# Simulation and mathematical modeling of Automatic Control

## of

## Wheeled Mobile Robots

Six-months practical training report

Submitted in partial fulfillment of the requirement for the award of the degree of

#### **BACHELOR OF TECHNOLOGY**

in

#### **Mechanical Engineering**

By

Subhasis Behera

Roll No.002120169

#### National Institute of Technology, Jalandhar

Under the guidance of **Prof. K. K. ISSAC** 

Work done at

Department if Mechanical Engineering, Indian Institute of Technology, Bombay Mumbai

September 2003

## CONTENTS

Acknowledgement	I
Abstract	II
List of Figures	III

## INTRODUCTION

1.1	Motivation	2
1.2	Organization of the Contents	2
1.3	Basic motion tasks	3
2	CONTROL DEVELOPMENT FOR A TRICYCLE MODEL	5
2.1	Kinematic model	6
2.2	Control development	7
2.3	The matlab model	8
2.4	Testing the strategy to reach destination points	8
2.5	The optimal values of the parameters k1 and k2	9
2.6	A simple modification in the control strategy	11
2.7	The results of the modification	11
2.8	Control Through Waypoints to Destination	12
2.9	The modified model	13
2.10	The control strategy	14
2.11	The result of simulation	14
2.12	Conclusion	16
3	CONTROL OF SINGLE-TRACK MODEL	17
3.1	The model	18
3.2	The Control Strategy	20
3.3	The results of the simulation	20
3.4	The problems encountered	21
3.5	The best values of the parameters	22
3.6	Improvements in the Control Strategy	25
3.7	A constant starting velocity	25
3.8	The result of simulation	26

3.9	Control Through Waypoints to Destination	27
3.10	The result of the simulation	27
3.11	Conclusion	28
4	CONTROL STRATEGY BASED ON THREE STEERING	29
	POSITIONS	
4.1	The model	30
3.2	The Control Strategy	32
4.3	The results of the simulation	33
4.4	The problems encountered	33
4.5	The identification and elimination of the problem	34
4.6	Modifying the model to make the variation in delta continuous	35
4.7	The result of simulation	36
4.8	Conclusion	36

## ACKNOWLEDGEMENT

I take this opportunity to express my deep sense of gratitude towards my guide Prof. K. K. ISSAC, for his support, guidance and encouragement throughout the work.

I am also thankful to my friends in IIT who helped me directly or indirectly throughout my work.

IIT, Bombay September 2003 SUBHASIS BEHERA Mechanical Engineering NIT, Jalandhar

### **Abstract:**

A detailed study on the analytical study of kinematics of WMR (Wheeled Mobile Robots) is done in the following work. The objective is to develop an appropriate kinematic model of a WMR and test various time varying feedback control algorithms on this model to control its motion from a given starting position to desired goal position. Various control strategies are reviewed and compared for trajectory tracking and posture stabilization in an environment free of obstacles. From the comparison of the obtained results, guidelines are provided for WMR end-users. Three different kinematic models are developed in the following work and tested using ode23 solver of MATLAB.

The first model is a tricycle type model having two rear wheels driven independently and a front wheel on a castor. The model is tested using a time varying smooth feedback control law satisfying liapunov's criterion for stability. Various modifications in the control strategy are tested and the results are presented. The strategy is then extended so as to make the vehicle trace a number of goal positions.

The second model is similar to a conventional vehicle in which the front wheels can be steered through a range of permitted values of angle in accordance with the longitudinal speed and length of the vehicle. Different control strategies are tested on this model and modified suitably to yield satisfactory results for all situations. This strategy is also extended so as to make the model trace a number of goal positions.

In the third model, the space in the vehicle frame of reference is divided into a number of different geometric regions. The behavior of the vehicle can be modeled in a particular way according to the presence of the vehicle in a particular region of space till it reaches to a very close vicinity of the desired point where it finally stops. A time varying control strategy is tested and the results are discussed. The problems encountered are tried to be eliminated by suitable improvements in the strategy.

## LIST OF FIGURES:

1.1	Basic motion tasks for a WMR 1	ł					
2.1	Relevant variables for the tricycle (top view) model 7						
2.2	The trajectories of the vehicle for values of parameters $k_1 = 1$ , $k_2 = 1$ 9						
2.3	The comparison of the trajectories for the optimum value of the parameters with the previous results	;					
2.4	The oscillatory nature of the trajectory when close to destination 10	)					
2.5	The result of the modification in the control strategy 11	I					
2.6	The enlarged view of the final closeness of the destination point 12	) -					
2.7	The trajectory of the vehicle while tracing a number of way-points 14	ł					
2.8	The plot of angular variation wrt the longitudinal error 15	5					
3.1	Relevant variables for the single-track model 18	}					
3.2	The trajectories for values of parameters <i>vpar</i> = 1, <i>cpar</i> = 1 20	)					
3.3	The typical motion of the vehicle as it reaches close to the destination point 21	]					
3.4	The comparison between the plots while varying the cpar, while keeping vpar constant	ŀ					
3.6	The trajectory of the vehicle with the modified strategy when the destination point lies in the lateral axis 26	t S					
3.8	The trajectory of the vehicle while tracing a number of way-points 28	3					
4.1	The diagrammatic representation of the tri-level steering input model with the different regions of space associated with it 31	;					
4.2	The trajectories of the vehicle for values of parameters: $v0 = 10$ , err = 0.01						
	and, c = 0.1	33					
4.3	The problem occurred when <i>c</i> was chosen to be, 0.01, which is equal to the width of the central region 34	1 1					
4.4	A result showing the trajectory of the vehicle, finally successful for the desired	1					
	values of parameters 36	5					

## CHAPTER 1 INTRODUCTION

#### Simulation of Automatic Control of Wheeled Mobile Robots:

Wheeled mobile robots (WMRs) have been an active area of research and development over the past three decades. This long-term interest has been mainly fueled by the myriad of practical applications that can be uniquely addressed by mobile robots due to their ability to work in large (potentially unstructured and hazardous) domains. A nonlinear control of a wheeled robot is discussed here.

#### 1.1 Objective

Wheeled mobile robots (WMRs) are increasingly present in industrial and service robotics, particularly when; flexible motion capabilities are required on reasonably smooth grounds and surfaces. Several mobility configurations (wheel number and type, their location and actuation, single- or multi-body vehicle structure) can be found in the applications. The most common for single-body robots are differential drive and synchrodrive (both kinematically equivalent to a unicycle), tricycle or car-like drive, and omni directional steering. A detailed reference on the analytical study of the kinematics of WMRs is [1]. Beyond the relevance in applications, the problem of autonomous motion planning and control of WMRs has attracted the interest of researchers in view of its theoretical challenges. In particular, these systems are a typical example of nonholonomic mechanisms due to the perfect rolling constraints on the wheel motion (no longitudinal or lateral slipping) [6]. In the absence of workspace obstacles, the basic motion tasks assigned to a WMR may be reduced to moving between two robot postures and following a given trajectory. From a control viewpoint, the peculiar nature of nonholonomic kinematics makes the second problem easier than the first; in fact, it is known that feedback stabilization [2] at a given posture cannot be achieved via smooth time-invariant control. This indicates that the problem is truly nonlinear; linear control is ineffective, even locally, and innovative design techniques are needed. As for posture stabilization, both discontinuous and/or time-varving feedback controllers have been proposed. Smooth time-varying stabilization was pioneered by Samson [7], [8]. A recent addition to this class was presented in [5], where dynamic feedback linearization has been extended to the posture stabilization problem. While comparative simulations of several of the above methods are given in [3] for a unicycle and in [4] for a car-like vehicle, there is no extensive experimental testing on a single benchmark vehicle. The objective of my work is therefore to evaluate and compare the practical design and performance of control methods for trajectory tracking and posture stabilization, highlighting potential implementation problems related to kinematic or dynamic nonidealities. Nonetheless, most of the methods selected for comparison can be generalized to vehicles with more complex kinematics.

#### 1.2 Basic motion tasks

The basic motion tasks that we consider for a WMR in an obstacle-free environment are (see Fig. 1.1):

- Point-to-point motion: The robot must reach a desired goal configuration starting from a given initial configuration.
- Trajectory following: A reference point on the robot must follow a trajectory in the Cartesian space (i.e., a geometric path with an associated timing law) starting from a given initial configuration.



Figure 1.1: Basic motion tasks for a WMR: (a) Point to point motion; (b) Trajectory following

Execution of these tasks can be achieved using either feed forward commands, or feedback control, or a combination of the two. Indeed, feedback solutions exhibit an intrinsic degree of robustness. However, especially in the case of point-to-point motion, the design of feedback laws for nonholonomic systems has to face a serious structural obstruction. The design of feed forward commands is instead strictly related to trajectory planning, whose solution should take into account the specific nonholonomic nature of the WMR kinematics.

#### 1.3. Organization of contents

In Chapter 2, a tricycle model will be modeled in MATLAB and tested using the ode23 solver. The model uses a time-varying smooth feedback control law. The model is tested for different conditions and the results of some typical stages will be presented (section 2.2 and 2.3). The problems in the results will be discussed (section 2.4), and the solutions will be proposed. Additional parameters will be introduced in the strategy, in order to improve the performance (section 2.5), and the improved results will be compared with the earlier results. Then the strategy will be extended to trace more than one point (section 2.6), that means enabling the vehicle to run through a number of waypoints. The result of that modification in the strategy is analyzed (section 2.7), and the reason of those irregularities in motion is explained. Some of the problems are solved and some could not be solved. Finally, the unsolved problems and their possible explanations are presented.

In chapter 3, another model and time varying feedback strategy was tested. The model is explained in section 3.1. The model is tested for different conditions and the results of some typical stages will be presented (section 3.2). The problems in the results will be discussed (section 3.3), and the solutions will be proposed. Additional parameters will be introduced in the strategy, in order to improve the performance (section 3.4), and the improved results will be compared with the earlier results. A typical problem was discovered in the strategy (section 3.5), while the model is required to get to very close points. The problem and the proposed solutions will be discussed. Then the strategy will be extended to trace more than one point (section 3.6), that means enabling the vehicle to run through a number of waypoints. The result of that modification in the strategy is analyzed (section 3.7). Finally, the unsolved problems and their possible explanations are presented.

In chapter 4, a single-track model using tri-level steering inputs and time varying feedback strategy was tested. The model is explained in section 4.1. The model is tested for different conditions and the results of some typical stages will be presented (section 4.2). The problems associated the results will be discussed (section 4.3). A typical problem was discovered in the strategy (section 4.4), while the model is required to stop at a very small final region. The problem and the proposed solutions will be discussed in section 4.5. The model is based on a discontinuous controller based on polar coordinates transformation. So, the problem was found to be with the discontinuity associated with the model. The detailed description of the problem is briefed in section 4.6. Some methods of tackling the problem are discussed in section 4.7. In chapter 6, using a stabilizing law based on dynamic feedback linearization solves the problem found in the previous chapter. The model is described in section 6.1. The detailed explanation of the solving of the problem is given in section 6.2. The best values for the parameters are discussed in section 6.3.

Chapter 2 Tricycle type model

#### Introduction

The kinematic model for wheeled mobile robots is described here for the nonholonomic constraint of pure rolling and non-slipping. Based on the kinematic model, the differentiable, time-varying kinematic controllers for the regulation control problem will be analyzed here. The model will be tested in different sets of conditions for a number of control strategies and the corresponding results will be analyzed.

#### 2.1 Kinematic model

The model is a tricycle type model having two rear wheels driven independently and a front wheel on a castor. The kinematic model for the nonholonomic constraint of pure rolling and non-slipping is given as follows.

Where q(t), qd(t) are defined as,

 $x_c(t)$  and  $y_c(t)$  denote the position of the center of mass of the WMR along the X and Y Cartesian coordinate frames and  $\theta_c$  (t) represents the orientation of the WMR,  $x_{cd}(t)$  and  $y_{cd}(t)$  denote the Cartesian components of the linear velocity, the matrix S(q) is defined as follows

$$S(q) = \begin{bmatrix} \cos\theta & 0\\ \sin\theta & 0\\ 0 & 1 \end{bmatrix}$$
 ... (2.4)

And velocity vector v(t) is defined as

$$\mathbf{v} = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} v_1 \\ \theta_d \end{bmatrix} \tag{2.5}$$

The control objective of regulation problem is to force the actual Cartesian position and orientation to a constant reference position and orientation. To quantify the regulation control objective, we define x(t), y(t),  $\theta(t)$  as the difference between the actual Cartesian position and orientation and the reference position as follows

$x(t) = x_c - x_{rc}$	(2.6)
$y(t) = y_c - y_{rc}$	
$\theta(t) = \theta_c - \theta_{rc}$	



Figure 2.1: Relevant variables for the tricycle (top view)

 $x_{rc}$ ,  $y_{rc}$ ,  $\theta_{rc}$  represent the constant position and orientation.

$q_1 = x\cos\theta + y\sin\theta$	
$q_2 = -x\sin\theta + y\cos\theta$	(2.10)
$q_1 = \theta$	(2.11)

Where  $q_1$ ,  $q_2$ ,  $q_3$  are the auxiliary error of the system. Taking the derivatives of the above and using the kinematic model given in equation (2.2), can be rewritten as follows

$\dot{q}_1 = v_1 + v_2 e_2$	(2.12)
$\dot{q}_2 = -v_2 e_2$	
$\dot{q}_3 = v_2$	(2.14)

#### 2.2 Control development:

The control objective is to design a controller for the transformed kinematic model given by equations (2.12), (2.13), (2.14) that forces the actual Cartesian position and orientation to a constant reference position and orientation. Based on this control objective, a differentiable, time-varying controller was proposed as follows:

$v_1 = -k_1 e_1$	(2.15)
$v_2 = -k_2 e_3 + e_2^2 \sin t$	(2.16)

Where  $k_1$  and  $k_2$  are positive constant control gains. After substituting the equations the following closed-loop error system was obtained:

$\dot{q}_1 = -k_1 q_1 + (-k_2 q_3 + q_2^2 \sin t) \cdot q_2$	
$\dot{q}_2 = -(-k_2q_3 + q_2^2\sin t) \cdot q_2$	
$\dot{q}_3 = \left(-k_2 q_3 + q_2^2 \sin t\right)$	

The control strategy adopted here is quiet simple. The linear velocity is directly proportional to the longitudinal error or the projected distance of the vehicle from the destination point alone. The rate at which the wheels should be turned is proportional to the angular orientation of the desired point with respect to the reference frame attached to the vehicle or the angular error and an additional time-varying term. This term plays a key role, when the vehicle gets stuck at a point. Such a situation occurs; when the longitudinal error of the vehicle vanishes that and it is oriented parallel to the desired direction, but the lateral error is not zero. In such a case, in the absence of the second term of the angular velocity control, the vehicle would get locked in that position and will fail to move any further; even though the vehicle has not reached the destination point. So this additional term is added to the steering control term. When the vehicle gets locked in the abovementioned situation, this term makes the angular error nonzero again and makes the vehicle turns a bit. This gives rise to a longitudinal error and the velocity is again non-zero. The term is sine function of time multiplied with the square of the lateral error. The sine term varies between -1 and 1 making this term vary in an oscillatory fashion. The lateral error term makes the quantity bigger when the lateral error is large. So, when the lateral error is large the disturbing steer is even larger. This quantity is smaller in comparison to the first quantity, so that the cyclic nature of the sine function does not affect the result much. Here  $k_1$  and  $k_2$  are positive constant control gains.

#### 2.3 The MATLAB model:

The behavior of the model and the strategy can be tested if we can obtain the trajectory of the path, when subjected to a given set of conditions. For that we need to get all the values of the state variables ( $q_1$ ,  $q_2$ ,  $q_3$ ) at small intervals of time, which can later be plotted to obtain the trajectory of the path followed. Hence the above set of first order differential equations have to be integrated in a time interval; given the values of the initial conditions and parameters using ode23; A powerful tool of matlab. Ode23 is a function for the numerical solution of ordinary differential equations. It can solve simple differential equations or simulate complex dynamical systems. It integrates a system of ordinary differential equations using 2nd & 3rd order Runge-Kutta formulas. This particular 3rd-order method reduces to Simpson's 1/3 rule and uses the 3rd order estimate for xout. The process of ode23 is as follows: A string variable with the name of the M-file that defines the differential equations to be integrated. The function needs to compute the state derivative vector, given the current time and state vector. It must take two input arguments; scalar t (time) and column vector q (state), and return output argument qdot, a column vector of state derivatives. The above set of first order differential equations was converted into the following M-file, to execute ode23.

#### 2.4 Testing the strategy to reach destination points:

The above model is modeled in MATLAB and the strategy is tested for the vehicle to reach different destination points in all the four different quadrants. The matlab program is in Appendix I(a). The following plot is a typical result of simulation, which shows the trajectories of the state variables  $q_1$  and  $q_2$ . Since in this model, the longitudinal and lateral errors are the state variables, the plot shows the errors decreasing and finally becoming zero. So the destination points in the plots are situated at zero and the

starting points are the longitudinal and lateral errors. In the following plots, the vehicle starts with error values  $e_1$  and  $e_2$  as, (5, 5). The parameters are taken as,  $k_1$  = 1 and  $k_2$  = 1.



Figure 2.2: (a) Vehicle starting from (5, 5, 0);  $k_1 = 1$ ,  $k_2 = 1$ ; (b) Vehicle starting from (-5, -5, 0);

#### **2.5** The optimal values of the parameters $k_1$ and $k_2$ :

The above result shows the plots of the lateral against the longitudinal errors of the vehicle when the parameters are chosen to be  $k_1 = 1$  and  $k_2 = 1$ . The values of  $k_1$  and  $k_2$  can be iterated to study their influence over the results and find out their optimal values that gives the best results. From the iteration, the best values of the parameters were found to be,  $k_1 = 2$  and  $k_2 = 0.1$ . A comparison between the two results is shown in the following figure:



Figure 2.3: (a) The trajectory of the vehicle when,  $k_1 = 1$ ;  $k_2 = 1$ ; (b) The trajectory of the vehicle with the optimal values of the parameters,  $k_1 = 2$ ;  $k_2 = 0.1$ ;

Form the figures it is clear that, the lateral error  $(e_2)$  could not be completely eliminated even for the optimal values of the parameters  $k_1$  and  $k_2$ . Still there remained a lateral error of about 0.5 units and it did not seem to improve much by the change of the values of the parameters. The amplitude of the longitudinal oscillations is however minimized; but the primary objective was to reduce the lateral error, which did not seem to be much affected by variation the parameters. The trajectory has however become reasonably smoother in the plot than the previous result. The vehicle now undergoes lesser wandering before reaching close to the goal position (see figure). The reason of this lesser wandering of the vehicle accounts for the following: By choosing a smaller value of  $k_2$  with respect to  $k_1$ , we are assigning lower weightage to the angular error and giving more weightage to the linear errors. This directly reduces the steering rate of the vehicle. Hence the vehicle remains in more control. To understand this, let us consider driving a real car. If the driver swings the steering wheel too fast in response to curves in the road or if the steering wheel is too much free, the vehicle experiences a lot of swagger in its motion. The motion is in control or smooth if he swings the wheel slowly, keeping a firm control over the steering wheel. This exactly happens here by choosing a lower value of the parameter  $k_2$ .

Plotting the trajectory for 100 seconds we find that the longitudinal error swings about 0 slowly n the amplitude decreases continuously while the lateral error approaches 0 at a very slow rate. This is clear from the figures shown below.



Fig 2.4: (a) the trajectory of the vehicle for 100 seconds:  $k_1 = 1$ ,  $k_2 = 1$ ; (b) the magnified view of the oscillatory nature of the motion.

The enlarged view of the final oscillatory character of the trajectory suggests that the longitudinal error swings about 0, while the lateral error very slowly nears 0. The reason for this is explained as follows. The angular velocity is sum of two components;  $-k_2e_3$  and  $e_2^2 \sin t$ . The presence of the sine term explains the orderly oscillations at the end part of the motion. Since the longitudinal and lateral errors become insignificant as the vehicle slowly proceeds towards the goal position, the sine term

becomes the predominant factor in effecting the motion of the vehicle. This makes angular speed vary regularly with time. Since the longitudinal speed decreases as the longitudinal error decreases (the vehicle approaches close to the goal position), the vehicle travels through lesser distance in the same time in which the sine term changes sign; hence resulting in decrease of the amplitude of the oscillations as the vehicle approaches the destination point.

#### 2.6 A simple modification in the control strategy:

So finally we tried to find out if any modification in the model could affect the results. A lot of modifications were tested and finally an introduction of a third constant  $k_3$ , yielded desirable results. When a constant  $k_3$  was multiplied with the sine terms, the lateral error seemed to almost reduce to zero at a much faster rate. The result was smoother trajectory and faster motion of the vehicle. Higher values of constant  $k_3$  yielded yet better results. The final oscillations seemed to almost come down to zero. The modified matlab program is in Appendix I (b).

$$v_1 = -k_1 e_1$$
 (2.20)  
$$v_2 = -k_2 e_3 + k_2 e_2^2 \sin t$$
 (2.21)

#### 2.7 The results of the modification:

The modified strategy was tested for various values of state variables and parameters. The value of the parameter  $k_3$  was iterated and it was observed that, the final result was greatly improved with the increase in the value of  $k_3$ . For value of  $k_3 = 10$ , the final lateral error was reduced to a very small value and also at a much faster rate. This is shown in the figure 3.9. Where,  $k_1 = 2$  and  $k_2 = 0.1$ . The vehicle starts with error values (5, 5, 1) and is simulated for 10 seconds. Figure 3.10 shows the result for  $k_3 = 100$  for the same values of  $k_1$  and  $k_2$ .



Figure 2.5: the trajectory of the vehicle with the modified strategy: (a)  $k_3 = 10$ ; and (b)  $k_3 = 100$ .

It is clearly seen that the result is greatly improved, by the application of this strategy. The result is also attained at a very fast rate. With the increase in the value of  $k_3$ , the result is also further improved. So in the following result the value of  $k_3$  is taken to be 100.

From the adjacent figure, we can clearly see the final position of the trajectory. The lateral error is 0 and the longitudinal error is less than  $0.005 \times 10^{-3}$ , which is small enough to neglect. So it is clearly seen how the model got modified. The lateral error is 0 in less than 3 seconds. The speed of the vehicle at this point is also small enough to assume it to be 0.



Figure 2.6: the enlarged view

From the above discussion, it is clearly seen that the final result has been greatly improved by the modification in the strategy. By choosing a higher value of  $k_3$ , we are actually, increasing the weightage of the second term in the angular velocity control term. That means the angle now changes faster than before, and hence the vehicle reaches to the point at a faster rate also. When the desired distance becomes very small, the angular change also becomes small. But by choosing higher values of  $k_3$ , it has become very fast and the amplitude has also increased, which makes the steps at which the vehicle nears the destination point, bigger. But since, the second term is a sine function of time; it will simply fluctuate about a mean value, with amplitude diminishing slowly.

#### 2.8 Controls through Waypoints to Destination:

After the objective of moving to a single point was achieved satisfactorily, the next task was to extend the strategy so as to trace a number of points. Given  $[x_i, y_i, \theta_i]$ ,  $i = 0, 1, 2, \dots, m$ , develop an algorithm to make the vehicle to reach  $x_m, y_m, \theta_m$ , starting from  $x_0, y_0, \theta_0$ , passing through the m-1 points in sequence. The above objective was attained by the following approach. A parent algorithm was made, in which all the points to be traversed were stated. The parent algorithm defines a small region of space around the way points which is known as error value. It then sets the first waypoint as the goal point for the original control strategy. When the vehicle reaches close enough to the first waypoint (in the parent algorithm), specified by the error value, it sets the next waypoint to be the destination point for the original control strategy taking its current position as the starting point. This way the vehicle travels through all the 'm-1' waypoints, till it reaches the final destination point. The control strategy used to travel between the waypoints is the same as that used to trace a single point in the previous model. The original model is however modified so that the vehicle now moves in a Cartesian coordinate plane. That means now the original coordinates of the point are taken as the state variables. Rather than taking the error values as the variables, which it did previously. The modified model is discussed in the next section.

#### 2.9 The modified model:

The model is modeled with respect to the global reference frame. The vehicle type however remains unchanged. That is, a tricycle type model having two rear wheels driven independently and a front wheel on a castor. Given a global reference plane in which the instantaneous position and orientation of the model is given by (q(1), q(2), q(3)) with respect to the global reference system. The vehicle is to start at a position (x, y,  $\theta$ ) and has to reach a given point ( $x_d$ ,  $y_d$ ,  $\theta d$ ) with respect to the global reference plane.

The longitudinal axis of the reference frame attached to the vehicle and the lateral axis perpendicular to the longitudinal axis. Since this reference frame's position changes continuously with respect to the global reference system, the instantaneous position of the origin of the reference frame attached to the vehicle is given by  $(q_1, q_2, q_3)$ . The position of the point to be traced in the reference frame attached to the vehicle, with respect to the global Coordinate system is given by  $(e_1, e_2, e_3)$ . Where,

 $e_1$  = The instantaneous longitudinal coordinate of the desired point to be traced with respect to the reference system of the vehicle.

 $e_{\rm 2}~$  = The instantaneous lateral coordinate of the desired point to be traced

with respect to the reference system of the vehicle.

 $e_3$  = The instantaneous angular coordinate of the desired point to be traced with respect to the reference system of the vehicle.

The conversions of the local values of

$$e_{1} = (x_{d} - q_{1})^{*} \cos q_{3} + (y_{d} - q_{2})^{*} \sin q_{3} \qquad \dots \qquad \dots \qquad \dots \qquad (2.22)$$

$$e_{2} = -(x_{d} - q_{1})^{*} \sin q_{3} + (y_{d} - q_{2})^{*} \cos q_{3} \qquad \dots \qquad \dots \qquad \dots \qquad (2.23)$$

$$e_{3} = \tan^{-1} \left( \frac{y_{d} - q_{2}}{x_{d} - q_{2}} \right) - \theta \qquad \dots \qquad \dots \qquad (2.24)$$

The kinematic model for the so-called kinematic wheel under the nonholonomic constraint of pure rolling and non-slipping is given as follows.

$\dot{q}_1 = v_1 * \cos q_3$	 	 	 	 	 (2.25)
$\dot{q}_2 = v_1 * \sin q_3$	 	 	 	 	 (2.26)
$\dot{q}_3 = v_2$	 	 	 	 	 (2.27)

 $v_1$  = The longitudinal velocity applied to the vehicle

 $v_2$  = The instantaneous angular deflection provided to the wheels of the vehicle

Or in other words, the angle by which the reference frame attached to the vehicle changes instantaneously. So these two variables have to be controlled by a control strategy, so that the vehicle reaches the desired point smoothly.

#### 2.10 The Control Strategy:

The control strategy is same as we are modifying the model only.

$$v_1 = -k_1 e_1$$
(2.28)  
$$v_2 = -k_2 e_3 + e_2^2 \sin t$$
(2.29)

After substituting eq(7) and eq(8) into eq(4) and eq(5) and eq(6), the following closed loop error system was developed.

#### 2.11 The results of simulation:

The above modifications are implemented in appropriate matlab models and the models are tested in ode solver. The matlab model is in Appendix I (c). Here are the results of the strategy at previous ideal values of  $k_1$ ,  $k_2$  and  $k_3$ .



Figure 2.7: (a) Here the figure shows the path followed by the vehicle, when the above strategy is applied. The values of  $k_1 = 2$ ,  $k_2 = 0.1$ ,  $k_3 = 100$ ; (b) The angular variation with respect to the x-distance.

The figure 2.7(a) represents the trajectory of the vehicle, while it traverses the three points. A first look shows it to be quite all right except that the trajectory is not very smooth. But when we plot the angular variation with respect to the x coordinates, we see that there is a large angular variation (see figure 2.7(b)). The animation of the motion of the vehicle reveals that it takes a lot of turns in between the motion, which is completely undesirable. The iterative investigation associates the reason of these

unnecessary rotations to the high value of the parameter  $k_3$ . Due to the high value of  $k_3$ , the weightage of the second term increases. Since the sine can have positive as well as negative values, the entire second term is positive at some times and negative at the other times. When it is positive, the whole sum of both the values becomes so high that it makes the increment in the angular velocity more than  $2\pi$  at one particular instant and the vehicle takes a complete revolution. This is the reason of the large angular change in that case. The vehicle takes a lot of revolutions in midway, which is not desirable. Previously this did not happen while tracing a single point, because the vehicle completed its motion before the angular speed is raised that high to make it rotate completely, that is before the sine term reaches that high. The high variation in the angle always happened while the vehicle approached close to a waypoint, which is because the vehicle slows down as it approaches the waypoints, and thus the sine term gets time to increases to high values making angular variations high.

So by lowering  $k_3$  and increasing  $k_2$  we can see that the angular variations are greatly reduced. The result of  $k_2 = 0.25$  and  $k_3 = 10$  are shown below.



Fig 2.8: (a) The angular variation q (3) with respect to the longitudinal error q (1), the waypoints to be traced is 15, 25 and 80.  $k_1 = 2$ ,  $k_2 = 0.25$ ,  $k_3 = 10$ ; (b) The angular variation versus x-coordinate for the set of intermediate points (15, 5, 0.675), (35, 45, 0.927), (80, 60, 0.54). The angular variation is limited to only 1.6 radians, less than  $\pi/2$ 

The above two results show the effect of lowering the value of the parameter,  $k_3$ . The figures are the plots of the angular orientation q (3), with respect to the longitudinal position q (1). The angular coordinate undergoes a variation of the order of about 3 radians or one half revolutions in figure 2.8(a). This proves that the result has improved a lot over the previous case where the angular variation was much higher. However the plot advocates another serious problem. After reaching close to the first waypoint (q (1) = 25) the angular orientation changes sharply by about 3 radians, which means half a revolution. That means that the vehicle will face opposite with that sharp rotation. Then instead of returning to its normal orientation, the vehicle starts moving towards the next waypoint. This type of motion (sharp rotations and backwards travel) is completely undesirable in a real vehicle. The next plot

figure 2.8(b) shows somewhat better results but this behavior is independent of the choice of parameters and primarily depends upon the relative location of the points. The reason of this type of behavior is the same as before; that is the choice of high value of the parameter  $k_3$ . But the problem can not be eliminated by choosing lower values of  $k_3$ , as that would again give rise to the other problems solved by the introduction of  $k_3$ . So the problem seems to be intrinsic to the strategy itself.

#### **Conclusion:**

The strategy could trace a single point quiet successfully and the improvement in the strategy yielded very good results. As the strategy is extended for multiple points, the strategy could not perform quiet satisfactorily. The same set of parameters which yielded very beautiful results in case of a single point yielded quiet cumbersome results for multipoint tracing. It is however observed that the angular variation is reduced greatly by choosing the parameters properly. The turning of the vehicle in the middle of the motion was eliminated and the motion overall became quiet smooth, but some problems could not be eliminated. A number of approaches were tried. One of them involved the introduction of an additional parameter,  $k_4$  with the sine term as the wave number ( $e_2^2 \sin(k_4 t)$ ). These values even though yielded some good results, still did not quiet improve the situation. The final problems that persisted are:

- The vehicle did not always face the goal positions while moving towards them. Or in other words, sometimes the vehicle approached the goal position while facing opposite to it. This is not quiet desired in case of a real vehicle.
- The vehicle took too sharp turns (as big as 3 radians) at a single point as the vehicle approached close to the waypoints. This may be neither practically achievable nor desired in a real vehicle.
- The vehicle became quiet slow while approaching an intermediate point. It approaches the goal point at a very fast rate and becomes too slow on reaching the intermediate waypoints. Such high acceleration or deceleration rates may not be achievable in a real vehicle.
- The vehicle took a lot of time to reach to a very close neighborhood of the waypoints. So the final error values (the regions around the desired position where the vehicle has to stop finally), have to be kept reasonably higher in order to make the motion swift.

All these problems could not be eliminated completely and is the scope for future work. By choosing the parameters properly may diminish some of these problems in magnitude. However the challenge of eliminating the problems by suitable improvement in the control strategy still remains.

Chapter 3 Single-track model

#### Introduction:

This model is similar to a conventional vehicle in which the front wheels can be steered through a range of permitted values of angle in accordance with the longitudinal speed and length of the vehicle. The case is similar to a conventional four wheeled vehicle. So the model can be described as follows.

#### 3.1 The model:

The model is a four wheeler where the front wheels steer and the rear wheels drive. The difference of this model with respect to the previous one lies in the fact that, the previous one could have differential motion at its rear wheels whereas no differential motion could be possible in this model. Only the speed can be controlled here. It is assumed that the steering angel can be arbitrarily varied with its natural limits. The velocity is taken to be proportional to the longitudinal error, the projection of the distance of the destination. So, when the vehicle is not oriented in the direction of the destination, it has a lower velocity and it attains the maximum value when oriented in the direction of destination.

The control objective of this model is again to force the actual Cartesian position and orientation to a constant reference position and Orientation. The model is modeled with respect to the global reference frame. Given a global reference plane in which the instantaneous position and orientation of the model is given by (q(1), q(2), q(3)) with respect to the global reference system. The vehicle is to start at a position (*x*, *y*,  $\theta$ ) and has to reach a given point (*xd*, *yd*,  $\theta$  *d*) with respect to the global reference plane.



Figure 3.1: Relevant variables for the single track model

The instantaneous position of the vehicle is considered to be at the rear end of a line segment of length I. This line segment is the line passing through the center of mass of the vehicle and parallel along the length. The longitudinal axis of the reference frame attached to the vehicle and the lateral axis perpendicular to the longitudinal axis. Since this reference frame's position changes continuously with respect to the global reference system, the instantaneous position of the origin of the reference frame attached to the vehicle is given by  $(q_1, q_2, q_3)$ . The position of the point to be traced in the reference frame attached to the vehicle, with respect to the global Coordinate system is given by  $(e_1, e_2, e_3)$ . Where,

- $e_1$  = The instantaneous longitudinal coordinate of the desired point to be traced with respect to the reference system of the vehicle.
- $e_2$  = The instantaneous lateral coordinate of the desired point to be traced with respect to the reference system of the vehicle.
- $e_3$  = The instantaneous angular coordinate of the desired point to be traced with respect to the reference system of the vehicle.

The conversions of the local values of

$$e_{1} = (x_{d} - q_{1})^{*} \cos q_{3} + (y_{d} - q_{2})^{*} \sin q_{3} \qquad \dots \qquad \dots \qquad \dots \qquad (3.1)$$

$$e_{2} = -(x_{d} - q_{1})^{*} \sin q_{3} + (y_{d} - q_{2})^{*} \cos q_{3} \qquad \dots \qquad \dots \qquad \dots \qquad (3.2)$$

$$e_{3} = \tan^{-1} \left( \frac{y_{d} - q_{2}}{x_{d} - q_{2}} \right) - \theta \qquad \dots \qquad \dots \qquad (3.3)$$

The kinematic model for the so-called kinematic wheel under the nonholonomic constraint of pure rolling and non-slipping is given as follows.

Here,  $\rho = \frac{l}{\tan \delta}$ , which is the instantaneous radius of curvature of the trajectory of the vehicle, and

v = the longitudinal velocity applied to the vehicle

 $\delta\,$  = The instantaneous angular deflection provided to the wheels of the vehicle

Or in other words, the angle by which the reference frame attached to the vehicle changes instantaneously. So these two variables have to be controlled by a control strategy, so that the vehicle reaches the desired point smoothly.

#### 3.2 The Control Strategy:

The Control objective is to design a controller for the kinematic model given by equation (1) that forces the actual Cartesian position and orientation to a constant reference position and orientation. Based on this control objective a simple time varying controller was proposed as follows.

This simply means that, the longitudinal velocity is directly proportional to the longitudinal error in the reference system attached to the vehicle. The rate at which the wheels should be turned is proportional to the angular orientation of the desired position with respect to the reference frame attached to the vehicle. Here *vpar* and *cpar* are positive constant control gains. After substituting eq(7) and eq(8) into eq(4) and eq(5) and eq(6), the following closed loop error system was developed.

$\dot{q}_1 = vpar * e_1 * \cos q_3$	(3.9)
$\dot{q}_2 = vpar * e_1 * \sin q_3$	(3.10)
$\dot{q}_3 = \left(\frac{vpar * e_1}{l}\right) * \tan(cpar * e_3)$	(3.11)

#### 3.3 The results of the simulation:

The above model and strategy was tested in MATLAB for various values of state variables and parameters. The matlab model is in Appendix II (a). Two plots are shown below in which, the vehicle starts from (0, 0, 0) and reaches a point in each quadrant. The parameters: cpar = 1 and vpar = 1. The model is simulated for 10 seconds. The plots show the trajectory of the vehicle in a plane.



Figure 3.2: (a) trajectory while tracing (5, 5); (b) trajectory while tracing (-5, -5)

#### 3.4 The problems encountered:

The model seems to perform quiet nicely from the above results except for the end part of the trajectory, where it seems to be turning about the destination point. But on more rigorous analysis, the model seems to face the following problems.

- The first and the most critical one is, the vehicle fails to start at all, when the destination point lies on the y axis; i.e., when xdes = 0. That is because; the value of ddist is zero when xdes is equal to zero. So, the velocity *v*, which is directly proportional to ddist, is also zero. Hence the vehicle does not start.
- The second problem is; after reaching sufficiently close to the destination point, the vehicle goes round about the point, without actually reaching that. This is not desired and has to be eliminated.

The above three points are discussed below and the attempts taken towards solving those problems. Here the problems are shown diagrammatically.



Figure 3.3: the above figure shows, how the vehicle goes round the destination point without actually reaching that

In the *figure 4.6* the above-mentioned problem of the circling about the destination point is illustrated. The reason for this unwanted circling round the destination point is the following: The vehicle follows such a path that the longitudinal and lateral errors do not decrease proportionately. The longitudinal error decreases faster than the lateral error. So when the longitudinal error approaches zero, the lateral error becomes quiet large in comparison to it and simultaneously the angular error also becomes quiet large. The linear velocity of the vehicle varies proportionately with the longitudinal error, whereas the steering angular velocity is proportional to the linear velocity as well as the tangent of the angular error. Since the value of the tangent becomes quiet large when the angle approaches  $\pi/2$ , the angular velocity of steering also increases indefinitely even though the linear velocity is very small. That means, even though the velocity decreases considerably when the longitudinal error approaches zero, the

angular velocity value increases and makes the vehicle circle about the desired point. So the problem can be solved if the linear velocity and the steering angular velocity can be controlled in such a way that the vehicle goes straight to the destination point. That means, both the longitudinal and lateral error reduce proportionately. The control may be done in the following ways.

- By choosing appropriate values for the parameters vpar and cpar.
- By choosing suitable functions for vpar and cpar and thereby controlling the linear velocity and angular velocity in the desired way.
- By making certain improvements in the control strategy to deal with the problem, if satisfactory results were not obtained by the above two strategies.

The above strategies are applied and tested in the above model. The objective of the above strategies is to smoothen the motion of the vehicle and to remove all the above mentioned problems so that finally, the vehicle should be able to reach all the points in the plane smoothly.

#### 3.5 The best values of the parameters:

The first strategy is employed and tested here. The objective here is to find out the best values of the parameters vpar and cpar so that the problem of unnecessary circling of the vehicle about the destination point is eliminated. So the parameters vpar and cpar are iterated, first keeping cpar constant and varying vpar to test the effect and then varying cpar keeping vpar constant. The objective of the iteration is to understand the behavior of the parameters cpar and vpar and then to find a suitable set of values for cpar and vpar, which would minimize the problem. Even though the relationship does not seem to be a direct one, still a relative understanding of how there variation affect the results, can be made. The result of iteration is discussed below. Since the problem is more pronounced, when the point to be traced is quiet close to the starting point, the destination point for the testing purpose is taken to be (0.5, 0.5). The different results for the different values of parameters are illustrated in a table and compared with the values of parameters cpar =1, vpar=1. Some results are illustrated in graphs for clearer visualization. The hypothesis, which was made as an explanation of the unwanted results are verified with the experimental results. The results are documented in the tables below. In table I, vpar is varied keeping cpar constant.

	vpar	cpar	X <sub>f</sub>	<b>y</b> <sub>f</sub>	$ heta_{f}$	t
1	1	0.2	0.5117	0.0053	0.0236	30
2	1	0.4	0.5252	0.0117	0.0516	20
3	1	0.6	0.5435	0.0206	0.0908	14
4	1	0.8	0.5778	0.0366	0.1662	12
5	1	1.0	0.6515	0.2663	6.9134	—
6	1	1.1	0.8041	0.5710	1.7994	12
7	1	1.2	0.7356	0.5762	1.8829	9.5
8	1	1.3	0.6817	0.5745	1.9600	8
9	1	1.4	0.6390	0.5696	2.0345	7.5

#### TABLE I:

10	1	1.5	0.4693	0.5209	4.1165	8.5
11	1	1.6	0.4565	0.4740	5.2540	15
12	1	1.7	0.5606	0.5479	2.2339	6
13	1	1.8	0.5460	0.5440	2.2882	5
14	1	1.9	0.5359	0.5336	2.3192	4.5
15	1	2.0	0.4990	0.5163	3.2165	4.5
16	1	2.1	0.4340	-0.04210	-0.1209	3.5
17	1	2.5	0.4811	-0.0194	-0.0362	3.5
18	1	3.0	0.7673	0.7823	8.6934	
19	1	5	0.4498	0.5438	3.9945	2

#### The symbols used in TABLE I:

- $x_f$  = the final x-coordinate of the vehicle in the global coordinate system at the end of 100 seconds.
- $y_f$  = the final y-coordinate of the vehicle in the global coordinate system at the end of 100 seconds.
- $\theta_{f}$  = the final angular orientation of the vehicle at the end of 100 seconds in radians.
- t = the time taken to reach the desired point by the vehicle in seconds.

From the above table it is clear that for the values of cpar less than 1 the steering angular velocity is too small. Hence the vehicle simply moves straight with a little amount of turning. So the vehicle covers the longitudinal distance only. By increasing the value of cpar, the vehicle performs better and more of the lateral distance is covered. For value of cpar equal to one, a drastic improvement in the lateral distance covered is observed. Then there is a great improvement in the behavior of the vehicle, when the value of cpar is chosen slightly greater than 1. Not only the lateral distance covered finally improved greatly, but also the angular position greatly reduces which indicates that the turning of the vehicle has decreased. The time taken to reach the desired point also decreases with the increase of value of vpar. That is clearly evident from the fact that, the velocity is proportional to vpar. But the drastic improvement in the time taken corresponds to the fact that the unnecessary rotations have also reduced. Since the vehicle slows down when it points away from the destination point, the less away it points, the faster it reaches the destination. As the value of cpar approaches 2, the behavior of the vehicle improves a lot. But when the value of the vpar exceeds 2, the behavior is again undesirable. For the value of vpar = 2 is the best value of cpar, with respect to vpar = 1.

Some of the above results are summarized in the figures 3.4(a) & (b) and 3.5(a) & (b) for clearer illustration. Figure 3.4 (b) shows the different trajectories followed by the vehicle when the cpar is varied. The values that cpar assumes is 0.2, 0.4, 0.6, 0.8, and 1 respectively. The trajectories are shown collectively to show how by varying cpar the result improves as the vehicle reaches close to the goal position. Figure 3.5(a) displays the best trajectory for any values of vpar = 1 and cpar = 1.2, which is the best trajectory that could be found by the iteration of the values of cpar. Figure 3.5 (b) shows that by

further increasing the value of cpar, the results detoriate yielding poorer trajectories. So the best value of cpar is held 1.2 (see figure 3.5(a)).



Figure 3.4: (a) the trace when cpar = 0.1, and vpar = 1; (b) the traces of the vehicle, when cpar varies from 0.2, 0.4. 0.6, 0.8, and 1 when, vpar = 1. The colors are respectively blue, green, red, black and magenta for the respective values of cpar.



Figure 3.5: (a) the trace when cpar = 1.2, and vpar = 1. The point to be traced (0.5, 0.5), is marked by a circle; (b) the trace when cpar = 3, and vpar = 1

After studying the effect of vpar on the result, the effect of cpar is studied. The parameter cpar is varied keeping vpar constant (vpar = 1). The result of the iteration is illustrated in TABLE II.

#### TABLE II:

	vpar	cpar	X <sub>f</sub>	У <sub>f</sub>	$ heta_{f}$
1	0.2	2	0.6515	0.2663	6.9134
2	0.5	2	0.6515	0.2663	6.9134
3	1	2	0.6515	0.2663	6.9134
4	2	2	0.6515	0.2663	6.9134
5	3	2	0.6515	0.2663	6.9134

So from table II, it is clear that cpar has very little effect on the behavior of the vehicle. It is almost negligible. But the time taken to reach the destination only depends upon it. So any suitable vpar value can be chosen, depending upon other factors. Hence from the above discussion it is clear that the best values of the parameters are vpar= 1 and cpar = 2.

#### 3.6 Improvements in the Control Strategy:

The problem of unwanted circling is solved to some extent by the suitable choice of the parameters. But another vital problem remains, which cannot be dealt with by the adjustment of parameters alone, but the control strategy needs modification itself to deal with the problem. That is the vehicle fails to start from its position when the longitudinal error of the destination point is zero. Or in other words the destination point lies in the line perpendicular to the starting point and passing through it. The reason for this problem lies in the control strategy itself. Since the linear velocity is proportional to the longitudinal error becomes zero the linear velocity also becomes zero. And since the linear velocity is zero, the angular velocity automatically becomes zero, as it is proportional to the linear velocity. Hence the vehicle fails to start in that case. Suitable improvements in the control strategy may solve the problem. The following improvements are suggested to tackle the problem:

- An additional constant term can be added to the linear velocity of the vehicle, so as to make sure that the vehicle starts even when the longitudinal error is zero.
- A function can be added to the linear velocity instead of adding a constant to be more sophisticated.

The above two improvements were employed in the strategy and tested in matlab. The results are discussed below.

#### 3.7 A Constant starting velocity:

A simple modification is made here in the parent strategy to make the vehicle trace the points that lie along the lateral axis in the vehicle frame of reference. That is by adding a small additional constant quantity to the previous equation of the linear velocity. So the linear velocity equation is modified to:

The angular velocity remains the same as in the previous strategy. This ensures that the vehicle would start at all the conditions. But then the problem now arisen due to this strategy is, the vehicle will neither

stop finally, even after reaching the point. It will continue to proceed in that direction with the same velocity  $v_0$ . This problem can be solved by choosing a strategy for the vehicle to finally stop, once it comes within a given closeness of the destination point. The strategy is as the following: the velocity of the vehicle greater than a particular error value would be according to the previous strategy and velocity of the vehicle for error less than that value is zero. The error value is the absolute distance of the instantaneous position value from the destination point. The strategy can be represented as follows:

if (abs(dist) >= near) else  

$$v = vpar^*e_1 + v_0;$$
  $v = 0;$ 

Where, dist = the absolute distance of the destination point from the instantaneous position of the vehicle. near = the required closeness at which the vehicle should stop.

#### 3.8 The result of simulation:

The modified strategy was developed in matlab and tested for various values of state variables and parameters. The matlab program is in Appendix II (b). Some plots are shown below in which, the vehicle starts from (0, 0, 0) and reaches a point in the lateral axis in the vehicle frame of reference in both the directions. That is, it is to reach (0, y) and (0, -y), where y is a variable. The parameters cpar and vpar are taken as, cpar = 1 and vpar = 1. The parameter v0 is iterated to find the most suitable value for it, so that the previous problem of getting stuck at certain points is solved satisfactorily. The strategy is simulated for 10 seconds. Some plots showing some typical results are also shown for clearer visualization of the solution. The following plots show the trajectory of the vehicle in a



Figure 3.6: (a) point to be traced is (0, 0.1): cpar = 1, and vpar = 1; (b) point to be traced is (0, 5): cpar = 1, and vpar = 1.



Figure 3.7: (a) point to be traced is (0, 50): cpar = 1, and vpar = 1; (b) point to be traced is (0.5, 0.5): cpar = 1, and vpar = 1.

#### 3.9 Controls through Waypoints to Destination:

In the previous section the objective of tracing a single point was discussed with a single-track model. The various limitations of the original strategy were discussed and appropriate modifications were made in the control strategy so as to overcome those shortcomings. The next task was to extend the strategy so as to trace a number of points. Given  $[x_i, y_i, \theta_i]$ ,  $i = 0, 1, 2, \dots, m$ , develop an algorithm to make the vehicle to reach  $x_m$ ,  $y_m$ ,  $\theta_m$  from  $x_0$ ,  $y_0$ ,  $\theta_0$  passing through the *m*-1 points in sequence. This is done in the following way: The vehicle starts from the starting point,  $(x_0, y_0)$  and assumes the first waypoint  $(x_1, y_1)$ , to be the destination point. When the vehicle reaches close enough to the first destination point, specified by an error value (erv), it assumes that it has reached the destination point and takes the next waypoint to be the destination point. This way the vehicle travels through all the 'm-1' waypoints, till it reaches the final destination point. The control strategy used to travel between the waypoints is the same as that used to trace a single point in the previous strategy.

#### 3.10 The result of the simulation:

The above strategy, when tested for various points, yielded very good results. Some of those results are shown below. Starting from (0, 0), the vehicle has to pass through the following points: (25, 35), (10, 60), (50, 75). The points are arbitrarily chosen and the parameters are taken to be the best values as found previously. That is, vpar = 2 and cpar = 1. The plots are shown in the Figure 3.8 (a) and (b). The motion looks quiet smooth from the trajectories that the vehicle follows.



Figure 3.8: (a) the path followed by the vehicle, while tracing (25, 35), (10, 60), (50, 75). The values of vpar = 2, cpar = 1; (b) vehicle tracing (0, 25), (40, 60), (30, 25).

#### 4.7 Conclusions:

The strategies still could not solve the problems satisfactorily. The problem of circling round the point, when the destination point lies close to the starting point is not solved satisfactorily and is the scope for future work. However the problem of getting locked, in case the destination point lied in the lateral axis was successfully solved. The strategy exhibited a very beautiful property. In case the destination point lied behind the starting point, instead of turning all the way round like in the previous model it traveled backwards till it faced the goal position and then moved straight to that. This is quiet similar to the way we drive our vehicles. There was not too much of swagger in the motion. However the strategy was unable to orient the vehicle in a final desired orientation. So it is not possible to park this vehicle at a chosen orientation. The strategy can be further modified to achieve this objective.

Chapter 4

# Control Strategy based on three steering positions

#### Introduction:

In this model the problem of reaching a given point can be tackled from another approach. That can be summarized as follows. The space in the vehicle frame of reference is divided into a number of different geometric regions. The behavior of the vehicle can be modeled in a particular way according to the presence of the point in a particular region of space till it reaches in a very close vicinity of the desired point when it finally stops. Thus the behavior of the vehicle in this model is pretty predictable and hence various control strategies can be applied to the model easily. The model can be described as follows.

#### 4.1 The model:





The model as shown in the figure 5.1 above, is very geometric. The space is divided into five discrete regions with respect to the vehicle frame of reference. These regions can be defined as follows.

1. This portion can be defined in the vehicle Cartesian coordinate reference plane as,

$$x_l \ge 0$$
, and  $-\varepsilon \le y_l \le \varepsilon$ 

This is the thin rectangular strip of width  $2\varepsilon$  along the longitudinal axis in the vehicle frame of reference.

2. This portion can be defined in the vehicle Cartesian coordinate reference plane as,

$$y_i > \varepsilon$$
, for  $x_i \ge 0$ 

 $y_i \ge 0$  , for  $x_i \le 0$ 

and,  $x_l^2 + (y_l - r_b)^2 > r_b^2$ 

This is the entire positive y plane, except for portion 1 and 3.

3. This portion can be defined in the vehicle Cartesian coordinate reference plane as,

$$x_l^2 + (y_l - r_b)^2 \le r_b^2$$

This is the portion inside a circle of radius  $r_{b}$  in the positive y plane, as shown in the figure.

4. This portion can be defined in the vehicle Cartesian coordinate reference plane as,

$$y_l < \varepsilon$$
 , for  $x_l \ge 0$ 

$$y_1 \leq 0$$
 , for  $x_1 \leq 0$ 

and,  $x_l^2 + (y_l + r_b)^2 > r_b^2$ 

This is the entire negative y plane, except for portion 1 and 3.

5. This portion can be defined in the vehicle Cartesian coordinate reference plane as,

$$(x_l^2 + (y_l + r_b)^2 \le r_b^2)$$

This is the portion inside a circle of radius  $r_b$  in the positive y plane, as shown in the figure.

The instantaneous position of the vehicle is considered to be at the rear end of a line segment of length I. This line segment is the line passing through the center of mass of the vehicle and parallel along the length. The longitudinal axis of the reference frame attached to the vehicle and the lateral axis perpendicular to the longitudinal axis. Since this reference frame's position changes continuously with respect to the global reference system, the instantaneous position of the origin of the reference frame attached to the vehicle is given by  $(q_1, q_2, q_3)$ . The position of the point to be traced in the reference frame attached to the vehicle, with respect to the global Coordinate system is given by  $(e_1, e_2, e_3)$ . Where,

- $e_1$  = The instantaneous longitudinal coordinate of the desired point to be traced with respect to the reference system of the vehicle.
- $e_2$  = The instantaneous lateral coordinates of the desired point to be traced with respect to the reference system of the vehicle.
- $e_3$  = The instantaneous angular coordinate of the desired point to be traced with respect to the reference system of the vehicle.

The conversions of the local values of

$$e_1 = (x_d - q_1)^* \cos q_3 + (y_d - q_2)^* \sin q_3 \qquad \dots \qquad \dots \qquad \dots \qquad (4.1)$$

$$e_2 = -(x_d - q_1)^* \sin q_3 + (y_d - q_2)^* \cos q_3 \qquad \dots \qquad \dots \qquad \dots \qquad (4.2)$$

$$e_3 = \tan^{-1}\left(\frac{y_d - q_2}{x_d - q_2}\right) - \theta$$
 ... ... (4.3)

The kinematic model for the so-called kinematic wheel under the nonholonomic constraint of pure rolling and non-slipping is given as follows.

$$\dot{q}_1 = v^* \cos q_3$$
 ... ... ... ... ... ... ... ... (4.4)

$$\dot{q}_2 = v * \sin q_3$$
 ... ... ... ... ... ... ... ... (4.5)

Here,  $\rho = \frac{l}{\tan \delta}$ , which is the instantaneous radius of curvature of the trajectory of the vehicle, and

v = the longitudinal velocity applied to the vehicle

# $\delta\,$ = The instantaneous angular deflection provided to the wheels of the vehicle

Or in other words, the angle by which the reference frame attached to the vehicle changes instantaneously. So these two variables have to be controlled by a control strategy, so that the vehicle reaches the desired point smoothly.

#### 4.2 The Control Strategy:

The Control objective is to design a controller for the kinematic model given by equation (1) that forces the actual Cartesian position and orientation to a constant reference position and orientation. Based on this control objective a simple time varying controller was proposed as follows. For controlling the velocity the following strategy is adopted.

if 
$$(e_1^2 + e_2^2 \ge c^2)$$
  
 $v = V_0$   
else  
 $v = 0$ 

end

This means that the velocity is a constant and has a value  $v = V_0$ , for all the points in the space except for the points inside a circle of radius *c*. this circle is the region in which we can choose the vehicle to finally stop. This can be chosen as small as required for the vehicle to stop at a very close vicinity of the desired point.

For the angular velocity control, the following conditional control strategy is adopted. The angular velocity fed to the system is region specific. A step angular displacement value is fed to the system as follows.

1. For region 1 the angular deviation is,

$$\delta = 0;$$

2. For regions 2 and 5 the value of the angular deviation is,

$$\delta = \delta_0;$$

3. For regions 3 and 4 the value of the angular deviation is,

 $\delta = -\delta_0;$ 

This simply means that, the step angular displacement is constant and its sign +ve, -ve or 0 is chosen according to the instantaneous location of the desired point. This simple control strategy is applied in the vehicle and tested.

#### 4.3 The results of the simulation:

The above model and strategy was tested for various values of state variables and parameters. Four plots are shown below in which, the vehicle starts from (0, 0, 0) and reaches a point in each quadrant. The parameters are: v0 = 10, err = 0.01 and, c = 0.1. The model is simulated for 10 seconds. The plot shows the trajectory of the vehicle in a plane. The desired point is marked with a \*.



Figure 4.2: (a) trajectory while tracing (25, 25); (b) trajectory while tracing (-25, 25)

#### 4.4 The problems encountered:

The model seems to perform quiet nicely from the above results except for only one problem. The vehicle finally failed to stop at the destination point, when the error-range (error-range is the minimum nearness to the destination point that the vehicle is finally required to attain) was smaller than a particular value, and that value was found to be dependent upon the strip width ' $\varepsilon$ ' of the region 1. In the strategy

the vehicle was required to go about a circle of radius ' $r_b$ ' till the desired point comes into region 1 in the vehicle frame of reference. Given the strategy lies in any region initially. The vehicle reaches till region 1 quiet nicely. But as soon as it reaches very close to the desired point, instead of stopping at the required error-range, the vehicle keeps on tracing circles indefinitely. The problem is shown in the following plots. In the following plot the strip width and the radius of the error-range, both are taken to be equal to 0.01.



Figure 4.3: (a) The problem occurred when c was chosen to be, 0.01, which is equal to  $\varepsilon$ ; (b) The problem occurred when c was chosen to be, 0.01, which is equal to  $\varepsilon$ , while tracing the point (5, 5). The starting point and the desired point are marked.

#### 4.5 The identification and elimination of the problem:

The problems were investigated for the possible reasons of failure. A number of predictions were made and the solutions were proposed. The first task was to isolate the region that contained the problem.

From a lot of iterative investigation of the above plots, it was observed that the region, which is the intersection of regions 1, 3 and 5, is the region where the problem occurs. To ensure this intuition the abovementioned region was also included inside the error range, where the vehicle has to stop finally. Thus from simple geometrical inspection, the modified error range was found to be,

$$c = \sqrt{2 \times r_b \times \varepsilon}$$
 (4.7)

This improvement in the strategy was implemented and simulated. The result clearly showed the problems removed. So the strategy changes a bit. Now in the modified strategy, the error-range is not a parameter, but is predefined. The corresponding matlab program is in Appendix III (b).

So the results clearly indicate that the problem is centered to the above-mentioned region only. After isolating the region where the problem was centered, the next task was to investigate the real reason, why that problem occurs. Finally, the problem was found to be the following: when the vehicle touches region 1, the angular velocity instantly becomes zero. So the desired point, instead of getting into region 1, lies at the boundary and slowly proceeds longitudinally. In the vehicle frame of reference, the desired point moves along the boundary towards the vehicle slowly. This happens because of the discontinuous nature of the angular velocity shift. So finally when the desired point comes to the point where the border of the circular region intersects the region 1, it behaves according to the conditions defined for being inside region3. Hence it attains an angular velocity of either +  $\delta_0$  or -  $\delta_0$  and instead of stopping there, it moves away from the desired point and stopping there. The problem can be solved, if the point lies within the region 1, instead of lying at the boundary. This can be done by another approach: By making the change of angular velocity continuous, instead of discrete.

#### 4.6 Modifying the model to make the variation in delta continuous:

In the previous strategy, the variation in angular changes was discrete, which is not possible practically. So the model needed a modification so as to make the process continuous. That means, the angular variation will not take place in steps, but it will take place continuously. For this a fourth state variable  $q_4$  is included in the model. This variable is the actual angular change that takes place when a step change in the desired angle change takes place. That means there is a time lag now between the required value of angular change and the actual change. This time lag factor is taken care by a parameter k. So now the angular change that takes place is  $q_4$ , rather than  $\delta$ . The modified model is given below.

$\dot{q}_1 = v * \cos q_3$	 	 	 	 	 (4.8)
$\dot{q}_2 = v * \sin q_3$	 	 	 	 	 (4.9)
$\dot{q}_3 = \frac{v}{\rho} = \left(\frac{v}{l}\right) * \tan q_4$	 	 	 	 	 (4.10)
$\dot{q}_4 = \left(\delta_d - q_4\right)_k$	 	 	 	 	 (4.11)

Here,  $\rho = \frac{l}{\tan \delta}$ , which is the instantaneous radius of curvature of the trajectory of the vehicle, and

v = the longitudinal velocity applied to the vehicle.

 $\delta_d$  = the desired value of instantaneous angular deflection provided to the wheels of the vehicle.

This  $\delta_d$  is a function of the region in which it lies. It is the same step function that was for  $\delta$ . It is defined as follows.

1. For region 1 the angular deviation is,

$$\delta = 0;$$

2. For regions 2 and 5 the value of the angular deviation is,

$$\delta_d = \delta_0;$$

3. For regions 3 and 4 the value of the angular deviation is,

$$\delta_d = -\delta_0;$$

The modified matlab model is in appendix III(c).

#### 4.7 The result of simulation:

The result of the simulations is shown below. The vehicle starts from (0, 0), and reaches the point, (45, 55). For values of err = 0.01 and c = 0.01. The same values showed problems previously. Now the problem seems to have been solved.



Figure 4.4: a result showing the trajectory of the vehicle, finally successful for the desired values of parameters.

#### **Conclusion:**

The model discussed above was so far the most successful in moving to a single point (among the 3 models discussed). However the challenge of extending the model to trace more than one point still remains to be solved. Moreover attaining a final desired orientation is still to be achieved by the use of suitable control strategies.

#### **APPENDIX I**

#### The programs for the first model

#### I(a) The matlab program for the Tricycle type model:

% usage: qd = regulation1( t, q, flg, k1, k2 ); % inputs: % k1 - forward velocity constant; % k2 - angular constant; % p - the coefficient of angle; % q - state variables[x; y; theta]; % t - time; % outputs: % qd - derivatives of states[xd; yd; thetad]; function qd = regulation1(t, q, flg, k1, k2);

 $\begin{aligned} qd(1,1) &= -k1^*q(1) + (-k2^*q(3) + ((q(2)^2)^*sin(t))^*q(2); \\ qd(2,1) &= -(-k2^*q(3) + (q(2)^2)^*sin(t))^*q(1); \\ qd(3,1) &= -k2^*(q(3)) + (q(2)^2)^*sin(t); \end{aligned}$ 

#### I(b) The modified matlab program for the tricycle model:

```
% develop steering controls
% usage: qd = regulation(t, q, flg, k1, k2, k3);
% inputs:
% k1 - forward velocity constant;
% k2 - angular constant;
% k3 - amplitude;
% q - state variables[x; y; theta];
% t - time;
% outputs:
% qd - derivatives of states[xd; yd; thetad];
```

function qd = regulation(t,q,flg,k1,k2,k3);

 $\begin{aligned} qd(1, 1) &= -k1^*q(1) + (-k2^*q(3) + (k3^*(q(2))^2)^*sin(t))^*q(2); \\ qd(2, 1) &= -(-k2^*q(3) + k3^*(q(2)^2)^*sin(t))^*q(1); \\ qd(3, 1) &= -k2^*(q(3)) + k3^*(q(2)^2)^*sin(t); \end{aligned}$ 

#### I(c) The matlab program for the model in global reference frame:

```
% usage: qd = wmr(t, q, flg, k1, k2, k3);
% Inputs:
% k1 - forward velocity constant;
% k2 - angular velocity constant;
% k3 - amplitude ;
% q = the actual state variables [x, y, theta]
% t - time;
% Outputs:
% qd - derivatives of states[xd; yd; thetad];
function qd = wmr(t,q,flg,x,y,a,k1,k2,k3)
```

```
\begin{aligned} qd(1, 1) &= -k1^*((q(1) - x)^* \cos(q(3)) + (q(2) - y)^* \sin(q(3)))^* \cos(q(3)); \\ qd(2, 1) &= -k1^*((q(1) - x)^* \cos(q(3)) + (q(2) - y)^* \sin(q(3)))^* \sin(q(3)); \\ qd(3, 1) &= -k2^*(q(3) - a) + k3^*(((x - q(1))^* \sin(q(3)) + (q(2) - y)^* \cos(q(3)))^2)^*(\sin(t)); \end{aligned}
```

#### I(d)The parrent program for multipoint tracing:

- % usage:
- % dpts the matrix containing all the waypoints
- $\%~\mbox{erv}$  the final closeness at which the vehicle stops at the waypoints
- % inst the initial state
- % ninst the initial state of the state variable
- % tmint time
- % cdpt the waypoints
- % sttm the state variable in the main program
- %~ ndtm the time variable in the main program
- dpts = [15 20 0.675; 35 60 0.927; 80 90 0.54];
- erv = 0.7;
- inst = [0 0 0];
- tmint = 5;

k1 = 2; k2 = 0.1; k3 = 100; tt = []; qt = []; n = size(dpts, 1); ninst = inst; sttm = 0; ndtm = tmint;

```
near = 1;
cdpt = dpts(i, :)
while (near ~ = 0)
  [t, q]= ode23('wmr', [sttm ndtm], ninst, [], cdpt(1), cdpt(2), cdpt(3), k1, k2, k3);
  tt=[tt; t]; qt=[qt; q];
  if (norm(qt(end, :) - cdpt ) < erv)
      near = 0;
  end
  sttm = tt(end); ndtm = sttm + tmint;
  ninst = qt(end,:);
  end
end
```

#### **APPENDIX II**

#### Programs for the second model

#### II(a) The matlab program for the single track model:

```
% To model single track kinematics - provides derivatives. Only steer control;
% Assumes uniform speed;
% Usage: qd = stm1der(t, q, flg, xdes, ydes, thetades, I, vpar, cpar)
% Inputs:
% t - time
% q - sate variables [x; y; theta];
% flg - not used;
% I - length of car;
% vpar - vehicle parameters kv - length and velocity
% cpar - control parameters
% Outputs:
% qd - derivatives of state [xd; yd; theta];
```

function qd = stm1der(t, q, flg, xdes, ydes, thetades, l, cpar, vpar)
v=velocitycontrol1(t, q, xdes, ydes, vpar);
delta=steercontrol1(t, q, xdes, ydes, cpar);
qd(1,1)=v\*cos(q(3));
qd(2,1)=v\*sin(q(3));
qd(3,1)=v/l\*tan(delta);

#### Velocity control:

% To provide velocity command for controlling the vehicle, allows reversing the car;

```
% Usage: delta = velocitycontrol1(t, q, vpar)
```

- % Inputs:
- % t time
- % q state [x; y; theta]
- % vpar gain
- % Outputs:

```
% v - velocity;
```

```
function v = velocitycontrol1(t, q, xdes, ydes, vpar, dpar)
ddist=(xdes-q(1))*cos(q(3))+(ydes-q(2))*sin(q(3));
v=vpar*ddist;
```

#### Steer control:

% To provide steer angle for controlling the vehicle;
% Usage: delta = steercontrol1(t, q, cpar)
% Inputs:
% t - time
% q - state [x; y; theta]
% cpar - k gain
% Outputs:
% delta - steer angle
function delta = steercontrol1(t, q, xdes, ydes, cpar)
alpha=atan2((ydes-q(2)),(xdes-q(1)))-q(3);

delta=cpar\*alpha;

#### II(b) The modified matlab program for the single track model:

#### **Velocity control**

% To provide velocity command for controlling the vehicle, allows reversing the car;

% Usage: delta = velocitycontrol2(t, q, vpar)

```
% t - time

% q - state [x; y; theta]

% vpar - gain

% v0 - velocity constant

% Outputs:

% v - velocity;

function v = velocitycontrol1(t, q, xdes, ydes, vpar, v0)

dist = sqrt( (ydes - q(2))^2 + (xdes - q(1))^2);

ddist = (xdes - q(1))*cos(q(3)) + (ydes - q(2))*sin(q(3));

if (abs(dist) <= 0.005)

v = 0;

else
```

```
v = vpar*ddist + v0 ;
end
```

#### The main program:

% To model single track kinematics - provides derivatives. Only steer control;

% Assumes uniform speed;

% Usage: qd = stm2der(t, q, flg, xdes, ydes, thetades, l, vpar, cpar,v0)

% Inputs:

% t - time

% q - sate variables [x; y; theta];

% flg - not used;
% I - length of car;
% vpar - vehicle parameters
% cpar - control parameters
% Outputs:
% qd - derivatives of state [xd; yd; thetad];

```
function qd = stm2der(t, q, flg, xdes, ydes, thetades, l, cpar, vpar, v0)
v = velocitycontrol2(t, q, xdes, ydes, vpar, v0);
delta = steercontrol2(t, q, xdes, ydes, cpar);
qd(1,1) = v*cos(q(3));
qd(2,1) = v*sin(q(3));
qd(3,1) = v/l*tan(delta);
%endfunction
```

#### The steer control

% To provide steer angle for controlling the vehicle;
% Usage: delta = steercontrol2(t, q, cpar)
% Inputs:
% t - time
% q - state [x; y; theta]
% cpar - angular velocity parameter
% Outputs:
% delta - steer angle

```
function delta = steercontrol2(t, q, xdes, ydes, cpar)
```

```
alpha = atan2( (ydes-q(2)), (xdes-q(1)) ) - q(3);
```

```
delta = cpar*alpha;
```

#### II(c) The matlab program for the model in global reference frame:

dpts = the matrix containing all the waypoints

 $\%~\mbox{erv}$  - the final closeness at which the vehicle stops at the waypoints

% inst - the initial state

- %~ ninst the initial state of the state variable
- % tmint time
- % cdpt the waypoints
- $\%\;$  sttm the state variable in the main program
- %~ ndtm the time variable in the main program
- dpts = [ 25 35 1; 10 60 1; 50 75 1 ];

erv = 0.1;

```
inst = [ 0 0 0 ];
tmint = 5;
I = 5; vpar = 2; cpar = 1;
tt = []; qt = [];
n = size(dpts, 1);
ninst = inst;
sttm = 0; ndtm = tmint;
for i = 1:n
  near = 1;
  cdpt = dpts(i, :)
  while (near \sim = 0)
  [t,q] = ode23( 'stm2der', [sttm ndtm], ninst, [], cdpt(1), cdpt(2), cdpt(3), I, vpar, cpar, dpar );
     tt = [ tt; t ]; qt=[ qt; q ];
     if (norm( qt(end, : ) - cdpt ) < erv)
        near = 0;
     end
           sttm = tt( end ); ndtm = sttm+tmint;
           ninst = qt( end, : );
           if (norm( qt( end, 3) - cdpt ) > erv)
               erv = erv + 1;
           end
      end
end
```

#### **APPENDIX III**

#### Programs for the third model

#### III(a) The matlab program for the single track model:

```
function qd = perfect3(t, q, flg, l, delta, v0, x, y, err, c);
r = I /(tan(delta));
e1 = -(q(1) - x)*cos(q(3)) - (q(2) - y)*sin(q(3));
e2 = -(x - q(1))*sin(q(3)) - (q(2) - y)*cos(q(3));
v = velcontrol(t, q, x, y, v0, c);
if ( (e1 >= 0) & (abs(e2) <= err) )
               qd(1, 1) = v*cos(q(3));
              qd(2, 1) = v*sin(q(3));
              qd(3, 1) = 0;
elseif ( ( ( (((e1 >= 0) \& (e2 > err)) ) (((e1 < 0) \& (e2 >= 0)) ) & (e1^2 + (e2 - r)^2 > r^2)) | ( ( ((e1 >= 0) \& (e2 > err)) ) ) ( ((e1 > err)^2 > r^2)) | ( ((e1 > err)^2 > r^2)) | ( ((e1 > err)^2 > r^2)) | ((e1 < err)^2 > r^2) | ((e1 < err)^2 > r^2)
<-err)) | ( (e1<0) & (e2<0))) & (e1^2 + (e2 + r)^2 < r^2)))
                                   qd(1, 1) = v*cos(q(3));
                                   qd(2, 1) = v*sin(q(3));
                                   qd(3, 1) = (v / I)*tan(delta);
elseif ( ( ( ( ((e1 >= 0) \& (e2 < -err) ) | ( (e1 < 0) \& (e2 < 0 ) ) ) \& (e1^2 + (e2 + r)^2 > r^2) | ( ( ((e1 >= 0) \& (e1 < 0) ) ) ) | ( (e1 < e1 < e1) ) | ( (e1 < e1 < e1) ) | (e1 < e1 < e1) | (e1 < e1 < e1) | (e1 < e1)
(e2 > err) ) | ( (e1 < 0) & (e2 > 0) ) ) & (e1^2 + (e2 - r)^2 < r^2)))
                          qd(1, 1) = v*cos(q(3));
                           qd(2, 1) = v*sin(q(3));
                           qd(3, 1) = (v/l)*tan(delta);
```

end

#### III(b) The modified matlab program for the model having three strategic positions:

```
function qd = perfect2( t, q, flg, l, delta, v0, x, y, err );

r = l /( tan(delta) );

e1 = -(q(1) - x)*cos(q(3)) - (q(2) - y)*sin(q(3));

e2 = -(x - q(1))*sin(q(3)) - (q(2) - y)*cos(q(3));

v = velcontrol2( t, q, x, y, v0 ,delta );
```

```
if ( (e1 >= 0) & (abs(e2) <= err) )
qd(1, 1) = v*cos(q(3));
qd(2, 1) = v*sin(q(3));
qd(3, 1) = 0;</pre>
```

```
elseif ( ( ( ((e1 >= 0) & (e2 > err)) | ((e1 < 0) & (e2 >= 0)) ) & (e1^2 + (e2 - r)^2 > r^2)) | ( ( ((e1 >= 0)&(e2 < -err)) | ( (e1 < 0) & (e2 < 0) ) ) & (e1^2 + (e2 + r)^2 < r^2) ) 

qd(1, 1) = v*cos(q(3));

qd(2, 1) = v*sin(q(3));

qd(3, 1) = (v / l)*tan(delta);

elseif ( ( ( ((e1 >= 0) & (e2 < -err)) | ((e1<0) & (e2 < 0)) ) & (e1^2 + (e2 + r)^2 > r^2)) | ( ( ((e1 >= 0) & (e2 > err)) | ((e1 < 0) & (e2 > 0)) ) & (e1^2 + (e2 - r)^2 < r^2))) 

qd(1, 1) = v*cos(q(3));

qd(2, 1) = v*sin(q(3));

qd(2, 1) = v*sin(q(3));
```

```
qd(3, 1) = (v/l)*tan(delta);
```

end

#### The velocity control:

function v = velcontrol2(t, q, x, y, l, v0, err, delta) r = l /(tan (delta)); e1 = - (q(1) - x)\*cos(q(3)) - (q(2) - y)\*sin(q(3)); e2 = - (x -q (1))\*sin(q(3)) - (q(2) - y)\*cos(q(3)); c = sqrt(2\*r\*err); if(e1^2 + e2^2 >= c^2) v = v0; else

## v = 0;

end

#### III(c) Matlab program for the modified model:

```
function qd = perfect5( t, q, flg, l, delta, v0, x, y, err, c, k );
r = l /(tan(delta));
deltad = deltad( t, q, flg, l, delta, v0, x, y, err, c, k );
v = velcontrol( t, q, x, y, v0, c );
qd(1, 1) = v*cos(q(3));
qd(2, 1) = v*sin(q(3));
qd(3, 1) = (v / l)*tan(q(4));
qd(4, 1) = ( deltad - q(4) ) / (k);
```

#### The function for deltad:

```
function deltad = deltad(t,q,flg,l,delta,v0,x,y,err,c,k);
r=l/(tan(delta));
e1 = -(q(1) - x)*cos(q(3)) - (q(2) - y)*sin(q(3));
e^{2} = -(x - q(1))*sin(q(3)) - (q(2) - y)*cos(q(3));
v = velcontrol(t, q, x, y, v0, c);
if( (e1 >= 0) & ( abs(e2) <= err ) )
        deltad = 0;
elseif( ( ( (e1 >= 0) & ( abs(e2) <= err) ) & ( e1^2 + (e2 + r)^2 < r^2) ) | ( ( ( e1 >= 0 ) &
(abs(e2) \le err) \& (e1^2 + (e2 + r)^2 \le r^2))(((e1 \ge 0) \& (abs(e2) \le err)) \& (e1^2 + (e2 + r)^2)
< r^2 ) )
        deltad=0;
elseif (((((e1 >= 0) & (e2 > err)))(((e1 < 0) & (e2 >= 0)))) & (e1^2 + (e2 - r)^2 >= r^2))(((((e1 < 0))))) = ((e1 < r^2)))
>= 0) & (e_2 < -e_r )) | ((e_1 < 0) & (e_2 < 0)) ) & (e_1^2 + (e_2 + r)^2 < r^2))
        deltad=delta;
elseif (((((e1 >= 0)) & (e2 < -err))|((e1 < 0) & (e2 < 0))) & (e1^2 + (e2 + r)^2 >= r^2))|((((
e1 \ge 0 ) & (e2 \ge err)) | ((e1 < 0) & (e2 \ge 0))) & (e1^{2} + (e2 - r)^{2} \le r^{2})))
        deltad=-delta:
```

end

#### References

- 1. Alexander J C, Maddocks J H 1989 on the kinematics of wheeled mobile robots. *International Journal of Robotics Research* 8(5): 15–27
- Campion G, d'Andrea-Novel B, Bastin G 1991 Modeling and state feedback control of nonholonomic mechanical systems. In: *Proceedings of 30th IEEE Conference on Decision and Control* Brighton, UK, pp 1184–1189
- Canudas de Wit C, Khennouf H, Samson C, Sørdalen O J 1993 Nonlinear control design for mobile robots. In: Zheng Y F (ed) *Recent Trends in Mobile Robots*. World Scientific Publisher, pp 121–156
- De Luca A, Oriolo G, Samson C 1998 Feedback control of a nonholonomic car like robot. In: Laumond J P (ed) *Robot Motion Planning and Control.* Lecture Notes in Control and Information Sciences (229), Springer, London, UK, pp 171–25346 A. DeLuca, G. Oriolo, M. Vendittelli
- 5. De Luca A, Oriolo G, Vendittelli M 2000 Stabilization of the unicycle via dynamic feedback linearization. In: *Preprints of 6th IFAC Symposium on Robot Control* Vienna, A, pp 397–402
- 6. Neimark J I, Fufaev F A1972 *Dynamics of Nonholonomic Systems.* American Mathematical Society, Providence, RI
- 7. Samson C 1993 Time-varying feedback stabilization of car-like wheeled mobile robots. *International Journal of Robotics Research* 12(1): 55–64
- 8. Samson C 1995 Control of chained systems. Application to path following and time-varying pointstabilization of mobile robots. *IEEE Transactions on Automatic Control* 40(1): 64–77