

# IMPLEMENTATION OF HEADING AND VELOCITY CONTROL ALGORITHMS FOR FRONT WHEEL STEERED AND DIFFERENTIAL-DRIVE ROBOTS

Authors: Sofia Yousuf<sup>1</sup>, Muhammad Bilal Kadri<sup>1</sup>

<sup>1</sup>Department of Mechatronic Engineering, Karachi Institute of Economics & Technology, Karachi, Pakistan  
sofia.yousuf@kiet.edu.pk (Sofia Yousuf), bilal.kadri@kiet.edu.pk (Bilal Kadri)

Received: 31-January-2021 / Revised: 20-February-2021 / Accepted: 23-February-2021

Karachi Institute of Economics and Technology || Technology Forces Journal, Issue 2, Volume 3, 2021

## ABSTRACT

In this paper, various algorithms are implemented for both the front-wheel steered (Ackermann) as well as differential drive robots. For the Ackermann Car-like robot, the robot motion can be modelled using the steering angle as well as the translational speeds whereas the differential drive (DD) robot motion is described by its left and right wheel velocities. The differential drive robot is a two-wheeled robotic system with two independent actuators for controlling the velocities of each of the two wheels with a common axis. For the testing of various heading and velocity control algorithms, MATLAB as well as the Virtual Robotic Experimentation Platform (V-REP) by Coppelia Robotics was employed. All the controllers for each of the two robots were implemented in MATLAB. For testing the performance of each controller, a separate MATLAB script was written to interface MATLAB with V-REP software utilizing various MATLAB Remote API functions and various linear and non-linear controllers were tested. For simulation, the models for both the Ackermann and differential drive robots were utilized from built-in mobile robot library provided by the V-REP software.

**Key Words :** Ackermann, Differential Drive, V-REP, Front-wheel Steered Robot, Heading Control.

## 1. INTRODUCTION

Mobile robotics being a young field, it has its roots in multidisciplinary areas such as electrical, electronics, mechanical engineering as well as computer and cognitive sciences. Mobile robots are a particular class of robots that have the capability to move as compared to fixed robotic manipulators. Robot manipulators such as robotic arms are widely used in various industrial and commercial applications, however, the major disadvantages of these robotic platforms are the limited ranges for motion and they lack mobility [1].

Mobile robots can be majorly classified into following types including ground-based robots known as Unmanned Ground Vehicles known as (UGVs),

aerial robots more commonly known as Unmanned Aerial Vehicles (UAVs) and Unmanned Underwater Vehicles robots [2, 3]. Apart from mobility, other notable features (advantages) of these robots include the certain level of autonomy and the ability of sensing and perception necessary for interaction with the environment. For the different sensing technologies employed in mobile robotics, the interested reader is referred to [4].

Also, mobile robots have been substantially proved to effective in various applications domains such as military, healthcare, homes, security, rescue and various industrial as well as commercial applications [5]. One of the major challenges in mobile robotics is the determination of exact position information of

robot during its travel from its starting position to some destination [6]. Localization refers to a method for the estimation of robot x and y position co-ordinates utilizing sensor measurements [7,8]. Similarly, mobile robot autonomous navigation essentially depends on robot path-planning and self-localization for effective control of the robot in its environment. Robot path-planning can be considered as an extension of robot self-localization dealing with a robot's goal position as well as robot's current position in some reference frame. This navigation capability of a mobile robot becomes important specifically in way-finding and collision avoidance. However, robot navigation becomes difficult in situations where the prior knowledge of the environment is not known [9-11].

In this paper, various linear non-linear and hybrid control algorithms have been implemented and tested for autonomously navigating mobile robots towards the desired goal position and moving it in some desired pre-defined trajectory. For simulation purpose, the Ackermann and Pioneer P3-DX robot built-in models in V-REP software have been utilized. Individual scripts have been written in MATLAB programming environment to establish connections with V-REP robot experimentation platform software. A number of important MATLAB remote API functions have been utilized for interfacing MATLAB with V-REP and controlling robots in V-REP with commands implemented in MATLAB. More than a hundred of these MATLAB remote API functions are available at Coppelia Robotics website [12].

This paper is structured as follows: Section 2 briefly describes the kinematic models of the front wheel steered as well as the differential-drive robot. Section 3 discusses the heading and velocity control algorithms for Ackermann robot. Section 4 comprises of various heading and velocity control strategies for the differential drive robot. In Section 5, the simulation results and performance of different controllers are presented. Finally, Section 6 concludes the paper.

## 2. Kinematics Of Mobile Robots

### 2.1 Kinematics of Ackermann Robot

Today, various mobile work machines and road vehicles utilize the well-known mechanical steering mechanism of the four-wheeled Ackermann robot which uses a four-bar linkage system as shown in Fig. 1 [13, 14]. The

Ackermann robot or the Front-Wheeled steered robot is also known to be a "Car-like Robot". The Ackermann model in a V-REP scene utilized in the simulated experiments described in this paper is shown in Fig.2.



Fig.1 Ackermann Robot (Adapted from [14])

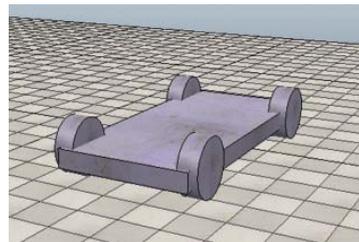


Fig.2 Ackermann robot model in V-REP Scene

The kinematics of Ackermann Robot can be mathematically expressed as:

$$\begin{aligned} v_x &= 0 \\ V &= V_{rear\_wheel} \\ \psi &= \frac{V_{rear\_wheel}}{L} \tan(\alpha) \end{aligned} \quad (1)$$

### 2.2 Kinematics of Differential Drive Robots

The Pioneer P3-DX robot shown in Fig.3 is an example of a differential-drive robot and is considered to be the most popular robotic platform for research and experimentation purposes.



Fig.3 Pioneer P3-DX Robot (Adapted from [15])

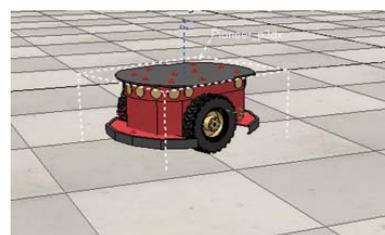


Fig.4. Pioneer Robot P3-DX model in V-REP Scene

The kinematic equations of motion of a differential-drive robot are given as [16]:

$$\begin{aligned}
 v_x &= 0 \\
 v_y &= \frac{(v_{left} + v_{right})}{2} \\
 \dot{\psi} &= \frac{(v_{right} - v_{left})}{w}
 \end{aligned} \quad (2)$$

### 3. Velocity And Heading Control Of The Ackermann Robot

This section presents the details of the various linear, non-linear as well as the hybrid controllers for the velocity as well as steering angle control, i.e., the heading control of Ackermann robot. The various control strategies were tested using MATLAB and VREP cross-platform. The details of the State Feedback Control (SFBC) mechanism implemented for the velocity as well as steering control is presented Section 4.

To navigate a robot with a particular heading towards a certain goal (destination), the desired robot heading angle can be described utilizing a pre-determined goal position as well as current robot position coordinates as:

$$\psi_{des} = -\tan\left(\frac{x_{des} - x}{y_{des} - y}\right) \quad (3)$$

The various control strategies have been simulated and tested for performance using MATLAB and V-REP environments. In V-REP and MATLAB based simulation, the following important MATLAB Remote API functions listed in Table-I were used. For MATLAB client, these remoteAPI functions start with the prefix 'simx'. In order to utilize the remote API functions three files namely remoteApi.dll, remoteApiProto.m and remApi.m must be copied into MATLAB working directory at MATLAB path along-with the separately written MATLAB script. Care must be taken selecting the correct remoteApi.dll files from V-REP installation folder, since it should match the 32-bit or 64-bit architecture of the MATLAB installed on the system to avoid errors during simulation [17].

Table I: Matlab Remote API Functions

S.no	RemoteAPI Functions <sup>a,b</sup>	Description
1.	simxStart	Starts a communication thread with server
2.	simxStart	Starts the simulation

Simulation		
3.	simxGetObject Handle	Retrieve the object handle
4.	simxGetObject Orientation	Retrieve the Euler Angles
5.	simxGetObject Velocity	Retrieve the object angular/linear velocity
6.	simxGetObject Position	Retrieve the object (x,y,z) position
7.	simxFinish	Stop communication thread
8.	simxSetJoint TargetPosition	Set target position of joint
9.	simxSetJoint TargetVelocity	Set target position of velocity
10.	simxStop Simulation	Stops the simulation

<sup>a</sup> Functions used in the simulation; <sup>b</sup> For MATLAB synopsis refer to [12]

### 3.1 Proportional Control

This type of steering controller is mathematically expressed by equation (5). Using the small angle approximation,

$$\tan(\alpha) \approx \alpha \quad (4)$$

$$\alpha = K(\psi_{des} - \psi) \quad (5)$$

The above equation shows that the steering angle is proportional to the error in heading angle. The proportional controller gain K can be adjusted with suitable value for desired response. The steering control signal  $\alpha$  increases when the error between desired heading and the actual heading of the robot is larger and vice-versa.

### 3.2 Sliding Mode Control (SMC)

This controller is also known as the 'Bang-Bang' controller. The Sliding Mode Controller is a popular technique employed for controlling complex non-linear systems specially in the presence of uncertainties. One of the major advantages of SMC controllers is the effectiveness against disturbances. Another advantage of SMC controllers is the rapid convergence which makes it widely applicable to mobile systems such as autonomous robots and vehicles [18]. Here, the controlling actions are represented as 'discontinuous functions' that can be implemented with lesser complexity. The SMC

controller is mathematically expressed as:

$$\alpha = \frac{\pi}{4} \text{sgn}(\psi_{des} - \psi) \quad (6)$$

In the above equation, the *sgn* function is known as *signum* function (Latin for 'sign'). The signum function extracts the sign of a non-zero number and exhibits the characteristic of indeterminacy at zero. This is the reason behind the fact when the heading error in eq.4 becomes zero, a highly undesirable, *chattering* effect (i.e., the back and forth switching effect of steering angle) is encountered during robot motion. The signum function is mathematically expressed as:

$$\text{sgn} = \frac{x}{|x|} \quad x \neq 0 \quad (7)$$

Using the small angle approximation in (4), this control algorithm steers the robot towards the specified destination with a steering angle of  $\pm \frac{\pi}{4} = 45^\circ$ .

### 3.3 SMC plus Proportional Hybrid Control

The SMC control strategy is suitable for driving the heading error to zero resulting in rapid convergence, however, when the heading error reaches a very small value, a chattering effect is introduced. A possible solution to deal with this undesirable condition is to utilize a hybrid control strategy that combines the features of both rapid convergence as well as no eliminate chattering using the SMC + Proportional controller combination.

$$\alpha = K(\psi_{des} - \psi) \quad \text{when} \quad K(\psi_{des} - \psi) \geq \frac{\pi}{4} ;$$

And

$$\alpha = \frac{\pi}{4} \text{sgn}(\psi_{des} - \psi) \quad \text{when} \quad K(\psi_{des} - \psi) \leq \frac{\pi}{4} \quad (8)$$

### 3.4 Steering Angle Derivative Control

The steering angle derivative control is described in equation (9) showing the control signal is now the derivative of steering angle, however, this control strategy is still linear.

$$\dot{\alpha} = K(\dot{\psi}_{des} - \dot{\psi}) \quad (9)$$

The steering angle can be obtained at any time-step by discrete-time integration methods, such

as Trapezoidal Integration used in this paper. The major drawback of this control strategy is that it produces oscillations by introducing imaginary poles in the closed loop system with equation (1).

### 3.5 Steering Angle Derivative Control with Feedback

A much better control strategy dealing with steering derivative control signals is to incorporate a feedback term which is based on the steering angle (again using the small angle approximation), the purpose of this additional term is to provide a damping mechanism to minimize the oscillatory effect introduced by the steering derivative control described in equation (9).

$$\dot{\alpha} = K_1(\psi_{des} - \psi) - K_2 \frac{V}{L} \alpha \quad (10)$$

*Feedback (Damping) Term*

### 3.6 Velocity Control

Apart from the steering control strategies, the velocity of the robot can be controlled by taking into account the distance error from the destination and remaining time. The velocity controller command formulations based on this technique are described as follows:

$$V_{des} = \frac{\sqrt{(x_{des} - x)^2 + (y_{des} - y)^2}}{t_r} \quad (11)$$

The above velocity control is simpler from implementation perspective, however, in practical situations, the velocity commanded using the above control can exceed the highest velocity that is achievable itself by the robot. This problem may arise due to various factors related to the physical attributes of the robot, such as mass, weight and most importantly the robot inertia. To cater this issue, a more realistic approach towards the robot velocity control can be defined by following expression which represents the velocity control of the robot with a saturation command  $V_{max}$  as follows:

$$V_{des} = \min \left\{ \frac{\sqrt{(x_{des} - x)^2 + (y_{des} - y)^2}}{t_r}, V_{max} \right\} \quad (12)$$

### 3.7 Velocity Derivative Control

The expression for velocity control in (11) assumes

that the robot will attain the value directed by the control velocity signal instantaneously. A good choice for the control strategy could be to utilize a velocity rate control with a certain delay or more specifically with a time-constant  $\tau$ :

$$\dot{V} = \frac{\min\{V_{des}, V_{max}\} - V}{\tau} \quad (13)$$

### 3.8 Non-Linear Steering Control

The SMC control algorithm defined in equation (6) is an example of a non-linear heading control strategy. The major disadvantage with this control is quick control switching or the chattering effect. From stability perspective, the Lyapunov function for this controller is negative if the disturbances, delays or lags are within the distance  $L$  the robot reaches the final destination. Here,  $L$  represents the length of the robot.

To ensure stability for distances greater than  $L$ , one approach is to utilize the following non-linear steering control of the form:

$$\tan \alpha = \sin(\psi_{des} - \psi) \quad (14)$$

This control strategy restricts the control output to,  $\pm$  rendering a small magnitude output when the heading error goes to zero. With these output limits, the steering angle  $\alpha$  is restricted to  $\pm \frac{\pi}{4}$  radians. The advantage is that it eliminates the chattering effect, the disadvantage is that it does not provides rapid convergence towards the desired heading.

### 3.9 Non-Linear Steering Hybrid Control

To exploit the benefits of both the characteristics of rapid convergence towards destination with no chattering, a nonlinear hybrid control mechanism obtained by the synergy of SMC with the non-linear tangent steering control could be an optimal solution. This non-linear hybrid control strategy is defined as follows:

$$\alpha = \frac{\pi}{4} \text{sgn}(\psi_{des} - \psi) \quad \text{when} \quad |\psi_{des} - \psi| > th$$

$$\tan \alpha = \sin(\psi_{des} - \psi) \quad \text{when} \quad |\psi_{des} - \psi| < th \quad (15)$$

The above hybrid control provides rapid convergence when the heading error  $|\psi_{des} - \psi|$  is larger than a certain threshold  $th$ . Additionally,

this method eliminates the chattering effect due to tangent steering control when the heading error is closer to the desirable angle.

## 4. State-Feedback Control Design For Reference Trajectory

### 4.1 Generation of Reference Elliptical Trajectory

This section describes the implementation of a state-feedback control mechanism for steering the Ackermann robot on a pre-defined *reference* trajectory. Here, the reference trajectory  $T$  is selected to be elliptical in shape, described by the following equation:

$$T = |(r \sin(2\pi Ft) + jr \cos(2\pi Ft))| \quad (16)$$

Where, the parameter  $r$  is assigned the value  $r=4$  m, which was selected to be greater than the length  $L=2.55$  m of the Ackermann robot in V-REP simulation. It was tested and observed that the particular value of  $r$  with this strategy generated a smooth elliptical trajectory during robot manoeuvre. The parameter  $F$  in above equation was given a value :  $F=0.016$  Hz to control the rotation speed of the robot Following this trajectory, the Ackermann robot's position  $(x_r, y_r, \psi_r)$  reference coordinates are easily measurable with MATLAB API functions from V-REP simulation, these values were recorded in a .mat file for implementation of the state-feedback incremental control. The video showing the Ackermann robot elliptical trajectory generation is available online at [19].

### 4.2 Implementation of Incremental State-Feedback Controller

The open-loop incremental model for the Front-Wheeled Steered Ackermann robot along the reference trajectory is given by:

$$\begin{bmatrix} \delta \dot{x} \\ \delta \dot{y} \\ \delta \dot{z} \end{bmatrix} = \begin{bmatrix} 0 & 0 & -V_r \cos \psi_r \\ 0 & 0 & -V_r \sin \psi_r \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \delta x \\ \delta y \\ \delta \psi \end{bmatrix} + \begin{bmatrix} -\sin \psi_r & 0 \\ \cos \psi_r & 0 \\ \frac{\tan \alpha_r}{L} & \frac{V_r}{L \cos^2 \alpha} \end{bmatrix} \begin{bmatrix} \delta V \\ \delta \alpha \end{bmatrix} \quad (17)$$

In the above incremental model, the values of reference velocity, heading and steering angle  $V_r$ ,  $\psi_r$  and  $\alpha_r$  are assumed to be constant during the entire reference trajectory rendering the

above model linear. Hence, with this assumption, a linear control mechanism is applicable to be designed for this system.

In this paper, the performance of state-feedback controller is investigated with an elliptical trajectory. The above state-space model in equation 17 is equivalent to a Linear Time Invariant (LTI) system, described in its general mathematical form with the matrices A and B being constant:

$$\dot{x} = Ax + Bu \quad (18)$$

The state-feedback controller gains are represented by:

$$u = -Kx \quad (19)$$

Defining the state-feedback controller gains in terms of the state vector  $[\delta x \quad \delta y \quad \delta \psi]^T$  to be:

$$\delta V = -K_1 \delta y \quad (20)$$

$$\delta \alpha = K_2 \delta x - K_3 \delta \psi \quad (21)$$

The close-loop system is then expressed as:

$$\dot{x} = (A - BK)x \quad (22)$$

The stability of the open-loop system is checked with the MATLAB function *eig*. The matrix A was evaluated with the following values:  $V_r = 0.3$  m/s,  $\psi_r = 5^\circ$  and  $\alpha_r = 0^\circ$ , from equation (1) for heading rate.

$$A = \begin{bmatrix} 0 & 0 & -0.0851 \\ 0 & 0 & 0.2877 \\ 0 & 0 & 0 \end{bmatrix} \quad (23)$$

An LTI system is controllable if the controllability matrix is full-rank, equal to the number of states n of the system. The controllability of the above third order system was calculated using the MATLAB functions *rank* and *ctrb*.

$$\text{rank}(\text{ctrb}(A,B)) \quad (24)$$

Furthermore, for this system, we assume that all the states of the system are observable. The state feed-back controller is designed by pole-placement method and setting the desired poles at:

$$\begin{aligned} p_1 &= -0.0392 + 0.1012j \\ p_2 &= -0.0392 - 0.1012j \\ p_3 &= -0.2963 + 0.0000j \end{aligned} \quad (25)$$

The corresponding state-feedback gains are

calculated to be:

$$[K_1 \quad K_2 \quad K_3] = [0.5 \quad 0.2 \quad 2] \quad (26)$$

## 5. Velocity And Heading Control Of Differential Drive Robot

The velocity and heading control of the Pioneer P3-DX differential-drive robot can be obtained by controlling the robot's left and right wheel velocities. This section discusses both the linear as well as non-linear control algorithms for controlling the speeds and heading angles (i.e., yaw angle orientation, about the z-axis) of the Pioneer P3-DX robot for the following two specific cases of robot motion:

i. Turn-Then-Travel (T-T-T), and

ii. Turn-While-Travel (T-W-T) Control

### 5.1 Heading and Velocity Linear Control (T-T-T)

The Turn-Then-Travel control of the differential-drive robot such as Pioneer P3-DX can be achieved by steering towards the pre-defined heading by setting the velocities of the two independent wheels such that  $v_{right} = v$  and  $v_{left} = -v$ . This heading control is described as:

$$\dot{\psi} = \frac{v_{right} - v_{left}}{W} \quad (27)$$

The zero-velocity control is achieved using:

$$V = \frac{v_{right} + v_{left}}{2} \quad (28)$$

Once the desired heading is achieved by the robot, the robot velocity can be controlled to navigate it towards the desired destination with the straight-line motion using the velocity saturation commands described in equations (12-13).

### 5.2 Heading and Velocity Linear Control (T-W-T)

The simultaneous Turn-While-Travel control of the robot can be obtained by turning the robot towards the desired heading using the control command for heading rate in equation (27) and defining a non-zero desired robot average velocity command:

$$V = V_{des} \quad (29)$$

### 5.3 Heading and Velocity Non-Linear Control (T-T-T)

In the non-linear control of heading and velocity of differentially-steered robot, first it is assumed that the velocity actuation commands for the two separate wheels have been set directly, i.e.,

$$V = \frac{v_{right} + v_{left}}{2} = V_{des} \quad (30)$$

In above equation,  $V_{des}$  is set to zero until a desired robot heading angle is attained by the robot. The heading rate (angular velocity) control signal is adjusted according to the following non-linear controller commands:

$$\dot{\psi} = \text{sign}(\psi_{des} - \psi) \quad \text{when} \quad |\psi_{des} - \psi| > th$$

and

$$\dot{\psi} = \sin(\psi_{des} - \psi) \quad \text{when} \quad |\psi_{des} - \psi| \leq th \quad (31)$$

When the heading error  $|\psi_{des} - \psi|$  is less than a certain threshold value, the velocity control is made non-zero in equation 29 to navigate the robot towards the destination. This combination of the two non-linear hybrid control algorithms eliminate the chattering effect and provides rapid convergence towards zero heading error.

### 5.4 Heading and Velocity Non-Linear Control (T-W-T)

In order to achieve a stable Turn-While-Travel control of a differentially-steered robot, the strategy just described can be utilized. The only difference in the control algorithm would be to adjust the velocity control in equation (9) to a non-zero suitable value to drive the robot towards destination while achieving the desired heading with strategy defined in equation 30.

## 6. Results And Discussion

Fig.1-2 show the V-REP based simulation results for the P-controller for steering control of the Ackermann robot. The proportional controller gain  $K$  was set to  $K=5$ . It can be observed that the steering angle is proportional to heading-error of the robot. As the heading error decreases, the magnitude of the steering angle  $\alpha$  also decreases and vice-versa. In these figures, the steering angle as well as heading error slightly increases since

the robot did not stop (i.e., velocity was not set to zero) and moved towards the obstacle colliding it during the simulation.

The response of the Sliding Mode Controller is shown in Fig.3-4, where from the two plots, the effect of chattering is evident. The controller gain is selected as  $K = \pi / 4$  radians. It can be clearly observed from the two plots as the heading error of the robot goes to nearly zero (as shown in the magnified plot in Fig.4) the steering angle starts switching back and forth  $\pm 45$  degrees.

The controller action for the combined hybrid strategy for the SMC+P controller is shown in Figs.5-6. With this combination, a more effective control strategy is achieved ensuring both the rapid convergence towards smaller heading error and simultaneously eliminating the chattering effect. In this hybrid strategy, the P-controller gain is adjusted to be  $K=5$ . In Figs.7-8, the response of the steering angle derivative controller is presented. Following the assumption of small angle approximation defined in Section 3, a smaller magnitude for the proportional gain has been selected, i.e.,  $K = 0.006$ . The robot rear-wheel velocity was set to  $V=0.4$  m/s. The steering angle signal was obtained by discrete-time integration choosing a sampling time of  $dt = 0.01$  seconds. In Fig.7 sustained oscillations are evident in the steering angle obtained with this control strategy due to purely imaginary poles i.e.,  $\pm 0.0305j$ , of the close-loop system, an undesirable effect of this scheme.

The results of the steering derivative control with the introduced feed-back *damping* are depicted in Figs.9-10, it can be clearly observed from the two plots, by adding the feed-back term in the steering derivative controller command represented by equation (10), the oscillations in the steering angle are significantly reduced. The controller gains in the simulation were selected to be  $K_1 = 0.001$  and  $K_2 = 0.002$ . The rear-wheel robot velocity  $V$  was set to  $V=0.1$  m/s and the robot dimensions were set according to the V-REP-based Ackermann-Car model, i.e.,  $d=0.755$  meters and  $L=2.5772$  meters, where  $d$  represents distance between the wheels and  $L$  represents the length of the robot. The controller was designed satisfying the small angle

approximation assumption in equation (4). The initial steering angle of the Ackermann-Car was set to zero degrees.

The performance of the velocity control algorithms represented by equations (11-12) are depicted in Figs.12-13. The velocity is adjusted using as saturation velocity command, i.e.,  $V_{\max} = 0$  m/s as the robot reaches its destination.

Similarly, Figs.13-14 show the simulation results of the velocity derivative control algorithm designed to control velocity with a time-constant  $\tau = 2$  seconds with a saturation velocity of  $V_{\max} = 0.1$  m/s. This control strategy is designed to navigate the robot in its environment attaining a velocity with a certain delay defined by the time constant.

The response of the tangential non-linear steering control algorithm is shown in Figs. 15-16. This control strategy restricts the tangential steering control signal between  $\pm 1$  (Fig.15). The major drawback of this control strategy is that the convergence towards the desired heading is slow as evident from (Fig.16).

In Fig.17-18, the results of the hybrid non-linear control strategy combining the characteristics of rapid convergence towards desired heading utilizing the SMC control with the eliminated undesirable chattering effect by the tangential control is presented. It can be observed from the two graphs that although a rapid decrease in the heading error is achieved, initially, with the benefit of chattering effect of the SMC control in Fig.3-4 being eliminated, the change in the steering control becomes slower as soon as the tangential control is envisaged.

Fig.19 depicts the performance of the designed state-feedback control strategy in Section 5. For an elliptical trajectory, the performance of the linear state feed-back controller observed is not good as it can be clearly seen from Fig.19, the robot follows the reference trajectory for a very short period of time and then the movement is drifted off the actual trajectory. Hence, this controller is not a good control strategy for navigating a robot in a highly non-linear trajectory since the heading  $\psi_r$  cannot be a constant as initially assumed in matrices 'A' and 'B' in equation (17). Also, Figs.20-

22 show how the three states of the system, i.e., begin to diverge after a short period of time, thus diverting the robot from its desired reference elliptical path.

In Figs.23-24, the performance of the *linear* Turn-Then-Travel approach is presented for navigating a differentially steered two-wheeled Pioneer P3-DX robot. In this control strategy, the proportional gain for the angular velocity command was set to be  $K = 5$ , whereas the robot was driven towards its destination by first attaining the desired orientation and then travelling in a straight-line with an average velocity control signal of 0.5 m/s (Fig.24).

Similarly, the results of the Turn-While-Travel control scheme have been depicted in Figs.25-26. In this scheme, the robot turns towards its destination while attaining the desired heading simultaneously as can be observed from Fig.26. The proportional gain for the controller was set to be  $K = 5$ , while the velocity command was set to a constant non-zero value of 5 m/s driving the robot towards its destination.

Finally, the performance of the non-linear hybrid control schemes in equations (29-30) providing another stable control solution for Turn-Then-Travel as well as the Turn-While-Travel mechanism are shown in Figs.27-28 and Figs.29-30 respectively. For the Turn-Then-Travel control, the gains for the SMC controller is set to  $K_1 = 1.5$  and for the tangential control, the gain  $K_2$  was set to 1. The velocity command is set to 5 m/s as soon as the robot attains the desired heading, otherwise it is set to zero. For the Turn-While-Travel control, the values of the hybrid controller gains are adjusted to be  $K_1 = 5$  and  $K_2 = 1.2$ . The robot is commanded with a constant velocity control signal  $V = 5$  m/s during its navigation. The velocity control signal magnitude is decreased when the robot reaches its final position.

The simulations showing the above controller responses for the Pioneer P3-DX robot are available online at [19].

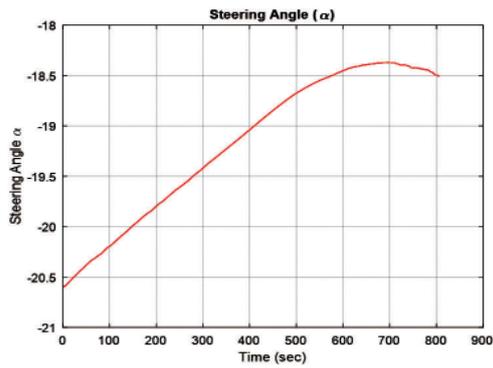


Fig.1 Steering Angle Control (P-controller;  $K=5$ )

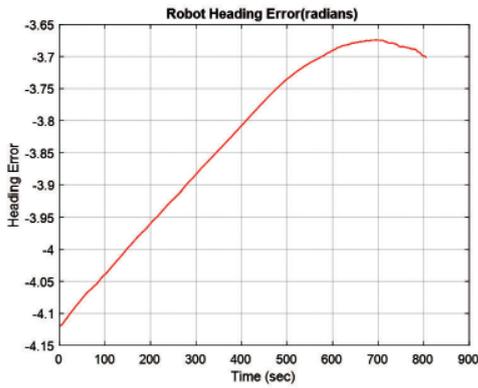


Fig.2 Heading Error (P-controller;  $K=5$ )

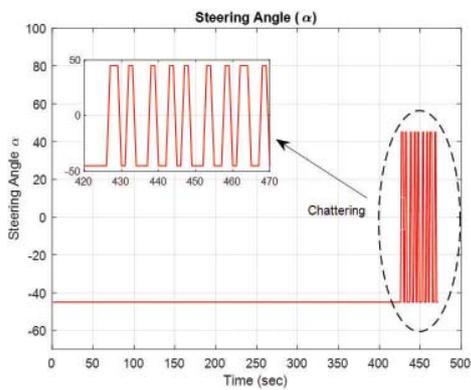


Fig.3 Steering Angle Control (SMC-Controller)

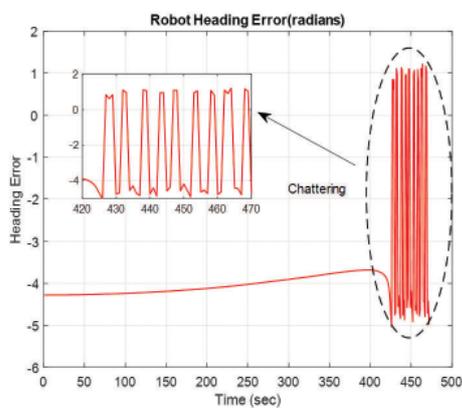


Fig.4 Heading Error (SMC-Controller)

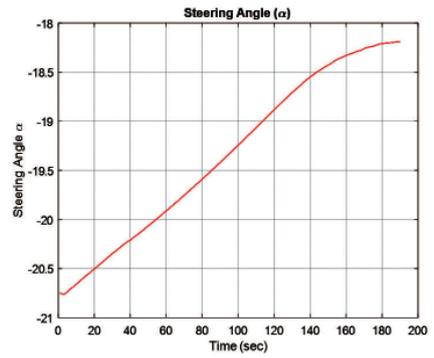


Fig.5 Steering Angle Control (SMC+P-Controller)

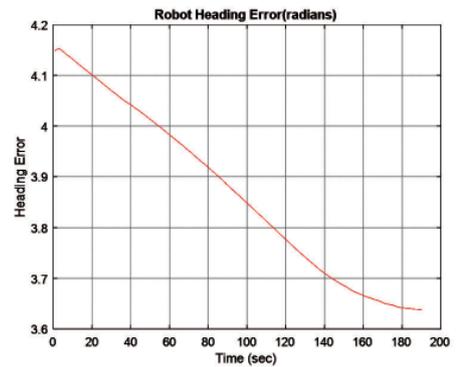


Fig.6 Heading Error (SMC+P-Controller)

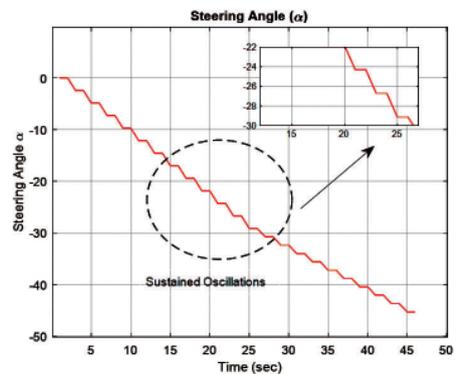


Fig.7 Steering Angle (Steering Derivative Control)

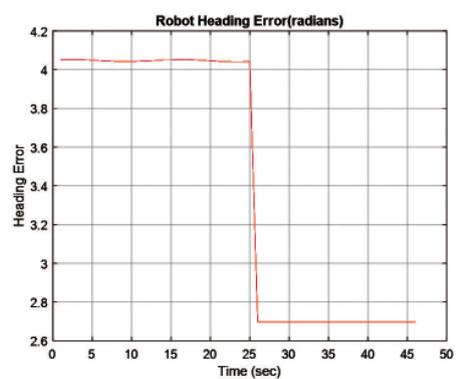


Fig.8 Heading Error (Steering Derivative Control)

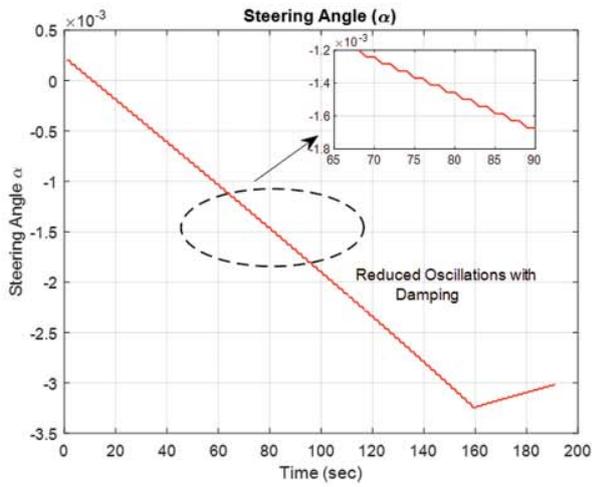


Fig.9 Steering Angle (Steering Derivative with Feedback Control)

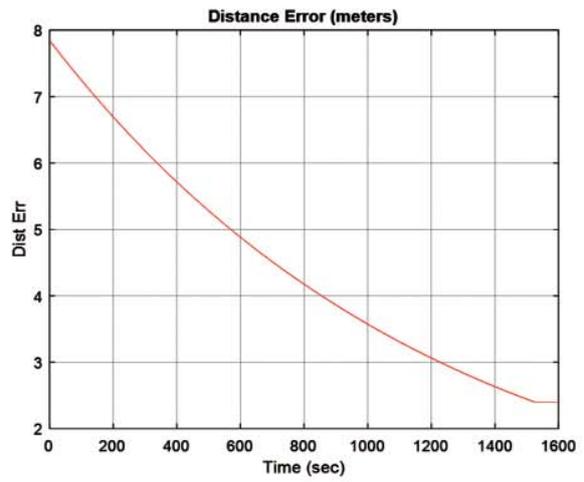


Fig.12 Distance Error (Velocity Saturation Command)

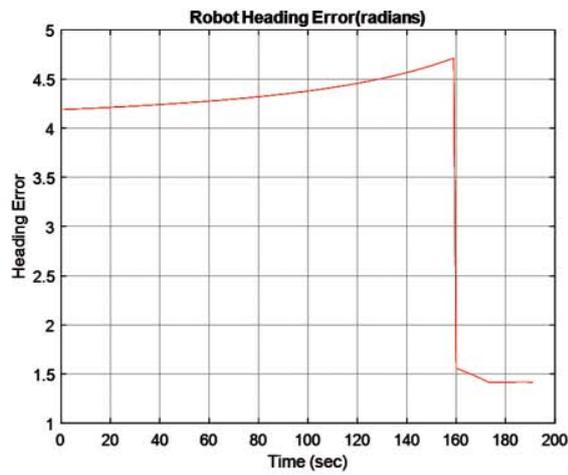


Fig.10 Heading Error (Steering Derivative Control with Feedback)

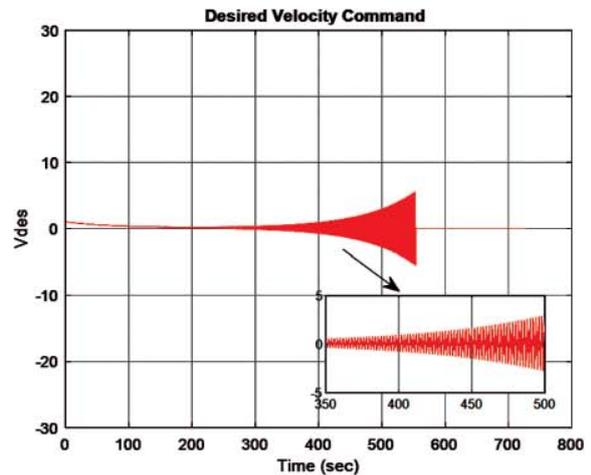


Fig.13 Velocity Derivative Control with Time-Constant

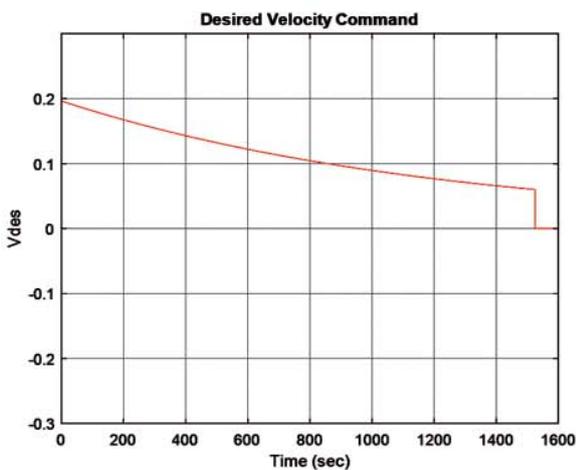


Fig.11 Velocity Control with Saturation Command

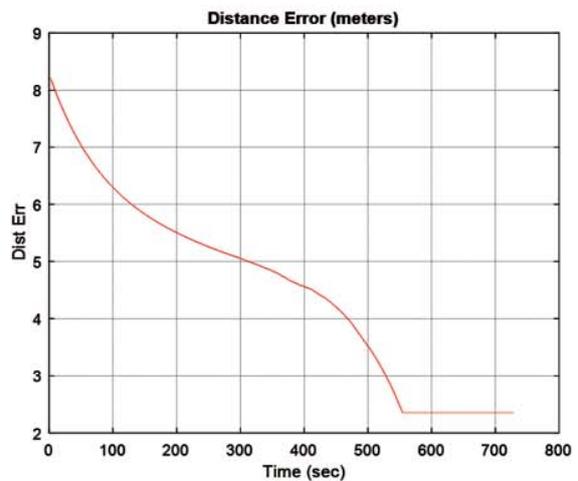


Fig.14 Distance Error (Velocity Derivative Control)

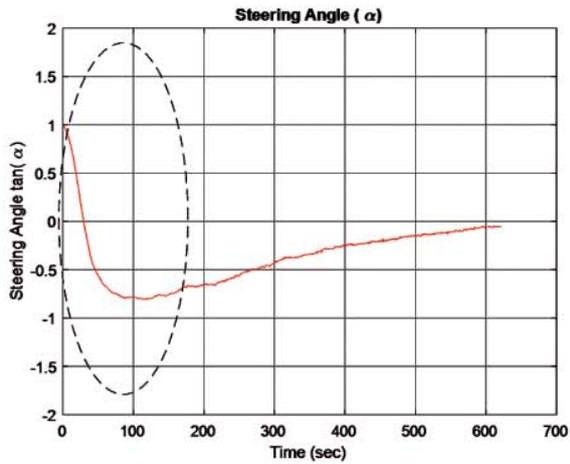


Fig.15 Steering Angle (Tangential Control with Small Angle Approximation)

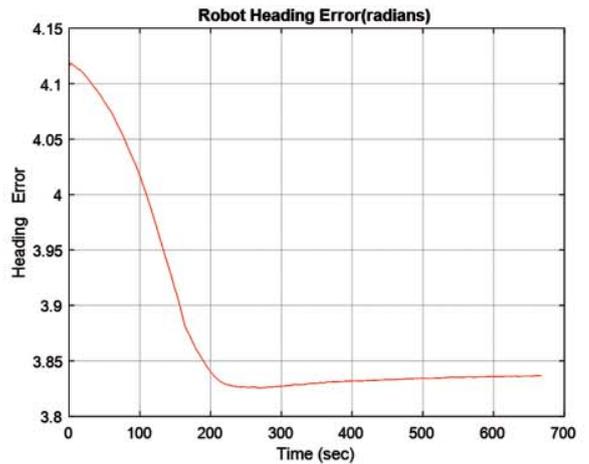


Fig.18 Heading Error (Hybrid SMC+tangential Control)

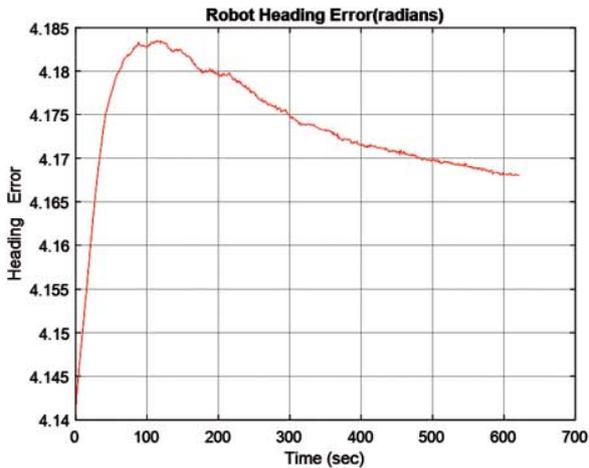


Fig.16 Heading Error (Tangential Control with Small Angle Approximation)

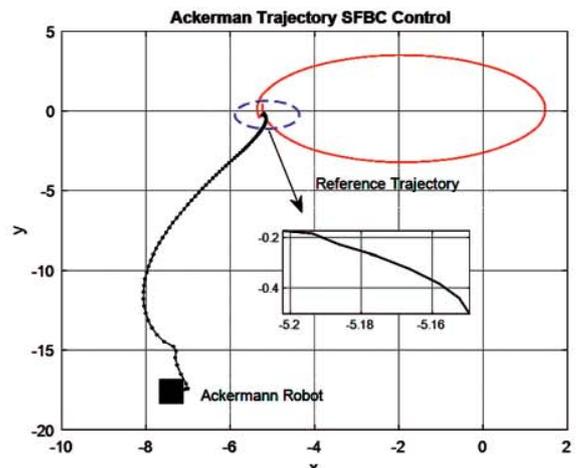


Fig.19 Response of the State-Feedback Incremental Control

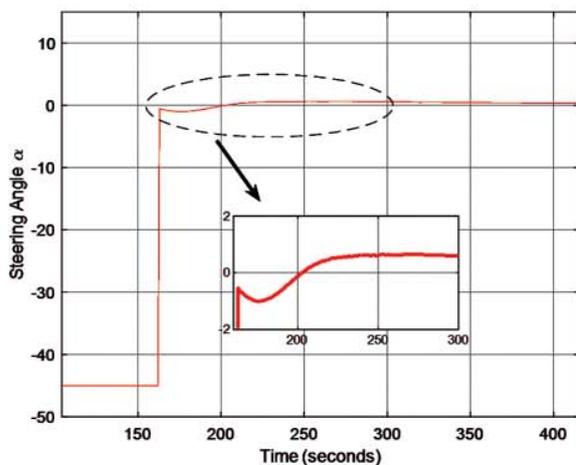


Fig.17 Steering Angle (Hybrid SMC+ Tangential Control)

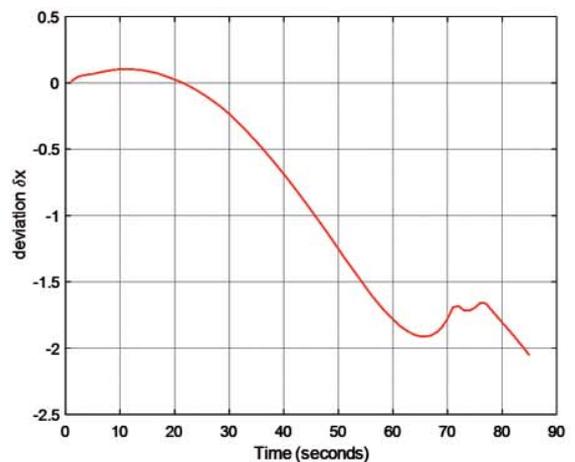


Fig.20 State Feedback Controller: Deviation in x Position

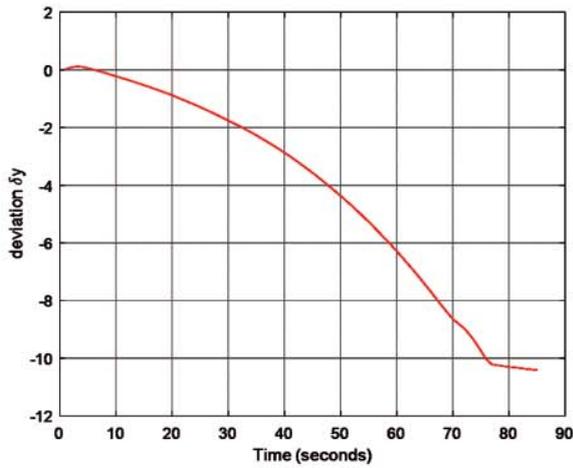


Fig.21 State Feedback Controller: Deviation in y Position

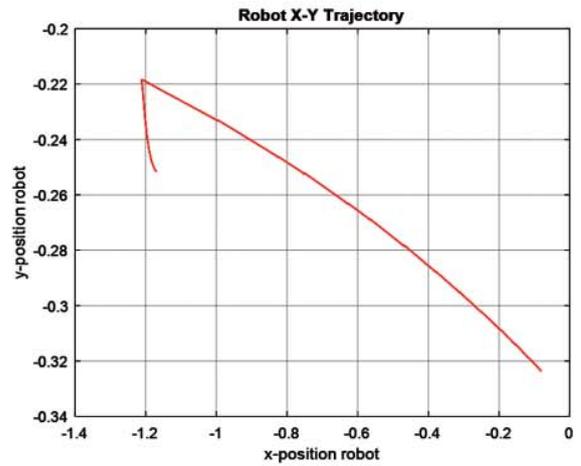


Fig.24 Robot XY Position (Turn-Then-Travel Linear Control)

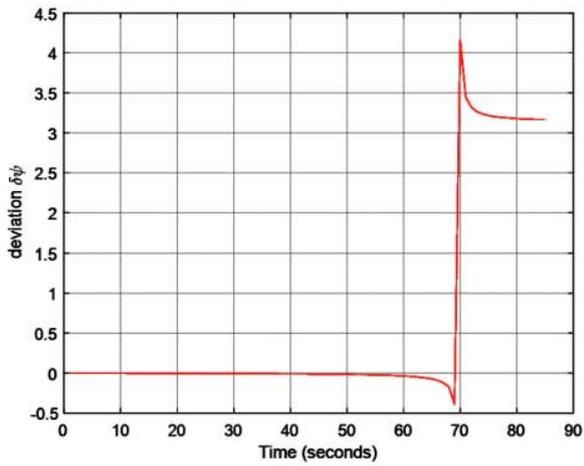


Fig.22 State Feedback Controller: Deviation in  $\psi$

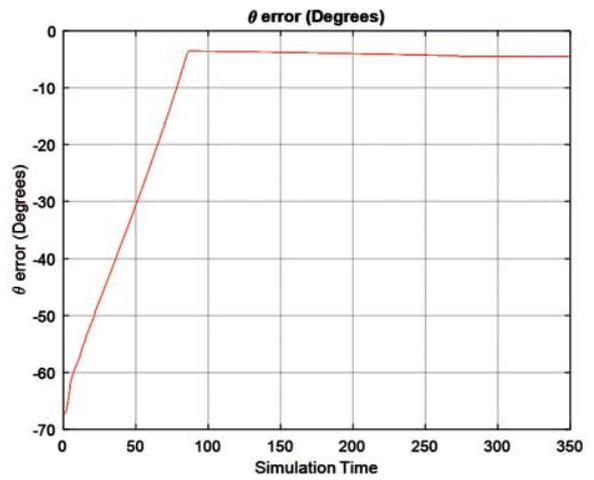


Fig.25 Heading Error (Turn-While-Travel Linear Control)

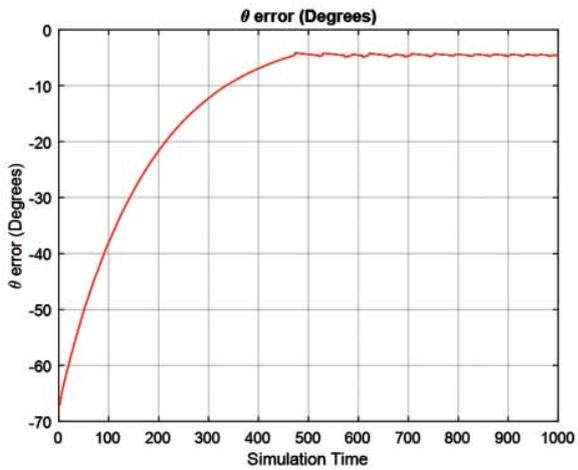


Fig.23 Heading Error (Turn-Then-Travel Linear Control)

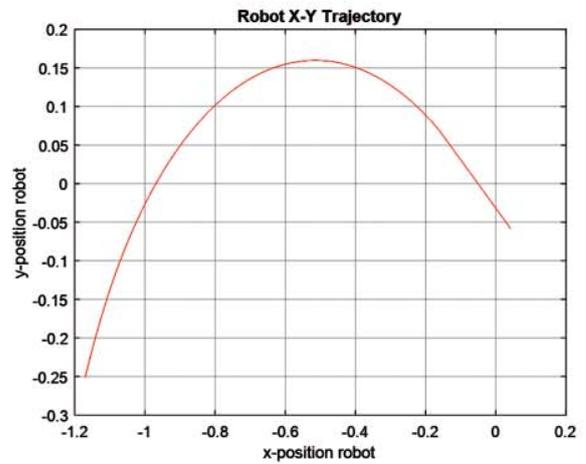


Fig.26 Robot XY Position (Turn-While-Travel Linear Control)

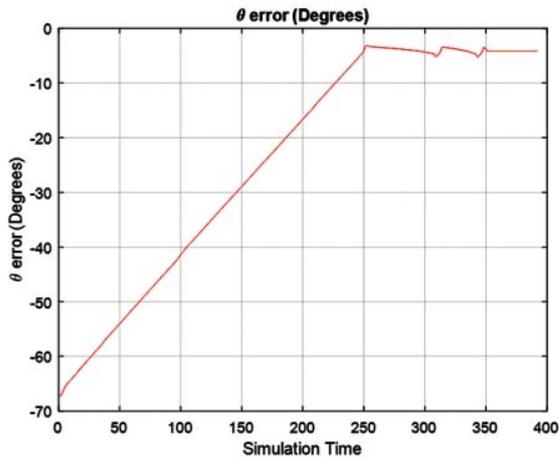


Fig.27 Heading Error (Turn-Then-Travel Non-Linear Hybrid Control)

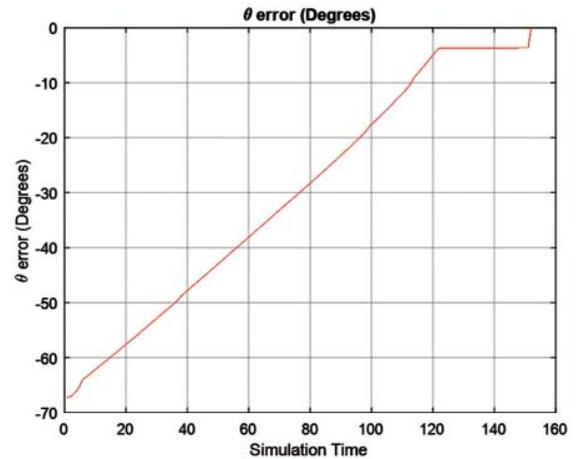


Fig.29 Heading Error (Turn-While-Travel Non-Linear Hybrid Control)

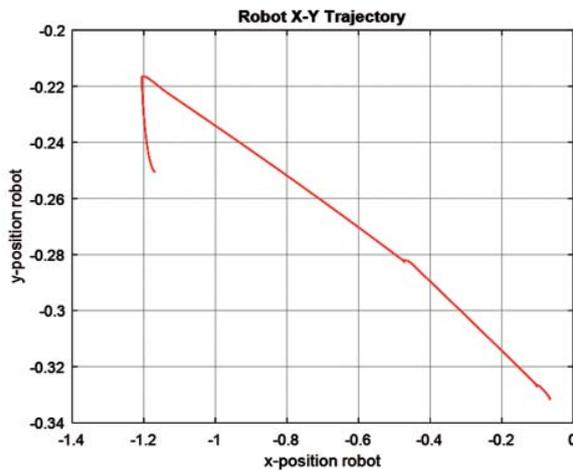


Fig.28 Robot XY Position (Turn-Then-Travel Non-Linear Hybrid Control)

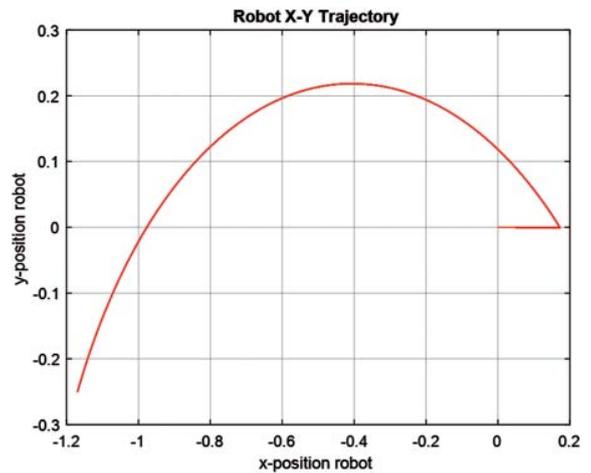


Fig.30 Robot XY Position (Turn-While-Travel Non-Linear Hybrid Control)

## 7. Conclusion

In this paper, the advantages and drawbacks of various linear, non-linear and hybrid control algorithms for the Front-Wheeled steered Ackermann robot as well as the differentially steered P3-DX robot have been studied in detail. The performance of these control algorithms has been investigated by designing simulation-based experiments with MATLAB and V-REP cross-platform.

It has been shown that the hybrid control strategies for the two robot types can yield a better control

solution such as in cases of SMC+P controller and SMC+tangential control. Also, from the simulation results, it was inferred that the performance of the state-feedback incremental controller designed in this paper did not produce satisfactory results for the case of a highly non-linear trajectory. This behaviour of the state-feedback controller under the assumption of an LTI system model (with a constant heading,  $\psi_r$ ) has been proved to be inadequate utilizing a non-linear elliptical shaped trajectory, thus, revealing the drawback of the original assumption of LTI approach for the system model.

## References

- [1] Siegwart, R., Nourbakhsh, I.R. and Scaramuzza, D., 2011. Introduction to autonomous mobile robots. MIT press. H. Simpson, Dumb Robots, 3rd ed., Springfield: UOS Press, 2004, pp.6-9.
- [2] Lima, P. and Ribeiro, M.I., 2002. Mobile robotics. Course Handouts, Instituto Superior Técnico/Instituto de Sistemas e Robótica.
- [3] Li, Y., Li, B., Ruan, J. and Rong, X., 2011, September. Research of mammal bionic quadruped robots: A review. In 2011 IEEE 5th International Conference on Robotics, Automation and Mechatronics (RAM) (pp.166-171). IEEE.
- [4] Borenstein, J., Everett, H.R., Feng, L. and Wehe, D., 1997. Mobile robot positioning: Sensors and techniques. *Journal of robotic systems*, 14(4), pp.231-249.
- [5] Tzafestas, S.G., 2018. Mobile robot control and navigation: A global overview. *Journal of Intelligent & Robotic Systems*, 91(1), pp.35-58.
- [6] Marinho, L.B., Almeida, J.S., Souza, J.W.M., Albuquerque, V.H.C. and Rebouças Filho, P.P., 2017. A novel mobile robot localization approach based on topological maps using classification with reject option in omnidirectional images. *Expert Systems with Applications*, 72, pp.1-17.
- [7] Colle, E. and Galerne, S., 2019. A robust set approach for mobile robot localization in ambient environment. *Autonomous Robots*, 43(3), pp.557-573.
- [8] Garrote, L., Torres, M., Barros, T., Perdiz, J., Premebida, C. and Nunes, U.J., 2019. Mobile robot localization with reinforcement learning map update decision aided by an absolute indoor positioning system. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*.
- [9] Lin, J., Wang, W.J., Huang, S.K. and Chen, H.C., 2017, June. Learning based semantic segmentation for robot navigation in outdoor environment. In 2017 Joint 17th World Congress of International Fuzzy Systems Association and 9th International Conference on Soft Computing and Intelligent Systems (IFSA-SCIS) (pp. 1-5). IEEE.
- [10] Yang, S. and Li, C., 2017, September. Behavior control algorithm for mobile robot based on Q-Learning. In 2017 International Conference on Computer Network, Electronic and Automation (ICCNEA) (pp. 45-48). IEEE.
- [11] Hafez, H.M., Marey, M.A., Tolbah, F.A. and Abdelhameed, M.M., 2017. Survey of Visual Servoing Control Schemes for Mobile Robot Navigation. *Scientific Journal of October 6 University*, 3(1), pp.41-50.
- [12] <https://www.coppeliarobotics.com/helpFiles/en/remoteApiFunctionsMatlab.htm>
- [13] Zhao, J.S., Liu, X., Feng, Z.J. and Dai, J.S., 2013. Design of an Ackermann-type steering mechanism. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 227(11), pp.2549-2562.
- [14] <https://www.volksbot.de/ackermann.php>
- [15] <https://www.generationrobots.com/en/402395-robot-mobile-pioneer-3-dx.html>
- [16] Dudek, G. and Jenkin, M., 2010. *Computational principles of mobile robotics*. Cambridge university press.
- [17] R Shamshiri, R., Hameed, I.A., Pitonakova, L., Weltzien, C., Balasundram, S.K., J Yule, I., Grift, T.E. and Chowdhary, G., 2018. Simulation software and virtual environments for acceleration of agricultural robotics: Features highlights and performance comparison
- [18] Utkin, V.I., 2008. Sliding mode control: mathematical tools, design and applications. In *Nonlinear and optimal control theory* (pp. 289-347). Springer, Berlin, Heidelberg.
- [19] <https://drive.google.com/file/d/1cIVED5yWzyAa8lBy0qOgxUYPd7PQiG83/view?usp=sharing>