

ADAPTIVE CONTROLLER DESIGN AND SELF-BALANCING TWO-WHEELED TRANSPORTER (SBTWT)

A DISSERTATION

*Submitted in partial fulfillment of the
requirements for the award of the degree*

of

MASTER OF TECHNOLOGY

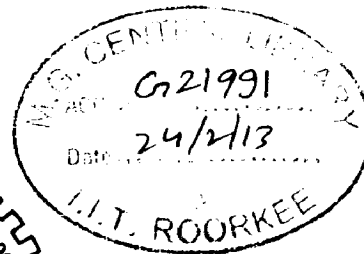
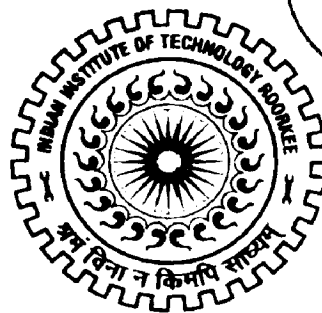
in

ELECTRONICS & COMMUNICATION ENGINEERING

(With Specialization in System Modeling and Control)

By

RAHUL AGRAWAL



**DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY ROORKEE
ROORKEE - 247 667 (INDIA)**

MAY, 2012

CANDIDATE'S DECLARATION

I hereby declare that the work, which is being reported in this dissertation report, entitled “**Adaptive Controller Design and Self-Balancing Two-Wheeled Transporter (SBTWT)**”, is being submitted in partial fulfillment of the requirements for the award of the degree of **Master of Technology in System Modeling and Control**, in the Department of Electronics and Computer Engineering, Indian Institute of Technology, Roorkee is an authentic record of my own work, carried out from June 2011 to May 2012, under guidance and supervision of **Dr. R. Mitra**, Professor Emeritus, Department of Electronics and Computer Engineering, Indian Institute of Technology, Roorkee.

The results embodied in this dissertation have not submitted for the award of any other Degree or Diploma.

Date: 1- Jun-2012

Place: Roorkee



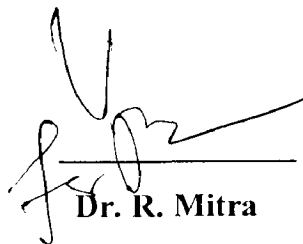
Rahul Agrawal

CERTIFICATE

This is to certify that the statement made by the candidate is correct to best of my knowledge and belief.

Date: 1- Jun-2012

Place: ROORKEE



Dr. R. Mitra

Professor Emeritus

Department of E & CE

Indian Institute of Technology, Roorkee

ACKNOWLEDGEMENT

First and foremost, I would express my gratitude to Dr. R. Mitra Emeritus Professor Department of Electronics and Computer Engineering at Indian Institute of Technology, Roorkee for his invaluable guidance, support and encouragement. Without him, I could not have realized my skill and potential. Interaction and discussion under his guidance helped in building my concepts.

I am deeply obliged to Dr. M.J. Nigam for his moral support. I owe a huge debt of thanks to the faculty of System Modeling and Control group, department of Electronics and Computer Engineering for their technical assistance and constant motivation to carry out my dissertation.

I would like to acknowledge the resources and information provided by IIT Roorkee regarding research work being conducted worldwide on the topic. I thank the Institute authorities for granting me the use of laboratory facilities and other software required for this work.

Rahul Agrawal
M.Tech (SMC)

ABSTRACT

With tremendously increased use of the personal transporter vehicle in comparison of (public transporters), the problems like damage caused by carbon dioxide and other greenhouse gas emissions, and environmental and political forces have de-stabilized the global petroleum supply and fuel price hike are growing day by day. So the personal transporter like SegwayTM is an effective solution, the Segway PT can drastically reduce dependence on foreign oil, pollution, greenhouse gas emissions and substantially increase energy efficiency by replacing short-distance single-occupancy car journeys. Its compact yet robust design makes it suitable for a variety of everyday uses and commercial applications, allowing riders to cover distances which would have previously required the use of a traditional vehicle for the purpose of travelling some miles. Such Self-balancing Two-Wheeled Transporters (SBTWT) which are less expensive in comparison to Segway PT can be built using low-tech, off-the-shelf inexpensive components so that such type of vehicle can be easily used like traditional bicycle or scooter to serve the purpose of human transportation. The controlling of these low cost vehicles in such a way so that they offer sturdy capability in rugged off-sidewalk terrains such as trails, bike paths or beachfronts is an open research problem.

This work contributes to design the controller for self-balancing two-wheeled transporter so that rider can experience the comfortable standing posture and the motion control in riding. In this work firstly a tracking and controlling of the SBTWT using feedback linearization technique is discussed to achieve the self-balancing and the yaw motion control of the vehicle and the performance is examined with system uncertainty, parameter variation and for different riders and comparative study with existing control techniques. Since the self-balancing two-wheeled transporter is an application of the inverted pendulum, an Adaptive Neuro-Fuzzy Inference Structure (ANFIS) controller is designed for Rotary Inverted Pendulum (RIP). The controller and the inverted pendulum are simulated in the Matlab Simulink environment with the help of ANFIS editor GUI and the performance of this controller is shown in comparison to conventional PID and fuzzy logic controller. The problem with the existing controllers is that the control signal is dependent upon the mathematical modeling of the vehicle. An indirect adaptive controller is designed for SBTWT. Since the control signal is not depend upon the mathematical dynamic equation of

the system, the performance of the proposed controller is not affected from environment changes, system parameters changes, uncertainty and disturbances.

CONTENTS

CANDIDATE'S DECLARATION.....	ii
ABSTRACT.....	v
LIST OF FIGURES.....	ix
LIST OF TABLES	xi
INTRODUCTION.....	2
1.1 Importance	3
1.2 System architecture	3
1.3 Objective.....	4
1.4 Literature review	5
1.5 Dissertation organization.....	6
NONLINEAR MATHEMATICAL MODELING OF SBTWT AND CONTROL	7
2.1 Nonlinear mathematical modeling	7
2.2 Linearized State model.....	12
2.3 Tracking and control using Feedback linearization	13
2.3.1 Feedback law for mobile inverted pendulum subsystem	14
2.3.2 Feedback control law for yaw motion control subsystem.....	14
2.3.3 Control results and discussion	15
2.3.4 Trajectory tracking result and discussion	19
ADAPTIVE NEURO-FUZZY INFERENCE STRUCTURE CONTROLLER FOR RIP	22
3.1 Introduction.....	22
3.2 Rotary inverted pendulum model.....	23
3.3 ANFIS Controller.....	26
3.3.1 Simulation Results and Discussion	28
3.3.2 Comparison of ANFIS and conventional PID and fuzzy control.....	30
INDIRECT ADAPTIVE CONTROLLER FOR SBTWT	34

4.1	Subsystems identification using RBFNNs	35
4.1.1	Mobile inverted pendulum subsystem identification	36
4.1.2	Yaw motion subsystem identification	36
4.2	Learning algorithm.....	37
4.3	Indirect adaptive controller design using network inversion.....	38
4.3.1	Indirect adaptive self-balancing controller design using network inversion.....	39
4.3.2	Indirect adaptive Yaw motion controller using network inversion	41
4.4	Simulation results.....	43
4.4.1	Online data generation.....	43
4.4.2	Mobile inverted pendulum subsystem identification	43
4.4.3	Yaw motion subsystem identification	45
4.4.4	Indirect adaptive self-balancing controller	46
4.4.5	Indirect adaptive yaw motion controller	48
	CONCLUSION AND FUTURE SCOPE.....	50
	REFERENCES.....	51
	PUBLICATIONS.....	54

LIST OF FIGURES

Chapter – 1	
Fig.1. 1 (a) Laboratory built personal two wheeled scooter. (b) Segway i2 from Segway Inc.	3
Fig.1. 2 Block diagram of SBTWT control system	4
Fig.1. 3 Block diagram of the SBTWT controller	5
Chapter – 2	
Fig.2. 1 (a) free body diagram of right wheel, (b) Free body diagram of the human transporters [6].	8
Fig.2. 2 Comparison of simulation results of the pitch angle control	16
Fig.2. 3 Comparison of simulation results of the Yaw angle control	17
Fig.2. 4 Comparison of simulation results of the pitch angle control when $f_4=10N$	18
Fig.2. 5 Comparison of simulation results of the Yaw angle control when $f_6=10N$	18
Fig.2. 6 Self-balancing control using feedback linearization when mass of the vehicle is changed to 435kg.	19
Fig.2. 7 Pitch angle tracking of self-balancing Controller (a) Desired and plant pitch angle (b) Desired and plant pitch rate	20
Fig.2. 8 Yaw angle tracking of yaw Controller (a) Desired and plant yaw angle (b) Desired and plant yaw rate	20
Chapter – 3	
Fig.3. 1 Rotary Inverted Pendulum System	24
Fig.3. 2 Simplified model of the rotary inverted pendulum system	24
Fig.3. 3 Block Diagram of Inverted Pendulum System with feedback ANFIS controller	25
Fig.3. 4 Plant and controller block diagram.	26
Fig.3. 5 structure of ANFIS controller	28
Fig.3. 6 Falling angle and voltage applied plot.	29
Fig.3. 7 Desired arm position and arm response of ANFIS controller.	29
Fig.3. 8 Falling pendulum angle of (a) PID (b) efficient fuzzy controller	31
Fig.3. 9 Falling pendulum angle response of (a) PID controller (b) ANFIS controller when mass is changed to 0.85 Kg.	32
Chapter – 4	
Fig.4. 1 Structure of RBFNN model of inverted pendulum subsystem	36

Fig.4. 2 Structure of RBFNN model of yaw motion subsystem	37
Fig.4. 3 Subsystem identification using feed-forward network	38
Fig.4. 4 Indirect adaptive self-balancing controller	40
Fig.4. 5 Indirect adaptive yaw motion controller	42
Fig.4. 6 Mobile inverted pendulum: identification (a) the state θP and (b) the state ωP	44
Fig.4. 7 Yaw motion subsystem: identification (a) the state θy and (b) the state ωy	46
Fig.4. 8 Pitch angle tracking for the indirect adaptive self-balancing controller using network inversion technique (a) the state θP and (b) the state ωP	47
Fig.4. 9 Yaw angle tracking for the indirect adaptive Yaw motion controller using Network inversion technique (a) the state θy and (b) the state ωy	48

LIST OF TABLES

Chapter – 1

Table2. 1 Symbols definition 11

Table2. 2 Performance comparisons specification-wise 21

Chapter – 3

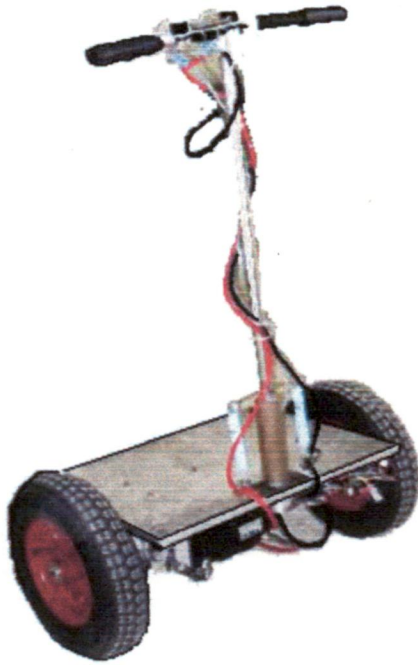
Table3. 1 Values used in the simulation 28

CHAPTER – 1

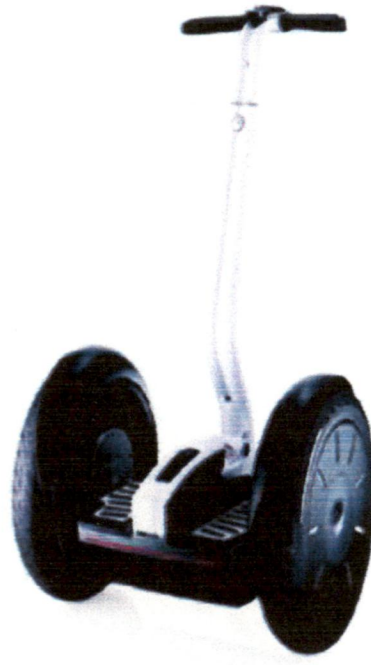
INTRODUCTION

Now days, self-balancing two-wheeled transporters (SBTWT) have become popular for the purpose of personal transportation. SegwayTM [1] has already launched such type of vehicles in the market for local transportation. These kinds of transporters can be constructed by synthesis of mechatronics, control techniques and software. The Segway's scooter is made up of quite high-tech and high-quality dedicated components, such brushless servomotor with neodymium magnets, precision gearbox, nickel metal hydride (NiMH) batteries, silica-based wheels, a digital signal processor as a main controller, motor drivers, six gyroscopes, and several safety accessories. Many researchers [2]-[5] have constructed self-balancing transporter which are less expensive in comparison to Segways's scooter. In order to reduce the cost of vehicle, researchers use low-tech, off-the-shelf inexpensive components. Now the main focus of researchers is to obtain a personal transporter with more reduced cost and sufficient safety guards, so that such type of vehicle can be easily used like traditional bicycle or scooter to serve the purpose of human transportation

This vehicle is application of mobile inverted pendulum like nBOT (a two-wheeled balancing robot) [5] and JOE built by Grasser *et al.* [2], a scaled-down prototype of a digital-signal-processor (DSP) controlled two-wheeled vehicle based on the inverted pendulum with weights attached to it. Such type of systems, inherently non-linear and unstable, can be used as a platform or benchmark to investigate the performance of any linear and nonlinear control methods. However, the inverted pendulum system can neither be used to provide more interesting and versatile applications, nor be adopted to examine the performance of any control systems with multiple inputs and outputs. These shortcomings can be avoided if the low-tech self-balancing two-wheeled transporter is used. Once the system structure and mathematical model of the SBTWT are described and established, the overall system model will be divided into two subsystems, the yaw control subsystem and the inverted pendulum subsystem, thus leading to the design of two kinds of controllers for yaw control and self-balancing.



(a)



(b)

Fig.1. 1 (a) Laboratory built personal two wheeled scooter. (b) Segway i2 from Segway Inc.

1.1 Importance

Such type of systems can serve for the individual, business purpose, patrolling, roaming. These vehicles can help reduce dependence on foreign oil, use the existing energy supply more efficiently, and reduce pollution. They provide stress-free commute, free from public transportations schedule, singular door-to-door transportation solution. This can take you places that a car or bicycle can't, including inside many stores, office buildings, businesses, airports, elevators, and trains.

1.2 System architecture

Fig. 1.2 illustrates the block diagram of the overall SBTWT system along with controller. The DSP based controller, along with built-in A/D converters, is responsible for executing the control algorithms for both self-balancing and yaw control. The feedback signals from the gyroscope and the tilt sensor are used by the controller to maintain the human body on the footplate without falling. The operating principle of the self-balancing control is simply interpreted as; if the rider leans forward, then the vehicle will move forward in order to maintain the rider without falling, and vice versa. The signal taken from the potentiometer is

used in the controller to rotate the yaw axis of the transporter to the desired angle i.e. to rotate the vehicle left or right or any angle that rider intended to achieve.

The signals coming from the sensors are passed through the two first order low-pass filters. These filters process the measured pitch angle rate ω_p from the gyroscope and the measured pitch angle θ_p from one tilt sensor via the first-order low pass filters, thus removing unwanted signals. The potentiometer is adopted to measure the shaft angle of the handlebar and the angular signal is directly measured by the DSP controller with necessary signal processing. Two high torque DC motors were specially chosen not only for their ability to support the load of human riders of various weights, but also for their ability to provide enough power to carry out high-performance driving. The motor driver is used to simultaneously drive both two dc motors by using its PWM input signals.

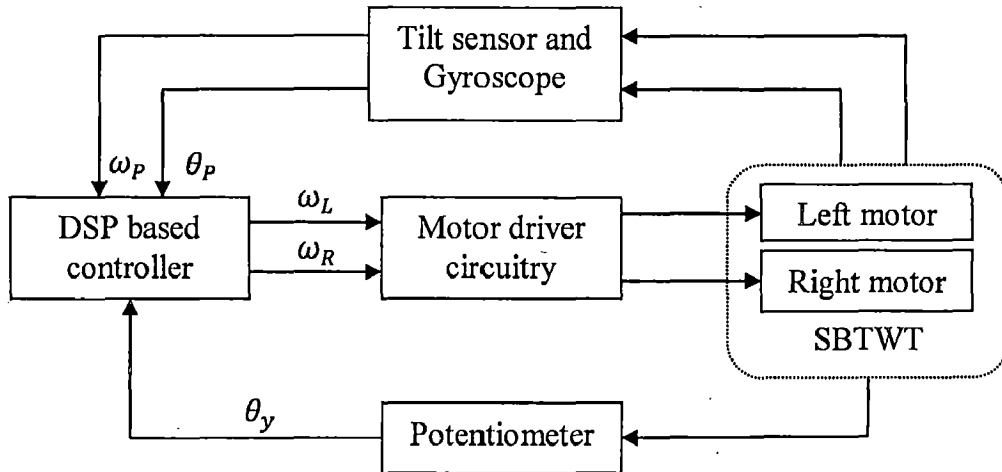


Fig.1. 2 Block diagram of SBTWT control system

1.3 Objective

The problem of controlling and steering the Self-Balancing two-wheel transporter (SBTWT) is a very interesting problem with mass variations and system uncertainties unknown and unmodelled parameters. The objective of controlling the vehicle includes the self-balancing to achieve standing posture maintenance, yaw motion control to steer. Since the overall system model will be divided into two subsystems, the yaw control subsystem and the inverted pendulum subsystem,

This leads to the design of two controllers by different technique.

(a) Yaw controller

(b) Self-balancing controller

Since the potentiometer is employed to measure the angle difference between the equilibrium point and the yaw angle θ_y the rider intended to achieve, the yaw motion control problem is reduced to a regulation problem. The objective of self-balancing controller is to control the pitch angle θ_P and pitch angle rate ω_P to reach the command signal θ_{PC} and ω_{PC} without any error. In the below figure Fig. 1.3 illustrate the overall system of SBTWT with the controller and sensors part.

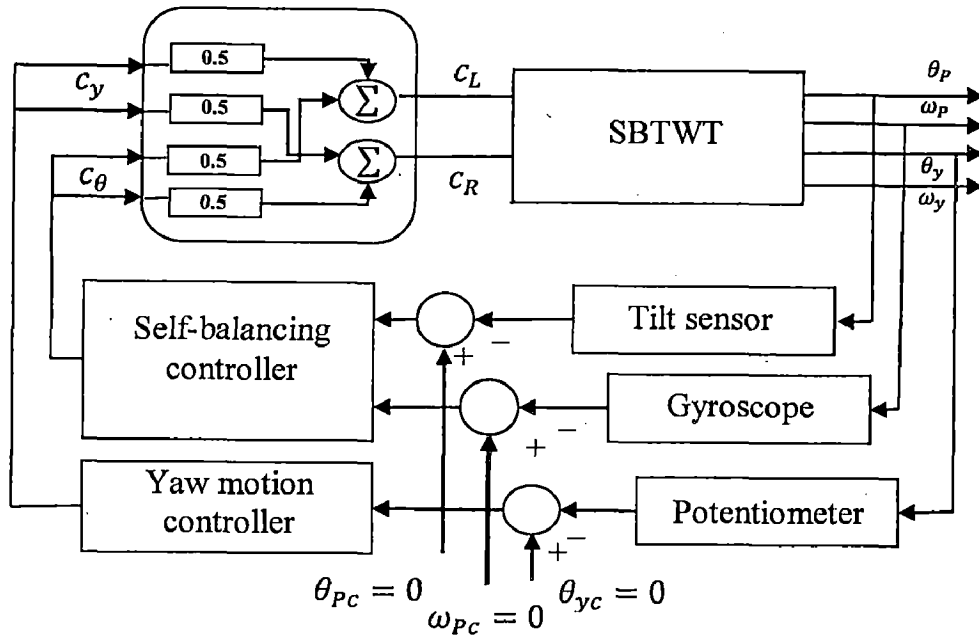


Fig.1. 3 Block diagram of the SBTWT controller

1.4 Literature review

Grasser *et al.* [2] built a scaled-down prototype of a digital-signal-processor (DSP) controlled two-wheeled mobile inverted pendulum and weights attached to the system; however they are test prototypes, aiming at providing several theoretical design and analytical approaches. Tsai *et al.* [3] built a similar two-wheeled transport vehicle in lab and used linearized model to build Proportional plus derivative (PD) controller and design the compensators for the vehicle. In [4] Tsai presents the adaptive neural network controller for linearized two-wheeled transporter estimating the viscous force and the static friction between the wheels and the motion surface using two RBFNNs: however, the method in [4] has not yet considered the fully nonlinear modeling and control problem for this kind of transporter. Blackwell [5] constructed a low-tech, low-cost self-balancing off-the-shelf inexpensive

scooter. Tsai *et al.* in [6] presents the non-linear dynamics of the two-wheeled self-balancing transporter and proposed an adaptive controller using fuzzy basis function and in [7] gives the adaptive controller for the motion control of SBTWT. Pathak *et al.* [8] gives the position and velocity control of mobile inverted pendulum while in [9] Ha *et al.* presents the trajectory tracking of the inverse pendulum mobile robot. Furthermore, several researchers in [10]-[13] have proposed useful control and implementation techniques for four-wheeled vehicles, two-wheeled vehicles with differential driving, and electric scooter. However, these methods cannot be directly applied to the self-balancing and the yaw motion control of the self-balancing two-wheeled transporter.

1.5 Dissertation organization

The rest of the thesis is outlined as follows. In Chapter 2 the nonlinear mathematical modeling of the SBTWT incorporating the friction between the wheels and the motion surface is discussed and the basic controller using feedback linearization is designed for the self-balancing two-wheeled transporter and the performance and the merits of the proposed controller is examined by conducting several simulations. Since the SBTWT is the application of Inverted pendulum, so first in Chapter 3 the adaptive neuro-fuzzy structure based controller is designed for the rotary inverted pendulum system and simulations are conducted to compare the performance in comparison to the conventional PID and fuzzy logic controller. In chapter 4 the indirect adaptive controller is proposed for the SBTWT which is not depends upon dynamic model and architecture of the system and simulation results are discuss the performance of the proposed controller. Chapter-5 concludes the report and gives future research suggestion.

CHAPTER-2

NONLINEAR MATHEMATICAL MODELING OF SBTWT AND CONTROL

In this Chapter firstly the nonlinear mathematical modeling of the SBTWT is derived by incorporating the frictions between the wheels and the motion surface, and modeling error. Such a nonlinear model is constructed based on the Newtonian Mechanics and the force diagram. The linear state model is also presented for SBTWT. In last feedback linearization technique is used for trajectory tracking and controlling of SBTWT in which a nonlinear control law is designed which makes the overall system linear. The performance and merits of this controller is evaluated by conducting various simulations. This is established by changing the system uncertainty terms, and varying the mass on the vehicle and comparison to the state-feedback controller in [2]

2.1 Nonlinear mathematical modeling

The nonlinear mathematical model of the SBTWT is developed in more detailed. Similar to the literature [4], the model can be decomposed into two independent subsystems: inverted pendulum and yaw motion subsystems. In what follows describes the procedure to establish the mathematical model of the SBTWT. In Fig. 2.1 the free body diagram of the right wheel and the free body diagram of human transporter. As shown in Fig. 2.1(a), the static friction force depends on speed and opposites to the moving direction. In order to simplify the derivation of the nonlinear model of the vehicle, one assumes that the wheels of the SBTWT do not slip and Table 2.1 lists all the used symbols and their definitions. By using the Newtonian mechanism, one develops the following set of motion equations to describe the vehicle.

The second Newton's law is used to derive the motion equation of the right wheel

$$M_{RR}\ddot{x}_{RR} = f_{dRR} + H_{TR} - H_R - b\dot{x}_{RR} \quad (2.1)$$

$$M_{RR}\ddot{y}_{RR} = V_{TR} - M_{RR}g - V_R \quad (2.2)$$

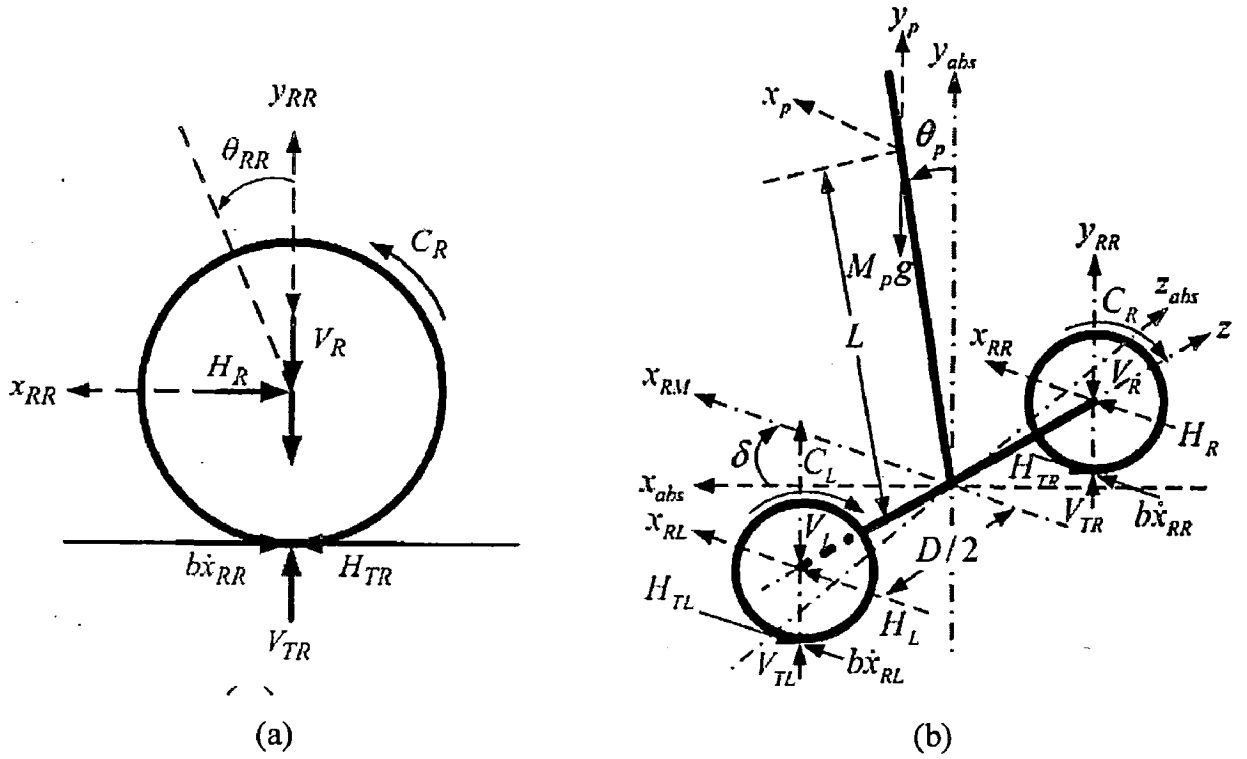


Fig.2. 1 (a) free body diagram of right wheel, (b) Free body diagram of the human transporters [6].

By inertia moment formula as

$$J_{RR}\ddot{\theta}_{RR} = C_R - H_{TR}R \quad (2.3)$$

Similarly the motion equation of the left wheels as

$$M_{RR}\ddot{x}_{RL} = f_{dRL} + H_{TL} - H_L - b\dot{x}_{RL} \quad (2.4)$$

$$M_{RL}\ddot{y}_{RL} = V_{TL} - M_{RL}g - V_L \quad (2.5)$$

$$J_{RL}\ddot{\theta}_{RL} = C_L - H_{TL}R \quad (2.6)$$

Since $\ddot{y}_{RR} = 0$, from (2.2) we have $V_{TR} = M_{RR}g + V_R$. Because $x_{RR} = R\theta_{RR}$, from it follows that

$$M_{RR}R\ddot{\theta}_{RR} = f_{dRR} + H_{TR} - H_R - b\dot{x}_{RR} \quad (2.7)$$

And from (2.3) gives

$$\theta_{RR} = (C_R - H_{TR}R)/J_{RR} \quad (2.8)$$

Substituting (2.8) into (2.7) gives

$$M_{RR}RC_R - \left(\frac{H_{TR}R}{J_{RR}}\right) = f_{dRR} + H_{TR} - H_R - b\dot{x}_{RR} \quad (2.9)$$

$$\frac{M_{RR}R}{J_{RR}}C_R - f_{dRR} + H_R + b\dot{x}_{RR} = \left(\frac{M_{RR}R^3}{J_{RR}} + 1\right)H_{TR} \quad (2.10)$$

$$H_{TR} = [(M_{RR}RC_R/J_{RR}) - f_{dRR} + H_R + b\dot{x}_{RR}]/[(M_{RR}R^2/J_{RR} + 1)] \quad (2.11)$$

Because both wheels are identical, we have

$$H_{TL} = [(M_{RL}RC_L/J_{RL}) - f_{dRL} + H_L + b\dot{x}_{RL}]/[(M_{RL}R^2/J_{RL} + 1)] \quad (2.12)$$

For the chassis, the force in the free body diagram of the chassis in the horizontal direction is described by

$$M_P\ddot{x}_P = f_{dP} + H_L + H_R \quad (2.13)$$

Similarly, in the vertical direction, one obtains:

$$M_P\ddot{y}_P = V_L + V_R - M_Pg \quad (2.14)$$

By the moment formula, we have

$$J_{P\theta}\ddot{\theta}_P = (V_L + V_R)L \sin \theta_P - (H_L + H_R)L \cos \theta_P - (C_L + C_R) \quad (2.15)$$

$$J_{P\delta}\ddot{\theta}_y = (H_L - H_R)D/2 - b_\delta\omega_y \quad (2.16)$$

If $\dot{y}_P = 0$, then $V_R + V_L = M_Pg$. since $x_P = L \sin \theta_P$, $\ddot{x}_P = L\ddot{\theta}_P \cos \theta_P - L\dot{\theta}_P^2 \sin \theta_P$, substituting, \ddot{x}_P into (2.13) and (2.15) we obtain

$$\ddot{\theta}_P = (f_{dP} + H_R + H_L + M_PL\dot{\theta}_P^2 \sin \theta_P)/M_PL \cos \theta_P \quad (2.17)$$

Substituting (2.17) into (2.15) yields

$$\begin{aligned} & [(J_{P\theta}/M_PL \cos \theta_P) + L \cos \theta_P](H_R + H_L) \\ & = -(J_{P\theta}/M_PL \cos \theta_P)f_{dP} - J_{P\theta}\dot{\theta}_P^2 \tan \theta_P \\ & + M_PgL \sin \theta_P - (C_L + C_R) \end{aligned} \quad (2.18)$$

And

$$\begin{aligned} (H_R + H_L) = & \{1/[(J_{P\theta}/M_PL \cos \theta_P) \\ & + (L \cos \theta_P)]\}[-J_{P\theta}f_{dP}/M_PgL \cos \theta_P \\ & - J_{P\theta}\dot{\theta}_P^2 \tan \theta_P + M_PgL \sin \theta_P - (C_R + C_L)] \end{aligned} \quad (2.19)$$

By the definitions in Fig. 2.1 (b), the following equation of motion can be obtained:

$$x_{RM} = (x_{RL} + x_{RR})/2 = R(\theta_{RL} + \theta_{RR})/2 \quad (2.20)$$

$$\dot{v}_{RM} = \ddot{x}_{RM} = R(\ddot{\theta}_{RL} + \ddot{\theta}_{RR})/2 \quad (2.21)$$

From (2.3) and (2.6), it yields $\theta_{RR} = (C_R - H_{TR}R)/J_{RR}$, $\theta_{RL} = (C_L - H_{TL}R)/J_{RL}$. Thus one obtains

$$v_{RM} = (R/2)\{[(C_L - H_{TL}R)/J_{RL}] + (C_R - H_{TR})/J_{RR}\} \quad (2.22)$$

Assuming (2.11) and (2.12) into (2.21), and then equating the result to (2.21) yield

$$\begin{aligned} \dot{v}_{RM} = & A_{22}v_{RM} + A_{23} \sin \theta_P + B_2(C_L + C_R) + B_{23}f_{dRL} \\ & + B_{24}f_{dRR} + B_{25}f_{dP} \end{aligned} \quad (2.23)$$

where $A_{22} = -R^2b/\alpha J_R$, $B_2(R/2J_R)[(1/\alpha) + (R/\alpha\beta)]$,

$$A_{23} = -(R^2)(\gamma - (\dot{\theta}_P^2/\cos \theta_P))/2J_R\alpha\beta,$$

$$B_{23} = B_{24} = R^2/2J_R\alpha,$$

$$\alpha = (M_R R^2/J_R) + 1,$$

$$B_{25} = (R^2/2J_R\alpha\beta)(J_{P\theta}/M_P L \cos \theta_P),$$

$$p = 1 + (D^2 \cdot (J_R + M_R R^2)/2J_{P\theta} R^2),$$

$$\beta = (J_{P\theta}/M_P L \cos \theta_P) + L \cos \theta_P,$$

$$\gamma = LM_P g.$$

Next, derive the dynamics of the pitch angle. Because $\dot{\theta}_P = \omega_P$ from (2.15) gives

$$\begin{aligned} \omega_P = \dot{\theta}_P = & \{[(L \sin \theta_P M_P g - L \cos \theta_P)J_{P\theta}]/\beta\}\{-J_{P\theta} \\ & /M_P L \cos \theta_P\}f_{dP} - J_{P\theta}\dot{\theta}_P^2 \tan \theta_P + \gamma \sin \theta_P \\ & - (C_R + C_L)\} - (C_L + C_R)/J_{P\theta} \end{aligned} \quad (2.24)$$

which, consequently, becomes

$$\dot{\omega}_P = A_{43} \sin \theta_P + B_4(C_L + C_R) + B_{45}f_{dP} \quad (2.25)$$

where $A_{43} = (\gamma/J_{P\theta}) - [L(\gamma - J_{P\theta}\dot{\theta}_P^2)/J_{P\theta}\beta]$,

$B_{45} = 1/\beta M_P$, $B_4 = [(L \cos \theta_P/\beta) - 1]/J_{P\theta}$. Furthermore, from (2.16), it follows

$$\ddot{\theta}_y = \dot{\omega}_y = [D(H_L - H_R)/2J_{P\delta}] - [b_\delta \omega_y/J_{P\delta}] \quad (2.26)$$

$$\omega_y = (v_{RL} - v_{RR})/D \quad (2.27)$$

Subtracting (2.4) from (2.1), one obtains

$$\begin{aligned} H_{TL} - H_{TR} = & [(M_R R/J_R)(C_L - C_R) + (f_{dRR} - f_{dRL}) + (H_L \\ & - H_R) + b(\dot{x}_{RL} - \dot{x}_{RR})]/\alpha \end{aligned} \quad (2.28)$$

And rearranging (2.26) give

$$\begin{aligned} H_L - H_R = & (f_{dRR} - f_{dRL}) + (C_L - C_R)/R - bD\omega_y - [(J_R \\ & + M_R R^2)/R^2]D\dot{\omega}_y \end{aligned} \quad (2.29)$$

Therefore, solving (2.25) for $H_L - H_R$ and substituting it into (2.27), we have

$$\dot{\omega}_y = A_{66}\omega_y + B_{61}C_L + B_{62}C_R + B_{63}f_{dRL} + B_{64}f_{dRR} \quad (2.30)$$

where $A_{66} = -(bD^2 + 2b)/2pJ_{P\delta}$,

$B_6 = B_{63} = -B_{64} = -D/2pJ_{P\delta}$,

$B_{61} = -B_{62} = D/2pJ_{P\delta}R$. Define the following six state variables

$$X = [x_{RM} \quad v_{RM} \quad \theta_p \quad \omega_p \quad \theta_y \quad \omega_y]^T \quad (2.31)$$

The nonlinear system model of the human transporter is then described in the following state-space form

$$\begin{bmatrix} \dot{x}_{RM} \\ \dot{v}_{RM} \\ \dot{\theta}_p \\ \dot{\omega}_y \\ \dot{\theta}_y \\ \dot{\omega}_y \end{bmatrix} = \begin{bmatrix} v_{RM} \\ A_{22}v_{RM} + A_{23} \sin \theta_p \\ \omega_p \\ A_{43} \sin \theta_p \\ \omega_y \\ A_{66}\omega_y \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ B_2 & B_2 \\ 0 & 0 \\ B_4 & B_4 \\ 0 & 0 \\ -B_6 & B_6 \end{bmatrix} \begin{bmatrix} C_L \\ C_R \end{bmatrix} + \begin{bmatrix} 0 \\ \bar{f}_2 \\ 0 \\ \bar{f}_4 \\ 0 \\ \bar{f}_6 \end{bmatrix} \quad (2.32)$$

where $f_2 = B_{23}f_{dRL} + B_{24}f_{dRR} + B_{25}f_{dP}$, $f_6 = B_{63}f_{dRL} + B_{64}f_{dRR}$, $f_4 = B_{45}f_{dP}$.

From (2.30), assuming $f_i = 0$ for $i = 2, 4, 6$, and if the human transporter has reached steady-state inclination θ_p and $\dot{\theta}_p = 0$, then it will generate a steady-state linear speed $v_{RM_{ss}}$

$$v_{RM_{ss}} = (B_2A_{43} - A_{23}B_6) \sin \theta_p / A_{22}B_6 |_{\dot{\theta}_p=0}$$

In order to move the human transporter without falling down, transforming C_y and C_θ into the wheel torques C_L and C_R yields

$$C_L = 0.5C_\theta + 0.5C_y, C_R = 0.5C_\theta - 0.5C_y$$

Thus (2.32) can be decomposed into two independent subsystems: One is concerned with mobile inverted pendulum subsystem describing the yaw motion about the z axis i.e.

$$\begin{bmatrix} \dot{x}_{RM} \\ \dot{v}_{RM} \\ \dot{\theta}_p \\ \dot{\omega}_y \end{bmatrix} = \begin{bmatrix} v_{RM} \\ A_{22}v_{RM} + A_{23} \sin \theta_p \\ \omega_p \\ A_{43} \sin \theta_p \end{bmatrix} + \begin{bmatrix} 0 \\ B_2(C_\theta + \bar{f}_2) \\ 0 \\ B_4(C_\theta + \bar{f}_4) \end{bmatrix}, \bar{f}_i = \frac{f_i}{B_i}, i = 2, 4 \quad (2.33)$$

And other is yaw motion control subsystem modeling the rotation about the y-axis, i.e.

$$\begin{bmatrix} \dot{\theta}_y \\ \dot{\omega}_y \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & A_{66} \end{bmatrix} \begin{bmatrix} \theta_y \\ \omega_y \end{bmatrix} + \begin{bmatrix} 0 \\ B_6 \end{bmatrix} [C_y + \bar{f}_6], \bar{f}_6 = \frac{f_6}{B_6} \quad (2.34)$$

From (2.5) and (2.6), it is clear that two controllers for C_θ and C_y can be synthesized independently from each other and combine together to achieve the control goal.

Table2. 1 Symbols definition

Symbol and unit	PARAMETER AND VARIABLE NAME	Value
x_{rm} [m] v_{rm} [m]	Position and speed	–
θ_p [rad] ω_p [rad/sec]	Pitch angle and pitch rate	–
θ_y [rad] ω_y [rad/sec]	Yaw angle and yaw rate	–
C_L [N.m] C_R [N.m]	Applied torque on left wheel Applied torque on right wheel	–
J_{RR} [Kg.m ²] J_{RL} [Kg.m ²]	Moment of inertia of the rotation mass with respect to the Z axis	0.1Kg.m ²
M_{RR} [Kg] M_{RL} [Kg]	Mass of the rotation mass connected to the left and right wheel	–
$J_{P\theta}$ [Kg.m ²]	Moment of inertia of the chassis with respect to the Z axis	27.6 Kg.m ²
$J_{P\delta}$ [Kg.m ²]	Moment of inertia of the chassis with respect to the Y axis	3.478 Kg.m ²
f_{aRL} (f_{aRR})	Force applied to the center of the left (right) wheels	–
f_{dp}	Force applied to the center of gravity	–
M_P [Kg]	Mass of the chassis	135
R [m]	Radius of the wheels	0.2m
D [m]	Lateral distance between the contact patches of the wheels	0.6m
L [m]	Distance between the Z axis and the CG of the chassis	1m
b	Friction coefficient	0.01~0.05

2.2 Linearized State model

Assume that θ_p be small, i.e., $y_p = L \cos \theta_p \approx L$, $\dot{y}_p = 0$. Using the linearization procedure, one obtains the linearized system model of the SBTWT in the following state-space form:

$$\begin{bmatrix} \dot{x}_{RM} \\ \dot{v}_{RM} \\ \dot{\theta}_P \\ \dot{\omega}_y \\ \dot{\theta}_y \\ \dot{\omega}_y \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & A_{22} & A_{23} & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & A_{43} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & A_{66} \end{bmatrix} \begin{bmatrix} x_{RM} \\ v_{RM} \\ \theta_P \\ \omega_P \\ \theta_y \\ \omega_y \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ B_2 & B_2 \\ 0 & 0 \\ B_4 & B_4 \\ 0 & 0 \\ -B_6 & B_6 \end{bmatrix} \begin{bmatrix} C_L \\ C_R \end{bmatrix} + \begin{pmatrix} 0 \\ f_2 \\ 0 \\ f_4 \\ 0 \\ f_6 \end{pmatrix} \quad (2.35)$$

2.3 Tracking and control using Feedback linearization

Feedback linearization is the technique in which the closed loop dynamics of the non-linear system made the linear through suitable control law. In fact, there exists a class of non-linear system for which feedback linearization is possible.

Considering a class of single input affine non-linear system defined as

$$\frac{d}{dt} \begin{bmatrix} x_1 \\ x_2 \\ \dots \\ x_{n-1} \\ x_n \end{bmatrix} = \begin{bmatrix} x_2 \\ x_3 \\ \dots \\ x_n \\ f(x) + g(x)u \end{bmatrix} \quad (2.36)$$

$$y = x_1$$

Assume that $g(x) \neq 0$. if the control law is chosen

$$u = \frac{1}{g(x)} [-f(x) + k_v r + \lambda_1 e^{(n-1)} + \dots + \lambda_{n-1} e^{(1)} + \dots + x_{nd}] \quad (2.37)$$

where $e = y^d - y$ is the output tracking error and $r = e^{(n-1)} + \lambda_1 e^{(n-2)} + \dots + \lambda_{n-1} e$ (power denotes respective derivatives).

The closed loop error dynamics becomes $\dot{r} = -k_v r$ which is linear as well as stable. where k_v and λ s are positive design parameters.

The system is decomposed into two subsystem i.e. mobile inverted pendulum subsystem and yaw motion control subsystem. This leads to driving the independent control law for both the two subsystem by using the law (2.37), so this made the overall closed loop dynamics of the two subsystems linear. This results two controller: self-balancing controller and yaw motion controller

2.3.1 Feedback law for mobile inverted pendulum subsystem

The inverted pendulum subsystem described in (2.33) can be written as

$$\begin{bmatrix} \dot{\theta}_p \\ \dot{\omega}_p \end{bmatrix} = \begin{bmatrix} \omega_p \\ A_{43} \sin \theta_p \end{bmatrix} + \begin{bmatrix} 0 \\ B_4(C_\theta + \bar{f}_4) \end{bmatrix}, \bar{f}_4 = \frac{f_4}{B_4} \quad (2.38)$$

The control objective is to control the pitch angle and pitch rate of the mobile inverted pendulum subsystem in order to follow the command signal $\theta_{pc} = 0, \omega_{pc} = 0$ without any error from Fig. 1.3. Therefore SBTWT will remain stable at vertical upright position and rider can stand on the footplate of the vehicle without falling. For deriving the control law for this mobile subsystem the nonlinear term \bar{f}_4 in (2.38) is assumed to be zero and the control law is derived from (2.37).

From (2.38) and (2.36) the $x_1 = \theta_p, x_2 = \omega_p, f(x) = A_{43} \sin \theta_p$ and $g(x) = B_4$ and for the control law.

$$\begin{aligned} y^d &= \theta_{pc} = 0 \text{ and } e = -y = -\theta_p \text{ and } x_{nd} = y^{nd} = 0 \\ r &= e^{(1)} + \lambda_1 e \end{aligned} \quad (2.39)$$

Substitution of (2.39) into the control law (2.37) which yields the control law is to be

$$c_\theta = \frac{1}{B_4} [-A_{43} \sin \theta_p + k_v r + \lambda_1 e^{(1)}] \quad (2.40)$$

Applying the control law (2.40) to the subsystem (2.38) the error dynamics becomes $\dot{r} = -k_v r$. While k_v and λ_s are the positive constants, selection of the appropriate values of these tuning parameter ensures the convergence of θ_p, ω_p and error dynamics as well. This can make the overall subsystem stable and follow the command signal.

2.3.2 Feedback control law for yaw motion control subsystem

The control law for yaw motion control of the subsystem (2.34). Since the potentiometer is employed to measure the angle difference between the equilibrium point and yaw angle the rider intended to achieve, the yaw motion control is designed such that the yaw angle of the vehicle is to follow command signal given by the rider. Similarly this control law is derived from the feedback linearization technique (2.37) so that the closed loop dynamics of the system becomes linear, while deriving the control law the nonlinear term \bar{f}_6 is assumed to be zero in (2.34).

From (2.34) and (2.36) the $x_1 = \theta_y, x_2 = \omega_y, f(x) = A_{66}\omega_y$ and $g(x) = B_6$ and for the control law

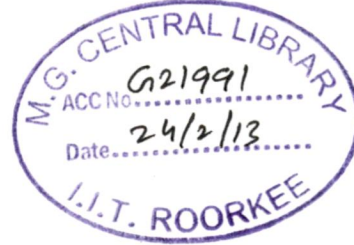
$$y^d = \theta_{yc} = 0 \text{ and } e = -y = -\theta_y \text{ and } x_{nd} = y^{nd} = 0$$

$$r = e^{(1)} + \lambda_1 e \quad (2.41)$$

Substitution of (2.41) into the control law (2.37) which yields the control law is to be

$$c_y = \frac{1}{B_6} [-A_{66}\omega_y + k_v r + \lambda_1 e^{(1)}] \quad (2.42)$$

This control signal (12) is applied to the subsystem (6), substitution of this control law in (6) gives the closed loop linear dynamics in r . i.e. $\dot{r} = k_v r$, where k_v and λ_s are the positive constants, these parameters can be calculated from pole placement and this ensures the convergence of the r, θ_y . As a result of this the yaw angle is follow the riders command yaw angle θ_y .



2.3.3 Control results and discussion

In this section two sets of experiments for each subsystem (self-balancing and yaw motion control subsystems) are conducted to show the effectiveness and performances of the proposed controller using feedback linearization technique. Table I lists the all parameters used for the simulation, two uncertain terms f_4 and f_6 are taken to be nonzero quantity but the viscous friction depends upon the vehicle speed and slip of the wheels are neglected. In the first set the uncertainty terms is assumed to be fixed quantity $f_4 = f_6 = 0.5$ N while in second set of simulation the uncertain terms f_4 and f_6 are increase to 10 N.

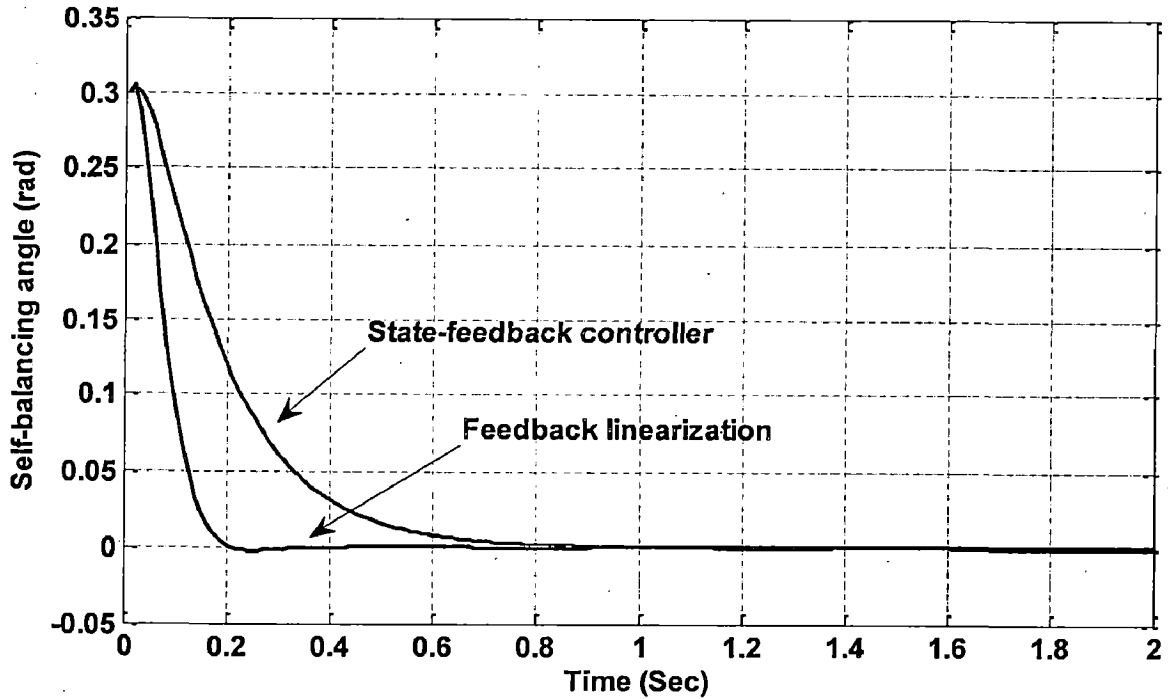


Fig.2. 2 Comparison of simulation results of the pitch angle control

The first simulation of first set is performed to verify the performance of the proposed feedback linearization controller for self-balancing controller. Fig. 2.2 shows the comparative simulation results of the pitch angle tracking for the state-feedback controller [2] and the feedback linearization control. As the Fig. 2.2 shows the feedback linearization control has an excellent convergent speed than that of the state feedback controller when the uncertain term f_4 is taken to be 0.5N. The second simulation of first set is conducted to examine the effectiveness of the proposed controller using feedback linearization for yaw motion control subsystem. Fig 2.3 shows the yaw angle tracking performance for the state-feedback controller and proposed yaw controller using feedback linearization. The results in Fig. 2.2 and 2.3 shows that the proposed feedback linearization controller has far better performance compare to the conventional state-feedback controller. In contrast Fig 2.2 and 2.3 have shown that the proposed controller outperforms the state-feedback controller.

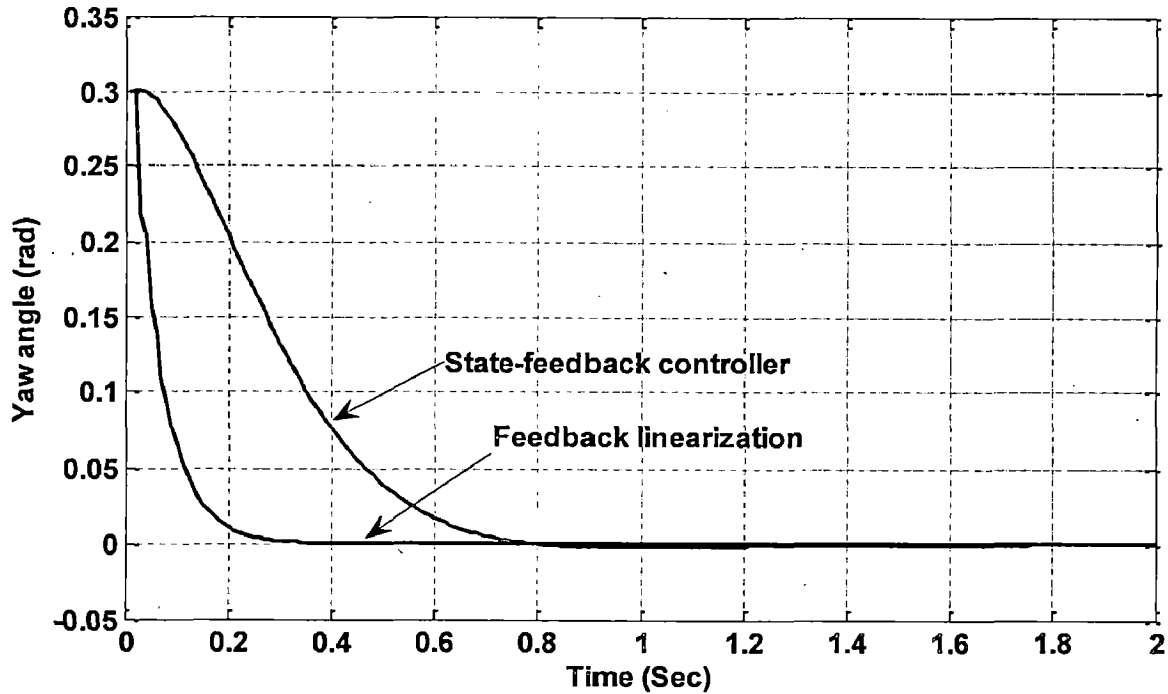


Fig.2. 3 Comparison of simulation results of the Yaw angle control

In the second set of experiment, the simulation of feedback linearization controller and state-feedback controller are designed with the same tuning parameters while the uncertain term f_4 and f_6 is increase to 10N for both the self balancing and yaw motion control subsystem. Fig. 2.4 shows the performance of both feedback linearization and state-feedback controller for self-balancing subsystem. From the Fig. 2.4 it can be stated that feedback linearization controller works fine with negligible steady state error 0.02 *rad* (1.15°) on other side state-feedback controller gives the large steady state error 0.07 *rad* (4°). In case of yaw motion control subsystem Fig. 2.5 shows the responses of both feedback linearization and state-feedback controller. From Fig. 2.5 shows that performance of feedback linearization controller is in limit and gives the steady state error of 0.02 *rad* (1.15°) while the state-feedback controller totally fails and gives the steady state error of 0.27 *rad*(15.4°). In a result Fig. 2.4 and 2.5 have shown that the proposed controller totally outperformance the state-feedback controller when the uncertainty terms are changed to 10N.

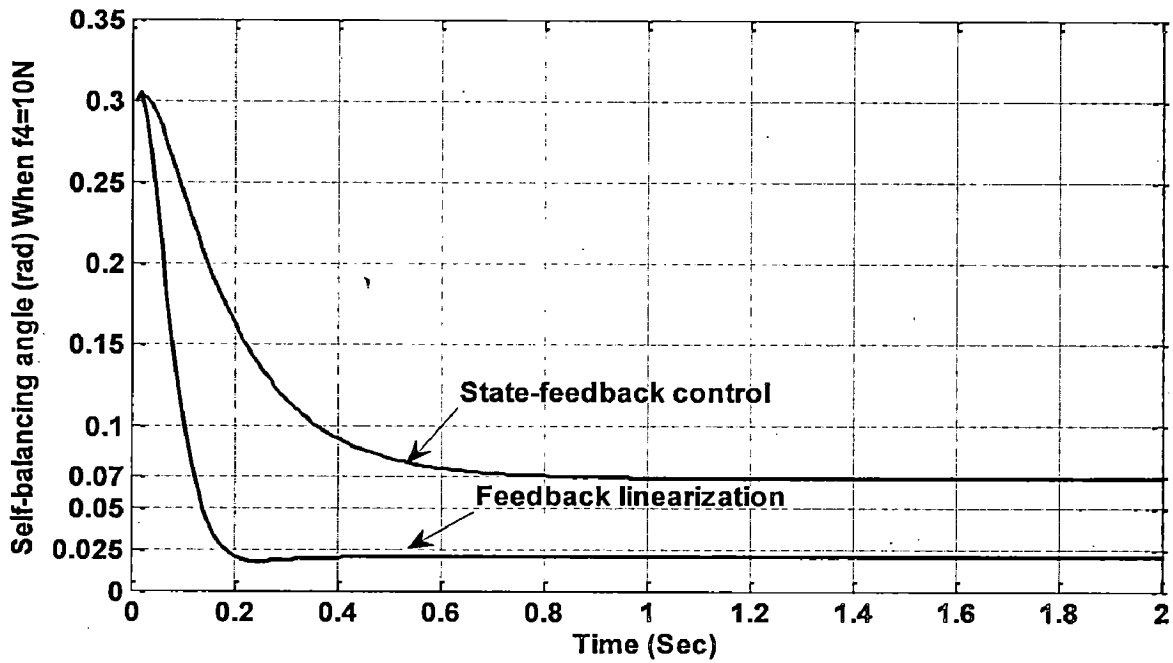


Fig.2. 4 Comparison of simulation results of the pitch angle control when $f_4=10N$

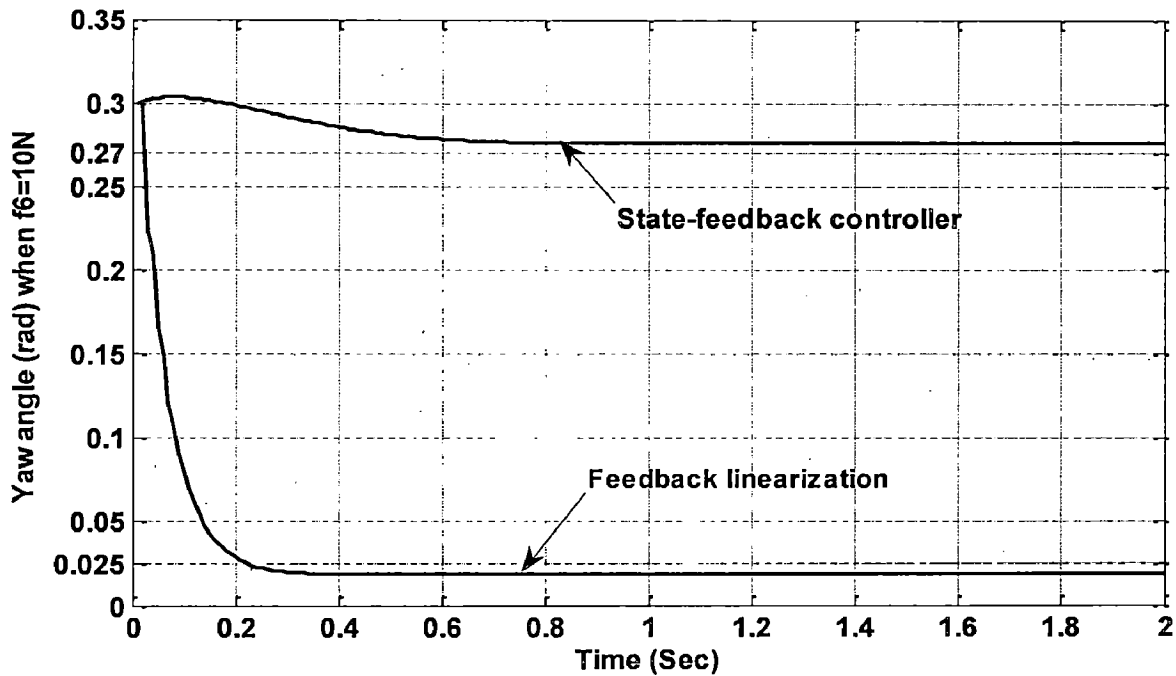


Fig.2. 5 Comparison of simulation results of the Yaw angle control when $f_6=10N$

Now the performance of the controller is evaluated in system parameter variation. The mass of the vehicle is changed for the different rider from 135 kg. to 435 kg. Since the dynamics of the inverted mobile pendulum is dependent on the mass of the vehicle M_p so the performance get affected. While yaw motion subsystem is independent to the vehicle mass, the performance of it remains unchanged so overall the performance of controller is get affected.

The response of the self-balancing controller is shown in Fig. 2.6 with mass variation. From figure it is clear that the settling time is increased to 0.7 sec and overshoot is 0.06.

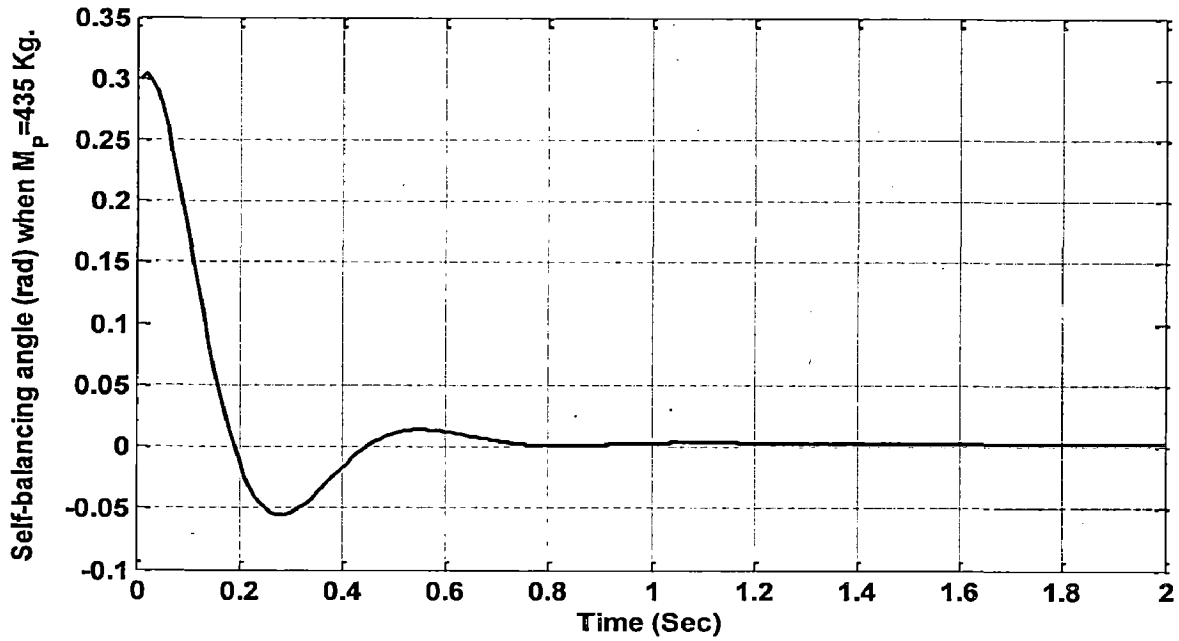
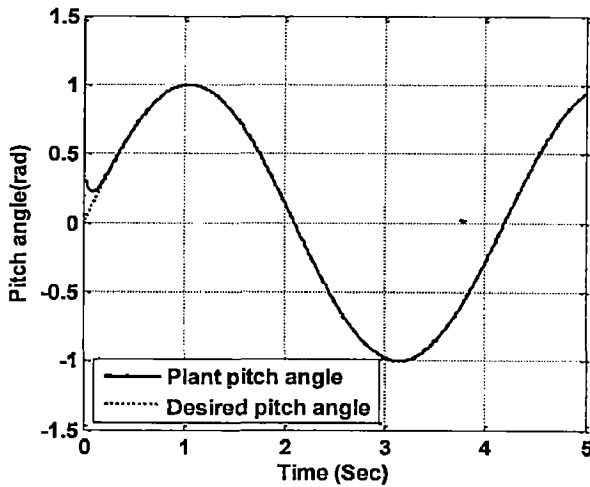


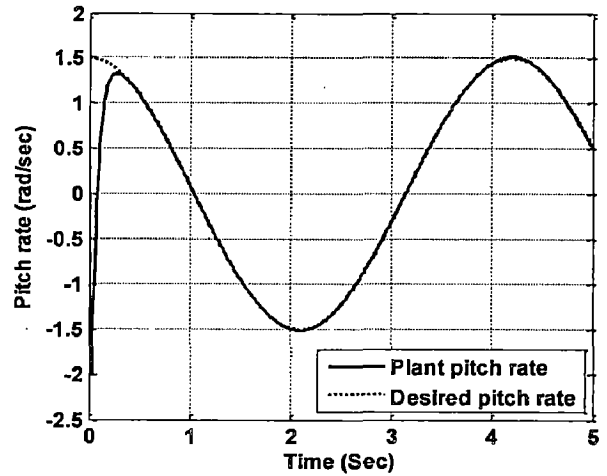
Fig.2. 6 Self-balancing control using feedback linearization when mass of the vehicle is changed to 435kg.

2.3.4 Trajectory tracking result and discussion

For tracking of desired trajectory, the control is again derived from the (2.37). The desired trajectory is chosen for the self-balancing controller is $\sin 1.5t$. The response of the mobile inverted pendulum subsystem is shown in the Fig. 2.7. The Fig. 2.7 (a) is shown the self-balancing angle i.e. pitch angle response and the desired trajectory and Fig. 2.7 (b) is shown the pitch rate response of the subsystem and the desired pitch rate trajectory. Form the figure it is clear that the subsystem is starts tracking the desired trajectory in short time and then exactly track it. Similarly for the yaw motion subsystem, for tracking the desired trajectory the control law is again calculated from (2.37) and applied to the subsystem (2.34).



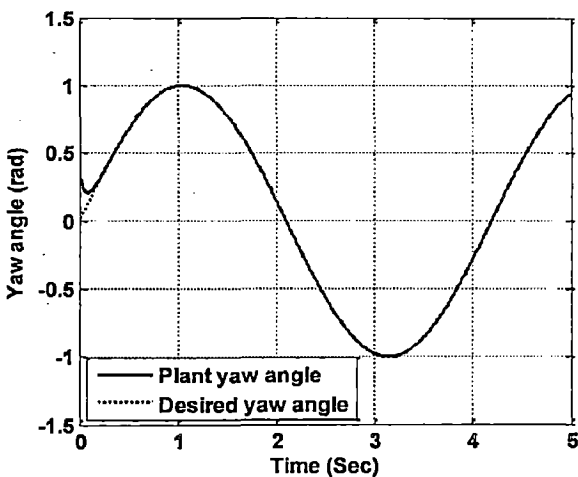
(a)



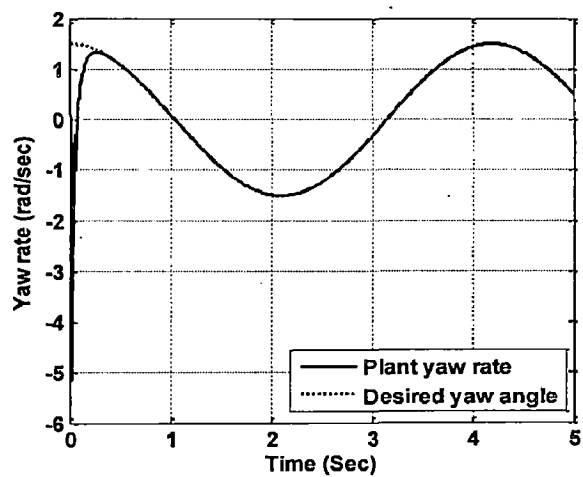
(b)

Fig.2. 7 Pitch angle tracking of self-balancing Controller (a) Desired and plat pitch angle (b) Desired and plant pitch rate

The desired trajectory is selected for this subsystem is $\sin 1.5t$. The response of the yaw subsystem is shown in Fig. 2.8. In Fig. 2.8 (a) the yaw angle response and desired yaw angle trajectory is shown and in Fig. 2.8 (b) the yaw rate response and desired yaw rate signal is shown. From the figure it is clear that the subsystem is starts tracking the desired trajectory in short time and then exactly track it.



(a)



(b)

Fig.2. 8 Yaw angle tracking of yaw Controller (a) Desired and plat yaw angle (b) Desired and plant yaw rate

In Table 2.2 the performance comparison between feedback linearization and state-feedback controller is shown. From the table it can be concluded the feedback linearization controller totally outperformance the state-feedback controller

Table2. 2 Performance comparisons specification-wise

Controller Specification	State-feedback controller		Feedback linearization	
	Yaw motion Controller	Self- balancing controller	Yaw motion Controller	Self- balancing controller
Settling time (sec)	0.8	0.8	0.25	0.2
Steady-state error (<i>rad</i>) When $f_4 = f_6 = 0.5N$	0	0	0	0
Steady-state error (<i>rad</i>) When $f_4 = f_6 = 10N$	0.28	0.07	0.024	0.024

In this chapter of the report the nonlinear mathematical equation of the self-balancing two-wheeled transporter is derived by incorporating the frictions between the wheels and the motion surface, and modeling error. Tracking and control of this vehicle is effectively achieved through the feedback linearization control and the performance of this controller is far better in comparison to state-feedback controller [2] and adaptive controller [3]. Since SBTWT is an application to Inverted Pendulum (IP), in next Chapter an adaptive neuro-fuzzy inference structure based controller is designed for the rotary inverted pendulum.

CHAPTER – 3

ADAPTIVE NEURO-FUZZY INFERENCE STRUCTURE CONTROLLER FOR RIP

SBTWT can be seen as an application of the inverted pendulum. In linear motion of the vehicle the rider can be assumed as a pendulum at its uprights position and wheels with the footplate can be assumed as the cart to this inverted pendulum while in rotation of vehicle the system is assumed to be a rotary inverted pendulum. So in this Chapter an adaptive neuro fuzzy inference (ANFIS) controller is present for the Rotary inverted pendulum to balance it at it's the up-right position. The steps for implementation of four input controller is presented and shown that designing of this controller is very simple and at the same time it reduces the time and space complexity of the controller. The controller and the inverted pendulum are simulated in the Matlab Simulink environment with the help of ANFIS editor GUI. Simulation result shows that ANFIS controller is much better in comparison to conventional PID and Fuzzy logic controller in terms of settling time, overshoot and parameter variation.

3.1 Introduction

A mechanical system which has greater number of joints than the number of actuator present in the system such system is called the underactuated system [34]. Because of this, the strategies developed for fully actuated system may not be directly applied to underactuated system. The control study of underactuated system has drawn a great interest in last few decades as most of the physical systems have underactuated dynamics as those in robotics, aerospace engineering and marine engineering including the example of flexible-link robots, walking robots acrobatic robots, helicopter, satellite, space robot, spacecraft etc.

The Rotary Inverted Pendulum is a widely investigated nonlinear system due to its property of unstable, higher order, multi-variable and highly coupled which can be treated as highly non-linear control system. Rotary system provides an excellent experimental platform for examining specific control theories or typical solution and thus promoting the development of new theories. This system can be taken as the problem of balancing the pendulum at up-right position which is the most common issue in robotics. This explains the fact that many investigations have been carried out on the rotary inverted pendulum problem

[34]-[37].

For control the balancing act of the rotary pendulum, a control system is needed. As known the ANFIS can be used as controller as it can model the human decision making based on the IF-THEN rules and become a very popular tool for the approximation and inexpensive tool to implement and shows the adaptive and robust behavior in comparison to more commonly used conventional controller like PID and compensator like lead-lag and fuzzy controller. As known the conventional controller completely relies on the mathematical model of underlying system while efficient fuzzy controller, designed with the help of LQR parameters can be easily implemented to linear and nonlinear systems [36].

In ANFIS, fuzzy inference system is blended with the neural networks and uses the human intelligence to design the controller. The rotary inverted pendulum and controller is model before putting them into the simulation and controller is train, validate and check the performance with the noise and varying parameter. Then the controller and system model is implemented in Simulink environment of MATLAB and the performance of the controller is measured and run in real-time workshop. This is followed by the implementation and comparison of the PID and fuzzy controller with ANFIS controllers through simulations.

3.2 Rotary inverted pendulum model

The Rotary system, as shown in Fig. 3.1, consists of a vertical pendulum, a horizontal arm, a gear chain, and a servomotor which drives the pendulum through the gear transmission system. The rotating arm is mounted on the output gear of the gear chain. An encoder is attached to the arm shaft to measure the rotating angle of the arm. At the end of the rotating arm there is a hinge instrumented with an encoder. The pendulum is attached to the hinge.

The inverted pendulum (mechanical part only) is sketched in Fig 3.2, α and θ are employed as the generalized coordinates to describe the inverted pendulum system. The pendulum is displaced with a given α while the arm rotates an angle of θ . We assume the pendulum to be a lump mass at point B which is located at the geometric center of the pendulum. The x, y, z frame is fixed to the arm at point A .

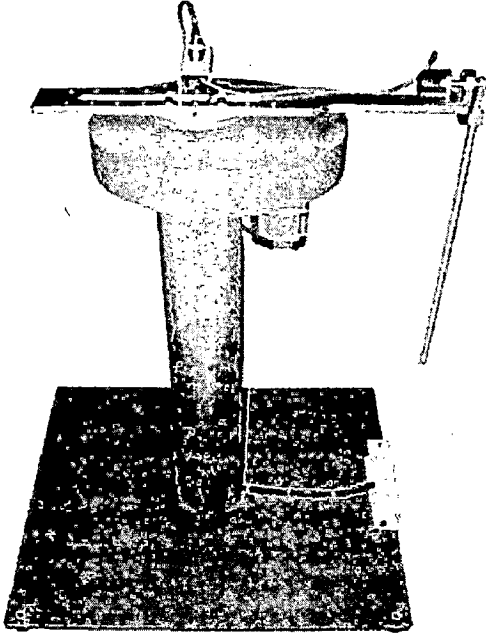


Fig.3. 1 Rotary Inverted Pendulum System

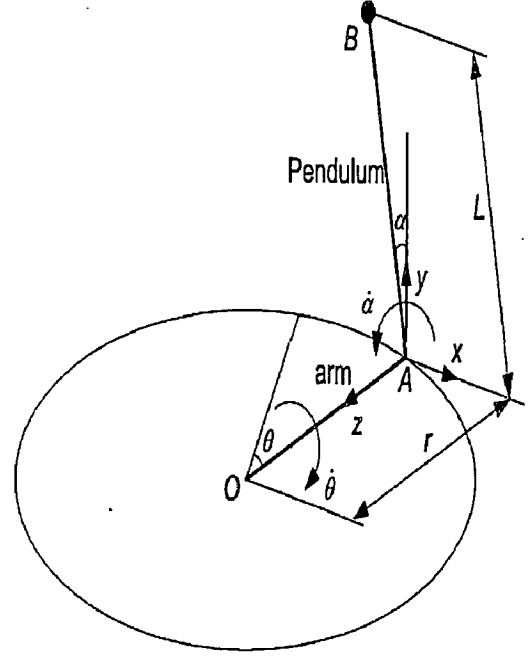


Fig.3. 2 Simplified model of the rotary inverted pendulum system

Non-linear dynamic equation

$$a\ddot{\theta} - b \cos(\alpha) \ddot{\alpha} + b \sin(\alpha) \dot{\alpha}^2 + e\dot{\theta} = fV_m \quad (3.1)$$

$$-b \cos(\alpha) \ddot{\theta} + c\ddot{\alpha} - d \sin(\alpha) = 0 \quad (3.2)$$

Where $a = J_{eq} + mr^2 + \eta_g K_g^2 J_m$, $b = mLr$, $c = \frac{4}{3}ml^2$, $d = mgl$,

$e = B_{eq} + \frac{\eta_m \eta_g K_t K_g^2 K_m}{R_m}$, $f = \frac{\eta_m \eta_g K_t K_g}{R_m}$, J_{eq} is moment of inertia of pendulum and arm about the axis of θ and η_m and η_g are the motor and gear efficiency respectively. K_g , K_m , K_t are the servo system gear ratio, back-emf constant and motor torque constant respectively.

Linearizing (3.1, 3.2) under the assumption that $\alpha \approx 0$ and $\dot{\alpha} \approx 0$, we get the linearized model as follows:

$$a\ddot{\theta} - b\ddot{\alpha} + e\dot{\theta} = fV_m \quad (3.3)$$

$$-b\ddot{\theta} + c\ddot{\alpha} - d\alpha = 0 \quad (3.4)$$

The overall block diagram of a rotary pendulum system with a feedback ANFIS control block is shown in Fig. 3.3. The output of the plant $(\theta, \dot{\theta}, \alpha, \dot{\alpha})$ is fed back to the controller to produce subsequent amount of voltage to balance the pendulum to its up-right position and at the same time maintaining the arm at the initial position.

The model of the inverted pendulum and the controller is created using Simulink. As a whole the algorithm for the controller to balance the pendulum at up-right position is to

calculate the voltage which needs to give to the servomotor. Fig. 3.4 shows how the voltage is calculated from the pendulum angle, pendulum angle acceleration, arm angle, arm angle acceleration measured from their respected sensor. The Fig. 3.4 shows that motor shaft encoder gives the arm angle while another shaft encoder placed at the end of the arm gives the pendulum angle and then the angle accelerations are derived from arm and pendulum angle.

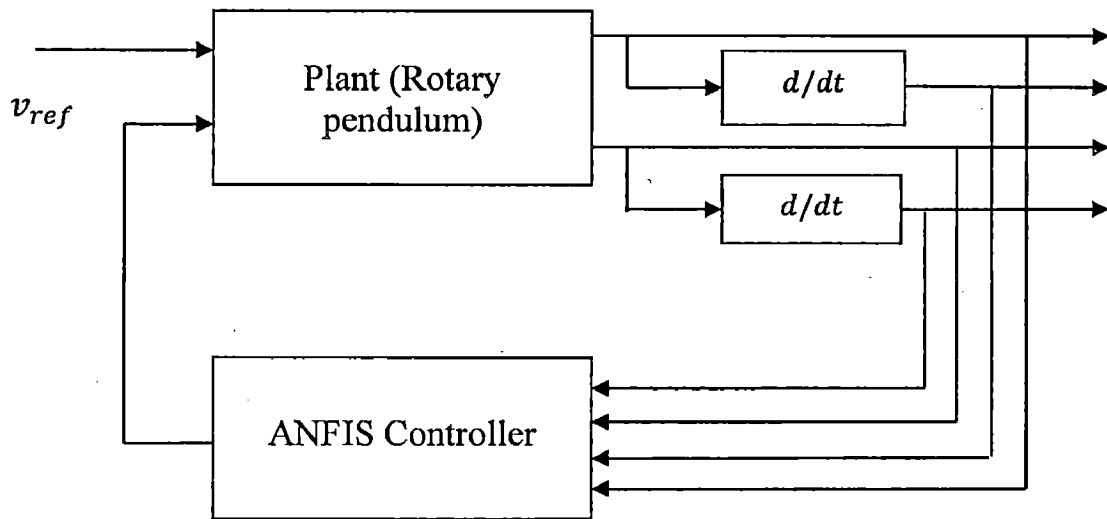


Fig.3. 3 Block Diagram of Inverted Pendulum System with feedback ANFIS controller

The four circles (Fig. 3.4) K_1 , K_2 , K_3 , k_4 are four “knobs” used to provide the gain to the four feedback signals. They are summed together and feed-back to the system as to give the voltage to the motor to rotate the arm. This can be expressed as

$$V = (K_1 * \theta) + (K_2 * \dot{\theta}) + (K_3 * \alpha) + (K_4 * \dot{\alpha}) \quad (3.5)$$

The controller input gains K_1 , K_2 , K_3 and K_4 are determined using the Linear-Quadratic Regulator (LQR) method described by Friedland [34]. This method finds the optimal K based on the state feedback law and the state-space equation derived earlier. For finding out the closed loop stability analysis of inverted pendulum we find out the root locus analysis, frequency analysis and many techniques.

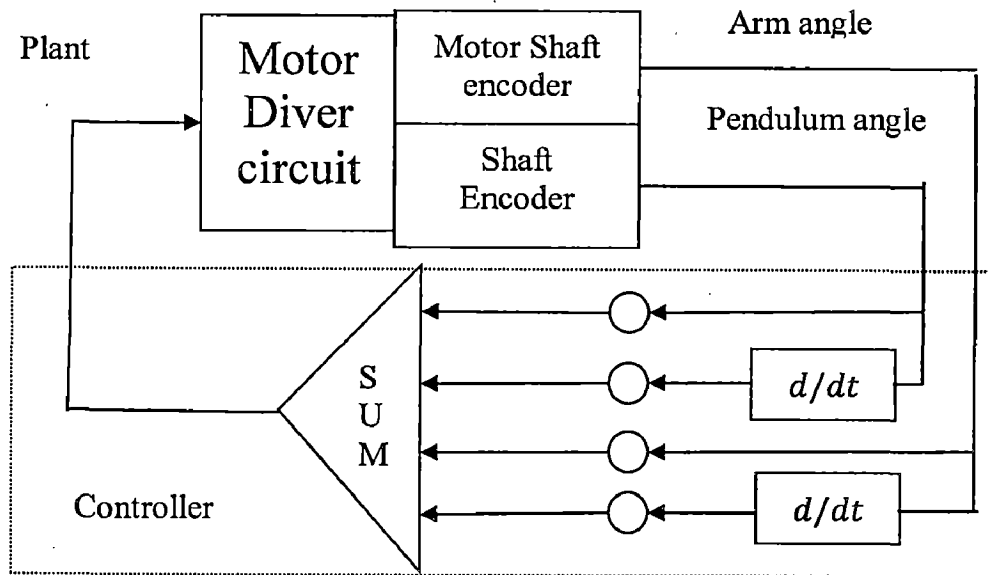


Fig.3. 4 Plant and controller block diagram.

3.3 ANFIS Controller

ANFIS adaptive Neuro-Fuzzy system was first introduced by J. Jang in 1993 [40]. ANFIS constructs a fuzzy inference system (FIS) whose membership function parameters are tuned (adjusted) using either a backpropagation algorithm alone or in combination with a least squares type of method. This uses a network-type structure similar to that of a neural network, which maps inputs through input membership functions and associated parameters, and then through output membership functions and associated parameters to outputs, can be used to interpret the input/output map. The parameters associated with the membership functions changes through the learning process. The computation of these parameters (or their adjustment) is facilitated by a gradient vector. This gradient vector provides a measure of how well the fuzzy inference system is modeling the input/output data for a given set of parameters. When the gradient vector is obtained, any of the learning algorithms is applied in order to adjust the parameters to reduce the error (squared difference between actual and desired outputs).

The structure of the fuzzy inference system (FIS) is taken as Takagi-Sugeno type and four input variables arm angle, arm angular velocity, pendulum angle and pendulum angular velocity are considered and all input variables are having two membership functions. The parameter values of these membership functions are trained by ANFIS to provide the appropriate value of the voltage applied to motor which achieves the goal of balancing the pendulum. Fig. 3.5 shows the structure of the ANFIS controller

For generating the FIS structure ANFIS editor GUI, already available in MATLAB is used. In editor grid portion type structure is selected and hybrid learning is chosen. Training data is available from the above mentioned LQR method which is randomly divided into training, testing and checking data. After training checking and testing of the ANFIS controller, above shown (Fig.3.5) structure is obtained. This structure is considered as five layer feed-forward neural network

- A. The first layer- This layer is a basic input Fuzzification layer where the crisp inputs are allocated relative fuzzy values.
- B. The second layer- This layer of the nodes labels defines the specified membership functions for each input created in the layer one. Gaussian shaped fuzzy memberships are utilized here.
- C. The third layer- The nodes in this layer represent the rules generated for different combinations and instances of inputs. This layer will give the information regarding which rules are to be fired for different possibilities of inputs.
- D. The fourth layer- This layer produces the defuzzified Takagi-Sugeno-type output for each previous i^{th} output. Here a particular defuzzified value is getting generated for each and every rule fired.
- E. The fifth layer- The single node in this layer computes the overall outputs as the summation of all incoming signals. That gives the overall output that is generated from all the rules fired for particular set of input values.

Therefore the output of the ANFIS is clearly is a linear function of all the inputs. This can be seen as the Rule-Base of this controller is given by

$$\text{Rule Base: If } \theta \text{ is A1 and } \dot{\theta} \text{ is B1 and } \alpha \text{ is C1 and } \dot{\alpha} \text{ is D1 then } V_m = (K1 * \theta) + (K2 * \dot{\theta}) + (K3 * \alpha) + (K4 * \dot{\alpha}) + K5$$

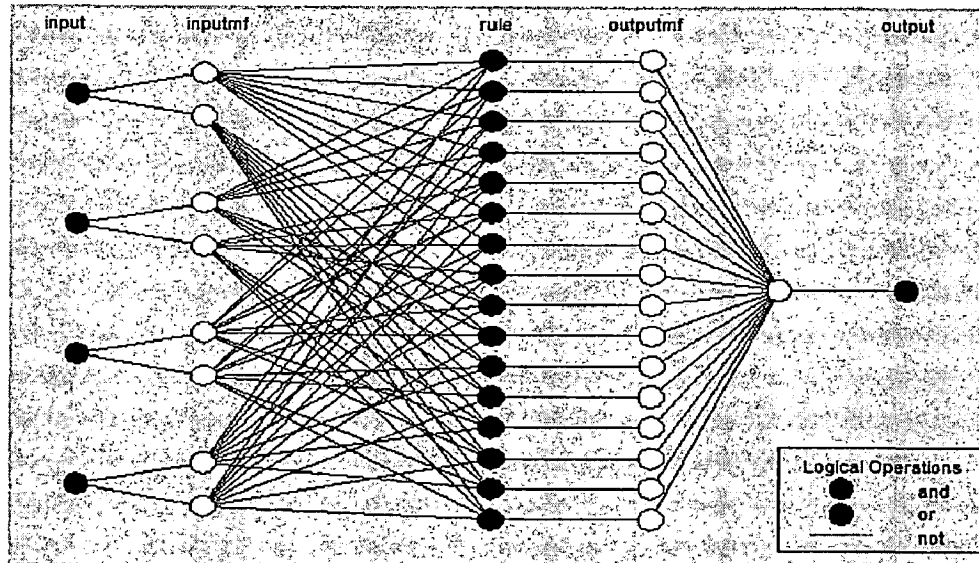


Fig.3. 5 structure of ANFIS controller

3.3.1 Simulation Results and Discussion

The rotary inverted pendulum and controller are implemented in Matlab Simulink environment. For the controller firstly FIS file is generated from the ANFIS editor GUI and used in a fuzzy logic controller block in the Simulink. The non-linear model of rotary pendulum is designed in the Matlab Simulink. The experiment is tested in real-time also

Table3. 1 Values used in the simulation

Parameter	Values	Parameter	Values
B_{eq}	0.004	K_m	0.00767
J_{eq}	0.0035842	K_t	0.00767
J_m	$3.87e-7$	L	0.1675
K_g	70	r	0.215
R_m	2.6	η_g	0.9
η_m	0.69	g	9.8
m	0.125		

The Simulink model is simulated with ode5 solver and 0.001s sampling time. In Fig. 3.6 the falling angle (α) of the pendulum and voltage applied to the servomotor is shown. The applied voltage is calculated by the ANFIS controller which has a maximum and minimum limit of $\pm 6V$. In figure it can be seen that the pendulum is get stable in 1.3s.

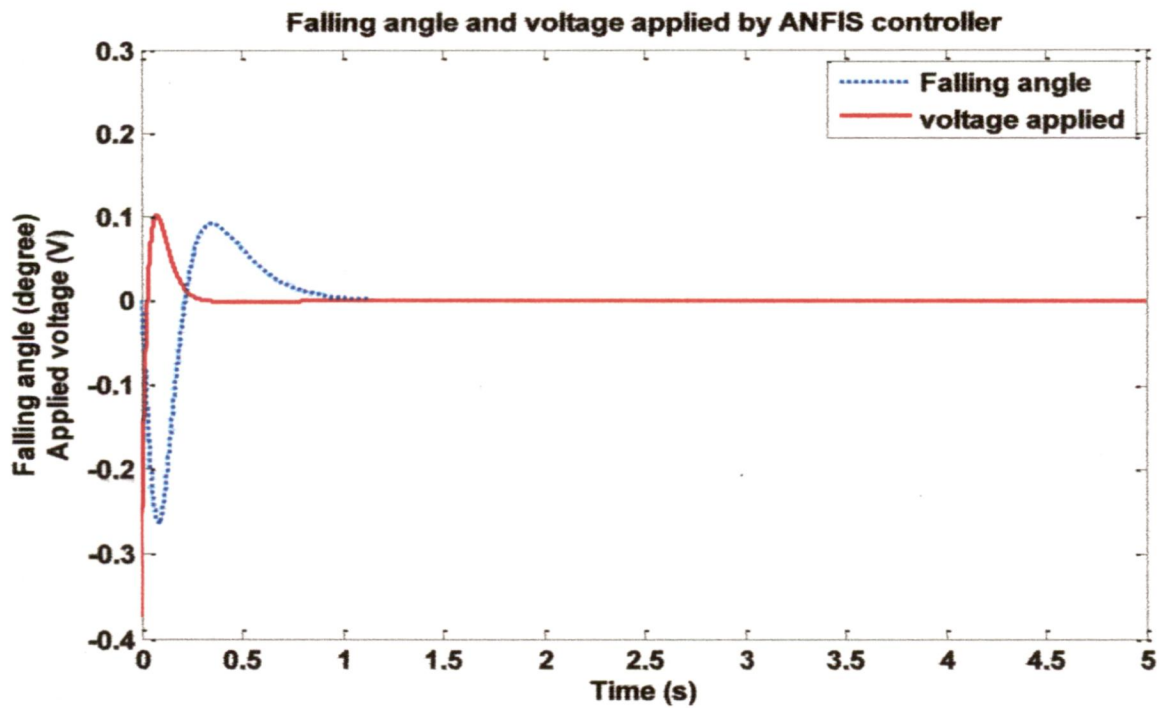


Fig.3. 6 Falling angle and voltage applied plot.

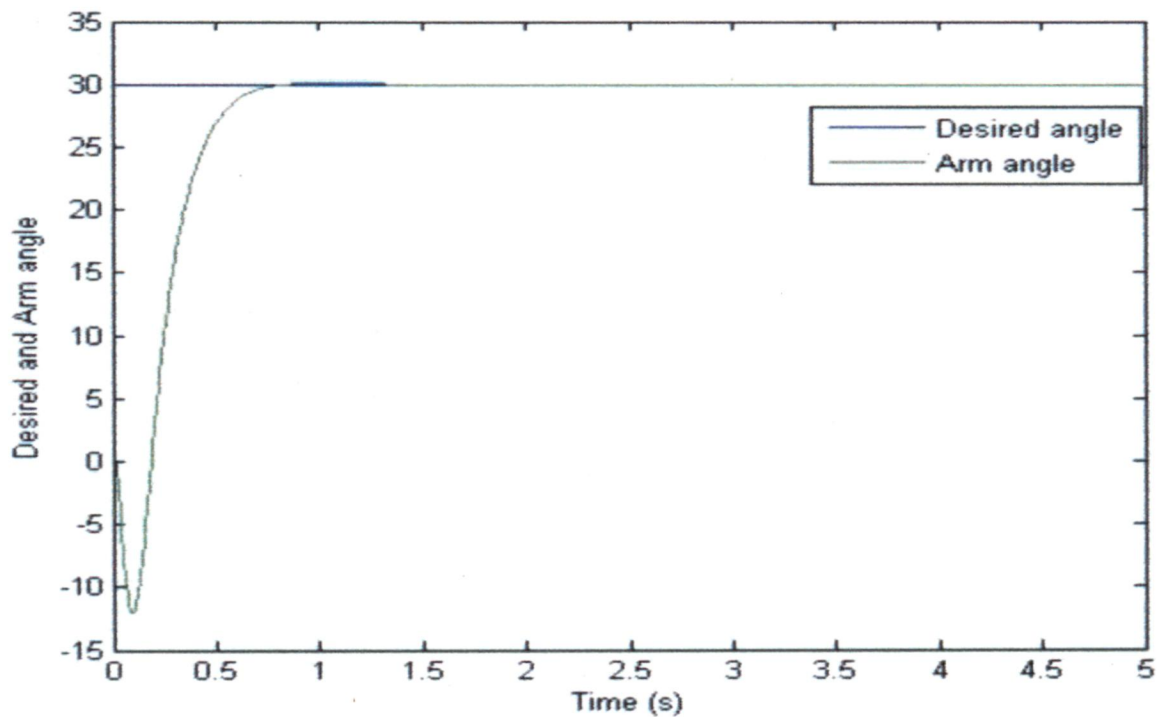


Fig.3. 7 Desired arm position and arm response of ANFIS controller.

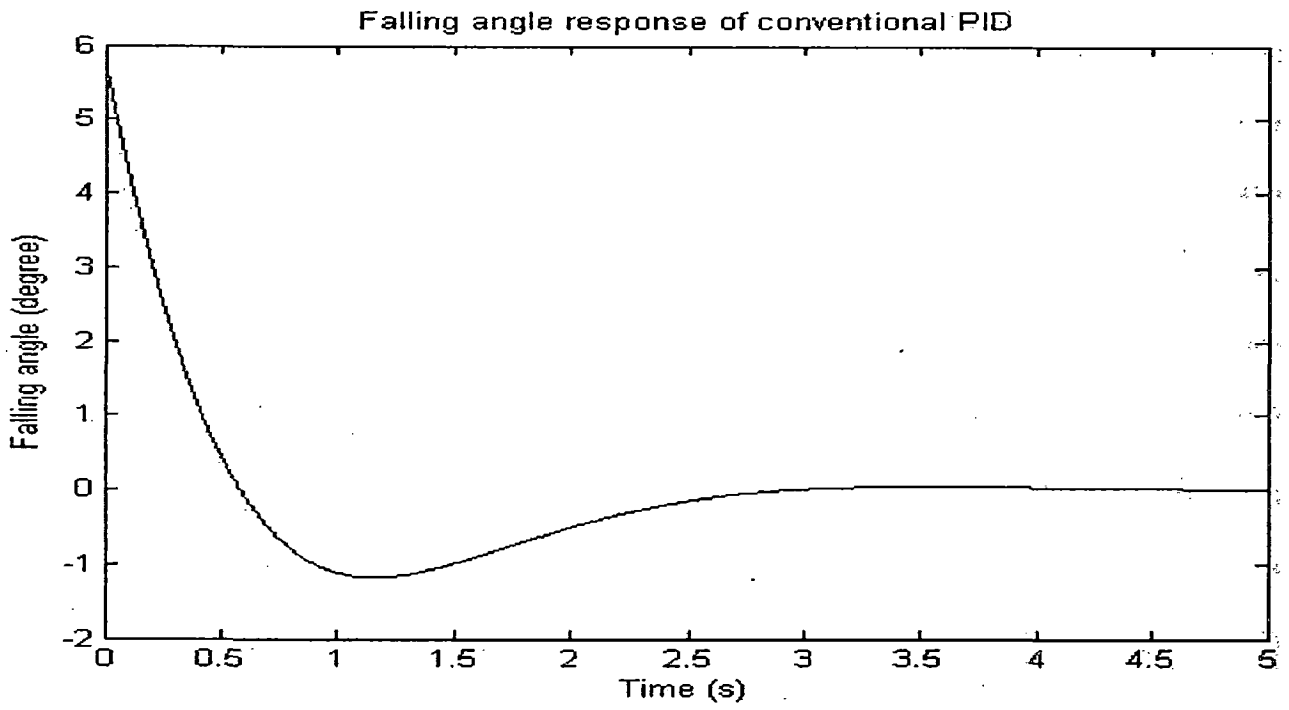
The response of the arm angle versus desired angle is plotted in the Fig. 3.7. This shows that the desired arm angle which is 30 degree in this case is achieved in the nominal time about 1sec

3.3.2 Comparison of ANFIS and conventional PID and fuzzy control

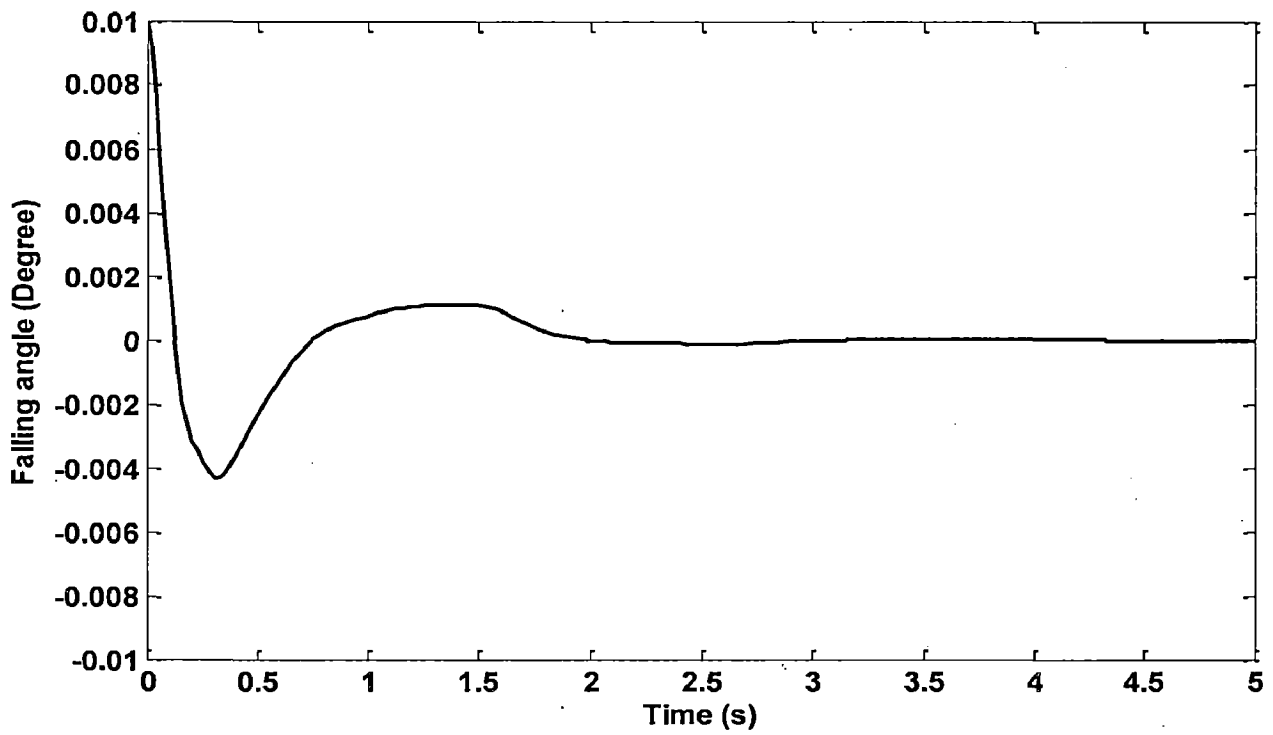
The conventional PID controller and fuzzy controller are designed for the same rotary inverted pendulum problem to compare the result with the proposed ANFIS controller. For the same plant parameter a PID controller is designed with proportional gain K_P , derivative gain K_D , integral gain K_I 5, 11, and 0.02 respectively and efficient fuzzy controller [36] based on LQR. The falling angle of the pendulum in case of ANFIS, PID and efficient fuzzy is plotted in Matlab shown in Fig. 3.8. The graph shows that the settling time and overshoot of the ANFIS controller are much less than PID and efficient fuzzy controller.

In another set of experiment the robustness of the two ANFIS and PID has been checked by changing the mass of the Pendulum is changed from 0.125 to 0.85 kg without changing the parameters of the controllers. Fig. 3.9 shows that the conventional controller gives the un-damped oscillation and unable to stabilize the pendulum anymore while the ANFIS gives the reasonable response however poor than the previous one when no mass has been changed and stabilize the pendulum in 1.7 sec. This proves that proposed ANFIS controller is more robust and does not rely on mathematical description of the plant.

In another set of experiment the robustness of the two ANFIS and PID has been checked by changing the mass of the Pendulum is changed from 0.125 to 0.85 kg without changing the parameters of the controllers. Fig. 3.9 shows that the conventional controller gives the un-damped oscillation and unable to stabilize the pendulum anymore while the ANFIS gives the reasonable response however poor than the previous one when no mass has been changed and stabilize the pendulum in 1.7 sec. This proves that proposed ANFIS controller is more robust and does not rely on mathematical description of the plant.

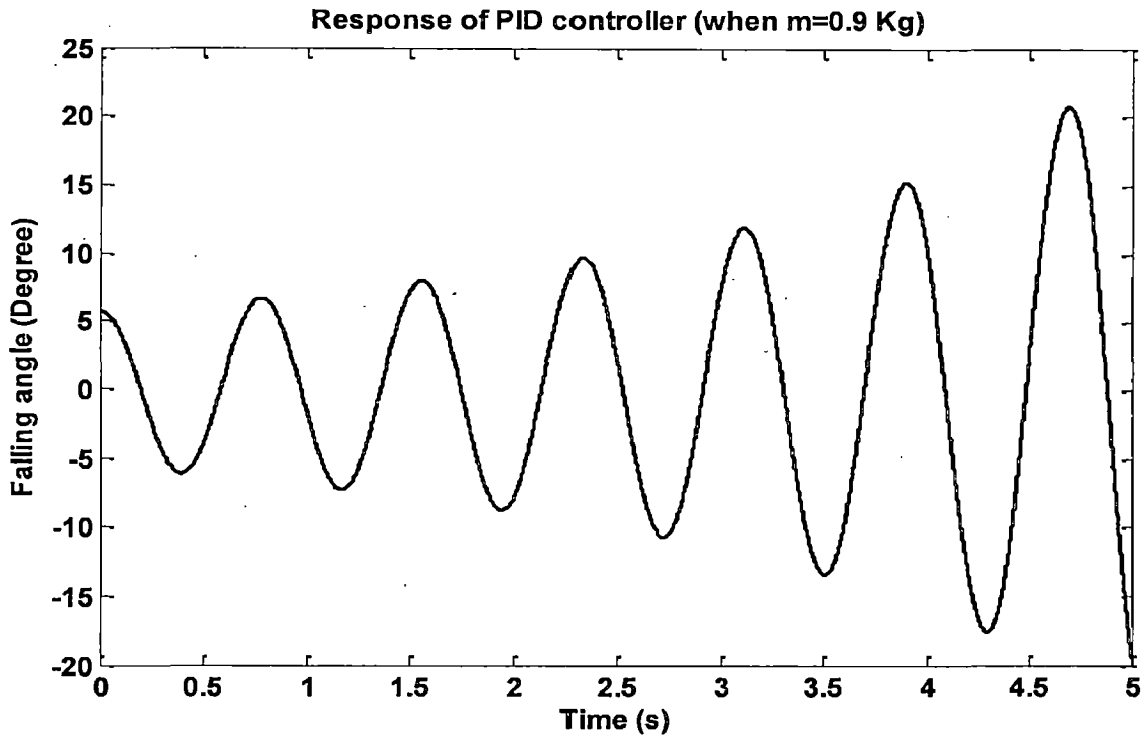


(a)

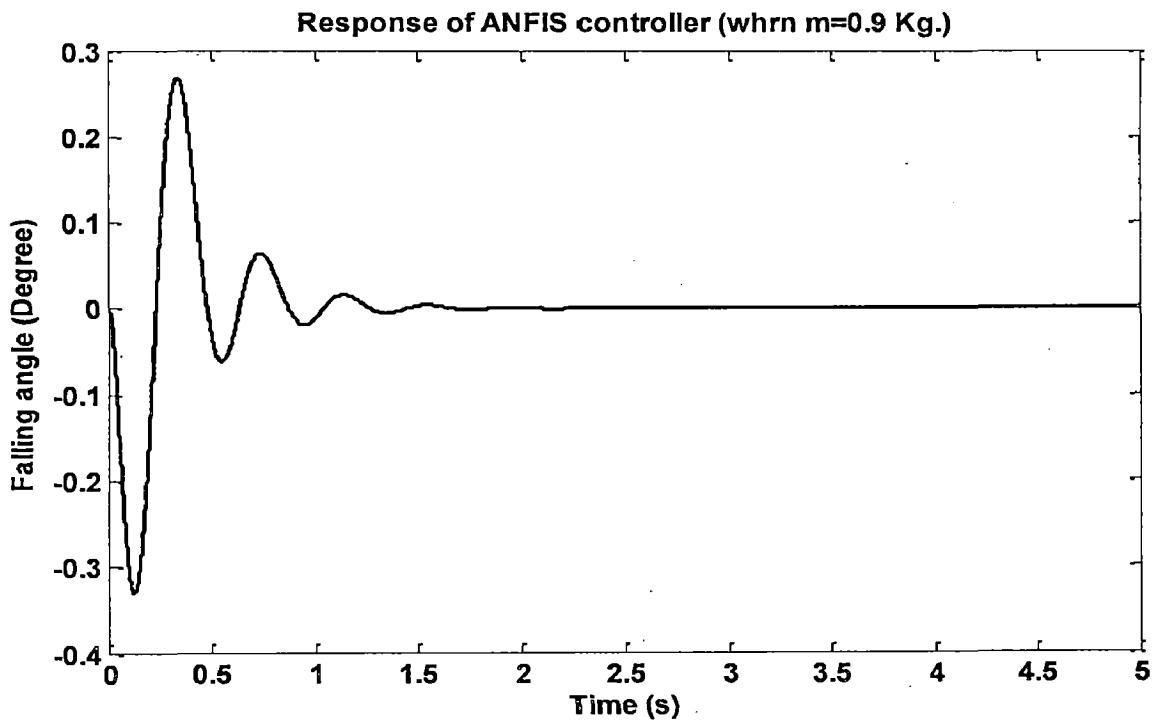


(b)

Fig.3. 8 Falling pendulum angle of (a) PID (b) efficient fuzzy controller



(a)



(b)

Fig.3.9 Falling pendulum angle response of (a) PID controller (b) ANFIS controller when mass is changed to 0.85 Kg.

In this Chapter, ANFIS controller is designed for rotary inverted pendulum in Matlab Simulink with the help of ANFIS editor GUI. The designing of this controller has the

advantages of both the intelligent technique Fuzzy and Neural networks together. In comparison of the modern control design technique, ANFIS is simpler to implement as it eliminates the complicated mathematical process and uses soft computing techniques. In the simulation result it is shown that ANFIS controller is more robust to system parameter variation in comparison to conventional PID and fuzzy controller. In the next Chapter an indirect adaptive controller is presented for the self-balancing two-wheeled transporter which firstly explicitly identified the system using neural network and then controller is designed through the inversion of trained neural network.

CHAPTER – 4

INDIRECT ADAPTIVE CONTROLLER FOR SBTWT

The problem of steering the Self-Balancing Two-Wheel Transporter (SBTWT) in the presence of uncertainty is a very interesting problem. The SBTWT is a mechatronic system which is decomposed into two subsystems: yaw motion and mobile inverted pendulum. Steering of the vehicle can be achieved by controlling of the both subsystem i.e. yaw motion control for the first subsystem and self-balancing of the second subsystem in the presence of uncertainties, unknown and unmodeled parameters and changing environment condition. In this Chapter indirect adaptive controller using network inversion technique is proposed for the two subsystems to achieve self-balancing of the vehicle and yaw motion control to steer the vehicle respectively. Simulation results indicate the identification of the two subsystems using radial-basis function neural network (RBFNN) and the performance of the proposed controller. Simulations also show that the proposed controller is capable of steering the vehicle in desired manner.

For the nonlinear modeling and control, the radial-basis function neural networks (RBFNNs) is more preferred over the multi-layered network (MLN), as RBFNN has a simpler structure and better capability of functional representation [14]. Since its response is linear to weights, learning in RBFNN is expected to be faster. Thus neural networks (which are shown to be universal function approximators [15, 16] have used to explicit model the dynamics of the SBTWT system. Within the framework of indirect adaptive control, in [17] Narendra presents four possible neural network model of the plant to design the indirect adaptive control. In the area of indirect adaptive control design, the forward-inverse-modeling approach [18], feedback error-learning scheme [19], tuning of the controller parameters using back-propagation [20] are some notable contribution. Many researchers present indirect adaptive control design using network inversion [21-27] as well.

Here the indirect adaptive controller for the self-balancing two-wheeled transporter is designed. Since the system is decomposed into two subsystems mobile inverted pendulum and yaw motion subsystem so like JOE [2] by Grasser *et al.* and in [3] Tasi *et al.*, two indirect adaptive controllers i.e. indirect adaptive self-balancing and indirect adaptive yaw motion controller are designed using network inversion technique. For designing the controller firstly

two subsystems are explicitly identified through the two RBFNNs and then from these identified neural network emulators the controllers are designed through network inversion technique. In comparison to state-feedback controller presented by Grasser *et al.* [2] and adaptive controller using RBFNNs given by Tasi *et al.* [3] the designing of proposed indirect adaptive controllers are not dependent on the dynamical model of the subsystems, unknown plant parameters, unknown frictions, uncertainty and linearization error because the control signals are derived from the identified neural network emulators which take inputs directly from the SBTWT through sensors. So the proposed controller is capable of steering the vehicle for different riders and maintaining its standing posture under the unknown model parameter, uncertainty and frictions.

4.1 Subsystems identification using RBFNNs

As shown in the Fig. 4.1, 4.2 RBFNN is selected for modeling the two subsystems of the SBTWT, as the RBFNN performs the excellent approximation for the curve fitting problems and the response of the network is linear in terms of its weights. This helps us to extend linear adaptive control methods to develop non parametric non-linear adaptive system. [14].

Consider the non-linear discrete time dynamics of the each subsystem of SBTWT is described by the equation

$$x(k + 1) = f(x(k), u(k)) \quad (4.1)$$

where $x(k) \in \mathcal{R}^n$ and $u(k) \in \mathcal{R}^p$ represent respectively states and input vector of the system at the k th sampling instant. The states of the systems are assumed to be accessible and nonlinear function $f(x)$ is assumed to be unknown.

The i th output of such a network can be expressed as

$$\hat{x}_i = f_i(v) = \sum_{j=1}^l \theta_{ij} \phi_j(\|v - c_j\|) \quad (4.2)$$

where $v \in \mathcal{R}^m$ is the network input vector; $\|\cdot\|$ denotes the Euclidean norm; $c_j \in \mathcal{R}^m, 1 \leq j \leq l$, are the RBF centers; $\phi(\cdot)$ is the j th activation function of hidden layer; $\theta_{ij}, 1 \leq j \leq l, 1 \leq i \leq n$ are the connection weights from hidden layer to output layer; and l is the number of hidden units in the first layer. Gaussian radial function has been used as the activation function because it has good modeling capability. Of course the RBFN is a general function approximator, and its performance does not depend critically on the choice of $\phi(\cdot)$ [16].

4.1.1 Mobile inverted pendulum subsystem identification

The mobile inverted pendulum subsystem is approximated using the RBFNN model as shown in Fig. 4.1, taking $v = [\theta_p \ \omega_p \ C_\theta]^T$ and $\hat{x}_i = [\hat{\theta}_p \ \hat{\omega}_p]^T$ and output of the network is given as

$$\begin{aligned} \hat{x}_i &= f_i(v) = \theta_{ij} \phi^T \\ &= [\theta_{i1} \ \dots \ \theta_{il}] [\phi_1 \ \dots \ \phi_l]^T \end{aligned} \quad (4.3)$$

where θ_{ij} are the connection weights from hidden layer to output layer; $\phi(\cdot)$ is the j th activation function of hidden layer and given as

$$\phi_j = \exp[-\{(\theta_p - c_1)^2 + (\omega_p - c_2)^2 + (C_\theta - c_3)^2\}/\sigma_j] \quad (4.4)$$

where $c_i, i = 1,2,3$ are the centers of the receptive field and $\sigma_j, j = 1,2..l$ is the inverse of the width of Gaussian function.

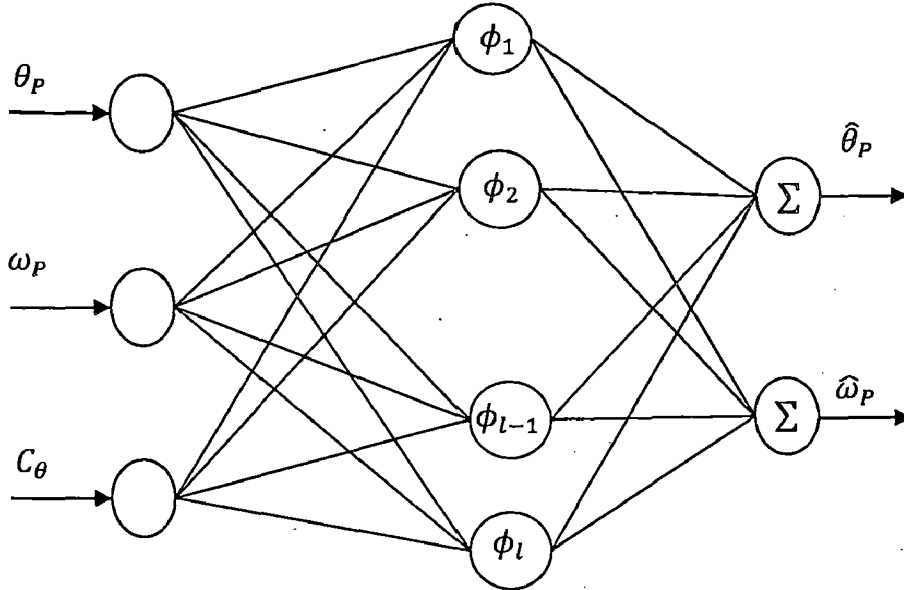


Fig.4. 1 Structure of RBFNN model of inverted pendulum subsystem

4.1.2 Yaw motion subsystem identification

The yaw motion subsystem is approximated using the RBFNN model as shown in Fig. 4.2, taking $v = [\theta_y \ \omega_y \ C_y]^T$ and $\hat{x}_i = [\hat{\theta}_y \ \hat{\omega}_y]^T$ and output of the network is given as

$$\begin{aligned} \hat{x}_i &= f_i(v) = \theta_{ij} \phi^T \\ &= [\theta_{i1} \ \dots \ \theta_{il}] [\phi_1 \ \dots \ \phi_l]^T \end{aligned} \quad (4.5)$$

where θ_{ij} are the connection weights from hidden layer to output layer; $\phi(\cdot)$ is the j th activation function of hidden layer and given as

$$\phi_j = \exp[-\{(\theta_y - c_1)^2 + (\omega_y - c_2)^2 + (C_y - c_3)^2\}/\sigma_j] \quad (4.6)$$

where $c_i, i = 1,2,3$ are the centers of the receptive field and $\sigma_j, j = 1,2..l$ is the inverse of the width of Gaussian function.

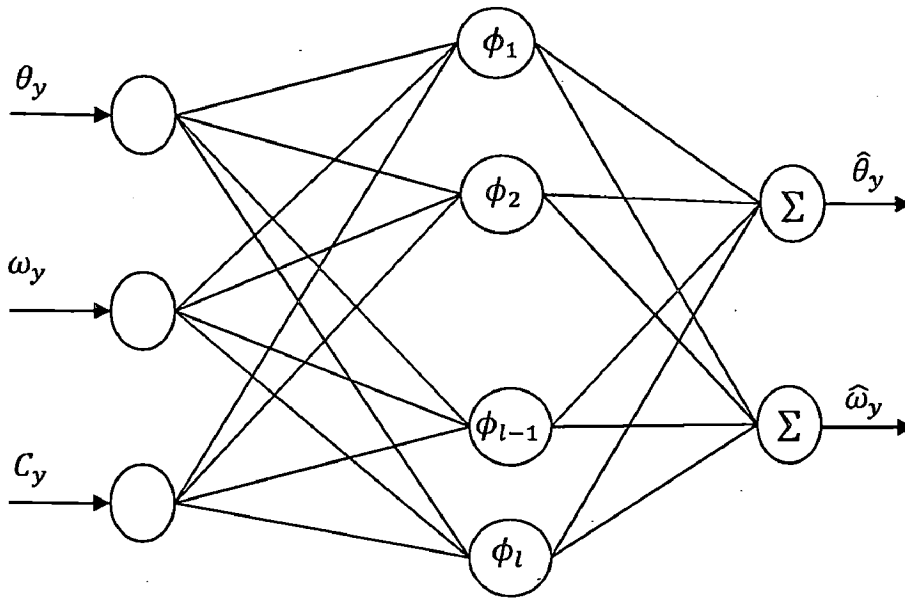


Fig.4. 2 Structure of RBFNN model of yaw motion subsystem

4.2 Learning algorithm

The centers and weights of an RBFN can be tuned using ideas from nonlinear system identification theory such as parallel recursive prediction error (PRPE) algorithm [28] or extended Kalman filtering (EKF) algorithm [25, 29]. The simplest approach is to update the centers using gradient descent algorithm and the weights can be updated using simple LMS [30] algorithm. Although computational requirement increases by adjusting centers, the number of radial centers can be substantially reduced by this approach [31]. The generalization performance of such a network is much better as compared to hybrid learning scheme where centers are fixed or learned unsupervised and the weights are updated using recursive least squares algorithm. In this fixed number of radial centers are taken to be of uniform random distribution over the input space.

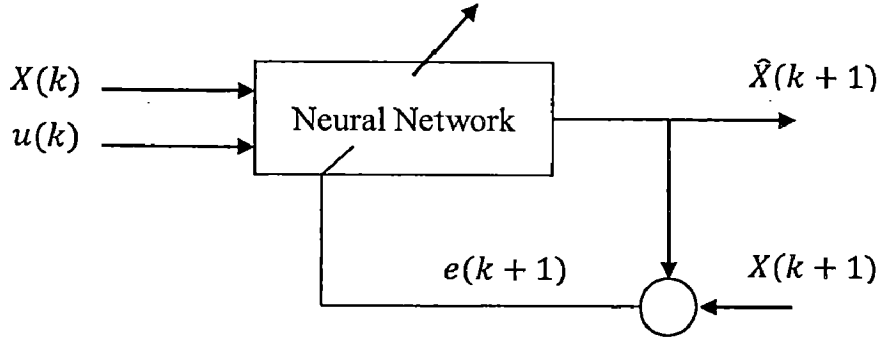


Fig.4. 3 Subsystem identification using feed-forward network

The learning procedure is shown in the Fig. 4.3, the inputs to the network are current states of the subsystem and the control signal $X(k), u(k)$ and the output is the next states $\hat{X}(k+1)$ of the system. $X(k+1)$ is the states of the plant so the network weights are updates such that the error signal $e(k+1)$ get minimized. The training data set should be dimensionally sufficient to obtain a valid neural model of the subsystems. Since the two subsystems are open loop unstable the training data samples are obtained using a PD controller with additive dither signal. RLS [32] algorithm is selected to learn the subsystems dynamics.

The i th output of the RBFN described earlier, can be written as

$$\hat{x}_i = f_i(v) = \theta_{ij} \phi^T \quad (4.7)$$

The weight update equations as per RLS algorithm [32], are described as

$$\hat{\theta}_i(k) = \hat{\theta}_i(k-1) + \mathbf{P}(k) \phi(k-1) [x_i(k) - \phi(k-1)^T \hat{\theta}_i(k-1)] \quad (4.8)$$

$$\begin{aligned} \mathbf{P}(k) = & \mathbf{P}(k-1) \\ & - \mathbf{P}(k-1) \phi(k-1) \\ & - 1) (1 + \phi(k-1)^T \mathbf{P}(k-1))^{-1} \phi(k-1)^T \mathbf{P}(k-1) \end{aligned} \quad (4.9)$$

where $\mathbf{P}(k) \in \mathcal{R}^{l \times l}$ and it is taken as $50I$ for both the two subsystems.

4.3 Indirect adaptive controller design using network inversion

Inverse mapping of the RBFNNs will generate the input pattern for a desired output pattern [21]-[27]. Though inversion process it is possible to obtain the required control input for the

desired output trajectory of the system.

The RBFNN models given by (4.3) and (4.5) represent a non-linear mapping from a 3-dimensional input space $(\theta_P, \omega_P, C_\theta)$ in case of mobile inverted pendulum subsystem) to a 2-dimensional output space $(\hat{\theta}_P, \hat{\omega}_P)$. The objective of inverse operation on these models is to predict only one input (C_θ) out of three inputs. The remaining two inputs are known *a priori* (present subsystem states). The predicted input can be mathematically expressed as

$$u(k) = g(x(k), x^d(k+1), c, W) \quad (4.10)$$

The inversion algorithm predicts the control signal by updating input activation $\hat{u}(k)$ iteratively till the desired output activation is achieved or the number of iteration reaches a maximum t_{max} . This upper bound is decided by the sampling interval and computation time required per iteration. The initial guess of the input activation function $\hat{u}(k)$ during each sampling interval is taken as the input activation $u(k-1)$ predicted in the previous sampling instant. For the case first sampling interval, the initial guess is selected arbitrary from the input space.

4.3.1 Indirect adaptive self-balancing controller design using network inversion

In this Section the indirect adaptive self-balancing controller is designed using network inversion. The block diagram of the proposed controller is shown in Fig. 4.4. Since the mobile inverted pendulum subsystem is explicitly identified through RBFNN, the control law is derived from this neural network emulator. The inverse mapping is achieved by Lyapunov based approach. This approach is presented by Lee [33] for pattern recognition problem; same concept is used [24] to deriving the control law in following way. The advantage of this approach is that the convergence is guaranteed since the algorithm is derived using Lyapunov stability concept.

The Lyapunov function candidate $V(x(t), t)$ is chosen to be quadratic error function in the desired trajectory of subsystem (2.38)

$$V = \frac{1}{2} (\tilde{x}^T \tilde{x}) \text{ where } \tilde{x} = x_d - \hat{x} \quad (4.11)$$

where $x^d = (\theta_{PC}, \omega_{PC}) = 0$ is the desired trajectory of subsystem and \hat{x} is the actual output of the RBFNN model (4.3). The time derivative of the Lyapunov function V is given by

$$\dot{V} = -\tilde{x}^T \frac{\partial \hat{x}}{\partial C_\theta} \dot{C}_\theta \quad (4.12)$$

$$= -\tilde{x}^T J \dot{C}_\theta$$

$$\text{where } J = \frac{\partial \hat{x}}{\partial C_\theta} \quad J \in R^{2 \times 1} \quad (4.13)$$

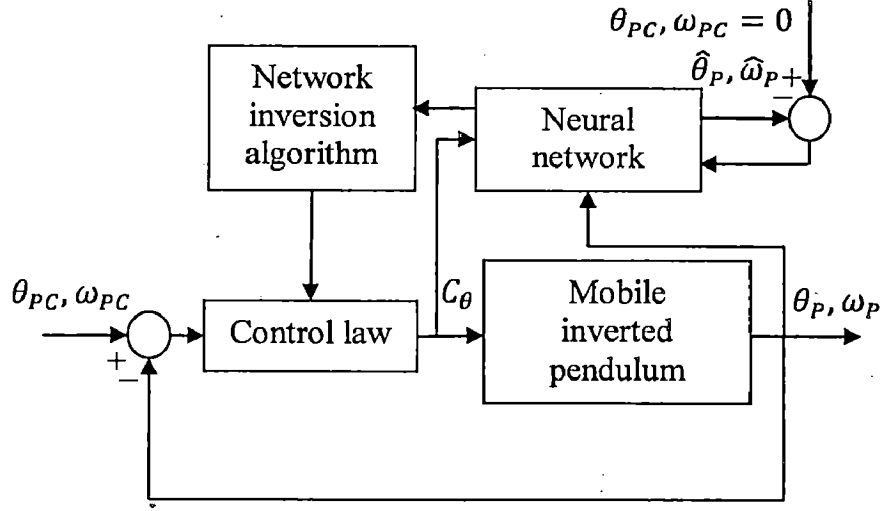


Fig.4. 4 Indirect adaptive self-balancing controller

Theorem1: if an arbitrary input activation $C_\theta(0)$ is updated by

$$C_\theta(t') = C_\theta(0) + \int_0^t \dot{C}_\theta dt \quad (4.14)$$

Where

$$\dot{C}_\theta = \frac{\|\tilde{x}\|^2}{\|J^T \tilde{x}\|^2} J^T \tilde{x} \quad (4.15)$$

Then \tilde{x} converges to zero under the condition that \dot{C}_θ exists along the convergence trajectory.

Proof: Substitution of C_θ in \dot{V} field (4.12).

$$\dot{V} = -\|\tilde{x}\|^2 \leq 0$$

where $\dot{V} < 0$ for all $\tilde{x} \neq 0$ and $\dot{V} = 0$ if and only if $\tilde{x} = 0$ thus update law is stable and \tilde{x} converges to zero in time. The iterative input action is update rule can be given by

$$C_\theta(t') = C_\theta(t' - 1) + \mu \dot{C}_\theta(t' - 1) \quad (4.16)$$

where μ is a small constant representing the update rate and t' represents the iteration index.

The positive numerical instability associated with the weight update law can be avoided by adding a small positive constant ϵ in the denominator. In this case \dot{C}_θ becomes

$$\dot{C}_\theta = \frac{\|\tilde{x}\|^2}{\|J^T \tilde{x}\|^2 + \epsilon} J^T \tilde{x} \quad (4.17)$$

From this (4.12) \dot{V} becomes

$$\begin{aligned}\dot{V} &= -\|\tilde{x}\|^2 \frac{\|\tilde{x}\|^2}{\|J^T \tilde{x}\|^2 + \epsilon} \\ &= -\alpha \|\tilde{x}\|^2\end{aligned}\quad (4.18)$$

where $0 < \alpha < 1$, since α is positive, \dot{V} is negative semi-definite. Thus V will decrease with the update of C_θ , so as the tracking error \tilde{x} . Once the update of C_θ is over, input $C_\theta(k)$ at the k th instant is assigned to the update value $C_\theta(t')$ and applied to the actual subsystem.

4.3.2 Indirect adaptive Yaw motion controller using network inversion

This Section is devoted to design the indirect adaptive yaw motion controller using network inversion technique, since the potentiometer is used to measure difference between equilibrium point and the yaw angle that rider intended to achieve, the indirect adaptive yaw control problem is reduced to an indirect adaptive regulation problem of the SBTWT. In Fig. 4.5 the controller structure is shown. Therefore similar to previous section, the Lyapunov based approach is used for inversion of the RBFN model of subsystem (2.34). The yaw motion subsystem is identified using RBFNN (4.5) and this is used to deriving the control law for achieving the yaw motion of the SBTWT. The inverse mapping is achieved by Lyapunov based approach because the advantage of this approach is that the convergence is guaranteed since the algorithm is derived using Lyapunov stability concept. The Lyapunov function candidate $V(x(t), t)$ is chosen to be quadratic error function in the desired trajectory of subsystem (2.34)

$$V = \frac{1}{2} (\tilde{x}^T \tilde{x}) \text{ where } \tilde{x} = x_d - \hat{x} \quad (4.19)$$

where $x^d = (\theta_{yc}, \omega_{yc}) = 0$ is the desired trajectory of subsystem and \hat{x} is the actual output of the RBFNN model (4.5). The time derivative of the Lyapunov function V is given by

$$\begin{aligned}\dot{V} &= -\tilde{x}^T \frac{\partial \hat{x}}{\partial C_y} \dot{C}_y \\ &= -\tilde{x}^T J \dot{C}_y\end{aligned}\quad (4.20)$$

$$\text{where } J = \frac{\partial \hat{x}}{\partial C_y} \quad J \in R^{2 \times 1} \quad (4.21)$$

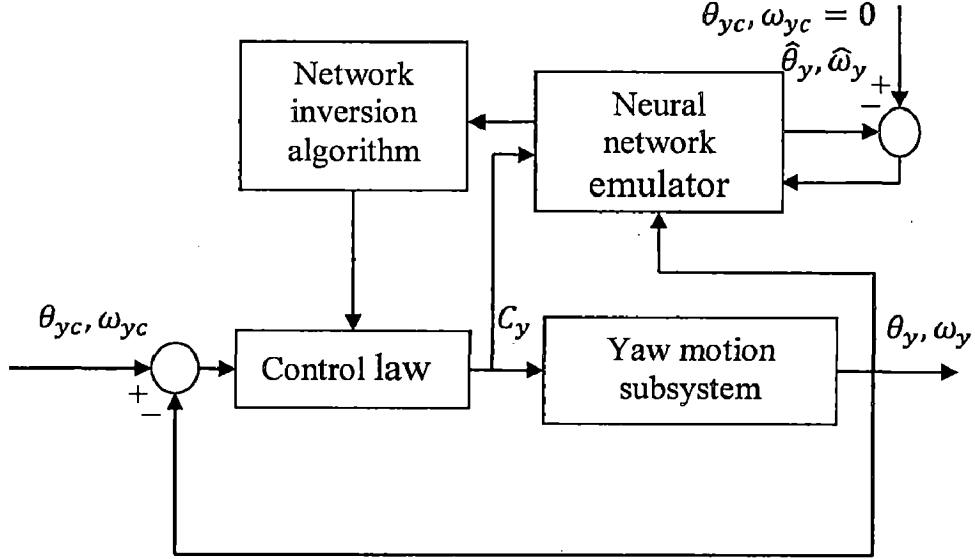


Fig.4. 5 Indirect adaptive yaw motion controller

Theorem 2: if an arbitrary input activation $C_y(0)$ is updated by

$$C_y(t') = C_y(0) + \int_0^t \dot{C}_y dt \quad (4.22)$$

Where

$$\dot{C}_y = \frac{\|\tilde{x}\|^2}{\|J^T \tilde{x}\|^2} J^T \tilde{x} \quad (4.23)$$

Then \tilde{x} converges to zero under the condition that \dot{C}_y exists along the convergence trajectory.

Proof: Substitution of C_y in \dot{V} field (4.20).

$$\dot{V} = -\|\tilde{x}\|^2 \leq 0$$

where $\dot{V} < 0$ for all $\tilde{x} \neq 0$ and $\dot{V} = 0$ if and only if $\tilde{x} = 0$ thus update law is stable and \tilde{x} converges to zero in time. The iterative input action is update rule can be given by

$$C_y(t') = C_y(t' - 1) + \mu \dot{C}_y(t' - 1) \quad (4.24)$$

where μ is a small constant representing the update rate and t' represents the iteration index.

The positive numerical instability associated with the weight update law can be avoided by adding a small positive constant ϵ in the denominator. In this case \dot{C}_y becomes

$$\dot{C}_y = \frac{\|\tilde{x}\|^2}{\|J^T \tilde{x}\|^2 + \epsilon} J^T \tilde{x} \quad (4.25)$$

From this Eq. (4.20) \dot{V} becomes

$$\dot{V} = -\|\tilde{x}\|^2 \frac{\|\tilde{x}\|^2}{\|J^T \tilde{x}\|^2 + \epsilon}$$

$$= -\alpha \|\tilde{x}\|^2$$

where $0 < \alpha < 1$, since α is positive, \dot{V} is negative semi-definite. Thus V will decrease with the update of C_y , so as the tracking error \tilde{x} . Once the update of C_y is over, input $C_y(k)$ at the k th instant is assigned to the update value $C_y(t')$ and applied to the actual subsystem.

4.4 Simulation results

In this section two sets of simulation are conducted to show the performance and effectiveness of the proposed indirect adaptive controllers using the network inversion technique. Table 2.1 shows the all parameter used for the simulation and the online input-output data is generated for the two radial-basis function neural network (RBFNN) model (4.3) and (4.5) to learn the dynamics of the two subsystems (2.38) and (2.34). In the first set of simulation result the identification of the two subsystems using RNFNN are discussed and in the second set of simulation the performance of the two indirect adaptive controllers is shown.

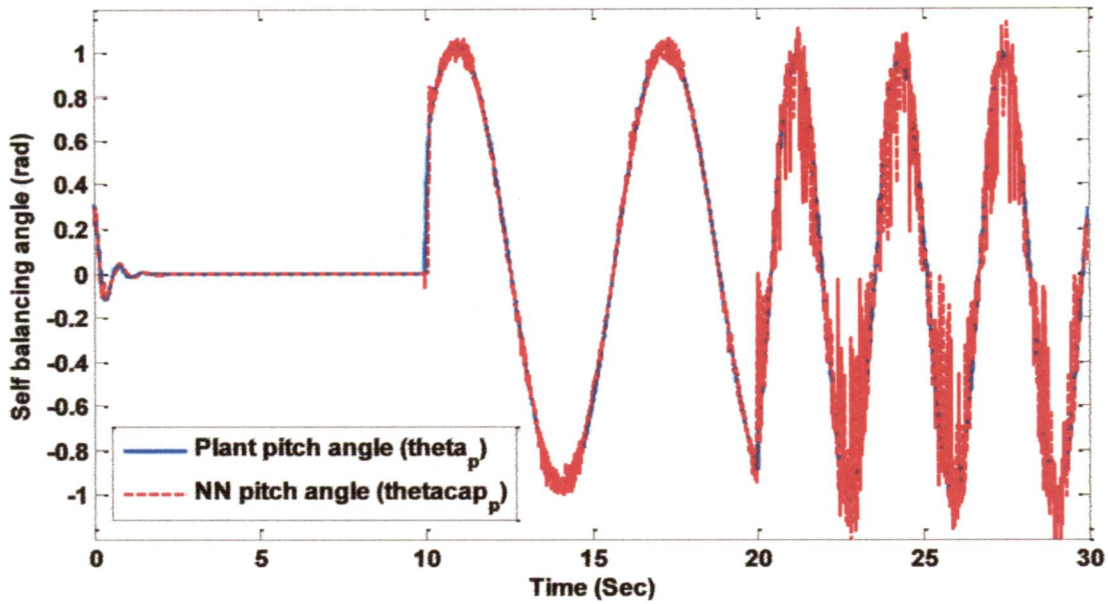
4.4.1 Online data generation

The Euler representation of the two subsystems (2.38) and (2.34) with the sampling time $T = 0.01 \text{ sec.}$ is used for generation of the training data. Since the two subsystems are open loop unstable, the training data samples for the neural network model of two subsystems are generated using the PD controller to self-balance and yaw motion of the vehicle. While tracking the trajectory at each sampling instant various dither signal as in the form of white noise, impulses, step functions, ramp and parabolic type of functions are added to PD controller output to improve generalization capabilities of RBFN model. In this way the 3000 input-output data samples are collected taking the sampling interval to be 10ms.

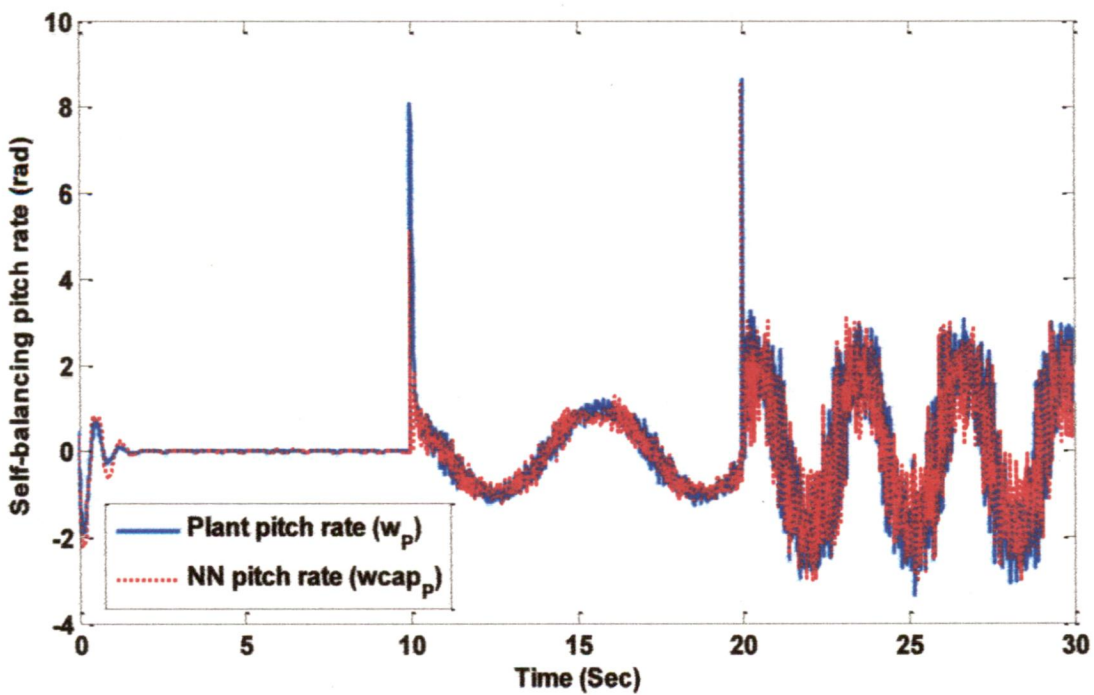
4.4.2 Mobile inverted pendulum subsystem identification

In Fig. 4.6 the identification of mobile inverted pendulum subsystem using radial-basis function network is shown. The number of hidden layer neurons for the RBFNN is taken as 100. The basis-function is assumed to be Gaussian for which the centers are fixed and randomly within its input range. The input-output data are normalized. The training of the neural network is done through recursive least-square (RLS) using (4.8) and (4.9). In Fig. 4.6 (a) shows the pitch angle (self-balancing angle) θ_p of subsystem (2.38) and the identified

state (pitch angle $\hat{\theta}_p$) of neural network (4.3). While In Fig. 4.6 (b) the pitch angle rate ω_p of subsystem and the identified state (pitch angle rate $\hat{\omega}_p$) of neural network is shown. The RMS errors for the training of the self-balancing subsystem are found to be 10.54 and 14.43.



(a)

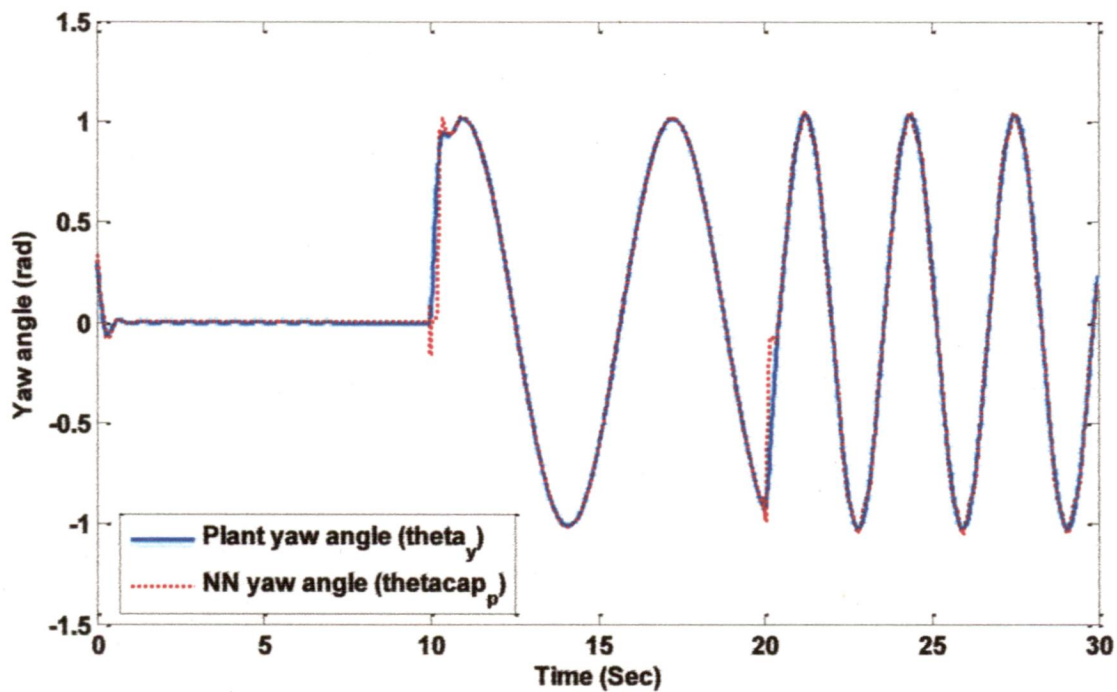


(b)

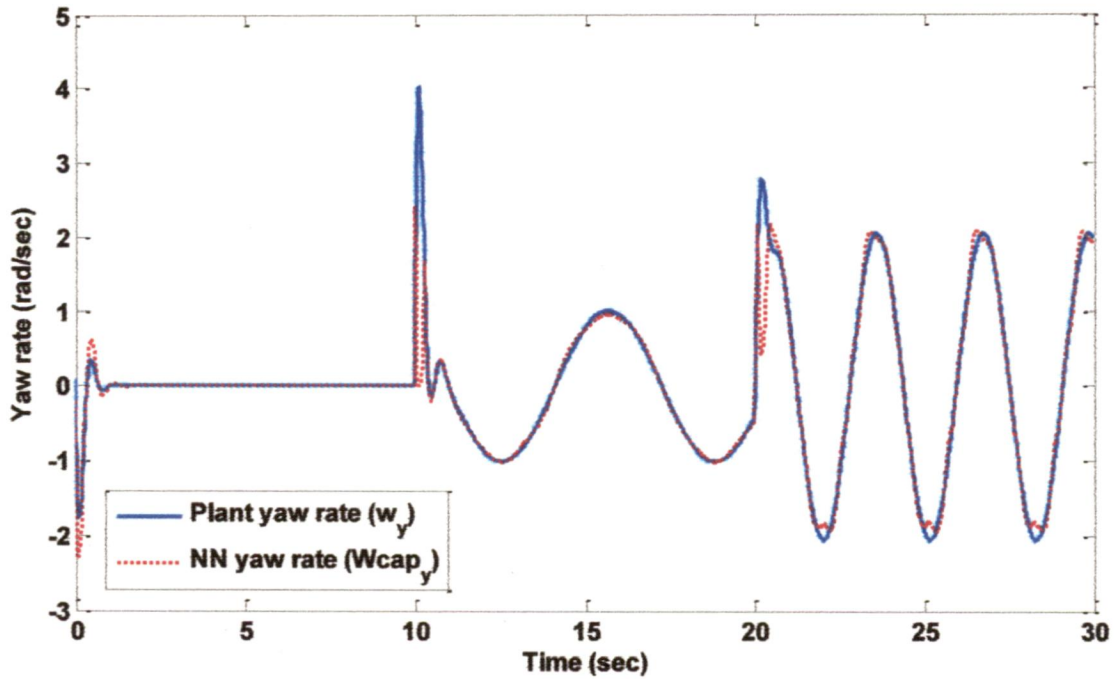
Fig.4. 6 Mobile inverted pendulum: identification (a) the state θ_p and (b) the state ω_p

4.4.3 Yaw motion subsystem identification

In Fig. 4.7 the identification of yaw motion subsystem using radial-basis function network is shown. Similar to mobile inverted pendulum subsystem, the number of hidden layer neurons for the RBFNN is taken as 100. The basis-function is assumed to be Gaussian for which the centers are fixed and randomly within its input range. The input-output data are normalized. The training of the neural network is done through recursive least-square (RLS) using (4.8) and (4.9). In Fig. 4.7 (a) shows the yaw angle θ_y of yaw motion subsystem and the identified state (yaw angle $\hat{\theta}_y$) of neural network (4.5). While In Fig. 4.7 (b) the yaw angle rate ω_y of subsystem (2.34) and the identified state (yaw angle rate $\hat{\omega}_y$) of neural network is shown. The RMS errors for the training of the yaw motion subsystem are found to be 5.74 and 8.54.



(a)

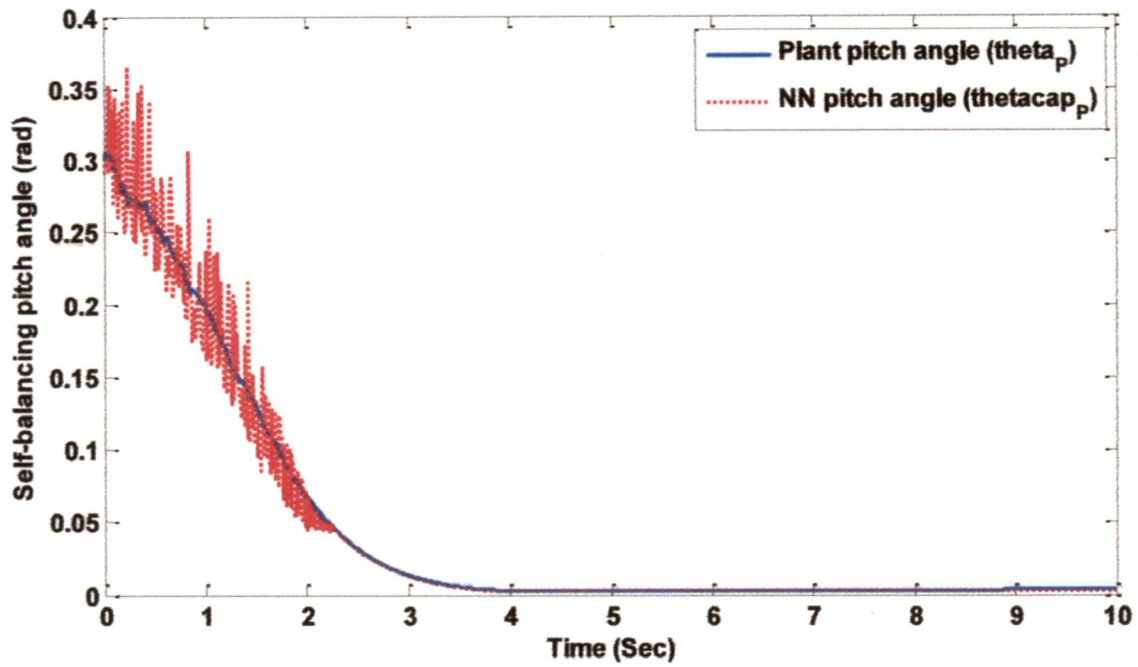


(b)

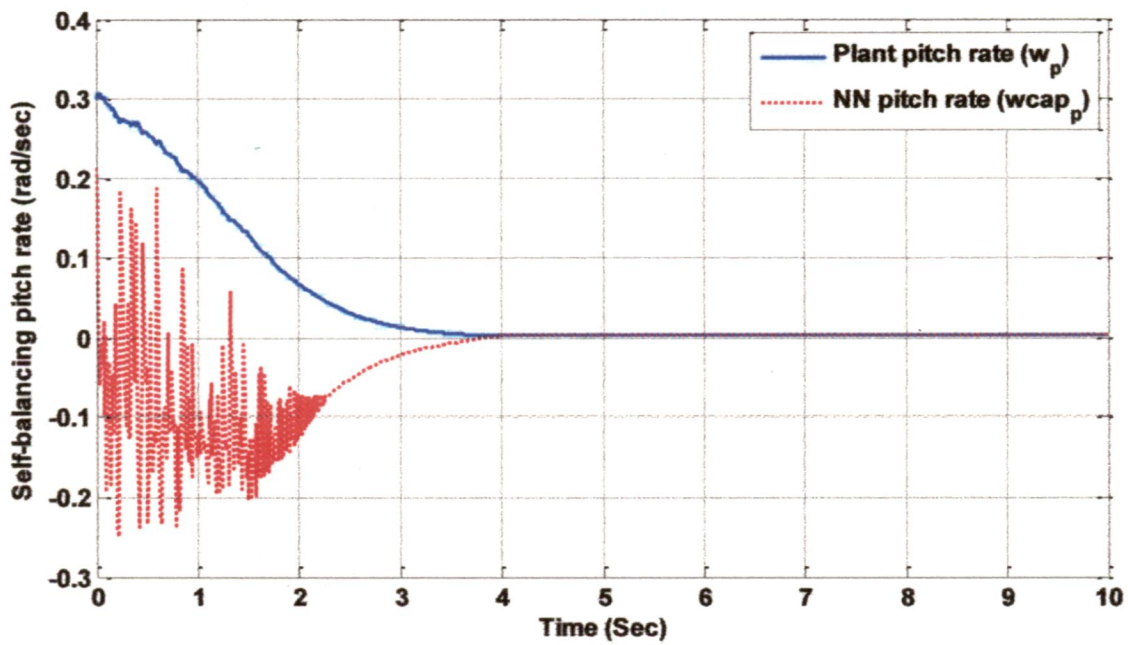
Fig.4. 7 Yaw motion subsystem: identification (a) the state θ_y and (b) the state ω_y

4.4.4 Indirect adaptive self-balancing controller

In this set of simulation the performance of the proposed indirect adaptive self-balancing controller is shown in Fig. 4.8. The indirect adaptive controller is designed through identified neural network model using network inversion technique discussed before and the control signal is calculated using (4.14), (4.15) and (4.16). The value of t_{max} is taken to 30 and the update rate μ is taken 0.05. In Fig 4.8 (a) the pitch angle θ_p of the mobile inverted subsystem and the pitch angle $\hat{\theta}_p$ of the RBFNN model is shown and in Fig 4.8 (b) the pitch rate ω_p of the subsystem (2.37) and pitch rate $\hat{\omega}_p$ of the neural network model is shown. The pitch angle and pitch angle rate both converges to zero in around 4 sec. This shows that the rider can stand properly on the vehicle without falling and the proposed indirect adaptive controller works fine.

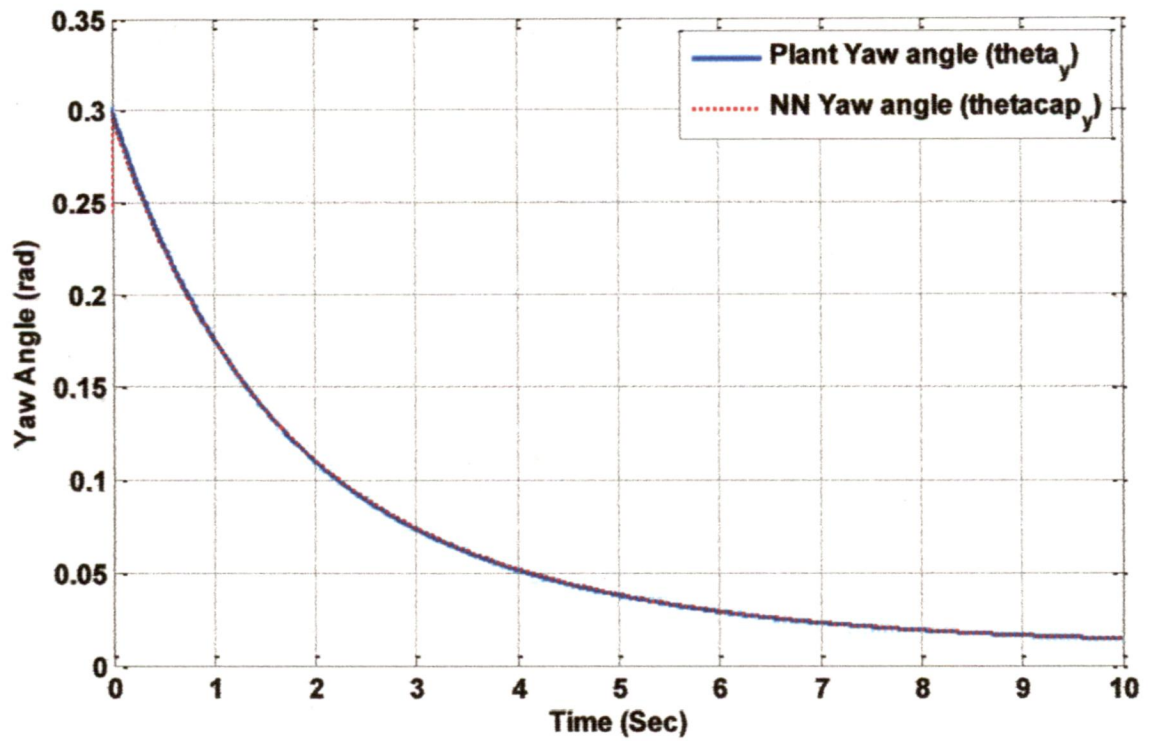


(a)

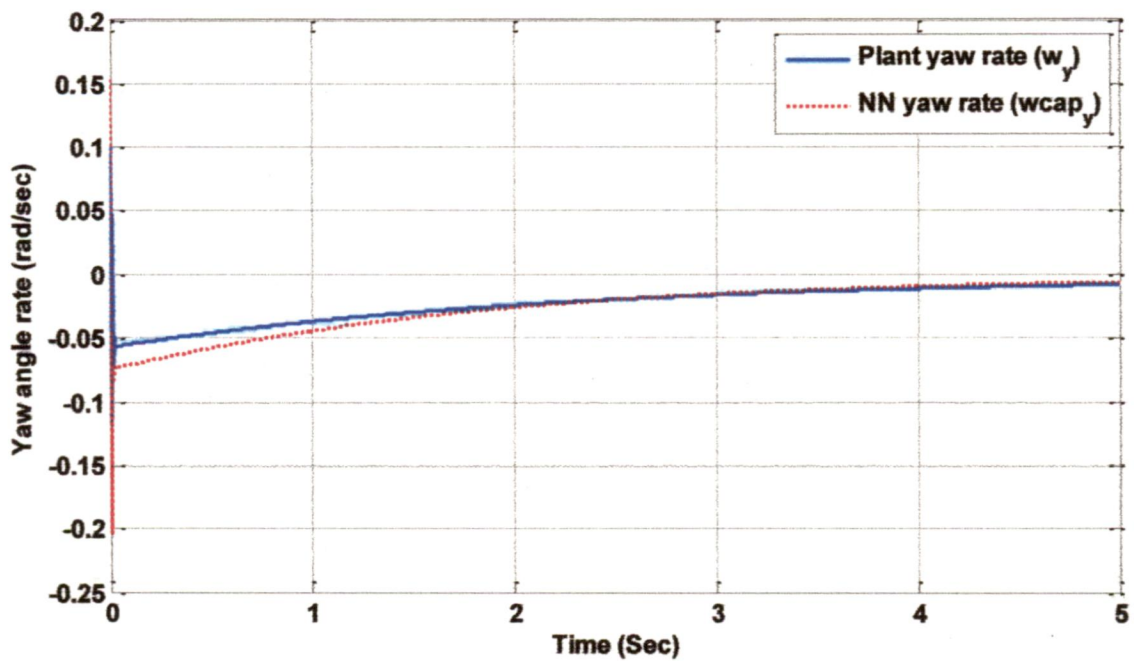


(b)

Fig.4. 8 Pitch angle tracking for the indirect adaptive self-balancing controller using network inversion technique (a) the state θ_p and (b) the state ω_p



(a)



(b)

Fig.4. 9 Yaw angle tracking for the indirect adaptive Yaw motion controller using Network inversion technique (a) the state θ_y and (b) the state ω_y

4.4.5 Indirect adaptive yaw motion controller

The aim of the yaw motion controller is to achieve the yaw angle tracking that rider intended

to achieve. This is why it becomes the indirect adaptive regulation problem of SBTWT. The performance of the proposed indirect adaptive yaw controller is shown in Fig. 4.9. The indirect adaptive controller is designed by inversion of identified neural network model and the control signal is calculated using (4.22), (4.23) and (4.24). The value of t_{max} is taken to 30 and the update rate μ is taken 0.05. In Fig 4.9 (a) the yaw angle θ_y of the yaw motion subsystem and the yaw angle $\hat{\theta}_y$ of the RBFNN model is shown and in Fig 4.9 (b) the yaw rate ω_y of the subsystem (2.34) and yaw rate $\hat{\omega}_y$ of the neural network model is shown. The pitch angle and pitch angle rate both converges to zero in around 4 and 2 sec respectively. This shows that the rider can stand properly on the vehicle without falling and the proposed indirect adaptive controller works fine

In this way an indirect adaptive controller has been proposed for the self-balancing two-wheeled transporter. The system has been explicitly, identified using two radial-basis function neural networks. The current states of the subsystem and the control signal to the subsystem have taken as the inputs while the next states of the subsystem have taken as output for the identification of these subsystems. The input-output data sets for the training of these two neural network emulators have generated using PD yaw motion controller and PD self-balancing controller. The training of two neural network emulators has been done through recursive least square (RLS) method. For designing the indirect adaptive self-balancing and indirect yaw motion controller the network inversion algorithm has used. The control law for the both the adaptive controllers is derived from inversion of trained neural network emulators. Two indirect adaptive controllers have been synthesized respectively to achieve the self-balancing and yaw motion control so that the efficient steering of the vehicle is achieved. Through the simulation the identification of the two subsystems using RBFNN is shown and the performance of the proposed indirect adaptive controller is also shown. Since controller is independent of the vehicle's mathematical dynamic model, the presence of uncertainty and plant unknown and unmodelled parameters and friction do not affect the performance of the proposed controller.

CHAPTER – 5

CONCLUSION AND FUTURE SCOPE

In this dissertation report, the mathematical modeling of the self-balancing two-wheeled transporter is derived. With the decomposition of SBTWT into two subsystems: yaw motion and mobile inverted pendulum subsystem, the overall controller is synthesized with the two controller: yaw motion controller and self-balancing controller. The feedback linearization technique is used for trajectory tracking and control of this vehicle and its performance is evaluated in the variation of system parameters and for different weight riders and comparison with the existing state-feedback controller. This technique shows much improved response on grounds of settling time, overshoot and steady state error in comparison to state-feedback controller.

Another controller is also presented using inversion of trained neural network emulator of the system. This is an indirect adaptive controller which is not depends upon dynamic model of system rather depends upon identified model of the system that adapts environment changes, system parameter changes and uncertainty as well.

As future research suggestion to design indirect adaptive controller different inversion algorithms of neural network emulator can be applied to further reduce settling time of the system. Any adaptive control schemes can be used to design the controller for SBTWT which steer the vehicle properly in different terrains such as trails, bike paths or beachfronts. With the advent of modern technology, such transporters with sophisticated safety features can be cost down so that they, like traditional bicycles, have high potential to become prevalent two-wheeled scooters, satisfying human transportation requirements.

REFERENCES

- [1] Segway Simply moving Segway Inc., 2007 [Online]. Available: <http://www.segway.com>
- [2] F. Grasser, A. D. Arrigo, and S. Colombi, "JOE: A mobile, inverted pendulum," *IEEE Trans. Ind. Electron.*, vol. 49, no. 1, pp. 107–114, Feb. 2002.
- [3] S. C. Lin and C. C. Tsai, "Development of a self-balancing human transportation vehicle for the teaching of feedback control," *IEEE Trans. Educ.*, vol. 52, no. 1, pp. 157–168, Feb. 2009.
- [4] C. C. Tsai, H. C. Huang, S. C. Lin, "Adaptive Neural Network Control of a Self-balancing Two-wheeled Scooter," *IEEE Transactions on Industrial Electronics*, vol. 57, no. 4, April 2010.
- [5] T. Blackwell, Building a Balancing Scooter, 2007 [Online]. available: <http://www.tlb.org/scooter.htm>
- [6] C. C. Tsai and S. C. Lin, "Intelligent Adaptive Motion Control Using Fuzzy Basis Function Networks for Self-Balancing Two-Wheeled Transporters" *Fuzzy Systems (FUZZ), 2010 IEEE International Conference on*, vol., no., pp.1-6, 18-23 July 2010
- [7] C. C. Tsai, H. C. Huang, S. C. Lin, "Direct Adaptive Fuzzy-Basis-Function-Network Motion Control for Self-Balancing Two-Wheeled Transporters" *Systems Man and Cybernetics (SMC), 2010 IEEE International Conference on*, vol., no., pp.1191-1197, 10-13 Oct. 2010
- [8] K. Pathak, J. Franch, and S. K. Agrawal, "Velocity and position control of a wheeled inverted pendulum by partial feedback linearization," *IEEE Trans. Robotics and Automation*, vol.21, no.3, pp.505-513, Jun. 2005.
- [9] Y.-S. Ha and S. Yuta, "Trajectory tracking control for navigation of the inverse pendulum type self-contained mobile robot," *Robot. Autonom. Syst.*, vol. 17, pp. 65–80, 1996.
- [10] H. Ohara and T. Murakami, "A Stability Control by Active Angle Control of Front-Wheel in a Vehicle System" *IEEE Trans. Ind. Electron.*, vol.55, no.3, pp.1277-1285, Mar. 2008.
- [11] I. Baturone, F. J. Moreno-Velo, V. Blanco, and J. Ferruz, "Design of Embedded DSP-Based Fuzzy Controllers for Autonomous Mobile Robots" *IEEE Trans. Ind. Electron.*, vol.55, no.2, pp.928-936, Feb.2008.
- [12] C. H. Chen, M.Y. Cheng "Implementation of a Highly Reliable Hybrid Electric Scooter Drive" *IEEE Trans. Ind. Electron.*, vol.54 no.5, pp.2462-2473, Oct. 2007.

- [13] N. Mutoh, Y. Hayano, H. Yahagi, and K. Takita, "Electric Braking Control Methods for Electric Vehicles with Independently Driven Front and Rear Wheels" *IEEE Trans. Ind. Electron.*, vol.54, no.2, pp.1168-1176, Apr. 2007.
- [14] Choi JY, Farrell JA, "Nonlinear adaptive control using networks of piecewise linear approximators" *IEEE Trans Neural Networks* 2000; 11(2):390-401
- [15] Hornik K, Stinchcombe M, White H., "Multilayered feed forward networks are universal approximators" *Neural Networks* 1989;2:359-66.
- [16] Powell MJD., "Radial basis function approximations to polynomials" *Proceedings 12th Biennial Numerical Analysis Conference*, 1987
- [17] Narendra KS, Parthasarathy K., "Identification and control of dynamic systems using neural networks" *IEEE Trans Neural Networks* 1990: 1(1):4-27.
- [18] Jordan MI., "Supervised learning and systems with excess degree of freedom" *COINS Tech. Rep. no. 88-27*, 1988.
- [19] Miyamoto H, Kawato M, Setoyama T, Suzuki R., "Feedback error learning neural networks for trajectory control of a robotic manipulator" *Neural Networks* 1988;1:251-65.
- [20] Nguyen D, Widrow B., "Neural networks for self-learning control systems" *Int J Control* 1991;54(6):1439-51.
- [21] Gomi H, Kawato M., "Neural network model control for a closed loop system using feedback-error learning" *Neural Networks* 1993;6(7):933-46.
- [22] Hwang JN, Choi JJ, Oh S, Marks II RJ., "Query based learning applied to partially trained multilayered perceptrons" *IEEE Trans Neural Networks* 1991;2(1):131-6.
- [23] Hoskins DA, Hwang JN, Vagners J., "Iterative inversion of neural networks and its application to adaptive control of nonlinear systems" *IEEE Trans Neural Networks* 1992;3(2):292-301.
- [24] Behera L, Gopal M, Chaudhury S., "On inversion of rbf network and its application to adaptive control of nonlinear systems" *Proc IEE Control Theory Appl* 1995;143(6).
- [25] Behera L, Gopal M, Chaudhury S., "On adaptive trajectory tracking of a robot manipulator using inversion of its neural emulator" *IEEE Trans Neural Networks* 1996;7(6):1401-14.
- [26] Poznyak AS, Sanchez EN, Wu W, Perez JP., "Nonlinear adaptive trajectory tracking using dynamic neural networks" *IEEE Trans Neural Networks* 1999;10(6):1402-11.
- [27] Ahmed MS., "Neural net based mrac for a class of nonlinear plants. *Neural Networks*" 2000;13:111-24.

- [28] Chen S, Cowans CFN, Billings SA, Grant PM. Parallel recursive prediction error algorithm for training layered neural networks. *Int. J Control* 1990;51(6):1215–28.
- [29] Iiguni Y, Sakai H, Tokumaru H. A real-time learning algorithm for a multilayered network based on extended kalman filter. *IEEE Trans Signal Process* 1992;40. Neural networks for self-learning control system
- [30] Haykin S., " *Adaptive filter theory*". Englewood Cliffs, NJ: Prentice Hall; 1991.
- [31] Haykin S., " *Neural networks, a comprehensive foundation*" NY: Macmillan College Publishing Company; 1994.
- [32] Goodwin GC, Sin KS., " *Adaptive filtering, prediction and control*" Englewood Cliffs, NJ: Prentice Hall; 1991
- [33] Lee S, Kill RM., "Inverse mapping of continuous functions using local and global information" *IEEE Trans Neural Networks* 1994;5(3):409–23
- [34] R.O. Saber, Nonlinear control of underactuated mechanical systems with application to robotics and aerospace vehicles , PhD Thesis, MIT, 2001
- [35] Akhtaruzzaman, M.; Shafie, A.A. "Modeling and control of a rotary inverted pendulum using various methods, comparative assessment and result analysis," International Conference on Mechatronics and Automation (ICMA), vol., no., pp.1342-1347, 4-7 Aug. 2010
- [36] Krishen, J.; Becerra, V.M.; "Efficient fuzzy control of a rotary inverted pendulum based on LQR mapping," *IEEE International Symposium on Intelligent Control*, vol., no., pp.2701-2706, 4-6 Oct. 2006
- [37] ozbek, N.S., Efe M.O. , "Swing up and stabilization control experiments for a rotary inverted pendulum An educational comparison," *Systems Man and Cybernetics (SMC)* vol., no., pp.2226-2231, Oct. 2010
- [38] Khanesar, M.A.; Teshnehlab, M.; Shoorehdeli, M.A. "Sliding mode control of Rotary Inverted pendulum," *Control & Automation, 2007. MED '07. Mediterranean Conference on* , vol., no., pp.1-6, 27-29 June 2007
- [39] Tatikonda, R.C.; Battula, V.P.; Kumar, V. "Control of inverted pendulum using adaptive neuro fuzzy inference structure (ANFIS)," *Circuits and Systems (ISCAS), Proceedings of 2010 IEEE International Symposium on* , vol., no., pp.1348-1351, May 30 2010-June 2 2010
- [40] Jang, J.-S. R., "ANFIS: Adaptive-Network-based Fuzzy Inference Systems," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 23, No. 3, pp. 665-685, May 1993

PUBLICATIONS

- [1] Agrawal R.; Mitra R.” Adaptive Neuro Fuzzy Inference Structure Controller for Rotary Inverted Pendulum,” advances in Intelligent and Soft Computing (Springer series), July 2012. (*Accepted for presentation*)
- [2] Agrawal R.; Mitra R.” Self-balancing and Yaw Motion Control of SBTWT Using Feedback linearization, “ IEEE International Conference on Advances in Power Conversion and Energy Technologies, APCET- Aug. 2012. (*Accepted for presentation*)
- [3] Agrawal R.; Mitra R.” Indirect Adaptive Neural Network control For Self-Balancing Two-Wheeled Transporter, IEEE International Conference on Computing Communication and Networking Technologies, ICCCNT- July 2012 (*communicated*)

INDIAN INSTITUTE OF TECHNOLOGY ROORKEE
ROORKEE - 247 667, INDIA

GRADE SHEET 2011-12 SESSION

AUTUMN SEMESTER

ENROLLMENT NO. : 10532007

NAME : RAHUL AGRAWAL

CLASS: M.TECH. E & CE (SYSTEM MODELING AND CONTROL) II YEAR

SUBJECT CODE	TITLE OF SUBJECT	CREDITS	GRADE OBTAINED
EC-601	SEMINAR	2	B
EC-602	PROJECT	4	A
EC-603	DISSERTATION	0	S

* Additional subject(s) not considered for awarding credits.

REGISTERED CREDIT IN SEMESTER : 6
EARNED CREDIT IN SEMESTER : 6
TOTAL EARNED CREDITS : 41
S.G.P.A. : 8.33
C.G.P.A. : 7.49

DATE : February 09, 2012

Prepared by..... Checked by..... ASSTT. REGISTRAR (PGS&R)