FREE MOTION AND OBSTACLE AVOIDANCE CONTROL FOR MOBILE ROBOTS

BY

BUMSOO KIM

A thesis 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, August 1992

ISBN 0-315-93587-1

UNIVERSITÉ D'OTTAWA
UNIVERSITY OF OTTAWA

TO MY WIFE AND SON

YEONSOON & JINSEO

ABSTRACT

Mobile robot motion control has been extensively studied based on a simplified mobile robot and actuator dynamic model. Free motion and obstacle avoidance require suitable robot and actuator models and the integration of sensor information into the system controller. In the present thesis, artificial impedance approach is used for developing the controller of an autonomous mobile robot. The control law is based on a Newtonian dynamics model of the mobile robot. Range sensor information is used in the trajectory correction for avoiding collision with obstacles. Simulation results illustrate the performance of the impedance based controller.

ACKNOWLEDGEMENT

I wish to acknowledge my appreciation to all those who helped me with this research. I am particularly grateful to Prof. Necsulescu whose participation in the present research was both valuable and perceptive. He showed that he was very dependable, thorough and responsive. I am also grateful to my colleagues in the control lab who always gave good advice and encouraged me to keep working on this research. I would like to give special thanks to Mr. Jassemi and Mr. De Carufel.

I am also very grateful to my parents, my wife Yeonsoon, and my son Jinseo for their patience and continuing support throughout my education.

TABLE OF CONTENT

LIST OF FIGURE

NOMENCLATURE

A_T, A_Q	Components of acceleration with respect to global reference frame
A_{RT}, A_{RQ}	Control command for collision avoidance
a_x, a_y	Components of acceleration with respect to local reference frame
B	Damping constant for artificial impedance
B_r	Damping constant for virtual repulsive force
D	Desired position of the mobile robot
$\bar{F}$	Reaction forces of the mobile robot
$\bar{F}_a$	Virtual attractive force
$\bar{F}_{ext}$	External forces exerted to the mobile robot
$\bar{F}_r$	Virtual repulsive force
$\bar{G}$	Frictional forces between the wheels and ground
I	Initial position of the mobile robot
I_{cm}	Moment of inertia of the mobile robot with regard to the centre of mass of the mobile robot frame
I_{wi}	Moment of inertia of each wheel with regard to their axis of rotation (i=1,2,3)
$\bar{i}$, $\bar{j}$	Unit vectors in x-y moving reference frame
K	Spring constant for artificial impedance
K_r	Spring constant for virtual repulsive force
m	Mass of the mobile robot
m_{wi}	Mass of each wheel (i=1,2,3)
T, Q	Current position of the mobile robot with respect to global reference frame
$\bar{t}$, $\bar{q}$	Unit vectors in T-Q fixed reference frame

T_d, Q_d	Target position of the mobile robot with respect to global reference frame
T_0, Q_0	Initial position of the mobile robot with respect to global reference frame
γ	Orientation angle between the vehicle's moving reference frame and the fixed global reference frame
δ	Steering angle of the front wheel with respect to moving reference frame
θ	Orientation angle between the vehicle's fixed reference frame and the fixed global reference frame
τ_1	Actuator propulsion torque to the front wheel
ω_i	Angular velocities of the three wheels which have radii r_{wi} (i = 1,2,3), respectively

I INTRODUCTION

Since its appearance in the early 1960s, robotics has generally focused on the design and control of robot manipulators. To a great extent, this has been motivated by the needs of manufacturing industry. However, a robot manipulator can manipulate only objects that it can reach. As most industrial robots are fixed in place, their work space is confined by the maximum extension of their linkage. There are two approaches under consideration to overcome the problems caused by the limited reach of robot manipulators (McKerrow, P.J. [3]). One is the flexible manufacturing cell, where the robot is fixed in place and the machines that it services are placed around it. This approach works well if the machines are small enough to fit close to the robot, and if all machines are carrying the same task, or all are working on different stages of the same task, and no machine has to wait for the robot to serve it. The second approach is to make the mobile robot by some form of guidance system so that it can move from one machine to another. These kind of guided vehicles are generally called AGVs (Automated Guided Vehicles). The motion of these vehicles is restricted to certain predefined path, and once the guidance mechanism is installed, a new path can be achieved only by restructuring the guidance mechanism. Recently, the manufacturing industries have searched for means to make their manufacturing systems more flexible in order to meet expectations

such as a variety of products and fast delivery to the market. The rising cost of keeping large amounts of inventory, which is inevitable for conventional manufacturing systems, has stimulated the development of the just-in-time manufacturing concept. One of the most serious problems in this environment is the transportation of parts and sub-assemblies between the manufacturing cells. In this environment, traditional AGVs are often too cumbersome and inflexible, smaller and more flexible autonomous mobile robots are required.

The autonomous mobile robot can be defined as "the vehicle that is capable of intelligent motion and actions without requiring either a guide to follow or teleoperator control."(Wilfong, G.T. [47]). One of the advantages of the mobile robot is that it can be used to find the trajectory to the target position while avoiding obstacles in a partially unknown environment. There is a variety of potential applications of the mobile robots (industrial, military, scientific, household, and humanitarian) in which a mobile robot will operate in a large and unstructured environment, for example, delivering parts inside a large factory, cleaning an industrial waste site, maintaining a nuclear plant, inspecting and repairing underwater structures, assembling structures in outer space, fire fighting, cleaning windows, aiding handicapped, etc. It is believed that autonomous mobile robots will be commonplace in near future.

An important research topic for the mobile robots is the control of the mobile robot from a current position to a goal position while avoiding the collision with any obstacles in real time. In order to achieve this goal, the mobile robot must be capable of sensing its environment, interpreting the sensor information, updating its knowledge of its position and the environment's structure, followed by planning or generating a route from an initial to a goal position in the presence of known or unknown obstacles.

In this thesis, the impedance control is applied for the generation of the trajectory of the mobile robot instead of preplanning the trajectory. Impedance control is a Cartesian space control and is one of the new control techniques which can generate trajectories for both obstacle free and obstacle avoidance cases in real time. One of the advantages of impedance control for the mobile robot motion control is the fact that it enables a mobile robot to perform obstacle avoidance tasks without knowing the full geometry of the obstacles and of the environment.

In this thesis, a simplified three wheeled mobile robot with a driven and steered front wheel and two passive rear wheels is considered. In order to apply the impedance control, in chapter III a complete kinematic and dynamic model, for this particular three wheeled mobile robot, is developed and the non-holonomicity of the mobile robot is also examined. In chapter IV, the basic

concept of impedance control is explained and is employed for trajectory generation and obstacle avoidance. Chapter V shows the simulation results for the free motion and obstacle avoidance of a mobile robot using impedance control approach. Chapter VI presents conclusions regarding the utilization of impedance control for the real-time control of the mobile robot.

II LITERATURE SURVEY

2.1 Introductory Remark

Interest in mobile robots is growing rapidly because of the very broad range of their potential applications. The challenge is that these robots move intelligently so that they can perform various actions without human intervention. Research on mobile robots began in the late sixties with the pioneering work of Stanford Research Institute. Two versions of SHAKEY, an autonomous mobile robot, were built in 1968 and 1971. The main purpose of this project was "to study processes for the real-time control of a robot system that interacts with a complex environment" (Nilsson, N.J. [1]). Indeed, mobile robots were and still are a very convenient and powerful support for research on artificial intelligence oriented robotics. A second and quite different trend of research began around the same period. It was aimed at solving the problem of robot vehicle locomotion over a rough terrain. Part of this research focused on the design and the study of the kinematics and dynamics of multilegged robots (McGee, R.B. et al, [2]). However, the research in this field progressed slowly for various reasons, such as the lack of efficient on-board instrumentation (computer,sensors, etc.). Meanwhile the so-called industrial robots (i.e., robot manipulator) become the main body of a fast growing field of robotics.

The present renewal of interest in mobile robots started in the late seventies fostered by powerful on-board signal and data processing capacities offered by microprocessor technology. Today the scientific reasons for using mobile robots as a support for conceptual and experimental work in advanced robotics hold more than ever. Furthermore a number of real-world applications can now be realistically envisioned, some for the near future. These applications range from intervention robots operating in hostile or extremely dangerous environments to day-to-day machines in highly automated factories using flexible manufacturing systems (FMS) technology.

The literature review has been performed in order to serve the focus of this thesis which is on the autonomy feature of mobile robots based on the kinematics, dynamics modelling and motion control aspects of the autonomous mobile robots development.

2.2 Configurations of Wheeled Mobile Robots

Most mobile robots use either wheels, chains or legs to move around. While people and many animals walk on legs, most mobile machines roll on wheels. Wheels are simpler to control, pose fewer stability problems, use less energy per unit distance of motion, and move faster than legs (McKerrow, P.J. [3]). Since the mobile robot described in this thesis is a wheeled mobile robot, only the publications on wheeled mobile robots are reviewed. However,

wheels are only usable on relatively smooth, solid terrain; on soft ground they can slip and get bogged down. In order to scale rough terrain, wheels have to be larger than the obstacles they encounter (McKerrow, P.J. [3]).

The most familiar wheel layout for a vehicle is a four wheel configuration in which the wheels are placed at the corners of a rectangle. Often, the two rear wheels are used for driving and the front two for steering. Alternative arrangements include front wheel drive, four wheel drive, and four wheel steering allowing some limited sideways motion. Most four wheeled vehicles have limited manoeuvrability because they are unable to move sideways. Also, a wheel suspension system is required to ensure that the wheels are in contact with the ground at all times. When moving in a straight line, all wheels rotate by the same amount, when turning, the inside wheels rotate slower than the outside wheels to avoid skidding because the contact distance travelled by the inside wheels is shorter. For a mobile robot, to meet these requirements, we need good mechanical design and independent controlling of the speed of the drive wheels. Shiller, Z. and Y.R. Gwo [4], Sekiguchi, M. et al [5], and Crowley J.L. [6] are some of the major contributors using this four wheel configuration. The vehicle model used in Graettinger, T.J. and B.H. Krogh [7] has two wheels, rather than four, but retains approximately the same characteristics as the four-wheeled vehicle as it is mentioned by Ellis, J.R. [8] and Hatwal, H., and Mikulcik, E.C. [9].

One way to simplify the problems of four-wheeled vehicles is to replace the coupled steering wheels with one wheel and keep the two rear wheels driven. Still the two rear driving wheels must rotate at slightly different speeds for accurate control of turning. Three wheeled vehicles have the advantage that wheel-to-ground contact can be maintained on all wheels without a suspension system. The centre of a three wheeled vehicle is the centre of the triangle defined by the ground contact points of the three wheels. This type of wheel configuration can be found in (Steer, B. [10]), (Samson, C. [11]), and (Hemami, A. et al [12]). In other three-wheeled vehicles, two wheels are driven independently and the other is idle caster. In order to steer the vehicle, the wheels should be driven at different speeds. For the robot to follow straight line and curves accurately, motor speeds must be controlled precisely. (Giralt, G. et al [13]), (Canudas De Wit, C. and Sordalen, O.J. [14]), (Samson, C. [15]), (Saha, S.K. and Angeles, J. [16]), and (Sordalen, O.J. and Canudas De Wit, C. [17]). (Canudas De Wit, C. and Roskam, R. [18]) used the same wheel configuration as above but with the two driving wheels in front. Other variants of the three-wheeled vehicle configuration are found in (Nelson, W.L. [19]) and (Necsulescu, D.S. and Kim, B. [20]). In one, the single wheel is the drive wheel as well as the steering wheel, enabling the other wheels to idle. Combining drive and steering mechanisms in one wheel results in a more complex mechanical design. In this thesis, this type of three-wheel

configuration is used and the kinematics and Newtonian dynamics are analyzed.

Some mobile robots have three wheels controlled by a synchronous drive system (e.g., K2A and Denning mobile robots) (Cybermation [21]), (Denning mobile robots Inc. [22]). All wheels are used for driving and steering. However, the wheels are coupled with a belt drive or gears, so that they can be steered by the one motor (Holland, J.M. [23]). The body of the robot always maintains a fixed orientation to the external world, and a sensor platform above the body always points in the direction of motion. All wheels are driven by a single motor. (Borenstein, J. and Y. Koren [24], [25]) uses this type of robot vehicle (K2ARS) for obstacle avoidance control.

The mobile robots described up until now were non-omnidirectional mobile robots, which cannot move sideways. In order to minimize the floor space required to turn corners and to eliminate the control problems associated with steering, it is needed that the vehicle has to be omnidirectional and compact. Some omnidirectional vehicles use special wheels. One is Stanford wheel (Calisle, B. [26], Campion, G. et al [27]) which has the rollers perpendicular to the axis of the hub, and the other is Illanator wheel (Muir, P.F., and Newman, C.P. [28], Feng, D. et al [29], and Daniel, D.J. et al [30]) which has rollers 45 degrees to the axis of the hub. Another type of wheel has a hub which is

driven, and the rollers are idle. The Stanford wheel system uses three wheels, one at each corner of an equilateral triangle, aligned so that their axes intersect at the centre of the robot. This arrangement does not need a suspension system, but has less resistance to tipping than a four-wheel system and if one roller jams the robot is immobilized. Due to the small diameter of the rollers, they have difficulty traversing obstacles lying parallel to their axis of rotation. Each wheel has two modes of motion: a) rotation about the axis of the hub with the rollers remaining still, and b) translation in the direction of the hub axis with the rollers in contact with the floor spinning and the hub fixed. Motion in any other directions involves a combination of hub rotation and roller rotation. An Illanator wheel, as used on the Carnegie-Mellon robot Uranus, can rotate about hub with rollers still, or move at 45 degrees with the hub still and the roller in contact with the ground spinning. Left-handed and right-handed arrangements of the wheel are possible, where left or right is the direction wheel motion with only the rollers spinning. Uranus uses four wheels, two left-handed and two right handed, and requires a suspension system. The wheels are arranged so that the diagonal lines through wheel contact points intersect at the centre of the vehicle. With these wheels, the vehicle can still move forwards or backwards if a roller jams. Omnidirectional mobile robots are the most suitable for applying fully operational impedance control law because of their ability to achieve decoupled dynamics. However, it is important to evaluate the performance and limitations of an

impedance controller on a non-omnidirectional robot which is much less complicated than the omnidirectional one.

In this thesis, the feasibility, the performance and the limitations of an impedance controller for non-omnidirectional robots will be investigated.

2.3 Kinematics and Dynamics Model for Trajectory Generation

The feedback control of an autonomous robot vehicle can sometimes present subtle and surprising problems particularly due to non-holonomic constraints, i.e. diffenential constraints which are not integrable (Campion, G. et al [32]). The position or pose of a vehicle is represented for planar motion by three parameters (x, y, θ), two for orthogonal translation axes one for orientation. However, in contrast to robotic manipulators, which are holonomic systems, rolling robots are in general non-holonomic systems. The non-holonomic constraints in the wheels make kinematic and dynamic analysis more complicated than those of holonomic systems (Saha, S.K. and Angeles, J. [16]). For common tricycle and differential-drive vehicle configurations, there are only two degree of freedom for control, that is, steering angle and velocity or the independent velocity of the two wheels of a differential drive vehicle and these vehicles are non-holonomic (Graettinger, T.J. and Krogh, B.H. [7], Samson, C. [11], Hemami, A. et al [12], Canudas de Wit, C. [14], Saha, S.K. and Angeles, J. [16], Canudas de Wit, C.

and Koskam, R. [18], Necsulescu, D.S. and Kim, B. [20], Campion, G. et al [32], Alexander, J.C. and Maddocks, J.H. [33], Kanayama, Y. et al [34], Barraquand, J. and Latombe, J.C. [35], and Nakamura, Y. and Hukherjee, R. [36]). Necsulescu and Kim [20] examined the non-holonomic constraints on the tricycle model, and Saha and Angeles [16] used the concept of orthogonal complement of the matrix of non-holonomic constraints on the differential-drive vehicle configuration for the development of the dynamic equations of motion of the problem. The kinematics and dynamics modelling are essential for the autonomous operation of mobile robots. They are used for the purpose of trajectory planning, simulation, and control.

There is a variety of results in the literature concerning kinematics and dynamics based control for trajectory planning or generation of the mobile robots. A kinematic description of both a tricycle and "turtle" robot architecture for the trajectory planning of a mobile robot was developed for these two architectures and a Gaussian envelope method for modulating the steering angle was proposed by (Steer, B. [10]). A kinematics based piecewise continuous controller, which exponentially stabilizes the robot about the origin for two degrees of freedom mobile robot with non-holonomic constraints, is proposed in (Sordalen, O.J. and Canudas de Wit, C. [17]). However, the mobile robots in these contributions follow a preplanned trajectory. For an autonomous mobile robot given the initial and final goal

positions and orientation, it is necessary to generate a path linking the two points. Trajectory generation and obstacle avoidance using virtual force field method based only on kinematics model of the mobile robot is described in (Borenstein, J. and Koren, Y. [24]). Kinematics based controller for trajectory generation is also presented in (Alexander, J.C. and Maddocks, J.H. [33], and Kanayama, Y et al [34]). In order to move fast and accurate, mobile robots need not only a kinematics model but also a dynamics model. Using the kinematics and dynamics models, the goal is to generate a trajectory rather than follow a preplanned trajectory. There are some dynamic based control results presented in the literature. Kinematics and Newtonian dynamics model for the control of a simplified two wheeled vehicle is reported in (Graettinger, T.J. and Krogh, B.H. [7]). Saha, S.K. and Angeles, J. [16] also developed a kinematics and Newtonian dynamics model for a three wheeled vehicle. Shiller, Z and Gwo, Y.R. [4] did the same work as above for four wheeled vehicles. All these contributions were developed for following a preplanned trajectory, rather than generating one. Canudas de Wit, C. and Roskam, R. [18] utilized the Lagrangian dynamic model for on line path generation in an obstacle free environment.

In this thesis, a complete kinematics and Newtonian dynamics model for a tricycle is developed, rather than the two wheeled equivalent proposed in (Graettinger, T.J. and Krogh, B.H. [7]). These models are used for trajectory generation and obstacle

avoidance control.

2.4 Obstacle Avoidance using Artificial Impedance Control

Recently, most of the research for mobile robots focuses on achieving collision free path to the goal position. Mainly there are two types of obstacle avoidance algorithms, graph searching and potential field methods (Warren, C.W. [37]). The graph searching methods use complex sensor systems which give the full geometry of work space and require lots of memory capacity for calculation in computer. Other shortcomings when using only the information from the sensors are poor directionality, frequent misreadings, specular reflections, etc (Borenstein, J. and Koren, Y. [24]). Since these methods are used for high level control only which needs more memory capacity for computation, and generally are not feasible to implement in real time (Khatib, O. [38]).

Artificial potential field and artificial impedance are two of the most recent approaches for obstacle avoidance developed in the last decade. These control methods create trajectories and avoid obstacles along the way to the goal position. The potential field method uses the resultant forces of an artificial potential field applied to obstacles and goal positions. These methods require less memory capacity and increase the speed of algorithm which makes it possible to be implemented in real-time, and easy to extend to higher dimensions. Khatib, O. [38] is the researcher who

used the APF (Artificial Potential Field) approach. Later APF algorithm proved very similar to impedance control system, developed independently (Hogen, N. [43]). In the APF approach, an attractive potential is also posed at the goal position and acts as a PSP(points subject to the potential) located on manipulator's end effector or the centre of mass of mobile robot. Also, a repulsive potential field is posed around the obstacles and PSP on the manipulator links or mobile robot body. In this form an attractive pole for the robot acts as the target position and the repulsive surface for the robot parts acts as the obstacles. The robot's equation of motion is obtained as the sum of the potentials which placed into a Lagrangian formulation along with the manipulator or mobile robot kinetic energy. This algorithm is very susceptible to local minima which limits its usefulness. One of the other early researchers in the application of potential field to path planning is Hogan, N. [43]. He developed a complete manipulator controller not just of path planner, reformulating APF as an impedance control. He developed a control scheme both for free and contact motion. Free motion control refers to motion in which no physical contact with environment occurs while in contact motion control, the control system uses an extra feedback from the contact force between the robot and the environment. This control scheme operates in Cartesian space and all dynamic terms of the robot model have to be decoupled for the Cartesian space. Recently, Necsulescu, D.S. and Jassemi-Zargani, R. [46] applied the impedance control scheme in Cartesian space for trajectory generation and

obstacle avoidance of single and dual arm robot manipulators.

For the mobile robot control, force field based obstacle avoidance had been suggested by Krogh, B.H. [39], Khatib, O. [38], and Krogh, B.H. and Thorpe, C.E. [40], although these methods were not used for real-time navigation. Brooks, R.A. [41] and Arkin, R.C. [42] use force field methods on experimental mobile robots (equipped with ultrasonic sensors). Brooks, R.A. [41] uses only the current set of range readings to compute the resultant repulsive force for implementation. Borenstein, J. and Koren, Y. [24],[25] developed VFF (Virtual Force Field) method for real-time obstacle avoidance algorithm. In this thesis, an impedance control scheme is applied to the three-wheeled mobile robot for generating trajectory and for avoiding any obstacles in the way to the goal position using a complete model of nonlinear and cross coupling dynamic terms for inverse dynamics.

III DYNAMIC MODELLING

3.1 Description of the Mobile Robot

In this work a simplified three wheeled mobile robot is considered which has front wheel steering and two fixed orientation rear wheels (Fig. 3.1). In this configuration, the front wheel is driven and steered and the two rear wheels are passive. The mobility of the wheeled mobile robot highly depends upon the structure, dynamics, and the controller of the mobile robot. Generally, the dynamic modelling of a wheeled vehicle is quite complicated as a result of the influence of the suspension system, brake system, roll and yaw dynamics, and wheel slip conditions, etc. The development of dynamics based controllers for mobile robots have been the subject of several recent papers (Krogh and Thoupe, 1986; Graettinger and Krogh, 1989; Saha and Angeles, 1989; Campion and others, 1990; Shiller and Gwo, 1991; Canudas and Sordalen, 1991).

Mobile robots, being non-holonomic systems, require multiple reference frames chosen in such a way to facilitate the development of the kinematic model, dynamic model and control law. Inertial and moving reference frames were chosen for this purpose (Graettinger and Krogh, 1989). A fixed world reference frame in Cartesian space (T-Q-S) is needed to describe the motion of the mobile robot. For the development of the dynamic equations, it is

helpful to define inertial reference frames which are aligned with the mobile robot body and originated at the centre of mass of the mobile robot. Finally, a moving reference frame can be attached to the mobile robot in such a way that the x-axis is along the orientation of the mobile robot and the y-axis is parallel to the rear axle and the origin is at the centre of mass of the mobile robot. At each instant of the motion of the mobile robot, the moving reference frame of the mobile robot will be aligned with the inertial reference frame attached to the particular point on the path visited by the centre of mass of the mobile robot and the dynamic equations can be written with regard to that instantaneously coincident inertial reference frame. With regard to a fixed reference frame T-Q-S, the planes T-Q and x-y are assumed parallel for the study of planar motion, i.e. the unit vector s and k are parallel (Fig. 3.1).

The coordinate transformation gives the position (T,Q) in T-Q plane of a point defined by the position (x,y) in x-y plane where (T_0, Q_0) is the position of the origin of x-y plane in T-Q plane, and θ is the orientation angle between T and x axes of the two inertial frames, T-Q-S and x-y-z. The derivatives of θ are zero given that it refers to fixed reference frames.

$$\begin{bmatrix} T \\ Q \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} T_0 \\ Q_0 \end{bmatrix} \qquad (3.1)$$

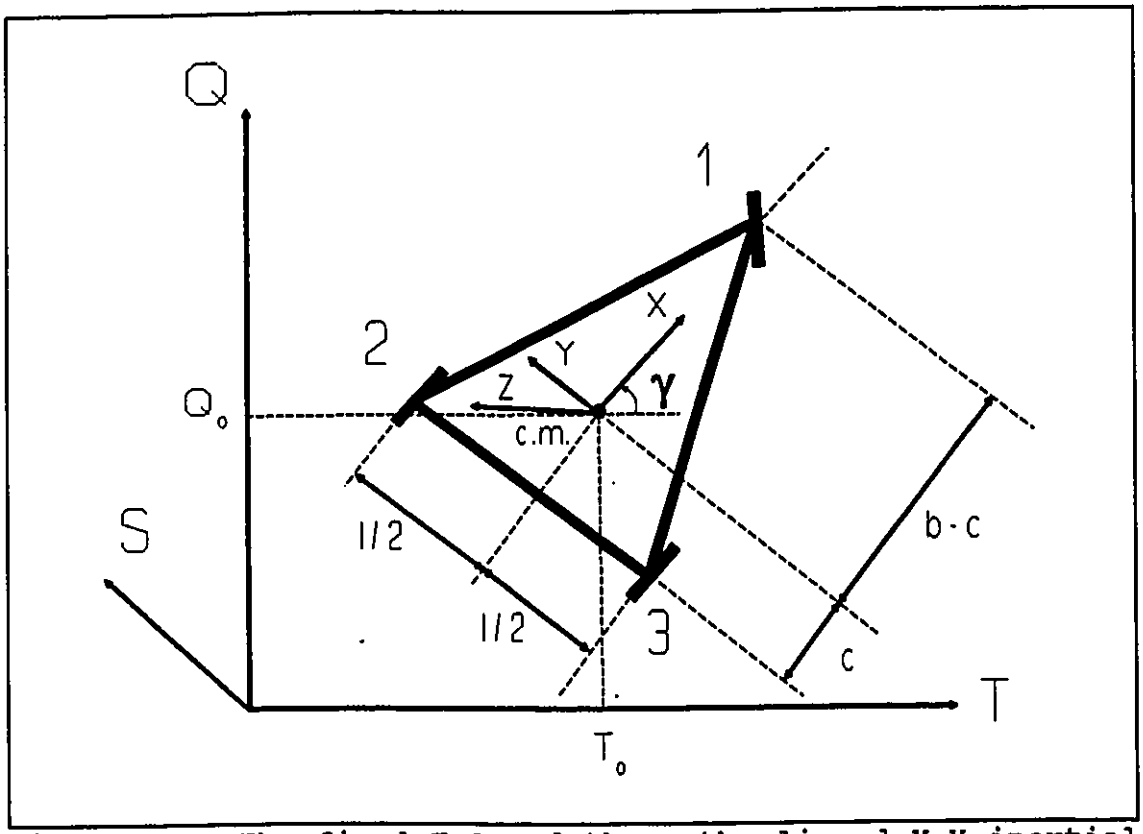

Figure 3.1. The fixed T-Q and the path aligned X-Y inertial reference frames

In vectorial form, the coordinate transformation gives the eq.(3.1) and its derivatives as,

$$\bar{R}_{TQ} = \bar{R}_{TQ_o} + \bar{r} \tag{3.2}$$

$$\dot{\bar{R}}_{TQ} = \dot{\bar{r}} \tag{3.3}$$

$$\ddot{\bar{R}}_{TQ} = \ddot{\bar{r}} \tag{3.4}$$

where,

$$\bar{R}_{TQ} = T\bar{t} + Q\bar{q} \tag{3.5}$$

$$\bar{R}_{TQ_o} = T_0\bar{t} + Q_0\bar{q} \tag{3.6}$$

$$\bar{r} = x\bar{i} + y\bar{j} \tag{3.7}$$

$\bar{t},\bar{q}$ are the unit vectors in T-Q fixed reference frame and $\bar{i},\bar{j}$ are the unit vectors in x-y moving reference frame, and they are related by a rotational transformation

$$\begin{bmatrix} i \\ j \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} t \\ q \end{bmatrix} \tag{3.8}$$

3.2 Kinematic model of the Mobile Robot

Under rigid body assumption, the absolute speed and the absolute acceleration of an arbitrary point $i(x_i,y_i)$ on the frame of the mobile robot with regard to the aligned inertial x-y-z frame($\bar{i}=\bar{I}$, $\bar{j}=\bar{J}$, $\bar{k}=\bar{K}$) can be obtained as follows, (Anand, D.K. and

Cuniff, P.F. [48]) (Fig. 3.2).

$$\bar{V}_i = \bar{V} + \dot{\theta}\bar{k} \times \bar{r}_i \qquad i=1,2,3 \qquad (3.9)$$

and,

$$\bar{a}_i = \bar{a} + \ddot{\bar{\theta}} \times \bar{r}_i + \dot{\theta}\bar{k} \times (\dot{\theta}\bar{k} \times \bar{r}_i) \qquad i=1,2,3 \qquad (3.10)$$

where,

$\bar{V}$ is the absolute speed of the centre of mass of the mobile robot ($\bar{V}=\dot{\bar{R}}=V_X\bar{I}+V_Y\bar{J}$) ;

$\bar{a}$ is the absolute acceleration of the centre of mass of the mobile robot ($\bar{a}=\ddot{\bar{R}}=a_X\bar{I}+a_Y\bar{J}$) ;

$\bar{V}_i$ is the absolute velocity of the centre point of each wheel ($\bar{V}_i=V_{Xi}\bar{I}+V_{Yi}\bar{J}$), (i=1,2,3) ;

$\bar{a}_i$ is the absolute acceleration of the centre point of each wheel ($\bar{a}_i=a_{Xi}\bar{I}+a_{Yi}\bar{J}$), (i=1,2,3) ;

$\bar{r}_i$ is the position vector measured in the x-y moving frame ($\bar{r}_i=x_i\bar{I}+y_i\bar{J}$), (i=1,2,3) ;

$\dot{\theta}$ is the absolute angular velocity of the x-y moving reference frame (mobile robot). ($\dot{\bar{\theta}}=\dot{\theta}\bar{k}$) ;

$\ddot{\bar{\theta}}$ is the absolute angular acceleration of the x-y moving reference frame. ($\ddot{\bar{\theta}}=\ddot{\theta}\bar{k}$);

Obviously, due to the parallelism of the axes z and S, the vertical component of the angular acceleration is the same in all systems of reference. At any instant of time the angles γ and θ are identical, but only θ has non-zero derivatives given that it refers to the relationship between a fixed and moving reference frame. We define the angular velocities ω_1, ω_2 and ω_3 of the three wheels which have radii r_{w1}, r_{w2} and r_{w3}, respectively.

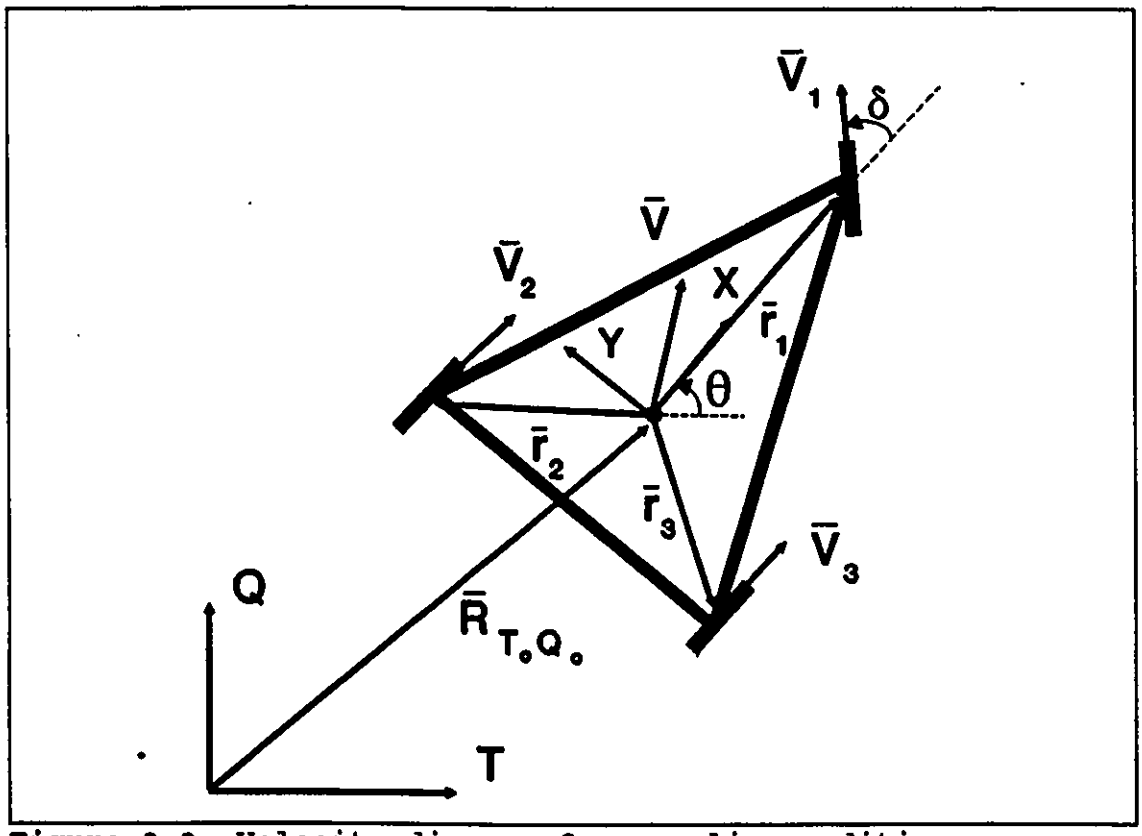

Figure 3.2. Velocity diagram for no slip conditions.

In the case that no sideway slip occurs,

$$V_{Y2} = V_{Y3} = 0 \tag{3.11}$$

and from Fig. 3.2,

$$V_{X1} = \omega_1 r_{w1} \cos\delta \tag{3.12}$$

$$V_{Y1} = \omega_1 r_{w1} \sin\delta \tag{3.13}$$

$$V_{X2} = \omega_2 r_{w2} \tag{3.14}$$

$$V_{X3} = \omega_3 r_{w3} \tag{3.15}$$

where δ is the steering angle of the front wheel with respect to moving reference frame. The accelerations $\bar{a}_i$ are dependent on ω_i, $\dot{\omega}_i$ and $\dot{\delta}$. The derivation of $\bar{a}_i$ (Equation (3.16) - (3.21)) are illustrated in Appendix C.

$$a_{X1} = r_{w1} \dot{\omega}_1 \cos\delta - r_{w1} \omega_1 (\dot{\theta}+\dot{\delta}) \sin\delta \tag{3.16}$$

$$a_{Y1} = r_{w1} \dot{\omega}_1 \sin\delta + r_{w1} \omega_1 (\dot{\theta}+\dot{\delta}) \cos\delta \tag{3.17}$$

$$a_{X2} = r_{w2} \dot{\omega}_2 \tag{3.18}$$

$$a_{Y2} = r_{w2} \omega_2 \dot{\theta} \tag{3.19}$$

$$a_{X3} = r_{w3}\ \dot{\omega}_3 \tag{3.20}$$

$$a_{Y3} = r_{w3}\ \omega_3\ \dot{\theta} \tag{3.21}$$

Since we have the velocities of the wheels in terms of the centre of mass in equation (3.9) from the rigid body assumption, we can combine equation (3.9) and equations (3.11) ~ (3.15), and obtain non-holonomic constraints.

$$\omega_1\ r_{w1}\ (\bar{i}\ \cos\delta + \bar{j}\ \sin\delta) = \bar{V} + (\dot{\theta}\bar{k}) \times \bar{r}_1 \tag{3.22}$$

$$\omega_i\ r_{wi}\ \bar{i} = \bar{V} + (\dot{\theta}\bar{k}) \times \bar{r}_i \qquad i=2,3 \tag{3.23}$$

The 12 scalar equations (3.22), (3.23),and (3.10) describe the kinematics of the mobile robot under no slip conditions. The six scalar equations (3.22) and (3.23) are not all linearly independent. In fact the imaginary part of equation (3.23) for i=2 and i=3 give identical scalar equations

$$0 = V_Y - c\dot{\theta} \tag{3.24}$$

The five scalar linearly independent equations (3.22) and (3.23) have seven variables, $\dot{\theta}$, ω_1, ω_2, ω_3, V_x, V_y and δ. Two variables will be considered as independent variables and the other five will be dependent variables.

Equations (3.22) and (3.23) can be used for forward kinematics and inverse kinematics. In the case of forward kinematics, for given ω_1 and δ solutions for the unknowns ω_2, ω_3, $\dot{\theta}$, V_x and V_y will be obtained.

If we express equations (3.22) and (3.23) in scalar form, equation (3.22) becomes,

$$V_X = \omega_1 \, r_{w1} \cos\delta \tag{3.25}$$

$$V_Y = \omega_1 \, r_{w1} \sin\delta - (b-c) \, \dot{\theta} \tag{3.26}$$

and equation (3.23) becomes,

$$V_x = r_{w2} \, \omega_2 + (1/2) \, \dot{\theta} \tag{3.27}$$

$$V_Y = c\dot{\theta} \tag{3.28}$$

$$V_x = r_{w3} \, \omega_3 - (1/2) \, \dot{\theta} \tag{3.29}$$

First of all, we obtain V_x from equation (3.25). Combining equations (3.26) and (3.28), we can obtain $\dot{\theta}$ as,

$$\dot{\theta} = (1/b) \, r_{w1} \, \omega_1 \sin\delta \tag{3.30}$$

Inserting equation (3.30) into equation (3.28) gives,

$$V_y = (c/b)\ r_{w1}\ \omega_1 \sin\delta \qquad (3.31)Hx$$

Also, equations (3.27) and (3.29) give,

$$\omega_2 = (1/r_{w2})\ [\ V_x - (1/2)\ \dot{\theta}] \qquad (3.32)$$

$$\omega_3 = (1/r_{w3})\ [\ V_x + (1/2)\ \dot{\theta}] \qquad (3.33)$$

For inverse kinematics, with given V_x and V_y, the unknowns ω_1, ω_2, ω_3, $\dot{\theta}$ and δ will be obtained, and δ is the result of the only nonlinear dependence

$$\delta=\tan^{-1}[\ (V_y/V_x)\ (b/c)] \qquad (3.34)$$

This equation will be analysed later in this chapter with regard to non-holonomicities of this robot.

From equations (3.27), (3.28) and (3.29), we can determine ω_2, ω_3 and $\dot{\theta}$ as,

$$\omega_2 = (1/r_{w2})\ [\ V_x - [1/(2c)\ V_y]] \qquad (3.35)$$

$$\omega_3 = (1/r_{w3})\ [\ V_x + [1/(2c)\ V_y]] \qquad (3.36)$$

$$\dot{\theta} = (1/c)\ V_y \qquad (3.37)$$

Combining equations (3.25) and (3.26), we can obtain,

$$\omega_1 = (1/r_{w1}) \left[V_x^2 + [V_y + (b-c)\ \dot{\theta}]^2 \right]^{1/2} \qquad (3.38)$$

The integration of $\dot{\theta}$ for a given initial value $\theta(0)$ permits the calculation of $\theta(t)$ used in the transformation equation (3.2) of the vectors from the x-y frame into the fixed T-Q frame.

Equations (3.10) and (3.16) to (3.21) give,

$$r_{w1}\dot{\omega}_1 (\bar{i}\cos\delta + \bar{j}\sin\delta) + r_{w1}\omega_1 (\dot{\theta}+\dot{\delta}) (-\bar{i}\sin\delta + \bar{j}\cos\delta) =$$

$$\bar{a} + \ddot{\bar{\theta}} \times \bar{r}_1 + (\dot{\theta}\bar{k}) \times (\dot{\theta}\bar{k} \times \bar{r}_1) \qquad (3.39)$$

$$r_{wi} (\omega_i \bar{i} + \omega_i \dot{\theta} \bar{j}) = \bar{a} + \ddot{\bar{\theta}} \times \bar{r}_i + (\dot{\theta}\bar{k}) \times (\dot{\theta}\bar{k} \times \bar{r}_i), \qquad i=2,3 \qquad (3.40)$$

The real parts of equation (3.23) for i=2 and i=3 give

$$\omega_2 r_{w2} = V_X - \dot{\theta}(1/2)$$

$$\omega_3 r_{w3} = V_X + \dot{\theta}(1/2)$$

Eliminating V_x, we can obtain

$$\omega_2 r_{w2} = \omega_3 r_{w3} - 2\dot{\theta}(1/2)$$

which used in equation (3.40) for i=2.

$$\omega_2 r_{w2}\dot{\theta} = a_Y - (1/2)\dot{\theta}^2 - c\ddot{\theta}$$

gives

$$(\omega_3 r_{w3} - l\dot{\theta})\dot{\theta} = a_Y - (1/2)\dot{\theta}^2 - c\ddot{\theta}$$

or

$$\omega_3 r_{w3}\dot{\theta} = a_Y + (1/2)\dot{\theta}^2 - c\ddot{\theta}$$

This is in fact the derivative of the real part of equation (3.23) for i=3.

This fact proves that only five of the six equations (3.39) and (3.40) are linearly independent. Equations (3.39) and (3.40) contain twelve unknowns ω_1, ω_2, ω_3, $\dot{\omega}_1$, $\dot{\omega}_2$, $\dot{\omega}_3$, $\dot{\theta}$, $\ddot{\theta}$, δ, $\dot{\delta}$, a_x and a_y. Following the direct kinematics or inverse kinematics solutions for the velocity equations, all variables ω_1, ω_2, ω_3, $\dot{\theta}$ and δ are determined. Using these variables as independent variables together with the given values for a_x and a_y, the five linearly independent equations (3.39) and (3.40) can be solved for the five dependent variables, the accelerations $\dot{\omega}_1$, $\dot{\omega}_2$, $\dot{\omega}_3$, $\ddot{\theta}$ and the speed $\dot{\delta}$.

From equation (3.39),

$$\dot{\omega}_1 = (1/r_{w1}) \{ [a_x - (b-c)\dot{\theta}^2] \cos\delta + [a_y + (b-c)\ddot{\theta}] \sin\delta \} \quad (3.41)$$

and from equation (3.40),

$$\ddot{\theta} = (1/c) [a_y - r_{w2}\dot{\omega}_2\theta - (1/2)\dot{\theta}^2] \quad (3.42)$$

$$\dot{\omega}_2 = (1/r_{w2}) \{a_x - [1/(2c)]a_y+(1/2)r_{w2}\dot{\omega}_2\theta+[(4c^2+l^2)/(4c)]\dot{\theta}^2\} \quad (3.43)$$

$$\dot{\omega}_3 = (1/r_{w3}) [a_x + c\dot{\theta}^2 + (1/2)\ddot{\theta}] \quad (3.44)$$

The steering rate $\dot{\delta}$ results from the derivative of steering angle δ,

$$\dot{\delta} = (a_y V_x - a_x V_y) / \{ [(c/b) + (b/c)(V_y/V_x)^2] V_x^2 \} \quad (3.45)$$

3.3 The Newtonian Dynamic Model of The Mobile Robot

The dynamic model of the mobile robot is obtained using free body diagrams for the rigid mobile robot frame and for each of the three wheels. For planar motion, three equations are obtained for each free body. The Newtonian dynamics formulation presented here for three wheeled mobile robot is similar to the one used by (Grattinger, T.J. and Krogh, B.H. [7]) for the simplified case of an equivalent two wheeled mobile robot.

The reaction forces between the frame and the wheels are given in two orthogonal components, F_{xi}, F_{yi}, (Fig. 3.3). The twisting moment and viscous frictions at the axles are neglected.

The following equations result:

-for the vehicle frame(fig. 3.3)

$$ma_x = F_{x2} + F_{x3} + F_{x1}\cos\delta - F_{y1}\sin\delta \quad (3.46)$$

$$ma_y = F_{y2} + F_{y3} + F_{x1}\sin\delta + F_{y1}\cos\delta \quad (3.47)$$

$$I_{cm}\ddot{\theta} = -c(F_{y2}+F_{y3}) - (1/2)(-F_{x2}+F_{x3}) + (b-c)(F_{x1}\sin\delta+F_{y1}\cos\delta) \quad (3.48)$$

where,

m, I_{cm} are the mass and the moment of inertia with regard to the centre of mass of the mobile robot frame, respectively.

b, c and l are refer to fig. 3.1.

-for the front wheel (fig. 3.4a)

$$m_{w1}a_{x1} = (G_{x1}-F_{x1})\cos\delta - (G_{y1}-F_{y1})\sin\delta \quad (3.49)$$

$$m_{w1}a_{y1} = (G_{x1}-F_{x1})\sin\delta + (G_{y1}-F_{y1})\cos\delta \quad (3.50)$$

$$I_{w1}\dot{\omega}_1 = \tau_1 - G_{x1}r_{w1} \quad (3.51)$$

-for the rear wheels, i=2,3 (fig. 3.4b)

$$m_{wi}a_{xi} = G_{xi} - F_{xi} \quad (3.52)$$

$$m_{wi}a_{yi} = G_{yi} - F_{yi} \quad (3.53)$$

$$I_{wi}\dot{\omega}_i = -G_{xi}r_{wi} \quad (3.54)$$

where,

m_{wi}, I_{wi} (i=1,2,3) are the mass and the moment of inertia of each wheel with regard to their axis of rotation.

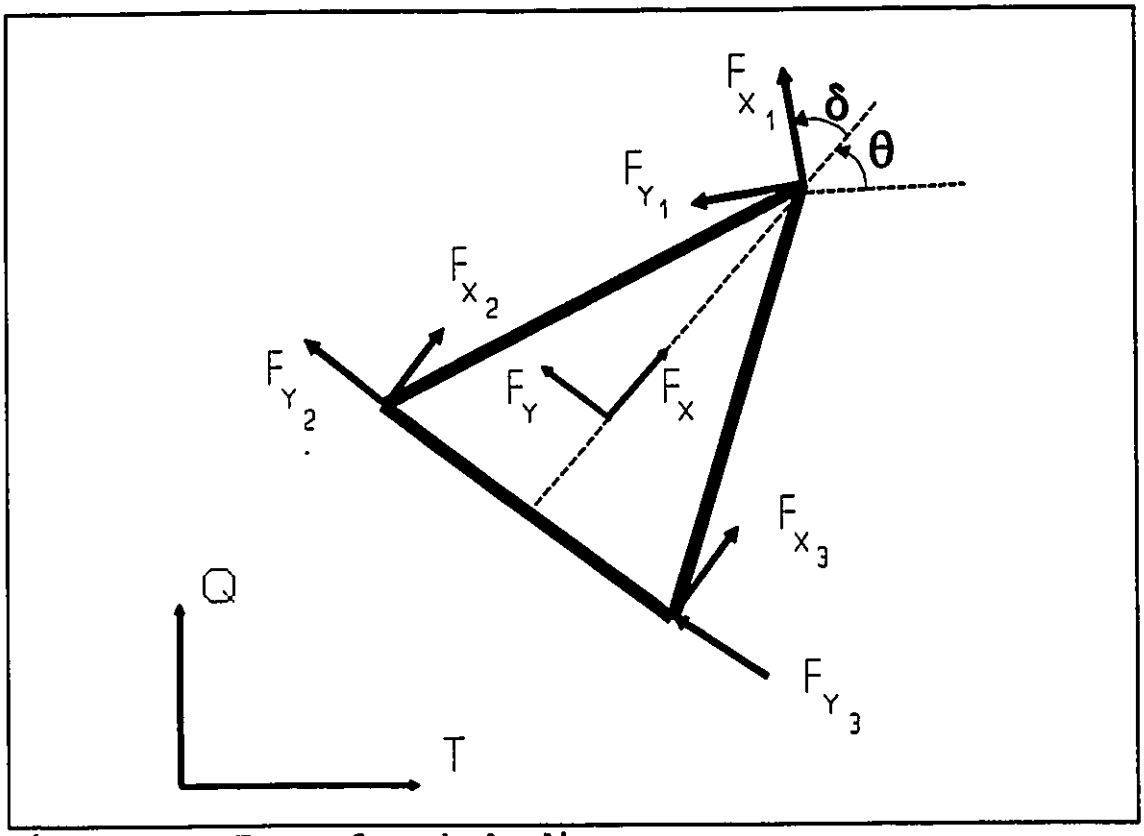

Figure 3.3. Frame free body diagram.

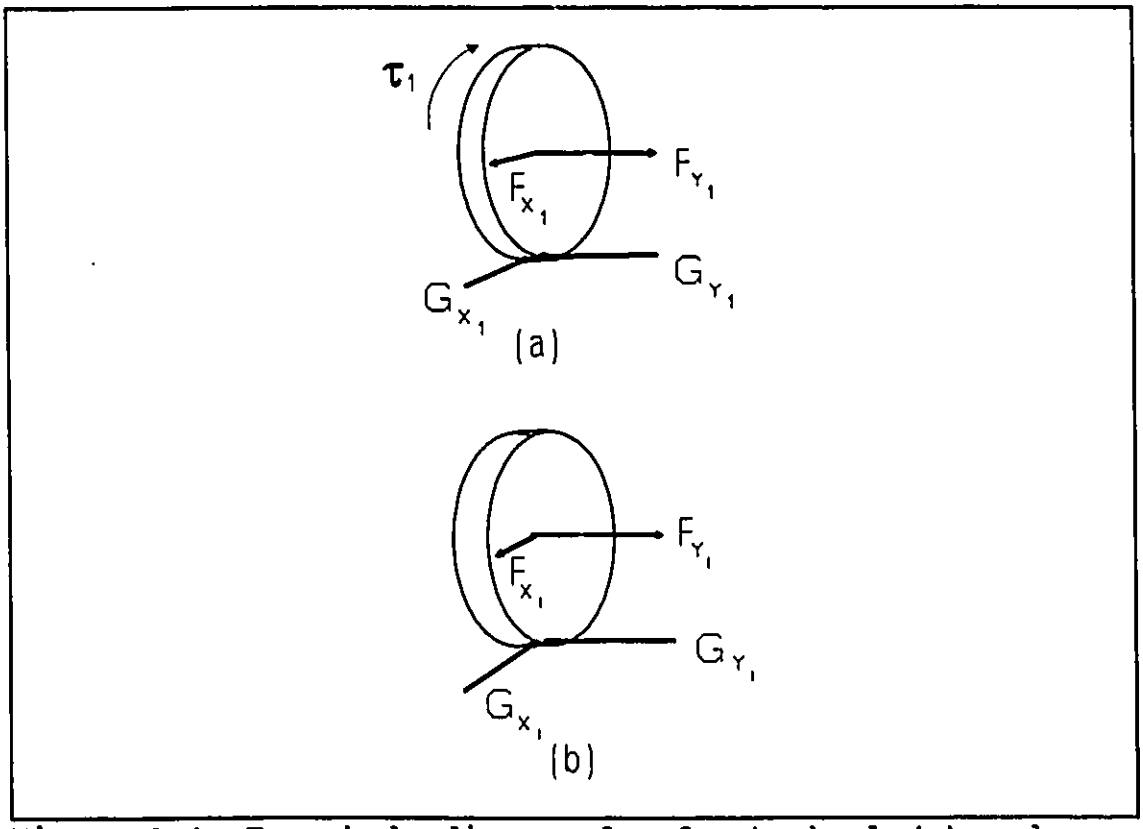

Figure 3.4. Free body diagram for front wheel (a) and rear wheels i=2,3 (b).

These twelve equations contain twelve unknown forces (which are the reaction forces (F_{xi}, F_{yi}), (i=1,2,3), the frictional forces between the wheels and ground (G_{xi}, G_{yi}), (i=1,2,3)) and an input torque τ_1 to the front wheel (#1). For given positions, speeds and accelerations and a given state dependent control law, which commands the input torque, the dynamic model has a unique solution at each instant of time. In order to obtain a solution for τ_1, one extra equation is needed. Assuming that the rear axle is elastic but with a very high stiffness, we can prove that we can solve the equations (3.46) to (3.54) considering,

$$F_{y2} \approx F_{y3} \qquad (3.55)$$

The derivation of this equation is given in Appendix A.

The kinematic equations (3.9) to (3.45) and the dynamic equations (3.46) to (3.55) together determine the reaction forces $\bar{F}$ and the friction forces $\bar{G}$.

$\bar{F} = [\ F_{x1},\ F_{y1},\ F_{x2},\ F_{y2},\ F_{x3},\ F_{y3}\ ]^T$

$\bar{G} = [\ G_{x1},\ G_{y1},\ G_{x2},\ G_{y2},\ G_{x3},\ G_{y3}\ ]^T$

and the propulsion torque command τ_1.

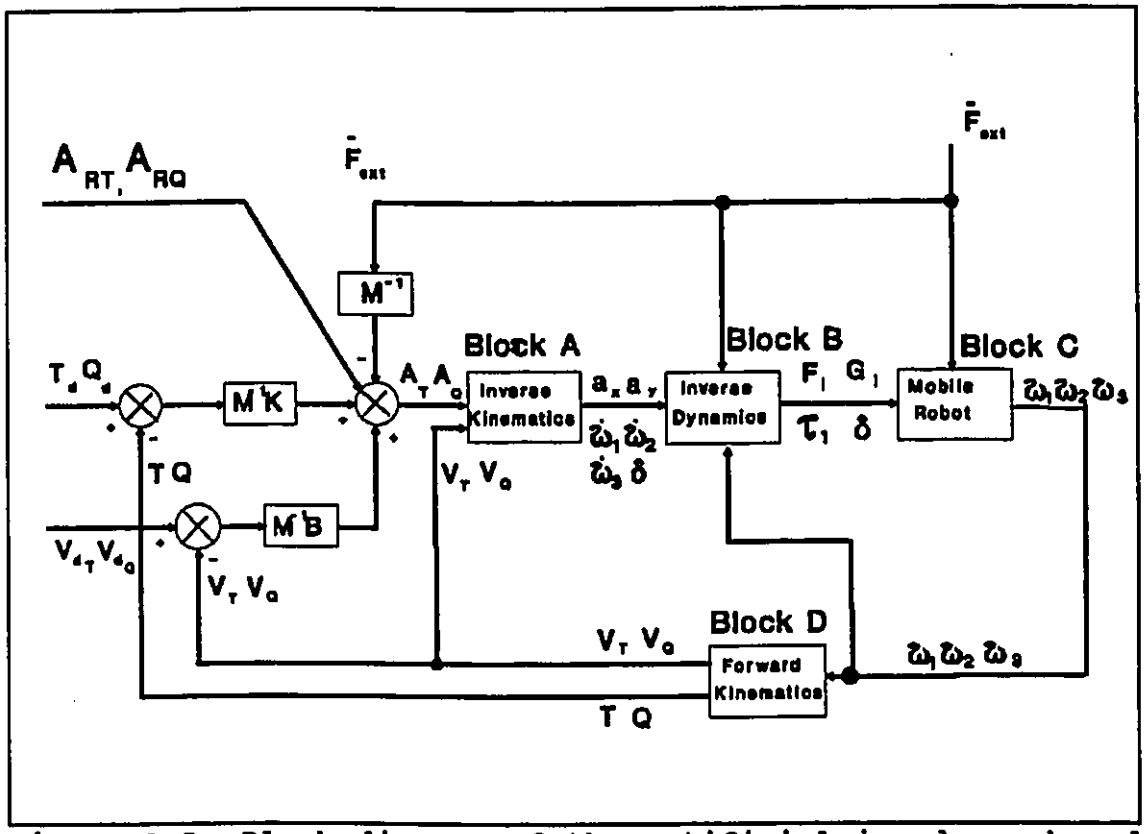

Figure 3.5 Block diagram of the artificial impedance based controller for free motion, collision avoidance and contact motion (A_{RT}, A_{RQ} are generated by collision avoidance)

Equations (3.46) to (3.55) are the basis for the results presented in Appendix B, Block B(inverse dynamics), used for

nonlinear compensation and decoupling of the mobile robot, and Block C (mobile robot system dynamics), used for the simulation of the mobile robot dynamics, as shown in the block diagram (Fig. 3.5).

The inverse dynamics give the control inputs required to execute the trajectory. In matrix form;

$$\begin{bmatrix} \cos\delta & -\sin\delta & 1 & 0 & 1 \\ \sin\delta & \cos\delta & 0 & 2 & 0 \\ q_1\cos\delta - q_4\sin\delta & -q_1\sin\delta - q_4\cos\delta & q_3+q_8 & 2q_2 & q_1-q_3 \\ q_{10}\sin\delta & q_{10}\cos\delta & q_6 & 2(q_1+q_5) & -q_6 \\ q_1\cos\delta + q_4\sin\delta & -q_1\sin\delta - q_4\cos\delta & q_1-q_3 & -2q_2 & q_3+q_8 \end{bmatrix} \begin{bmatrix} F_{x_1} \\ F_{y_1} \\ F_{x_2} \\ F_{y_2} \\ F_{x_3} \end{bmatrix} = \begin{bmatrix} ma_x \\ ma_y \\ -c\dot{\theta}^2 \\ r_{w2}\omega_2\dot{\theta} + \frac{I}{2}\dot{\theta}^2 \\ -c\dot{\theta}^2 \end{bmatrix} \tag{3.56}$$

And the torque command for the front wheel is given by,

$$\tau 1 = [(q_1+q_{13}+q_{12}\sin^2\delta)F_{x1} + (q_{12}\sin\delta\ \cos\delta)F_{y1} + (q_1\cos\delta - q_{11}\sin\delta)F_{x2} + (2q_{10}\sin\delta)F_{y2} + (q_1\cos\delta + q_{11}\sin\delta)F_{x3} - (b-c)\cos\delta\ \theta^2] / q_{14} \tag{3.57}$$

The frictional forces for the three wheels are given by,

$$G_{x1} = [(q_{28}(q_1+q_{13}+q_{12}\sin^2\delta)+q_{18}]F_{x1} + (q_{12}q_{28}\sin\delta\cos\delta)F_{y1} + [q_{28}(q_1\cos\delta - q_{11}\sin\delta)]F_{x2} + (2q_{10}q_{28}\sin\delta)F_{y2} + [q_{28}(q_1\cos\delta + q_{11}\sin\delta)]F_{x3} - q_{28}(b-c)\dot{\theta}^2\cos\delta$$

$$G_{y1} = (q_{12}q_{27}\sin\delta\cos\delta)F_{x1} + (1+q_1q_{27}+q_{12}q_{27}\cos^2\delta)F_{y1} - [q_{27}(q_1\sin\delta+q_{11}\cos\delta)]F_{x2} + (2q_{10}q_{27}\cos\delta)F_{y2} - [q_{27}(q_1\sin\delta+q_{11}\cos\delta)]F_{x3} + [q_{27}(b-c)\dot{\theta}^2\sin\delta]$$

$$G_{x2} = q_{16}\, F_{x2}$$

$$G_{y2} = F_{y2} + q_{26}r_{w2}\dot{\omega}_2\theta$$

$$G_{x3} = q_{16}\, F_{x3}$$

$$G_{y3} = F_{y3} + q_{26}r_{w2}\dot{\omega}_2\theta + m_{w2}l\dot{\theta}^2 \qquad (3.58)$$

The notations (q_i ... q_{28}) used in the inverse dynamic equations have been analyzed in Appendix B, Block B.

3.4 The non-holonomic constraints

For the calculation of F_i (reaction forces) and G_i (frictional forces between the wheels and ground), we need the velocity commands v_x and v_y added to the inputs from inverse kinematics. For given values of $\dot{\omega}_2$ and $\dot{\theta}$, v_x and v_y are determined by the non-holonomic constraints, i.e. equations (3.27) and (3.28).

If the constraint is non-holonomic, these equations expressing the constraint cannot be integrated and cannot be used to eliminate

the dependent coordinates. Therefore, the systems containing non-holonomic constraints always require more coordinates for their description than their number of degree of freedom.

Since we have the mobile robot doing the planar motion rolls without slippage on the horizontal T, Q plane, we can choose as coordinates the location (T, Q) of the centre of mass of the mobile robot, and orientation angle θ with respect to the T axis of the fixed global reference coordinates T-Q-S. The equations (3.27) and (3.28) imply the requirement of rolling without slippage.

Through the rotational transformation we can rewrite equations (3.27) and (3.28) with regard to the fixed global reference frame T-Q-S:

$$V_T \cos\theta + V_Q \sin\theta - r_{w2}\omega_2 - (1/2)\dot{\theta} = 0 \qquad (3.59)$$

$$-V_T \sin\theta + V_Q \cos\theta - c\dot{\theta} = 0 \qquad (3.60)$$

The constraints equations (3.59) and (3.60) can be written in pfaffian form (Frank D'Souza, A. and Greg, V.K. [51]) as:

$$df = dT + dQ \tan\theta - (1/\cos\theta) r_{w2}\omega_2 - (1/\cos\theta)(1/2)\dot{\theta} = 0 \qquad (3.61)$$

$$df = dT \tan\theta - dQ + (1/\cos\theta) c\dot{\theta} = 0 \qquad (3.62)$$

The pfaffian has to satisfy the integrability requirements:

$$\frac{\partial}{\partial T}\left(\frac{\partial f}{\partial Q}\right) = \frac{\partial}{\partial Q}\left(\frac{\partial f}{\partial T}\right) \tag{3.63}$$

$$\frac{\partial}{\partial T}\left(\frac{\partial f}{\partial \theta}\right) = \frac{\partial}{\partial \theta}\left(\frac{\partial f}{\partial T}\right) \tag{3.64}$$

$$\frac{\partial}{\partial Q}\left(\frac{\partial f}{\partial \theta}\right) = \frac{\partial}{\partial \theta}\left(\frac{\partial f}{\partial Q}\right) \tag{3.65}$$

in order to make the pfaffian reducable to perfect differential:

$$df(T,Q,\theta) = \frac{\partial f}{\partial T}dT + \frac{\partial f}{\partial Q}dQ + \frac{\partial f}{\partial \theta}d\theta = 0 \tag{3.66}$$

If any of the pfaffian equations is not satisfied, the constraint equation is not integrable and is non-holonomic.

Let us execute the pfaffian for the equation (3.59). When equation (3.65) is performed, the result is;

$0 \neq \sec^2\theta$

And similarly, performing equation (3.64) for equation (3.60) gives same result. Therefore, the pfaffian is not satisfied for the constraint equations (3.59) and (3.60). This proves that the constraints (3.59) and (3.60) are non-holonomic constraints.

IV ARTIFICIAL IMPEDANCE APPROACH OF THE MOTION CONTROL

4.1 Control Algorithm

4.1.1 The basic concept of impedance control

The basic idea in using impedance control for mobile robots is to create a desired artificial mechanical impedance between the centre of mass of mobile robot and the desired target point by modulating the force and torque output of the robot actuators (Necsulescu D.S. et al [46]). The purpose of this impedance is to attract the mobile robot so that it will reach the desired target point. Considering an artificial link, consisting of a spring and a damper, between the centre of mass of the mobile robot and the target point, we can establish so called virtual mechanical M-B-K system. (Fig. 4.1). The spring component in this mechanical system will produce an attractive force in the direction of a straight line from the centre of mass of the mobile robot to the destination point in Cartesian space. The damping component is regulating the velocity to make the centre of mass of the mobile robot reach and stop at the desired target point.

This approach imposes a desired dynamics, normally the dynamics of a M-B-K linear system between the current position of the mobile robot R and the destination position R_D, i.e.

$$(Ms^2 + Bs + K)\ [r(s)-r_D(s)] = -F_{ext} \qquad (4.1)$$

where F_{ext} is the contact force. In the case of free motion $F_{ext}=0$, and equation (4.1) can be used to obtain the desired motion of a representative point of the mobile robot, say the centre of mass. Solving eq.(4.1) for $\ddot{\bar{R}}(t)$, we can use these solutions as the instantaneous desired motion to be followed by the mobile robot in the fixed inertial frame T-Q-S for measured $\dot{\bar{R}}$ and $\ddot{\bar{R}}$, i.e. a resolved acceleration method.

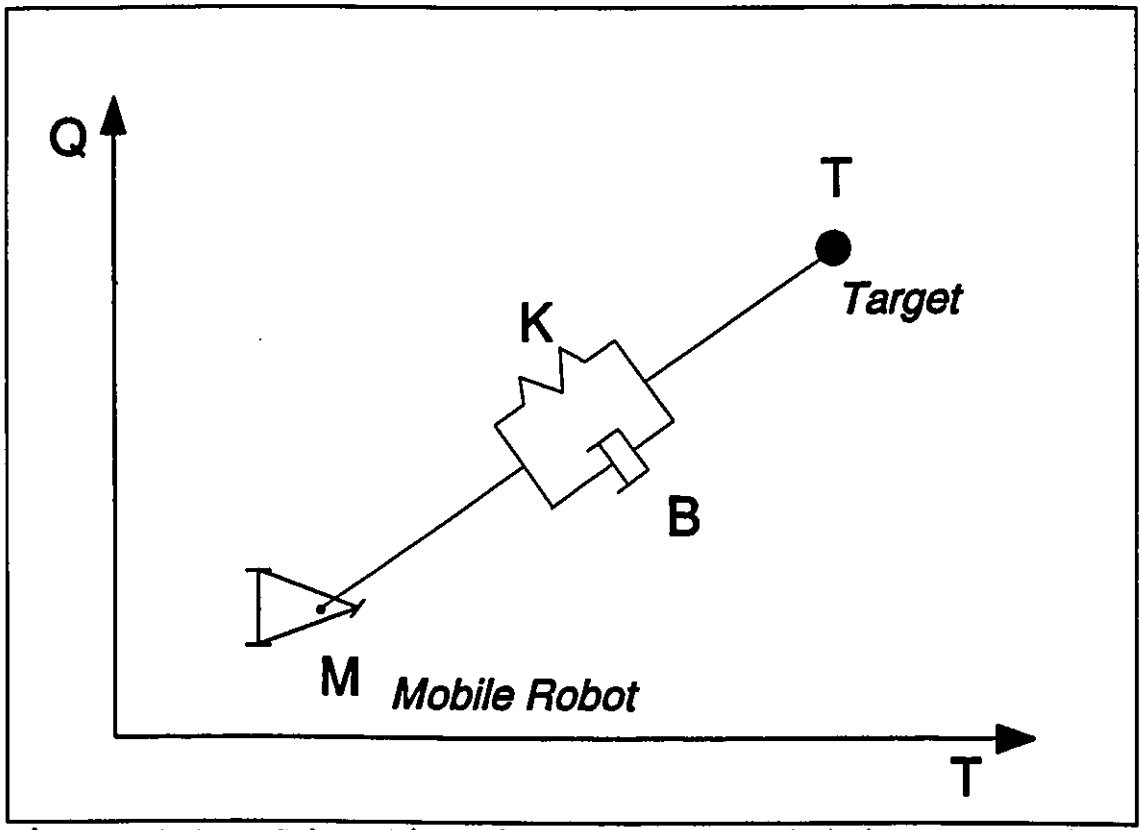

Figure 4.1 Schemetic diagram for artificial mechanical impedance M-B-K system of a mobile robot

Further, $\ddot{\bar{R}}(t) = A_T(t)\bar{t} + A_Q\bar{q}$ can be transformed into $\bar{a} = a_x\bar{I} + a_y\bar{J}$ using the transformation given in eq.(3.1). Using the resulting V and the similarly obtained A, in eq.(3.57), the mobile robot is forced to behave like a M-B-K linear system, i.e. virtual impedance described by eq.(4.1). The impedance matrices are normally chosen diagonal and in such a way to result in a critically damped motion toward the destination (Necsulescu, D.S. [49]).

4.1.2 Realization of the Impedance Control

The realization of the impedance control of a mobile robot requires the decoupling and linearization of the nonlinear dynamics terms resulting in an equivalent to a mass and vertical moment of inertia concentrated in the centre of mass of the mobile robot. The planar motion in this case implies position and orientation of the decoupled robot at a desired T_d, Q_d, θ_d which form a vector $\bar{R}_d$. Actual position and orientation being defined by T, Q, θ, or the vector $\bar{R}$, a simple spring - damper impedance affects the position and speed terms in the command in the position $\bar{e} = \bar{R} - \bar{R}_d$ and speed $\dot{\bar{e}} = \dot{\bar{R}} - \dot{\bar{R}}_d$ error terms. The translation of the speed term of the command into inputs to the servomotors implies a inverse Jacobian transformation which can be achieved only if the 3 x 3 Jacobian has linealy independent terms.

In case of a mobile robot with front wheel drive and

steering, only ω and δ are controlled and the Jacobian for the front wheel has two linearly dependent columns (Shilling, R. [50]). The situation is the same if the two parallel rear wheels are driven, i.e. only ω_2 and ω_3 are controlled (Canudas de Wit, C. and Sordalen, O.J. [14]). A piecewise continuous controller can lead to a point-to-point motion control, but this type of controller cannot be used for collision avoidance with unexpected obstacles. We can have better performance of artificial impedance control of a mobile robot by applying it to omnidirectional mobile robot. Such a robot can track x - y path in any desired orientation θ, i.e. is capable of a three degree of freedom planar controlled motion (Shilling, R. [50]). However, it is of interest to study a partial decoupling and linearization of a non-omnidirectional robot and the effect of a chosen artificial mechanical impedance on the motion in such case. In this thesis this study is performed for trajectory generation and collision avoidance of a front wheel steered and driven mobile robot, i.e. a tricycle.

4.2 Obstacle Avoidance of the Mobile Robot

4.2.1 The Realization of Obstacle Avoidance

Real-time obstacle avoidance is one of the major issues to successful application of mobile robots. Various methods have been proposed to plan a collision-free path. These methods are

classified into two groups. The first one plans the path based on local information about the obstacles in the work space, and the second one takes into account the global information about them. The impedance control method presented in this thesis belongs to the first type of collision free motion control algorithm.

The impedance controller generates the on-line trajectory which enables the mobile robot to execute the real-time obstacle avoidance. Only the distance between the closest point of the obstacle and the position of the centre of mass of the mobile robot is needed in this case. Therefore, since the robot controller does not need the knowledge of the full geometry of the obstacle, it does not have to be equipped with complex vision system or to perform replanning of new trajectories in case of unexpected obstacles. The trajectory of the mobile robot can be corrected in real time by the impedance control system to avoid the obstacles. The mobile robot moves toward the target point and avoids obstacles simultaneously, and it stops in safe distance from obstacle only in some critical situations.

There are basically two virtual force fields for obstacle avoidance applying impedance control, namely, virtual attractive force field for the desired target point and virtual repulsive force fields for the obstacles. The virtual attractive force field was already explained in section 4.1.1. And the virtual repulsive force field will be explained in next section.

4.2.2 The Repulsive Force Field of Obstacle Avoidance

A repulsive force field is introduced by the impedance controller in order to avoid obstacles. The repulsive force field is created around all obstacles within an area of effectiveness. The virtual repulsive forces increase as the mobile robot gets closer to the centre of the repulsive force field or in other word the obstacle. Eventually, this force field pushes the mobile robot away from colliding with the obstacle. If the mobile robot passes within this boundary, the resultant of the virtual attractive and the virtual repulsive forces leads the mobile robot to a new feasible trajectory.

If the mobile robot position is beyond the boundary of effectiveness the virtual repulsive force becomes zero and has no effect on the robot trajectory. That is, only the virtual attractive force is applied to the mobile robot to reach the target point.

To perform obstacle avoidance with impedance controller, the mobile robot has to be equipped with range sensors which give the information needed for the calculation of the shortest distance between the mobile robot and the surface of obstacles. An ultrasonic proximity sensor can be used for this purpose. Using the information sent by the range sensors, the virtual repulsive forces in the repulsive force fields are generated in the direction

toward the mobile robot from the obstacle. Figure (4.2) shows the resultant $\bar{F}$ of the sum of the virtual repulsive force $\bar{F}_r$ and the virtual attractive force $\bar{F}_a$. And the resultant $\bar{F}$ directs the mobile robot to a new trajectory.

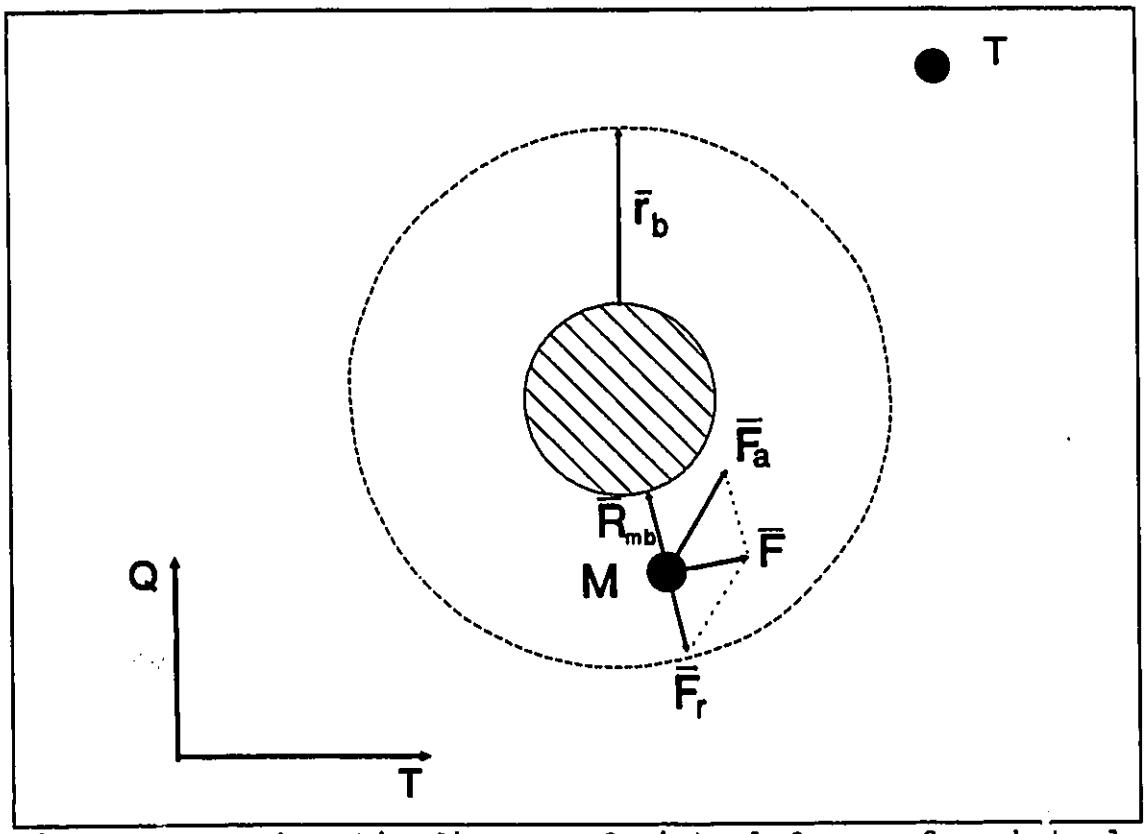

Figure 4.2 Schemetic diagram of virtual forces for obstacle avoidance control of a mobile robot

Refering to figure (4.2), we can illustrate the general equations of this repulsive force field algorithm.

$$\bar{F}_r = (F_{rT})\bar{i} + (F_{rQ})\bar{j} \qquad , \quad R_{mb} < r_b$$

$$\bar{F}_r = 0 \quad , \quad R_{mb} > r_b \tag{4.2}$$

$$F_{rT} = K_r \ (T_{mb}/R_{mb}) \ (R_{mb} - r_b)^2 \tag{4.3}$$

$$F_{rQ} = K_r \ (Q_{mb}/R_{mb}) \ (R_{mb} - r_b)^2 \tag{4.4}$$

where,

$T_{mb} = T_b - T_m$

$Q_{mb} = Q_b - Q_m$

$\bar{R}_{mb} = T_{mb}\bar{i} + Q_{mb}\bar{j}$

$R_{mb} = (T_{mb}^2 + Q_{mb}^2)^{1/2}$

where,

K_r : the spring constant for the virtual repulsive force.

T (T_d , Q_d) : The position of the target, where T_d is the T component and Q_d is the Q component of the target position.

M (T_m , Q_m) : The position of the mobile robot, where T_m is the T component and Q_m is the Q component of the mobile robot position.

B (T_b , Q_b) : The position of the boundary of an obstacle which the mobile robot can detect with the range sensors

at each period of time, where T_b is the T component and Q_b is the Q component of the obstacle boundary position.

T_{mb}: the T component of the distance between the mobile robot and the surface of obstacle.

Q_{mb}: the Q component of the distance between the mobile robot and the surface of obstacle.

R_{mb}: the shortest distance between the mobile robot and the surface of obstacle.

r_b : the radius of the repulsive force field effectiveness from the surface of obstacle.

$\bar{i}$, $\bar{j}$: the unit vectors in the Cartesian coordinates, where the bars denote vectors.

The resultant of the virtual attractive and repulsive forces is given by

$$\bar{F} = \bar{F}_a + \bar{F}_r \tag{4.5}$$

where,

$\bar{F}_a$: the virtual attractive force to the target.

$\bar{F}_r$: the virtual repulsive force from the obstacle.

$\bar{F}$: the resultant of the sum of $\bar{F}_a$ and $\bar{F}_r$.

The repulsive force field is designed to have no influence beyond r_b. And there are no discontinuities in force entering the area of effectiveness as repulsive forces are zero on both sides of

the boundary. The virtual repulsive force increases as the square as the centre is approached, reflecting a higher priority of defending the obstacle as the mobile robot gets closer. The repulsive force field is designed here rotationally symmetric and can defend the obstacle when approached from any direction. (Hogan, N. [43]).

4.2.3 The effect of damping coefficient on the repulsive force field

The damping effect on repulsive force field helps to generate a smooth path of the mobile robot by reducing the robot speed and preventing the mobile robot from approaching to the obstacle at high speed. Eventually it achieves a safe distance between the mobile robot and the obstacle. Since the damping effect reduces the robot speed, it increases the execution time. In some cases, the mobile robot can develop oscillatory motion. For example, in the case that the mobile robot is trapped in between two obstacles an oscillatory motion of the mobile robot could damage the mobile robot itself or offset the tasks it performs such as transporting some liquid from one position to the other. Choosing a suitable damping constant avoids the oscillation of the mobile robot. The selection of the damping constant(BR) is based on taking into account that high BR slows down the mobile robot by a large factor. If the damping coefficient is present, the equations used for the repulsive force field, equations (4.3) and (4.4), become

$$F_{rT} = K_r \ (T_{mb}/R_{mb}) \ (R_{mb} - r_b)^2 - B_r\dot{T}_m \qquad (4.6)$$

$$F_{rQ} = K_r \ (Q_{mb}/R_{mb}) \ (R_{mb} - r_b)^2 - B_r\dot{Q}_m \qquad (4.4)$$

where,

B_r : the damping constant for the virtual repulsive force.

The block diagram of the artificial impedance based controller in shown in Fig.3.5. Appendix B presents the equations of block A to D of Fig. 3.5.

V. SIMULATION RESULTS

5.1 Trajectory Generation

In the case of a mobile robot, for which the dynamic model, the parameters and the variables are exactly known, the motion resulting from an impedance controller described by equation (4.1) is generated by the linear virtual impedance M-B-K. Fig. 5.1 to Fig. 5.19 show the simulation results for a non-omnidirectional mobile robot, the tricycle, shown in Fig. 3.1.

Fig. 5.1a and fig.5.1b represent the case of moving from its initial position I(1,1) to target position D(10,10) for $\theta(0) = 0$. The result shown in figure 5.1a was obtained for the robot which achieves very fast steering angle command δ_r for the vertical inertia of the wheel #1, $J_{w1} = 0$ and for an ideal steering controller (Ks = 1 and Bs = 0). The Ks and Bs are the position gain and velocity gain for the steering controller, respectively. As the result shows, the impedance controller generates the quasi-straight trajectory for obstacle free case. This result could not be considered representative for actual non-omnidirectional mobile robot motion. And the simulation model has been modified to include a more realistic front wheel steering (J_{w1} is not zero and a PD control for the steering angle). The simulation result shown in fig. 5.1b for the obstacle free case shows that a curved trajectory would be generated in this case, and this fact

illustrates the effect of the transient response of the mobile robot steering control to the δ_r command. T(t) and Q(t) for the case of Fig. 5.1b are shown in Fig. 5.2 to show the amount of time spent by the mobile robot to reach to the target. Also, the time behaviour of actuator torque $\tau_1(t)$ for the case of Fig. 5.1b are shown in Fig. 5.3. The mobile robot reached to the target within 65 seconds.

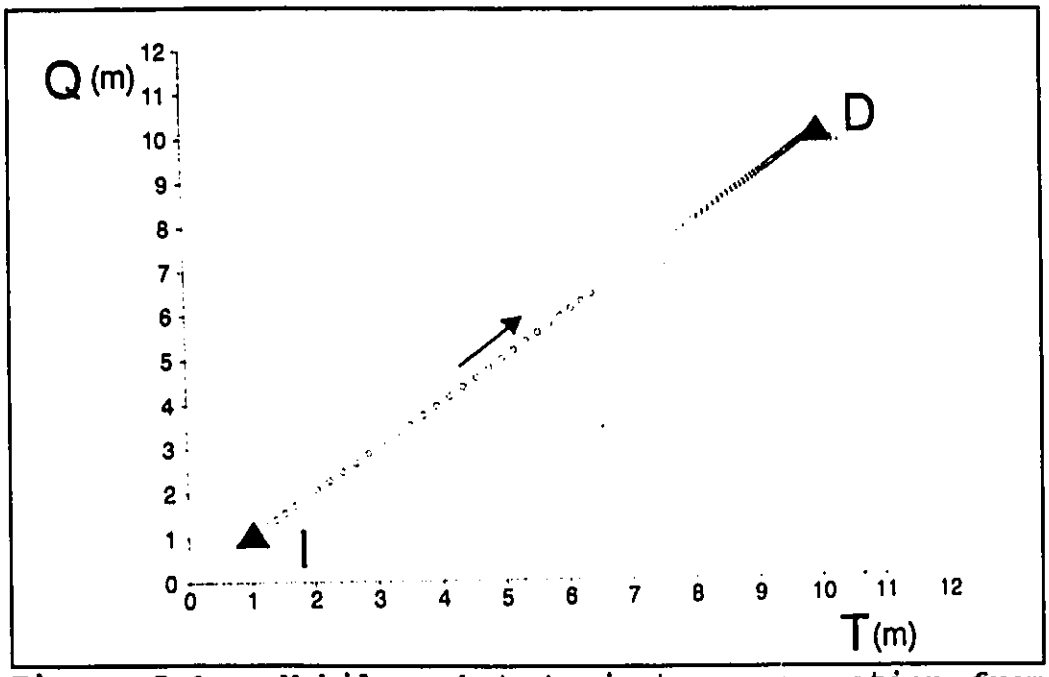

Figure 5.1a Mobile robot trajectory generation from point I(1,1) to point D(10,10) with ideal steering controller and the integration step of 10 ms.

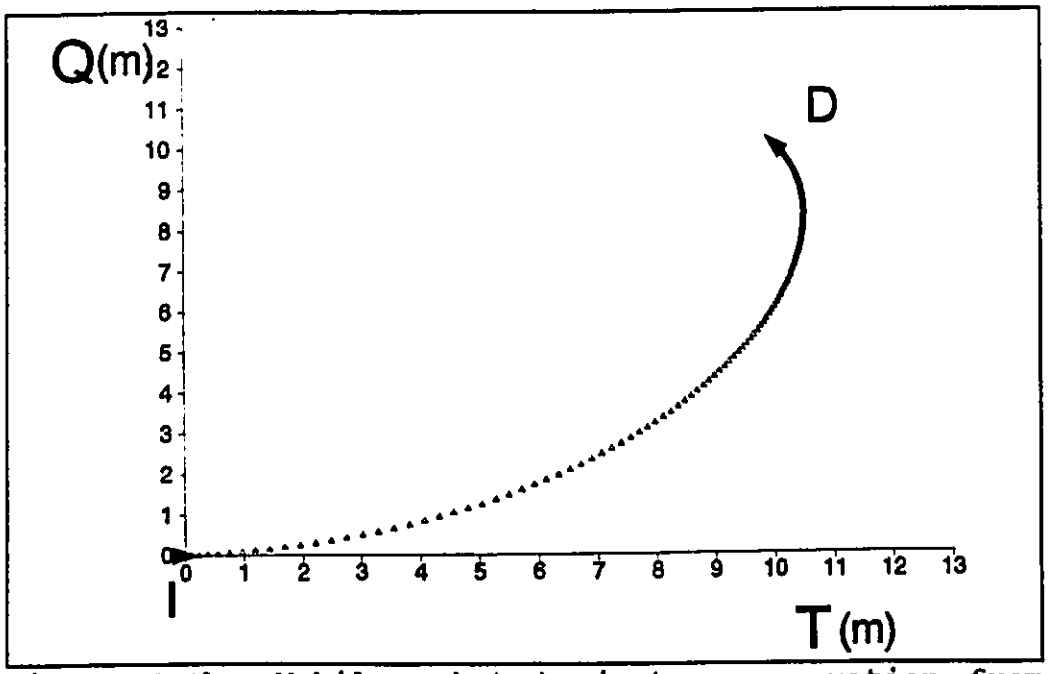

Figure 5.1b Mobile robot trajectory generation from point I(1,1) to point D(10,10) with realistic steering controller.

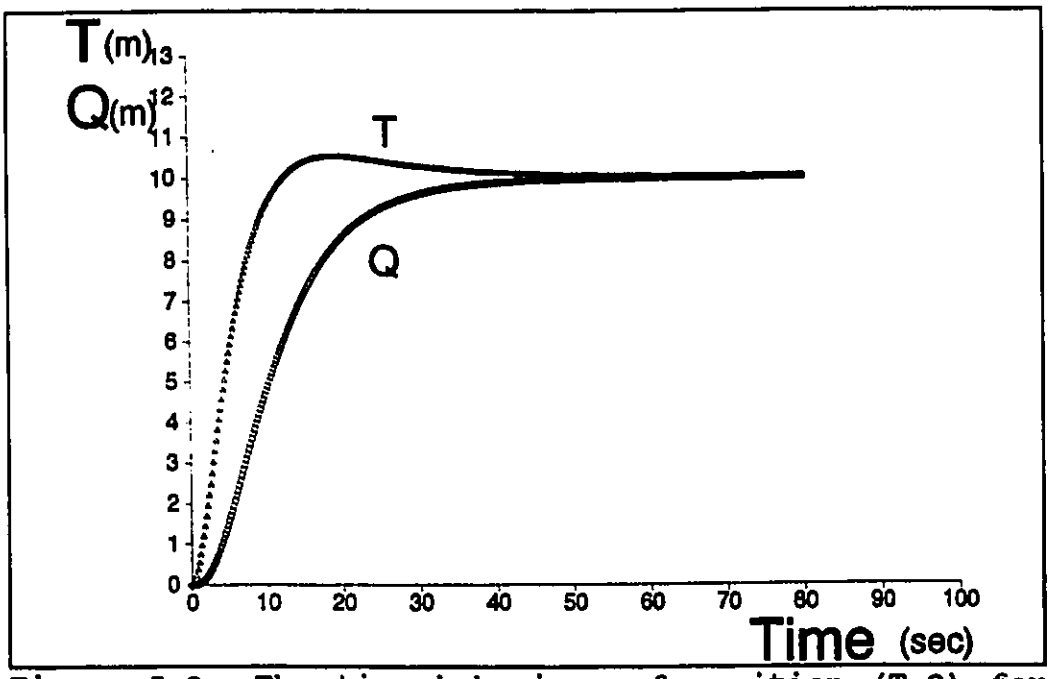

Figure 5.2 The time behaviour of position (T,Q) for Figure 5.1b.

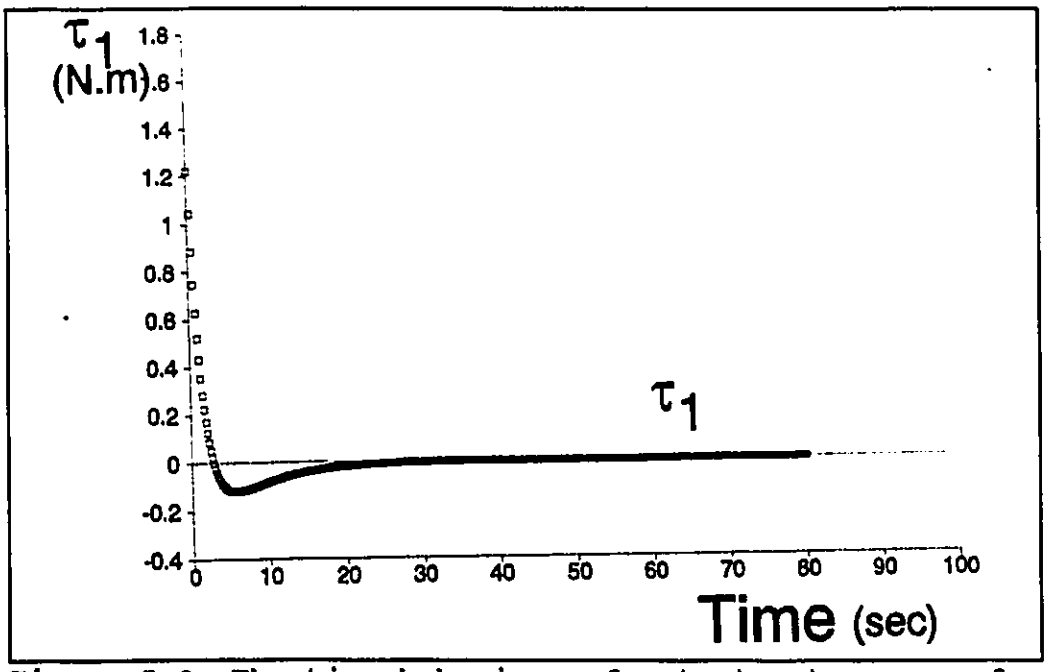

Figure 5.3 The time behaviour of actuator torque τ_1 for Figure 5.1b.

5.2 Obstacle Avoidance

Similar to the case of a jointed robot manipulator subject to an impedance controller, a Cartesian trajectory can be generated for the case of a work volume containing obstacles (Fig. 5.4a,b, Fig. 5.7a,b, Fig. 5.10a,b, and Fig. 5.13) by posing repulsive virtual impedance between the mobile robot and the obstacles. Fig. 5.4a shows the simulation result of the trajectory from I to D of a mobile robot avoiding an obstacle with a circular shape located on the path between the current robot position and the target position. Only spring constant is considered for the virtual repulsive force exerted from the obstacle and ideal steering

controller is used. The mobile robot in this case has a feasible trajectory which avoids the obstacle and reaches the target and shows fast response of the steering command. The same simulation is done in the case of Fig. 5.4b with realistic steering command and it shows a longer and wider curved trajectory. As shown in Fig. 5.4a,b, the robot controller needs only the value of the shortest distance between the obstacle and the robot at any time. Therefore, the robot could sense only the part of the obstacle, marked by small dots, rather than sensing the complete shape of the obstacle. T(t) Q(t), and $\tau_1(t)$ for the case of Fig. 5.4b is shown in Fig. 5.5, and Fig. 5.6, respectively. It takes 77.21 seconds to reach to the target position.

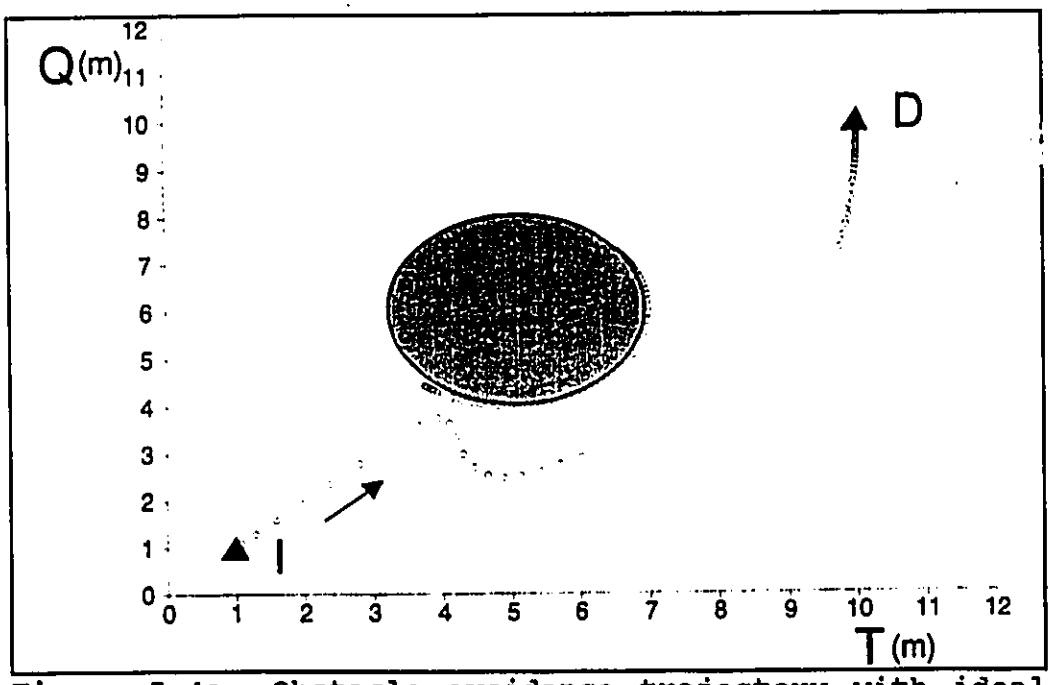

Figure 5.4a Obstacle avoidance trajectory with ideal steering controller having virtual repulsive force of K_o = 10(N.m), B_o = 0(N/ms^{-1}).

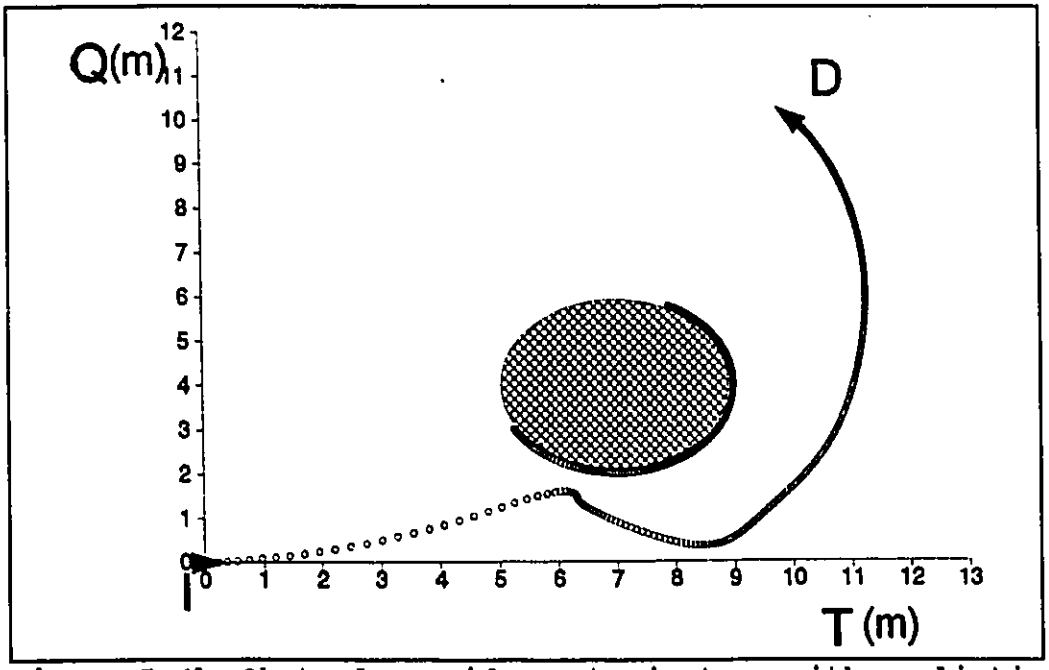

Figure 5.4b Obstacle avoidance trajectory with realistic steering controller having virtual repulsive force of K_o = 10(N.m), B_o = 0(N/ms^{-1}).

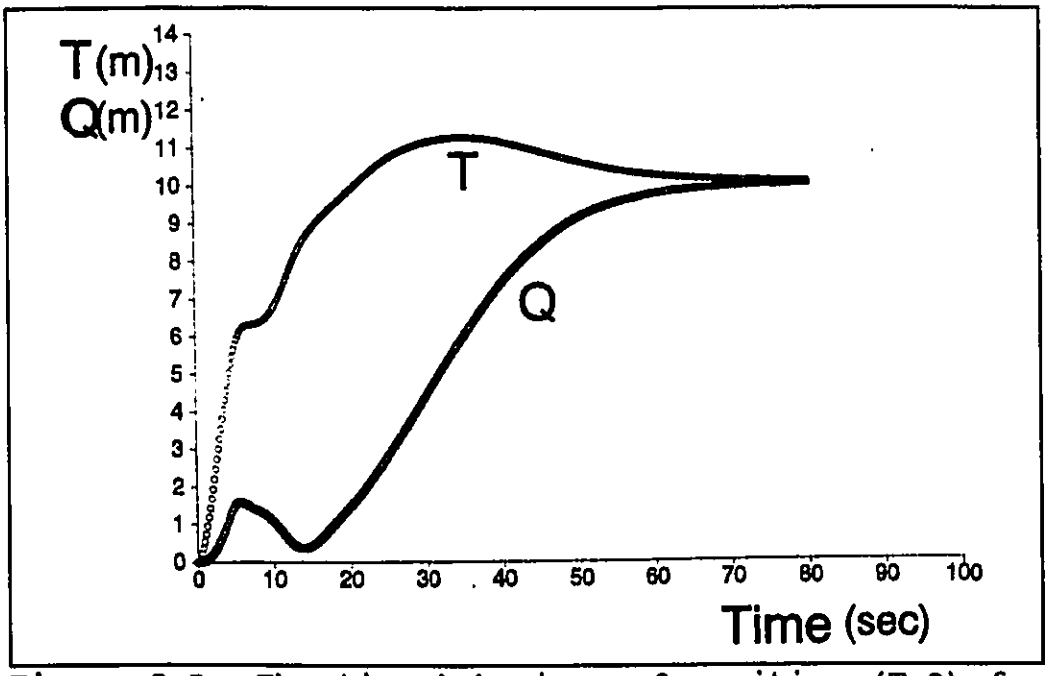

Figure 5.5 The time behaviour of position (T,Q) for Figure 5.4b.

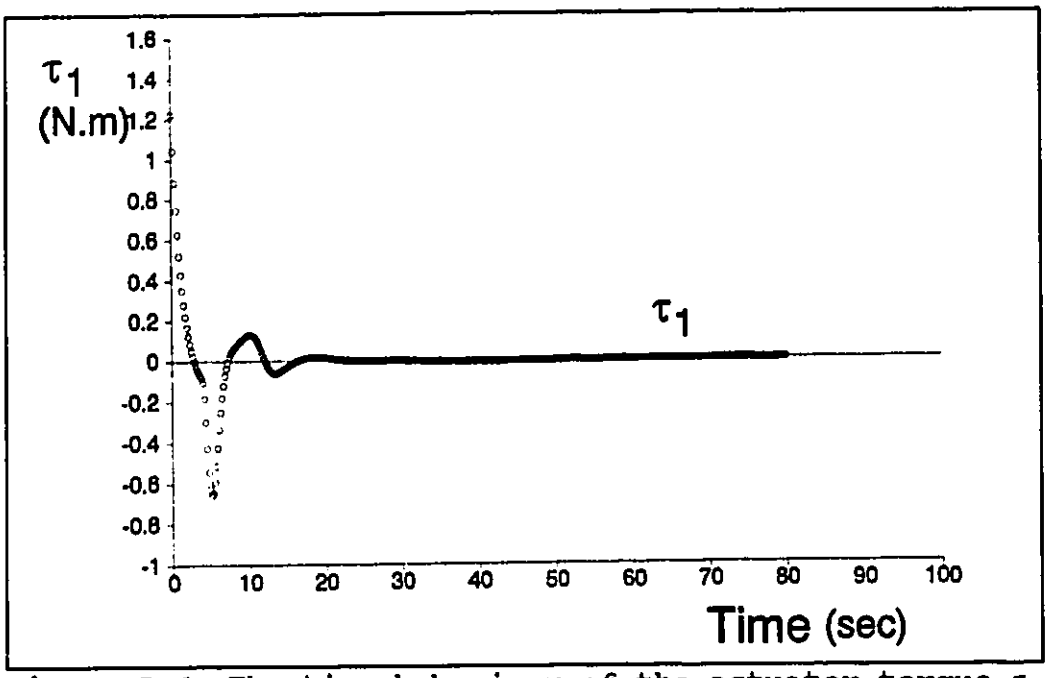

Figure 5.6 The time behaviour of the actuator torque τ_1 for Figure 5.4b.

In Fig. 5.7a,b, the damping effect is added to the virtual repulsive force and this results in a smoother trajectory, but it took more time to reach the target in comparison with the case of Fig. 5.4a,b. The non-omnidirectional effect of the mobile robot is clearly shown in Fig. 5.7b. The time taken for the case of Fig. 5.7b is 92.47 seconds. T(t) Q(t), and τ_1(t) for the case of Fig. 5.7b is shown in Fig. 5.8, and Fig. 5.9, respectively.

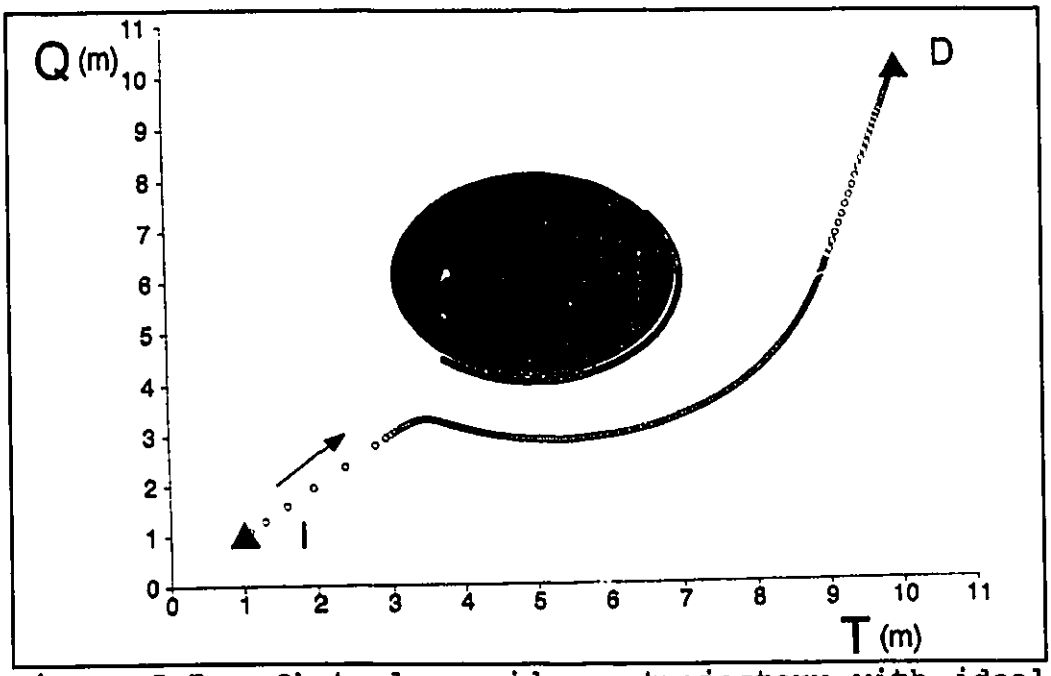

Figure 5.7a Obstacle avoidance trajectory with ideal steering command having virtual repulsive force of K_o = 10(N.m), and B_o = 1(N/ms^{-1}).

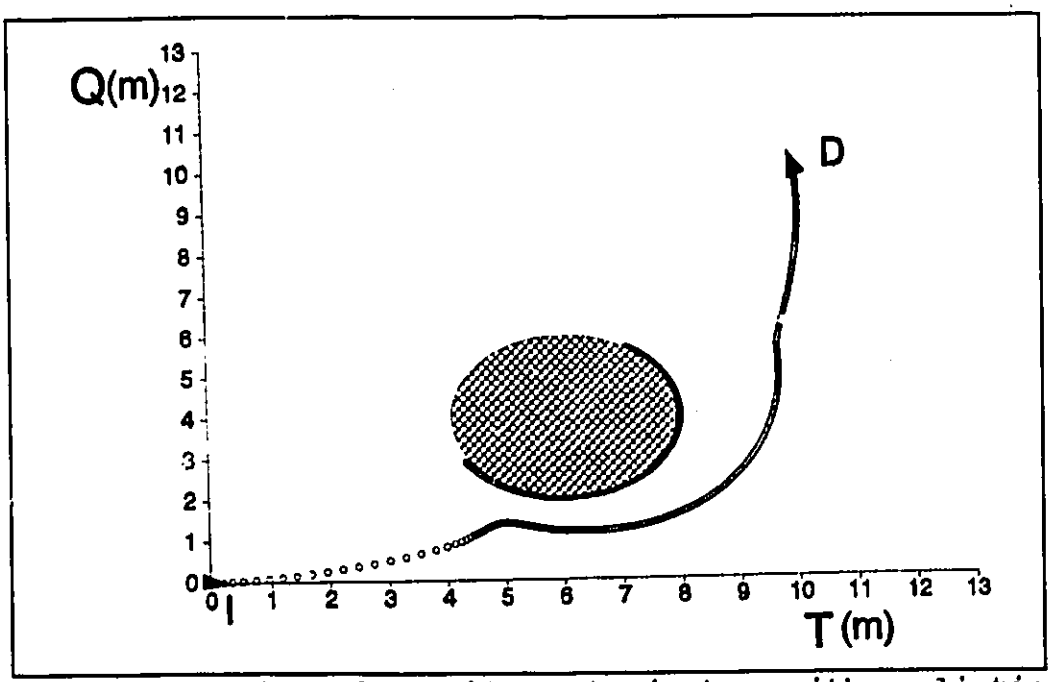

Figure 5.7b Obstacle avoidance trajectory with realistic steering command having virtual repulsive force of K_o = 10(N.m), B_o = 1(N/ms^{-1}).

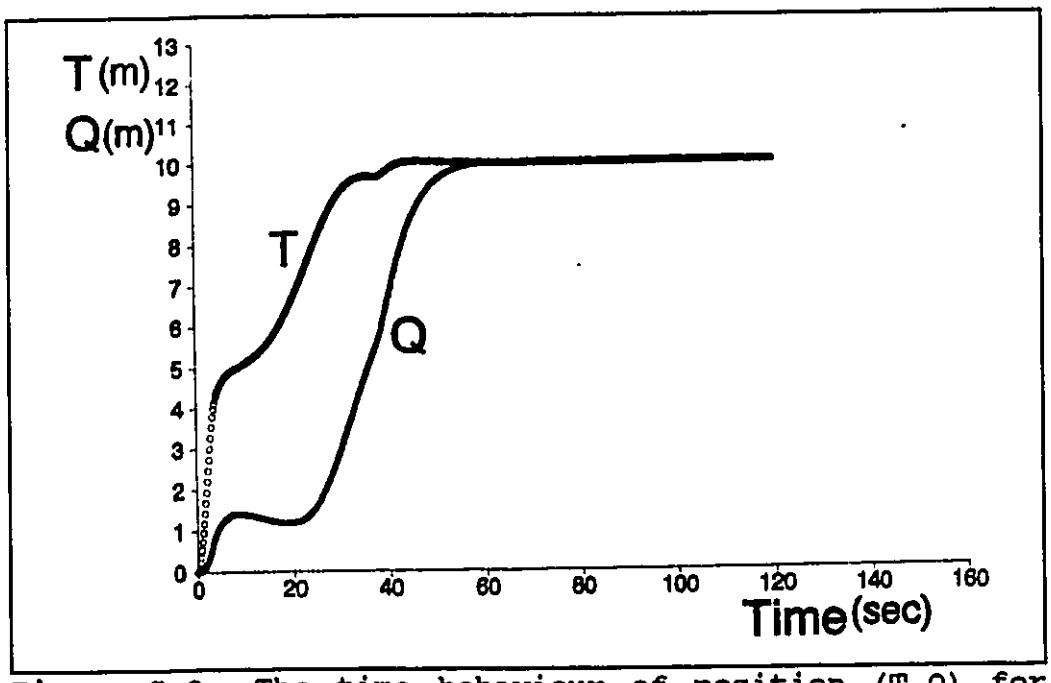

Figure 5.8 The time behaviour of position (T,Q) for Figure 5.7b.

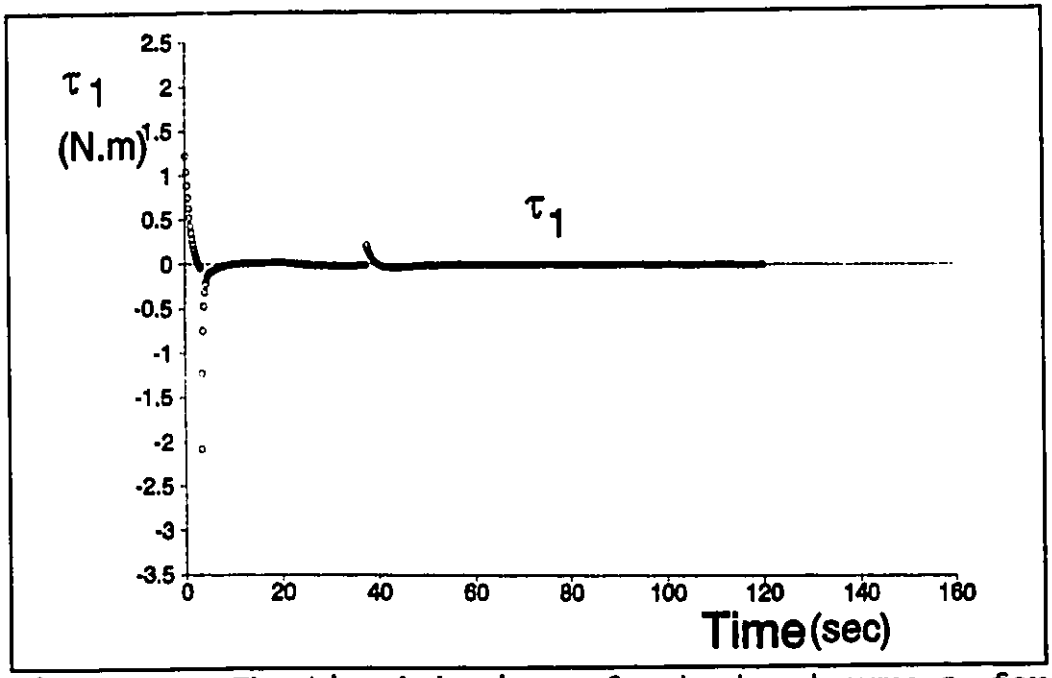

Figure 5.9 The time behaviour of actuator torque τ_1 for Figure 5.7b.

Figure 5.10a,b, and figure 5.13 present the performance of impedance controller for an obstacle cluttered environment. Figure 5.10a shows a smooth trajectory generated between the initial(I) and target(D) position with ideal steering command in three circular shaped obstacles environment. But a distorted trajectory is generated as expected with realistic steering command in figure 5.10b. $T(t)Q(t)$, and $\tau_1(t)$ for the case of Fig. 5.10b is shown in Fig. 5.11, and Fig. 5.12, respectively. The time spent to reach to the target for the case of Fig. 5.10b is 112.08 seconds. In Fig. 5.13, for the case of two obstacles at a 1m minimum distance, the controller for this non-omnidirectional mobile robot of 0.3m width with realistic steering command cannot generate a trajectory passing in between the two obstacles, while for an omnidirectional robot the controller could easily generate such a trajectory. $T(t)Q(t)$, and $\tau_1(t)$ for the case of Fig. 5.13 is shown in Fig. 5.14, and Fig. 5.15, respectively.

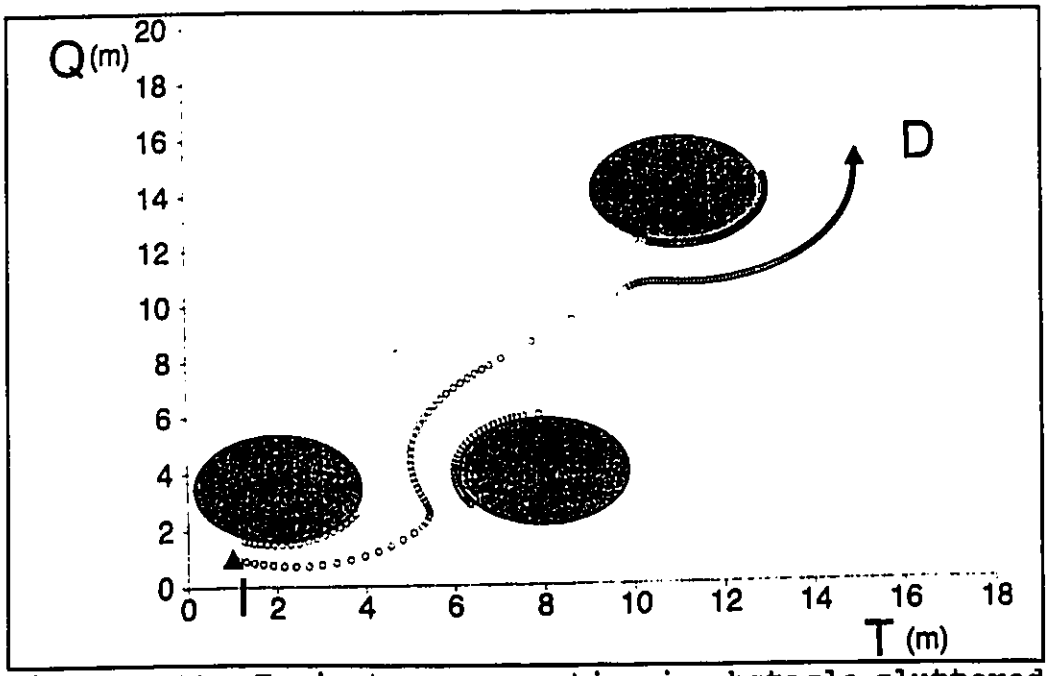

Figure 5.10a Trajectory generation in obstacle cluttered environment with ideal steering command(K_o = 10(N.m), B_o = 1(N/ms^{-1})).

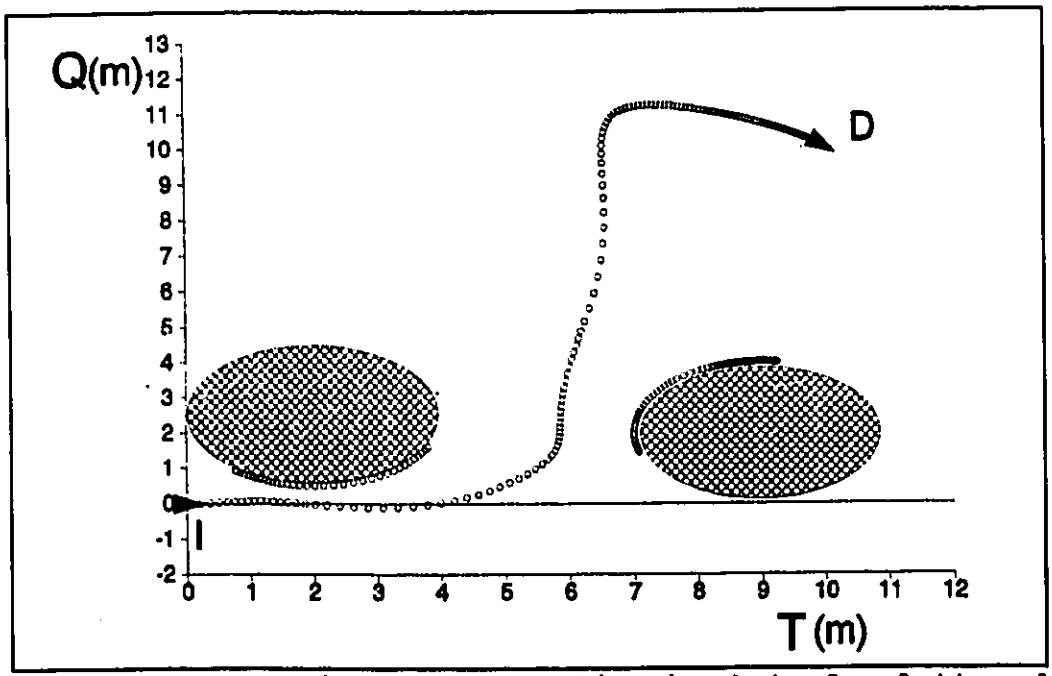

Figure 5.10b Trajectory generation in obstacle cluttered environment with realistic steering command (K_o = 10(N.m) B_o = 1(N/ms^{-1})).

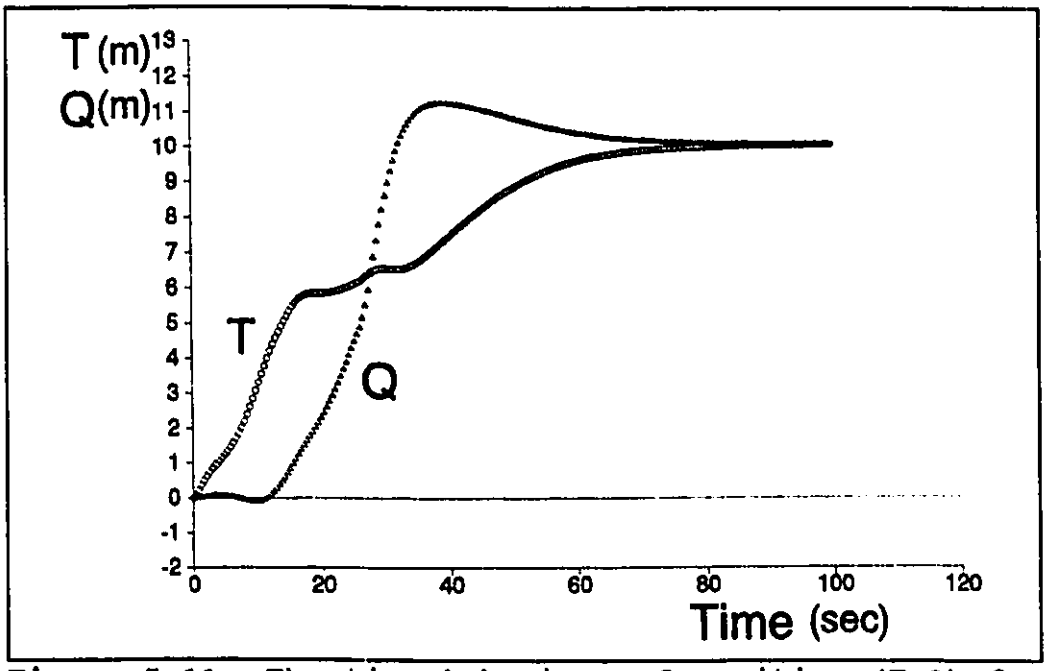

Figure 5.11 The time behaviour of position (T,Q) for Figure 5.10b.

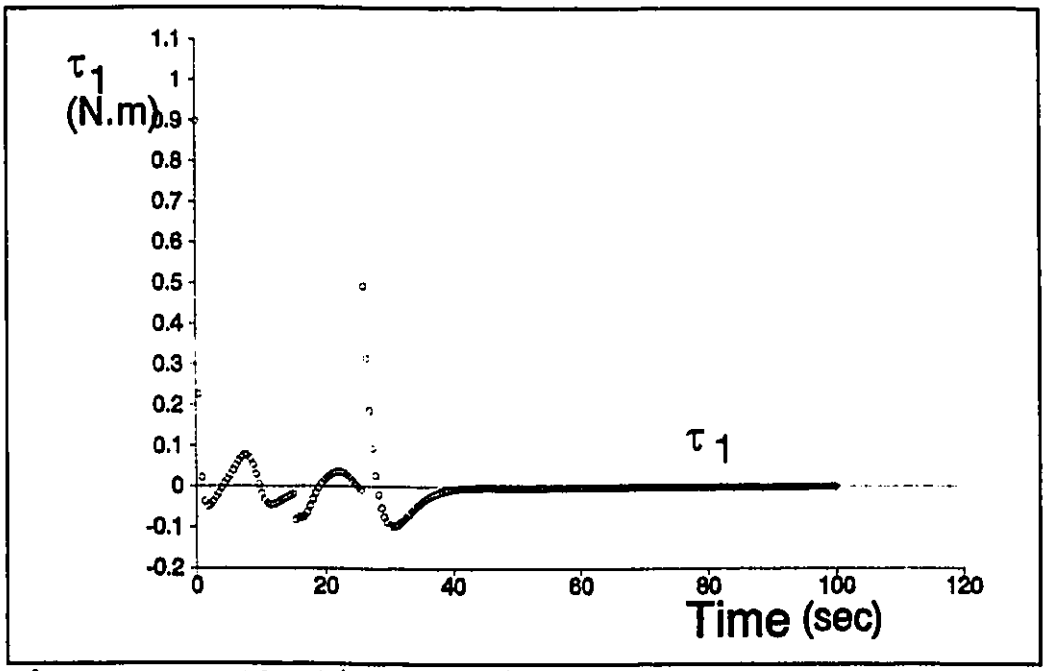

Figure 5.12 The time behaviour of actuator torque τ_1 for Figure 5.10b.

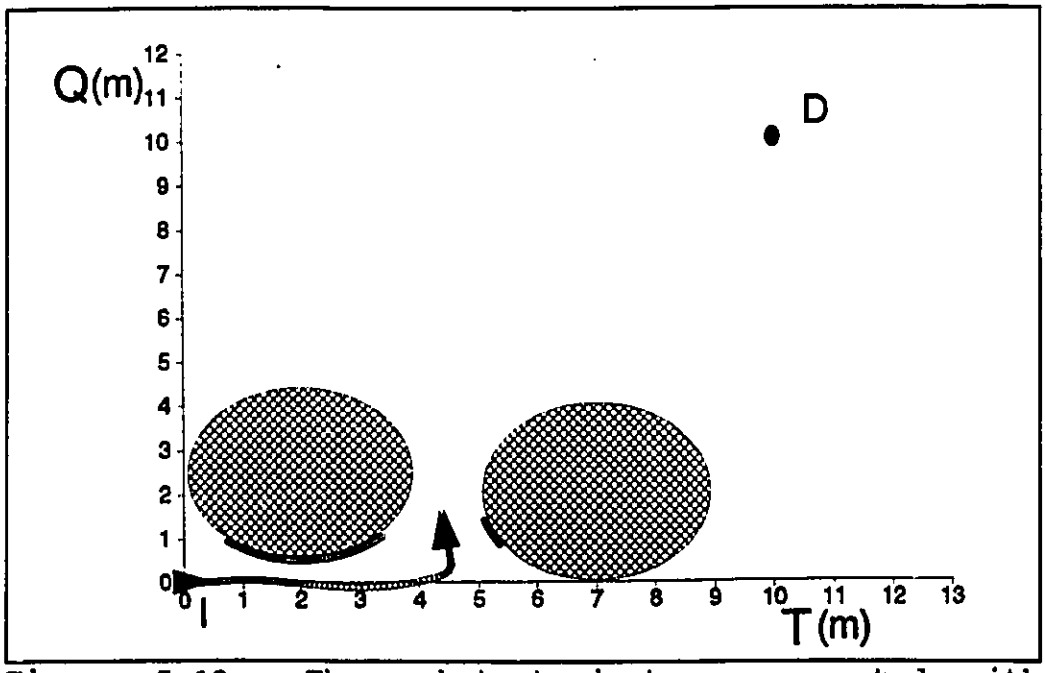

Figure 5.13 The robot trajectory generated with realistic steering command caught in between two obstacles.

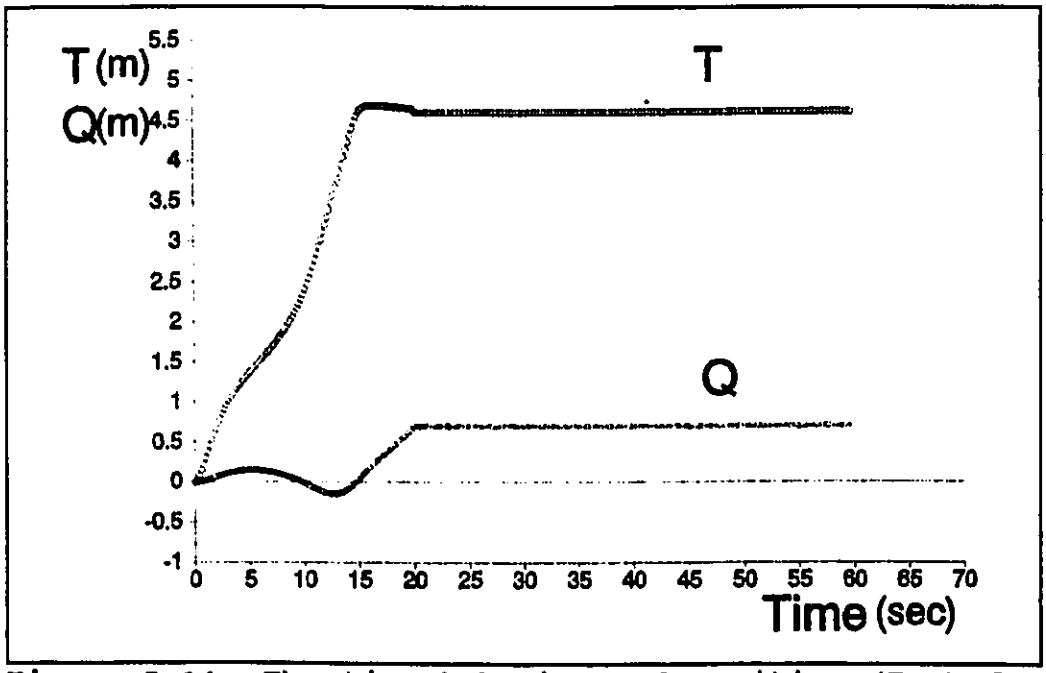

Figure 5.14 The time behaviour of position (T,Q) for Figure 5.13.

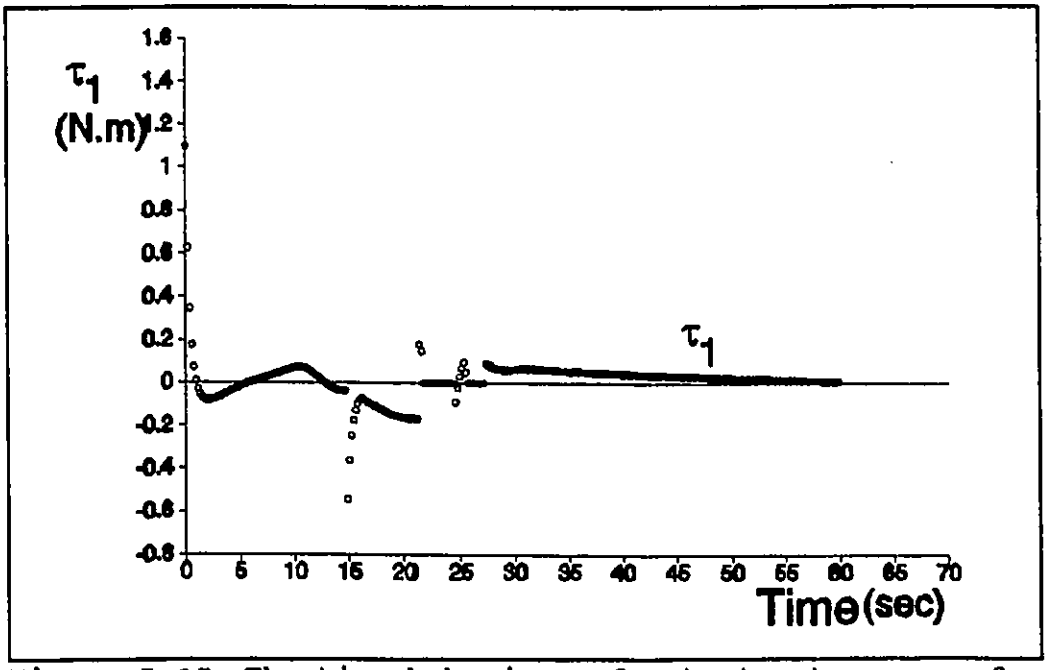

Figure 5.15 The time behaviour of actuator torque τ_1 for Figure 5.13.

VI Conclusion

The utilization of impedance control for the trajectory generation of the mobile robot and for the collision avoidance with obstacles is analyzed in this thesis. The results of the simulation analysis prove that the impedance control can be an important candidate for trajectory generation and obstacle avoidance tasks.

The kinematic and Newtonian dynamic model of the mobile robot facilitates the development of the artificial impedance based controller of a mobile robot by imposing to behave like a virtual impedance linking the centre of mass of the mobile robot to the destination position. This controller permits the generation of the trajectory in both obstacle free and obstacle avoidance cases.

The major advantage of the proposed impedance based controller for the motion of a mobile robot is the fact the trajectory, rather than preplanned, is generated and is corrected to avoid collisions with obstacles in real time. The impedance control enables a mobile robot to avoid obstacles without knowing the full geometric description of the obstacles which usually requires a complex vision system. The only information needed is the closest point of the surface of an obstacle from the mobile robot at each time, provided by a simple range sensors.

The distance between the current position and the destination position is normally communicated to the controller from a higher level of task planning while the distance to the unexpected obstacles is assumed obtained and updated in real time from range sensors. This fact gives further advantages to the impedance control in Cartesian space with regard to the incorporation of the limits imposed to the servomotor torque command. Torque saturation and other motion constraints can be incorporated in the controller using a Cartesian force rescaling scheme [Necsulescu, 1990]. The impedance control approach also permits the control of contact forces when the mobile robot perform docking and during object handling. In general, omnidirectional mobile robots perform more efficiently under an operational space controller as for example the impedance based controllers.

The contribution of this thesis is in investigating the utilization of impedance control for non-omnidirectional mobile robot. While a quasi-straight line trajectory is obtained with ideal steering controller in an obstacle free space, a curved trajectory is generated with realistic steering controller which is expected in real situation. There are initial positions and orientations of the mobile robot from which goal point can be reached in real time. Also, for similar conditions, obstacle avoidance is possible in real-time. For the case of two obstacles at a certain short distance, even if the distance between the two obstacles is wider than the width of mobile robot, the non-

omnidirectional robot with realistic steering command get caught in the middle of two obstacles. And in the case of short maneuvaring distances between the current robot position and obstacles and the goal position and for particular initial orientations, the efficiency of an impedance control of a non-omnidirectional mobile robot is reduced and is further limited by servomotors saturation torque. Non-smooth controllers (piece-wise linear, see Canudas) can be considered in such situations.

References

1. Nilsson, N.J.,"A Mobile Automation : An Application of Artificial Intelligence Techniques", Proc. of the 1st IJCAI, Washington, D.C., May, 1969

2. McGee et al,"Adaptive Locomotion of a Multilegged Robot over Rough Terrain", IEEE Trans. Syst., Man., and Cyber., Vol. SMC9, 1979

3. McKerrow, P.J.,"Introduction to Robotics", Addison Wesley, 1991

4. Shiller, Z. and Gwo, Y.R.,"Dynamic Motion Planning of Autonomous Vehicles", IEEE Trans. on Robotics and Automation, Vol. 7, No. 2, April 1991, pp 241-249

5. Sekiguchi, M. et al,"Behavior Control for a Mobile Robot by Multi-Hierarchical Nueral Network", Proc. 1989 IEEE International Conference on Robotics and Automation, Vol. 3, Scottsdale, Arizona, May 14-19, 1989, pp 1578-1583

6. Crowley, J.L.,"Asynchronous Control of Orientation and Displacement in Robot Vehicle", Proc. 1989 IEEE International Conference on Robotics and Automation, Vol. 3, Scottsdale, Arizona, May 14-19, 1989, pp 1277-1282

7. Graettinger, T.J. and Krogh, B.H.,"Evaluation and Time Scaling of Trajectories for Wheeled Mobile Robots", ASME Journal of Dynamic Systems Measurement and Control, Vol. 111, June 1989, pp 222-231

8. Ellis, J.R.,"Vehicle Dynamics", Business Books, London, 1969

9. Hatwal, H. and Mikulcik, E.C.,"Some Inverse Solutions to an Automobile Path-Tracking Problem with Input Control of Steering and Brakes", Vehicle System Dynamics, Vol. 15, 1986, pp 61-71

10. Steer, B.,"Trajectory Planning for Mobile Robot", The International Journal of Robotics Research, Vol. 8, No. 5, Oct, 1989, pp 3-14

11. Samson, C.,"Time-Varying Feedback Stabilization of A Nonholonomic Car-like Mobile Robot", The 30th IEEE Conference on Decision and Control, 1991, pp 1-24

12. Hemami, A. et al,"A New Control Strategy for Tracking in Mobile Robots and AGV's", Proc. 1990 IEEE International Conference on Robotics and Automation, Vol. 2, Cincinnati,

Ohio, May 13-18,1990, pp 1122-1127

13. Giralt, G. et al,"An Integrated Navigation and Motion Control System for Autonomous Multisensory Mobile Robots", The First International Symposium on Robotics Research, MIT Press, 1984, pp 191-214

14. Canudas de Wit, C. and Sordalen, O.J.,"Exponential Stabilization of Mobile Robots with Nonholonomic Constraints", accepted in IEEE TAC, Aug, 1992

15. Samson, C.,"Velocity and Torque Feedback Control of a Nonholonomic Cart", Proc. of the Interational Workshop on Nonlinear and Adaptive Control Issues in Robotics, Grenoble, France, Nov. 21-23, 1990, pp 125-151

16. Saha, S.K. and Angeles, J.,"Kinematics and Dynamics of a Three Wheeled AGV", Proc. 1989 IEEE International Conference on Robotics and Automation, Vol. 3, Scottsdale, Arizona, May 14-19, 1989, pp 1572-1577

17. Sordalen, O.J. and Canudas de Wit, C.,"Path Following and Stabilization of A Mobile Robot", International Workshop in Adaptive and Nonlinear Control : Issues in Robotics, Nov 21-23, 1990, Granoble, France

18. Canudas de Wit, C., and Roskam, R.,"Path following of 2-DOF Mobile Robot under Path and Input Torque Constraints", Proc. 1991 IEEE Conference on Robotics and Automation, Vol. 2, Sacramento, California, April 9-11, 1991 pp 1142-1147

19. Nelson, W.L. and Cox, I.J.,"Local Path Control for An Autonomous Vehicle", Proc. of the 1988 IEEE International Conference on Robotics and Automation, Vol. 3, Philadelphia, Penn., 1988, pp 1504-1510

20. Necsulescu, D.S. and Kim, B.,"Free Motion, Collision Avoidance and Contact Motion Control for Mobile Robots", 7th IFAC Symposium on Information Control Problems in Manufacturing Technology, Toronto, Ont., May 25-28, 1992, pp 331-336

21. Cybermation,"K2A Mobile Platform", Commercial offer, 5457 JAE Valley Road, Roanoke, Virginia 24014, 1987

22. Denning Mobile Robots, Inc.,"Securing the Future", Commercial offer, 21 Cunnings Park, Woburm, MA. 01801, 1985

23. Holland, J.M.,"A Mobile Platform for Industrial Research", Robotics Research Conference, Scottsdale, Arizona, SME Technical Paper, No. MS80-786, 1989, 13 pages

24. Borenstein, J. and Koren, Y.,"Real-time Obstacle Avoidance for

Fast Mobile Robots", IEEE Trans. on System and Cybermation, Vol. 19, No. 5, 1989, pp 1179-1187

25. Borenstein, J. and Koren, Y.,"Real-time Obstacle Avoidance for Fast Mobile Robots in Cluttered Environments", Proc. 1990 IEEE International Conference on Robotics and Automation, Vol. 1, Cincinnati, Ohio, May 13-18,1990, pp 572-577

26. Carlisle, B.,"An Omni-directional Mobile Robot", Development in Robotics, IFS, 1983, pp 79-87

27. Campion, G. et al,"External Linearization Control for a Mobile Robot", IFAC, Syroco, 1988

28. Muir, P.F. and Newman, C.P.,"Kinematic Modelling for Feedback Control of An Omnidirectional Wheeled Mobile Robot", Proc. 1987 IEEE International Conference on Robotics and Automation, Vol. 3, Raleigh, North Carolina, Mar 31- Apr 3, 1987, pp 1772-1778

29. Feng, D. et al,"The Servo-Control System for An Omnidirectional Mobile Robot", Proc. 1989 IEEE International Conference on Robotics and Automation, Vol. 3, Scottsdale, Arizona, May 14-19, 1989, pp 1566-1571

30. Daniel, D.J. et al,"Kinematics and Open-loop Control of An Illonator-based Mobile Platform", Proc. 1985 IEEE International Conference on Robotics and Automation, St.Louis, MO., Mar 1985, pp 346-353

31. Yuh, J.,"Modelling and Control of Underwater Robotic Vehicle", IEEE Trans, on Systems, Manufacturing and Cybernetics, Vol. 20, No. 6, Nov/Dec 1990, pp 1475-1483

32. Campion, G. et al,"Controllability and State Feedback Stabilization of Non-holonomic Mechanical Systems", International Workshop in Adaptive and nonlinear Control : Issues in Robotics, Nov 21-23, 1990, Granoble, France

33. Alexander, J.C. and Maddocks, J.H.,"On the Kinematics of Wheeled Mobile Robots", International Journal of Robotics Research, 1990

34. Kanayama, Y. et al,"A Stable Tracking Control Method for An Autonomous Mobile Robot", Proc. 1990 IEEE International Conference on Robotics and Automation, Vol. 1, Cincinnati, Ohio, May 13-18, 1990, pp 384-389

35. Barraquand, J. and Latombe, J.C.,"Controllability of Mobile Robots with Kinematic Constraints", Technical Roport No. STAN-CS-90-1317, Dept. of Conmputer Science, Stanford University

36. Nakamura, Y. and Mukherjee, R.,"Nonlinear Control for the Nonholonomic Motion of Space Robot Systems", International Workshop in Adaptive and Nonlinear Control: Issues in Robotics, Nov 21-23, 1990, Granoble, France .

37. Warren, C.W.,"Global Path Planning Using Artificial Potential Fields", Proc. 1989 IEEE International Conference on Robotics and Automation, Vol. 1, Scottsdale, Arizona, May 14-19, 1989, pp 316-321

38. Khatib, O.,"Real-time Obstacle Avoidance for Manipulators and Mobile Robots", The International Journal of Robotics Research, Vol. 5, No. 1, Spring 1986, pp 396-404

39. Krogh, B.H.,"A Generalized Potential Field Approach to Obstacle Avoidance Control", Proc. SME Conference Robotics Research: The next five years and Beyond, Bethlehem, PA., Aug 1984 pp 1-15

40. Krogh, B.H. and Thorpe, C.E.,"Integrated Path Planning and Dynamic Steering Control for Autonomous Vehicles", Proc. 19th IEEE International Conference on Robotics and Automation, San Francisco, CA., Apr 7-10 1986, pp 1664-1669

41. Brooks, R.A.,"A Robust Layered Control System for A Mobile Robot", IEEE Journal on Robotics and Automation, Vol. RA-2, No.1, 1986, pp 14-23

42. Arkin, R.C.,"Motor Schema-Based Mobile Robot Navigation", The International Journal of Robotics Research, Aug 1989, pp 92-112

43. Hogan, N.,"Impedance Control: An Approach to Manipulation", Proc. Numerical Control Conference, San Diago, California, June 6-8, 1984, pp 304-313

44. Hogan, N.,"Stable Execution of Contact Tasks Using Impedance Control", Proc. of the IEEE Intarnational Conference on Robotics and Automation, Vol. 1, 1987, pp 1047-1054

45. Andrews, J.R.,"Impedance Control as a Framework for Implementing Obstacle Avoidance in a Manipulator", Master's Thesis, MIT, Feb 1983

46. Necsulescu, D.S. et al,"Impedance Control for Robotic Manipulation", Second workshop on Military Robotics Applications, Aug 8-11, 1989, RMC, Kingston, Canada

47. Wilfong, G.T.,"Autonomous Robot Vehicles", Springer-Verlag, 1990

48. Anand, D.K. and Cuniff, P.F.,"Engineering Mechanics, Dynamics.", Allyn and Bacon, 1984

49. Necsulescu, D.S.,"Artificial Impedance Approach of the Trajectory Generation and Collision Avoidance for Single and Dual Arm Robots", Proc. International Workshop on Adaptive and Nonlinear Control, Grenoble, Springer, Verlag, 1990

50. Shilling, R. ,"Fundamentals of Robotics", Prentice Hall

51. Frank D'Souza, A. and Greg, V.K.,"Advanced Dynamics: Modeling and Analysis", Prentice Hall, 1984

APPENDIX A

The proof of equation (3.55)

Consider the bar between wheel #2 and wheel #3.

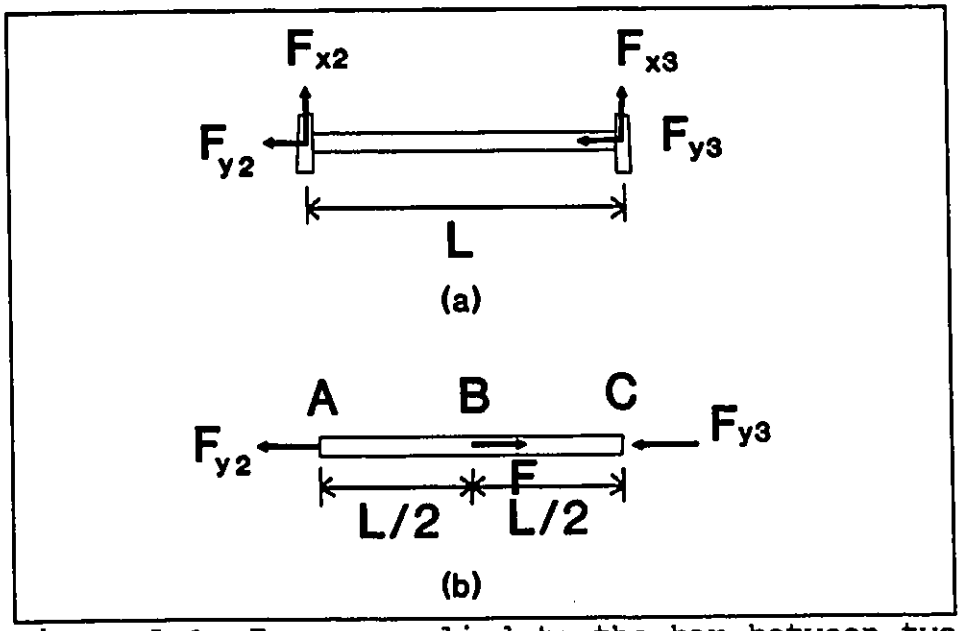

Figure A-1 Forces applied to the bar between two rear wheels: (a)force relation at the bar, (b) free body diagram of the bar

From the free body diagram, the equilibrium equation of the y-component of the forces has the relation as,

$$F = F_{y2} + F_{y3} \tag{A-1}$$

There are two unknown reactions and only one equilibrium equation; therefore, the problem is statically indeterminate. Consequently, we must also consider the deformations. Since we consider the bar as a rigid body, the total elongation of the bar must be zero. Thus, the compatibility condition is,

$$\delta = \delta_{AB} + \delta_{AB} = 0 \qquad \text{(A-2)}$$

From equation (A-2), we have

$$(1/2)F_{y2} + [-(F-F_{y2})(1/2)] = 0 \qquad \text{(A-3)}$$

so that

$$F_{y2} = F/2 \qquad \text{(A-4)}$$

After we insert equation(A-4) into equation(A-3), we have

$$F_{y3} = F/2 \qquad \text{(A-5)}$$

Therefore,

$$F_{y2} = F_{y3} \qquad \text{(A-6)}$$

Appendix B

The equations of block A-D of the block diagram

Block A Inverse Kinematics

Inputs : A_T, A_Q, V_T, V_Q

Outputs : ω_1, ω_2, ω_3, $\dot{\theta}$, δ

a_x, a_y and V_x, V_y are obtained from A_T, A_Q and V_T, V_Q, respectively, using the transformation derived from Eq.(3.1) for $\dot{\theta}=\ddot{\theta}=0$. We assume no slip conditions. The rear wheels cannot rotate with respect to the robot frame plane, and these wheels and frame have the same orientation. From Eq.(3.25) and (3.29) we obtain,

$$\dot{\theta} = V_y/c$$

$$\omega_2 = (V_x - (1/2)\dot{\theta})/r_{w2}$$

$$\omega_3 = (V_x + (1/2)\dot{\theta})/r_{w3}$$

$$\omega_1 = \{V_x^2 + [V_y + (b-c)\dot{\theta}]^2\}^{1/2} / r_{w1}$$

$$\delta = \tan^{-1}([V_y + (b-c)\dot{\theta}] / V_x)$$

Block B Inverse Dynamics

Inputs: a_x, a_y, ω_1, ω_2, ω_3, $\dot{\theta}$, δ, F_{ext}, α

outputs: τ_1, F_{xi}, F_{yi}, G_{xi}, G_{yi} (i=1,2,3)

The inverse dynamics presented for a three wheel mobile robot is obtained from Eqs. (3.46 - 3.55) and is presented here in a format similar to the one used by (Graettinger and Krogh, 1989) for the simplified case of an equivalent two wheel mobile robot.

From Eq.(3.55), $F_{Y2} \approx F_{Y3}$, and Eq. (3.46), (3.47) and considering F_{ext} applied under an angle α at the tip of the frame , we obtain

$$F_{x1}\cos\delta - F_{y1}\sin\delta + F_{x2} + F_{x3} = ma_x + F_{ext}\cos\alpha$$

$$F_{x1}\sin\delta + F_{y1}\cos\delta + 2F_{y2} = ma_y + F_{ext}\sin\alpha$$

For $F_{ext} = 0$, eq. (3.36)-(3.54) give

$$(q_1\cos\delta - q_4\sin\delta)F_{x1} - (q_1\sin\delta + q_4\cos\delta)F_{y1} + (q_3+q_8)F_{x2} + 2q_2F_{y2} + (q_1-q_3)F_{x3} = -c\dot{\theta}^2$$

$$(q_{10}\sin\delta)F_{x1} + (q_{10}\cos\delta)F_{y1} + q_6F_{x2} + 2(q_1+q_5)F_{y2} - q_6F_{x3} = r_{w2}\omega_2\dot{\theta} + (1/2)\dot{\theta}^2$$

$$(q_1\cos\delta + q_4\sin\delta)F_{x1} - (q_1\sin\delta - q_4\cos\delta)F_{y1} + (q_1-q_3)F_{x2} - 2q_2F_{y2} + (q_3+q_8)F_{x3} = -c\dot{\theta}^2$$

The torque command for the front wheel is given by

$$\tau_1 = [(q_1+q_{13}+q_{12}\sin^2\delta)F_{x1} + (q_{12}\cos\delta\sin\delta)F_{y1} + (q_1\cos\delta - q_{11}\sin\delta)F_{x2} + 2q_{10}\sin\delta F_{y2} + (q_1\cos\delta + q_{11}\sin\delta)F_{x3} - (b-c)\cos\delta\dot{\theta}^2] / q_{14}$$

The frictional forces for the three wheels are given by

$$G_{x1} = [\, q_{28}\,(\, q_1 + q_{13} + q_{12}\sin^2\delta\,) + q_{18}\,]\, F_{x1} + (\, q_{28}\, q_{12}\, \sin\delta\, \cos\delta\,)\, F_{y1} + [q_{28}\,(\, q_1\, \cos\delta - q_{11}\, \sin\delta\,)]\, F_{x2} + (\, 2q_{28}\, q_{10}\, \sin\delta\,)\, F_{y2} + [\, q_{28}\,(\, q_1\, \cos\delta + q_{11}\, \sin\delta\,)\,]\, F_{x3} - q_{28}\,(\, b-c\,)\,\dot{\theta}^2\, \cos\delta$$

$$G_{y1} = (q_{27}\, q_{12}\, \sin\delta\, \cos\delta)\, F_{x1}$$
$$+ (1 + q_1\, q_{27} + q_{12}\, q_{27}\, \cos^2\delta)\, F_{y1}$$
$$- [q_{27} (q_1\, \sin\delta + q_{11}\, \cos\delta)\, F_{x2}$$
$$+ (2\, q_{27}\, q_{10}\, \cos\delta)\, F_{y2}$$
$$- [q_{27} (q_1\, \sin\delta + q_{11}\, \cos\delta)]\, F_{x3} + [q_{27}\, (b-c)\, \dot{\theta}^2\, \sin\delta]$$

$$G_{x2} = q_{16}\, F_{x2}$$

$$G_{y2} = F_{y2} + q_{26}\, r_{w2}\, \omega_2\, \dot{\theta}$$

$$G_{x3} = q_{16}\, F_{x3}$$

$$G_{y3} = F_{y3} + q_{26}\, r_{w2}\, \omega_2\, \dot{\theta} + m_{w2} l \dot{\theta}^2$$

The following notations are used in the inverse dynamics equations:

$q_1 = 1/m$

$q_2 = (cl)/(2Icm)$

$q_3 = (l/2)^2/Icm$

$q_4 = (b-c)(l/2)/Icm$

$q_5 = c^2/Icm$

$q_6 = q_2$

$q_7 = r_{w2}^2/(I_{w2} + m_{w2}\, r_{w2}^2)$

$q_8 = q_1 + q_7$

$q_9 = c(b-c)/Icm$

$q_{10} = q_1 - q_9$

$q_{11} = (b-c)(l/2)/Icm$

$q_{12} = (b-c)^2/Icm$

$q_{13} = r_{w1}^2/(I_{w1} + m_{w1}\, r_{w1}^2)$

$q_{14} = q_{13}/r_{w1}$

$q_{15} = m_{w1}r_{w1}\ /(I_{w1} + m_{w1}\, r_{w1}^2)$

$q_{16} = I_{w2}/(I_{w2} + m_{w2}\, r_{w2}^2)$

$q_{17} = m_{w2}r_{w2}$

$q_{18} = I_{w1}/(I_{w1} + m_{w1}\, r_{w1}^2)$

$q_{19} = m_{w1}r_{w1}$

$q_{26} = q_{17}/r_{w2}$

$q_{27} = q_{19}/r_{w1}$

$q_{28} = q_{15}\ /\ q_{14}$

Block C Mobile Robot System Dynamics

Inputs : τ_1, F_{xi}, F_{yi}, G_{xi}, G_{yi},(i=1,2,3), F_{ext}, α, δ

Outputs : a_x, a_y, $\ddot{\theta}$, $\dot{\omega}_1$, $\dot{\omega}_2$, $\dot{\omega}_3$

$$a_x = 1/m(F_{x2}+F_{x3}+F_{x1}\cos\delta-F_{y1}\sin\delta-F_{ext}\cos\alpha)$$

$$a_y = 1/m(2F_{y2}+F_{x1}\sin\delta+F_{y1}\cos\delta-F_{ext}\sin\alpha)$$

$$\ddot{\theta} = (1/I_{cm})[(b-c)(F_{x1}\sin\delta+F_{y1}\cos\delta-F_{ext}\sin\alpha)$$

$$-(1/2)F_{x2}+(1/2)F_{x3}-cF_{y2}-cF_{y3})$$

$$\dot{\omega}_1 = (\tau_1 - G_{x1} r_{w1}) / I_{w1}$$

$$\dot{\omega}_2 = (-G_{x2} r_{w2}) / I_{w2}$$

$$\dot{\omega}_3 = (-G_{x3} r_{w3}) / I_{w3}$$

By integration, ω_1, ω_2, ω_3 are obtained.

Block D Forward Kinematics

Inputs : ω_1, ω_2, ω_3

Outputs : V_T, V_Q, T, Q, θ, $\dot{\theta}$

$$\dot{\theta} = 1/l(\omega_3 r_{w3} - \omega_2 r_{w2})$$

$$V_x = r_{w2}\omega_2 + (l/2)\dot{\theta}$$

$$V_y = c\dot{\theta}$$

APPENDIX C

Illustration of the derivation of Eqn(3.16)-Eqn(3.21)

To encounter the effect of θ for a_{xi} and a_{yi} (i=1,2,3), let us express equation (3.12) - equation (3.15) with respect to the T,Q components of the global reference frame by the transformation equations,

$$V_{Ti} = v_{xi}\cos\theta - v_{yi}\sin\theta \qquad \text{(C-1)}$$

$$V_{Qi} = v_{xi}\sin\theta + v_{yi}\cos\theta \qquad \text{(C-2)}$$

Only for the case of deriving a_{x1} and a_{y1} are illustrated in this appendix and the rest of the accelerations are derived in similar procedures.

Differentiating equations (D-1) and (D-2) for i=1, we get,

$$A_{T1} = v_{x1}\cos\theta - v_{x1}\sin\theta\dot{\theta} - v_{y1}\sin\theta - v_{y1}\cos\theta\dot{\theta} \qquad \text{(C-3)}$$

$$A_{Q1} = v_{x1}\sin\theta + v_{x1}\cos\theta\dot{\theta} + v_{y1}\cos\theta - v_{y1}\sin\theta\dot{\theta} \qquad \text{(C-4)}$$

By the transformation to the local frame,

$$a_{x1} = A_{T1}\cos\theta + A_{Q1}\sin\theta \qquad \text{(C-5)}$$

$$a_{y1} = -A_{T1}\sin\theta + A_{Q1}\cos\theta \qquad \text{(C-6)}$$

With the velocity equations (3.12) and (3.13), executing equations (C-5) and (C-6) gives exactly same equations as appeared in equations (3.16) and (3.17) in the text.

Appendix D

Computer codes

```
/*
* TRAJ.C
* Impedance control for mobile robot.
*
*/

#include <stdio.h>
#include <stdlib.h>
#include <math.h>

#define L (0.3)
#define B (0.4)
#define C (B*(2.0/3.0))
#define Rw1 (0.07)
#define Rw2 (0.1)
#define Rw3 (0.1)
#define M (20.0)
#define mw1  (0.6)
#define mw2  (0.8)
#define mw3  (0.8)
#define Icm  (0.5*(M*(B-C)*(B-C)))
#define Iw1  (0.5*(mw1*Rw1*Rw1))
#define Iw2  (0.5*(mw2*Rw2*Rw2))
#define Iw3  (0.5*(mw3*Rw3*Rw3))
#define Jw1  (0.3333*(mw1*Rw1*Rw1))

/* function prototype */
void invkin(float,float,float,float,float *,float *,float *,float
            *,float *);
void invdyn(float,float,float,float,float,float *,float *,float
            *,float *,float *,float *,float *,float *,float
            *,float *,float *,float *);
void robot(float,float,float,float,float,float,float,float,
           float, float,float,float,float,float,float *,float
           *,float *,float *,float *,float *);
void forkin(float,float,float,float,float,float *,float *,float
            *,float *);

void main(void)
{

     FILE *fptr;
     FILE *pt;
     FILE *qt;
     FILE *rt;
     FILE *st;
     float x0,y0,x,y,X,Y,xd,yd,Vx0,Vy0,Vx,Vy,Vxd,Vyd,Ax,Ay,
           vx,vy,ax,ay;
     float ohm1,ohm2,ohm3,dohm1,dohm2,dohm3,qk,dqk,ddqk,
           del,dela,ddel,dddel;
     float TMAX,kp,kv,ks,bs,dt,t;
     float Fx1,Fy1,Fx2,Fy2,Fx3,Fy3,Gx1,Gy1,Gx2,Gy2,Gx3,Gy3,Tor;
     int i, count,k,kmax;
```

```
      fptr = fopen("traj.dat", "w");
      pt = fopen("xytime.dat","w");
      qt = fopen("tortime.dat","w");
      rt = fopen("qktime.dat","w");
      st = fopen("deltime.dat","w");
/*  initialization */
      x0 = .0;              /* initial position of robot */
      y0 = .0;
      Vx0 = .0;             /* initial velocity of robot */
      Vy0 = .0;
      qk = .0;              /* initial orientation of robot */
      dqk = .0;    /* initial angular velocity of robot c. of m. */
      del =.0;
      ddel = .0;
/* desired values */
      xd = 10.0;
      yd = 10.0;
      Vxd = .0;
      Vyd = .0;

      count = 1;
      printf("\nSIMULATION TIME ? ");
      scanf("%f",&TMAX);
      printf("\nSpring constant,KP = ");
      scanf("%f",&kp);
      printf("\nDamping constant,KV = ");
      scanf("%f",&kv);
      printf("\nDamping constant, KV = %.4f", kv);
      printf("\nSampling Time, dt = ");
      scanf("%f",&dt);
      printf("\nSteering Spring constant, KS = ");
      scanf("%f",&ks);

      kmax = (int)(TMAX/dt);
      printf("\nThe maximum count : %d",kmax);
      X = x0;
      Y = y0;
      Vx = Vx0;
      Vy = Vy0;
      Ax = 0.0;
      Ay = 0.0;

      for (k=1; k<=kmax; k++)
        {
        t = k*dt;    /* before invkin() put control law */
        Ax = (kp/M)*(xd-X)+(kv/M)*(Vxd-Vx);
        Ay = (kp/M)*(yd-Y)+(kv/M)*(Vyd-Vy);
        vx = Vx*cos(qk) + Vy*sin(qk);
        vy = Vy*cos(qk) - Vx*sin(qk);
        ax = Ax*cos(qk) + Ay*sin(qk);
        ay = Ay*cos(qk) - Ax*sin(qk);
        invkin(vx,vy,ax,ay,&ohm1,&ohm2,&ohm3,&dqk,&dela);
```

```
            bs = 5.*sqrt(ks*Jw1);
            dddel = (ks/Jw1)*(dela-del) - (bs/Jw1)*ddel;
            ddel = ddel+dddel*dt;
            del = del+ddel*dt+0.5*dddel*dt*dt;
          invdyn(ax,ay,ohm2,dqk,del,&Fx1,&Fy1,&Fx2,&Fy2,&Fx3,
              &Fy3,&Gx1,&Gy1,&Gx2,&Gy2,&Gx3,&Gy3,&Tor);
          robot(Fx1,Fy1,Fx2,Fy2,Fx3,Fy3,Gx1,Gy1,Gx2,Gy2,Gx3,Gy3,
                Tor,del,&ddqk,&dohm1,&dohm2,&dohm3,&ax,&ay);
          ohm1 = ohm1 + dohm1*dt;
          ohm2 = ohm2 + dohm2*dt;
          ohm3 = ohm3 + dohm3*dt;
          forkin(ohm1,ohm2,ohm3,ay,del,&dqk,&ddqk,&vx,&vy);
          vx = vx + ax*dt;
          vy = vy + ay*dt;
          x = x + vx*dt + 0.5*ax*dt*dt;
          y = y + vy*dt + 0.5*ay*dt*dt;

          qk = qk + dqk*dt + 0.5*ddqk*dt*dt;
          Vx = vx*cos(qk) - vy*sin(qk);
          Vy = vx*sin(qk) + vy*cos(qk);
          Ax = ax*cos(qk) - ay*sin(qk);
          Ay = ax*sin(qk) + ay*cos(qk);
          X = x*cos(qk) - y*sin(qk) + x0;
          Y = x*sin(qk) + y*cos(qk) + y0;

         if (count == 20)
               count = 1;
         if ( count == 1)
              {
              printf("%14.5f,%14.5f,%14.5f,%14.5f,%14.5f\n"
                  , X, Y,qk*180/3.141592,del*180/3.141592,t);
              fprintf(fptr,"%14.5f %14.5f\n", X, Y);
              fprintf(pt,"%14.5f %14.5f %14.5f\n", t, X, Y);
              fprintf(qt,"%14.5f %14.5f\n", t, Tor);
              }
          count++;

          }
          fclose(fptr);
}

/*
 * Block A
 * function invkin()
 * This program executes inverse kinematics
 * given x_acc, y_acc, x_vel, y_vel by the control law.
 *
 */

void invkin(float vx,float vy,float ax, float ay,float
            *ohm1,float *ohm2,float *ohm3,float *dqk,float *dela)
 {
```

```
      float co,si;
      float dohm1,dohm2,dohm3;
      *dqk = vy/C;
      *ohm2 = (vx-0.5*L*(*dqk))/Rw2;
      *ohm1 = sqrt(vx*vx + (vy+(B-C)*(*dqk)) *
              (vy+(B-C)*(*dqk)))/Rw1;
      *ohm3 = (vx+0.5*L*(*dqk))/Rw3;
      if (vx == 0.0 && vy == 0.0)
        *dela = .0;
      else
      *dela=atan2((vy)*B,(vx)*C);
      si = sin(*dela);
      co = cos(*dela);
}


/*
 * Block B in algorithm
 * Invdyn.c
 * ********************************************************
 * This program receives all properties of mobile robot   *
 * from kinematics and executes inverse dynamics.          *
 * Also, executes Gauss Elimination Algorithm.             *
 * ********************************************************
 * For now the properties are given.
 */


#define ROWS 5
#define COLUMNS 6



void invdyn(float ax,float ay,float ohm2,float dqk,float del,
         float *Fx1,float *Fy1,float *Fx2,float *Fy2,float *Fx3,
         float *Fy3,float *Gx1,float *Gy1,float *Gx2,float *Gy2,
         float *Gx3,float *Gy3,float *Tor)
{
    /*    FILE *fptr; */


      float a [ROWS] [COLUMNS];
      float ddel;
      float si, co, Q1, Q2, Q3, Q4, Q5, Q6, Q7, Q8, Q9, Q10, Q11,
            Q12, Q13, Q14, Q15, Q16, Q17, Q18, Q19, Q26, Q27, Q28;
      int    j, k, pospiv, l;
      float  pivot, temp, coef, sum;
      float x,y,vx,vy;
      int i,n,m;
      int noninv=0;
```

```
si = sin(del);
co = cos(del);

Q1 = 1/M;
Q2 = (C*L)/(2*Icm);
Q3 = (L*L)/(4*Icm);
Q4 = ((B-C)*L)/(2*Icm);
Q5 = (C*C)/Icm;
Q6 = Q2;
Q7 = (Rw2*Rw2)/(Iw2+mw2*Rw2*Rw2);
Q8 = Q1+Q7;
Q9 = (C*(B-C))/Icm;
Q10 = Q1-Q9;
Q11 = (L*(B-C))/(2*Icm);
Q12 = ((B-C)*(B-C))/Icm;
Q13 = (Rw1*Rw1)/(Iw1+mw1*Rw1*Rw1);
Q14 = Q13/Rw1;
Q15 = (mw1*Rw1)/(Iw1+mw1*Rw1*Rw1);
Q16 = Iw2/(Iw2+mw2*Rw2*Rw2);
Q17 = mw2*Rw2;
Q18 = Iw1/(Iw1+mw1*Rw1*Rw1);
Q19 = mw1*Rw1;
Q26 = Q17/Rw2;
Q27 = Q19/Rw1;
Q28 = Q15/Q14;


a[C][0] = co;
a[0][1] = -si;
a[0][2] = 1.0;
a[0][3] = 0.0;
a[0][4] = 1.0;
a[0][5] = M*ax;
a[1][0] = si;
a[1][1] = co;
a[1][2] = 0.0;
a[1][3] = 2.0;
a[1][4] = 0.0;
a[1][5] = M*ay;
a[2][0] = Q1*co-Q4*si;
a[2][1] = -Q1*si-Q4*co;
a[2][2] = Q3+Q8;
a[2][3] = 2*Q2;
a[2][4] = Q1-Q3;
a[2][5] = -C*dqk*dqk;
a[3][0] = Q10*si;
a[3][1] = Q10*co;
a[3][2] = Q6;
a[3][3] = 2*(Q1+Q5);
a[3][4] = -Q6;
a[3][5] = Rw2*ohm2*dqk+(L/2.0)*dqk*dqk;
a[4][0] = Q1*co+Q4*si;
```

```
 a[4][1] = Q4*co-Q1*si;
 a[4][2] = Q1-Q3;
 a[4][3] = -2*Q2;
 a[4][4] = Q8+Q3;
 a[4][5] = -C*dqk*dqk;
/*******************************
 * Gauss Elimination Algorithm *
 *******************************/

    i=0;
     while (noninv==0 && i < ROWS-1)
            {
            x=a[i][i];
            pivot = fabs(x);
            pospiv = i;
            for ( l=i+1; l<ROWS; l++)
              {
              y=a[l][i];
              if (fabs(y) > pivot)
                 {
                 pivot = fabs(y);
                 pospiv = l;
                 }
              }
            if (pivot > 0)
                 {
                 if ( pospiv != i)
                   {
                   for (k=i; k<=ROWS; k++)
                     {
                     temp = a[i][k];
                     a[i][k] = a[pospiv][k];
                     a[pospiv][k] = temp;
                     }
                   }

                 for (j=i+1; j<ROWS; j++)
                   {
                   coef = a[j][i]/a[i][i];
                   a[j][i]=0;
                   for (k=i+1; k<=ROWS; k++)
                     a[j][k] = a[j][k] - a[i][k]*coef;
                   }
                 }
            else
                 noninv = 1;
            ++i;
            }

 /* reduction */
          if (a[ROWS-1][ROWS-1] != 0)
             a[ROWS-1][ROWS] = a[ROWS-1][ROWS]/a[ROWS-1][ROWS-1];
```

```
          else
            noninv = 1;
          if ( noninv == 0)
            {
            for (i=ROWS-2; i>=0;i--)
                {
                 sum = 0;
                 for (j=i+1; j<ROWS; j++)
                   {
                   sum = sum + a[i][j]*a[j][ROWS];
                   }
                 a[i][ROWS] = (a[i][ROWS] - sum)/a[i][i];
                      }
          }
/*************************************************************/
          for (i=0; i<ROWS; i++)
             {
             if (i == 0)
               {
               *Fx1 = a[i][ROWS];
               }
             else if (i == 1)
               {
               *Fy1 = a[i][ROWS];
               }
             else if (i == 2)
               {
               *Fx2 = a[i][ROWS];
               }
             else if (i == 3)
               {
               *Fy2 = a[i][ROWS];
               }
             else if (i == 4)
               {
               *Fx3 = a[i][ROWS];
               }
             }
                          *Fy3 = *Fy2;
/*************************************************************
  Calculation of friction forces
 *************************************************************/

*Gx1 = (Q28*(Q1+Q13+Q12*si*si)+Q18)*(*Fx1) +Q28*Q12*si*co*(*Fy1)
      + Q28*(Q1*co-Q11*si)*(*Fx2) + 2.0*Q28*Q10*si*(*Fy2)
      + Q28*(Q1*co+Q11*si)*(*Fx3) - Q28*(B-C)*dqk*dqk*co ;
*Gy1 = Q27*(Q12*si*co)*(*Fx1) + (1+Q1*Q27+Q27*Q12*co*co)*(*Fy1)
     -Q27*(Q1*si+Q11*co)*(*Fx2) + 2.0*Q27*Q10*co*(*Fy2)
     -Q27*(Q1*si-Q11*co)*(*Fx3) + Q27*(B-C)*si*dqk*dqk;
*Gx2 = Q16*(*Fx2);
*Gy2 = (*Fy2) + Q26*Rw2*ohm2*dqk;
*Gx3 = Q16*(*Fx3);
```

```
*Gy3 = (*Fy2) + Q26*Rw2*ohm2*dqk + mw2*L*dqk*dqk;

*Tor = ((Q1+Q13+Q12*si*si)*(*Fx1) + (Q12*co*si)*(*Fy1)
       + (Q1*co-Q11*si)*(*Fx2) + 2.0*Q10*si*(*Fy2)
       + (Q1*co+Q11*si)*(*Fx3) - (B-C)*co*dqk*dqk)/Q14;

}

/*
 * Block C
 * System.c
 * This program executes forward dynamics
 * Inputs : dela, ddel, Fx1, Fy1, Fx2, Fy2, Fx3, Fy3, Tor.
 *          Gx1, Gy1, Gx2, Gy2, Gx3, Gy3
 * Outputs : ohm1, ohm2, ohm3, x, y, the.
 *
 */


void robot(float Fx1,float Fy1,float Fx2,float Fy2,float  Fx3,
           float Fy3,float Gx1,float Gy1,float Gx2,float Gy2,
           float Gx3,float Gy3,float Tor,float del,float *ddqk,
           float *dohm1,float *dohm2,float *dohm3,float *ax,
           float *ay)
{

     float co,si;
     float Q1,Q2,Q3,Q4,Q5,Q6,Q7,Q8,Q9,Q10,Q11,Q12;
     float vx,vy;


     Q1 = 1.0/M;
     Q2 = (0.5*L*C)/Icm;
     Q3 = (0.25*L*L)/Icm;
     Q4 = (0.5*L*(B-C))/Icm;
     Q5 = C*C/Icm;
     Q6 = Q2;
     Q7 = (Rw2*Rw2)/(Iw2+mw2*Rw2*Rw2);
     Q8 = Q1+Q7;
     Q9 = (C*(B-C))/Icm;
     Q10 = Q1 - Q9;
     Q11 = (0.5*L*(B-C))/Icm;
     Q12 = ((B-C)*(B-C))/Icm;


     co = cos(del);
     si = sin(del);
     *ax = Q1*Fx2 + Q1*Fx3 + Q1*Fx1*co - Q1*Fy1*si;
     *ay = 2.0*Q1*Fy2 + Q1*Fx1*si + Q1*Fy1*co;
     *ddqk = (1.0/Icm)*((B-C)*Fx1*si + (B-C)*Fy1*co - (0.5*L*Fx2)
```

```
            + (0.5*L*Fx3)-C*Fy2-C*Fy3);
     *dohm1 = (Tor - Gx1*Rw1)/Iw1;
     *dohm2 = (-Gx2*Rw2)/Iw2;
     *dohm3 = (-Gx3*Rw3)/Iw3;
}

/*
 * Block D
 *
 * This program executes forward kinematics
 * given ohm1, ohm2, ohm3 by robot system.
 *
 */

void forkin(float ohm1,float ohm2,float ohm3,float ay,float del,
          float *dqk,float *ddqk,float *vx,float *vy)
{

     float si,co,x,y,qk;


     *dqk = (Rw3*ohm3-Rw2*ohm2)/L;
     *ddqk = (ay-0.5*L*(*dqk)*(*dqk)-Rw2*ohm2*(*dqk))/C;
     si = sin(del);
     co = cos(del);
           *vx = Rw1*ohm1*co;
           *vy = Rw1*ohm1*si - (B-C)*(*dqk);
}


/*
 * OBSK.C
 * Obstacle Avoidance of Mobile Robot using Artificial Impedance
 * Approach.
 * Repulsive spring constant only.
 */

#include <stdio.h>
#include <stdlib.h>
#include <math.h>

#define L (0.3)
#define B (0.4)
#define C (B*(2.0/3.0))
#define Rw1 (0.07)
#define Rw2 (0.1)
#define Rw3 (0.1)
#define M (20.0)
#define mw1  (0.6)
#define mw2  (0.8)
#define mw3  (0.8)
#define Icm  (0.5*(M*(B-C)*(B-C)))
```

```
#define Iw1   (0.5*(mw1*Rw1*Rw1))
#define Iw2   (0.5*(mw2*Rw2*Rw2))
#define Iw3   (0.5*(mw3*Rw3*Rw3))
#define Jw1   (0.3333*(mw1*Rw1*Rw1))

/* function prototype */
void invkin(float,float,float,float,float *,float *,float *,float
            *,float *);
void invdyn(float,float,float,float,float,float *,float *,float
            *,float *,float *,float *,float *,float
            *,float *,float *,float *,float *);
void robot(float,float,float,float,float,float,float,
           float,float,float,float,float,float,float *,float
           *,float *,float *,float *,float *);
void forkin(float,float,float,float,float,float *,float *,float
            *,float *);
void obstacle(float,float,float,float,float,float,float *,float
              *);

void main(void)
{

     FILE *fptr;
     FILE *pt;
     FILE *qt;
     FILE *rt;
     FILE *st;
     float x0,y0,x,y,X,Y,xd,yd,Vx0,Vy0,Vx,Vy,Vxd,Vyd,Ax,Ay,
           vx,vy,ax,ay;
     float ohm1,ohm2,ohm3,dohm1,dohm2,dohm3,qk,dqk,ddqk,del,
           dela,ddel,dddel;
     float TMAX,kp,kv,ks,bs,dt,t,xo,yo,ro,xm,ym,xmo,ymo,
           Rmo,KR,BR;
     float Arx,Ary,Amx,Amy,rb,xb,yb,xmb,ymb,Rmb;
     float Fx1,Fy1,Fx2,Fy2,Fx3,Fy3,Gx1,Gy1,Gx2,Gy2,Gx3,Gy3,Tor;
     int i, count,k,kmax;

      fptr = fopen("obspos.dat", "w");
      pt = fopen("obsxyt.dat","w");
      qt = fopen("obstor.dat","w");
      rt = fopen("qtime.dat","w");
      st = fopen("dtime.dat","w");
/*   initialization */
      x0 =.0;              /* initial position of robot */
      y0 = .0;
      Vx0 = .0;             /* initial velocity of robot */
      Vy0 = .0;
      qk = .0;              /* initial orientation of robot */
      dqk = .0;    /* initial angular velocity of robot c. of m. */
      del = .0;
      ddel = .0;
```

```
/* target values */
     xd = 10.0;
     yd = 10.0;
     Vxd = .0;
     Vyd = .0;
/* obstacle values */
     xo = 7.;
     yo = 4.;
     ro = 2.;
     rb = 2.;

     count = 1;
     printf("\nSIMULATION TIME ? ");
     scanf("%f",&TMAX);
     printf("\nSpring constant,KP = ");
     scanf("%f",&kp);
     printf("\nDamping constant,KV = ");
     scanf("%f",&kv);
     printf("\nDamping constant, KV = %.4f", kv);
     printf("\nSampling Time, dt = ");
     scanf("%f",&dt);
     printf("\nSteering spring constant, KS = ");
     scanf("%f",&ks);
     printf("\nThe spring constant of repulsive force ?\n");
       scanf("%f",&KR);

     kmax = (int)(TMAX/dt);
     printf("\nThe maximum count : %d",kmax);
     X = x0;
     Y = y0;
     Vx = Vx0;
     Vy = Vy0;
     Ax = 0.0;
     Ay = 0.0;

     for (k=1; k<=kmax; k++)
       {
       t = k*dt;   /* before invkin() put control law */

       /* obstacle avoidance */

       xm = X;
       ym = Y;
       xmo = xm-xo;
       ymo = ym-yo;
       Rmo = sqrt(xmo*xmo + ymo*ymo);
       Rmb = Rmo - ro;
       xb = xo + (ro/Rmo)*xmo;
       yb = yo + (ro/Rmo)*ymo;
       xmb = xm - xb;
       ymb = ym - yb;
```

```
        if ( Rmb <= rb)
             obstacle(xmb,ymb,Rmb,rb,KR,BR,&Arx,&Ary);
        else
        {
        Arx = 0.0;
        Ary = 0.0;
        }
        Amx = (kp/M)*(xd-xm)+(kv/M)*(Vxd-Vx);
        Amy = (kp/M)*(yd-ym)+(kv/M)*(Vyd-Vy);

        Ax = Amx + Arx;
        Ay = Amy + Ary;
        vx = Vx*cos(qk) + Vy*sin(qk);
        vy = Vy*cos(qk) - Vx*sin(qk);
        ax = Ax*cos(qk) + Ay*sin(qk);
        ay = Ay*cos(qk) - Ax*sin(qk);
        invkin(vx,vy,ax,ay,&ohm1,&ohm2,&ohm3,&dqk,&dela);

        bs = 5.*sqrt(ks*Jw1);
        dddel = (ks/Jw1)*(dela-del)-(bs/Jw1)*ddel;
        ddel = ddel+dddel*dt;
        del = del+ddel*dt+0.5*dddel*dt*dt;

        invdyn(ax,ay,ohm2,dqk,del,&Fx1,&Fy1,&Fx2,&Fy2,&Fx3,
            &Fy3,&Gx1,&Gy1,&Gx2,&Gy2,&Gx3,&Gy3,&Tor);
robot(Fx1,Fy1,Fx2,Fy2,Fx3,Fy3,Gx1,Gy1,Gx2,Gy2,Gx3,Gy3,Tor,del,
     &ddqk,&dohm1,&dohm2,&dohm3,&ax,&ay);
        ohm1 = ohm1 + dohm1*dt;
        ohm2 = ohm2 + dohm2*dt;
        ohm3 = ohm3 + dohm3*dt;
        forkin(ohm1,ohm2,ohm3,ay,del,&dqk,&ddqk,&vx,&vy);

        vx = vx+ax*dt;
        vy = vy+ay*dt;
        x = x + vx*dt + 0.5*ax*dt*dt;
        y = y + vy*dt + 0.5*ay*dt*dt;
        qk = qk + dqk*dt + 0.5*ddqk*dt*dt;
        Vx = vx*cos(qk) - vy*sin(qk);
        Vy = vx*sin(qk) + vy*cos(qk);
        Ax = ax*cos(qk) - ay*sin(qk);
        Ay = ax*sin(qk) + ay*cos(qk);
        X = x*cos(qk) - y*sin(qk);
        Y = x*sin(qk) + y*cos(qk);


        if (count == 20)
             count = 1;
        if ( count == 1)
             {
             printf("%14.5f,%14.5f,%14.5f,%14.5f\n", X, Y,qk,t);
             fprintf(fptr,"%14.5f %14.5f\n", X, Y);
             fprintf(fptr,"%14.5f %14.5f\n", xb,yb);
```

```
            fprintf(pt,"%14.5f %14.5f %14.5f\n", t, X, Y);
            fprintf(qt,"%14.5f %14.5f\n", t, Tor);
            fprintf(rt,"%14.5f %14.5f\n", t, qk);
            fprintf(st,"%14.5f %14.5f\n", t, del);
             }
        count++;

        }
        fclose(fptr);
}

/* obstacle avoidance algorithm using only spring in repulsive
force field. */
void obstacle(float xmo,float ymo,float Rmo,float ro,float
              KR,float BR,float *Arx,float *Ary)
{

     *Arx = (1/M)*(xmo/Rmo)*KR*(Rmo-ro)*(Rmo-ro);
     *Ary = (1/M)*(ymo/Rmo)*KR*(Rmo-ro)*(Rmo-ro);
}

/* obstacle avoidance algorithm using spring  and damper in
repulsive force field. */

void obstacle(float xmo,float ymo,float Rmo,float ro,float
              vxmo,float vymo,float KR,float BR,float *Arx,float
              *Ary)
{

     *Arx = (1/M)*(xmo/Rmo)*KR*(Rmo-ro)*(Rmo-ro)-BR*vxmo;
     *Ary = (1/M)*(ymo/Rmo)*KR*(Rmo-ro)*(Rmo-ro)-BR*vymo;
}

/* Multiobstacle avoidance algorithm using spring  and damper in
repulsive force field. */
/*
 * MULobskb.C
 * Multiple Obstacle Avoidance of Mobile Robot
 * using Artificial Impedance Approach.
 * Obstacle is a circle having radius of 2 meters.
 *  Repulsive forces exerted by spring constant and damping
constant.
 */

void obstacle(float xmo,float ymo,float Rmo,float ro,float
              vxmo,float vymo,float KR,float BR,float *Arx,float
              *Ary)
{

     *Arx = (1/M)*(xmo/Rmo)*KR*(Rmo-ro)*(Rmo-ro)-BR*vxmo;
     *Ary = (1/M)*(ymo/Rmo)*KR*(Rmo-ro)*(Rmo-ro)-BR*vymo;
}
```

```
void multiobs(float ox1,float oy1,float ox2,float oy2,
              float xm,float ym, float *xo,float *yc)
{
     float R1,R2;

     R1 = sqrt((ox1-xm)*(ox1-xm) + (oy1-ym)*(oy1-ym));
     R2 = sqrt((ox2-xm)*(ox2-xm) + (oy2-ym)*(oy2-ym));
     if (R1 < R2)
     {
     *xo = ox1;
     *yo = oy1;
     }
     else
     {
     *xo = ox2;
     *yo = oy2;
     }
}
```