Fuzzy-neural robot motion control system

Author: Oktay Alizada

Supervisor: Dr. John A. Bullinaria

MSc Computer Science

University of Birmingham

September 2013

Abstract

Nowadays, fuzzy systems is becoming one of the most important direction especially in scientific area. One of the main implementation of fuzzy systems is Fuzzy Logic Controller for controlling motion of robotic systems. In this dissertation fuzzy logic controller implemented, which carries action to reach the desired target with avoiding obstacles, with various intelligent layers and simulated. Fuzzy Logic Controller advantages relatively to other system are shown. The theory of avoiding moving obstacle proposed. Doppler effect sensor technology for measurement of moving obstacles' speed and Ultrasonic sensors for distance measurement are used.

Acknowledgement

I would to thank you my supervisor Dr. John A. Bullinaria for all of his support, assistance and valuable comments during our meetings in these three months of research and study of my summer project.

I would also like to thank Dr. Alan Sexton as my inspector for his comments during my demonstration.

Contents

Abstract……………………………..……………………………………………………..1

Acknowledgement…………..………………………………………………………… 2

Contents………………..……...…………………………………………………… .3

List of Figures………………………………………………………………… .. .5

1. Introduction…………………………………………….……………………......7

    1.1 Progress……………………………………………………………………..…8

    1.2 Aim…………………………………………………………………………...9

    1.3. Problem……………………………………………………………………9

    1.4. Report structure………………………………………………………..10

2. Background Research…………………………………….……………….. 11

    2.1 Fuzzy Systems………………………………………………………………14

3.Fuzzy Logic……………………………………….....……………………..15

    3.1 Definitions and terminologies………………………………………………15

    3.2. Comparison with PID controller……………………………….…………..23

4. Project Specification…………………………………………………………..26

    4.1 Functional Requirements……………………………………………………...26

    4.2 Non-Functional requirements………………………………………………26

5. Mathematical Implementation…………………………………………………..27

5.1 Model of the system……………………………………………………………..27

5.2 Kinematics of the System…………………………………………………..29

5.3 Dynamics of the System ……………………………………………………32

6. Software Implementation ……………………………………………………..38

6.1 Fuzzy Logic Controller……………………………………………………..38

6.2 Ultrasonic sensors for detecting obstacles…………………………………..44

6.3 Obstacle avoiding behaviour………………………………………………..44

6.3.1. Static Obstacles avoidance…………………………………………..44

6.3.2 Moving Obstacles avoidance…………………………………………51

6.4 Testing…………………………………………………………………………57

6.5 Evaluation………………………………………………………………...58

6.6 Future Works…………………………………………………………………..59

7.Conclusion……………………………………………………………………..60

Reference…………………………………………………………………………...61

Appendix…………………………………………………………………………...62.

List of Figures

Figure 1: General structure of intelligent controlling system………………………11

Figure 2. Membership functions for fuzzy sets of linguistic variable w …………….21

Figure 3. Membership functions for fuzzy sets of linguistic variable y………………21

Figure 4. Activation step………………………………………………………………22

Figure 5. Acccumulation………………………………………………………………22

Figure 6. Falling angle response for fuzzy control……………………………………23

Figure 7. Falling angle response for conventional PID……………………………….24

Figure 8. Falling angle response for fuzzy control…………………………………….24

Figure 9. Falling angle response for conventional PID control……………………….25

Figure 10. Scheme of modelling the robot control system……………………………27

Figure 11. Kinematics of mobile robot………………………………………………...29

Figure 12. Inputs of fuzzy controller 1st type………………………………………….38

Figure 13. Inside fuzzy logic controller………………………………………………..39

Figure 14.splitted fuzzy controller(abstraction)………………………………………..40

Picture 15. Distance input membership function………………………………………41

Figure 16. Linear velocity output membership function………………………………..41

Figure 17. angle input membership function……………………………………………42

Figure 18. angular velocity output membership function………………………………43

List of Figures

Figure 19. Overview of the splitted fuzzy controller type 2(abstraction)………………..45

Figure 20. Fuzzy logic controller 2nd type…………………………………………….45

Figure 21. DistanceObs Membership function…………………………………………46

Figure 22. Membership function for input angleObs……………………………………47

Figure 23. Overview of the system……………………………………………………….53

Figure 24. Membership function for input G-ε...……………………………………………55

Figure 25. Kinematics of the system……………………………………………………..56

Figure 26. The entire model in Simulink;………………………………………………..57

Figure 27. Goal reahing behavior………………………………………………………..57

Figure 28. Obstacle avoidance behaviour……………………………………………….58

1. Introduction

Nowadays, robotics systems have been actively implemented in all spheres of human activity. This trend is dictated by the rapid rise in the development of computer and microprocessor technology. The development of technology requires new production areas, so number of the threats, which can happen, is increasing. As a result of these emergencies, significant part of emergency rescue operations for their removal to be conducted under pollution with radioactive and chemical substances. In carrying out such works an essential objective is to reduce risk to the life of rescuers. The solution is the development of more wide capable applications, which are increasing modern efficiency of robotic systems. Specificity of operations, the terms of operation and functional purpose of a mobile robot determine its structural features and the complexity of the control system, and the weight and size special-purpose equipment. Thus it is important to create application and mechanisms which can handle those processes and correctly respond for them. The paper will present the robot system which can move in this environment towards to the object by avoiding obstacles on the path. An important role in this case is the problem of creating intelligent control system that allows the robot to autonomously perform the task with minimal human intervention.

In some researches, mobile robots are belong to one operator-human which is controlling the robot movement and velocities. These appproaches consider the system that mobile robot has not got self-behavior and intelligent.Therefore they have number of drawbacks:

-the need for permanent interaction between person and robot and if the cable or device is damaged the goal will not be obtained. Furthermore, it is decreasing efficiency of the system by decreasing the speed of robot because of information processing.

-during the motion of robot, person is using information, gained from robot sensors and trying to choose next stage of operation. Person is not always able to correctly analyse the data from cameras and activate necessary behaviour.

-Delay in activating behavior can lead to serious problems especially in unknown environment(for example collision with moving obstacle)

These drawbacks can be avoided if the person(operator) is not performed at movement stage but goal reaching stage. In this case, the robot must independently or with minimal intervention perform the assigned tasks. The relevance of project lies in the fact that the proposed architecture of the control system will improve the efficiency of performed works and expand scope applications mobile robots by providing them with autonomy in the partially nondeterministic conditions. The system which will be implemented is based on modelling mobile robot on the cheapest way and can be easily updated to perform new tasks. The paper also presents moving obstacle algorithm which can be applied to systems operated on very complex environments. Many applications of mobile robots, in particular task management of autonomous mobile robots, have high uncertainty environment that requires the system management ability to make decisions.

The proposed system is using Fuzzy Logic Control system for moving robot towards the target in particially-known and unknown environments. With this approach we do not need to use complex algorithms to perform tasks and if we need to add new behavior we are able to add very easily. The dissertation also provides relevant sensors which data will be collected and used by fuzzy logic controller.

1.1 Progress

The project was able to give me to use several strategies in one problem to evaluate their performance. The gain of these project to use several architectures and several subjects as Physics, Mathematics, Programming at the same time and compare my approach with different researches which are done todays. The technologies and tools which are learned and used for created the system are following:

1. Matlab
2. Simulink
3. FCL-Fuzzy Control Language
4. Ultrasonic sensors
5. Visual Basic

6. Robotics applications such as SimRobot and Webots

1.2 Aims

The aim of the thesis is the development of architecture, algorithms and software of Intelligent Systems control of mobile robots based on the method of fuzzy logic and sonic wave mechanism for autonomous execution tasks. The work was set and solved the following problems:

1. Development of the Doppler effect speed measurement sensor.
2. Development of a generalized architecture and the control algorithm mobile robots based on the method of fuzzy logic;
3. Development of a method of obstacle detecting in the cheapest way, allowing implement tracking of moving objects in real time;
4. Simulation, carrying out independent motion control of mobile robot based on the method of fuzzy logic;
5. Development of the algorithm for moving obstacle avoiding system

1.3 Problem

Development of a control system for a mobile robot that is a complex, dynamic system acting in conditions uncertainties and external disturbances operating environment, requires attract new , non-traditional approaches using methods processing knowledge new types feedback ties modern intellectual, information and telecommunication technologies. First of all, the construction of intellectual control system need a clear understanding of the specifics of the problem at the level of basic concepts. The following tasks are important to be solved:

1. Control speed of robot to reach target
2. Moving in unknown and partially known environment

3. Obstacle avoiding mechanisms(moving and static)
4. The cheapest implementation of robot
5. System which can be easily modifiable.

1.4 Report Structure

2. Background Research- the research which has been done to solve some of the problems listed above(section 1.3 Problems)

3. Fuzzy logic this chapter about fuzzy logic terminology and specification of fuzzy logic

4. Project specification. This chapter is about requirements which is used to develop system such as functional and non-functional reuqirements

5. Mathematical Implementation. This chapter is about Mathematical model of the robot Kinematics and Dynamics

6. Software Implementation. Fuzzy Logic Controller implementation in matlab, behaviours of the system and Sensor information. Achievements due the project and Evaluation.

7. Conclusion

2. Background Research

First of all, the obvious task for understanding is the term "intelligent". Intelligent systems are systems which can process knowledge. The absence of a full-scale clarification of the term " intelligent" due to difficulties definition of knowledge.

Knowledge – is proven by practical results, action of understanding the problem which can be used to solve it;[4]

Thinking - the process of reflection of objective reality in views, opinions, concepts ;[4]

The concept - a modern logically thought about a class of objects.[4]

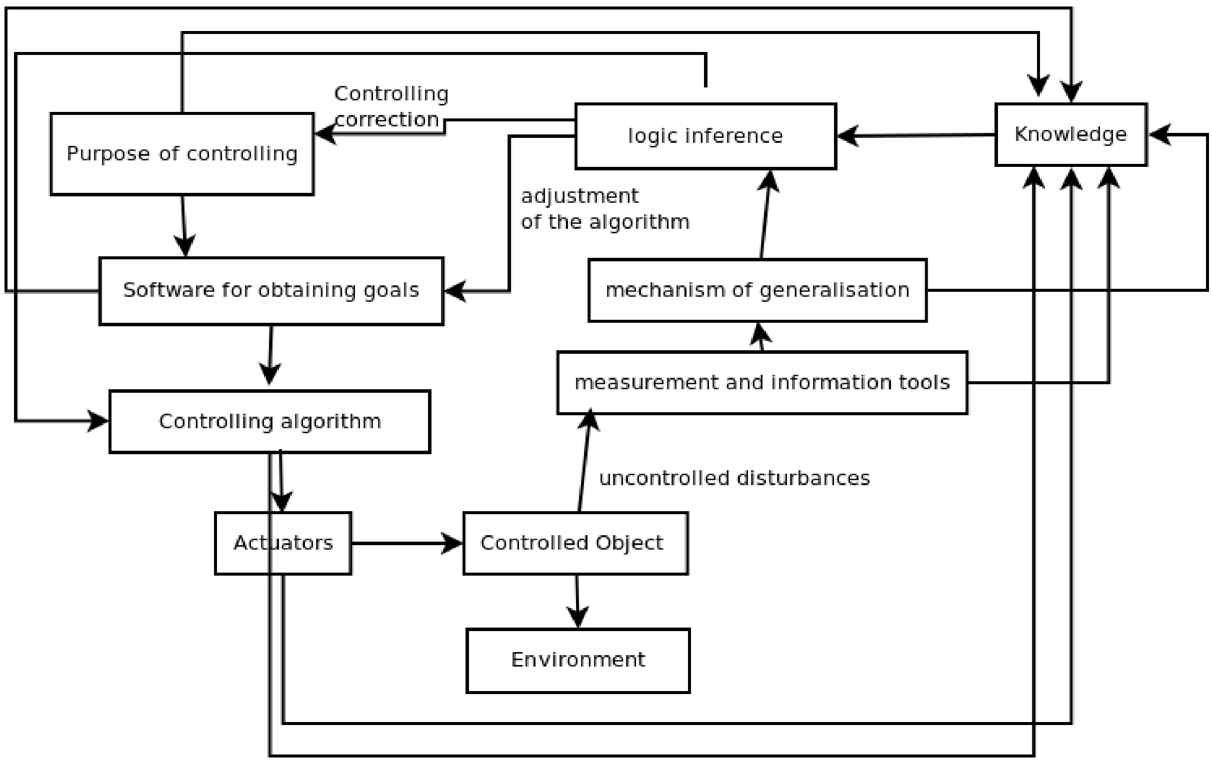


Figure 1. General structure of intelligent controlling system

There are several systems which can be considered as intelligent systems.

-expert systems technology ;

- fuzzy logic technology ;

- neural network technology ;

- associative memory technology .

Expert systems is a software, used expert knowledges for providing high effficiency solutions for unformalized problems in some area. The expert systems generally based on Knowledge Base which formed according to subject area. Knowledge Base is generated during the process of building and using the Expert System[2].

Expert system has several properties:

1. Application for solving a problem regarding to experience
2. Prognostic tools exist, Expert System is showing how results is changing in various situations.
3. Institutional memory based on Knowledge Base, which will be the set of opinions and permanently updated strategies and methods, used by users. The experts can leave but knowledge will remain.
4. Posibility to use Expert systems to teach new staff.

But in real word the problems are not static so new approach of expert system is shown as a dynamical Expert systems. Comparing with static Expert systems, Dynamical Expert systems have two properties

- Subsystem to model theoutside world
- Subsystem interface with the outside world

The disadvantages of the Expert System is that they are related to some limited areas where expert knowledge is important to get some results. Furthermore, Expert Systems are very heavily changeable.

Artificial Neural Networks are computer structures which model associated biological processes, which are related to human brain processes.Neural Networks are parallelized systems capable of learning by analysis of positive and negative effects[3]

The meaning of Neural Networks leaning is for all input variables giving a set of outputs. Each of inputs are presented as a vector[4].The training input is carried out by sequential presentation of the input simultaneous adjusting the weights in accordance with a specific procedure.

Assosiative memory- hardware support for intelectual systems of managing, which always operates with Database, is associative memory.

Associative memory has following properties[5]:

-Operations in memory are relevant to groupt or all elements;

-Operations in memory is done simultaneously to all element, which should be updated

-The main operation in memory is search and comparing

-time needed to search does not depend on number of saved data.

There are several types of associative memory organisation:

1. Memory with paralel access
2. Memory with sequential processing of digits
3. Memory with sequential processing of words
4. Memory organized by blocks

2. 1 Fuzzy systems

Fuzzy System is fuzzy logic based system which that takes input variables in interval [0,1]. Fuzzy Systems takes important role in controlling applications. Some reaserches [14], indicates to use layered structure of fuzzy sets, by appliying new behaviour as a separate system. For example, desired target is indicated and mobile robot is moving towards to target when the sensor is detecting the obstacle new fuzzy logic controller is activating to handle the obstacle and activate avoidance strategy.

Another research [15], refers to get information from all sensors as a several inputs. In detailed, each sensor is an input for fuzzy logic controller. If left sensor is detecting obstacle the value input of the left sensor is changing and that indicates to the system that left sensor detected obstacle.

Another research[16], refers the robot with digital image processing to detect moving obstacles. Digital camera using some filters give related information(input) to the fuzzy logic controller to operate on it.

3. Fuzzy Logic

Fuzzy logic is a form of many-valued logic; it deals with reasoning that is approximate rather than fixed and exact. Compared to traditional binary sets (where variables may take on true or false values) fuzzy logic variables may have a truth value that ranges in degree between 0 and 1. Fuzzy logic has been extended to handle the concept of partial truth, where the truth value may range between completely true and completely false. [1].

During the past several years, fuzzy control has emerged as one of the most active and fruitful areas for research in the applications of fuzzy set theory, especially in the realm of industrial processes, which do not lend themselves to control by conventional methods because of a lack of quantitative data regarding the input-output relations[3].

In general, the purpose of controlling based on analysis of the current state of the object to determine the desired control behavior, the implementation of which is to determine the desired control management. Nowadays, for solving related tasks various algorithms has been defined for finding optimal rules for managing objects with different physical nature.

One of the most important practical use of fuzzy systems is tasks of management and controlling different objects or processes. In this case, based on formal viewing of characteristics exploring system in terms of linguistic variables.

Linguistic variable in the theory of fuzzy logic is variable which can take arguments as words or sentences from natural language. For example, water can have values such as “hot”, “very hot”, “very cold”, “cold”. There words also indicates fuzzy sets of variables.

3.1 Definitions and terminologies

The formulization of using fuzzy variables based on theory of fuzzy sets. “Fuzzy sets- are sets, elements of which has degrees of membership”. [1]

Membership functions are the most widely used in application. “Membership function for a fuzzy set A on the universe of discourse X is defined as μ_A:X → [0,1], where each element of X is mapped to a value between 0 and 1. This value, called membership value or degree of membership, quantifies the grade of membership of the element in X to the fuzzy set A”. [1]

Membership functions takes values in range [0, 1].

$$\mu_A(x) \epsilon [0, 1] \; \forall x \in X,$$

where A ϵ X- set A is a subset of universal set X. $\mu_A(x)$-membership function of set A.

$$\mu_{A\cap B}(x)= \mu_A(x) \wedge \mu_B(x)$$

$$\mu_{AUB}(x)= \mu_A(x) \vee \mu_B(x)$$

. $$\mu_{\bar{A}} (x)= . 1- \mu_A(x)$$

Where is $\mu_A(x)$:X→[0,1], $\mu_B(x)$:X→[0,1]

Furthermore, logic operation like disjunction and conjuction can also be done with fuzzy sets. İn this case we use minimum interpolation of these operations,

$$\mu_A(x) \wedge \mu_B(x)=\min_x\{\mu_A(x), \mu_B(x)\}$$

$$\mu_A(x) \vee \mu_B(x)=\max_x\{ \mu_A(x), \mu_B(x)\}$$

One of the most important consepts of fuzzy sets are fuzzy relations. Fuzzy relations between several sets is the fuzzy subset of direct multiplication of these sets. Fuzzy relation is given by the relevant membership function, which showing direct multiplication of sets.

$$\mu_R(x, y): X*Y\rightarrow[0,1], x \in X, y \in Y$$

where $\mu_R(x, y)$- membership function of fuzzy relation R, X,Y- arbitrary sets

The considered logic operations over the fuzzy sets are based on mechanisms of logic inference, on which we are consistently aplying fuzzy rules. In applications rules are defined in the following way

If (W is A) then (Z is B),

W-is input variable, Z-is output variable. A, B fuzzy sets, each of them described by membership functions, and the assertion of type (X is A)-fuzzy assertion. List of these rules is called rule-base and showing the connection between input and output parameters of the system, which is building logic-linguistic model of the system. In each rule there can be several fuzzy statements of inputs and outputs. These staments can be connected by logic functions, as, and, or, not. Statements on the left side of **THEN** are called preconditions, on the right side called conclusions. For example:

IF(W_1 is A_{11}) and ….and (W_n is A_{1n}) THEN(Z_1 is B_{11}) and … (Z_k is B_{1k})

IF(W_1 is A_{21}) and ….and (W_n is A_{2n}) THEN(Z_1 is B_{21}) and … (Z_k is B_{2n})

………..

IF(W_1 is A_{t1}) and ….and (W_n is A_{tn}) THEN(Z_1 is B_{t1}) and … (Z_k is B_{tk})

Where n, k- is number of input and outputs of linguistic variables: W_i and Z_i(where i=1,2,..,n; j=1,…k);

T-number of rules in linguistic model,

$A_{ti} \in X_i$ $B_{tj} \in Y_j$- linguistic values of input and output variables respectively;

X_i V_j set of linguistic values, given for each used variable.

Therefore, all parameters of the model are described by proper linguistic variables, values of which are given by fuzzy terms. Creating and applying fuzzy inference systems consist of several steps, realising of which is done by main statements of fuzzy logic.

Data, which is the input of the fuzzy inference system, are somehow measured input variables. These variables are real variables of controlling the process. Information, which is formed on the output of the system, is control variables of the process.

The main steps of the fuzzy inference system:

-fuzzification of the inputs

-aggregation of conditions in fuzzy rules

-activation or composition of the conclusions in the fuzzy rules;

-accumulation conclusions fuzzy rules

-defuzzification of the outputs

"In fuzzy inference systems linguistic variables, which are used in fuzzy statements on the left side are called input lingustic variables. Variables, on the right side of the fuzzy statement called output linguistic variables"[3].

"Rule base is a finite number of fuzzy rules, agreed the use of linguistic variables" [1]

In the context of fuzzy logic, fuzzification not only separate step of fuzzy inference system, but also the process to find values of membership functions of fuzzy sets, based on simple(not fuzzy) data. The purpose of fuzzification is a process to find relation between the ordinary value of input variable and value of membership function and fuzzy set of input linguistic variable related to it.

After finishing this stage, concrete values of membership function, for each of linguistic fuzzy sets, should be assigned to all input variables

Aggregation operations on fuzzy sets are operations by which several fuzzy sets are combined in a desirable way to produce a single fuzzy set. "The inputs of the operation is a set of membership functions, aggregation process is taking these inputs and combining it to one fuzzy set for each output variable".[7]

Activation is a process to find truth degree for each conclusions of fuzzy rules. The step of activation is considered to be finished, when for all output linguistic variables, included in different conclusions of fuzzy rules, membership functions are defined.

Accumulation- in fuzzy inference systems is the procedure for finding membership functions for each output linguistic variables of fuzzy set.

The purpose of accumulation is joining or acccumulate all truth degree of conclusions for obtaining membership function for all output variables. The reason of doing this step is conclusions, which are related to the same output linguistic variable, can belong to different rules of fuzzy inference system.

The stage of accumulation is considered to be finished, when for each output linguistic variable, summarized membership function will be determined.

Defuzzification in the fuzzy inference systems is the process or procedure of finding an ordinary(not fuzzy) values for all output linguistic variables.

The purpose of defuzzification is to use results of accumulation for all output variables, to get an ordinary quantitive value for each output variable, which can be used by special devices, outside of fuzzy inference systems.

Nowadays, devices and mechanisms in real-world applcations are taking only traditional commands described in numerical and quantitive way. The stage of defuzzification is considered to be finished, when for each of output linguistic variable, final numerical values will be determined. These values will be real number, ie u_1, u_2, …, u_n, where n-is number of linguistic outputs.

There are many methods for doing defuzzification procedure. The most famous are followings:

1. Method of gravity
2. Center of area method
3. Mean of Maxima
4. Middle of Maximum

The aggregation of different methods described above is determining the algorithm of fuzzy inference system. “There are many algorithms Tsukamoto, Takagi-Sugeno, Mamdani, Larsen and etc”.[8]

Mamdani algorithm is the most popular algorithms for fuzzy inference systems which is related to control systems:

Rule base in the example will be:

P_1: IF (w is A_{11}) THEN y is B_1

P_2: IF (w is A_{21}) THEN y is B_2

P_3: IF (w is A_{31}) THEN y is B_3

P_4: IF (w is A_{41}) THEN y is B_4

P_5: IF (w is A_{51}) THEN y is B_5

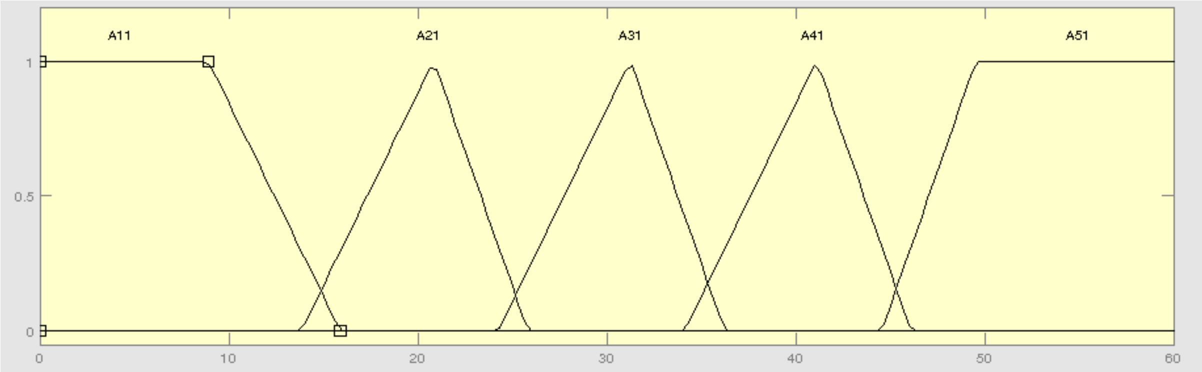
Where P_i- is rule number in rule base; W_i-input variables, A_{ij}-the value of input linguistic variable; y-output variable; B_i- the value of output linguistic variable; i,j=1,2.

First Step. Fuzzification. Determining of truth degree of each precondition and each rule for given values of input variables $\mu_{Aij}(w'_j)$.

As a fuzzy set of first variable (w) we use set $T_1=\{A_{11}, A_{21}, A_{31}, A_{41}, A_{51}\}$. As a fuzzy set of second variable(y) we use $T_2=\{ B_{1,} B_{2,} B_{3,} B_{4,} B_5)$.

Using Mamdani Algorithm, assume that w=35, In this case fuzzification of the variable is leading to truth degree of 0.1.

Membership functions are shown of w and y are shown on figure 2 and figure 3 respectively.



. Figure 2. Membership functions for fuzzy sets of linguistic variable w

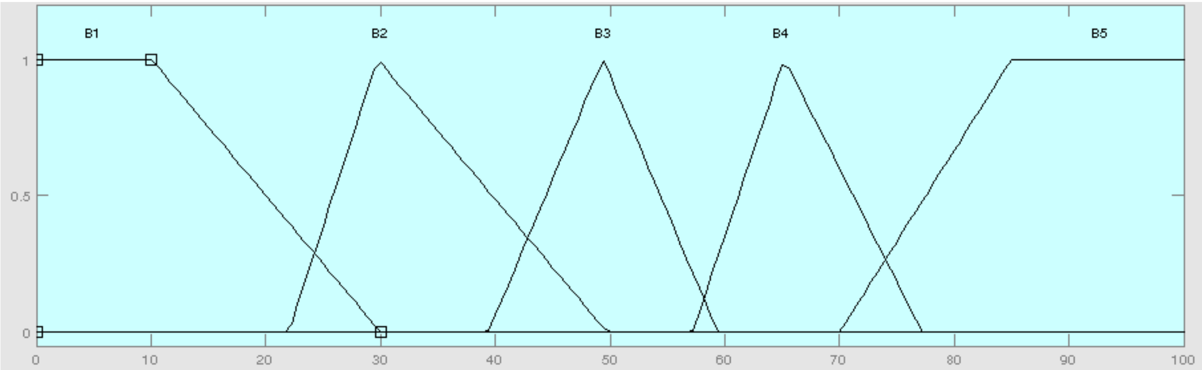


Figure 3. Membership functions for fuzzy sets of linguistic variable y

Second step. Aggregation of truth degrees of prerequisites for each of the rules.

$$\alpha_1 = \min\{\mu_{A11}(w_1'), \mu_{A12}(w_2')\}$$

$$\alpha_2 = \min\{\mu_{A21}(w_1'), \mu_{A22}(w_2')\}.$$

(In our case we do not have 2 inputs so this step will not be used in our example)

Third Step. Activation(figure 4). Determining truth degree of conclusions for each rule based on method called min-activation As we proposed in fuzzification step the variable w=35 and this assumption activating rules 3 and 4 in rule base of fuzzy inference system

$$\boldsymbol{\mu_{B'1} = \min\{\alpha_1, \mu_{B1}(y)\},}$$

$$\boldsymbol{\mu_{B'2} = \min\{\alpha_1, \mu_{B2}(y)\}.}$$

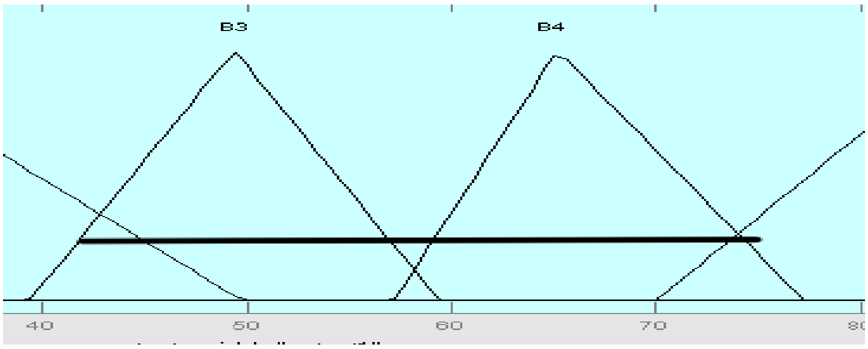


Figure 4. Activation step

Fourth Step(figure 5). Accumulation of conclusions, gained on the prior step, over all rules.Assosiation of truncated fuzzy sets are done by using max-disjunction method.As the result fuzzy set for output variable are formed.

$$\mu_{B'} = \max\{ \mu_{B1'}(y), \mu_{B2'}(y)\},$$

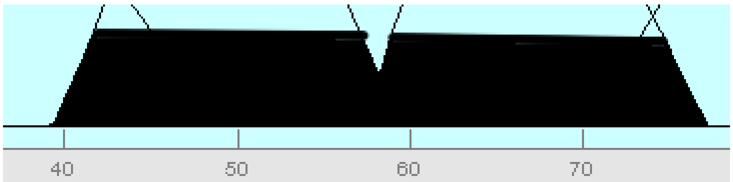


Figure 5. Accumulation

Fifth step. Defuzzification- the step is bringing results to clarity. If results should be presented as an ordinary numbers(not fuzzy sets). In Mamdani algorithm of fuzzy inference system, center of gravity method is used for $\mu_{B'}(y)$:

$$y' = \frac{\int_{Nmin}^{Nmax} y * \mu_{B'}(y) * dy}{\int_{Nmin}^{Nmax} \mu_{B'}(y) * dy}$$

where N_{max}, N_{min} – interval limits of vector of output varible y fuzzy set.

The main problems, which are solving during generation control systems with fuzzy logic, are comparison of object descriptions with conditions of rules' truth degree.

In solving problems of management, where the object or his environment can poorly be identified by mathematics descriptions, fuzzy controller is used, ie fuzzy logic is used with other traditional control algorithms or instead of them. Overview, of the research and practical work has shown that fuzzy control in a number of cases gives better results than classical methods. Fuzzy variables, gained from the results “not quite accurate measurement” in many ways are similar to the theory of probability distributions, but compared with probabilistic methods, techniques, fuzzy logic dramatically reduce the amount of calculations made, which, lead to increase performance of fuzzy systems.The disadvantage of the system, the original set of fuzzy rues formulated by a human-expert, and can be incomplete or controversial, furthermore, view and paramenters of membership functions, which describe input and output variables of the system, are chosen subjectively and can be not fully reflective for the reality.

3.2 Comparison with PID

We can compare two main approaches of mobile robot controlling: PID and Fuzzy Logic Controller.”Fig 6 and fig 7 show two sets of results comparing PID and Fuzzy Logic Controllers technigues to the inverted pendulum problem simulation for the same parameters here, the PID controller proportional gain, Kp, derivative gain, Kd and integral gain, Ki are found to be 9, 14, and 0.06 respectively. The first two graph show that the fuzzy logic controller gives a smaller overshoot and shorter settling time”[7]

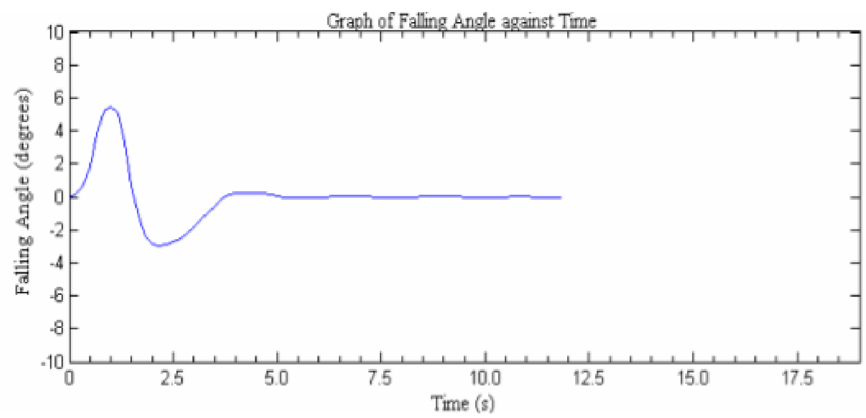


Figure 6. Falling angle response for fuzzy control.

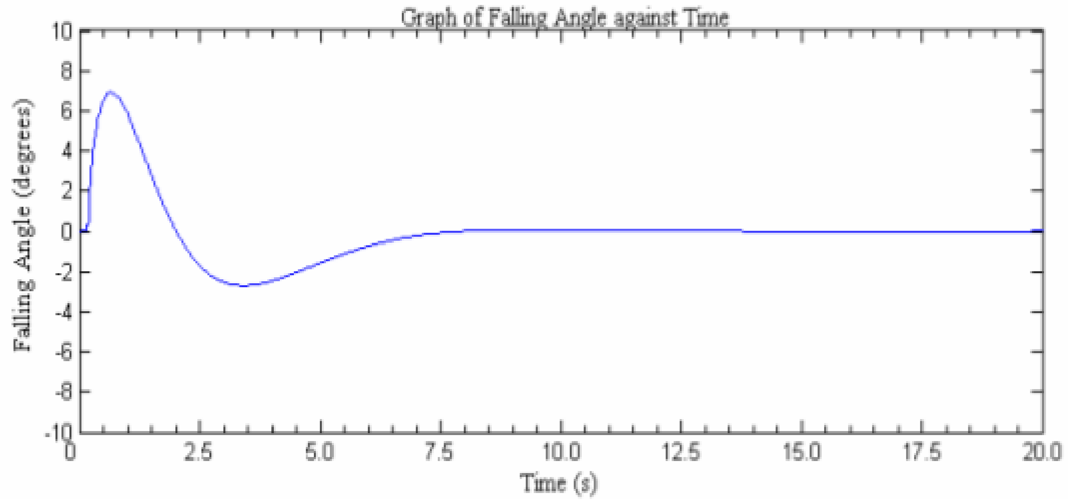


Figure 7. Falling angle response for conventional PID.

“Fig.9 shows that the conventional controller totally failed to balance the pendulum as it was designed for the nominal value of cart mass. On the other hand, the fuzzy logic controller exhibited small performance degradation due to this parameter change as shown in 8. This proves that fuzzy logic is not based on the mathematical model and more robust to mass variations.”[7]

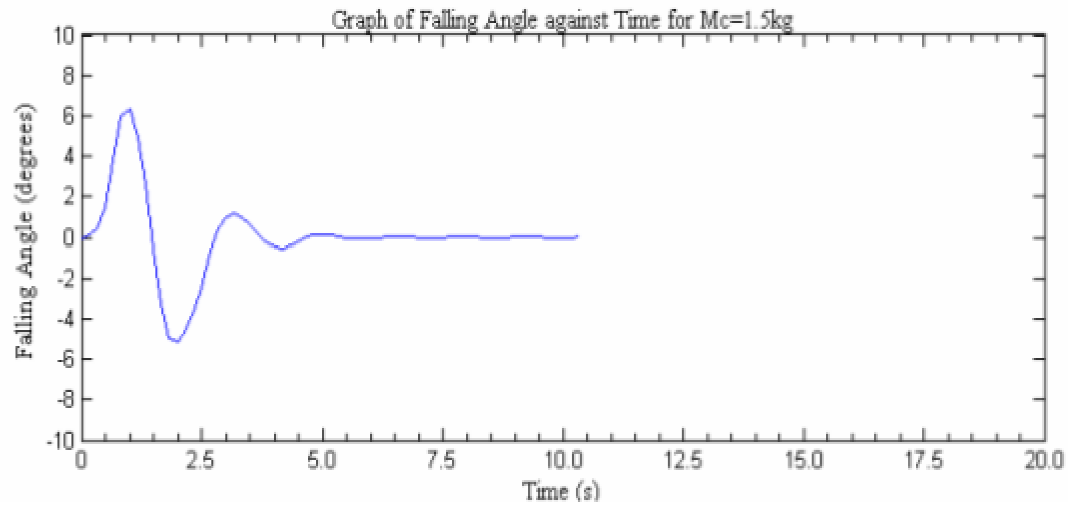


Figure 8. Falling angle response for fuzzy control

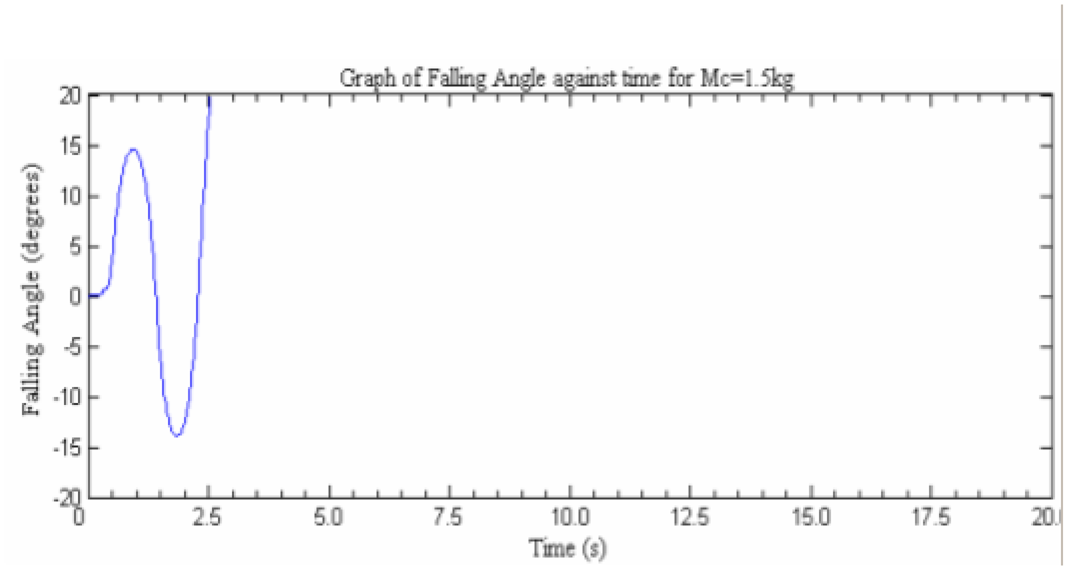


Figure 9. Falling angle response for conventional PID control.

Furthermore, comparing traditional controllers and Fuzzy Logic Controller, we can see various of advantages of FLC

-it is allowing computers to operate more like humans with phrases such as ‘too fast’, ‘slow’, ‘very slow’, (in traditonal controllers fast or lslow)[10].For example, situations like car is moving fast. This statement doesn’t determine the speed of car and also for each person “fast” means different speeds(for one person 100 km/h is fast, for another 100 it is medium speed). For that reason we are using membership function.

-for improving the performance of the system we should just add new rules.[8]

-“fuzzy control can be used to improve existing traditional controller systems by adding an extra layer of intelligence to the current control method”[8]

4 Project Specification

Proposed system built on Fuzzy Logic architecture with static and moving obstacle avoidance with kinematics(later Chapter 5) which define robots current position and the system calculate distance to the target and sends signal to move the robot, also system calculate possible collision and avoiding it.

4.1 Functional Requirements

1. System must calculate current position of the robot

2. System must allow user to define desired target position (X, Y)

3. System must detect obstacles

4. System must avoid obstacles and reach target

5. System must save all coordinates which robot passed

4.2 Non-Functional Requirements

-Maintability: The system should include well written coments for later use

-Efficiency: System should take coordinates of target as soon as user passed it to system

-Usability: System should be easy defined and easy understandable

-Extensibility: The should have enough features to be easy upgradable

5 Mathematical Implementation

For realizing proposed system, firstly, we should generate computer model of the system. Therefore, we need to create mathematical model of the mobile robot. Mathematical model consists of two main parts:

1. Kinematics, which defines the motion of point without consideration of causes of motion.[17]

2. Dynamics, is the part of Physics which defines causes of motion.

5.1 Model of the system

We proposed that robot has 3 wheels two of them are non-rotating wheel drive with independent motors and one passing wheel. The rotating of the robot is done by difference between angular velocities of two wheels.On these robot we setting up the ultrasonic sensor for getting distance to the obstacles. Actions, on which mobile robots is doing, are operations such as the moving of robot to desired target and avoiding obstacles. Scheme of robot's controlling presented on figure 10

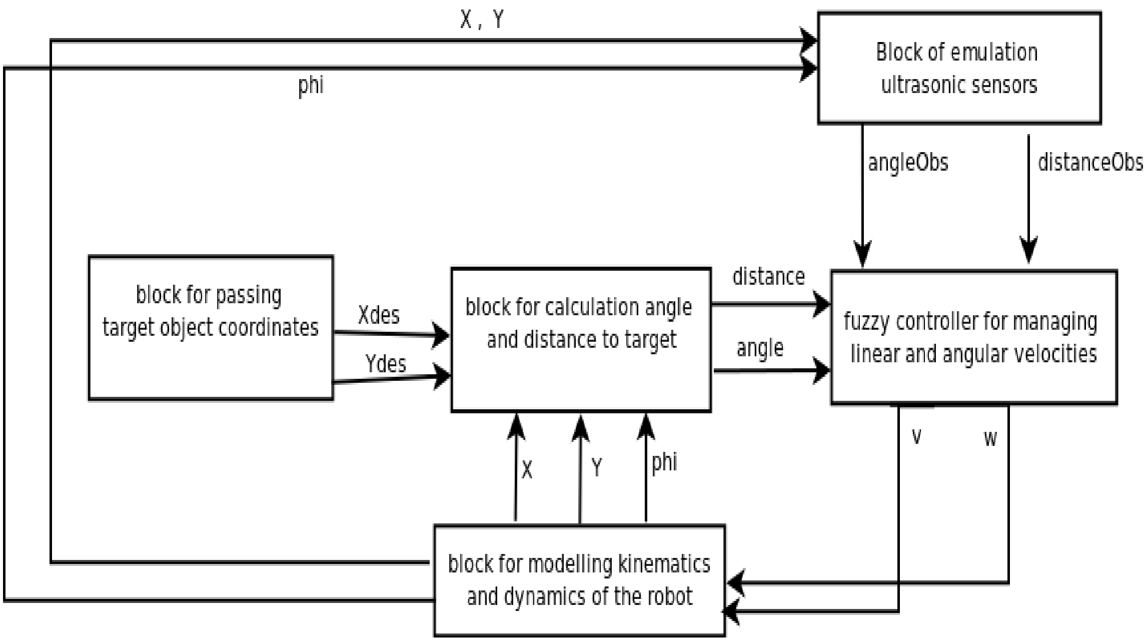


Figure 10. Scheme of modelling the robot control system

Block for passing target object coordinations-this block requires to system administrator to tell the robot target coordinates(Xdes,Ydes), in fact, human-operator is passing to system a goal.

Then these coordinates will be passed to fuzzy contoller with coordinates of obstacles, block of emulation ultrasonic sensors passing detecting obstacles and putting these obstacles distance to list when all obstacles detected. It passing to fuzzy controller the nearest obstacle distance(distanceObs) and angle(angleObs)(according to robot position)

Fuzzy controller block gets these 4 variables and first by with angle and distance to angle calculating linear and angular velocities of the robot(v, w)

Block for modelling kinematics and dinamics of the robot based on linear and angular velocities, the current position(X,Y) and orientation angle(phi) of robot will be computed

For describing the behavior of robot we need to develop its mathematical model. This will allow us, without hardware implementation of mobile robot, to analyze the work of system and add necessary corrections. The mathematical model of mobile robot consists of two main parts: dynamic and kinematic models.

Kinematic model is a simple description of mobile robot and allow the investigetion of its properties.

Dinamic model is more detailed description of mobile robot and takes into account the force-torgue effects produced by the actuators.

Proposed mathematical model of robot’s motion is right only for motion on horizontal plane.

In order to mathematically describe a mobile robot to introduce a number of conditions set out in the form of the following assumptions

-We consider that mobile robot’s mechanisms are rigidly bound;

-Wheels are underformable

-Movement of the robot is carried out without slipping

-The platform is considered to be solid body, on which wheel system is fixed

5.2 Kinematic model of mobile robot

The robot motion is on horizontal plane, it is sufficient to consider the two-dimensional case. Figure 11, depicts robot with relevant coordinate systems. Wheels arranged in points L and R, axes of rotation coincide with the LR segment of length l. We connect the robot with X_p, O_p, Y_p coordinate system and locating O_p in the middle of LR and sending axis X_p perpendicular to LR in motion direction. Then position of robot, as rigid body, can be defined by 3 values: x,y, phi.

where x,y- the coordinates of O_p on coordinate system XOY

phi- angle between the axes X and X_p

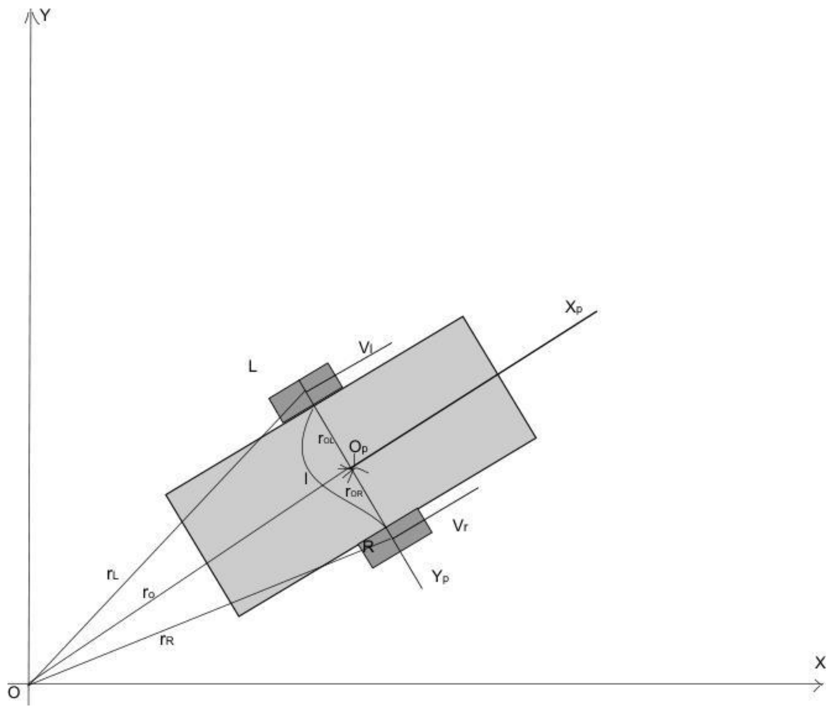


Figure 11. Kinematics of mobile robot

The position of L,R and O_p in absolute system corespond to vectors r_L,r_R and r_O. The location of O_p n also be found from these system of equations:

$$\mathbf{r_O}=\mathbf{r_r}+\mathbf{r_{OR}}$$

$$\mathbf{r_R}=\mathbf{r_L}+\mathbf{r_{OL}}$$

From the kinematics of motion it is known that speed of motion of any solid body can be presented:

$$\mathbf{V}=\mathbf{V_o}+\mathbf{w}*\mathbf{r_p}$$

V- translational velocity of any point

V_o- translational velocity of solid body

w-angular velocity of solid body reatively to pole O

r_O- vector, determining the location of any point relatively to pole O.

Therefore, according to figure 11, the robot velocity can be determined from the folowing:

$$\begin{cases} \mathbf{V}=\mathbf{V_R}+\mathbf{w}*\mathbf{r_{OR}} \\ \mathbf{V}=\mathbf{V_l}+\mathbf{w}*\mathbf{r_{OL}} \end{cases}$$

After some operations on these formulas we can get

$$\begin{cases} \mathbf{w}=\frac{\mathbf{1}}{\mathbf{l}}\left(\mathbf{V_R}-\mathbf{V_L}\right) \\ \mathbf{V}=\frac{\mathbf{1}}{\mathbf{2}}\left(\mathbf{V_R}-\mathbf{V_L}\right) \end{cases} \qquad (1)$$

For x, y, φ we get:

$$\begin{cases} x' = V * \cos\varphi \\ y' = V * \sin\varphi \\ \varphi' = w \end{cases} \qquad (2)$$

$$\begin{cases} x(t) = \int_{t0}^{t} V(\chi)\cos(\varphi(\chi))d\chi \\ y(t) = \int_{t0}^{t} V(\chi)\sin(\varphi(\chi))d\chi \\ \varphi(t) = \int_{t0}^{t} w(\chi)d\chi \end{cases} \qquad (3)$$

Substituting (1) into (2) we obtain:

$$\begin{cases} x' = \frac{1}{2}(V_R + V_L)cos\varphi \\ y' = \frac{1}{2}(V_R + V_L)sin\varphi \\ \varphi' = \frac{1}{l}(V_R - V_L) \end{cases} \qquad (4)$$

(4) is a system of nonlinear differential equations of the third order with respectto the phase vector components(x,y,φ). V_R and V_L are defining robot motion.

Finally (4) can be defined as a kinematic model of mobile robot. (1) is determining linear and angular velocities of robot.

5.3 Dynamic model of the system

Mobile robot is moving on plane, having two drive wheels with radius σ. Wheels rotate without slippage, and their axes rotation coincide with a line, LR(fig 11) This condition sets constraints on the position and velocity and is one of coupling equations, the geometric interpretation of which is that velocity vector $V=(x',y')^T$ is perpendicular to Y_p.

Robot, in fact, is a combination of three rigid bodies: platform(robot itself) and two driving wheels. The location of this system can be defined by 3 values: x,y, φ-the location of platform and lets assume ψ_L and ψ_R are angles of rotation of right and and left wheel respectively.

$$\begin{cases} \mathbf{V_R=\sigma * \psi_R'} \\ \mathbf{V_L= \sigma * \psi_L'} \end{cases} \qquad (5)$$

The conditions of slippage we obtain from(4) and (5)

$$\begin{cases} \mathbf{x'=\frac{\sigma}{2}(\psi'_R + \psi'_L)cos\varphi} \\ \mathbf{y'=\frac{\sigma}{2}(\psi'_R + \psi'_L)sin\varphi} \\ \mathbf{\varphi'=\frac{\sigma}{l}(\psi'_R - \psi'_L),} \end{cases} \qquad \mathbf{(6)}$$

By integrating last equation we gain

$$\boldsymbol{\varphi = \frac{\sigma}{l}(\psi'_R - \psi'_L) + C}$$

C-is constant after integrating

Final equations are

$$\begin{cases} x' = \frac{\sigma}{2}(\psi'_R + \psi'_L)\cos\varphi \\ y' = \frac{\sigma}{2}(\psi'_R + \psi'_L)\sin\varphi \\ \varphi = \frac{\sigma}{l}(\psi'_R - \psi'_L) + C \end{cases} \qquad (7)$$

In drawing up the dinamic model we will neglect the inertia moment of wheels as they are lower than the corresponding moments of the platform given to the motor shaft.

Now we can consider of motion equation. For that we need the Langrange equation of second type:

$$\frac{d}{dt}\left(\frac{\partial E_k}{\partial \psi'_i}\right) - \frac{\partial E_k}{\partial \psi_i} = \lambda_i \qquad i=1,2,\ldots,n$$

E_k-kinetic energy

ψ_i-generalized coordinates

n-number of generalized coordinates

λ_i-generalized forces

The mobile robot has two drive wheels. As wheels is moving without slippage, then rotation moment will be T_i can be presented as:

$$F_i = \frac{T_i}{\sigma}$$

Our robot system can be presented as a motion of figit body under the force of F_i

As we know from physics: kinetic energy of any object can be defined as:

$$E_k = \frac{mv^2}{2} \tag{8}$$

As our system is consist of two rigid bodies so whole system can be defined as**:**

$$\begin{cases} E_{k1} = \frac{m*V^2}{2} \\ E_{k2} = \frac{I*w^2}{2} \end{cases} \tag{9}$$

From (9) full kinetic energy will be calculated as:

$$E_k = E_{k1} + E_{k2} = \frac{m*V^2}{2} + \frac{I*w^2}{2}$$

m-mass of platform

V-the speed of center of platform's mass

I-the moment of inertia

w-angular velocity

With (x,y,φ) our formula will look like:

$$E_k = \frac{m}{2}((x')^2 + (y')^2) + \frac{I * \varphi^2}{2}$$

Using Langrage equation of type two we will gain these system of equations

$$\begin{cases} m * x'' = (F_r + F_l)cos\varphi \\ m * y'' = (F_r + F_l)sin\varphi \\ I * \varphi'' = \frac{l}{2}(F_r - F_l) \end{cases} \tag{10}$$

System of differential equations(10), defining motion of the platform in coordinate system XOY by force F_r and F_l, are relevant to system of differential equation determining the motiong of plane-paralel motion, which is defining moving of mass centres of rigid body under outside forces:

$$\begin{cases} \boldsymbol{M * x'' = \sum F_x^e} \\ \boldsymbol{M * y'' = \sum F_y^e} \\ \boldsymbol{I * \varphi'' = \sum M} \end{cases}$$

With above equations we can write:

$$\begin{cases} \boldsymbol{V' = \frac{1}{\sigma m}(T_R + T_L)} \\ \boldsymbol{w' = \frac{l}{2*\sigma*I}(T_R - T_L)} \end{cases} \qquad (11)$$

(11) is a system nonlienar differential equations relative to phase vector $(V, w)^T$

T_r and T_l components of control vector, generating motion system

Combining all equations

$$\begin{cases} \boldsymbol{x' = \frac{1}{2}(V_R + V_L)cos\varphi} \\ \boldsymbol{y' = \frac{1}{2}(V_R + V_L)sin\varphi} \\ \boldsymbol{\varphi' = \frac{1}{l}(V_R - V_L)} \\ \boldsymbol{V' = \frac{1}{\sigma*m}(T_R + T_L)} \\ \boldsymbol{w' = \frac{l}{2*\sigma*I}(T_R - T_L)} \end{cases} \qquad (12)$$

Then we can look at situation, when wheels are controlled by direct current. According to physics, the equation of motion of direct current :

$$\boldsymbol{U = L * I + R * I + K_w * w_{ms}} \qquad (13)$$

L-inductance of winding

I-current strength

R-resistance of winding

w-angular velocity of rotation of motor shaft

U-voltage

K-constructive factor which is determined by the relation

$$\boldsymbol{K = pN/(2a)}$$

p-the number of pole pairs

N-the number of face conductors

a-the number of paralel branches of winding

Angular velocities of motor shaft rotation and rotation of wheels can be shown:

$$\boldsymbol{w_{ms} = w * i}$$

i-gear ration of reducer;

For moment T, the equation looks like

$$\boldsymbol{T = K * I} \qquad (14)$$

So for if we put (14) to (13)

$$\boldsymbol{T' = -\frac{R}{L}T - \frac{K * K_w * i}{L}w + \frac{K}{L}U_R} \qquad (15)$$

Finally we can get equations for both wheels, according to (15) we get

$$\begin{cases} T_R' = -\frac{R}{L}T_R - \frac{K*K_w*i}{L}w_R + \frac{K}{L}U_R \\ T_L' = -\frac{R}{L}T_L - \frac{K*K_w*i}{L}w_L + \frac{K}{L}U_L \end{cases} \qquad (16)$$

K- constructive factor for motor

K_w-constructive factor for motor shaft

Then we can w_l and w_R through the phase coordinates.

$$\begin{cases} w_R = \frac{1}{\sigma}(V + \frac{w*l}{2}) \\ w_L = \frac{1}{\sigma}(V - \frac{w*l}{2}) \end{cases} \qquad (17)$$

Using (16) and (17) we will get system of differential equations, defining the robot motion with direct current

$$\begin{cases} x' = V * cos\varphi \\ y' = V * sin\varphi \\ \varphi' = w \\ V' = \frac{1}{\sigma*m}(T_R + T_L) \\ w' = \frac{l}{2*\sigma*I}(T_R - T_L) \\ T_R' = -\frac{R}{L}T_R - \frac{K*K_w*i}{L*\sigma}(V + \frac{w*l}{2}) + \frac{K}{L}U_R \\ T_L' = -\frac{R}{L}T_L - \frac{K*K_w*i}{L*\sigma}(V - \frac{w*l}{2}) + \frac{K}{L}U_L \end{cases} \qquad (18)$$

System (18) is a system of nonlinear differential equation with phase vectror $(x,y,\varphi, V,w, T_r,T_L)^T$

6 Software Implementation

Proposed fuzzy logic inference system(Fuzzy Logic Controller) is done by Fuzzy Control Language(FCL) in Matlab environment. The simulation is done in Simulink package the results will be written to csv file, which will be read by Visual Basic Program and graph of Robot movement will be built. Fuzzy Logic Controller is built based on Mamdani algorithm with 4 inputs and 2 outputs.

6.1 Fuzzy logic controller

For controlling robot motion we should implement fuzzy controller. Fuzzy controller is a mechanism for conversion vector of input signals to vector of output signals. For that conversion mechanism of fuzzy inference is used, based on knowledge, which are added to system by expert-operator.Inputs and outputs signals in fuzzy logic system are logic-linguistic variables, values of which are fuzzy sets. Rule base of fuzzy logic system consists of rules, which determine dependancy between input and ouput fuzzy sets, and membership functions, which showing the extent of accordance between real values and consept, determined by fuzzy sets.Firstly, the input signals of fuzzy controller are distance, between the robot and target and angle, between robot direction and straight line between robot center and target.

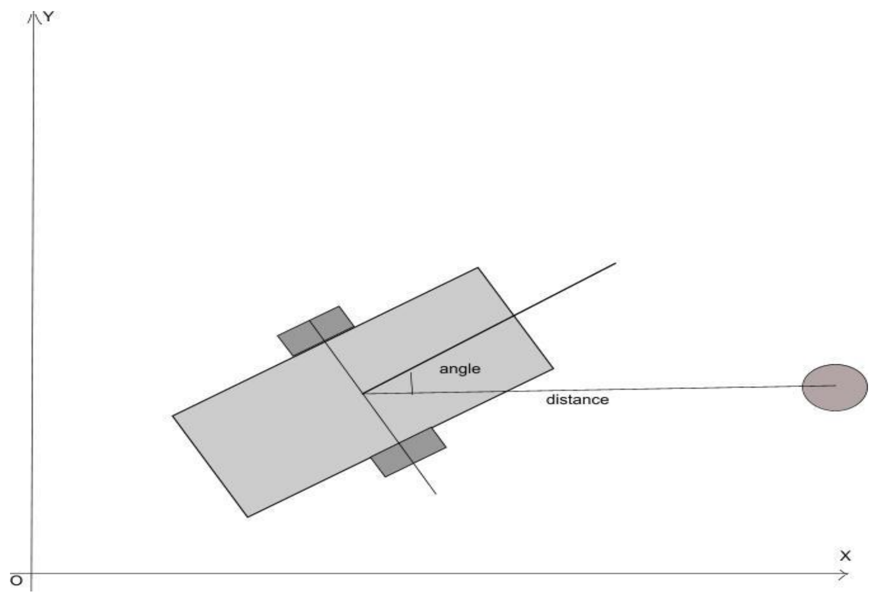


Figure 12. Inputs of fuzzy controller 1st type

Secondly, the outputs of fuzzy controller are linear V and angular w velocities of mobile robot.

Mamdani algorithm will be chosen as the mechanisms of fuzzy logic controller. Numerical output will be computed based on “centre of gravity” as:

$$\boldsymbol{y} = \frac{\int y_i * \mu_p(y_i) * dy}{\int \mu_p(y_i) * dy}$$

$\mu_p(y_i)$-fuzzy result of logic inference;

y-output value of fuzzy controller

During the work of fuzzy controller, the following algorithm will be done:

-Fuzzification input variables-converting accurate values of input signals to input values of linguistic variables

-Fuzzy logic inference

-Defuzzification-gaining real values of output variables.

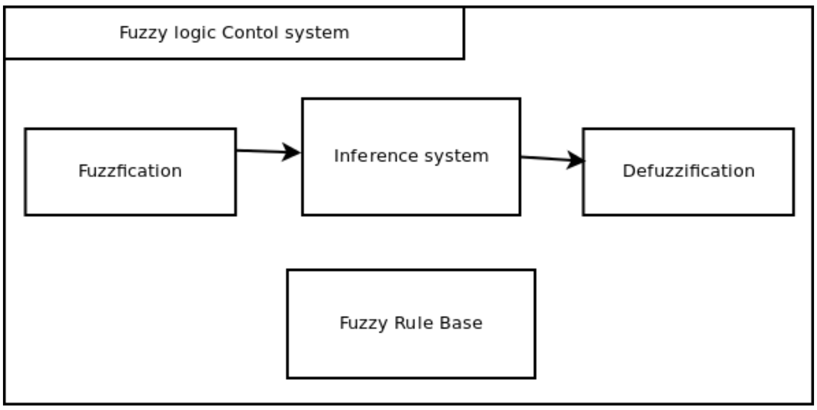


Figure 13. Inside fuzzy logic controller

By generating fuzzy controller we can assume that it consists of two main structures, fuzzy controller for angular velocity and fuzzy controller for linear velocity and inputs of them are angle, between robot orientation and target, and distance to target respectively.(Fig 12)

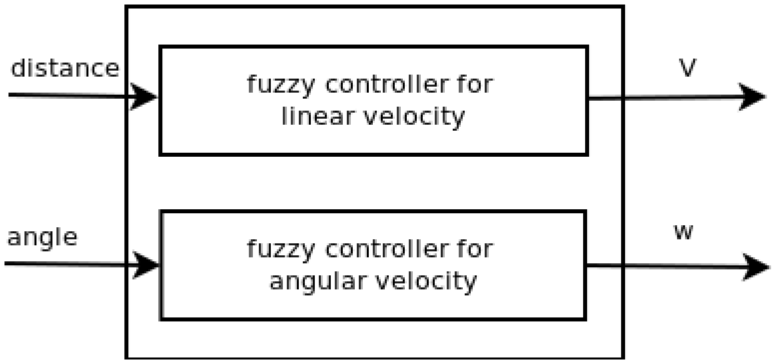


Figure 14.splitted fuzzy controller(abstraction)

It is important that controller should change linear velocity by increasing it when the target is far away and decreasing gradually when the robot reaching the target.

Lets consider linear velocity fuzzy controller, fuzzy variable is distance and it has the following fuzzy sets: Close, Medium, Far

The membership function is taking view of trapezoidal function:

$$\boldsymbol{\mu_A(x)} = \begin{cases} \mathbf{0}, & \boldsymbol{if\ (x < a) or\ (x > d)} \\ \dfrac{\boldsymbol{x-a}}{\boldsymbol{b-a}}, & \boldsymbol{if\ a \le x \le b} \\ \mathbf{1}, & \boldsymbol{if\ b \le x \le c} \\ \dfrac{\boldsymbol{d-x}}{\boldsymbol{d-c}}, & \boldsymbol{if\ c \le x \le d} \end{cases}$$

View of first input distance is shown on figure 15.

Output variable is linear velocity of robot as a trapezoidal function with the following fuzzy sets: Very Low,Low, Fast, Very Fast

View of the first output linear velocity is shown figure 16.

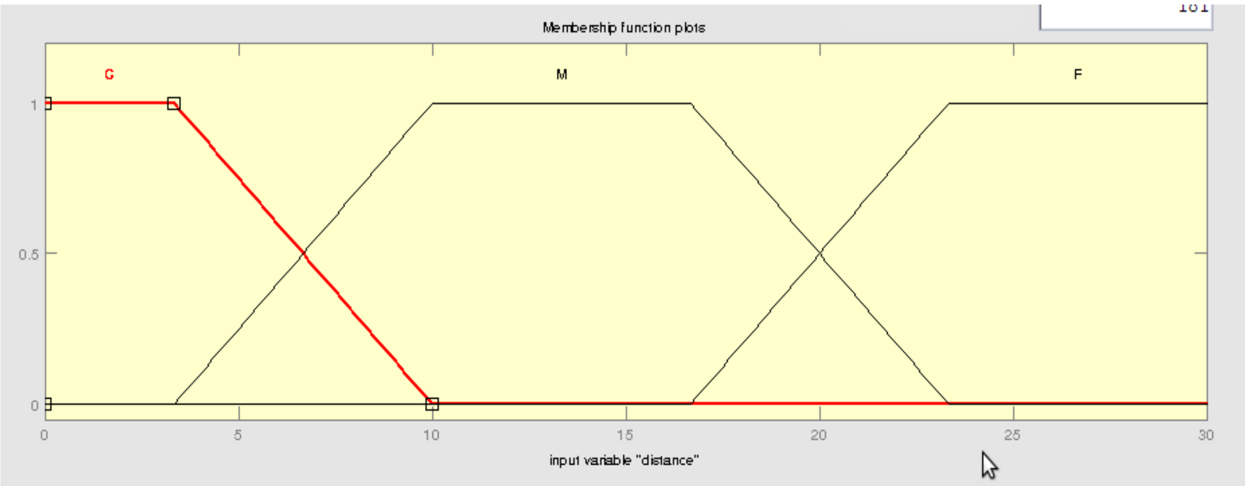


Figure 15. Distance input membership function

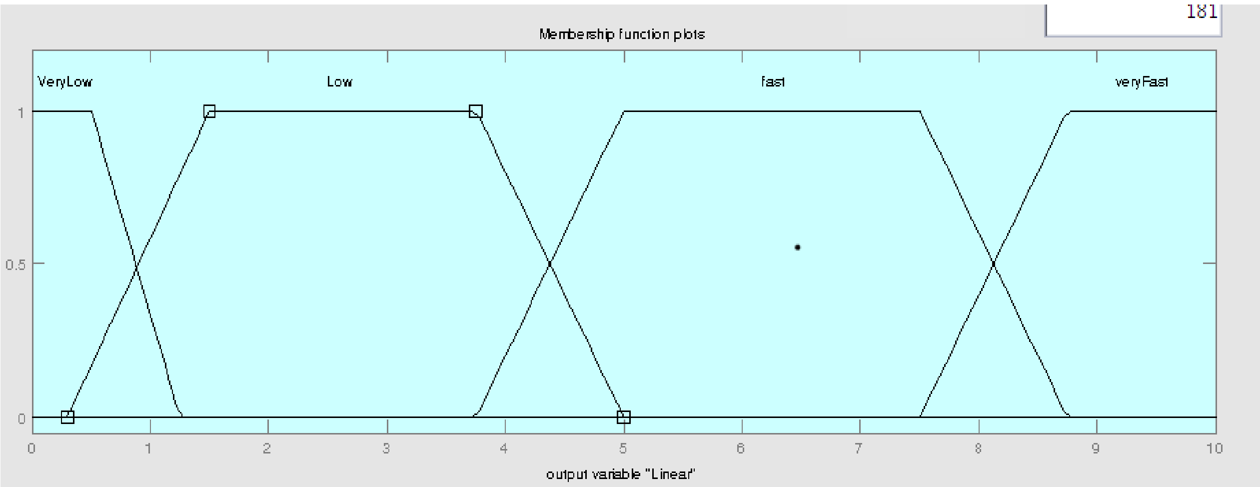


Figure 16. Linear velocity output membership function

For defining output from input we should generate rule base system. Rule base:

1. IF (distance is C) THEN (Linear Velocity is Low)
2. IF (distance is M) THEN (Linear Velosity is Fast)
3. IF (distance is F) THEN (Linear Velosity is VeryFast)

For controlling orientation of mobile robot relatively to target, ie rotation of robot, we are using the same technique as with linear velocity. Input linguistic variable of fuzzy contoller for angular velocity is angle direction to the target(angle variable), which is calculated with the following formula:

$$\boldsymbol{angle = arctan\frac{ydes-ycurrent}{xdes-xcurrents}} \quad (1)$$

angle- is an angle between target position and robot orientation

ydes-desired position of target correspond to Y axis

xdes-desired position of target correspond to X axis

ycurrent-current position of robot obtained from kinematic correspond to Y axis

xcurrent-current position of robot obtained from kinematic correspond to X axis

Input variable is angle of robot as a trapezoidal function with the following fuzzy

sets(figure 16): Big Negative, Intermediate Negative, Negative, Negative Too Small, Zero, Positive Too Small, Positive, Intermediate Positive, Big Positive

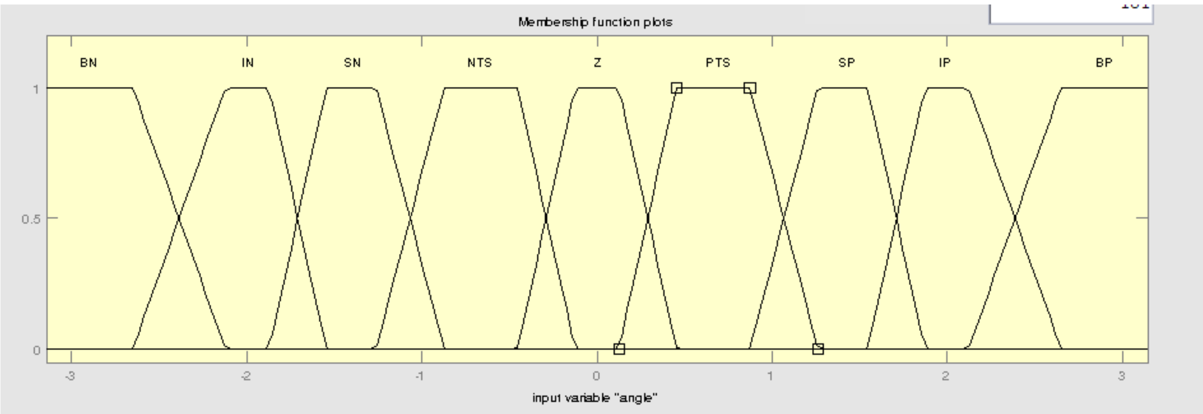


Figure 16. angle input membership function

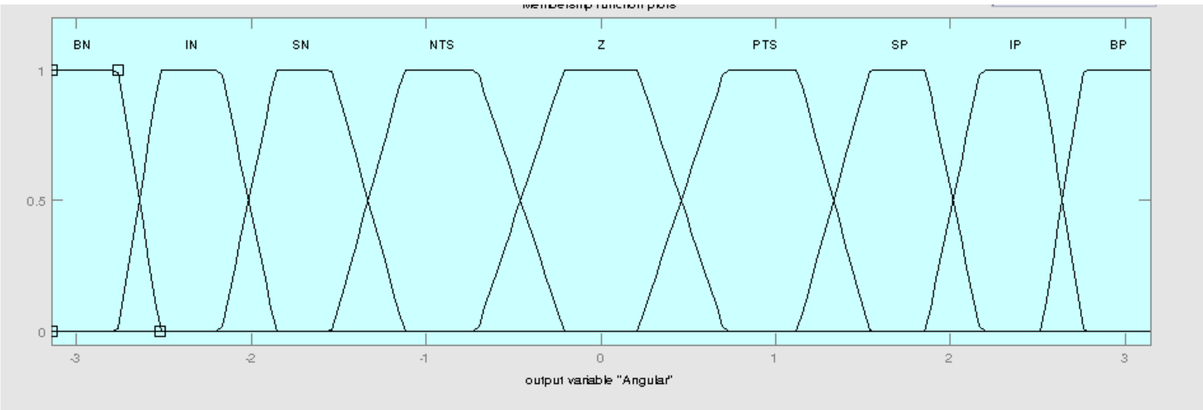


Figure 18. angular velocity output membership function

Output linguistic variable for fuzzy controller is angular velocity w: with the following fuzzy sets(figure 18): Big Negative, Intermediate Negative, Negative, Negative Too Small, Zero, Positive Too Small, Positive, Intermediate Positive, Big Positive

Rule Base is following

1. IF (angle is BN) THEN (angular is BN)
2. IF (angle is IN) THEN (angular is IN)
3. IF (angle is N) THEN (angular is N)
4. IF (angle is NTS) THEN (angular is NTS)
5. IF (angle is Z) THEN (angular is Z)
6. IF (angle is PTS) THEN (angular is PTS)
7. IF (angle is P) THEN (angular is P)
8. IF (angle is IP) THEN (angular is IP)
9. IF (angle is BP) THEN (angular is BP)

Before implementation of obstacle avoidance,first, we should tell about detecting these obstacles. Obstacle detecting procedure is based on Ultrasonic sensors theory. Next section will be about Sonic Wave and Ultrasonic Sensors.

6.2 Ultrasonic sensors for detecting obstacles.

Ultrasonic Sensors are devices which can determine distance from robot to obstacle with sonic waves. It generates high frequency sound waves and send it in all directions. Every object reflect and absorb a portion of wave. Ultrasonic Sensors send and receive the portion of waves after receiving the device calculate time between sending and obtaining and can calculate actual distance to the object[13]. For that reason sensors should know about the environment. The main parts determine the density, pressure and temperature. For example: speed of sound in air 343 metres per second at 20^0 C but it can vary because of changes in the envorenment.

Lets assume that sensor sent sonic waves at T and received it at T_1 and temperature is 20^0 C and we will neglect the pressure. The distance between sender and receiver would be

$$\mathbf{S = (T_1 - T) * speedOfSound}$$

It is a cheap implementation of obstacle detecting. Unlike digital image processing, Ultrasonic Sensors do not give information how is obstacle look like and contours but in our implementation we need only avoid obstacles and we do not want to know how they look like.

6.3 Obstacle avoidance behaviour

6.3.1 Static Obstacles avoidance

For avoiding static obstacles we should add another layer of intelligence to our system. Because these system is fuzzy controller we will do it by adding new inputs to the system. That is an advantage of the system for adding new layer of intelligence we are not changing whole system.

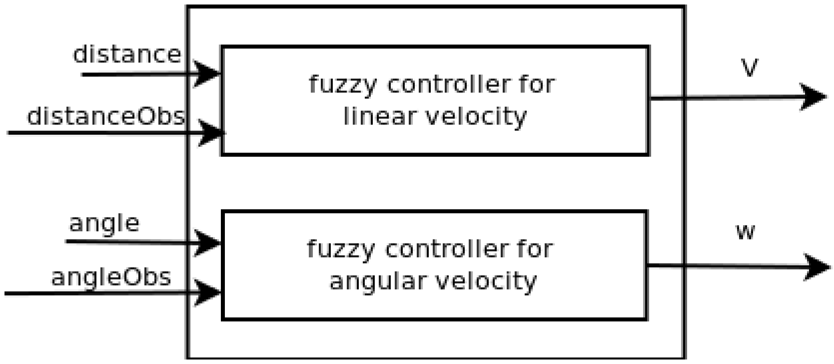


Figure 19. Overview of the splitted fuzzy controller type 2(abstraction)

New view of the system presented on figure 19.

A new layer of intelligence will be distanceObs and angleObs inputs of the system.[10]. Unlike our system should avoid obstacles in a faster way, because if the the robot detect obstacle it is avoiding it in the smalest angle as possible so it increases the system intelligense and robot gets the target earlier than proposed in [10].

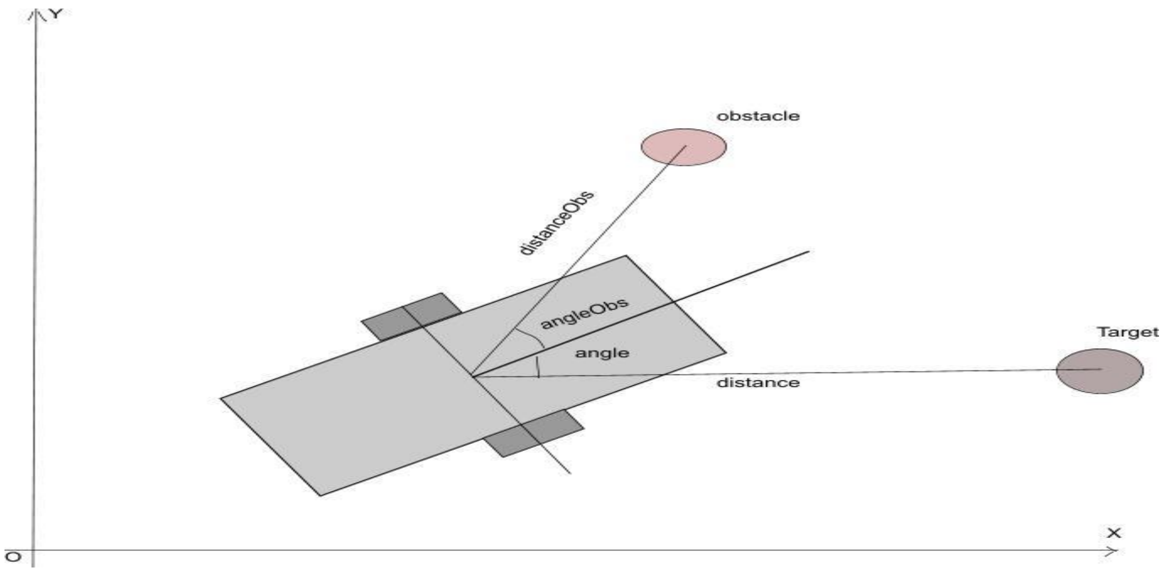


Figure 20. Fuzzy logic controller 2nd type.

New input fuzzified variable is distanceObs. It is actual distance from robot to obstacle. This variable are not computed but measured through the Ultrasonic sensors, which located on robots platform. Each sensor is computing the obstacle distance. Detected obstacles are appended to List of Obstacles. This list is examined by robot and if the robot detects the obstacle on his way obstacle avoidance behavior is activated.

DistanceObs has following fuzzy sets: Near Close, Near,Medium, Far

DistanceObs membership function is trapezodial function and viewed on figure 21.

Combination of rules will be shown later on this section.

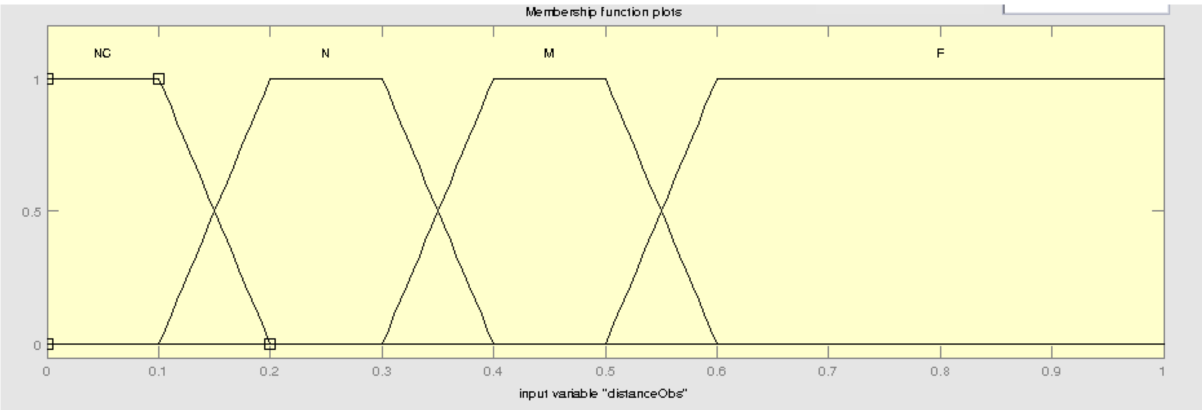


Figure 21. DistanceObs Membership function

Second input to controller is angleObs this angle is calculated acccording to sensor's information. Assume that each sensor can detect objects in angle between 0-60^0. As in this section we proposed that obstacles are static we calculate obstacles' angle for [–pi;+pi] . There are0 should be 6 sensors and if we get information from second sensor it means that the angleObs is between 60-120. These sensor information are combined and one angleObs for each object are created. According the figure 20, angleObs is the angle between robot orientation and obstacle position.

AngleObs membership function is presented on figure 22. AngleObs has following fuzzy sets: Big Negative, Intermediate Negative,Small Negative, Negative Too Small, Zero,

Pozitive Too Small, Small Positive, Intermediate Positive, Big Positive

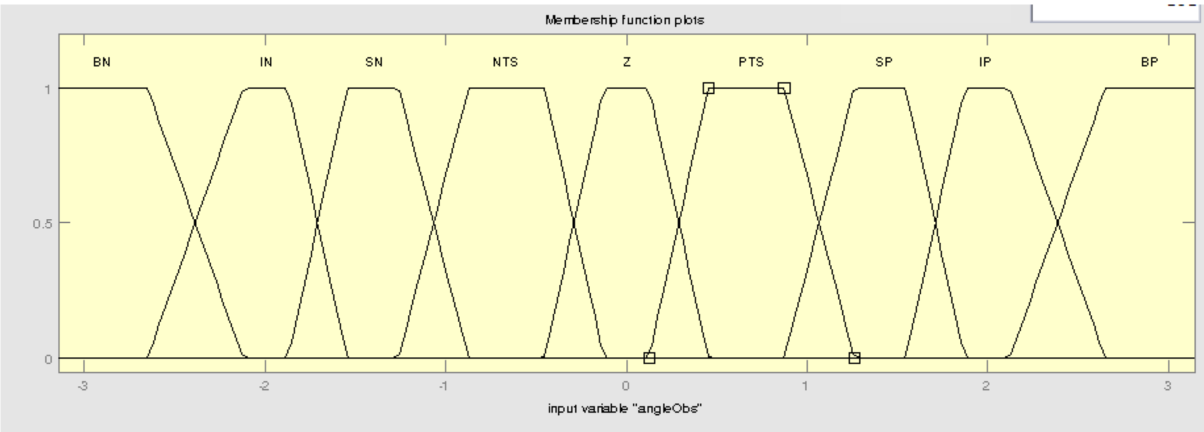


Figure 22. Membership function for input angleObs

Rule Base for whole system will be:

1. IF (angleOBs is Z) AND (distanceObs is F) AND (distance is M) THEN (Linear is M)(Angular is NTS);
2. IF (distanceObs is NC) AND (angleObs is Z) THEN (Linear is VL)(Angular is BN);
3. IF (distanceObs is N) AND (angleObs is Z) THEN (Linear is L)(Angular is BN);
4. IF (distanceObs is M) AND (angleObs is NTS) THEN (Linear is M)(Angular is BN);
5. IF (angle is BN) AND (distanceObs is F) AND (distance is C) THEN (Linear is L)(Angular is BN);
6. IF (angle is IN) AND (distanceObs is F) AND (distance is C) THEN (Linear is L)(Angular is IN);
7. IF (angle is SN) AND (distanceObs is F) AND (distance is C) THEN (Linear is L)(Angular is SN);
8. IF (angle is NTS) AND (distanceObs is F) AND (distance is C) THEN (Linear is L)(Angular is NTS);
9. IF (angle is PTS) AND (distanceObs is F) AND (distance is C) THEN (Linear is L)(Angular is PTS);
10. IF (angle is SP) AND (distanceObs is F) AND (distance is C) THEN (Linear is L)(Angular is SP);

11. IF (angle is IP) AND (distanceObs is F) AND (distance is C) THEN (Linear is L)(Angular is IP);
12. IF (angle is BP) AND (distanceObs is F) AND (distance is C) THEN (Linear is L)(Angular is BP);
13. IF (angle is BN) AND (distanceObs is F) AND (distance is M) THEN (Linear is M)(Angular is BN);
14. IF (angle is IN) AND (distanceObs is F) AND (distance is M) THEN (Linear is M)(Angular is IN);
15. IF (angle is SN) AND (distanceObs is F) AND (distance is M) THEN (Linear is M)(Angular is SN);
16. IF (angle is NTS) AND (distanceObs is F) AND (distance is M) THEN (Linear is M)(Angular is NTS);
17. IF (angle is PTS) AND (distanceObs is F) AND (distance is M) THEN (Linear is M)(Angular is PTS);
18. IF (angle is SP) AND (distanceObs is F) AND (distance is M) THEN (Linear is M)(Angular is SP);
19. IF (angle is IP) AND (distanceObs is F) AND (distance is M) THEN (Linear is M)(Angular is IP);
20. IF (angle is BP) AND (distanceObs is F) AND (distance is M) THEN (Linear is M)(Angular is BP);
21. IF (angle is BN) AND (distanceObs is F) AND (distance is F) THEN (Linear is H)(Angular is BN);
22. IF (angle is IN) AND (distanceObs is F) AND (distance is F) THEN (Linear is H)(Angular is IN);
23. IF (angle is SN) AND (distanceObs is F) AND (distance is F) THEN (Linear is H)(Angular is SN);
24. IF (angle is NTS) AND (distanceObs is F) AND (distance is F) THEN (Linear is H)(Angular is NTS);
25. IF (angle is PTS) AND (distanceObs is F) AND (distance is F) THEN (Linear is H)(Angular is PTS);
26. IF (angle is SP) AND (distanceObs is F) AND (distance is F) THEN (Linear is H)(Angular is SP);

27. IF (angle is IP) AND (distanceObs is F) AND (distance is F) THEN (Linear is H)(Angular is IP);
28. IF (angle is BP) AND (distanceObs is F) AND (distance is F) THEN (Linear is H)(Angular is BP);
29. IF (distanceObs is NC) AND (angleObs is BN) THEN (Linear is L)(Angular is PTS);
30. IF (distanceObs is NC) AND (angleObs is IN) THEN (Linear is L)(Angular is SP);
31. IF (distanceObs is NC) AND (angleObs is SN) THEN (Linear is L)(Angular is IP);
32. IF (distanceObs is NC) AND (angleObs is NTS) THEN (Linear is VL)(Angular is BP);
33. IF (distanceObs is NC) AND (angleObs is PTS) THEN (Linear is VL)(Angular is BN);
34. IF (distanceObs is NC) AND (angleObs is SP) THEN (Linear is L)(Angular is IN);
35. IF (distanceObs is NC) AND (angleObs is IP) THEN (Linear is L)(Angular is SN);
36. IF (distanceObs is NC) AND (angleObs is BP) THEN (Linear is L)(Angular is NTS);
37. IF (distanceObs is N) AND (angleObs is BN) THEN (Linear is L)(Angular is PTS);
38. IF (distanceObs is N) AND (angleObs is IN) THEN (Linear is L)(Angular is SP);
39. IF (distanceObs is N) AND (angleObs is SN) THEN (Linear is L)(Angular is IP);
40. IF (distanceObs is N) AND (angleObs is NTS) THEN (Linear is L)(Angular is BP);
41. IF (distanceObs is N) AND (angleObs is PTS) THEN (Linear is L)(Angular is BN);
42. IF (distanceObs is N) AND (angleObs is SP) THEN (Linear is L)(Angular is IN);
43. IF (distanceObs is N) AND (angleObs is IP) THEN (Linear is L)(Angular is SN);
44. IF (distanceObs is N) AND (angleObs is BP) THEN (Linear is L)(Angular is NTS);
45. IF (distanceObs is M) AND (angleObs is BN) THEN (Linear is M)(Angular is PTS);
46. IF (distanceObs is M) AND (angleObs is IN) THEN (Linear is M)(Angular is SP);
47. IF (distanceObs is M) AND (angleObs is SN) THEN (Linear is M)(Angular is IP);
48. IF (distanceObs is M) AND (angleObs is NTS) THEN (Linear is M)(Angular is BP);
49. IF (distanceObs is M) AND (angleObs is PTS) THEN (Linear is M)(Angular is BN);

50. IF (distanceObs is M) AND (angleObs is SP) THEN (Linear is M)(Angular is IN);
51. IF (distanceObs is M) AND (angleObs is IP) THEN (Linear is M)(Angular is SN);
52. IF (distanceObs is M) AND (angleObs is BP) THEN (Linear is M)(Angular is NTS);

6.3.2 Moving obstacle avoidance

Until that moment in the dissertation, we talked about static obstacles(walls, barriers and so on.) But in real world there are also moving obstacles which can lead to problems too. If obstacle is moving it does not have special position on space. The object has size and velocities.

Before going to robot respons to moving obstacle, we should first discuss how to detect obstacle and moreover, calculate speed of the obstacle and size of it.

For speed calculation we will use Doppler effect. In 1842 Christian Doppler(Austrian physicist) proposed theory of Dopler effect. The theory is based on changes in frequency of a wave(in our case sound wave).

Chrisitian Doppler(1842) states "When the source of the waves is moving toward the observer, each successive wave crest is emitted from a position closer to the observer than the previous wave.

Therefore each wave takes slightly less time to reach the observer than the previous wave. Therefore the time between the arrival of successive wave crests at the observer is reduced, causing an increase in the frequency. While they are travelling, the distance between successive wave fronts is reduced. if the source of waves is moving away from the observer, each wave is emitted from a position farther from the observer than the previous wave, so the arrival time between successive waves is increased, reducing the frequency. "

In real word system there are a lot of objects which can be an obstacle for the robot. When the object is near the wave frequency is higher than the object is far away. So for calculating the speed of the object we will use formula:

$$f_{obser} = (\frac{c+v_{rv}}{c+v_{sv}})f_{dis}$$

f_{obser}-observed frequency

f_{dis}-discharged frequency

c-the velocity of waves in the environment

v_{rv}-the velocity of the receiver relative to the environment

$_{vsv}$-the velocity of the sender relative to the environment

v_{os}=v_{rv}- v_{sv}-

After few calculation we will get

$$\frac{f_{obser}}{f_{dis}}c - c = v_{rv} - v_{sv} \xRightarrow{v_{os}=v_{rv}-v_{sv}} \quad v_{os} = \left(1 - \frac{f_{obser}}{f_{dis}}\right) * c \qquad (1)$$

Based on 1, we can set up device to measure speed of object.

To find size of the object the technique [10] can be used.

Now we can turn to mathematical model of the proposed obstacle. As we know from kinematics of the robot, to define the location of the robot on space three varibles should be defined (x,y, φ):

$$\begin{cases} x'(t) = v(t) * cos\varphi(t) \\ y'(t) = v(t) * cos\varphi(t) \\ \varphi' = w(t) \end{cases} \qquad (2)$$

The obstacle's position will be the same, obstacle is like the second robot. The second robot will have the same positions.

In our case we assume that obstacle does not change its speed and angular velocity. So in mathematic language they are constant Any moving obstacle can be defined with vector:

$$(x,y,\varphi,V,w,R)^T$$

x-coordinate of x, location on plane relatively to X

y-coordinate of y, location on plane relatively to Y

φ-angle of orientation of the robot

V-linear velocity of the moving obstacle

w-angular velocity of the moving obstacle

R- radius of circumscribed sphere

We consider that the size of object does not change and environment is not able to change the size of object. Firstly,we circumscribe the sphere around the object

We also assume that V and w are constant so:

$$(x,y,\varphi,V,w,R)^T \sim (x,y,\varphi,R)$$

According to the figure 23, PTL-Perpendicular to Target Line it is perpendicular from obstacle to robot trajectory to target;

PCL-it is proposed direction of the obstacle

θ is a angle between PTL and distanceObs

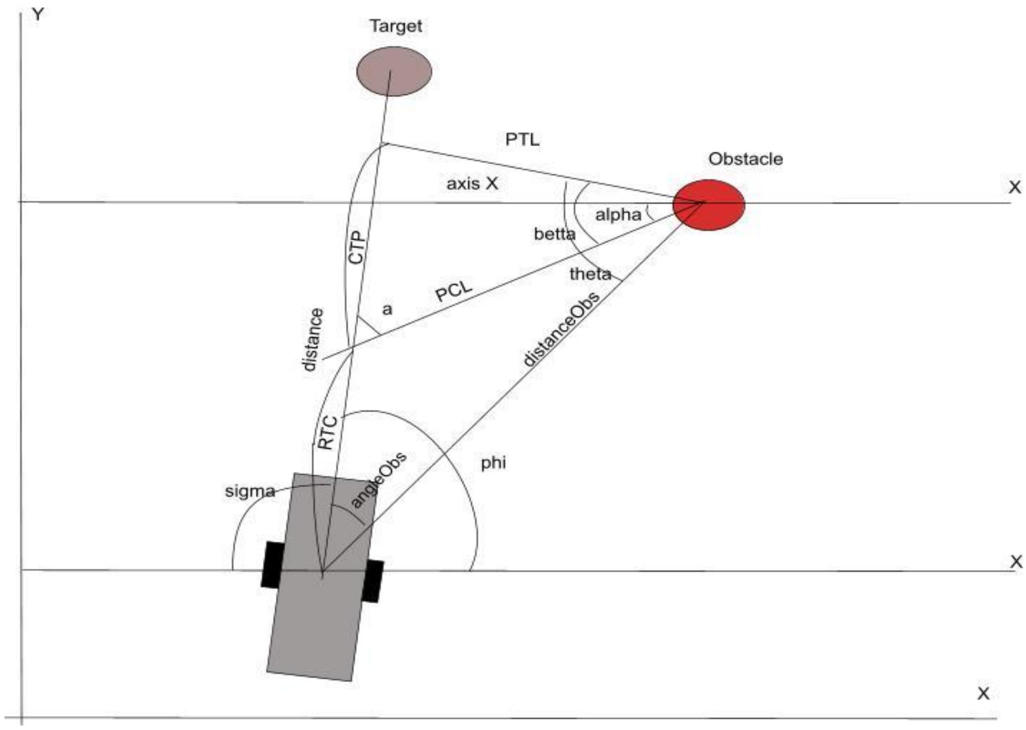


Figure 23. Overview of the system

CTP the part of distance line distance between perpendicular from obstacle and its actual trajectory.

β is angle between PTL and CTL

α- is angle on which obstacle is moving relatively to axis X.

a-angle between PCL and CTP

RTC-Robot Trajectory to possible Collision

$$\boldsymbol{\theta = phi - angleObs} \quad (1)$$

Based on (1) we can calculate PTL

$$\boldsymbol{PTL = distanceObs * sin\varphi} \quad (2)$$

$$\mathbf{a=180}-(\boldsymbol{\alpha}+\boldsymbol{\sigma}) \quad (3)$$

RTC=distance-CTL

Based on (3) angle β will be

$$\boldsymbol{\beta}=\mathbf{90}-\mathbf{a} \quad (4)$$

$$\boldsymbol{CTL}=\boldsymbol{PTL}*\boldsymbol{tan\beta} \quad (5)$$

$$\boldsymbol{RTC}=\boldsymbol{distance}-\boldsymbol{CTl}$$

$$\boldsymbol{PCL}=\sqrt{\boldsymbol{PTl}^{2}-\boldsymbol{CTP}^{2}}$$

As robot and obstacle has own velocities we need to find time necessary to get collision point for robot and obstacle, if time is neglected small collision can be appeared. For that reason we should be sure that robot and obstacle will be in safe distance when one of them will reach collision point. When the robot velocity is changing the collision point(G) should be calculated again. G is a time difference beetwen robot reaches the point and obstacle reaches the point. For avoiding obstacle we should maximize the G

$$G=|\boldsymbol{T}_{\boldsymbol{rob}}-\boldsymbol{T}_{\boldsymbol{obs}}| \rightarrow \mathbf{max}$$

$$\boldsymbol{G}=\left|\frac{\boldsymbol{PCl+R+RobotWidth}}{\boldsymbol{v}_{\boldsymbol{obs}}}-\frac{\boldsymbol{RTC-RobotLength-R}}{\boldsymbol{v}_{\boldsymbol{rob}}}\right|>\boldsymbol{\varepsilon} \quad (6)$$

ε is a small number which added to secure the robot movement, it can be 1/10 part of RobotLength

RobotWidth-is a width from the center to wheels(so total width/2)

RobotLength- is a length of robot from the center to the back(total length/2)

R is a radius of circumscribe sphere around obstacle.

If G is too small it means obstacle will pass near robot, if G is too big then obstacle will not be near robot in collision time so robot can move with the same velocities.

For new behavior we should add new layer of intelligent on our system.Firstly, in some situations it is hard to define what variable to add to Fuzzy Logic Controller, G-ε will be the best solution of this problem.

$$G - \varepsilon > 0$$

If G-ε>0 then collision will be avoided with this velocity and trajectory

If G-ε<0 collision will happen, obstacle avoidance behavior should be activated

If G-ε=0 collision can happen

Secondly, we should define rules, based to which output variable the changes of G-ε will be applied. In my opinion linear velocity can be the answer of that question. Because the changes in linear velocity also can change time to reach the collision point

Figure 24, shows membership function of G-ε.Finally, fifth input of the system is G-ε, which has following fuzzy sets: Too Small, Small Big, Very Big

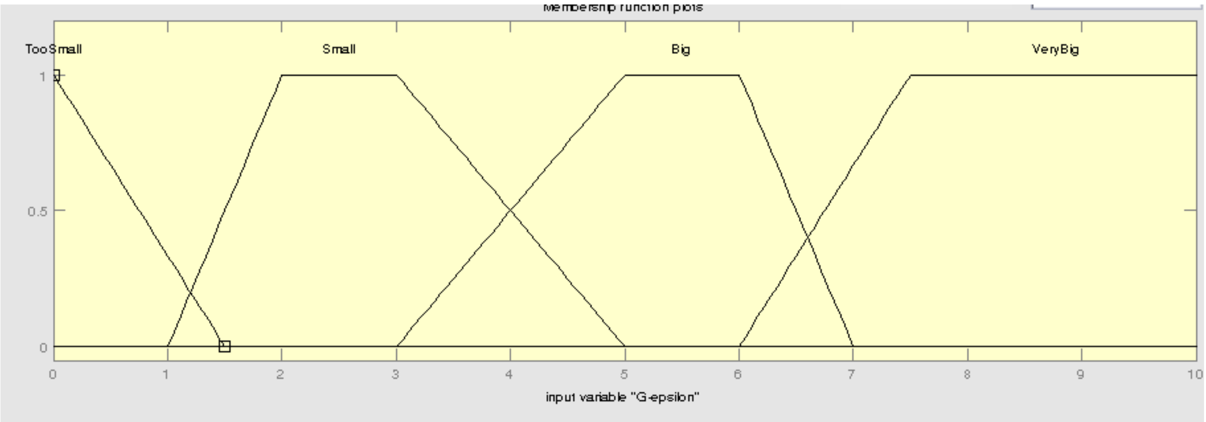


Figure 24. Membership function for input G-ε

Fuzzy rules-Rule Base:

1. IF (G is TooSmall) THEN (Linear is VeryLow)
2. IF (G is Small) THEN (Linear is Low)
3. IF (G is Big) THEN (Linear is High)
4. IF (G is VeryBig) THEN (Linear is VeryHigh)

The same strategy with List will apply on moving obstacle as a static obstacles after calculations on multiple objects they will be put on the list. The nearest and more possible to collision obstacle will be taken from List and operation on avoiding obstacle will be activated for this obstacle

6.4 Testing

In this section we will test the system obtained through the paper.

In first section we will built simulation environment. This is done by MATLAB/SIMULINK. Results copied to the file and the graphs are built by Visual Basic program in Excel. The kinematics of the system was discussed in Chapter 5.3, according to this chapter kinematics of the system was built.

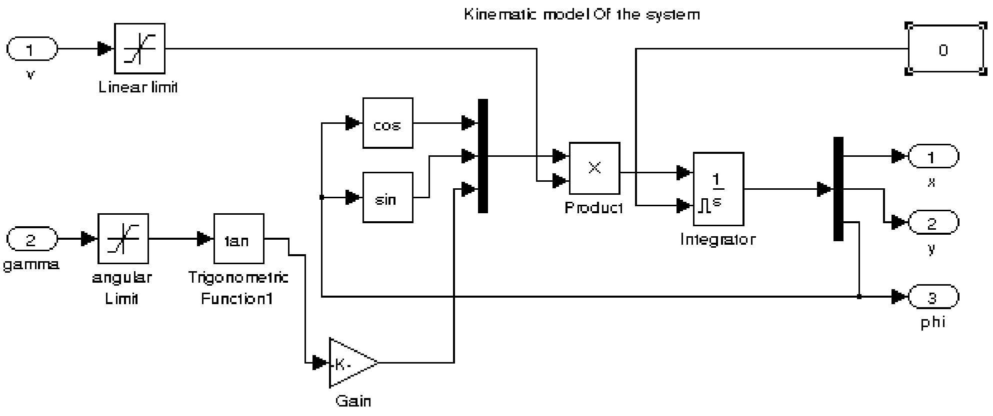


Figure 25. Kinematics of the system

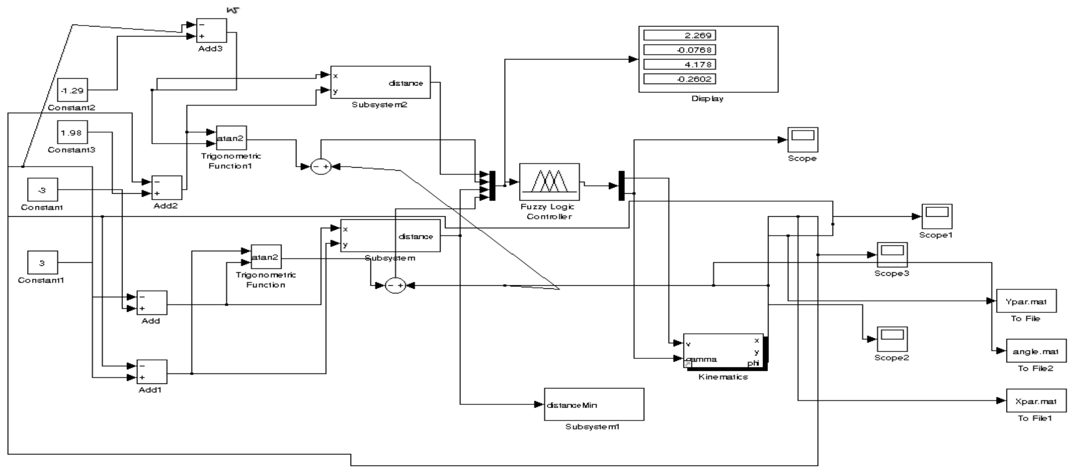


Figure 26 the entire model in Simulink;

During the testing section, both behaviours(target reaching and obstacle) has been tested. One of the tests I present in the dissertation as an example: the target destinaion was defined as (-3; 3). As we proposed in our system that the robot's length is 0.2 then the minimum distance between the target and robot is put 0.3 it allows robot to use its arm to operate on it or if it is only coordinate of the object to not collide with it. Figure 27 shows the system without obstacle and figure 28 shows with added obstacle.

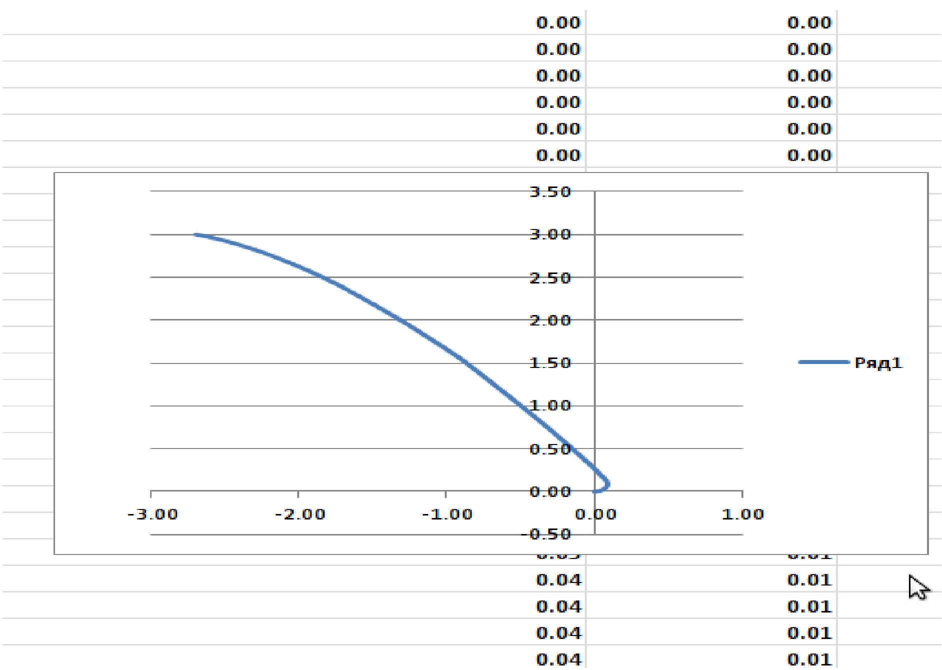


Figure 27. Goal reahing behavior

-2.10	2.98
-2.10	2.98
-2.10	2.98
-2.10	2.98
-2.10	2.98
-2.10	2.98
-2.10	2.98
-2.10	2.98
-2.10	2.98
-2.10	2.98
-2.10	2.98
-2.10	2.98
-2.10	2.98
-2.10	2.98
-2.10	2.98
-2.10	2.98
-2.10	2.98

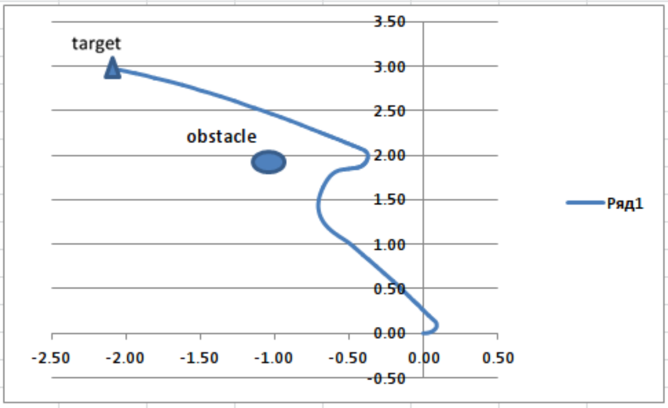


Figure 28. Obstacle avoidance behaviour

6.5 Evaluation

All aims defined in chapter 1.2 have been implemented. Plus, all functional requirements have been implemented. The system is designed on matlab and can be moved to different operating systems. Mamdani algorithm used as algorithm for converting inputs to outputs. The aims was to create the cheapest and autonomous system which can move in various environment and reach the goal. As we proposed on sections 6.1 , 6.3.1, 6.3.2 the contoller is able to handle different situations. This shows that system is supporting extensibility requirement.

Furthermore, the system upgrade was quite easier we did not change the system we were only adding new layers of intelligent and rules and system is ready to handle other processes. The system was tested with many inputs and it showed high level of robustness.

Different technologies for developing the system. Matlab environment used for handling FCL(Fuzzy Control Language). Simulink used for simulation the system. Visual Basic used for obtaining results of simulation. Ubuntu 12.04(Visual Basic is used by wine) is used as a operating system during all project.

According the results used sensors, robot is using cheap devices to detect the obstacle and its parameters, Dopler effect used for speed measurement ultrasonic sensors for distance measurement.

6.6 Future Work

This project is only the beginning of whole project which I want to do. It helped me to understand the robotics from inside. In future, the system should be improved to deal with real robot an real environment. The following should be done by me in future:

-Implementation of hardware and the use of real robots with the system.

-Adding complete task of moving obstacles and moving target

-Implementation of user and robot interface program

-object recognition, when the target characteristics(size, speed and so on) are given however the desired coordinates are not given

7. Conclusion

The project is devoted to solve the actual and current problem, creation of intelligent system for contolling autonomous robotic systems based on multifunctional information system . The system has successfully developed

This project ends up with the creation of tested system which meets all funcional reuqirements. Moreover, it is classic implementation of Fuzzy Logic which is generally the system on which we can apply various new layers. Furthermore, traditional controllers such as PID controller can be also upgraded to use fuzzy sistems.[18]. These approaches gives traditional controller more robustness and more ability to be easily upgraded(adding new ability to handle some processes).

These systems can also be improved by adding new layer of intelligence which is able to get the target, which is not static(moving target).

The main results of the dissertation can be grouped as following:

1) Proposed algorithm to detect moving obstacles, which are using cheap sensors based on Doppler effect, and avoid them by mathematical calculations and fuzzy logic Controller
2) By analysing different systems it is clearly seen that Fuzzy Logic Controller is better aproach to handle these type of systems
3) The kinematics and dynamics of the system are clearly stated
4) The software for simulation of Fuzzy Logic controller implemented and tested.

Although the system has success in testing and all functional requirements are implemented. This is unlike that indormation after simulation should be used in other environment(Visual Basic, Excel) to provide information about the robot movement.

But the software is providing all functional requirements and at least two of non-functional requirements. However, working on these project was quite interesting and useful. Now I have understanding of at least 4 system and knowledge related to Robotics.

References:

1. Lotfi Asker Zadeh “Fuzzy Sets, Fuzzy Logic, and Fuzzy Systems” 1996
2. Peter Jackson “Introduction to Expert Systems” February 1999
3. Simon Haykin “Neural Networks and Learning Machines” March 2009
4. Kevin Gurney “An Introduction to Neural Networks” 1997
5. Teuvo Kohonen “Self-organization and associative memory” 1988
6. Hans-Jargen Zimmermann “Fuzzy Set Theory-And Its Applications” 2001
7. Chun Chin Lee “Fuzzy Logic in Control Systems: Fuzzy Logic Controller” April 2, 1990
8. George Bojadziev Maria Bojadzieva “Fuzzy Sets, Fuzzy Logic, Applications” 1995
9. “Fuzzy Logic Control vs. Conventional PID Control” M. I. H. Nour, J. Ooi, , and K. Y. Chan International Conference on Intelligent and Advanced Systems 2007
10. Vamsi Mohan Peri “Fuzzy Logic Controller For An Autonomous Mobile Robot” May, 2005
11. Gyula Mester “Sensor Based Control of Autonomous Wheeled Mobile Robots” University of Szeged, Department of Informatics 2009
12. Muhammad Wali Ullah Bhuiyan and Fayaz Shahdib Chowdhury “Obstacle detection and size measurement for autonomous mobile robot using sensor fusion.” October, 2012
13. Phillip J McKerrow, Sherine M. Antoun “Research Into Navigation with CTFM Ultrasonic Sensors” The University of Wollongong April 2007
14. Xiaoyu Yang, M. Moallem, R. V. Patel “A Layered Goal-Oriented Fuzzy Motion Planning Strategy for Mobile Robot Navigation” December 2005 Dept. of Electr. & Comput. Eng., Univ. of Western Ontario, Ont., Canada
15. H. Maaref , C. Barret “Sensor-based navigation of a mobile robot in an indoor environment” December 1998
16. Jongcheol Kim and Yasuo Suga “An Omnidirectional Vision-Based Moving Obstacle detection in Mobile Robot” December 2007
17. Joseph Stiles Beggs “Kinematics”. 1983
18. Constantin Volosencu “Tuning Fuzzy PID Controllers” 2011

Appendix A

Contents of the DVD

-Project Report :

CD/dissertation/project.pdf

-codes

CD/codes/

Run source code

Firstly, environment for running code should be installed:

- Linux Ubuntu 12.04(with wine)/Windows
- Matlab
- Simulink
- Microsoft Excel

All files from codes catalogue should be copied the same place and they should be in the same folder. In matlab we should run:

>>fuzzy('last'); and then export to workspace by menu(File>export to Workspace)

After this is done simulink model(last1) can be run (by constant variables on the left).

After simulink is finished in matlab we should run rob.m file(it will create 2 csv files: Xp.csv and Yp.csv)

>>rob

After files created run **start.xlsm** file and press button extract data. Data will be inserted into the excel file. As we know matlab is putting in files: Array of variable and Timeseries. For that reason all numbers on A and B columns while they are equals should be deleted the remains will be coordinates of robot motion.