Development of Stable Telecontrol and Teleoperation of The AutoMerlin Mobile Robot

von der Naturwissenschaftlich-Technischen Fakultät (Fakultät IV),
Department Elektrotechnik und Informatik
der Universität Siegen

zur Erlangung des akademischen Grades
Doktor der Ingenieurwissenschaften
(Dr.-Ing.)

genehmigte Dissertation
von

M.Sc. Aamir Shahzad

1. Gutachter : Prof. Dr.-Ing. Hubert Roth

2. Gutachter : Prof. Dr. Mian Muhammad Awais

Tag der mündlichen Prüfung: 01 June 2017

Abstract

Mobile robots have played a crucial role since their inception and the research in the field of mobile robotics is ever increasing day by day because they have vast applications. These robots have brought easiness and comfort to human's life. There are different ways in which mobile robots navigate, e.g. some move autonomously while others have a human operator as their motion planner and executor. The involvement of a human operator gave rise to the idea of the development of teleoperated robots. Teleoperated robots have helped in planetary explorations, landmines clearance, and explosive materials handling in dangerous environments, e.g. they are actively involved in the radioactive material handling of the nuclear power plants. Telecontrol and teleoperation consist of a human operator, a control station, a communication medium, a slave robot, and the remote environment. Initially, a dedicated medium like a radio link was used as a communication link between the control station and the slave robot. The communication involved some delay in the data and signal transmission. The delay was constant and the control laws were modified according to the new scenario. The dedication of a specific link for the communication was costly and it could not attract many researchers. But with the advent of the Internet, the researchers started utilizing it for the development of teleoperated robots. The main advantage of the Internet is its availability and cost. But as the Internet is a shared medium, therefore, it has an inherent delay in it. This delay is random in nature. Apart from delay it also has other limitations like packet drops, duplication of data, out of order arrival of packets, etc. The design of a controller with such limitations was a challenging task. Therefore, different solutions have been proposed to develop a stable telecontrol of the mobile robot AutoMerlin. At first, the Event-based Control was implemented to limit the execution time of the input commands so that the robot remains stable even in the presence of a delay. After that fuzzy soft computing was employed to design a controller, which was immune to a network delay. It has two inputs, one input comes from the human operator and the other input comes from the sonar sensors which map the environment and calculate the distance to the obstacle and provide it to the speed controller for an appropriate output speed. An ancillary intelligence has been provided to

avoid obstacles autonomously in case of connection loss between the human operator and the mobile robot. Finally, Time Domain Passivity Control has been implemented to design a bilateral controller. It is a non-model-based controller which doesn't add any extra damping to the system and also there is no need to make any compromise on any parameter of the system. It is based on Two-port network. The controller has been designed with only one port active. Then, the work has been extended to design a bilateral controller with both ports active having a constant and stochastic delay and other network impediments using Time Delay Power Network approach. The force feedback has been rendered back to the human operator. Several experiments have been performed to test the performance and robustness of the controllers. The performance evaluation data has been plotted. Stable teleoperation has been achieved and it has been deployed to the mobile robot AutoMerlin. In the end, the Probabilistic Neuro-fuzzy and ANFIS have been used to design a leader-follower setup of multiple mobile robots. A simulation has been done to visualize the performance of the proposed algorithm and the Probabilistic Neuro-fuzzy has been implemented on the real robot.

Zusammenfassung

Mobile Roboter haben seit ihrer Entwicklung eine entscheidende Rolle gespielt und die Forschung im Feld mobiler Robotik nimmt andauernd zu, da diese in vielen Anwendungen eingesetzt werden. Diese Roboter haben Leichtigkeit und Komfort in das Leben des Menschen gebracht. Es gibt verschiedene Arten in denen mobile Roboter navigieren. Manche bewegen sich autonom, während andere einen menschlichen Anwender zur Planung und Ausführung ihrer Bewegungen haben. Die Beteiligung eines menschlichen Anwenders förderte die Idee der Entwicklung von teleoperierten Robotern. Teleoperierte Roboter werden bereits bei planetarischen Erkundungen, bei der Landminen Beseitigung und dem Transport von explosivem Material in gefährlichen Umgebungen eingesetzt. Zum Beispiel werden sie aktiv zur Handhabung von radioaktiven Materialien in Kernkraftwerken eingesetzt.

Zur Fernsteuerung und Tele-Operation wird ein menschlicher Anwender, eine Kontrollstation, eine Kommunikationsverbindung, ein Roboter und ein entferntes Einsatzgebiet benötigt. Anfangs wurde eine Radioverbindung als eine Kommunikationsverbindung zwischen dem Sender und dem Roboter verwendet. Die Kommunikation beinhaltete eine Verzögerung bei der Daten- und Signalübertragung. Die Verzögerung war konstant, und die Regelungsgesetze wurden entsprechend dem neuen Szenario modifiziert. Die Entwicklung einer spezifischen Kommunikationsverbindung war kostspielig und konnte nicht viele Forscher überzeugen. Mit dem Aufkommen des Internets begannen Forscher dieses in die Entwicklung von teleoperierten Robotern einzubinden. Der Hauptvorteil des Internets ist seine Verfügbarkeit und die geringen Kosten. Da das Internet ein geteiltes Kommunikationsmedium ist, hat es eine inhärente Verzögerung. Neben der Verzögerung können auch Informationspakete verloren gehen, doppelt ankommen oder zu einer anderen Zeit eingehen.

Deshalb werden verschiedene Lösungen vorgeschlagen, um eine stabile Fernsteuerung vom mobilen Roboter AutoMerlin zu entwickeln. Zuerst wurde eine Ereignis basierte Regelung eingeführt, um die Ausführungszeit der Eingangsbefehle zu begrenzen , so dass der Roboter, trotz auftretender Verzögerungen in der Signalübertragung, stabil bleibt. Hiernach wurde Fuzzy-Logik verwendet um einen Regler zu entwerfen, welcher unabhängig von Netzverzögerungen funktioniert.

Dieser hat zwei Eingänge. Ein Eingangssignal kommt von dem menschlichen Anwender, während der andere Eingang von den Sonarsensoren kommt, die die Umgebung kartografisch darstellen, die Entfernung zum Hindernis berechnen und diese Information dem Geschwindigkeitsregler zur Verfügung stellen der eine angemessene Ausgangsgeschwindigkeit bestimmt. Um Hindernissen autonom im Falle von Verbindungsverlust zwischen dem menschlichen Anwender und dem mobilen Roboter auszuweichen, wurde eine untergeordnete Intelligenz implementiert. Zuletzt wurde Time Domain Passivity Control eingeführt, um einen bilateralen Regler zu entwerfen. Dabei handelt es sich um einen Nicht-modellbasierten Regler, der dem System keine zusätzliche Dämpfung zufügt. Ausserdem müssen keine Kompromisse bei den Parametern des Systems eingegangen werden. Dies basiert auf einem zwei-Kanal Netzwerk. Der Regler wurde mit nur einem aktiven Kanal entwickelt. Unter Verwendung des Time Delay Power Network Ansatzes wurde die Arbeit um einen bilateralen Regler mit zwei aktiven Kanälen, einer konstanten und stochastischen Verzögerung und anderen Netzwerk Beeinträchtigungen erweitert. Das Force Feedback wurde an den menschlichen Anwender zurückgeführt. Verschiedene Versuche zur Überprüfung der Leistung und der Stabilität des Reglers wurden durchgeführt und dargestellt. Stabile Teleoperation wurde erreicht und an dem mobilen Roboter AutoMerlin eingesetzt. Zuletzt wurden der probabilistische Neuro-fuzzy und ANFIS eingesetzt, um ein leader-follower Setup von multiplen mobilen Robotern aufzubauen. Die Leistung des vorgestellten Algorithmus wurde in einer Simulation dargestellt. Der probabilistische Neuro-fuzzy wurde auf dem realen Roboter implementiert.

Acknowledgements

First of all, I would like to thank Allah Almighty for granting me the opportunity to pursue my Ph.D. in the field of Mobile Robotics and Telecontrol. It is indeed His blessing that I am able to write and submit my thesis. During Ph.D. I have gone through literature review, implemented different algorithms and published the results at different International Conferences and Journals. This contribution was not possible without the active involvement of Prof Roth. Therefore, I am thankful to Prof Roth for supervising me during my Ph.D. His trust and confidence in me have enabled me to do this work. Prof Roth vision, guidance, and dynamic leadership qualities have benefited me at each and every step during my Ph.D. I would also like to thank Prof Mian Muhammad Awais from LUMS Lahore, Pakistan for his invaluable suggestions for the improvement of my thesis and also his role as a second supervisor.

I would like to thank HEC/DAAD for providing me funding for the Ph.D. I am also thankful to the International Office of the University of Siegen for proving scholarship in the last phase of my Ph.D.

I would like to appreciate a few people whom I met in Germany like Muhammad Mubeen Anwar, Muhammad Tanveer Baig, and Muhammad Khurram Ehsan. These people have made my stay in Germany pleasant and I have really enjoyed their friendship. I am also grateful to all RST like Dipl. Ing. Peter Sahm, Dipl. Ing. Christopher Hille, Frau Birgit Hoffmann and Frau Kristina Borchert

Last but not least, it is really important to acknowledge the efforts of my parents throughout my studies, especially during Ph.D. Their untiring support, motivation, and prayers have always helped me. I am deeply indebted to my brothers and sisters for their adored love. I am deeply thankful to my wife for her patience and understanding while I was away. All these people are a precious asset of my life.

Contents

List of Figures

List of Tables

Abbreviations

ANFIS	**Adaptive Neuro-Fuzzy Inference System**
FIN	**Free Intelligent Navigation**
FLC	**Fuzzy Logic Control**
f_d	**Force due to Distance**
f_e	**Environment Force**
f_f	**Force of Friction**
f_h	**Human Force**
PC	**Passivity Controller**
PD	**Proportional Derivative**
PFLS	**Probabilistic Fuzzy Logic System**
PO	**Passivity Observer**
s_ρ	**Scaling Factor**
TDPC	**Time Domain Passivity Control**
TDPN	**Time Delay Power Network**
$V_{m,M}$	**Velocity of the Master Device**
$V_{s,S}$	**Velocity of the Salve Robot**

Physical Constants

Environment Damping Coefficient	b_e	=	0.5
Environment Elasticity Coefficient	k_e	=	0.05
Friction Coefficient	μ	=	0.3
Normal Reaction	N	=	98.1N

Symbols

B	Damping Matrix
d	Delay
f	Force
H	Hybrid Matrix
M	Inertia Matrix
N	Normal Reaction
P	Power
R	Real Number
s	Scattering Variable
S	Scattering Operator
t	Time
T	Time Delay
u	wave variables
Z	Impedance

Dedicated to my family

Chapter 1

Introduction

The skill of controlling the remote, inaccessible and dangerous environment with the help of a robot is called teleoperation. *Tele* stands for the distance and *Operation* means an action or an activity[1–5]. Hence, any action/activity performed from a distance is regarded as teleoperation. It has many potential applications in the recent times. The usefulness of this approach is vivid in the medicine, nuclear power plants, deep sea and space exploration missions, land-mines clearance, surveillance and combat operations, etc. The sensory feedback from the remote site like the vision, touch and auditory is very crucial and necessary to rebuild and grasp the remote environment. Among this sensory information, the force feedback from the remote environment is very important as it plays a key role in teleoperation. The human operator experiences a sense of alertness via a haptic device while maneuvering the robot in a remote environment. The force feedback stimulates the haptic device. This dissertation describes the control approaches which have established the proprioception information exchange between the human operator and the mobile robot located in a remote place. A human operator and the remote environment have been brought into an interaction through the communication channels like LAN, Wifi or the Internet, etc[6–9].

Teleoperation is an active field of research throughout the world because of its vast applications and its modifications for the desired scenarios ranging from the manufacturing to the handling material, exploration to surgery etc. In teleoperation, a human operator induces an input energy to manipulate the distant robot. The distance can vary from a few millimeters to thousands of kilometers. This variation in the distance is the main challenge in teleoperation. The small distance

creates the accessibility problem and the long range operation induces a delay in the control loop. The energy injected by the human operator travels through a certain mechanism to the remote robot to perform the desired actions. The next section describes a few notable teleoperated robots.

1.1 Teleoperated Robots

There are so many sophisticated teleoperated robots. They have been developed so far for the various applications. A few of them have been described here with a concise detail.

1.1.1 The Victor 6000

The remotely operated vehicle Victor 6000 as shown in the Fig. 1.1, has been developed by a team of engineers and technologists during the 1990s[10]. The main purpose of this underwater teleoperated robot was to study an ecosystem beneath the water in the deep oceans. It has been used in West Africa for the continental margin ecosystem observation at a depth of 4059m. The Victor 6000 has also enabled the scientists to study the cold seep at the continental margin of the Norway.

FIGURE 1.1: The Victor 6000

1.1.2 The Mars Rover

The Lunakhod, Mariner, Ranger and Surveyor Robotic System have been developed and utilized for the space and planetary exploration[11]. These robots have performed the various tasks like the maintenance, assembly, and the inspection in the space. The Mars Rover has been shown in the Fig. 1.2, is among one of such robots which were sent to the other planets. It was deployed to the Mars in 2004. It was a semi-autonomous robot with some features purely based on teleoperation like the capturing of images from the Mars and transmitting them back to the Earth[12].

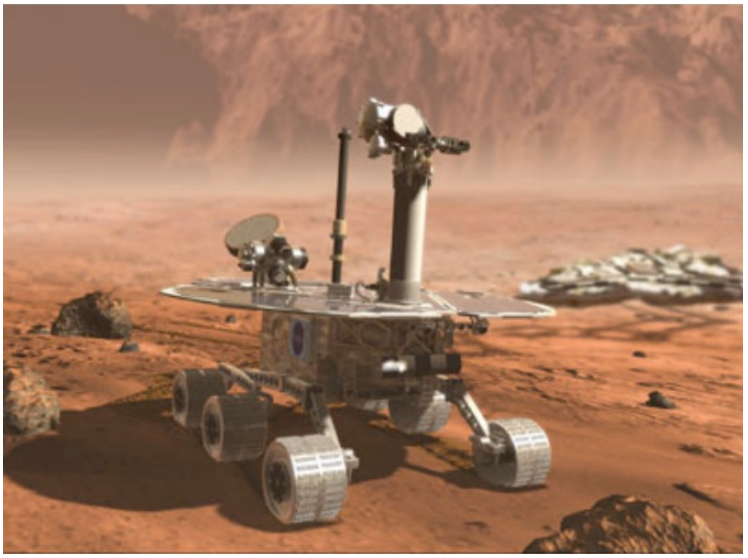

FIGURE 1.2: The Mars Rover

1.1.3 The ALTUS

The Unmanned Ariel Vehicles (UAVs) are also teleoperated for different applications. The ALTUS as depicted in the Fig. 1.3, is the best example of the teleoperated UAV. It is a state of the art flying robot for the environmental research. It is a part of the NASA[1]'s ERAST(Environmental Research Aircraft and Sensor Technology) program. The ALTUS is equipped with the sensors to carry out research about the atmosphere at high altitude. It can fly up to 71000ft height[13].

[1]National Aeronautics and Space Administration

FIGURE 1.3: The ALTUS

1.1.4 The ZEUS

Telesurgery is another best use of the teleoperation. It is helpful in conducting surgery around the world without the physical presence of a surgeon. It saves money, time and the hectic traveling by bringing the remote operation theater close to the fingertips of a surgeon. The Fig. 1.4, shows the ZEUS Robotic System which is being used for the remote surgery. The main purpose of this system is to conduct telesurgery for the removal of a gall bladder[14].

FIGURE 1.4: The ZEUS robotics system for the telesurgery

1.1.5 Numbat

The Numbat as shown in the Fig. 1.5, is primarily used for the rescue purpose in an underground coal mine in the events of an emergency. It provides an accurate

information about the disaster and state of the stranded people with the help of a state-of-the-art equipment mounted on it. It is an amalgam of a diverse range of communications, actuation, mobility, power, control and software technologies. The communication between the Numbat and a human operator is established through a fiber optic cable[15].

FIGURE 1.5: The Numbat vehicle

1.2 Paramount Interpretations

There are some important terms that are related to teleoperation of a mobile robot. They have been defined in this section[16].

1.2.1 Robotics

Robotics is the branch of engineering which deals with the conception, design, manufacturing and operation of a robot. It is a combination of Electronics, Computer Science, Artificial Intelligence and Mechatronics.

1.2.2 Mobile Robot

A mobile robot is a mechanical structure which can move. It is equipped with different sensors and control board for the locomotion control. It has different types or means of motion like the wheels or legs or chains etc.

1.2.3 Teleoperator

A robot that is being commanded to execute different tasks from a distance is called teleoperator.

1.2.4 Teleoperation

It is an extension of a human sensing and manipulative capabilities to a remote location[17].

1.2.5 Telepresence

Telepresence is the feeling of being present at the remote site. It is achieved by providing every information regarding teleoperator to a human operator in such a natural way that s/he gets the feeling of directly controlling it. Sheridan presented the idea of Telepresence measure as the quality and extent of sensory information feedback from the remote cite to a human operator along with exploration and manipulation capabilities[18].

1.2.6 Telerobotics

Telerobotics deals with the control of robots from a distance by using some communication medium like the Wifi, Bluetooth, LAN, Radio Link or the Internet. It is a merger of teleoperation and telepresence.

1.2.7 Bilateral Teleoperation

In bilateral teleoperation, a human operator sets the motion command for a robot by maneuvering a master haptic device through a communication channel and receives the sensory information along with the force feedback from the remote robot. Ideally, the robot present in a remote environment is called slave and is controlled in such a way that the slave velocity should be equal to the haptic device velocity called master velocity and the force feedback should be equal to the environmental force acting on the remote robot.

1.2.8 Impedance

Stable positioning requires at a minimum, a static relation between force and position; some spring like element must be included in the equivalent physical network. The controller must specify a vector quantity such as the desired position, but it must also specify a quantity which is fundamentally different: a relationship, an impedance, which has properties similar to those of a second-rank, twice covariant tensor; it operates on a contravariant vector of deviations from the desired position to produce a covariant vector of interface forces[19]. Imdepance is a ratio of force to velocity. $Z_m = \frac{f_m}{V_m}$ is master Impedance and $Z_s = \frac{f_s}{V_s}$ is slave impedance.

1.2.9 Transparency

The extent to which the operator commands and the feelings about the environment are preserved is called transparency of teleoperation. If the positions and the forces on the both sides are equal, then the ideal kinesthetic coupling is achieved. The force and velocity are correlated to each other by the impedance factor as $f = Zv$, and the transparency is an impedance match on the both sides[20]. Transparency of system is dependent on impedance match on both side. A system is 100% transparent if ($Z_m = Z_s$) i.e. master impedance is equal to slave impedance.

1.2.10 Force

The force exerted by the environment on a slave robot or teleoperator is called environment force represented by f_e. The force acting on the slave robot f_s is termed as the slave force and the force feedback acting on a master haptic device is f_m. The force applied by the human operator to set the velocity command is f_h. f_h is a resultant of f_m and the force exerted by the human arm.

1.2.11 Haptics

The science of applying a tactile sensation to a human interaction with a computer is called haptics. The user can feed the information into a computer with the help

of a haptic device and can receive a feeling of alertness from it by mean of some kind of vibrations.

1.2.12 Haptic Device

The haptic device is such a source of an input to a system which can generate a sense of physical interaction with the remote environment through a force feedback. The sensation of directly interacting with the remote environment is brought by it to a human operator[21].

1.2.13 Supervisory Control

In supervisory control, a human operator intermittently programs and continuously receives the information from a computer which is an integral part of a control loop.

1.3 Components of a Teleoperation System

The master haptic device along with a computer acts as a control station, which sets the different commands for the mobile robot. The force feedback from the environment is played over a haptic device. The slave mobile robot AutoMerlin executes the motion commands given by a human operator through a master haptic device. The communication medium is always chosen based on the applications and resources. For the long distance applications like the deep space missions, a dedicated Radio Link is used. This medium has an inherent delay, but it is measurable and deterministic in nature. Most of the applications use the Internet as a communication medium in the daily life and industrial activities. There are some challenges involved in teleoperation with the Internet like the time delay, and distortion in data.

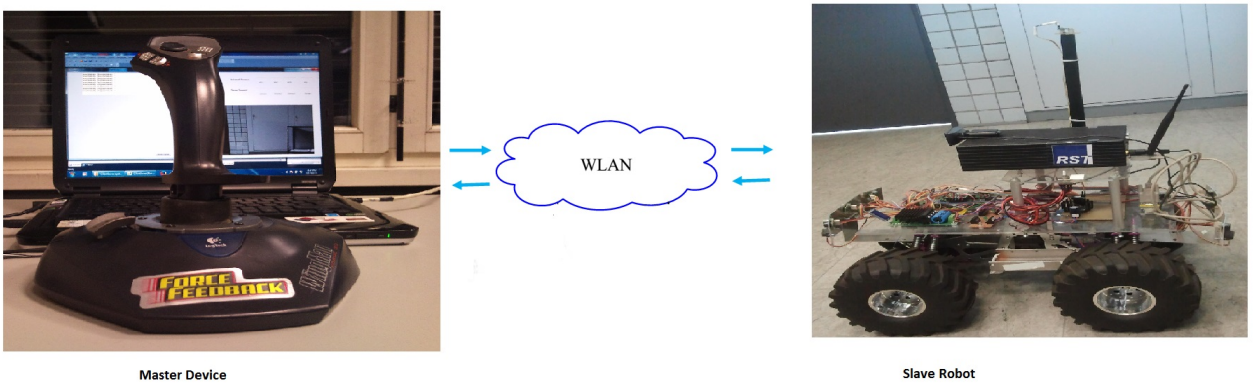

FIGURE 1.6: The mobile robot teleoperation

1.4 Thesis Outline

This thesis consists of nine chapters. The basic idea of telecontrol and teleoperation has been discussed in the Ch. 1. It also has a few state of the art teleoperators. Ch. 2, sheds light on the Problem Description. It contains some plots of a linear time invariant system. Its has been established in this chapter that the delay has an adverse effect on the performance and stability of a control system. Ch. 3, has a brief literature review. It describes a few notable approaches which have been used for the teleoperation of the robots.

Ch. 4, has implementation of the Event Based Control for the teleoperation of a mobile robot. It has the relevant theory as well as the experimental results. Ch. 5, describes the implementation of the Fuzzy Logic in telecontrol. It has all the steps involved in the design of a controller. The performance has been enhanced by integrating FIN (Free Intelligent Navigation) Algorithm which helps the robot in avoiding the obstacles autonomously. Ch. 6, presents the implementation of Time Domain Passivity Control. The controller has been designed with a bad communication link. Ch 7, extends the notion of Time Domain Passivity Control and contains the controller design which is effective in a constant as well as random delay scenarios. Ch 8, has the description of a team of robots and PFLS and ANFIS has been used to design a leader follower formation of the mobile robots. Ch. 9, describes the Conclusion and Future Work.

Chapter 2

Problem Description

There are some issues which deteriorate the performance of a teleoperated system. The utmost is the time delay. This delay is due to some distance between the local and the remote location and it causes a time gap between the issuance of a command and its execution. The effect of the time delay on the stability of the bilateral teleoperation was presented in 1966[22]. Since then various methods, approaches, and techniques have been proposed to address this issue. The force feedback from the remote environment was introduced in the same year and it greatly enhanced the teleoperation performance, but its inclusion raised many issues and destabilization is most important among them. Another issue pertaining to teleoperation is a correct presentation of the environment to a human operator so that he can take appropriate decisions. For example, shadows and curved areas cannot be viewed using a narrow view angle camera, especially in an environment with a bad illumination and several obstacles.

2.1 Effect of Delay on Stability

The telecontrol suffers from the time delay due to communication between the control station and the robot. This delay depends on the distance between the control station and the mobile robot. It has a fix value when the distance is constant and a dedicated medium of communication between the control station and the robot is used, like planetary exploration or it has the random value like communication through the Internet. This delay is fatal for teleoperation of a mobile robot. This delay causes the system to become unstable.

Nyquist plot can elaborate the effect of a time delay in the control loop. In a control process, when the reference input and feedback become nearly equal, then the loop gain is greater than unity in the operational frequencies range for all the physical plants[23]. In such cases, the negative feedback becomes positive if the delay is more than half of the time constant of the system. Consequently, the controller will keep on adding energy into the system and the input would keep on growing leading to an instability. Frequencies of interest are relying on the transparency. For higher a transparency, the upper bound limit of the frequencies of interest should be increased at the cost of smaller time delay or reduced stability margin. Thus, stability is an inversely proportional phenomenon to the transparency of teleoperation.

The first order system is given in the Eq. 2.1, which can be considered for the problem statement. It is a linear time-invariant system with an input delay d.

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t-d) \\ y(t) &= Cx(t) \end{aligned} \tag{2.1}$$

The transfer function of the system can be written as given in the Eq. 2.2.

$$\frac{y(s)}{u(s)} = P(s) = P_{\circ}(s)e^{-ds} \tag{2.2}$$

$P_{\circ}$ of the Eq. 2.2, can be calculated as follows:

$$P_{\circ} = C(sI - A)^{-1}B \tag{2.3}$$

The closed loop transfer function of the Fig. 2.1, with $G(s) = K$, $H(s) = \frac{1}{\tau s+1}$,

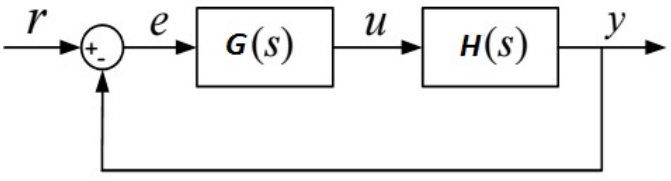

FIGURE 2.1: A close loop control system

and input delay d, can be written as given in the Eq. 2.4.

$$\frac{y(s)}{r(s)} = \frac{Ke^{-ds}}{1 + \tau s + Ke^{-ds}} \tag{2.4}$$

The characteristic equation of the transfer function of the delayed system has a boundless entity which results in a distributed system with infinite poles. The delay also causes a phase shift as described in the Eq.2.5.

$$\phi(\omega) = -\omega d \tag{2.5}$$

Hence, time delay and operating frequencies are inversely proportional to each other for given stability margin. In other words, the stability decreases as $\frac{d}{\tau}$ increases. The plant in the Eq. 2.4, becomes delay free as K approaches to infinity and $\frac{d}{\tau}$ approaches to zero. The root locus plot of $\frac{d}{\tau}$ to be infinity and K approaches to 1 with the ratio of delay to the time constant of the system to be 1×10^6, has been shown in the Fig. 2.2.

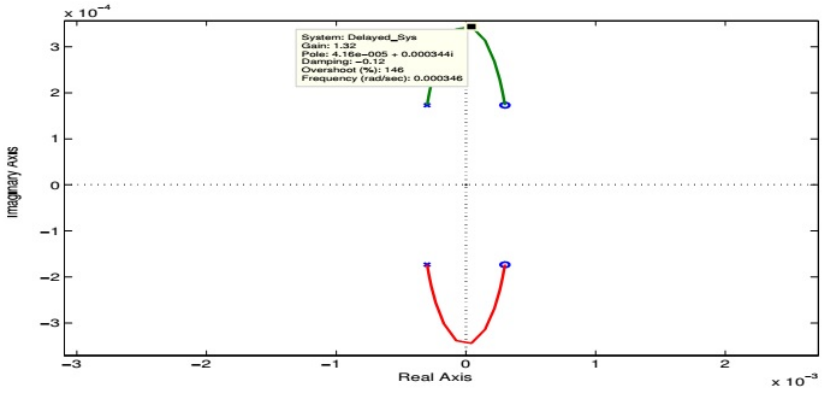

FIGURE 2.2: The root locus of the plant with a delay

Let consider the open loop system with delay d as

$$\frac{y(s)}{r(s)} = \frac{Ke^{-ds}}{1 + \tau s} \tag{2.6}$$

The Nyquist plot has been shown in the Fig. 2.3, and Bode plot has been shown in the Fig. 2.4, when d=0. It is clear from these plots that the system is stable with infinite gain margin and -180° phase margin.

The delay makes the system unstable. The Fig. 2.5, shows the Nyquist plot and the Fig. 2.6, shows the Bode plot. These plots have been drawn with a delay equal to half of the time constant of the system. The system with unity gain shifts toward instability. The Bode plot has a gain margin of 11.6dB and a phase margin of $-180°$.

The slight increase in gain from unity to 5 makes the system completely unstable as shown in the Fig. 2.7, and 2.8. The Nyquist plot encircles (-1,0) point multiple times due to unstable behavior of the plant. The Bode plot has a gain margin of -2.37dB and a phase margin of $-38.8°$.

The inference from the above discussion is that the delay is the main cause of instability in the control system. Teleoperation has a much higher delay in comparison to desired command frequency. Therefore, it is an inevitable problem for stable teleoperation.

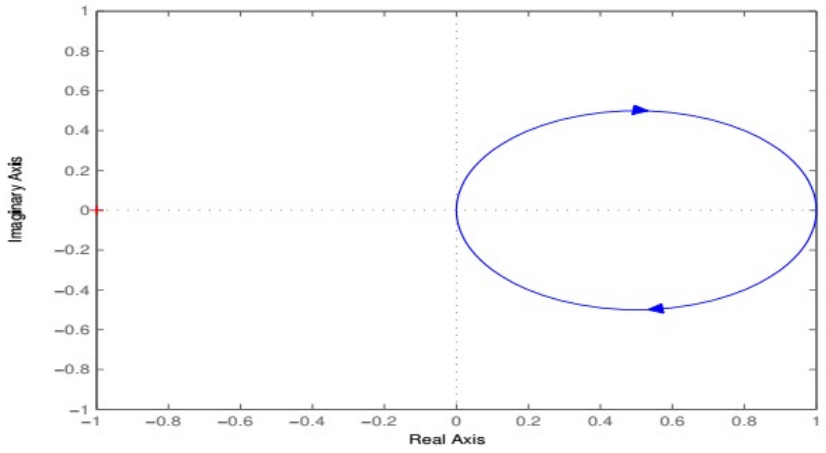

FIGURE 2.3: The Nyquist plot for an open loop system without a delay

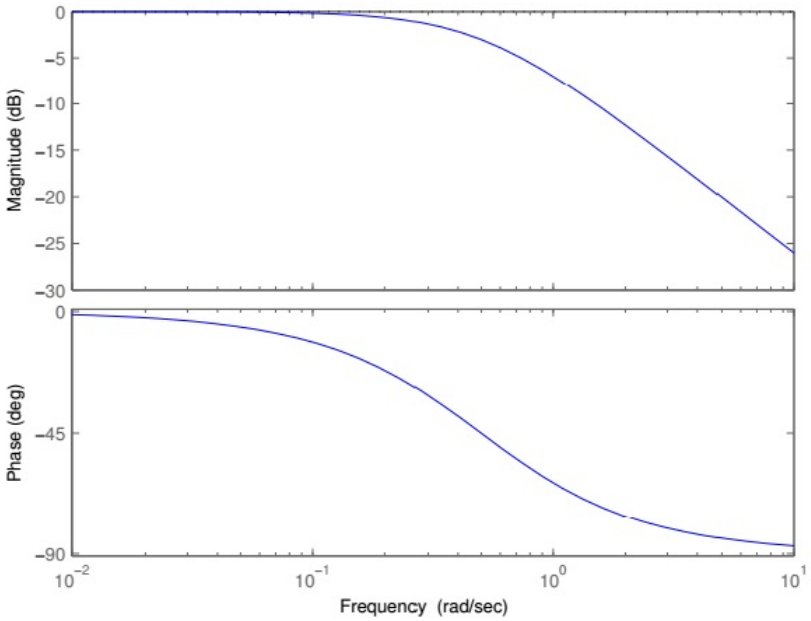

FIGURE 2.4: Bode plot for an open loop system without a delay

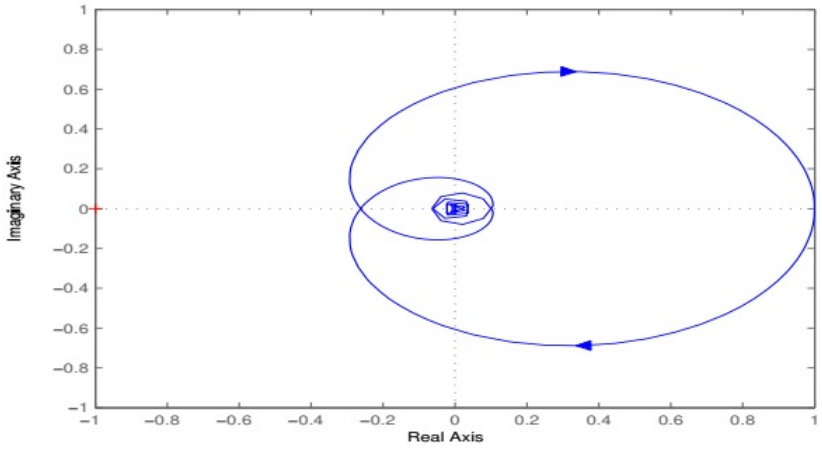

FIGURE 2.5: The Nyquist plot with a delay equal to the half of the time constant

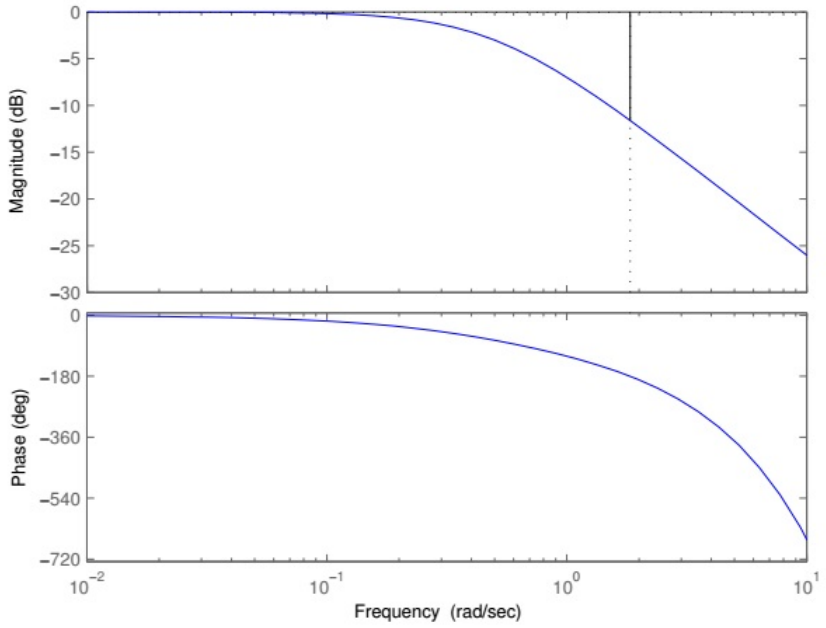

FIGURE 2.6: Bode plot with a delay equal to the half of the time constant

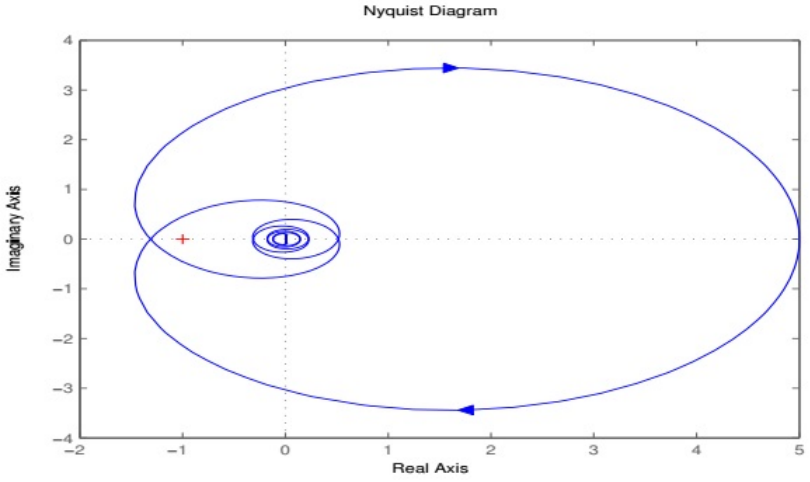

FIGURE 2.7: The Nyquist plot with a delay equal to the half of the time constant and a gain of five

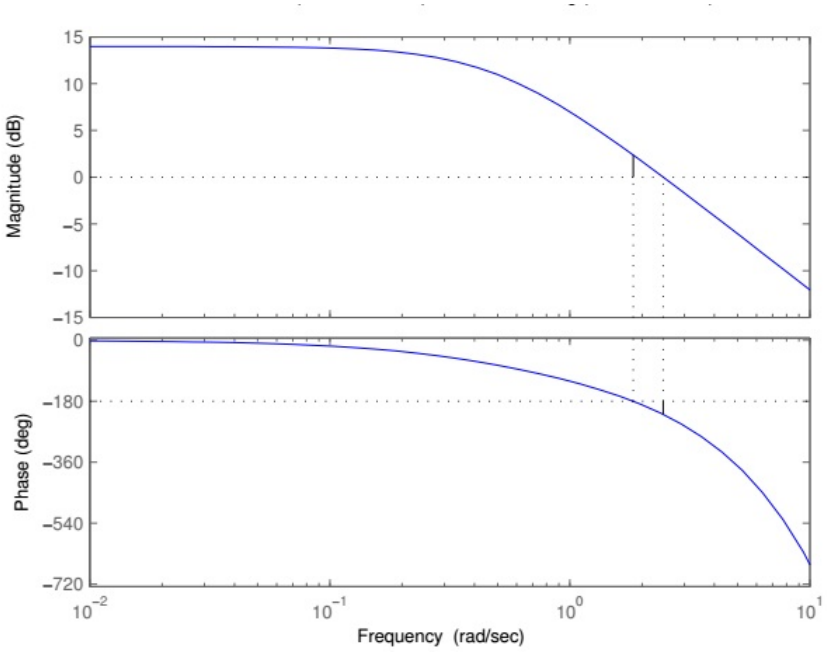

FIGURE 2.8: Bode plot with a delay equal to the half of the time constant and a gain of five

2.2 Teleoperation of AutoMerlin

The following is a list of hardware involved in teleoperation of AutoMerlin mobile robot.

- Haptic device
- Data processor, e.g. computer
- Slave mobile robot AutoMerlin

The haptic device used is called WingMan FORCE 3D from Logitech as shown in the Fig. 2.9. It is a true force feedback device that delivers realistic effects. It has seven programmable buttons with a comfortable stick and also has a weighted base for stability. It provides an experience of adrenaline pumping during operation so that the operator can immerse himself in the desired task with great interest.

FIGURE 2.9: A haptic force feedback joystick

This joystick is connected to a computer through Microsoft Direct X. An algorithm developed in C sharps maps the inputs from the joystick and forwards them to slave robot. The Graphical User Interface as shown in the Fig. 2.10, displays the proximity sensor values and the live video feed from the slave robot. TCP/IP and UDP protocols have been used for data and visual image transmission respectively.

AutoMerlin as shown in the Fig. 2.11, is based on a chassis of a vehicle model type HPI Savage 2.1. It is a two-track vehicle that is equipped with Ackermann

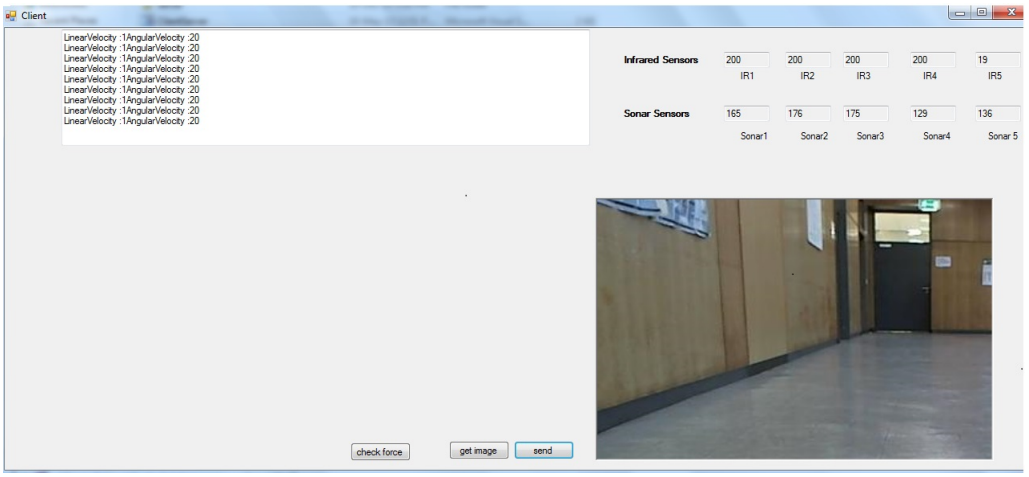

FIGURE 2.10: The GUI

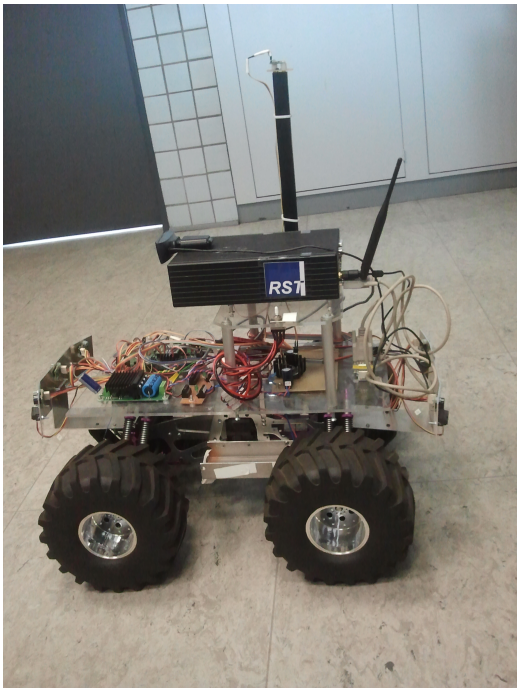

FIGURE 2.11: The AutoMerlin mobile robot

steering. This means that it can turn with the two front wheels. Additionally, the robot is four-wheel driven and the force is transmitted to all four wheels in the same direction. Thus, similar conditions prevail here as in a conventional four-wheel driven automobile. Turning on the spot is not possible. AutoMerlin is using the powerful DC motor TruckPuller 3, 7.2V batteries for the drive, and the powerful servo motor HiTec HS-5745MG for steering. The drive motor is also equipped with an optical position encoder which can be used to determine the speed and direction. The dsPic microcontroller controls the behavior of the robot depending on different factors like user preferences, environmental influences, and

controls. An x86 microcomputer is an integral part of the robot. It contains the user program through which the user can parameterize the robot. The computer also has a wireless interface to operate the robot wirelessly.

Chapter 3

Literature Review

The inception of teleoperation was introduced during the 1940s when the first mechanically controlled master and slave teleoperator was built by Goertz. The design was later improved in 1954 by replacing the mechanical parts with the electrical force reflecting position servomechanism so that the master and the slave can be separated[24]. The delay issue was studied first in the 1960s. Different experiments were conducted to analyze the effect of the delay on the force feedback[25]. During the 1980s the concept of Lyapunov analysis was used for the stability of teleoperation[26]. Network theory through impedance representation, hybrid representation, and passivity control were introduced in the 1990s[27–29].

3.1 Early Teleoperation Strategies

The human operator performance was evaluated during the 1960s by conducting some manipulation experiments with a delay[25, 30]. The objective was to calculate the total time necessary to perform a specific task and to study human behavior. The outcome of these experiments was the adoption of a *Move and Wait Approach* to perform the required action by a human operator. In this approach, the human operator stipulated a command and waited for its execution. In this way, the required tasks were performed in steps.

The number of individual steps taken to complete one task in *Move and Wait Approach* was denoted by N(I). N(I) was a function of difficulty of the task and it was independent of time delay. The total completion time t(I) of a single task

was calculated as given in the Eq. 3.1[31].

$$t(I) = t_r + \sum_{i=1}^{N(I)} (t_{mi} + t_{wi}) + (t_r + t_d) + t_g + t_d \tag{3.1}$$

Where t_r is the human reaction time, t_{wi} is waiting time after each move, t_{mi} is movement time, t_d is delay time and t_g is grasping time in the Eq. 3.1. The experiment ended with the conclusion that the completion time is linear with respect to the time delay.

After the development of the *Move and Wait Approach* the supervisory control approach was proposed[32, 33]. The task was performed partly by a human operator and partial autonomy was granted to the remote manipulator. Depending upon the distant controller autonomy and task complexity, there were some direct commands from the operator to the manipulator while some indirect local linguistic instructions to perform subtasks. The remote controller was made intelligent enough to execute subtasks automatically. The manipulator dynamics were not considered in this approach and focus was on the static geometric aspect of the problem.

The development of the microprocessor during the 1980s led the teleoperation to a new horizon by enabling the human operator to issue high-level commands to the teleoperator[34–38]. A special programming language pertinent to teleoperation was introduced. It was based on two types of commands, one in which human operator involvement was not required like close end effector, and the other type was the specific action commands need to be issued by the skilled operator. Many improvements were suggested later to enhance it.

3.2 Passivity Based Teleoperation

Passivity approach in teleoperation originated from the network theory is an input–output property of a dynamical system. It primarily deals with the exchange of energy between interconnected systems. The passive system should absorb more energy than it emits. It means that during any process the system cannot produce energy[39, 40]. In order to study passivity and stability, consider a linear model

of the master and the slave robot as given in the Eq. 3.2, Eq. 3.3, respectively.

$$M_m\ddot{x}_m + B_m\dot{x}_m = f_m + f_h \tag{3.2}$$

$$M_s\ddot{x}_s + B_s\dot{x}_s = f_s - f_e \tag{3.3}$$

Here $x \in \Re^1$ is the generalized coordinate for the master and slave, f represents input forces, M is the positive inertia matrix and B is a damping matrix. The nonlinear mathematical model for the master is given in the Eq. 3.4, and for the slave is given in the Eq. 3.5.

$$M_m(x_m)\ddot{x}_m + C_m(x_m, \dot{x}_m)\dot{x}_m = f_m + f_h \tag{3.4}$$

$$M_s(x_s)\ddot{x}_s + C_s(x_s, \dot{x}_s)\dot{x}_s = f_s - f_e \tag{3.5}$$

Here M is inertia matrix and C matrix contains Coriolis and Centrifugal entities. There are several properties associated with nonlinear equations of motion, such as ($M = M^T$ is positive definite) and ($M - 2C$ is skew symmetric)[41].

$$\dot{x} = f(x, u) \tag{3.6}$$

$$y = h(x, u) \tag{3.7}$$

The Eq. 3.6, and Eq. 3.7, describe the dynamic system. This system can only be passive if there exists a continuously differentiable semidefinite scalar function $V(x) : \Re^n \to \Re$(called the storage function) such that the condition in the Eq 3.8, is satisfied.

$$\dot{V} \leq u^T y \equiv \left(\int_0^t u^T(\tau)y(\tau)d\tau \geq V(t) - V(0) \right) \tag{3.8}$$

If the system is lossless then the Eq. 3.8 is modified to take the shape given in the Eq. 3.9.

$$\dot{V} = u^T y \equiv \left(\int_0^t u^T(\tau)y(\tau)d\tau = V(t) - V(0) \right) \tag{3.9}$$

Proposition: Given the positive definite and skew-symmetric properties and taking the human and the environment passive, i.e. $\int_0^t \left[f_h^T(\tau)\dot{x}_m(\tau) - f_e^T(\tau)\dot{x}_s(\tau)\right]d\tau \geq 0$, then the nonlinear system with inputs $\left[f_m^T, f_s^T\right]$ and outputs $\left[\dot{x}_m, \dot{x}_s\right]$ is passive

[1]Real number

with following storage function V.

$$V = \frac{1}{2}\begin{bmatrix}\dot{x}_m \\ \dot{x}_s\end{bmatrix}^T \begin{bmatrix}M_m & 0 \\ 0 & M_s\end{bmatrix}\begin{bmatrix}\dot{x}_m \\ \dot{x}_s\end{bmatrix} \tag{3.10}$$

V function contains the dynamics of master and slave. The differential of V should be less than net energy of the system to ensure passivity. The nonlinear system is lossless when forces are inputs and velocities are outputs. Hence the lossless system allows examining the energy exchange within the teleoperator and with the human operator and the remote environment. The effectiveness of the passivity has led to many approaches based on it for teleoperation.

3.2.1 Two-port networks

The inception of teleoperation as a Two-port network was presented during the 1980s[28, 42, 43]. The teleoperator, master device, controllers and remote environment were modeled as Two-port network.

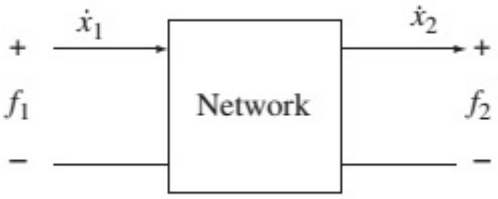

FIGURE 3.1: Two port network

The Two-port network is shown in the Fig. 3.1. It is vivid from the figure that external signals are efforts and flows. The effort corresponds to the force in the Mechanical System and voltage in the Electrical System. The flow corresponds to a velocity in the Mechanical System and current in the Electrical System. The behavior of the network is described by different matrices like Impedance Matrix or Hybrid Matrix. The Impedance Matrix relates the forces to the velocities and the Hybrid Matrix establishes the relation between the mixed force-velocity vectors. Each representation has its own applications depending upon the control input and available measured signals.

In order to elaborate the Impedance Matrix, consider the case in which the flow is input on both sides of a Two-port network. The Impedance Matrix relates the force and velocity vector of a PD[1] controlled master-slave network according to the Eq. 3.11[27].

$$\begin{pmatrix} f_1 \\ f_2 \end{pmatrix} = \begin{bmatrix} z_m(s) - c_{11}(s)_m & -c_{12}(s) \\ -c_{21}(s) & z_s(s) - c_{22}(s) \end{bmatrix} \begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \end{pmatrix} \tag{3.11}$$

z is the impedance of the master and the slave. The controller c has an effect on all the elements of the Impedance Matrix. The passivity of the network can be achieved by manipulating the elements of the Impedance Matrix to make it a positive real transfer matrix. If the force is measured at the slave side then alternative representation with the Hybrid Matrix is given in the Eq. 3.12[44].

$$\begin{pmatrix} f_1(s) \\ -\dot{x}(s) \end{pmatrix} = \begin{bmatrix} h_{11}(s) & h_{12}(s) \\ h_{21}(s) & h_{22}(s) \end{bmatrix} \begin{pmatrix} \dot{x}(s) \\ f_2(s) \end{pmatrix} \tag{3.12}$$

h_{11} is the input impedance, h_{12} is the force scaling term, h_{21} is the velocity scaling element and h_{22} is the output impedance as given in the Eq. 3.13. Ideally, it is desirable to have zero input impedance and infinite output impedance because the ideal Hybrid Matrix would achieve the kinesthetic feedback between the environment and human operator. The Hybrid Matrix would have following entities for the ideal case as given in the Eq. 3.14.

$$H(s) = \begin{bmatrix} Z_{input} & f_{scaling} \\ V_{scaling} & Z^{-1}_{output} \end{bmatrix} \tag{3.13}$$

$$H_{ideal}(s) = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \tag{3.14}$$

Hannaford proposed a method which made a Two-port network behavior ideal. He estimated the human and the environment dynamics and transmitted them in the opposite direction as shown in the Fig. 3.2[44]. Another approach of feedback linearization of equations of motion has been presented for teleoperation by Strassberg, Goldenberg, and Mills[45, 46]. Two controllers were designed for the master, one for the local position control and other for the force error between two manipulators. Additionally, two compensators provided the control signals to the slave robot. These compensators relied upon two factors i.e. the velocity

[1] Proportional derivative

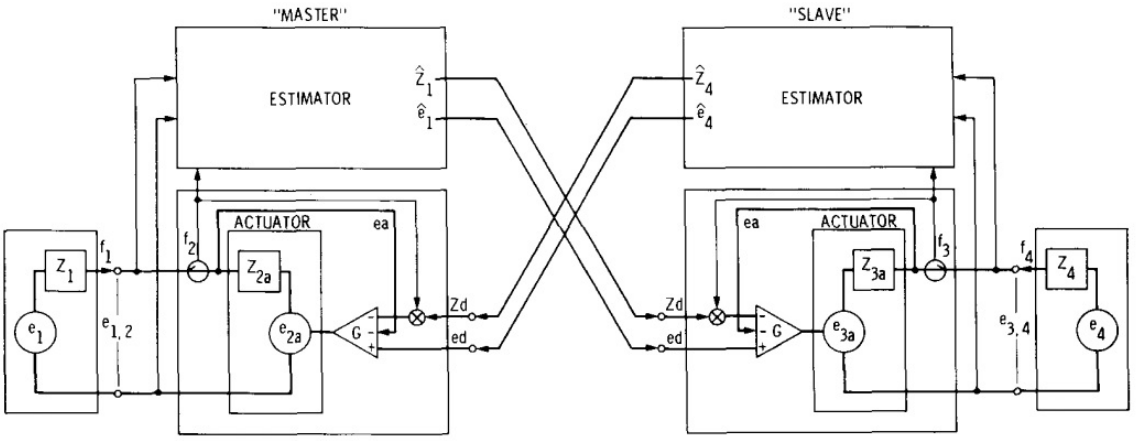

FIGURE 3.2: Master slave teleoperation architecture

difference and the local force on the slave. The design helped in attaining the ideal Hybrid Matrix condition given in the Eq. 3.14.

3.2.2 Scattering Approach

The idea of scattering variables was presented by Anderson and Spong during the 1980s for the bilateral teleoperation[29] which were previously utilized in transmission line theory. The bilateral control loop was modeled as a combination of One-port and Two-port networks as shown in the Fig. 3.3. These networks exchanged flow and effort at each port.

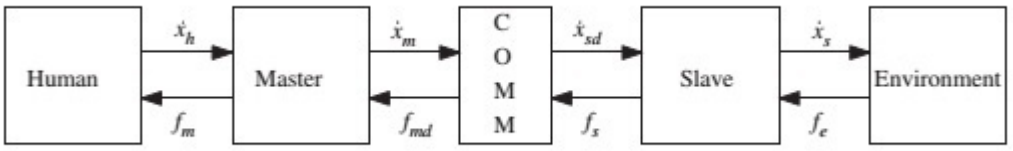

FIGURE 3.3: Teleoperation control loop

The scattering operator was defined by correlating the incident and the reflected waves. The incident wave $(f(t) + \dot{x}(t))$ and the reflected waves $(f(t) - \dot{x}(t))$ were related to each other by scattering operator S as given in the Eq. 3.15.

$$f(t) - \dot{x}(t) = S(f(t) + \dot{x}(t)) \tag{3.15}$$

The scattering matrix for a Two-port network using Hybrid Matrix was driven as given in the Eq. 3.16.

$$S(s) = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} (H(s) - I)(H(s) + I)^{-1} \tag{3.16}$$

In order to ensure passivity of a teleoperated system, the scattered wave cannot have more energy than the incident wave. Therefore, a P-port system is passive if $\| S(j\omega) \|_\infty \leq 1$, is true. It was further elaborated for a Two-port system with force f and velocity v. The net power (δP) as given in the Eq. 3.17, was calculated in term of scattering variables.

$$\delta P = P_{in} - P_{out} = f^T v \tag{3.17}$$

$$\delta P = \left(\frac{f+v}{2}\right)^T \left(\frac{f+v}{2}\right) - \left(\frac{f-v}{2}\right)^T \left(\frac{f-v}{2}\right) \tag{3.18}$$

The scattering variables (s_+,s_-) were defined as given in the Eq.3.19, and Eq. 3.20, and net power was calculated in term of scattering matrix and scattering variable as described in the Eq. 3.22.

$$s_+ = \left(\frac{f+v}{2}\right) \tag{3.19}$$

$$s_- = \left(\frac{f-v}{2}\right) \tag{3.20}$$

$$\delta P = {s_+}^T s_+ - {s_-}^T s_- \tag{3.21}$$

$$\delta P = {s_+}^T (I - S^T S) s_+ \tag{3.22}$$

S is scattering matrix in the Eq. 3.22. The power difference should be positive for the passivity as the scattered wave should have less power than the incident wave. Therefore, to satisfy this condition the maximum singular value of the scattering matrix should be less than or equal to unity as given in the Eq. 3.23.

$$\| S \|_\infty = \bar{\sigma}\Big(S^T(jw)S(jw)\Big) \leq 1 \tag{3.23}$$

The scattering approach was also applied to a linear system which was transmitting signals with a constant time delay T_c, and control laws were derived for it as given

in the following equations[47, 48].

$$f_m(t) = f_s(t - T_c) \tag{3.24}$$

$$\dot{x}_{sd}(t) = \dot{x}_m(t - T_c) \tag{3.25}$$

$$f_s(t) = K_s \int (\dot{x}_{sd}(t) - \dot{x}_s(t))dt + B_s(\dot{x}_{sd}(t) - \dot{x}_s(t)) \tag{3.26}$$

f(s) is coordinating torque. The Hybrid Matrix of the above signals has delay elements due to which the norms of the scattering matrix comes out to be infinity ($\| S(j\omega) \|_\infty = \infty$), which is clearly an active behavior. This issue was resolved by passivating the communication channel. The control laws were modified as follows.

$$f_m(t) = f_s(t - T_c) + Z(\dot{x}_m(t) - \dot{x}_{sd}(t - T_c)) \tag{3.27}$$

$$\dot{x}_{sd}(t) = \dot{x}_m(t - T_c) + Z^{-1}(f_m(t - T_c) - f_s(t)) \tag{3.28}$$

Z was impedance and it was a ratio of transformer placed between the forces and the velocities to make their values comparable. The scattering matrix for the control scheme is given in the Eq. 3.29.

$$S(s) = \begin{bmatrix} 0 & e^{-sT_c} \\ e^{-sT_c} & 0 \end{bmatrix} \tag{3.29}$$

The scattering matrix has norm $\| S \|_\infty = 1$, hence the system is passive and stable. An extension of the scattering approach was power scaling presented by Colgate[49]. It was observed that for some applications like telesurgery the power transmitted from one side to the other should be scaled to enable the human to deal with the mismatch with the environment. The human operator and the environment were taken passive while applying some scaling coefficients with the scattering matrices S_h and S_{en} respectively. The compact matrix of the human operator and environment scattering matrices was $S_{hen}(s)$ as given in the Eq. 3.30.

$$S_{hen}(s) = \begin{bmatrix} S_h & 0 \\ 0 & S_{en} \end{bmatrix} \tag{3.30}$$

The scattering matrix containing dynamics of the master and the slave along with power and impedance was taken as written in the Eq. 3.31.

$$S_{ms}(s) = \begin{bmatrix} s_{11} & s_{12} \\ s_{21} & s_{22} \end{bmatrix} \tag{3.31}$$

The passivity was determined by calculating the value of the scattering matrix. A similar approach of scaling in the time domain has been described in[50]. The interaction with the environment was scaled by introducing scaling factors as given in the Eq. 3.32, and Eq. 3.33.

$$v_s = k_v v_e \tag{3.32}$$

$$f_e = k_f f'_e \tag{3.33}$$

k_v and k_f were scaling factors of power applied and received from the environment respectively. These factors did not change the sign of the power($f_e^T v_e = k_v k_f (f'_e) v_e$) and therefore, passivity was preserved.

3.2.3 Wave Variables

In 1991 the wave variables concept was introduced in teleoperation by Niemeyer and Slotine[51]. The power variables like velocity v_m and force f_s were replaced by the wave variables u_m and u_s as shown in the Fig. 3.4.

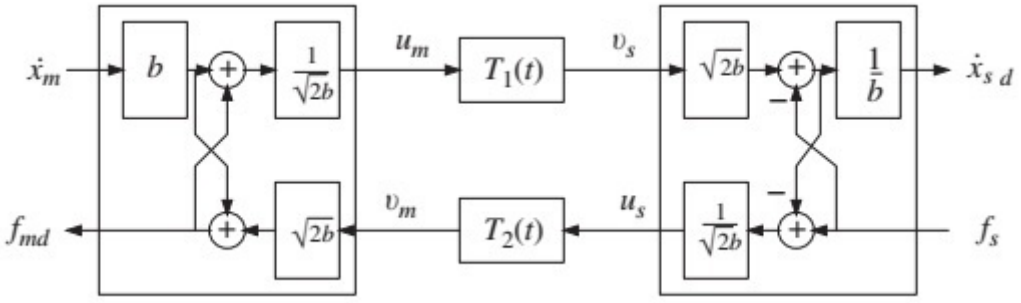

FIGURE 3.4: Wave variables

$$u_m(t) = \frac{1}{\sqrt{2b}}(f_{md}(t) + b\dot{x}_m(t)) \tag{3.34}$$

$$u_s(t) = \frac{1}{\sqrt{2b}}(f_s(t) - b\dot{x}_{sd}(t)) \tag{3.35}$$

f_{md} in the Eq. 3.34, is the received power variable on the master side and $\dot{x}_{sd}$ in the Eq. 3.35, is the power variable obtained on the slave side. b is the characteristic impedance of the transmission line. The corresponding total power in terms of wave variables can be written as given in the Eq. 3.37.

$$P(t) = f_{md}^T(t)\dot{x}_m(t) - f_s^T(t)\dot{x}_{sd}(t) \tag{3.36}$$

$$P(t) = \frac{1}{2}(u_m^T(t)u_m(t) - v_m^T(t)v_m(t)) + \frac{1}{2}((u_s^T(t)u_s(t) - v_s^T(t)v_s(t)) \tag{3.37}$$

v_m and v_s in the Eq. 3.37, are output signals on the both side of a communication channel and corresponding force and velocity signal can be written as:

$$f_{md}(t) = b\dot{x}_m(t) + \sqrt{2b}v_m \tag{3.38}$$

$$\dot{x}_{sd} = \frac{1}{b}(\sqrt{2b}v_s(t) - f_s(t)) \tag{3.39}$$

$$E(t) = \frac{1}{2}\left(\int_{t-T}^{t}(u_m^T(\tau)u_m(\tau) + u_s^T(\tau)u_s(\tau)d\tau\right) \geq 0 \tag{3.40}$$

The passivity of the wave variables is guaranteed under a constant time delay T if the condition in the Eq. 3.40, hold true. Due to the essential passivity of the wave formulation, several control strategies have been developed in the wave domain that otherwise would have caused the loss of the passivity when executed directly in the power variables domain. For example introducing a passive filter like Smith predictor in the wave variables does not affect the passivity of a communication channel[52, 53]. Let consider a predictor on the master side of the communication channel given as in the Eq. 3.41.

$$G_p(s) = (1 - e^{s(T_1+T_2)})\hat{G}_s(s) \tag{3.41}$$

T_1 and T_2 are the constant time delays in the forward and backward path and $\hat{G}_s(s)$ is the estimated slave model. A Smith predictor helps to reduce the effect of a constant time delay and provides a predicted wave variable $v_p = G_p(s)u_m$ which is added to the received delayed wave variable u_s to correct the value of v_m at the master side. In order to retain the passivity with the predicted wave the expression in the Eq. 3.42, should hold true. A similar application of the Smith

predictor has been discussed in[54, 55].

$$\int_0^t u_s^T(\tau)u_s(\tau)d\tau \geq \int_0^t v_m^T(\tau)v_m(\tau)d\tau \tag{3.42}$$

3.2.3.1 Matching Impedances

In the background of transmission line theory, it is well-established fact that if the load terminating the line has a different impedance than the transmission line characteristic impedance then wave reflection happens. In the context of bilateral teleoperation, such reflections downgrade and weaken the performance of the system. This gave rise to the inclusion of impedance matching elements b at each side of the communication network of Fig. 3.4, to get Fig. 3.5. This setting of the impedance matching as described by Niemeyer and Slotine was adapted when the slave was under impedance or force control. The offset between master and slave can be written as given in Eq. 3.45. The background theory and mathematics of the equation has been explained in[56, 57].

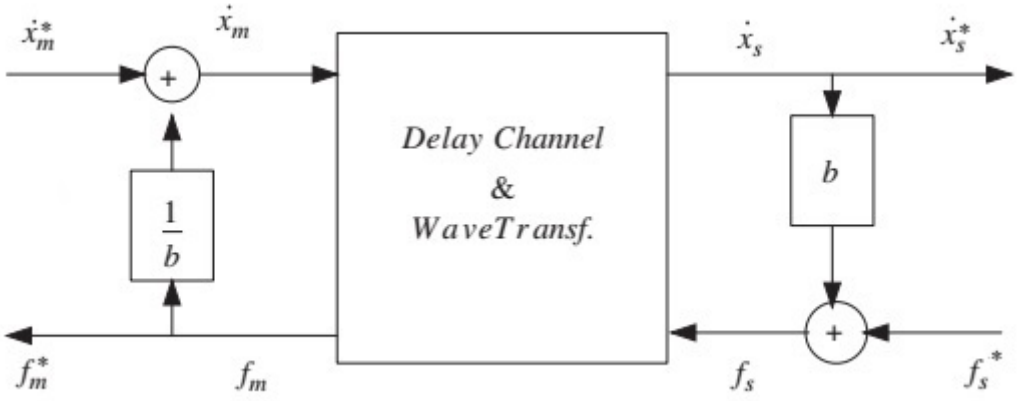

FIGURE 3.5: Impedance matching

$$\dot{x}_m = x_m^* - \frac{1}{b}f_m \tag{3.43}$$

$$f_s = f_s^* + b\dot{x}_s \tag{3.44}$$

$$x_m^*(t-T) - x_s(t) = \frac{1}{b}\int_0^t f_s^*(\tau)d\tau + x_s(t) \tag{3.45}$$

Impedance matching element b affected the position tracking when it was placed on the both sides of the communication block. Therefore, b was only placed on the slave side and the shunt element on master side of the Fig. 3.5, was removed which

resulted in a small position drift as given in the Eq. 3.46. The small position drift was achieved due to smaller gain, shorter integration time and without the effect of x_s.

$$x_m^*(t-T) - x_s(t) = \frac{1}{2b}\int_{t-2T}^{t} f_s^*(\tau)d\tau \tag{3.46}$$

It has been inferred that larger value of b resulted in a smaller position drift but it required extra damping which affected the performance adversely. The difference in $\dot{x}_s$ is $\dot{x}_s^*$ due to wave reflection which occurs at junctions and terminations due change in impedance of wave carrier. Similarly f_m is different from f_m^* due to wave reflection.

3.3 Teleoperation over the Internet

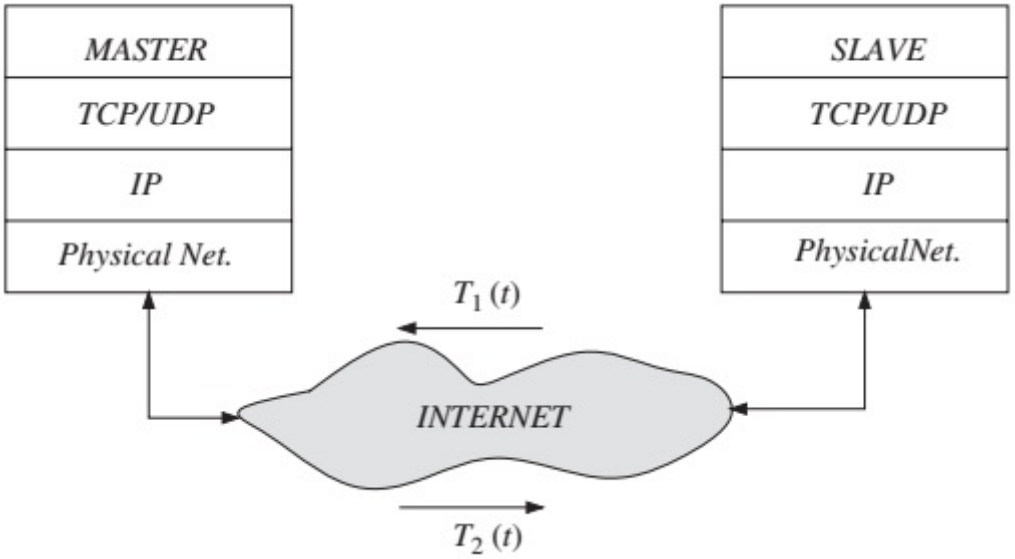

FIGURE 3.6: Teleoperation via the Internet

With the advent of the Internet, the researchers started utilizing it during the 1990s in teleoperation and it is an active field of research till to date[58–64]. The exchange of information across a packet-switched network resulted in random delays in the communication. This delay grew bigger and eventually resulted in packet loss. Additionally, the scenario of discrete time stability also emerged. These issues caused the deteriorated performance of teleoperation with the poor stability.

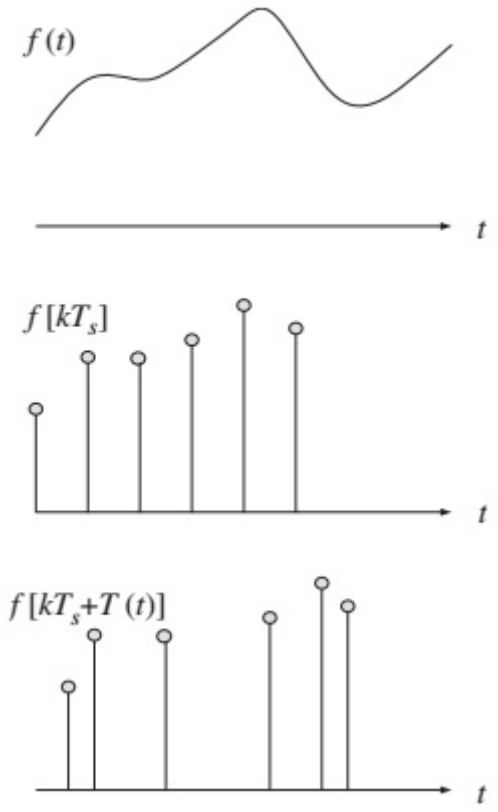

FIGURE 3.7: Distortion of signals due to delays in the channel

In the Internet-based teleoperation, master and slave have to move their discrete data through different layers to physical layers as shown in the Fig. 3.6. There are two protocols for the data exchange like transmission control protocol (TCP) or user datagram protocol (UDP). Each has to be selected based on its performance and application. Both reside at the transport layer. TCP provides reliable bilateral communication with a guarantee of data transmission at the cost of resending of data with long timeouts. UDP does not have reception acknowledgments at the expense of permanent packet loss, eliminating long waiting time, which makes it suitable for real-time applications. Data traveling through the Internet with the forward and backward delay along with distortion in it has been depicted in the Fig. 3.7.

Control laws have been revised over the years under the newly emerging communication medium like the Internet because some existing issues such as position drift between the master and slave has been worsened by the random time delay. Moreover, performance has been drastically affected and stability has been compromised due to information loss and consequently, revision of methods according to new scenario became necessary.

Naturally, having some knowledge about the delay in the communication over the Internet, or at least an idea about the upper bound on the delay, is crucial in designing adequate control laws for the stable teleoperation. Delay estimation schemes have been proposed using an autoregressive model and using some linear or nonlinear time series analysis[65, 66]. Different control approaches using the Internet as a communication medium like Passivity under time-varying delays [67, 68], position drift[69–74], continuous to discrete-time systems [75–77], discrete-time scattering approach [78], information loss[79, 80] have been reported in the literature.

3.4 Passive Decomposition

The recent approach of splitting the teleoperation into two subsystems i.e, the shape and locked systems is called passive decomposition[81–87]. The shape system establishes the coordination between the master and slave and the behavior of the overall motion of the system is determined by locked system. Consider a nonlinear system with power scaling factor (ρ) such that $\rho > 0$. The supply rate (s) can be defined for passive operation as given in Eq. 3.47.

$$s(\dot{x}_m, f_h, \dot{x}_s, f_e) = \rho f_h^T \dot{x}_m + f_e^T \dot{x}_s \tag{3.47}$$

The supply rate is sum of power due to f_e and power factor times power due to f_h. The stable teleoperation is achieved if the condition in the Eq. 3.48, is true.

$$\int_0^t s(\dot{x}_m, f_h, \dot{x}_s, f_e) d\tau \geq -c^2 \tag{3.48}$$

3.5 Adaptive Control

An adaptive control scheme has also been applied to teleoperation which deals with parametric uncertainty[88]. A new mixed position-velocity signal has been transmitted. Consider a nonlinear system as given below.

$$M_m(q_m)\ddot{q}_m + C_m(q_m, \dot{q}_m)\dot{q}_m = f_h + f_m \tag{3.49}$$

$$M_s(q_s)\ddot{q}_s + C_s(q_s, \dot{q}_s)\dot{q}_s = f_s - f_e \tag{3.50}$$

Where q is displacement vector, M is inertia matrix and C matrix contain Centrifugal and Coriolis terms. The control laws for the nonlinear system can be written as follows:

$$f_m = -f_{md} - \hat{M}_m(q_m)\gamma\dot{q}_m - \hat{C}_m(\dot{q}_m, q_m)\gamma q_m \tag{3.51}$$

$$f_s = -f_{sd} - \hat{M}_s(q_s)\gamma\dot{q}_s - \hat{C}_s(\dot{q}_s, q_s)\gamma q_s \tag{3.52}$$

$\hat{M}$ and $\hat{C}$ are the estimates of the M and C matrix. γ is constant. The force acting on the master and slave can be written by using feedback law and the fact that nonlinear system is linearly parameterizable[89] i.e $M(q)\ddot{q}+C(q,\dot{q})\dot{q} = Y(q,\dot{q},\ddot{q})\theta$. Y is a matrix of generalized coordinates (q_m, q_s) and their derivatives. θ is a vector of inertia parameters. The Eq. 3.51, and 3.52, can be written as:

$$f_m \equiv -f_{md} - Y_m\hat{\theta}_m \tag{3.53}$$

$$f_s \equiv -f_{sd} - Y_s\hat{\theta}_s \tag{3.54}$$

Substituting the Eq. 3.53, and 3.54, in above Eq. 3.49, and 3.50, respectively, the following equations are obtained.

$$M_m\dot{r}_m + C_m r_m = f_h - f_{md} + Y_m\tilde{\theta}_m \tag{3.55}$$

$$M_s\dot{r}_s + C_s r_s = +f_{sd} - f_e + Y_s\tilde{\theta}_s \tag{3.56}$$

A new variable r has been used which comprises of position and velocity as given in Eq. 3.57, and 3.58. The control laws for master and slave can be written as given in Eq. 3.51, and 3.52.

$$r_m \equiv \gamma q_m + \dot{q}_m \tag{3.57}$$

$$r_s \equiv \gamma q_s + \dot{q}_s \tag{3.58}$$

$$\tilde{\theta}_m = \theta_m - \hat{\theta}_m \tag{3.59}$$

$$\tilde{\theta}_s = \theta_s - \hat{\theta}_s \tag{3.60}$$

The Eq. 3.59, and Eq. 3.60, are the estimation errors. This scheme makes the estimation errors bounded and converge them to zero asymptotically.

3.6 Sliding-mode Control

The sliding-mode control approach for teleoperation of one degree of freedom (DOF) system without delay was presented by Buttolo in 1994[90]. It was later extended to teleoperation with time delayed by Cho and Park[91–93]. The main idea presented in their work is to consider a sliding surface, which consists of the error between the positions and velocities of the master and slave.

$$s = \dot{\tilde{x}} + \lambda \tilde{x} \tag{3.61}$$

where $\tilde{x}$ in the Eq. 3.61, is equivalent to the position of master and slave relation:

$$\tilde{x} \equiv x_s - k_s x_m \tag{3.62}$$

The k_s is a position scaling factor in the Eq. 3.62. Standard methods to derive stable control laws for surface s are dependent on the human and environment forces. Large gains are needed when there is a huge difference between estimated and actual forces to enforce the dissipation condition ($s\dot{s} \leq -\eta \parallel s \parallel < 0$).

3.7 H_∞ Control

Leung presented the H_∞ and μ-synthesis approaches to derive compensators for delayed teleoperation in 1995[94]. The approach is based on an approximation of the maximum allowable time delay between the master and slave in the forward and feedback communication. A linear system with master model $P_m(s)$ and slave model $P_s(s)$ was considered in the frequency domain. The control involved the design of compensators for free motion using H_∞ control approach and then it was extended to delayed constrained motion using $\mu - synthesis$. The controller parameters for the master controller C_m for free motion were driven as follows:

$$z = \begin{pmatrix} W_{m1}(f_h - \dot{x}_m) \\ W_{m2} f_{m1} \end{pmatrix} \tag{3.63}$$

$$w = \begin{pmatrix} f_h \\ d_{m1} \end{pmatrix} \tag{3.64}$$

$$y = \dot{x}_m + W_{m3} d_{m1} \tag{3.65}$$

$$u = f_{m1} \tag{3.66}$$

The controller parameters for the slave controller C_s for free motion were driven as follows.

$$z = \begin{pmatrix} W_{s1}(\dot{x}_m - \dot{x}_s) \\ W_{s2} f_{s1} \end{pmatrix} \tag{3.67}$$

$$w = \begin{pmatrix} f_h \\ d_{s1} \\ d_{s2} \end{pmatrix} \tag{3.68}$$

$$y = \begin{pmatrix} \dot{x}_m + W_{s3} d_{s1} \\ \dot{x}_s + W_{s4} d_{s2} \end{pmatrix} \tag{3.69}$$

$$u = f_{s1} \tag{3.70}$$

z is the performance output, w is an exogenous signal, y is the measured output, u is the control input, in both cases. The disturbance and weighting matrices are d and W respectively. First, the controller was designed for free motion and then it was extended to the case of constrained motion with a delay while the slave was interacting with the environment having impedance Z_e. An extra controller K to account for the delays was introduced. The time delay was taken as a disturbance to constrained motion. The total time delay T was considered which was combined forward and backward delay. The $\mu-$synthesis design was used to calculate the controller K for the system as shown in the Fig. 3.8. The $\mu-$synthesis produces a robust and stable controller for closed loop system by properly choosing weighting functions. Mathematical description of $\mu-$synthesis is given in[95]. G is plant in Fig. 3.8, which contains dynamics of both master and slave. w is noise signal acting on the plant. u is control input. $\triangle$ is perturbation due to time delay. The Eq. 3.71, is performance output for the constrained motion of the system (i.e. slave interacting with environment along with time delay while) the Eq. 3.67, shows the performance output of the slave when there is no delay and also there is no constrains like the environment force acting on slave is zero.

$$z = \begin{pmatrix} W_1(\dot{x}_m - \dot{x}_s) \\ W_2(f_s - (f_{m1} + f_{m2})) \\ W_3 f_{m2} \\ W_4 f_{s2} \\ z_5 \end{pmatrix} \tag{3.71}$$

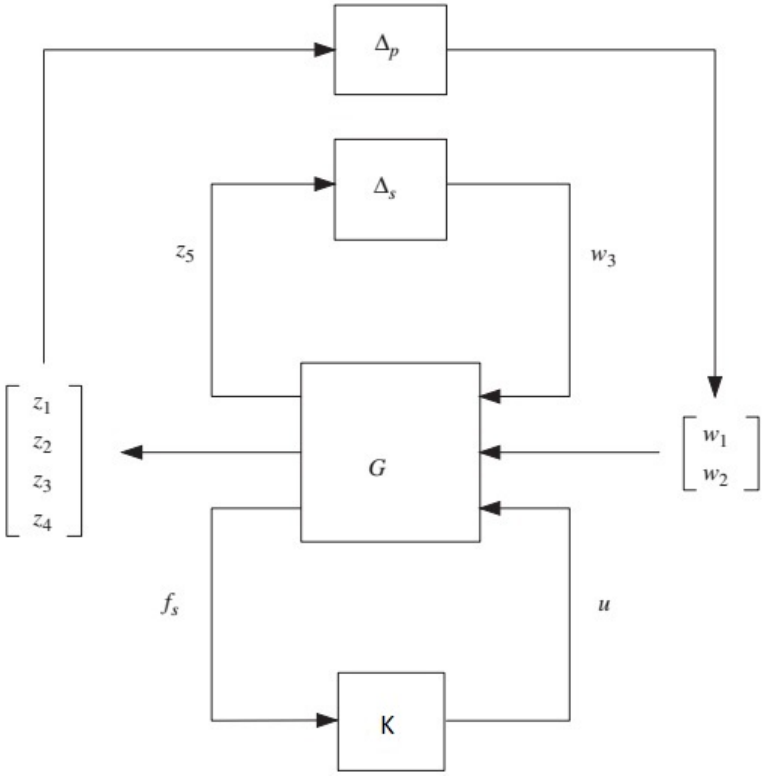

FIGURE 3.8: $\mu - synthesis$

$$w = \begin{pmatrix} f_h \\ f_b \\ w_3 \end{pmatrix} \tag{3.72}$$

$$y = f_s \tag{3.73}$$

$$u = \begin{pmatrix} f_{m2} \\ f_{s2} \end{pmatrix} \tag{3.74}$$

The designed controller worked fine for all delay values that were within allowable range, but it was too conservative. In order to cope with the issue of conservation, a solution was presented later by Sano[96]. Gain scheduling was proposed which had the provision of designing a controller for multiple values of delay. In another development, the scaling elements were introduced in the control loop and the design was performed under a specified range of scaling elements[97]. 4-channel formulation has been utilized to design controller based on H_∞ as well[98, 99].

Chapter 4

Event Based Telecontrol

4.1 Introduction to Event Based Control

The event based sampling and control has gained much attention in recent years because it allows a straightforward implementation and better resource utilization. Numerous event based control approaches and algorithms have been proposed. Event based sampling and control has potentially solved many issues related to network-controlled robots. Instead of periodic sampling, the events are triggered when they are needed, e.g. on reaching some threshold point, so that the number of generated events like control actions or network messages are less. Consequently, resulting in a decrease in the network traffic and the associated problems like delay in the commanded actions and dropouts become less critical. Thus, the battery in wireless devices who are using event based sampling is utilized more efficiently because of less communication events load and this elongates the battery lifetime. Additionally, malfunctioning of active control elements like valves is also retarded due to less number of control actions. The event based control has smaller latencies as compared to the periodic control because it does not need to wait until the next periodic activation.

Potential applications of event based control are embedded systems, building automation and wireless sensor networks. However, the design and implementation of the event based control are still challenging field. Hence, special control algorithms have been developed over the years in the recent times that have exhibited the usefulness and applicability of event based control. The event based sampling

and control can be implemented at different elements of the control loop which increases the number of possible exertion variants[100–102].

4.2 Teleoperation with Event Based Control

In recent years, teleoperation through the Internet has gained substantial attention from researchers across the globe because Internet is easily available and it has low cost. Major problems in Internet-based teleoperation are the random time delay, packet loss, and disconnection, etc. These issues make it difficult to develop stable teleoperation. The prime objective in teleoperation is to develop a stable and transparent system. Traditionally, control signals and trajectories are referred to time. That means, conventional control systems have time as their action reference. These kinds of systems suffer from different issues due to Internet-based teleoperation like instability and de-synchronization caused by a random time delay. But, if time is replaced by a non-time based action reference, then the time-varying delay has a very restricted effect on performance. Therefore, event based planning and control which is a non-time based planning and control methodology has been utilized to suppress the deadly adverse effects of time-varying delay.

The Fig. 4.1, shows the conventional control block with time as a reference and the Fig. 4.2, shows necessary amendments in a conventional control loop to elaborate the event based control.

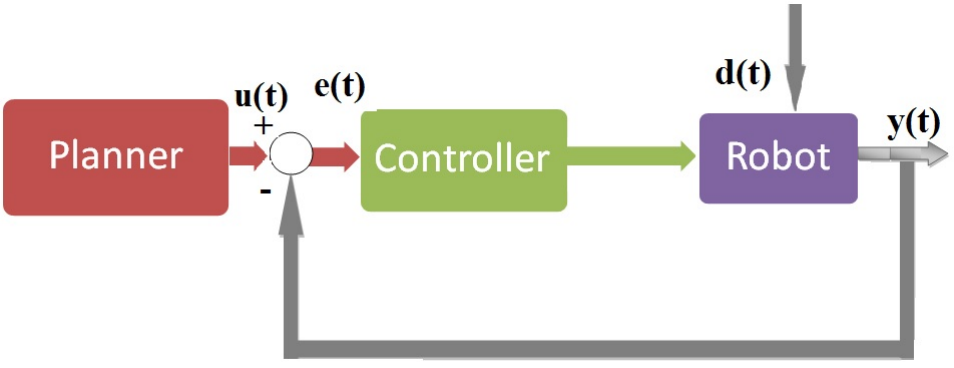

FIGURE 4.1: Conventional control loop

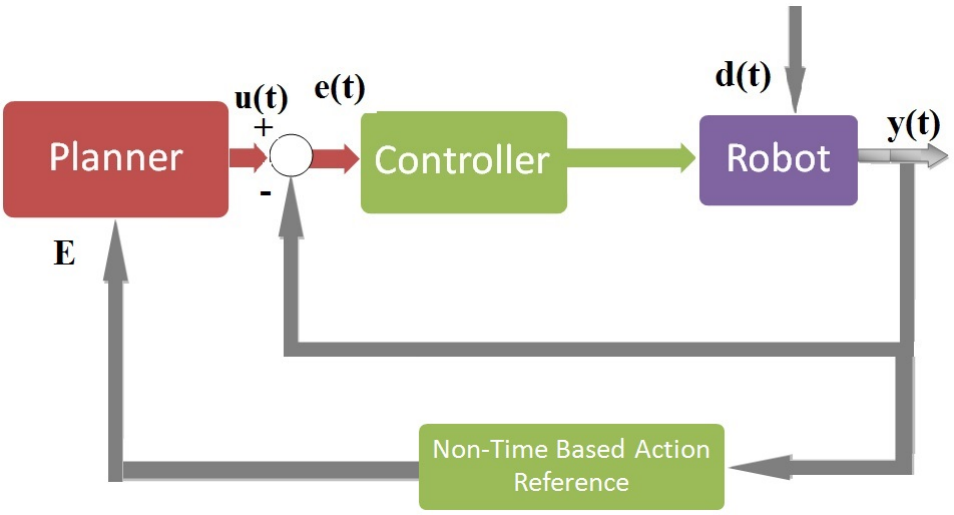

FIGURE 4.2: Non-time based control

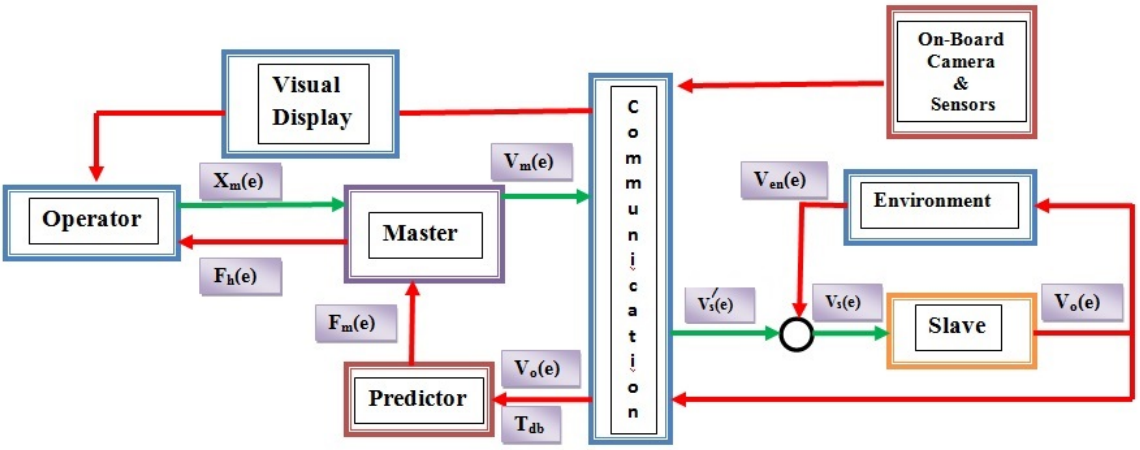

FIGURE 4.3: Block diagram of the teleoperated system

4.3 System Implementation Model

The telecontrol of a mobile robot has been implemented as shown in the Fig. 4.3. It has two parts separated by a communication channel. The human operator resides on the left of the communication channel and all the control commands are generated by him. A haptic force feedback joystick which is very crucial in telecontrol is maneuvered by the human operator to set different commands. He can also visualize the vision and sensory feedback to perceive the environment around the robot. Additionally, s/he can compensate the certain instabilities in telecontrol. The force acting on the slave robot is fed to master device so that operator can feel the real impact of force acting on slave robot. In telecontrol,

there is a delay due to which force acting on the haptic device is a delayed response. The force is modeled as a virtual force which is acting on the robot and is inversely proportional to distance to an obstacle in front of the robot. With the predictor block, it has been ensured that the real-time force is generated by using the T_{db}(delay time backward), the velocity of slave and on board proximity sensors to calculate the real time position and hence virtual force.

The human operator applies the force F_h as given in the Eq. 4.1, to master device to generate the new position X_m as described in the Eq. 4.2.

$$F_h(e) = \begin{pmatrix} F_{hx}(e) \\ F_{hy}(e) \\ F_{h\phi}(e) \end{pmatrix} \tag{4.1}$$

$$X_m(e) = \frac{F_h(e)}{C_h} \tag{4.2}$$

Where C_h is constant and e is the event

$$X_m(e) = \begin{pmatrix} X_{mx}(e) \\ X_{my}(e) \\ X_{m\phi}(e) \end{pmatrix} \tag{4.3}$$

F_h is a resultant of force feedback and applied force by the human operator. The force is fed back only in x and y directions. $F_{h\phi}$ is for the rotation of the master device. The velocity of master device V_m corresponding to a position change of the joystick can be written as given in the Eq. 4.4.

$$V_m(e) = C_m X_m(e) \tag{4.4}$$

C_m is scaling constant from position to velocity.

$$V_m(e) = \begin{pmatrix} V_{mx}(e) \\ V_{my}(e) \\ V_{m\phi}(e) \end{pmatrix} \tag{4.5}$$

V_m travels through a communication channel to slave robot. The communication channel is a simple Internet with inherent delay characteristics in it. It helps in transmitting and receiving data. Contrary to conventional control with time as a

reference, the event based control is not affected by advancement in time but only with the generation of next event. In the case of connection loss between the master and slave, the slave would stop and wait for a resumption in communication. During disconnection, the slave remains stable and resumes its action upon receiving next event. The understanding of the environment around the slave robot is crucial for a human operator to perform teleoperation. The contact with obstacles is avoided and the force is calculated as a function of distance. $V_s'(e)$ is received velocity on the slave side from the communication link and it is fed to slave robot with slight modification as $V_s(e)$ as described in the Eq. 4.6.

$$V_s(e) = V_s'(e) - V_{en}(e) \tag{4.6}$$

$V_s(e)$ is the input velocity of the robot, $V_{en}(e)$ is the effect of the environment and $V_s'(e)$ is the input velocity without the environmental effect coming out of a communication channel. $V_\circ(e)$ is the output velocity of the slave robot as given in the Eq. 4.7.

$$V_\circ(e) = \begin{pmatrix} V_{\circ x}(e) \\ V_{\circ y}(e) \\ V_{\circ \phi}(e) \end{pmatrix} \tag{4.7}$$

The virtual environmental force is calculated based on this velocity and the distance to obstacles. Then, that force is sent back as feedback from the robot location to a human operator.

4.4 Stability Analysis

Theorem: If the original robot dynamic system (without remote human operator/autonomous controller) is asymptotically stable with time t as its action reference and the new non-time action reference $e = \prod(y)$ is a (monotone increasing) non-decreasing function of time t, then the system is asymptotically stable with respect to new action reference e[103].

In order to ensure stable teleoperation, the statement in above theorem should hold true, i.e. the robotic system should be stable with time t as its action before switching to non-time action reference. The non-time action reference should be increasing with advancing time. This leads to asymptotically stable system with new action reference. The main benefit of this technique is that stability

is independent of the human model and time delay. The challenges involved in the selection of event e were the presence of uncertainties like unstructured environment and trajectory of the robot, bad Internet connection, etc. Normally, the event e is selected based on certain position or distance from a specific location or angle with origin but these options cannot be exercised in teleoperation. Therefore, after careful inspection and keeping in mind the sensitive nature of the system, it was decided that the execution time should be fixed and it should be taken as an event. Hence, each event duration is 200ms and after executing every input 200ms system waits for new input as next event. If the next event is not available due to longer delay then robot would stop and wait for next event.

4.5 Experimental Results

Some experiments were carried out to evaluate the behavior of the controller under different circumstances, like when there was a perfect connection between client and server. Later, the performance was evaluated with disconnection and reconnection. These figures have been plotted with events on the horizontal axis and other parameters like the linear and angular velocity and force against the vertical axis. The duration of each event is 200ms.

Fig. 4.4, shows the heading angle values on the vertical axis and events on the horizontal axis. This plot has the angular velocity values of the master device like the steering wheel of a car. The slave heading angle has been plotted in the Fig. 4.5. These two figures have been drawn while there was a perfect connection between master and slave robot. It is obvious from these plots that there is a complete harmony and coordination between the master and slave heading angle values over the span of nearly 300 events. At some events the slave heading angle is slightly different than the master because the slave robot is moving on the ground and it needs certain time to turnaround because on spot turning is not possible due to friction between the robot wheels and the ground. The force provided by the servo motor is insufficient for turning in static position. Also due to data communication and its processing in microcontroller to calculate particular control action for servo motor, there is slight non-synchronization at certain events.

The Fig. 4.6, shows the linear velocity of the master device on the vertical axis and events on the horizontal axis. These values were generated by moving the

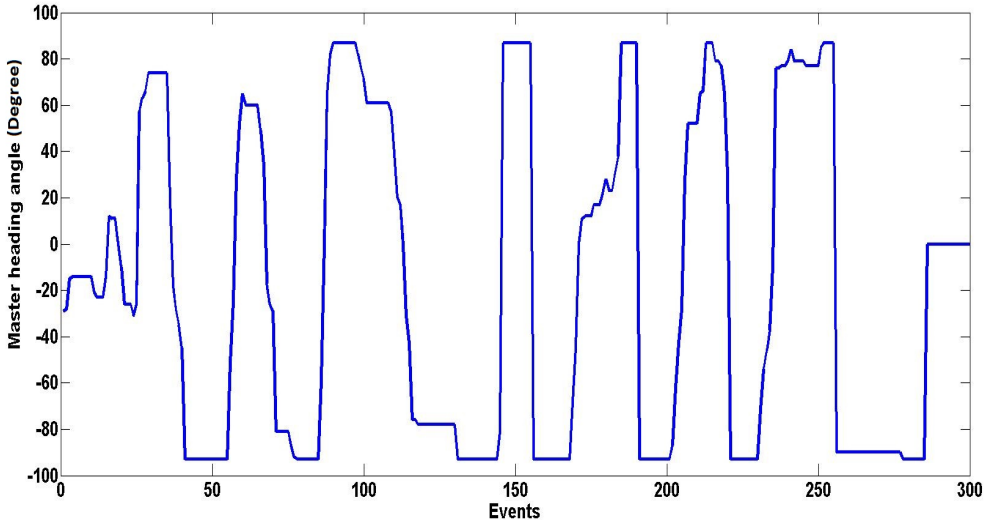

FIGURE 4.4: The master angular velocity

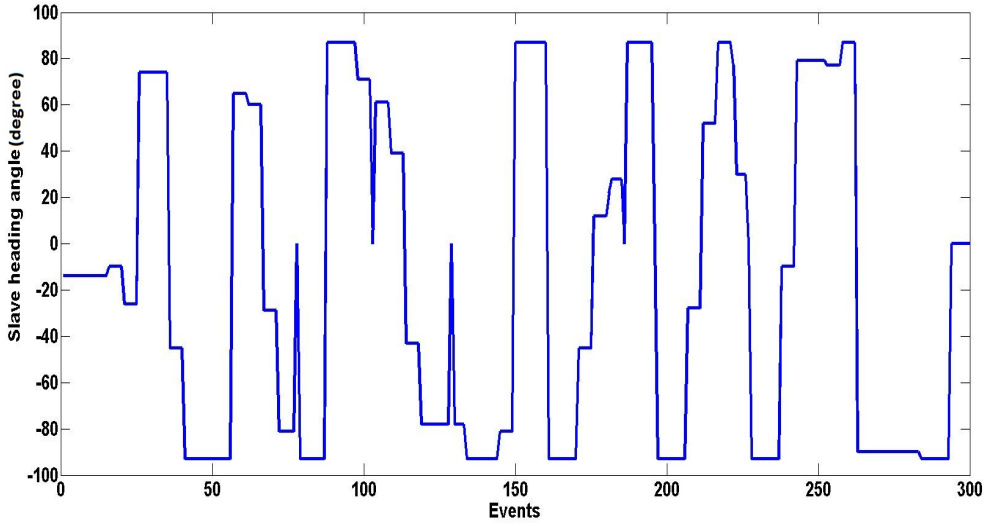

FIGURE 4.5: The slave heading angle

master device back and forth to set the linear velocity commands for the slave robot like a gas pedal in a car. The slave linear velocity has been plotted in the Fig. 4.7. The slave robot is following the master robot's linear velocity. There is a smooth motion in forward and reverse direction according to the aspirations of the human operator.

In the second scenario, the controller was tested with a bad communication link i.e, which has connection loss at different intervals. The Fig. 4.8, exhibits the master linear velocity with some interruption in communication between master and slave. The Fig. 4.9, shows slave linear velocity with disconnection. In the beginning the

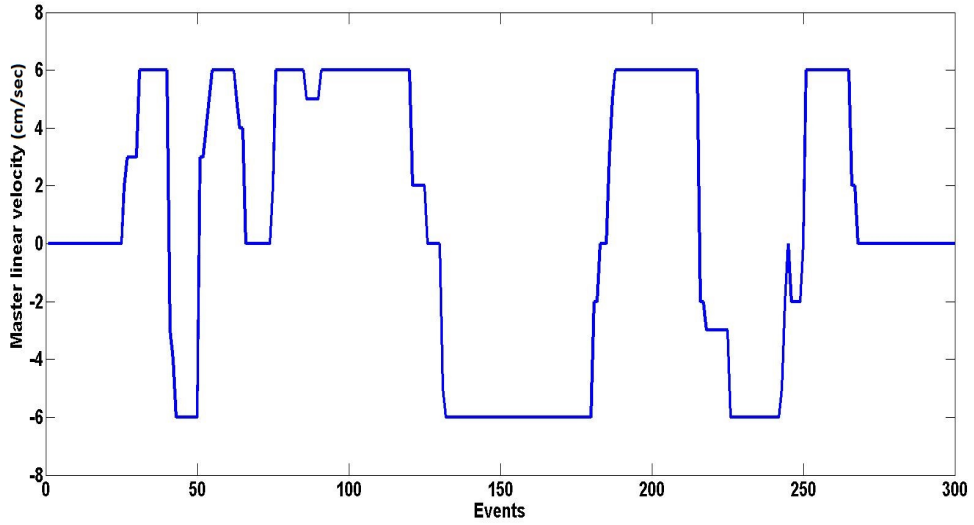

FIGURE 4.6: The master linear velocity

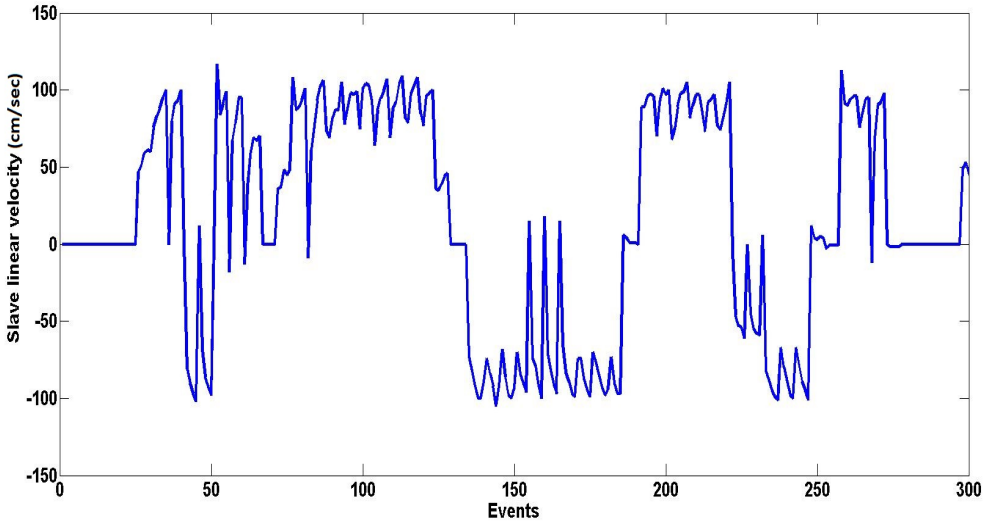

FIGURE 4.7: The slave linear velocity

master linear velocity is zero and the corresponding slave linear velocity is also zero. After 40 events there is variation in the master velocity until 150 events and the slave follows the master velocity. Then, there is a connection loss between the master and slave robot and the slave robot has zero velocity which is critical for the stability so that the slave does not collide with the surrounding objects. Hence, the robot is safe and does not navigate in unstructured environment which is the prime objective i.e it should always listen to master commands. After some events at 230 approximately, the connection is re-established and the slave follows the master again. The master velocity is shown zero when there is connection loss

because the algorithm developed sends data through TCP/IP and it only records data when there was established communication between master and slave.

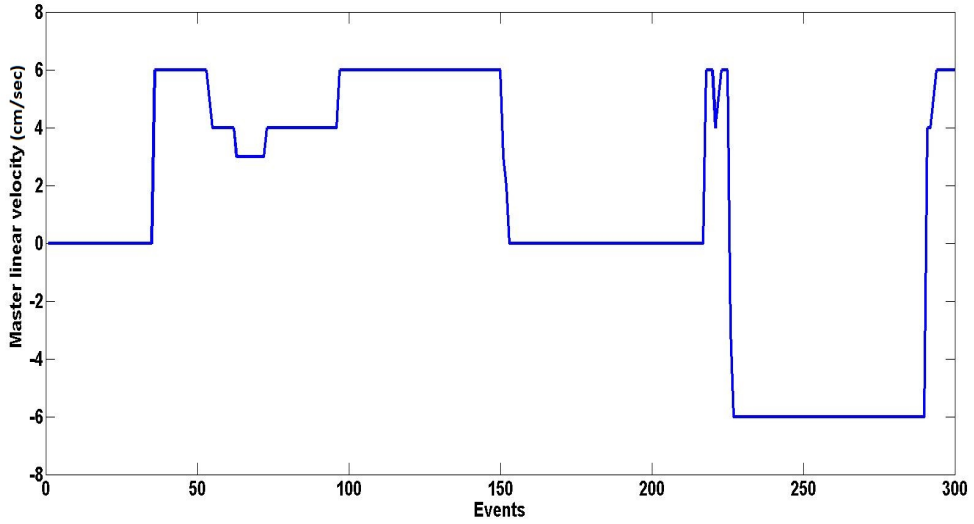

FIGURE 4.8: The master linear velocity with disconnection

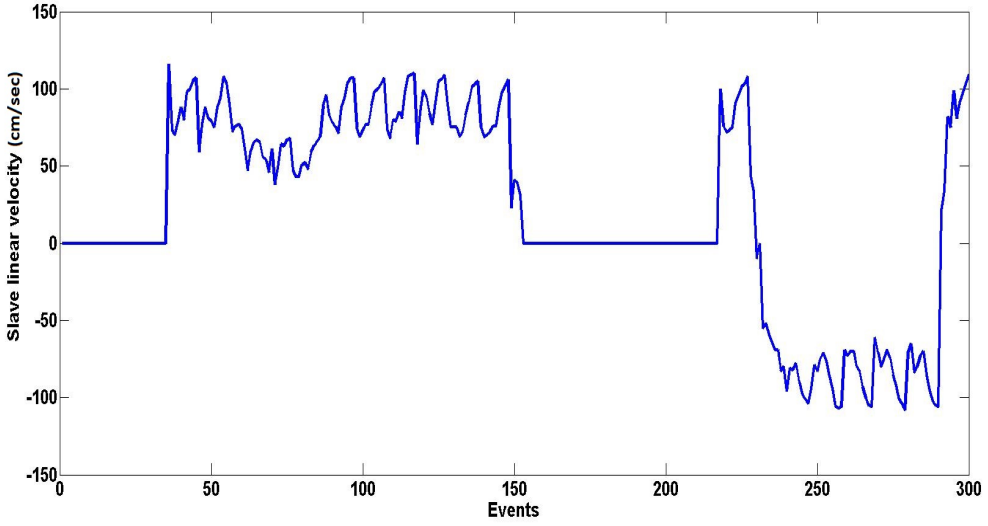

FIGURE 4.9: The slave linear velocity with disconnection

The above two experiments were performed without the obstacles around the mobile robot during its navigation. The real world has an unstructured environment with different objects scattered in a random fashion. Therefore, it was necessary to evaluate the performance of the robot in such a clustered and cluttered environment which should be similar to the real world. The Fig. 4.10, shows the linear velocity of the master haptic device. The Fig. 4.11, exhibits the corresponding

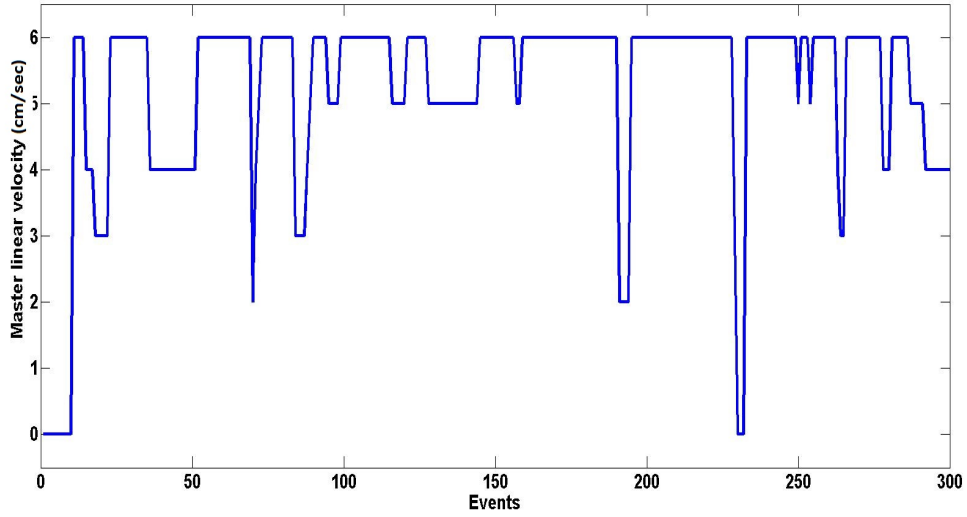

FIGURE 4.10: The master linear velocity when there are obstacles

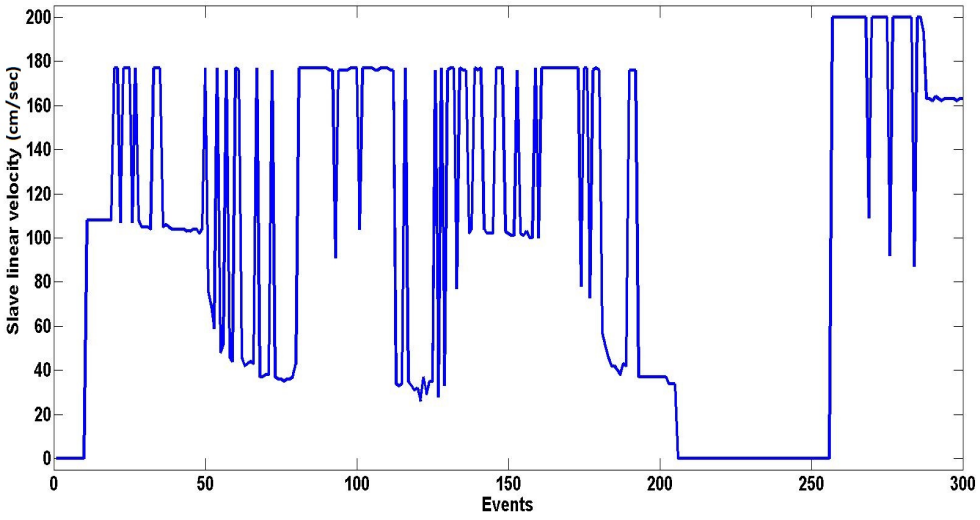

FIGURE 4.11: The slave linear velocity when there are obstacles

slave linear velocity. The last plot shown in the Fig. 4.12, presents the force acting on the slave robot which has been brought back to master device as force feedback. These plots are strongly correlated with each other. In the beginning, the slave robot follows the master device commands and navigates according to the human operator instructions. When the obstacle is detected by onboard sensors, then the value of input velocity V_s is adjusted to reduces the speed of the robot so that the human operator can take necessary actions to avoid collision with it i.e. maneuver the robot. The Fig. 4.12, shows the small variation in force until 50 events. After that, there is an increase in the force due to the presence of some

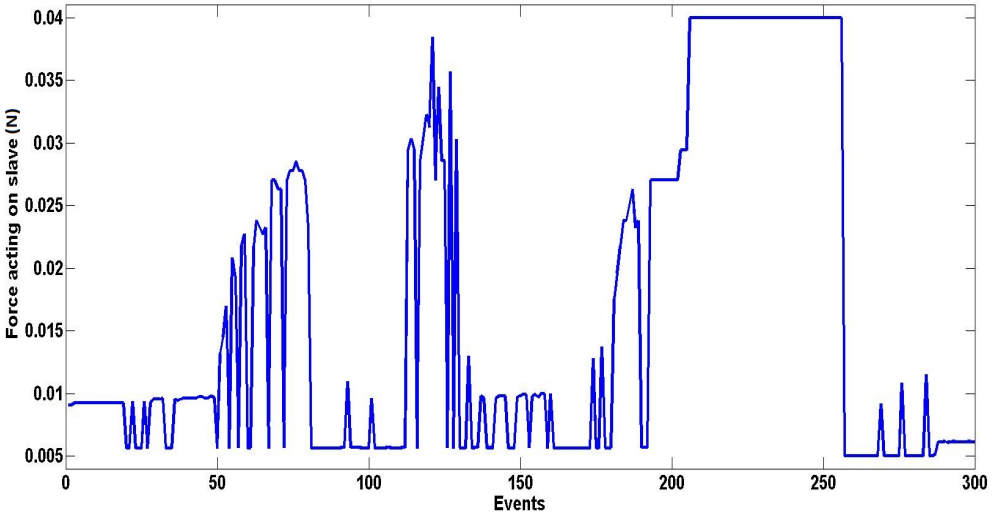

FIGURE 4.12: The force acting on the slave robot which is reflected back to the master device

object in the vicinity of the robot. This rise in force is due to an obstacle which resulted in decrease in the slave robot navigation speed even though the human operator has sometimes sent the maximum speed values. This feature is really important because it helped to avoid the obstacles with ease by providing more time to a human operator for making intelligent decisions. So when there is maximum force the velocity of slave is minimum regardless the velocity of master device and when the object is in the critical range of 0.2m the slave stops and force becomes maximum and slave does not listen to master device for forward motion.

Chapter 5

Telecontrol with Fuzzy Logic

5.1 Introduction

Fuzzy Logic is a form of numerous value logic in which the truth value of a variable varies and can have any real number value between 0 and 1. As compared to Boolean Logic, which has truth value only 0 or 1, fuzzy logic has the concept of partial truth. The true value in fuzzy logic has always ranged between totally true to totally false. This means everything is true up to a certain degree in the Fuzzy Logic as opposed to the notion of things being either true or false in the classical Boolean logic. In order to model a process, or in general the world, and make decisions about it, Fuzzy Logic uses the fuzzy sets and the fuzzy rules. Fuzzy Logic is a tool for the soft computing and it has the ability to cope with the ill-defined and ambiguous problems. Fuzzy set based approximate reasoning was first introduced by the Prof. Lotfi A. Zadeh in Berkeley in 1965, forming the basis of the Fuzzy Logic. The concept has evolved over the years and has helped in the development of many applications[104, 105].

The Fuzzy Logic Control(FLC) uses a mode of approximate reasoning, which enables the system to make the decisions based solely on the vague and incomplete information, which is similar to that of a human being. Two concepts, i.e. the linguistic variables and the fuzzy *If–Then* rule base play a central role in the Fuzzy Logic. The linguistic variable is interpreted as a label of a fuzzy set that is characterized by a membership function. A fuzzy rule is decomposed into the antecedents and the consequents that contain the linguistic variables. However, the total number of fuzzy rules and system parameters increase exponentially

with the number of input variables in the standard fuzzy reasoning process. This imposes a heavy burden on the system because of the control robustness and cost. Therefore, a system with many input variables needs a special fuzzy reasoning system for the robust control and less rules[106].

5.1.1 Fuzzy Set

A fuzzy set is a class of objects with a continuum of grades of membership. Such a set is characterized by a membership (characteristic) function which assigns to each object a grade of membership ranging from 0 to 1. Due to its inherent abilities, a fuzzy set is capable of modeling uncertain situations and can be instrumental in the formalization of interpolative reasoning. The concept of fuzzy set is similar to the widely known classical set, with the difference that the elements of the fuzzy set belong to the set to various degrees from 0 to 1, known as the membership grade (μ). In contrast to a fuzzy set, the classical set has elements belonging to it with a membership grade of either 0 or 1[107].

Given a fuzzy set F, defined on the universe of discourse U, it is possible to define a membership function that assigns to every $\mu \epsilon U$, a value from the unit interval [0, 1].

$$\mu_F : U \rightarrow [0, 1] \tag{5.1}$$

Where μ_F is the membership function of the fuzzy set F. This implies that for every element u from U, there exists a membership degree.

$$\mu_F(u)\epsilon[0, 1] \tag{5.2}$$

Where F is defined by a set of tuples

$$F = [(u, \mu_F(u)) \mid u\epsilon U] \tag{5.3}$$

Fuzzy sets have different characteristic (shape) functions over which the membership functions of their elements are defined, these characteristic functions are as follows.

- Triangular

- Trapezoidal function
- Gaussian function
- Generalized bell function
- Sigmoid function

The major operations in the fuzzy set theory are the intersection, union and complement operations. Fuzzy Logic was proposed to design the control systems by Mamdani in 1977 and later by Sugeno in 1985. Hence, there are two major types of the fuzzy controllers, namely, the Mamdani type and the Takagi-Sugeno type. A brief discussion about the Mamdani is as follows.

5.2 Fuzzy Logic in Control

The application of the Fuzzy Logic in the design of the control system has brought many successful consumer and industrial products. Although, there are the Generalized Modus Ponens and the Compositional Rule of Inference used in the deduction of a conclusion from the fuzzy rules, involving the use of many fuzzy sets and fuzzy relations, there is still no general framework for the design of controllers using Fuzzy Logic. This implies, that the method employed in the every control system design using the fuzzy logic is mostly application specific.

Usually, when the dynamics of a system are nonlinear, complex, ill-defined, difficult to model mathematically, and has large disturbances, then the control design for such system is challenging and teleoperation falls under such category. Linear controllers are not usually robust enough to bring about the desired behavior and performance. Fuzzy Logic, on the other hand, helps in the development of the controller for such systems. The choices of the linguistic variables and their ranges, the membership functions (fuzzy sets) and their spans and a well-structured rule base are the key features of the Fuzzy Logic[108, 109]. The stepwise description of the Fuzzy Logic Control (FLC) is as follows.

- Thoroughly study and understand the system.
- Define the linguistic variables, i.e. the input(s) and output(s) variables for the FLC.

- Define the ranges (Universe of Discourse) for the defined linguistic variables.
- Define the membership functions (MFs) for each linguistic variables and systematically specify a spans for each MF.
- Create a well-organized rule base.
- Define the methods for the conjuncture (AND), disjuncture (OR), implication, and aggregation for the inference engine.
- Choose an appropriate technique to convert the fuzzy output variable (control variable) to a crisp output variable to be sent to the plant.

There are four components of a FLC, a fuzzification module, a rule base, an inference engine and a defuzzification module as shown in the Fig .5.1.

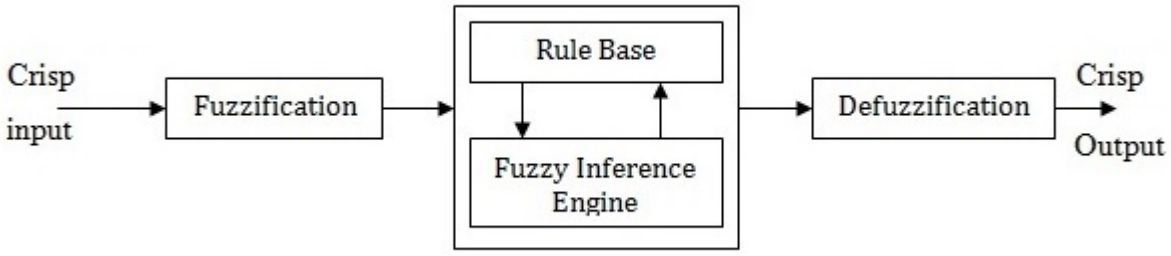

FIGURE 5.1: The components of a Fuzzy Logic Control

5.2.1 Fuzzification Module

Fuzzification is the first step in the fuzzy inferencing process. This involves a domain transformation where the crisp inputs are transformed into the fuzzy inputs. The crisp inputs are exact inputs measured by the sensors and passed into the control system for the processing, such as temperature, pressure, rpm etc. Fuzzification generally involves determining the degree of membership of a crisp value in an appropriate fuzzy set of a linguistic variable, i.e. the degree to which a crisp value belongs to a fuzzy set. Fuzzification can be achieved through the membership functions.

Given a linguistic variable X, with the fuzzy sets A_1, A_2, and A_3 and with the crisp input $x_\circ$, the fuzzified values are as follows and these values are deduced from the Fig .5.2.

$\mu_{A_1}(x_\circ) = 0.0$

$\mu_{A_2}(x_\circ) = 0.1$

$\mu_{A_3}(x_\circ) = 0.8$

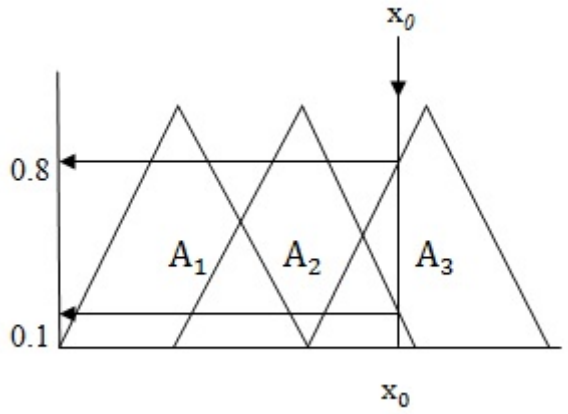

FIGURE 5.2: The fuzzification of a crisp input data

5.2.2 Rule Base

There are several rules contained in a rule base and these rules depend on the number of input linguistic variables as well as the number of fuzzy sets. Some general format of rules in the rule base is as follows:

- If a variable Is a property, Then action
- If (a set of conditions are satisfied) Then (a set of consequences can be inferred)
- If (fuzzy proposition) Then (fuzzy proposition)
- If the Temperature is High, Then Increase the Fan speed

The statements (*The Temperature is High* and *Increase the Fan speed*) are known as the fuzzy propositions. These propositions can be atomic or compound. When

they are compound, then they are usually joined together by some connective operators such as *AND* or *OR*. Therefore, the statements (*The Temperature is High*, and *Increase the Fan speed*) are examples of atomic fuzzy propositions while (*Error is Positive AND Change in error is Negative*) is a compound fuzzy proposition.

5.2.3 Fuzzy Inference Engine

The processes involved in the fuzzy relations, including an implication and aggregation, take place in the inference engine. The inference engine fires the appropriate rules from the rule base. The inference engine deduces the overall fuzzy control output as follows.

- The fuzzified inputs are applied to the antecedents of the rules, the *AND* or *OR* operators are applied to the rules with multiple antecedents in order to obtain the result of the evaluation of the antecedents as a single number.
- The result of the antecedent evaluation is then related to the membership functions in order to obtain an output as a fuzzy set. This process is called implication, and it is carried out for all available rules.
- As a result of implication, each rule produces an output fuzzy set, these fuzzy sets are then combined in such a way to obtain a single fuzzy set. This process of combining the output fuzzy sets is known as aggregation. Aggregation produces one fuzzy set for each output variable.

5.2.4 Defuzzification Module

In many practical processes, a crisp control signal is sent to actuate the plant, therefore the process of defuzzification is necessary. In the defuzzification module, a mapping from a space of fuzzy control actions defined over an output universe of discourse into a space of crisp control actions is carried out. In general, defuzzification involves the conversion of the fuzzy control actions inferred from the fuzzy control system into a crisp control action. There are different methods/strategies available for the defuzzification, the most common ones are:

- Centroid Method (Centre of Gravity Method or Centre of Area Method)

- Mean of Maximum Method
- Max Membership Method (Height Method)
- Weighted Average Method

As there is no specific procedure for the selection of a defuzzification method, the choice of a defuzzification method is basically dependent on the particular design application.

The reason behind the utilization of Fuzzy Logic Control (FLC) for the development of telecontrol of a mobile robot is that the fuzzy system is robust even in case of a vague environment. It also has the additional advantage like the low sensitivity to a variation of parameters or noise levels.

5.3 Telecontrol Setup

The telecontrol setup is shown in the Fig. 5.3. It consists of a joystick which is connected to a laptop. The laptop hosts the algorithm which translates the joystick movement into the input velocity of the robot. It also displays the visual feedback from the robot. This whole setup is regarded as a control station. The human operator stays at the control station and sets different commands for the robot navigation at the remote location. The velocity command travels from the control station to the robot through a communication channel. The communication has been established using TCP/IP protocol. The human operator interacts with a client algorithm at the control station. A micro PC which has the server algorithm in it is mounted on the robot. When the velocity command reaches the micro PC, it sends it to the robot via serial port and receives the visual and sensory feedback from the robot.

Fuzzy logic has been implemented to design a suitable controller, which can work in the ill-structured environments potentially dangerous to the humans, in the presence of random communication delay. The design is based on two inputs. One input comes from the human operator and the other from an algorithm developed and deployed on the robot, which takes the values of onboard proximity sensors and sorts them out to find the shortest reading. This distance value from robot to the obstacle is fed as a second input to the fuzzy controller to give an appropriate

output speed. The implication of the fuzzy logic enables the human to teleoperate the AutoMerlin robot in the presence of a random time delay. The robot listens to a human operator and follows him/her until there is no obstacle in front of him. But if it detects the obstacle, then it calculates the speed based on the objects in front of it and slows down to zero speed if the robot reaches very near to the object. Then the robot can listen to the operator for reverse movement only, but ignores the commands for the advancement. Additionally, the live video feed is made available to the operator so that s/he can perceive the environment accurately.

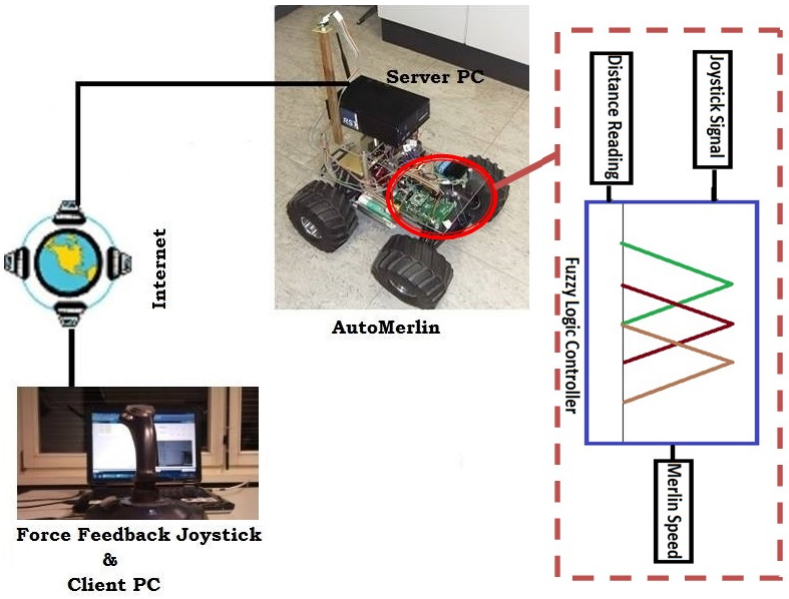

FIGURE 5.3: Telecontrol loop along with the control station and the robot

5.4 Intelligent Fuzzy Set Model For AutoMerlin Telecontrol

The frequently used controller for the nonlinear systems is fuzzy controller. The fuzzy system is able to have a robust control of the robot and it has low sensitivity to a variation of parameters or noise levels. The fuzzy logic is a powerful problem-solving methodology with a myriad of applications in embedded control and information processing. It can provide a remarkably simple way to draw definite conclusions from the vague information. In fuzzy reasoning, the most important is the Generalized Modus Ponens (GMP), which uses *If-Then* rules that

implicitly represent a fuzzy relation. The fuzzy rules define the connection between the input and output fuzzy (linguistic) variables. The fuzzification stage is defined as the process to convert the crisp value input $u_\circ$ to a fuzzy point A' which is defined in the Eq. 5.4. The degree of membership can be 0 to 1 for a particular input as shown in the Fig. 5.2.

$$\mu_{A'}(u) = \begin{cases} 1, & \text{if } u = u_\circ. \\ 0, & \text{otherwise.} \end{cases} \tag{5.4}$$

In the rule-based system for more than one rule, the fuzzy rules that describe the fuzzy system are given as the following:

r_1 : **IF** the Joystick is High Negative **AND** the Obstacle Distance is Close **THEN** the Speed is Low Negative

.....

r_k : **IF** the Joystick is Low Positive **AND** the Obstacle Distance is Far **THEN** the Speed is Low Positive

.....

r_n : **IF** the Joystick is High positive **AND** the Obstacle Distance is Very Far **THEN** the Speed is High Positive

Therefore, the translation of these rules is done by constructing the fuzzy relation R_n for each rule r_k and then combining these relations into a fuzzy relation R as described in the Eq. 5.5. This process of combining the fuzzy rules into a fuzzy relation is called aggregation. This aggregation relies on the type of the relation used to represent the rule in the rule base system. In the case of Mamdani approach, R_k is represented by using the conjunction operator and then the aggregation is done by using the disjunction operator.

$$R = \bigcup_k R_k \tag{5.5}$$

When the rule is represented by Mamdani (conjunction relation) then the inference is obtained as given in the Eq. 5.6.

$$\mu_{c^*}(w) = \bigvee_u \bigvee_v \Big[(\mu_A(u) \wedge \mu'_A(u)) \wedge (\mu_B(v) \wedge \mu'_B(v)) \wedge \mu_C(w) \Big] \forall w \in W \tag{5.6}$$

Defuzzification converts the fuzzy sets into crisp value by Center of Gravity Method as described in the Eq. 5.7.

$$w^* = \frac{\sum_i \mu_{c^*}(w_i) w_i}{\sum_i \mu_{c^*}(w_i)} \tag{5.7}$$

The intelligent fuzzy sets model is divided into two main parts. First, there are three sonar sensors (S1, S2, S3) mounted in front of the robot to detect the obstacles in the environment. The angle between these sensors is 20° as shown in the Fig. 5.4. These sensor readings are fed to an algorithm. This algorithm finds the shortest distance between the robot and the obstacle. The view angle of the sonar sensor is 24.5°. Therefore, there is no conflict between the sensor readings. Accordingly, there are three probabilities for the sensors: S1 is the shortest or S3 is the shortest or S2 is the shortest. Second, fuzzy control has been designed in order to control the speed of the robot along with the random delays due to communication through the shared medium like the Internet. The controller is based on two inputs and one output. The first input is the joystick command that comes from the user with five membership functions as shown in the Fig. 5.5, and the linguistic variables are (high negative HN, low negative LN, zero Z, low positive LP, and high positive HP). Second input comes from the algorithm. It has five triangular membership functions and five linguistic variables (very close VC, close C, medium M, far F and very far VF) as illustrated in the Fig. 5.6. The controller has one output that is the speed of the robot with five triangular membership functions (negative high NH, negative low NL, Zero Z, positive low PL, and positive high PH) as shown in the Fig. 5.7. The generalized modus ponens and T-norms have been used to design the fuzzy rule base with 25 rules in terms of the relationship between the inputs and the output as it is illustrated in Table. 5.1. These rules have been defined in such a way that they combine both inputs with AND operator. This means that until and unless both inputs are not available then there would not be any output. Therefore delayed input from operator and sensors reading produce output speed. The Fig. 5.8, demonstrates the fuzzy surface generated by two inputs and one output. It shows the output speed based on the input from the joystick and the sonar readings.

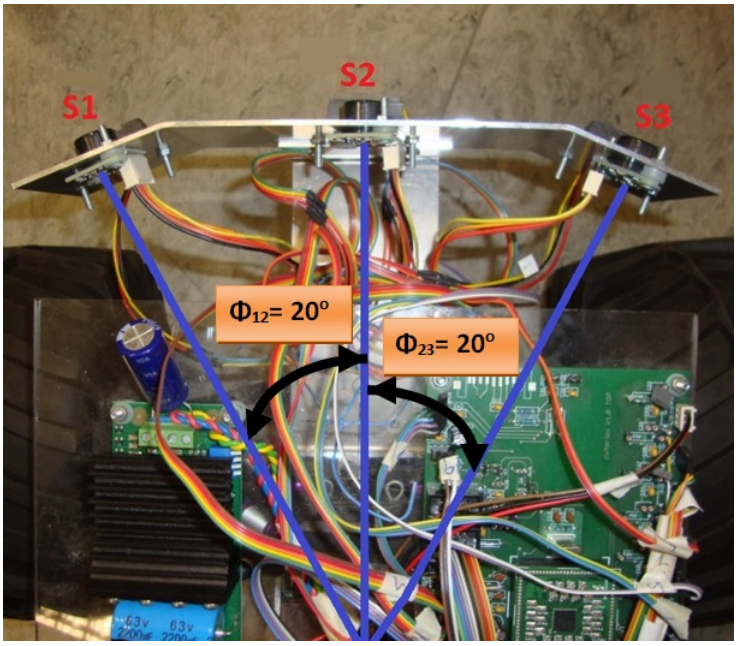

FIGURE 5.4: The sonar sensors mounted in front of the robot

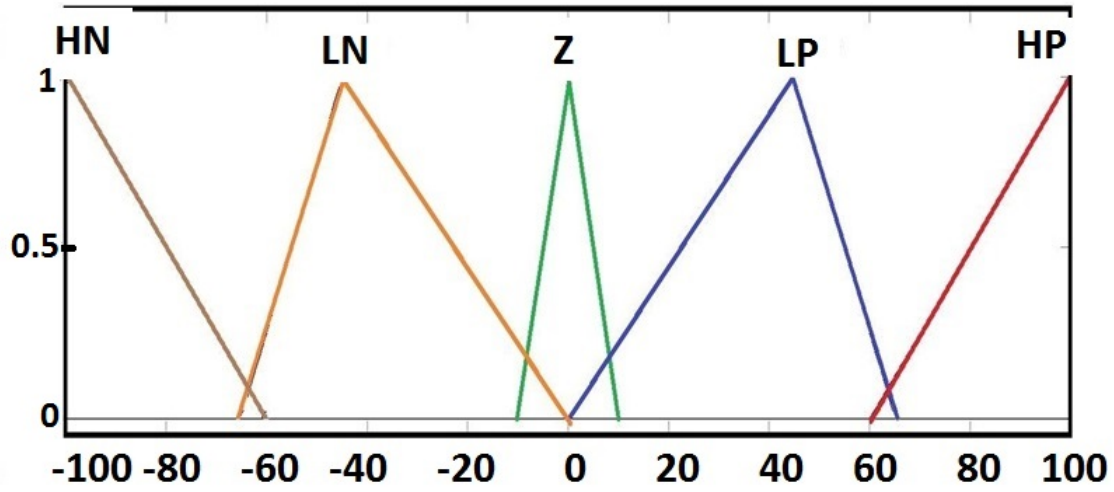

FIGURE 5.5: Fuzzy membership functions for the first input

TABLE 5.1: The fuzzy rules base for the speed controller

Sonar Readings	***Joystick Commands***				
	HN	***LN***	***Z***	***LP***	***HP***
VC	NL	NL	Z	Z	Z
C	NL	NL	Z	Z	Z
M	NL	NL	Z	PL	PL
F	NL	NL	Z	PL	PL
VF	NL	NL	Z	PL	PH

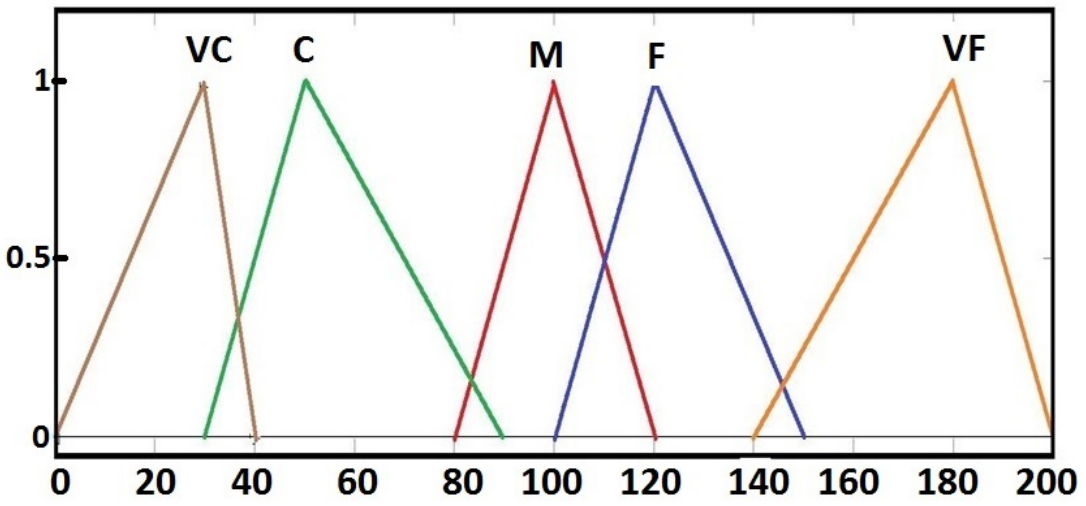

FIGURE 5.6: Fuzzy membership functions for the second input

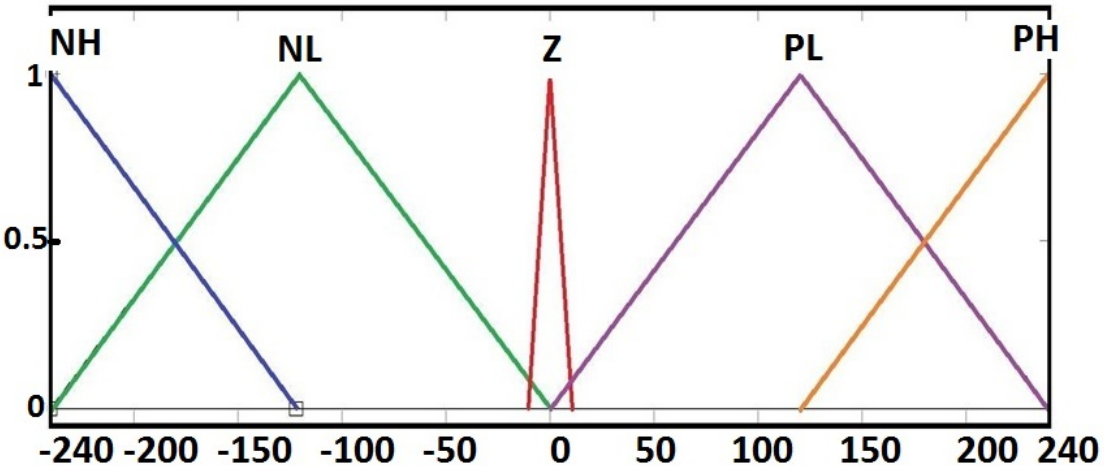

FIGURE 5.7: Fuzzy membership functions for the output speed

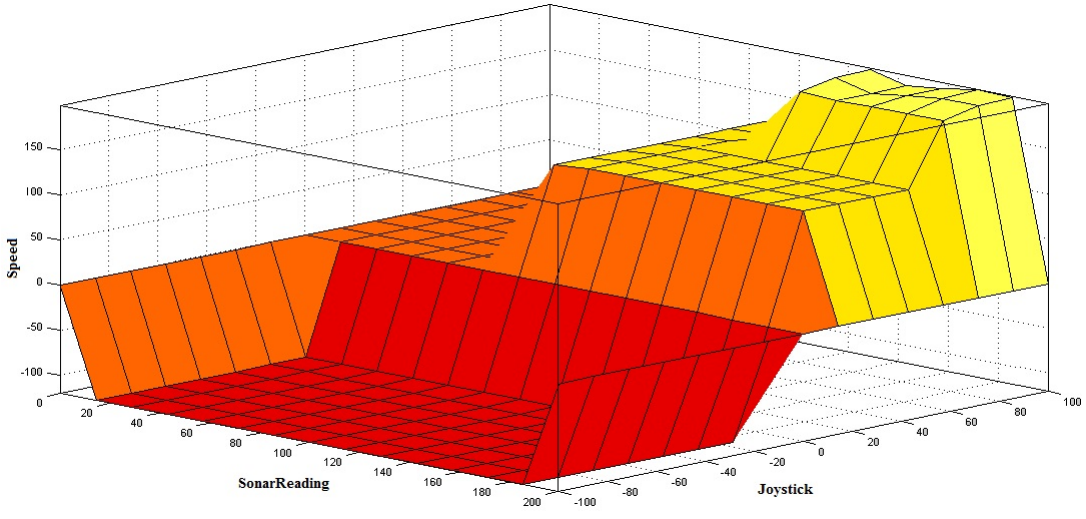

FIGURE 5.8: The fuzzy surface

5.5 Experimental Results

Several experiments have been carried out to analyze the behavior of the controller. The robot has been tested in the indoor environment and the results have been plotted to show the performance during the experiments. Fig 5.9, shows the input velocity. Fig. 5.10, shows the distance sensor S1 reading over the span of 130 samples with the blue line during the test run. Similarly, the Fig. 5.11, exhibits the middle sensor values with a red line and the Fig. 5.12, illustrates the values of the third sensor readings with a dark green line. The Fig. 5.13, is showing fuzzy speed controller output with a light green line.

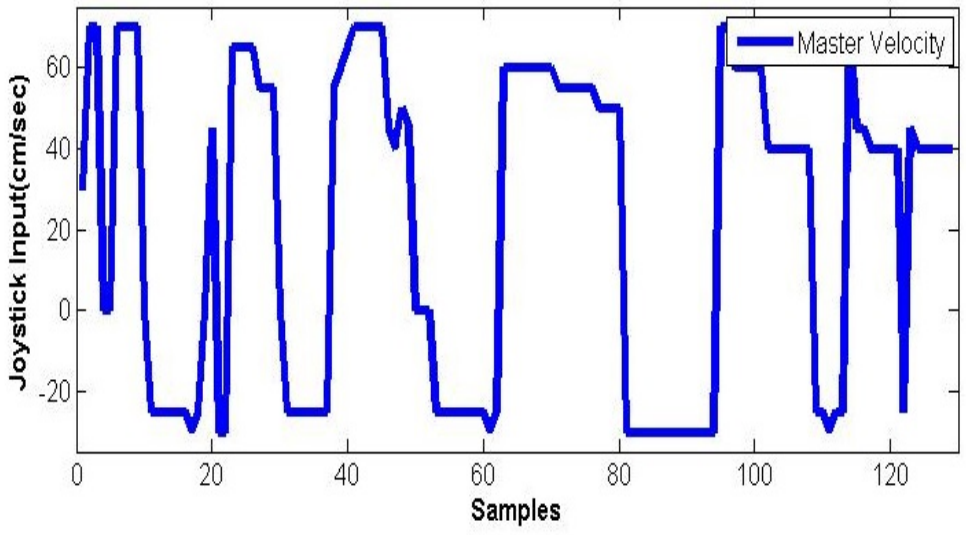

FIGURE 5.9: The joystick input

The robot follows the human operator commands when there is no obstacle and the variation in the speed is brought by the operator. When there is no obstacle, then the sensors are showing the maximum distance i.e. 200cm. In this case, the robot is following the human operator and the variation in the speed is being brought by the operator commands according to the fuzzy rules defined in the Table. 5.1. The operator is driving the robot in forward or backward directions, but when the objects are being detected by the sensors in the vicinity of the robot then the speed is reduced as shown in the speed plot, and the robot stops when it goes very close to the obstacle as shown in the same figure after 120 samples.

The Fig. 5.14, shows the performance of the controller. These sequential images showing the performance of the controller from top left to bottom right. The robot

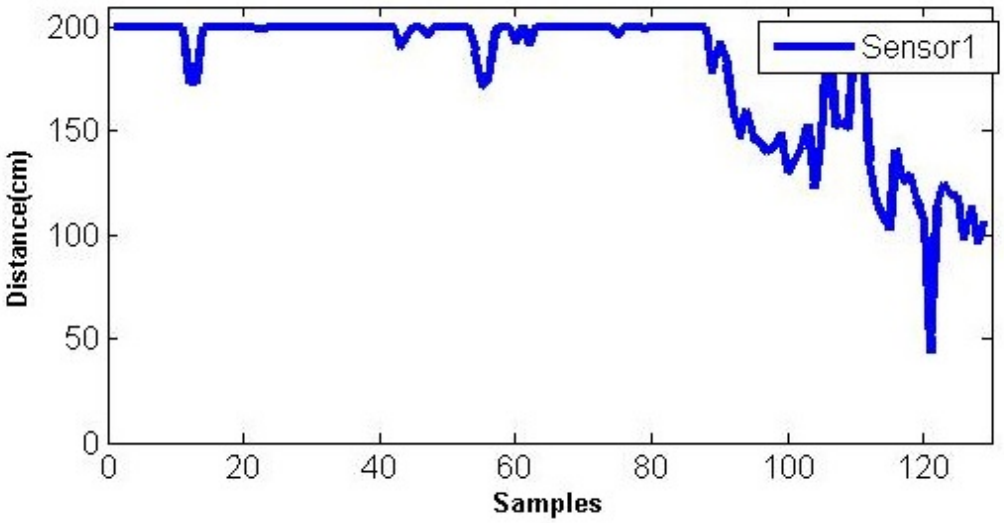

FIGURE 5.10: The readings of the sensor 1

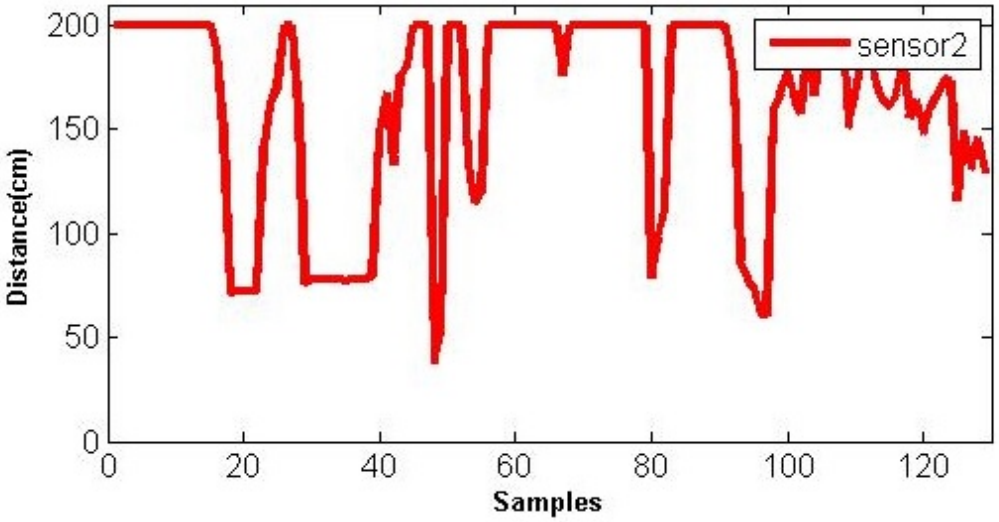

FIGURE 5.11: The readings of the sensor 2

moves according to the commands from the human operator, but once it detects the dynamic obstacle it reduces its speed regardless of the value of the input from the human operator and stops when it goes very close to the obstacle and only follows the human operator command in the backward direction and ignore the commands for the forward motion. The Fig. 5.15, shows the teleoperation using the wireless network in the RST Institute. The robot is calculating the output speed based on the environment. It is navigating from one place to another place safely as indicated by sequential images. The first image on the top left in the figure shows the start point and the last image on the bottom right shows the end point of robot navigation.

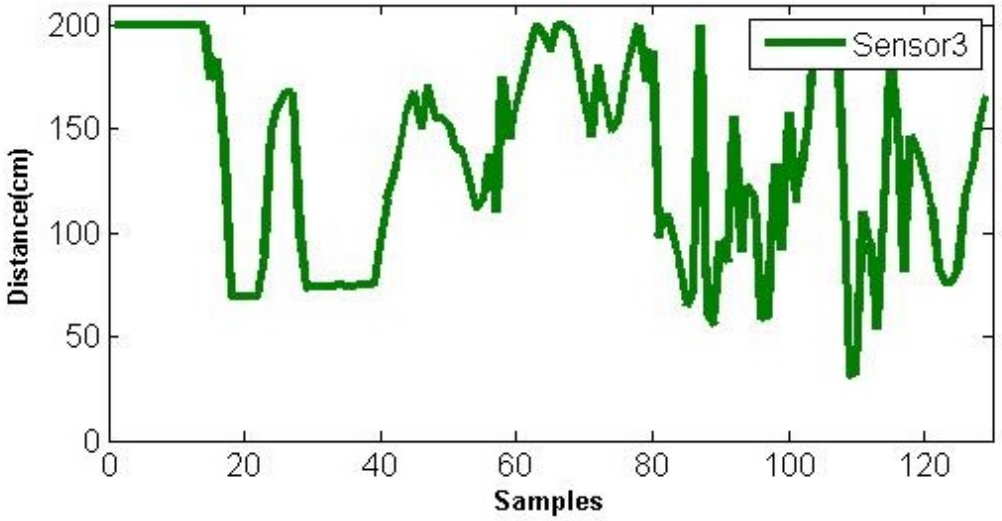

FIGURE 5.12: The readings of the sensor 3

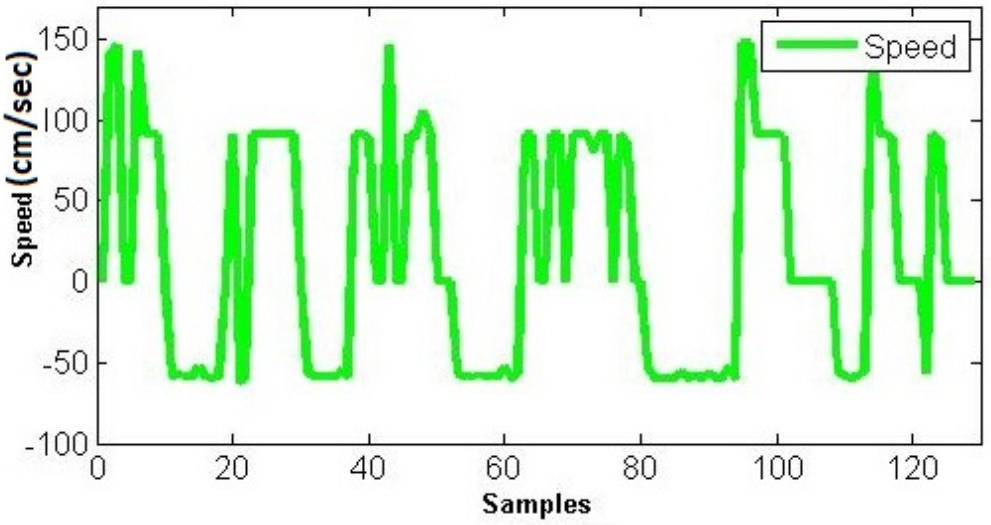

FIGURE 5.13: The output speed of the AutoMerlin mobile robot

FIGURE 5.14: The robot is detecting the human and the obstacles and reducing the speed

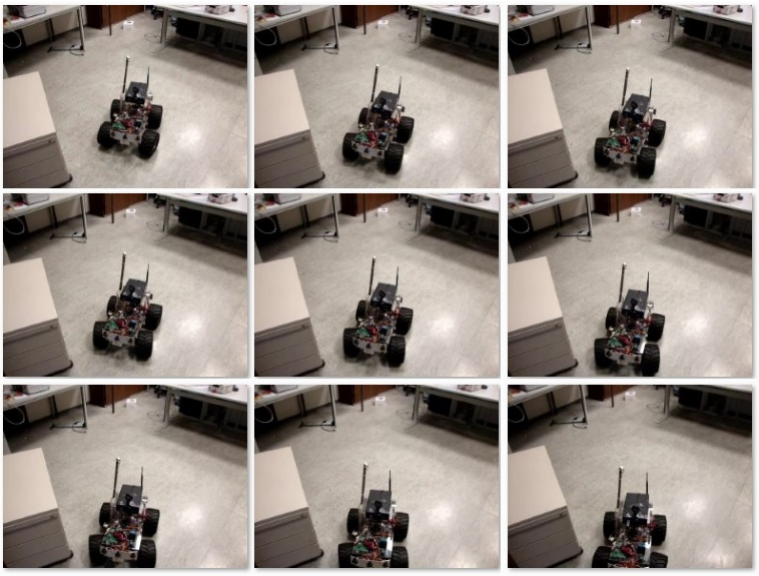

FIGURE 5.15: The robot's navigation in the lab

5.6 Free Intelligent Navigation (FIN) Algorithm

The teleoperation of a mobile robot becomes a difficult task if the environment around the mobile robot is not presented to a human operator precisely. The lower quality of the information delivered to the operator has a negative impact on the perception of the remote environment and often leads to the incorrect decisions. For instance, relying exclusively on the video feedback commonly leads to the disorientation, incorrect depth estimation, or failure to detect the obstacles in the unstructured environments. These negative effects due to the separation of the operator from the point of action site become even more significant in the applications where the precise maneuvering is required. In order to teleoperate the mobile robot AutoMerlin, the human operator sets the linear and the angular velocities using a joystick. The robot follows the human operator, but sometimes due to a delay in teleoperation or incorrect vision information or operator's false perception of the environment around the mobile robot, the robot goes very close to the obstacles and collides with them before the operator takes the necessary steps. Therefore, an ancillary intelligence has been added to the existing speed controller to avoid the obstacles autonomously during teleoperation when they are in the critical range.

The obstacle avoidance algorithm called *Free Intelligent Navigation (FIN) Algorithm* detects the obstacles in the vicinity of the robot with the help of proximity sensors. Then, it takes the control of the robot for the safe navigation and diverts the robot to avoid collision with the obstructions and then shifts the control back to the operator. The FIN Algorithm adjusts the servo and the speed of the robot. It helps in teleoperation in the presence of a random time delay. The AutoMerlin mobile robot listens to the human operator commands but sometimes in the catastrophic environments where the availability of the commanded signals is drastically limited or unavailable for the certain time intervals then the robot does not collide with the surrounding objects but instead the FIN Algorithm takes over and guides the robot with the constant speed and the suitable steering value so that it does not collide with the objects and when the connection is resumed then the robot again follows the human operator provided that there is no obstacle in the critical range. However, in some emergency cases, the collision monitoring is also realized on the front infrared sensors. If a very near object is detected by

the IR sensors then, the robot is instructed to stop. The security distance, verified by the IR sensors is about 40cm for the usual objects in the lab. The sonar measurements are in the range of 0.3m to 2m.

The FIN Algorithm described in the Table 5.2, has been designed for the obstacle avoidance. The front sensors S1, S2, S3 provide the distance readings R1, R2, R3 respectively. Its first objective is to find the shortest distance SD among these three readings, and based on SD to find whether the obstacle is on the right, middle or left side of the robot is. These scenarios are shown in the Fig. 5.16, Fig 5.17 and Fig. 5.18. In the light of the preceding conclusion, the algorithm turns the robot for the best orientation to avoid the obstacle. The working of the FIN Algorithm is depicted in the Fig. 5.19. The step by step execution is described in the Table. 5.2.

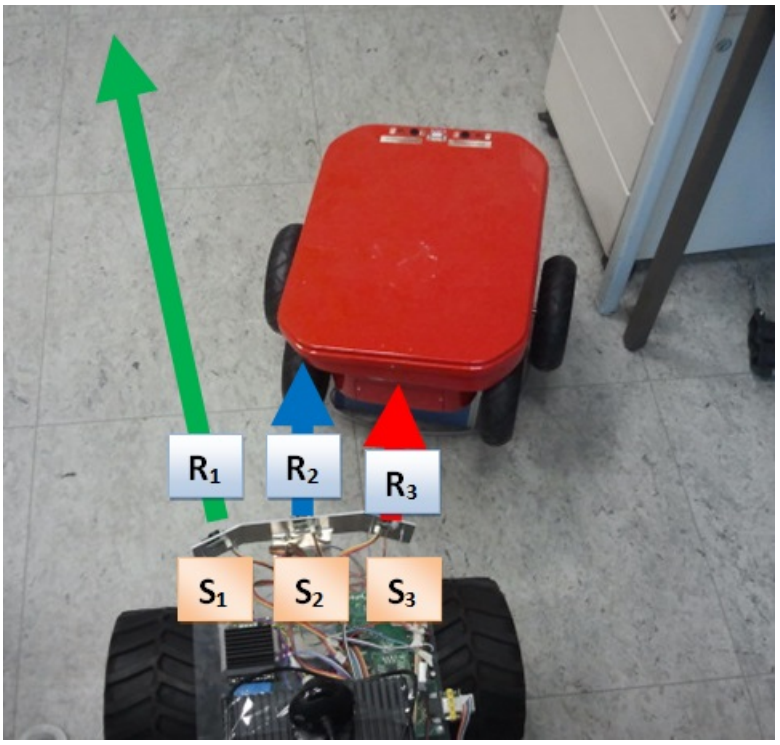

FIGURE 5.16: The obstacle is on the right of the robot

5.6.1 Sub Controller

The well-known limitations of the ultrasonic sensors are the uncertainties and the drawbacks in the information. Motlagh demonstrated that the fuzzy sets systems might model the uncertainties using the linguistic rules[110]. Cliff Joslyn introduced a method to construct the possibility distribution and the fuzzy logic

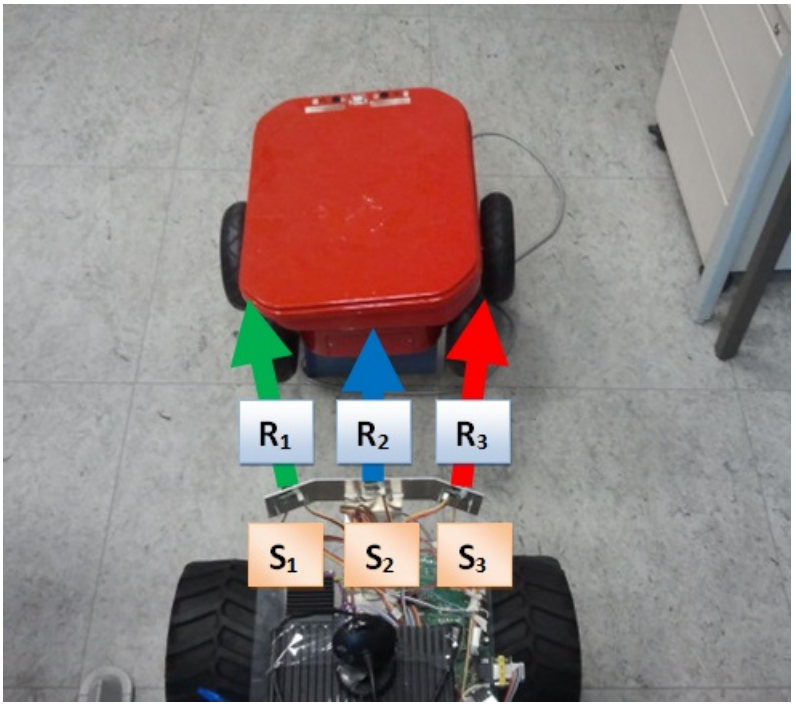

FIGURE 5.17: The obstacle is infront of the robot

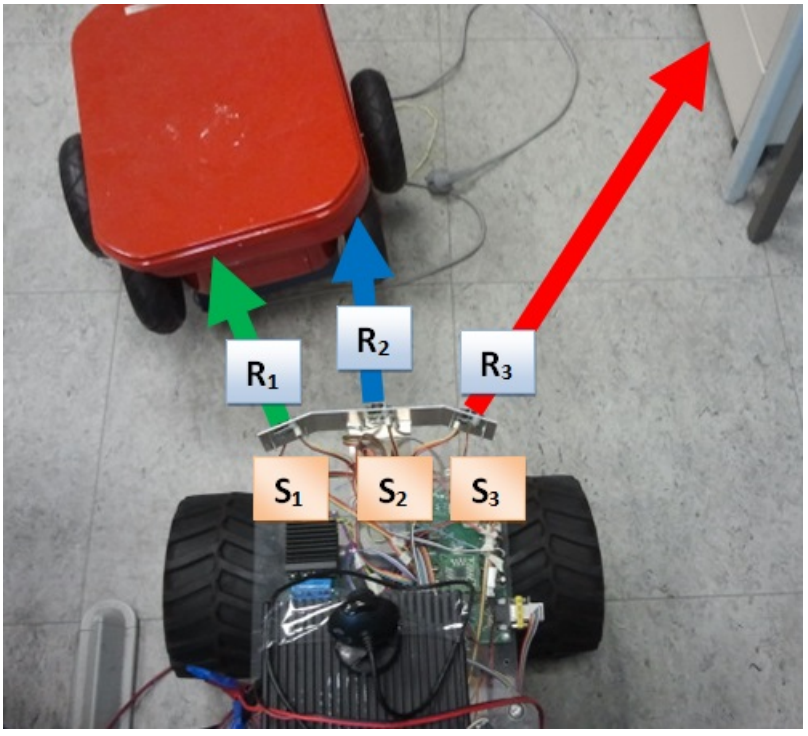

FIGURE 5.18: The obstacle is on the left of the robot

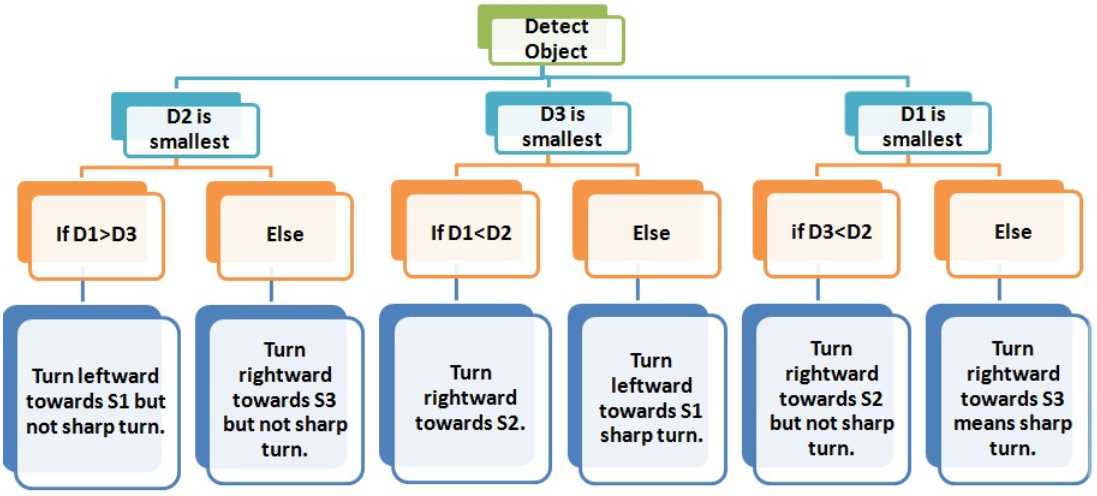

FIGURE 5.19: The execution of the FIN Algorithm

from the empirical data by collecting the data and constructing the interval set statistics with the random sets[111, 112]. Therefore, there are two main stages that have been designed and implemented. At the first stage, Probabilistic Fuzzy Logic System (PFLS) described in Ch. 8, deals with the drawbacks in the sensors, rectify the sensor readings and then uses the proper fuzzy sets to find the shortest distance between the AutoMerlin mobile robot and the obstacles. The second stage uses the output of the sub-fuzzy set model to generate the main behavior of the robot.

In order to reduce the drawbacks in the sonar sensors, a fuzzy set model was modeled by using the possibilities distribution theory. However, the experimental data show the values of these errors are related to the range of view β as well as the distance between the sensor and the object. As a result, these errors can be reduced and modeled by the fuzzy sets and the possibility distributions as shown

TABLE 5.2: The FIN Algorithm

Input Sensor Readings	***(S1,D1), (S2,D2), (S3,D3)***

Correct Sensor readings by Fuzzy set Model
Obtain New Distances: (S1,D1*), (S2,D2*),(S3,D3*)
Calculate the shortest distance SD
$FIN_{Initial}$= {min(D1, D2), min(D1,D3),min(D2,D3)}
FIN_{Final}= min { D}
While: Navigate
IF: SD >Safety Distance Navigation.
Else: Find $FIN_{initial}$
Then estimate the FIN_{Final}
End IF
END WHILE.
Return Navigation

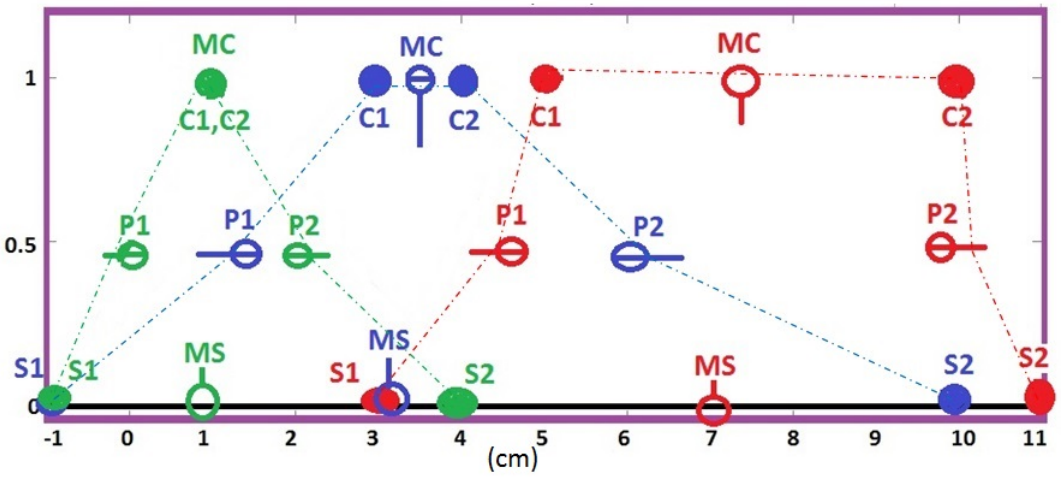

FIGURE 5.20: The possibilities histograms

TABLE 5.3: The analysis of the frequency data

$\mathbf{A}_i$	$\mathbf{S}_i$
A_1=<[0,1],[1,3],[0,3],[-1,4]>	S_1={[0,1]=0.25},[1,3]=0.25,[0,3]=0.25,[-1,4]=0.25}
A_2=<[-1,4],[0,5],[2,7],[3,10]>	S_2={[-1,4]=0.25,[0,5]=0.25,[2,7]=0.25,[3,10]=0.25}
A_3=<[3,10],[4,11],[5,11]>	S_3={[3,10]=1/3,[4,11]=1/3,[5,11]=1/3}

TABLE 5.4: The analysis of the possibilities information

$\mathbf{E}_i^L$	$\mathbf{E}_i^R$	$\mathbf{C}_i(\pi)$	$\mathbf{Supp}_i(\pi)$
{-1,0,0,1}	{1,3,3,4}	[1,1]	[{-1,0},{0,1},{1,1},{1,3},{3,4}]
{-1,0,2,3}	{4,5,7,10}	[3,4]	[{-1,0},{0,2},{2,3},{3,4},{5,7},{7,10}]
{3,4,5	{10,11,11}	[5,10]	[{3,4},{4,5},{5,10},{10,11}]

in the Table. 5.3. The vectors of endpoints, cores, and support for the possibilities distribution are shown in the Table. 5.4. The possibilities histograms are shown in the Fig. 5.20, which can be converted to the fuzzy membership functions as shown in the Fig. 5.21, where C_i are the cores for the histograms, P_i are other points, S_1 and S_2 are the start and the end point for the support, M_C is the midpoint for the core of each histogram. The data given in these tables have been used to construct the histograms in the Fig. 5.21. These histograms show the range of error in the sensor readings.

The FIN Algorithm decides the smallest distance reading with the known radial error (ϵ) and the view angle as given in the Eq. 5.8. α represents support of fuzzy set i.e full range of membership function i.e from -1 to 4 for green membership function of Fig. 5.21. Then, rotates the reading distance (RD) to the original axis coordinate to find the shortest distance (SD) as given in the Eq. 5.9. Finally, to

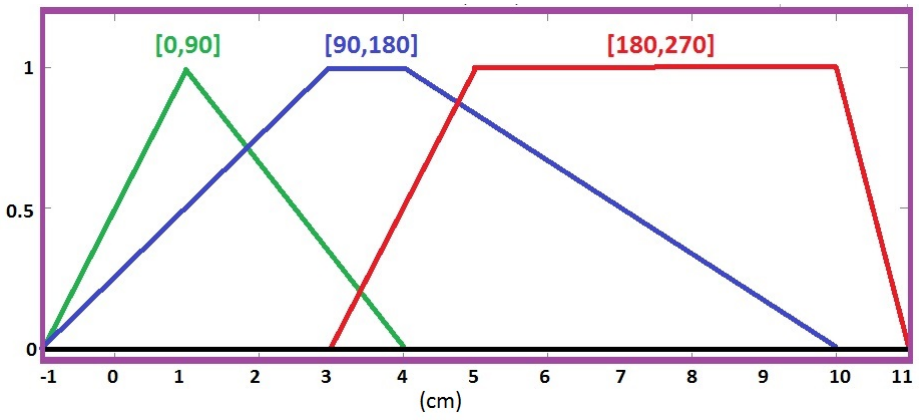

FIGURE 5.21: The fuzzy membership functions for the fuzzy sets model

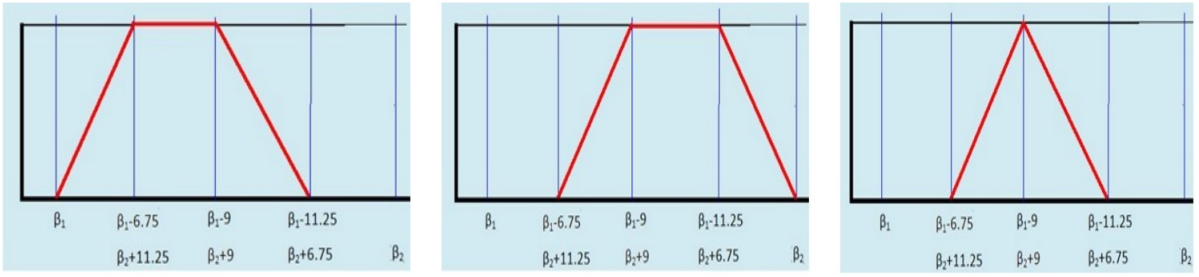

FIGURE 5.22: The fuzzy membership functions for β (R1, R2, R3)

estimate the shortest distance the T-norms have been used as given in the Eq. 5.10.

$$\mu_{SD}(x) = Supp\{\mu(\epsilon, \beta)\}, \alpha^1 < \beta < \alpha^4 \tag{5.8}$$

$$SD_x = [\epsilon \pm RD] \cos(\mid \beta \mid) \tag{5.9}$$

$$Supp_\mu = \sum_{\mu=1}^{4} min(SD) \tag{5.10}$$

For simplicity, assume the R1 is the shortest distance and it is 80 cm, then, the R1 has the green membership function as it is shown in the Fig. 5.21, because 80 ϵ [0, 90]. The radial error has four values ϵ= [-1, μ=0], [1, μ=1], [1, μ=1], [4, μ=0]. Therefore, SD= [80-1, μ=0]* cos(β1), [80+1, μ=1]* cos(β1), [80+1, μ=1]* cos(β1), [80+4, μ=0]* cos(β1). Then, SD= [80+1, μ=1]* cos(β1).

The value of β, can be obtained by three possibilities for the readings: R1, R2 or R3 is the shortest reading. The membership functions for the three possibilities based on view angle are shown in the Fig. 5.22. Now the value β in case R1 is the shortest is a trapezoidal shape with 4 values 0, 6.75, 9, 11.25. Thus, SD= $\{[80+1, \mu = 1] * cos(0), [80+1, \mu = 1] * cos(6.75), [80+1, \mu = 1] * cos(9), [80+1, \mu =$

1]$*cos(11.25)$} and by using the Eq. 5.10, the SD= min(81,80.439, 80.003, 79.44)= 79.44 cm.

5.7 Experimental Results

The Fig. 5.23, Fig. 5.24, Fig. 5.25, and Fig. 5.26, show the three sensor readings and the steering angle value of the robot. The robot is following the human operator when there is no obstacle. But, when any of the sensors detect any object within a range of 1m, then, the FIN Algorithm comes into play to avoid it and steers the robot in an appropriate direction. The robot would turn in the rightward direction when the value of the steering is positive and leftward direction when the value is negative e.g. at approximately 80th sample the sensor1 and the sensor2 mounted at the left and the middle of the robot respectively, detect some object, and the robot turns rightward indicated by the positive value in the steering plot. Same behavior is repeated from sample 95 to 100.

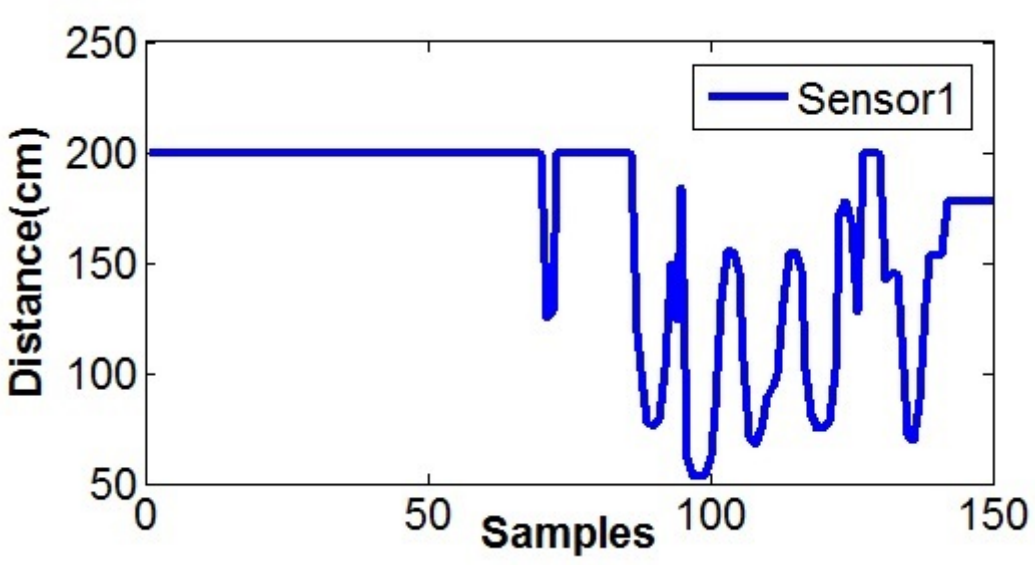

FIGURE 5.23: The sonar sensor S1 plot

In another experiment, the performance of the FIN Algorithm has been tested as shown in the Fig. 5.27. It is vivid from the figure that the sensor1 is reading the maximum distance in the beginning, so the robot is following the human operator. When, the obstacle is detected, i.e. the reading of sensor is below 100cm then either the operator can move the robot backward or the FIN Algorithm starts functioning and reduces the speed to a constant value and steers the robot in a

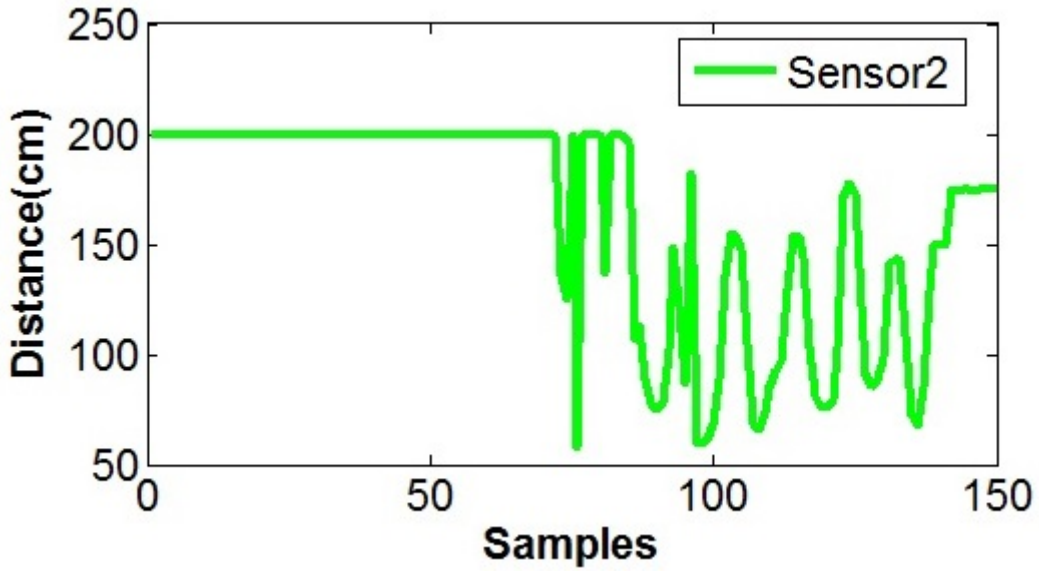

FIGURE 5.24: The sonar sensor S2 plot

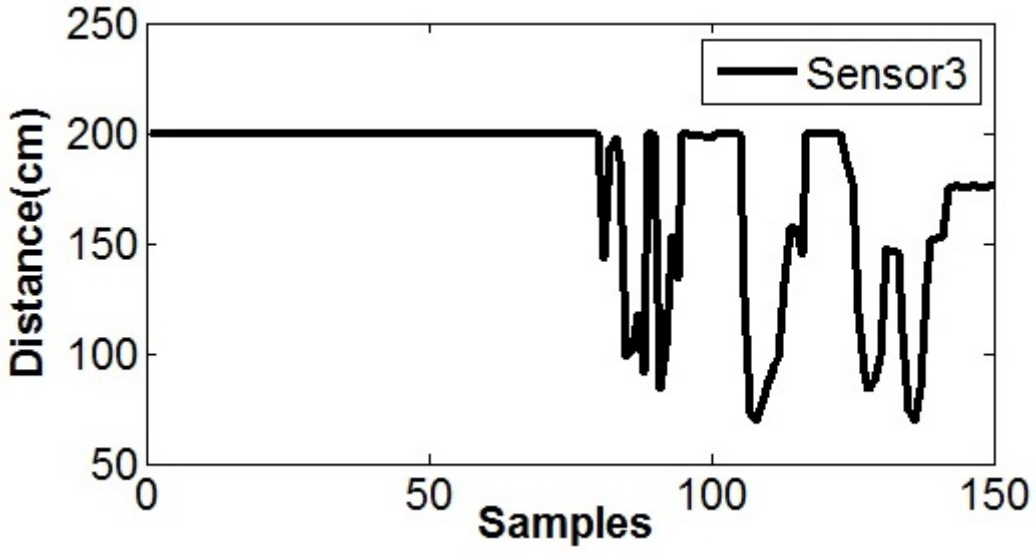

FIGURE 5.25: The Sonar sensor S3 plot

suitable direction. In this scenario, the obstacle is on the left side of the robot and therefore the steering action is positive, i.e. in the rightward direction and the positive speed for the forward movement is constant.

The Fig. 5.28, and the Fig. 5.29, demonstrate the performance of the FIN Algorithm to avoid the obstacles present in the environment in the form of sequences images in the two test runs. The robot moves across the RST lab without colliding with the other robots and objects. Top left image is start point and bottom right image is end point of the test run. From these sequential images it is clear that robot is turning leftward after detecting obstacle on the right side.

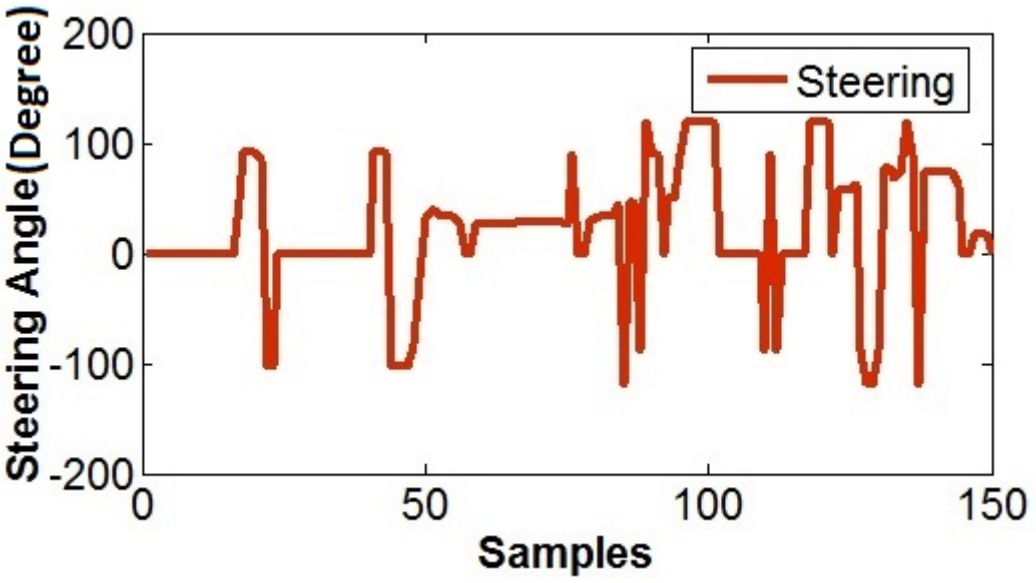

FIGURE 5.26: The steering angle plot

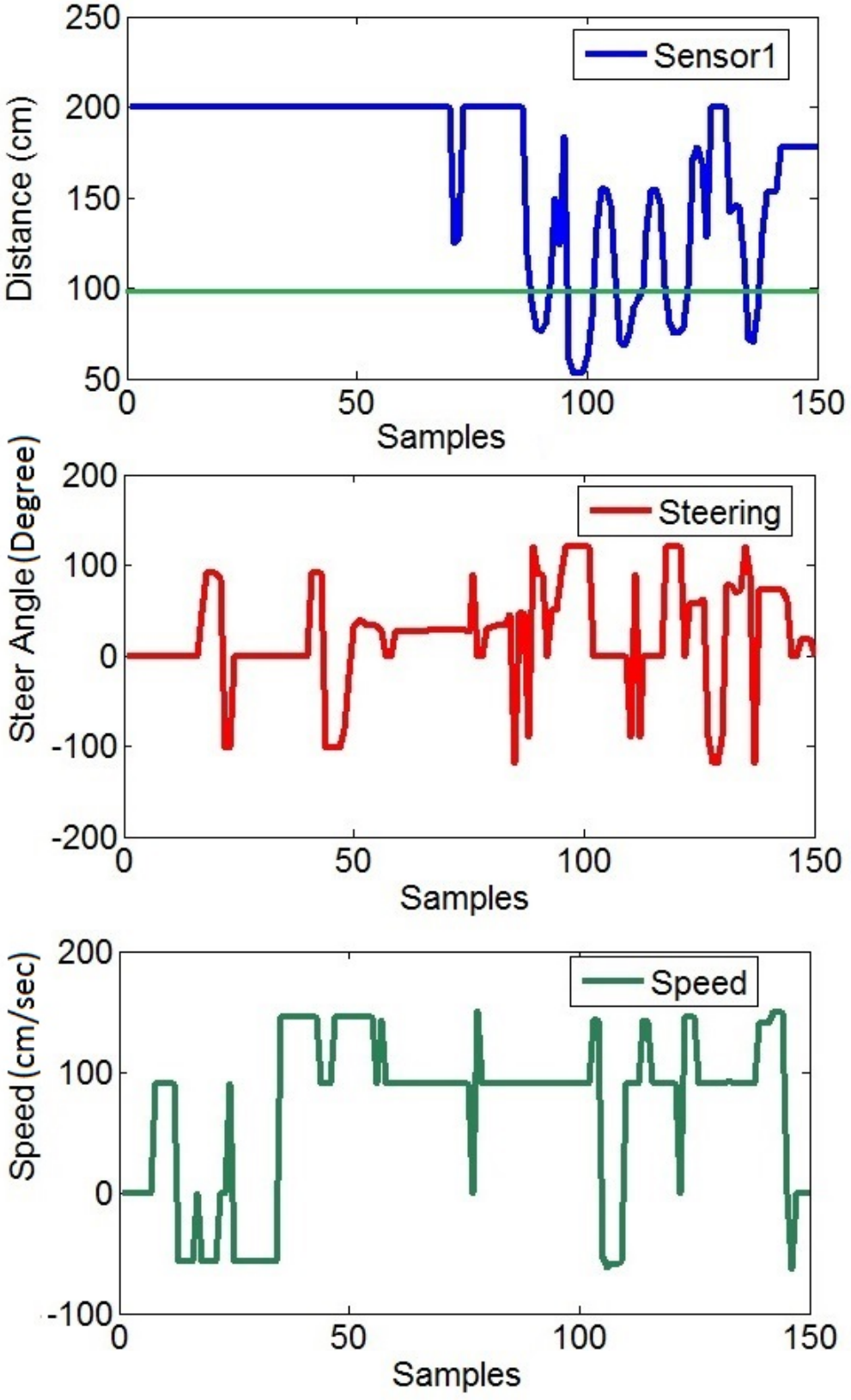

FIGURE 5.27: The Sensor1 reading, the steering angle, and the output speed

FIGURE 5.28: The FIN Algorithm performance test 1

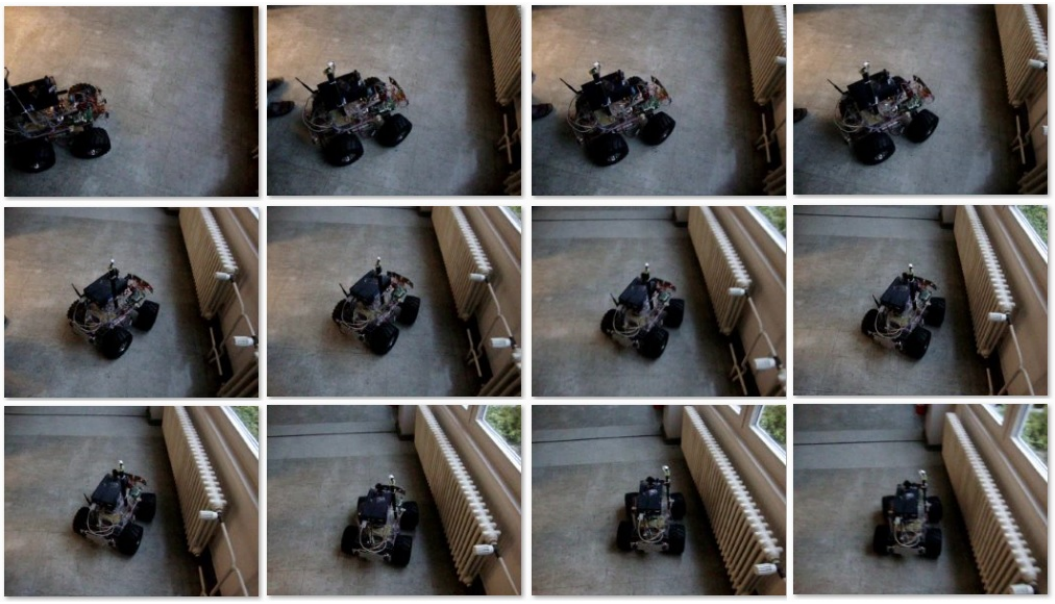

FIGURE 5.29: The FIN Algorithm performance test 2

Chapter 6

Time Domain Passivity Control

Passivity is an efficient approach that can be utilized in different ways to establish the stability of a dynamic system. Some control approaches based on passivity have been described in Ch. 3. For example the wave variables approach revamps the power variables into the wave variables and ensures the passivity of the time delayed systems using the passive transmission property of the waves. Time Domain Passivity Control is a remarkable technique to ensure the stability in the passivity domain. Time Domain Passivity Control(TDPC) has been proposed by the Ryu and Hannaford for the control of a haptic interface[113, 114]. TDPC is based on the continuously monitoring passivity of the system in real-time and the calculation of a control action based on the real-time observability of the system energy. Hence, passivity is no longer a design constraint which affects the controller design and transparency of the system, but rather based on the observation of the excess energy and its dissipation to keep the system passive. Stable control of a complex system such as teleoperation of a mobile robot with the force feedback can be developed with Time Domain Passivity Control. It is based on a Passivity Observer which monitors the net energy and passivity of the system and a Passivity Controller to dissipate the extra active energy to make it passive again.

6.1 Force Modeling

The virtual force acting on the mobile robot is given in the Eq. 6.1. It has been calculated based on two factors, i.e. the relative distance between the robot and

the obstacles and the force of friction acting on the mobile robot. The force f_d given in the Eq. 6.2, is based on the distance between the robot and the obstacle and the linear velocity of the robot. It has been modeled as a spring-damper system. The angle α is 20° or zero between the robot and the obstacle based on the position of the obstacle as shown in the Fig. 6.1. The f_f as described in the Eq. 6.3, is the force of friction acting on the mobile robot during navigation and the coefficient of friction μ for indoor movement is 0.3. N is normal reaction of the surface. The environment force f_e is the addition of these two forces as given in the Eq. 6.4.

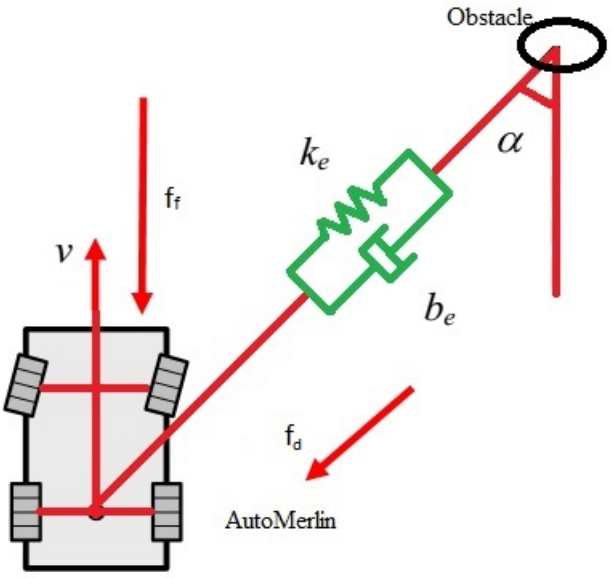

FIGURE 6.1: Forces acting on the AutoMerlin

$$f_e = f_f + f_d \tag{6.1}$$

$$f_d = (b_e V + k_e X)\cos\alpha \tag{6.2}$$

$$f_f = \mu N \tag{6.3}$$

$$f_e = \mu N + (b_e V + k_e X)\cos\alpha \tag{6.4}$$

6.2 Time Domain Passivity Control

Bilateral teleoperation loop has been shown in the Fig. 6.2. It consists of a human operator, a master device, a communication channel, a slave robot, and an environment. The master device and the robot are passive as they dissipate the

energy due to the presence of the actuators in them. The network adds the extra energy in the system to make it active and unstable. To cope with this issue the network has been modeled as a Two-port network as shown in the Fig. 6.3. V_M and V_S' are the velocities of the master and slave robot respectively. The slave force f_S acts on the slave robot and f_M' is the force acting on the master device. The teleoperation is passive and stable if the Two-port network is passive along with a human operator and the environment. The passivity of a Two-port network can be defined as follows.

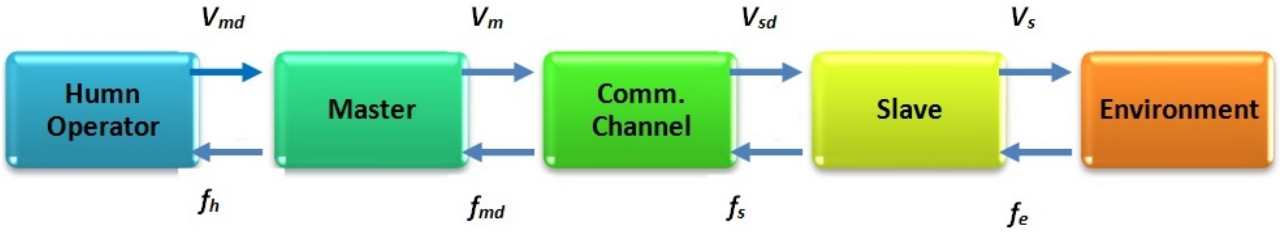

FIGURE 6.2: Bilateral teleoperation

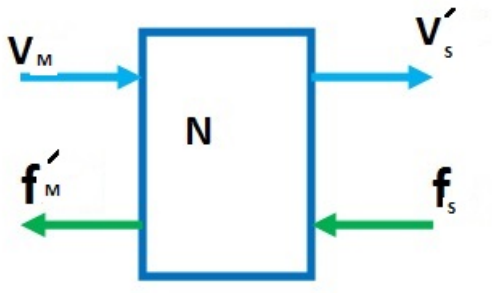

FIGURE 6.3: A Two-port network

Let E(0) represent the initial energy of the system. The total energy E(t) after time t can be written as given in the Eq. 6.5. The teleoperation would be passive and stable for all the time t if E(t) is positive.

$$E(t) = \int_0^t P(\tau)d\tau + E(0) \geq 0 \tag{6.5}$$

E(t) represents the total energy of the system at time t. By considering the initial energy to be zero, the Eq. 6.5, can be written in term of power as a product of the input and the output as given in the Eq. 6.6.

$$E(t) = \int_0^t P(\tau)d\tau = \int_0^t u^T(\tau)y(\tau)d\tau \geq 0 \tag{6.6}$$

u and y are input and output of the system in Eq. 6.6. In teleoperation the input is the f(force, voltage) and the output is the V(velocity, current) and the above equation with these parameters can be modified as given in the Eq. 6.7.

$$E(t) = \int_0^t ((f'_M(\tau)V_M(\tau)) - (f_S(\tau)V'_S(\tau)))d\tau \geq 0 \tag{6.7}$$

For passivity, the output energy should never be more than the input energy at all times.

6.2.1 Passivity Observer

A Passivity Observer (PO) as shown in the Fig. 6.4, keeps the track of the energies which go into the system and comes out of the system. The value of the energy is calculated according to the relation given in the Eq. 6.8. If at any moment the net value of the energy is negative, then the system is no more passive, but becomes active and then Passivity Controller dissipates any surplus energy to make the system passive again[113, 114].

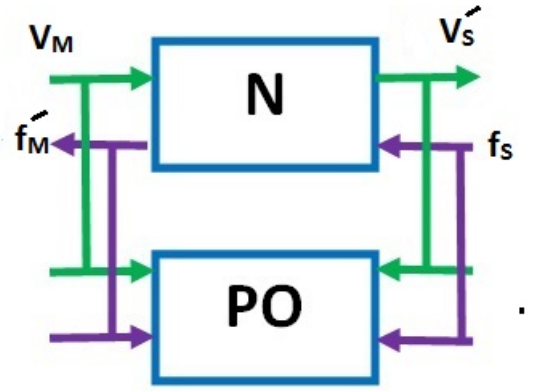

FIGURE 6.4: A Passivity Observer

$$\begin{aligned} E(n) = E(n-1) + f'_M(n)V_M(n) + f_S(n)V'_S(n) + \\ \alpha_M(n-1)V_M(n-1)^2 + \alpha_S(n-1)f_S(n-1)^2 \end{aligned} \tag{6.8}$$

6.2.2 Passivity Controller

Whenever the net energy reported by the Passivity Observer (PO) is negative, then the Passivity Controller takes the necessary action to dissipate the surplus

energy. Here, there are two controllers, i.e. the master controller α_M and the slave controller α_S. The α_M is present on the master side and α_S is present on the slave side. There are different scenarios to calculate both the master and the slave controllers. If the net energy E(n) calculated by Passivity Observer (PO) is negative, then the system behavior is active and then the controller monitors which side is showing an active behavior whether it is the master or the slave or both. If only the master side is active, then the value of the master controller α_M would be calculated as given in the Eq. 6.9, and the α_S would be zero. On the other hand, if the slave side is active, then the value of the slave controller would be calculated as given in the Eq. 6.10, and the master controller would be zero. If both sides are active, then both controller would dissipate the surplus energy. If both sides are active for the two consecutive samples like E(n) and E(n-1), then the values of both controllers are calculated according to the Eq. 6.11, and the Eq. 6.12, for the additional conservation of the energy[113, 114].

$$\alpha_M(n) = -\frac{E(n)}{V_M(n)^2} \tag{6.9}$$

$$\alpha_S(n) = -\frac{E(n)}{f_S(n)^2} \tag{6.10}$$

$$\alpha_M(n) = -\frac{f'_M(n)}{V_M(n)} \tag{6.11}$$

$$\alpha_S(n) = -\left[\frac{E(n-1) + f_S(n)V'_S(n)}{V'_S(n)^2}\right] \tag{6.12}$$

After the calculation of the master and slave controller, the value of the slave velocity and the force feedback is modified as shown in the Fig. 6.5. It is vivid from the Eq. 6.13, and the Eq. 6.14, that if any controller has zero value, then the corresponding velocity or force would not be modified and if any or both have some value, then they would modify the corresponding signal.

$$f_M(n) = f'_M(n) + \alpha_M(n)V_M(n) \tag{6.13}$$

$$V_S(n) = V'_S(n) + \alpha_S(n)f_S(n) \tag{6.14}$$

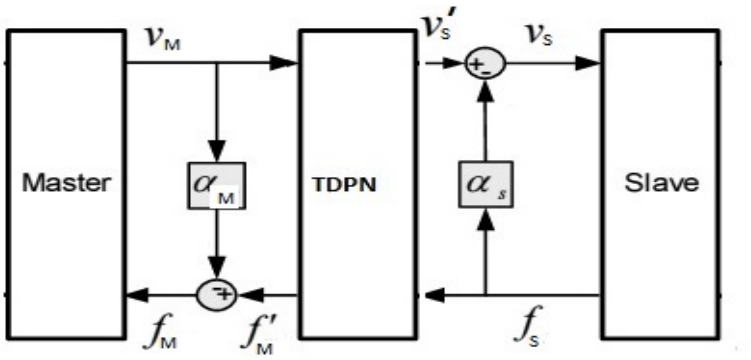

FIGURE 6.5: The Passivity Controller adjusting the velocity and the force feedback

6.3 Experimental Results

The performance and stability of the Time Domain Passivity Control(TDPC) have been tested and some experimental results are presented here. These curves have been plotted while considering 10% drop in data without time delay. Only one port has been considered as an active port i.e slave side. Open source software called Clumsy has been used to manipulate data. Clumsy enable users to produce alteration in data which usually data suffers while going through the Internet. The Fig. 6.6, shows the master linear velocity. The Fig. 6.7, shows the slave linear velocity. There is a certain surge in the slave velocity as compared to the master velocity at certain intervals. The Fig. 6.8, shows the energy of the system with blue line representing the energy on the master side, the light green line is representing the energy on the slave side and the red line is representing the net energy. The value of the net energy is zero in the beginning, but it is becoming more and more negative with the passage of time because of the accumulation of a surplus energy. These plots have been drawn without Passivity Controller. On closely examining these plots, it is clear that after two seconds the system becomes active and the surplus energy is increasing with the passage of time making the system unstable. The slave velocity is having certain surges as compared to the input master velocity.

The Fig. 6.10, shows the linear velocity of the master device and the Fig. 6.11, shows the slave linear velocity. These plots have been drawn after turning on the Passivity Controller. The robot was driven with a constant linear velocity as it is vivid from the velocity plots. The Fig. 6.12, shows the master energy, the slave energy, and the net energy. Passivity Observer indicated active behavior

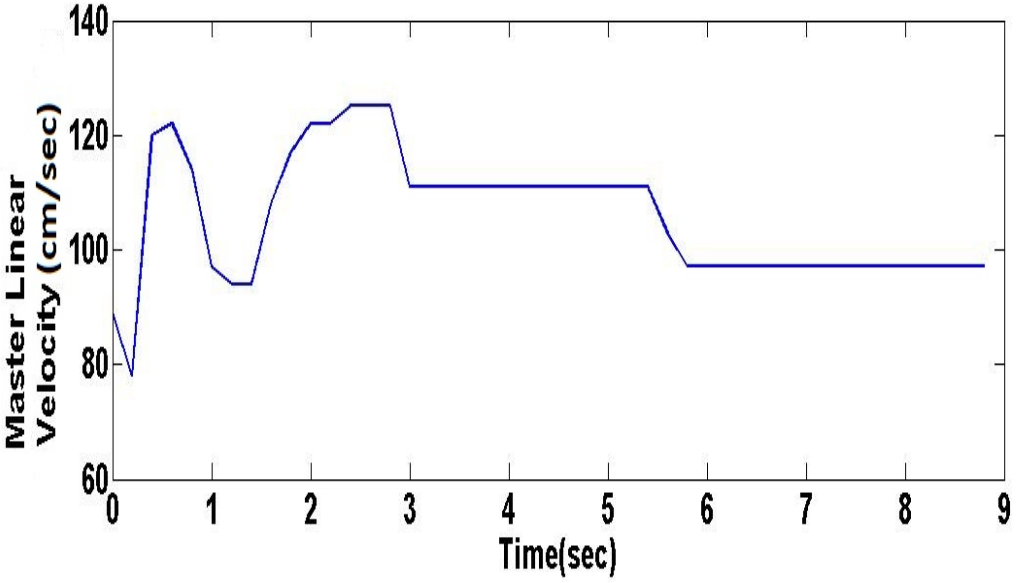

FIGURE 6.6: The master linear velocity

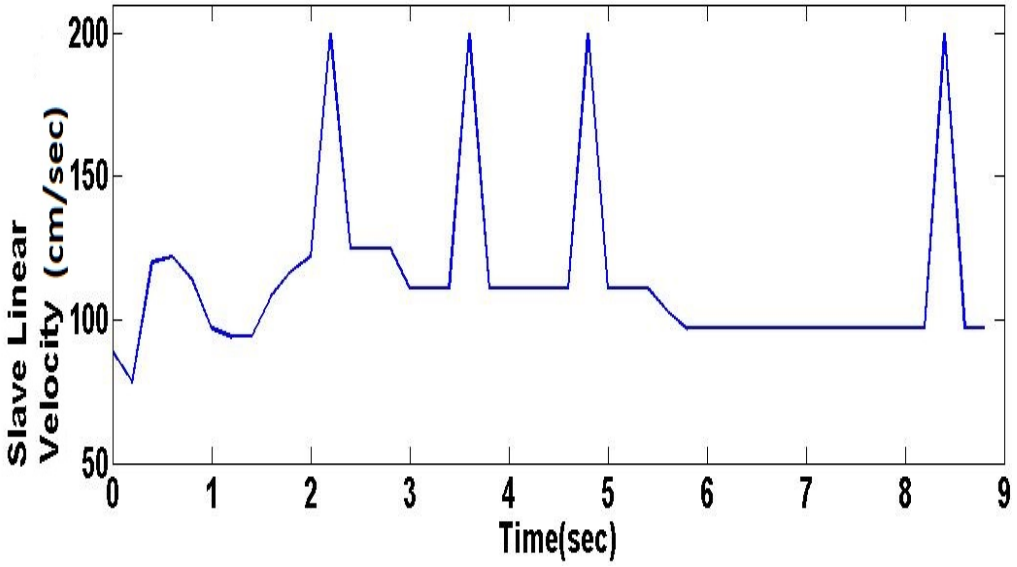

FIGURE 6.7: The slave linear velocity

twice i.e after first and fourth second. Passivity Controller dissipated the extra energy to make the system passive and net energy became zero again as compared to the accumulation of the energy in the Fig. 6.8, without controller. The Fig. 6.14, shows the slave controller calculated the suitable control action at the same specific moments and dissipates the extra energy to keep the system passive and stable at all time.

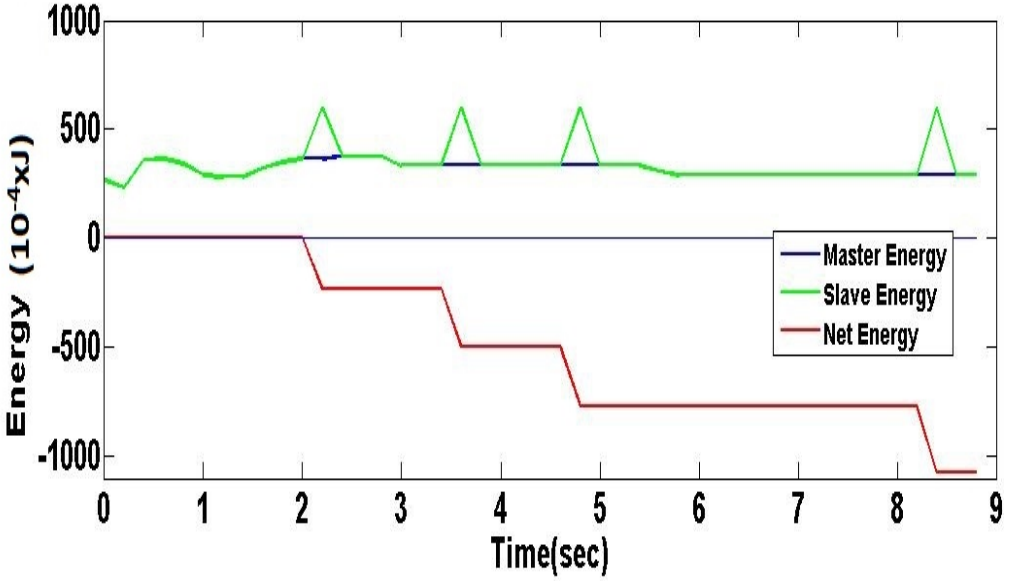

FIGURE 6.8: The master energy, the slave energy and the net energy

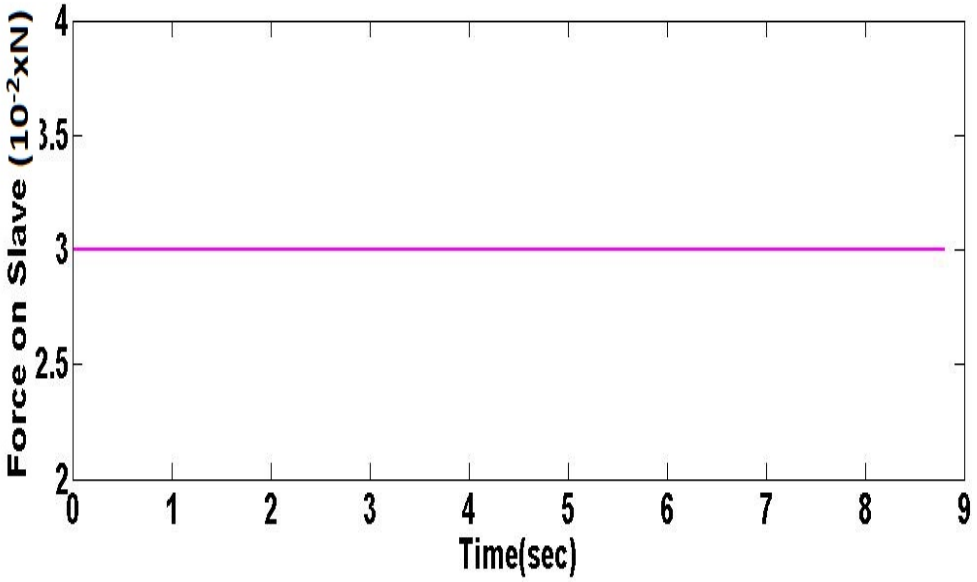

FIGURE 6.9: The force acting on the slave robot

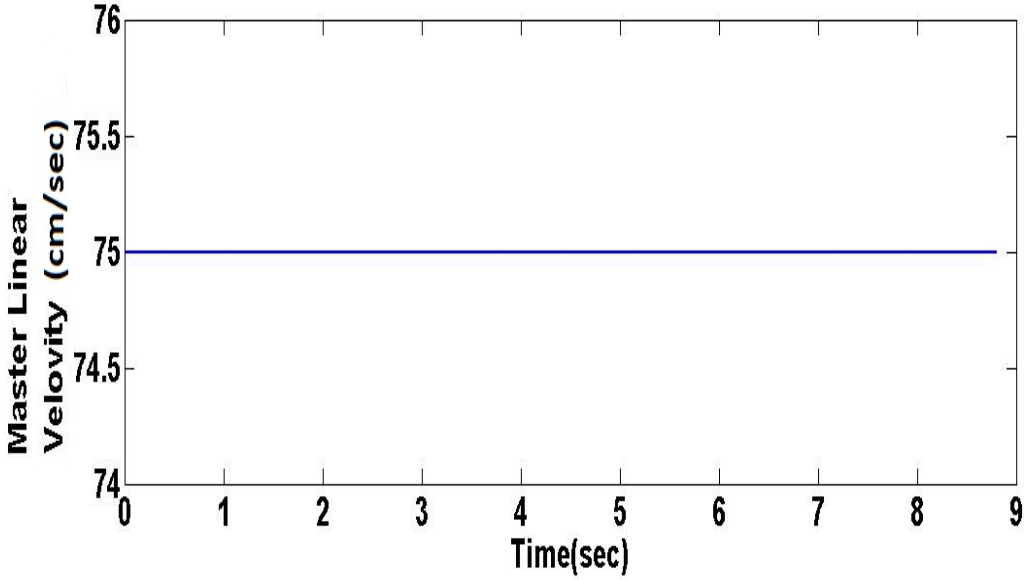

FIGURE 6.10: The master linear velocity

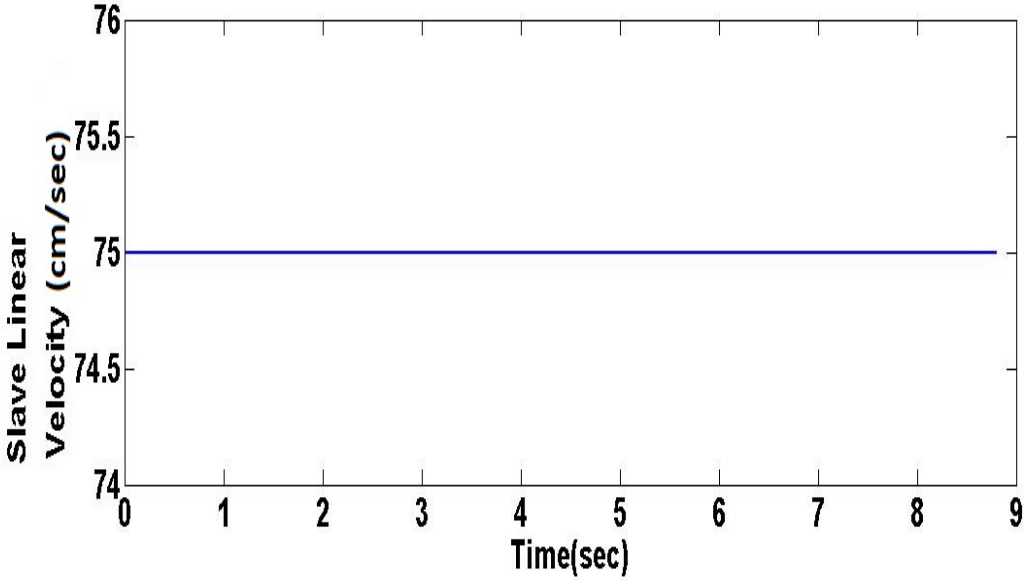

FIGURE 6.11: The slave linear velocity

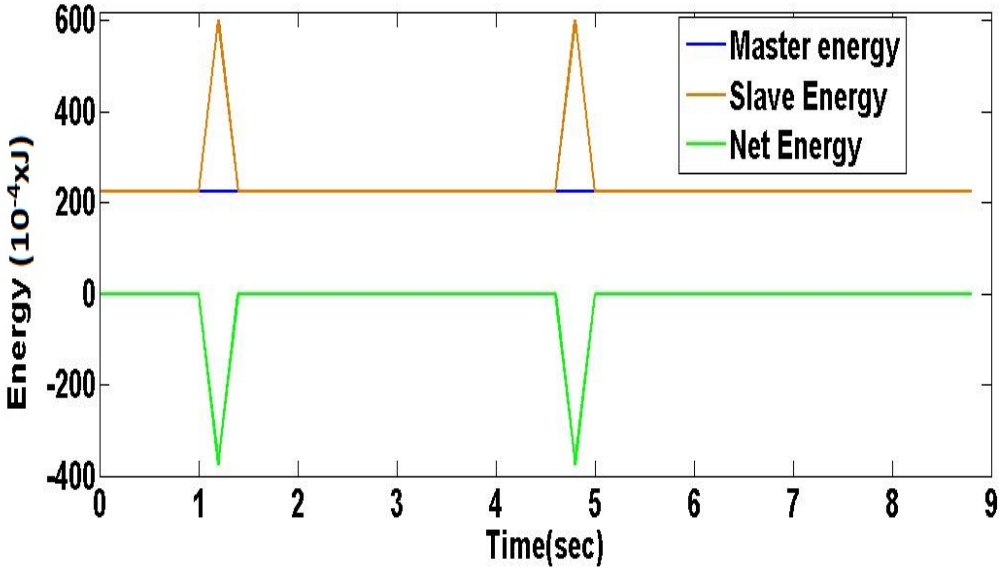

FIGURE 6.12: The master energy, the slave energy and the net energy

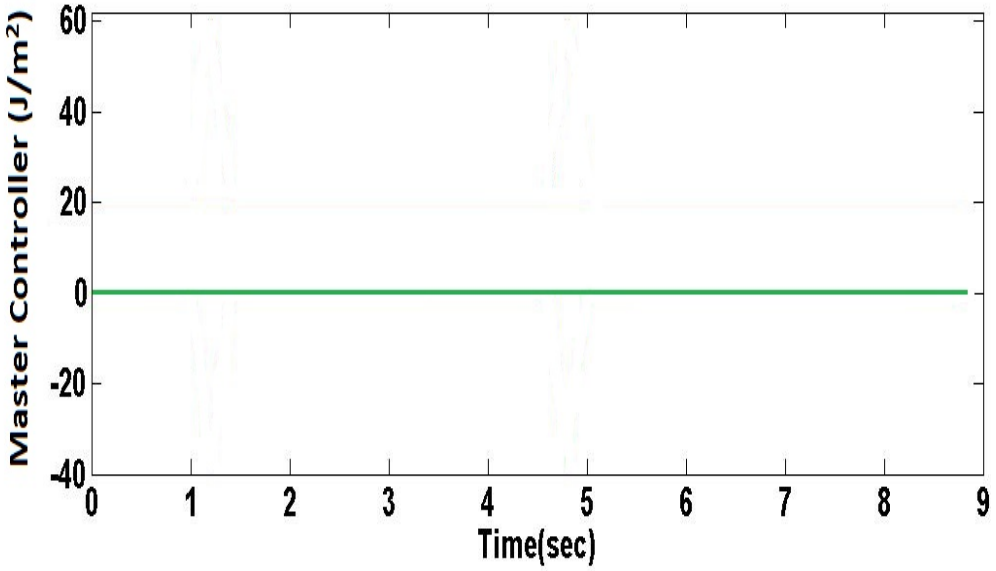

FIGURE 6.13: α_M for energy dissipation on master side

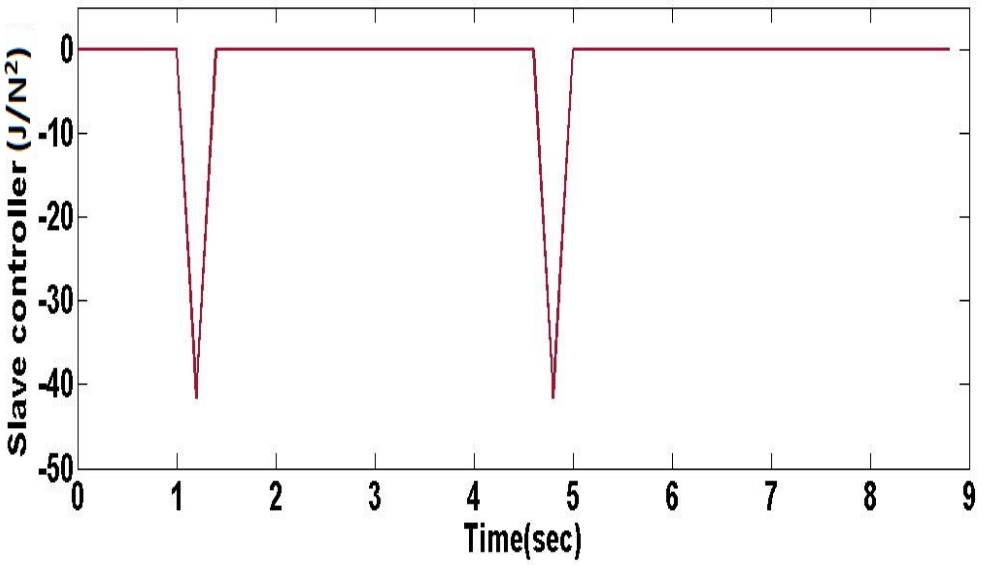

FIGURE 6.14: α_S for energy dissipation on slave side

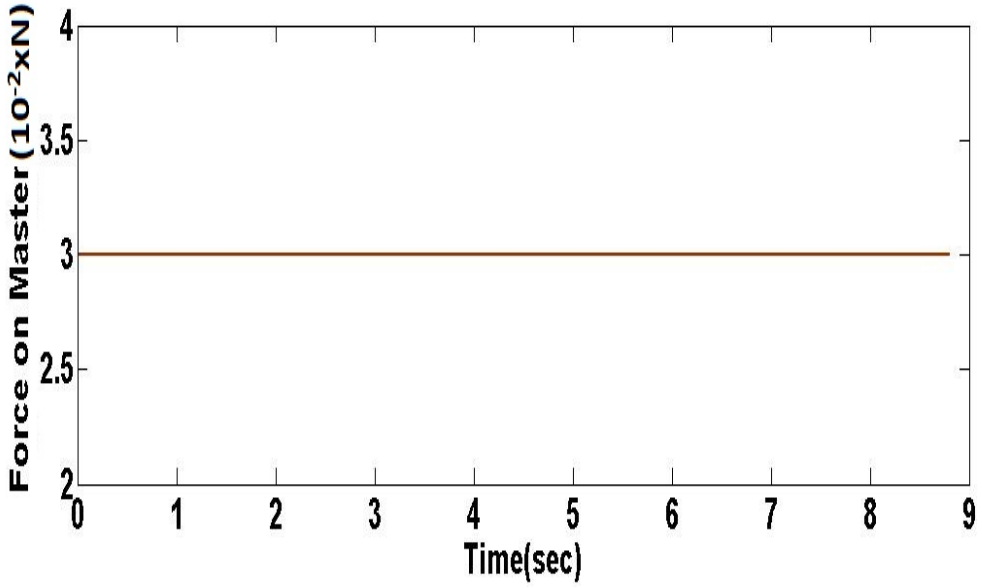

FIGURE 6.15: The force acting on the master device

Chapter 7

Time Delay Power Network

Most approaches used for the delayed teleoperation systems turned out to be conservative techniques that proved to be detrimental to the transparency and usability of a teleoperation system. Bilateral control often has elements that dissipate more energy introduced by the delayed communications channel than is strictly required to keep the control loop passive. Wave variable-based approach presents a lossless characteristic after applying the wave transformation, but damping elements are needed to minimize the wave reflection and to achieve impedance matching between the master robot and the slave robot[115].

The method proposed in the Chapter. 6, to teleoperate the mobile robot is based on the Time Domain Passivity Control (TDPC). The design has been performed by considering the distortion in data like packet loss, throttling, duplication, out of order and tempering in data without the considering network delay. Time Domain Passivity Control works fine without delay because the input and output of the system are power conjugated. This implies that each input is related to an output and their product is power. But, when there is a delay in the network, then it is not possible to calculate the net energy in real-time in the Passivity Observer. To cope with this issue the Time Delay Power Network (TDPN) approach has been used. It was presented to establish coordination between two fixed mechanisms in a simulated environment[116]. TDPN approach has been modified for the real mobile robot teleoperation with a network delay. Time Delay Power Network represents a Two-port network that has an inherent delay with the pair of conjugated

variables at each port. This approach is useful in modeling the input and the output through the communication medium with a delay as an ideal flow and effort source respectively.

7.1 Passivity of TDPN

A communication channel like the Internet has a delay and other impediments which are the main source of activity in a passive teleoperation while exchanging the velocity and the force. It has been molded as a Two-port network where velocity and force travel in the opposite directions with time delay as shown in the Fig. 7.1. Each port represents a real-time signal and its conjugate delayed signal. The product of these two quantities is the power. This representation of the network is called Time Delay Power Network (TDPN). TDPN behaves in the similar way as the normal communication network. But it helps in solving the issue of time delay in the communication channel. Instead of exchanging the velocity and the force, energy is being exchanged at both ports. The energy is divided into positive and negative energy at each port. The energy entering the port is taken as positive and the energy leaving the port is taken as negative. The analysis of the energy at the output of each port is done with reference to an input energy. Passivity Observers at the master and the slave side is installed to monitor the net energy at each port. When any Observer indicates the active behavior, then the corresponding Passivity Controller calculates the necessary action to dissipate the surplus energy to keep the system passive and stable[116].

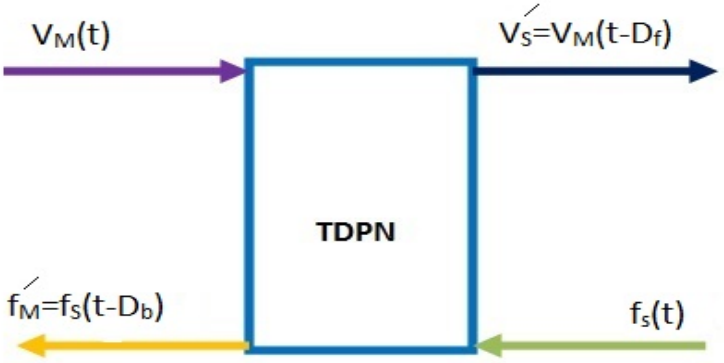

FIGURE 7.1: Time Delay Power Network (TDPN)

The net power of the whole system can be written as given in the following Eq. 7.1.

$$P_N(t) = P_M(t) + P_S(t) \tag{7.1}$$

The power on the left side of the communication channel is $P_M(t)$ and the power on the right of the communication channel is $P_S(t)$. The corresponding energy on the master side and the slave side can be written as given in the Eq. 7.2 and Eq. 7.3.

$$E_M(t) = \int_0^t P_M(\tau)d\tau \tag{7.2}$$

$$E_S(t) = \int_0^t P_S(\tau)d\tau \tag{7.3}$$

$$E_M(t) = \int_0^t f'_M(\tau)V_M(\tau)d\tau \tag{7.4}$$

$$E_S(t) = \int_0^t f_S(\tau)V'_S(\tau)d\tau \tag{7.5}$$

$E_M(t)$ is the energy on the master side in the Eq. 7.4, and $E_S(t)$ is the energy on the slave side in the Eq. 7.5. The network remains passive such that the condition in the Eq. 7.6, is satisfied.

$$P_N(t) \geq 0, \qquad \forall t \geq 0 \tag{7.6}$$

$E_M(t)$ and $E_S(t)$ are not available simultaneously. Therefore, to calculate the net energy of the system in the presence of a delay, the power at each port can be breached into positive and negative power as given in the Eq. 7.7-7.10.

$$P_M^+(t) = P_M(t), \quad \forall t \geq 0 \quad \& \quad f'_M(t)V_M(t) \geq 0 \tag{7.7}$$

$$P_M^-(t) = -P_M(t), \quad \forall t \geq 0 \quad \& \quad f'_M(t)V_M(t) \leq 0 \tag{7.8}$$

$$P_S^+(t) = P_S(t), \quad \forall t \geq 0 \quad \& \quad f_S(t)V'_S(t) \geq 0 \tag{7.9}$$

$$P_S^-(t) = -P_S(t), \quad \forall t \geq 0 \quad \& \quad f_S(t)V'_S(t) \leq 0 \tag{7.10}$$

The corresponding positive and negative energies of the master and the slave can be written as follows:

$$E_M^+(t) = \int_0^t P_M^+(\tau)d\tau \qquad \forall t \geq 0 \tag{7.11}$$

$$E_M^-(t) = \int_0^t P_M^-(\tau)d\tau \qquad \forall t \geq 0 \tag{7.12}$$

$$E_S^+(t) = \int_0^t P_S^+(\tau)d\tau \qquad \forall t \geq 0 \tag{7.13}$$

$$E_S^-(t) = \int_0^t P_S^-(\tau)d\tau \qquad \forall t \geq 0 \tag{7.14}$$

$E_M^{in}(t)$ is the entering energy in the port on the master side and is taken as positive as given in the Eq. 7.15, and the $E_M^{out}(t)$ is the energy coming out of the port on the master side and is taken as negative as given in the Eq. 7.16. Similarly, the energy entering and leaving the port on the slave side is also taken as positive and negative respectively, as given in the Eq. 7.17, and Eq. 7.18.

$$E_M^{in}(t) = E_M^+(t) \qquad \forall t \geq 0 \tag{7.15}$$

$$E_M^{out}(t) = E_M^-(t) \qquad \forall t \geq 0 \tag{7.16}$$

$$E_S^{in}(t) = E_S^+(t) \qquad \forall t \geq 0 \tag{7.17}$$

$$E_S^{out}(t) = E_S^-(t) \qquad \forall t \geq 0 \tag{7.18}$$

The net energy of the network can be written as given in the Eq. 7.19 and it can be modified in term of entering and leaving energies as given in the Eq. 7.20.

$$E_N(t) = E_M(t) + E_S(t) \tag{7.19}$$

$$E_N(t) = E_M^{in}(t) - E_M^{out}(t) + E_S^{in}(t) - E_S^{out}(t) \tag{7.20}$$

Here $E_{M\to S}(t)$ is the net energy from master to slave as described in the Eq. 7.22, and $E_{S\to M}(t)$ is the net energy from slave to master as given in the Eq. 7.23.

$$E_N(t) = E_{M\to S}(t) + E_{S\to M}(t) \tag{7.21}$$

$$E_{M\to S}(t) = E_M^{in}(t) - E_S^{out}(t) \tag{7.22}$$

$$E_{S\to M}(t) = E_S^{in}(t) - E_M^{out}(t) \tag{7.23}$$

The network remains passive until and unless the expressions in the Eq. 7.24, and Eq. 7.25, are true.

$$E_{M\to S}(t) \geq 0 \tag{7.24}$$

$$E_{S\to M}(t) \geq 0 \tag{7.25}$$

The net energy from the master to the slave and from the slave to the master with a delay can be written as given in the Eq. 7.26, and Eq. 7.27, respectively. D_f stands for the forward delay and D_b represents the backwards delay.

$$E_{M\to S}(t) = E_M^{in}(t - D_f) - E_S^{out}(t) \tag{7.26}$$

$$E_{S\to M}(t) = E_S^{in}(t - D_b) - E_M^{out}(t) \tag{7.27}$$

7.1.1 Passivity Observer

There are two observers to monitor the net energy on both sides as given in the Eq. 7.28, and Eq. 7.29. $E_M^{obs}(n)$ is the observer on the master side to monitor the active energy and E_S^{obs} is the observer on the slave side. These observers calculate the surplus energy and indicate the behavior of the system[116].

$$E_M^{obs}(n) = E_M^{obs}(n-1) + E_M^{in}(n - D_f) - E_S^{out}(n) + \alpha_M(n-1)V_M(n-1)^2 \tag{7.28}$$

$$E_S^{obs}(n) = E_S^{obs}(n-1) + E_S^{in}(n - D_b) - E_M^{out}(n) + \alpha_S(n-1)f_S(n-1)^2 \tag{7.29}$$

7.1.2 Passivity Controller

In order to ensure the stability of teleoperation system, it is mandatory to dissipate the surplus energy which is induced in the system due to an active behavior of the network. The designed passivity controller acts as a dissipative element at each output port of the TDPN. This dissipation should be enough to dilute the effect of the active energy to zero. The following expressions calculate the master and the slave controllers when the values of the observers are negative. α_M as given in the Eq. 7.30, is a master controller on the master side and α_S as given in the Eq. 7.31, is a slave controller on the slave side. These two make the system passive whenever there is an active behavior.

$$\alpha_M(n) = \begin{cases} 0 & \text{if } E_M^{obs} \geq 0; \\ -\frac{E_M^{obs}(n)}{V_M(n)^2} & \text{if } E_M^{obs} < 0. \end{cases} \tag{7.30}$$

$$\alpha_S(n) = \begin{cases} 0 & \text{if } E_S^{obs} \geq 0; \\ -\frac{E_S^{obs}(n)}{f_S(n)^2} & \text{if } E_S^{obs} < 0. \end{cases} \tag{7.31}$$

$$V_S(n) = V'_S(n) + \alpha_S(n) f_S(n) \tag{7.32}$$

$$f_M(n) = f'_M(n) + \alpha_M(n) V_M(n) \tag{7.33}$$

The corrective action is applied to the velocity at the slave side as given in the Eq. 7.32, and to the force feedback on the master side as described in the Eq. 7.33, respectively, and also shown in Fig. 7.2. It is an extension of the Fig. 6.5 , with forward and backward delay in the control loop.

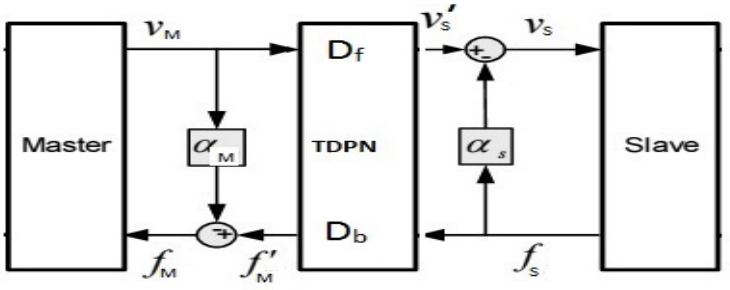

FIGURE 7.2: The master and slave controller adjusting force and velocity

7.2 Problem Description

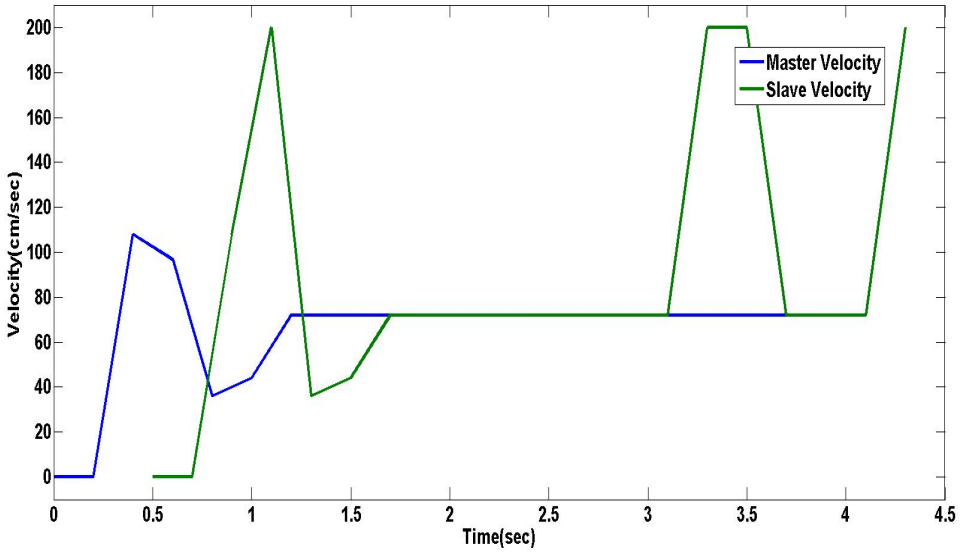

FIGURE 7.3: The master and the slave linear velocity

In order to show the behavior of the system and to illustrate the problem vividly, different plots are presented in this section. The time delay has been taken as

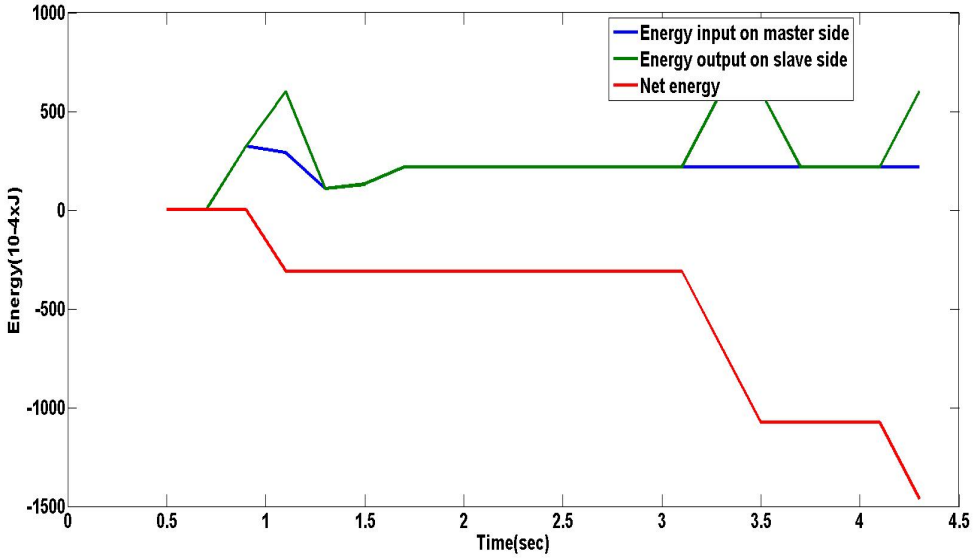

FIGURE 7.4: Input, output and net energy from the master side to the slave side

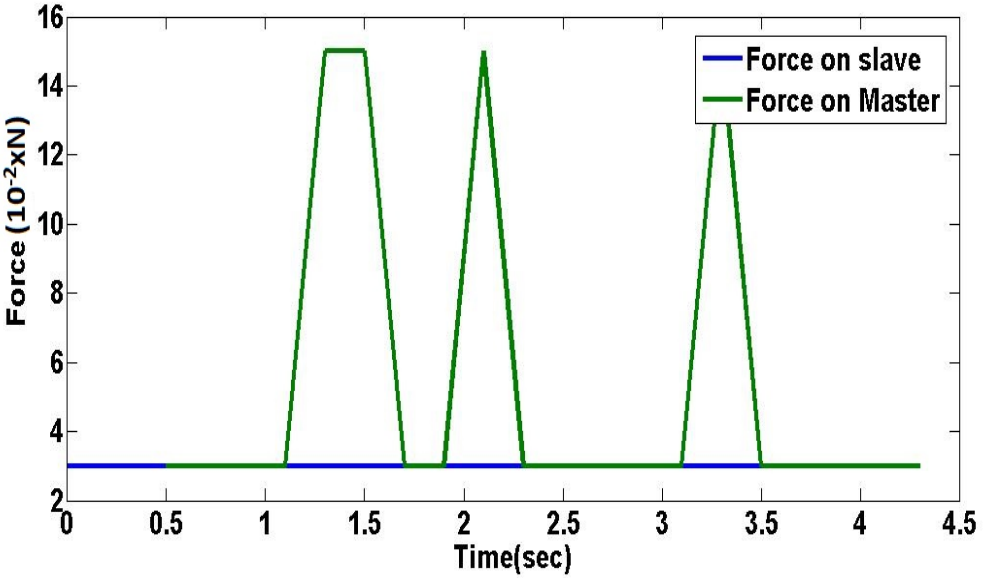

FIGURE 7.5: The force acting on the master and the slave robot

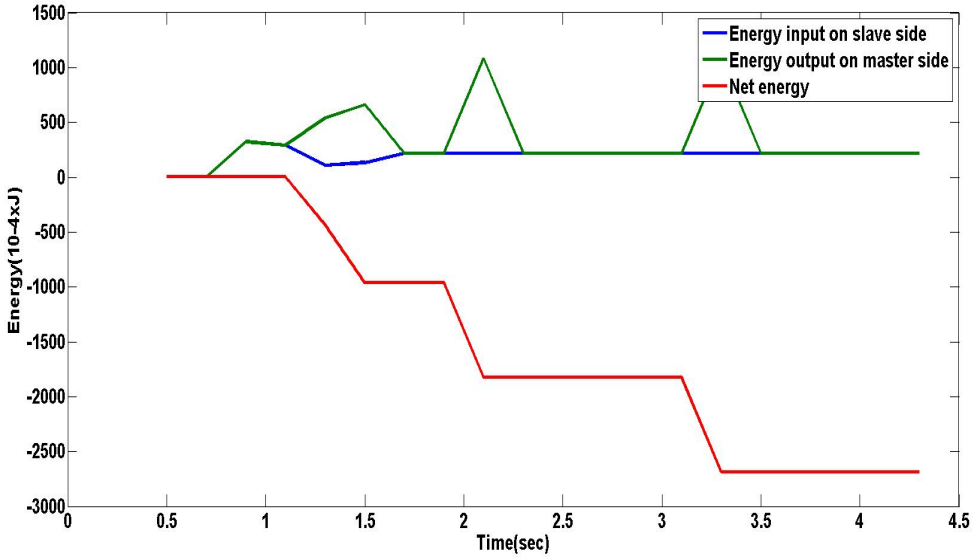

FIGURE 7.6: Input, output and net energy from the slave side to the master side

500ms in both directions. Therefore, the round trip time delay is one second. The network behaves similar to the Internet, the only difference is that it has fix communication delay. The Fig. 7.3, shows the master velocity with the blue line and the slave velocity with the green line. It can be seen that after passing through the network the slave velocity is much higher than the desired master velocity at certain time intervals. It is vivid from the green line that the slave velocity is higher than the master before first second for nearly one second. Furthermore, there is a surge in slave velocity after 3 seconds for nearly one second too. The Fig. 7.4, shows the input energy on the master side and the output energy on the slave side. It is obvious that the output energy is more than the input energy at certain intervals and surge in the output energy is obvious from the plot. This is due to an active behavior of the network, which induces surplus energy in the system and makes it unstable. The force acting on the slave robot and the force feedback acting on the master haptic device have been plotted with blue and green line respectively, as shown in the Fig. 7.5. The Fig. 7.6, shows the energy input on the slave side and the output energy on the master side. It is clear after closely examining these two plots that whenever system behavior is active then, there is a surge in the output energy and the force feedback. Clearly, the feedback force is not similar to the slave force and an active behavior can be seen after one second as shown in the Fig. 7.5. This behavior is being repeated multiple times leading to instability. The corresponding output energy on the master side is also higher

than the input energy on the slave side.

7.3 Experimental Results with Fix Delay

The following plots have been drawn to reveal the performance of the designed controller. AutoMerlin mobile robot has been used to test the controller. The Fig. 7.7, exhibits the master velocity with a blue line and the slave velocity with a green line. The Fig. 7.8, represents the input energy on the master side with a blue line and the output energy on the slave side with a light green line and the net energy with a red line. Input energy has a delay of 500ms. It has been plotted with an offset to show the symmetry between input, output and net energy. The Fig. 7.9, shows the slave controller with a dark green line. The output energy is slightly more than input energy around one second and after 3 seconds and the net energy is negative as shown in the Fig. 7.8. Therefore, the control action to dissipate this surplus energy has been taken by the slave controller as shown in the Fig. 7.9. The purpose of the slave controller is to dissipate the active energy. An active behavior has been observed twice. The slave controller has dissipated the surplus energy to keep the system passive. Due to a timely dissipation of the active energy, the slave follows the master velocity commands as shown in the Fig. 7.7. The Fig. 7.10, presents the force acting on the slave and the force feedback acting on the master haptic device. The Fig. 7.11, shows the input energy on the slave side, output energy on the master side and the net energy. By taking a close look at these energies, it can be seen that the output energy is more than the input energy after 2 seconds. The surplus energy has been added by the network and this energy has been dissipated by the master controller as shown in the Fig. 7.12, to avoid any surge in the force feedback at the active intervals. It is obvious from these plots that there is a surplus energy and it is making the system active and unstable, but the master passivity controller is keeping the system passive and stable by dissipating the active energy.

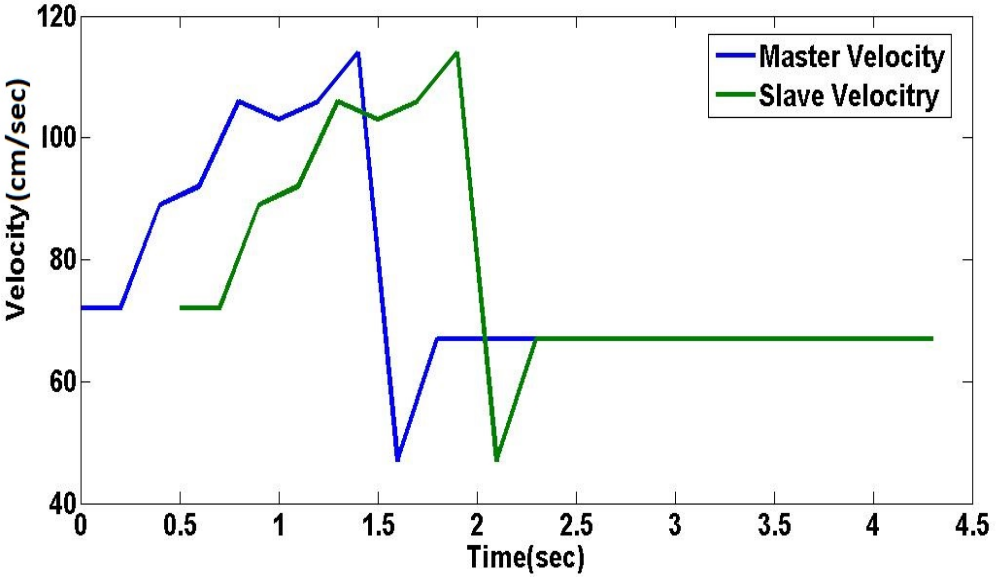

FIGURE 7.7: The master and the slave linear velocity

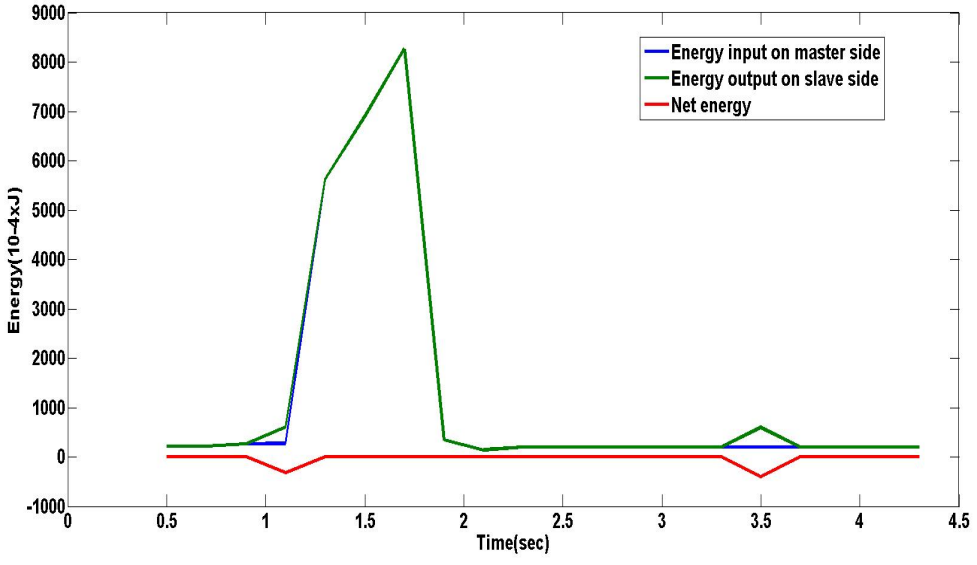

FIGURE 7.8: Input, output and net energy from the master side to the slave side

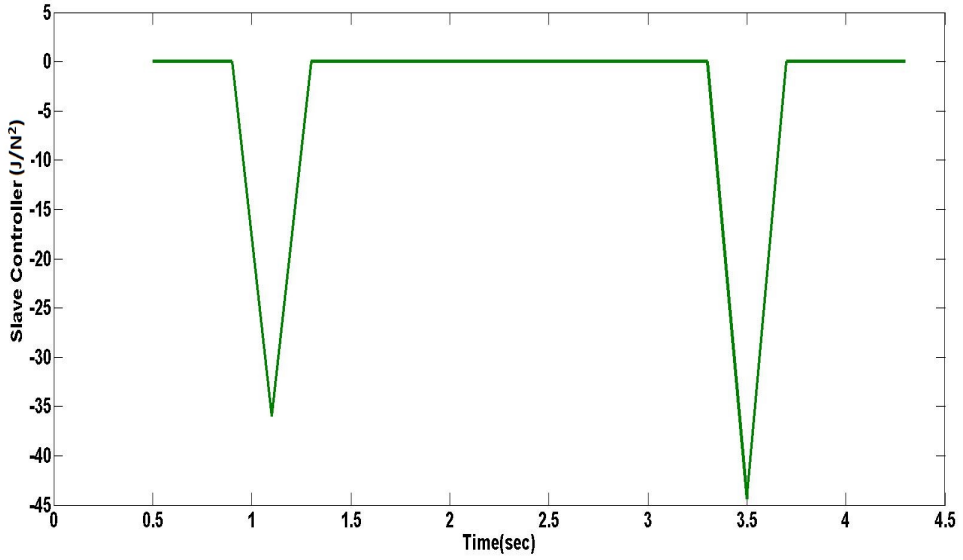

FIGURE 7.9: α_S for energy dissipation on slave side with constant delay

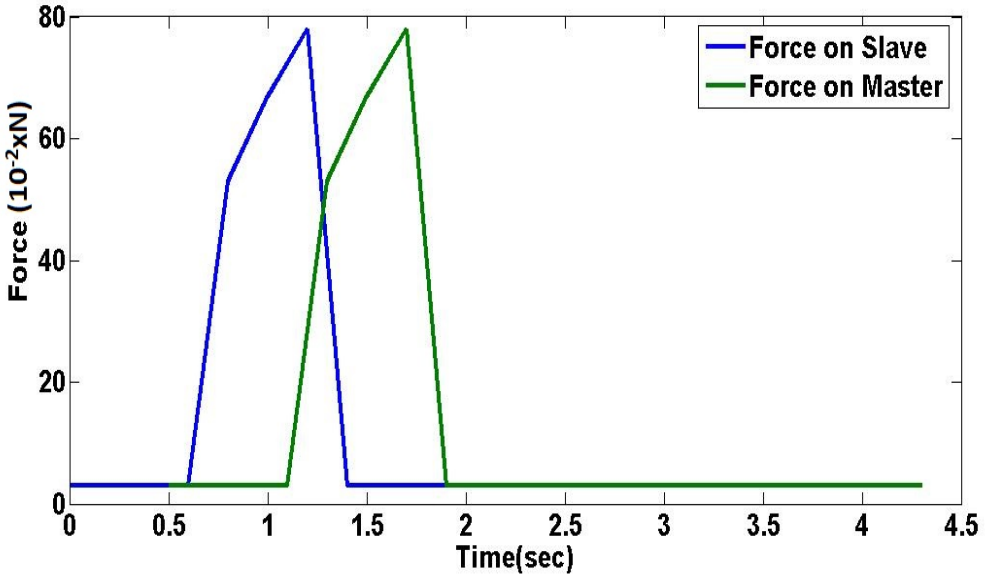

FIGURE 7.10: The force acting on the master and the slave robot

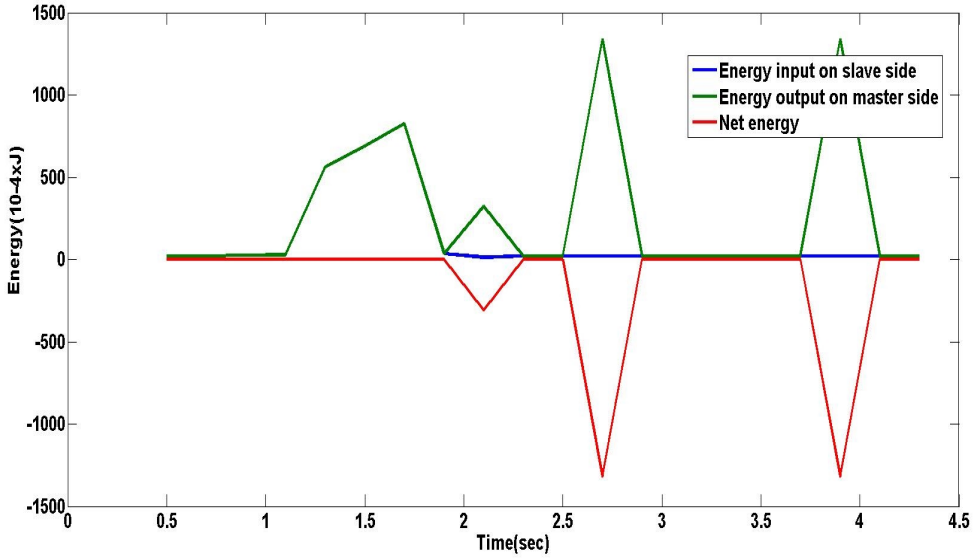

FIGURE 7.11: Input, output and net energy from the slave side to the master side

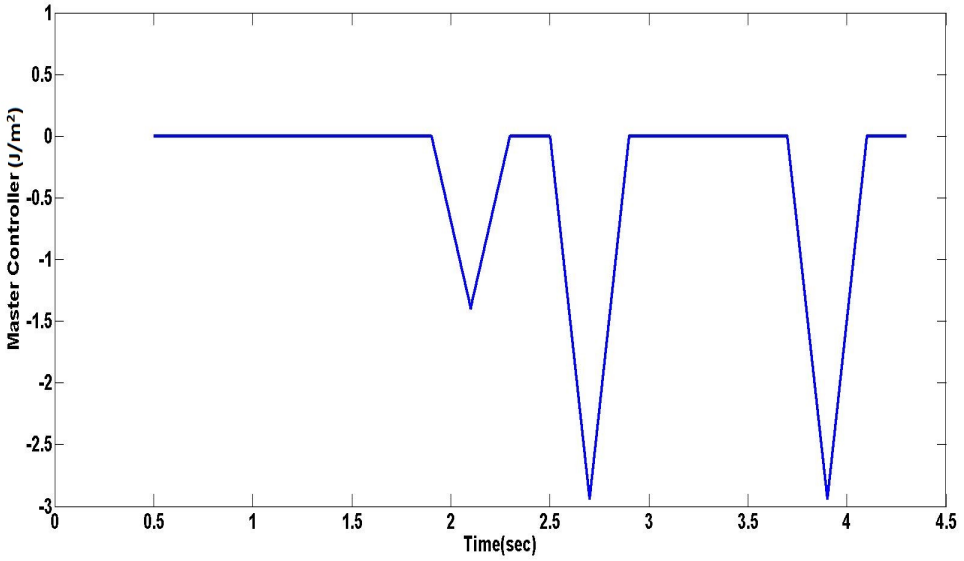

FIGURE 7.12: α_M for energy dissipation on master side with constant delay

7.4 Experimental Results with Stochastic Delay

The behavior of the system has been tested with a random delay. This delay has been generated by using the opens source software called Clumsy. It can produce delay as well as other limitations of the network like drop in data, duplication or out of order arrival of data. Twenty percent drop in data with a random delay of upto 700ms have been used for the experimentation. First, the performance has been evaluated without the obstacles. The Fig. 7.13, shows the coordination between the master and the slave velocity and the slave velocity has some offset due to the time delay in the communication channel. The Fig. 7.16, shows the force acting on the slave and the force feedback on the master. The Fig. 7.14, provides the comparison of the input energy on the master side and the output energy on the slave side and the net energy from the master to the slave. The output energy is more than the input energy as indicated by the green line. The slave controller monitors the surplus energy and dissipates the extra energy to keep the forward communication passive as indicated by the blue line in the Fig. 7.15. The backward communication from the slave to the master carries force feedback. The Fig. 7.17, exhibits the input energy on the slave side and the output energy on the master side and the net energy from the slave to the master. It is vivid from the green line that there is a surge in the energy at certain intervals due to an active behavior of the network. As the force on the slave and the force feedback are similar therefore the energy observed as an active energy has been dissipated by the master controller as depicted with the blue line in the Fig. 7.18. Whenever the network has added the surplus energy in the system the master and the slave controllers have calculated the necessary action and have dissipated the redundant energy to keep the system passive and stable.

These plots have been plotted while teleoperating the robot with the obstacles in the remote environment. The Fig. 7.19, shows the coordination between the velocities. Fig. 7.20, presents the track of energies monitored by the Passivity Observer. Fig. 7.21, is showing the control action taken by slave controller α_S to dissipate active energy. Fig. 7.22, shows the force on the robot and the feedback force. After eight seconds the robot encounters the first obstacle and after seventeen seconds there is another obstacle. The rise in force can be seen in the Fig. 7.22. Fig. 7.23, shows the input energy on slave side, output energy on master side and net energy. Fig. 7.24, is illustrating the master controller. It is vivid from the plot that the master controller α_M is dissipating the active energy. These plots

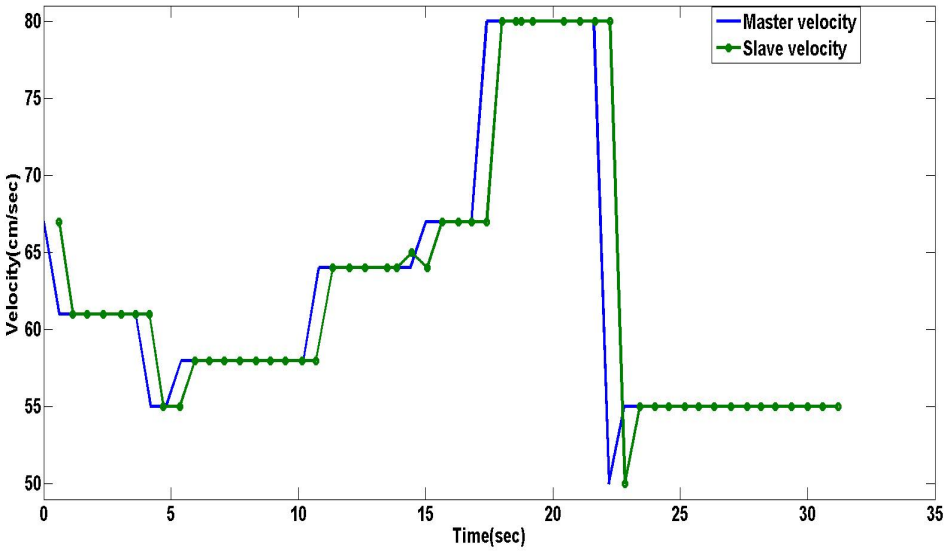

FIGURE 7.13: The master and the slave linear velocity without obstacles

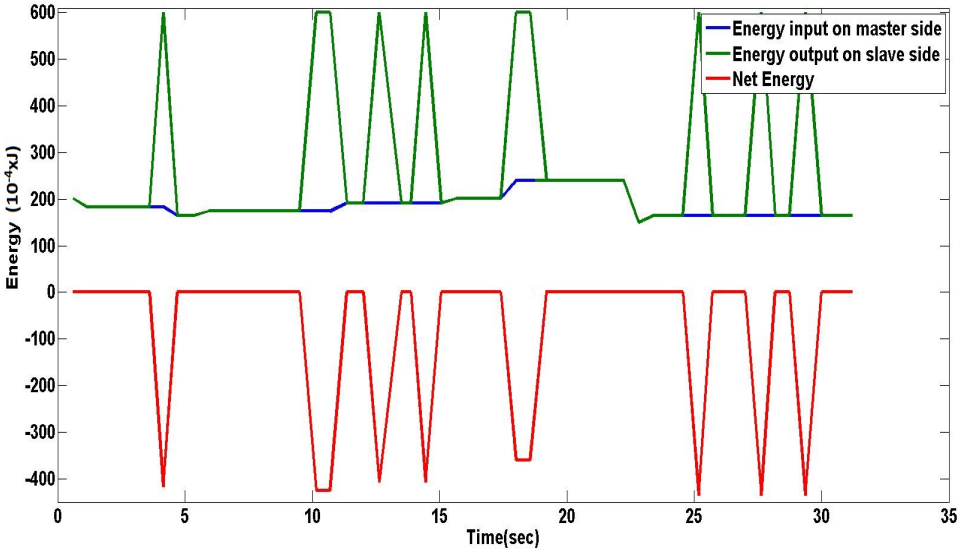

FIGURE 7.14: The input, output and net energy from the master side to the slave side without obstacles

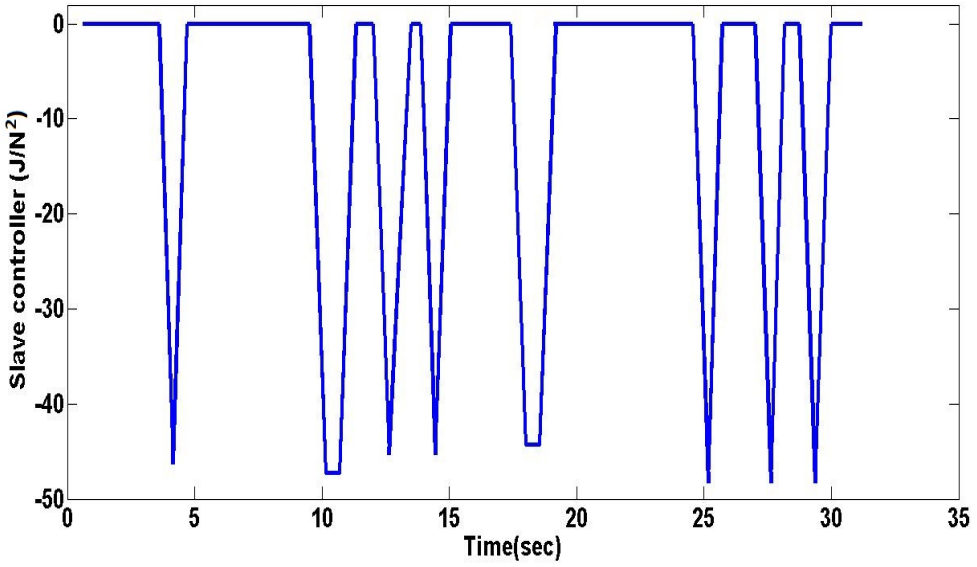

FIGURE 7.15: α_S for energy dissipation on slave side delay without obstacles

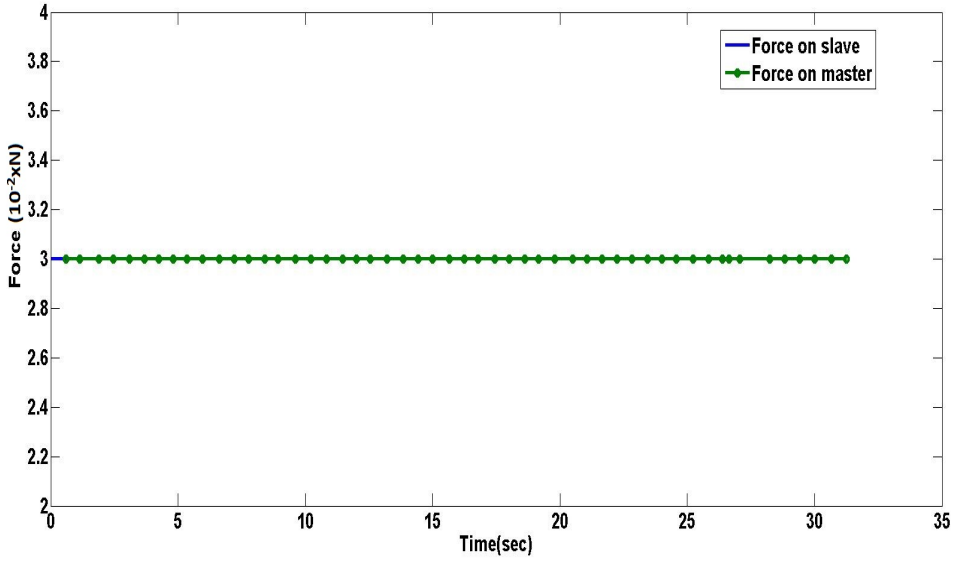

FIGURE 7.16: The force acting on the master and the slave without obstacles

show that the controller perform equally good in the presence of the obstacles too.

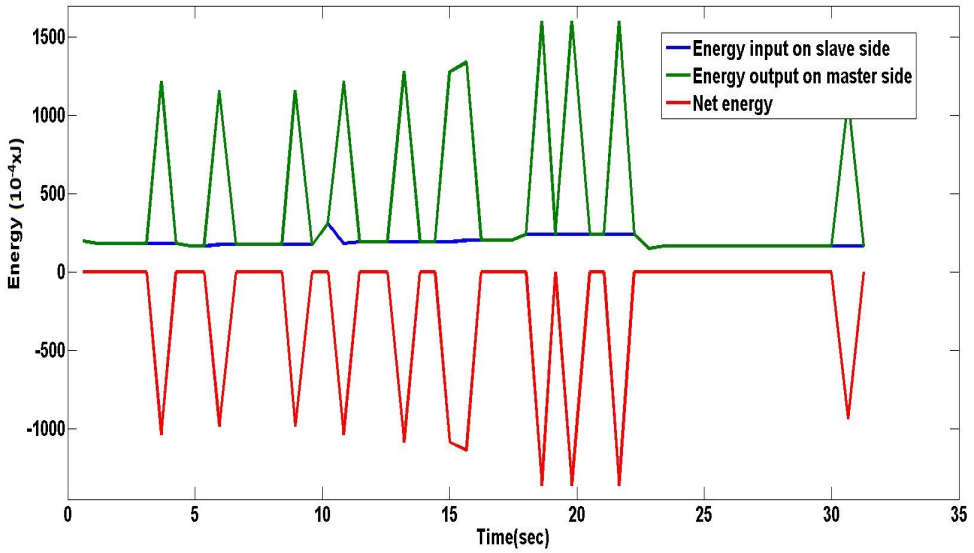

FIGURE 7.17: The input, output and net energy from the slave side to the master side without obstacles

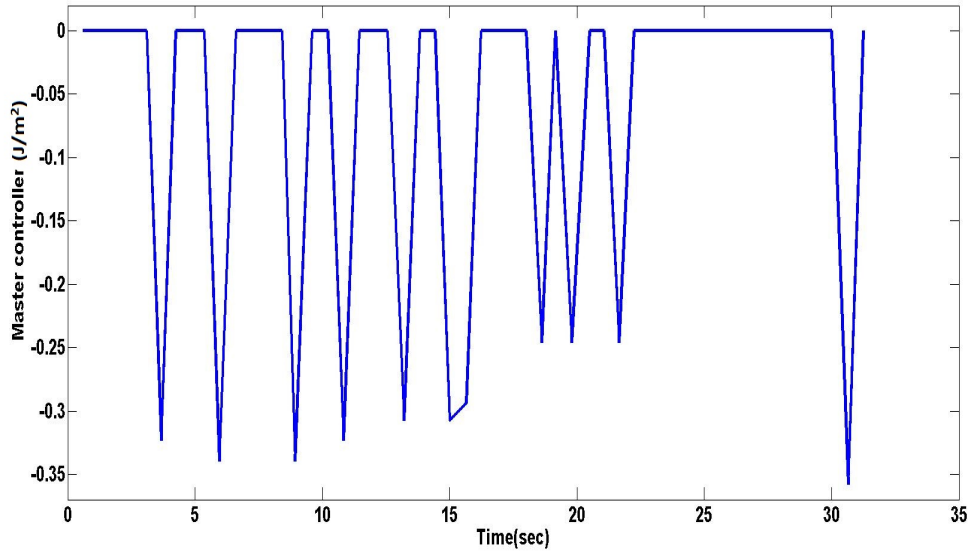

FIGURE 7.18: α_M for energy dissipation on master side delay without obstacles

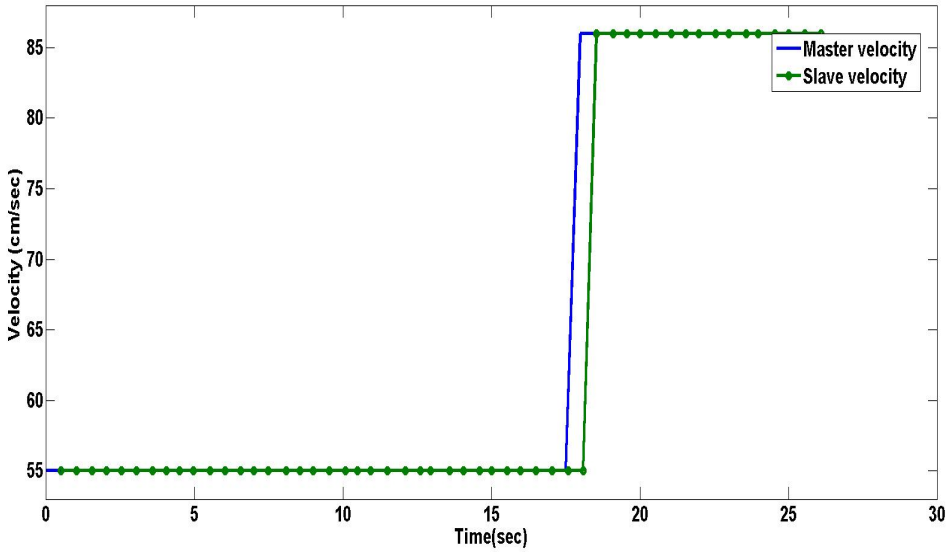

FIGURE 7.19: The master and the slave linear velocity with obstacles

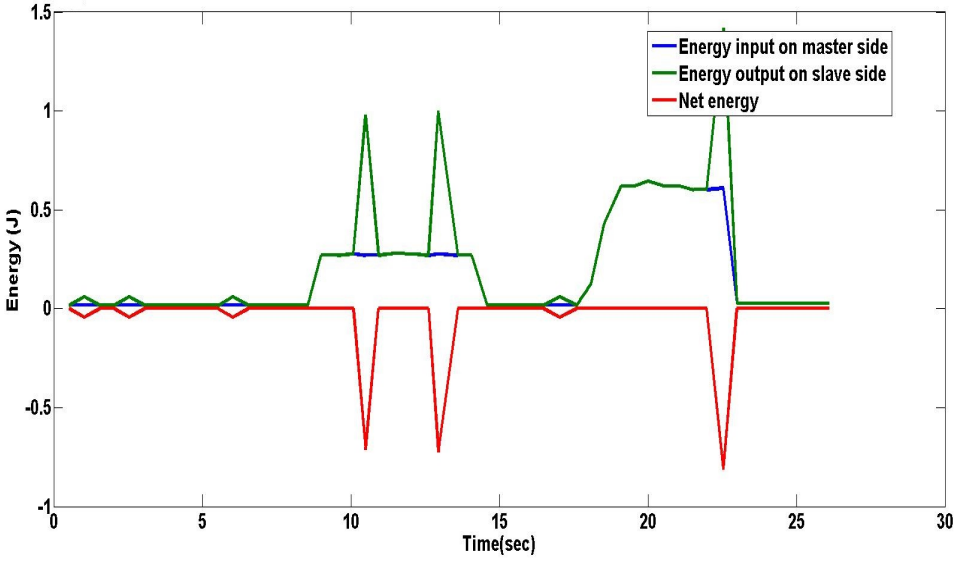

FIGURE 7.20: The input, output and net energy from the master side to the slave side with obstacles

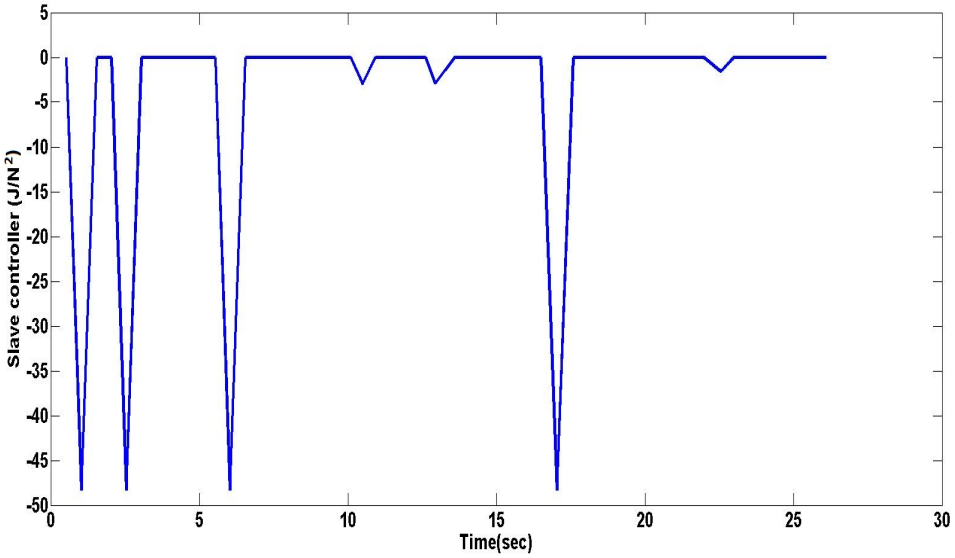

FIGURE 7.21: α_S for energy dissipation on slave side with obstacles

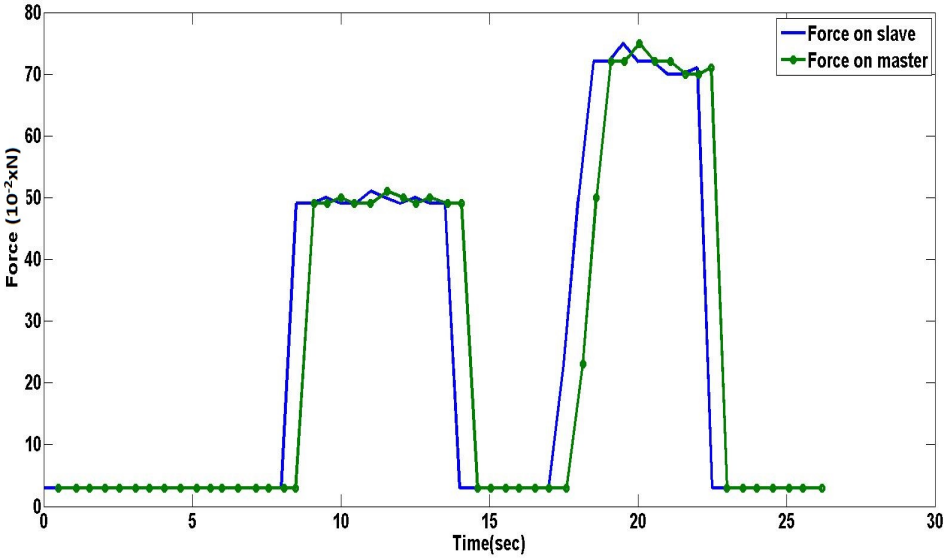

FIGURE 7.22: The force acting on the master and the slave with obstacles

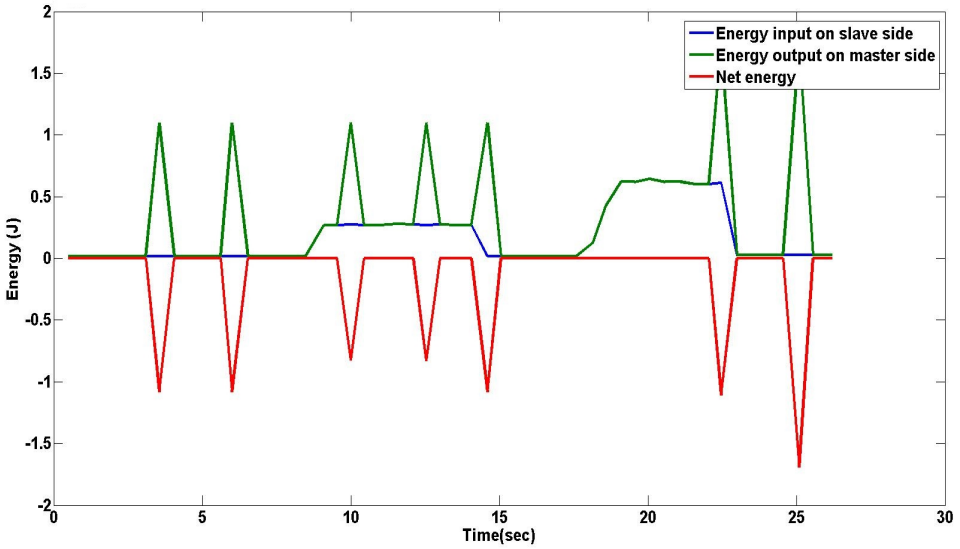

FIGURE 7.23: The input, output and net energy from the slave side to the master side with obstacles

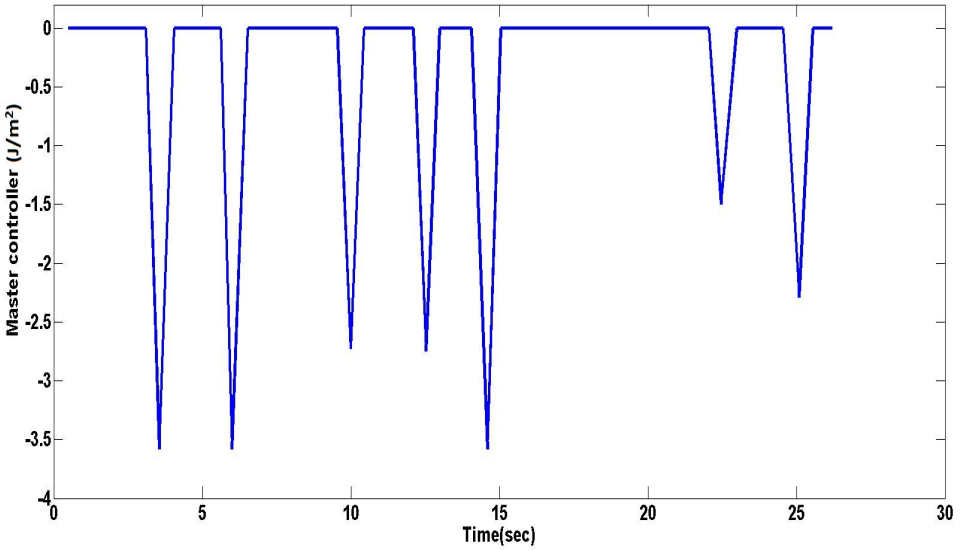

FIGURE 7.24: α_M for energy dissipation on master side with obstacles

Chapter 8

Path Planning and Motion Coordination of Multiple Robots

The development of the mobile robots which coordinate with each other has been dramatically increased because it is an appropriate solution in relation to the performance, efficiency, and reliability. The development of such cooperation is one of the most demanding goals in the artificial intelligence and robotics research. The first study about the multi-robot systems was done under the project called the *Cellular Robotics System*. It was a decentralized and hierarchical architecture[117, 118]. The swarm robot system was composed of a large number of homogeneous robots aimed to represent a group of self-organizing robots in the new patterns. In addition, the control strategy was based on the individual behavior where each robot could perceive other robots by using the proximity sensors[119]. The work on the coordination among multiple agents was inspired from the biological systems of motion coordination among the insects. The pioneer work initiated significant efforts in the study of the multi-robots formation and then, three different approaches towards the cooperative formation control of the mobile robots were evolved. The behavior-based approach suitable for the large flocks of the robots was one of them. Its aim was to assign several desired behaviors to each single robot such as the obstacle avoidance and target seeking. The leader-follower was another approach in which one robot among the flock was specified as the leader and the rest robots were the followers. The leader was driven on the predefined trajectories whilst the follower robots followed the leader according to a relative posture. The virtual structure was the third approach in

which the priority of the robots could not be changed. They had to either follow their individual trajectories or had to maintain the group formations[120].

Path planning is to find an optimal path and appropriate trajectory which makes the multi-robots team navigation automatic from the start point to the target point without any collision. The path planning based on the environmental information is usually classified into two main categories. The first category depends on the full environment model and the complete environmental information. The other is based on the mounted sensors because the mobile robots do not master the full environmental information. Several approaches have been investigated in this area which includes the map-based navigation approach, the fuzzy reasoning, the neural network approach, and the genetic algorithm approach. The fuzzy logic control FLC and type 2-FLC have been studied widely and have been implemented on the mobile robots due to their ability to control the mobile robot without knowing the mathematical model. However, both of them couldn't catch the stochastic uncertainty which cannot be predicted in advance or can't be recognized accurately during the measurement process. On the other hand, the probabilistic fuzzy logic system PFLS is different from the FLC since it uses the probabilistic fuzzy sets instead of the fuzzy sets in order to capture the information with the stochastic uncertainties. It can dramatically improve the performance of the measurement and reasoning for the mobile robots, especially in an unknown dynamic environment. It provides an alternative way for the robots to acquire data-driven human-like intelligence. As a result, PFLS are more valuable for processing the various uncertainties and the reactive navigation control of mobile robots[121, 122]. This work has been done in collaboration with Rami Al-Jarrah[123].

8.1 Kinematic of The Mobile Robot

Path planning is an essential strategy for the mobile robot navigation. The main problem is how to compute the path to move the formation from an initial point to a target point in an unknown environment. Therefore, in order to control the behaviors of the multiple mobile robots the kinematic analysis of each robot is required. The kinematics analysis of the differentially driven wheeled mobile robot in a two-dimensional plane has been considered in such a way that the mobile robot moves without slipping or skidding of wheels on a plane surface.

The modeling of the mobile robot has been done in the Cartesian Coordinates. Each robot has two fix standard wheels and a caster wheel and is differentially driven by skid steer motion. The two wheels are independently driven by the two motors for the motion and orientation. The Kinematics Model of the mobile robot has been driven according to the Fig. 8.1, and is given in the Eq. 8.1.

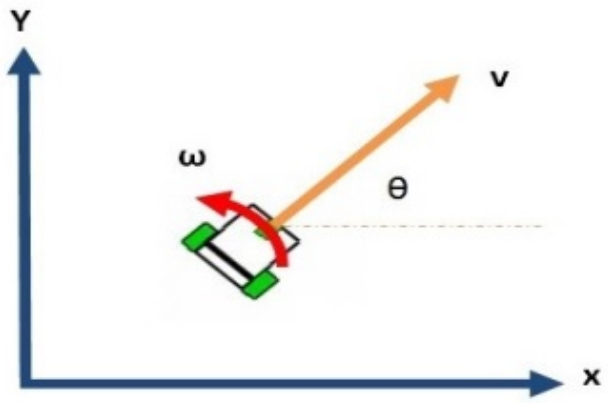

FIGURE 8.1: The mobile robot kinematics

$$\begin{bmatrix} \dot{x}(t) \\ \dot{y}(t) \\ \dot{\theta}(t) \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ w \end{bmatrix} \tag{8.1}$$

The linear velocity v can be calculated according to the relation given in the Eq. 8.2. Where v_r and v_l are the linear velocities of the right and left wheels respectively. Similarly, the angular velocity w is given in the Eq. 8.3. l is the distance between two rear wheels i.e. the width of the robot. v_r is the product of the angular velocity w_r of the right wheel and its radius as given in the Eq. 8.4. v_l is the product of the angular velocity w_l of the left wheel and its radius as described in the Eq. 8.5

$$v = \frac{v_r + v_l}{2} \tag{8.2}$$

$$w = \frac{v_r - v_l}{l} \tag{8.3}$$

$$v_r = rw_r \tag{8.4}$$

$$v_l = rw_l \tag{8.5}$$

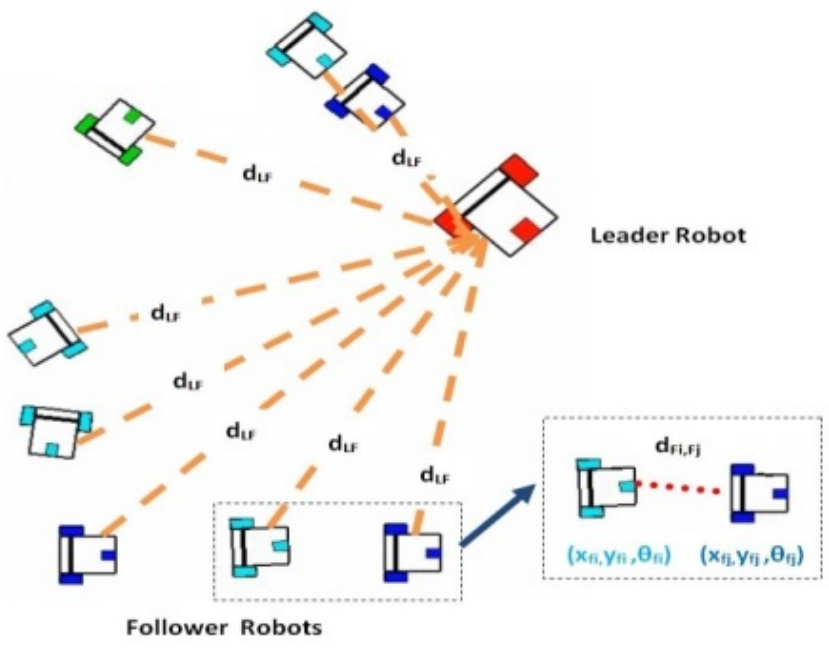

FIGURE 8.2: The multi-robot system

It has been assumed that the leader robot and the follower robots are given sensors(e.g., compass and encoder etc) that enable them to relate their own position to each other to execute the leader-follower model. Furthermore, it has also been assumed that the robots can find the distances from any moving obstacles including the robots themselves by using the ultrasonic sensors. The Fig. 8.2, shows the multi-robot system. The leader robot, as well as other agents can localize themselves and find distances between them and the probability of the existence of an obstacle around. The d_{LF} and $d_{F,F}$ are the distances between the follower and the leader as well as the distance between any two followers, respectively.

8.2 Probabilistic Fuzzy Logic System

The fuzzy logic is based on the theory of fuzzy set which is composed of discrete or continuous elements possessing a degree of membership. The fuzzy set can be represented as $A = (I, M)$, where the input variable $x \epsilon I$ and $M : I \rightarrow [0, 1]$. M is the fuzzy membership function. In case that $I = \{x_1, x_2, x_3, ..., x_i\}$ and i is the number of the elements in a fuzzy set, then the fuzzy set can be written as given

in the Eq. 8.6[121, 122].

$$A = (I, M) = \frac{u(x_1)}{x_1} + \frac{u(x_2)}{x_2} + ...\frac{u(x_n)}{x_n} = \sum_{i=1}^{n} \frac{u(x_i)}{x_i} \tag{8.6}$$

The probabilistic fuzzy set $\hat{A}$ can be described as a probability space of $\hat{A} = (A, P)$, where the set of all possible events $A = \{A_j\} = \{(x, u_j), j = 1, 2, ..., m\}$ and $x \epsilon I$ is the input variable. The probabilistic fuzzy set can be formulated as the union of the finite space as given in the Eq. 8.7, and for all the element event $S_j \epsilon S$ the $P(S_j) \geq 0$, P(S)=1.

$$\hat{A} = \bigcup_{x \in I} ((I, U), P) \tag{8.7}$$

The operation of defuzzification is realized by computing the centroid with the association of the mathematical expectation. For each possible input event, the output has a fuzzy set which has $\hat{N}$ elements and each element is assigned a value $\rho_k (k = 1, 2, ..., \hat{N})$ and every number is corresponding to a membership function $\mu_\rho(x, \rho_k)$. The centroid output $\rho^\star$ is computed as given in the Eq. 8.8.

$$\rho^\star = \frac{\sum_{k=1}^{E} \rho_k u_k(x, \rho_k)}{\sum_{k=1}^{E} u_k(x, \rho_k)} \tag{8.8}$$

E are the possible events. The overall system for the probabilistic fuzzy sets is described in the Fig. 8.3. For each control step, the first layer will reduce the uncertainties in the ultrasonic sensors based on the probabilistic fuzzy set. Thus, each robot can calculate the distance between itself and any dynamic robot in its path. The second layer of the probabilistic fuzzy controller has been used to avoid an obstacle. When there is a dynamic robot moving close to the robot it will reduce its speed based on the shortest distance among the ultrasonic sensor readings (D_1, D_2, D_3) and also it will change the orientation of the robot based on the comparison between the three distances. In other words, the intelligent control has the ability to decide which orientation is better (Turn Left, or Turn Right , or keep going forward with low speed). Finally, each robot has two navigation control, each with 2 layers. First, i^{th} Rules in order to control the speed of the robot. Second, the j^{th} Rules which is responsible for controlling the orientation of the robot.

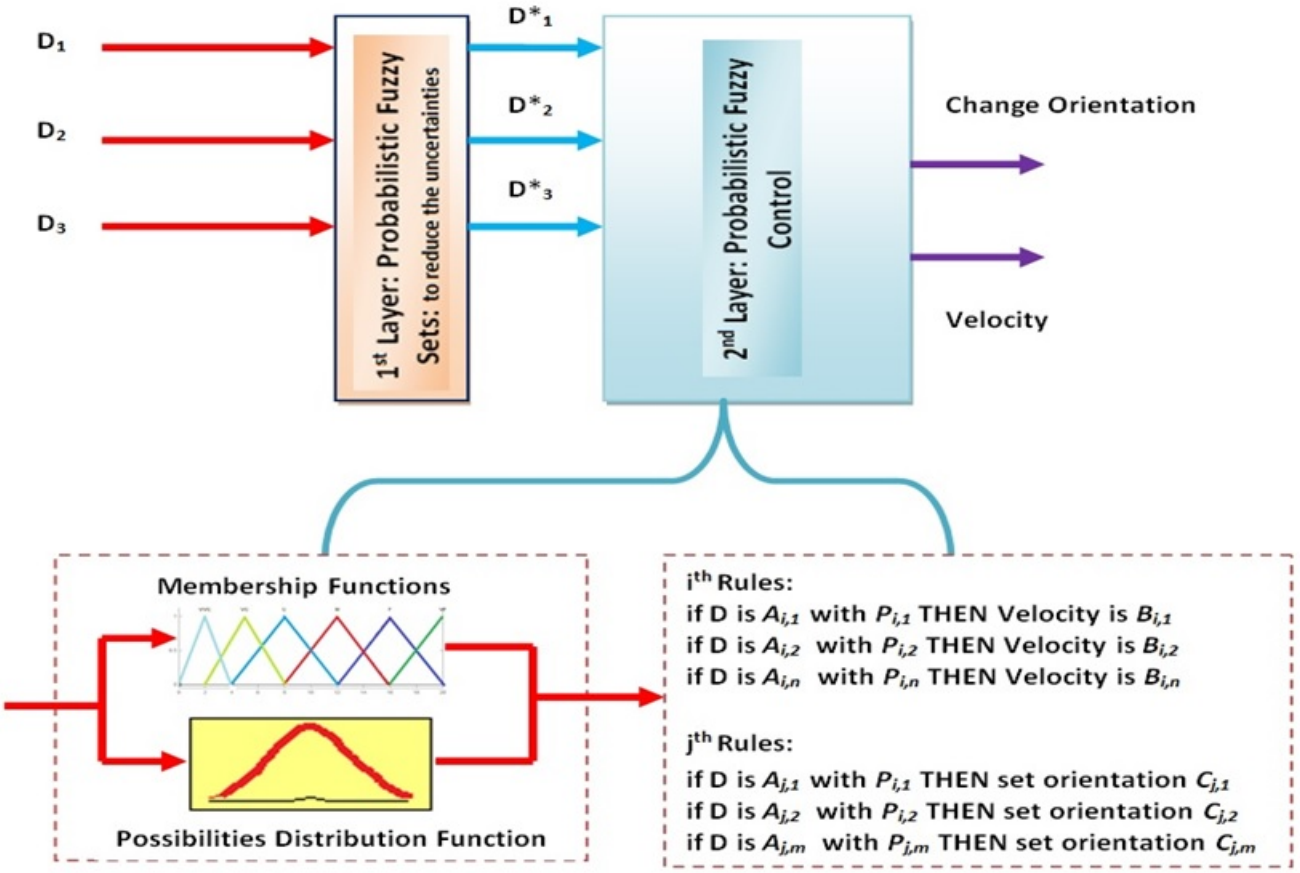

FIGURE 8.3: The overall system for the probabilistic fuzzy sets

8.3 ANFIS

The ANFIS incorporated with the class of rules extracting systems using a decompositional strategy, where rules are extracted at the level of individual nodes within the neural network and then aggregating these rules to form the global behavior description[124, 125]. The ANFIS was divided into two parts: one was trained to obtain the fuzzy rules for the leader robot whilst the other to extract the rules for the follower robots. The fuzzy inference system for the leader robot has three inputs: $\boldsymbol{x_{L,i}}$ leader x position, $\boldsymbol{y_{L,i}}$ leader y position and leader orientation $\boldsymbol{\theta_{L,i}}$ and two outputs (linear speed $\boldsymbol{V_{L,i}}$ and angular speed $\boldsymbol{\omega_{L,i}}$). For the followers the fuzzy inference system has three inputs: $\boldsymbol{x_{L,i}}$ leader x position, $\boldsymbol{y_{L,i}}$ leader y position , and the leader orientation $\boldsymbol{\theta_{L,i}}$ and two outputs (linear speed $\boldsymbol{V_{F,i}}$ and angular speed $\boldsymbol{\omega_{F,i}}$). The training data will construct a mapping between the leader robot and the follower robots based on the linear and angular velocities of the leader. Thus, the follower robots will keep moving toward the leader robot based on its location. For a first order Sugeno fuzzy model the rules are

IF $x_{L,i}$ is A_i and $y_{L,i}$ is B_i and $\theta_{L,i}$ is C_i
THEN $V_{L,i} = a_{1i}x_{L,i} + a_{2i}y_{L,i} + a_{3i}\theta_{L,i} + a_{4i}$
$\omega_{L,i} = a_{5i}x_{L,i} + a_{6i}y_{L,i} + a_{7i}\theta_{L,i} + a_{8i}$

IF $x_{L,i}$ is A_i and $y_{L,i}$ is B_i and $\theta_{L,i}$ is C_i and $d_{FL,i}$ is D_i
THEN $V_{F,i} = P_{1i}x_{L,i} + P_{2i}y_{L,i} + P_{3i}\theta_{L,i} + P_{5i}$
$\omega_{F,i} = P_{6i}x_{L,i} + P_{7i}y_{L,i} + P_{8i}\theta_{L,i} + P_{10i}$

The ANFIS structure of the fuzzy model is shown in the Figure 8.4. The architecture of the system includes five layers, the fuzzy layer, product layer, normalized layer, de-fuzzy layer and the total output layer. In the first layer, every node is an adaptive node with a node function. The first node membership functions $\mu_A(x)$ for the output is described in the Eq. 8.9,-8.11.

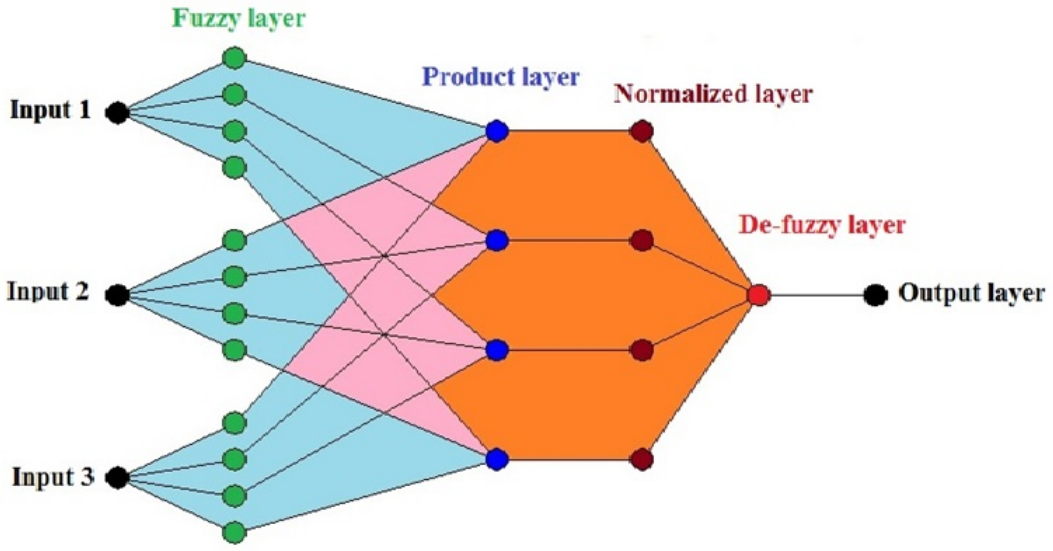

FIGURE 8.4: The ANFIS structure

$$\mu_A(x_{L,i}) = \frac{1}{[1 + (|x_{L,i} - c_i|/a_i)^{2b_i}]} \tag{8.9}$$

$$\mu_B(y_{L,i}) = \frac{1}{[1 + (|y_{L,i} - c_i|/a_i)^{2b_i}]} \tag{8.10}$$

$$\mu_C(\theta_{L,i}) = \frac{1}{[1 + (|\theta_{L,i} - c_i|/a_i)^{2b_i}]} \tag{8.11}$$

(a_i, b_i, c_i) are the parameters of the bell shaped functions. Every node in layer two is fixed node and uses the product for the firing output as given in the Eq. 8.12.

$$O_2 = \mu_A(x_{L,i})\mu_B(y_{L,i})\mu_C(\theta_{L,i}) \tag{8.12}$$

The third layer is also fixed node and it calculates the normalized strengths according to the Eq. 8.13.

$$O_3 = \frac{w_i}{w_1 + w_2 + w_3} \tag{8.13}$$

Every node in the fourth layer is an adaptive node with a node function as described in the Eq. 8.14.

$$O_4 = O_3(f(i)) \tag{8.14}$$

Where $f(i) = V_{L,i}$ or $\omega_{L,i}$ or $V_{F,i}$ or $\omega_{F,i}$. In the final layer a single node is obtained to compute the overall output as given in the Eq. 8.15.

$$O_{5,i} = \frac{\sum_i O_4}{\sum_i O_3} \tag{8.15}$$

The system output is the weighted sum of the results of the rules. The number of fuzzy sets is determined by the number of nodes in the layer 1. On the other hand, the dimension of the layer 3 determines the number of fuzzy rules employed in the architecture that shows the complexity and flexibility.

In order to develop the controller based on ANFIS, the data was divided into 85% training data and 15% for checking. The approach started by generating the input/output data, then subtractive clustering algorithm, namely the fuzzy c-means was used as a data clustering technique wherein each data point belongs to a cluster to some degree that is specified by a membership grade. Then, the data was trained to identify the parameters of Sugeno-type fuzzy inference system based on the hybrid algorithm combining the least square estimation (LSE) and the backpropagation gradient descent method. After training, fuzzy inference calculations of the developed model were performed. Then, the input vectors from the test data set were presented to the trained network. Finally, the criterion used for the measurement of the network performance was the root mean square error (RMSE) given in the Eq.8.16, and the mean relative error (MRE) given in the Eq. 8.17. MRE shows the closeness between predicted and eventual outcomes. MRE is average of absolute errors. Y_{fuzzy} is predicted and $Y_{desired}$ is true desired value. All errors have equal weight in MRE. RMSE is squaring the difference between two values and then taking average and then square root. Hence high weight is given to large errors. The estimation errors for the linear and angular velocities

TABLE 8.1: The network performance for the leader robot

Criterions	V_L	ω_L
MRE	**0.00202**	**0.0031**
RMSE	**0.014**	**0.057**

TABLE 8.2: The network performance for the follower robot

Criterions	V_F	ω_L
MRE	**0.0018**	**0.004**
RMSE	**0.024**	**0.016**

for leader and followers are given in Table. 8.1, and Table. 8.2.

$$RMSE = \sqrt{\frac{\sum_i^N (Y_{fuzzy,i} - Y_{desired,i})^2}{N}} \tag{8.16}$$

$$MRE = \sum_i^N \frac{|Y_{fuzzy,i} - Y_{desired,i}|}{N} \tag{8.17}$$

Fig. 8.5, and Fig. 8.6, show the membership functions for the input variables for leader robot controllers where bell-shaped membership functions are used to describe the variable. Fig. 8.7 and Fig. 8.8, show the automatically generated membership functions for the three input variables for follower robots.

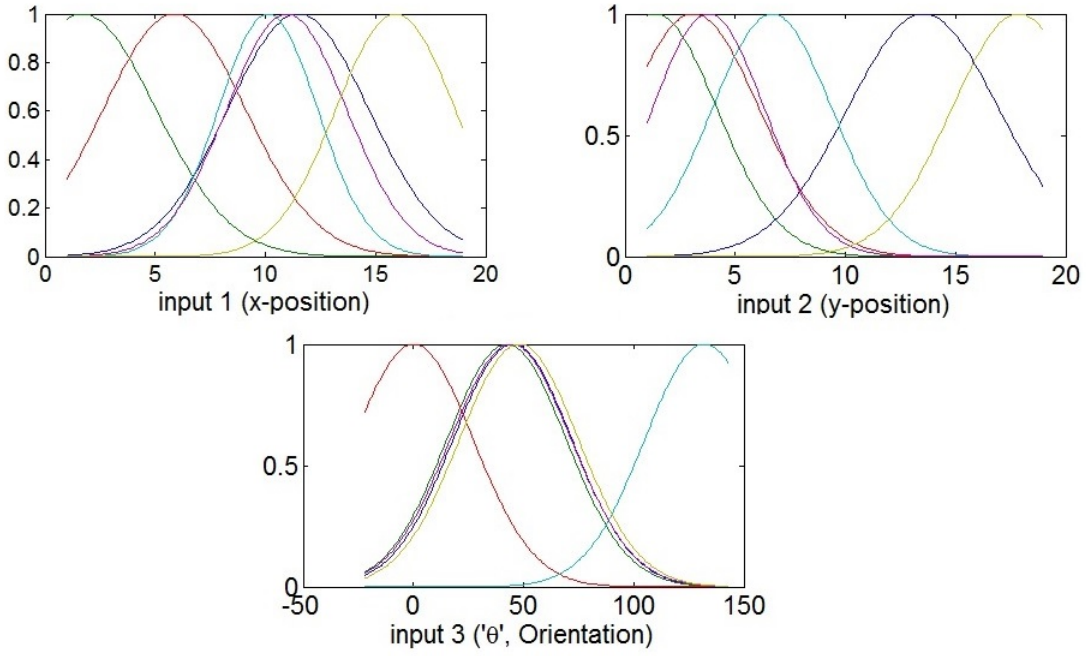

FIGURE 8.5: The membership functions for the leader linear speed

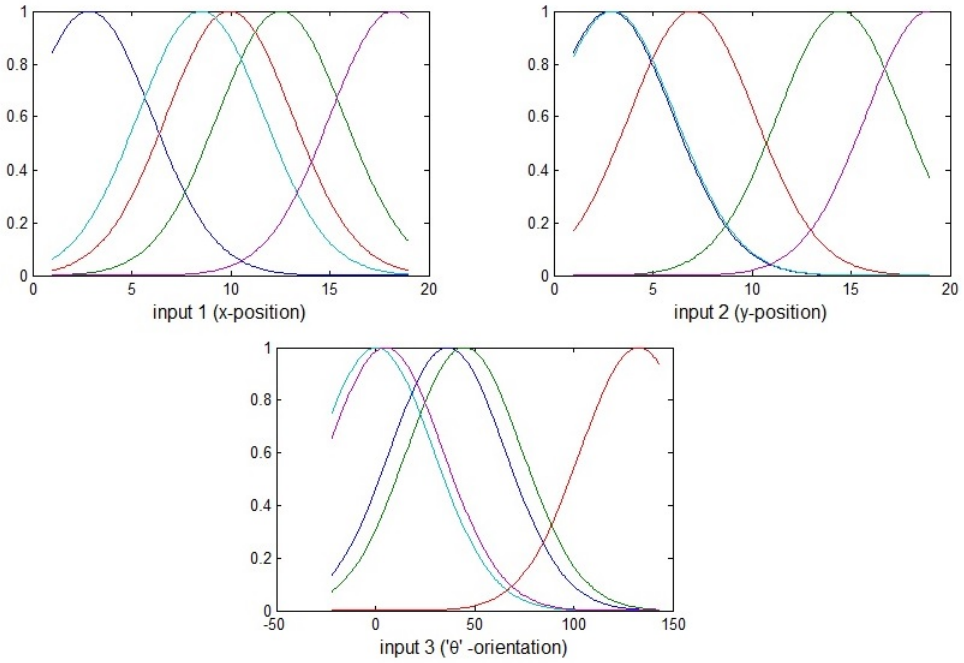

FIGURE 8.6: The membership functions for the leader angular speed

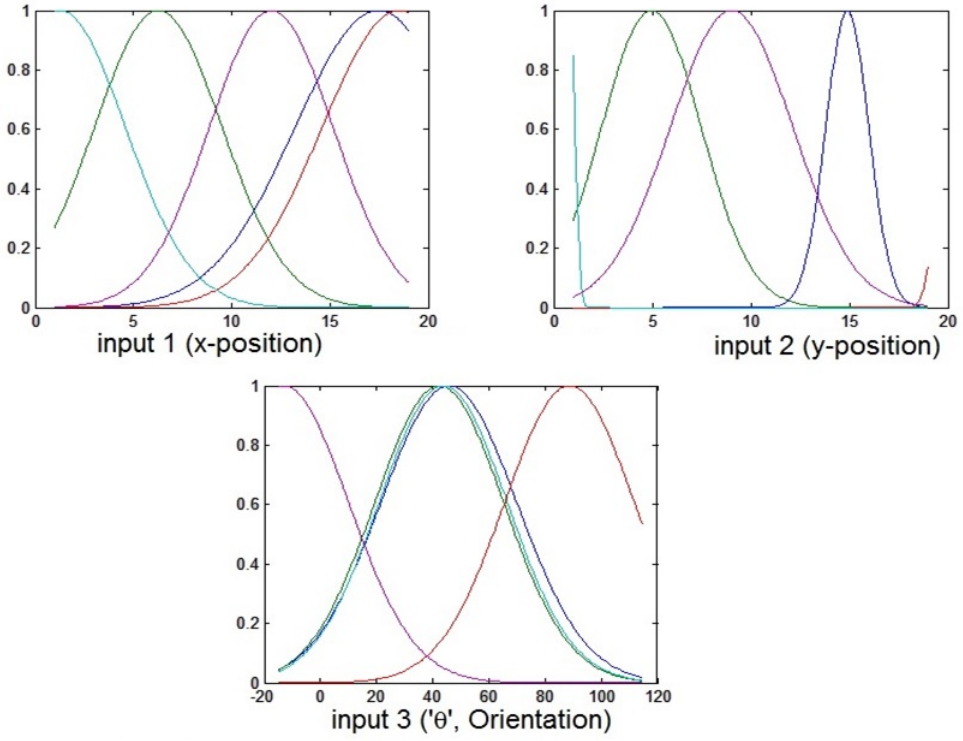

FIGURE 8.7: The membership functions for the follower linear speed

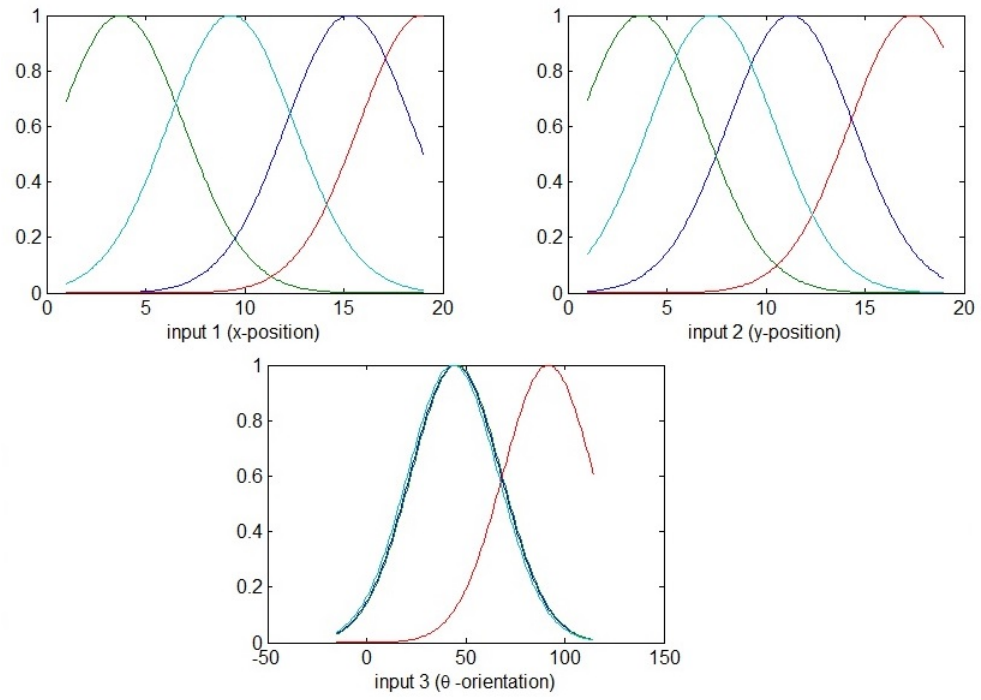

FIGURE 8.8: The membership functions for the follower angular speed

8.4 Experimental Results

In order to demonstrate the performance of the presented system, several experiments were carried out using a simulated platform. In order to implement the conception of leader-follower formation. First part of the simuation i.e. PFLS has been implemented on real robot AutoMerlin. This implementation is given in Ch. 5. In the first simulation four robots were used and all were equipped with ultrasonic sensors. The environment is assumed to be completely unknown to them as shown in the Fig. 8.9. The PFLS helped the four robots to navigate safely and effectively. In addition, it shows the follower robots moved toward the leader based on its positions. The Fig. 8.10, is illustrating the performance of the proposed approach in order to help the follower robots to navigate safely without any dangerous collision and risk.

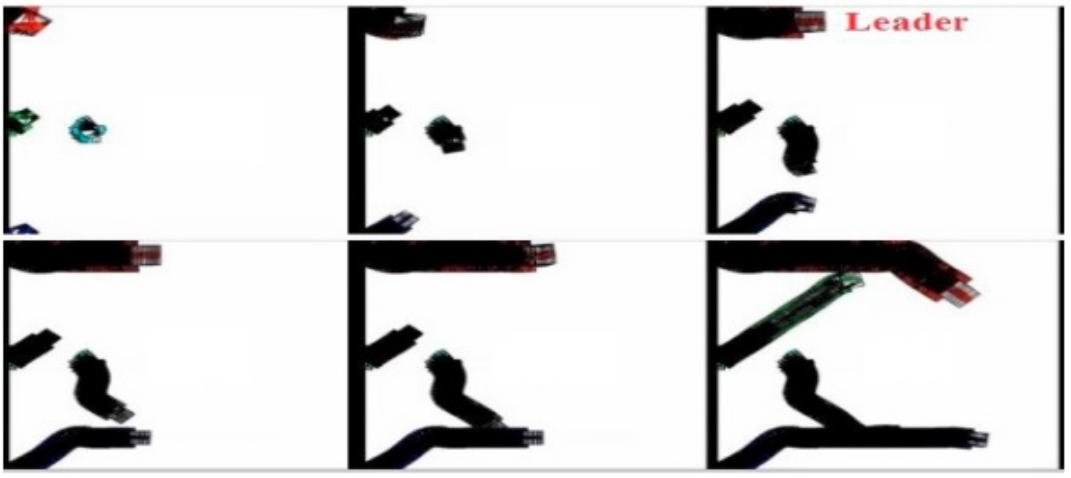

FIGURE 8.9: Simulation 1

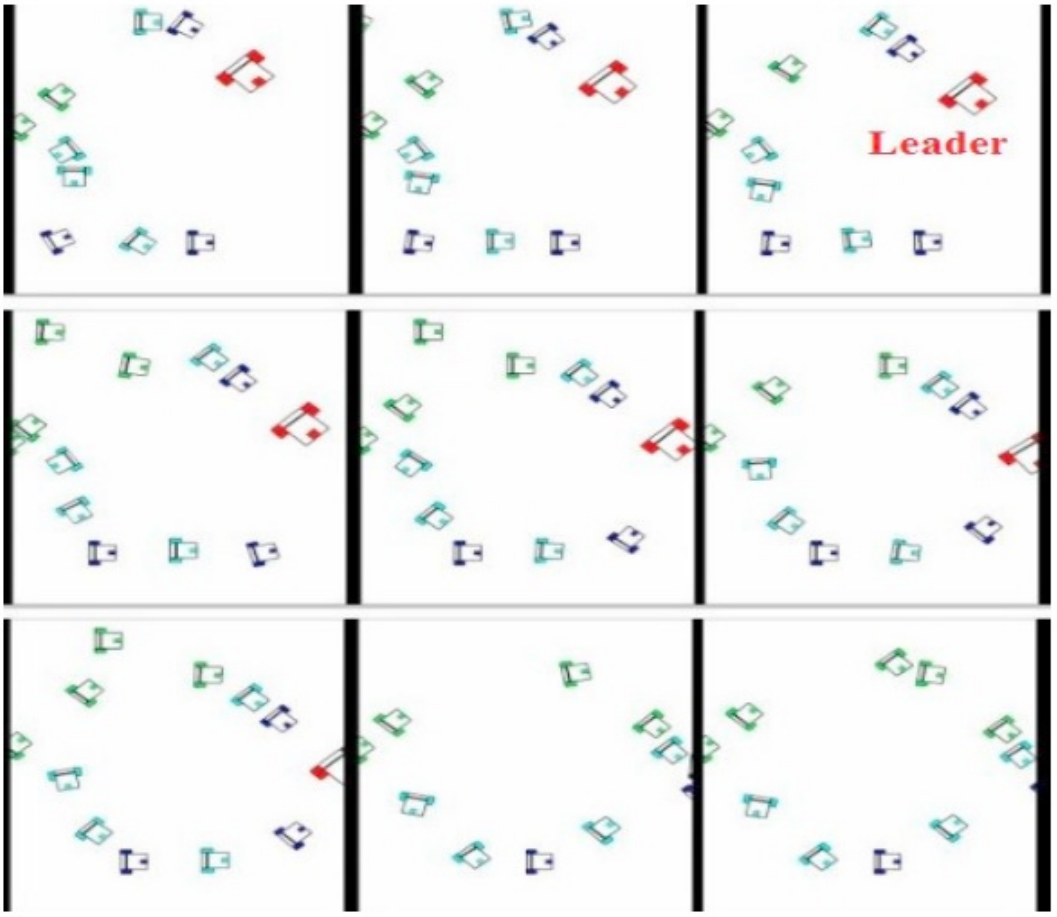

FIGURE 8.10: Simulation 2

Chapter 9

Conclusion and Future Work

9.1 Conclusion

In this dissertation, telecontrol of the mobile robot AutoMerlin has been investigated. Different approaches like Event Based Control Approach, Fuzzy Logic Control, and Time Domain Passivity Control have been used to develop the stable teleoperation system. The proposed stabilization schemes have been implemented in the following different scenarios.

- Fix execution time of the commanded actions
- Fuzzy Logic for the controller design and autonomous navigation for the obstacle avoidance
- Time Domain Passivity Control (TDPC) for teleoperation
- Time Delay Power Network (TDPN) with a constant delay
- Time Delay Power Network (TDPN) with a random delay

In all of the above cases, a very good stabilization has been achieved as discussed in the Ch. 4, to Ch. 8. Event Based Control for the teleoperation of the mobile robot has been described in the Ch. 4. The human operator controls and navigates the robot and receives the sensory feedback. The environment force acting on the slave robot has been modeled as a virtual force based on the obstacles in front of the mobile robot using proximity sensors. Thus, the operator feels that he

is driving the robot like a car while he is present at the remote location. The commanded action execution time has been fixed to 200ms in order to ensure stable teleoperation with a delay. In the case of connection loss or longer delays, the robot remains safe. The designed controller shows the excellent coordination between the master haptic device and the slave robot. The slave robot follows the master device and the communication delay has no effect on the performance and stability of the teleoperated robot. Ch. 5, presents a Fuzzy Logic Control technique to cope with the issue of the time delay from the Internet communication for the ground robot teleoperation. The controller has been designed based on two inputs. The first input comes from the human operator and the second input is the sonar reading. The controller calculates the output speed of the robot based on these two parameters. Thus, the controller functions only when both parameters are available. In the case of a delay, the controller would not keep running the robot it would wait for the next input from the operator. Similarly, the controller would not move the robot very fast when there are obstacles close to it. Additionally, FIN Algorithm has been designed to work along with the speed controller in teleoperation to safely navigate and avoid the collisions with the obstacles in an indoor environment during teleoperation. The FIN Algorithm has the ability to deal with the drawbacks in the ultrasonic sensors and to reduce these drawbacks. The addition of the algorithm is helpful to a human operator in assisting him to teleoperate the robot with a time delay without colliding with different objects present in the environment.

Time Domain Passivity Control keeps the track of the energy of the system. The controller has been designed with a communication network which imitates the Internet. In Ch. 6, Time Domain Passivity Control has been implemented for the teleoperation with a distortion in data only without a delay. The controller performance has been described in the Section. 6.3. The controller keeps the system passive and dissipates the surplus energy whenever the system shows an active behavior. The experiments have been performed by considering drawbacks in the Internet communication like throttling, packet loss, duplication, out of order and tempering in data. Then, the work has been extended to address the issue of the constant and stochastic delay in the communication. In Ch. 7, the bilateral controller has been designed which performs equally well in the presence of a fix or random delay. The surplus energy has been dissipated on the master and the slave side and to make the bilateral telecontrol passive and stable. The robot follows the human operator commands and the operator experiences the same

force feedback acting on the mobile robot. It is vivid that the TDPN approach is really beneficial and effective for the teleoperation of a mobile robot. TDPN is not based on the system model and it does not impose any stringent rules on the system parameters. It only functions when needed. The perforamce of TDPN approach has been described in Section. 7.3, and 7.4.

Path planning and motion coordination for the multiple robots have been presented in the Ch. 8. The approach is based on the PFLS to map the typical nonlinear relation of the input/output model with a stochastic and the fuzzy uncertainties. Then, second layer controls the obstacles avoidance behavior of the robots. Also, the approach is based on the leader-follower coordination. The higher level controller has been designed by using the ANFIS that enables the leader robot to reach the target and enables the follower robots to follow the desired path so that they keep on following the leader robot. The simulations show the presented probabilistic fuzzy approach with ANFIS makes the system robust and improves the navigation performance. The experiments on a real mobile robot for the only probabilistic fuzzy approach had been tested and it showed a good performance to avoid static or moving obstacles.

9.2 Contribution

During the tenure of the Ph.D. research, the author's contribution is listed as follows. Main achievements are

- Event based teleoperation
- Implication of Fuzzy Logic
- Time Domain Passivity control
- Time delay power network
- Leader-follower concept in the multiple robots with the probabilistic fuzzy and ANFIS

The following publications have been done related to research work in telecontrol.

1. A. Shahzad and H. Roth, Teleoperation of mobile robot using event based controller and real time force feedback, *Scientific Cooperations International Workshops on Electrical and Computer Engineering Subfields 22-23 August 2014, Koc University, ISTANBUL/TURKEY*

2. A. Shahzad, R. Al-jarrah and H. Roth, Telecontrol of AutoMerlin Robot by Employing Fuzzy Logic, *International Journal of Mechanical Engineering and Robotics Research (IJMERR), Volume 5, No. 1, January, 2016*

3. A. Shahzad, R. Al-jarrah and H. Roth, Teleoperation of AutoMerlin by Inculcating FIN Algorithm, *International Journal of Mechanical Engineering and Robotics Research (IJMERR), Volume 5, No. 1, January, 2016*

4. A. Shahzad, H. Roth, Bilateral Telecontrol of AutoMerlin Mobile Robot, *9Th IEEE international conference on open source systems and technologies, Lahore, Pakistan, 17-19 December, 2015*

5. A. Shahzad and H. Roth, Bilateral Telecontrol of AutoMerlin Mobile Robot with Fix Communcation Delay, *International Conference on Automation, Quality and Testing, Robotics, Cluj Napoca, Romania, May 2016*

6. R. Al-Jarrah, A. Shahzad and H. Roth,Path Planning and Motion Coordination for Multi-Robots System Using Probabilistic Neuro-Fuzzy, *2nd IFAC Conference on Embedded Systems, Computational Intelligence and Telematics in Control, CESCIT'15, to be held in Maribor, Slovenia, June, 22nd -24th, 2015*

7. R. Al-Jarrah, B. Kadhim, A. Shahzad, and H. Roth, Towards a Heterogeneous Navigation Team of Aerial-Ground Robots Based on Fuzzy Image Processing, *International Journal of Materials Science and Engineering, Volume 3, No. 1, March, 2015*

9.3 Future Work

In the future work, more robots can be added to conduct teleoperation of the multiple robots. A team of the heterogeneous robots can be build in which an aerial robot can provide the aerial view of the remote environment. The teleoperation can be performed much accurately based on this aerial information.

Bibliography

[1] O. Linda and M. Manic. "Self-organizing fuzzy haptic teleoperation of mobile robot using sparse sonar data". *IEEE Transactions on Industrial Electronics*, 58(8), August 2011.

[2] K. M. Al-Aubidy, M. M. Ali, A. M. Derbas, and A.W. Al-Mutairi. "Gprs-based remote sensing and teleoperation of a mobile robot". *10th International Multi-Conference on Systems, Signals and Devices, Hammamet, Tunisia*, March 2013.

[3] I. Farkhatdinov, J. H. Ryu, and J. An. "A preliminary experimental study on haptic teleoperation of mobile robot with variable force feedback gain". *IEEE Haptics Symposium Waltham*, March 2010.

[4] Z. Szanto, L. Marton, P. Haller, and S. Gyorgy. "Performance analysis of WLAN based mobile robot teleoperation". *IEEE International Conference on Intelligent Computer Communication and Processing (ICCP), Cluj-Napoca, Romania*, September 2013.

[5] S. K. Cho, H. Z. Jin, J. M. Lee, and B. Yao. "Teleoperation of a mobile robot using a force-reflection joystick with sensing mechanism of rotating magnetic field". *IEEE/ASME TRANSACTIONS ON MECHATRONICS*, 15(1), February 2013.

[6] A. Shahzad and H. Roth. "Teleoperation of mobile robot using event based controller and real time force feedback". *Scientific Cooperations, International Workshops on Electrical and Computer Engineering Subfields, Koc University, Istanbul/Turkey*, August 2013.

[7] A. Shahzad and H. Roth. "Telecontrol of AutoMerlin robot by employing fuzzy logic". *International Journal of Mechanical Engineering and Robotics Research (IJMERR)*, 5(1), January 2016.

[8] A. Shahzad and H. Roth. "Teleoperation of AutoMerlin by inculcating FIN algorithm". *International Journal of Mechanical Engineering and Robotics Research (IJMERR)*, 5(1), January 2016.

[9] A. Shahzad and H. Roth. "Bilateral telecontrol of AutoMerlin mobile robot". *9Th IEEE international conference on open source systems and technologies, Lahore, Pakistan*, December 2015.

[10] J.-L. Michel, M. Klages, F. J. A. S. Barriga, Y. Fouquet, M. Sibuet, P.-M. Sarradin, P. Siméoni, and J.-F Drogou. "Victor 6000: Design, Utilization and First Improvements". *The Thirteenth International Offshore and Polar Engineering Conference Honolulu, Hawaii, USA*, May 2003.

[11] S. B. Skaar and C. F. Ruoff. "Teleoperation and Robotics in Space". *Published by American Institute of Aeronautics and Astronautics*, 1994.

[12] L.Pedersen, D.Kortenkamp, D. Wettergreen, and I. Nourbakhsh. "A Survey of Space Robotics". *NASA Technical Documents, USA*, March 2003.

[13] J. Hollingum. "Military look to flying insect robots". *Industrial Robot: An International Journal*, 25(2):124–128, 1998.

[14] C.-Yi. Su, Y. Xia, and Z. Li. "Intelligent Networked Teleoperation Control". *Springer*, pages 3–4, May 2015.

[15] D. W. Hainsworth. "Teleoperation User Interfaces for Mining Robotics". *Autonomous Robots*, 11(1):19–28, July 2001.

[16] A. Iqbal. "Stabilization Of Delayed Teleoperation Systems Using Time Domain Passivity Control". *PhD Thesis, University of Siegen, Germany*, June 2007.

[17] T. B. Sheridan. "Teleoperation, telepresence, and telerobotics: Research needs for space". *Human Factors in Automated and Robotic Space Systems: Proc. of a Symposium*, pages 279–291, 1987.

[18] G. Niemeyer and J. E. Slotine. "Telemanipulation with time delays ". *The International Journal of Robotics Research*, 23(9):1873–890, September 2004.

[19] N. Hogan. "Impedance control: An approach to manipulation, part i- theory". *ASME Journal of Dynamic Systems, Measurement, and Control*, 107: 1–7, March 1985.

[20] S. Hirche and M. Buss. "Study of teleoperation using realtime communication network emulation". *Proc of IEEE/ASME Internationa Conference on Advanced Intelligent Mechatronic*, 2003.

[21] A. Frisoli, E. Sotgiu, C.A. Avizzano, D. Checcacci, and M. Bergamasco. "Forcebased impedance control of a haptic master system for teleoperation". *Sensor Review*, 24(1):42–50, 2004.

[22] W. R. Ferrell. "Delayed force feedback". *Human Factors*, 8:449–455, 1966.

[23] T. B. Sheridan. "Space teleoperation through time delay: Review and prognosis". *IEEE Transactions on Robotics and Automation*, 9(5):592–602, October 1993.

[24] R. C. Goertz. "Electronically controlled manipulator". *Nucleonics*, 12(11): 46–47, 1954.

[25] T. B. Sheridan and W. R. Ferrell. "Remote manipulative control with transmission delay". *IEEE Transactions on Human Factors in Electronics*, 4: 25–29, 1963.

[26] F. Miyazaki, S. Matsubayashi, T.Yoshimi, and S. Arimoto. "A new control methodology toward advanced teleoperation of master–slave robot systems". *In Proceedings of the IEEE international conference on robotics and automation*, 3:997–1002, 1986.

[27] G. J. Raju, G. C. Verghese, and T. B. Sheridan. "Design issues in 2-port network models of bilateral remote manipulation". *In Proceedings of the IEEE international conference on robotics and automation*, 3(11):1316–1321, 1989.

[28] B. Hannaford and P. Fiorini. "A detailed model of bilateral teleoperation". *In Proceedings of the IEEE international conference on systems, man and cybernetics*, 1:117–121, 1988.

[29] R. J. Anderson and M. W. Spong. "Bilateral control of teleoperators with time delay". *In Proceedings of the IEEE conference on decision and control, Austin,Texsas*, 1:167–173, 1988.

[30] W. R. Ferrell. "Remote manipulation with transmission delay". *IEEE Transactions on Human Factors in Electronics*, 6:24–32, 1965.

[31] P. F. Hokeyem and M. W. Spong. "Bilateral teleoperation: An historical surve". *Automatica*, 42(12):2035–2057, 2006.

[32] W. R. Ferrell and T. B. Sheridan. "Supervisory control of remote manipulation". *IEEE Spectrum*, pages 81–88, 1967.

[33] D. Whitney. "State space models of remote manipulation tasks". *IEEE Transactions on Automatic Control*, 14(6):617–623, 1969.

[34] C. Fong, R. Dotson, and A. Bejczy. "Distributed microcomputer control system for advanced teleoperation". *In Proceedings of the IEEE international conference on robotics and automation*, 3:987–995, 1986.

[35] S. Lee, G. Bekey, and A. K. Bejczy. "Computer control of spaceborne teleoperators with sensory feedback". *In Proceedings of the IEEE international conference on robotics and automation*, 2:205–214, 1985.

[36] A. Madni, Y. Y. Chu, and A. Freedy. "Intelligent interface for remote supervision and control of underwater manipulation". *OCEANS*, 15:106–110, 1983.

[37] R. paul, T. Lindsay, and C. Sayers. "Time delay insensitive teleoperation". *In Proceedings of the IEEE/RSJ international conference on intelligent robots and systems*, 1:247–254, 1992.

[38] T. Sato and S. Hirai. "Language-aided robotic teleoperation system(larts) for advanced teleoperation". *IEEE Journal of Robotics and Automation*, 3 (5):476–481, 1987.

[39] D. A. Lawrence. "Stability and transparency in bilateral teleoperation". *IEEE Transactions on Robotics and Automation*, 9(5):625–637, 1992.

[40] Y. Yokokohji and T. Yoshikawa. "Bilateral control of master–slave manipulators for ideal kinesthetic coupling—formulation and experiment". *IEEE Transactions on Robotics and Automation*, 10(5):605–620, 1994.

[41] M. W. Spong, S. Hutchinson, and M. Vidyasagar. "Robot modeling and control". *Wiley. New York*, 2005.

[42] F. T. Buzan and T. B. Sheridan. "A model-based predictive operator aid for telemanipulators with time delay". *In the Proceedings of the IEEE International conference on systems, man and cybernetics*, 1:138–143, 1989.

[43] B. Hannaford. "Stability and performance tradeoffs in bilateral telemanipulation". *In Proceedings of the IEEE international conference on robotics and automation*, 3:1764–1767, 1989.

[44] B. Hannaford. "A design framework for teleoperators with kinesthetic feedback". *IEEE Transactions on Robotics and Automation*, 5(4):426–434, 1989.

[45] Y. Strassberg, A. A. Goldenberg, and J.K Mills. "A new control scheme for bilateral teleoperating systems: Lyapunov stability analysis". *In Proceedings of the IEEE international conference on robotics and automation*, 1:837–842, 1992.

[46] Y. Strassberg, A. A. Goldenberg, and J.K Mills. "A new control scheme for bilateral teleoperating systems: Performance evaluation and comparison". *In Proceedings of the IEEE/RSJ international conference on intelligent robots and systems*, 2:865–872, 1992.

[47] R. J. Anderson and M. W. Spong. "Asymptotic stability for force reflecting teleoperators with time delay". *In Proceedings of the IEEE international conference on robotics and automation*, 3:1618–1625, 1989.

[48] R. J. Anderson and M. W. Spong. "Bilateral control of teleoperators with time delay". *IEEE Transactions on Automatic Control*, 34(5):494–501, 1989.

[49] J. E. Colgate. "Power and impedance scaling in bilateral manipulation". *In Proceedings of the IEEE international conference on robotics and automation*, 3:2292–2297, 1991.

[50] K. Kosuge, T. Itoh, and T. Fukuda. " Scaled telemanipulation with communication time delay". *In Proceedings of the IEEE international conference on robotics and automation*, pages 2019–2024, 1996.

[51] G. Niemeyer and J. J-. E. Slotine. "Stable adaptive teleoperation". *IEEE Journal of Oceanic Engineering*, 16(1):152–162, 1991.

[52] S. Munir and W. J. Book. "Internet based teleoperation using wave variables with prediction". *In Proceedings of the IEEE/ASME international*

conference on advanced intelligent mechatronics,Como, Italy, pages 43–49, 2001.

[53] S. Munir and W. J. Book. "Wave-based teleoperation with prediction". *In Proceedings of the IEEE American control conference*, 6:4605–4611, 2001.

[54] S. Ganjefar, H. Momeni, and F. J-Sharifi. "Teleoperation systems design using augmented wave-variables and Smith predictor method for reducing time-delay effects". *In Proceedings of the IEEE international symposium on intelligent control,Vancouver, Canada*, pages 333–338, 2002.

[55] G. Niemeyer and J. J. E. Slotine. "Designing force reflecting teleoperators with large time delays to appear as virtual tools". *In Proceedings of the IEEE international conference on robotics and automation,Albuquerque, NM, USA*, 3:2212–2218, 1997.

[56] G. Niemeyer and J. J. E. Slotine. " Transient shaping in force-reflecting teleoperation". *In International conference on advanced robotics*, 1:261–266, 1991.

[57] C. Benedetti, M. Franchini, and P. Fiorini. "Stable tracking in variable time-delay teleoperation". *In Proceedings of the IEEE/RSJ international conference on intelligent robots and systems*, 4:2252–2257, 2001.

[58] K. Goldberg, M. Mascha, S. Gentner, N. Rothenberg, C. Sutter, and J. Wiegley. "Desktop teleoperation via the world wide web". *In Proceedings of the IEEE international conference on robotics and automation*, 1:654–659, 1995.

[59] K. Brady and T. J. Tran. "Internet-based remote teleoperation". *In Proceedings of the IEEE international conference on robotics and automation*, 1:65–70, 1998.

[60] I. Elhaij, H. Hummert, N. Xi, W. K. Fung, and Y. H. Liu. "Realtime bilateral control of internet-based teleoperation". *In Proceedings of the third World congress on intelligent control and automation*, 5:3761–3766, 2000.

[61] R. Oboe. "Web-interfaced, force-reflecting teleoperation systems". *IEEE Transactions on Industrial Electronics*, 48(6):1257–1265, 2001.

[62] R. Oboe. "Force-reflecting teleoperation over the internet". *The JBIT project. Proceedings of the IEEE*, 91(3):449–462, 2003.

[63] N. Xi and T. J. Tarn. "Action synchronization and control of internet based telerobotic systems". *In Proceedings of the IEEE international conference on robotics and automation*, 1:219–224, 2000.

[64] N. Xi and T. J. Tarn. "Stability analysis of non-time referenced internetbased telerobotic systems". *Journal of Robotics and Autonomous Systems*, 32(2-3): 173–178, 2001.

[65] T. Mirfakhrai and S. Payandeh. "A delay prediction approach for teleoperation over the internet". *In Proceedings of the IEEE international conference on robotics and automation*, 3:2178–2183, 2002.

[66] X.Ye, M. Q.-H. Meng, P. X. Liu, and G. Li. "Statistical analysis and prediction of round trip delay for internet-based teleoperation". *In Proceedings of the IEEE/RSJ international conference on intelligent robots and systems*, 3:2999–3004, 2002.

[67] X.Ye, M. Q.-H. Meng, P. X. Liu, and G. Li. "Passivation of force reflecting bilateral teleoperators with time varying delay". *Mechatronics, Entschede, Netherlands*, 2002.

[68] S. I. Niculescu, C. T. Abdallah, and P.F. Hokayem. "Effects of channel dynamics on the stability of teleoperation". *In IFAC Workshop on TimeDelay Systems INRIA, Rocquencourt, France*, pages 2999–3004, 2003.

[69] G. Niemeyer and J. J. E. Slotine. "Using wave variables for system analysis and robot control". *In Proceedings of the IEEE international conference on robotics and automation,Albuquerque, NM, USA*, 3:1619–1625, 1997.

[70] N. Chopra, M. W. Spong, R. Ortega, and N. E. Barabanov. "On position tracking in bilateral teleoperation". *IEEE Transactions on Robotics*, 22(4): 861–866, 2006.

[71] N. Chopra, M. W. Spong, S. Hirche, and M. Buss. "Bilateral teleoperation over the internet: The time varying delay problem". *In Proceedings of the IEEE American control conference*, 1:155–160, 2003.

[72] G. Niemeyer and J. J. E. Slotine. "Towards force-reflecting teleoperation over the internet". *In Proceedings of the IEEE international conference on robotics and automation*, 3:1909–1915, 1998.

[73] Y. Yokokohji, T. Imaida, and T. Yoshikawa. "Bilateral teleoperation under time-varying communication delay". *In Proceedings of the IEEE/RSJ international conference on intelligent robots and systems*, 3:1854–1859, 1999.

[74] Y. Yokokohji, T. Imaida, and T. Yoshikawa. "Bilateral control with energy balance monitoring under time-varying communication delay". *In Proceedings of the IEEE international conference on robotics and automation, San Francisco, CA, USA*, 3:2684–2689, 2000.

[75] S. Stramigioli, C. Secchi A. van der Schaft, and C. Fantuzzi. "A novel theory for sample data system passivity". *In Proceedings of the IEEE/RSJ international conference on intelligent robots and systems, Lausanne, Switzerland*, pages 1936–1941, 2002.

[76] C. Secchi and S. Stramigioli C. Fantuzzi. "Dealing with unreliabilities in digital passive geometric telemanipulation". *In Proceedings of the IEEE/RSJ international conference on intelligent robots and systems*, 3:2823–2828, 2003.

[77] C. Secchi and S. Stramigioli C. Fantuzzi. "Digital passive geometric telemanipulation". *In Proceedings of the IEEE international conference on robotics and automation*, 3:3290–3295, 2003.

[78] K. Kosuge and H. Murayama. "Bilateral feedback control of telemanipulator via computer network in discrete time domain". *In Proceedings of the IEEE international conference on robotics and automation, Albuquerque, NM, USA*, 3:2219–2224, 1997.

[79] P. Berestesky and N. Chopra M. W. Spong. "Discrete time passivity in bilateral teleoperation over the internet". *In Proceedings of the IEEE international conference on robotics and automation, New Orleans, LA, USA*, 2004.

[80] S. Mastellone and D. Lee M. W Spong. "Master–slave synchronization with switching communication through passive model-based control design". *In Proceeding of American control conference, Minneapolis, MN*, pages 3203–3208, 2006.

[81] D. Lee and P. Y. Li. "Passive control of bilateral teleoperated manipulators: Robust control and experiments". *In Proceedings of the IEEE American control conference*, 6:4612–4618, 2001.

[82] D. Lee and P. Y. Li. "Passive coordination control of nonlinear bilateral teleoperated manipulators". *In Proceedings of the IEEE international conference on robotics and automation*, 3:3278–3283, 2002.

[83] D. Lee and P. Y. Li. "Passive tool dynamics rendering for nonlinear bilateral teleoperated manipulators". *In Proceedings of the IEEE international conference on robotics and automation*, 3:3284–3289, 2002.

[84] D. Lee and P. Y. Li. "Formation and maneuver control of multiple spacecraf". *In Proceedings of the IEEE American control conference*, 3:278–283, 2003.

[85] D. Lee and P. Y. Li. "Passive bilateral feedforward control of linear dynamically similar teleoperated manipulators". *IEEE Transactions on Robotics and Automation*, 19(3):443–456, 2003.

[86] D. Lee and P. Y. Li. "Toward robust passivity: A passive control implementation structure for mechanical teleoperators". *Symposium on haptic interfaces for virtual environment and teleoperator systems*, pages 132–139, 2003.

[87] P. Y. Li. "Passive control of bilateral teleoperated manipulators". *In Proceedings of the IEEE American control conference*, 6:3838–3842, 1998.

[88] N. Chopra M. W. Spong and R. Lozano. "Adaptive coordination control of bilateral teleoperators with time delay". *In Proceedings of the IEEE conference on decision and control*, pages 4540–4547, 2004.

[89] M. W Spong and S. Hutchinson M. Vidyasagar. "Robot modeling and control". *New York: Wiley*, 2005.

[90] P. Buttolo, P. Braathen, and B. Hannaford. "Sliding control of force reflecting teleoperation". *Preliminary studies. In PRESENCE*, 3:158–172, 1994.

[91] H. C. Cho, J. H. Parkand K. Kim, and J. O Park. "Sliding-mode-based impedance controller for bilateral teleoperation under varying time-delay". *In Proceedings of the IEEE international conference on robotics and automation,Seoul, Korea*, 1:1025–1030, 2001.

[92] J. H. Park and C. H. Cho. "Sliding-mode controller for bilateral teleoperation with varying time delay". *Proceedings of the IEEE/ASME international conference on advanced intelligent mechatronic*, pages 311–316, 1999.

[93] J. H. Park and C. H. Cho. "Sliding mode control of bilateral teleoperation systems with force-reflection on the internet". *In Proceedings of the IEEE/RSJ international conference on intelligent robots and systems*, 2: 1187–1192, 2000.

[94] J. H. Park and C. H. Cho. "Bilateral controller for teleoperators with time delay via $\mu - synthesis$". *In Proceedings of the IEEE/RSJ international conference on intelligent robots and systems*, 2:1187–1192, 1995.

[95] D. Gu, P. H. Petkov, and M. M. Konstantinov. "Robust Control Design with MATLAB". *Springer London*, pages 203–219, 2013.

[96] A. Sano and H. Fujimoto M. Tanaka. "Gain-scheduled compensation for time delay of bilateral teleoperation systems". *In Proceedings of the IEEE international conference on robotics and automation*, 3:1916–1923, 1998.

[97] M. Boukhnifer, A. Ferreira, and J. G. Fontaine. "Scaled teleoperation controller design for micromanipulation over internet". *In Proceedings of the IEEE international conference on robotics and automation*, 5:4577–4583, 2004.

[98] Z. Hu, S. E. Salcudean, and P.D. Loewen. " Robust controller design for teleoperation systems". *In Proceedings of the IEEE international conference on systems, man and cybernetics*, 3:2127–2132, 1995.

[99] J. Yan and S. E. Salcudean. "Teleoperation controller design using h_∞-optimization with application to motion-scaling". *IEEE Transactions on Control Systems Technology*, 4(3):244–258, 1996.

[100] K. J. Astroem. "Event Based Control". *Analysis and Design of Nonlinear Control Systems*, pages 127–147, 2008.

[101] V. Vasyutynskyy and K. Kabitzsch. "Event-based control: Overview and generic model". *8th IEEE International Workshop on Factory Communication Systems (WFCS)*, pages 271–279, 2010.

[102] D. Lehmann and J. Lunze. "Event-Based Control: A State-Feedback Approach ". *Proceedings of the European Control Conference*, pages 1716–1721, 2009.

[103] N. Xi and T. J. Tarn. "Action synchronization and control of Internet based telerobotic systems". *IEEE International Conference on Robotics and Automation*, 1:219–224, 1999.

[104] M. J. Patyra and D. J. Mlynek. "Fuzzy Logic: Implementation and Applications". *John Wiley and Sons Inc*, 1996.

[105] L. A. Zadeh. "Fuzzy Sets". *Information and Control*, 8:338–353, 1965.

[106] C. C. Lee. "Fuzzy Logic in Control Systems: Fuzzy Logic Controller, Part I1". *IEEE Transactions on Systems, Man, and Cybernetics*, 20(2):419–435, 1990.

[107] D. Driankov, H. Hellendoorn, and M. Reinfrank. "An Introduction to Fuzzy Control". *Spriger, 2nd Edition*, 1996.

[108] P. Guillemin. "Fuzzy Logic Applied to Motor Control". *IEEE Transaction on Industry Application*, 32(1):51–56, 1996.

[109] I. Iancu. "A Mamdani Type Fuzzy Logic Controller". *Fuzzy Logic - Controls, Concepts, Theories and Applications, INTECH*, pages 325–350, 2012.

[110] O. R. E. Motlagh, T. S. Hong, and N. Ismail. "Development of a new minimum avoidance system for a behavior-based mobile robot". *Fuzzy Sets and Systems Journal*, 20:1929–1946, 2009.

[111] C. Joslyn. "Empirical possibility and minimal information distortion". *in Fuzzy Logic: State of the Art, R. Lowen, Ed., Kluwer*, 1992.

[112] C. Joslyn. "Towards an empirical semantics of possibilities through maximum uncertainty". *in Proc IFSA*, pages 86–89, 1991.

[113] B. Hannaford and J-H. Ryu. "Time-Domain Passivity Control of Haptic Interfaces". *IEEE TRANSACTIONS ON ROBOTICS AND AUTOMATION*, 18(1):1–10, 2002.

[114] J-H. Ryu, D-S. Kwon, and B. Hannaford. "Stable Teleoperation with Time Domain Passivity Control". *IEEE International Conference on Robotics 8 Automation Washington, DC*, pages 3260–3265, 2002.

[115] G. Niemeyer. "Using wave variables in time delayed force reflecting teleoperation". *Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts*, 1996.

[116] J. Artigas, J-H. Ryu, and C. Preusche. "Time Domain Passivity Control for Position-Position Teleoperation Architectures". *MIT Press Journal*, 19(5), 2010.

[117] T. Fukuda and S. Nakagawa. "A dynamically reconfigurable robotic system (concept of a system and optimal configurations)". *In International Conference on Industrial Electronics, Control, and Instrumentation*, pages 585–595, 1987.

[118] T. Fukuda and G. Iritani. "Construction mechanism of group behavior with cooperation". *In IEEE/RSJ IROS*, pages 535–542, 1995.

[119] G. Beni. "The concept of cellular robotic system". *In IEEE International Symposium on Intelligent Control*, pages 57–62, 1988.

[120] K. H. Tan and M. A. Lewis. "Virtual structures for high-precision cooperative mobile robot control". *Autonomous Robots Journal*, 4:287–403, 1997.

[121] Z. Liu and H. X. Li. "A Probabilistic Fuzzy Logic System for Modeling and Control". *IEEE Transactions on Fuzzy Systems*, 13(6):848–859, 2005.

[122] C. Chen, G. Rigatos, D. Dong, and J. Lam. "Partial Feedback Control of Quantum Systems Using Probabilistic Fuzzy Estimator". *Proceedings of the 48th IEEE Conference on Decision and Control*, pages 3805–3810, 2009.

[123] R. A. Al-Jarrah and H. Roth. "Development of Cooperation Between Flying Robot, Ground Robot and Ground Station with Fuzzy Logic and Image Processing". *University of Siegen*, 2015.

[124] J. R. Jang. "Adaptive network based fuzzy inference system". *IEEE Transactions on Systems, Man and Cybernetics*, pages 665–685, 1993.

[125] R. Andrews and A. B. Tickle. "A survey and critique of techniques for extracting rules from trained artificial neural network". *Knowledge Based Systems*, 8:373–389, 1995.