1-1-2008

Reconfigurable fuzzy logic system for high-frame rate stereovision object tracking

Oleg Samarin
Ryerson University

RECONFIGURABLE FUZZY LOGIC SYSTEM FOR HIGH-FRAME RATE STEREOVISION OBJECT TRACKING

by

Oleg Samarin, B. Eng
Tel-Aviv University, 1996

A project
presented to Ryerson University
in partial fulfillment of the
requirement for the degree of
Master of Engineering
in the Program of
Electrical and Computer Engineering

Author's Declaration

I hereby declare that I am the sole author of this thesis.

I authorize Ryerson University to lend this thesis to other institutions or individuals for the purpose of scholarly research.

I further authorize Ryerson University to reproduce this thesis by photocopying or by other means, in total or in part at the request of other institutions or individuals for the purpose of scholarly research,

Instructions on Borrowers

Ryerson University requires the signatures of all persons using or photocopying this thesis. Please sign below, and give address and date.

Name	Signature	Address	Date

RECONFIGURABLE FUZZY LOGIC SYSTEM FOR HIGH-FRAME RATE STEREOVISION OBJECT TRACKING SYSTEM

Oleg Samarin

Master of Engineering

Department of Electrical and Computer Engineering

Ryerson University, Toronto, 2008

Abstract

This study investigates the applicability of fuzzy logic control to high-frame rate stereovision object tracking. The technology developed in this work is based on utilizing a disparity map produced by the Stereovision Tracking System (STS) to identify the object of interest. The coordinates of the object are used by the fuzzy logic control system to provide rotation and focus control for object tracking. The fuzzy logic control was realized as a reconfigurable hardware module and implemented on Virtex-2 FPGA platform of the STS. The fuzzy reasoning was implemented as a reconfigurable look-up table residing in FPGA's internal memory. A set of software tools facilitating creation of look-up table and reconfiguration of fuzzy logic control system was developed. Finally, the experimental prototype of the system was built and tested. Experimental results have demonstrated that the fuzzy logic system was able to provide fast and accurate control for the tracking of static and dynamic objects.

Acknowledgments

I would like to express my deep gratitude to my supervisor Dr. Lev Kirischian, whose expertise, knowledge, patience, and understanding added considerably to my graduate experience. This work would not be possible without his motivation, vision, and advice at times of critical need. His personal guidance and involvement made him more than a professor to me, but rather, a mentor and a role model.

Many thanks to the members of the review committee for their participation and consideration. A special thank you to Dr. Vadim Geurkov, for the assistance he provided during my graduate studies.

I would also like to express my gratitude to the members of Embedded Reconfigurable Systems Lab (ERSL), especially Ph.D. student Mr. Valery Kirischian and M.Sc. student Mr. Jamin Islam for their assistance during the various phases of this work. This project would not be possible without their hard work and dedication in developing a system that served as a foundation for this work.

I would also like to thank the Department of Electrical and Computer Engineering for providing facilities and technical resources needed for this work.

Last but not least, I would like to thank my family for the support they provided during my graduate education.

Contents

List of Tables

List of Figures

List of Abbreviations

STS	Stereovision Tracking System
DPML	Dynamic Programming Maximum Likelihood
PID	Proportional-Integral-Derivative
FL	Fuzzy Logic
FS	Fuzzy Set
FR	Fuzzy Reasoning
FI	Fuzzy Inferencing
FLS	Fuzzy Logic System
FLC	Fuzzy Logic Controller
MF	Membership Function
FPGA	Field Programmable Gate Array
CPLD	Complex Programmable Logic Device
RSS	Root Sum Square
FRM	Fuzzy Rule Matrix
FR	Fuzzy Reasoning
RSS	Root-Sum-Square
LUT	Look-Up Table
ACG	Architectural Configuration Graph
ISE	Integrated Software Environment
VHDL	Very High Speed Integrated Circuit (VHSIC) Hardware Description Language
TCP	Transmission Control Protocol
FCS	Fuzzy Control Space
SPI	Serial Peripheral Interface
SMC	Servo-Motor Controller

Chapter 1

Introduction

1.1 Motivation

The motivation of this study is to investigate the applicability of fuzzy logic control for the High-Frame Rate Stereovision Object Tracking system. Fuzzy logic control is known as a robust, fast, and reliable technique. It is mostly applicable to systems with nonlinearity, uncertainty, and where a mathematical model is difficult or impossible to develop. In these situations, fuzzy logic provides an accurate, human-like control that surpasses conventional Proportional-Integral-Derivative (PID) techniques. The main goal of the project is to develop an experimental platform – a reconfigurable fuzzy logic control system for the Stereovision Tracking System (STS). This system can serve as a foundation for applying fuzzy logic to stereovision object tracking in the future.

1.2 Objective

The main objective of this study is to develop, implement, and test a reconfigurable fuzzy logic control system for stereovision object tracking. The list of objectives for this thesis is as follows:

1) To develop a concept of rotation and focus control for STS.
2) To develop a parameterized fuzzy logic controller model for rotation and focus control adapted for STS system requirements.
3) To develop reconfigurable, real-time, embedded fuzzy logic controller architecture suitable for FPGA hardware implementation.

4) To develop a fuzzy logic control system using VHDL, as an independent module that can be included in STS system project.
5) To develop a set of tools to facilitate the reconfiguration process of a fuzzy logic control system.
6) To implement a fuzzy logic control system for STS's FPGA platform.
7) To conduct a series of static and dynamic experiments to validate fuzzy logic control system performance in STS's FPGA platform.

1.3 Original Contributions

The contributions made to the High-Frame Rate Stereovision Object Tracking project in the scope of this thesis are as follows:

- The development of the concept of rotation and focus control for STS.
- The development of the concept of selecting the closest object from the disparity map and providing inputs for the fuzzy logic controller.
- The conduction of a literature review on fuzzy logic control and the design of fuzzy logic systems, outlining the relevant topics in fuzzy logic control theory.
- The development of a parameterized fuzzy logic controller model for rotation control, based on the theory of fuzzy logic and its applications.
- The design and implementation of reconfigurable fuzzy logic control system hardware architecture, which contains two parts: a static servicing hardware part, and a reconfigurable fuzzy logic Look-Up Tale (LUT) part.
- The design and implementation of a set of tools to facilitate reconfiguration of a fuzzy logic LUT part. The programs in the set of tools include: (i) LUT generator (Matlab Program) that creates a LUT using a set of fuzzy parameters, and (ii) LUT builder (automatic built environment) that creates a LUT file in VHDL format.
- The participation in the design of the Disparity Processor module – the object identification module of STS system that provides inputs for the fuzzy logic controller.
- The integration of the fuzzy logic control system with Stereovision Object Tracking.
- The conduction a series of static and dynamic experiments with STS and the fuzzy logic control system.

1.4 Chapters' Description

Chapter 1 summarizes the motivation, objectives, contributions, results, and conclusions of this work.

Chapter 2 describes the architecture of the Reconfigurable Fuzzy Logic System for High-Frame Rate Stereovision Object Tracking. The topics covered include system specifications, principles of operation of a proposed system, the comparison between fuzzy logic and conventional Proportional-Integral-Derivative (PID) control as well as the reasons for selecting fuzzy logic control for STS, the reconfiguration requirements of the system, the main architectural components of the system, and the description of the automatic built environment for facilitating reconfigurability of the proposed system.

Chapter 3 describes the theory of fuzzy logic and its applications. The first topic covered is the theory of fuzzy logic systems. This encompasses fuzzy sets, membership functions, a linguistic approach to fuzzy reasoning, and fuzzy sets operations. Another topic covered is the design of fuzzy systems. This includes fuzzification, fuzzy reasoning and rules, and defuzzification. The last topic of this chapter is an overview of fuzzy logic systems.

Chapter 4 describes the implementation of the Reconfigurable Fuzzy Logic System for High-Frame Rate Stereovision Object Tracking. The topics covered include a description of hardware components and development tools, fuzzy logic controller implementation, automatic built environment and reconfiguration tools, as well as FPGA realization of fuzzy logic control system.

Chapter 5 describes the results of experiments conducted with the fuzzy logic control system. The topics covered include the results of simulation with ModelSim software, the description of the experimental set-up and the design of static and dynamic experiments, as well as a summary of results and conclusions.

Chapter 6 summarizes the conclusions derived from the work conducted in this thesis.

Chapter 2

Architecture

2.1 Introduction

In this chapter, the architecture of Reconfigurable Fuzzy Logic System for High-Frame Rate Stereovision Object Tracking System or Stereovision Tracking System (STS) is presented. The main objective of the proposed system is to enhance the STS with the capability to follow an object of interest by the rotation of the system head as well as to control the focus of the stereo-cameras of STS by rotation of the adjustable focus lenses (see Figure 2.1).

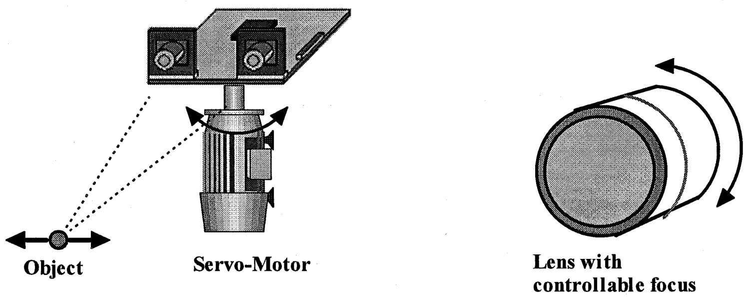


Figure 2.1 : Rotation of SOT system and lens with adjustable focus.

In the following sub-chapters, the system specifications will be outlined, the challenges of creating the system satisfying these specifications will be discussed. The principle of operation and proposed architecture for practical realization of the system will be given and the benefits of using fuzzy logic controller to cope with many challenges of the task will be outlined.

2.2 System specifications

The following system specifications or a list of system requirements can be outlined:

- The system must be able to automatically identify the object of interest. For proof of concept, the closest object in camera view can be chosen as the object of interest.
- The system must be able to position itself so that the object of interest will appear in the center of stereovision view. For proof of concept, the positioning can be done only by STS rotation along the horizontal axis. Generally, the horizontal and vertical rotation gives opportunity to track an object in tree-dimensional space.
- The system must be able to control the focus of two stereovision cameras by rotation of the lens adjustable focus to keep the object of interest within the focal depth. For proof of concept, a long focus depth lenses can be used, so little adjustments will be necessary.
- The system must be able to track the moving object with horizontal and vertical rotation of the stereovision head. For proof of concept, the tracking can be done only by rotation along horizontal axis.
- The system must be able to switch to new object if a) previous object of interest is out of the tracking angle[1] b) previous object is no longer the closest one (this is given the assumption that object selection is performed by the criteria of the closest object).
- The system must maintain tracking regardless of physical characteristics of the object i.e. size, shape, form, or color. The rotation of the object, moving, and changing shape and form shall not affect the tracking characteristics.
- The system must have a threshold distance for detection of the objects. The object shall be detected only if it is within the detection distance, thus separating object and the background. If no object is detected within the detection distance, the system should stay still. Similarly, if size of the object is too small, comparable to the noise level, the object must be ignored.

---

[1] Tracking angle is the maximum rotation angle of the STS that is supported by the binocular head platform. The maximum angel of STS platform is +/-90°

2.3 Principles of operation

In order to achieve desired system specifications, the a few basic steps have to be taken:

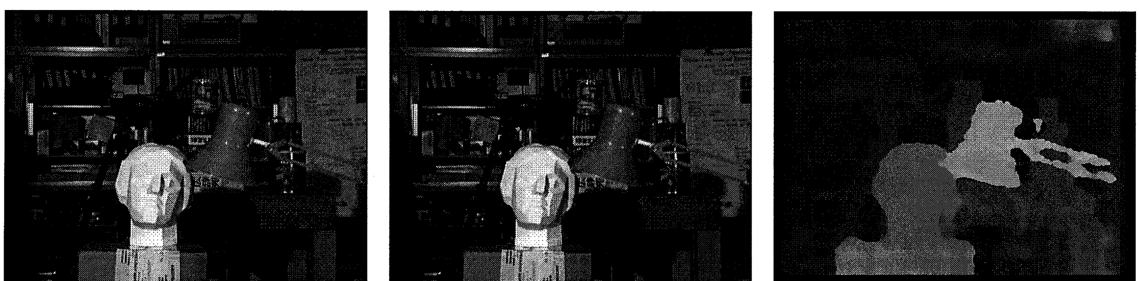
a) Mount STS system (a binocular head) on a flexible platform with servo or stepper motor attached to control the rotation (See Figure 2.1). The system must be able to rotate +/-90°.

b) Use the stereovision cameras with adjustable focus lenses.

For achieving desired tracking, the system must be able to identify the following three items without human intervention:

- The object of interest
- The position of the object relative to the center of the image
- The approximate distance to the object

Knowing all three items, it is possible to control the rotation of the head of the system so that the object will appear in the centre while maintaining proper focus. All this information can be excluded from the disparity map provided by Stereovision Tracking System (STS). A disparity map is a depth map where the depth information is derived from offset images of the same scene. In the STS, the disparity map is derived from offset images of two stereo-vision cameras. The objects, in the disparity map, are identified by the disparity value or disparity magnitude, so that close objects has higher disparity magnitudes than the distant ones. They also appear brighter on the disparity map image. The contours of object in the disparity image can serve the purpose of identifying the object, while the disparity value can identify the approximate distance from the object to the camera.



a b c

Figure 2.2 : a – left camera, b – right camera, c – disparity map image.

Figure 2.2 shows the disparity map combined from left and right vision cameras. Looking at the disparity map, the lamp can be identified as a closest object since having maximal disparity magnitude in the scene. The distance from the lamp to the camera can be derived relative to the distance to objects and requires calibration. The approximation of the distance can be applied to adjust the focus of lenses to maintain the object within focal depth.

The identification of the object can be done by analyzing the disparity map for maximum disparity values representing the closest objects, and deriving the object boundaries. As soon as the object boundaries are identified, it is possible to calculate the center of the object applying one of Centroid algorithms. Since, in this work, only a horizontal rotation of the binocular head is implemented, a different algorithm for locating the center of the object is presented (see §2.6.2).

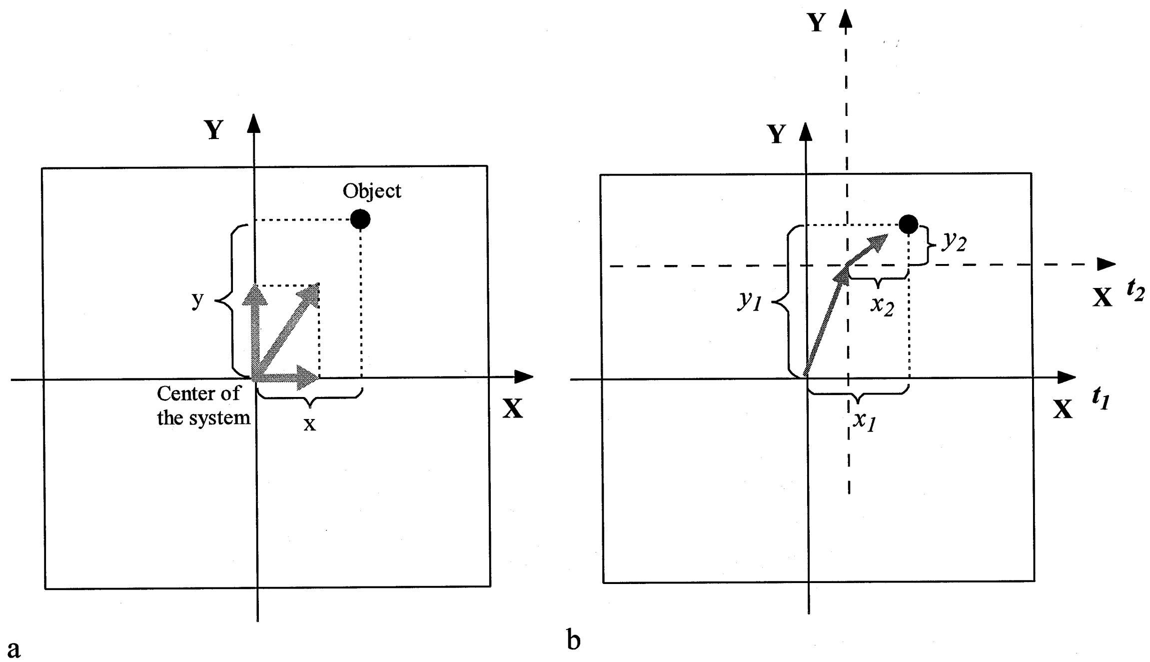


Figure 2.3 : a – displacement coordinates, b – two iterations if feedback control.

As soon as the object is identified and the center of the object is calculated, it is possible to change the angle of the camera to bring the center of the object in the middle of the view. Figure 2.3 (a) demonstrates how the displacement of the object from the center of the view can be identified by *x* and *y* coordinates reflecting the displacements relative to X and Y axes. Knowing the displacement of the object from the center of the view it is possible to approximate the required angle to move the binocular head in order to bring the object in the center, however many factors make the calculation complicated and imprecise. Nonlinearity in characteristics of the lens, nonlinearity in motor characteristics introducing the angle to the system and many other real-life factors make the precise mathematical model for calculation of the exact angle very complex and very component dependant. Such a model would require extensive calibration procedures to achieve desired result. It is clear that a feedback control system with multiple iterations can achieve better results. Two iterations of the feedback system are demonstrated in Figure 2.3 (b), where after the first iteration the center of the system is brought closer to the object and $x_2 < x_1$, $y_2 < y_1$. A number of iterations might be required to bring the object of interest in the center, depending on system dynamics and how well the required angle shifts are calculated. The fuzzy logic approach brings the ability to control the system without knowing a precise mathematical model with all nonlinearities and deviations. Moreover, it allows controlling the system with different dynamics and uncertainties by applying human like control and experience that can accommodate large variances in system parameters. The knowledge of desired control of the STS is transferred to membership functions, fuzzy rules, and fuzzy parameters.

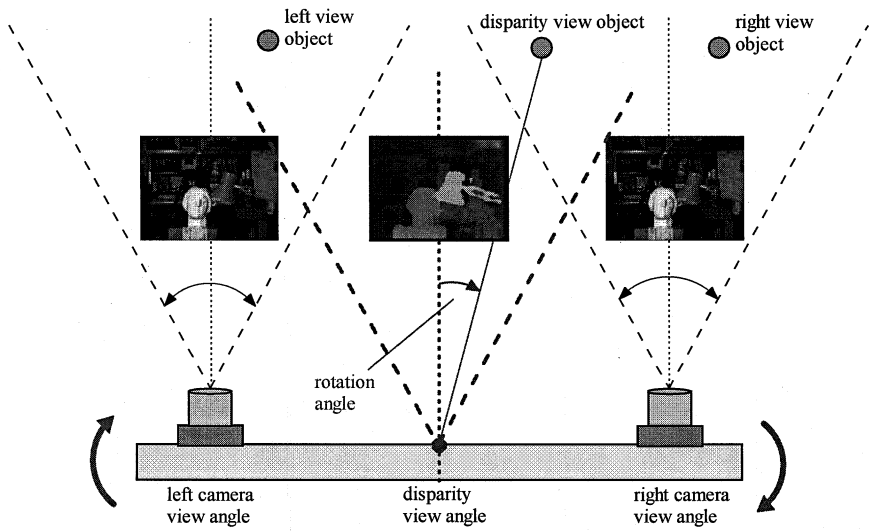


Figure 2.4 : The rotation center of the system (the disparity angle).

The rotation center of the system is illustrated in Figure 2.4. The angle to the object in the left view camera is different from the angle in the right view camera. The disparity map correlates those differences into one angle of disparity view object. The angle to the object is referenced to the virtual center of the system – the disparity center. If left and right view cameras are set in exact 90° angle from the holding platform, the disparity center will be in the middle between the two cameras. This is going to be the rotation angle of the system.

The ability to track moving objects largely depends on the responsiveness of the system. The overall responsiveness comprises of the sum of all response times of the system components. The responsiveness of the system will define the relative speed of the object that can be tracked depending on the distance from the binocular head. The maximum angular speed of the head and the system responsiveness will define limitations for tracking of moving objects.

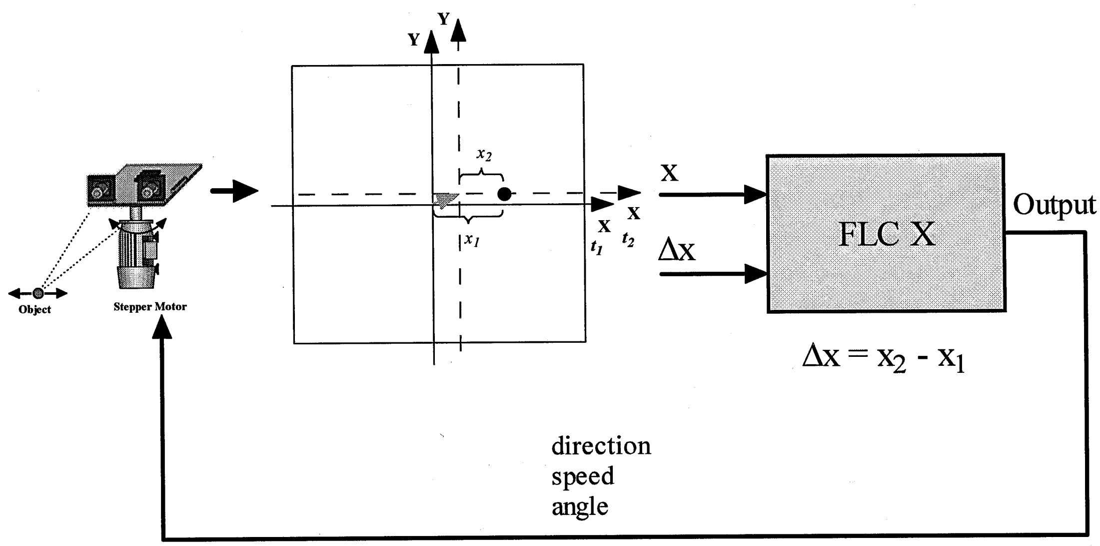
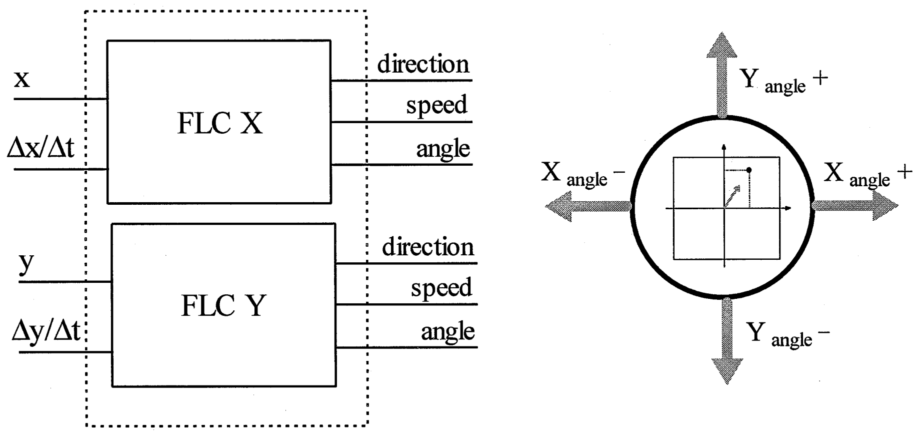


Figure 2.5 : Rotation control concept

In order to accommodate system dynamics and provide smooth control while tracking the object, the two input parameters to Fuzzy Logic Controller (FLC) are required: displacement and the derivative of the displacement (speed). If at the moment t_1, the coordinates of the object were (x_1, y_1), and after the feedback loop iteration at moment t_2, the coordinates of the object are (x_2, y_2), the derivatives of the displacement are $\Delta x/\Delta t = (x_2 - x_2)/(t_2 - t_1)$ and $\Delta y/\Delta t = (y_2 - y_1)/(t_2 - t_1)$.

A use of speed as an additional input parameter to FLC provides new dimension to the control, overseeing how fast the system is aiming on the object. Having displacement and speed as input parameters; the FLC can produce all necessary outputs for the direction control: the direction of rotation, the speed of rotation, and the angle (see Figure 2.5).

Since X and Y corrections are virtually independent processes, they can be controlled by two separate fuzzy logic controller. This architecture will provide a simultaneous tracking in two-dimensional space. The concept of this architecture is presented in Figure 2.6.



$$\Delta x/\Delta t = (x_2 - x_1) / (t_2 - t_1)$$
$$\Delta y/\Delta t = (y_2 - y_1) / (t_2 - t_1)$$

Figure 2.6 : Two parallel FLC controllers (X and Y).

The two groups of challenges and limitations for the proposed concept can be outlined.

Group 1 (challenges):

- Dependency on clarity of disparity map image
- Distortion and nonlinearity in characteristics of the lens
- Nonlinear system dynamics
- Vibration of the system during the rotation and as distorted image
- Different shapes and sizes of the objects
- Unpredictability in object movements (acceleration, rotation, vibration etc.)

- Different uncertainties due to viewing real-life events

Group 2 (limitations):

- Limitation in speed for tracking of fast moving objects
- Clarity of image necessary for object identification
- Minimum angle achievable by rotation control
- Maximum tracking angle of the system

The challenges and limitations of proposed concept mostly depend on many real-life factors and characteristics of system components. The usage of fuzzy logic can significantly improve system performance with respect to overcoming challenges of group 1, especially nonlinearity and uncertainties. The group 2 represents fundamental limitations of the system depending on system components.

2.4 Fuzzy logic versus PID control

From observation of fuzzy logic theory and applications (see §3), the major strengths of fuzzy logic control over conventional PID can be outlined:

- Ability to deal with nonlinear systems. The nonlinearity of the system can be reflected in fuzzy logic membership functions and the desired control can be achieved by fuzzy rules matrix and fuzzy parameters (see §3.1.1, §3.2.2).
- Ability to control systems with hardly achievable or unachievable mathematical models. Just by gaining human control experience on the system, the fuzzy membership functions can be designed to achieve the desired control without knowing mathematical model of the system. No precise numerical interpretations are needed.
- All possible decision making can be intuitively implemented in fuzzy logic system providing human-like reasoning and control (see §3.1.2 - §3.2.2).
- Large number of fuzzy parameters that can be modified to exert the desired degree of control (see §3.2.3). In typical PID control, the only 3 parameters can be modified. In fuzzy logic multiple parameters can be modified at each stage of the fuzzy logic approach: fuzzification, fuzzy reasoning, and defuzzification. The optimal combination of fuzzy logic parameters can provide any required degree of control. In tracking, fuzzy logic can provide faster aiming and quicker response to disturbances with minimal overshoot and undershoot.

- Proper set-up of fuzzy sets and membership functions guarantees system conversion to steady state (less chance of oscillation if wrong parameters are chosen). With human knowledge of system behavior, the set-up of fuzzy membership functions is fast and intuitive.

In order to achieve listed above qualities natural for fuzzy logic, the traditional PID requires a dynamic adaptation of its constants, which makes it cumbersome and ineffective in many cases. For instance of temperature control, if system requires faster warm-up and slower cool-down phase, and also a quicker response to disturbances with minimal overshoot and undershoot when set-point changes, the PID control cannot meet these extra challenges, however fuzzy logic is capable of accommodating all of them. For object tracking, the simplest example would be a requirement to track an objects going left in fast and aggressive way, and object going right in smooth and overshoot free way, the membership functions can by modified to provide this degree of control. The PID would require dynamic adaptation of its constants depending on the direction of moving object. The membership functions for rotation and focus control can accommodate all nonlinearity of all STS system components in one parameterized model of fuzzy logic controller.

The large number of fuzzy parameters (each can influence the overall control behavior of the system) make fuzzy logic highly applicable to reconfigurable systems. Changing these parameters statically or dynamically can significantly improve tracking characteristics providing additional degree of control and adaptation.

For all reasons listed above, the fuzzy logic control was selected as a control mechanism for rotation and focus control of high-frame rate stereovision object tracking.

2.5 System reconfiguration requirements

Reconfigurability can be divided into static and dynamic. The static reconfigurability denotes changing system configuration offline, while dynamic reconfigurability denotes changing system configuration at runtime.

The static reconfigurability of fuzzy logic controller implies changing the fuzzy logic parameters to fit best the specifications of the system (motor characteristics, nonlinearity of system components, etc.). The following fuzzy logic parameters can be changed in order to achieve best control: the number of fuzzy membership functions, the shapes of membership functions, the fuzzy operator, inferencing operators, and the defuzzification method (see §3). The fuzzy logic parameters also influence the way the application is controlled: fast and aggressive tracking versus smooth and slow. The preferable way of fuzzy control can be experimentally found adjusting fuzzy logic parameters. Static reconfigurability gives opportunity to find set of fuzzy parameters achieving desired tracking performance.

Dynamic reconfiguration of fuzzy logic controller implies changing fuzzy logic parameters on the fly, when system is working – between the frames of the camera. The fuzzy parameters can be changed depending on image qualities (brightness, contrast etc.), the size of the object, the speed and direction with which the object is moving, the distance from the object. The dynamic adaptation can significantly improve tracking, since for each particular instance the best-fitting fuzzy logic parameters can be used.

All of above requires a very high degree of reconfigurability from the fuzzy logic control system architecture. In order to achieve required degree of reconfigurability, the Look-Up Table (LUT) architecture of Fuzzy Logic Controller (FLC) is proposed in this work. This architecture virtually achieves maximum degree of reconfigurability, since all fuzzy input/output responses are pre-calculated and organized in the form of look-up table. Each LUT can contain any combination of fuzzy logic parameters and exert any control over the application. Reconfiguration can be achieved with loading a new LUT to the system. The proposed architecture is reconfigurable, easy for implementation and also is the fastest real-time architecture, since virtually no real-time calculations are required.

2.6 Main architectural components

The top level system block diagram is presented in Figure 2.7. This diagram represents the proposed architecture and divides system into top level modules.

In this diagram, the STS module provides the disparity map for the Disparity Processor Module. The Disparity Processor Module is responsible for the identification of the object of interest and delivering the displacement coordinates to the Fuzzy Logic Module. The FLC module performs calculation of the derivative of the displacement, and holds reconfigurable look-up tables with fuzzy logic values. The outputs of the FLC module are connected to the Servo-Motor Controller and are the direction of the rotation, rotation angle and the speed. The Servo-Motor Controller is responsible for controlling the motor to perform requested by fuzzy logic operation. All top-level system modules will be described in details in the following chapters.

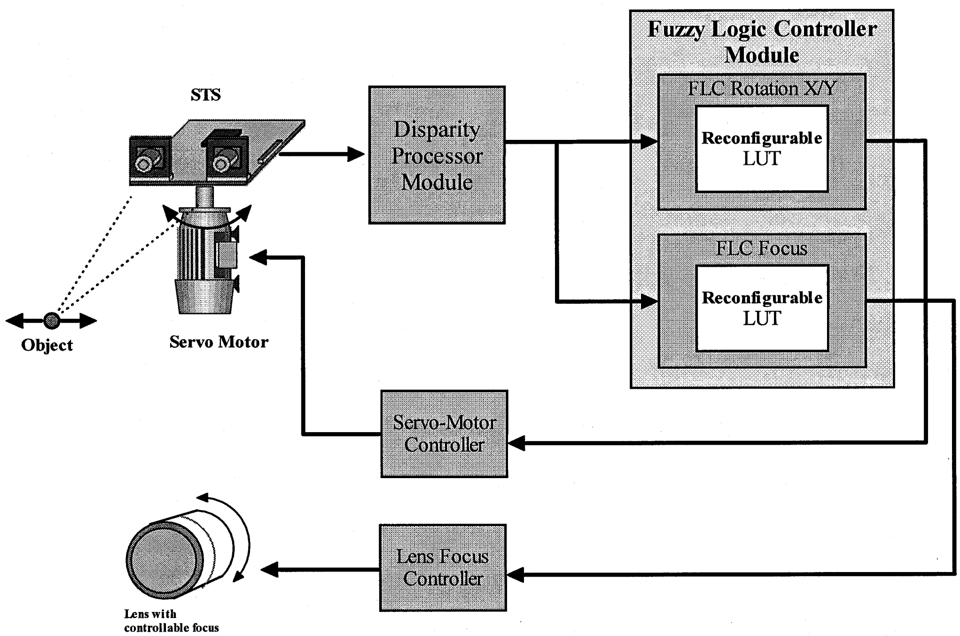
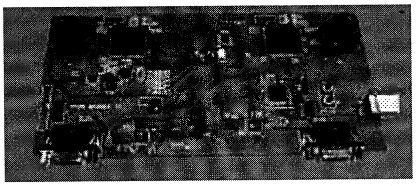


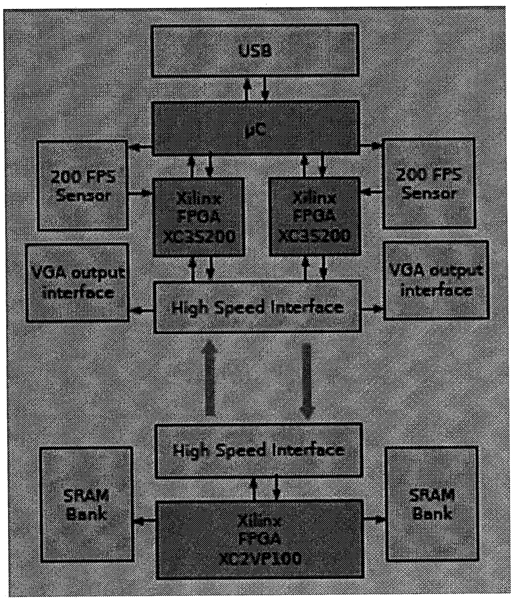
Figure 2.7 : Top level system block diagram.

2.6.1 STS Module

The High-Frame Rate Stereovision Object Tracking or Stereovision Tracking System (STS) has been developed at Ryerson University Embedded Reconfigurable System Lab as a project associated with investigation and development of space-borne computer stereovision system for automated satellite grasping and automated satellite docking for recent space robotic systems developed at MDA Space Missions [1] [2]. The project is associated with the development of application specific (200 frames per second) stereo video sensors with on-board preprocessing FPGA-based reconfigurable multi-stream platform and implementation of stereovision tracking algorithms on the FPGA based computing platform. See Figure 2.8.



a



b

Figure 2.8 : STS module (a) photograph of front view, (b) block diagram.

The STS architecture is based on a Dynamic Programming Maximum Likelihood (DPML) approach developed by Cox [3]. The original maximum likelihood stereo algorithm for producing a disparity image map for high-frame rate stereovision systems was enhanced and optimized for hardware pipelined architecture in [4]. This algorithm was implemented on the STS FPGA platform.

2.6.2 Disparity Processor Module

The disparity processor module was developed at Ryerson University Embedded Reconfigurable System Lab as a part of the High-Frame Rate Stereovision Object Tracking project. The module was developed on VHDL and implemented on the STS FPGA platform. The main objective of the disparity processor is to identify the object of interest and calculate the displacement from the center of the object to the center of the system. For that reason object identification and the Centroid algorithm has to be used. For proof of concept of fuzzy logic applicability, the following simplifications to the functionality of disparity processor algorithm have been made:

a) The object identification is made on the criteria of the object close than a certain distance. The threshold disparity value is representing a distance threshold.
b) The object detection is performed only on the middle horizontal line of the image.
c) The Calculation of the center of the object and the displacement is done only on X axis.

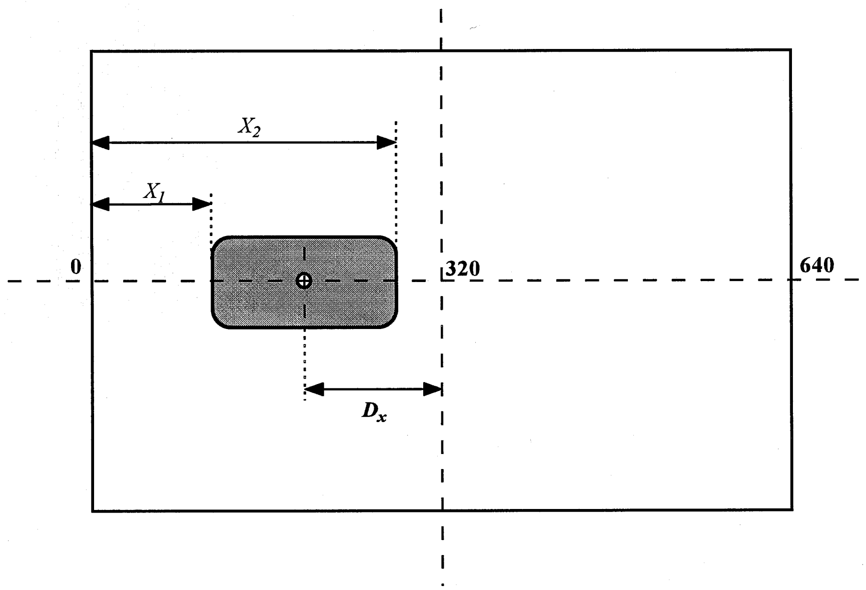


Figure 2.9 : Object displacement identification.

Figure 2.9 illustrates the proposed algorithm for calculation of the displacement. x_1 and x_2 are the edges of the object detected by the threshold principle, D_X is the actual displacement of the center of the object from the center of the camera. For 680 by 480 disparity pixel image, the D_X is given by (2.1).

$$D_X = \frac{x_1 - (640 - x_2)}{2} \tag{2.1}$$

The computer notation of (2.1) for software/hardware implementation if given by (2.2)

$$D_X = (x_1 - (640 - x_2)) >> 1 \tag{2.2}$$

where

symbol (>> 1) represents 1 bit shift to the right
x_1, x_2 – 10 bits unsigned (to cover all range from 0 to 640)
D_X – 10 bits signed

The center point of the object could be anywhere from -320 to +320.

2.6.3 Fuzzy Logic Controller Module

Fuzzy logic controller module is the main module presented in this work. This module incorporates fuzzy reasoning to produce direction, angle, and speed for each input displacement value. This module also calculates derivative of the displacement and uses the displacement and the derivative as input parameters for fuzzy look-up table (See Figure 2.10). The module was developed in VHDL with reconfigurable fuzzy look-up table located in Block RAM and practically implemented on the STS FPGA platform. The number of Block RAM units used by the system depends of the size of fuzzy look-up table, which depends on resolution and number of inputs and number of outputs. The details of look-up table implementation are described in §4.2.7.

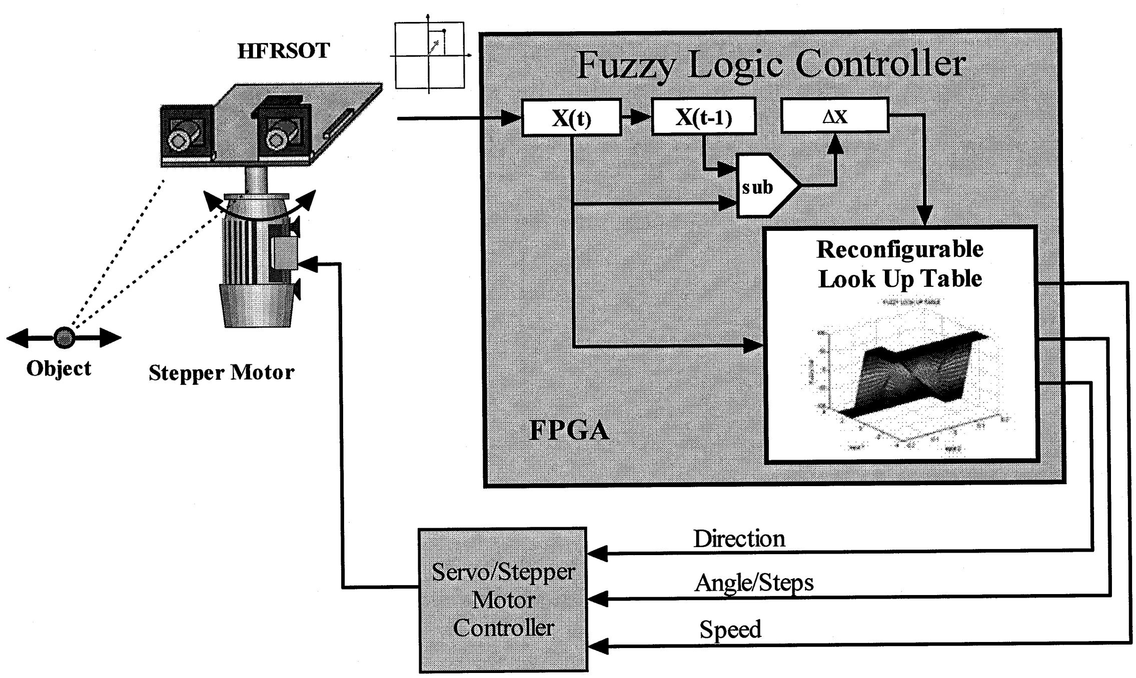


Figure 2.10 : Fuzzy Logic Controller Architecture.

To achieve high degree of reconfigurability, the two parts of fuzzy logic controller module can be separated: the reconfigurable look-up table (located in BRAM on the FPGA device), and the servicing functionality such as derivative calculation and address calculation of the look-up table. With this architecture, it is possible to reload look-up table with new fuzzy logic control values while maintaining identical servicing functionality. In this case the only BRAM initialization context must be changed leaving the rest of VHDL context of the module untouched. This architecture makes also the dynamic reconfiguration possible. If look-up table is placed in the external memory or FLASH, and the servicing functionality in the FPGA, the reconfiguration can be achieved just by reloading memory or FLASH.

The output of the fuzzy logic controller module is equipped with SPI interface to the Servo-Motor Controller providing it with direction, angle, and the speed. The implementation of the fuzzy logic controller module is described in details in §4.6.

2.6.4 Servo-Motor Controller

The Servo-Motor Controller (SMC) module was developed at Ryerson University Embedded Reconfigurable System Lab as a part of the High-Frame Rate Stereovision Object Tracking project. The module is responsible for controlling the servo-motor that enables rotation of the STS system head. The detailed interface the SMC module is described in §4.6.6.

2.7 Automatic Look-Up Table Build Environment

An automatic look-up table build environment was developed to facilitate and automate a creation of fuzzy logic controller VHDL project with built in fuzzy look-up table. A batch build file includes a sequence of commands initiating the process illustrated in Figure 2.11. The process starts with evoking LUT Generator in MATLAB environment to create fuzzy look-up tables and convert them into VHDL project file with BRAM initialization. Then, one of two options is available: simulation with ModelSim and test bench (VHDL) program, or synthesis of the project with Xilinx ISE environment. The build option is selectable by the parameter of the batch file. The implementation details of automated build environment and FLC project are given in §4.4.

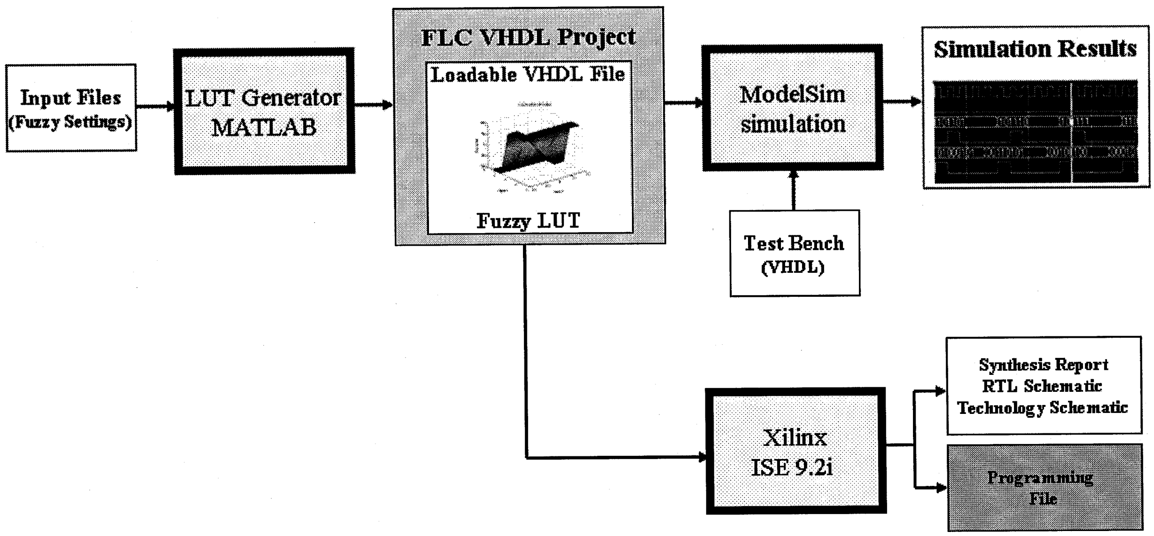


Figure 2.11 : Automatic Build Environment Block Diagram.

The LUT Generator is a Matlab program created to provide a simulation of fuzzy algorithm for the entire input range of values, generating fuzzy look-up table (see Figure 2.12). LUT Generator is presented here as a file that can be called in Matlab environment command line. LUT Generator is optimized for two input fuzzy logic controller with three membership functions for each. These functions, along with output-defuzzification membership function can be of any desired shape, since uploaded as a text files with pre-set numeric values. The implementation of fuzzy sets is organized in a sub-module that can generate a triangular membership functions. However, any arbitrary software can be used to create desired form of the membership functions. The output of LUT Generator is a text file with values of two-dimensional fuzzy look-up table organized in a form of matrix. Thus, for FLC with two inputs and one output, the LUT will represent all possible output values on all combinations of inputs. The implementation details of LUT Generator are given in §4.3.

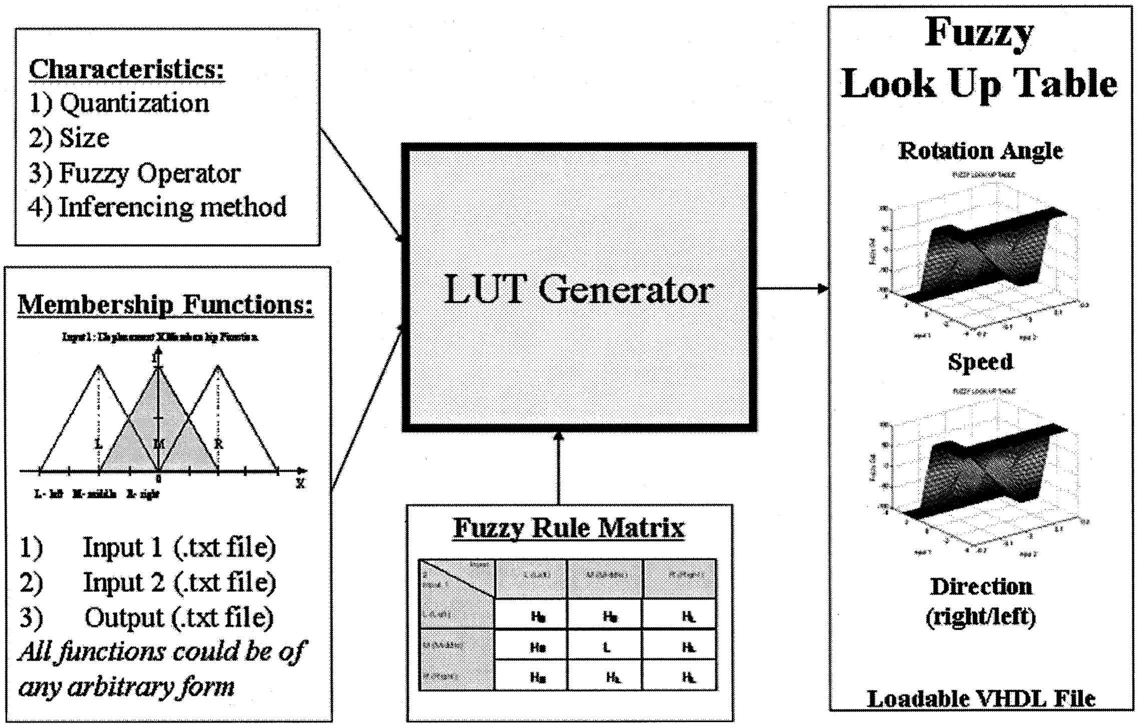


Figure 2.12 : Look-up Table Generator Block Diagram.

Practically LUT can be seen as a *k*-dimensional matrix, where *k* is equal to the number of inputs, organized in such a way that every element of the LUT is positioned corresponding to the index of the element in input vector. For instance of k=2, if $fuzzy_1$ and $fuzzy_2$ - are inputs to a fuzzy controller, such as $fuzzy_1 \in X$, $fuzzy_2 \in Y$, where X and Y are linguistic variables that could be transformed into a discrete universe with quantization level of N:

$$X \overset{N}{\Rightarrow} X' \in [x_1, x_2, ..., x_N] \qquad Y \overset{N}{\Rightarrow} Y' \in [y_1, y_2, ..., y_N] \tag{2.3}$$

The LUT matrix will have the following form:

$$\begin{bmatrix} fuzzy\{x_1, y_1\} & fuzzy\{x_2, y_1\} & \dots & fuzzy\{x_N, y_1\} \\ fuzzy\{x_1, y_2\} & fuzzy\{x_2, y_2\} & \dots & fuzzy\{x_N, y_2\} \\ \dots & \dots & \dots & \dots \\ fuzzy\{x_1, y_N\} & fuzzy\{x_2, y_N\} & \dots & fuzzy\{x_N, y_N\} \end{bmatrix} \tag{2.4}$$

where $fuzzy\{x, y\}$ is an entire fuzzy controller function, with crisp inputs x and y, producing crisp output.

2.8 Summary

In this chapter, the main concept of Reconfigurable Fuzzy Logic System for High-Frame Rate Stereovision Object Tracking was presented. The objective of the system and set of system requirements was listed. The main principle of X and Y control over the rotation of the system head to track the object of interest was presented. The architecture satisfying system requirements was proposed and major components of the system were described. Along with that, the benefits of fuzzy logic applied to cope with many challenges of the task were outlined. This chapter is finalized with the description of reconfigurable architecture of fuzzy logic controller and an automatic build environment that was developed for fuzzy logic controller reconfiguration and realization. This chapter covered only the architectural and functional description of the proposed solution, the implementation details will be described in §4.

Chapter 3

Fuzzy Logic Theory

When Aristotle and his predecessors devised their theories of logic and mathematics, they came up with the so-called Law of the Excluded Middle [5]. This law states that every proposition must either be true or false. We can compare it to a today's binary logic with only two logic states "1" – true and ""0" – false, which was created in the 19th century by George Boole as a system of algebra and set theory that could deal mathematically with such two-valued logic [6]. It is true that all contemporary digital world technology is based on these binary concepts, but what would we say of the following statement "the apple is either red or not red"; it clearly cannot be both red and not red. But it could be in the middle. Having that in mind, not everyone agreed with Aristotle, and even Plato indicated there was a third region, beyond true and false, where these opposites "tumbled about" [7]. In the early 20th century, Jan Lukasiewicz proposed a three-valued logic "true", "possible" or "false" [8], which was a first step towards development of mathematical theory that deviated from strict conventional logic. This theory never gained wide acceptance. Many great minds were addressing and discussing the possibility of creating an alternative logic, until in 1965, the initial concept of Fuzzy Logic (FL) was first presented by Lotfi Zadeh, a professor at University of California at Berkley [9].

Zadeh had observed that conventional computer logic couldn't manipulate data that represented subjective or vague ideas, and his attempt was to create an alternative logic to allow computers to determine the distinctions among data with shades of gray, similar to the process of human reasoning. He presented his ideas not as a control methodology but as a mathematical fuzzy set theory, the way of processing data by allowing partial set membership rather then crisp set membership or non-membership [10]. Although, the new concepts were introduced in the America, American and European scientist and researchers largely ignored it for years. They refused to take seriously something that dealt with imprecise numerical computation and subjective logic. Even unconventional name for the theory led to suspicion and distrust. Some mathematicians argued that fuzzy logic was merely probability in disguise [11], and doubted that it could ever gain wide acceptance and find its way in industrial applications.

Fuzzy logic as an approach to set theory was not applied to the field of control systems until late 70's due to insufficient small computer capability prior to that time. But later with increasingly fast growth of the computer technology, fuzzy logic has found its way in many industrial applications and even led to unreachable for conventional control results. In 80^{th}, fuzzy logic was readily accepted in Japan, China and other Asian countries, later it became very popular in Europe [12].

One of the basic reasoning behind the Professor Zadeh's fuzzy logic concept is that people do not require precise numerical data to instinctively produce highly adaptive and effective control [10]. If we could reveal the mystery of human adaptive and constantly improving with experience control, we would be able to produce systems that mimic this behavior. If, for instance, feedback controllers could be programmed the same way as human brain works, they would be able to accept noisy, imprecise input data, and still produce desired and adaptive response. The fuzzy logic is the tool allowing to mimics human intelligent control [13]. Fuzzy systems are capable of processing vague, imprecise and noisy data, and still produce effective and smooth control. They are able to effectively control environments with very complex mathematical models or even environments in which mathematical models are not achievable due to complexity and non-linearity. In all cases, fuzzy logic brings intuitive, human-like control and decision making, which can be easily adjusted through fuzzy parameters, operators and algorithms to fit desired behavior [14]. All of the above gives fuzzy logic big advantage over conventional strict logic, and summarizes the reasons why fuzzy logic became commonly used in various industrial applications and areas of modern life [15], [16].

3.1 Theory of Fuzzy Logic Systems

Fuzzy logic is a mathematic theory of fuzzy sets, which provides basic mechanism for fuzzy set-based approximation-reasoning. In this approach, each granule of knowledge is represented by a Fuzzy Set (FS). Fuzzification is a transformation of traditional crisp domain granules of knowledge to fuzzy domain (FS). Once in fuzzy domain, the approximation-reasoning operations could be performed on the FS to produce a fuzzy response. This process is called Fuzzy Reasoning (FR) or Fuzzy Inferencing (FI). It includes operations on FS, Fuzzy Relations, Fuzzy Rules, Linguistic Fuzzy Approaches, and Fuzzy Inferencing Methods. The result of Fuzzy Inferencing is a fuzzy set that represents the output of the fuzzy system in fuzzy domain. The conversion of fuzzy response from fuzzy domain to crisp domain is referred to as Defuzzification [13].

The conventional fuzzy logic system accommodates all three processes of the fuzzy logic: Fuzzification, Fuzzy Reasoning and Defuzzification. See Figure 3.1. The inputs of the system are transformed to fuzzy domain by Fuzzification, where Fuzzy Reasoning is applied. The last stage of the fuzzy system is the Defuzzification that generates crisp system response.

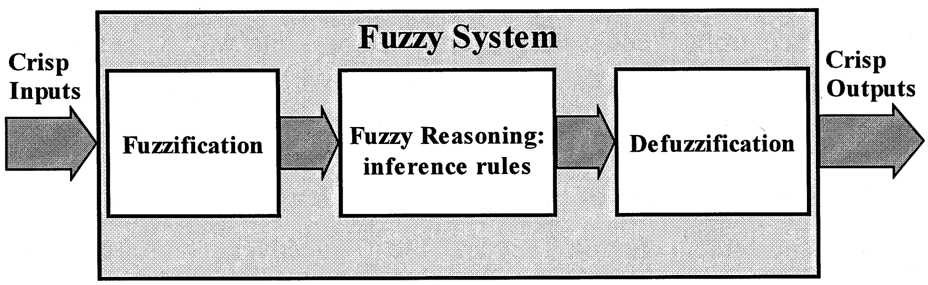


Figure 3.1 : Fuzzy system simplified block diagram

3.1.1 Fuzzy Sets and Membership Functions

Fuzzy set, by definition, is a collection of objects with membership values between 0 (complete exclusion) and 1 (complete membership). Each set is selected to have a unique feature or property that distinct this set from other sets. The membership values of each object express the degrees to which that object is compatible with the properties of the set [10] [17].

Definition 3.1 A fuzzy set is characterized by a membership function mapping the elements of a domain, space, or universe of discourse X to the unit interval [0,1]. (Zadeh 1995)

The notation of a fuzzy set can be expressed in a different way in the following format: A:X →[0,1]. The fuzzy set A in the universe X may be represented as a set of generic elements x∈X and its grade of membership A = {(A(x)/x) | x∈X}. The set of values A(x) describe a degree of membership of the generic elements x in the fuzzy set A and is referred to as Membership Function (MF). The MF is a representation of the magnitude of participation of each element x in set A. The membership values range from 0 to 1. Given a universe of elements U, a fuzzy membership function f_A describes a set A, and represents a degree of membership for each object x [18].

$$f_A : U \to \{0,1\}$$

$$\begin{cases} f_A(x) = 1, & if \quad x \in A \\ f_A(x) = 0, & if \quad x \notin A \end{cases} \tag{3.1}$$

Since fuzzy domain can be discrete or continuous, the common mathematical notation for a fuzzy set A with a continuous domain x and membership function f_A is represented by (3.2), for the discrete domain x €$\{x_1, x_2, ..., x_N\}$ with membership function f_A, the fuzzy set A is represented by (3.3):

$$f_A(x)/x \tag{3.2}$$

$$f_A(x_1)/x_1 + f_A(x_2)/x_2 + f_A(x_3)/x_3 + ... + f_A(x_N)/x_N \tag{3.3}$$

For example a typical triangular fuzzy set with continuous domain can be mathematically described by (3.4)

$$\begin{cases} f_A(x) = k_L x + h_L, & if \quad x \le a \\ f_A(x) = -k_R x + h_R, & if \quad x > a \end{cases} \tag{3.4}$$

where k_L and h_L are linear coefficients of the left side of the triangle, k_R and h_R are linear coefficients of the right side of the triangle and a is the center of the triangle, or the highest degree of membership in this set . If triangle is isosceles, the more compact notation (3.5) is used [19].

$$f_A(x) = \max(0.1 - |a - x| / d) \tag{3.5}$$

Fuzzy sets can be also regarded as constraints imposed on the element of a universe or a domain. The elements could be a discrete or continuous. It could be days of week, sides of the dice, or an analog continuous domain such as time, temperature, pressure, etc. The constraints are those that define affiliation of fuzzy elements with the particular fuzzy set – membership values. The sets could be a temperature range (see Figure 3.2) classified as high, medium or low; or time domain classified as early, on-time, late; or pressure domain classified as low, normal, high, or the classifications can be regarded as unacceptable low, acceptable, unacceptable high, etc. In fact, fuzzy sets could have any type of constraints arbitrary assigned to them. The creator of the constraints subjectively imposes the terms of constrains based on known or desired system behavior and a set of qualities pertained to the system. For instance, for controlling the room temperature, the fuzzy sets can be as follows: low- describing range of low temperatures (the heating is needed), medium – describing normal range of temperatures (no action is needed), high – describing high range of temperatures (cooling is needed. Membership functions for the temperature control will represent degree of membership for each temperature range of temperatures from 0 to 1 (see Figure 3.2).

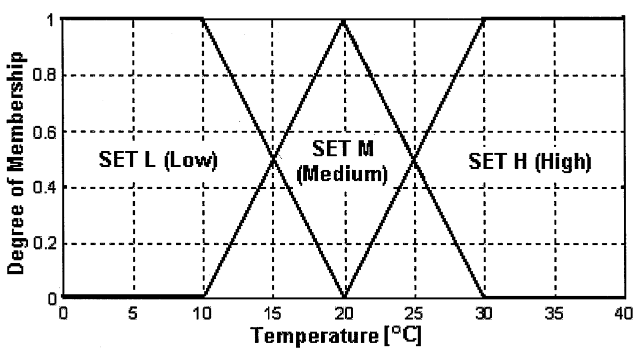


Figure 3.2 : Graphical representation of the membership functions (triangular shape, temperature control domain)

Generally, membership functions could be considered as a graphical representation of the magnitude of participation of each fuzzy set or fuzzy input [17]. Membership functions can be of any shape and form, depending on physical characteristics of the control domain. The most popular is the triangular shape. For most of the control applications triangular shapes give best quality characterization for fuzzy sets: high, medium, low (see Figure 3.2), or very high, high, medium-high, medium, medium low, low, very low (see Figure 3.3). Changing the position of each triangular membership function along axis x gives opportunity to describe system quality most accurately [20]. Although many applications can be well described with the triangular membership functions, there are cases when other forms must be used for best description of the system: trapezoidal, s-shape, bell, haversine, Gaussian, and exponential shape (see Figure 3.4). Matlab-Simulink fuzzy logic toolbox contains vast variety of membership functions and provides powerful software tools for creating ones [21]. The software toolbox allows users to start with choosing initial shape of membership function (triangular, trapezoidal, s-shape), and modify the shape and position of each membership function to create best application description. The selection of form and position of membership functions is very important especially in the design of Fuzzy Logic Controller (FLC), since the whole control of the system is based on accurate description of its qualities. The better system characteristics are described in the fuzzy membership functions, the better level of control can be achieved by FLC.

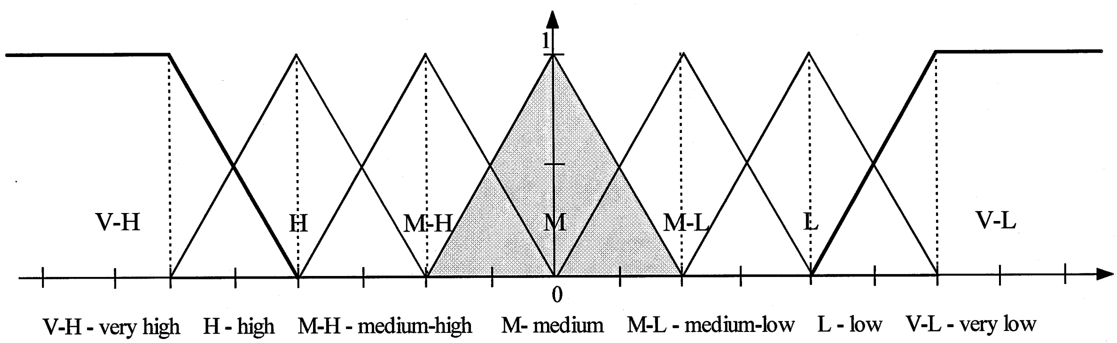


Figure 3.3 : Seven FS (Fuzzy Sets) with triangular FMF (Fuzzy Membership Functions)

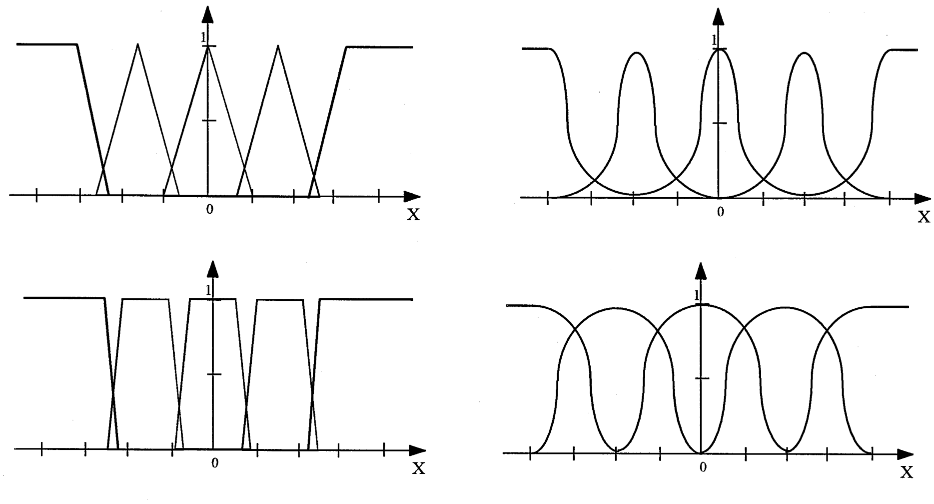


Figure 3.4 : Different forms and shapes of FMF (Fuzzy Membership Functions)

3.1.2 The Linguistic Approach

The linguistic approach defines fundamentals of the fuzzy theory. It is based on two main concepts: the linguistic variables and the linguistic term. A linguistic variable represents a concept that objectively or subjectively is measurable. It could be a temperature, pressure, time and so on. A linguistic term is a certain characteristics that the linguistic variables can satisfy. In other words, a linguistic term is a fuzzy set and a linguistic variable is its domain.

Every adequate representation of fuzzy sets involves the basic understanding of five related conceptual symbols:

- The set of elements $x \in \theta$, as in "person" from "group of friends", "room environment conditions" from the "environment" or "visible light" from "sun light".
- The linguistic variable V, that is a label for one of the attributes of the elements $x \in \theta$, as "age" of "person", "temperature" from "room environment conditions" and "visible light" of "sun light".
- The linguistic term A, is an adjective or adverb describing the linguistic variable by certain characteristic, feature, property, quality or virtue. It is totally subjective as "young" of "age", "hot" of "temperatures" and "red light" of "visible light".

- A referral set $X \subset \Re$, is the numerical measurable interval for the linguistic variable V, as "[0, 100] years" for "age", "[0, 50] °C" for " temperature" and "[400, 700] nm" for "visible light".
- A subjective numeric attribution $f_A(x)$, is a membership degree value of each element x, labeled by the linguistic variable V and described by linguistic term A.

Operating with linguistic symbols, it is possible to say that if there is a measurement process resulting in a measured value $m_V \in \Re$ for each element of $x \in \theta$, then to interpret this measurement, it is necessary to define subjective notions such as linguistic terms $A_0, A_1, A_2, ..., A_N$ together with their membership functions $f_0(x), f_1(x), f_2(x), ..., f_N(x)$, with domain $\Re$ and degree of membership ranging from 0 to 1. Applying f_i to m_V, we obtain the degree of membership for element x in the set A_i.

An example of the linguistic terms for linguistic variable of "temperature" could be the following: "almost frozen", "very cold", "cold", "pleasant", "warm", "too warm", "hot", "very hot", "almost boiling". As it was mentioned before, all notions are totally subjective.

3.1.3 Fuzzy Set Operations

The fuzzy sets, in fuzzy logic theory, can be treated like conventional sets in inter-set operations. Most common fuzzy set operations include union, intersection and complement. Thus, using an interval [0,1], instead of a set {0,1} gives infinite possibilities for the operations. It will be mostly used in aggregation operations, to average out different expert opinions [22].

Intersection of Fuzzy Sets

An intersection of two fuzzy sets is represented by a symbol $\cap$, and is a binary operation. This operation satisfies conditions which define a class of functions known as triangular-norms or t-norms. The standard t-norm is the minimum operation - min(A,B), or when applicable to a membership function - $\min(f_A(x), f_B(x))$, where $f_A(x)$ - is a membership function of the set *A* and $f_B(x)$ - is a membership function of the set *B*.

Complement of Fuzzy Sets

A compliment of a fuzzy set *A* is a set *A'* that contains all elements of the universal set, but not does not have elements belonging to set *A*. The mostly used compliment operation *A'* of a fuzzy set *A* is: $A'(x) = 1 - A(x) \quad or \quad f_A^{|}(x) = 1 - f_A(x)$

Union of Fuzzy Sets

Fuzzy union is a class of functions known as triangular-conorms (t-conorms) or s-norms, represented by a symbol $\cup$. The standard t-conorm is the maximum operation - max(A,B), or when applicable to a membership functions - $\max(f_A(x), f_B(x))$, where $f_A(x)$ - is a membership function of the set *A* and $f_B(x)$ - is a membership function of the set *B*. Other frequently used t-norm operations are listed in the Table 3.1.

Table 3.1 : t-norm, t-co-norm fuzzy operations

Type	t-norm	t-conorm
Standard	$\min(f_A(x), f_B(x))$	$\max(f_A(x), f_B(x))$
Algebraic	$f_A(x) \cdot f_B(x)$	$f_A(x) + f_B(x) - f_A(x) \cdot f_B(x)$
Limited	$\max(0, f_A(x) + f_B(x) - 1)$	$\min(1, f_A(x) + f_B(x))$
Robust	$\begin{cases} f_A(x), & if \quad f_B(x) = 1 \\ f_B(x), & if \quad f_A(x) = 1 \\ 0, & otherwise \end{cases}$	$\begin{cases} f_A(x), & if \quad f_B(x) = 0 \\ f_B(x), & if \quad f_A(x) = 0 \\ 1, & otherwise \end{cases}$

Cartesian Product of Fuzzy Sets

A Cartesian product of fuzzy sets A and B is denoted by $R = A$ x B, and if represented by membership function has a following format $f_R(x,y) = f_{A\times B}(x,y) = \min(f_A(x), f_B(x))$, where $f_A(x)$ and $f_B(x)$ are membership functions of the fuzzy sets A and B.

Fuzzy Set Aggregation

An aggregation operation in fuzzy set theory is a combination of several sets together, usually in application oriented manner, to produce a unique fuzzy set. For instance, in the intersection and union of any number of fuzzy sets, the unique set is produced. In general, many other types of aggregation may be performed on fuzzy sets.

Symmetric Sum of Fuzzy Sets

Symmetric sum provides another option for aggregation of fuzzy sets and membership functions. More formally, these are *n*-argument functions, denoted by *S_sum*. The symmetric sum can be represented by (3.6),

$$S\_sum(x_1, x_2, \ldots, x_N) = \left[1 + \frac{p(1-x_1, 1-x_2, \ldots, 1-x_N)}{p(x_1, x_2, \ldots, x_N)}\right]^{-1} \tag{3.6}$$

where p is an increasing continuous function witch satisfy: $p(0,0,\ldots,0) = 0$

Averaging Operations on Fuzzy Sets

Generalized mean average for N arguments is represented by (3.7) where $p \in R$ and $p \neq 0$

$$A(x_1, x_2, \ldots, x_N) = \sqrt[P]{\frac{1}{N}\sum_{i=1}^{N}(x_i)^P} \tag{3.7}$$

A detailed description of family of means generalized by parameter p is given in Table 3.2

Table 3.2 : Family of mean operations on FS (Fuzzy Sets)

Type of Mean	**Formula**
Arithmetic Mean ($p = 1$)	$A(x_1, x_2, ..., x_N) = \frac{1}{N}\sum_{i=1}^{N} x_i$
Geometric Mean ($p \to 0$)	$A(x_1, x_2, ..., x_N) = (x_1, x_2, ..., x_N)^{1/N}$
Harmonic Mean ($p = -1$)	$A(x_1, x_2, ..., x_N) = \frac{N}{\sum_{i=1}^{N} \frac{1}{x_i}}$
Minimum ($p \to -\infty$)	$A(x_1, x_2, ..., x_N) = \min(x_1, x_2, ..., x_N)$
Maximum ($p \to \infty$)	$A(x_1, x_2, ..., x_N) = \max(x_1, x_2, ..., x_N)$

A certain generalization of all averaging operations is a family of so-called quasi-arithmetic means (3.8), where f is any continuous, strictly monotonic function.

$$A(x_1, x_2, ..., x_N) = f\left(\sqrt[P]{\left[f^{-1}x_1, x_2, ..., x_N)\right]^P}\right) \qquad (3.8)$$

Linguistic Modifiers

Linguistic modifiers, or modified linguistic terms, are fuzzy models counterpart of adverbs. When linguistic term is defined, for example "hot", it is expected that the concept of "hot" can be broken into stages, such as "very hot" or "almost hot". Even though it is difficult precisely to determine what affect the modifier "very" has, it does have an intensifying effect. The modifier "more or less" or "almost", has the opposite affect. The modifiers are often approximated by the mathematical expressions: $very \quad A \equiv A^2$, $almost \quad A \equiv A^{1/2}$. Operator "very" is referred to as *concentration*, and operator "almost" is referred to as *dilation.* The power function in linguistic modifier is applied to each element of set A. If $f_A(x)$ represents a membership function of a set A, the power function is applied to a membership function, producing $(f_A(x))^2$ for "very" modifier, and $(f_A(x))^{1/2}$ for "almost" modifier. The modifier "very-very", by deduction will produce $very - very \quad A \equiv A^4$

A whole family of linguistic modifiers is generated by A^P, where *p* is any power between zero and infinity. For $p = \infty$, the modifier could be named *exactly* because it would suppress all membership lower than 1. Most common linguistic modifiers and their corresponding membership functions are listed in Table 3.3

Table 3.3 : Fuzzy linguistic modifiers

Modifier	Function
Very	$(f_A(x))^2$
Extremely	$(f_A(x))^3$
Very-very	$(f_A(x))^4$
Somewhat, More or Less	$f_A(x)$
Almost	$(f_A(x))^{1/2}$
Slightly	$(f_A(x))^{1/3}$

3.2 Design of Fuzzy Systems

The design and development of fuzzy system could be summarizes in a number of steps described below [13] [18] [23] [27].

1) Gain sufficient knowledge on the desired behavior and characteristics of the system.
2) Apply this knowledge in the design of fuzzy sets and membership functions.
3) Create fuzzy rules that best suites desired behavior of the system.
4) Apply one of the fuzzy reasoning operations (max-min, max-dot, averaging, root-sum-square) that best fits the expected outcome of fuzzy reasoning.
5) Design defuzzification. This step contains the following sub-steps: a) design output fuzzy sets and membership functions that best fit the transformation from fuzzy to crisp domain; b) Choose one of the following output defuzzification principles: max-membership, average membership, fuzzy Centroid, root-sum-square.
6) Validate your system and perform adjustments to steps 2 to 5 until desired behavior is achieved.

3.2.1 Fuzzification

Fuzzification is a process of dividing control domain into fuzzy sets and choosing membership functions for each set. These two steps of the Fuzzification process are largely depending on the knowledge of a real-life system behavior. The more knowledge and human–like experience of the system behavior is obtained, the more precisely it could be reflected by the shapes of membership functions. It is reasonable strategy to start with the development using simple membership functions and refine the design later with more complex application specific function, when the proper knowledge of the system behavior is obtained.

The following are known methods to determine membership function from the expert knowledge [20] [24] [25]:

1) Intuition. In this approach, the designers use their previous knowledge of the system trying to decide what curves are to be used.

2) Horizontal method. According to this statistical method, the information about the membership value of a given set is gathered from polling expert opinions on the system characteristics, accepting only "yes" or "no" answers. The estimated value of a membership function is the ratio of number of positive replies to the total number of replies.

3) Vertical method. According to this statistical method, the set is built by using its α-cuts. The question that experts are polled is "is the element *K* is α % compatible with the concept that describes the set?" The estimated value of a membership function is the ratio of number of positive replies to the total number of replies.

4) Pairwise-comparison method. In this method every pair is compared using some discrete level of preference (usually 7±2), and algebraic operations determine membership values.

5) Inference. This is rather class of methods that, according to this method, the previous knowledge is used to perform deductive reasoning.

3.2.2 Fuzzy Rules

Fuzzy rules is a collection of statements that characterizes the behavior and a response of a fuzzy system [26] [27]. These rules can be represented in the form of "if-then" statements,

$$\text{IF } a_1 \text{ AND } a_2 \text{ AND } \ldots a_{i\ldots} \text{ AND } a_n \text{ THEN } b \tag{3.9}$$

where a_i, $i = 1 \ldots n$, and b are fuzzy propositions. Each rule represents an assumption of the system's response to a specific input, where both the input and the output are represented by fuzzy sets.

In control systems, fuzzy rules can be arranged such that number of a_i propositions is the same as the number of system input variables. Thus, for a system that has two inputs, x and y, with three membership functions $f_1(x), f_1(x), f_1(x)$ for input x, three membership functions for $f_1(y), f_1(y), f_1(y)$ for input y, and output (defuzzification) membership functions $f_{OUTt1}(z))$, $f_{OUTt2}(z))$, $f_{OUTt3}(z))$ for output z, the following system of rules will cover all possible combinations of input–output relations:

$$\begin{aligned}
&1)\ \text{IF } [x \text{ is } f_1(x)] \text{ AND } [y \text{ is } f_1(y)] \text{ THEN } z_1 = f_1(x) \circ f_1(y), \quad z_1 \in f_{OUT3}(z) \\
&2)\ \text{IF } [x \text{ is } f_1(x)] \text{ AND } [y \text{ is } f_2(y)] \text{ THEN } z_2 = f_1(x) \circ f_2(y), \quad z_2 \in f_{OUT3}(z) \\
&3)\ \text{IF } [x \text{ is } f_1(x)] \text{ AND } [y \text{ is } f_3(y)] \text{ THEN } z_3 = f_1(x) \circ f_3(y), \quad z_3 \in f_{OUT1}(z) \\
&4)\ \text{IF } [x \text{ is } f_2(x)] \text{ AND } [y \text{ is } f_1(y)] \text{ THEN } z_4 = f_2(x) \circ f_1(y), \quad z_4 \in f_{OUT3}(z) \\
&5)\ \text{IF } [x \text{ is } f_2(x)] \text{ AND } [y \text{ is } f_2(y)] \text{ THEN } z_5 = f_2(x) \circ f_2(y), \quad z_5 \in f_{OUT2}(z) \\
&6)\ \text{IF } [x \text{ is } f_2(x)] \text{ AND } [y \text{ is } f_3(y)] \text{ THEN } z_6 = f_2(x) \circ f_3(y), \quad z_6 \in f_{OUT1}(z) \\
&7)\ \text{IF } [x \text{ is } f_3(x)] \text{ AND } [y \text{ is } f_1(y)] \text{ THEN } z_7 = f_3(x) \circ f_1(y), \quad z_7 \in f_{OUT3}(z) \\
&8)\ \text{IF } [x \text{ is } f_3(x)] \text{ AND } [y \text{ is } f_2(y)] \text{ THEN } z_8 = f_3(x) \circ f_2(y) \quad z_8 \in f_{OUT1}(z) \\
&9)\ \text{IF } [x \text{ is } f_3(x)] \text{ AND } [y \text{ is } f_3(y)] \text{ THEN } z_9 = f_3(x) \circ f_3(y) \quad z_9 \in f_{OUT1}(z)
\end{aligned} \tag{3.10}$$

where $f_1(x) \circ f_1(y)$ - is a designated composition or inference operations. For most systems, a standard *t-norm* and *t-conorm* operators as OR - $\min(f_A(x), f_B(x))$, or AND - $\max(f_A(x), f_B(x))$ are used. However, to improve performance, more complex operations listed in Table 3.1, can be considered.

The same set of expressions can be rearranged in the form of a matrix (See Figure 3.5). This matrix representation of fuzzy rules is referred to as Fuzzy Rule Matrix (FRM).

y\x	$f_1(x)$	$f_2(x)$	$f_3(x)$
$f_1(y)$	1 $z_1 = f_1(x) \circ f_1(y)$ $z_1 \in f_{OUT3}(z)$	2 $z_2 = f_1(x) \circ f_2(y)$ $z_2 \in f_{OUT3}(z)$	3 $z_3 = f_1(x) \circ f_3(y)$ $z_2 \in f_{OUT3}(z)$
$f_2(y)$	4 $z_4 = f_2(x) \circ f_1(y)$ $z_4 \in f_{OUT3}(z)$	5 $z_5 = f_2(x) \circ f_2(y)$ $z_5 \in f_{OUT2}(z)$	6 $z_6 = f_2(x) \circ f_3(y)$ $z_6 \in f_{OUT1}(z)$
$f_3(y)$	7 $z_7 = f_3(x) \circ f_1(y)$ $z_7 \in f_{OUT3}(z)$	8 $z_8 = f_3(x) \circ f_2(y)$ $z_8 \in f_{OUT1}(z)$	9 $z_9 = f_3(x) \circ f_3(y)$ $z_9 \in f_{OUT1}(z)$

Figure 3.5 : Fuzzy Rule Matrix (FRM)

Hence, a system with 2 inputs, each carrying 3 fuzzy membership functions yields 9 variations of fuzzy rules, which can be arranged in an FRM of dimensions 3x3. The same system, with 3 inputs of *m* membership functions yields m^3 rules, which can be arranged in cube matrix $m \times m \times m$. Thus, a k-input system, with *m* membership functions yields m^k rules, which comprise *k* dimensional matrix of *m*.

The implementation of fuzzy system with k inputs can be complicated with *k* exceeding 3, since fuzzy rule matrix becomes k-dimensional and each input is associated with a number of membership functions. The number of membership functions *m*, has similar influence on the complexity of fuzzy system. Systems with large number of inputs and membership functions can be difficult in real-time implementation due to calculation complexity.

3.2.3 Fuzzy Reasoning (Inferencing)

A fuzzy reasoning engine performs a composition of outputs of fuzzy rules into consolidated fuzzy system response. Fuzzy reasoning is divided into two basic steps:

1) Calculation of the consequent value of the each fuzzy rule.

2) Calculation of the consolidated result.

Fuzzy Reasoning (FR) is also known as the compositional inference engine. First step of fuzzy reasoning calculates the compositional value of each fuzzy rule using one of the implication operators listed in Table 3.1. Second step derives logical output from compositional values of each rule by using one of the inference techniques. The following inferencing techniques are available for fuzzy reasoning:

1) The ***max-min*** (Mamdani) inferencing method is based on testing the magnitude of each fuzzy rule value.

$$I_{OUT} = Inf\{z_1, z_2, ..., z_N\} = \max\{z_1, z_2, ..., z_N\} \tag{3.11}$$

   where N denotes the maximum number of fuzzy rules that fall under specific output membership function.

2) The ***max-dot*** or *max-product* method scales each member function to it under its respective peak value and takes the horizontal coordinate of the fuzzy centroid of the composite area under the functions as the output. This method combines the influence of all active rules and produces smooth, continuous output.

$$I_{OUT} = Inf\{z_1, z_2, ..., z_N\} = \frac{f_{OUT}(z_1)\cdot z_1 + f_{OUT}(z_2)\cdot z_2 + ... + f_{OUT}(z_N)\cdot z_N}{z_1 + z_2 + ... + z_N} \tag{3.12}$$

3) The ***averaging*** method is another approach that is used in fuzzy inferencing, but fails to give increased weighting to more rule votes per output membership function. For instance, if three "negative" rules fire, but only one "zero" rule, the averaging will not reflect the difference, and each function will be clipped at the average.

$$I_{OUT} = Inf\{z_1, z_2, ..., z_N\} = \frac{z_1 + z_2 + ... + z_N}{N} \tag{3.13}$$

4) The ***root-sum-square*** (RSS) method combines effects of all applicable fuzzy rules. It scales the functions at their respective magnitudes and computes the Fuzzy Centroid of the composite area. This method is more complicated for implementation than other methods since it involves a square root computation; however it gives the best weighted influence of all firing rules.

$$I_{OUT} = Inf\{z_1, z_2, ..., z_N\} = \sqrt{z_1^2 + z_2^2 + ... + z_N^2} \quad (3.14)$$

3.2.4 Defuzzification

Defuzzification or Fuzzy Decoding transforms a fuzzy quantity into a crisp number. Most of control systems require this step, since feedback control loops require crisp values [23]. In control systems [28], defuzzification rely on output membership functions, which determine the sort of actions to be taken by the system. The output control domain is a linguistic variable z, where $f_{OUT1}(z), f_{OUT2}(z), f_{OUT3}(z)$ - are output membership functions that determine output values of the system. The output fuzzy sets are similar to the input fuzzy sets. Each set is associated with its membership function, and represents physical characteristics of the controlled domain. For instance, in case of temperature control, the output fuzzy sets can be "apply heat", "do nothing", "apply cool". The output crisp value for temperature control is deducted by defuzzification from inference results $I_{OUT1}, I_{OUT2}, I_{OUT3}$, and is given by (3.15), where N is a number of output membership functions.

$$OUTPUT = Deff\{I_{OUT1}, I_{OUT2}, ..., I_{OUTN}\} \quad (3.15)$$

Figure 3.6 demonstrates defuzzification process that yields 63.5% of cooling to be applied to the system as result of fuzzy Centroid defuzzification.

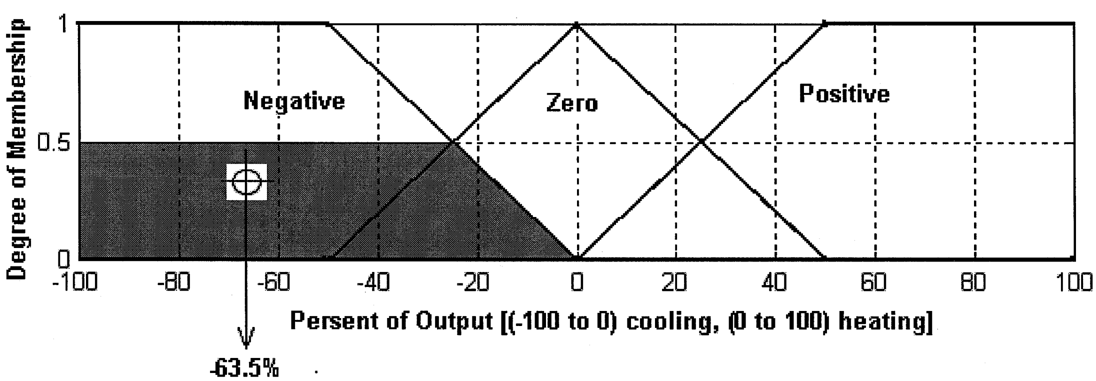


Figure 3.6 : The horizontal coordinates of fuzzy Centroid is taken as a crisp output

Mostly used defuzzification principles are listed below.

1) ***Max-membership*** principle (height method). According to this method, the maximum relative membership is used to determine the output value.

$$OUTPUT = MAX\{I_{OUT1}, I_{OUT2}, \ldots, I_{OUTN}\} \tag{3.16}$$

2) ***Average membership*** principal. According to this method, an average of all output inferencing products is taken as a crisp output.

$$OUTPUT = \frac{I_{OUT1} + I_{OUT2} + \ldots + I_{OUTN}}{N} \tag{3.17}$$

3) ***Fuzzy Centroid*** principle. According to this method, the weighted strengths of each output member function multiplied by their respective output membership function center points and summed. Ultimately, this area is divided by the sum of weighted number function strengths and the result is taken as the crisp output.

$$OUTPUT = \frac{I_{OUT1} \cdot center\{f_{OUT1}(z)\} + I_{OUT2} \cdot center\{f_{OUT2}(z)\} + \ldots + I_{OUTN} \cdot center\{f_{OUTN}(z)\}}{I_{OUT1} + I_{OUT2} + \ldots + I_{OUTN}} \tag{3.18}$$

4) The ***root-sum-square (RSS)*** principle is similar to the RSS inference technique, with the difference that it is applied to the inferencing products of fuzzy reasoning. This method produces crisp output that is a combination of all inferencing products.

$$OUTPUT = \sqrt{{I_{OUT1}}^2 + {I_{OUT2}}^2 + \ldots + {I_{OUTN}}^2} \tag{3.19}$$

The defuzzification and inference technique can be combined together in one process that is referred to as a Defuzzification-Inferencing technique. The Mean-Max, for example, is a combination of average inferencing and max-membership defuzzification.

3.3 Fuzzy Logic Systems Overview

Any system equipped with fuzzy logic reasoning is considered to be a fuzzy system. It can be a sub-module of a non-fuzzy system, or an entire system designed to perform fuzzy reasoning. The implementation strategy and nature of fuzzy logic call for a fuzzy component to be isolated from the non-fuzzy (crisp) parts of the system. The connections between fuzzy and crisp components are practically realized through *crisp to fuzzy* and *fuzzy to crisp* converters also known as Fuzzification and Defuzzification modules. These modules are the integrated parts of any fuzzy system. The top-level block diagram of a generic fuzzy logic system is illustrated in Figure 3.1.

Fuzzy logic systems can be designed to perform many different tasks and can be used in many applications. Broadly fuzzy systems can be divided into four categories:

1) Prescriptive systems. These systems undertake specific decisions in response to the changes of the input signals. Control systems fall under this category.
2) Descriptive systems. These systems perform identifying event, problem, or object by classification characteristics. Detection, monitoring, pattern recognition and diagnostic systems are classified as descriptive systems.
3) Optimization systems. These systems establish conditions and actions necessary to achieve some degree of efficiency in performance characteristics of the system. The optimization can be performed for complexity, time, speed, or any other controlled domain.
4) Predictive systems. These systems require predicting possible future outcome of some process or controlled domain.

Fuzzy logic systems have found their ways into many industrial applications. They are currently used in medical field, aviation, aero-space industry, robotics, seismology, heavy machinery, navigation, domestic electronics, shipbuilding, automobile industry, and even in education. These systems are designed to perform many different tasks as monitoring, identifying, controlling, tracking, diagnosing, predicting and optimizing.

This chapter presents an overview of modern fuzzy logic systems and applications in which the fuzzy logic is applied. Fuzzy systems are broken into classifications according to their usage and each class is described in details with references to existing or currently proposed fuzzy systems, as well as those that are still in research. This chapter also presents an overview of design and implementation methodology of fuzzy logic, which is practically used for development of fuzzy systems and applications.

3.3.1 Monitoring Systems

Monitoring systems are designed to monitor environment for specific events and produce response on occurrence of these events. Monitoring systems with fuzzy logic reasoning has found large usage in many different fields. One of them is medical. In [29], a fuzzy logic system for monitoring SIDS risk infants life threatening events is presented. In this system, fuzzy algorithms are used to monitor respiration, electrocardiogram (ECG) and blood oxygen saturation (SpO_2), to make an intelligent assessment of the potential life threatening events. Each monitoring signal has a corresponding fuzzy logic identification module that includes membership functions and rule matrix to produce an alarm when decided so by defuzzification process with fuzzy Centroid. The system is able to monitor simultaneously all input events. The fuzzy logic algorithm is implemented as a software program on a microcontroller, and consists of multiple inference composition and 6-rules based fuzzy logic. A system with very similar 6-rules fuzzy logic concept is presented in [30]. This system performs monitoring cardiovascular signals for cardiac problems detection. This fuzzy logic approach of this system is similar to [29] except that this system incorporates a unique fuzzy logic decision-function that smoothes probability distribution for patient parameters.

A little different fuzzy logic approach is presented in [31], where an online advisory system for monitoring and controlling the depth of inhaled anesthesia is described. This system monitors online measurements of pressure, heart rate and other clinical information, and advises of recommended dosage of anesthesia for the patient. The system combines a feedback of patient reaction to the introduced level of anesthesia and fuzzy reasoning controller producing a recommended dosage. The fuzzy logic controller has two multiple inputs corresponding to clinical data and flexible fuzzy rules mechanism that allows modification of fuzzy rules depending on level of monitoring.

Another system for monitoring morphological changes in the spinal cord for people with back injury is presented in [32]. A fuzzy logic tissue classification algorithm is used to segment array and white matter regions for morphometric analysis. The fuzzy logic system uses 2-inputs, 3-Gaussian membership functions with 3x3 fuzzy rule matrix, and fuzzy Centroid based defuzzification mechanism, to classify grey matter regions and intact white matter tracts. The fuzzy logic Toolbox in MATLAB is used for implementation of the system.

In aviation, health monitoring system has been developed for composite helicopter rotor [33], where different physical parameters of the rotor blades are analyzed by fuzzy algorithms. Genetic fuzzy system is developed for global online prediction of physical damage and life consumption using displacement- and force-based measurement deviations between damaged and undamaged conditions. The authors report an excellent performance of the system when processing noisy imprecise data. In [34], the jet engine health monitoring system using fuzzy logic controller is presented. The fuzzy logic controller has been developed to control engine start-up sequence and several discrete health and safety monitoring controls for run time and shut down sequence of the engine. The Bayesian network has been developed to alter fuzzy logic control through modification of fuzzy rules to adapt to sensor malfunctioning without having to shut down the engine.

Fuzzy logic has been successfully applied to other industrial applications. In [35], fuzzy logic has been applied to monitoring of oil-insulated transformers, where fuzzy logic is replacing conventional logic to cope with difficult task of maintenance and diagnostic of different transformers. In this application fuzzy logic has shown ability to deal with imprecise characteristics of the transformers to produce reliable detection of major faults. In [36], the authors present a monitoring system for drill wear. This system incorporates fuzzy logic with back-propagation neural network, which allows adaptive monitoring and human-like assessment of drill wear. The triangular membership functions have been chosen for this application to reduce computational load, and 3 layers of fuzzy neural networks have been chosen for adaptive control. Another application of monitoring and protecting power systems for voltage and current disturbance is presented in [37]. This system uses fuzzy logic for detection and classification of disturbances in power lines. Each voltage and current monitoring has individual fuzzy logic control modules. The systems uses one-input, 6-triangular membership functions, and 6-rules rule evaluation module to monitor disturbances. To achieve fast response time, the system is implemented on Field Programmable Gate Array (FPGA), and is presented as a Hardware Description Language (VHDL) model.

An induction motor fuzzy logic monitoring systems is presented in [38]. The main difficulty in induction motor diagnosis is the lack of an accurate analytical model to describe a faulty motor. A fuzzy logic approach successfully copes with this problem. This work presents a reliable method for the detection of stator winding faults based on monitoring the line/terminal current amplitudes. The fuzzy logic control consists of 3-inputs, up to 4 trapezoidal and triangular membership functions and 13 rules. The whole algorithm is implemented and simulated on MATLAB environment.

Another important field, that fuzzy logic monitoring has been applied to, is a seismic activity monitoring. A fuzzy logic system has been developed to monitor an earthquake activity and to reduce the absolute motion of the structure base [39]. The structure is controlled through a single force applied from a controllable MR damper placed on the first storey of the structure. The restraining control force is computed in real time, using an evolutionary fuzzy logic controller. The functionality of the proposed system has been verified through extensive simulation of a six-storey structure, using disparate earthquake ground motions.

Even in education, the monitoring students' actions system using teachers' expertise was developed and presented in [40]. This system uses fuzzy logic to produce subjective assessments based on teachers' experience. The system incorporates a networking model of fuzzy logic modules, where each input has its fuzzifier and each fuzzy module incorporates inputs from all fuzzifiers. The defuzzification stage is individual for each fuzzy module, thus producing multiple outputs. This system has very large reasoning capability equipped with neural networks, however the fuzzy rules matrixes are very large that require extensive computational power. That approach is suitable for off line computations or simulations.

3.3.2 Detection and Identification Systems

Detecting and identifying are similar tasks to monitoring, but usually more emphasis is done on detection and identification parts as oppose to monitoring process. In medical field, the biggest usage of fuzzy logic systems is the detection of different factors that help diagnosing the patient. Some of these can be addressed only by fuzzy logic, based on its ability to process noisy and imprecise data. Some of the systems use dedicated pattern recognition techniques for medical images that often have the property of fuzziness. A new image segmentation technique using intuitionistic fuzzy set has been proposed in [41], and may be applied to many real applications such as medical informatics and bioinformatics, where information is imprecise or uncertain. This paper present very effective fuzzy logic based pattern recognition technique. In [42], the similar approach is presented. The detection severity of traumatic brain injury system is presented. Based on Electroencephalography, Trauma and Glasgow coma scores were used for evaluating the system and the results were compared with the findings of neurologists. The proposed fuzzy logic controller has two inputs: trauma source with 9 trapezoidal membership functions, and EEG frequency with 5 trapezoidal membership functions. The output fuzzy variable is a degree of trauma (%) has 5 trapezoidal membership functions. The authors have found significant relationship between the findings of neurologists and systems output for normal, mild and severe electroencephalography tracing data.

Another medical system, a distributed adverse drug reaction detection system using fuzzy logic recognition-primed decision model is presented in [43]. The system comprises of a multi-node framework of intelligent agents (typically a specialists) connected to the Recognition-Primed Decision (RPD). The RPD model utilizes fuzzy logic technology to not only represent, interpret, and compute imprecise and subjective cues that are commonly encountered in the drug reaction detection problem but also to retrieve prior experiences by evaluating the extent of matching between the current situation and a past experience. Fuzzy sets are employed to formalize the representation of imprecise cues. Due to a large number of interconnected agents, the realization of this system requires networking and extensive computational power.

An interesting research in genomic using fuzzy logic concept is presented in [44]. In genome projects, typically, large quantities of data are processed to produce useful biological knowledge. Fuzzy systems are suitable for uncertain or approximate reasoning when systems are difficult to describe by mathematical modeling. The authors propose two applications of fuzzy logic: building a gene interaction model and assessing the accuracy of DNA bases called by a DNA base-calling algorithm. The first application contains a two-input 3 triangular membership functions, and 3x3 fuzzy rules matrix. The second application contains more complex fuzzy model with 3 two-input fuzzy logic modules connected to one 3-inout fuzzy module with trapezoidal membership functions. It is concluded that two-level fuzzy logic system produces very reliable results for assessing the quality of DNA.

Computer security threat detection system based on fuzzy logic concept is proposed in [45]. The system performs computer security threat evaluation modeling and anomaly detection based on fuzzy logic ability deal with the ambiguities and imprecision. The authors introduce a hierarchical fuzzy inference system to capture normal behavior deviations. Very similar fuzzy inference system, but for networking application, is presented in [46]. This fuzzy logic system monitors TCP transmission and detects transmission problems by distinguishing losses packets between bit error and congestion. The main difference is that this system uses adaptive neural network in conjunction with fuzzy inferencing. Similar system but without adaptive capability but processing larger amount of inputs is presented in [47]. This system detects electric power disturbances caused by voltage sags, swells, momentary outage and capacitor switching transient events. Another very complex system for analysis of water supply is proposed in [48], where the fuzzy logic classification inferencing is combined with artificial neural network to produce learning and adapting analysis system.

In [49], fuzzy logic control algorithm is used for in-field plant sensing system. This system uses a multi-spectral imaging sensor to produce image for nitrogen detection. The image quality is essential in this approach. The fuzzy logic algorithm is applied to automatically adjust the camera exposure and gain to control image brightness. The fuzzy logic controller has one input with 4 triangular asymmetrical membership functions, and 4 output membership functions for gain/exposure control. Unlike PID, FLC in this application achieves excellent and steady performance for this non-linear control domain, since human knowledge of exposure and gain adjustment was implemented in membership functions and fuzzy rules. The whole algorithmical part of this system including FLC was implemented on a portable computer connected to the sensing system.

3.3.3 Control Systems

Control systems are designed to perform control on an environment that is referred to as a control domain. In control systems, fuzzy logic is vastly used as an alternative to conventional Proportional-Integral-Derivative (PID) control. Many systems incorporate both Fuzzy and

PID control, since there are cases when the control achieves best results through strict conventional logic (when linear behavior of the system is expected and high degree of precision is needed), and there are cases when conventional logic cannot deal with real-world uncertainties, nonlinearity or noises and distortions. Fuzzy logic can be applied in all these cases and achieve astounding results.

In [50], a system of automatic control vehicle control is presented. In this article, the authors present an automated version of Citroen Berlingo equipped with embedded fuzzy logic control system for speed and steering control. This system incorporates human intelligence and behavior to implement an automatic driving. The inputs of the guidance systems are a CCD color camera, high precision global positioning system, tachometer, and wireless LAN; the outputs are DC motor for steering wheel, analog card for throttle, and DC motor for brake. The inputs are broken down into sub-components, thus camera input is analyzed by image recognition systems to produce member of fuzzy inputs: Angular Error, Lateral Error etc. Each fuzzy input has 3 triangular membership functions that best describe human experience on driving control. The large fuzzy reasoning and rule matrix brings all fuzzy inputs together to generate 3 outputs: steering wheel, throttle, and brake. All fuzzy reasoning is implemented on PC-based computer onboard the vehicle. The fuzzy logic controller perfectly mimics the human driving behavior driving on a freeway, keeping distance from other vehicles and even overtaking. The system adequately responds to highly dynamic road situation involving stop-and-go. The future work on this topic includes creating possibility do deal with more complex situations on the road involving intersections, turning and extensive maneuvering.

Fuzzy logic controllers have been proposed to many different control applications in vehicle control. In [51], an active suspension control system of a vehicle model that has five degrees of freedom with a passenger seat using a fuzzy logic controller is proposed and studied. The goal of this study is to achieve vibration compensation for the passenger seat to maintain best comfort. Similar fuzzy logic system controlling electrical vehicle power supply is presented in [52]. The system controls a combination of a fuel cell power source and two energy storage devices, i.e., batteries and ultra-capacitors. The control strategy is designed to achieve the high-efficiency operation region of the individual power source and to regulate current and voltage at peak and average power demand, without compromising the performance and efficiency of the overall system. In this system, the fuzzy logic control is based on classical 2-inout 9-rule fuzzy matrix with triangular and trapezoidal membership functions, and defuzzification Centroid, however an original fuzzy approach is taken. The systems uses tree different sets of fuzzy rules for three different driving conditions: normal, acceleration, and braking. The experimental testing has shown ability to maintain the dc output stable under different driving conditions.

In [53], a Hybrid Electric Vehicle (HEV) control strategy using fuzzy logic is proposed. HEV combines an internal combustion engine and electric generators. To make HEV effective, the proper management of different energy sources is required. The proposed generic HEV control strategy can manage produced, saved and used energy to minimize fuel consumption and emissions, while maintaining driving performance of the vehicle. The system uses classical 2-input, 3MP (triangular shape), 9-rules fuzzy logic controller. The inputs to the FLC are the state of change – a measured value of engines performance, and the required torque. The output is the engine management signals. The optimized pattern to minimize vehicle emission is embedded in the membership functions of fuzzy logic controller, and general vehicle-dependant tuning strategy to achieve maximum performance and minimum emission is advised.

3.3.4 Tracking Systems

Tracking, or object tracking systems are considered to be a sub-classifications of control systems. Tracking systems are usually designed to track or follow specific object or event. In most applications, tracking systems follow visual image of moving object and predict its trajectory. The fuzzy logic concept has not been introduced to tracking systems until early 90^{th}. Since then, the new horizons for tracking control have been opened. The possibility to apply human-like reasoning to deal with uncertainties that are experienced in most tracking applications has been achieved. One of the first tracking system exercising fuzzy logic concepts is proposed in [54]. The fuzzy logic system has been designed to track moving objects with constant or accelerating speed on straight forward road or space with almost zero tracking errors. The proposed system can trace a car, tank, artillery, or a similar object. The FL based approach allows targets to be unclear, imprecise, noisy, and still maintain good tracking characteristics. This system does not use a stereo-vision camera tracking identification of the object based on occlusion and disparity, neither this system has a stereovision center of the object identification partially imbedded in the fuzzy membership functions set of rules. The system bares a strict software implementation. Later on, more complicated object tracking systems have been proposed [55] [56]. These systems are based on two new video tracking segmentation algorithms. The authors present a novel Centroid tracker based on distance features in cluttered image sequencing. The proposed target classifier adopts the fuzzy-reasoning segmentation instead of the estimation of the state-conditional densities. Comparative experiments show that the performance of the proposed fuzzy-reasoning segmentation algorithms is superior to that of the conventional thresholding methods. The real-life experiments with real target images have shown that the tracking results are good and stable. A medical image segmentation system, sharing similar fuzzy-reasoning approach is proposed in [41]. In this system, the intuitionistic fuzzy set theory has been used to extract information by reflecting and modeling the uncertainty present in real-fife situations.

A multi-object evolutionary fuzzy modeling for docking maneuver of an automated guided vehicle is presented in [57]. In many real world applications, mobile robots require interacting with objects by performing docking tasks in precise manner. The authors present a soft computing technique based on a multi-objective evolutionary algorithm in order to find multiple fuzzy logic controllers which optimize specific objectives and satisfy specific constraints for docking task. The system has applied for a fork-lift truck docking that must perform docking maneuvers to load pallets in conveyor belt. Another multi-object tracking system is presented in [58]. In this paper, the authors propose a new method for determination of a motion trajectory based on a trisectional structure. The method distinguishes between real-world objects and abstract objects, as well as determines the motion trajectories for moving objects.

A fuzzy color tracking system for robotic tasks is presented in [59]. In this paper, the authors present a method of track an object by using a color cue. The colors are presented as fuzzy membership functions in the color space. Target initialization is done through color selection. The system is tracking all objects of selected color. The target search is based on examining the pixel intensities over a test region of the current image using a fuzzy logic rules. As soon as object of interest is acquired, the system tracks its location and reflects dynamic color change of the object.

In [60], the authors present an innovative fuzzy logic controller for tracking objects in the context of robotic soccer games. This system uses type-2[2] fuzzy logic connected by four rules [61]. The image processing is used to estimate the angle between the robotic agent and the tracking target, which introduces uncertainty in the control loop. The other sources of uncertainty are related to the defects in the control loop. The paper investigates how the type-2 fuzzy logic overcomes these sources of uncertainty.

In [62], a fuzzy logic control system to suppress noises of a laser racking system is presented. This system can be classified as a strict control application, with an exception that is applied to tracking system. In this research paper is shown how Fuzzy Logic Controller (FLC) can outperform a conventional Proportional-Integral-Derivative (PID) controller in this application. A practical comparison between PID and FLC is presented in this article. The conclusion drawn by this study is that FLC is able to suppress strong coupling effects and nonlinearities existing in a laser tracking system, which cannot be handled by PID.

Another important field, where FL is being implemented is a robotic and vehicle tracking control. In [63], a fuzzy logic based trajectory control and tracking system for very large crude carrier is presented. A very complex control system of a ship motion in confined waters is based on two different controllers connected in parallel. One controller is based on a robust technology and used for maneuvering control (geometric and dynamic task). The other controller is based on FL technique and used for trajectory tracking (geometric task) using forward trust of the main propeller for speed control and the rudder deflection to minimize cross tracking error. The decision which controller to use is dependant on the velocity of the vessel. The control system was initially implemented on a nonlinear multi-variable simulation model developed on MATLAB/Simulink, and later tested on a real-time object – floating model of very large crude carrier.

---

[2] Type-2 fuzzy logic represents uncertainty using a function which is itself a type-1 (regular) fuzzy number. This way a two level fuzzification is achieved. If type-1 fuzzy set maps elements in the range of [0,1], a type-2 fuzzy sets maps element in the crisp domain to type-1 fuzzy numbers. Since values at each point in a type-2 fuzzy set are given by a function, type-2 fuzzy sets are three-dimensional [61]. Sometimes type-2 fuzzy logic is referred to as fuzzy-fuzzy.

Another tracking controller for motion control and drives is presented in [64]. The authors present an implementation of most optimal fuzzy logic tracking controller using low-cost Motorola microprocessor. The optimization goal was to design low cost yet practical controller that can be implemented and marketed and give respectable performance. The proposed controller is robust and is based on set of simple rules that are derived form engineering and experimental results. The results of the system controlling DC brushless motors indicate effectiveness of proposed controller for both speed and position trajectories. The experiments, presented in the paper, have shown that FLC performs significantly better than bang-bang controller, providing smooth and overshoot free control.

In [65], a tracking path algorithm based on fuzzy logic has been implemented on a FPGA platform for an autonomous mobile robot. The algorithm integrates planning activity, which provides goals for the robot, with behavior based reactivity, implemented by fast control module. The FLC is designed on a base of one input, three membership functions for control a rotation of a steering wheel of tricycle robot. The FPGA implementation gives many advantages to the control in terms of configurability and design reprogramming. The VHDL modeling of the controller gives increased modular reusability and reconfigurability.

An automatic focusing system for optical tracking instrumentation based on fuzzy logic control is presented in [66]. The system was implemented on large telescope system used for optical data collection including attitude and miss-distance information. The system is given only target range and is providing a highly accurate focusing on the target. The initial controller was implemented as PID, which gave sufficient control tracking targets with slow velocity dynamics. For high velocity moving targets, the fuzzy model was chosen as an alternative to a PID model, since its capability of coping with nonlinear errors and difficult models. A 2-input, 5-membership function FLC was designed and tested, which indicated that FLC is capable to maintain necessary control to keep target within depth of focus even for very difficult cases. The optical FLC system presented in this paper does not use stereovision tracking and des not utilize disparity and occlusion capabilities.

All referenced here tracking systems have been designed based on mono-vision camera (single camera). The only tracking system that has been based on stereovision (two cameras) is proposed in [67]. The proposed system combines a stereophonic vision with the 3D model that allows accurate pose estimation in the presence of partial occlusions by non rigid objects.

Using a second camera in the system improves stability of tracking and simplifies the algorithm. This system, however, does not embed fuzzy logic and is not equipped with uncertain reasoning.

3.3.5 Diagnostic Systems

Diagnostic systems are largely divided into medical systems and fault diagnostic systems. Some medical systems that can be classified as detection and identification systems are covered in 3.3.2, while others are typical diagnostic systems for pattern classification. In [68] and [69], new diagnostic reasoning approaches based on fuzzy logic are presented. The authors propose generalized fuzzy algorithms that apply alternative fuzzy inference approach to decision making and diagnosis. In [68] the proposed diagnosis algorithm is generalized especially for medical systems. Unlike other medical systems [42] [43], this algorithm can be applied to any field by modifying its membership functions and fuzzy inferencing with expert knowledge in this field. That gives generalization of fuzzy approach for diagnostic systems.

Unlike [68], an application specific diagnostic approach is presented in [70]. This system is identifying lung cancer by analyzing a signal acquired through an array of MOS sensors. The system uses complicated fuzzy reasoning algorithm for pattern classification which requires large computational power and is not suited for real-time processing applications. However, the authors report a very high accuracy in classification, exceeding 90% of cases. A similar system, but for breast cancer detection is presented in [71]. This system is different from [70] by using an image classification for cancer detection.

In [72] [73], the fuzzy set framework has been utilized in several different approaches for modeling the diagnostic process of diabetics and atherosclerosis diseases. In both, the capability of fuzzy logic to express professional knowledge in linguistic ways is utilized, allowing a system to be described by simple, human-friendly rules. In [74], a similar approach is applied to aphasia diagnosis, with addition of hierarchical fuzzy rule-based structure that considers the effect of different features of aphasia by statistical analysis in its construction. The addition of statistical analysis in this approach gives an advantage in terms of accuracy. The simulation and test results of proposed systems show that fuzzy algorithms can be used as an effective classifier for medical problems.

Another important field, where fuzzy logic has been applied is fault diagnosis. In [75], a general fault diagnostic method based on fuzzy logic and evidence theory is presented. This method is designed for multi-sensor input systems where basic reliability distribution of evidence theory is obtained by using membership functions, which significantly improves the accuracy of diagnosis.

A similar approach has been applied to automated diagnostics of analog systems [76]. This method is aimed to detect and localize multiple system faults in noisy conditions. The fuzzy decision making approach used in the system is based on three trapezoidal and thee triangular membership functions, which simplifies computations and allows system to be implemented in a real-time environment, comparing to approaches presented in [70] [71], where a complexity of membership functions does not allow real time implementation.

In [77], the authors present a similar fuzzy logic system for fault diagnostic of lead-zinc smelting process. The system is able to effectively diagnose multiple system faults by using fuzzy abductive inferencing. In [78], a very similar system for fault diagnosis of a wheel loader is presented. The advantage of these systems is in utilization of neural networks for fault symptom extraction, which gives diagnostic process ability of adaptation.

Much less complicated system comparing to [77], however capable of accurately diagnose faults in a scooter engine platform is presented in [79]. The system uses tachometer signals as well as microphone signals to produce fuzzy reasoning diagnostics. The triangular and π membership functions are used in the system, and an adaptive reasoning is provided by knowledge data base that influences the inference rules. The system does not utilize neural networks, yet provides an adaptive diagnosis and due to simplicity of membership functions is suitable for real time implementation. The experimental results indicated that proposed system is effective for accurate fault detection under various operating conditions.

The fuzzy logic has found its way into many diagnosis applications, and even as important and critical as nuclear plant accident diagnostics. In [80], a complex nuclear plant accident diagnosis method incorporating fuzzy logic approach is presented. The fuzzy-logic inference technique is used to calculate the operator's confidence, or degree of belief, that a given plant event has occurred based on the observed symptoms. In this system the operator diagnosis and fuzzy algorithm diagnosis cooperate in order to provide as accurate diagnosis as possible.

3.3.6 Predicting Systems

Predicting systems are systems that establish probability of occurrence of specific event based on analysis of events of the past as well as current conditions. The fuzzy logic reasoning found large utilization in this area of research. In [81], a dynamic fuzzy preemption algorithm for soft real-time operating systems is presented. In this work, a fuzzy logic prediction system to predict task switching in the real-time operating system is used to improve scheduling algorithm of system kernel. The correct prediction makes number of task switches smaller and improves real-time system performance. The fuzzy logic system was implemented on a simulation model and produced results surpassing a traditional slack first scheduling algorithm. The main challenge of this algorithm is achieving real-time constraints when implemented in a real kernel for real-time operating system. Due to complexity of fuzzy logic calculations, to achieve real-time requirements, the hardware acceleration might be necessary.

In [82], the authors present a prediction system based on fuzzy logic approach for prediction of engine emission. The system is designed to help the manufacturers of the automobile engines to predict pollution of their engines in absence of the pollutant measurement devices. The systems is based on analysis of peak pressure, load, indicated mean effective pressure, ignition delay, and combustion duration to produce prediction of omission of harmful to the health oxides of nitrogen (NOx). The authors reported excellent prediction results of the system. The system uses large number of inputs to perform fuzzy reasoning and is very complex computational wise. The practical implementation is limited to the offline applications only.

A similar prediction system for prediction of compressive strength of concrete is presented in [83]. The system uses 9 input parameters identifying concrete components to predict 7, 28 and 90 days compressive strength of concrete. The prediction fuzzy reasoning, in this system, is equipped with a three-layer neural network to produce learning capability. The fuzzy logic inference is based on trapezoidal membership functions that are simple in calculation and achieve good prediction results for given application. The neural network in this system requires training process that has to be performed offline of operation, as well as the large database of samples has to be applied to the training. However the results of simulation model implemented in MATLAB has shown very good prediction. The advantage of this system is adaptivity based on past experience.

Another predicting system using neuro-fuzzy approach is presented in [84]. The system is designed to predict human operator behavior while operating complex control systems, as driving a car, piloting an airplane etc. The human behavior depends on many factors that can not be mathematically modeled. Therefore, fuzzy logic is a best method to model this behavior, and the adaptive-network-based fuzzy inference system can produce learning experience. The system is based on 32-rules fuzzy inferencing and involves a complex computational overhead. Therefore, the real time implementation of the systems is difficult. However, the simulation of the proposed system gives very food results, given the uncertainty and variability of human operator decision making.

Unlike [84], a real-time prediction system for risk assessment is presented in [85]. The system involves a fuzzy approach to access a set of important plant specific inputs to produce prediction of frequencies of occurrence of abnormal events, failure probabilities of safety systems involving equipment and human actions. The fuzzy membership functions are simple and assigned to various critical zones of the controlled plant. The system is reported to produce a good estimation of human and equipment risk probabilities, and is capable of a real-time prediction.

3.3.7 Fuzzy Systems Design Techniques

Fuzzy systems are much different from conventional systems, and therefore the development process caries many differences that are specific to fuzzy logic. Good knowledge of a control domain characteristics decreases development time and post development tuning of a fuzzy system. Reconfigurability can also benefit to the post development tuning. The following steps summarize general recommendations for fuzzy system development process [17] [18] [23] [26]:

1) Describe fuzzy system as a set of measurable attributes. Sometimes there is no need to prepare precise measurements, since fuzzy systems can deal with uncertainty.

2) Consider each attribute as a linguistic variable. Build a reasonable set of linguistic terms that describe those variables. Do not try to select a membership function at this point.
3) Repeat steps 1 and 2 until desired input selection for linguistic terms is achieved. For control system, the controllable attributes for the system must be achieved, in other situation the human vocabulary may play role of controllable attributes.
4) Build the membership functions for the linguistic terms. In the beginning, when no knowledge of system behavior is known, the use of simple triangular functions is recommended. Later on, more complex functions can be used to achieve better results.
5) Gain knowledge of the relationship between the attributes of the system and the desired behavior such as input to output response.
6) Design fuzzy rule matrix reasoning and inference rules.
7) Implement fuzzy system either on simulation environment or on a real-world environment (preferably reconfigurable such as FPGA or Software)
8) Verify system behavior against desired.
9) If system behavior is different from the expected, analyze what step of the listed above items would have an influence on the system behavior and repeat starting from that step (post development fuzzy tuning).
10) Validate final system performance and conclude the design.

Many steps of fuzzy system design could be facilitated with Matlab/Simulink fuzzy logic environment that provides vast verity of tools for fuzzy system design and simulation [21].

3.3.8 Fuzzy Systems Implementation Techniques

Fuzzy logic systems can be implemented in many different ways. The actual implementation of a fuzzy logic system is much dependant on the requirements imposed on the system performance. The 3 main categories of system implementation can be outlined:

1) Software Implementation
2) Hardware Implementation
3) Hybrid (Hardware-Software) Implementation

In *Software Implementation*, a fuzzy logic algorithm is implemented as a program or a subroutine. It may run on different processors starting from simple embedded controllers up to modern RISC processors and embedded DSP processors. Software implementation is the cheapest way to implement fuzzy reasoning. It also has a high degree of reconfigurability that is highly suitable for fuzzy systems requiring tuning. Software implementation, however, may not yield high performance for a high speed real-time control applications. For these systems, a high speed feedback is needed to achieve required real-time constraints. Software timing characteristics are vastly dependant on a CPU clock, number of clocks per fuzzy reasoning interaction, and the operating system. Since typically thousands of clock cycles are needed to execute a function or a sub-routine with fuzzy reasoning (including a clock cycles spent on fetch and decode CPU instructions), the hardware implementation of fuzzy logic achieves better performance [86].

In *Hardware Implementation*, the whole fuzzy logic systems is implemented on a hardware without using a CPU and thus not spending time on fetch and decode instructions. The hardware architecture is the fastest and most suitable for the systems with hard real-time constraints. The FPGA implementation combines fast speed and reconfigurability [87], while fuzzy logic module can be designed in VHDL [88]. Pure hardware implementation, such as ASIC lacks reconfigurability and is not recommended for the development stage of fuzzy logic systems.

Hybrid (software-hardware) implementations are less common, since there is little point in dividing fuzzy logic into blocks that partially implemented on hardware and partially on software, since such a systems would be overcomplicated by passing non-crisp fuzzy values from software to hardware and vice versa. Most commonly, the software system is used to control hardware fuzzy logic modules to produce a response in crisp domain and report back to software. Another way is to combine software and hardware implementation is to design different independent fuzzy logic modules in one system, that modules with hard real-time constraints will be implemented on hardware, while modules with soft real-time constraints will have software implementation.

Regardless which way the system is chosen to be implemented, there are 3 main techniques that can be used for the implementation of fuzzy logic reasoning:

1) Run-Time Implementation
2) Look-Up Table (LUT) Implementation
3) Hybrid (Run-Time and Look-Up-Table) Implementation.

In *Run-Time* implementation technique, the system performs calculation of a fuzzy response on every change of the input signal. The input signal is going through all stages of fuzzy algorithm: fuzzification, inferencing, fuzzy rules, and defuzzification. After defuzzification, the output response is formed. The run-time implementation achieves very precise response and requires almost no memory resources, however is costly computational wise.

In *Look-Up Table* implementation, all possible system responses are pre-calculated and organized in form of a look-up table. Thus, the input signal does not have to go trough any of regular fuzzy computation, and is only used to establish the corresponding fuzzy response from the table. This method is the fastest, since there are little calculations involved, usually only to transfer input signal to the corresponding address of a fuzzy response stored in the memory. However this method can be costly memory wise, since all possible input-output relations must be stored.

In *Hybrid (Run-Time and Look-Up-Table)* implementation, a combination of run-time calculations and a look-up table usage are combined together. Sometimes in a trade off situation - accuracy versus memory, the accuracy of look-up table can be increased by interpolation. In this case a look-up table is equipped with interpolation arithmetic and the response time of the system increases.

Generally, the design of fuzzy logic system is a trade off situation between a number of factors: response time, accuracy, memory, reconfigurability, and price. The design space exploration brings all these factors into one chart that explores possibilities for system implementation [89]. Appendix A elaborates on the design space exploration for fuzzy logic system.

Chapter 4

Implementation

In this chapter the implementation of experimental prototype of Reconfigurable Fuzzy Logic Controller for High-Frame Rate Stereovision Object Tracking System is presented. Since the reason of implementation is the proof of concept only, a number of simplifications were made to the main concept of operation of the system in order to simplify the implementation of the first experimental prototype. Only X-axis (horizontal) object tracking control was implemented in fuzzy logic controller and in the entire stereovision tracking system. The automatic focusing on the object is supported by fuzzy logic controller module, however was not implemented in the rest of the STS system. The following chapters describe in details the implementation of the system.

4.1 Hardware Components and Development Tools

Table 4.1 presents a list of major hardware components of the first experimental prototype.

Table 4.1 : Hardware components

Module	Component	Details
STS	Stereo Lens	High Resolution Miniature Fixed Focal Lens V-4405.6-2.HR 1/3" CCD.
STS	FPGA	Virtex2P - XC2VP2 - FF672-6
SMC	Servo-Motor Controller	AVSI-USBPIC Development Board with PIC Microchip® PIC 18LF442.
SMC	Servo-Motor	HiTec HS-300 Servo Motor

The list of the development tools that were used for implementation of the system prototype is presented in Table 4.2.

Table 4.2 : Implementation Tools

Tool	Details
MATLAB	5.1.0.421 Simulation environment
Xilinx	Xilinx ISE 9.2.03i (synthesis)
ModelSim	ModelSim SE 9.2g (behavioral and post synthesis simulation)

4.2 Fuzzy Logic Control Implementation

This chapter describes the design of fuzzy logic rotation control. It includes selection of membership functions, fuzzy rule matrix, selection of fuzzy parameter, and creation of fuzzy control space.

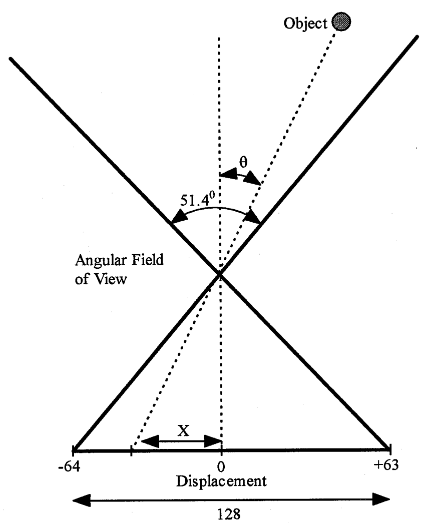


Figure 4.1 : Mapping Angular Field of View onto Displacement

4.2.1 Input Membership Functions

With the disparity image resolution 380 x 640, the horizontal displacement that is calculated by the disparity processor can take values [-320:319]. The displacement of 0 indicates that the center of the object is in the middle of the system. The displacement range is scaled down to 7 bits signed value with range [-64 to 63]. The scaling factor is 5. The quantization constant is 640/5 = 128. The mapping of the angular view of the camera on the displacement value is illustrated in Figure 4.1. The angular view of the camera according to specification of the lens V-4405.6-2.0-HR is 51.4°. With quantization level of displacement 128, the minimum angle resolution is 0.4°. The derivative $\Delta X/\Delta t$ is a 4 bit signed value with range [-8: 7].

Table 4.3 gives description of fuzzy sets assigned to the system inputs. Three fuzzy sets have been assigned to describe object displacement from the center of the view, and the direction of the object's movements.

Table 4.3 : Fuzzy sets for the input signals

Input Parameter	Fuzzy Set	Description
Displacement X	L – left	The object is located to the left of the center
	M – middle	The object is in the middle of the view
	R – right	The object is located to the right of the center
Derivative $\Delta X/\Delta t$	L – left	The object is moving left relative to the center
	M – middle	The object is not moving relative to the center
	R – right	The object is moving right relative to the center

The conventional triangular membership functions have been selected for each fuzzy set of the input signals of the fuzzy logic controller. Figure 4.2 illustrates the selection. Generally, any type of membership function best describing physical characteristics of the system can be used. Fuzzy logic membership functions give representation of the degree of membership to every input value of displacement and the derivative of the displacement. For instance, for the displacement value -32, fuzzy membership values: L [-32] = 0.5, M [-32] = 0.5, R [-32] = 0.

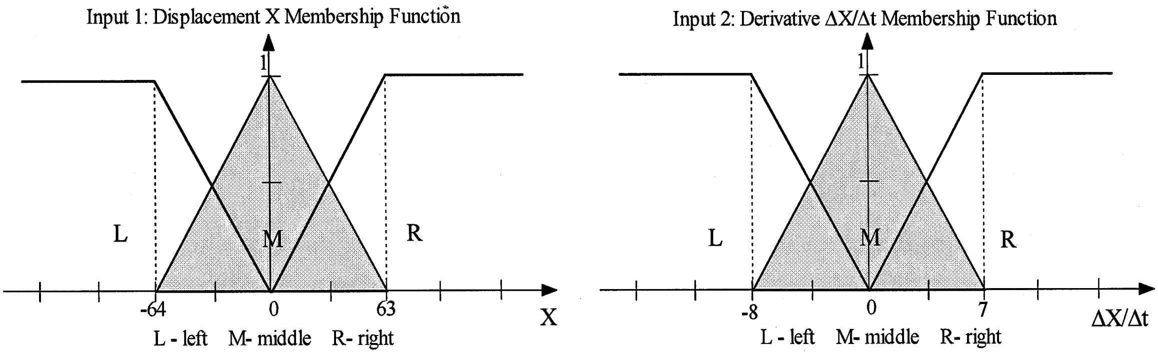


Figure 4.2 : Input Membership Functions

4.2.2 Output Membership Functions

Output membership functions define the defuzzification range of crisp values that necessary to control servo-motor rotation. The crisp output is transferred to the servo-motor controller to perform the rotation. The output value represents the direction of the rotation and also how many basic angle steps are to be taken. The basic angle step (or resolution angle) is the minimum angle that the servo-motor can be turned when controlled by servo-motor controller. It can be referred to as a resolution of the servo-motor controller turn. The output of fuzzy logic controller is a signed value, the sign represents the direction of the turn, and value represents the number of basic angle steps to be taken.

For basic angle 1.6° and angular view of the camera 51.4°, the output range is [-16:16]. 0 – means that no turn is necessary, 1 – move 1.6° to the right, 4 - move 6.4° to the right, -14 – move 22.4 ° to the left.

Table 4.4 describes fuzzy sets assigned to the system output. Each set represents an action that must be taken to control the system. Three fuzzy sets have been selected to describe all possible actions to control the rotation.

Table 4.4 : Fuzzy sets for the output signal

Output Parameter	Fuzzy Set	Description
Rotation X	Left	Turn the servo-motor to the left
	DN	Do Nothing
	Right	Turn the servo-motor to the right

The conventional triangular membership functions have been selected for each fuzzy set of the output signal of the fuzzy logic controller. Figure 4.3 illustrates the selection. Generally, any type of membership function best describing physical characteristics of the system can be used. The physical characteristics of servo-motor behavior can be reflected by the shapes of the membership functions. Output membership functions are used in the defuzzification process to form the crisp output value for the servo-motor controller.

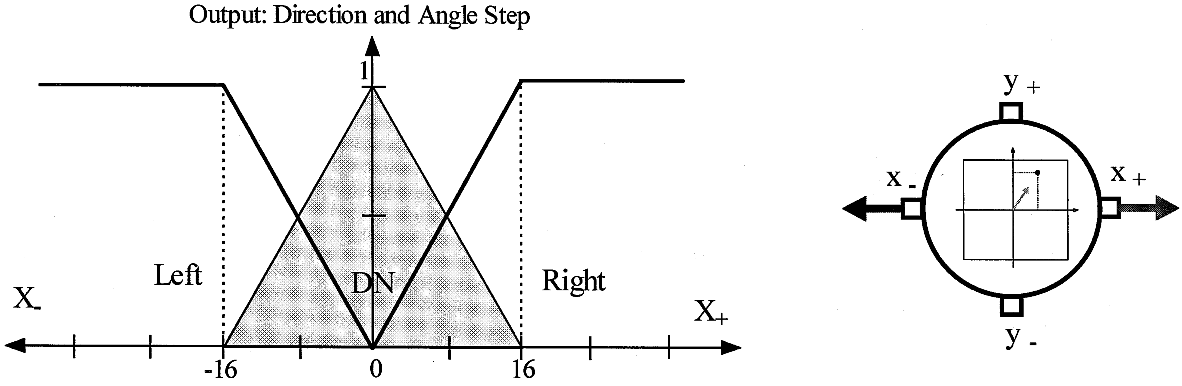


Figure 4.3 : Output Membership Functions

4.2.3 Fuzzy Rule Matrix

The fuzzy matrix for the rotation control is presented in Figure 4.4.

X \ ΔX/Δt	L (Left)	M (Middle)	R (Right)
L (Left)	**Left**	**Left**	**Right**
M (Middle)	**Left**	**DN**	**Right**
R (Right)	**Left**	**Right**	**Right**

Figure 4.4 : Fuzzy Rule Matrix for Rotation Control.

Fuzzy rules is a collection of statements that characterizes the behavior and a response of a fuzzy system. These rules can be organized in the form of "if-then" statements, or fuzzy rule matrix. The fuzzy matrix summarizes the desired control of rotation. It accounts not only for the displacement but also for the direction of the moving object. The fuzzy rule matrix can be also written in the form of "if-then" linguistic statements:

IF [*Object is on the left*] AND [*Moving left*] THEN [*Move left*]
IF [*Object is on the left*] AND [*Not moving*] THEN [*Move left*]
IF [*Object is on the left*] AND [*Moving right*] THEN [*Move right*]
IF [*Object is in the middle*] AND [*Moving left*] THEN [*Move left*]
IF [*Object is in the middle*] AND [*Not moving*] THEN [*Do nothing*]
IF [*Object is in the middle*] AND [*Moving right*] THEN [*Move right*]
IF [*Object is on the right*] AND [*Moving left*] THEN [*Move left*]
IF [*Object is on the right*] AND [*Not moving*] THEN [*Move right*]
IF [*Object is on the right*] AND [*Moving right*] THEN [*Move right*]

4.2.4 Fuzzy Logic Parameters

Table 4.5 summarizes all fuzzy logic parameters that identify fuzzy reasoning engine. The last column specifies the selection of the parameters for the rotation control.

Table 4.5 : Fuzzy Logic Parameters for Rotation Control

Fuzzy Parameter	Description	Selection
X Quantization	Level of quantization for membership functions of input X, defines the actual size of the LUT	128
$\Delta X/\Delta t$ Quantization	Level of quantization for membership functions of input $\Delta X/\Delta t$, defines the actual size of the LUT	16
Output Quantization	Level of quantization for output membership functions	32
Fuzzy Operator	One of the following: OR, AND	AND
Inferencing	One of the following: MAX, MIN, Average, RSS	Average
Defuzzification	One of the following: MAX, Aver, Centroid, RSS	Centroid

Every fuzzy parameter has a different influence on the fuzzy control space. Appendix B presents how different parameters influence the fuzzy control space. It is visible from printouts of 3-d control space visualization that the selected fuzzy logic parameters give the smoothest control line.

4.2.5 Fuzzy Control Space

Fuzzy Control Space (FCS) is a composition of all possible responses of fuzzy system on various inputs. Plotting the FCS visualizes a degree of control which fuzzy system applies on every particular set of inputs. A 3-d plot of a fuzzy look-up table is direct visualization of FCS. Figure 4.5 presents a Matlab 3-d plot of the FCS of the rotation control fuzzy look-up table.

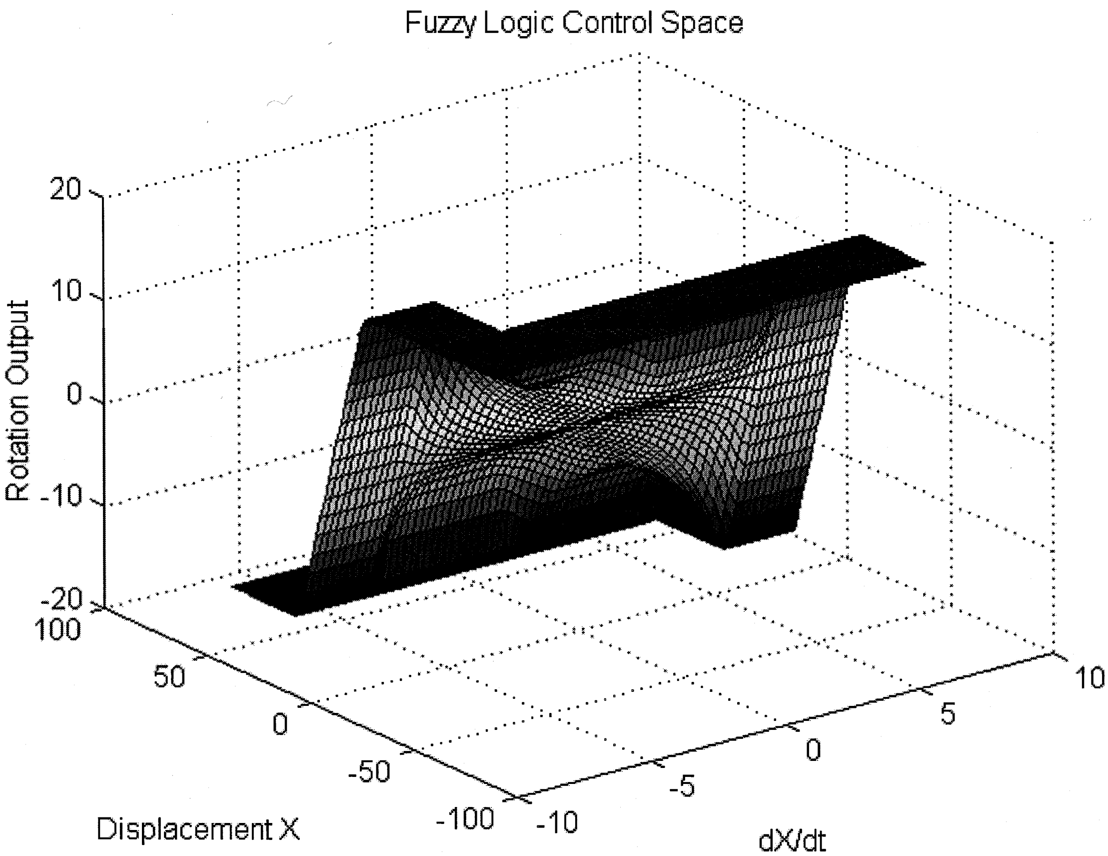


Figure 4.5 : Fuzzy Control Space of the Rotation

4.2.6 Correlation between Displacement and Speed

The correlation between the displacement of the object and the speed that servo-motor rotates the system head depends on the way we desire to control the rotation. On one hand, the smooth rotation is desired; on the other hand the fast tracking is also necessary. The following correlation provides both: smooth and slow correction for little angles of displacement while fast and aggressive for large. The main idea is to have a constant *execution time* for the motor to cover small and large angles. Thus, if the motor has to turn 1.6°, it would take same time to cover 25.7 °. The speed for each turn will be different. The following is derived relying on two constants *min_speed, max_speed* – minimum and maximum speed of the motor.

$$execution\_time = \frac{quantization}{2 \cdot \max\_speed} \tag{4.1}$$

$$speed = \frac{rotation\_steps}{execution\_time} = \frac{2 \cdot \max\_speed \cdot rotation\_steps}{quantization} \tag{4.2}$$

The speed for each element of look-up table is given by (4.3).

$$\begin{cases} speed = \dfrac{2 \cdot \max\_speed \cdot rotation\_steps}{quantization} \\ if \quad speed < \min\_speed \quad \Rightarrow \quad speed = \min\_speed \end{cases} \tag{4.3}$$

where

quantization – level of quantization for output membership functions
min_speed – minimum speed of motor
max_speed – maximum speed of motor
rotation_steps – number of rotation steps calculated by fuzzy logic

For *min_speed* = 0.083 [step/msec], *max_speed* = 0.333 [step/msec], *quantization* = 32, the execution time will be *execution time* = 48 [msec]. The speed value for each rotation value will be calculated by (4.4).

$$speed = \frac{rotation\_steps}{48[m\sec]}[steps / m\sec] \tag{4.4}$$

The speed calculations are carried out by look-up table generator Matlab program.

4.2.7 Look-Up Table Structure

Look-up table structure contains the control values for all system inputs. The control value consists of three elements: direction of the rotation, number of basic steps, and the speed. Look-up table Generator Matlab program is build to produce a look-up table for rotation control. Each control value is organized the way that can be interpreted by the servo-motor controller (See Figure 4.6).

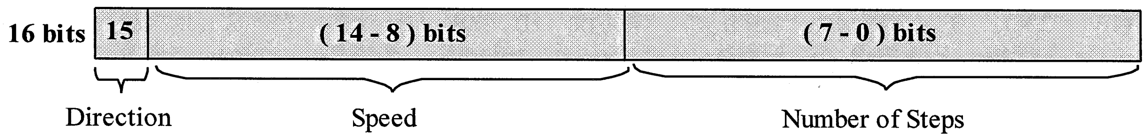


Figure 4.6 : Look-up table element structure

The description of each element is given by the Table 4.6.

Table 4.6 : Description of the element of look-up table.

Description	Bits	Min Value	Max Value	Resolution	Bits required	Bits available
Steps	0-7	0 (0°)	16 (25.7°)	1 (1.6°)	4	8
Speed	8-14	3 ms/step (0.33 step/ms)	12 ms/step (0.08step/ms)	1	7	7
Direction	15	0 (left)	1 (right)	1	1	1

Since the size of each element of LUT is 2 bytes, the total look-up table size depends on the resolution of input signals and is given by (4.5).

$$Size(bytes) = \left(2^{size\_input\_1[bits]} \cdot 2^{size\_input\_2[bits]}\right) \cdot 2[bytes] \tag{4.5}$$

Where *size_input_1,* and *size_input_2* are number of bits of X and ΔX/Δt input signals.

For *size_input_1*= 7, and *size_input_2*= 4, the size of look-up table is 4096 bytes.

4096 bytes has to be devoted to look-up table for rotation control. With Virtex II - 2KB BRAM organization, 2 BRAM blocks has to be devoted for rotation control. The BRAM organization is described in §4.6.4.

4.3 Look-Up Table Generator Matlab Program

LUT Generator Matlab program was developed in the scope of this work to facilitate creation of fuzzy look-up table. The program includes a number of routines. Each routine is organized as a separate Matlab file. Table 4.7 presents a description of LUT Generator routines.

Table 4.7 : Look-up table generator program components.

File name	Routine	Description
`fuzzy.m`	`fuzzy()`	This is a main script calling a sequence of Matlab programs for fuzzy look-up table generation. The sequence of calls is as follows: config(), mfunc_gen(), fuzzy_gen(), lut_builder(), lut_bram(). This script generates LUT and VHDL files without displaying membership function and the 3-D control space graphs.
`fuzzy_show.m`	`fuzzy_show()`	This script is similar to fuzzy(), however displays fuzzy parameters and graphical representation of membership functions and 3-D control space.
`config.m`	`config()`	This routine holds all configuration parameters for fuzzy logic controller and look-up table.
`mfunc_gen.m`	`mfunc_gen()`	This routine creates input/output fuzzy membership functions of triangular shape. Each membership function is stored as a separate text file in ...\input_files folder.
`fuzzy_gen.m`	`fuzzy_gen()`	This routine implements fuzzy logic reasoning based on input membership functions (defined as input files) and fuzzy parameters and rule matrix (defined in config). The routine simulates fuzzy reasoning to all possible input values and generates fuzzy look-up table. The look-up table text file is generated in ...\output_files folder. The routine also plots the membership functions and 3-D fuzzy control space.
`lut_builder.m`	`lut_builder()`	This routine generates final LUT structure transferring fuzzy outputs to direction, speed and angle of the motor. The values are packed in the 16 bit format and prepared for the implementation in the

		VHDL for BRAM initialization form. The input file to this routine is the output file from fuzzy_gen(). The resulting files are stored in ...\output_files folder.
lut_bram.m	lut_bram()	This routine generates VHDL project file containing BRAM initialization of fuzzy look-up table values. The input file to this routine is the output file of lut_builder(). The output files are copied to ...\output_vhd folder.

The configuration file of LUT Generator contains two sections: a fuzzy logic parameters section and an implementation constants section. The fuzzy logic parameters section initializes all fuzzy logic parameters described in §4.2.4. The implementation constant section contains servo-motor constants, communication constants (format of servo-motor controller word), and FPGA constants (BRAM constants).

```
% ====================================================================
%                    FUZZY LOGIC PARAMETERS INITIALIZATION
% ====================================================================
% GLOBAL QUANTIZATION
  q_in1 = 128;  % Input 1
  q_in2 = 16;   % Input 2
  q_out = 32;   % Output

% OUTPUT CONTROL RANGE
  low_out  = -16;
  mid_out  = 0;
  high_out = 16;

% FUZZY OPERATOR
  operator = Operator_AND;

% FUZZY INFERENCING METHOD
  inf_method = Infer_RSS;

% FUZZY RULE MATRIX
% ----------------------------------------
rules = [ L, L,  R;
          L, DN, R;
          L, R,  R];
% ----------------------------------------
% ====================================================================
%                  IMPLEMENTATION CONSTANTS INITIALIZATION
% ====================================================================
% Servo Motor Contrants
MaxSpeed  = 3;   %[ms/step]
MinSpeed  = 12;  %[ms/step]
SpeedStep = 0.1; %[ms/step]
```

```
% Servo Motor Controller Communication Constants
step_bits      = 8;
speed_bits     = 7;
direction_bits = 1;

% parameters check
if ( (2^speed_bits - 1) < (MinSpeed/SpeedStep))
   disp('ERROR: Wrong parameters for Stepper Motor Constants');
end;
% FPGA Constants (BRAM)
bram_size_bytes    = 2048; %[Bytes]
segment_size_bites = 32;   %[Bytes]
```

LUT Builder input/output files are listed in Table 4.8.

Table 4.8 : LUT Generator input/output files.

File name	In/Out	Description

`func1_1.txt`
`func1_2.txt`
`func1_3.txt`
	Input	Three input membership functions of the displacement X

`func2_1.txt`
`func2_2.txt`
`func2_3.txt`
	Input	Three input membership functions of the derivative of the displacement $\Delta X/\Delta t$

`input1.txt` `input2.txt`	Input	The range of fuzzy input membership functions for X and $\Delta X/\Delta t$

`out1.txt`
`out2.txt`
`out3.txt`
	Input	Three output membership functions of the rotation control

`output_levels.txt`	Input	The range of fuzzy output membership functions for rotation control
`fuzzy_lut.txt`	Output	Generated fuzzy look-up table file
`conv_lut.txt`	Output	Generated look-up table file converted to FPGA BRAM initialization format for rotation control
`lut_bram_rotation.vhd`	Output	VHDL project file with BRAM initialization for rotation control. Rotation FLC.
`lut_bram_focus.vhd`	Output	VHDL project file with BRAM initialization for focus control. Focus FLC.[3]

The printouts of LUT Generator run for fuzzy rotation control are presented in Appendix C.

[3] The fuzzy logic look up tables for focus control was not developed as a part of this work, however the hardware implementation supporting focus fuzzy logic controller was developed and implemented.

4.4 FLC Project Structure

The FLC project structure was developed to enable an automatic build of the system. For that reason, the folder structure illustrated in Figure 4.7 was developed.

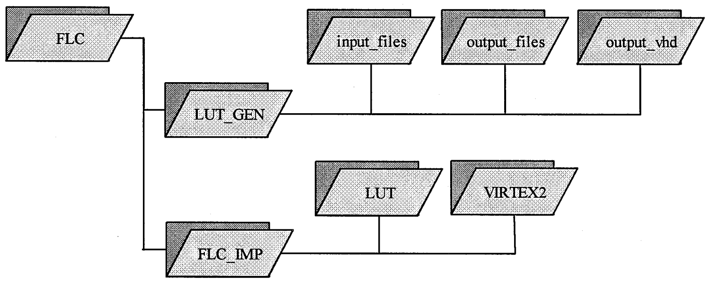


Figure 4.7 : Structure of folders for FLC project.

Each element of the folder structure is described in the Table 4.9.

Table 4.9 : Project's folders description.

Folder	Description
FLC	Main project folder. Contain `lut_gen.bat` automatic build batch file.
LUT_GEN	Contains LUT Generator project Matlab files and sub-folders
input_files	Contains input files for LUT Generator Matlab program. (Input/output membership functions text files)
output_files	Contains output files generated by LUT Generator. (Fuzzy look-up table and BRAM initialization text files)
output_vhd	Contains BRAM initialization VHDL files generated by LUT Generator. (`lut_bram_rotation.vhd` and `lut_bram_focus.vhd`)
FLC_IMP	Contains FLC VHDL project files.
LUT	Contains a copy of files generated by LUT Generator in output_vhd.
VIRTEX2	Contains Xilinx ISE project files.

The automatic build environment was developed to automate the process of building the fuzzy logic controller of proposed architecture. Using this environment, it is possible to change the FLC parameters and automatically generate VHDL project files for further synthesis with

Xilinx ISE or behavioral simulation with ModelSim. Practically the results of LUT Generator Matlab program are copied to ...\FLC\FLC_IMP\LUT folder and become a part of Xilinx FLC project. Since then, the Xilinx synthesis or simulation is possible.

The following environments have to be installed for generation, synthesis and simulation of FLC project: Matlab – for generation of FLC LUT, Xilinx ISE – for synthesis, ModelSim – for behavioral/post-routing simulation.

The batch file `lut_gen.bat` contains the script for the automatic build. The run of the script first invokes the LUT Generator Matlab program that creates the look-up table files and BRAM initialization VHDL files. After that, the script calls either Xilinx ISE or ModelSim simulation environment depending on the parameter. The batch file can be called with the parameters listed in Table 4.10.

Table 4.10 : Automatic build batch file parameters.

Call	`C:WORK\FLC\lut_gen`
Description	Generates fuzzy LUT and creates BRAM initialization VHDL files for Xilinx synthesis.
Call	`C:WORK\FLC\lut_gen -show`
Description	Generates fuzzy LUT, creates BRAM initialization VHDL files, presents Matlab graphs of input/output membership functions, invokes ModelSim simulation environment and performs behavioral simulation with test bench VHDL program for main input/outputs of the fuzzy logic controller.
Call	`C:WORK\FLC\lut_gen -show -all`
Description	Performs all actions as with `-show` parameter, with exception that all debug outputs of the fuzzy logic controller module are simulated with ModelSim. The script invokes a different instantiation of the FLC main module that includes additional debug outputs. The different test bench VHDL program is called. This option is useful for debugging purposes.

The detailed description of the Xilinx ISE VHDL project files is given in §4.6.1.

4.5 FPGA Implementation - System Architecture

The High-Frame Rate Stereovision Object Tracking or Stereovision Tracking System (STS) was implemented on STS FPGA Virtex2 platform. The fuzzy logic controller extension was implemented on the same FPGA platform as two additional modules: Disparity Processor and FLC. Appendix D presents the top level schematics of STS FPGA implementation. The top level schematic for entire STS system is presented in Figure D.1. The simplified test model version for FLC testing, where the disparity image is pre-stored in the BRAM, is presented in Figure D.2.

4.6 Fuzzy Logic Controller

Figure 4.8 represents pipelined hardware architecture of look-up table fuzzy logic controller for FPGA implementation. The proposed architecture supports BRAM implementation of look-up tables and includes the following sub-modules: *address decoder* – to decode the address of the element in the look-up table, *check value module* – to avoid jittering of the motor around set value, *SPI output interface module* – to communicate with servo-motor controller and set the appropriate control value.

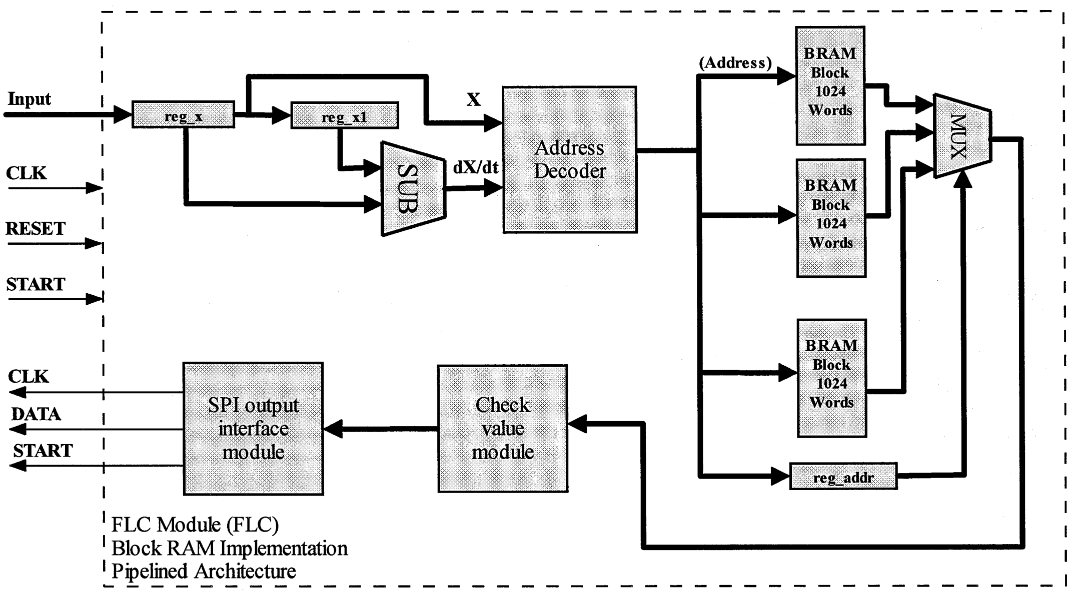


Figure 4.8 : Architecture of LUT based FLC

4.6.1 Architecture

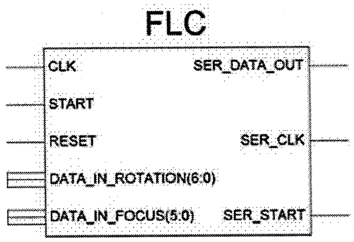


Figure 4.9 : Fuzzy logic controller hardware module.

Figure 4.9 presents fuzzy logic controller implemented as hardware module. The inputs FLC are described in Table 4.11, and are the outputs of Disparity Processor module. The outputs of FLC are connected to serial servo-motor controller SPI interface (see Table 4.12).

Table 4.11 : FLC input signals.

Signal Name	Description
DATA_IN_ROTATION(6:0)	Signed integer. Rotation input value provided by the disparity processor. (Displacement value X)
DATA_IN_FOCUS(5:0)	Signed integer. Focus input value.
CLK	62.5 MHz system clock
RESET	Asynchronous reset. Resets all internal registers and stops serial communication with servo-motor controller.
START	Start pulse initiates a single iteration of fuzzy logic controller. The values on the data lines must be set before the rising edge of START signal. The with of START signal is 1 CLK width.

Table 4.12 : FLC output signals.

Signal Name	Description
SER_CLK	Serial clock 15.2 KHz. Serial clock line is disabled when SER_START is low.
SER_DATA_OUT	Serial DATA output to the Stepper Motor Controller. The

	DATA is transmitted on the falling edge of the serial clock. The receiving must be triggered on the rising edge of the serial clock. The data is transmitted in the following sequence (32bit …0bit) The format of the data is presented in §4.6.5.
SER_START	This signal enables serial transmission to servo-motor controller. When high, the transmission is in progress.

The top-level schematic of FLC module is presented in Figure 4.10. The responses of two parallel fuzzy logic controllers for rotation and focus control are merged together in one serial system response produced by FLC_SPI_OUT module. The outputs of fuzzy logic controllers are validated by FLC_CHECK modules and START_TX signal to initiate serial transmission is produced. Table 4.13 describes sub-modules of FLC module.

Table 4.13 : FLC sub-modules.

Entity Name	Description
FLC_ROTATION	Fuzzy Logic Rotation Controller – controls rotation of the servo-motor. This module contains fuzzy look-up table located in 2 BRAM modules.
FLC_FOCUS	Fuzzy Logic Focus Controller – controls focus of stereovision cameras. This module contains fuzzy look-up table located in a single BRAM module.
FLC_CHECK	This module checks control data produced by fuzzy logic controller and initiates serial transmission to servo-motor controller. The transmission is initiated only if number of steps is equal or greater than instantiation constant < check_val>. This is designed to ignore a single step jittering around set value and avoid motor vibration. The FLC_CHECK is instantiated twice in the FLC module for FLC_ROTATION and FLC_FOCUS.
FLC_SPI_OUT	This module provides one way serial SPI communication interface to the servo-motor controller, transmitting rotation and focus control values.

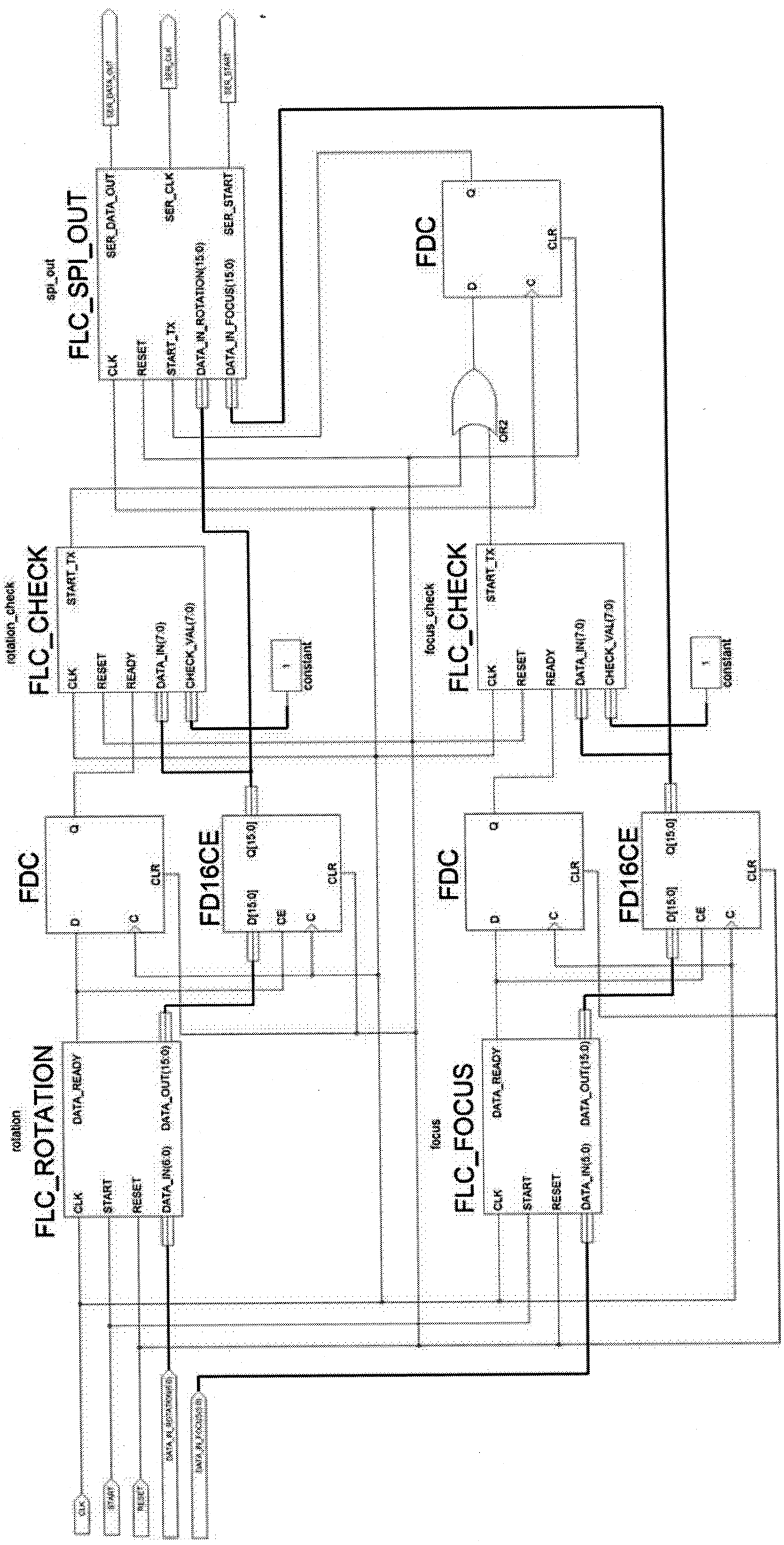


Figure 4.10 : Top-level schematic of FLC module

4.6.2 Project Files

Figure 4.11 presents the hierarchy of instantiated sub-modules in FLC module in Xilinx ISE project view window. The VHDL project files are organized the way that each module entity is located in a separate file. The content of each project file is given by the Table 4.14.

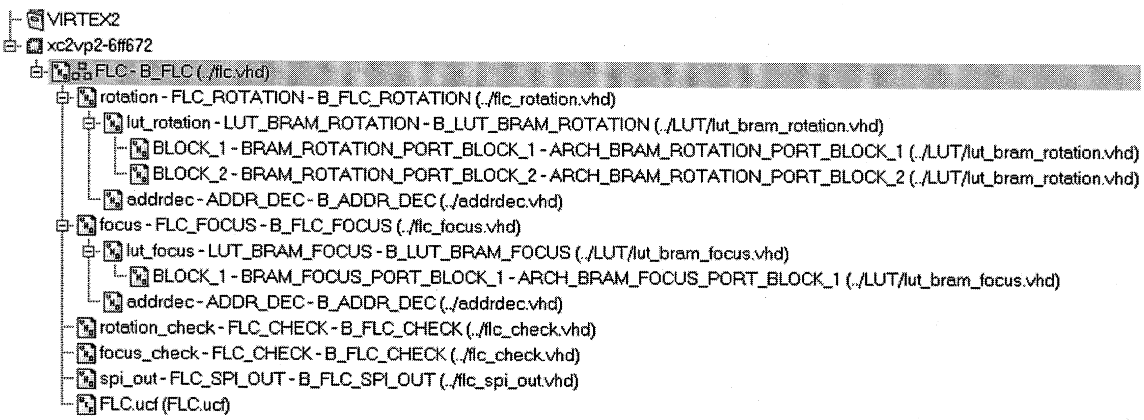


Figure 4.11 : FLC project modules hierarchy.

Table 4.14 : FLC project files.

File Name	Description
flc.vhd	Main project file. Contains FLC entity
flc_rotation.vhd	Contains FLC_ROTATION entity
flc_focus.vhd	Contains FLC_FOCUS entity
flc_check.vhd	Contains FLC_CHECK entity
addrdec.vhd	Contains ADDR_DEC entity
flc_spi_out.vhd	Contains FLC_SPI_OUT entity
LUT\flc_lut_bram_rotation.vhd	Contains LUT_BRAM_ROTATION entity - rotation BRAM look-up table
LUT\flc_lut_bram_focus.vhd	Contains LUT_BRAM_FOCUS entity - focus BRAM look-up table
flc_tb.vhd	Contains test bench - TEST_FLC entity. This entity is used in behavioral/post-routing simulation of FLC module.

4.6.3 Instantiation constants

In order to achieve high degree of reconfigurability, not only in fuzzy logic control, but also in the hardware supporting fuzzy logic controller, the whole design of FLC module and its components is parameterized. In each entity the input/output signals and crucial constants are parameterized. The top level module combines all instantiation constants. The description of each constant is given in Table 4.15.

```
-- Rotation Control
input_rotation_data_width  :integer:= 7;  -- input  data width
output_rotation_data_width :integer:= 16; -- output data width
rotation_delta_width       :integer:= 4;  -- delta width

-- Focus Control
input_focus_data_width     :integer:= 6;  -- input  data width
output_focus_data_width    :integer:= 16; -- output data width
focus_delta_width          :integer:= 4;  -- delta width

-- Output SPI check settings
output_check_width         :integer:= 8;  -- output rotation check bits
rotation_check_val         :integer:= 1;  -- output rotation check value
focus_check_val            :integer:= 1   -- output focus check value
```

Table 4.15 : FLC instantiation constants.

Instantiation constant	Description
`input_rotation_data_width`	Input rotation data width in bits (Displacement X)
`output_rotation_data_width`	Output serial rotation data width in bits
`rotation_delta_width`	Rotation delta register width in bits (dX/dt)
`input_focus_data_width`	Input focus data width in bits (Focus F)
`output_focus_data_width`	Output focus data width in bits
`focus_delta_width`	Focus delta register width in bits (dF/dt)
`output_check_width`	Output rotation check bits
`rotation_check_val`	Minimum steps of the servo-motor necessary to start the transmission for rotation control.
`focus_check_val`	Minimum steps of the servo-motor necessary to start transmission for focus control.

4.6.4 BRAM organization

Physical properties of FPGA Virtex2P 2VP2 with regards to BRAM organization are listed below [90].

Block RAM – 4 columns, 64 blocks each, 294,912 Bits, 288KBits

Block organization: 1Kx16 (2 + parity), RAM configured as a single port

Organization	Memory Depth	Data Width	Parity Width	DI/DO	DIP/DPO	ADDR	Single-Port Primitive
1Kx18	1024	16	2	(15:0)	(1,0)	(9:0)	RAMB16_S18

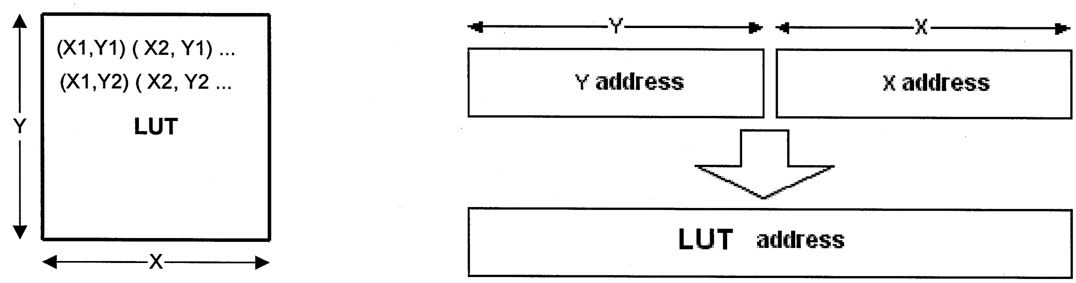


Figure 4.12 : BRAM organization and addressing.

Figure 4.12 describes the addressing principle used in ADDRESS_DECODER to address the corresponding value in the LUT or (BRAM structure). The X and Y addressing lines are concatenated together so that Y list significant bit is attached to X most significant bit. The resulting value is a LUT address value. For rotation controller the X = Displacement X (7 bits), Y= Derivative dX/dt (4 bits). Table give calculation of how many BRAM blocks has to be used depending on the number of input address bits (the total space of LUT is given by equation (4.5).

Table 4.16 : BRAM organization for rotation and focus control.

FLC Addressing	X [bits]	Y[bits]	LUT Size [Bytes]	BRAM blocks	Unused bytes
Rotation	7	4	4096	2	0
Focus	6	4	2048	1	0

4.6.5 Timing Specifications

The waveform of input signals to the FLC is presented in Figure 4.13. In order to maintain proper functionality, the input signal specifications have to be met by the Disparity Processor. The output waveform signals are presented in Figure 4.14. The output signal specifications are in accordance with servo-motor controller SPI interface requirements.

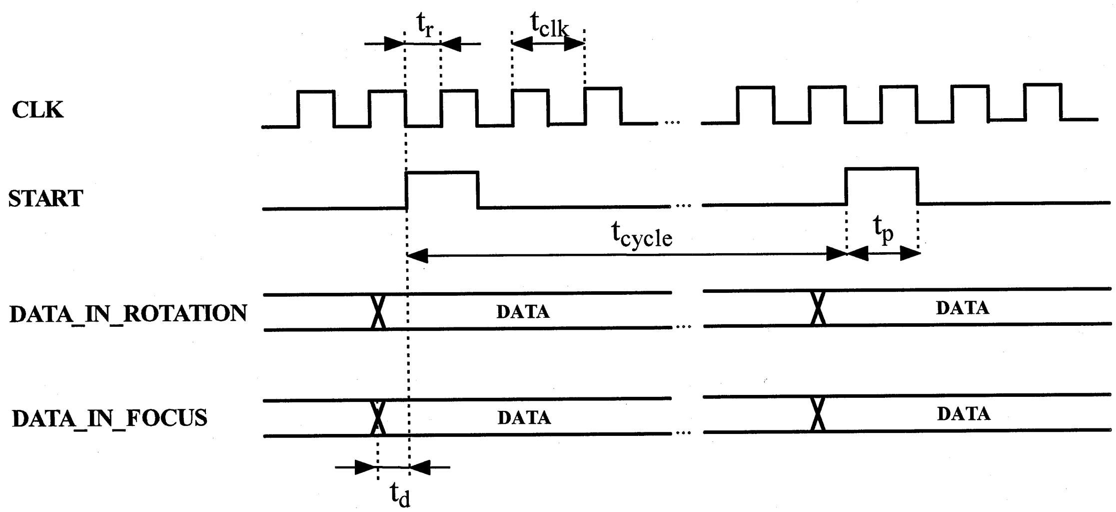


Figure 4.13 : Input signal waveform.

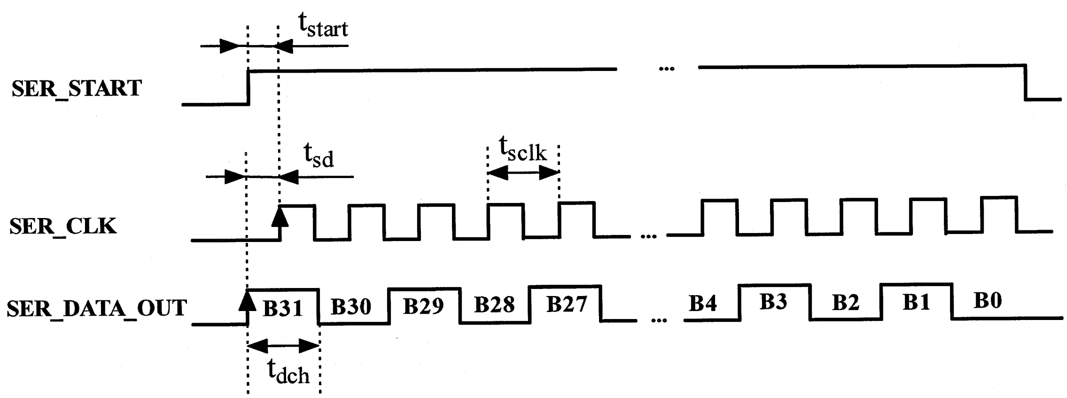


Figure 4.14 : Output signal waveform.

The main principle of FLC timing operation is as follows. The Disparity Processor module sets the data on the DAT_IN_ROTATION and DATA_IN_FOCUS lines. In t_d, the START pulse with width of one clock cycle initiates FLC data acquisition. Each time the data acquisition is initiated, the FLC prepares the value to be sent to servo-motor controller from fuzzy look-up table. The serial response is produced only if control value is different from the control value sent by the previous acquisition cycle and if control value is within certain limits defined by FLC_CHECK module. The time between each acquisition cycle or between consecutive START pulses shall not exceed t_{cycle}.

Table 4.17 summarizes timing constrains imposed on input/output signals of FLC module. All calculations are done based on assumption that system clock is 62.5 MHz and the timing constant (execution time) for FLC cycle is 48 ms (see §4.2.6).

Table 4.17 : Timing constants.

Symbol	Parameter	Value for CLK = 62.5 MHz
t_{clk}	= 1/CLK	16 ns
t_r	= $t_{clk}/2$	8 ns
t_p	= t_{clk}	16 ns
t_{cycle}	> 32x t_{dch}	> 2 ms, typical 36 - 48 ms
t_d	≥ t_r	≥ 8 ns
t_{sclk}	Serial clock = 15.2 KHz	65.5 us
t_{sd}	= $t_{sclk}/2$	32.75 us
t_{start}	= $t_{sclk}/2$	32.75 us
t_{dch}	= t_{sclk}	65.5 us

4.6.6 Serial interface to Servo-Motor Controller

The servo-motor controller module is based on AVSI-USBPIC development board with PIC18F442 microcontroller that is accessible by SPI interface. The format of the data sent to servo-motor controller through serial SPI interface is presented in Figure 4.15. The rotation control word is presented in Figure 4.16. The focus control word is presented in Figure 4.17.

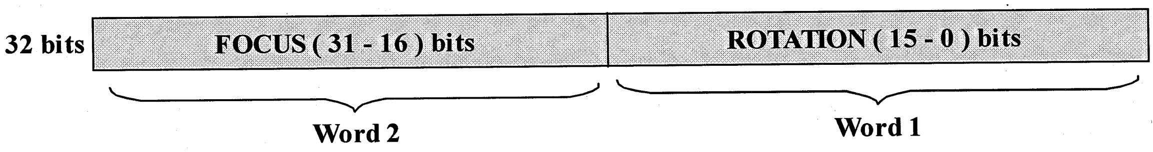


Figure 4.15 : Output data format.

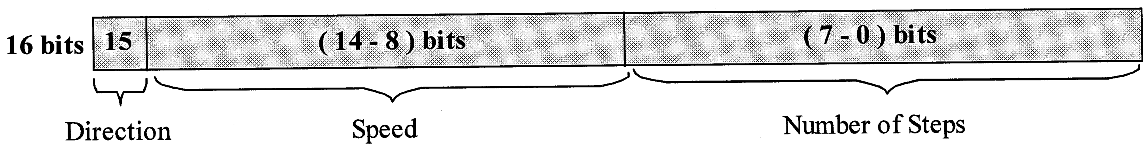


Figure 4.16 : Rotation control word.

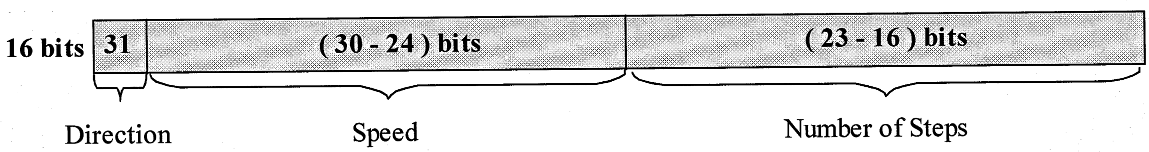


Figure 4.17 : focus control word.

4.6.7 Synthesis

Synthesis results of FLC module are presented in Appendix E. The synthesis was generated using Xilinx ISE Version 9.2.03i for the following device properties:

FPGA Type: Virtex2P
Device: XC2VP2
Package: FF672
Speed: -6

Chapter 5

Analysis of experiment

This chapter describes results of experiments that have been conducted to validate functionality of the fuzzy logic control system integrated with STS. First, the behavioral and post-routing simulation was performed to validate the functionality of the design, and then a number of static and dynamic experiments with the fuzzy logic system on the STS platform were conducted. This chapter includes the simulation results acquired software, the description of experimental set-up, the description of procedures used in each experiment, collected results of static and dynamic experiments, and the analysis of the results. The results are analyzed and conclusions are drawn. Some of the results, schematics, print-outs and tables referenced in this chapter are located in Appendix F.

5.1 Simulation Results

The results of simulation of FLC module with ModelSim (SE6.2g) are presented in Appendix F.1. The simulation includes behavioral simulation and post routing simulation with post-routing synthesis model produced by Xilinx ISE (9.2i).

The simulation test-bench module produces all necessary signals for FLC and serves as a substitution to the *disparity_processor* module and the entire STS system. The displacement values used for simulation are the same values produced by *disparity_processor* module in the dynamic experiment described in §5.4.2. The simulation model of FLC is enhanced with additional debug outputs: rotation control value and focus control value. Both values are sent through the SPI interface to servo-motor controller.

The displacement values and the corresponding look-up table rotation control values of the simulation are presented in Table 5.1. The speed ($\Delta X/\Delta t$) is calculated manually knowing current and previous displacement. The expected rotation control values are taken from look-up table presented in Appendix F.4.

Table 5.1 : Simulation results.

Displacement X (7bits)	Speed ($\Delta X/\Delta t$) (4 bits)	Expected output (hex) [LUT rotation value]	Simulation output value (hex)
-30	-8	4008	4008
-24	6	4B06	4B06
-6	7	7802	7802
0	6	7800	7800
3	3	F801	F801
40	7	B10A	B10A
35	-5	B908	B908
29	-6	C107	C107

It is visible from the simulation results that all values transmitted through SPI interface correspond the expected values drawn from rotation control look-up table.

5.2 Description of experimental set-up

The experimental set-up for fuzzy logic control system integrated with STS is illustrated in Figure 5.1. The following components are used in the experimental set-up:

1) STS Image Capture Module. This module includes a Stereo Frame Grabber Board with two high-speed CMOS sensors.
2) STS Video Processing Module. This module includes the Amirix AP1100 FPGA Development Board with Xilinx Virtex-II Pro FPGA (XC2VP2 - FF672-6) with two SRAM Banks and PCI Mezzanine card slot.
3) Xilinx JTAG Programmer
4) Logic Analyzer HP-54620C (16 channels 500Msa/s)
5) Multi-channel cable connected to Logic Analyzed

6) PC with the following software: Xilinx ISE 7.1 Synthesis tool, Xilinx Loader IMPACT – Loading utility, ChipScope Pro Analyzer 8.2 – on chip debugging tool.

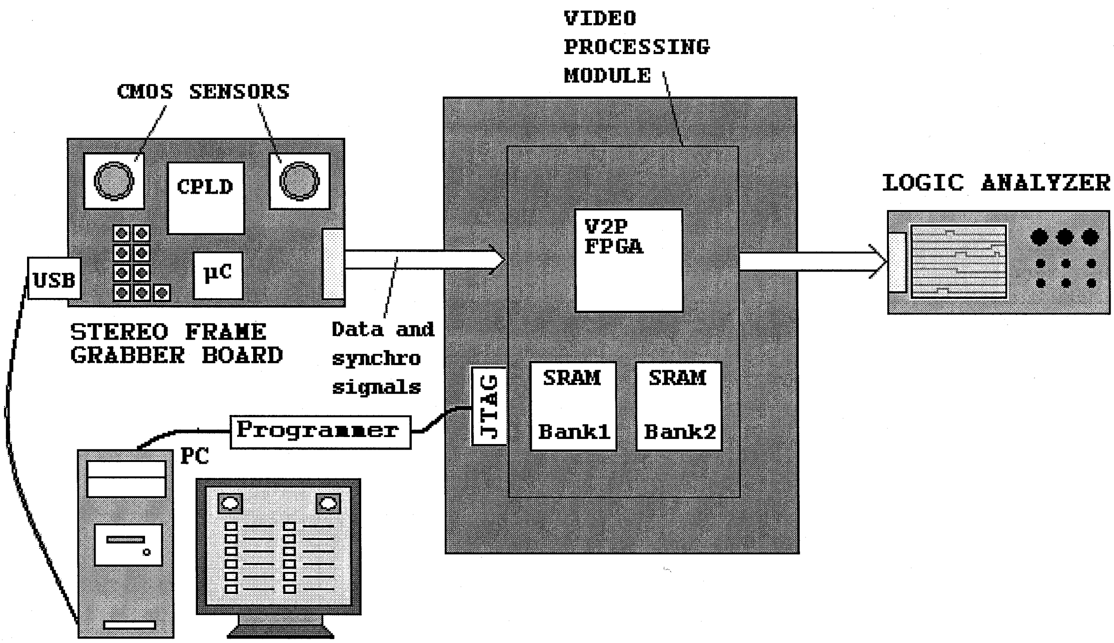


Figure 5.1 : Experimental set-up block diagram.

Xilinx software tools are used to synthesize the STS project and load it into FPGA. The loading is performed through the Xilinx JTAG Programmer connected to the JTAG port on the Video Processing Module. The Stereo Frame Grabber Board is used to provide a stereovision image for further processing in the Video Processing Module. The FPGA on the Video Processing Module, after loading the design, contains FLC module integrated with the STS modules. The experimental top-level schematic of FLC and STS modules is described in Appendix F.2. The debug port on the Video Processing Module is used to monitor the output signals from the FLC module. The port is connected to logic analyzer through multi-channel cable. The pin-out connections of debug port and logic analyzer's channels are listed in the Table 5.2. The dip-switches located on the Video Processing Module (Amirix AP1100 FPGA Development Board) are used in the experiments to switch between pre-loaded images, stored in the BRAM of the FOGA (see Appendix F.2).

Table 5.2 : Logic analyzer pin-outs.

Chan	Pin-out on debug connector	Direction	Description
CH15	NET "ROTATION<15>" LOC = "g19";	NA	NA
CH14	NET "ROTATION<14>" LOC = "h16";	NA	NA
CH13	NET "ROTATION<13>" LOC = "e19";	NA	NA
CH12	NET "ROTATION<12>" LOC = "m18";	IN	Displacement (bit 0)
CH11	NET "ROTATION<11>" LOC = "D19";	IN	Displacement (bit 7)
CH10	NET "ROTATION<10>" LOC = "L17";	IN	Displacement (bit 6)
CH09	NET "ROTATION<9>" LOC = "K20";	IN	Displacement (bit 5)
CH08	NET "ROTATION<8>" LOC = "H17";	IN	Displacement (bit 4)
CH07	NET "ROTATION<7>" LOC = "j21";	IN	Displacement (bit 3)
CH06	NET "ROTATION<6>" LOC = "F17";	IN	Displacement (bit 2)
CH05	NET "ROTATION<5>" LOC = "J20";	IN	Displacement (bit 1)
CH04	NET "ROTATION<4>" LOC = "K18";	IN	Displacement (bit 0)
CH03	NET "SDO" LOC = "C20";	OUT	SER_DATA_OUT - Serial Data Out
CH02	NET "SCLK" LOC = "F21";	OUT	SER_CLK – Serial Clock
CH01	NET "SSTART" LOC = "G18";	OUT	SER_START – Serial Start
CH00	NET "start" LOC = "E17";	IN	START – start signal

Figure 5.2 shows the STS module components and connections. Figure 5.3 and Figure 5.4 are showing the STS module on the four-wheel rotation platform attached to the servo-motor.

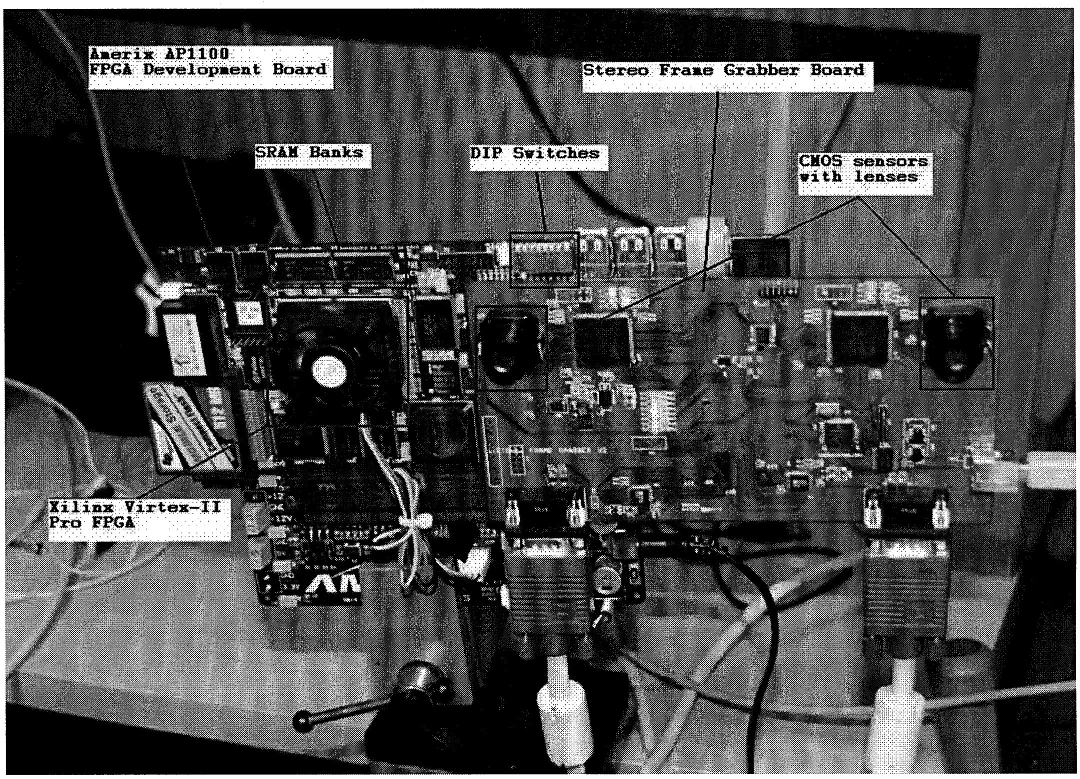


Figure 5.2 : STS Module components and connections.

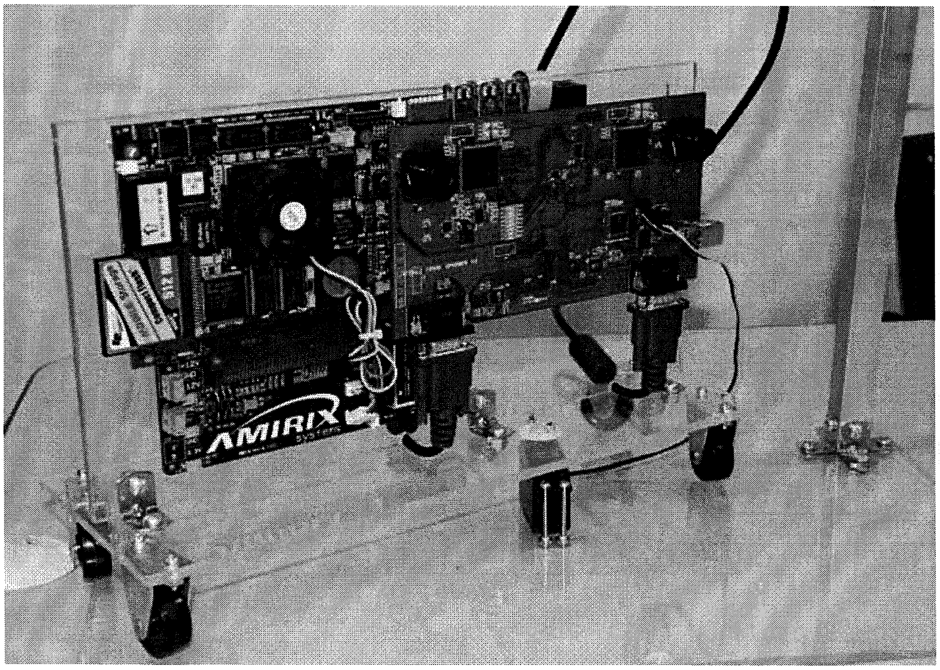


Figure 5.3 : STS Module on the rotation platform turning right.

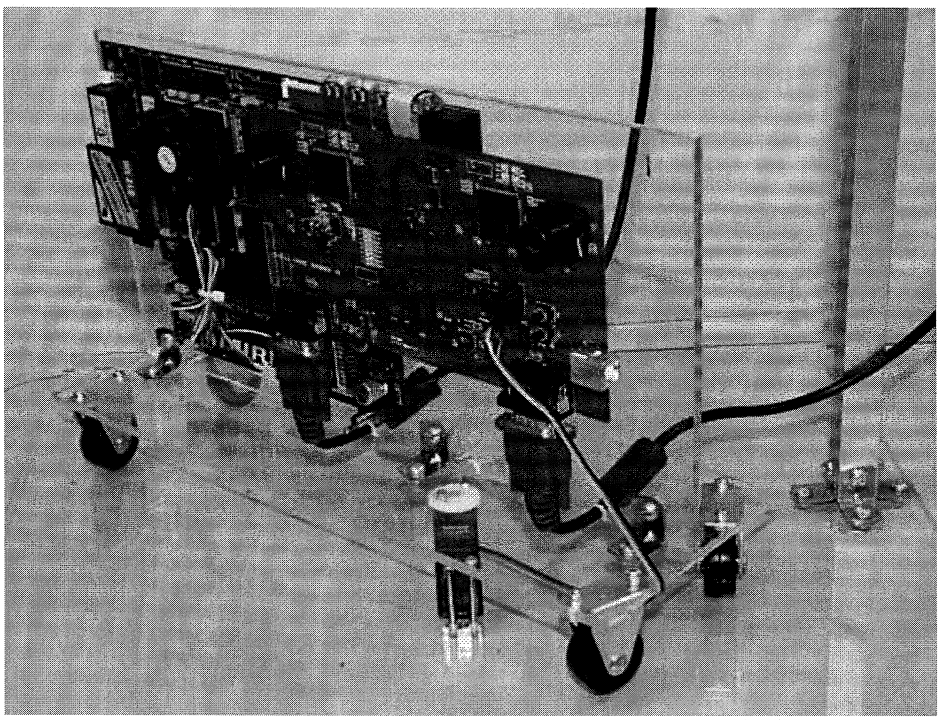


Figure 5.4 : STS Module on the rotation platform turning left.

5.3 Procedures of the experiment

The experiment is divided into two parts: static and dynamic. In static experiment, the set of static images are introduced to the system, and the response of the system is captured by logic analyzer and analyzed for correctness. In dynamic experiment, dynamically changing images representing moving object are introduced to the system with periodic intervals; the response on each image is captured and analyzed. The dynamic experiment is designed to investigate the tracking characteristics of the system when tracking the moving object.

5.3.1 Static experiment

In static experiment, three static images are pre-loaded into experimental set-up module (See Appendix 0). Each image is loaded into BRAM and organized as a separate module. The image selector transfers selected image to Disparity Processor that generates inputs for the FLC module. The image selection is done by the dip switches located on the STS platform. Each image is ideal representation of the object's disparity map. The only middle horizontal line of the image is used by the Disparity Processor for displacement calculations. The BRAM initialization (.coe) files of the images are presented in Appendix F.3.

The procedure of the static experiment is described below:

1) Load the experimental design into FPGA on the STS platform.
2) Select dip switches to load first image (left edge 172, right edge 375)
3) Capture FLC output waveforms with Logic Analyzer.
4) Select dip switches to load second image (left edge 70, right edge 103)
5) Capture FLC output waveforms with Logic Analyzer.
6) Select dip switches to load third image (left edge 552, right edge 572)
7) Capture FLC output waveforms with Logic Analyzer.
8) Analyze acquired waveforms for a) correctness of data transmitted to servo-motor controller: acquired data versus manually calculated and referenced to look-up table; b) conformance with timing constraints.

5.3.2 Dynamic experiment

In dynamic experiment, a similar approach as in static experiment is taken. The test images are pre-loaded into test module's BRAM, each organized as a separate module. The main difference from static experiment is that images are automatically selected by image selector every system cycle (START pulse), or 48ms. Eight images producing different displacement X-values are loaded to the system. The selection is done starting from image 1 and following through all images up to image 8, after that the process starts from the beginning. The selection is synchronized by START pulse produced by the Disparity Processor, so that with each START pulse, the different displacement value is loaded to the FLC. The displacements produced by pre-loaded images are listed in Table 5.3.

Table 5.3 : Displacement of dynamic images.

Image	Displacement X
Image 1	-30
Image 2	-24
Image 3	-6
Image 4	0
Image 5	3
Image 6	40
Image 7	35
Image 8	29

The procedure of the dynamic experiment is described below:

1) Load the experimental design into FPGA on the STS platform.
2) Capture FLC output waveforms with Logic Analyzer. Make sure to capture the responses on every image. Since images are rotating, the constant response of the system will be generated.
3) Analyze acquired waveforms for a) correctness of data transmitted to servo-motor controller: acquired data versus manually calculated and referenced to look-up table;

b) dynamic image motion tracking characteristics; c) conformance with timing constraints.

5.4 Obtained results

This chapter presents a summary and an analysis of acquired results. Each result corresponds to a photograph presented in Appendix F.

5.4.1 Static experiment results

The results of static experiments are presented in Table 5.4. The values of ΔX/Δt are manually calculated knowing the displacement, and the expected control values are taken from fuzzy look-up table. (See Appendix F.4) The found control values are recorded from monitoring serial transmission of FLC to servo-motor controller on SPI buss. (See photograph of captured transition for every image in Appendix F.5) It is visible from static experiment results that expected control values are identical to the values captured during serial transition.

Table 5.4 : Static experiment rotation value results.

Image	Edges of the image	Displacement X (7bits)	ΔX/Δ t (4 bits)	Expected Control Value (hex)	Found Control Value (hex)
1	left: 172 right: 375	-6	-6	7802	7802
2	left: 70 right: 103	-29	-8	4207	4207
3	left: 552 right: 572	30	8	C008	C008

5.4.2 Dynamic experiment results

The results of dynamic experiment are presented in Table 5.5. The values of ΔX/Δt are manually calculated knowing current and previous displacements. The expected control values are taken from fuzzy look-up table presented in Appendix F.4. The found control values are recorded from monitoring serial transmission of FLC to servo-motor controller on SPI buss. (See photograph of captured transition for every image in Appendix F.6) It is visible from static experiment results that expected control values are identical to the values captured during serial transition.

Table 5.5 : Dynamic experiment rotation value results.

Image	Displacement X (7bits)	ΔX/Δ t (4 bits)	Expected Control Value (hex)	Found Control Value (hex)
1	-30	-8	4008	4008
2	-24	6	4B06	4B06
3	-6	7	7802	7802
4	0	6	7800	7800
5	3	3	F801	F801
6	40	7	B10A	B10A
7	35	-5	B908	B908
8	29	-6	C107	C107

5.5 Analysis of experimental results

This chapter presents an analysis of acquired static and dynamic results. The following notations and are used in this chapter:

X — Displacement – 7 bit signed value generated by disparity processor module.

ΔX/Δt — The derivative of the displacement (speed) - 4bit sighed value calculated by FLC module.

Control Value — Control value generated by FLC and sent through SPI interface to servo-motor controller. This value is taken from fuzzy LUT of FLC module.

θ — The angle of the object relative to the center of the camera. (See §4.2). With angular view of 51.4°, the X range of [-64 to 63], the angle is calculated from the displacement by (5.1).

$$\begin{cases} x \le 0: & \theta = \dfrac{25.7 \cdot X}{64} \\ x > 0: & \theta = \dfrac{25.7 \cdot X}{63} \end{cases} \tag{5.1}$$

Dir — Direction of the rotation of servo-motor (part of control value see §4.2.7)

Steps — Number of basic steps of servo-motor controller (part of control value see §4.2.7) Each step is equivalent to 1.6° of rotation.

Speed — Speed value in [ms/step] of servo-motor controller (part of control value see §4.2.7)

φ — Rotation angle for servo-motor, calculated by (5.2)

$$\varphi = steps \cdot 1.6° \tag{5.2}$$

Static Error — Static error of the experiment represents a difference between the actual angle of the object and the rotation angle sent to servo-motor controller. This error can also be referred to as an angular tracking error of the static object. The error is calculated by (5.3)

$$StaticError = \theta - \varphi \tag{5.3}$$

Dynamic Error — Dynamic error is an angular error that represents a difference between the actual angle of the object and the angle achieved by servo-motor during one execution cycle. For execution cycle of 48ms and basic step angle of 1.6°, the dynamic error is given by (5.4).

$$DynamicError = \left(steps - \frac{48}{speed} \right) \cdot 1.6° \tag{5.4}$$

The dynamic error makes sense only if positive, i.e *steps* > 48/*speed*, where *steps* represents the number of basic angular steps required to aim on the object, the 48/*speed* represents the number of steps that servo-motor can achieve in 48msec. If *steps* > 48/*speed,* the motor will not be able to achieve required number of steps within 48ms and the dynamic error value makes sense, if *steps* < 48/*speed* the motor will stop after required number of steps before 48ms elapses and the dynamic error will be 0.

5.5.1 Static experiment

The analysis of static experimental results is presented in this chapter. The angular error calculations for static experiment are presented in Table 5.6. The angle of the object θ is calculated from the displacement by (5.1), the direction, the number of steps and the speed are derived from the control value by bit interpretation given in §4.2.7. The rotation angle φ is calculated by (5.2), and the static system error for each image is calculated by (5.3). The static error for first image is -0.8°, for the second is +0.4°, and for the third is -0.6°. It is visible that static errors are less than basic rotation angle of servo-motor (1.6°.).

Table 5.6 : Analysis of static errors.

Image	Displacement X	θ	control value (hex)	Dir	Steps	Speed [ms/step]	φ	Static Error (θ-φ)
1	-6	2.4°	7802	left	2	12.0	3.2°	-0.8°
2	-29	11.6°	4207	left	7	6.6	11.2°	0.4°
3	30	12.2°	C008	right	8	6.4	12.8°	-0.6°

Table 5.7 presents timing analysis of captured FLC input/output signals versus timing specifications defined in §4.6.5. It is visible that all timing specifications are met. (See Appendix F.5) The START pulse appears only after the DATA_IN values are set, the SER_START line goes high before first bit is transmitted, and stays high for the whole period of transmission, the SER_CLK is active only when SER_START is high, and the data on SER_DATA_OUT is transmitted on the falling edge of the clock (the receiving is done on the rising edge).

Table 5.7 : Timing results of SPI output.

Time constant	Expected time	Measured time
t_{cycle}	> 2 ms, typical 36 - 48 ms	48 ms
t_{sclk}	65.5 us	65.5 us
t_{dch}	65.5 us	65.5 us

5.5.2 Dynamic experiment

The analysis of dynamic experimental results is presented in this chapter. The angular error calculation results for dynamic experiment are presented in Table 5.8. The static errors are calculated using (5.3), and the dynamic errors are calculated using (5.4). The total error is equal to the sum of static and dynamic errors. The total error can also be interpreted as follows: if the object angle is θ, and the total error is *error*, then after 48ms the servo-motor will rotate θ + *error* angle. Thus for image 1, the object angle is 12.0°, so after 48ms the servo-motor will rotate 12.3° having total error of -0.3°. Also, for image 1, the static error is negative - meaning that the servo-motor set value is going to overshoot an object, but the dynamic error is positive - meaning that in 48ms of iteration cycle, the motor will not be able to complete the rotation and will undershoot the set value, thus the total error will be smaller.

Table 5.8 : Analysis of static and dynamic errors.

#	X	θ	Output (hex)	Dir	Steps	Speed	φ	Static Error	Dynamic Error	Total Error
1	-30	12.0°	4008	left	8	6.4	12.8°	-0.8°	0.5°	-0.3°
2	-24	9.6°	4B06	left	6	7.8	9.6°	0°	0°	0°
3	-6	2.4°	7802	left	2	12	3.2°	-0.8°	0°	-0.8°
4	0	0°	7800	NA	0	NA	0°	0°	0°	0°
5	3	1.2°	F801	right	1	12	1.6°	-0.4°	0°	0.4°
6	40	16.3°	B10A	right	10	4.9	16°	0.3°	0.2°	0.5°
7	35	14.0°	B908	right	8	5.7	12.8°	1.2°	0°	1.2°
8	29	11.8°	C107	right	7	6.5	11.2°	0.6°	0°	0.6°

It is visible from the Table 5.8 that angular errors for all images are less than the basic rotation angle of servo-motor that is 1.6°. The biggest error of the experiment is the error of 1.2° for image N7. Since all errors are less than the basic rotation angle of servo-motor, they can be considered negligible. The experimental results show excellent performance of the FLC while tracking a highly mobile object presented in images 1 to 8.

The dynamics of the system receiving a constant flow of experimental images can be summarized into the following statements:

1) Object located 12.0° left of the center.
2) After camera rotation of 12.5°, the object was located 9.6° left of the center. The object had moved left from last location with angular speed of 9.6°/48ms (200°/sec).

3) After camera rotation of 9.6°, the object was located 2.4°left of the center. The object had moved left from last location with angular speed of 2.4°/48ms (50°/sec).
4) After camera rotation of 3.2°, the object was located in the center. The object did not move since last iteration. The object is in the center.
5) Camera did not move, the object was located 1.2° right of the center. The object had moved from right last location with angular speed of 1.2°/48ms (25°/sec).
6) After camera rotation of 1.6°, the object was located 16.3° right of the center. The object had moved right from last location with angular speed of 16.3°/48ms (339°/sec).
7) After camera rotation of 15.8°, the object was located 14.0° right of the center. The object had moved right from last location with angular speed of 14.0°/48ms (291°/sec).
8) After camera rotation of 12.8°, the object was located 11.8° right of the center. The object had moved right from last location with angular speed of 11.8°/48ms (245°/sec).

In all these cases the FLC is able to provide new rotation angle and keep the object within a view of the camera. The maximum angular speed of the object in the experimental images was 339°/sec, depending on a distance from the object this is extremely high speed (for object located 15 m away from the camera, the speed can be 62.5 m/s) Since tracking is a continues process and is achieved by many iterations, if object stops even for one iteration cycle, the system will be able to bring the center of the object in the center of the camera view with an angular error smaller than a basic rotation angle of servo-motor.

5.6 Conclusions

Static and dynamic experiments have shown that the system is able to provide correct values for servo-motor controller for object tracking. The values are sent through SPI interface satisfying all timing specifications. The static experiment has shown that system is able to set rotation parameters for servo-motor controller to aim on the object with an angular error smaller than a basic rotation angle of servo-motor. The dynamic experiment has shown that system was able to track the object moving with constantly changing speed with total (static and dynamic) angular error less than a basic rotation angle of the system. In the dynamic experiment, the images of extremely mobile object with dramatically changing speed were introduced to the system, yet system was able to set correct rotation parameters to keep the object of interest within the view of the camera.

Chapter 6

Summary

6.1 Results

In the scope of this project, all objectives listed in §1.2 were accomplished. A concept of rotation control for the STS system and parameterized model of a fuzzy logic controller were developed. Based on these models, reconfigurable hardware architecture of a fuzzy logic control system was designed and implemented on the STS FPGA platform. To facilitate reconfiguration of the system, a set of tools and programs was developed. The reconfiguration process was automated minimizing the overall building time of the system. The fuzzy logic control system was successfully integrated with the STS system and the servo-motor control system.

A series of static and dynamic experiments validating the functionality of the system were conducted. In the static experiments, a set of static images were introduced to the system, and system response was measured and analyzed. In the dynamic experiments, a set of dynamically changing images representing moving objects with constantly changing speeds was introduced to the system. The system response was captured and analyzed. Static and dynamic experiments have shown that the system was able to provide accurate control values for the servo-motor controller for tracking the static objects. The response of the system was measured and found to satisfy all timing requirements of the system. The experiments have shown that the system was able to set rotation parameters for the servo-motor controller to bring the object into the center of view with an angular error smaller than a basic rotation angle of the system. It was also noticed that a single iteration was required to bring the static object into the center of view. The dynamic experiment has shown that the system was able to track the moving objects with total (static and dynamic) angular error smaller than a basic rotation angle of the system. In the dynamic experiment, the images of a fast moving object with rapidly changing velocities were introduced to the system. The system was able to provide rotation parameters to keep the object of interest within the field of view of the camera.

6.2 Conclusions

In the scope of this project, a Reconfigurable Fuzzy Logic Control System for the High-Frame Rate Stereovision Object Tracking was developed. A set of tools and programs facilitating reconfiguration process of the system was developed. The system was implemented on the STS platform, and a number of experiments were conducted. The experiments have shown that the system was able to track mobile and static objects with an angular error smaller than the basic rotation angle of the system.

The following conclusions can be drawn from the results of this work:

- Fuzzy logic control can be successfully applied to stereovision object tracking. It can produce fast and accurate control accommodating all nonlinearities and uncertainties of the system.
- A parameterized model of fuzzy logic controller for stereovision object tracking based on LUT approach can be created.
- A reconfigurable fuzzy logic control system can be implemented on the FPGA platform as a part of STS system.
- A fast way to reconfigure fuzzy logic control system can be found.
- The resulting system can accurately track static and mobile objects with characteristics depending on system components.
- The developed system can serve as a foundation for applying fuzzy logic to stereovision object tracking in the future.

Appendix A

Design Space Exploration of Fuzzy Logic System

The design space exploration method presented in [89] is used to determine possible architectural variances of the design space of Fuzzy Logic System. It gives basic layout of different fuzzy logic implementation techniques (hardware versus software), as well as different inferencing methods (min-max, average, RSS) over the system characteristics as speed, memory and accuracy. The Architectural Configuration Graph (ACG), presented in A.2, gives visual representation of the design space implementation and the trade-off between speed, memory usage and accuracy.

A.1 Components of the Design Space Exploration

The top level Design Space Explorations components are chosen to be the main implementation techniques of FLS.

a) ***Run-Time*** implementation technique. The systems designed with this approach perform calculation of fuzzy response on every change of the input signals. Typically these systems give high accuracy and low memory usage, the trade-off is speed. The speed decreases with more complicated inferencing methods.

b) ***Look-Up Table (LUT)*** implementation technique. The systems designed with this approach have pre-calculated LUT for input-output referencing. Typically these systems give high speed and high memory usage, the trade-off is accuracy.

c) ***Hybrid*** implementation technique is a combination of LUT and Run-Time calculations in the implementation of FLS. An example of such system can be LUT implementation equipped with interpolation to increase accuracy of the calculations.

The sub-components of the implementation space are inferencing methods (see 3.2.3).

a) ***Min-Max*** inferencing. The simplest inferencing method with smallest computational overhead. This method gives good results for simple FLS.
b) ***Average*** inferencing. This method requires averaging to be implemented in the calculation of fuzzy inferencing and defuzzification modules, which increases computational overhead, but provides better results than min-max technique.
c) ***RSS*** inferencing. This method requires most computational overhead, since it requires root square calculations, however, produces best results with smooth control lines. This method is best to be implemented with LUT approach.

The components of the characteristic space are

a) ***Speed CC*** – characterizes overall speed of the system in Clock Cycles (CC)
b) ***Memory Usage-*** characterizes memory usage of the system.
c) ***Accuracy*** – characterizes the degree of precision in fuzzy logic calculations.

In the design of FLS, many system components can be defined by system requirements. For instance, accuracy can be a requirement to achieve certain level of control. Similarly, speed, can be a requirement for high speed systems, and memory usage can be limited by selected hardware. In general, these three parameters define the design characteristics of the FLS, and are in trade-off relations with each other. The Design Space Exploration gives opportunity to select the implementation method that best suites the requirements of the system. These trade-off relations can be graphically represented by Architectural Configuration Graph (ACG).

A.2 Architectural Configuration Graph (ACG)

The Architectural Configuration Graph of Fuzzy Logic System is presented in Figure A.1.

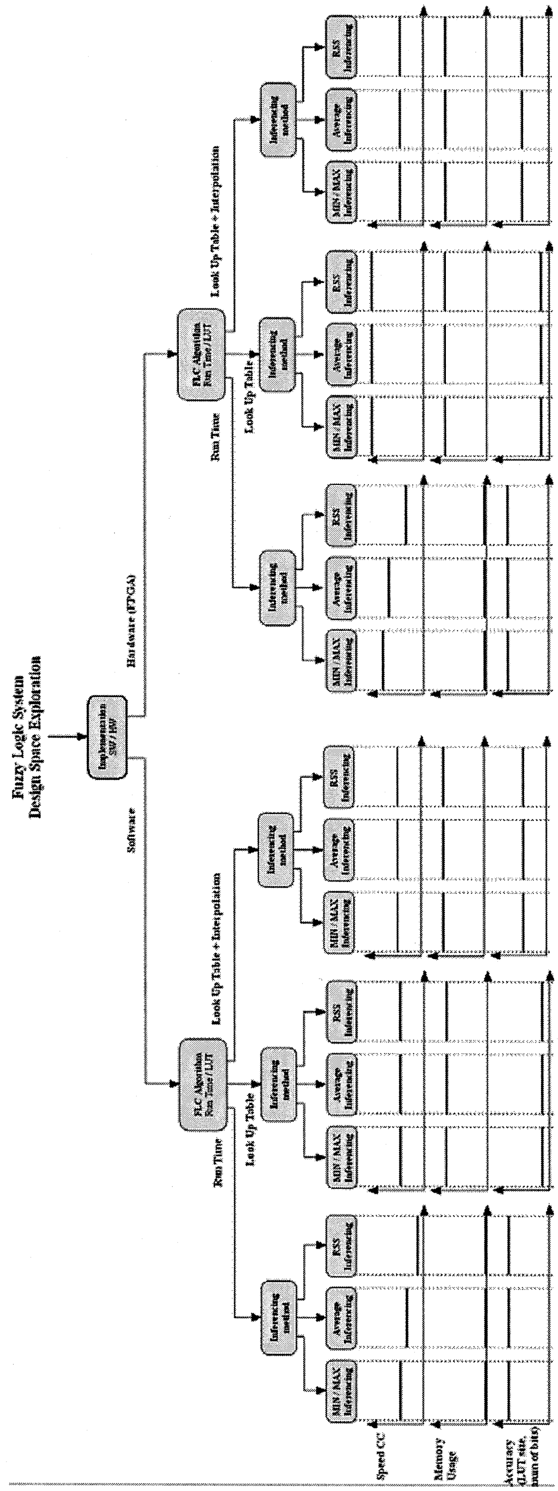


Figure A.1 : Architectural Configuration Graph of Fuzzy Logic System

Appendix B

Fuzzy Control Spaces

Fuzzy Control Space (FCS) is a composition of all possible responses of fuzzy system on various inputs. Plotting the FCS visualizes a degree of control which fuzzy system applies on every particular set of inputs. A 3-d plot of a fuzzy LUT is direct visualization of FCS.

Graphical visualization of FCS gives an indication of the degree and quality of the control in terms of smoothness and boundaries. Two fuzzy logic controllers with the same set of membership functions and same rule matrixes, the FCS can be different if those controllers have different fuzzy parameters. The Combinational Operator, Inferencing Operator, and Defuzzification method vastly influence the shape of the FCS. Reviewing differences in FCS shapes makes it possible to find optimal fuzzy parameters providing required degree of control for the application.

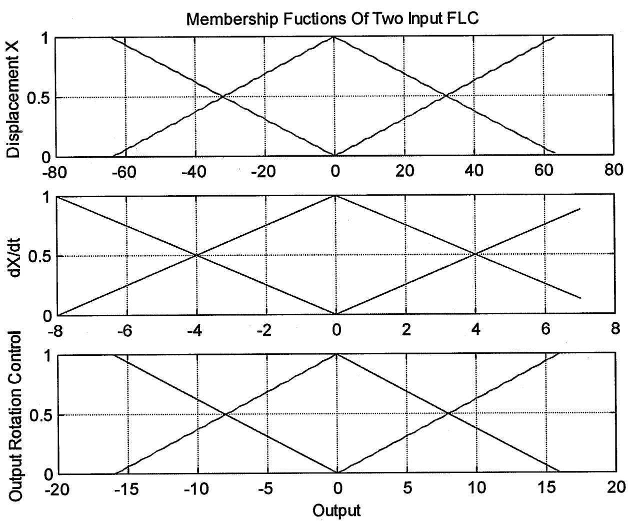
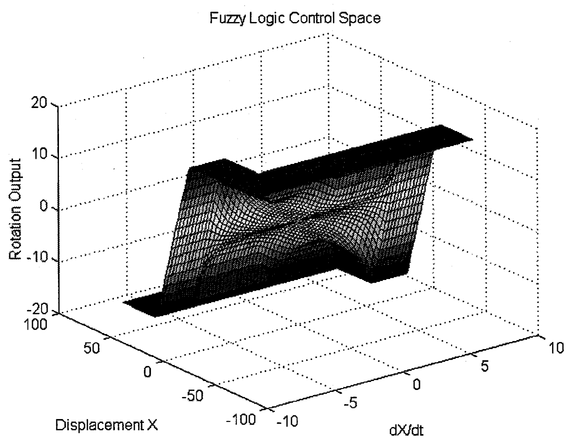


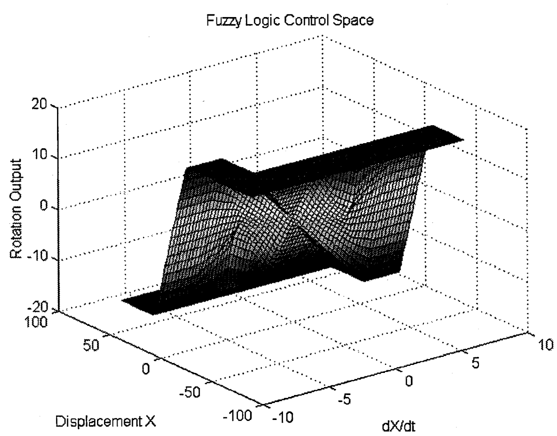
Figure B.1 : Input/Output membership functions for STS FLC

A number of different control space graphs for FLC for STS with identical membership functions (See Figure B.1) and different fuzzy parameters are presented below. All graphs are results of Matlab simulation. Every case presented below is capable of achieving required control of STS application; however each set of parameters give different degree of smoothness/sharpness to control.

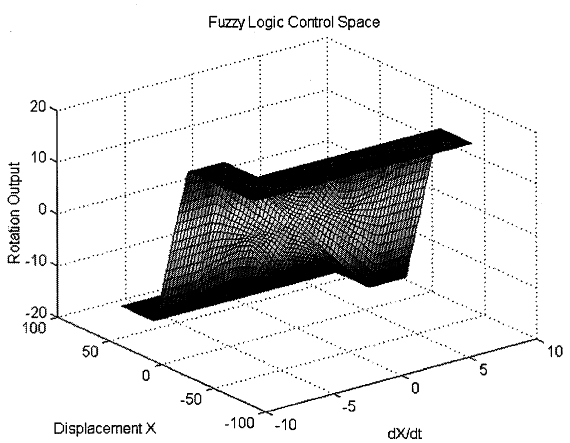
Combinational Operator: AND
Inferencing: Average
Defuzzification: Fuzzy Centroid



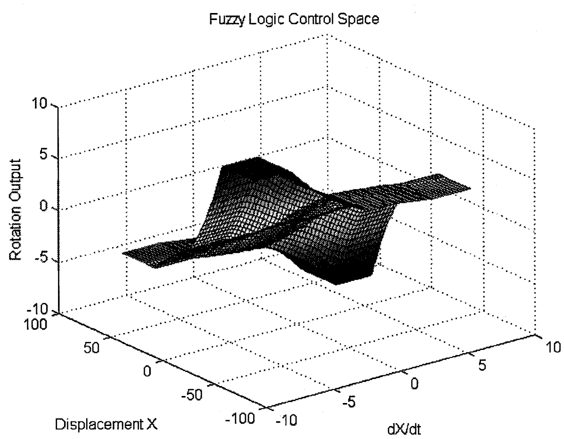
Combinational Operator: AND
Inferencing: MAX
Defuzzification: Fuzzy Centroid



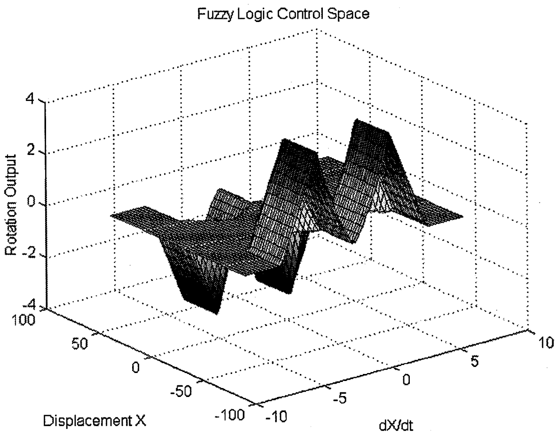
Combinational Operator: AND
Inferencing: RSS
Defuzzification: Fuzzy Centroid

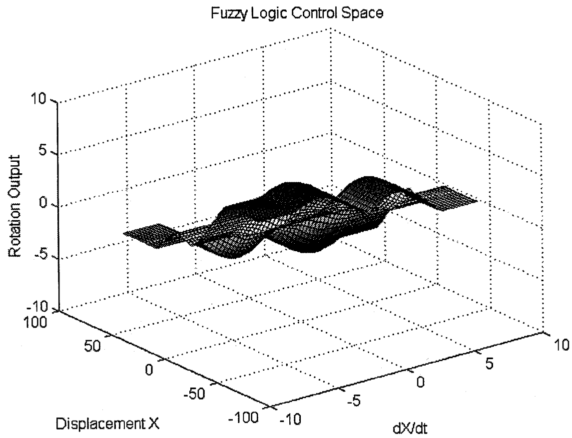


Combinational Operator: OR
Inferencing: Average
Defuzzification: Fuzzy Centroid



Combinational Operator:	OR	Combinational Operator:	OR
Inferencing:	MAX	Inferencing:	RSS
Defuzzification:	Fuzzy Centroid	Defuzzification:	Fuzzy Centroid





Appendix C

LUT Generator run results

The LUT Generator run-time printout results in Matlab command window:

```
--------------------------------------------
Fuzzy Logic Controller Configuration
--------------------------------------------
Fuzzy MP Shape: Triangular

Output Control Range : [-16:0:16]

Quantization <Input Function 1> : 128
Quantization <Input Function 2> : 16
Quantization <Output Function>  : 32

Fuzzy Rule Matrix:
    ---------
    L   L   R
    L   DN  R
    L   R   R
    ---------

Fuzzy Logic Options:
Operator:    Average
Inferencing: RSS

--------------------------------------------
Implementation Initializations
--------------------------------------------
Stepper Motor Controller Constants:
Step Bits      : 8
Speed Bits     : 7
Direction Bits : 1

Maximum Speed : 3 [ms/step]
```

```
Minimum Speed : 12 [ms/step]
Speed Step    : 1.000000e-001 [ms/step]

FPGA constants:
BRAM size : 2048 [Bytes]
Segm size : 32 [Bytes]

--------------------------------------------
Starting Simulation
--------------------------------------------
...Fuzzy Membership Functions Initialization
...Creating Input Files
...Fuzzy Logic Calculations
...Building Look-Up Table
   ExecutionTime = 48 [ms]
...Creating VHDL LUT file < lut_bram_rotation.vhd>
...End
»
```

The graphical representation of input/output membership functions and fuzzy logic control space is presented in Figure C.1.

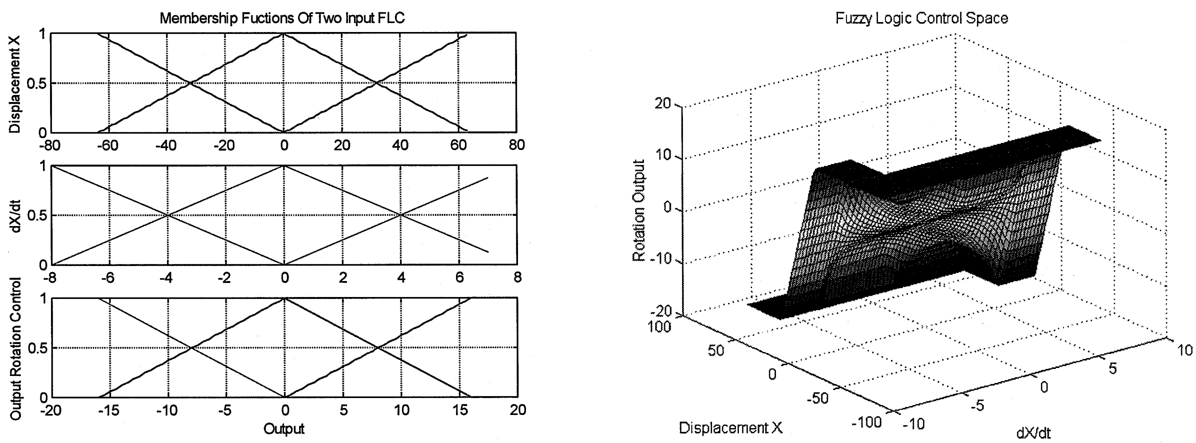


Figure C.1 : Membership functions and fuzzy logic control space

Appendix D

Stereovision Tracking System Schematics

The two implementation versions for proof of concept of Stereovision Tracking System with Fuzzy Logic Control were developed. The first one is a fully operational version of STS implemented on the Virtex2 STS platform. The top-level schematic[4] of fully operational version of STS is presented in Figure D.1.

For the purpose of testing and validation of Disparity Processor and Fuzzy Logic Controller, the simplified version of STS was developed. This version does not produce the disparity data from stereovision cameras at real time, but introduces to the system already pre-stored in BRAM disparity map of the test sample. Figure D.2 presents the top-level schematic of this version. The *flcdisp_controller* module is serving the purpose of loading the pre-stored disparity map and substitutes the entire section of the camera control, rectification and disparity map generation. The *disparity_processor* and *FLC* modules are the same as in fully operation version.

[4] The Xilinx ISE project of STS is organized that the top-level schematic (.sch) is the main project file. It allows top-level schematic to serve two purposes: a) schematically represent all system modules and components b) perform synthesis of the design with Xilinx ISE.

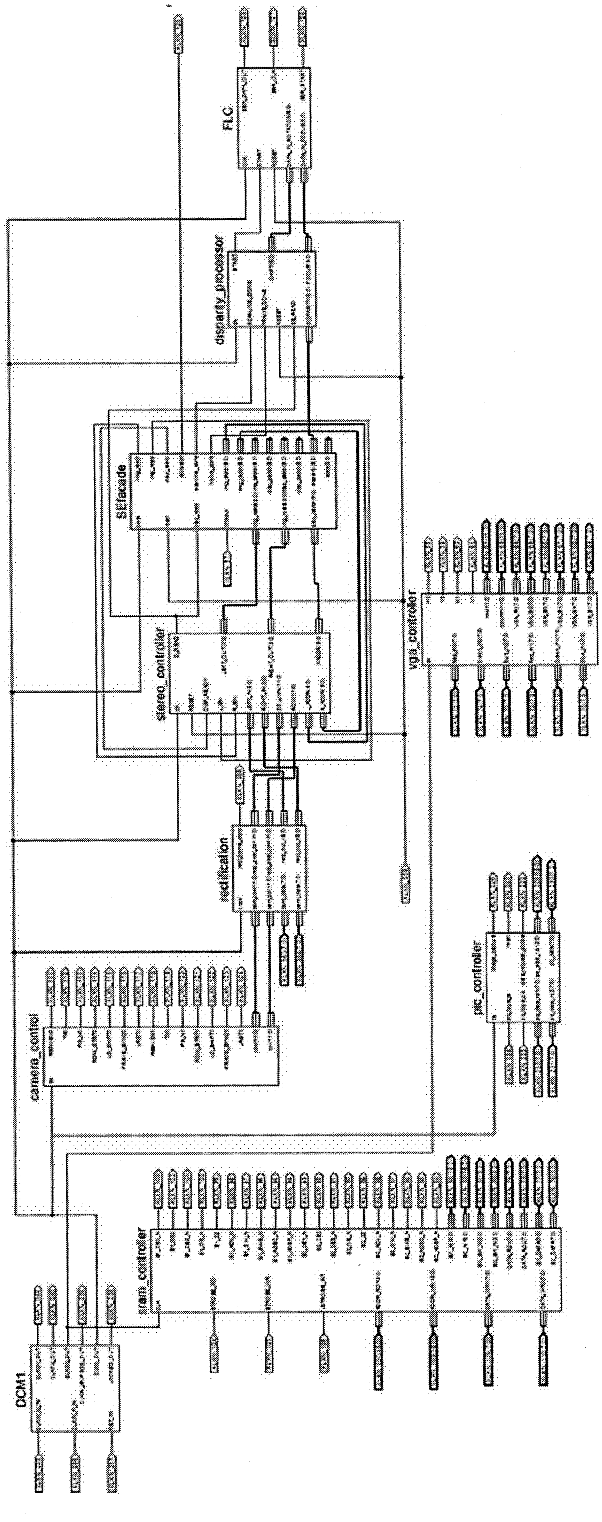


Figure D.1 : Top-level schematic of STS with FLC - FPGA implementation.

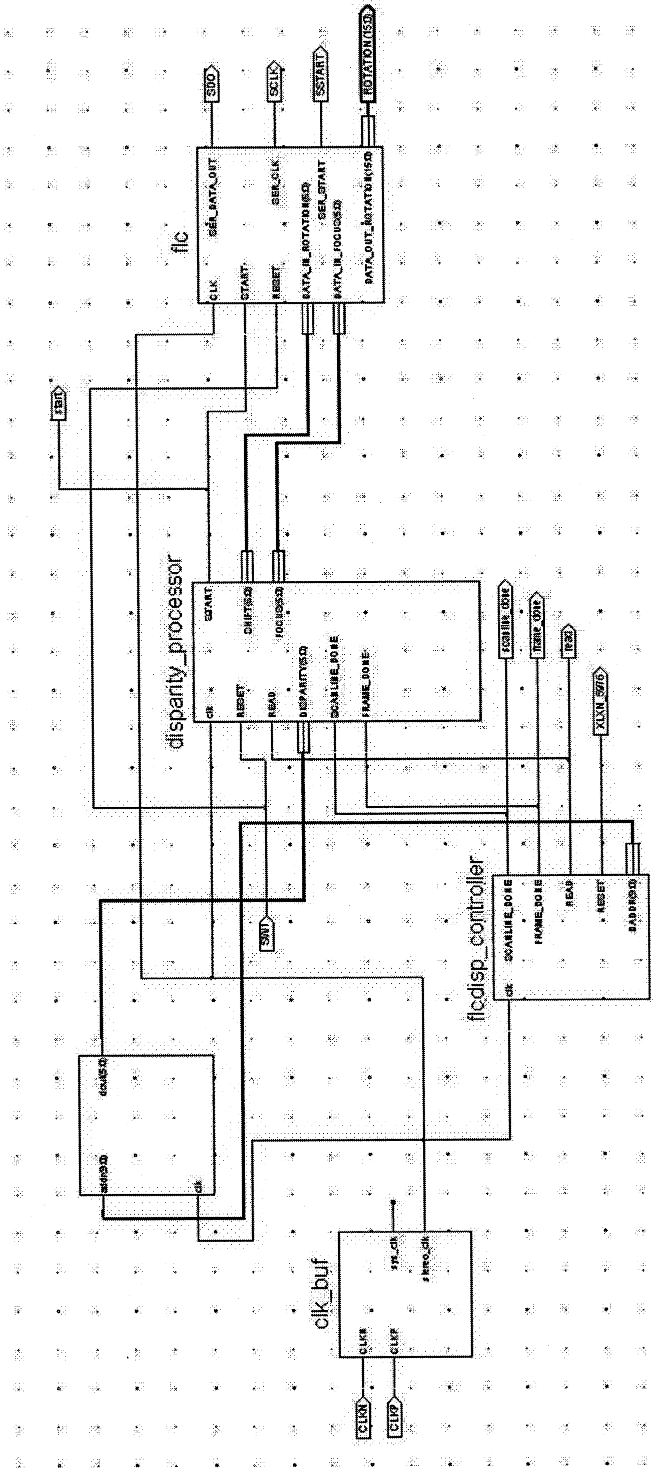


Figure D.2 : Top-level schematic of test module of FLC - FPGA implementation.

Appendix E

Fuzzy Logic Controller Synthesis

Synthesis results of FLC module are presented in this section. The synthesis was generated using Xilinx ISE Version 9.2.03i. The device properties are listed below:
Virtex2P - XC2VP2 - FF672 -6

E.1 Synthesis Results

VIRTEX2 Project Status			
Project File:	VIRTEX2.ise	**Current State:**	Placed and Routed
Module Name:	FLC	• **Errors:**	No Errors
Target Device:	xc2vp2-6ff672	• **Warnings:**	8 Warnings
Product Version:	ISE 9.2.03i	• **Updated:**	Tue Aug 5 20:33:41 2008

VIRTEX2 Partition Summary
No partition information was found.

Device Utilization Summary				
Logic Utilization	**Used**	**Available**	**Utilization**	**Note(s)**
Number of Slice Flip Flops	181	2,816	6%	
Number of 4 input LUTs	129	2,816	4%	
Logic Distribution				
Number of occupied Slices	157	1,408	11%	
Number of Slices containing only related logic	157	157	100%	

Number of Slices containing unrelated logic	0	157	0%	
Total Number of 4 input LUTs	140	2,816	4%	
Number used as logic	129			
Number used as a route-thru	11			
Number of bonded IOBs	19	204	9%	
IOB Flip Flops	15			
Number of PPC405s	0	0	0%	
Number of Block RAMs	3	12	25%	
Number of GCLKs	1	16	6%	
Number of GTs	0	4	0%	
Number of GT10s	0	0	0%	
Total equivalent gate count for design	199,139			
Additional JTAG gate count for IOBs	912			

Performance Summary			
Final Timing Score:	0	**Pinout Data:**	Pinout Report
Routing Results:	All Signals Completely Routed	**Clock Data:**	Clock Report
Timing Constraints:	All Constraints Met		

Detailed Reports					
Report Name	**Status**	**Generated**	**Errors**	**Warnings**	**Infos**
Synthesis Report	Current	Tue Aug 5 20:18:31 2008	0	8 Warnings	9 Infos
Translation Report	Current	Tue Aug 5 20:33:21 2008	0	0	0
Map Report	Current	Tue Aug 5 20:33:26 2008	0	0	3 Infos
Place and Route Report	Current	Tue Aug 5 20:33:37 2008	0	0	2 Infos
Static Timing Report	Current	Tue Aug 5 20:33:40 2008	0	0	3 Infos
Bitgen Report					

Figure E.1 : Synthesis results summary.

Timing synthesis results produced by Xilinx ISE are as follows:

```
Timing Summary:
---------------
Speed Grade: -6

   Minimum period: 3.752ns (Maximum Frequency: 266.489MHz)
   Minimum input arrival time before clock: 1.984ns
   Maximum output required time after clock: 4.516ns
   Maximum combinational path delay: No path found

=========================================================================
Process "Synthesize" completed successfully
```

E.2 RTL Schematics

RTL schematics of FLC module and sub-modules are presented in this section. The main FLC module RTL schematic is presented in Figure E.2.

The RTL schematic of Rotation sub-module FLC_ROTATION is presented in Figure E.3. This module represents a main concept of look-up table fuzzy logic controller. The input of this module is a displacement X, the speed dX/dt is calculated at first stage, then the X and the dX/dt is used by address decoder to calculate the address of output element in BRAM memory block LUT_ROTATION (see Figure E.4). The focus control module FLC_FOCUS has similar architecture with exception that it uses one BRAM in LUT_FOCUS as oppose to 2 BRAMs in LUT_ROTAION.

The RTL schematic of CHECK module, LUT_FOCUS module, and basic BRAM module are presented in Figure E.5.

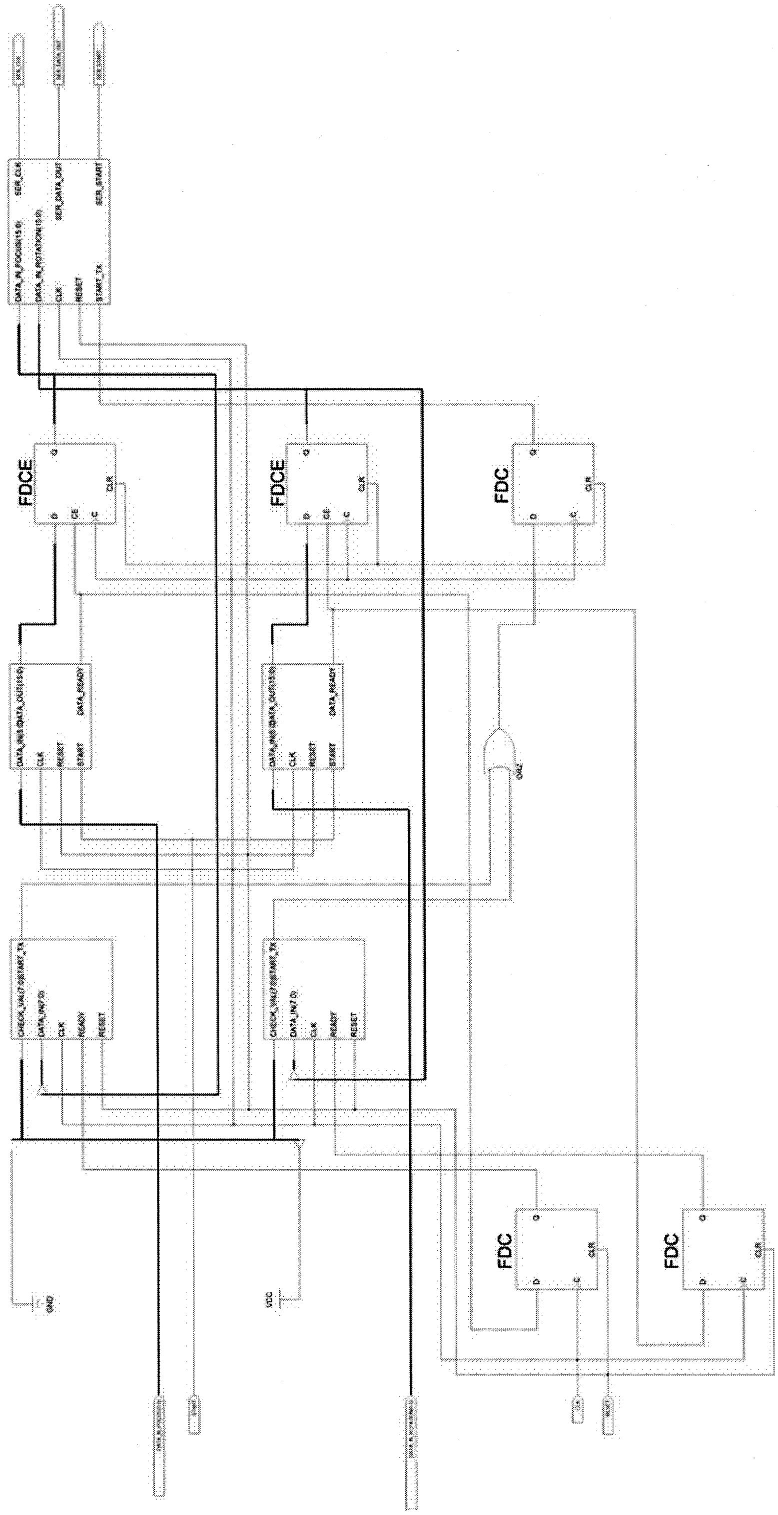


Figure E.2 : RTL schematics of FLC module.

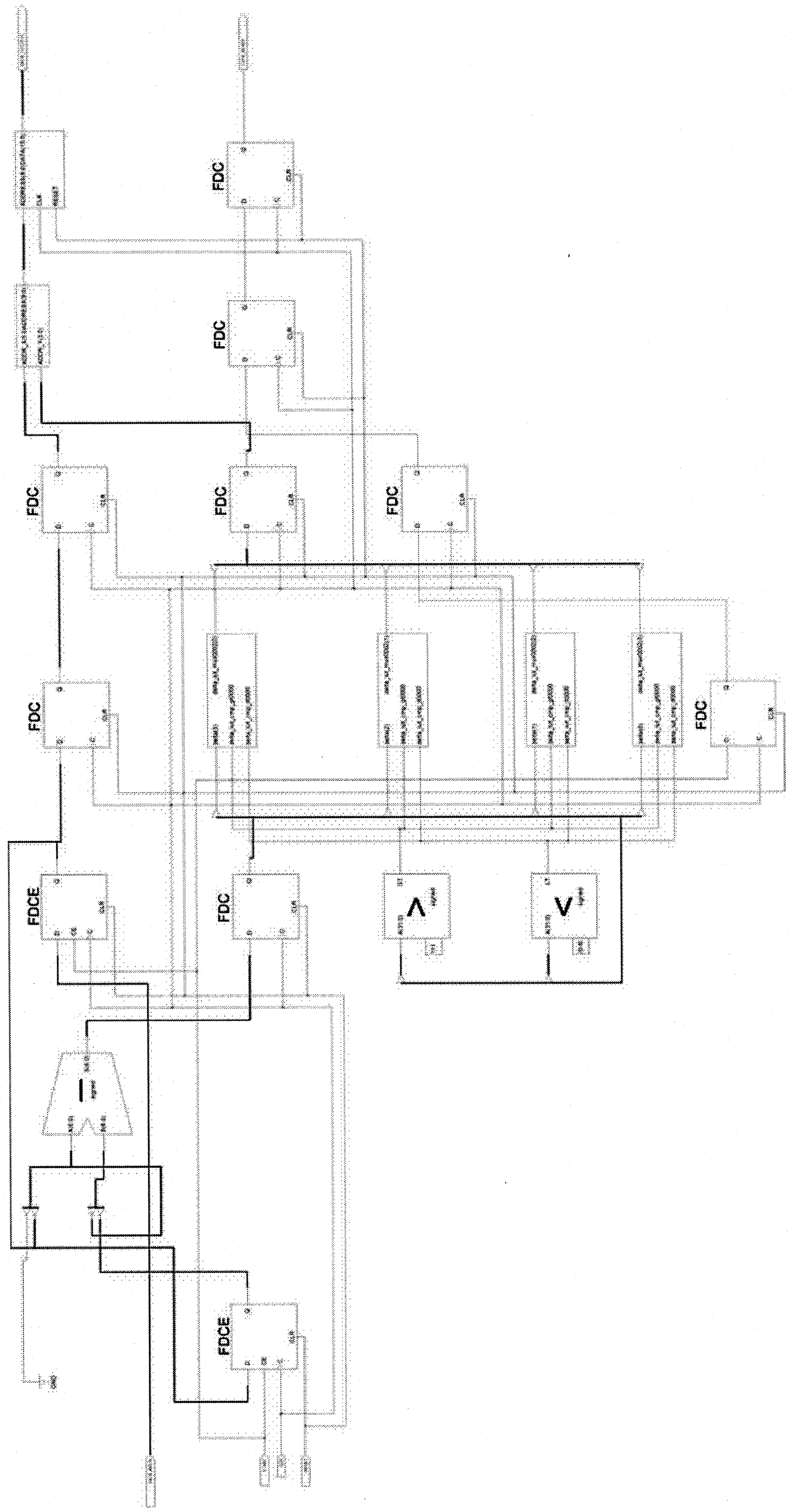


Figure E.3 : RTL schematics of Rotation module.

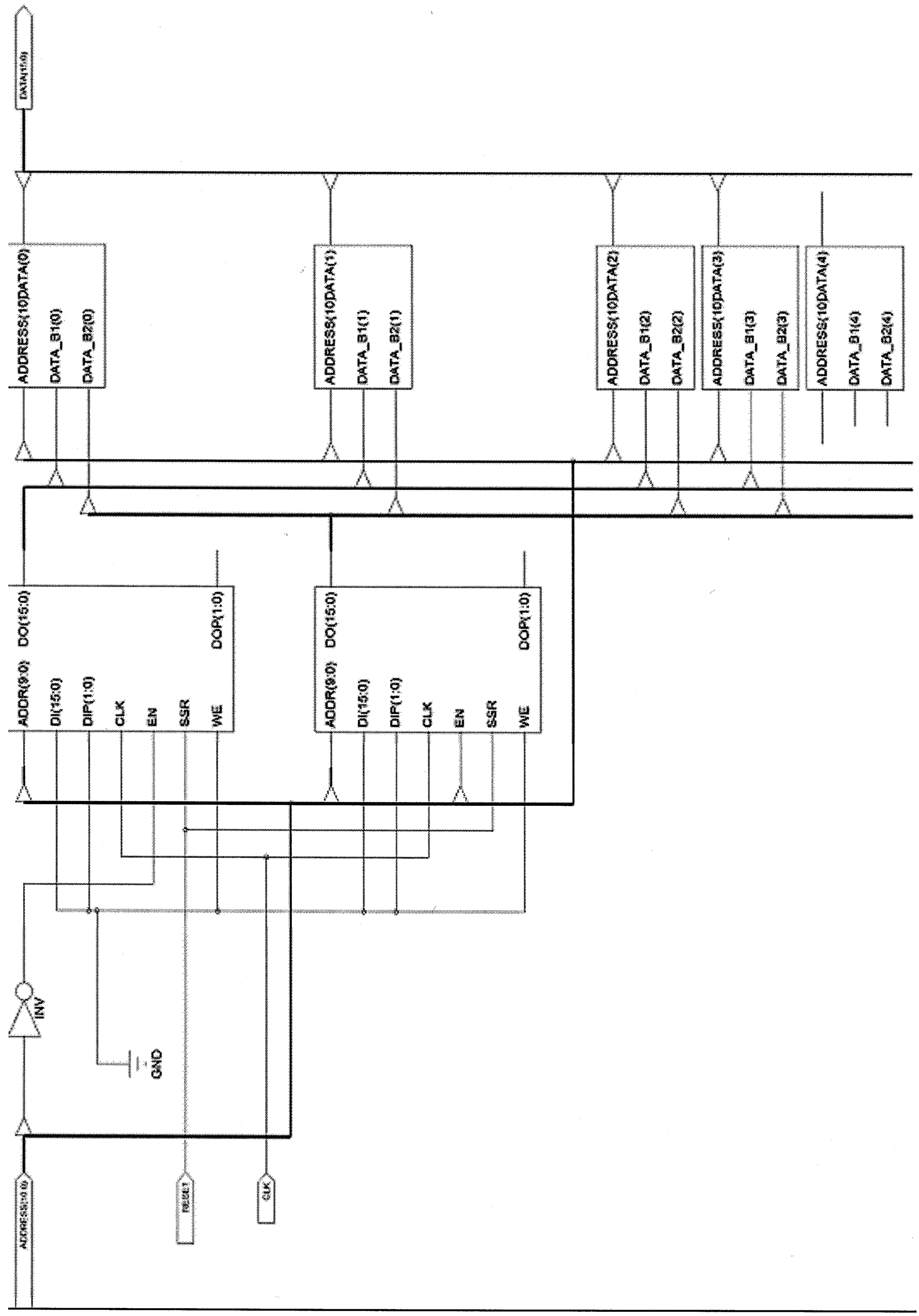


Figure E.4 : Fragment of RTL schematic of LUT_ROTATION module

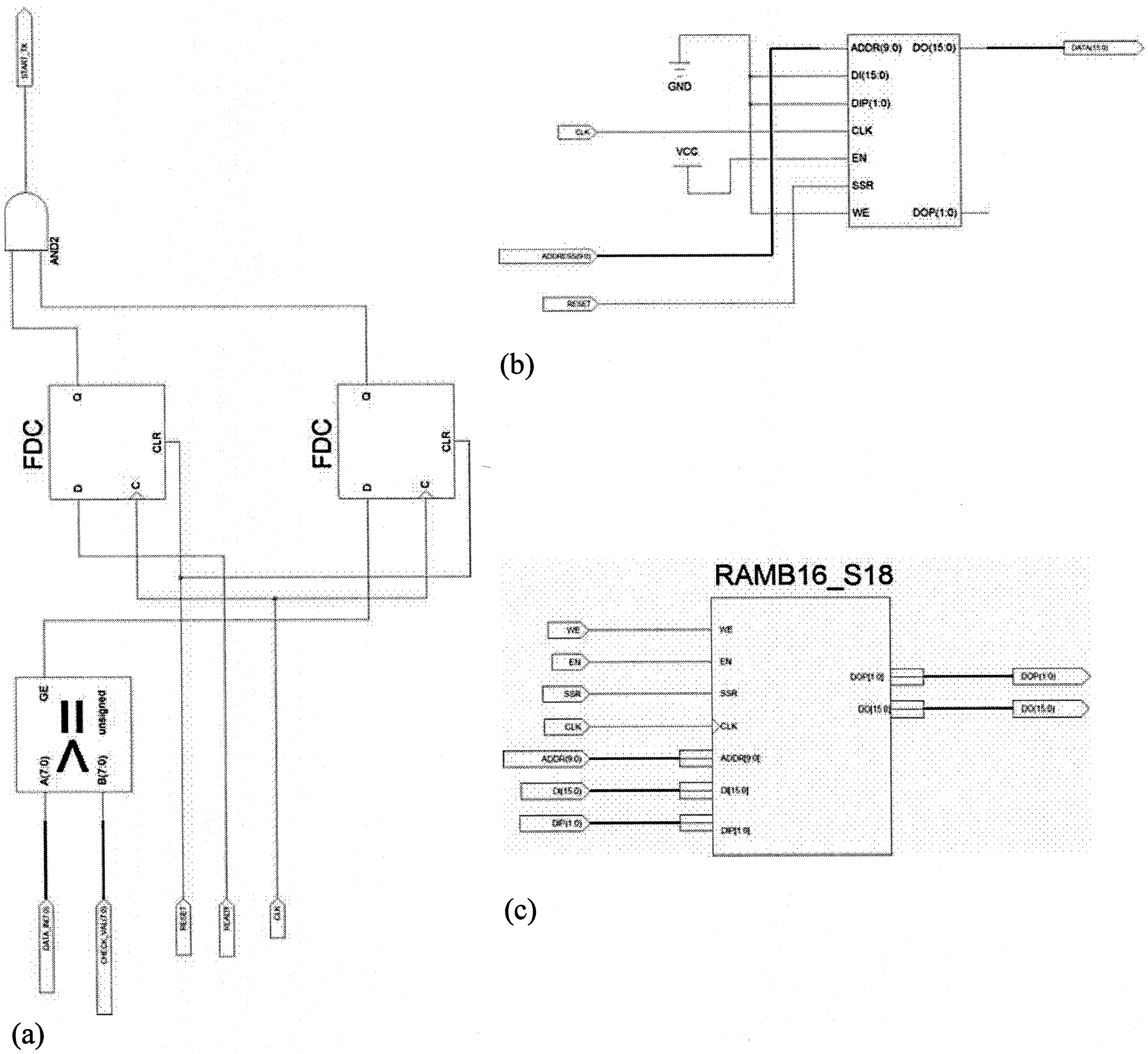


Figure E.5 : RTL schematic of (a) CHECK module, (b) LUT_FOCUS module, (c) basic BRAM module

Appendix F

Experiment

This appendix presents details of simulation and experimental testing: ModelSim simulations results, top-level schematics, pre-loaded images, reference tables and values that are used in interpreting experiment results.

F.1 Simulation results waveforms

All simulation results have been acquired using ModelSim SE Plus 6.2g software.

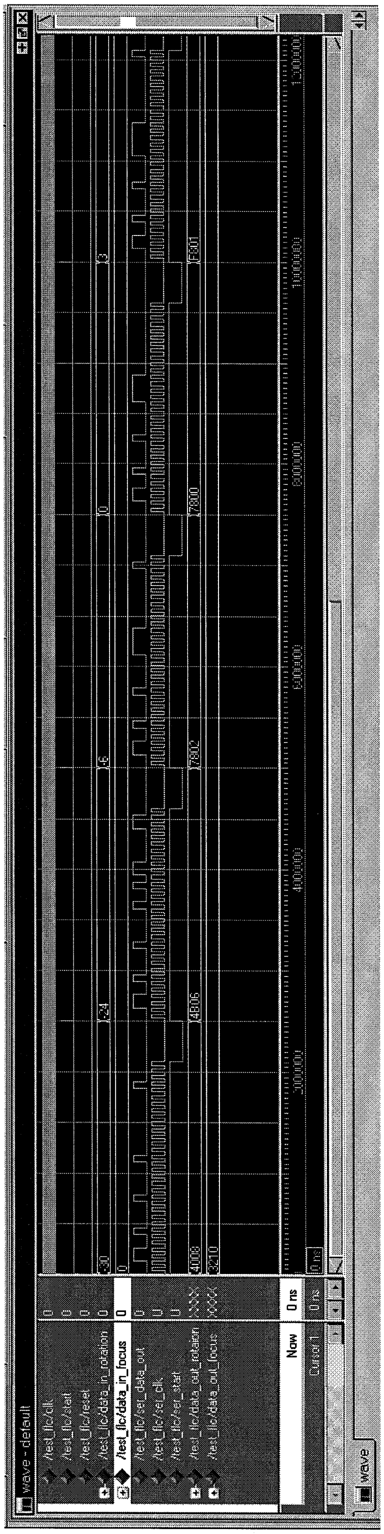
The results of behavioral simulation of FLC module are presented in Figure F.1.

The results of post-routing simulation of FLC module are presented in Figure F.2.

The displayed signals are as follows:

/test_flc/clk	CLK
/test_flc/start	START
/test_flc/RESET	RESET
/test_flc/data_in_rotaiton	DATA_IN_ROTATION (6:0)
/test_flc/data_in_focus	DATA_IN_FOCUS (5:0)
/test_flc/ser_data_out	SER_DATA_OUT
/test_flc/ser_clk	SER_CLK
/test_flc/ser_start	SER_START
/test_flc/data_out_rotation	Debug output - rotation control value
/test_flc/data_out_focus	Debug output - focus control value

Figure F.1 : Behavioral simulation results



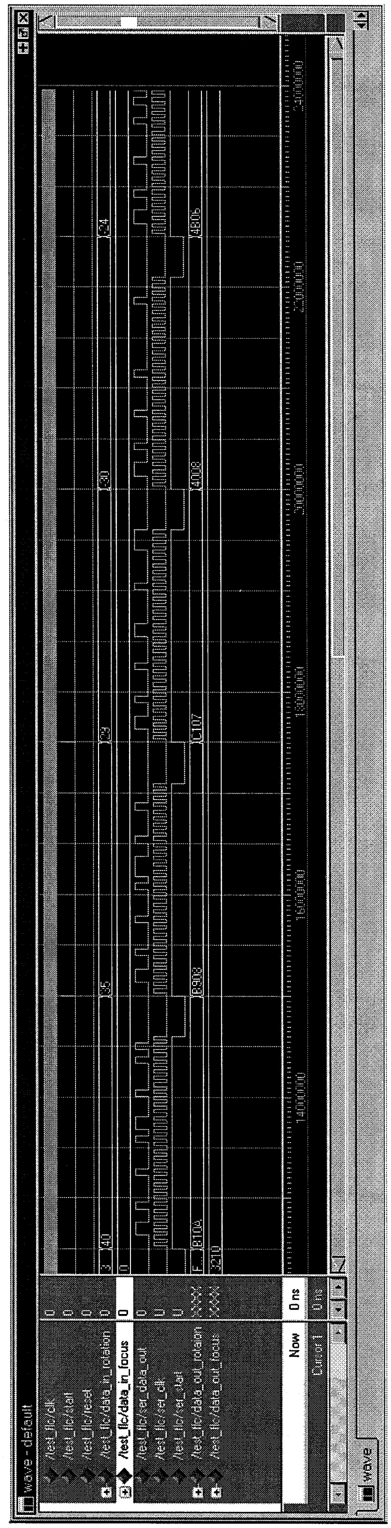


Figure F.2 : Post-routing simulation results

F.2 Experimental set-up schematic

The experimental set-up is designed to introduce preloaded images to *disparity_processor* module and subsequently to *FLC*. (See top-level schematics Figure F.4) The *data_select* module is designed to select between number of images pre-stored in the BRAM. Each image is organized as a separate module. The static experiment uses 3 images stored in BRAM. The selection between images is done by reading the dip-switches on the STS Video Processing Module (Amirix AP1100 FPGA Development Board). (See Figure F.3) The dynamic experiment uses 8 pre-loaded images, and the selection between images is done automatically. Every START pulse the next image is loaded to the *disparity_processor*.

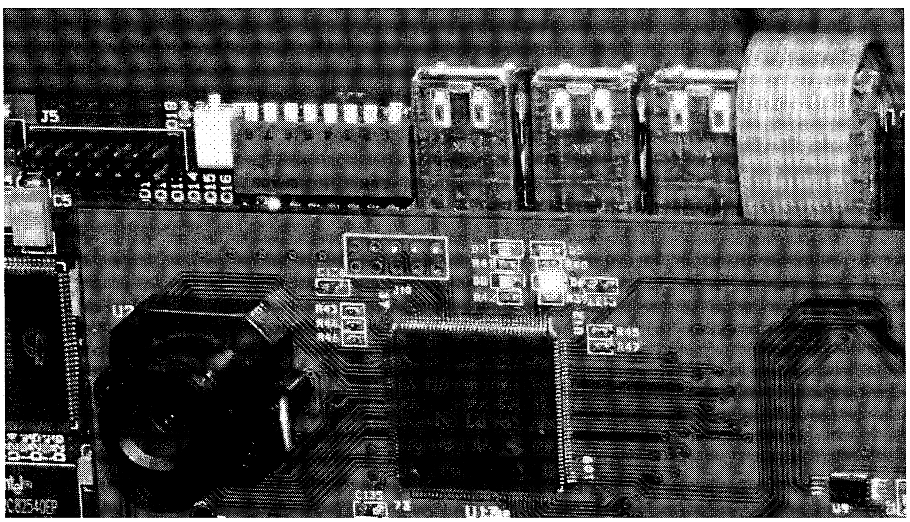


Figure F.3 : Dip switches on Amirix AP 1100 FPGA Development Board

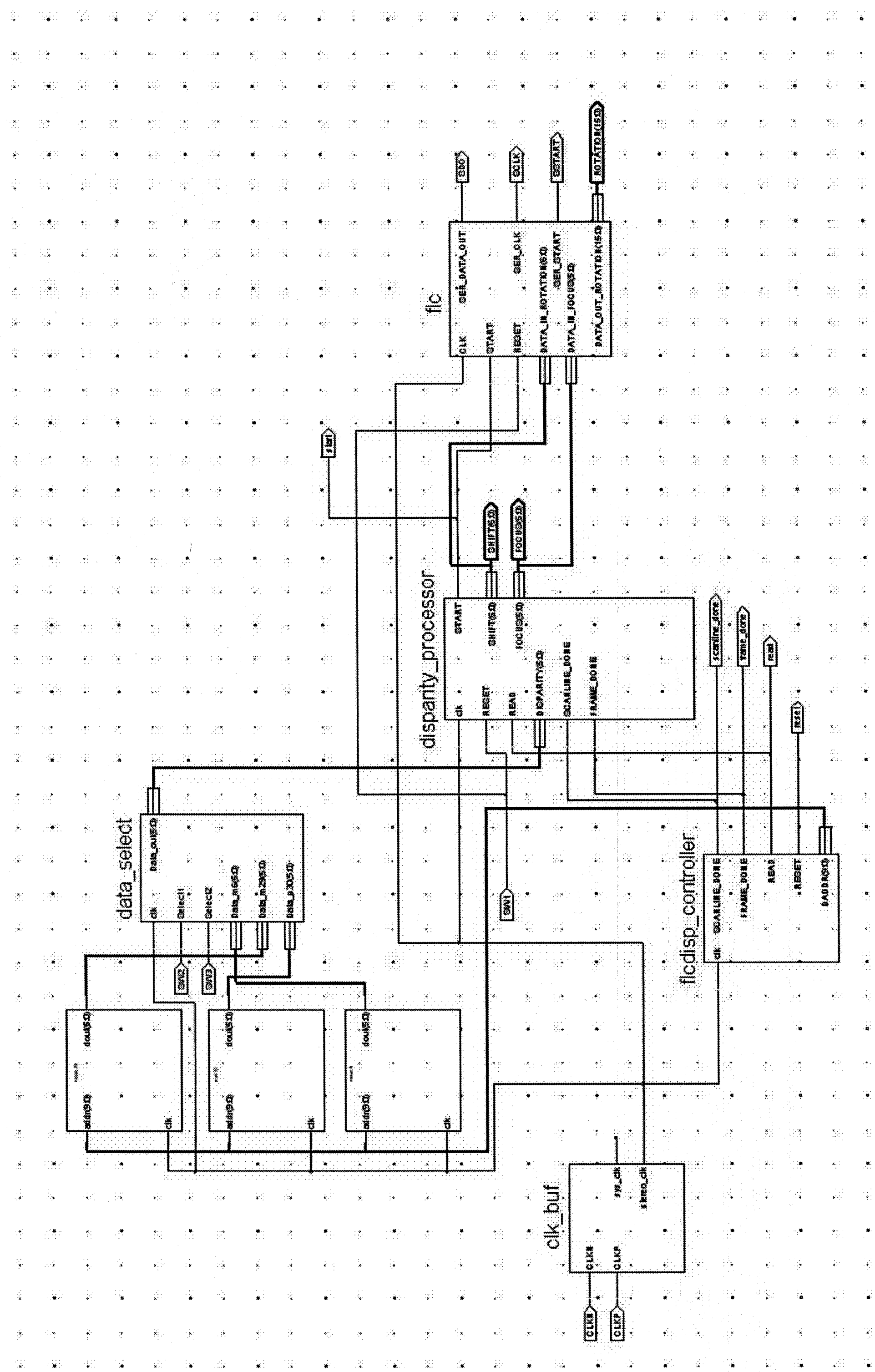


Figure F.4 : Top-level schematic of experimental set-up module with preloaded images.

F.3 Preloaded images

The following preloaded images are used in the static experiment. Each image is an ideal representation of object's disparity map. The only middle horizontal line of the image is used in calculation of the center of the object.

Image 1:

left edge 172, right edge 375 => Displacement = -6

```
; This .COE file specifies initialization values for a
; block memory of depth=640, and width=8. In this case,
; values are specified in hexadecimal format.

memory_initialization_radix=16;
memory_initialization_vector=

00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,
00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,
00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,
00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,
00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,
00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,0F,0F,0F,0F,0F,0F,0F,0F,0F,0F,0F,0F,0F,0F,
0F,0F,0F,0F,0F,0F,0F,0F,0F,0F,0F,0F,0F,0F,0F,0F,0F,0F,0F,0F,0F,0F,0F,0F,0F,0F,0F,0F,0F,0F,
0F,0F,0F,0F,0F,0F,0F,0F,0F,0F,0F,0F,0F,0F,0F,0F,0F,0F,0F,0F,0F,0F,0F,0F,0F,0F,0F,0F,0F,0F,
0F,0F,0F,0F,0F,0F,0F,0F,0F,0F,0F,0F,0F,0F,0F,0F,0F,0F,0F,0F,0F,0F,0F,0F,0F,0F,0F,0F,0F,0F,
0F,0F,0F,0F,0F,0F,0F,0F,0F,0F,0F,0F,0F,0F,0F,0F,0F,0F,0F,0F,0F,0F,0F,0F,0F,0F,0F,0F,0F,0F,
0F,0F,0F,0F,0F,0F,0F,0F,0F,0F,0F,0F,0F,0F,0F,0F,0F,0F,0F,0F,0F,0F,0F,0F,0F,0F,0F,0F,0F,0F,
0F,0F,0F,0F,0F,0F,0F,0F,0F,0F,0F,0F,0F,0F,0F,0F,0F,0F,0F,0F,0F,0F,0F,0F,0F,0F,0F,0F,0F,0F,
0F,0F,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,
00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,
00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,
00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,
00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,
00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,
00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,
00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,
00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00;
```

Image 2:

left edge 70, right edge 103 => Displacement = -29

```
; This .COE file specifies initialization values for a
; block memory of depth=640, and width=8. In this case,
; values are specified in hexadecimal format.

memory_initialization_radix=16;
memory_initialization_vector=

00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,
00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,
00,00,00,00,00,00,0F,0F,0F,0F,0F,0F,0F,0F,0F,0F,0F,0F,0F,0F,0F,0F,0F,0F,0F,0F,0F,0F,0F,0F,
0F,0F,0F,0F,0F,0F,0F,0F,0F,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,
00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,
00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,
00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,
00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,
```

```
00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,
00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,
00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,
00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,
00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,
00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,
00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,
00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,
00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,
00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,
00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,
00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,
00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00;
```

Image 3:

left edge 552, right edge 572 => Displacement = 30

```
; This .COE file specifies initialization values for a
; block memory of depth=640, and width=8. In this case,
; values are specified in hexadecimal format.

memory_initialization_radix=16;
memory_initialization_vector=

00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,
00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,
00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,
00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,
00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,
00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,
00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,
00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,
00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,
00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,
00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,
00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,
00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,
00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,
00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,
00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,
00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,
00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,0F,0F,0F,0F,0F,0F,0F,0F,0F,
0F,0F,0F,0F,0F,0F,0F,0F,0F,0F,0F,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,
00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,
00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00;
```

F.4 Reference LUT

The look-up table for rotation control generated by LUT Generator build environment is presented below. The values are presented in hexadecimal format. Each raw represents a displacement X, each column represents a speed (ΔX/Δt). The values marked in bold were referenced in the experiment.

	-8	-7	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6	7
-64	1E10	1E10	1E10	1E10	1E10	1E10	1E10	1E10	1E10	1E10	1E10	1E10	1E10	1E10	1E10	1E10
-63	1E10	1F10	1F10	1F10	1F10	1F10	1F10	1F10	1E10	1F10	1F10	1F10	1F10	1F10	1F10	1F10
-62	1F10	200F	200F	200F	200F	200F	200F	200F	1F10	200F	200F	200F	200F	200F	200F	200F
-61	1F0F	200F	210F	210F	210F	210F	210F	200F	1F0F	200F	210F	210F	210F	210F	210F	200F
-60	200F	210F	210E	220E	220E	220E	210E	210F	200F	210F	210E	220E	220E	220E	210E	210F
-59	210F	220E	220E	230E	230E	230E	220E	220E	210F	220E	220E	230E	230E	230E	220E	220E
-58	210F	230E	230E	230E	240D	230E	230E	230E	210F	230E	230E	230E	240D	230E	230E	230E
-57	220E	230E	240D	240D	250D	240D	240D	230E	220E	230E	240D	240D	250D	240D	240D	230E
-56	220E	240D	250D	250D	260D	250D	250D	240D	220E	240D	250D	250D	260D	250D	250D	240D
-55	230E	240D	260D	260D	260C	260D	260D	240D	230E	240D	260D	260D	260C	260D	260D	240D
-54	240E	250D	260D	270C	270C	270C	260D	250D	240E	250D	260D	270C	270C	270C	260D	250D
-53	240D	260D	270C	280C	280C	280C	270C	260D	240D	260D	270C	280C	280C	280C	270C	260D
-52	250D	260D	280C	290C	290C	290C	280C	260D	250D	260D	280C	290C	290C	290C	280C	260D
-51	260D	270C	290C	2A0B	2A0B	2A0B	290C	270C	260D	270C	290C	2A0B	2A0B	2A0B	290C	270C
-50	260D	280C	2A0B	2B0B	2B0B	2B0B	2A0B	280C	260D	280C	2A0B	2B0B	2B0B	2B0B	2A0B	280C
-49	270C	280C	2B0B	2C0B	2C0B	2C0B	2B0B	280C	270C	280C	2B0B	2C0B	2C0B	2C0B	2B0B	280C
-48	280C	290C	2B0B	2D0B	2D0B	2D0B	2B0B	290C	280C	290C	2B0B	2D0B	2D0B	2D0B	2B0B	290C
-47	290C	2A0B	2C0B	2D0B	2E0A	2D0B	2C0B	2A0B	290C	2A0B	2C0B	2D0B	2E0A	2D0B	2C0B	2A0B
-46	2A0C	2B0B	2D0B	2E0A	2F0A	2E0A	2D0B	2B0B	2A0C	2B0B	2D0B	2E0A	2F0A	2E0A	2D0B	2B0B
-45	2B0B	2C0B	2E0B	2F0A	300A	2F0A	2E0B	2C0B	2B0B	2C0B	2E0B	2F0A	300A	2F0A	2E0B	2C0B
-44	2C0B	2C0B	2E0A	300A	310A	300A	2E0A	2C0B	2C0B	2C0B	2E0A	300A	310A	300A	2E0A	2C0B
-43	2D0B	2D0B	2F0A	310A	320A	310A	2F0A	2D0B	2D0B	2D0B	2F0A	310A	320A	310A	2F0A	2D0B
-42	2E0B	2E0A	300A	320A	3309	320A	300A	2E0A	2E0B	2E0A	300A	320A	3309	320A	300A	2E0A
-41	2F0A	2F0A	310A	3309	3409	3309	310A	2F0A	2F0A	2F0A	310A	3309	3409	3309	310A	2F0A
-40	300A	310A	320A	3409	3509	3409	320A	310A	300A	310A	320A	3409	3509	3409	320A	310A
-39	310A	320A	3309	3509	3509	3509	3309	320A	310A	320A	3309	3509	3509	3509	3309	320A
-38	330A	3309	3409	3609	3609	3609	3409	3309	330A	3309	3409	3609	3609	3609	3409	3309
-37	3409	3409	3509	3709	3709	3709	3509	3409	3409	3409	3509	3709	3709	3709	3509	3409
-36	3509	3609	3709	3809	3809	3809	3709	3609	3509	3609	3709	3809	3809	3809	3709	3609
-35	3709	3709	3809	3908	3908	3908	3809	3709	3709	3709	3809	3908	3908	3908	3809	3709
-34	3809	3908	3908	3A08	3A08	3A08	3908	3908	3809	3908	3908	3A08	3A08	3A08	3908	3908
-33	3A08	3A08	3B08	3B08	3B08	3B08	3B08	3A08	3A08	3A08	3B08	3B08	3B08	3B08	3B08	3A08
-32	3C08	3C08	3C08	3C08	3C08	3C08	3C08	3C08	3C08	3C08	3C08	3C08	3C08	3C08	3C08	3C08
-31	3E08	3E08	3E08	3D08	3D08	3D08	3E08	3E08	3E08	3E08	3E08	3D08	3D08	3D08	3E08	3E08
-30	**4008**	4008	3F08	3E08	3E08	3E08	3F08	4008	4008	4008	3F08	3E08	3E08	3E08	3F08	**4008**
-29	**4207**	4207	4107	4008	3F08	4008	4107	4207	4207	4207	4107	4008	3F08	4008	4107	4207
-28	4507	4407	4307	4107	4007	4107	4307	4407	4507	4407	4307	4107	4007	4107	4307	4407
-27	4707	4607	4507	4307	4207	4307	4507	4607	4707	4607	4507	4307	4207	4307	4507	4607
-26	4A07	4907	4707	4407	4307	4407	4707	4907	4A07	4907	4707	4407	4307	4407	4707	4907
-25	4D06	4C06	4907	4607	4407	4607	4907	4C06	4D06	4C06	4907	4607	4407	4607	4907	4C06
-24	5006	4E06	4B06	4707	4607	4707	4B06	4E06	5006	4E06	4B06	4707	4607	4707	**4B06**	4E06
-23	5306	5106	4D06	4907	4807	4907	4D06	5106	5306	5106	4D06	4907	4807	4907	4D06	5106
-22	5706	5506	5006	4B06	4A07	4B06	5006	5506	5706	5506	5006	4B06	4A07	4B06	5006	5506
-21	5B05	5805	5206	4D06	4C06	4D06	5206	5805	5B05	5805	5206	4D06	4C06	4D06	5206	5805
-20	6005	5C05	5506	4F06	4E06	4F06	5506	5C05	6005	5C05	5506	4F06	4E06	4F06	5506	5C05
-19	6505	6105	5805	5206	5106	5206	5805	6105	6505	6105	5805	5206	5106	5206	5805	6105
-18	6B05	6505	5B05	5506	5306	5506	5B05	6505	6B05	6505	5B05	5506	5306	5506	5B05	6505
-17	7104	6A05	5E05	5805	5606	5805	5E05	6A05	7104	6A05	5E05	5805	5606	5805	5E05	6A05
-16	7804	7004	6105	5C05	5A05	5C05	6105	7004	7804	7004	6105	5C05	5A05	5C05	6105	7004
-15	7804	7604	6605	6005	5E05	6005	6605	7604	7804	7604	6605	6005	5E05	6005	6605	7604
-14	7804	7804	6B05	6505	6305	6505	6B05	7804	7804	7804	6B05	6505	6305	6505	6B05	7804
-13	7803	7804	7104	6A05	6805	6A05	7104	7804	7803	7804	7104	6A05	6805	6A05	7104	7804
-12	7803	7803	7704	7004	6E04	7004	7704	7803	7803	7803	7704	7004	6E04	7004	7704	7803
-11	7803	7803	7804	7804	7504	7804	7804	7803	7803	7803	7804	7804	7504	7804	7804	7803
-10	7803	7803	7803	7804	7804	7804	7803	7803	7803	7803	7803	7804	7804	7804	7803	7803
- 9	7802	7803	7803	7803	7804	7803	7803	7803	7802	7803	7803	7803	7804	7803	7803	7803
- 8	7802	7803	7803	7803	7803	7803	7803	7803	7802	7803	7803	7803	7803	7803	7803	7803
- 7	7802	7802	7803	7803	7803	7803	7803	7802	7802	7802	7803	7803	7803	7803	7803	7802
- 6	7802	7802	**7802**	7802	7803	7802	7802	7802	7802	7802	7802	7802	7803	7802	7802	**7802**
- 5	7801	7802	7802	7802	7802	7802	7802	7802	7801	7802	7802	7802	7802	7802	7802	7802
- 4	7801	7801	7802	7802	7802	7802	7802	7801	7801	7801	7802	7802	7802	7802	7802	7801
- 3	7801	7801	7801	7801	7801	7801	7801	7801	7801	7801	7801	7801	7801	7801	7801	7801
- 2	7801	7801	7801	7801	7801	7801	7801	7801	7801	7801	7801	7801	7801	7801	7801	7801
- 1	7800	7800	7800	7800	7800	7800	7800	7800	7800	7800	7800	7800	7800	7800	7800	7800
0	7800	7800	7800	7800	7800	7800	7800	7800	7800	7800	7800	7800	7800	7800	**7800**	7800

1	F800	F800	F800	F800	F800	F800	F800	F800	F800	F800	F800	F800	F800	F800	F800	F800
2	F801	F801	F801	F801	F801	F801	F801	F801	F801	F801	F801	F801	F801	F801	F801	F801
3	F801	F801	F801	F801	F801	F801	F801	F801	F801	F801	F801	**F801**	F801	F801	F801	F801
4	F801	F801	F802	F802	F802	F802	F802	F801	F801	F801	F802	F802	F802	F802	F802	F801
5	F801	F802	F802	F802	F802	F802	F802	F802	F801	F802	F802	F802	F802	F802	F802	F802
6	F802	F802	F802	F802	F803	F802	F802	F802	F802	F802	F802	F802	F803	F802	F802	F802
7	F802	F802	F803	F803	F803	F803	F803	F802	F802	F802	F803	F803	F803	F803	F803	F802
8	F802	F803	F803	F803	F803	F803	F803	F803	F802	F803	F803	F803	F803	F803	F803	F803
9	F802	F803	F803	F803	F804	F803	F803	F803	F802	F803	F803	F803	F804	F803	F803	F803
10	F803	F803	F803	F804	F804	F804	F803	F803	F803	F803	F803	F804	F804	F804	F803	F803
11	F803	F803	F804	F804	F504	F804	F804	F803	F803	F803	F804	F804	F504	F804	F804	F803
12	F803	F803	F704	F004	EE04	F004	F704	F803	F803	F803	F704	F004	EE04	F004	F704	F803
13	F803	F804	F104	EA05	E805	EA05	F104	F804	F803	F804	F104	EA05	E805	EA05	F104	F804
14	F804	F804	EB05	E505	E305	E505	EB05	F804	F804	F804	EB05	E505	E305	E505	EB05	F804
15	F804	F604	E605	E005	DE05	E005	E605	F604	F804	F604	E605	E005	DE05	E005	E605	F604
16	F804	F004	E105	DC05	DA05	DC05	E105	F004	F804	F004	E105	DC05	DA05	DC05	E105	F004
17	F104	EA05	DE05	D805	D606	D805	DE05	EA05	F104	EA05	DE05	D805	D606	D805	DE05	EA05
18	EB05	E505	DB05	D506	D306	D506	DB05	E505	EB05	E505	DB05	D506	D306	D506	DB05	E505
19	E505	E105	D805	D206	D106	D206	D805	E105	E505	E105	D805	D206	D106	D206	D805	E105
20	E005	DC05	D506	CF06	CE06	CF06	D506	DC05	E005	DC05	D506	CF06	CE06	CF06	D506	DC05
21	DB05	D805	D206	CD06	CC06	CD06	D206	D805	DB05	D805	D206	CD06	CC06	CD06	D206	D805
22	D706	D506	D006	CB06	CA07	CB06	D006	D506	D706	D506	D006	CB06	CA07	CB06	D006	D506
23	D306	D106	CD06	C907	C807	C907	CD06	D106	D306	D106	CD06	C907	C807	C907	CD06	D106
24	D006	CE06	CB06	C707	C607	C707	CB06	CE06	D006	CE06	CB06	C707	C607	C707	CB06	CE06
25	CD06	CC06	C907	C607	C407	C607	C907	CC06	CD06	CC06	C907	C607	C407	C607	C907	CC06
26	CA07	C907	C707	C407	C307	C407	C707	C907	CA07	C907	C707	C407	C307	C407	C707	C907
27	C707	C607	C507	C307	C207	C307	C507	C607	C707	C607	C507	C307	C207	C307	C507	C607
28	C507	C407	C307	C107	C007	C107	C307	C407	C507	C407	C307	C107	C007	C107	C307	C407
29	C207	C207	**C107**	C008	BF08	C008	C107	C207	C207	C207	C107	C008	BF08	C008	C107	C207
30	C008	C008	BF08	BE08	BE08	BE08	BF08	C008	C008	C008	BF08	BE08	BE08	BE08	BF08	**C008**
31	BE08	BE08	BE08	BD08	BD08	BD08	BE08	BE08	BE08	BE08	BE08	BD08	BD08	BD08	BE08	BE08
32	BC08	BC08	BC08	BC08	BC08	BC08	BC08	BC08	BC08	BC08	BC08	BC08	BC08	BC08	BC08	BC08
33	BA08	BA08	BB08	BB08	BB08	BB08	BB08	BA08	BA08	BA08	BB08	BB08	BB08	BB08	BB08	BA08
34	B809	B908	B908	BA08	BA08	BA08	B908	B908	B809	B908	B908	BA08	BA08	BA08	B908	B908
35	B709	B709	B809	**B908**	B908	B908	B809	B709	B709	B709	B809	B908	B908	B908	B809	B709
36	B509	B609	B709	B809	B809	B809	B709	B609	B509	B609	B709	B809	B809	B809	B709	B609
37	B409	B409	B509	B709	B709	B709	B509	B409	B409	B409	B509	B709	B709	B709	B509	B409
38	B30A	B309	B409	B609	B609	B609	B409	B309	B30A	B309	B409	B609	B609	B609	B409	B309
39	B10A	B20A	B309	B509	B509	B509	B309	B20A	B10A	B20A	B309	B509	B509	B509	B309	B20A
40	B00A	B10A	B20A	B409	B509	B409	B20A	B10A	B00A	B10A	B20A	B409	B509	B409	B20A	**B10A**
41	AF0A	AF0A	B10A	B309	B409	B309	B10A	AF0A	AF0A	AF0A	B10A	B309	B409	B309	B10A	AF0A
42	AE0B	AE0A	B00A	B20A	B309	B20A	B00A	AE0A	AE0B	AE0A	B00A	B20A	B309	B20A	B00A	AE0A
43	AD0B	AD0B	AF0A	B10A	B20A	B10A	AF0A	AD0B	AD0B	AD0B	AF0A	B10A	B20A	B10A	AF0A	AD0B
44	AC0B	AC0B	AE0A	B00A	B10A	B00A	AE0A	AC0B	AC0B	AC0B	AE0A	B00A	B10A	B00A	AE0A	AC0B
45	AB0B	AC0B	AE0B	AF0A	B00A	AF0A	AE0B	AC0B	AB0B	AC0B	AE0B	AF0A	B00A	AF0A	AE0B	AC0B
46	AA0C	AB0B	AD0B	AE0A	AF0A	AE0A	AD0B	AB0B	AA0C	AB0B	AD0B	AE0A	AF0A	AE0A	AD0B	AB0B
47	A90C	AA0B	AC0B	AD0B	AE0A	AD0B	AC0B	AA0B	A90C	AA0B	AC0B	AD0B	AE0A	AD0B	AC0B	AA0B
48	A80C	A90C	AB0B	AD0B	AD0B	AD0B	AB0B	A90C	A80C	A90C	AB0B	AD0B	AD0B	AD0B	AB0B	A90C
49	A70C	A80C	AB0B	AC0B	AC0B	AC0B	AB0B	A80C	A70C	A80C	AB0B	AC0B	AC0B	AC0B	AB0B	A80C
50	A60D	A80C	AA0B	AB0B	AB0B	AB0B	AA0B	A80C	A60D	A80C	AA0B	AB0B	AB0B	AB0B	AA0B	A80C
51	A60D	A70C	A90C	AA0B	AA0B	AA0B	A90C	A70C	A60D	A70C	A90C	AA0B	AA0B	AA0B	A90C	A70C
52	A50D	A60D	A80C	A90C	A90C	A90C	A80C	A60D	A50D	A60D	A80C	A90C	A90C	A90C	A80C	A60D
53	A40D	A60D	A70C	A80C	A80C	A80C	A70C	A60D	A40D	A60D	A70C	A80C	A80C	A80C	A70C	A60D
54	A40E	A50D	A60D	A70C	A70C	A70C	A60D	A50D	A40E	A50D	A60D	A70C	A70C	A70C	A60D	A50D
55	A30E	A40D	A60D	A60D	A60C	A60D	A60D	A40D	A30E	A40D	A60D	A60D	A60C	A60D	A60D	A40D
56	A20E	A40D	A50D	A50D	A60D	A50D	A50D	A40D	A20E	A40D	A50D	A50D	A60D	A50D	A50D	A40D
57	A20E	A30E	A40D	A40D	A50D	A40D	A40D	A30E	A20E	A30E	A40D	A40D	A50D	A40D	A40D	A30E
58	A10F	A30E	A30E	A30E	A40D	A30E	A30E	A30E	A10F	A30E	A30E	A30E	A40D	A30E	A30E	A30E
59	A10F	A20E	A20E	A30E	A30E	A30E	A20E	A20E	A10F	A20E	A20E	A30E	A30E	A30E	A20E	A20E
60	A00F	A10F	A10E	A20E	A20E	A20E	A10E	A10F	A00F	A10F	A10E	A20E	A20E	A20E	A10E	A10F
61	9F0F	A00F	A10F	A10F	A10F	A10F	A10F	A00F	9F0F	A00F	A10F	A10F	A10F	A10F	A10F	A00F
62	9F10	A00F	A00F	A00F	A00F	A00F	A00F	A00F	9F10	A00F	A00F	A00F	A00F	A00F	A00F	A00F
63	9E10	9F10	9F10	9F10	9F10	9F10	9F10	9F10	9E10	9F10	9F10	9F10	9F10	9F10	9F10	9F10
	-8	-7	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6	7

F.5 Static experiment photographs

The photographs of the FLC response on each image are presented below.

Image 1

left edge: 172

right edge: 375

displacement: -6

Rotation output value: 7802 h

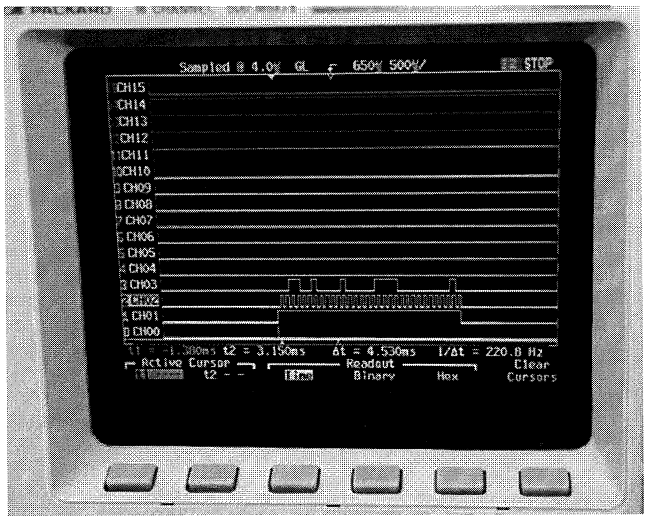


Image 2

left edge: 70

right edge: 103

displacement: -29

Rotation output value: 4207 h

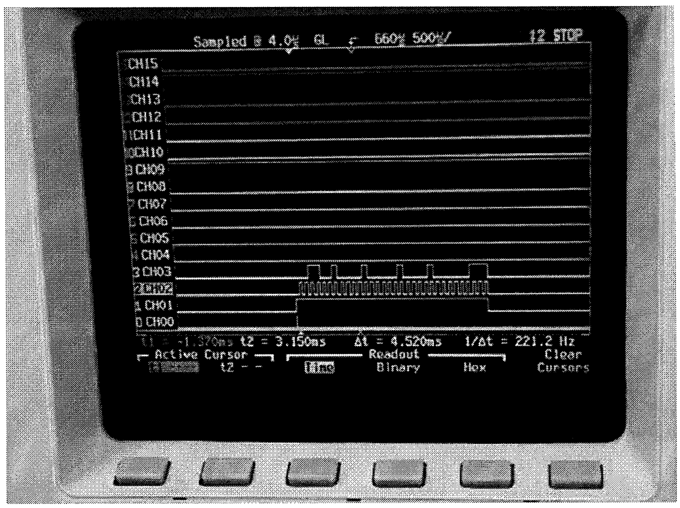


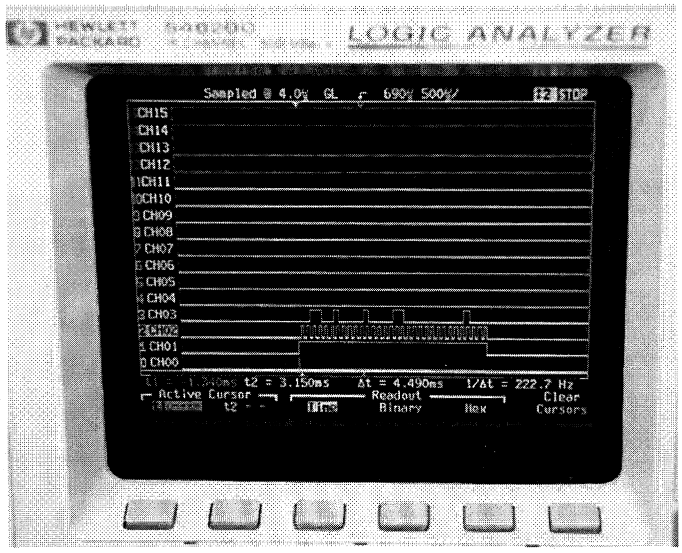
Image 3

left edge: 552

right edge: 572

displacement: 30

Rotation output value: C008 h



F.6 Dynamic experiment photographs

The photographs of the FLC response on each image are presented below.

Image 1

displacement: -30

Rotation output value: 4008 h

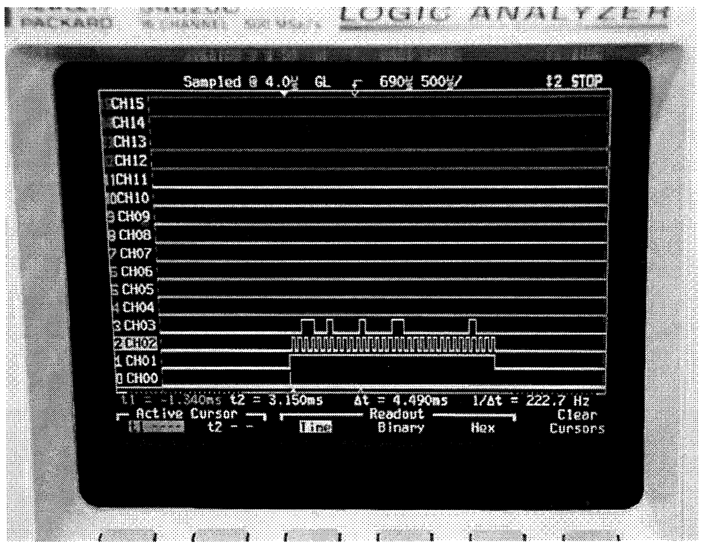


Image 2

displacement: -24

Rotation output value: 4B06 h

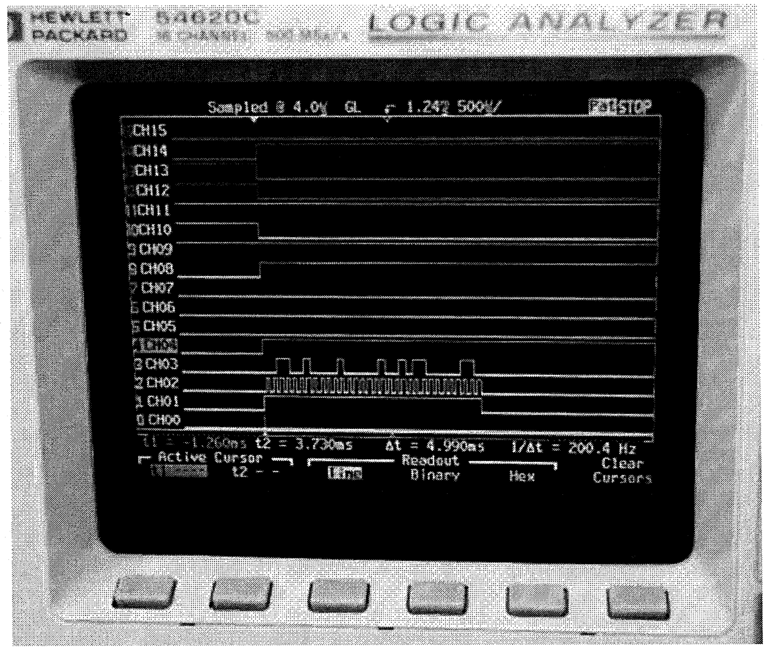


Image 3

displacement: -6

Rotation output value: 7802 h

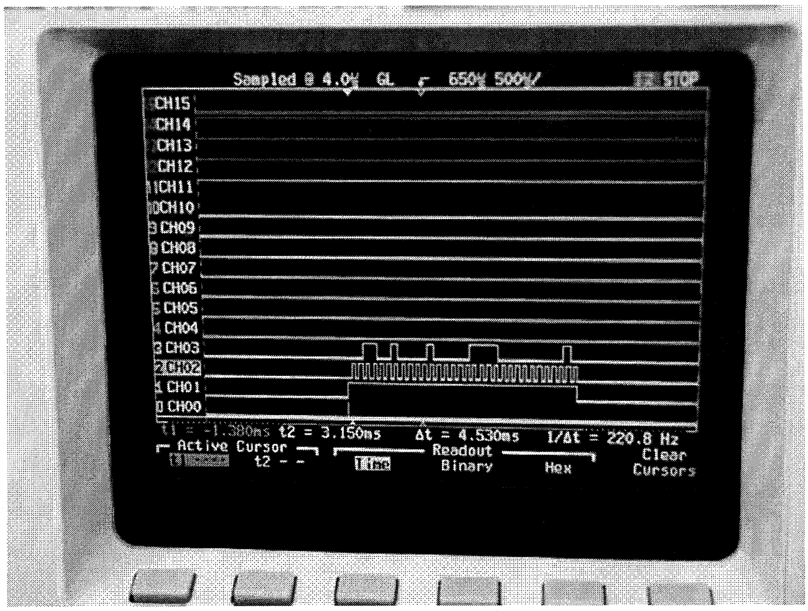


Image 4

displacement: 0

Rotation output value: 7800 h

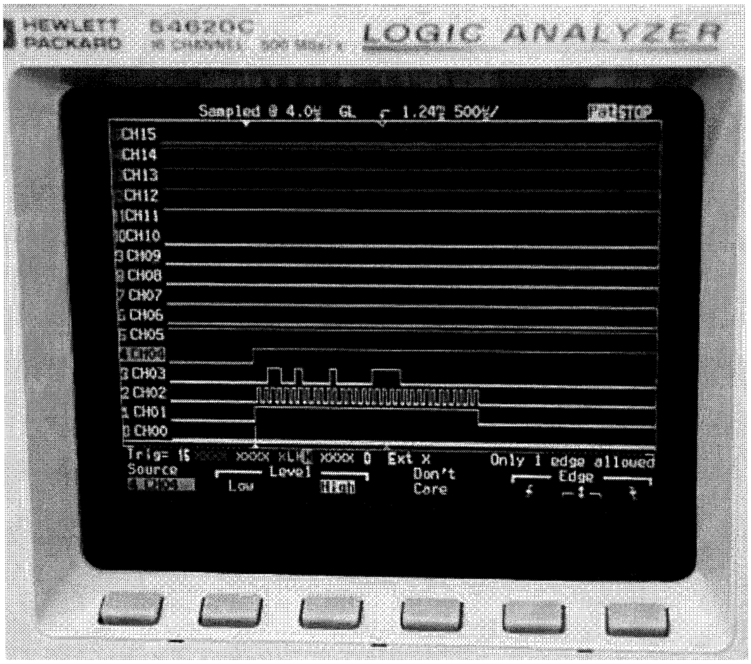


Image 5

displacement: 3

Rotation output value: F801 h

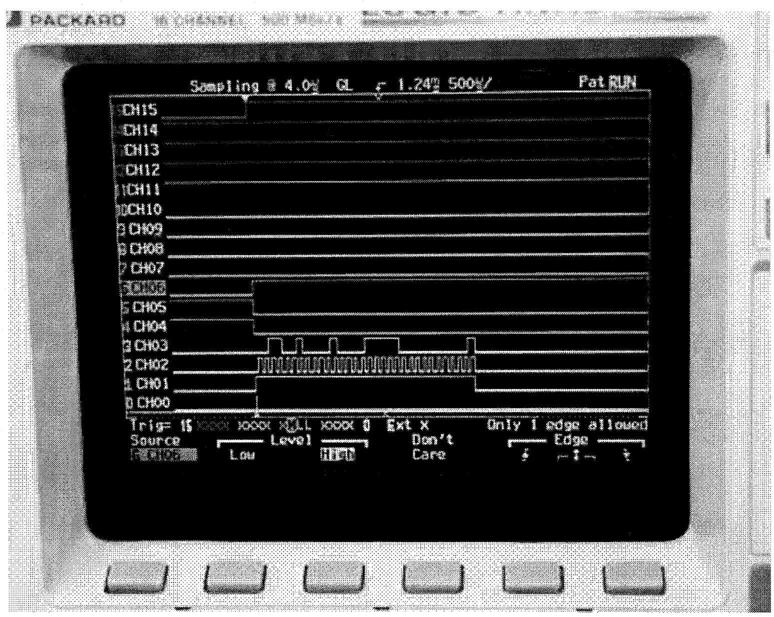


Image 6

displacement: 40

Rotation output value: B10A h

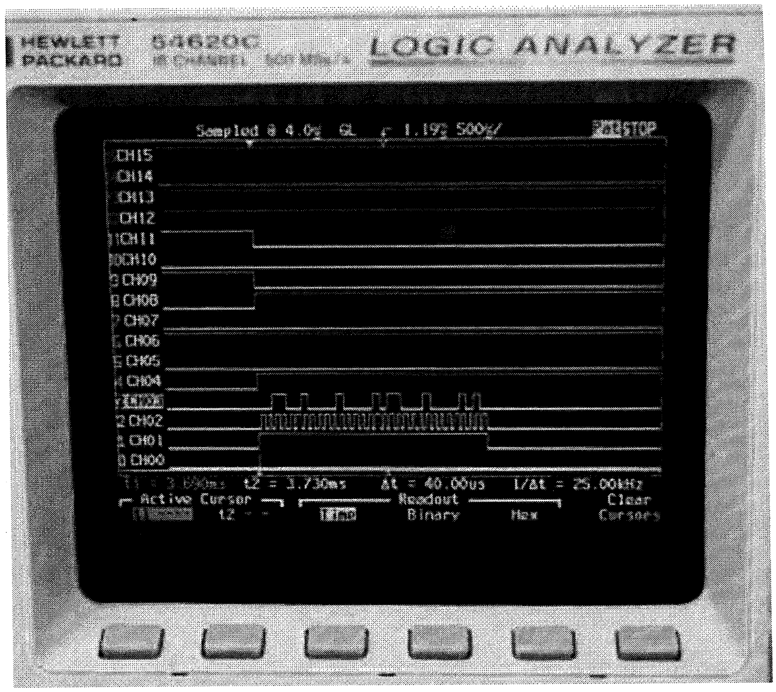


Image 7

displacement: 35

Rotation output value: B908 h

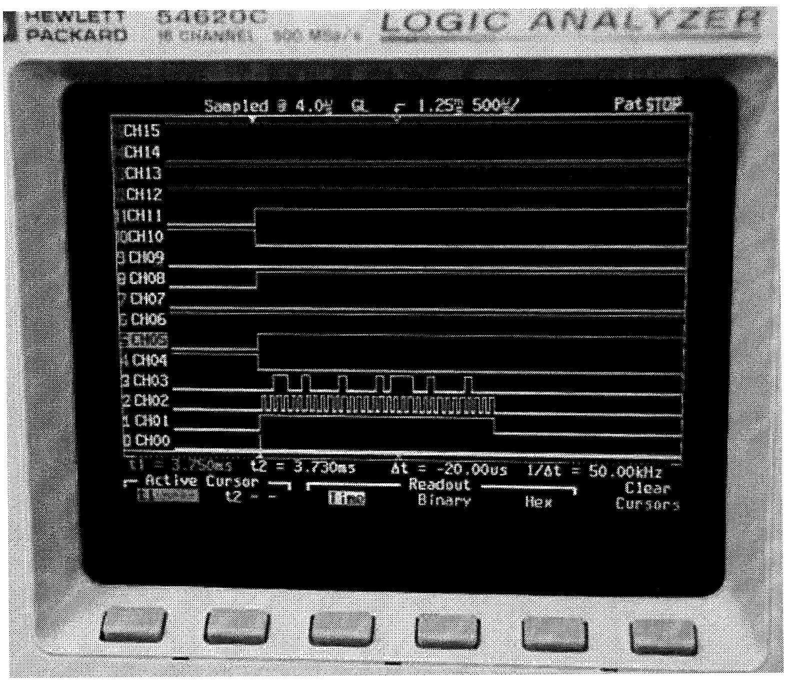
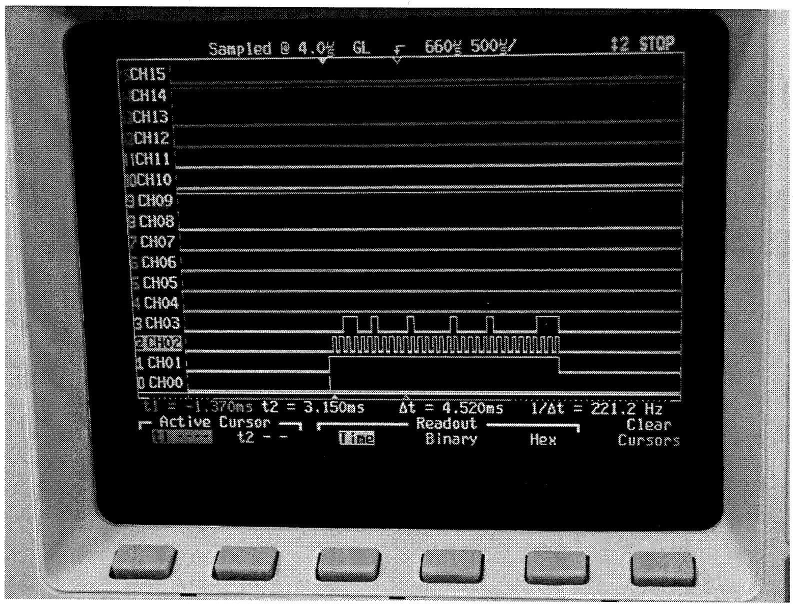


Image 8

displacement: 29

Rotation output value: C107 h



Bibliography

[1] http://www.ee.ryerson.ca/~lkirisch/research.htm

[2] http://sm.mdacorporation.com/

[3] I. J. Cox, S. L. Hingorani, S. B. Rao, "A Maximum Likelihood Stereo Algorithm", Computer Vission and Image Understanding, Vol. 63, No,3, May, pp. 542-567, 1996

[4] S. Sabihuddin, J. Islam, W. J. MacLean, "Dynamic programming approach to high frame-rate stereo correspondence: a pipelined architecture implemented on a field programmable gate array", Conference of Electricity and Computer Engineering, 2008

[5] C. J. Shields, "Aristotle", New York, Routledge, 2007

[6] R. L. Goodstein, "Boolean Algebra", Dover Publications; 1 edition, 2007

[7] S. Stewart, "Thinking about mathematics: the philosophy of mathematics", New York, Oxford University Press, 2000.

[8] J. Lukasiewicz, "Elements of Mathematical Logic", Pergamon Press; [2nd ed.] edition (1966)

[9] http://en.wikipedia.org/wiki/Lotfi_Asker_Zadeh

[10] L. A. Zadeh, "Fuzzy Sets, Fuzzy Logic, and Fuzzy Systems: Selected Papers by Lotfi A. Zadeh", Advances in Fuzzy Systems – Applications and Theory, vol. 6, Word Science Pub Co Inc, June 1996.

[11] H. T. Nguyen, E. A. Walker, "A First course in Fuzzy Logic", Chapman & Hall/CRC; 3rd edition, 2005

[12] D. Mcneill, "Europe Gets into Fuzzy Logic", Electronics Engineering Times, Nov 11, 1991

[13] A. Konar, "Computational Intelligence, Principles Techniques and Applications", Springer; 1 edition, 2005

[14] "Fuzzy Logic and Its Application to Technology and Management", Alpha Science International, 1 edition, 2007

[15] D. Simon, "Truth, American Culture, and Fuzzy Logic", Fuzzy Information Processing Society, 2006, NAFIPS 2006, Annual meeting of the North American, 3-6 June 2006, pp 430-435.

[16] J. Maturana, F. Saubion, "Towards a generic control strategy for evolutionary algorithms: An adaptive fuzzy-learning approach", 2007 IEEE Congress on Evolutionary Computation, CEC 2007, 2007 IEEE Congress on Evolutionary Computation, CEC 2007, 2008, p 4546-4553.

[17] J. J. Buckley, E. Eslami, "An Introduction to Fuzzy Logic and Fuzzy Sets", Physica-Verlag Heidelberg, 1 edition, 2002

[18] M. Bergmann, "An introduction to Many-Valued and Fuzzy Logic", Cambridge University Press; 1 edition, 2008

[19] V. Novak, I. Perfilieva, J. Mockor, "Mathematical Principles of Fuzzy Logic", Kluwer Academic Publishers, 1999

[20] Y. Bai, H. Zhuang, D. Wang, "Advanced Fuzzy Logic Technologies in Industrial Applications", Springer, 1 edition, 2006

[21] S.N. Sivanandam, S.Sumathi, S.N. Deepa "Introduction to Fuzzy Logic using MATLAB", Springer-Verlag Berlin Heidelberg, 2007

[22] M. Wygralak, "Cardinalities of Fuzzy Sets", Springer-Verlag Berlin Heidelberg 2003

[23] Y. Jin, "Advanced Fuzzy Systems Design and Applications", Physica-Verlag Heidelberg, 2003

[24] M. Hanss, "Applied Fuzzy Arithmetic: An Introduction with Engineering Applications", Springer, 1 edition, 2005

[25] R. Zhang, Y. A. Phillis, V. S. Kouikoglou, "Fuzzy Control of Queuing Systems", Springer, 1 edition, 2004

[26] K. Tanaka, H. O. Wang, "Fuzzy Control Systems Design and Analysis", John Wiley & Sons, Inc 2001

[27] K. Michels, F. Klawonn, R. Kruse, A. Nurnberger, "Fuzzy Control: Fundamentals, Stability and Design of Fuzzy Controller", Springer, 1 edition, 2006

[28] H.B Verbruggen, R. Babuska, "Fuzzy Logic Control: Advances in Applications", World Scientific Series in Robotics and Intelligent Systems – Vol. 23, 2001

[29] M.J. Burke, R. Downes, "A fuzzy logic based apnoea monitor for SIDS risk infants", Journal of Medical Engineering and Technology, v 30, n 6, Nov 1, 2006, p 397-411

[30] N. Kannathal, U. R Acharya, ; E.Y.K Ng,.; S.M. Krishnan, M. L. Choo, S. Laxminarayan, "Cardiac health diagnosis using data fusion of cardiovascular and haemodynamic signals", Computer Methods and Programs in Biomedicine, v 82, n 2, May, 2006, p 87-96

[31] J.S. Shieh, D.A. Linkens, A.J. Asbury, "A hierarchical system of on-line advisory for monitoring and controlling the depth of anesthesia using self-organizing fuzzy logic", Engineering Applications of Artificial Intelligence, v 18, n 3, April, 2005, p 307-316

[32] B. M. Ellingson, J. L. Ulmer, B. D. Schmit, "Morphology and morphometry of human chronic spinal cord injury using diffusion tensor imaging and fuzzy logic", Annals of Biomedical Engineering, v 36, n 2, February, 2008, p 224-236

[33] P.M. Pawar, R. Ganguli, "Fuzzy-logic-based health monitoring and residual-life prediction for composite helicopter rotor" Journal of Aircraft, v 44, n 3, May/June, 2007, p 981-995

[34] C. Nott, C. Sossamon, S,M. Olcmen, L.C. Trevino, "Jet-engine health monitoring and operation using fuzzy-logic controllers and bayesian belief networks", Collection of Technical Papers - 44th AIAA Aerospace Sciences Meeting, v 23, 2006, p 17710-17720

[35] L. Mokhnache, C. Kada, A. Boubakeur, A.N. Nait, "Condition monitoring of fuzzy logic system for oil insulated transformer diagnosis", International Journal of COMADEM, v 8, n 1, January, 2005, p 13-15

[36] S.S Panda, D. Chakraborty, S.K. Pal, "Monitoring of drill flank wear using fuzzy back-propagation neural network", International Journal of Advanced Manufacturing Technology, v 34, n 3-4, September, 2007, p 227-235

[37] M.B.I. Reaz, F. Choong, F. Mohd-Yasin, "VHDL modeling for classification of power quality disturbance employing wavelet transform, artificial neural network and fuzzy logic", Simulation, v 82, n 12, January, 2006, p 867-881

[38] J. Rodriguez, P. Vicente A. Antero, "Detection of stator winding fault in induction motor using fuzzy logic", Applied Soft Computing Journal, v 8, n 2, March, 2008, p 1112-1120

[39] A.J. Dounis, P. Tiropanis, G.P. Syrcos, D. Tseles, "Evolutionary fuzzy logic control of base-isolated structures in response to earthquake activity", Structural Control and Health Monitoring, v 14, n 1, February, 2007, p 62-82

[40] R. Stathacopoulou, M. Grogoriadou, M. Samarakou, D. Mitropoulos, "Monitoring students' actions and using teachers' expertise in implementing and evaluating the neural network-based fuzzy diagnostic model", Expert Systems with Applications, v 32, n 4, May, 2007, p 955-975

[41] T. Chaira, "Intuitionistic fuzzy set: Application to medical image segmentation", Studies in Computational Intelligence 85, 2008, pp. 51-68

[42] I. Guler, A. Tunca, E. Gulbandilar, "Detection of traumatic brain injuries using fuzzy logic algorithm", Expert Systems with Applications, v 34, n 2, February, 2008, p 1312-1317

[43] H. Ying, J. Yen, S. Zhu, D. C. Barth-Jones, R. E. Miller, R. M. Massanari, "A distributed adverse drug reaction detection system using intelligent agents with a fuzzy recognition-primed decision model", International Journal of Intelligent Systems, v 22, n 8, August, 2007, p 827-845

[44] H. Ressom, P. Natarajan, R. S. Varghese, M. T. Musavi, "Applications of fuzzy logic in genomics", Fuzzy Sets and Systems, v 152, n 1, May 16, 2005, Fuzzy Sets and Systems, p 125-138

[45] J. H. Graham, Y. Yu, "Computer system security threat evaluation based upon artificial immunity model and fuzzy logic", Conference Proceedings - IEEE Systems, International Conference on Systems, Man and Cybernetics, 2005, p 1297-1302

[46] H. Liu, H. Sheng, L. Li, Z. Yang, X. Huang, "An adaptive transmission control scheme based on TCP vegas in MANETs", Mobile Ad-hoc and Sensor Networks - Third International Conference, MSN 2007, Proceedings, 2007, p 387-396

[47] M. I. Chacon, J. L. Duran, L. A. Santiesteban, "A wavelet-fuzzy logic based system to detect and identify electric power disturbances", IEEE Symposium on Computational Intelligence in Image and Signal Processing, CIISP 2007, 2007, p 52-57

[48] S. R. Mounce, J. MacHell, J. B. Boxall, "Development of artificial intelligence systems for analysis of water supply system data", 8th Annual Water Distribution Systems Analysis Symposium 2006, 8th Annual Water Distribution Systems Analysis Symposium 2006, 2007, p 91

[49] Y. Kim, J. F. Reid, Q. Zhang, "Fuzzy logic control of a multispectral imaging sensor for in-field plant sensing", Computers and Electronics in Agriculture, v 60, n 2, March, 2008, p 279-288

[50] J. E. Naranjo, C. González, R. García, T. De Pedro, M. A. Sotelo, "Using fuzzy logic in automated vehicle control", IEEE Intelligent Systems 22 (1), 2007, pp. 36-45

[51] N. Yagiz, L. E. Sakman, R. Guclu, "Different control applications on a vehicle using fuzzy logic control", Sadhana - Academy Proceedings in Engineering Sciences, v 33, n 1, February, 2008, p 15-25

[52] A. A. Ferreira, J. A. Pomilio, G. Spiazzi, S. L. de Araujo, "Energy management fuzzy logic supervisory for electric vehicle power supplies system" IEEE Transactions on Power Electronics, v 23, n 1, January, 2008, p 107-115

[53] A. Poursamad, M. Montazeri, "Design of genetic-fuzzy control strategy for parallel hybrid electric vehicles", Control Engineering Practice, v 16, n 7, July, 2008, p 861-873

[54] Altas, H Ismail, "Fuzzy logic controlled tracking system for moving targets", IEEE International Symposium on Intelligent Control - Proceedings, 1997, p 43-48

[55] Chung, Kuo-Liang Yu, Shih-Wei; Yeh, Hsueh-Ju; Huang, Yong-Huai; Yao, Ta-Jen, "New fuzzy object segmentation algorithm for video sequences", Journal of Information Science and Engineering, v 24, n 2, March, 2008, p 521-537

[56] J. S. Cho, B. J. Yun, Y. H. Ko," Intelligent video tracking based on fuzzy-reasoning segmentation", *Neurocomputing* 70 (4-6), 2007, pp. 657-664

[57] J.M. Lucas, H. Martinez-Barbera, F. Jimenez, "Multi-objective evolutionary fuzzy modeling for the docking maneuver of an automated guided vehicle", IEEE International Conference on Fuzzy Systems, Proceedings of the IEEE International Conference on Fuzzy Systems, FUZZ-IEEE 2005, 2005, p 359-364

[58] K. L. Chung, S. W. Yu, H. J. Yeh, Y. H Huang, T. J. Yao, "Robust multi-object tracking under a wide range of real-world conditions", Proceedings - International Conference on Computational Intelligence for Modelling, Control and Automation, CIMCA 2005 and International Conference on Intelligent Agents, Web Technologies and Internet, 2005, p 815-820

[59] F. J. Montecillo-Puente, V. Ayala-Ramirez, A. Perez-Garcia, R. E. Sanchez-Yanez, "Fuzzy color tracking for robotic tasks", Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, v 3, 2003, p 2769-2773

[60] J. Figueroa, J. Posada, J. Soriano, M. Mclgarejo, S. Rojas, "A type-2 fuzzy logic controller for tracking mobile objects in the context of robotic soccer games", IEEE International Conference on Fuzzy Systems, Proceedings of the IEEE International Conference on Fuzzy Systems, FUZZ-IEEE 2005, 2005, p 359-364

[61] O. Castillo, P. Melin, "Type-2 Fuzzy Logic: theory and Applications", Studies in Fuzziness and Soft Computing, Springer-Verlag Berlin Heidelberg, 2008

[62] Y. Bai, H. Zhuang, Z. S. Roth, "Fuzzy logic control to suppress noises and coupling effects in a laser tracking system", IEEE Transactions on Control Systems Technology, v 13, n 1, January, 2005, p 113-121

[63] W. Gierusz, V. C. Nguyen, A. Rak, "Maneuvering control and trajectory tracking of very large crude carrier", Ocean Engineering, v 34, n 7, May, 2007, p 932-945

[64] A. Rubaai, D. Ricketts, M. D. Kankam, "Laboratory implementation of a microprocessor-based fuzzy logic tracking controller for motion controls and drives", Industry Applications, IEEE Transactions on Volume 38, Issue 2, March-April 2002 Page(s):448 – 456

[65] H. Abdelkrim, S. Ben Saoud, "Implementation of a fuzzy logic tracking path algorithm on a field programmable gate array", Design and Test of Integrated Systems in Nanoscale Technology, 2006. DTIS 2006. International Conference on 2006 Page(s):355 – 358

[66] Allen, John S. Nguyen, Eric M.; Timms, Steven; Prasad, Nadipuram R. "Design of an automatic focus control unit using fuzzy logic", Proceedings of SPIE - The International Society for Optical Engineering, v 2755, 1996, p 363-374

[67] H. Najafi, G. Klinker, "Model-based tracking with stereovision for AR", Mixed and Augmented Reality, 2003, Proceedings, The Second IEEE and ACM International Symposium on 7-10 Oct. 2003 Page(s):313 - 314

[68] Kao, Han-Ying, "Diagnostic reasoning and medical decision-making with fuzzy influence diagrams", Computer Methods and Programs in Biomedicine, v 90, n 1, April, 2008, p 9-16

[69] A.B. Rakityanskaya, A.P. Rotshtein, "Fuzzy relation-based diagnosis", Automation and Remote Control, v 68, n 12, December, 2007, p 2198-2213

[70] R. Blatt, A. Bonarini,E. Calabro, M. D Torre, M. Matteucci, U. Pastorino, "Fuzzy k-NN lung cancer identification by an electronic nose", Applications of Fuzzy Sets Theory - 7th International Workshop on Fuzzy Logic and Applications, WILF 2007, Proceedings, 2007, p 261-268

[71] S. Arita, T. Nomura, H. Sonoo, "Diagnostic system of breast cancer based on imaging data of mammography using fuzzy logic", 2006 World Automation Congress, WAC'06, 2006 World Automation Congress, WAC'06, 2007, p 4259849

[72] R. Radha, S. P. Rajagopalan, "Fuzzy logic approach for diagnosis of Diabetics", Information Technology Journal, v 6, n 1, Jan 1, 2007, p 96-102

[73] K. Polat, S. Kara, F. Latifoglu, S. Gunes, "A novel approach to resource allocation mechanism in artificial immune recognition system: Fuzzy resource allocation mechanism and application to diagnosis of atherosclerosis disease", Artificial Immune Systems - 5th International Conference, ICARIS 2006. Proceedings, 2006, p 244-255

[74] M. Akbarzadeh-T, M. Moshtagh-Khorasani, "A hierarchical fuzzy rule-based approach to aphasia diagnosis", International Journal of Automation and Computing, v 4, n 2, April, 2007, p 135-140

[75] Y. Guang, W. Xiaoping, "Synthesized fault diagnosis method based on fuzzy logic and D-S evidence theory", Advanced Intelligent Computing Theories and Applications: With Aspects of Artificial Intelligence - Third International Conference on Intelligent Computing, ICIC 2007, Proceedings, 2007, p 1024-1031

[76] P. Bilski, J.M. Wojciechowski, "Automated diagnostics of analog systems using fuzzy logic approach", IEEE Transactions on Instrumentation and Measurement, v 56, n 6, December, 2007, p 2175-2185

[77] W. H. Gui, C. H. Yang, J. Teng, "Intelligent fault diagnosis in lead-zinc smelting process", International Journal of Automation and Computing, v 4, n 2, April, 2007, p 135-140

[78] Z. Zheng, S. Xinyu, Y. Daoyuan, "Fault diagnosis of a wheel loader by artificial neural networks and fuzzy logic", IEEE Conference on Robotics, Automation and Mechatronics, 2006, p 4018820

[79] J. D. Wu, Y. H. Wang, M. R. Bai, "Development of an expert system for fault diagnosis in scooter engine platform using fuzzy-logic inference", Expert Systems with Applications, v 33, n 4, November, 2007, p 1063-1075

[80] Y. H. Chang, K. Coyne, A. Mosleh, "A nuclear plant accident diagnosis method to support prediction of errors of commission", IEEE Conference on Robotics, Automation and Mechatronics, 2006, p 4018820

[81] W. Ba, D. Zhang, "A novel dynamic fuzzy threshold preemption scheduling algorithm for soft real-time systems", Proceedings of the IEEE Conference on Decision and Control, Proceedings of the 46th IEEE Conference on Decision and Control 2007, CDC, 2008, p 4584-4589

[82] N. Bose, K. N. Senthil , "Prediction of engine emissions through fuzzy logic modeling", Proceedings - International Conference on Computational Intelligence and Multimedia Applications, ICCIMA 2007, 2008, p 420-428

[83] I. B Topcu, M. Saridemir, "Prediction of compressive strength of concrete containing fly ash using artificial neural networks and fuzzy logic", Computational Materials Science, v 41, n 3, January, 2008, p 305-311

[84] S. Ertugrul, "Predictive modeling of human operators using parametric and neuro-fuzzy models by means of computer-based identification experiment", Engineering Applications of Artificial Intelligence, v 21, n 2, March, 2008, p 259-268

[85] A. Meel, W. D. Seider, "Real-time risk analysis of safety systems", Computers and Chemical Engineering, v 32, n 4-5, Apr 5, 2008, p 827-840

[86] S. Singh, A. Roy, K. S. Rattan, J. M. Emmert, "Performance trade-offs for hardware/software implementation of a fuzzy logic controller on programmable hardware", Annual Meeting of the North American Fuzzy Information Processing Society, 2005, p 484-488

[87] Kim, Daijin , "Implementation of fuzzy logic controller on the reconfigurable FPGA system", IEEE Transactions on Industrial Electronics, v 47, n 3, Jun, 2000, p 703-715

[88] P. T. Vuong, A. M Madni, J. B. Vuong, "VHDL implementation for a fuzzy logic controller", World Automation Congress, WAC'06, 2007, p 4259884

[89] L. Kirischian, "Architectural Synthesis and Design of Digital Systems", ELE-8208 course, Ryerson University, 2007

[90] Xilinx user guide, "Virtex-II Pro and Virtex-II Pro X FPGA User Guide", November 2007