Dynamics based control of a mobile robot with non-holonomic constraints

Prepared by: Victor Lonmo

This thesis is presented to the University of Ottawa
in partial fulfilment of the requirement for the degree of

Master of Applied Science

in

Mechanical Engineering

Department of Mechanical Engineering
University of Ottawa
Ottawa, Ontario, January 1996

0-612-26344-4

Abstract

The development of a planetary rover for use on the moon or on Mars has generated further interest in autonomous vehicles which are capable of moving through a complicated terrain with minimal supervision. One approach for enhancing autonomy is a dynamics based controller which could ensure motion stability and avoid catastrophic events.. This approach allows the monitoring of wheel to ground contact forces which is helpful in avoiding slippage and tipping. As a prelude to more sophisticated tests in a three dimensional environment with random obstacles, a tricycle configuration mobile robot has been built and used to verify the performance of dynamics based control on a flat surface. This controller works by taking into consideration the forces involved in moving the vehicle and therefore can produce better control than controllers which consider only kinematics or simplified dynamics of the system. The work presented here describes the controller of a mobile robot based on dynamics and discusses design aspects of the robot used in this type of system. The dynamics based controller tested in simulations and experiments in this thesis contains several components: exact input-output linearization, model based predictive control of a linearized system for torque saturation verification and operational space linear control. These components were developed separately in previous projects and theses. This method of control is compared in simulations and experiments to a more conventional kinematic based controller. Simulations taking into account actuator torque saturation show a distinct advantage in using the dynamics based control over a kinematic based control in the case of low torque saturation limits. The experimental results show that the robot under dynamic model based control has a more consistent behaviour when compared to the case of the kinematic based controller. Moreover, the kinematic based controller requires for low actuator saturation limits that the controller gains be modified by trial and error for different trajectories while the dynamics based controller does not require any such modification.

Table of Contents

List of Figures

List of Tables

Nomenclature:

A1, A2, A3, A4	notations for functions of steering angle and orientation angle
A_X, A_Y	acceleration of the robot's centre of mass in the XY frame
b	length of the robot's wheelbase
c	distance to the robot's centre of mass from the rear axle
CLAMOR	acronym used for the mobile robot which stands for:"Cartesian Linearized Autonomous Mobile Observable Robot"
Den(δ)	A function of steering angle used in the dynamic equations
dSPACE	A computer produced by the dSPACE company of Germany. The computer used in experiments which uses a TMS320C30 digital signal processor.
I_{Y1}, I_{Y2}, I_{Y3}	moment of inertia of the wheels about their axis of rotation
J_1, J_2, J_3	moment of inertia of the wheels about their vertical axes
J_{SA}	moment of inertia of the steering assembly
l	distance between the robot's rear wheels
M	a transformation matrix from torques to U
m_1, m_2, m_3	masses of the given wheels
m_{FRAME}	mass of the frame
m_{SA}	mass of the steering assembly
m_{TOTAL}	total mass of the robot
N, Q	inertial coordinates defined in the horizontal NQ frame
Nv, Na, Nj	speed, acceleration and jerk of the robot's centre of mass in the N direction
Num(δ)	A function of steering angle used in the dynamic equations
P	a 1x2 vector defined as [N Q]

Qv, Qa, Qj	speed, acceleration and jerk of the robot's centre of mass in the Q direction
r_1 , r_2 , r_3	radii of the wheels
U	a 1x2 vector of command variables
U_1 , W_1 , W_2	command variables
W	a 1x2 vector of command variables
V	a vector of input jerks = (V_N, V_Q)
X, Y	non-inertial coordinates whose origin is the vehicle's centre of mass
Z	vertical coordinate (used with N,Q or X,Y axes)
Δ	Transformation from commands W to jerks $P^{(3)}$
θ	Orientation angle of the robot (from the inertial N axis to the moving X axis)
ζ, λ, ω_N	parameters used in the controlling equation
ω_1 , ω_2 , ω_3	angular velocities of the wheels
ω_δ	relative angular velocity of the steering assembly with respect to the moving X axis
ϖ_θ	absolute angular velocity of the robot's X axis relative to the inertial N axis
τ	a 1x2 vector of torques

Generally derivatives will be shown as:

$$\frac{dx}{dt} = \dot{x} \qquad \frac{d^2x}{dt^2} = \ddot{x}$$

The following convention to denote the ith derivative has also been used:

$$\frac{d^{(i)}(X)}{dt^{(i)}} = X^{(i)}$$

Chapter 1:

Introduction

The field of robotics continues to be a major area of research for mechanical and electrical engineers. Within this field there are many different branches including industrial fixed base robots, vision systems, and manipulator sensor development to name a few. The branch of mobile robots and autonomous vehicles has generally received less attention than other branches which have more immediate industrial applications. Currently, mobile robots are used in applications which are considered extremely hazardous. Some extreme cases include handling bombs, exploring hazardous natural areas [1] and entering areas where accidents involving highly toxic substances have occurred [2]. These vehicles are all very different and highly specialized.

The Planetary Explorer:

Another proposed use for mobile robots is that of planetary exploration. Unlike the Viking landing craft which was capable of running some experiments at a fixed location, a mobile robot would be able to gain specific data about individual locations on Mars. The Luna 17 mission included a mobile robot used in exploring the moon's surface. This robot was controlled by a human operator on Earth. Controlling a robot this way is referred to as "teleoperation". A rover on Mars will require a minimum of three minutes and a maximum of twenty two minutes to send or receive data from Earth (see Appendix I) as opposed to a few seconds for the moon. As explained by Sheridan [3], if supervisory control is to be implemented the following conditions must be met:

- The delay time must be smaller than the time for task execution.

- The unpredictable aspects of the remote environment are not changing too rapidly (low disturbance bandwidth)
- The subordinate automatic system is trustworthy.

These conditions can not be met unless the vehicle is barely moving or it is in an environment with nearly no obstacles. These conditions are not consistent with the type of terrain on Mars which is of most interest to the scientific community.

While an operator on Earth could specify a path to a new desired location, it is the controller that must generate its own path to arrive at the desired location. Although general trajectories can be specified, the vehicle itself will have to generate short portions of paths to avoid or traverse obstacles which only become apparent with closer proximity. This leads to the following requirements:

1) The rover must be capable of accurately observing and accessing its immediate environment.

2) The vehicles controller will have to modify the path of the vehicle as unexpected obstacles become apparent. The modified path must not violate the constraints of the rovers kinematic and dynamic capabilities. In simpler terms, the modified path must be feasible given the vehicle's dimensions, weight and actuators.

3) The vehicle's controller must be able to implement the new trajectory.

If the vehicle is unable to arrive to a desired position then a human operator on Earth would have to make the necessary changes to the path.

The limitation on the dynamic capabilities of the robot forces the system to avoid actuator saturation. Torque saturation is not commonly addressed for terrestrial robots because the actuators are chosen such that torque saturation will not occur. A planetary rover must be made as light as possible and considerable weight could be saved with smaller, less powerful motors, generators and energy storage systems. While conventional kinematic controllers can move a vehicle along a trajectory, they are not able to predict and compensate for slippage leading to potentially catastrophic results. A dynamics based controller which includes the higher order dynamics of the system would be able to deal with these problems. The model chosen for this particular study is a tricycle. While it is unlikely that this configuration would be chosen for a Mars rover, it is intended more as a first step to verifying the benefits of model based control and giving the designers some insight into the correct components needed for a more complex mobile robot.

The Martian Autonomous Rover:

The design of the rover is certainly dependant on the missions expected to be accomplished. It is extremely difficult to make any guess of what the best design would be however some generalizations can be made. An excellent overview of the many different rover designs and control options is provided in [4]. Another detailed analysis of a proposed mission and the rover design is done by Hayard [5]. The rover could use legs instead of wheels. Legged robots can traverse terrain that cannot be traversed by wheeled vehicles. Legged robots have limitations due to their weight and the need for more power compared to similar wheeled vehicles. More general information on legged mobile robots is given by McKerrow in section 2.1 [6] . The most common rover configuration shown in these articles is an articulated vehicle with three sections. Each section has two wheels. The vehicle's steering is controlled by active joints on the middle section. Many different designs have been proposed for a possible Mars rover, all of which are significantly more complex than a tricycle. [7], [8]

Generating the desired trajectory can be very complex particularly if the vehicle is non-holonomic. For known environments a path generating algorithm, such as the one described by Samuel and Keerthi [9] can be used. This simple algorithm uses a kinematic model and an algorithm to minimize some cost functions. Unexpected obstacles may become apparent after the rover has begun following the trajectory requiring a completely new trajectory. Alternatively, an unexpected obstacle may be avoided without the need for a completely new trajectory. In this case, the problem is one of obstacle avoidance instead of trajectory generation. The problem of evaluating the immediate environment in real time with sensors of limited accuracy is addressed by Borenstein and Koren [10, 11]. These papers use ultrasonic sensors whose operation will be impossible on Mars. It is likely that a rover on Mars would use laser rangefinders, and artificial vision systems as opposed to ultrasonics sensors, however the concepts of describing the immediate environment remain unchanged.

To ensure good odometry and to acknowledge any slippage, a dead reckoning system and accelerometers like that proposed by in [12] could be used. Although the model based controller is supposed to predict and eliminate slippage, it may not be able to compensate for sudden changes in contact friction between the wheels and the ground. A system to ensure that the position data available to the rover is as accurate as possible is considered a requirement. A mars rover is likely to move very slowly, and therefore any slippage which occurs will most likely be the result of a sudden change in terrain.

The Tricycle Configuration Mobile Robot:

Prior to sending a rover to Mars it is absolutely imperative that all systems be tested. The model based controller is the part being considered in this work. The development of the tricycle configuration mobile robot precedes that of the more complex 5 DOF articulated vehicle [13]. The tricycle was chosen as a comparatively simple non-holonomic configuration.

The use of dynamic modelling on autonomous vehicles has been carried out previously in many different contexts. In a paper by Samson [14], a simulation of a mobile robot was used to demonstrate time-varying feedback stabilization. The robot is modelled by three equations which describe the kinematics. Similarly Canudas de Wit and Roskam dynamically modelled and controlled experimental robot which was shown to be able to follow a trajectory despite actuator torque saturation [15]. These simulations and experiments feature three wheeled vehicles with two actuated rear wheels having the same axis of revolution, and a caster whose dynamics are ignored. This simplifies the dynamics of the system considerably, however it is very limiting in that most vehicles do not use castors. While there are many different configurations of vehicles proposed for planetary exploration, as yet the author has not seen one which features castors or negligible dynamics.

Simple kinematic controllers have been used for controlling mobile robots for good reason. They are easy to create for simple cases and fairly robust. A good example of improving tracking of kinematically controlled robots is given in [16]. This shows that by considering the posture of the robot and error in position, overall position error could be significantly reduced. A more complicated system may require more than simple kinematic controller because the motors will have to work together to move the robot effectively.

The dynamic model of the tricycle with steering dynamics included has been previously developed [17] using Lagrangian dynamics. Another article [18] shows a similar analysis with no numerical results. Similarly, there have been many studies of the stability of a tricycle configuration vehicle which usually include simulations. For example, Sarkar, Yun and Kumar [19] present a system to control a mobile robot with non-holonomic constraints. Their simulations are promising, but no experimental results are included. These papers present theoretical approaches, however their usefulness is questionable until they are verified experimentally.

The work presented by Kim [20] detailed a simple model for controlling a tricycle configuration mobile robot. The simulation used fast, stable steering. The steering/driving wheel was assumed to be pointed in the correct direction at any given instant, although a value for the steering torque is determined. It does however explain the kinematics and simplified dynamics of the tricycle. It also uses a virtual potential field algorithm which could be used for a vehicle with good sensors and odometry in unknown surroundings. The use of an impedance controller with this type of robot is simulated by Necsulescu, Kim and Kalaycioglu [21]. Continuing along these lines Necsulescu and Eghtesad [22] details another such model which included simplified steering dynamics. The next objective in this research was to add steering dynamics. For terrestrial experiments, the steering dynamics could not be neglected. Preliminary simulation results are given by Necsulescu, Lonmo, Kim and Vukovich [23]. Preliminary experimental results are by Lonmo, Necsulescu and Vukovich [24].

While developing the dynamic model it becomes apparent that torque saturation violates the smoothness required for the exact linearization in the model. It was considered predictive control could be beneficial in dealing with actuator torque saturation. Generalized predictive control, which commonly is used in industrial processes, is explained by Clarke [25] and a more theoretical description is provided by Garcia, Prett and Morari [26]. Del Re, Chapuis and Nevistic provide an excellent general description of predictive control with respect to general linearized systems [27]. Predictive control algorithms usually feature a cost function which is minimised to produce the input to the system. The model based prediction used in this thesis is very simple in comparison. There is no optimality condition to be minimized. Instead, when torque saturation is anticipated (the power requirements to move the vehicle cannot be met by the actuators) the controller recalculates the jerk command which produces a new set of feasible torques. This simplification was chosen to reduce computing time ensuring that the algorithm would be useable in real time with short time steps. The more complex predictive control algorithms are generally used in industrial processes, which are slower to react to new inputs.

Even though this work was stimulated by the issues specific to the development of dynamic model based control it is likely to have significant use in more conventional applications. For example, consider the use of anti-lock braking systems in cars introduced in the mid 1980's. Recently, acceleration slip restraint is being introduced in expensive new cars to reduce slippage on surfaces with low friction. These systems simply enhance control the vehicle by modifying the signal send to the brakes (and the throttle). They are crude in that they must detect the slippage by sensing a change in wheel position which is inconsistent with the kinematic motion of the car. A car controlled by a model based controller could provide the equivalent of anti-lock braking, acceleration slip restraint as well as sideways slippage avoidance and tip-over avoidance. It should be noted that nothing will prevent a vehicle from slipping in certain circumstances, however, this controller would reduce the frequency and severity of slippage. Further research is needed to use an approach tested for a tricycle with actuators on only one wheel and no suspension, however present results demonstrate significant potential.

Chapter 2:

A Brief Description of the Robot

The robot used in this analysis has a tricycle configuration ie. three wheels, two of which are idlers rotating about a common axis. The front wheel is both driven and steered. The robot is designed to move on a flat level surface. Any point on this surface can be given by inertial Cartesian coordinates N and Q. The robot's moving frame (X and Y) has it's origin located at the vehicle's centre of mass. The X axis is set such that it will intersect the steering axis, with the Y axis perpendicular as shown. A Z vertical axis is defined from the robot's centre of mass. The angle between the N axis and the X axis is referred to as the orientation angle θ. The angle between the direction that the driving wheel is pointed to and the X axis is called the steering angle δ (see Figure 2.1)

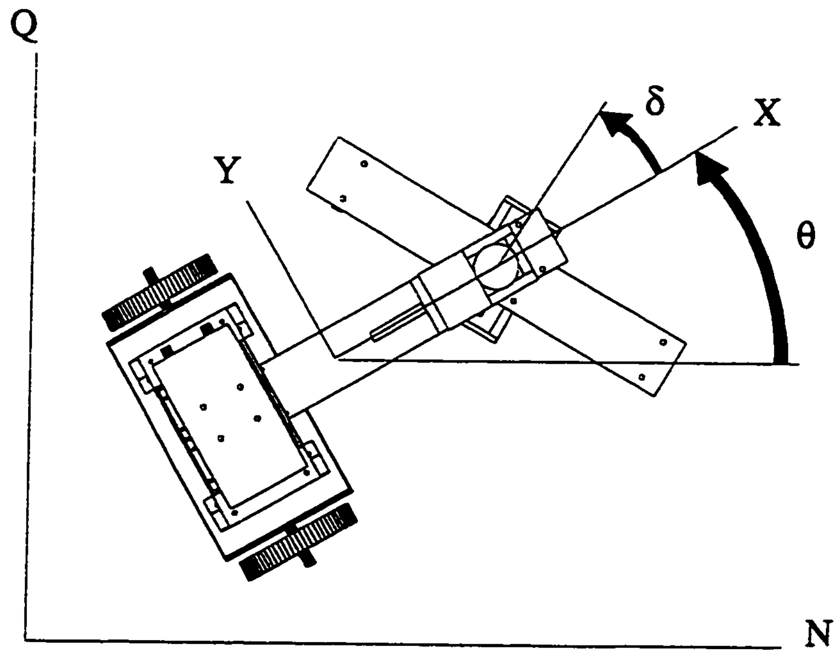

Figure 2.1: The Inertial and Vehicle Coordinate Frames

The dimensions of the robot needed for the kinematic relations are shown in figure 2.2. The wheelbase is denoted as "b" and the distance between the rear axle and the centre of mass is "c".

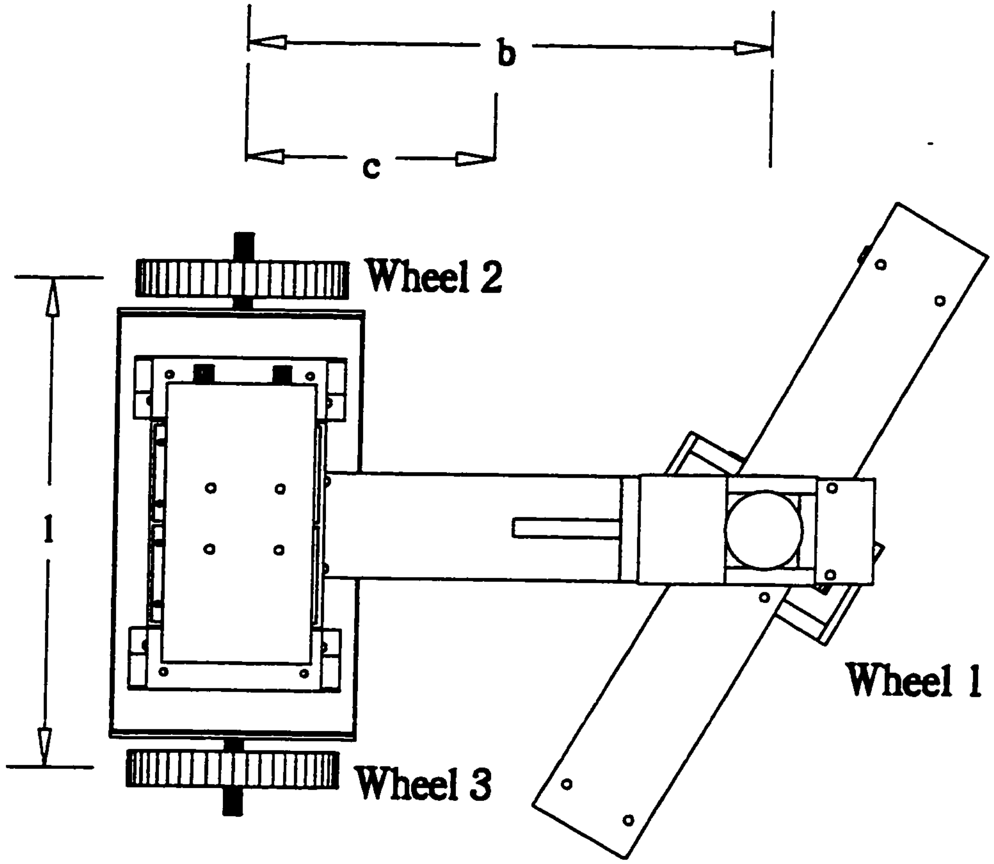

Figure 2.2: Base Dimensions of the Vehicle

It is important to note that the centre of mass of the robot must remain on the centre line of the chassis at a fixed distance "c". Keeping the centre of mass in one fixed position simplifies the dynamics of the system considerably.

The wheels are approximated as perfect disks with no deformation and they are supposed to resist any tendency to slide perpendicular to their current direction. This results in the following kinematic equations which relate the front wheel speed and the velocity of the centre of mass.

$$V_X = rl\ \omega_1 \cos(\delta) \tag{2.1}$$

$$V_Y = rl\ \omega_1 \frac{c}{b} \sin(\delta) \tag{2.2}$$

This gives a non-holonomic constraint written as:

$$\frac{V_X}{V_Y} = \frac{b}{c} \tan(\delta) \tag{2.3}$$

where Vx and Vy are the velocity of the centre of mass in the X and Y directions respectively.

This robot is designed to determine if a simple tricycle configuration system can be controlled with dynamic modelling. In later, more advanced simulations and experiments it is expected that a more sophisticated dynamic model will be able to evaluate and offer good predictions of the contact forces to determine if slip is impending, however at this stage it has been neglected.

The model of the robot includes some assumptions which are included to keep the dynamic model simple. Friction in the steering mechanism has been neglected. The vehicle is assumed to have perfect actuators which produce the desired torques instantaneously. These assumptions have been made for the simulation. In experiments, the friction must be identified and compensated for wherever possible. The dimensions and other properties of the experimental system (which will be referred to as CLAMOR) are given in Appendix H.

Chapter 3:

Dynamic Modelling and the Linearized Model

In this chapter the kinematic equations will be developed. The dynamic model will be produced from the Newtonian equations and their derivatives. Finally the exact linearization of the system and it limitations are discussed.

3.1 Kinematic Model

The basic kinematic equations used to develop the dynamic model of the mobile robot are those developed by Necsulescu, Kim and Kalaycioglu [22], and expanded on by Droguet [28]. Some of the equations have been modified for this thesis. Additional detail is given in Appendix C.

Given the following notations:

$$
\begin{aligned}
A1 &= \cos(\delta)\cos(\theta) - \frac{c}{b}\sin(\delta)\sin(\theta) \\
A2 &= \cos(\delta)\sin(\theta) + \frac{c}{b}\sin(\delta)\cos(\theta) \\
A3 &= \sin(\delta)\cos(\theta) + \frac{c}{b}\cos(\delta)\sin(\theta) \\
A4 &= \sin(\delta)\sin(\theta) - \frac{c}{b}\cos(\delta)\cos(\theta)
\end{aligned}
\tag{3.1}
$$

The robot's absolute velocity can be expressed in the NQ inertial frame by these expressions:

$$\begin{aligned} V_N &= r_1 \, A1 \; \omega_1 \\ V_Q &= r_1 \, A2 \; \omega_1 \end{aligned} \tag{3.2}$$

Differentiating with respect to time produces the corresponding equations of acceleration.

$$\begin{aligned} A_N &= r_1 \left(A1 \; \dot{\omega}_1 \; - (A2 \; \omega_\theta \; + A3 \; \omega_\delta) \; \omega_1 \right) \\ A_Q &= r_1 \left(A2 \; \dot{\omega}_1 \; - (-A1 \; \omega_\theta \; + A4 \; \omega_\theta) \; \omega_1 \right) \end{aligned} \tag{3.3}$$

3.2 The Dynamic model and Exact Linearization

The equations of the dynamic model are given in Appendix C. The expression of ω_1, given by Equation (C.6) in Appendix C substituted into Equation (3.3) gives:

$$\begin{aligned} A_N &= \frac{r_1 \, A1}{Den(\delta)} \left(\tau_d - \frac{r_1}{b} \tau_s \right) - \left[A2 \; \omega_\theta \; + \left(A3 \; + A1 \; \frac{Num(\delta)}{Den(\delta)} \; \omega_\delta \right) \right] r_1 \; \omega_1 \\ A_Q &= \frac{r_1 \, A2}{Den(\delta)} \left(\tau_d - \frac{r_1}{b} \tau_s \right) - \left[A1 \; \omega_\theta \; + \left(A4 \; + A2 \; \frac{Num(\delta)}{Den(\delta)} \; \omega_\delta \right) \right] r_1 \; \omega_1 \end{aligned} \tag{3.4}$$

Given a set of values for the steering angle (δ), the orientation angle (θ), their derivatives and the drive wheel speed (ω_1) there will be one set of accelerations for any set of torques. However, since the coefficients of τ_d and τ_s are linearly dependant it is clear that a set of torques cannot be generated for any given set of accelerations. The accelerations are differentiated again so that two independent control variables will be present in the dynamic equations. (See Appendix D for more detail.) The resulting jerks are given by:

$$J_N = \frac{r_1\ A_1}{Den(\delta)} W_1 - \left[(A_3 + A_1 \frac{Num(\delta)}{Den(\delta)}) \ \frac{r_1 \omega_1}{J_T} \right] W_2 + \Delta_{ON}$$

$$J_Q = \frac{r_1\ A_2}{Den(\delta)} W_1 - \left[(A_4 + A_2 \frac{Num(\delta)}{Den(\delta)}) \ \frac{r_1 \omega_1}{J_T} \right] W_2 + \Delta_{OQ} \tag{3.5}$$

with Δ_O defined as:

$$C1 = \dot{\omega}_\theta$$

$$C2 = \frac{Num(\delta)}{Den(\delta)}\ \omega_1 C1 - \frac{Num'(\delta)\ \omega_1 + Num(\delta)\dot{\omega}_1}{Den(\delta)} \omega_\delta - \frac{Den'(\delta)}{Den(\delta)}\ \dot{\omega}_1 - \omega_1(\omega_\theta + \omega_\delta)^2$$

$$\begin{aligned}\Delta_{ON} = {} & r_1 A1C2 - r_1\omega_1 A2C1 + 2r_1\dot{\omega}_1(-A2\omega_\theta - A3\omega_\delta) \\ & - r_1\omega_1\omega_\theta(A1\omega_\theta - A4\omega_\delta) - r_1\omega_1\omega_\delta(-A4\omega_\theta + A1\omega_\delta) \\ & - r_1\omega_1 A3C1\end{aligned} \tag{3.6}$$

$$\begin{aligned}\Delta_{OQ} = {} & r1A2C2 - r_1\omega_1 A4C1 + 2r_1\dot{\omega}_1(A1\omega_\theta - A4\omega_\delta) \\ & + r_1\omega_1\omega_\theta(-A2\omega_\theta - A3\omega_\delta) - r_1\omega_1\omega_\delta(\ A3\omega_\theta + A2\omega_\delta) \\ & + r_1 A1\omega_1 C1\end{aligned}$$

The definitions of Δ_{ON} and Δ_{OQ} are given here for convenience. The variable Δ will be defined completely later. U_1 , W_1 and W_2 referred to as the commands, result from Equations D.3 and D.6.

$$U_1 = \tau_d - \frac{r1}{b}\sin(\delta)\ \tau_s \tag{3.7}$$

$$W_1 = \dot{U}_1 \tag{3.8}$$

$$W_2 = \tau_s \tag{3.9}$$

The absolute coordinates N and Q have been differentiated three times to produce jerks. These are related to the commands by a decoupling matrix Δ.

$$\boldsymbol{V} = \Delta \boldsymbol{W} + \Delta_O \tag{3.10}$$

where,

$$\boldsymbol{V} = \boldsymbol{P}^{(3)}\ ,\quad \boldsymbol{P} = \begin{bmatrix} N \\ Q \end{bmatrix},\quad \boldsymbol{W} = \begin{bmatrix} W_1 \\ W_2 \end{bmatrix} \tag{3.11}$$

$$\Delta(\delta,\theta,\omega_1) = \begin{bmatrix} r_1 \dfrac{A_1}{Den(\delta)} & -\left(A_3 + A_1 \dfrac{Num(\delta)}{Den(\delta)}\right) \dfrac{r_1\omega_1}{J_T} \\ r_1 \dfrac{A_2}{Den(\delta)} & -\left(A_4 + A_2 \dfrac{Num(\delta)}{Den(\delta)}\right) \dfrac{r_1\omega_1}{J_T} \end{bmatrix} \tag{3.12}$$

$$\Delta_0(\delta,\theta,\omega_1,\omega_\delta,\omega_\theta,\dot{\omega}_1) = \begin{bmatrix} \Delta_{0N}\ (\ \delta,\theta,\omega_1,\omega_\delta,\omega_\theta,\dot{\omega}_1) \\ \Delta_{0Q}\ (\ \delta,\theta,\omega_1,\omega_\delta,\omega_\theta,\dot{\omega}_1) \end{bmatrix}$$

The conditions for invertibility will be discussed later in this section.

With these equations, the two torques and their derivatives can be converted into jerks in the NQ frame. The objective is to generate the desired torques as a function of a desired trajectory, so these equations must be inverted. To do this the following variables are added:

$$\boldsymbol{U} = \begin{bmatrix} U_1 \\ W_2 \end{bmatrix},\quad \boldsymbol{M} = \begin{bmatrix} 1 & \dfrac{rl}{b}\sin(\delta) \\ 0 & 1 \end{bmatrix},\quad \boldsymbol{\tau} = \begin{bmatrix} \tau_d \\ \tau_s \end{bmatrix} \tag{3.13}$$

such that,

$$\boldsymbol{U} = \boldsymbol{M}\boldsymbol{\tau} \tag{3.14}$$

Equation (3.10) can be rearranged to give a value of the commands (**W**) in terms of the input jerks (**V**),

$$W = \Delta^{-1}(V - \Delta_O) \tag{3.15}$$

The value of U_1 is determined by integrating **W**. Once **U** has been determined the torques can be generated.

$$\tau = M^{-1} U \tag{3.16}$$

The control of the robot in the NQ frame can now be separated in terms of N desired and Q desired, even though the system is non-holonomic. The only requirements needed are that the state variables (δ, θ and ω_1), their derivatives and the previous torques must all be known, and that the matrices (Δ and **M**) be invertible. The state variables are assumed known for the purposes of this research. The determinant of **M** is always 1, and therefore, **M** is always invertible. The decoupling matrix Δ, which relates the commands and the jerks, is not always invertible.

$$\det(\delta) = \frac{r_1^{\,2} c\ \omega_1}{b\ J_T\ Den(\delta)} \tag{3.17}$$

The matrix is invertible if its determinant is not zero. This forces the following conditions on the dynamic model.

-the $Den(\delta)$ function and J_T are never infinite.

-the ratio of c/b indicates that the centre of mass must not be located too close to the rear axle.

-the radius r_1 of the drive wheel must not be very small.

-the velocity of the drive wheel (ω_1) must not be zero.

This last result makes sense in that the robot's steering is completely ineffective (with respect to the centre of mass of the robot) when the robot is not moving. This is the only constraint which affects the normal operation of the robot, the others being design requirements all of which have been met (see Appendix H). Therefore, only one degree of freedom is accessible while the vehicle is stopped, thus, this dynamics based control scheme is not useable for starting the motion of the robot. Similarly, once the vehicle is moving it must not stop prematurely therefore it is suggested that a kinematic controller be used for starting and stopping the modelled robot.

It should also be noted that the linearization itself will break down in the event that either torque exceeds its saturation limits. In the event that a very high torque command is sent to a motor, the motor will reach the maximum output torque before the desired torque can be produced. If the feedback linearization system does not recognize the condition then the value of U_1 will not be integrated properly. Methods for avoiding torque saturation are discussed in Chapter 4.1: "The Controllers and the Simulation".

Chapter 4:

The Controllers and the Simulation

4.1: Simulation of the Mobile Robot subject to Model based control

Given that the desired input to the linearized system is the jerk, the control law for position N,Q is given by:

$$\begin{bmatrix} J_N \\ J_Q \end{bmatrix} = (s + \lambda)(s^2 + 2\zeta\omega_N s + \omega_N^2)\begin{bmatrix} N - N_{TRAJ} \\ Q - Q_{TRAJ} \end{bmatrix} \tag{4.1}$$

This control law corresponds to the PDD controller shown in the block diagram (Figure 4.1) where:

$$Kp = \lambda\omega_N^2$$
$$Kd = 2\zeta\omega_N\lambda + \omega_N^2$$
$$Ka = \lambda + 2\zeta\omega_N$$

Given the nonlinearities of the mobile robot ignored in the dynamic model and the fact that in this case only partial linearization can be achieved, the design of the controller cannot be realized with well established linear control theory methods. The control requirements were taken into consideration by an iterative process of choosing the gains in Equation 4.1. The zero at $s = -\lambda$ had to be positioned in such a way that it can not be ignored. In this situation, there is much more emphasis on maintaining speed as opposed to position. In simulations this results in a

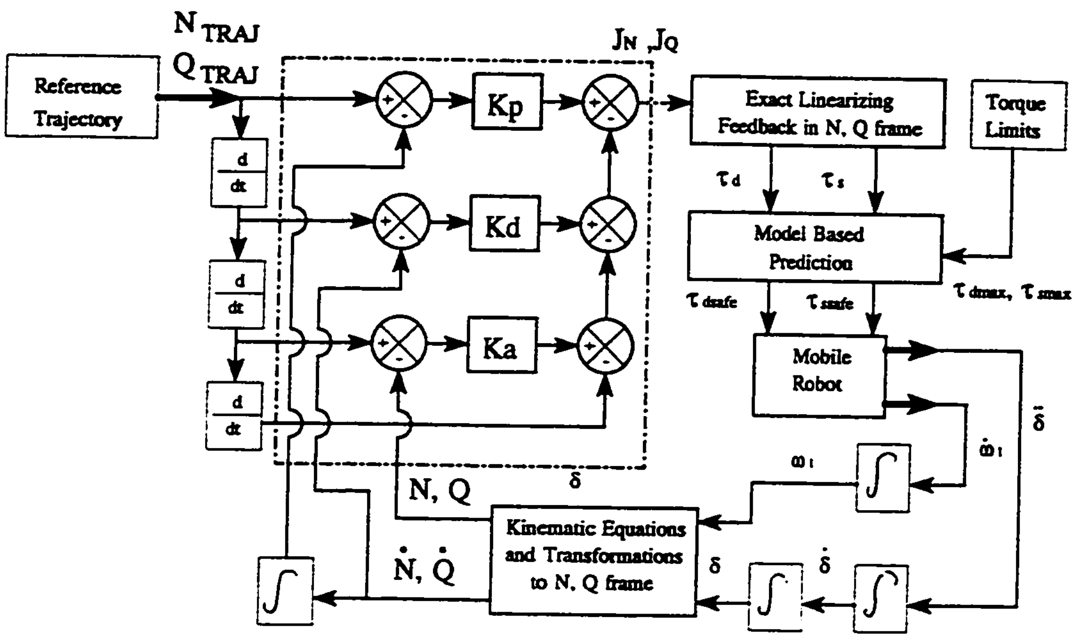

Figure 4.1: Block Diagram of the System using the Model Based Controller

The diagram does not explicitly show the filters which are included in the "Kinematic Equations and Transformations etc." box. The model based prediction box is shown in more detail in Figure 4.2.

tendency to keep the robot moving which causes it to approach the desired trajectory very slowly. One benefit of this is that the drive wheel is not likely to stop for a trajectory with continuous speed. Low values of ζ should be avoided in situations which involve saturation. If the saturation limit on the driving torque is very low compared to the unsaturated torque values then very high values of ζ should be considered.

The values recommended in Table 5.3 represent a reasonable compromise which allow the simulated robot to follow the trajectory with acceptable overshoot and a minimal tendency to stop prematurely. Simulations have been run with longer predictive periods, however the desired jerk begins to oscillate, usually causing instability. It had been expected that the value for λ could be made very high, corresponding to moving the associated zero to the left on the s-plane, and causing the system to behave similarly to a second order system. Unfortunately, no simulations were successful with high values of λ. When the values of λ or ω_N are set too high, there is a tendency to produce the oscillations in input jerk. To reduce this effect, the absolute value of the input jerks are limited to $10e^6$ m/s^3. Even so, high values of λ, ζ and ω_N can cause floating point overflows in the simulation. Similarly, the cut-off frequency to the filters should be 10 Hz or higher to avoid floating point overflow, presumably as a result of oscillations. In general, floating point overflows are reduced by shorter time steps, but this is not always practical. If lower cut-off frequencies are desired in simulation, it is suggested that the gains be reduced significantly.

The block diagram of the linear third order controller, exact linearizing controller with model based prediction is given in Figure 4.1. The model based prediction block is added in

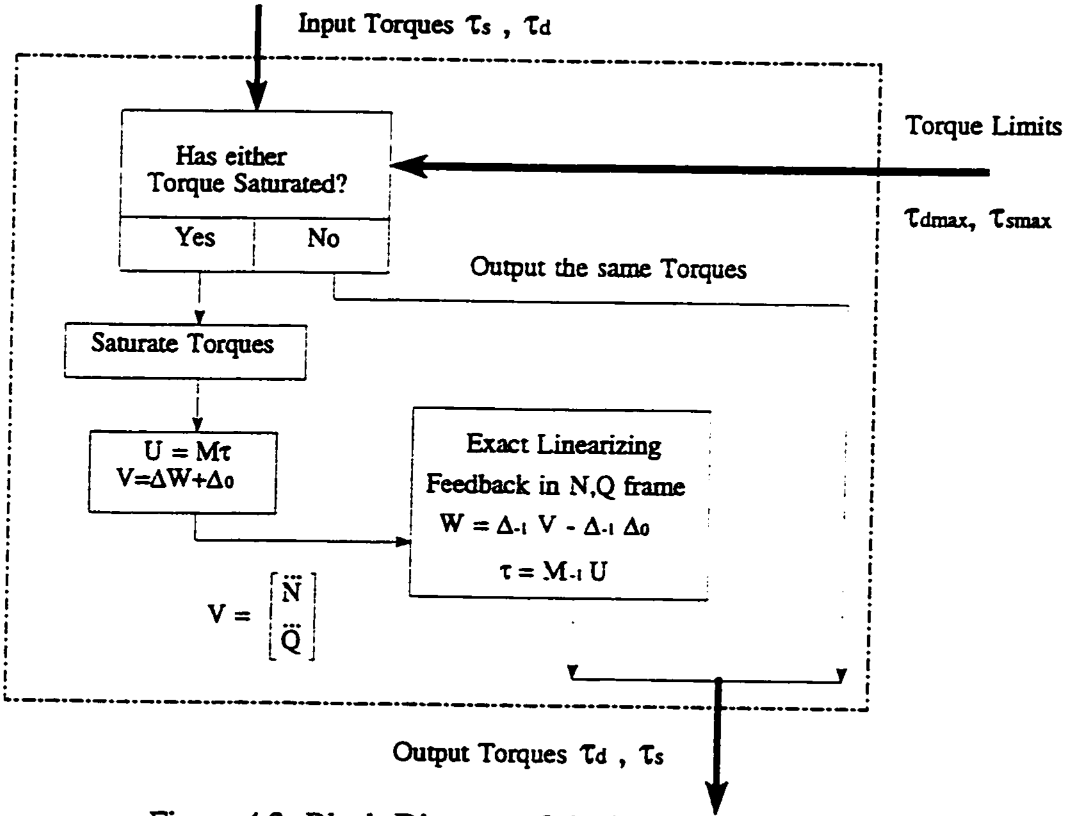

Figure 4.2: Block Diagram of the Model Based Prediction

order to preserve the smoothness condition (ie. not reaching the torque saturation limits) which is required for the exact linearizing controller. While the model based exact linearization approach has been used, given the inaccuracies of the model the actual result is a partial linearization of the real mobile robot and, consequently, experimental testing had to be pursued to validate the overall control system.

4.2: Time Scaling and Model Based Prediction

Any linearization requires that all of the necessary variables be estimated accurately. In simulation this is easily achieved, but in actual experimentation it is not always the case. In the model based system presented in the previous chapter, the linearization will not be achieved if either torque saturates. Thus, the controller must insure that any command it produces is within the associated torque limits. Although the torque is considered to be the variable that saturates, it should be noticed that the torque of a real motor is a function of its: direction, speed and input as well as other environmental factors such as temperature. For the purposes of the simulation and the experiment, the robot is considered to only saturate in output shaft torque. The torque limits given to the controller are all well within the torques that the actual motors on the CLAMOR can produce at the anticipated operating speed. However, the values of ω_t , ω_δ , driving torque and steering torque should be examined to ensure that the trajectory and robot parameters do not result in speeds and torques that the motors cannot produce.

The use of time scaling, zero and first order model based prediction are explained clearly by Droguet [28]. Time scaling is a simple method of avoiding actuator saturation by taking a desired command, determining the actuator values, ensuring that they are not saturated and determining a new set of jerks from them. If saturation has occurred then the actuator values produced are saturated separately and then the command jerks are calculated based on these new safe torques. This method is very effective and fairly easy to implement.

It should be noted that the torque being used in this method is being applied at the precise instant that it is needed. When using the actual robot, the torques would be applied slightly after they are needed. Since the computation time is very small this not considered a problem. (A thirty second simulation with a one millisecond timestep required thirty five seconds on a DEC 486 DX2/66.) The TMS320C30 digital signal processor can run the robot in real time easily.

A controller with model based prediction in a zero order case is very much like time scaling case except that the values of the commands to produce saturation torques are assumed constant over a longer period of time. The state of the robot is assumed to not change significantly during the period of time in which the model based controller is trying to anticipate torque saturation. To ensure that no saturation occurs the saturation torques are multiplied by a safety factor. As the ratio of the "control_step" (the length of time that the controller's commands must be used) divided by the "torque_step" (the time between the recalculation of the torques) increases the safety factor must be reduced. This method may still allow saturation if the path is highly non-linear or if the path requires significantly higher torques than those available. If saturation should occur, it tends to be limited and therefore less damaging to the system. Even with a safety factor of 100% the system remains controllable because the maximum torques are not much larger than the saturation limits.

Model based prediction can also be used in a first order case. Here the values of the commands U (see Equation 3.13), the data (θ, δ, ω_t and their derivatives) and the position (N, Q and their derivatives) are all predicted at a future time (t + control_step/2). These commands will be used from time = (t) to time = (t + control_step). The state variables can be extrapolated using their derivatives. The current values of acceleration are held constant. Thus, the commands can be determined and saturation avoided. This represents the most sophisticated of the three algorithms. The extra time associated with predicting the various data is minimal. This is the only algorithm which is truly predictive, allowing computation of the torques prior to the use of those torques.

4.3: Kinematic Based Controller Implementation

Kinematic based controllers are the standard means of controlling robots and autonomous

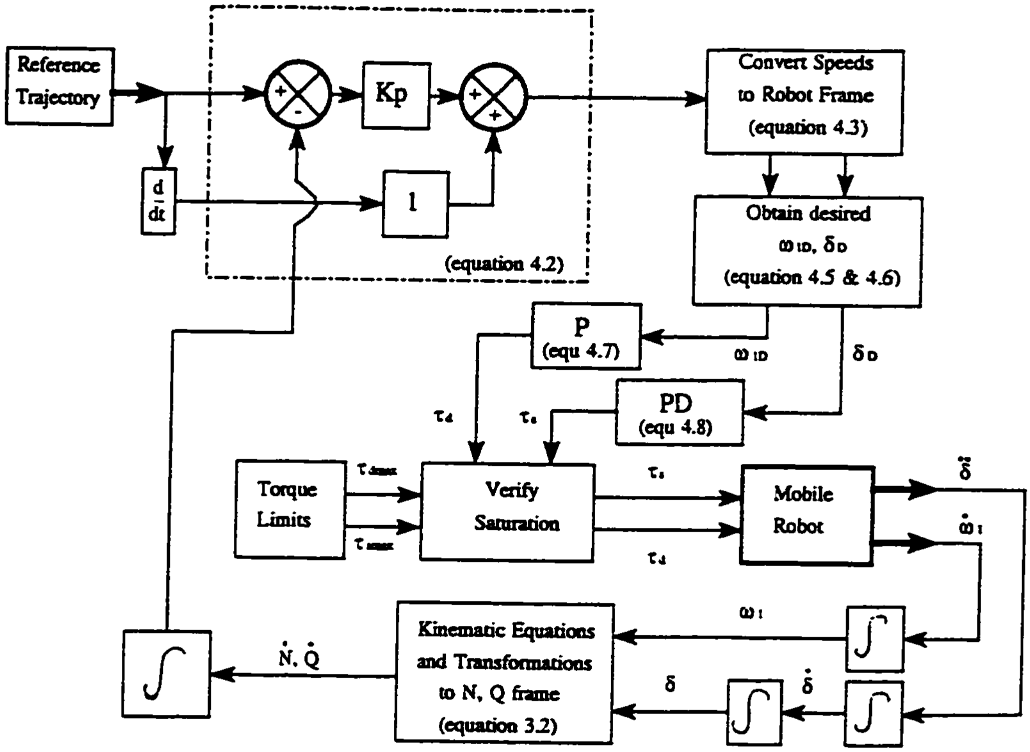

Figure 4.3: Block Diagram of the System using the Kinematic Based Controller

vehicles. This simple controller has been added for comparison with the model based controller.

Controlling the vehicle while neglecting much of the dynamics of the system results in a controller which is very easy to produce. The steps used to control the robot in simulation and in actual testing are shown below. First, a P controller gives resolved speeds.

$$\begin{aligned} \dot{N}^{(C)} &= \dot{N}_{Traj} + (N_{Traj} - N)\ Kp \\ \dot{Q}^{(C)} &= \dot{Q}_{Traj} + (Q_{Traj} - Q)\ Kp \end{aligned} \tag{4.2}$$

These speeds in NQ are transformed into the speeds in the robot frame.

$$\begin{aligned} \dot{X}^{(C)} &= \dot{N}^{(C)}\cos(\theta) + \dot{Q}^{(C)}\sin(\theta) \\ \dot{Y}^{(C)} &= -\dot{N}^{(C)}\sin(\theta) + \dot{Q}^{(C)}\cos(\theta) \end{aligned} \tag{4.3}$$

The superscript $^{(C)}$ will no longer be used in this chapter with regard to these velocities. The desired velocities in X and Y are used to generate a new desired position for the steering angle (δ_D). This is equation 2.3, the non-holonomic constraint, for steering angle.

$$\delta_D = Tan^{-1}\left(\frac{b\,\dot{Y}}{c\,\dot{X}}\right) \tag{4.4}$$

The steering torque (τ_S) is then determined using a PD controller for the steering angle (δ). The value of the derivative of δ_D is determined by numerically differentiating the δ_D .

$$\tau_s = (\dot{\delta}_D - \dot{\delta})K_{DS} + (\delta_D - \delta)K_{PS} \tag{4.5}$$

The desired value for the speed of the driving wheel (ω_l) is computed. This equation is based on equations 2.1 and 2.2, the kinematic constraints.

$$\omega_{1D} = \begin{bmatrix} \dfrac{\dot{X}}{rl\,\cos(\delta)}\ , \ for \quad |\cos(\delta)| > 0.1 \\ \dfrac{b\,\dot{Y}}{c\,rl\,\sin(\delta)}\ , \ for \quad |\cos(\delta)| \le 0.1 \end{bmatrix} \tag{4.6}$$

The driving torque (τ_d) is obtained by:

$$\tau_d = (\omega_{1D} - \omega_1)\ Kt \tag{4.7}$$

The two torques, τ_S and τ_d are independent, they affect each other only to the extent that the robot's position is changed and then new commands are generated. It was found by trial and error that a certain range of gains were needed to control the vehicle. Alternative forms of this controller were considered using PD and PI control on drive wheel speed, however the improvements were minimal and in some cases lead to instability. It should be noted that the robot will become unstable if it is unable to stay near it's position on the desired path for most trajectories. In other words, although a PI controller will not cause instability in the robot's speed, it may cause changes in the steering commands, resulting in unstable steering. While it would be easier to chose the gains empirically, no suitable method for transforming the gains was found. It should also be apparent that since the model based dynamics controller is conceptually different from this conventional controller, it would be impossible to generate gains which cause this controller to behave similarly to the model based controller. The gains used in this study are given Table 5.1:

In simulation the new torques were determined every 2ms. This is consistent with the model based controller. However, the model based controller recalculates the desired jerks every 4ms, so the kinematic controller has been given a significant advantage. A block diagram of the kinematic based controller is shown in Figure 4.3.

The kinematic based controller is much simpler to develop than the model based controller. Unlike the model based controller, very little knowledge of the system being represented is needed. This is an advantage in producing the controller, but it does not take into account the physical system and therefore may not produce correct torques needed to control the robot efficiently.

4.4 Features of the Simulation

The simulation has been designed to represent the actual CLAMOR platform as accurately as possible. It is written in Turbo C++ version 1.01, but it uses only ANSI C with the exception of the graphics output routines. The simulation allows the user to choose two sets of parameters for the robot which are called "actual" and "virtual" prior to compilation of the code. The "actual" parameters represent the values that the simulation is to use when determining the acceleration of the robot as a result of the actuator torques. The "virtual" parameters are the user's guess of the actual values. This was done simply to determine the effects of an incorrect value associated with any of the parameters. In experiments, there is only one set of parameters, which correspond to the virtual case. The feedback signals are retrieved, filtered and differentiated just as they are in the actual system.

The virtual (and actual) parameters are: the masses and moments of inertia of the wheels, the rotational inertia of the steering assembly, the size of the robot, and the location of the centre of mass (see Figure 2.2). From testing it was found that the position of the centre of mass is very important to the control of the robot. This value should be as accurate as possible. The other parameters mentioned are also important, however it was found that overestimating them by small amounts does not significantly affect the results of the simulation. In fact, it often improved performance. Underestimating these parameters was considerably more detrimental. The simulation rarely finished when the mass of the frame was 20% less than the "actual" value, unless another mass was significantly over estimated. Thus, it is recommended that for simulation and testing of the robot that the masses and moments of inertia be significantly higher than anticipated, except for those parameters which are known.

The resolution of the encoders on the motors is not infinite and therefore filters are used on the CLAMOR system as well as the simulation. The routines for differentiating and filtering the steering angle and the position of the drive wheel are the same ones used in the actual system. The filters are a low pass IIR second order type. Their cut-off frequency can be set to any one of ten values in the options menu (see Appendix A: Using the Simulation). The actual motors used have a resolution of 75100 steps per revolution. Since the simulation time step is 0.001 seconds and a normal operating value of ω_1 is roughly two radians per second, the encoder is expected to receive roughly 24 encoder steps per time step. With this in mind, it is recommended that the simulation not be run with a time step less than 0.0005 seconds to ensure that the driving motor filters operate normally. The steering motor will generally have much lower speeds than the driving motor, so problems with the determining the derivatives of the steering angle may occur. The filters used for each motor have the same cut-off frequency.

The simulation features four different path generating algorithms. The first is a Gaussian curve. This curve is ideal for testing because the derivatives all approach zero away from the point of maximum amplitude. It also allows the system time to return to the desired path after saturation occurs.

$$
\begin{aligned}
&Q = A\, e^{-\left(\frac{t\,-\,10}{sm}\right)^2} \\
&N = Nv\, t,\ Nv \textit{ is constant for all } t \\
&A\ :\textit{ is theamplitude of the trajectory (meters)} \\
&t\ :\textit{ is time (seconds)} \\
&sm :\textit{ is a smoothness parameter}
\end{aligned}
\tag{4.8}
$$

The second curve is a sinusoid which has been added because it represents a trajectory which does not stabilize. Also, the sinusoid can be used to generate other curves. Generating trajectories as summations of sinusoids may fail to control the robot due to the high frequencies needed to generate the desired path. (This will not be examined in this thesis, but may produce very interesting results.) Since the robot starts by moving in a straight line there has to be a trajectory which allows the transition to the sinusoid. The curve used to do this is fifth order polynomial which allow the position, velocity and acceleration to be matched at both ends.

$$
\begin{aligned}
&Q = Qc_3*(t-2.0)^3 + Qc_4*(t-2.0)^4 + Qc_5*(t-2.0)^5\ ,\ \textit{for } 2.0 < t < 5.0 \\
&Q = A/2.0\left[1.0 + \sin(\, per\,(t - 5.0)\right],\ \textit{for } t > 5.0 \\
&N = Nv\, t,\ Nv \textit{ is constant for all } t \\
&A\ :\textit{is the amplidude of the trajectory (meters)}
\end{aligned}
\tag{4.9}
$$

where,

$$\begin{aligned}
Qc_3 &= 0.370370(K1) \; -0.444444(K1\ per) \\
Qc_4 &= -0.185185(K1) \; +0.259259(K1\ per) \\
Qc_5 &= 0.024691(K1) \; -0.037037(K1\ per)
\end{aligned} \tag{4.10}$$

$$K1 = A / 2.0$$

The amplitude variable (A) is divided by two so that it will be consistent with other trajectories. Otherwise, the space needed in experimental testing for the different trajectories would vary significantly, requiring additional compiling in the experiments. The "per" variable is the period of the sinusoid divided by 2π.

The next trajectory is a square wave. This highly non-linear trajectory gives an interesting comparison with the smooth curves.

$$\begin{aligned}
&Q = 0 \ , \ t < 5.0 \\
&Q = A \ , \ 5.0 < t < 15.0 \\
&Q = 0 \ , \ t > 15.0 \\
&N = Nv\ t,\ Nv \textit{ is constant for all } t \\
&A \ : \textit{ the amplitude of the trajectory (meters)}
\end{aligned} \tag{4.11}$$

The last "trajectory" uses virtual potential fields to generate a trajectory which determines the desired accelerations to reach a target while avoiding obstacles in simulated time. This is an example of impedance control not unlike that shown in [20]. In this case, the target and the obstacles are fixed. Their position is set prior to compilation of the code. The algorithm that was used is simple, more elaborate algorithms can take into account the shape of the vehicle and other concerns. This trajectory was included to demonstrate the ease of converting a common algorithm, which normally is used in some conventional kinematic control into a similar algorithm with dynamics based control with model based prediction. Results using this trajectory and the Gaussian trajectory have not been included in this thesis to reduce redundancy.

Chapter 5:

Simulation Results

5.1: The Kinematic Controller

It was expected that the PD controller would have difficulty in dealing with saturation of the torques. While this result is apparent, the effect was not as destabilizing to the system as expected. Extended periods of driving torque saturation resulted in the robot being pointed in the correct direction, but ahead of or behind the desired position. Unlike the dynamics based controller, there is no need to ensure that the vehicle does not stop. This gives the user more freedom to chose the gains used in controlling the robot.

With very stiff steering, the robot tracks the sinusoid trajectory very well (see Figure 5.2). However, when saturation is used and the vehicle departs from the desired trajectory, there is a very strong tendency to produce motion which is not consistent with the desired path. Since the vehicle is no longer on the path, the feedback errors are large and thus the speed tends to be high. If the vehicle should return to the path it will likely overshoot it. Regardless, the kinematic controller achieved the best tracking of the sinusoid path with no saturation and stiff steering. It also behaved well when small amounts of saturation occurred. There is very little transition between what looked like very good tracking and a sudden inability to follow the desired trajectory. Also, with the stiff steering, it was unable to react to the square wave input when the saturation limit was reached (see Appendix E, Figures E3 and E4).

It was found that in order to follow the square wave input the steering stiffness had to be reduced significantly. This produced acceptable results with oscillations. The simulation was then run for the sinusoid trajectory and the robot was shown to be able to follow the trajectory, but with significant error. Through trial and error, it was found that the simulated robot could be made to perform most trajectories as well or better than the dynamics based controlled system, however each different trajectory required retuning the proportional steering gain. Using only one set of gains required that the steering stiffness be low to ensure that the simulated robot would stay near the desired path.

Gain	Value	Equation Number
Kp	0.429	4.2
K_{DS}	4.0	4.5
K_{PS} (stiff steering)	75.0	4.5
K_{PS} (loose steering)	25.0	4.5
Kt	1.0	4,7

Table 5.1: Gains used by the Kinematic Controller in Simulation

These saturation limits were also chosen for the dynamics based controller. The torque limits and other parameters of the simulation used are given in Table 5.2

Simulation Parameter	Value
Saturation limit for Driving Torque	0.55 Nm
Saturation limit for Steering Torque	1.40 Nm
Torque Time Step	2 ms
Integration Time Step	1 ms

Table 5.2: Simulation Parameters used with the Kinematic Controller

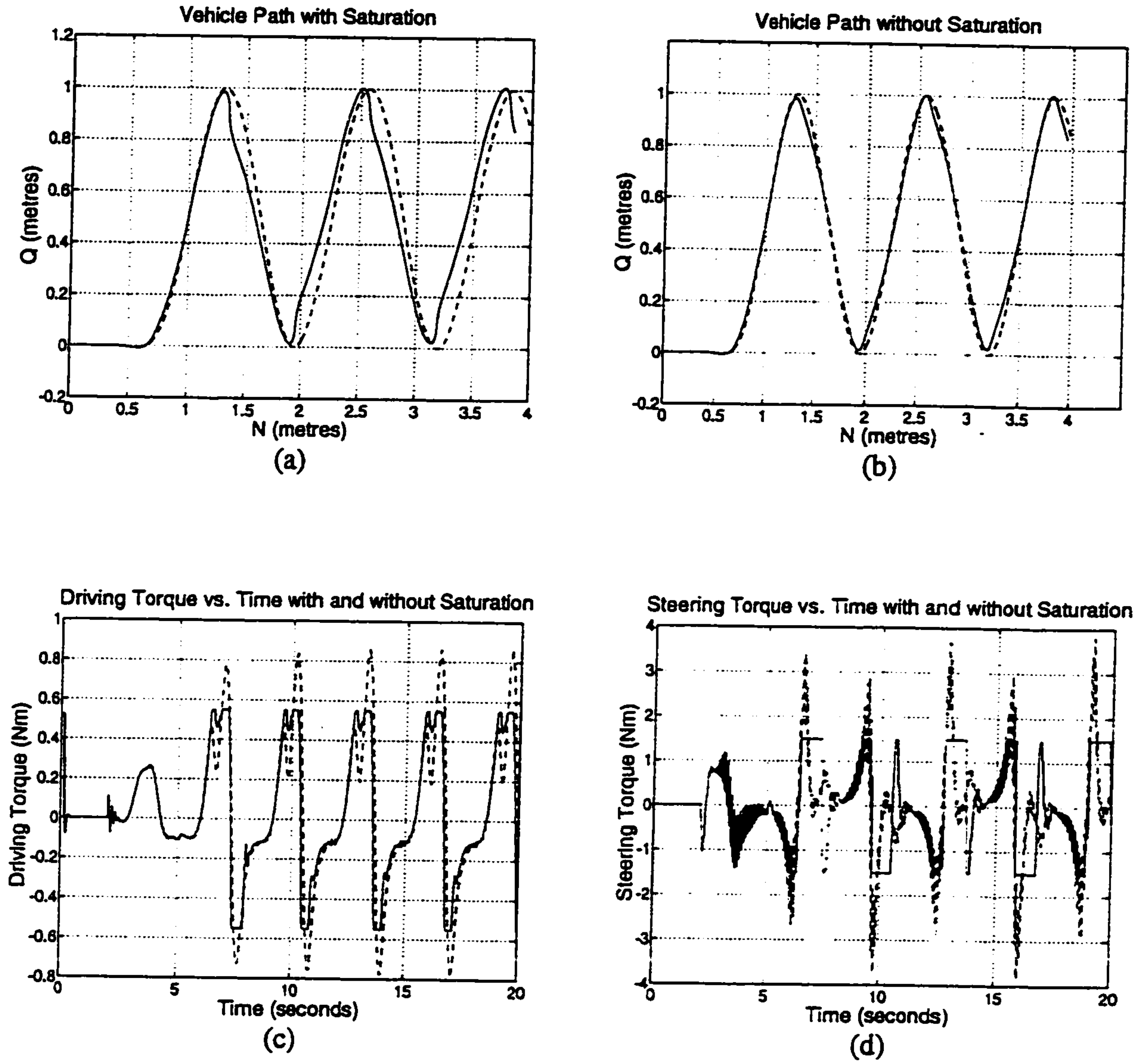

Figure 5.1: Simulations using a kinematic based controller with loose steering following a sinusoid trajectory

a) Path achieved with torque saturation (the dotted line is the desired path)

b) Path achieved without torque saturation (the dotted line is the desired path)

c) Driving Torque versus Time for both cases (dotted lines ignore torque saturation)

d) Steering Torque versus Time for both cases (dotted lines ignore torque saturation)

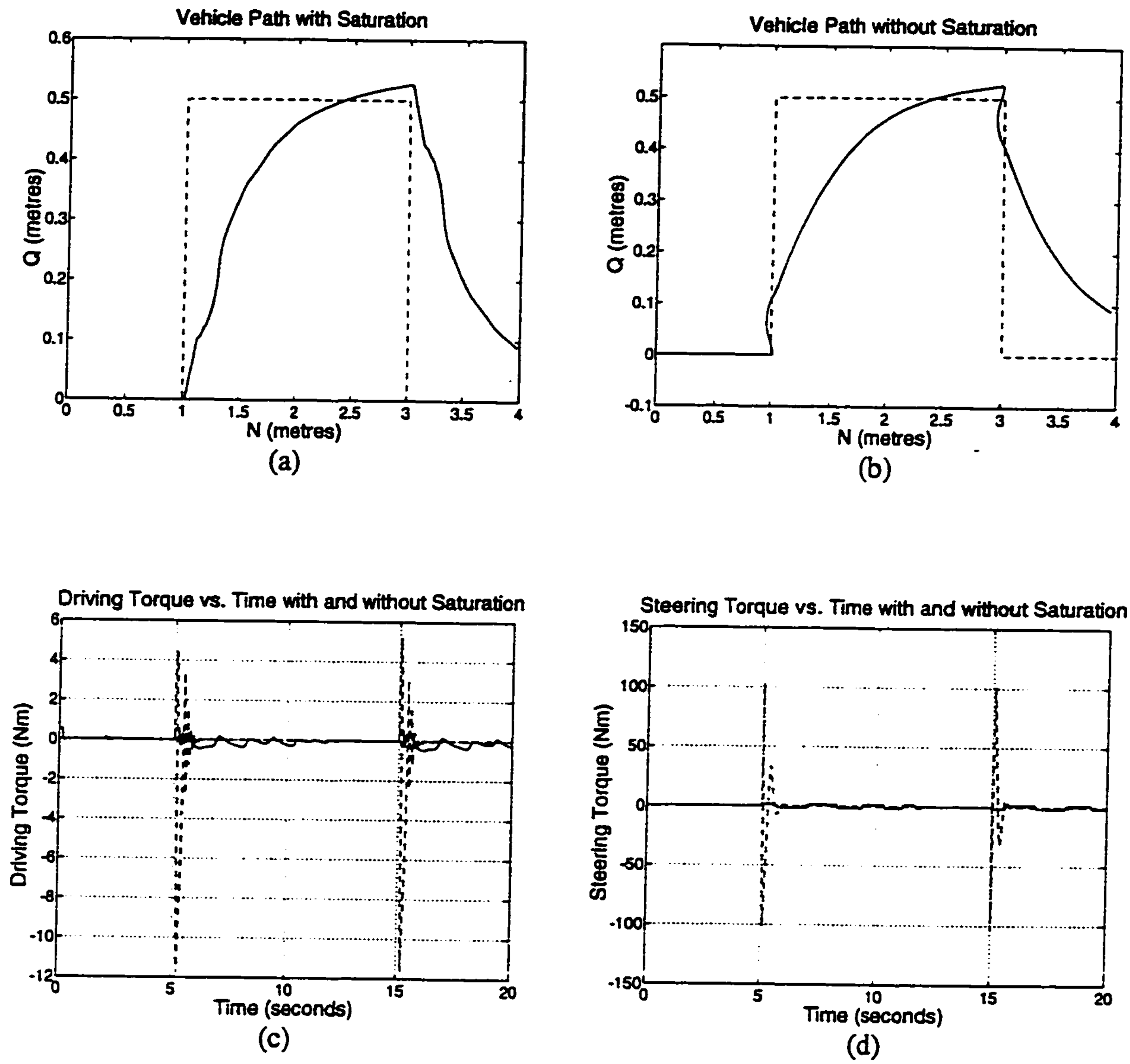

Figure 5.2: Simulation results for a kinematic controller using loose steering to follow a Square Wave Trajectory

a) Path achieved with torque saturation (the dotted line is the desired path)

b) Path achieved without torque saturation (the dotted line is the desired path)

c) Driving Torque versus Time for both cases (dotted lines ignore torque saturation)

d) Steering Torque versus Time for both cases (dotted lines ignore torque saturation)

5.2: Simulation Results using Model based control

Due to the precise nature of the feedback linearization the robot is expected to follow the desired trajectory very accurately. While the output graph shows the position matching the desired position almost exactly, the robot will often precede, or follow the desired trajectory as it returns to the original path. This is the result of the driving torque being saturated while the robot is moving in a fairly straight line. This can cause serious difficulty if the robot deviates significantly from the desired trajectory. However, saturation of the steering torque is more likely to cause problems as the robot attempts to follow the trajectory. If the steering assembly is not pointed in the correct direction there is a strong tendency to stop. This should be avoided as explained in Chapter 3. Using the predictive control algorithm it was found that the steering torque limit would have to be set to at least 90% of its unsaturated maximum, while the driving torque could be set to about 60%, depending on the trajectory. These recommended values are dependant on ω_N, ζ and λ. The limits can be set lower, but there will be a tendency to cause floating point errors, or a complete inability to follow the path.

Parameter	Value
λ	1.5
ζ	1.5
ω_N	2.25

Table 5.3: Parameters of the Control Equation used in Simulation

Parameter	Value
Saturation Limit on Driving Torque	0.55 Nm
Saturation Limit on Steering Torque	1.40 Nm
Safety Factor	0.95
Predictive Control Period	4 ms
Torque Time Step	2 ms

Table 5.4: Simulation Parameters used with Model based Simulations

When the simulation is run with saturation and without model based prediction, the robot often achieves an exit condition ($\omega_1 = 0$) and thus, the simulation ends prematurely. It should be noted that the exit condition is usually achieved as the robot overshoots the desired trajectory and tries to return to it by moving backward or being unable to steer quickly enough to return to the desired trajectory. Results, which have not been included, indicate significant saturation of both steering and driving torques after the expected saturations. It should be noted that the simulated vehicle without model based prediction is often capable of returning to the desired trajectory after significant saturation has occurred. During the simulations, many cases with and without model based prediction caused saturation. While these results are not included, in many cases the system remained controllable and the robot reached the desired end conditions. For the data presented here the simulation with saturation and no model based prediction always achieved a zero velocity prior to the end of simulation. In other cases, the robot's path deviated completely from the desired path.

When the saturation limits are omitted, the robot tracks the trajectory very well (see Figures 5.3b and 5.4b). The torques remain smooth as shown. It often appears that this scheme does not allow good tracking, however this is not the case. The gains have been chosen to keep the robot moving, and thus the overall position gain is small compared with the velocity and acceleration gains. The deviation looks significant because the error occurs mainly in the N direction. The root mean square (RMS) value of position error is lower for this case than the cases with saturation. The choice of filters has a significant impact on the smoothness of the output variables. Lower frequency cut-off tends to produce less variation in the state variables. Setting the cut-off frequency too low will cause the system difficulty in tracking the desired trajectory, often resulting in the exit condition.

The input jerks are recalculated every 4ms (the model based prediction period, ie. control_step) and the new torques are determined every 2 ms (torque_step). These values have significant impact on the simulation performance, but it has been observed that if the simulation is to avoid floating point overflows the ratio should be even. The choice of filter cut-off frequencies has more impact on the performance of the system as the periods are increased.

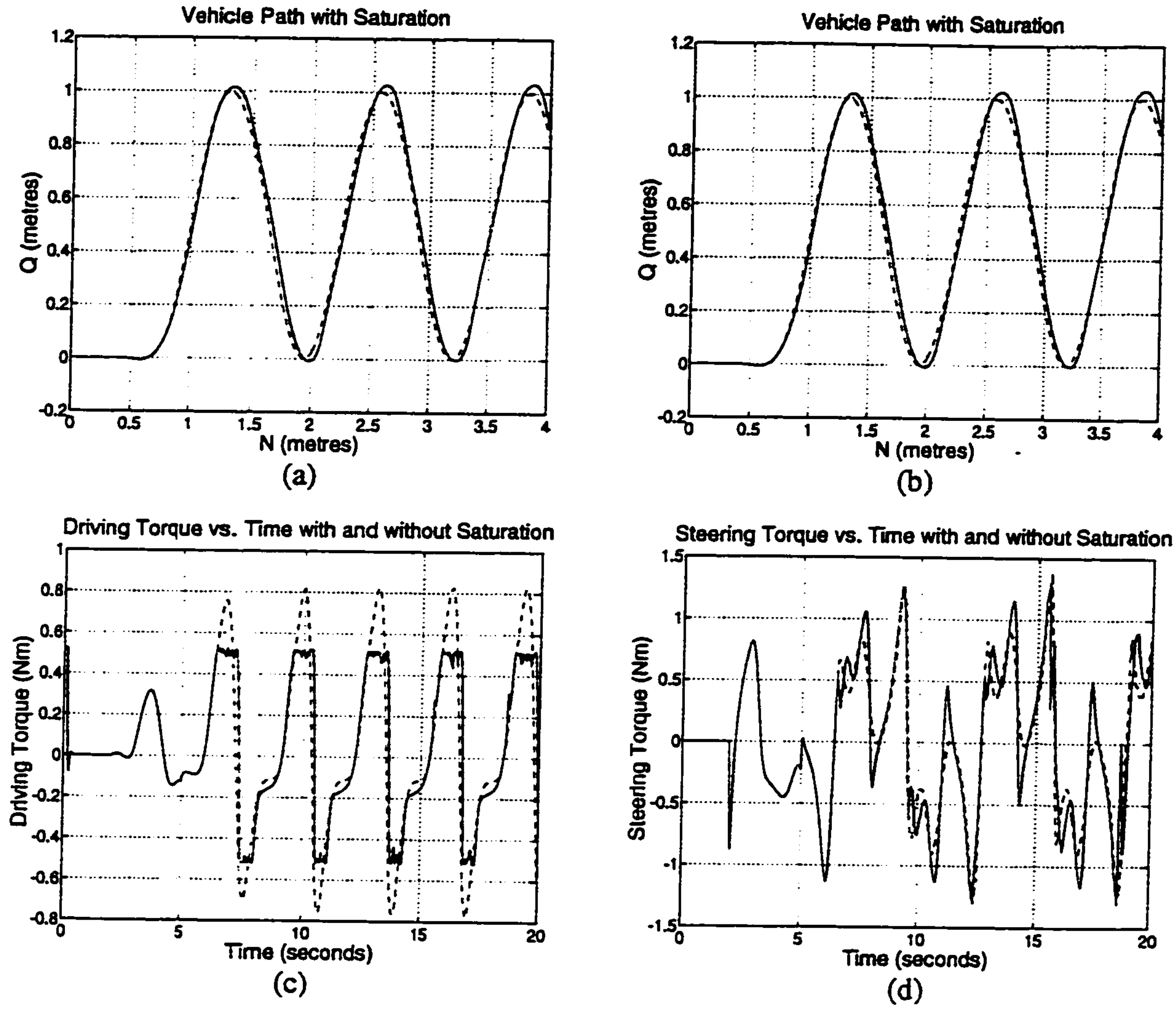

Figure 5.3: Simulation Results using first order model based prediction to follow a Sinusoid Path

a) Path achieved with torque saturation (the dotted line is the desired path)

b) Path achieved without torque saturation (the dotted line is the desired path)

c) Driving Torque versus Time for both cases (dotted lines represent the unsaturated case)

d) Steering Torque versus Time for both cases (dotted lines represent the unsaturated case)

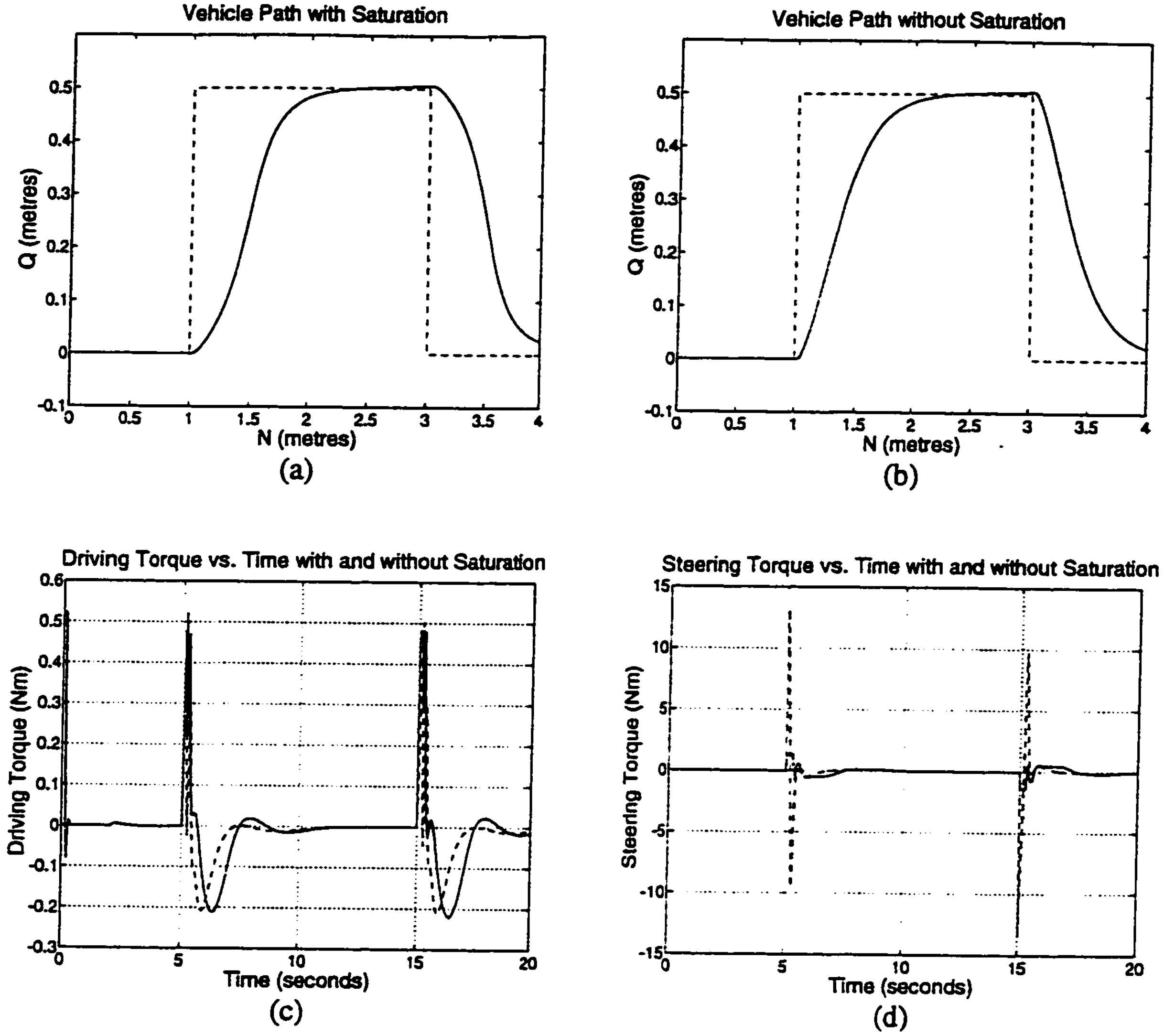

Figure 5.4: Simulation Results using first order model based prediction to follow a Square Wave Path

a) Path achieved with torque saturation (the dotted line is the desired path)

b) Path achieved without torque saturation (the dotted line is the desired path)

c) Driving Torque versus Time for both cases (dotted lines represent the unsaturated case)

d) Steering Torque versus Time for both cases (dotted lines represent the unsaturated case)

5.3: Comparison of the Controllers

In the simulations shown both the controllers were able to track the desired trajectories. The kinematic based controller was able to stay closer to the path for any of the trajectories even with saturation, however for different trajectories different values of steering stiffness had to be determined. The simulated vehicle was shown to follow the sinusoid curve well, however the square wave trajectory caused significant problems when saturation was introduced. When the steering stiffness was reduced the robot was able to follow both trajectories with actuator saturation but the accuracy of the resulting path suffered. The non-linear nature of the square wave trajectory causes the kinematic controller to request very large steering torques. These torques can be much larger than the saturation limit for steering torque. This often results in oscillation of the steering system with this trajectory.

In contrast the model based controller, was able to follow any of the trajectories with saturation of the torques and no modifications to the gains in both zero order model based prediction and first order model based prediction. It should be noted that floating point overflows were fairly common when the gains of the control equation 4.1 (λ, ζ, ω_N) were set too high or in the event that the saturation levels for steering or driving torque were very low. The torque curves generated by the model based controllers tend to be smoother than those of the kinematic based controller. There also seems to be less high frequency oscillations in the torques and state variables.

Chapter 6:

Design and System Parameterization of the CLAMOR Platform

6.1: General Requirements for the design

The main criteria to be dealt with in designing the robot was the balance of strength and weight. Sizes of most components were chosen based on the size of the motors. In an effort to ensure that no structural failures or large amplitude low frequency vibrations will occur the robot has been overbuilt. The main objective of the design was to allow experiments in which the dynamics based controller can be applied and compared to the more conventional kinematics based controller. The following factors were taken into consideration in designing the mobile robot:

- Low Cost
- Large enough to carry batteries and eventually an on board controller
- Be able to use the existing dSPACE system for controller and computer interaction
- Strong without too much weight
- Capable of operating in the area provided in the lab
- It must be consistent with the model
- Be useable for other future experiments such as inclined plane testing and steering dynamics experiments
- Stiff enough to avoid low frequency vibration
- Use equipment already available in the lab whenever possible

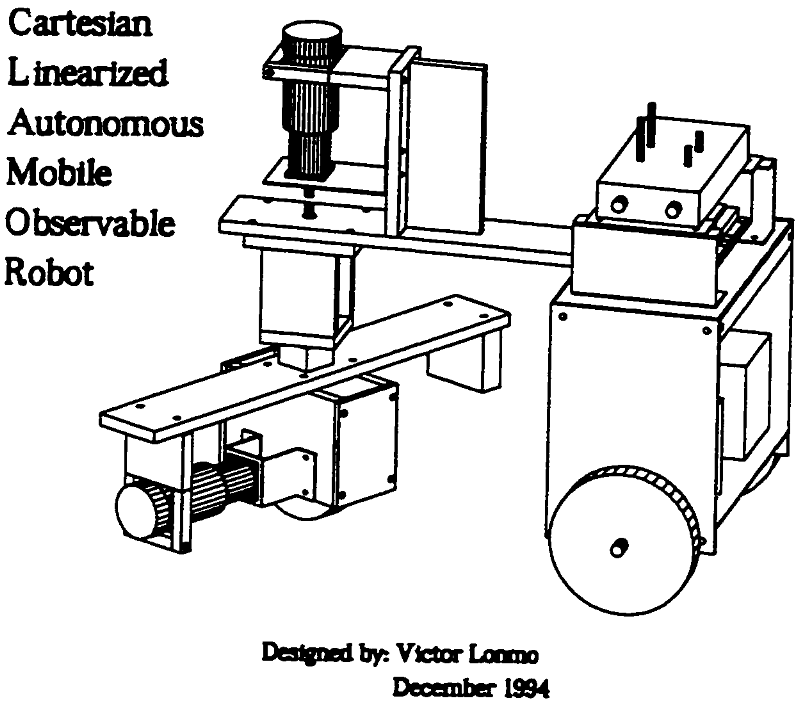

Figure 6.1: Side view of the CLAMOR Platform

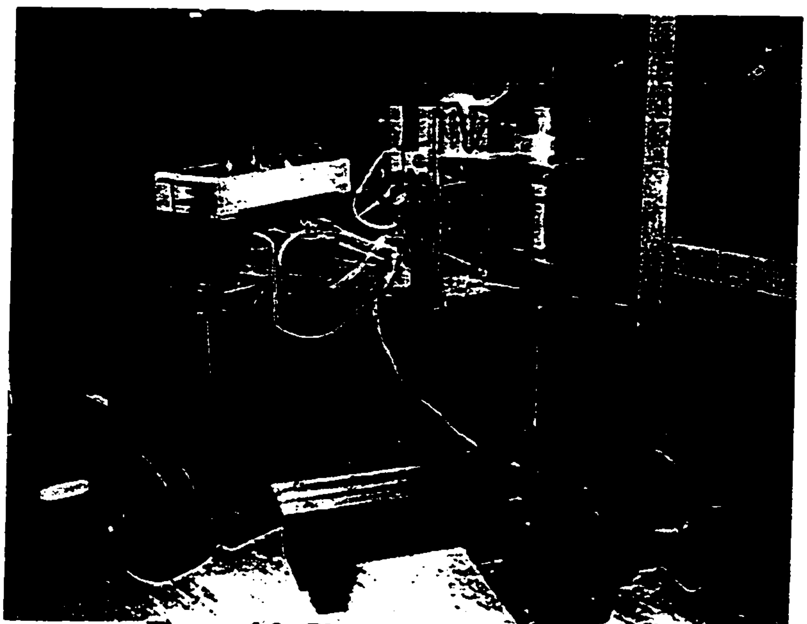

Figure 6.2: Photograph of CLAMOR

6.2: The Mechanical Design of the CLAMOR Platform

The tricycle configuration requires two motors which act on the front wheel, one for driving and the other for steering. In order that the model used remain simple and valid, the centre of mass of the vehicle must not move as the vehicle steers. Thus, the weight of the driving motor must be offset by a counterweight (see Figure 6.3). The counterweight's mass has been reduced by keeping it as far away from the steering axis as possible.

Having the mass of two motors and the counterweight on the front wheel results in a large normal force on the front wheel. To ensure that the weight of the vehicle is evenly distributed two batteries have been mounted at the rear of CLAMOR. The batteries have a 10 amp hour rating which allows very long periods of testing between recharging. The frame is made of half inch by three inch aluminium stock for strength. The plate metal used for the box near the rear wheels is 3/32 of an inch thick aluminium.

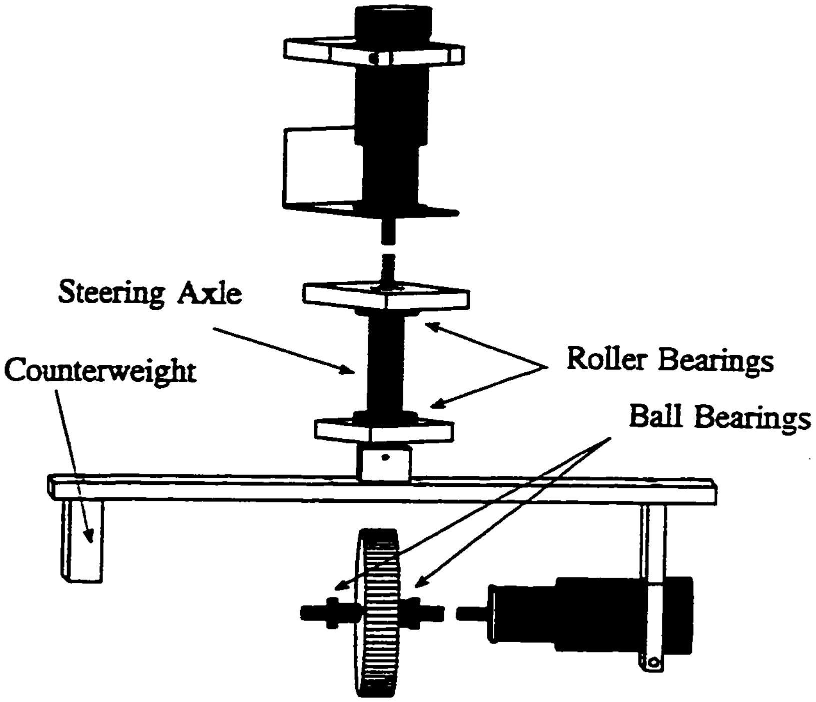

Figure 6.3: Driving and Steering Motors on the CLAMOR

The steering assembly is held in place by a steering axle. The steering motor should not bear any stresses except for those resulting from the steering torque; to do this the steering axle is held in place by a pair of roller bearings (see Figure 6.3). The distance between the bearings is roughly ten centimetres to reduce the amplitude of any vibrations. Steering experiments involving analysis of the steering wheel's contact with the ground required additional weight be placed over the front wheel. (This experiment is not otherwise referenced in this thesis.) The roller bearings for the steering axle are capable of supporting a 200kg load for this purpose. A flexible coupling joins the steering motor to the shaft.

The width of the steering assembly should be kept to a minimum in order to reduce the overall size of the vehicle. The wheel must be directly over the steering axis in order that the model remain valid. Therefore, the minimum width is given by the axial length of the motor plus half the wheel plus the coupling and the plate used to hold the wheel in place. The chassis of the vehicle must be long enough to allow the steering assembly to rotate an entire revolution.

To minimize friction the driving axle is held in place by bearings (see Figure 6.4). For symmetry the assembly extends the same distance in each direction. This was not required, but having it shorter would require that the mass of the counterweight be increased. The driving and steering motors are both protected by aluminum mounting brackets. While the motors themselves have a very strong case the encoders fitted to their ends are comparatively fragile.

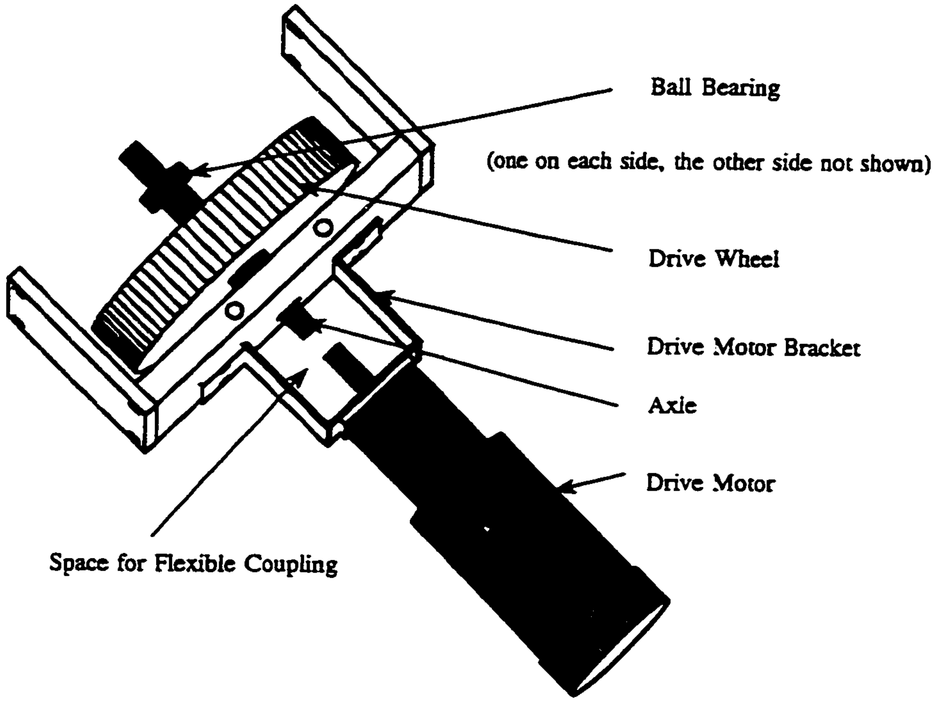

Figure 6.4: The Driving Wheel and Motor

Since the CLAMOR had to be designed before the dimensions of the batteries were known, it was decided that building a large box at the end would be appropriate. This would allow fairly large batteries and provide the extra space needed for an on board computer. The amplifiers (and their sockets) are mounted above the box, in their own isolated box. This allows them to be removed from the platform if the motors are to be used elsewhere. A plastic box shields the sockets from dust while providing space for co axial cable connectors (which are inputs for the amplifiers). Additional information about the sizes and masses of the CLAMOR platform and some of its major components are given in Appendix H.

During the initial testing of the vehicle some amplifiers were damaged. The exact reasons for it were unclear however, it was feared that they were overheating. Since they were already bolted to an aluminium plate (which has very high thermal conductivity) it was decided that air should be forced over the plate that holds them. This was accomplished by drilling two large holes in the box holding the batteries which allow a pair of small fans to blow air on the back of the plate.

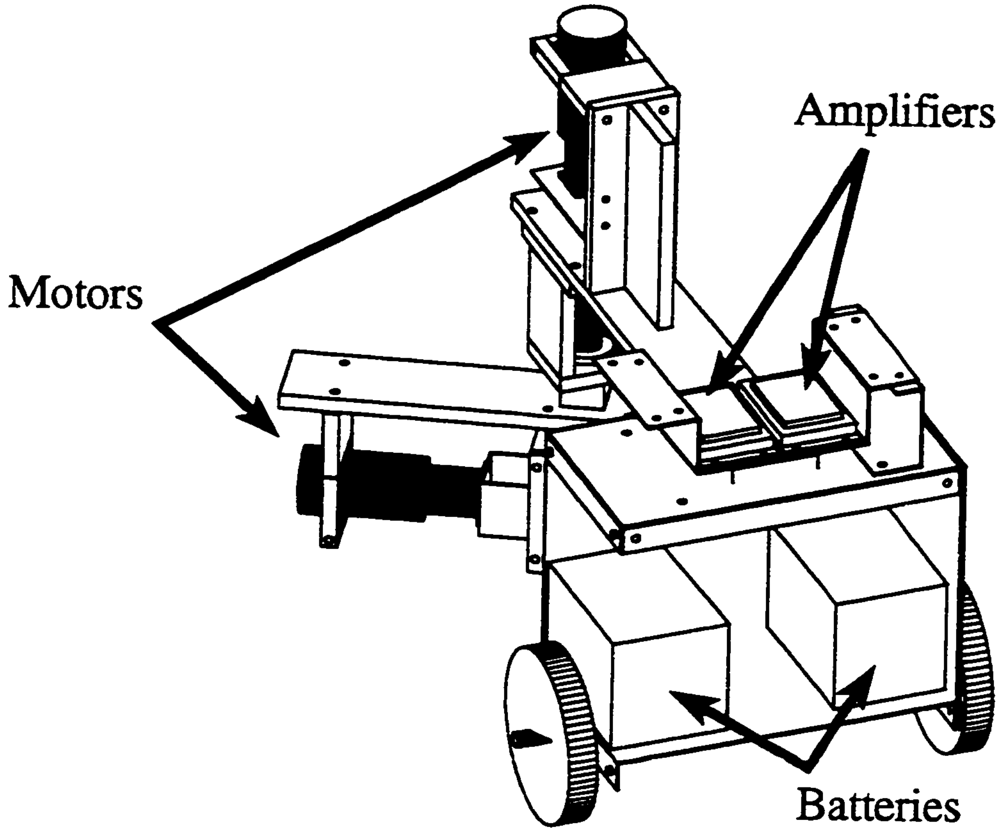

Figure 6.5: Major Components of the CLAMOR Platform

6.3: The Electrical Hardware of the CLAMOR Platform

The selection of electrical components for the robot is as critical as the mechanical design itself. The electrical hardware of the CLAMOR platform consists of two batteries, two motors, an interface which allows the dSPACE computer to read the encoders and a pair of signal buffers.

The motors used are Pitmo 14202 series D-C Servo Motors fitted to 75.1:1 planetary gear train. These components are from the Pittman corporation. Power for the motors is supplied by two 12 V batteries connected in series. The batteries have a 10 Ah (amp hour) rating. Each motor is controlled by a Cleveland Machine Controls PA228-001 amplifier with the PA221 current amplifier socket. The socket is connected to co-axial connectors for the input signals. Although the current amplifier is expected to process differential input signals, it was found that the return line had to be grounded in order to work properly. Due to the concern that grounding the return line of the ds2101 digital to analog converter (DAC) may cause ground loops, a buffer was introduced which would ground the return line while isolating the DAC board of the dSPACE computer.

Each motor is also fitted with a Hewlett Packard incremental optical encoder which has 1000 steps per revolution (providing 75100 steps per revolution for the output shaft). The encoder's outputs are boosted by Schmitt triggers. These components are all powered from the ds3001 board incremental encoder board (see Figure 6.6) but their power use is minimal.

The vehicle is controlled by a computer which is not on board and this results in a vehicle which is not truly autonomous. A computer could be mounted on CLAMOR if the equipment were made available.

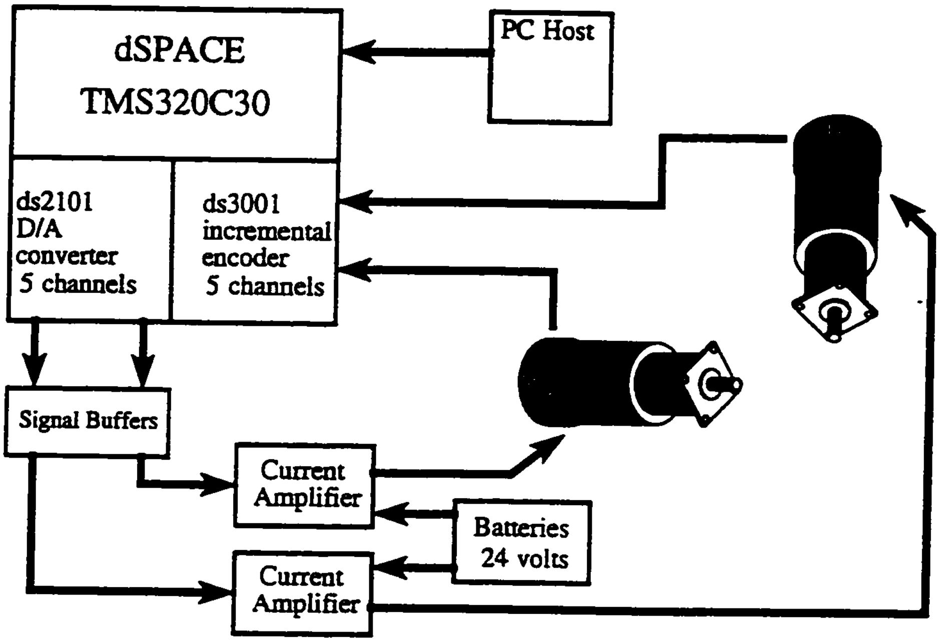

Figure 6.6: Electrical Hardware of the CLAMOR Platform

6.4: Parameterization of the amplifier and motor combination

In order to successfully implement linearizing feedback on the CLAMOR the output torques requested by the controller must be produced precisely. The simulation outputs the torques that are produced by one of the given controlling routines presented in detail in Chapters 3 and 4. The assumptions which have been made for the simulations are:

1) No dissipative friction

2) Thin, perfect disks for the wheels

3) Zero time constant for the motor

4) No backlash in the gear train

In the actual system, these assumptions must be carefully reconsidered. The first two assumptions are obviously incorrect but their effect is considered small. The other two are not unreasonable in that the time constant and backlash can be neglected if high quality components are used.. To compensate for the differences between the simulated torque and actual torque a simple friction model was used.

The amplifier-motor systems were parameterized using a friction identification routine. The parameterization attempts to evaluate: inertia, dry friction, viscous friction and Stribeck effect (which is represented by the coefficient alpha). No attempt has been made to deal with backlash or time delay in the system. The friction model is a Bo-Pavelescu type. The version for low rotational speeds is shown below:

$$V_{OUTPUT} = K\left[sign\,(\tau_{Model})\alpha e^{-\left(\frac{\omega_M}{\epsilon}\right)^{1.5}} + \tau_{dry} + f_{viscous}(\omega_M) + (\tau_{Model})\right] \tag{6.1}$$

where: V_{OUTPUT} represents output voltage which is proportional to motor output torque

$K = 0.4748$ (volts/Nm) as shown in Appendix F, Equation F.2

α is the Stribeck effect coefficient

ω_M is the rotational velocity of the given motor (rad/s)

ε is a small rotational velocity, 0.05 (rad/s)

τ_{dry} is the dry friction (Nm)

$F_{viscous}$ is the viscous friction in Nm/(rad/s)

τ_{Model} is the torque that is generated by the given controller

sign( ) is a cubic polynomial curve fit

At higher operating speeds the value of rotational velocity (ω_M) is substituted for τ_{Model} in the sign function.

The results shown in Figures 6.5 and 6.6 demonstrate that the routine has achieved stable final values for all four parameters, however the estimated error remains very high. The graphs showing the approach to the final values do not start at the same time as the friction identification of the system because the program reads data from the dSPACE system in fixed lengths of time. Thus, while the output shown corresponds to sixty seconds, the program was run for roughly twenty minutes when evaluating the steering system. The same program had been used successfully to parameterize direct drive motors. The parameterization of the direct drive motors was significantly more accurate as the estimated error was roughly one order of magnitude lower than the acceleration.

During the test the drive wheel was not allowed to make contact with the ground. This had the benefit of producing a value for the moment of inertia of the drive wheel, coupling and rotor, but produced no information about the rolling contact with the ground. The steering motor was tested while the drive wheel was in contact with the ground. Thus, the contact friction was identified in steering.

Parameter Name	Steering Motor	Driving Motor
Dry Friction (τ_{Dry}) (Nm)	1.2	1.1
Viscous Friction (Nm / (rad/s))	0.9	0.26
α (m^2 rad/kg s^2)	0.089	0.104
Estimated Inertia (kgm^2) (used in the model)	0.38	0.0000977
Actual Inertia (kgm^2)	0.58	0.23

Table 6.1: Friction and related parameters of the motors

The values used for the moment of inertia of the driving wheel and axle combination are very low due to an error in the program which caused the moment of inertia of the motor shaft as seen from the drive wheel to become negligible. The moment of inertia of the drive motor is fairly high as seen from the output shaft of the motor, however it is still quite small compared to the mass of the robot.

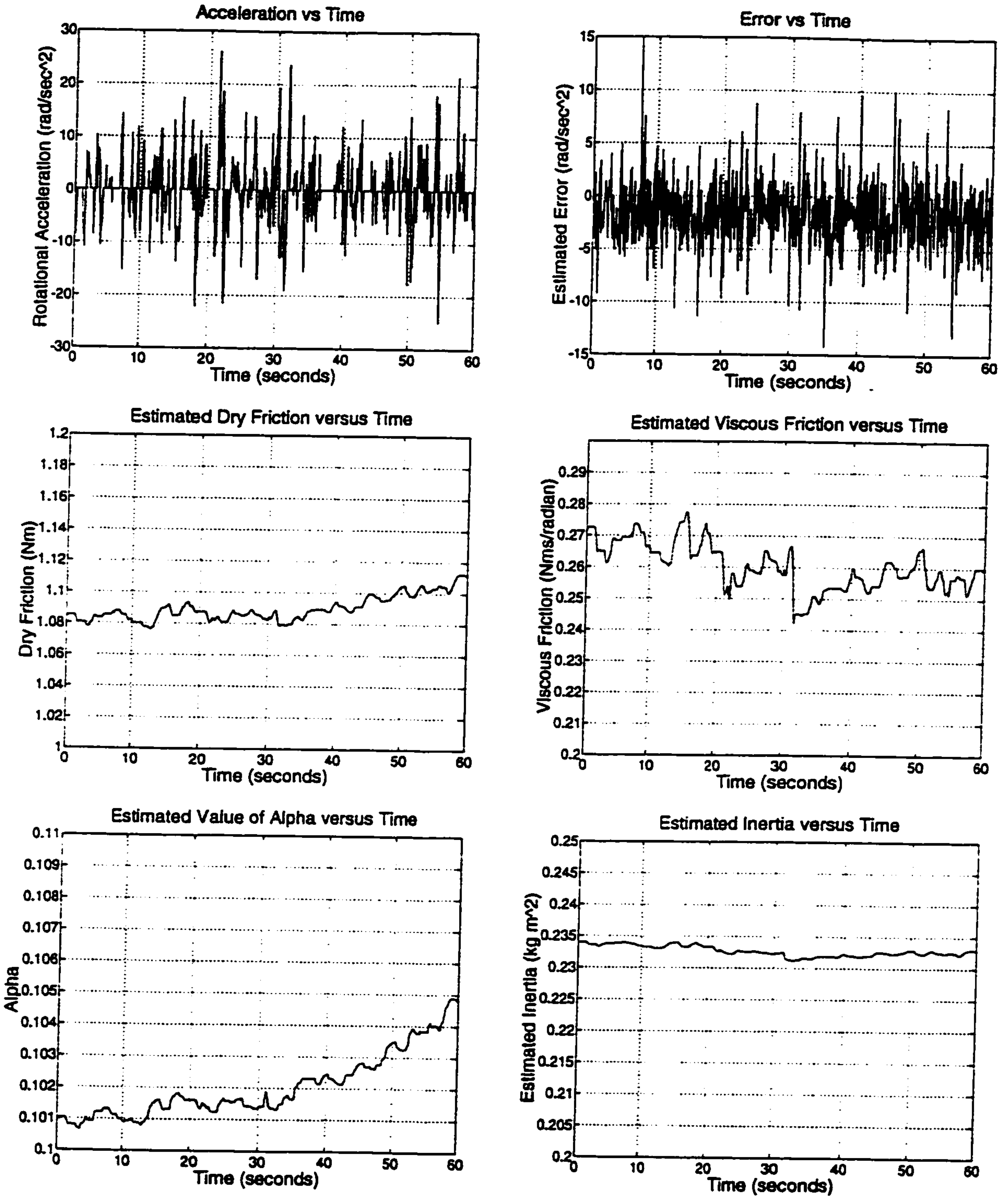

Figure 6.7: Parameters found for the Driving Motor

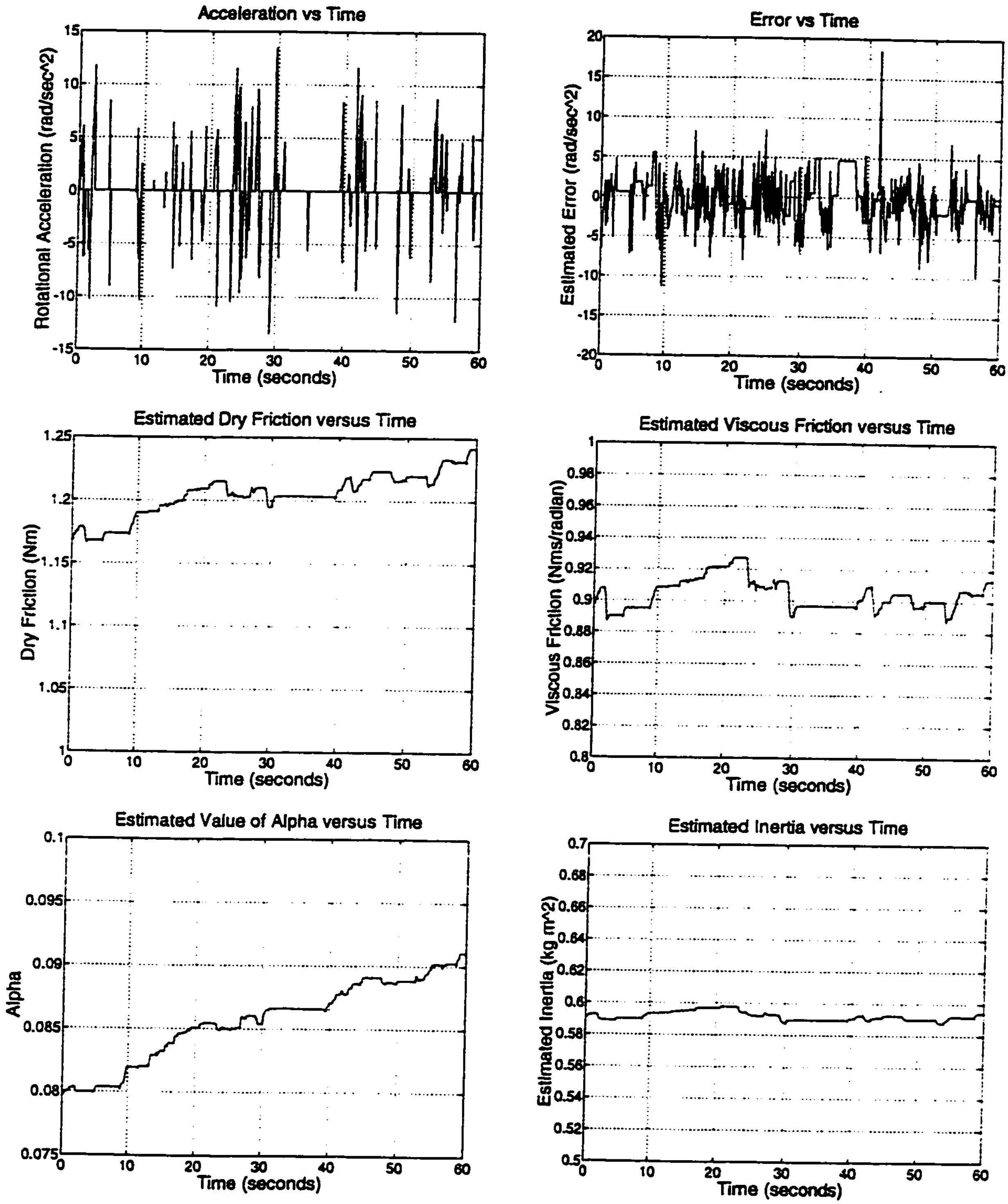

Figure 6.8: Parameters found for the steering motor

Chapter 7:

Experiments using the CLAMOR Platform

7.1: General Observations on the CLAMOR platform

The CLAMOR platform is more difficult to control than the simulated robot. This difference between the simulations and the experiments was not unexpected. The most likely cause of the differences is the model of the amplifier, motor and gearbox. The control of the robot is complicated by noise in the numerical derivatives of the encoder signals. This noise may be the result of backlash in the gears, randomness in the environment or some interference produced by the buffer/amplifier/motor combination. It was common to see the steering assembly shaking (about the steering axis) as the experiments were performed with filters having high cut-off frequencies. With lower cut-off frequencies the shaking is less apparent, but still present. Reducing the frequency too low has a tendency to cause the vehicle to ignore changes in the trajectory, so a compromise was reached with a cut-off frequency of 4 Hz. It should be noted that all the simulations were unstable with a filter cut-off at this value (or lower values) which can be explained by the inaccuracies of the model used in simulations.

In order to improve the quality of the encoder signal's derivatives a 2ms time step was tested. While the signals were improved significantly, the robot's performance was not improved. In an attempt to get better response from the robot the time step was reduced to 0.5ms. This caused a significant increase in the noise of the higher derivatives of the encoders' outputs with a slight worsening of the results. All results presented were generated using a 1ms time step.

The paths generated for the sinusoid curve and gaussian curve inputs are repeatable, which suggests that better knowledge of the unmodelled dynamics could lead to better results. The robot requires much higher torque limits than the simulated robot for the same trajectories. This is likely due to problems in the identification of the parameters of the motors as well as rolling resistance for the wheels. Another difference between the simulation and the experiments are errors in odometry. All of the outputs in this report are generated by the reading the encoder outputs and applying the kinematic relations to generate position in the NQ frame. In other words, the trajectories shown in the outputs are not always consistent with the actual motion of the real robot. The error is mainly noticeable in θ (the orientation angle of the robot's frame relative to the inertial N,Q frame). While the exact cause is uncertain, the error in θ is most noticeable when the robot is moving slowly, increasing speed (ω_1) with a high steering angle (δ) and when the filter cut-off frequency is low. Alternative methods for recording the position were considered. The only reasonable means of doing so is to have the robot draw a line on the floor as it runs. The position could then be compared with the position established by odometry and compared.

7.2: The Experimental Program

The simulation was designed to allow errors in odometry, however they were rare due to a higher average speed as well as much higher filter cut-off frequencies. This problem could be corrected with higher speeds, but the space available for testing made such tests impractical. Lower cut-off frequencies reduce high frequency information about the signal itself and introduce a phase lag, producing a significant difference between the actual value and the value which is computed, resulting in poor odometry.

The driving torque has been allowed to saturate during the first two seconds. This has been done to ensure that the robot's initial position and speed are consistent with the trajectory. The controller tested actually begins operating at time = 2 seconds. It continues operating until time = 20 seconds unless the absolute value of the steering angle is found to exceed 120 degrees (or $2\pi/3$ radians) or, in the model based controller case, if the robot stops ($\omega_t = 0$). After that point the signals from the dSPACE computer are set to zero which causes the robot to slow down due to friction. Since the speeds are fairly low, the robot stops quickly. There is presently no system designed to stop the robot, although a kinematic controller whose inputs are zero speed and no position error might perform adequately. The torque limits are given in table 7.1, and a safety factor of 0.9 was used. The system is theoretically capable of producing torques as high as 10 Nm for each motor, however since the actual output torques are also functions of current velocity and other variables (see Chapter 6.3), it was necessary to ensure that neither quantity would saturate. These limits were obtained by running the experiment with the dynamics based controller with first order model based prediction tracking a sinusoid trajectory. The same saturation limits are used with both controllers to keep the comparison consistent.

Case:	Motor:	Simulated Torque Limit: (Nm)
Unsaturated	Driving Torque	1.35
	Steering Torque	4.5
Saturated (Kinematic Based)	Driving Torque	no limit used (~10)
	Steering Torque	no limit used (~10)
Saturated (Model Based)	Driving Torque	8.0
	Steering Torque	8.0

Table 7.1 : Torque Limits and Safety Factor for the Experiments using CLAMOR

7.3: Experimental Results using the Kinematic Based controller

The kinematic controller performed well. Unlike the simulations which featured a set of cases with low steering stiffness and another with stiff steering, all of the experiments carried out with the kinematic based controller used low steering stiffness. As torque saturation was less common in the experiments, the robot performed very well. Table 7.2 lists the gains used with this controller.

Gain	Equation Number	Value
Feedforward Position with high gains (Kp)	4.2	1.5
Feedforward Position with low gains (Kp)	4.2	0.5
Steering Derivative gain (KDs)	4.4	3.0
Steering Proportional gain (KPs)	4.4	12.0
Driving Derivative gain (Kt)	4.6	1.0

Table 7.2: Gains used with the Kinematic Controller in Experimentation

The results in figures 7.1 and 7.2 clearly demonstrate the kinematic controller's strengths and limitations. This controller is fairly accurate with high gains, no saturation and a continuous path. This controller uses less torque than the dynamics based controllers in the case of following the sinusoid trajectory (see figure 7.1). When the path is discontinuous, the robot is unable to follow it without using very high torques. The path shown in figure 7.2a is truncated because the steering angle's absolute value exceeded $2\pi/3$ radians causing the experiment to stop.

In an effort to improve performance, the gains were reduced. The robot was able to follow both trajectories with torque saturation this way however the tracking was not particularly accurate (see figures 7.3 and 7.4). The effect of reducing the gains is most visible by comparing figures 7.4b with 7.2b.

Figure 7.2b illustrates a general problem with the kinematic controller. The steering PD controller is unable to properly position the steering angle. As a result the trajectory oscillates. If this controller were able to consider the effect of the steering angle on the actual motion of the vehicle then this oscillation might be avoided.

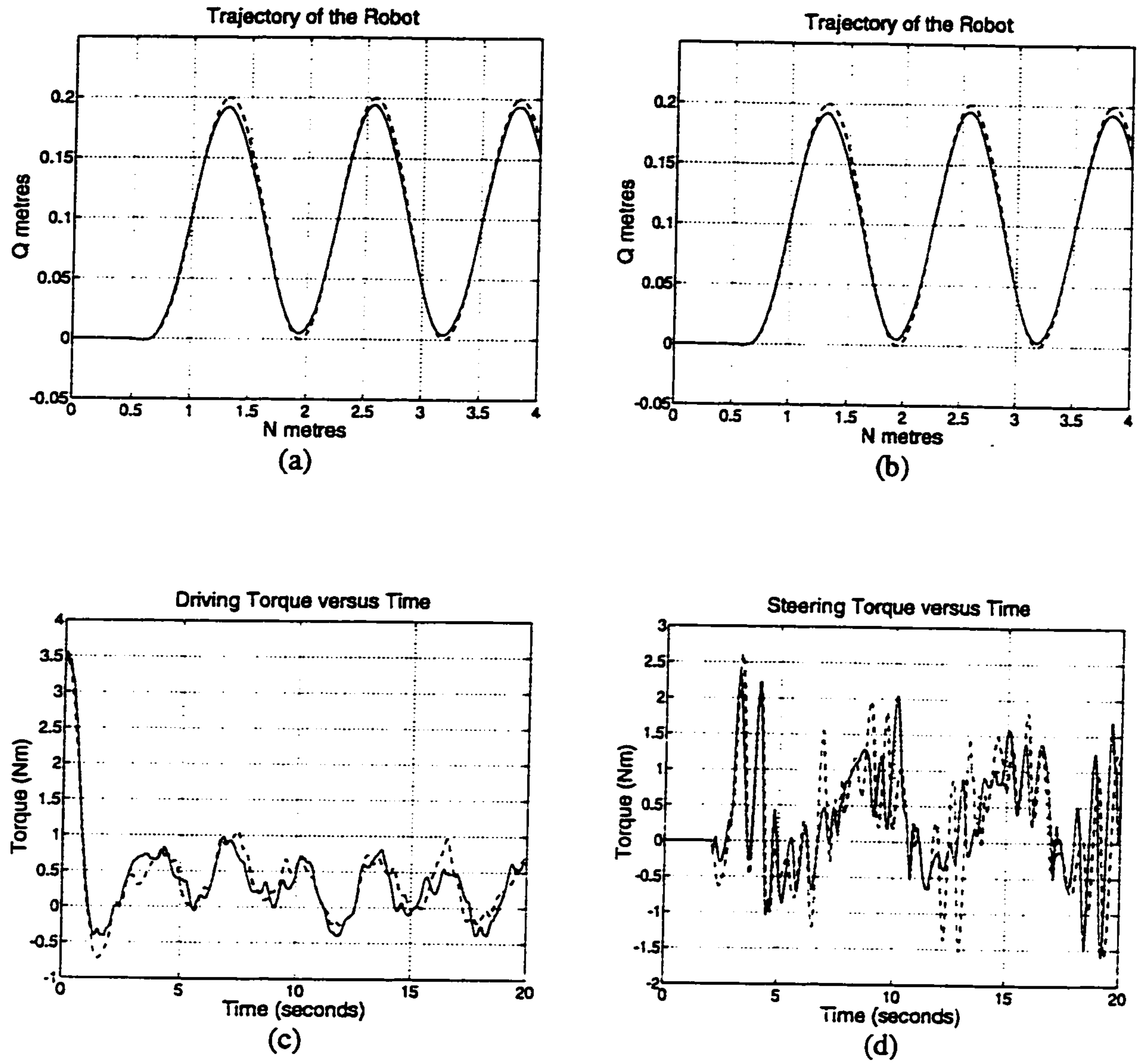

Figure 7.1: Experimental Results with a Kinematic based controller with high gains tracking a sinusoid trajectory

a) Trajectory with torque saturation (the dotted line is the desired path)

b) Trajectory without torque saturation (the dotted line is the desired path)

c) Driving Torque versus time (the solid line represents the saturated case)

d) Steering Torque versus time (the solid line represents the saturated case)

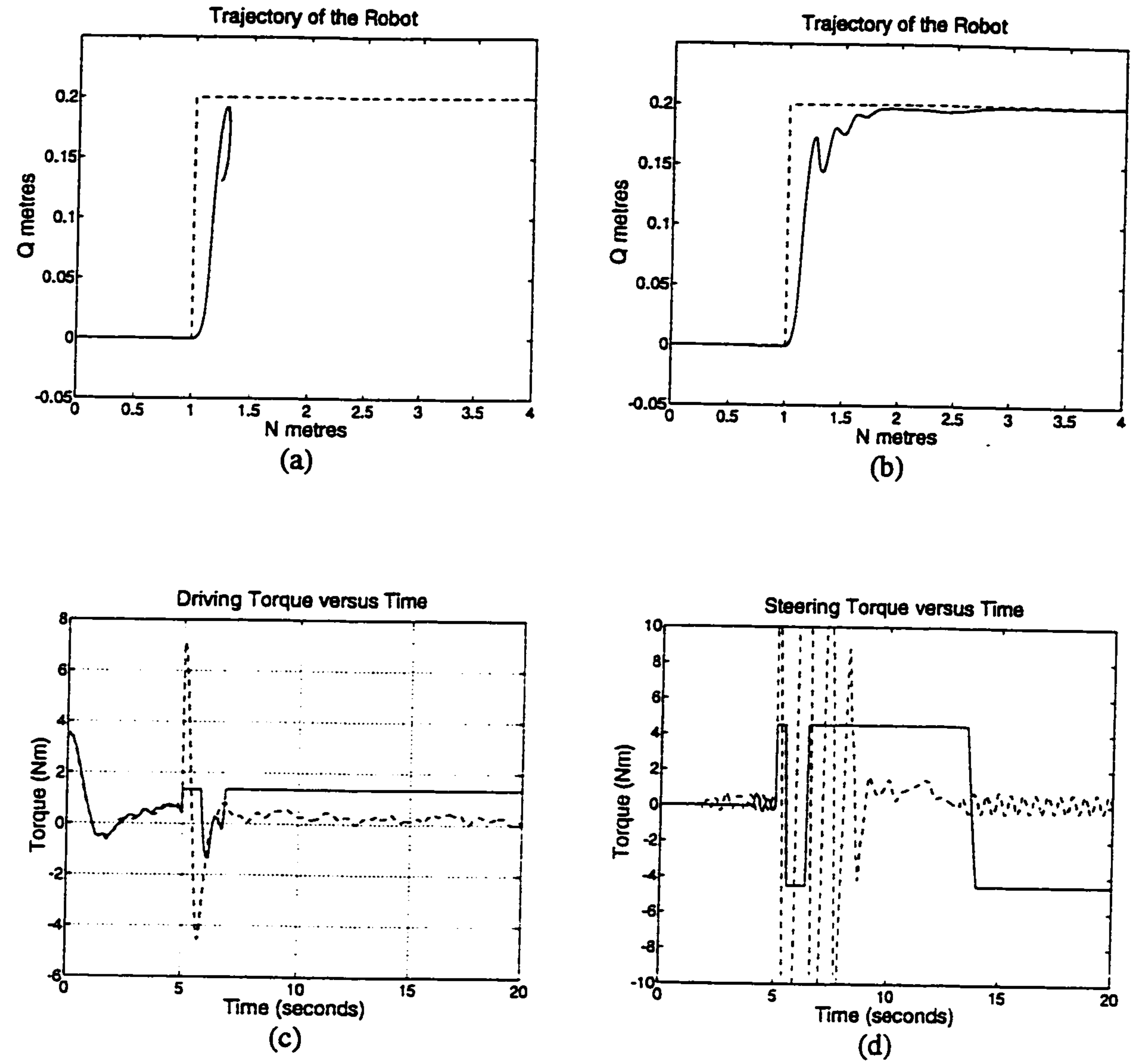

Figure 7.2: Experimental Results using a Kinematics based controller with high gains following a step input

a) Trajectory with torque saturation (the dotted line is the desired path)

b) Trajectory without torque saturation (the dotted line is the desired path)

c) Driving Torque versus time (the solid line represents the saturated case)

d) Steering Torque versus time (the solid line represents the saturated case)

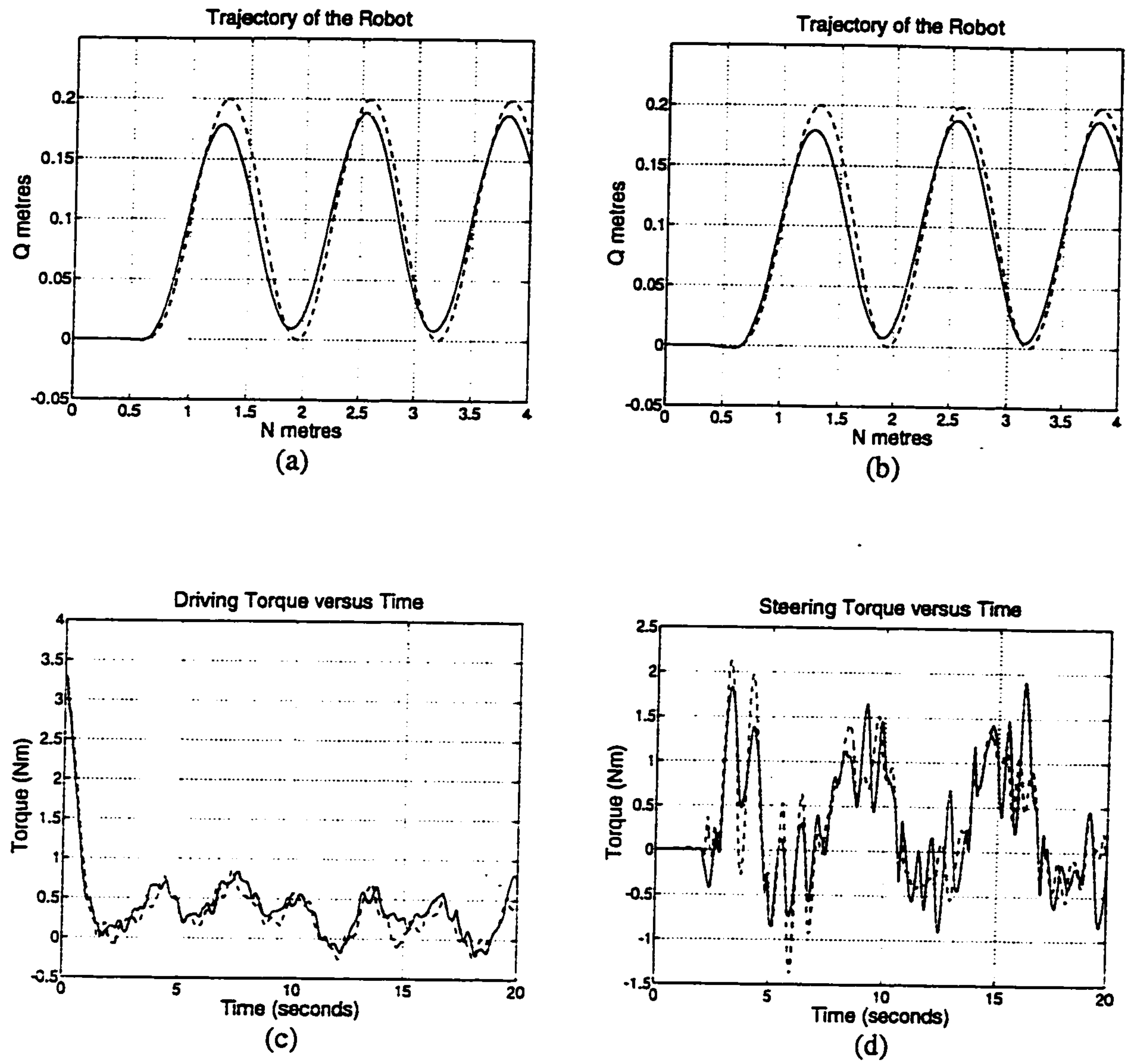

Figure 7.3: Experimental Results using a kinematic controller with low gains to follow a Sinusoid Trajectory

a) Trajectory with torque saturation (the dotted line is the desired path)

b) Trajectory without torque saturation (the dotted line is the desired path)

c) Driving Torque versus time (the solid line represents the saturated case)

d) Steering Torque versus time (the solid line represents the saturated case)

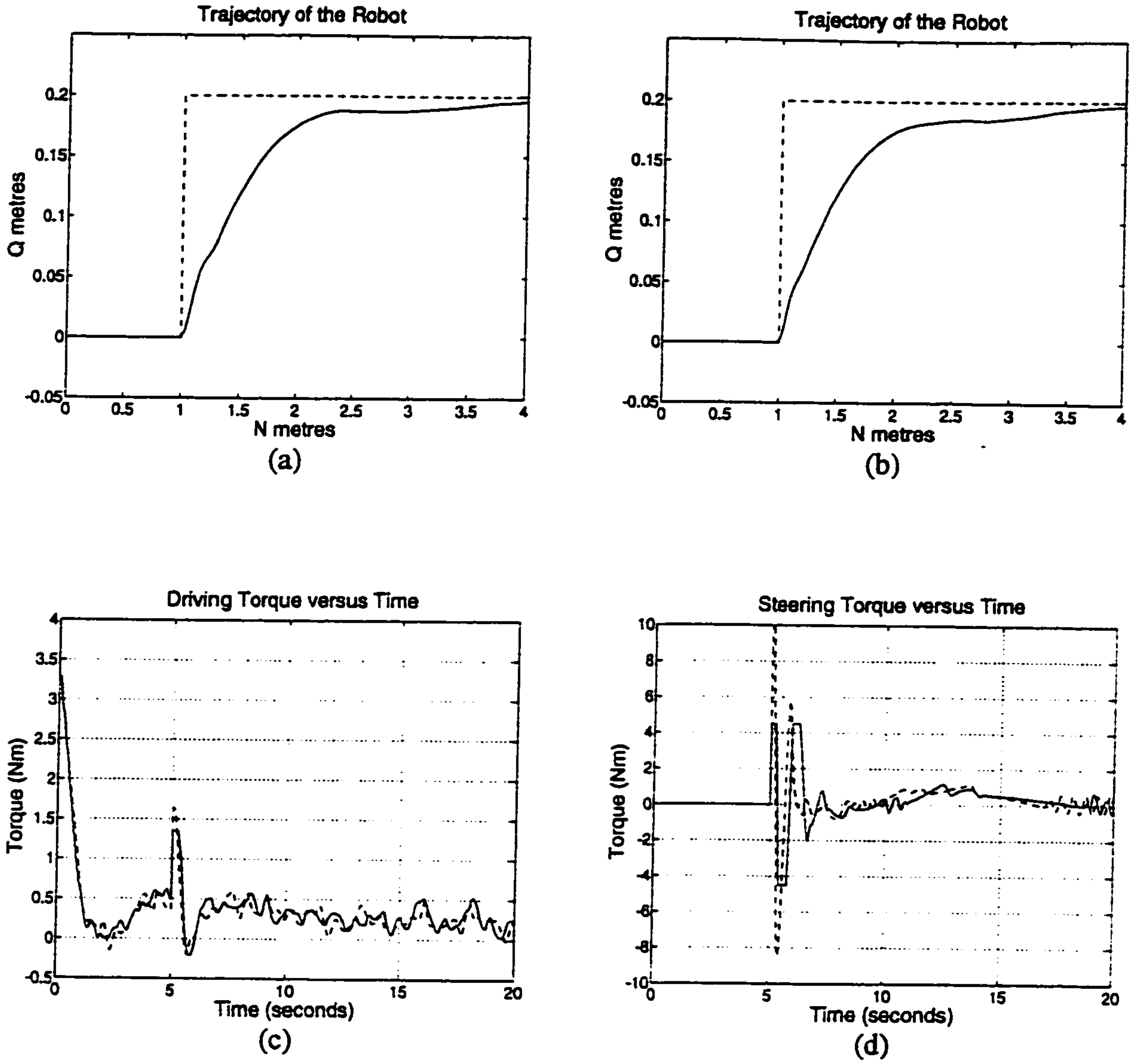

Figure 7.4: Experimental Results using a kinematic controller with low gains to follow a Step Trajectory

a) Trajectory with torque saturation (the dotted line is the desired path)

b) Trajectory without torque saturation (the dotted line is the desired path)

c) Driving Torque versus time (the solid line represents the saturated case)

d) Steering Torque versus time (the solid line represents the saturated case)

7.4: Experimental Results using the Model Based controller:

The model based controller was able to follow the trajectory of the robot without allowing the torques to saturate. The model based prediction period was set to 2ms and the desired torques were recalculated every 1ms and the filters used a 4Hz cut-off. The coefficients of the control equation (4.1) are given on Table 7.3:

Coefficient in the Controlling Equation	Value
λ	1.0
ζ	4.0
ω_N	3.0

Table 7.3: Coefficients of the Model Based Controlling Equation used in Experiments

This controller used more torque for driving and steering in the sinusoid trajectory case than the kinematic controllers however it was more accurate (see figure 7.5). The steering torque shows a considerable amount of oscillation. This is opposite to the result seen in the simulations. The reason for this fluctuating torque is most likely to be the result of using signals based on the second derivative of the encoder outputs as feedback to the controller. The saturation limit on driving torque used was found to be the lowest that would allow the robot to work without stopping prematurely. The system did not allow saturation with either the first order or zero order controllers. This controller with model based prediction was able to make the robot follow the step trajectory with minimal steering oscillation. Errors due to odometry were not nearly as apparent as they were with the kinematic based controller for this trajectory.

This controller is very versatile because it can follow both the continuous (figure 7.5) and discontinuous (figure 7.6) trajectories with no modifications of the control gains. This is an ideal controller for a system that must generate new trajectories as it runs.

With very low torque saturation limits, this controller often fails, causing the CLAMOR to stop prematurely. It is felt that the problem is a result of high input jerks which lead to floating point overflow, but there is no evidence to support this hypothesis except that it is consistent with similar failures of the simulation. Due to the way the dSPACE shares information with the PC host computer it would be very difficult to verify if that is the case. Additional results using zero order model based prediction are given in Appendix G.

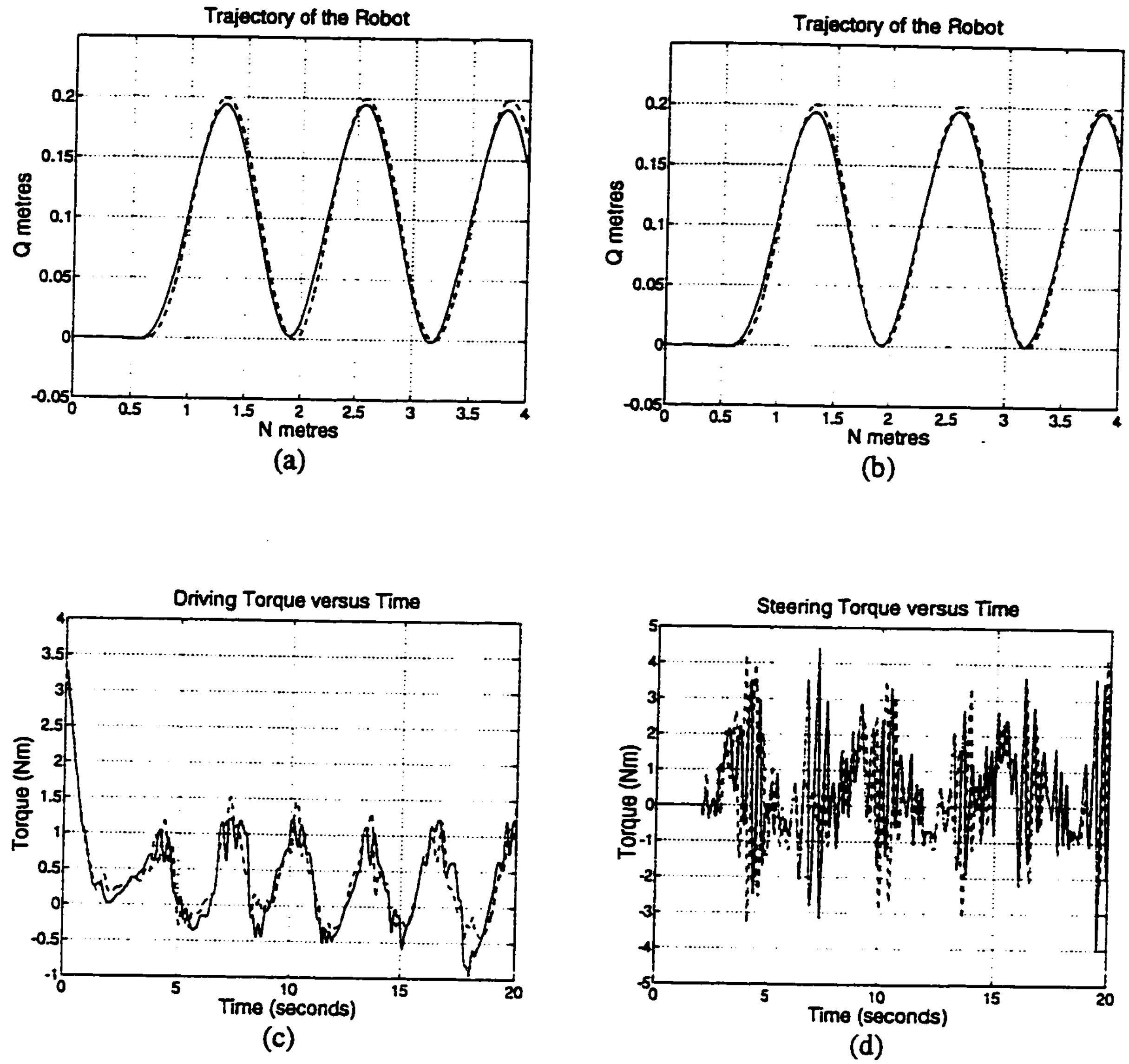

Figure 7.5: Experimental Results using first order Model Based Prediction to follow a Sinusoid Path

a) Trajectory with torque saturation (the dotted line is the desired path)
b) Trajectory without torque saturation (the dotted line is the desired path)
c) Driving Torque versus time (the solid line represents the saturated case)
d) Steering Torque versus time (the solid line represents the saturated case)

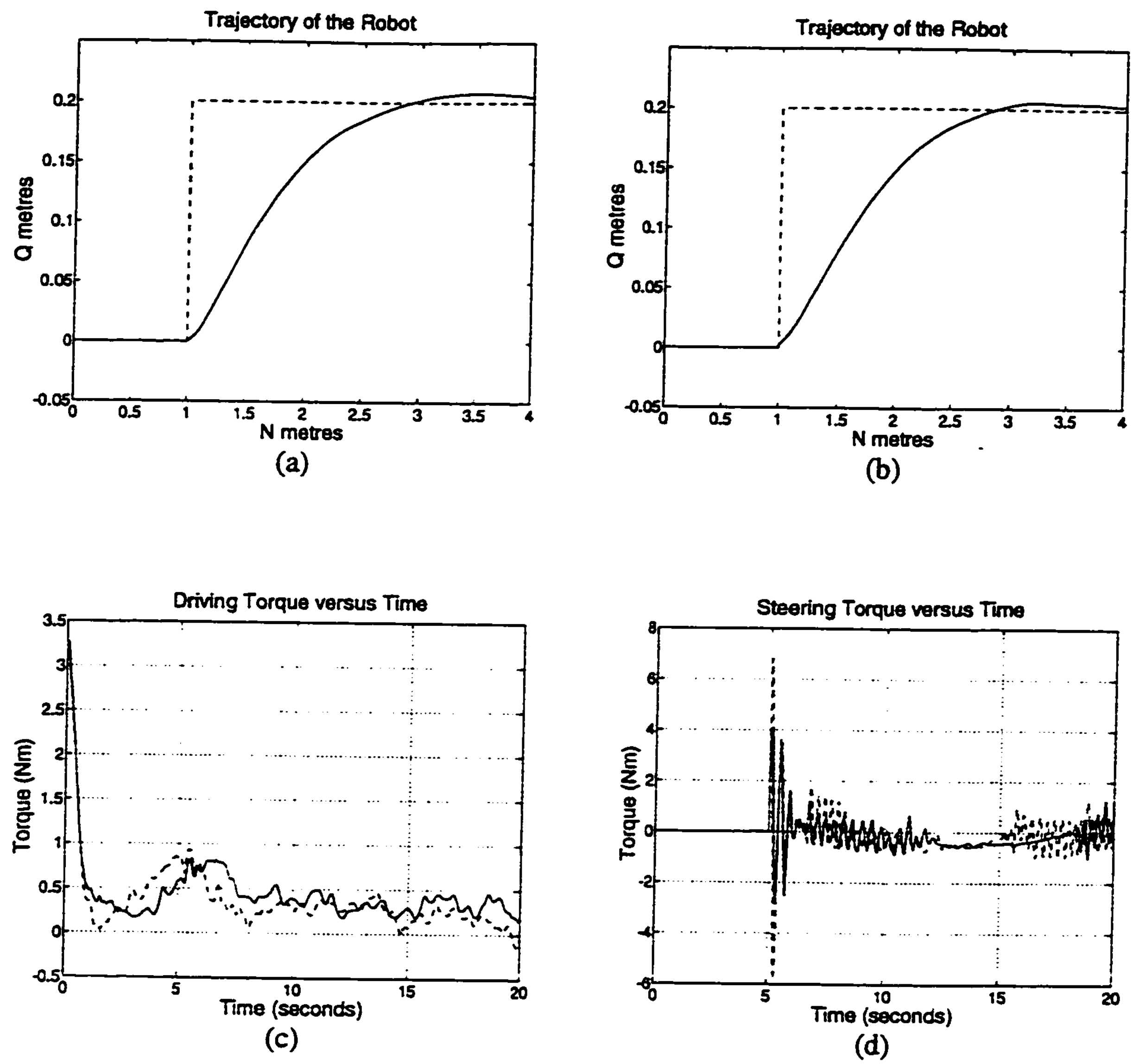

Figure 7.6: Experimental Results using first order Model Based Prediction to follow a Step Path

a) Trajectory with torque saturation (the dotted line is the desired path)

b) Trajectory without torque saturation (the dotted line is the desired path)

c) Driving Torque versus time (the solid line represents the saturated case)

d) Steering Torque versus time (the solid line represents the saturated case)

7.5: Comparison of the Model based and Kinematic based Controllers

The kinematic controller works by controlling the driving and steering systems separately and it's behaviour reflects this. This is especially apparent in the steering oscillation produced in the case with a step input trajectory and saturation (see Figure 7.2b). The kinematic controller is able to follow the sinusoid path quite well when it is using high gains.

The model based controller performs very well for all the trajectories with and without torque saturation. The slight steering oscillation present in the model based controller is attributed to the use of the second derivatives of the encoder's signals to determine feedback signals for the controller. This controller does have problems with low saturation limits causing errors which force the dSPACE to cease operating. (This error may be the result of floating point overflows in the dSPACE probably cased by high error signals.)

The kinematic controller requires a new set of gains to achieve the best accuracy in tracking. This is very inconvenient in terrestrial applications and unacceptable for a planetary rover.

Conclusions:

1) The simulations clearly demonstrate excellent performance with the model based control using zero order or first order model based prediction. The kinematic controller can produce better results than the model based controller but the results vary considerably between different reference trajectories. To improve the response with the kinematic controller the steering gains must be modified. The kinematic controller will control the robot well when the saturation limits on the actuators are omitted, however it will demand unrealistically high torques. The model based controller is able to give good results without the need to modify the control equation.

2) Experiments clearly confirm the conclusions obtained from the simulaltion. The kinematic controller must be modified for different pathes if torque saturation is present. The experiments also show that the feedforward gains used in simulation were not sufficient to control CLAMOR accurately; it should be noted however that the amplitude of the sinusoid path in simulations is much greater than it was in experimentation.

3) The CLAMOR platform has been used in experiments with lower filter cut-off frequencies than those used in the simulation. Noise in the second derivatives of the encoder signals was present so lower cut-off frequencies were used with the actual robot. These low frequency filters caused oscillations leading to failure in the simulations. Since the same oscillations were not present in the experiments it is suggested that the electrical time constant is higher than the literature (see appendix F) would indicate.

Recommendations:

1) In order to improve the performance of the robot, the motors should be parameterized with a more accurate model. The tests should be run with the inertias determined by the identification routine and not those arrived at mathematically. Alternatively, direct drive motors could be used to improve performance. Direct drive motors which can be powered by small batteries and can produce the kinds of torques needed were not available.

2) A full three dimensional model of the tricycle configuration mobile robot should be developed for testing the CLAMOR on an inclined plane.

3) The CLAMOR system should use an on board computer which will allow it full autonomy. This will also allow the robot to be tested in various locations. The tests shown in this thesis were conducted in a relatively small area. Genuine autonomy will also reduce any error associated with the cables including signal degradation.

4) If a larger testing area becomes available for the CLAMOR, then it is suggested that the robot be run at higher speeds than those used in the tests. The higher speeds will allow a reduction in the errors associated with odometry.

5) The amplifier and motor transfer functions should be analyzed more carefully. Testing the amplifiers will require a spectral analyzer and the motors will require some sort of dynamometer. A better understanding of this transfer function may explain why the simulation will not work with low cut-off frequencies although the experimental robot does.

6) Accelerometers could be added to the CLAMOR for improved sensor feedback. In testing it was found that the second derivatives of the encoder signals were often inaccurate. Adding the accelerometers with good sensor fusion systems will reduce this problem.

7) A more complete dynamic model could be produced which will allow the system to predict and compensate for slippage in a known environment. A sophisticated system might be able to evaluate changes in its environment (such as a slippery floor) and then make the appropriate changes (to its slippage criteria).

References

1) D. Wettergreen, C. Thorpe and R. Whittaker, "Exploring Mount Erebus by Walking Robot", Journal of Robotics and Autonomous Systems 11, 1993, pp 171-185

2) Stone, Edwards, "Hazbot: A hazardous material Emergency Response Mobile Robot", Proceeding of the 1992, IEEE International Conference of Robotics and Automation, pp 67-73

3) T. Sheridan, "Space Teleoperation Through Time Delay: Review and Prognosis", Proceeding of the 1993 IEEE International Conference of Robotics and Automation, pp 592-606

4) A. Spiessbach, R. Clark, S. Larimer, H. Tobey, B. Lindauer, R. Koenig and T. Lisec "Issues and options for a Mars Rover", Spie vol 852 Mobile Robots 2, 1987, pp 164-170

5) M. Hayard, "Mission, Technologies and Design of Planetary Mobile Vehicle", AGARD Lecture Series 193, 1994 ,Advanced Guidance and Control Aspects in Robotics, Page 6-1 to 6-7

6) P. McKerrow, "Introduction to Robotics", 1991 Addison-Wesley Publishing Company

7) A. Kemurdjian, V. Gromov, V. Mishkinyuk, P. Sologub "Small Marsokhod Configuration", Proceeding of the 1992 IEEE International Conference on Robotics and Automation ,pp 165-168

8) W. Zimmerman, "JPL Rover Technology Program", ISRAM 1994, fifth International Symposium on Robotics in Manufacturing, Wailea, Maui Hawaii, August 14-17, 1994, pp 345-354

9) S. Samuel and S. Keerthi, "Numerical Determination of Optimal Non-Holonomic Paths in the Presence of Obstacles", Proceeding of the 1993 IEEE International Conference of Robotics and Automation, pp 826-831

10) J. Borenstein and Y. Koren, "Obstacle Avoidance with Ultrasonic Sensors", IEEE Journal of Robotics and Automation Volume 4, No. 2, April 1988, pp 213-218

11) J. Borenstein and Y. Koren, "Real-Time Obstacle Avoidance for Fast Mobile Robots", IEEE Transactions on Systems, Man, and Cybernetics, Vol. 19, No. 5, Sept. to Oct. 1989 pp 1179-1187

12) D. Green, J. Sasiadek, D. Necsulescu and B. Kim, "Dead Reckoning for Mobile Robots via Sensor Fusion", Proceedings of the Knowledge Based Systems and Robotics Workshop 1993, pp 183-187

13) S. Rouseau, "Design of a Four Wheel and Two Body Mars Rover with Impedance Control", End of Study Report for: Diplome D'Ingenieur de Sup'Areo and Diplome D'études Approfondies en Automatique, Internal Paper,, University of Ottawa, 1993

14) C. Samson, "Path Following and Time-Varying Feedback Stabilization of a Wheeled Mobile Robot", RO-13.1.1 to 13.1.5

15) C. Canudas de Wit and R. Roskam, "Path Following of a 2-DOF Wheeled Mobile Robot under Path and Input Torque Constraints", Proceeding of the 1991 IEEE International Conference on Robotics and Automation, pp 1142 to 1147

16) L. Feng, Y. Koren and J. Borenstein, "Cross-Coupling Motion Controller for Mobile Robots", IEEE Dec 1993 0272-1708/93

17) B. d'Andréa_Novel, G. Bastin, and G. Campion, "Dynamic Feedback Linearization of Nonholonomic Wheeled Mobile Robots", Proceeding of the 1992 IEEE International Conference on Robotics and Automation, pp 2527 to 2532

18) Z. Deng and M. Brady, "Dynamic Tracking of a Wheeled Mobile Robot", Proceedings of the IEEE/RSI International Conference on Intelligent Robots and Systems, Yokohama, Japan, July 26-30, 1993, pp 1295-1298

19) N. Sarkar, X.Yun and V. Kumar, "Control of Mechanical Systems With Rolling Constraints: Application to Dynamic Control of Mobile Robots", The International Journal of Robotics Research, Vol 13, No. 1, Feb 1994, pages 55-69

20) B. Kim,"Free Motion and Obstacle Avoidance Control for Mobile Robots", Masters Thesis, University of Ottawa, 1992

21) D. Necsulescu, B. Kim and S. Kalaycioglu, "Dynamics Control of an Autonomous Wheeled Ground Vehicle", Trans. CSME, Vol 17, No. 4B, 1993, pp 735-758

22) D. Necsulescu, M. Eghtesad and S. Kalaycloglu, "Dynamic Based Linearization and Control of an Autonomous Mobile Robot", Paper presented to Second Biennial European Joint Conference on Engineering Systems Design and Analysis, University of London, July 14-17 1994

23) D. Necsulescu, V. Lonmo, B. Kim and G. Vukovich, "Autonmous Mobile Robot Control in Operational Space with Torque Saturation, Slippage and Tip-over Avoidance", IECON November 1995, Orlando, Florida

24) V. Lonmo, D. Necsulescu and G. Vukovich, "Experimental Verification of Operational Space Control Strategies for Autonomous Mobile Robots", Canadian Conference on Electrical and Computer Engineering, Sept. 1995

25) D. Clarke, "Application of Generalized Predictive Control to Industrial Processes", IEEE Control Systems Magazine, 1988, pp 49-55

26) C. Garcia, D. Prett and M. Morari, "Model Predictive Control: Theory and Practice- a Survey", Automatica vol 25, No.3, 1989, pp 335-348

27) L. Del Re, J. Chapuis, and Nevistic, "Predictive Control with Embedded Feedback Linearization for Bilinear Plants with Input Constraints", Proceedings of CDC, 1993, pp. 2984-2988

28) E. Droguet, "Autonomous Robot Control with Constraint Verification", paper presented to the University of Ottawa, August 1994, Internal Paper

APPENDIX A:
Using the Simulation

There are three simulation programs used in this thesis. They are fundamentally the same, and differ only in the type of control strategy used. They are:

PDTHESIS.EXE	: Conventional kinematics based controller
THESIS_1.EXE	: Controller with first order model based prediction
THESIS_0.EXE	: Controller with zero order model based prediction

These programs are all written in C with only the function MAIN() varying between them. They all require an EGA or VGA and a PC-AT equivalent or better with a math coprocessor. A 486DX with VGA is ideal. When any of these programs are run an option menu appears (see Figure A1). The option menu allows the user to modify the simulation without recompiling the code. The following keys are used to make changes:

```
Simulation of a Tricycle Configuration Vehicle  by: Victor Lonmo Jan 95

     The following options have been selected:
         following a Sinusoidal Trajectory
         filter cut-off freq: 20 Hz
         graphic output
         using predictive control to avoid torque saturation
         File: Outputpc.dat will be produced

         Driving Torque saturation limit:  0.300
        Steering Torque saturation limit:  1.500
                          Safety Factor:  0.950

       Press s to toggle saturation
             c to change the trajectory
             p to toggle predictive control
             g to toggle graphics
             d to modify driving torque
             t to modify steering torque
             f to modify the safety factor
             l to toggle file saving

       any other key will start the simulation
```

Figure A1: The Simulation Options Menu

Different filters can be chosen by pressing the number keys. Pressing "0" corresponds to a no filtering case with perfect reading of the variables and their derivatives. If text output mode has been chosen then the program will periodically stop, allowing the user to observe the input/output data. The options menu is updated as changes are made.

In graphics mode the simulation will immediately switch to the graphics screen which shows the options chosen, two torque output bars and a grid of the NQ plane. As the simulation progresses the desired trajectory and the actual path are shown. A robot is also shown to indicate the current value of steering angle orientation and driving wheel speed (see Figure A2). When the simulation is complete, the program will wait for the user to press a key before going to the end screen.

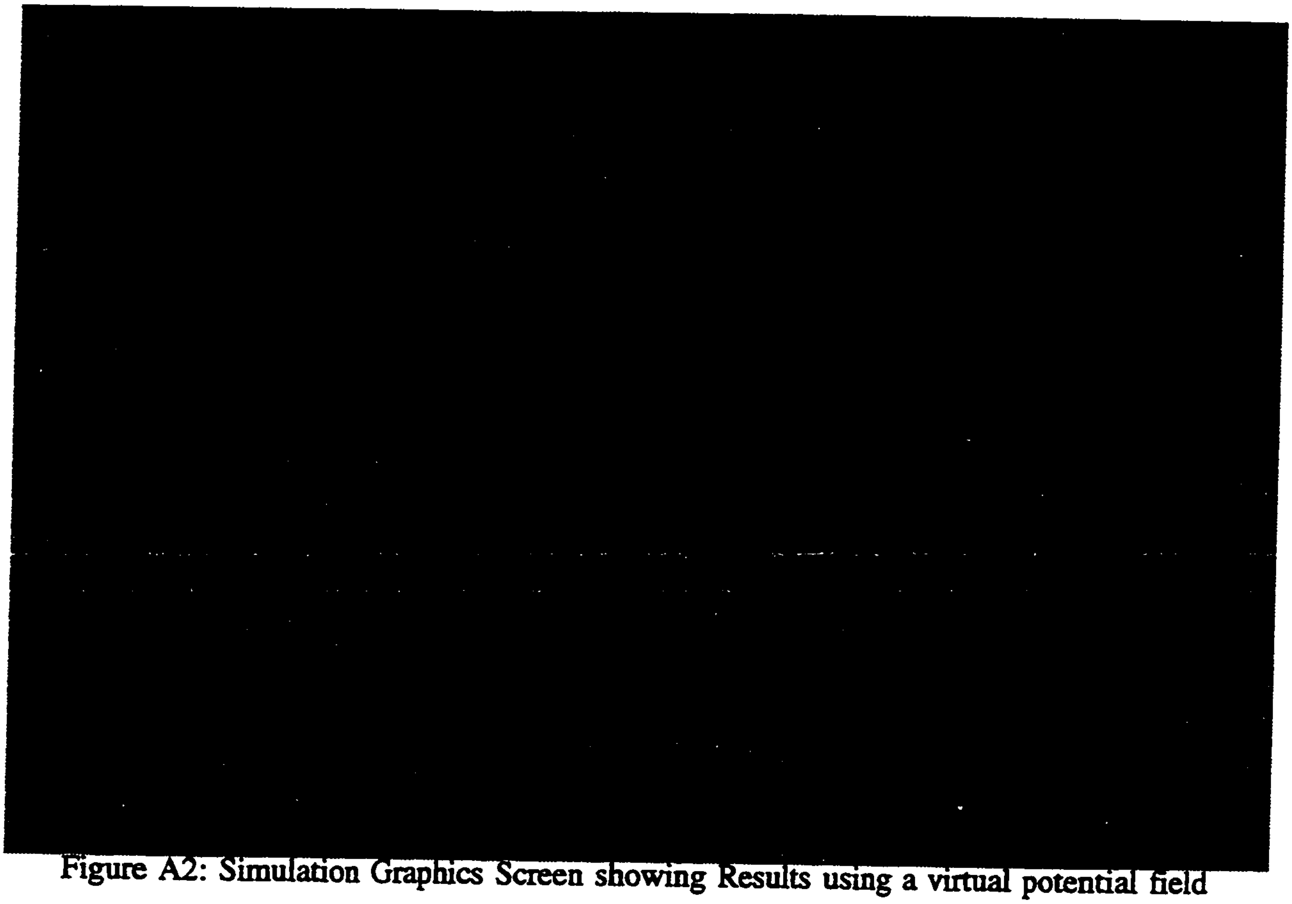

Figure A2: Simulation Graphics Screen showing Results using a virtual potential field

The end screen (see Figure A3) displays the maximum and minimum values for both torques, steering angle and orientation angle. The number of times that the torque was saturated follows. It shows the final desired and actual positions of the vehicle's centre of mass. The value of the root mean square of position error is indicated at the end. The root mean square error is defined to be the error in position (NQ) which is squared and summed to the previous error. Pressing any key at this point returns the user to the dos prompt.

```
finished at time = 20.000637
Td_Max      =     0.30000 at time = 19.1361
Td_Min      =    -0.30000           19.9996
Ts_Max      =     1.50000           19.0452
Ts_Min      =    -1.50000           19.9996
Theta_Max =     1.55668           12.2613
Theta_Min =    -1.67419           18.7034
Delta_Max =     3.78131           19.9996
Delta_Min =    -2.64119           13.7269

Predictive Control used for 0 loops

Torque Saturated for 14282 loops
Taud   Saturated for 14209 loops
Taus   Saturated for 10967 loops

Final Position   .N = 2.525716 .Q = 1.395322
Desired Position .N = 3.999928 .Q = 0.825282

This simulation used a PD controller
The RMS value of error in Position = 2.807109
press any key to exit
```

Figure A3: The Simulation Endscreen

If the user wishes to change any of the parameters of the platform itself, such as the mass of the frame, it will be necessary to recompile the code. These values are available in the file "THE_VAR.H". This header file is used by virtually all the function files therefore it will be necessary to recompile most of the program after modifying it. To rebuild the simulation, the following files are needed:

THE_MAI2.C (for first order predictive control)

THE_MAIN.C (for zero order predictive control)

THE_PDMA.C (for kinematic based control)

THE_PD.C and THE_PD.H (for kinematic based control only)

The other files are common to all three simulations:

THE_VAR.H (robot parameters and structure definitions)

THE_PROT.H (prototypes for all functions)

THE_MATH.C (model based control and integration functions)

THE_CONT.C (predetermined trajectories and controlling equations)

THE_OBST.C (potential field algorithm and related functions)

THE_GRA.C (graphics functions)

THE_IO.C (text screens and file output routines)

THE_TORQ.C (saturation functions for torque)

THE_F2CH.C (function to convert floating point numbers to character strings for graphics output, written by Jean de Carufel)

IIR_FIL.C and IIR_FIL.H (filters and derivatives, written by Jean de Carufel)

EGAVGA.OBJ (file needed for graphic screens)

The program also requires use of the graphics.h header file which is only common to Borland Dos compilers.

APPENDIX B:
Using the CLAMOR Platform

The CLAMOR platform and the simulation share most of their code. Since graphics and input/output functions are generally not available on the dSPACE computer system, those files which rely on STDIO.H, CONIO.H and GRAPHICS.H require modification for use on the dSPACE. Since THE_PROT.H includes a prototype which uses the FILE structure from STDIO.H, this file had to be modified. Most of the modifications were minor however. The new files are basically the same as the simulation files (although the simulation files have been modified since the conversion in some cases). The dSPACE files are:

DSP_MATH.C (dynamic model functions)
DSP_TORQ.C (torque saturation functions)
DSP_CONT.C (control equations and trajectories)
DSP_T7X.C (main and related functions for model based control)
PIT_MOTO.C, PIT_MOTO.H (functions to output torques)
VECTFUNC.C, VECTFUNC.H (used in definition of vectors and matrices)
FILTER.C (used for filters and derivatives)

To run the program on the system, the following steps must be taken:

1) Reset the PC host to DSP mode and reboot.
2) Compile the code (if necessary)
3) Link the code (if necessary)
4) Turn on the dSPACE.
5) Run the TRACE30 program on the PC host and verify that the correct TRC file is mapped.
6) Quit TRACE30 and run MON30

7) Load the object file. (DSP_ROBO for dynamic based control)
(DSP_PITT for kinematic based control)

8) Run the object file.

9) Quit MON30 (without disabling it).

10) Run TRACE30 again and begin tracing.

11) Turn on the robot. There are 3 switches that must be turned on. The first is the voltage source which is used to power the buffers. The second is a small switch at the back of the CLAMOR which controls the cooling fans. The last is the main power switch connected to the batteries. The program is designed to allow the user fifteen seconds from the time that the program is initiated to the time that the robot begins moving.

12) When the program has finished, turn off the main power switch on the CLAMOR. If the program will be run again, then leave the rest on.

It is strongly recommended that the user hold the main power switch at all times during operation of the robot. The robot's odometry will cause it to deviate from a desired path. It should only be used without the switch if the system has been previously tested and the user is absolutely certain that the robot will not hit anything.

APPENDIX C:
Kinematic and Dynamic Relations

The following are the kinematic and dynamic equations of a tricycle configuration mobile robot which is described in Chapter 3. The derivation of these equations is given in [20, 21, 23]. The kinematic equations of motion of the robot are:

$$\begin{aligned} \dot{N} &= r_1 \omega_1 (\cos\delta\cos\theta - \frac{c}{b}\sin\delta\sin\theta) \\ \dot{Q} &= r_1 \omega_1 (\cos\delta\sin\theta + \frac{c}{b}\sin\delta\cos\theta) \end{aligned} \tag{C.1}$$

$$\omega_\theta = \frac{1}{b} r_1 \omega_1 \sin\delta \tag{C.2}$$

$$\gamma_1 = \dot{\omega}_\theta \tag{C.3}$$

$$\omega_2 = \frac{1}{r_2}(\cos\delta - \frac{l}{2b}\sin\delta)\ r_1 \omega_1 \tag{C.4}$$

$$\omega_3 = \frac{1}{r_3}(\cos\delta + \frac{l}{2b}\sin\delta)\ r_1 \omega_1 \tag{C.5}$$

The dynamic equations of the robot are:

$$\dot{\omega}_1 = \frac{\tau_1 - \frac{r_1}{b}\sin(\delta)\tau_S - Num(\delta)\omega_1\omega_\delta}{Den(\delta)} \tag{C.6}$$

$$\dot{\omega}_\delta = \frac{\tau_S}{J_1 + J_{SA}} - \gamma_1 \tag{C.7}$$

Where Num(δ) and Den(δ) are defined as:

$$Q1 = \frac{I_{Y2}}{r_2^2} + \frac{I_{Y3}}{r_3^2} \tag{C.8}$$

$$Q2 = \frac{I_{Y2}}{r_2^2} - \frac{I_{Y3}}{r_3^2} \tag{C.9}$$

$$Q3 = (m_2 + m_3)c - (m_1 + m_{SA})(b - c) \tag{C.10}$$

$$Q4 = J_{FRAME} + J_2 + J_3 + (b - c)^2(m_{SA} + m_1) + (c^2 + \frac{l^2}{4})(m_2 + m_3) \tag{C.11}$$

$$Q5 = -Q3 + \frac{Q4}{c} \tag{C.12}$$

$$Q6 = Q1 + m_{TOTAL} \tag{C.13}$$

$$Q7 = Q2 + m_2 - m_3 \tag{C.14}$$

$$Q8 = \frac{l^2}{4c}Q1 + cm_{TOTAL} - Q3 + Q5 \tag{C.15}$$

$$Q9 = -cQ7 - \frac{l^2}{4c}Q1 - Q5 \tag{C.16}$$

$$Num(\delta) = r_1^2\left(\frac{l}{b} - Q7\cos(2\delta) + \frac{Q9}{2}\sin(2\delta)\right) \tag{C.17}$$

$$Num'(\delta) = \frac{d}{dt}Num(\delta) \tag{C.18}$$

$$Den(\delta) = I_{Y1} + r_1^2\left(Q6\cos^2(\delta) + \frac{c}{b^2}Q8\sin^2(\delta) - \frac{l}{2b}Q7\sin(2\delta)\right) \tag{C.19}$$

$$Den'(\delta) = \frac{d}{dt}Den(\delta) \tag{C.20}$$

Appendix D:

Development of the Equations of Jerk for the Robot

Equation (3.4) can be differentiated to produce:

$$\begin{aligned}
J_N = {} & r1A1\ddot{\omega}_1 + r1\dot{\omega}_1(-A2\omega_\theta - A3\omega_\delta) \\
& - r1A2\omega_\theta\dot{\omega}_1 - r1A2\dot{\omega}_\theta \\
& - r1\omega_\theta\omega_1(A1\omega_\theta - A4\omega_\delta) - r1A3\omega_\delta\dot{\omega}_1 \\
& - r1A3\dot{\omega}_\delta\omega_1 - r1(-A4\omega_\theta + A1\omega_\delta)\omega_\delta\omega_1 \\
J_Q = {} & r1A2\ddot{\omega}_1 + r1\dot{\omega}_1(A1\omega_\theta - A4\omega_\delta) \\
& + r1A1\omega_1\dot{\omega}_\theta + r1\omega_1\omega_\theta(-A2\omega_\theta - A3\omega_\delta) \\
& - r1A4\omega_\delta\dot{\omega}_1 - r1A4\dot{\omega}_\delta\omega_1 \\
& - r1\omega_\delta\omega_1(A3\omega_\theta + A2\omega_\delta)
\end{aligned} \tag{D.1}$$

These equations are needed for the terms in the second derivative of ω_1 and the first derivative of ω_δ. These are the variables of interest because their coefficients will determine if the we can invert the transformations between the torques and inertial space.

Differentiating equation C.6 gives:

$$\begin{aligned}
\ddot{\omega}_1 = {} & \frac{d}{dt}\left(\tau_d - \frac{r1\sin(\delta)}{b}\tau_S\right) \\
& + \frac{-Num(\delta)\omega_1\dot{\omega}_\delta - Num(\delta)\dot{\omega}_1\omega_\delta - Num'(\delta)\omega_1\omega_\delta}{Den(\delta)} \\
& - \frac{Den'(\delta)}{Den^2(\delta)}\left(\tau_d - \frac{r1\sin(\delta)}{b}\tau_S - Num(\delta)\omega_1\omega_\delta\right) \\
& - \omega_1(\omega_\theta + \omega_\delta)^2
\end{aligned} \tag{D.2}$$

The value of W_1 defined as:

$$W_1 = \frac{d}{dt}\left(\tau_d - \frac{r1\sin(\delta)}{b}\tau_S\right) \tag{D.3}$$

using this new variable, as shown in equation (C.7),

$$\dot{\omega}_\delta = \frac{\tau_S}{J_1 + J_{SA}} - \dot{\omega}_\theta \tag{D.4}$$

$$\ddot{\omega}_1 = \frac{W_1}{Den(\delta)} - \frac{Num(\delta)\omega_1\dot{\omega}_\delta + Num(\delta)\dot{\omega}_1\omega_\delta + Num'(\delta)\omega_1\omega_\delta}{Den(\delta)} - \frac{Den'(\delta)}{Den(\delta)}\dot{\omega}_1 - \omega_1(\omega_\theta + \omega_\delta)^2 \tag{D.5}$$

The value of W2 is defined as:

$$W2 = \tau_s \tag{D.6}$$

Our equation for the second derivative of ω_1 becomes:

$$\ddot{\omega}_1 = \frac{1}{Den(\delta)} W1 - \frac{Num(\delta)}{Den(\delta)}\frac{\omega_1}{J_T} W2 + \frac{-Num(\delta)\dot{\omega}_1\omega_\delta - Num'(\delta)\omega_1\omega_\delta}{Den(\delta)} + \frac{Num(\delta)\omega_1\dot{\omega}_\theta}{Den(\delta)} - \frac{Den'(\delta)\dot{\omega}_1}{Den(\delta)} - \omega_1(\omega_\theta + \omega_\delta)^2 \tag{D.7}$$

The constant C2 is defined as,

$$C2 = \frac{Num(\delta)\dot{\omega}_\theta\omega_1}{Den(\delta)} - \left(\frac{Num(\delta)\dot{\omega}_1\omega_\delta + Num'(\delta)\omega_1\omega_\delta}{Den(\delta)}\right) - \frac{Den'(\delta)\omega_1}{Den(\delta)} - \omega_1(\omega_\theta + \omega_\delta)^2 \tag{D.8}$$

This equation and the definitions of W1 and W2 are now used in the equation of J_N,

$$J_N = r1A1\left[\; W1\frac{1}{Den(\delta)} + W2\left(-\frac{Num(\delta)\omega_1}{Den(\delta)J_T}\right) + C2\right]$$
$$+r1\dot{\omega}_1(-A2\omega_\theta - A3\omega_\delta) - r1A2\omega_\theta\dot{\omega}_1$$
$$-r1A2\dot{\omega}_\theta\omega_1 - r1\omega_1\omega_\theta(A1\omega_\theta - A4\omega_\delta) \tag{D.9}$$
$$-r1A3\dot{\omega}_1\omega_\delta - r1A3\omega_1\left(\frac{W2}{J_T} - \dot{\omega}_\theta\right)$$
$$-r1\omega_1\omega_\delta(-A4\omega_\theta + A1\omega_1)$$

The terms of W1 and W2 are of particular interest, so they will be isolated.

$$J_N = \left[\frac{r1A1}{Den(\delta)}\right] W1 + \left[-\frac{r1A1\omega_1 Num(\delta)}{J_T\,Den(\delta)} - \frac{r1A3\omega_1}{J_T}\right] W2$$
$$+r1A1C2 + 2r1\dot{\omega}_1(-A2\omega_\theta - A3\omega_\delta) - r1A2\omega_1\dot{\omega}_\theta \tag{D.10}$$
$$-r1\omega_1\omega_\theta(A1\omega_\theta - A4\omega_\delta) + r1A3\omega_1\dot{\omega}_\theta$$
$$-r1\omega_1\omega_\delta(-A4\omega_\theta + A1\omega_\delta)$$

This is the form of the equation used in equation 3.5 for J_N. The terms which do not include W1 or W2 are grouped together and referred to as Δ_{ON} as per equation 3.6.

Using a similar approach in Q direction, we obtain,

$$\begin{aligned}
J_Q = r1A2\left(\frac{W1}{Den(\delta)} + W2\left[-\frac{Num(\delta)\ \omega_1}{Den(\delta)\ J_T}\right]\right) \\
+r1A2C2 + r1\dot{\omega}_1(A1\omega_\theta - A4\omega_\delta) \\
+r1A1\omega_1\dot{\omega}_\theta + r1A1\dot{\omega}_1\omega_\theta \\
+r1\omega_1\omega_\theta(-A2\omega_\theta - A3\omega_\delta) \\
+r1A4\dot{\omega}_1\omega_\delta - r1A4\left(\frac{W2}{J_T} - \dot{\omega}_\theta\right) \\
-r1\omega_\delta\omega_1(A3\omega_\theta + A2\omega_\delta)
\end{aligned} \tag{D.11}$$

which results in the relation used for Qj in the text:

$$\begin{aligned}
J_Q = W1\left[\frac{r1A2}{Den(\delta)}\right] + W2\left[-\frac{r1\omega_1 A2Num(\delta)}{J_T\ Den(\delta)} - \frac{r1\omega_1 A4}{J_T}\right] \\
+r1A2C2 + 2r1\dot{\omega}_1(A1\omega_\theta - A4\omega_\delta) \\
+r1A1\omega_1\dot{\omega}_\theta + r1\omega_1\omega_\theta(-A2\omega_\theta - A3\omega_\delta) \\
+r1A4\omega_1\dot{\omega}_\theta - r1\omega_1\omega_\delta(A3\omega_\theta + A2\omega_\delta)
\end{aligned} \tag{D.12}$$

Appendix E:
Additional Outputs from Simulations of the CLAMOR Platform

This appendix contains the outputs from the simulation using dynamics based zero order model based prediction with the sinusoid and square wave inputs as well as outputs from the kinematic controller for the same trajectories with stiff steering.

Figures E.1 and E.2 (zero order model based prediction) correspond to Figures 5.3 and 5.4 (first order model based prediction) respectively. The two sets of graphs are very similar due to the very short time step between the resetting of input jerks.

Figures E.3 and E.4 correspond to Figures 5.1 and 5.2. The simulation outputs for the sinusoid path shown in figure E.3 is the most accurate of all the simulations. Conversely, Figure E.4 (using the same controller with the same parameters as E.3) is the least accurate when torque saturation occurs.

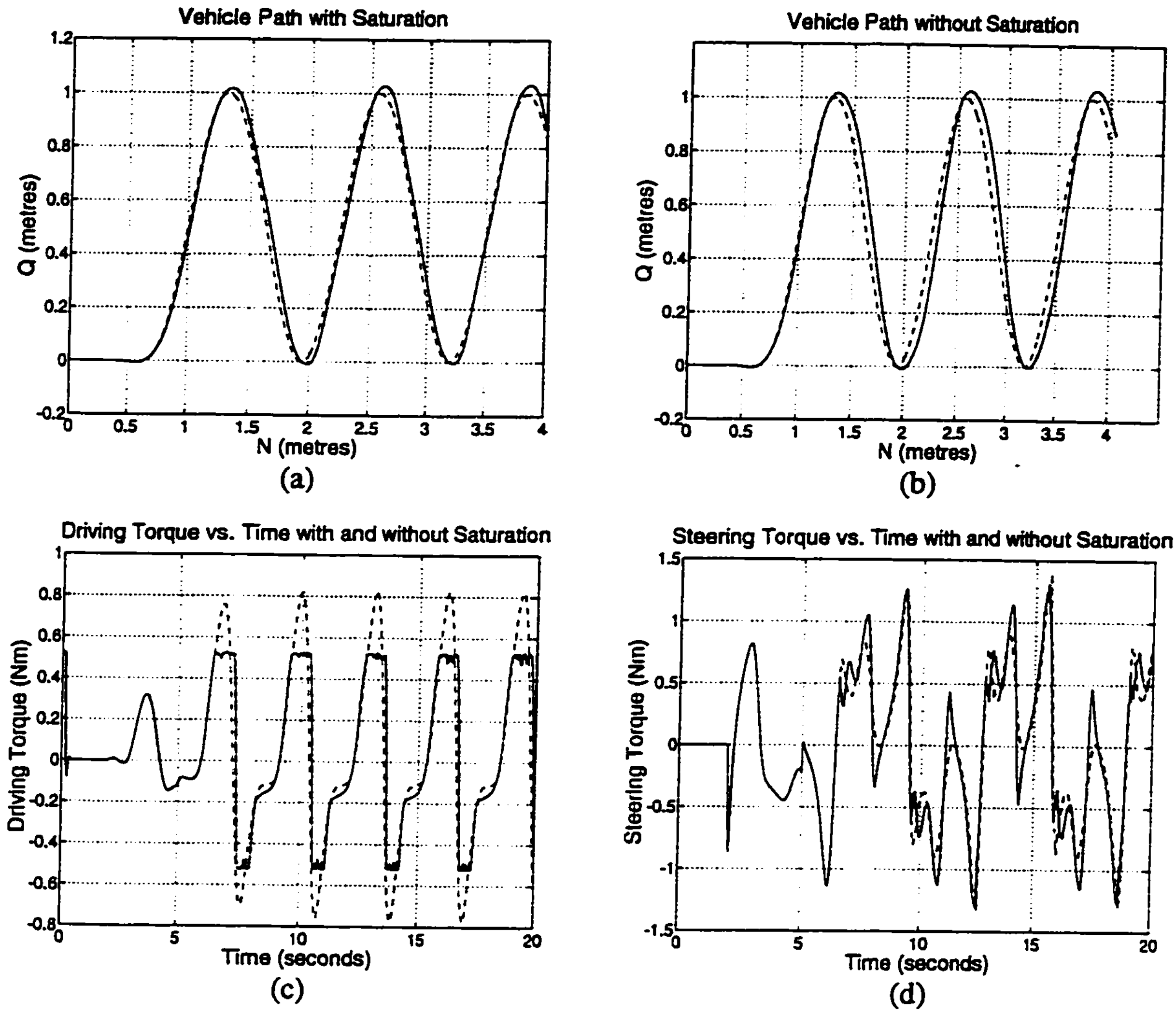

Figure E.1: Simulation Results using a Model Based Controller with Zero Order Prediction to track a Sinusoid Path

a) Path achieved with torque saturation (the dotted line is the desired path)

b) Path achieved without torque saturation (the dotted line is the desired path)

c) Driving Torque versus Time for both cases (solid lines show the saturated case)

d) Steering Torque versus Time for both cases (solid lines show the saturated case)

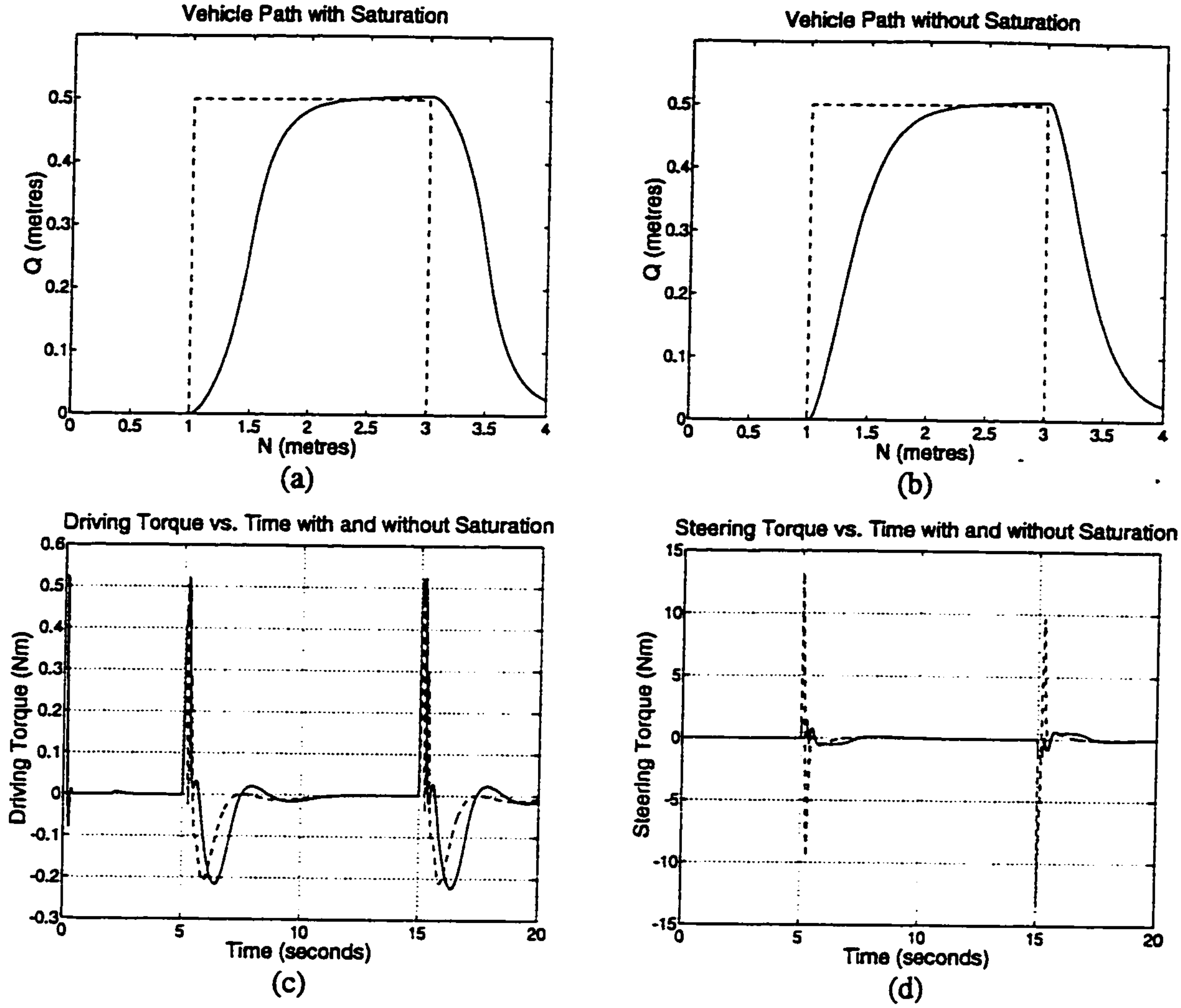

Figure E2: Simulation results using a Model Based Controller with Zero Order Prediction to track a Square Wave Path

a) Path achieved with torque saturation (the dotted line is the desired path)

b) Path achieved without torque saturation (the dotted line is the desired path)

c) Driving Torque versus Time for both cases (solid lines show saturated case)

d) Steering Torque versus Time for both cases (solid lines show saturated case)

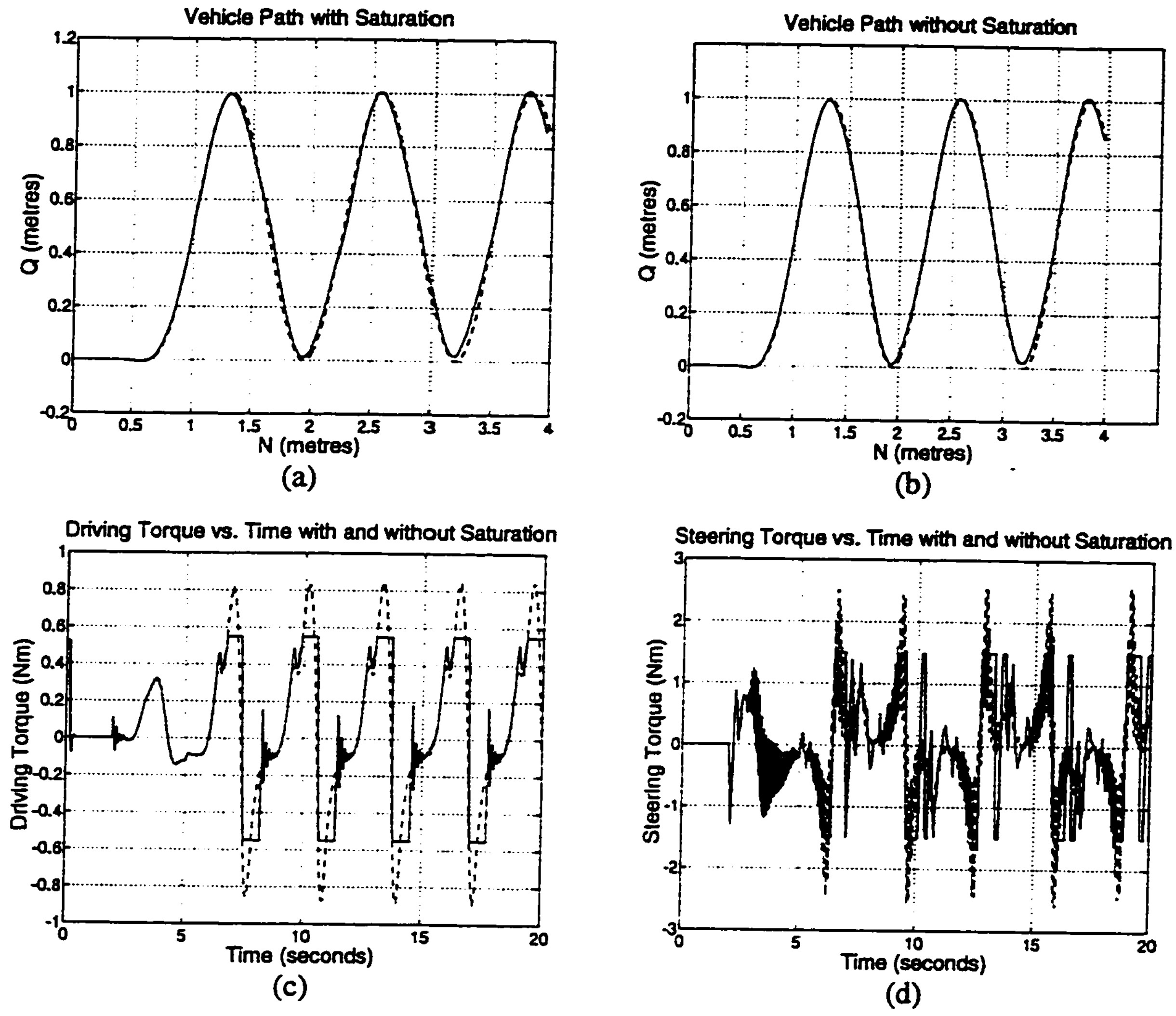

Figure E.3: Simulation results using a Kinematics Based Controller with stiff steering following a Sinusoid Path

a) Path achieved with torque saturation (the dotted line is the desired path)

b) Path achieved without torque saturation (the dotted line is the desired path)

c) Driving Torque versus Time for both cases (solid lines show the saturated case)

d) Steering Torque versus Time for both cases (solid lines show the saturated case)

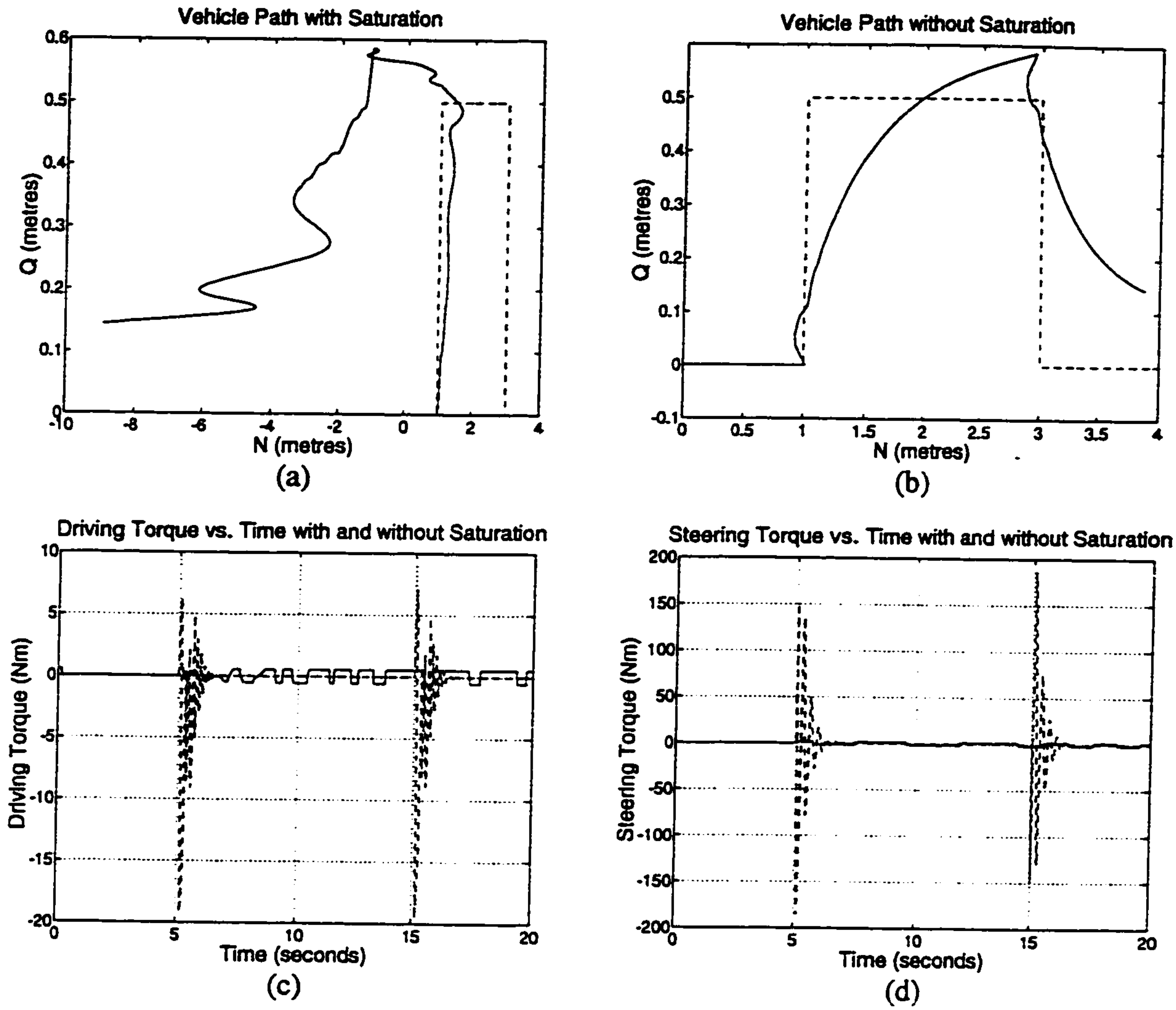

Figure E.4: Simulation results for a Kinematic Based Controller with stiff steering following a Square Wave Path

a) Path achieved with torque saturation (the dotted line is the desired path)

b) Path achieved without torque saturation (the dotted line is the desired path)

c) Driving Torque versus Time for both cases (solid lines show the saturated case)

d) Steering Torque versus Time for both cases (solid lines show the saturated case)

Appendix F:
Specifications of the Amplifiers, Gearheads and Motors

	Linear Servo Amplifier	
Ratings	Units	PA 228 Amplifier
Continuous Output Power	watts	160*
Continuous Output Current	amperes	7.5*
Peak Output Current	amperes	7.5*
Maximum internal dissipation (continuous)	watts	140*
Power Supply Voltage	+volts (dc)	19 - 32
	-volts (dc)	N.A.
Length	inches (cm)	4.7 (11.9)
Width	inches (cm)	2.8 (7.1)
Height	inches (cm)	0.72 (1.8)

*with heatsink @ 25 °C

Table F.1: Servo Amplifier Parameters

Current Amplifier Transfer Function:

$$\frac{i_o}{e_{input}} = 2 \ \frac{amperes}{volts} \tag{F.1}$$

Ratio	Gear Efficiency	Gearhead Mass (kg)
75.1:1	0.51	0.371

Table F.2: Gear Parameters

Motor Constant	Units	Model 14202
Peak Torque (stall)	Nm	0.75
Motor Constant	$mN{\cdot}m/(watts)^{0.5}$	41.1
Power for peak Torque	watts	333
Damping, zero source imped	mN·m/(rad/s)	1.75
No load speed	rad/s	400
Electrical Time Constant	ms	1.47
Mechanical Time Constant	ms	8.5
Friction Torque	N·m	14.1*10e-3
Armature Inertia	$kg{\cdot}m^2$	16.3*10e-5
Motor Mass	kg	0.74
Theoretical Acceleration	rad/s^2	46100
Thermal Time Constant	minutes	24.0
Ultimate temp. rise/watt	°C /watt	9.0
Maximum winding temp.	°C	155

Table F.3: Mechanical Characteristics of the Motors

Winding Constant	Units	Value for winding type #1
Voltage	volts	12.0
Current (stall)	amperes	27.8*10e-3
Torque Constant	N·m/ampere	27.5*10e-3
Terminal resistance	ohms	0.431
Back EMF	volts/(rad/s)	0.028
Inductance	mH	0.635
Current (No Load	amperes	0.460

Table F.4: Electrical Characteristics of the Motors

The overall gain of the dSPACE (voltage) to motor output (torque) is given by,

$$\frac{Torque}{Voltage\ input} = 2.107\ \frac{Nm}{volts} \tag{F.2}$$

which is determined by multiplying the amplifier transfer function by the motor torque constant, the gear ratio and the gear efficiency. Transient terms in the amplifiers were ignored.

Appendix G: Additional Experimental Results

This is a set of results obtained with zero order model based prediction and the CLAMOR platform. As with the simulations, there is very little difference between the first order model based prediction and the zero order model based prediction.

Figures G.1 and G.2 correspond to Figures 7.5 and 7.6 respectively. As with the simulations, due to a short time period the differences between the first order and zero order controllers are minimal.

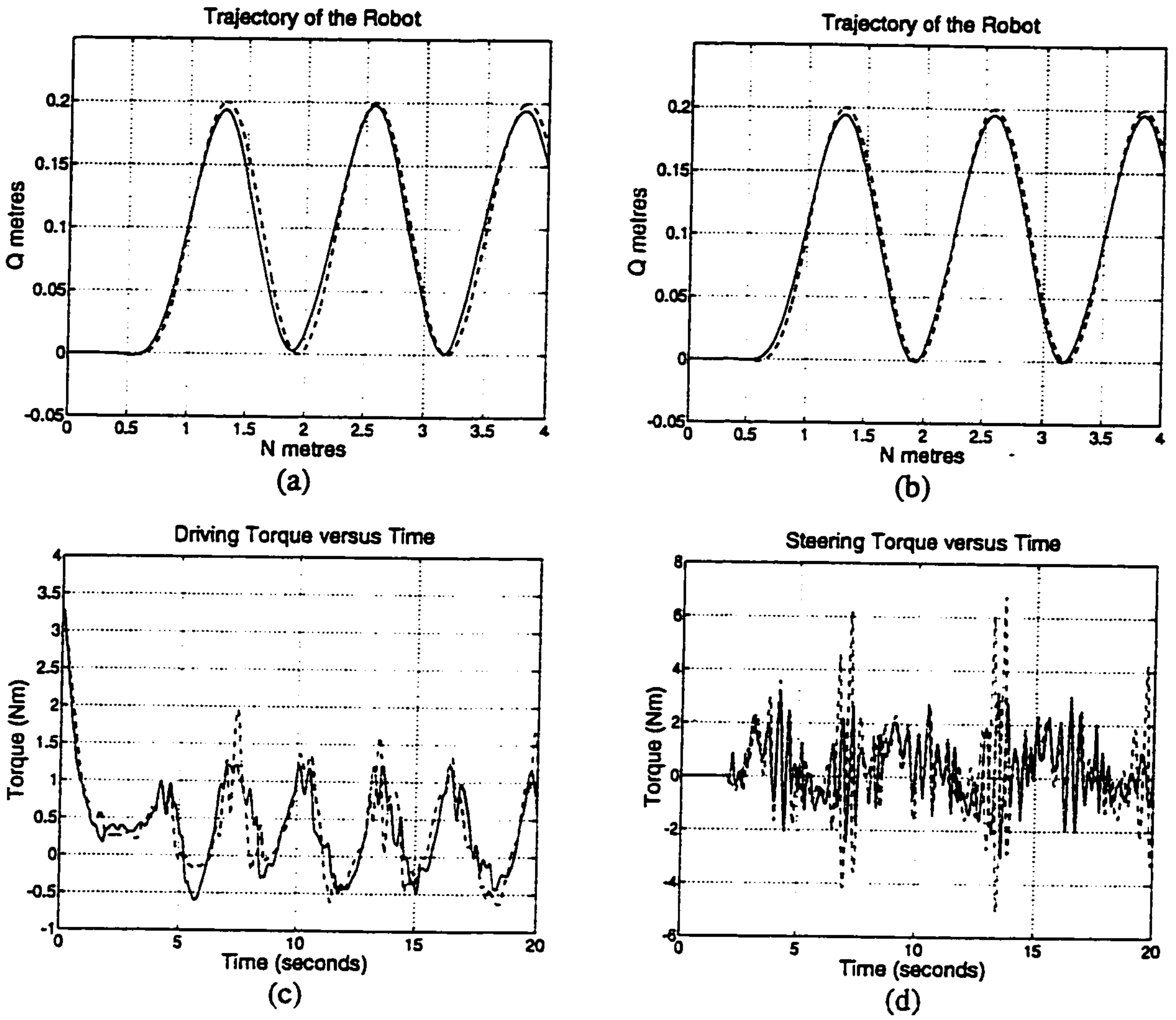

Figure G.1: Experimental Results using a Model Based Controller with Zero Order Prediction to track a Sinusoid Trajectory

a) Path with Torque Saturation (the dotted line is the desired path)

b) Path without Torque Saturation (the dotted line is the desired path)

c) Driving Torque versus Time (solid lines represent the torque saturation case)

d) Steering Torque versus Time (solid lines represent the torque saturation case)

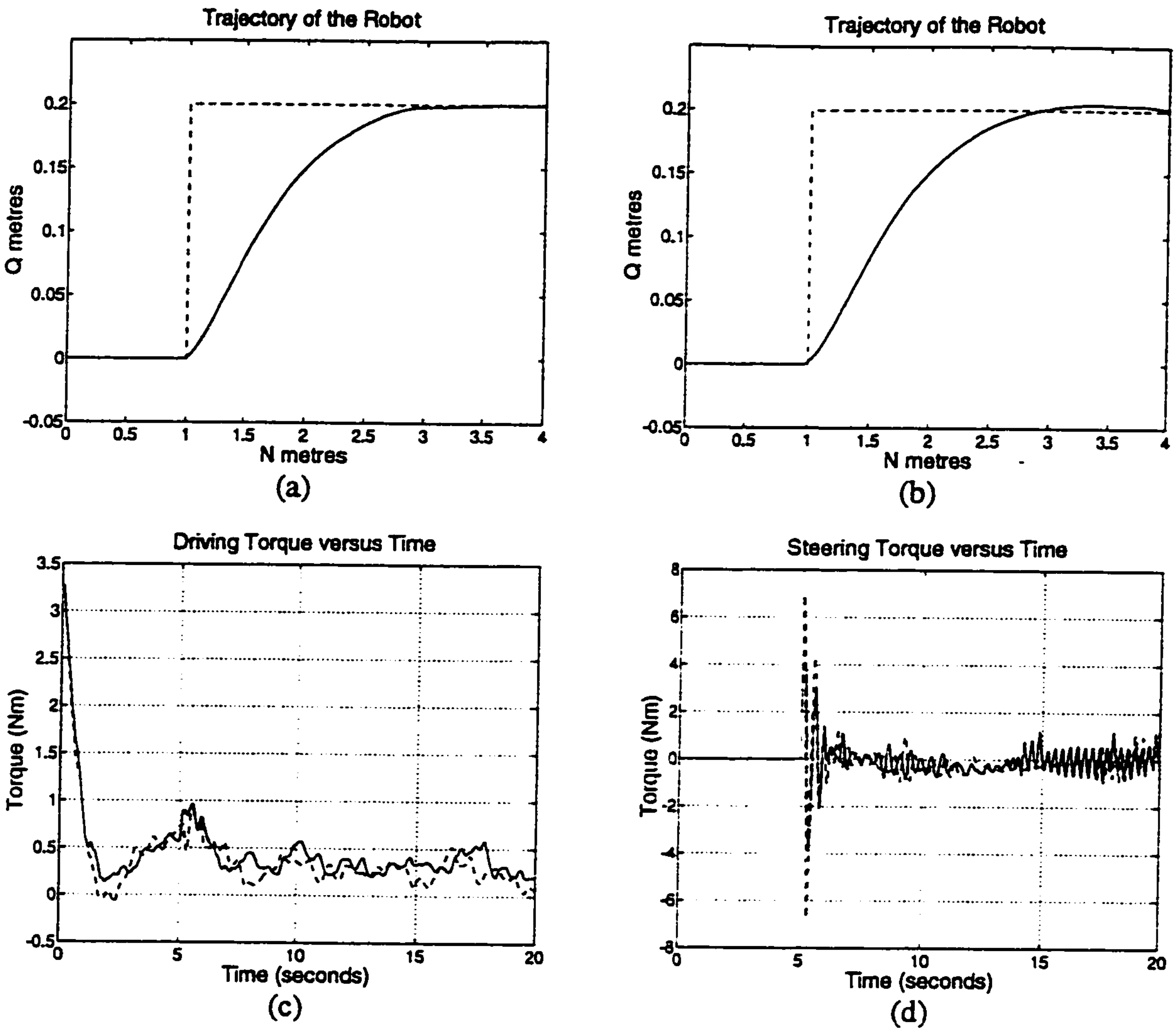

Figure G.2: Experimental Results using a Model Based Controller with Zero Order Prediction to track a Step Trajectory

a) Path with Torque Saturation (the dotted line is the desired path)

b) Path without Torque Saturation (the dotted line is the desired path)

c) Driving Torque versus Time (solid lines represent the torque saturation case)

d) Steering Torque versus Time (solid lines represent the torque saturation case)

Appendix H:
Dimensions and Properties of the CLAMOR Platform

This section provides the reader with a description of the CLAMOR platform and the dimensions which are needed in order to run the thesis program.

Dimension	Value (m)
Wheel 1: Radius (r_1)	0.0625
Wheel 2: Radius (r_2)	0.075
Wheel 3: Radius (r_3)	0.075
b	0.381
c	0.187
l	0.36

Table H.1: Dimensions of the CLAMOR used in the Experimental Program

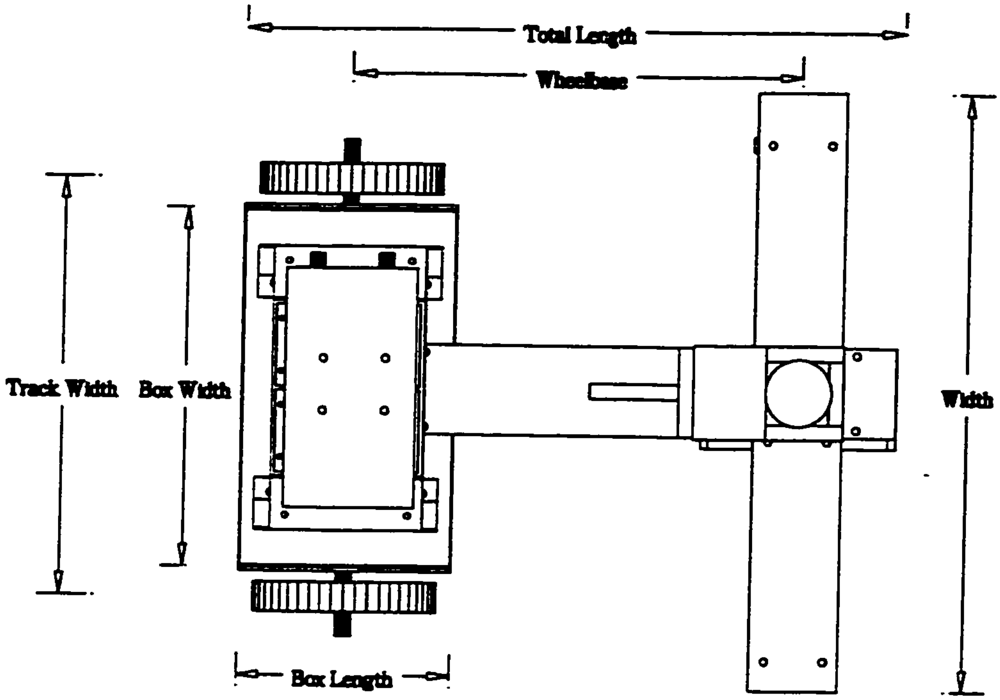

Figure H.1: Top View of CLAMOR

Dimension	Value m (inches)
Total Length	0.51 (20.)
Wheelbase (b)	0.38 (15.)
Width	0.51 (20)
Box Width	0.30 (12)
Track Width (l)	0.36 (14)

Table H.2: Dimensions of the CLAMOR

The centre of mass is located 0.187m ahead of the rear axle. Thus c = 0.187 for figure 2.2 and the relations given in appendix C.

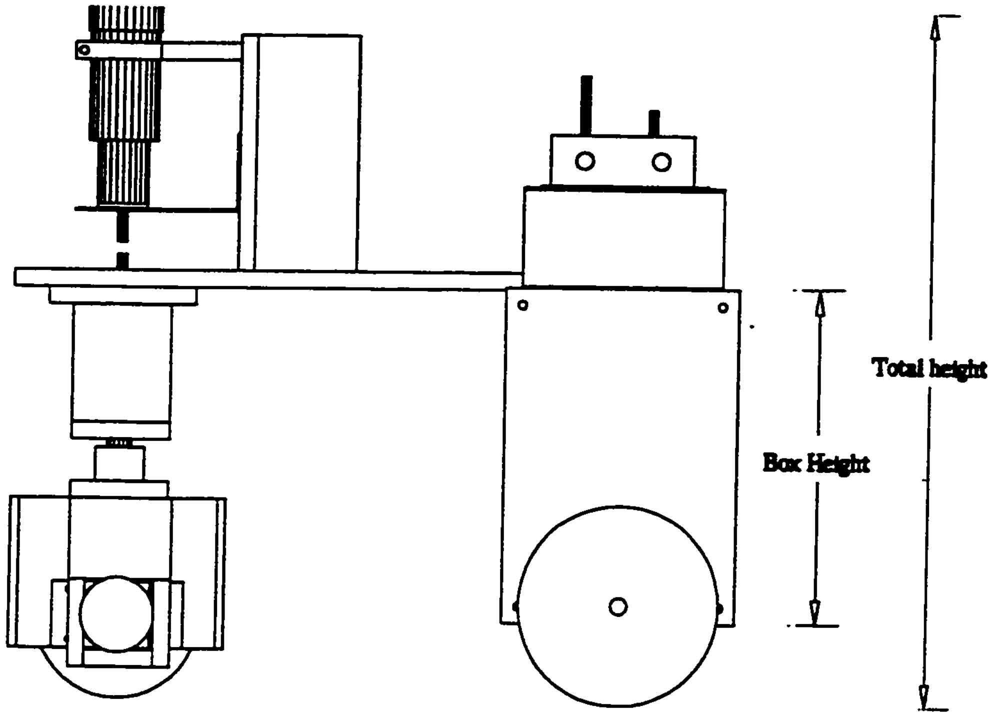

Figure H.2: Side View of CLAMOR

Dimension	Value m (inches)
Total Height	0.53 (21)
Box Height	0.25 (9.5)

Table H.3: Additional Dimensions of the CLAMOR

Masses of the components of CLAMOR	Mass (kg)
mass of the frame (including batteries)	18.0
mass of the steering assembly (m_{SA})	6.1
mass of wheel #1 (m_1)	0.1
mass of wheel #2 (m_2)	0.15
mass of wheel #3 (m_3)	0.15
Total Mass (m_{TOTAL})	24.5

Table H.4: Masses of the Components of CLAMOR

Rotational moments of inertia	Value (kg m^2)
Steering assembly (J_{SA})	0.354
Robot frame (with batteries) (J_{FRAME})	2.01
wheel #2 (J_2)	0.000844
wheel #3 (J_3)	0.000844
wheel #1 (I_{Y1})	0.000195
wheel #2 (I_{Y2})	0.000422
wheel #3 (I_{Y3})	0.000422

Table H.5: Rotational Moments of Inertia

Appendix I:
Calculation of time to send signal to Mars from Earth

The distance from Earth to Mars is given as 35 million miles to 248 million miles. The speed of light is 186 000 miles per second resulting in a minimum time to send a signal to Mar of three minutes and eight seconds. The maximum time to send the signal is just over twenty two minutes. It should also be noted that this time is only the length of time to send a signal, it does not include the time required to receive any feedback information.

The reference for the distance from Earth to Mars was taken from the World Book Encyclopedia Volume M, under the subject: "Mars", copyright 1973, U.S.A. Field Enterprises Educational Corporation.

IMAGE EVALUATION
TEST TARGET (QA-3)

150mm

6"

APPLIED IMAGE . Inc
1653 East Main Street
Rochester, NY 14609 USA
Phone: 716/482-0300
Fax: 716/288-5989