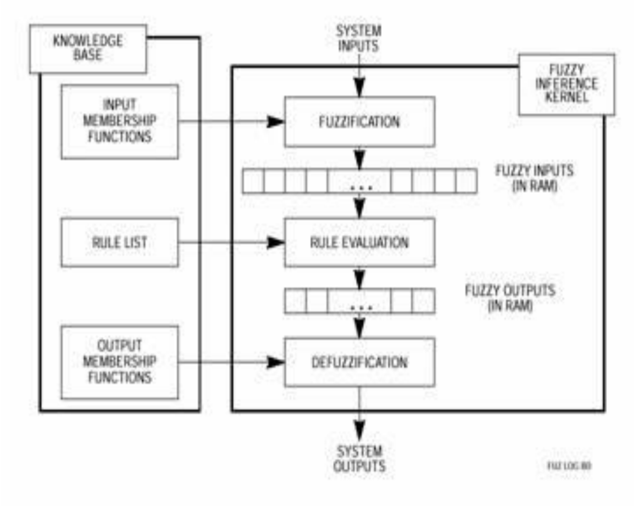

Figure 1 : Block diagram of a fuzzy logic system.

The second part is the inference kernel that takes the system inputs and produces the system outputs based on the knowledge base. The figure shows the basic structure of the fuzzy logic system.

B. Fuzzification Strategy [MEM]

Fuzzification is the process by which system inputs are evaluated to determine the degree at which they belong to a particular fuzzy set. The scale sets from 00 to FF in hexadecimal. The MEM instruction evaluates trapezoidal membership functions. These functions define fuzzy sets, the foundation of fuzzy logic. A fuzzy set is a set without a crisp, clearly define the boundary. It contains elements with only a partial degree of membership.

To define a trapezoidal membership function for the HC12. They are 4 values

- The start of the trapezoid.
- The first slope.
- Second slope.
- The end point of the trapezoid.

The user should define the trapezoid in memory as follows:

LABEL_MFDC.B $40,$D0,$08,$04.

The program should use a descriptive label because the programme will need this label during fuzzification.

C. *Rule definition and Evaluation [REV/REVW]*

Rule evaluation is how fuzzy logic performs calculations. The fuzzy values produced by the MEM function are passed through the rule list to find the fuzzy output. The two types of rules that the HC12 allows are weighted (REVW), where each rule can have a different weights and unweighted (REV) were all rules have equal weight.

An example of a rule list:

a. If temperature is COLD and wind is HIGH, then heat is on HIGH.
b. If temperature is WARM and wind is LOW, then heat is on LOW.
c. If temperature is HOT and wind is HIGH, then heat is on HIGH.

After the fuzzy inputs are evaluated with REV/REVW, the system's fuzzy outputs indicate the degree to which an output should have a specific value. These outputs must then undergo defuzzification before their values are useful.

D. *Defuzzification Strategy [WAV]*

The final step is the fuzzy logic calculation is defuzzification, when the raw fuzzy outputs are evaluated to create a composite system output. Unlike the input, the fuzzy output membership function is not trapezoidal but a singleton. This singleton indicates one system output value for each fuzzy output[2]. The output membership singleton is arraigned in memory in the same order as their corresponding fuzzy output.

WAV calculates a sum of products of each fuzzy output values times its singleton value and a sum of all of the fuzzy output values. The first sum is divided by the second using EDIV to produce an overall value that is the defuzzification output of the system. Defuzzification creates a weighted average system output based on the truth of the fuzzy outputs.

E. *Fuzzy Inference Kernel*

The inference kernel contains all of the instructions that make up the fuzzy system. The kernel utilities the knowledge base to create a system output from given system inputs. All the fuzzy Instructions require proper initialization of the accumulators, index registers and fuzzy values.

III. FUZZY METHOD

Fuzzy logic deals with uncertainly in engineering by attaching degrees of certainly to the answer to a logical question. Commercial fuzzy logic has been used with great success to control machines a consumer products. In this applications FL system are simple to design. It can be understood and implemented by non-specialists in control theory. In most cases someone with an intermediate technical background can design specialists in control theory and it can also design engineers also use it in applications where the on-board computing is very limited and adequate control is enough. FL is not the answer to all technical problems but for control problems where simplicity and speed of implementation is important then FL is a strong candidate.

A. *Define of Membership Function*

A Fuzzy set F in a space of points S= {s} is a set of elements with a varying grade of membership and is characterized by a membership function that maps each element of S to a real number in the interval[0 1]. The value of for any given s indicates the degree of s in F or the degree an s belongs to them.

B. *Fuzzification of Input*

Fuzzification of inputs is necessarily determining the degree to which they belong to each of the appropriate fuzzy sets via membership functions.

C. *Defining of Rules*

The next step is laying down certain rules, which relate the inputs to an output parallel nature of the rules is one of the most important aspects of FL systems. The transition from a region where the system's behavior is dominated by one rule to a region where another dominates it is smooth, avoiding sharp switching between modes based on breakpoints.

D. *Fuzzy Inference*

The decisions are based on the testing of all of the rules; the rules must be combined in some manner in order to make a decision. Aggregation is the process by which the fuzzy sets that represent the outputs of each rule are combined into single fuzzy sets. The output of the aggregation process is one fuzzy set for each output variables. All the rules are evaluated together and the output of each rule is combined or aggregated into a single fuzzy set whose membership function assigns a weighting for every output value.

E. *Defuzzification of the Output*

The defuzzification process transforms the fuzzy set single number. The aggregate of a fuzzy set encompasses a range of output values. It must be defuzzified in order to resolve a single output value from the set. This defuzzification method could employ methods like-centric, bisector, middle, largest and smallest of maximum.

IV. FL CONTROLLER

Fuzzy Logic control is a control algorithms based on a linguistic control strategy, it is derived from expert knowledge into an automatic control strategy. It doesn't need any difficult mathematical calculation like the other control systems. While the other control system use difficult mathematical calculation to provide model of the controlled plant it only uses simple mathematically calculation to simulate the expert knowledge. Although it does not need any difficult mathematical calculation, but it can give good performance in a control system.

A fuzzy logic control usually consists of the following as in the figure

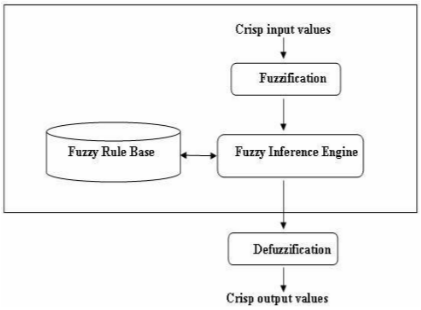

Figure.2 Fuzzy Logic controller

- *Fuzzification:* This process converts or transforms the measured inputs called crisp values, into the fuzzy linguistic values used by the fuzzy reasoning mechanism.

- *Knowledge Base:* A collection of the expert control rules needed to achieve the control goal.

- *Fuzzy Interface Engine:* This process will perform fuzzy logic operations and result the control action according to the fuzzy inputs.

- *Defuzzification Unit:* This process converts the result of fuzzy reasoning mechanism into the required crisp value.

V. FL ROBOTIC MANIPULATORS

The control engineers began their work with a mathematical model and did not acquire further knowledge of the system. Today the control engineers use all of the above sources of information. A relatively new concept, FL is being used in many engineering applications because it is considered by designers to be the simplest solution available for the specific problem. One more benefit of fuzzy controllers is they are basically non- linear and effective enough to provide the desired non-linear control actions by carefully adjusting their parameters.

A. Single Link Flexible Manipulator

A flexible link are is a distributed parameter system of infinite order, but it must be approximated by a lower-order model and controlled by a finite –order controller due to onboard computer limitations and sensor in accuracy[5]. The so-called “control spill over” and “observation spill over” effects then occur, which under certain conditions can lead to instability.

Several control schemes have recently been proposed for flexible robot arms a controller that is based upon a reduced-order model is proposed in to maintain reasonable computational loading. The aim is to simplify the software and hardware implementation of control algorithms while improving their robustness. A composite control approach, based on a two-time scale model of the flexible link arm has been derived in, and allows a definition of a slow subsystem that corresponds to a rigid body and a fast subsystem that describes the exile motion.

- *Mobile Manipulators*

Mobile manipulators are combined systems consists of a robotic manipulators mounted on a mobile platform. These systems are able to accomplish complicated tasks in large workspaces. It gives FL advantages over more traditional solutions for control of mobile robots is that it allows computers to reason more like humans, responding effectively to complex inputs to deal with linguistics notions such as “too hot” , “too cold” or “ just right”. Such systems can be easily upgraded by adding new rules to improve performance.

Fuzzy control can be used to improve existing traditional control systems in mobile robots by adding an extra layer of intelligence to the current control method. It combines some useful heuristic rules with the fuzzy resulted in the desired mapping between perception and motion. It provides much faster response to unexpected events. They are carried out simulations on a non-holonomic mobile robot to test the performance of the proposed fuzzy controller. The main feature of their method is that a priori assignment of fuzzy rules is not necessary and an online aligned fuzzy-logic method is proposed to generate rules automatically. Furthermore, a new FL method is developed for path control of mobile robot by means of a ceiling-mounted camera which observes the robot’s work space.

B. Two Links Flexible Manipulator

In some control tasks, such as those in robot manipulation, the systems to be controlled have constant or slowly-time varying uncertain parameters constant or slowly-time varying uncertain parameters reduced on-line by an appropriate adaptation or estimation mechanism. It may cause inaccuracy or instability for the control systems. In many task such as those in power systems, the system dynamics may have well known dynamics at the beginning. The initially appropriate controller design may not be able to control the changing plant well. In order to compensate the parametric uncertainties of the system, they used the FL system that has the capability to approximate any nonlinear function over the compact input space. A FL control strategy incorporates two fuzzy controllers substituted for the LQR state-space dynamic equations and a Linear Quadratic Gaussian strategy controls a two-link flexible robot manipulator tracking a two-dimensional square trajectory. Two fuzzy control schemes for a class of uncertain continuous-time multi-input multi-output nonlinear dynamical systems were derived.

The flexible manipulator system is firstly decomposed into two subsystems by modelling the joint angles and the corrected flexible modes as the slow and fast variables based on the singular perturbation method and two time-scale decomposition[7]. A non-singular terminal sliding mode manifold is proposed for the slow subsystem to realize fast convergence and better tracking precision. Meanwhile, a hybrid controller for the slow subsystem is proposed to ensure strong robustness as well as to weaken chattering phenomenon using fuzzy logic. Fuzzy Pre-compensated Proportional-Derivative controller in trajectory control of a two link rigid-flexible manipulator. FL control of a two-link flexible manipulator using genetic optimization techniques. Hence, dynamic modelling of the constrained two-link flexible manipulator is derived via finite element method, a Genetic Algorithm based hybrid fuzzy logic control strategy is also developed to reduce the end-point vibration of a flexible manipulator without sacrificing its speed of response.

VI. DEVELOPMENT OF BASIC BEHAVIORS

The architecture used to build and embody the robot behaviors consists of several fuzzy controllers. These behaviors include emergency, avoid-obstacles, left wall-following, right wall-followings and move-point.

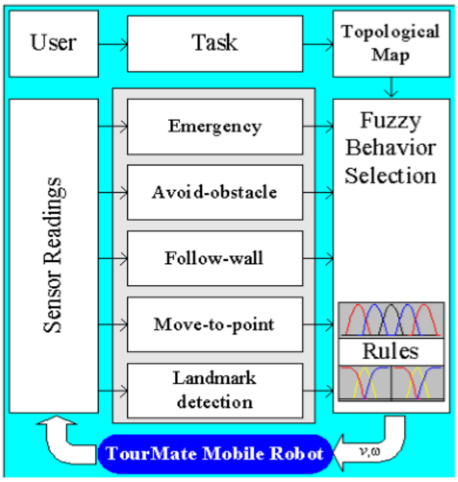

Figure.3 Intelligence Embodiment Architecture.

The behavior-based control architecture, as shown in Fig 3. It is organised vertically which shows that each behavior has full access to all sensor readings and process its own commands. The output of each behavior is the linear velocity and angular velocity of the mobile robot. The fuzzy rules are defined based on the tasks. The final robot command is dependent on the fuzzy selection that integrates and coordinates all behaviors.

A. *Emergency Behavior*

The most fundamental behavior that all mobile robots should have is the emergency behavior. It should have the highest priority in taking the control of the robot. In some situations, avoid-obstacles behavior may not function properly. The robot move too close to an obstacle. An emergency behavior is required to stop the robot or even, in some instances, move the robot backward. These two sub-behaviors are combined together to from one behavior called emergency behavior.

B. *Avoid-Obstacles Behavior*

A basic need for all autonomous mobile robots is an obstacles avoidance behavior. It is considered as a basic behavior that uses four sonar sensors on the front-left and four sonar sensors on the front-right. The fuzzy controller for the avoid-obstacles behavior is designed based on two fuzzy inputs receiving information from two symmetric sonar sensors one from the left another one from the right. The sonar readings provided by the eight sonar sensors on the front-left and front-right of the robot. Each fuzzy controller has two inputs and receives reading from a pair of sensors. Each sensor sends data directly to the controller input.

C. *Move-To-Point Behavior*

The next behavior developed is the move-to-point behavior that consists of angle and distance controller. It receives from the user a target position and orientation and computes the angle and the distance to the target from the current position of the robot. The angle controller keeps robot heading to the target while distance controller regulates the distance difference between the robot's current position and the target. Table lists the set of rules used for switching between behaviors relying on information gathered by the robot sonar sensors.

The design process of these controllers is presented in the following. A target point is specified by the user as an x-y position. The angle and distance to the target point is calculated with respect to the robot's current position. These two values are used by the fuzzy controllers as error values in I/O of fuzzification and defuzzification. Angle Fuzzy Controller : The angle error is computed and supplied as input to the angle fuzzy controller. Each membership function is a Gaussian curve velocity. The output of this controller is called the angular velocity.

Distance Fuzzy Controller: It is similar to the angle fuzzy controller in that it has one fuzzy input. The distance error is received as input by the distance fuzzy controller and its computes the required linear velocity for the robot to reach its final target. It also constructed seven Gaussian membership functions for the distance error input.

D. *Wall-Following Behavior*

For indoor mobile robot navigation wall-following behavior is essential. The sonar sensors on the left of the robot are used for wall-following behavior on the left side of the robot and the sonar sensors on the right of the robot are used for wall-following behavior on the right side of the robot. Using fuzzy logic control, the distance between the robot and the wall is regulated. While the distance error is close to zero, the robot will move forward and follow a wall. Two fuzzy controllers are developed for the wall-following behavior, one for left side another for the right side of the robot. The robot will be forced by this behavior to follow the closet wall to its two sides. Sonar readings on both left and right sides of the robot are used to develop the wall-following behavior.

CONCLUSION

In this paper, fuzzy-logic as an appropriate approach to control of mechanical robot manipulators. This method is explained and a review on applications of method in control of mobile robot and flexible links manipulators is presented. It provides a different way to approach a control or classification problem because this method focuses on what the system should do. Hence, one can concentrate on solving the problem rather than trying to model the robot system mathematically. Testing results showed that fuzzy behaviors made the robot move intelligently and adapt to changes in its environment. Four basic robot behaviors were designed using fuzzy control techniques. Each behavior was developed, implemented and tested separately. The individual testing results showed successful performance of each behavior.

Future work shall include development of more fuzzy behavior, generate more testing results and implementation of adaptation mechanism into the fuzzy behaviors using Genetic Algorithms.

ACKNOWLEDGEMENT

First of all, I am glad to thank THE LORD ALMIGHTY for giving me the spirit in completing this paper. I would thank my family for the constant support they provided throughout my preparation.

References

[1] Alavandara S, Jainb T and Nigamc M J (2013), "Bacterial Foraging Optimized Hybrid Fuzzy Precompensated PD Control of Two Link Rigid-Flexible Manipulator", International Journal of Computational Intelligence Systems, Vol. 2, No. 1, pp. 51-59.

[2] Green A and Sasiadek J Z (2001), "Fuzzy and Optimal Control of a Two-Link Flexible Manipulator", International Conference on Advanced Intelligent Mechatronics, Vol. 2, pp. 1169-1174.

[3] R.C.Arkin. Behavior-Based Robotics. MIT Press, Cambridge Massachusetts.

[4] R.A.Brooks. Cambrian Intelligence: The Early History of the New AI. The MIT Press, Cambridge, Massachusetts, 2015.

[5] Cheng J, Zhang Y and Wang Zh. (2015), “Backward Tracking Control of Mobile Robot with One Trailer via Fuzzy Line-ofSight Method”, International Confrence on Fuzzy System and Knowledge Discovery, Vol. 4, pp. 66-70.

[6] M. Nikodem and B. Wojciechowski. ―Upper Bounds on Network Lifetime for Clustered Wireless Sensor Networks‖. NTMS, pages 1-6, 2011.

[7] L.A. Zadeh, "Fuzzy algorithms," Info. & Ctl., Vol. 12, 1968, pp. 94 – 102.

[8] I. Ahrns, J. Bruske, G. Hailu, and G. Sommer. Neural fuzzy techniques in sonar-based collision avoidance. In L. C. Jain and T. Fukuda, editors, Soft Computing for Intelligent Robotic Systems, chapter 8, pages 185214. Physica-Verlag Heidelberg, Germany.

[9] Crespo, F.J., Sicicila, M.A., Cuadrado, J.J., On the use of fuzzy regression in parametric software estimation models: integrating imprecision in COCOMO cost drivers. WSEAS Transactions on Systems.

[10] USING FUZZY LOGIC MODEL, Malaysian Journal of Computer Science, Vol. 20(2)