California AHMCT Program
University of California at Davis
California Department of Transportation

DEVELOPMENT OF A TETHERED MOBILE ROBOT (TMR) FOR HIGHWAY MAINTENANCE

Scott E. Winters
Steven A. Velinsky

AHMCT Research Report
UCD-ARR-92-11-25-01

Interim Report of Contract
IA65Q168-MOU 92-9

November 25, 1992

DEVELOPMENT OF A TETHERED MOBILE ROBOT (TMR) FOR HIGHWAY MAINTENANCE

Interim Report

Scott E. Winters

and

Steven A. Velinsky

Department of Mechanical, Aeronautical, and Materials Engineering

University of California, Davis

November 25, 1992

Advanced Highway Maintenance and Construction Technology Program

DISCLAIMER/DISCLOSURE

"The research reported herein was performed as part of the Advanced Highway Maintenance and Construction Technology Program (AHMCT), within the Department of Mechanical, Aeronautical and Materials Engineering at the University of California, Davis and the Division of New Technology and Materials Research at the California Department of Transportation. It is evolutionary and voluntary. It is a cooperative venture of local, state and federal governments and universities."

"The contents of this report reflect the views of the author(s) who is (are) responsible for the facts and the accuracy of the data presented herein. The contents do not necessarily reflect the official views or policies of the STATE OF CALIFORNIA or the FEDERAL HIGHWAY ADMINISTRATION and the UNIVERSITY OF CALIFORNIA. This report does not constitute a standard, specification, or regulation."

ABSTRACT

Many highway maintenance operations, in addition to tasks in numerous other industries, involve the use of materials and tooling within close proximity to a support vehicle. For example, highway crack sealing operations involve maintenance personnel dispensing sealant from a wand which is attached to a vehicle housing the sealant melter. Other operations involve the use of tools which are powered by a supply on the support vehicle, such as painting equipment, etc. While the use of conventional robots to automate these operations seems at first consistent with many of the positional requirements of maintenance tasks, their use is hindered due to several reasons. First and foremost, commercial robots have a relatively low load carrying capacity relative to their weight. Considering the weight of many road maintenance devices, such as routers, and the forces that occur during their operation, the use of conventional robot/end effectors is not possible. Highway maintenance activities almost always require an end-effector to follow a specific path as opposed to merely moving from one location to another without path following requirements as in most manufacturing automation applications. Another aspect relates to the fact that most highway maintenance operations require the placement of the device within a specific height relative to the pavement (e.g., paint nozzles, routers, etc.) which additionally complicates the use of conventional robots.

In order to overcome the inherent disadvantages of the use conventional robots for the above tasks, an unique concept has been developed. This concept is the use of a self-propelled robot working in close proximity to a support vehicle for purposes of power, materials, etc., and allowing for the measurement of these robot's position relative to the support vehicle with high accuracy. The purpose of this thesis is to conceptually develop the Tethered Mobile Robot (TMR). In addition to presenting the conceptual TMR system layout, a simple prototype was built and tested to prove validity.

EXECUTIVE SUMMARY

Many highway maintenance operations, in addition to tasks in numerous other industries, involve the use of materials and tooling within close proximity to a support vehicle. For example, highway crack sealing operations involve maintenance personnel dispensing sealant from a wand which is attached to a vehicle housing the sealant melter. Other operations involve the use of tools which are powered by a supply on the support vehicle, such as painting equipment, etc. While the use of conventional robots to automate these operations seems at first consistent with many of the positional requirements of maintenance tasks, their use is hindered due to several reasons. First and foremost, commercial robots have a relatively low load carrying capacity relative to their weight. Considering the weight of many road maintenance devices, such as routers, and the forces that occur during their operation, the use of conventional robot/end effectors is not possible. Highway maintenance activities almost always require an end-effector to follow a specific path as opposed to merely moving from one location to another without path following requirements as in most manufacturing automation applications. Another aspect relates to the fact that most highway maintenance operations require the placement of the device within a specific height relative to the pavement (e.g., paint nozzles, routers, etc.) which additionally complicates the use of conventional robots.

In order to overcome the inherent disadvantages of the use conventional robots for the above tasks, an unique concept has been developed. This concept is the use of a self-propelled robot working in close proximity to a support vehicle for purposes of power, materials, etc., and allowing for the measurement of these robot's position relative to the support vehicle with high accuracy. The purpose of this thesis is to conceptually develop the Tethered Mobile Robot (TMR). In addition to presenting the conceptual TMR system layout, a simple prototype was built and tested to prove validity.

To begin the development, a general description of the problem was presented, outlining the need for a new method to overcome the inherent disadvantages of the use of conventional robots for highway maintenance operations. Additionally, a literature search was performed to investigate existing tethered mobile robot technology and other related topics. Based on an intensive literature review, it is the author's opinion that this concept has not been presented previously. The literature most closely related concerns autonomous mobile robots, semi-autonomous mobile robots and other mobile robot technologies.

Next, a general description of the TMR's conceptual configuration was presented. This included areas such as the wheel configuration, tracking configuration, control configuration, and local sensor configuration. In each section a literature review preceded the components discussion. After presenting the conceptual configuration of these specific areas, the TMR's integrated conceptual layout was given.

With the general conceptual configuration determined, the general linkage and workspace analysis used in designing the TMR system was presented. Additionally, the general workspace configuration was addressed which detailed the actual layout. Two workspace areas were decided upon in order to have a versatile and usable system, one which allowed the system to address a full lane width and the other which is located within the width confines of the support vehicle.

With the completion of the conceptual aspects of this thesis, the last chapters focused on modeling and experiments. The TMR was modeled in order to facilitate in the design of control algorithms, which is critical to the TMR's path tracking ability. A scaled down prototype was developed and built, which allowed testing of controllers, kinematics, and path trajectories covered earlier. The testing was separated into two sections, open loop and closed loop testing, respectively.

Finally, conclusions and recommendations were made. Based on the tests which were performed, it was concluded that the TMR can be accurately controlled to follow a

specified path. Recommendations concerning the TMR's development and operation were suggested to optimize performance.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1 - INTRODUCTION

1.1 Introduction

Many highway maintenance operations, in addition to tasks in numerous other industries, involve the use of materials and tooling within close proximity to a support vehicle. For example, highway crack sealing operations involve maintenance personnel dispensing sealant from a wand which is attached to a vehicle housing the sealant melter. Other operations involve the use of tools which are powered by a supply on the support vehicle, such as painting equipment, etc. While the use of conventional robots to automate these operations seems at first consistent with many of the positional requirements of maintenance tasks, their use is hindered due to several reasons. First and foremost, commercial robots have a relatively low load carrying capacity relative to their weight. Considering the weight of many road maintenance devices, such as routers, and the forces that occur during their operation, the use of conventional robot/end effectors is not possible. Highway maintenance activities almost always require an end-effector to follow a specific path as opposed to merely moving from one location to another without path following requirements as in most manufacturing automation applications. Another aspect relates to the fact that most highway maintenance operations require the placement of the device within a specific height relative to the pavement (e.g., paint nozzles, routers, etc.) which additionally complicates the use of conventional robots.

Accordingly, unique concepts have been developed to overcome the inherent disadvantages of the use of conventional robots for highway maintenance operations. A prime example is the positioning system concept being used on the automated crack sealing machine under development at the University of California, Davis. This work is being supported by the National Research Council's Strategic Highway Research Program (SHRP) and the California Department Of Transportation (CALTRANS). In this concept, shown in Fig.(1.1), a conventional SCARA manipulator is inverted and

mounted on a linear slide to provide a redundant degree of freedom allowing the manipulator to avoid singular positions in its motion and move through any prescribed path in its dexterous workspace. The SCARA manipulator is used to guide process carts over the pavement along specific paths (following cracks). Such an approach provides accurate and consistent relative positioning between the maintenance device and the pavement, and additionally relieves the manipulator of the burden of carrying the weight of that maintenance device. The determination of each cart's location is through the robots joint positioning. One typical problem with this configuration is that the mechanical advantage of the robot is dependent upon its joint positions.

A natural evolution of the U.C. Davis SHRP concept above, is the use of a self-propelled robot working in close proximity to a support vehicle for purposes of power, materials, etc., and allowing for the measurement of these robot's position relative to the support vehicle with high accuracy. As such, the support vehicle would contain the associated maintenance supplies (sealant, etc.), power supply (hydraulic power supply, electrical generator, etc.), and in many cases the primary maintenance operation sensing devices (e.g., machine vision for crack sealing operations). The mobile robot would be supplied with the necessary maintenance materials and power through a tether to the support vehicle, and some type of system would accurately determine the location of the robot relative to the support vehicle. This relative position could be based on any of a variety of technologies; i.e., it could be through a mechanical connection (i.e., linkage), an optical connection, etc. Furthermore, such an approach has the potential for application in a wide variety of applications including highway maintenance tasks, toxic waste clean-up, etc.

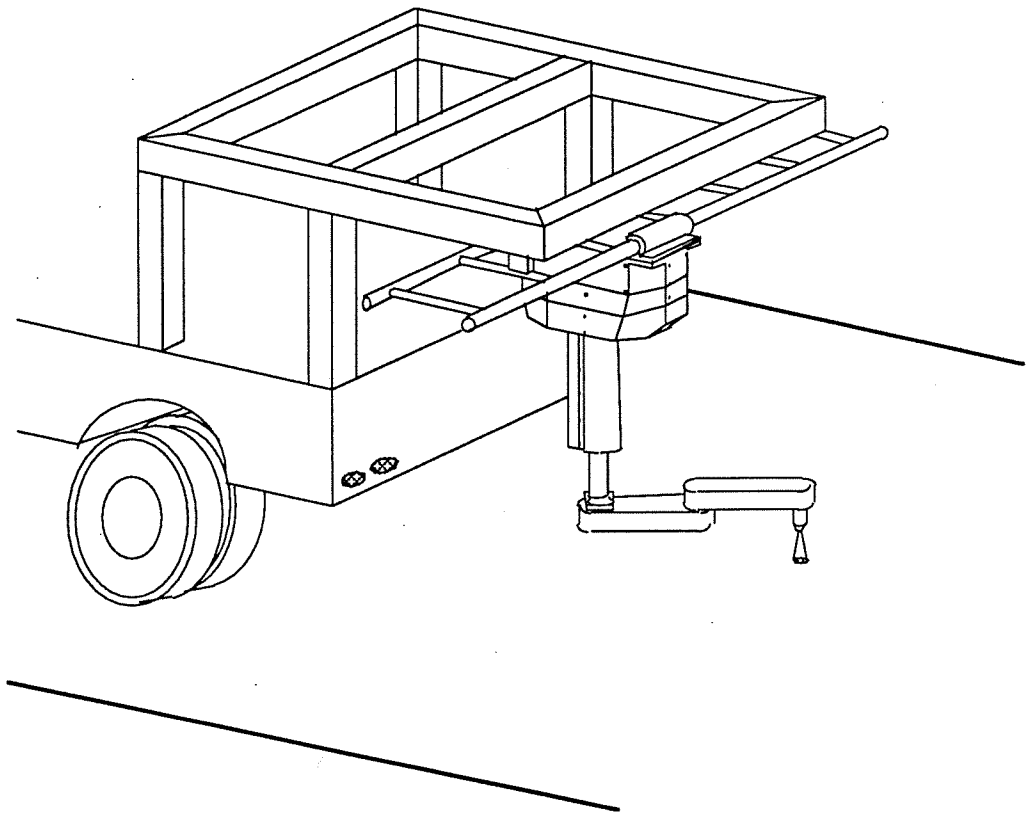


Figure 1.1 Inverted Conventional SCARA Manipulator

1.2 Literature Search

A literature search was initially performed to investigate existing tethered mobile robot technology and other related topics. Based on an intensive literature review, it is the author's opinion that this concept has not been presented previously. The literature most closely related concerns autonomous mobile robots, semi-autonomous mobile robots and other mobile robot technologies. The literature search, which will be presented in each corresponding section due to its diversity, reviews a wide variety of topics of value to the development of the Tethered Mobile Robot (TMR).

1.3 Problem Statement and Objective

The objective of this thesis is the conceptual development of a self-propelled robot which works in close proximity to a support vehicle for purposes of power, etc., and allowing for the measurement of these robot's position relative to the support vehicle with high accuracy. Such a device has the potential for use in a wide variety of applications. The development will involve a detailed literature search of each area, development of the general TMR configuration, development of the workspace, development of a dynamic model and simulations, and lastly the development of a small scale prototype which is used for testing/verification of the control equations.

CHAPTER 2 - GENERAL TMR CONFIGURATION

2.1 Introduction

The purpose of this chapter is to give a general description of the TMR's configuration, which includes areas such as the wheel configuration, tracking configuration, control configuration, and local sensor configuration. In each section a literature review will precede the components discussion. Before discussing the different components, a general description of the TMRs' objectives and related topics will be presented.

Objectives

The objectives of the Tethered Mobile Robot (TMR) are as follows: self-propelling, controllable, robust, and compact, with the ability to accurately follow a designated path. Since the TMR will be used primarily on asphalt and/or concrete roadways, which are considered fairly smooth and hard surfaces, only wheels for self-propelling will be considered. This is because wheels are much more energy efficient than legged or treaded concepts under these conditions (Muir and Neuman, 1986). The cart must be steerable, being able to follow a defined path in the roadway. The ability to control the position, velocity, and acceleration of the cart is a must. Robustness, the ability to attenuate disturbances, is also of prime importance. Disturbances, such as, irregular surfaces and sealant affixations (for highway maintenance) must not affect the performance.

Tractive Force Ability

In order for the cart to be self-propelled, it must be able to produce enough tractive force (between the drive wheels and contact surface) to counteract the router's resultant force (for highway maintenance) and other applied loads, while still accelerating the cart to proper speeds. During the routing operation, it has been determined that the resultant force from the router's blade is approximately 200 lbs. (890 N) in the direction

opposing the forward motion. Additionally, the total normal force necessary to ensure proper pavement cutting is approximately 500 lbs. (2224 N) (Bennet, 1992). From this information and assuming an adequate acceleration and minimal tire slip, the minimum tire tractive force needed to propel the cart is approximately 300 lbs. (1334 N). To determine which of the many wheel configurations will meet this criteria, we will need to make some reasonable engineering conjectures.

Robustness

As mentioned above, robustness is one of the criteria the TMR configuration chosen must exhibit. Robustness as defined here, will mean the ability of the wheel configuration (and wheels) to handle disturbances and irregularities without effecting its overall performance. In addition to the wheel configuration, the other components such as control hardware, tracking sensors, etc., must also exhibit this characteristic. In the highway maintenance work, the wheel disturbances will consist of, but are not limited to, the following; problems with sealant making contact with the wheel and the ability to travel on irregular surfaces such as roadways.

Controllability

Due to the relatively slow speeds at which our cart will be traveling, control will be based mainly on kinematic constraints, rather than a combination of kinematic and kinetic constraints. This method of ignoring vehicle dynamics is used widely in control strategies for automatic guided vehicles (AGV's) due to the low speeds and accelerations at which they operate (Reister, 1986; Kompass and Williams, 1990; Smith and Starkey, 1991). Control algorithms based on kinematics will allow any wheeled configuration to be controlled, and therefore is not a limiting constraint. However, actuating the different wheel configurations range from very difficult, e.g., ball wheels, to the simpler devices such as conventional wheels.

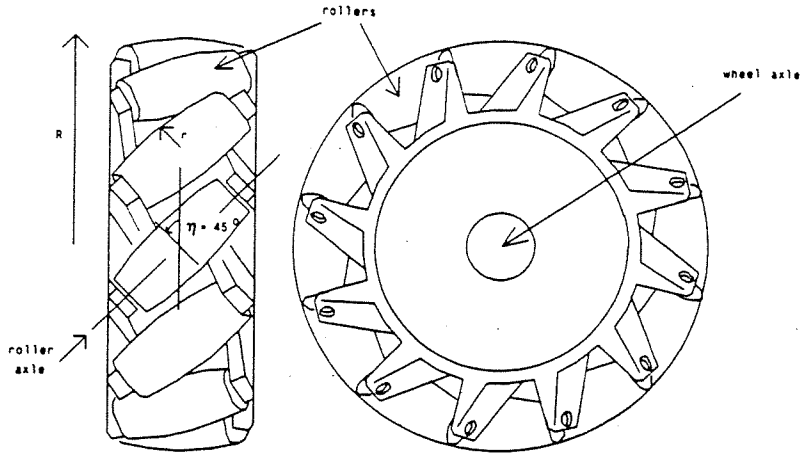
2.2 Wheel/Robot Configuration

Many different wheeled configurations were reviewed and investigated for use as the propelling method for the TMR. Documentation of wheeled configurations can be found in a variety of sources ranging from hobbyist literature to robotics (Muir and Neuman 1986, Killoughi 1991, Alexander 1989, Feng 1991, Smith 1991, Scott 1989). Of the many wheel configurations, three main wheel types were used; conventional, omnidirectional, and ball wheels. These wheel types can be seen in Fig. (2.1).

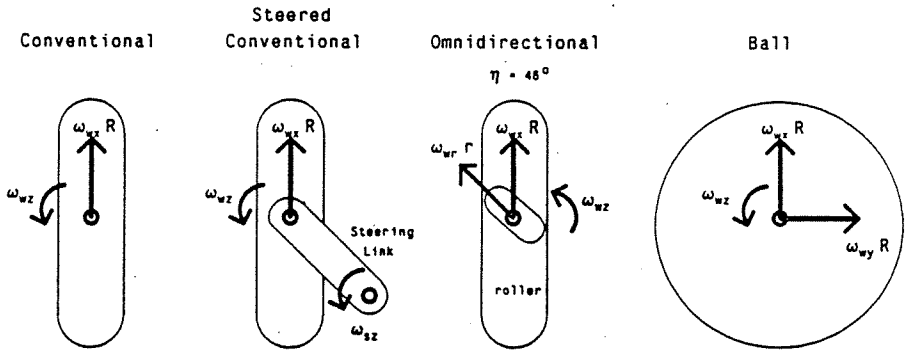
The conventional wheel having two degrees of freedom (DOF) is the simplest to construct (similar to a bicycle wheel). This type of configuration allows travel in the direction of wheel orientation, plus rotation about the point of contact between the wheel and the surface, therefore giving two DOF. The rotational DOF is slippage, however transverse slip (perpendicular to wheel orientation) is not a DOF due to the magnitude of force required for transverse motion.

The omnidirectional wheel having three DOF can be much more difficult to construct than the two DOF conventional wheel, however the benefit is one more DOF. The three DOF are as follows; one in the direction of the wheel orientation, the second in the direction of the rollers mounted on the circumference of the wheel, and the third DOF is the rotational slip between the rollers and the ground as in the conventional wheel.

The ball wheel has three DOF, yet it produces each DOF without slip, unlike the omnidirectional wheel. The disadvantage of the ball wheel is that it is very difficult to actuate.



Omnidirectional Wheel (Rollers at 45°)



Wheel DOFs

Figure 2.1 Conventional, Omnidirectional, and Ball Wheels (Muir and Neuman, 1986)

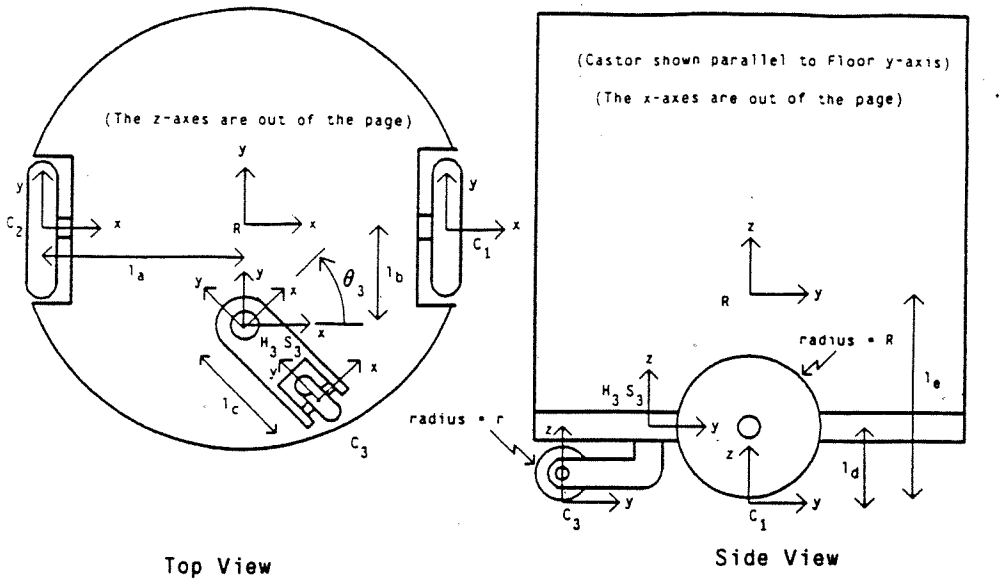
In this section, various wheeled configurations will be discussed, concentrating on their advantages and disadvantages related to our needs. Of the configurations discussed,

each will relate to a specific wheeled configuration, such as Newt is related to the "two parallel conventional wheel" configuration (Muir and Neuman, 1986).

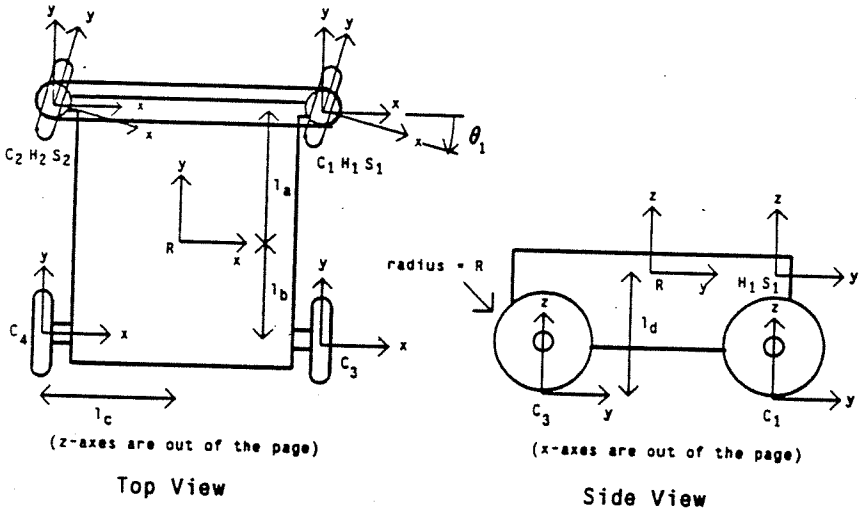
The following table (2.1) provides a list of mobile robots whose wheel configurations have aspects which apply to our TMR. Figs. (2.2), (2.3), and (2.4) show the general configuration of a few different mobile robots listed in table (2.1). Shown in Fig. (2.2) is Newt and the Stanford Cart, both of which have two degrees of freedom. Neptune and Rover are shown in Fig. (2.3), where Rover has three degrees of freedom and Neptune two. Lastly, the omnidirectional mobile robots Unimation and Uranus are presented in Fig. (2.4).

TWO (DOF) conventional	**TWO (DOF) steered conventional**	**THREE (DOF) omnidirectional**
Shakey	Neptune	Uranus
Newt	Hero-I	Unimation robot
Jason	Avatar	Easy Rollers
Hilare	Stanford Cart	Rover
Yamabiko	Blanche	Hermies III
ROBART II		

Table 2.1 Table of Different Wheel Configurations

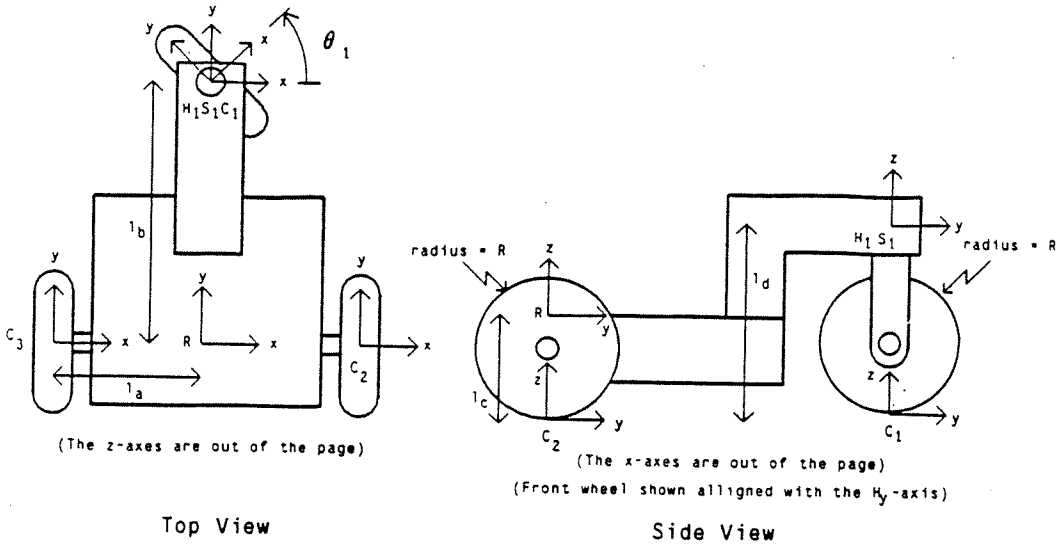


Newt

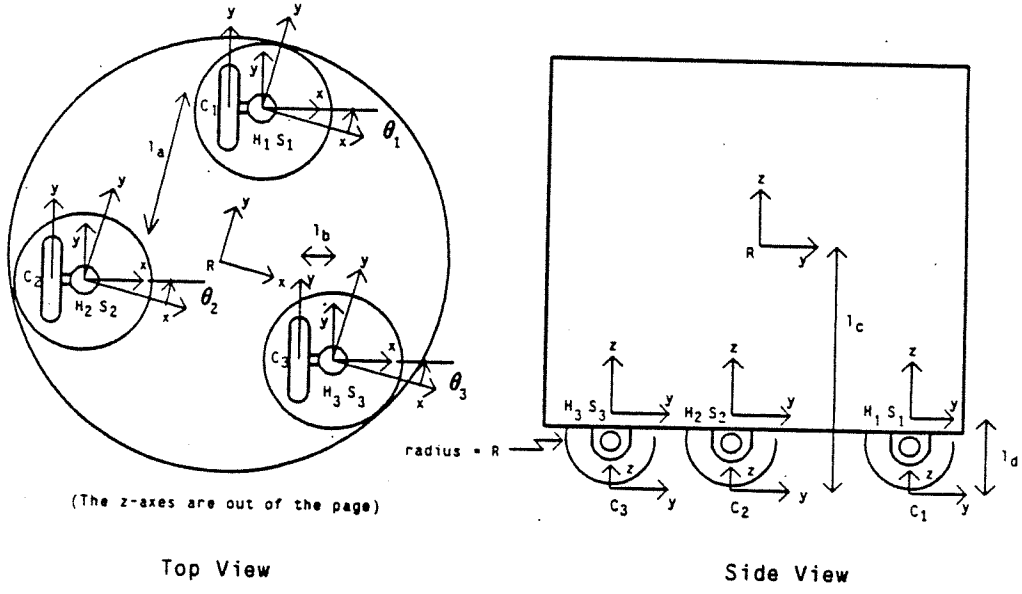


Stanford Cart

Figure 2.2 Newt and Stanford Cart (Muir and Neuman, 1986)

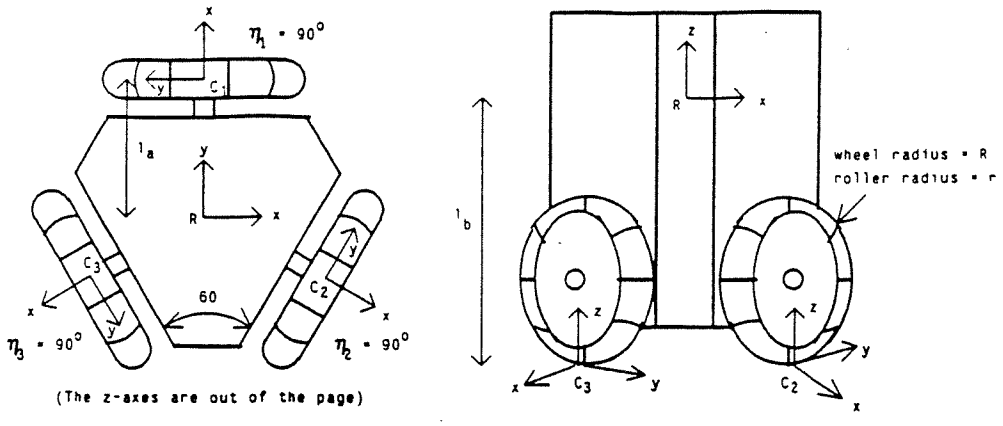


Neptune



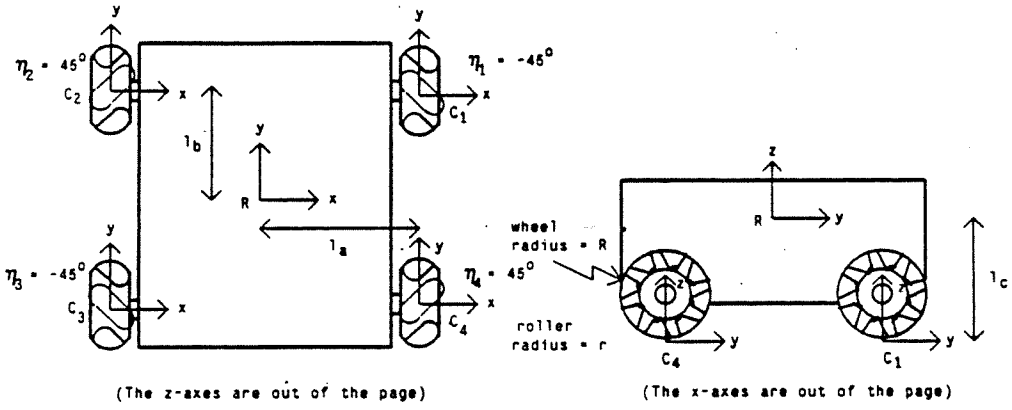
Rover

Figure 2.3 Neptune and Rover (Muir and Neuman, 1986)



Top View Side View

Unimation Robot



Top View Side View

Uranus

Figure 2.4 Unimation and Uranus (Muir and Neuman, 1986)

Newt

Newt has a configuration which consists of two diametrically opposed drive wheels and a single free-rolling castor. This general configuration will allow two dimensional motion, therefore any path in a plane can be traced (Muir and Neuman, 1986; Reister, 1990). This configuration is considered mechanically simple, due to its low quantity of parts required for assembly. The disadvantage of this two DOF configuration is that it contains singularities in its workspace. In general, any configuration which does not have three DOF will contain singularities. Singularity points for Newt lie along a line which is perpendicular to the wheels orientation and lie along its axle.

Stanford Cart

The Stanford Cart wheel configuration is very similar to that of an automobile, having two front non-driven steerable conventional wheels and two parallel non-steered driven conventional back wheels (Moravec, 1983). This configuration has two DOF, however its ability to follow a desired two-dimensional path is limited by its turning radius. Thus, its path following abilities do not match that of Newt. The Stanford Cart also contains a larger region of singularity points, due to its fixed axle and turning radius.

Uranus

Uranus has a wheel configuration which allows three DOF, and therefore contains no singularity points by definition (Muir and Neuman, 1986; Killoughi and Pin, 1991). This omnidirectional (3 DOF) configuration has four omnidirectional wheels with rollers mounted on their circumference at 45 degree angles to the wheels. Uranus' main advantage is the ability to travel in any direction, however it has a disadvantage of being very complicated. The use of four wheels is good for stability, although with four points of ground contact a wheel suspension system is required to ensure wheel-ground contact

at all times. In order to move directly sideways, all wheels must rotate perpendicular to the direction of travel, which results in poor efficiency (Carlisle,1983).

Rover

Rover's wheel configuration consists of three conventional wheels arranged in a symmetrical pattern, with each wheel independently driven and steered. This configuration classifies Rover as omnidirectional (3 DOF), and therefore has no singularity points. The drive system has both steering motors and drive motors connected to a spur gear differential, which is then connected to the wheels. The advantage of this configuration is that it permits low friction steering, which reduces the power required while the vehicle is stationary (Moravec, 1983).

Neptune

Neptune's wheel configuration is similar to that of a tricycle, having a single conventional steered and driven wheel, plus two fixed rear parallel conventional wheels. This configuration has two DOF, along with singularity points similar to the Stanford Cart. The advantage of this configuration over the Stanford Cart is that it has one less wheel, and therefore the Neptune configuration is mechanically simpler to build.

Above, many different wheel configurations have been discussed with their advantages and disadvantages being noted. The three main types of wheels consist of the following; conventional, omnidirectional, and ball wheels, however the ball wheels are rarely used.

The conventional wheel configuration, as used by Newt, Shakey, and Jason, for example, has two DOF. The disadvantage of the two DOF configuration is that it contains singularities in it workspace, although any path in a plane can be traced. The advantage of this configuration is that it is mechanically simple to build.

The other conventional wheel configurations, such as the Stanford Cart and Neptune, have 2 DOF, yet they can not follow any arbitrary path in a plane. This is due to their constrained turning radius.

The omnidirectional wheel configuration, as used by Uranus and Unimation robot, for example, has three DOF. The disadvantage of this configuration is its complexity. Although complex, the omnidirectional configuration allows motion in any direction unconstrained, which means there are no singularity points. In order to choose the appropriate wheel configuration for our need, a trade off will be made between the omnidirectional configuration with three DOF and the mechanically simpler conventional wheel configuration with two DOF.

It is proper design procedure to design for the worse conditions, and therefore the TMR should be designed in this manner. In the many conceivable operations that the TMR will operate (highway maintenance, toxic clean-up, etc.), the worst case condition is likely the pavement routing process. During the routing process, two main constraints apply to the system. First, the cart must always move tangent to the path it is routing. This is due to the router blade, which cuts properly only in a direction perpendicular to its rotation. Secondly, the cart's wheel configuration must produce enough tractive force to overcome the routing blade's resultant force.

In the survey of different wheel configurations, it was determined that certain two DOF systems, such as Newt and Jason for example, could follow any path in a plane. The other three DOF systems could also follow any path in a plane. Examination of the first constraint of the router and comparison of the wheel configurations, shows that a three DOF system would give a redundant DOF. The advantage of using a three DOF system would be in locating the TMR at its initial position.

The tractive force ability of each wheel configuration is a determining factor concerning the efficiency of each configuration. If the wheel configuration can not produce enough tractive force for our application, it is not practical. It was concluded

that all systems which use some type of pneumatic or semi-pneumatic tire could produce adequate force and all other non-compliant (plastics, metal, etc.) tires would most likely not. Robustness as defined here, means the ability of the wheel configuration (and wheels) to handle disturbances and irregularities. It was determined that the conventional wheel configuration is more robust than the omnidirectional configuration when looking at surface irregularities and sealant problems. Lastly, controllability as viewed from a kinematic sense, can be achieved in all above wheeled configurations. The main advantage of some systems over others is the reduced requirement of equipment to actuate the wheels.

Based on the above problem definition and design selection, the wheeled configuration similar to Newt, Shakey, and Jason for example, proves to be the best selection. This configuration meets all the requirements without any redundancies, and demonstrates to be the lowest cost to produce. The main disadvantage of this configuration is the added path planning required for initial conditions.

2.3 Tracking System Configuration

In order to control the position and path trajectory of the TMR, the spatial position must be known very accurately. The sensing methods literature review was geared towards tracking devices for autonomous and semi-autonomous vehicles. These methods will be researched for partial or complete systems to track the linear position and orientation of the TMR. The majority of autonomous and semi-autonomous vehicles use Dead reckoning as their tracking method, while other systems include Vision systems, Infrared systems, Sonar systems, and Knowledge based systems (Mcgillem, 1988; Sugimoto, 1988; Smith, 1991; Zelinsky, 1991; Dainis, 1985).

Although electric/electronic methods have been introduced above exclusively, we also investigated the possibility of using a mechanical linkage between the support vehicle and the Tethered Mobile Robot. In this configuration, ideally, no forces are applied to the linkage which freely follows the motion of the TMRs. It should be noted

that the concept of tethering the mobile robot to a support vehicle through the use of a linkage was not found, although water and air based systems have used umbilical cords as a method of tethering vehicles.

Dead reckoning is the process of position determination by measuring wheel rotations. Usually the rotations of each wheel on both sides of the vehicle are measured, and the results are processed to determine position and orientation. Although the equipment cost for dead reckoning is low, it suffers from error accumulation. Therefore, a backup system to periodically determine the absolute position should be used with a dead reckoning system. The cumulative errors of dead reckoning primarily arise from wheel slippage and uncertainty of the wheel rolling radius. There are a few methods which help reduce the dead reckoning errors, such as passive measuring wheels (non drive wheels) (Sugimoto et al, 1988) and also a castor sensing wheel (Culley and Daldur, 1988), However, these methods still do not reduce the error completely.

The infra-red optical scanner is a promising technique for accurately locating an automatic guided vehicle. The optical scanner is capable of measuring angles between pairs of beacons. The basic method for locating the automatic guided vehicle is to set up a three beacon system with an optical scanner mounted to the vehicle. The location (position and orientation) of the vehicle can then be determined, identical to the solution of the "3 point problem" in land surveying. The computation time is an important consideration since it governs the accuracy of a position measurement made by a vehicle while moving. In addition to large computation time for algorithms, the solution requires keeping track of the quadrants of the angles in order to avoid serious errors in the position calculations. Lastly, the position and orientation determination seem to be much more accurate than the dead reckoning method described above, along with the lack of accumulating errors (McGillem, 1988).

The determination of the position of a mobile robot in an environment using sonar has been investigated by Miller and Drumheller, although this approach assumes an

accurate map of the environment is on hand. The main problems with sonar are the invalid readings due to false reflections and sonar beam spread. These inaccuracies cause the measured objects to seem further away than they actually are, and therefore useless data is produced (Zelinsky, 1991). Although there are some proposed tests to check data for validation, an experimental based solution does not exist.

A two camera opto-electronics remote measuring system precisely tracks in three dimension the location of infrared light emitting LEDs (Dainis, 1985). The resolution of this system depends upon the resolution of the hardware (A/D converters) and the amount of noise present. A system exhibiting zero noise would have the same resolution as the A/D converters (i.e., 1 part in 4096 for 12 bit). This corresponds to estimated real time measurement rates of approximately 200 Hz using the 8086 processor (Dainis, 1985). The errors of this opto-electric method are due to reflections of the infrared light, causing the resolution to decrease considerably. Also, only approximately 80 percent of the detector field of view can be used.

The linkage system, is a passive device which allows the automatic guided vehicle to be tracked very accurately. Although this type of device has not yet been seen in the literature, it seems to offer more positive than negative features. The main advantage of this arrangement is its absolute positioning determination, without error accumulation. Another advantage is that it gives a means of running the power and other needed cables to the TMR from the support vehicle. The disadvantages of this system would be the possible errors involved, such as linkage deflection and resolution which could not be taken into account easily.

After examining the different possibilities in tracking the TMR, it was determined that the popular methods, such as dead reckoning, infrared optics, and sonar would not meet the needed accuracy and reliability of the tracking system requirements. Therefore, the initial design concept chosen has the TMR connected to the support vehicle via a mechanical linkage. Since the TMR has self-driving capabilities, passive elements

without driving capabilities are adequate for the linkage system which was adapted to handle irregular road surfaces and a wide range of workspaces. The general configuration of the planar linkage has an encoder mounted on each joint in order to calculate the relative position of the TMR to the support vehicle.

2.4 Control System Configuration

The newest method for controller hardware is that of Application Specific Integrated Circuit (ASIC) technology, which is much more attractive than the older bulky configurations. This high density semiconductor design method provides a highly integrated microelectronics component for industrial use, which results in higher reliability, lower cost and compactness for control equipment. Besides the compaction of hardware, these newer controllers allow flexible implementation of software for control purposes (Yamazaki, 1992). The control system configuration described below is based on this new technology.

The Control System which is being used is shown in Fig. (2.5). The TMR will be located on the crack starting position by controlling each wheel. In this system, the local sensing system (discussed later in section entitled Local Sensor) is mounted on the robot platform and interacts with the robot controller. The local sensing system determines exact crack position and gives the data to the robot controller, which then places the TMR on this crack position.

The control system consists of various components, that is, the robot controller, the actuators which drive the TMR wheels, the actuator controllers, and the corresponding control algorithms and software. These will be explained in the following sections in detail.

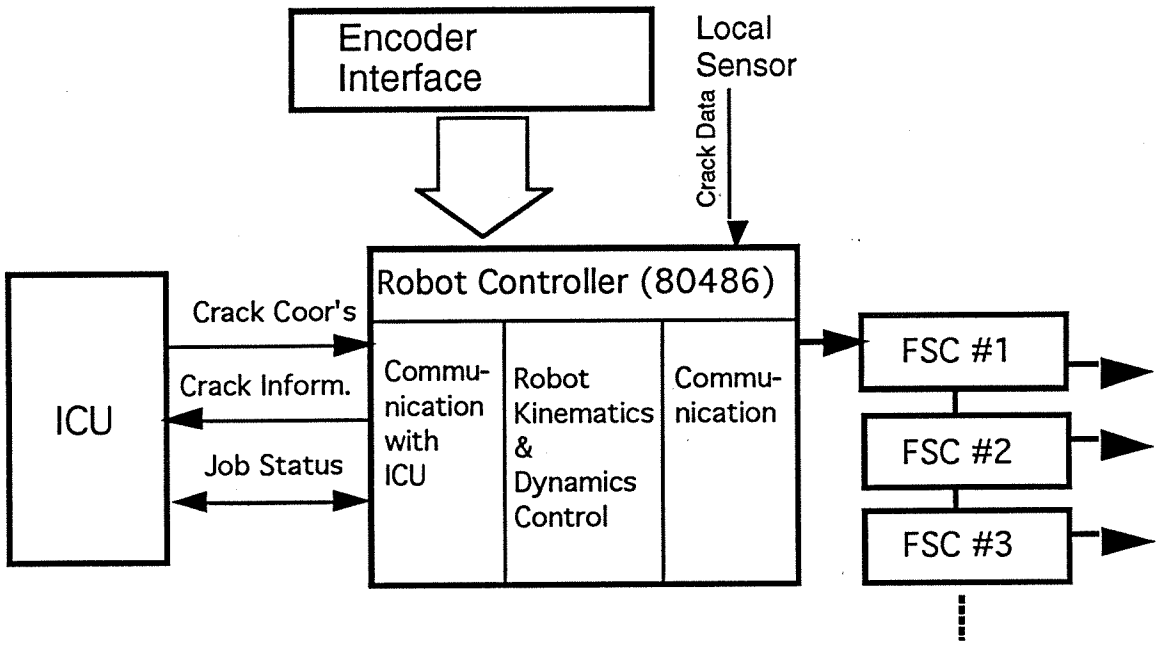


Figure 2.5 General Control System Configuration

Robot Controller

The robot controller supervises the interaction between the ICU (integration and control unit), and the robot positioning system and the control commands for each actuator controller. The ICU is the brain of the crack sealing machine, which handles all of the communication, monitoring, and general supervision of the individual components. To produce the control commands of the actuators, this controller performs the motion kinematics, path planning, display functions, and information transfer to each actuator controller through serial communication, RS232C. An 80486 microprocessor will be used for the controller. The processor receives the joint angle data from the encoder interface circuits Fig. (2.6). With these data, the position of the TMR is calculated and position and velocity control is accomplished.

Actuators-Motors

Electric motors can be used for the actuators to drive the wheels of the TMR, which have good controllability and control flexibility compared to other actuators (e.g., hydraulic motors, etc.). There are generally three kinds of electric motors, DC motors,

brushless DC (BLDC) motors, and AC induction motors. Table (2.2) shows a qualitative performance comparison.

	Low Speed	High Speed	Starting Torque	Speed Regulation	Maintenance
BLDC motor	Excellent	Excellent	Excellent	Excellent	Excellent
DC motor	Good	Poor-due to brush wear	Excellent	Good	Good-Brush Replacement
AC motor	Poor	Excellent	Poor-trouble with high inertia	Good-limited by slip of motor	Excellent

Table 2.2 Performance comparison of electric motors

In this table, it is evident that the BLDC motor is most adequate for our purposes. The DC motor has a maintenance problem even though its torque characteristics are excellent. The AC motor is not adequate because its stall torque is low. In terms of high torque, the DC motor is better than the BLDC motor, however high torque can be achieved by using an intermediate gear reduction assembly. Furthermore, since the high speed performance of the BLDC motor is better than the DC motor, this can be transformed to high torque via the gear reduction.

Since the TMR is being designed for the worse case, routing forces, the TMR needs a large traction force and heavy weight for stable and safe operation. If there are two driving wheels, having radii r, and the gear reduction ratio n, then the required torque T is;

$$T = \frac{Fr}{2n}$$

where F is the traction force. Thus, we must select the motor whose stall torque is greater than T. With the anticipated TMR velocity represented as v, the operating velocity of the motor is;

$$\omega = \frac{v}{r}$$

and the driving motor must have stable running characteristics at this speed.

Encoder Interface

Most optical encoders produce A, B, and Z pulses as outputs, which are able to inform the rotating direction and the origin of one complete revolution. Fig. (2.6) shows the block diagram of the encoder interface circuits. The direction detect circuit makes the A, B pulses a clockwise or counter-clockwise pulse train. The up-down counter counts them. The Z pulse resets the counter every one complete revolution of the encoder. After the counter produces the counting number, this data is latched by the parallel I/O. Then, the CPU can read that data any time, which represents each joint angles.

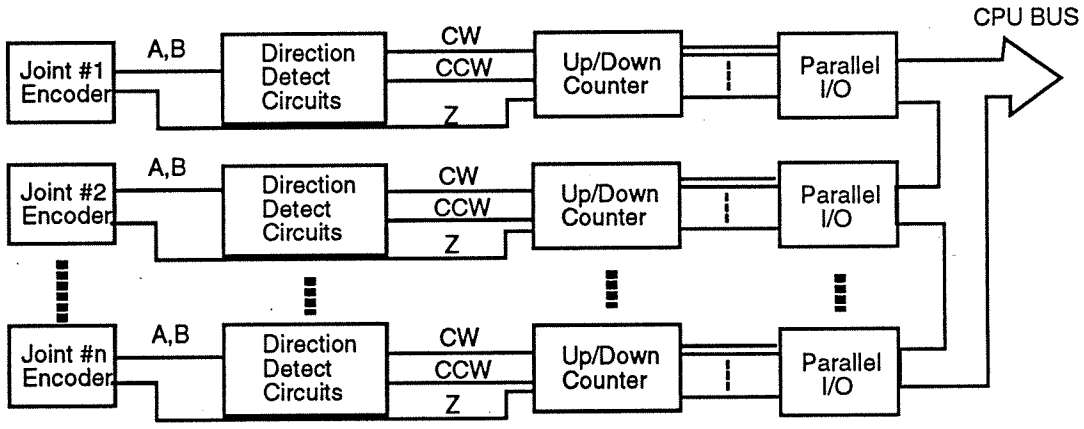


Figure 2.6 Encoder Interface Block Diagram

Motor Controller

There are many motor control technologies available today, with a wide variety of applications. The current state-of-the-art in motor control technology is the Flexible Servo Control (FSC) which directly controls each driving motor (Yamazaki, 1987). A block diagram of the FSC is shown in Fig. (2.7). This system can control three types of motors , namely DC, AC synchronous and AC induction, with one unique piece of

hardware and dedicated software. The core of the FSC is the custom designed VLSI chip, FSP-2, designed by Yamazaki (1988). The FSP-2 has been developed by using the ASIC (Application Specific Integrated Circuit) technology and can do the current loop servo control calculation including the feedforward control within 60 microseconds.

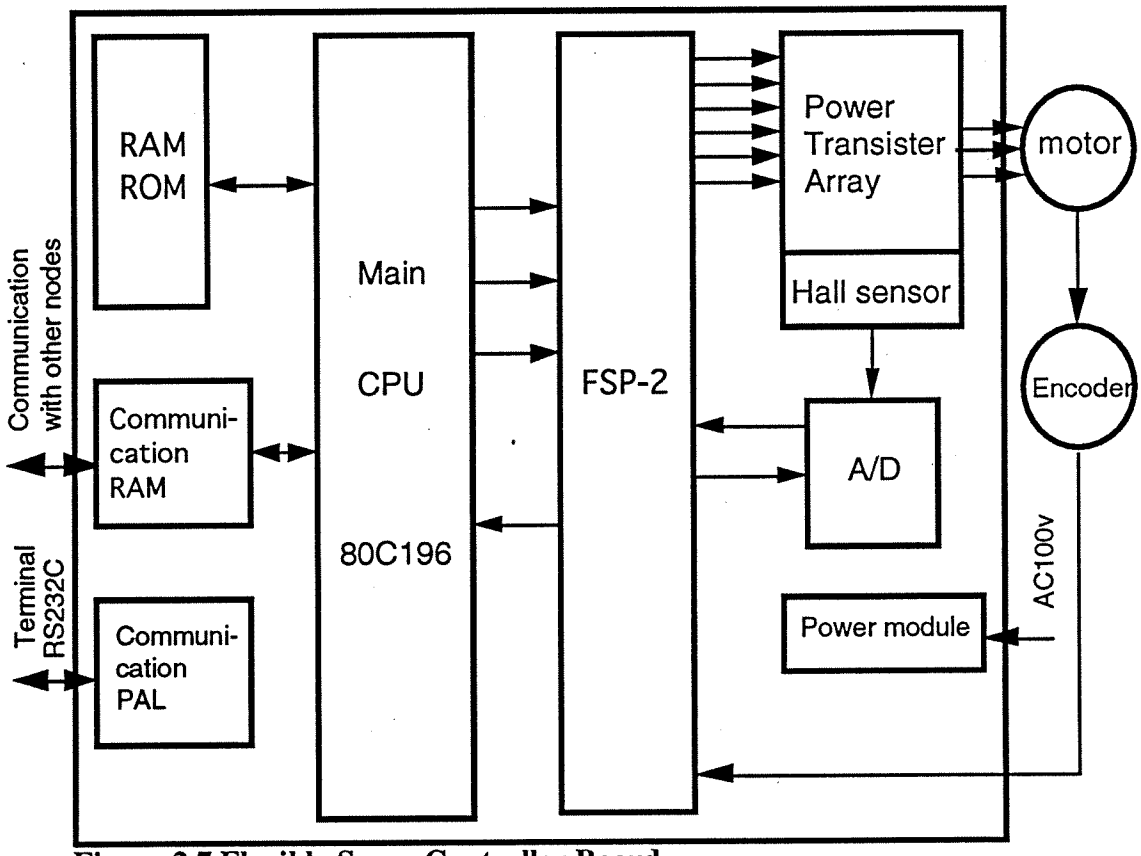


Figure 2.7 Flexible Servo Controller Board

Motor Control Algorithm

Basically, there are three different variables to control: the motor position, the motor speed, and the current supplied to the motor. The control target is to achieve a desired motion with sufficient internal stability. To detect these three variables, only two sensors are used: a Hall sensor to detect the current and an incremental optical encoder to detect the position and the velocity.

The adopted control configuration is the "multi-loop feedback" configuration shown in Fig. (2.8). Once a position command is given for generating a motion, the position loop controller calculates a velocity command so as to eliminate the position error. This velocity command is supplied to the velocity loop and having a similar treatment in the velocity loop and in current loop, the final voltage output is supplied via the PWM (Pulse Width Modulation) circuit to the motor.

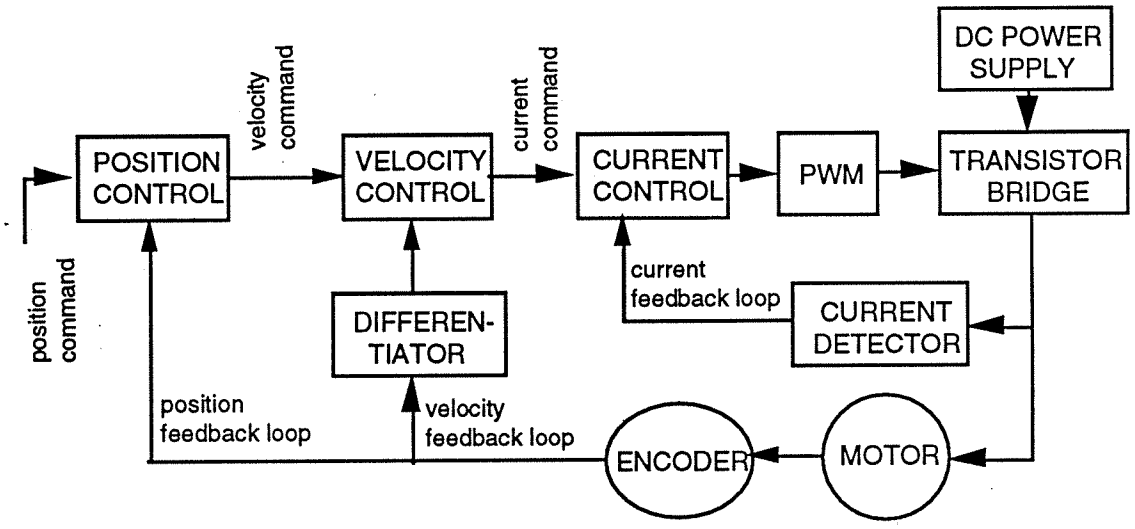


Figure 2.8 Multi-loop Feedback Control Algorithm

In typical machine motion control, a Proportional control algorithm is used for the position loop and a Proportional Integral control algorithm is used for the velocity and current loop. The Integral control action has the powerful feature that it eliminates the steady state error. This rule can be expanded as "Integral action eliminates all slow disturbances."

The primary task of the current controller is to control the current in a R-L circuit. A secondary task is to compensate the back voltage generated inside the motor proportional to the motor velocity. This back voltage is a "slow" disturbance compared

with the current behavior which can thus completely be compensated for by using integral control action.

The primary task of the velocity controller is to control the velocity of the rotor inertia. Note that in most applications the inertia of the load is negligible with respect to the motor inertia. A secondary task is to compensate for external forces such as frictional forces, load forces, etc. In most applications, these external force can be consider as "slow" disturbances compared with the mechanical behavior of the rotor inertia and thus in principle they can be completely compensated.

2.5 Local Sensor

A variety of sensor technologies have been researched in order to select a sensing system which best meets the requirements of the U.C. Davis SHRP automated crack sealing machine. Although the sensor parameters described here pertain to highway crack sealing, each specific application such as highway maintenance, toxic clean-up, etc. may have a different set of requirements.

The purpose of the local sensor is to detect cracks in pavement surfaces, so that a reference error can be produced which is used to help control the TMR. The local sensor will locate crack position and then send that information to the position controller, and along with the tracking system data the TMR can be controlled very accurately. Table (2.3) shows the sensor's main requirements.

A variety of sensor technologies have been researched in order to select a sensing system which best meets the requirements in this table. The sensor which is selected should be the most cost effective, off-the-shelf component which meets all the requirements. With this in mind, sensors which were considered for the task of local sensing are shown below. An extensive literature search was completed to gather background information in crack detection and tracking (Krulewich and Velinsky, 1992). Table (2.4) represents the variety of sensors considered (Jing, et al., 1990).

RESOLUTION ALONG SCAN	.0625 in. (1.588 mm)
VERTICAL RESOLUTION	.0625 in. (1.588 mm)
ACCURACY OF CRACK POSITION	.125 in. (3.175 mm)
FIELD OF VIEW	12 in. (304.8 mm) minimum
DISTANCE TO SURFACE	4 in. (101.6 mm)
SYSTEM RESPONSE FREQUENCY	18 Hz
HUMIDITY	0 to 85%
VIBRATION	3 g peak from 15 Hz to 100 Hz
SHOCK	10 g
OPERATING TEMPERATURE	0.0 to 71.1 degrees C
SENSOR MUST DISTINGUISH BETWEEN THE FOLLOWING	* previously filled cracks * oil spots * shadows * actual cracks

Table 2.3 Sensor Requirements for Crack Detection

Sensing technologies were divided into two categories; tactile and non-tactile. Tactile sensors recognize a crack through direct physical contact with the pavement. The main disadvantage of tactile sensing is that the sensors wear due to the physical contact and therefore have a shortened life. Non-tactile sensors on the other hand have no physical contact between the sensor and the pavement. Non-tactile sensors have the ability to detect remotely, and therefore have no inherent wear problems. Because of this longer life, a non-tactile sensor is preferred for the local sensing.

A complete description of each of the methods in table (2.4) can be found in Krulewich and Velinsky (1992). Based on this study, the most suitable sensing system is a laser range finding sensor based on the principle of triangulation, which proved to

perform well on both Asphalt Concrete (AC) and Portland Cement Concrete (PCC). It was concluded that a laser vision system utilizing structured light is most economical in addition to having a simpler design with less moving parts. Furthermore, the structured light sensor package is more compact and lighter. Therefore, it is determined that laser vision system using structured light is the optimum sensor for crack detection in pavement at the current time (Krulewich and Velinsky, 1992).

TACTILE	microswitches time domain reflectrometry
NON-TACTILE	spectral analysis capacitive inductive pneumatic far infrared temperature visible array acoustic ultrasonic optical modulated light light intensity displacement

Table 2.4 Sensor Technologies

2.6 TMR Conceptual Layout

The purpose of this section is to present the general conceptual layout of the TMR. The configuration will be presented in a conceptual context, and therefore not introducing actual dimensions or parameters. In the previous sections, specific configurations were chosen for the wheel layout, tracking method, control system, and local sensor. Each of these will now be presented, as they relate, in the conceptual layout. Additionally, specific details concerning the conceptual layout will be addressed.

Fig. (2.9) shows the conceptual layout of the TMR system, which has the TMR working in close proximity to the support vehicle for purposes of power, materials, etc., and allowing for the precise positional measurement by use of the mechanical linkage. Additionally, the mechanical linkage provides an useful method to route the power and material hoses to the TMR platform. The TMR platform is self-propelled and is used as the load carrying device. The workspace in this conceptual layout assumes that the linkage links can rotate a full 360 degrees with no restrictions and therefore easily reach the layout depicted.

Fig. (2.10) shows the general conceptual layout of the TMR platform, which is used as the load supporting and driving device for the TMR system. The wheel configuration is similar to Newt (Muir and Neuman, 1986), having two conventional drive wheels and two castors. Each drive wheel is connected through a planetary right angle gearbox to an electric motor. Having each wheel independently driven allows the platform to be controlled to follow any path in a plane. The local sensor is mounted at the front of the platform to detect errors in path following. Lastly, the linkage connection point is where the linkage and the platform are connected.

In order for the TMR system to operate properly, there are specific details which must be addressed while designing the TMR. The TMR should be able to follow a path on a surface which is not perfectly flat, which is usually the case. This requires a linkage configuration which is somewhat compliant, so that proper wheel/ground contact is made

at all times. The linkage structure must handle vibration problems and also bending due to its own weight. Power, material, etc., lines must be routed in such a way that they do not interfere with the mobility and workspace of the TMR. The platform wheels must be mounted as accurately as possible, so that the kinematic control equations are valid. The above details and other related topics should be addressed in detail in the design stage.

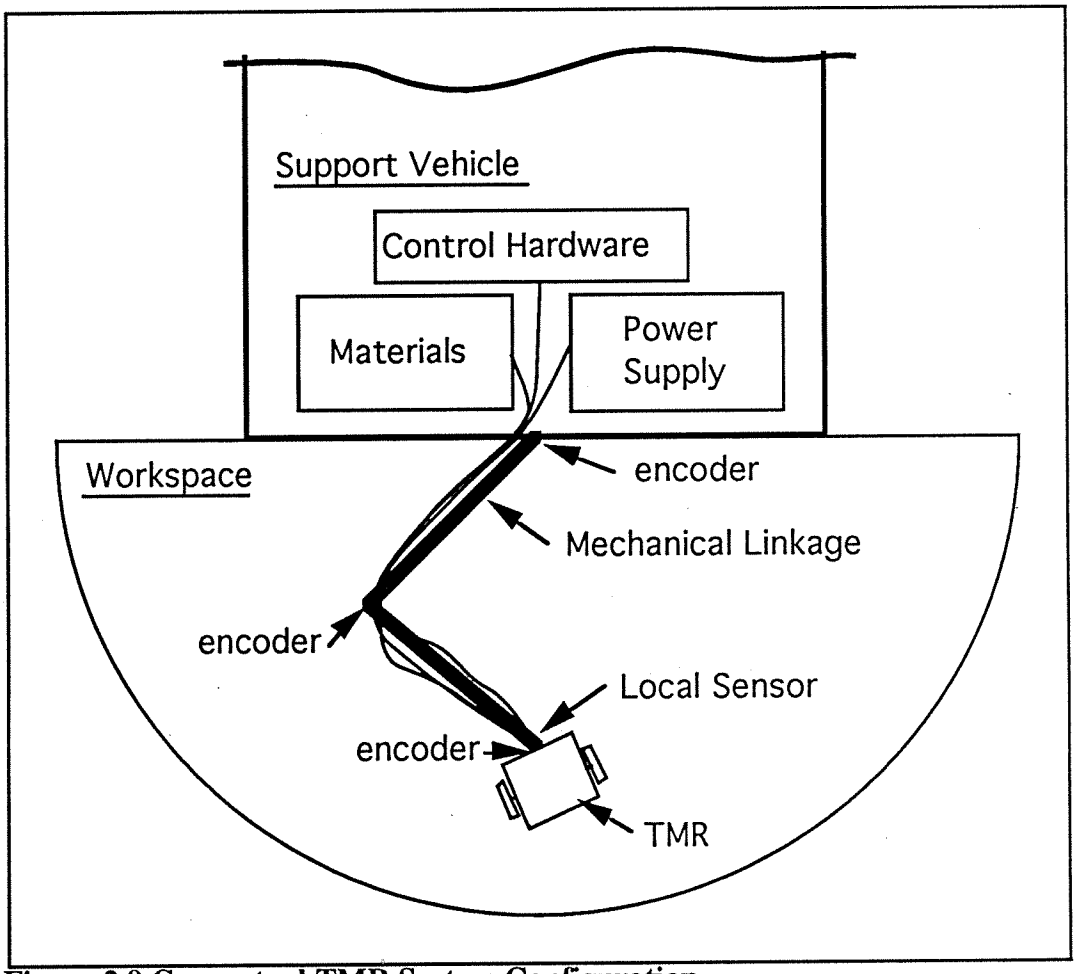


Figure 2.9 Conceptual TMR System Configuration

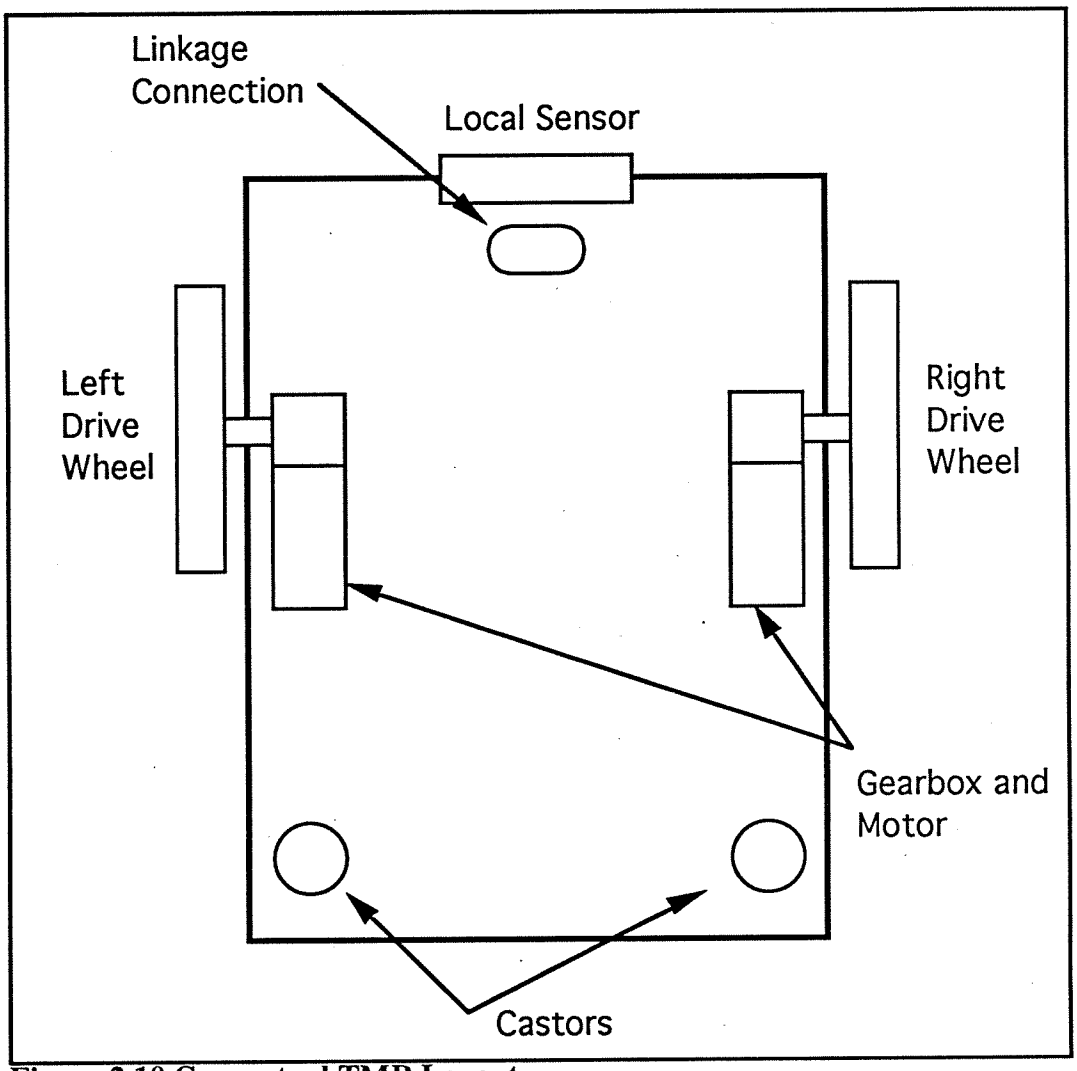


Figure 2.10 Conceptual TMR Layout

2.7 Conclusion

In this chapter the general configuration of the TMR was presented, which included areas such as the wheel configuration, tracking configuration, control configuration, and local sensor. There are two possibilities to choose from for the wheel configuration of the TMR, the omnidirectional wheel and the conventional wheel. Both wheels offer comparable path following abilities, however, factors such as the tractive

force and robustness are better for the conventional wheel. So, it is was concluded to use a configuration similar to Newt which uses conventional wheels.

Various kinds of sensing methods are considered for our tracking system, such as infrared, ultrasonic, etc. It was determined that a linkage type system would provide the best type of tracking method, due to it accuracy and simplicity. This mechanical connection also allows a method to route hoses and lines for material and power delivery.

The TMR control system will use the 80486 processor and it will communicate to the actuator controllers through the RS-232C link. Appropriate communication methods between the controller and the local sensor and the ICU (integration and control unit) will allow the TMR to be controlled properly. The BLDC electric motor is chosen for the actuators, after considering numerous approaches.

The Flexible Servo Controller (FSC), which was developed by Yamazaki, will be used for the motor controller. The motors will be controlled based upon PID control algorithm.

Based on requirements set by earlier work, it was determined that a laser range finding sensor based on the principle of triangulation would be most suitable for the sensing technology.

CHAPTER 3 - WORKSPACE ANALYSIS

3.1 Introduction

The workspace literature review is focused on determining a method to best optimize the usable workspace of the TMR, based on the total area of the workspace. This review is directly related to the use of mechanical manipulators (i.e., linkages) for optimization, which was chosen in the last chapter as the tracking method. In the literature, there are both numerical and analytical methods for this optimization (Gosselin, 1991; Jo, 1989; Gupta and Roth, 1982; Gupta, 1986). However in our situation the numerical approach is most beneficial.

An optimization method developed by Gosselin and Guillot (1991) for planar manipulators, focuses on synthesizing manipulators whose workspace is as close as possible to a prescribed workspace. The procedure begins with a geometric description of both the desired and actual workspaces, and then uses an optimization method which is based upon the intersection of the two areas. Jo and Haug (1989) developed an approach to numerical analysis of workspaces of multi-body mechanical systems. The general method is based on manifold theory and computational continuation methods. Other work by Gupta and Roth (1982) and Gupta (1986) discuss general configuration constraints of mechanical linkages. Other topics include kinematic design considerations of manipulators given a desired workspace and quantitative evaluation methods.

The purpose of this chapter is to describe the general linkage and workspace analysis used in designing the TMR system. Additionally, the general workspace configuration will be addressed which details the actual layout. Lastly, an error analysis will be used on the linkage configuration, generating an error equation which can be used for a given set of parameters. This error can then be used in determining the proper encoder resolution and link length accuracy's needed.

3.2 General Workspace/Parameters

In designing the configuration for the linkage system, initially, some general constraints for the workspace had to be addressed. It was decided that two main workspace areas would be needed in order to have a versatile and usable system. The first workspace should be 12' wide, therefore allowing the system to address a full lane width. The second area should be 8' wide extending significantly deeper than the 12' area, thus allowing a secure workspace which is located within the width confines of the transport vehicle (see Fig. 3.1). The second workspace may allow work to be accomplished without a lane closure. Although the workspaces are geared towards crack sealing, the workspace can be tailored to any particular task. Some additional constraints/parameters are also applicable in the general layout; the linkage system must be designed so that it can be stored easily within the confines of the truck and lastly the linkage must not have any singularity points within the workspace. Fig. (3.1) shows the conceptual workarea assuming the manipulator link lengths are equal and each joint allows a full 360 degrees of rotation without restriction.

Using the above criteria, and the design method similar to (Gosselin, 1991), the general linkage configuration will be selected. The objective of the optimization procedure described here is to obtain the geometric parameters of the manipulator that will lead to a workspace that is as close as possible to being identical to the prescribed one. The optimization procedure which will be performed over the manipulator's kinematic parameters, will determine the maximum usable workspace by examining the ratio of actual workspace to total workspace produced by the linkage. This will be accomplished by using

$$n = \frac{A_{workspace}}{A_{total}} * 100, \tag{3.1}$$

$$k_1 * A_{transverse} + k_2 * A_{longitudinal} - A_{overlap} = A_{workspace}, \text{ and} \tag{3.2}$$

$$k_1 + k_2 = 1, \tag{3.3}$$

where,

n = percentage of usable workspace covered by the manipulator,

$k_i (i = 1,2)$ = weighting factors,

$A_{transverse}$ = transverse workarea,

$A_{longitudinal}$ = longitudinal workarea,

$A_{overlap}$ = the area where the transverse and longitudinal workareas overlap,

$A_{workspace}$ = prescribed workspace, and

A_{total} = total area produced by manipulator.

In choosing the proper kinematic configuration, many different technical papers were reviewed, such as Gosselin and Guillot (1991), Jo and Haug (1989), Gupta (1986), and Gupta and Roth (1982). In these papers, the general conclusion was that for simple planar workspaces, a general serial two degree of freedom manipulator would allow the largest usable workarea and the simplest configuration.

In order to optimize Eqn. (3.1), a simple program was written which calculates the prescribed workspace and total area, for a variety of manipulator lengths. The program calculates Eqn. (3.1) for a variety of manipulator lengths, so that a corresponding length can be determined which optimizes the percentage of usable workspace. Although different weighting factors can be placed on each workspace (longitudinal or transverse) it was initially felt that they should be weighed equally. The weighing factors have the form of Eqns. (3.2) and (3.3). After running the program, with the above constraints, it was found that the optimal total length of the linkage was approximately seven feet. This corresponds to a workspace with a total area of 153 ft^2. The defined workspaces of 12 ft. and 8 ft. have a depth of 3.6 ft. and 5.7 ft. respectively, which can be seen in Fig. (3.1).

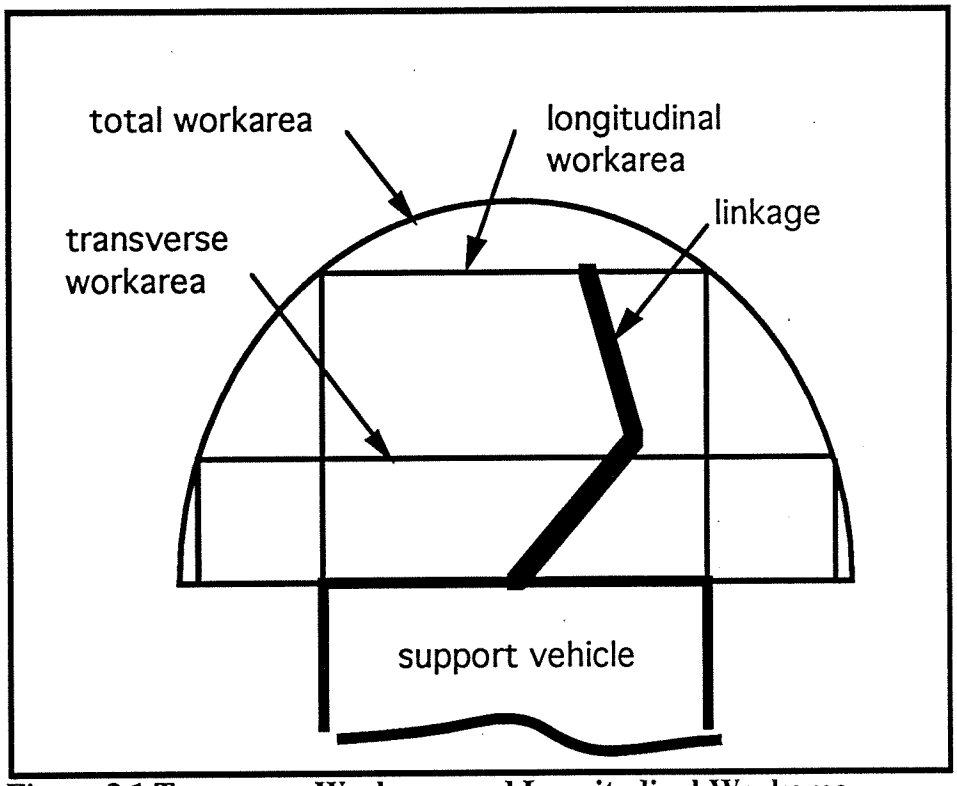


Figure 3.1 Transverse Workarea and Longitudinal Workarea

3.3 Error Analysis

In order to properly size and design the linkage, one must understand the general bias or systematic errors. These errors include items such as calibration errors, deformation errors, and limitations of system resolution to name a few. In this section, the limitations imposed by system resolution will be examined.

Using trigonometric identities and kinematic parameter uncertainties, the uncertainty equations will be derived for a simple planar two degree of freedom manipulator. The following kinematic equations represent the (x,y) location of the manipulator (see Fig. 3.2) as a function of θ_1 and θ_2, and are represented as

$$x = l_1 \cos(\theta_1) + l_2 \cos(\theta_1 + \theta_2) \text{ and} \tag{3.4}$$

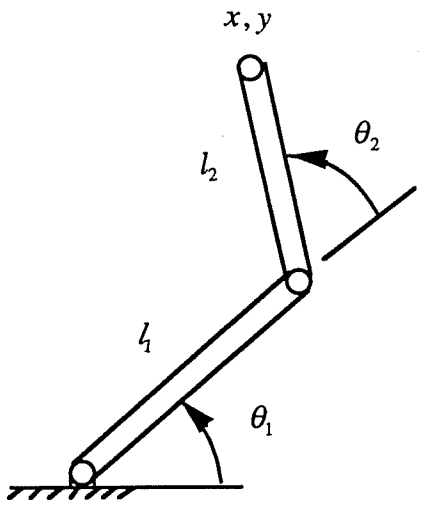


Figure 3.2 Planar Two Degree of Freedom Manipulator

$$y = l_1 \sin(\theta_1) + l_2 \sin(\theta_1 + \theta_2) \tag{3.5}$$

where,

x, y = the end effector position,

θ_1, θ_2 = link angles, and

l_1, l_2 = link lengths.

In this thesis, only the y-component uncertainty will be derived and the corresponding x-component uncertainty will be given. First, the uncertainties will be added to kinematic Eqn. (3.5), which is represented as

$$y' = (l_1 + \delta l_1)\sin(\theta_1 + \delta\theta_1) + (l_2 + \delta l_2)\sin(\theta_1 + \delta\theta_1 + \theta_2 + \delta\theta_2) \tag{3.6}$$

where,

δl_1 = discrete uncertainty of link l_1,

δl_2 = discrete uncertainty of link l_2,

$\delta\theta_1$ = discrete uncertainty of angle θ_1, and

$\delta\theta_2$ = discrete uncertainty of angle θ_2.

The following trigonometric function and small angle approximations,

$$\sin(a+b) = \sin(a)\cos(b) + \cos(a)\sin(b),$$
$$\sin(\delta\theta) \approx \delta\theta, \text{ and } \cos(\delta\theta) \approx 1$$

are now substituted into Eqn. (3.6) and reduced. This equation is now expressed as

$$y' = (l_1 + \delta l_1)[\sin(\theta_1) + \cos(\theta_1)\delta\theta_1] + (l_2 + \delta l_2)[\sin(\theta_1 + \theta_2) + \cos(\theta_1 + \theta_2)(\delta\theta_1 + \delta\theta_2)]. \quad (3.7)$$

The error equation is then the difference between Eqns. (3.7) and (3.5), and is represented as

$$Y_{error} = y' - y = l_1\cos(\theta_1)\delta\theta_1 + \delta l_1 \sin(\theta_1) + l_2\cos(\theta_1 + \theta_2)(\delta\theta_1 + \delta\theta_2) + \delta l_2 \sin(\theta_1 + \theta_2). \quad (3.8)$$

Now, with the completion of the error equation, the uncertainty equation is the quotient of Eqn. (3.8) and (3.5), and is expressed as

$$Y_{uncert.} = \frac{Y_{error}}{y}$$

$$= \frac{l_1\cos(\theta_1)\delta\theta_1 + \delta l_1 \sin(\theta_1) + l_2\cos(\theta_1 + \theta_2)(\delta\theta_1 + \delta\theta_2) + \delta l_2 \sin(\theta_1 + \theta_2)}{l_1\sin(\theta_1) + l_2\sin(\theta_1 + \theta_2)} \quad (3.9)$$

Evaluating Eqn. (3.9) numerically will give a general description of the uncertainty for a given set of parameters. It should be noted that this uncertainty, Eqn. (3.9) is the y-

component and that a corresponding x-component error can be derived. Similar derivation yields the x-component uncertainty equation as

$$X_{uncert.} = \frac{X_{error}}{x}$$

$$= \frac{-l_1 \sin(\theta_1)\delta\theta_1 + \delta l_1 \cos(\theta_1) - l_2 \sin(\theta_1 + \theta_2)(\delta\theta_1 + \delta\theta_2) + \delta l_2 \cos(\theta_1 + \theta_2)}{l_1 \cos(\theta_1) + l_2 \cos(\theta_1 + \theta_2)} \quad (3.10)$$

Eqns. (3.9) and (3.10) can now be used to show the positional uncertainty as a function of angular orientation. Values for encoders and the link length uncertainties must be included.

3.4 Conclusion

In this chapter the general linkage and workspace analysis used in designing the TMR system was presented. A literature review was completed which focused on optimization methods concerning both mechanical linkages and workspaces. A mechanical planar two degree of freedom linkage configuration was chosen based on its large usable workarea and simple configuration. It was decided that two workspaces would be ideal for our applications, and a numerical optimization method was implemented to size the linkage. The optimum linkage length for the chosen workspace was determined to be approximately 7 ft. Lastly, uncertainty equations were derived for the linkage configuration, which can be used for a given set of parameters. These uncertainty equations can be used in determining the proper encoder resolution and link length accuracy's needed.

CHAPTER 4 - TMR MODELLING

4.1 Introduction

The purpose of this chapter is to develop a computer model which will facilitate the design of the Tethered Mobile Robot. The TMR platform model will be used in the next chapter to simulate the path following abilities of the TMR by using an optimal control algorithm. The first part of this chapter will cover the derivation of the equations of motion of the TMR model, using an analytical method based on Kane's method. In the second part, these equations will be converted into state-space form for use in later simulations.

4.2 Derivation of Dynamical Equations of Motion

The method used for the derivation of the dynamical equations of motion is called Kane's method (Kane and Levinson, 1985). Kane's method, which is also known as Lagrange's form of D'Alemberts principle, is based on the generalized speeds of the system rather than its displacements, as in the Euler-Lagrange method for example, and this formulation directly results in a form that is solvable on a computer. Kane's method can be based on either an inertial coordinate reference frame or a relative coordinate frame, but has the disadvantage in either case of requiring the elimination of constraint variables during the formulation. The dynamical equations of motion will now be derived for the TMR model.

Nomenclature

The nomenclature listed below, describes the variables used in deriving the equations of motion, and are expressed as

q_i = Generalized coordinates,

u_i = Generalized speeds,

R_i, N_i = Represent unit vectors, and the corresponding subscript represents their direction,

${}^i\omega^j$ = Angular velocity of jth body in the ith reference frame,

${}^i v^j$ = Linear velocity of the jth point in the ith reference frame,

${}^i\alpha^j$ = Angular acceleration of the jth body in the ith reference frame,

${}^i a^j$ = Linear acceleration of the jth point in the ith reference frame,

R_i = Unit vectors attached to the TMR frame,

N_i = Unit vectors fixed in space,

R^* = Center of mass of TMR frame,

E^* = Center of mass of TMR's right wheel,

B^* = Center of mass of TMR's left wheel,

$\hat{E}$ = TMR's right wheel ground contact point,

$\hat{B}$ = TMR's left wheel ground contact point,

l = Distance equal to one-half the wheel base,

d = Radius of drive wheels,

m_R = Mass of TMR frame, and

m = Mass of right wheel and left wheel.

Assumptions

The following assumptions were made when deriving the equations of motion.

- Castor effects were neglected.
- TMR frame was assumed to remain parallel to the ground, therefore motor reaction torque's may be neglected.

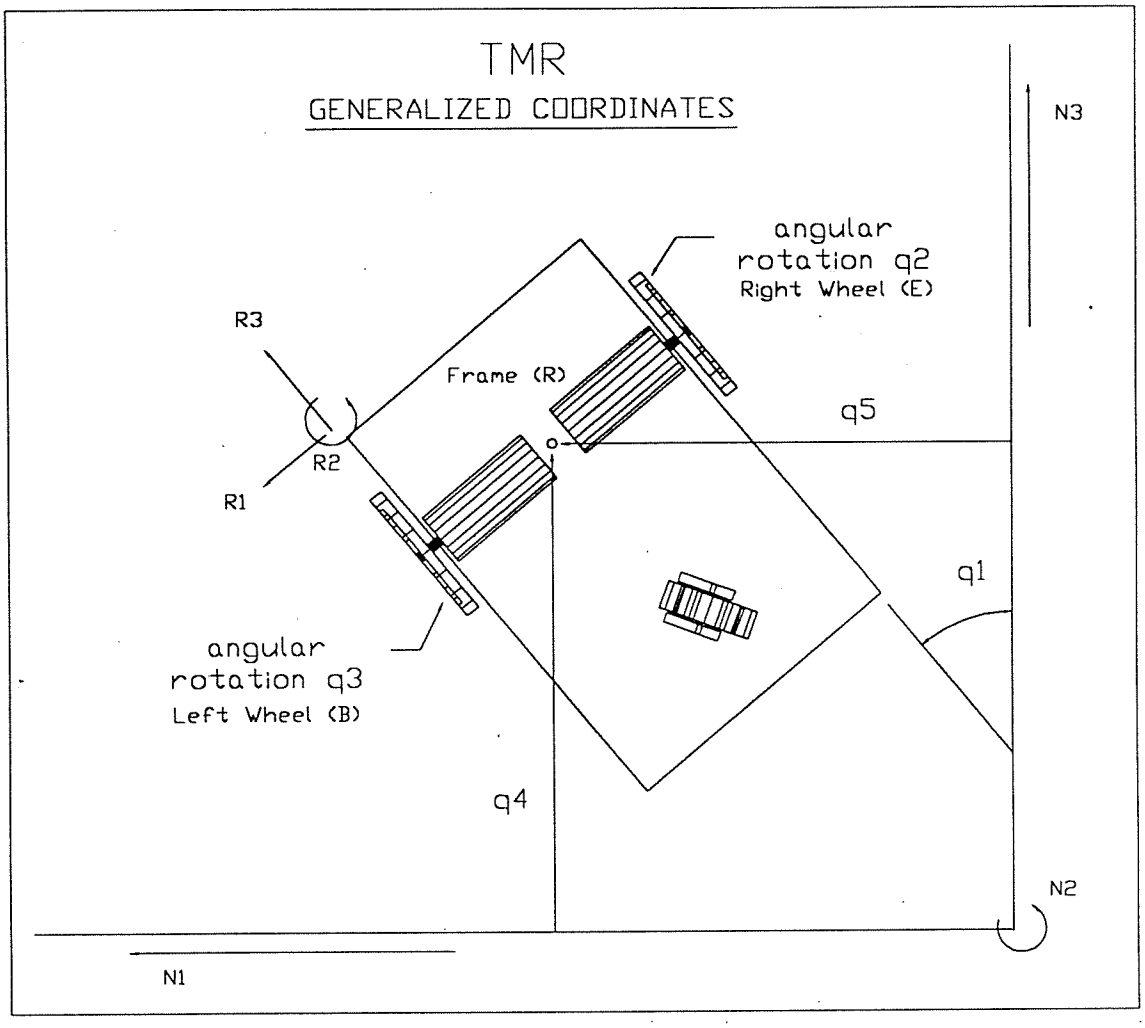


Figure 4.1 TMR generalized coordinates

The generalized coordinates, which represent the angular or linear displacements of each body represented in the model relative to a reference frame, are

$$q_1, q_2, q_3, q_4, q_5$$

where,

q_1 = the angular orientation of the TMR in reference frame N,

q_2 = the angular position coordinate of the right wheel relative to the TMR frame,

q_3 = the angular position coordinate of the left wheel relative to the TMR frame,

q_4 = the N_3-component of the TMRs' center of mass position, and

q_5 = the N_1-component of the TMRs' center of mass position.

With the generalized coordinates chosen, the generalized speeds are generated, which are the first time derivatives of the generalized coordinates and are expressed as

$$\dot{q}_1 = u_1, \dot{q}_2 = u_2, \dot{q}_3 = u_3, \dot{q}_4 = u_4, \dot{q}_5 = u_5$$

where,

u_1 = the angular speed of the TMR,

u_2 = the angular speed of the right wheel,

u_3 = the angular speed of the left wheel,

u_4 = the N_3-component linear velocity of the TMRs' center of mass, and

u_5 = the N_1-component linear velocity of the TMRs' center of mass.

Next relevant velocities are determined, which represent the angular and linear velocities of each of the bodies; right wheel, left wheel, and frame. They are expressed as

$$^N\omega^R = u_1 R_2, \tag{4.1}$$

$$^N\omega^E = u_1 R_2 + u_2 R_1, \tag{4.2}$$

$$^N\omega^B = u_1 R_2 + u_3 R_1, \tag{4.3}$$

$$^N v^{R^*} = u_4 N_3 + u_5 N_1, \tag{4.4}$$

$$^N v^{E^*} = u_4 N_3 + u_5 N_1 + l u_1 R_3, \tag{4.5}$$

$$^N v^{B^*} = u_4 N_3 + u_5 N_1 - l u_1 R_3, \tag{4.6}$$

$$^N v^{\hat{E}} = u_4 N_3 + u_5 N_1 + l u_1 R_3 - d u_2 R_3, \text{ and} \tag{4.7}$$

$$^N v^{\hat{B}} = u_4 N_3 + u_5 N_1 - l u_1 R_3 - d u_2 R_3, \tag{4.8}$$

where, the last two Eqns. (4.7) and (4.8), represent the velocity at each wheels ground contact point, respectively. Some general constraints will now be applied to the drive wheels. Due to the nature of wheel slippage and the difficulties in modelling it, the drive wheels have been assumed to be in a no-slip condition. This is a reasonable engineering assumption in light of the TMRs' characteristics and the slow speeds of travel. Therefore, each wheel's contact point velocity is set equal to zero, as

$$ {}^{N}v^{\hat{E}} = 0, \text{ and } \quad {}^{N}v^{\hat{B}} = 0. \tag{4.9} $$

Using the constraint equations above and applying them to Eqns. (4.7) and (4.8), the non-holonomic constraint equations will be derived. First, each of these velocities is expressed in terms of the unit vectors R_1, R_2, R_3. The transformation equations, which convert the N frame to the R frame are

$$ N_3 = -sq_1 R_1 + cq_1 R_3, \text{ and} \qquad N_1 = cq_1 R_1 + sq_1 R_3. \tag{4.10} $$

From Eqns. (4.7), (4.9), and (4.10), and setting components equal to zero, we have

$$ -sq_1 u_4 + cq_1 u_5 = 0, \text{ and} \qquad cq_1 u_4 + sq_1 u_5 + lu_1 - du_2 = 0. \tag{4.11} $$

Also from Eqns. (4.8), (4.9), and (4.10), and setting components equal to zero, we have

$$ -sq_1 u_4 + cq_1 u_5 = 0, \text{ and} \qquad cq_1 u_4 + sq_1 u_5 - lu_1 - du_3 = 0. \tag{4.12} $$

Now by solving Eqns. (4.11) and (4.12), the non-holonomic constraint equations can be determined. By applying the no-slip condition to the drive wheels, the non-holonomic constraint equations become

$$u_3 = (-2lu_1 + du_2) / d, \tag{4.13}$$

$$u_4 = (-lu_1 + du_2)cq_1, \text{ and} \tag{4.14}$$

$$u_5 = (-lu_1 + du_2)sq_1. \tag{4.15}$$

Now the above non-holonomic constraint equations will be substituted into Eqns. (4.1) through (4.6). With this substitution, the kinematic equations will only have two generalized speeds, reduced from the earlier number of five, and the following equations result:

$$^N\omega^R = u_1 R_2, \tag{4.16}$$

$$^N\omega^E = u_1 R_2 + u_2 R_1, \tag{4.17}$$

$$^N\omega^B = u_1 R_2 + (\frac{-2lu_1 + du_2}{d})R_1, \tag{4.18}$$

$$^N v^{R*} = (-lu_1 + du_2)cq_1 N_3 + (-lu_1 + du_2)sq_1 N_1, \tag{4.19}$$

$$^N v^{E*} = (-lu_1 + du_2)cq_1 N_3 + (-lu_1 + du_2)sq_1 N_1 + lu_1 sq_1 N_1 + lu_1 cq_1 N_3, \text{ and} \tag{4.20}$$

$$^N v^{B*} = (-lu_1 + du_2)cq_1 N_3 + (-lu_1 + du_2)sq_1 N_1 - lu_1 sq_1 N_1 - lu_1 cq_1 N_3. \tag{4.21}$$

Next, a partial velocity table is created in order to simplify the process of generating the dynamical equations of motion. Partial angular velocity and partial velocity are defined as follows. Let ω be the angular velocity of a body of interest, and v the velocity of a point. Then it is possible to express these vectors as

$$\omega = \sum_{r=1}^{n} \omega_r u_r + \omega_t \tag{4.22}$$

and

$$v = \sum_{r=1}^{n} v_r u_r + v_t, \tag{4.23}$$

where ω_r, v_r, ω_t, and v_t are functions of the generalized coordinates and time. The vector ω_r is called the r'th partial angular velocity and v_r is referred to as the r'th partial velocity.

	$r=1$	$r=2$
${}^N\omega_{ur}^R$	R_2	0
${}^N\omega_{ur}^E$	R_2	R_1
${}^N\omega_{ur}^B$	$R_2 - 2\frac{l}{d}R_1$	R_1
${}^N v_{ur}^{R*}$	$-lcq_1N_3 - lsq_1N_1$	$dcq_1N_3 + dsq_1N_1$
${}^N v_{ur}^{E*}$	0	$dcq_1N_3 + dsq_1N_1$
${}^N v_{ur}^{B*}$	$-2l(cq_1N_3 + sq_1N_1)$	$dcq_1N_3 + dsq_1N_1$

Table 4.1 Partial Velocity Table.

Now that the partial velocity table has been completed, the rest of the kinematics can be completed which includes the angular and linear accelerations. The angular accelerations were derived by taking the time derivative of the angular velocities, which has the form

$${}^i\alpha^j = \frac{{}^i d^i \omega^j}{dt} \tag{4.24}$$

where,

j = The body in question and

i = The reference frame.

The angular accelerations are expressed as

$$^{N}\alpha^{R} = \dot{u}_1 N_2, \tag{4.25}$$

$$^{N}\alpha^{E} = \dot{u}_1 N_2 + \left(-u_2 u_1 sq_1 + \dot{u}_2 cq_1\right)N_1 + \left(u_2 u_1 cq_1 + \dot{u}_2 sq_1\right)N_3, \text{ and} \tag{4.26}$$

$$^{N}\alpha^{B} = \dot{u}_1 N_2 + \{-\left(\frac{-2lu_1 + du_2}{d}\right)sq_1 u_1 + \left(\frac{-2l\dot{u}_1 + d\dot{u}_2}{d}\right)cq_1\}N_1 - \{+\left(\frac{-2lu_1 + du_2}{d}\right)cq_1 u_1 + \left(\frac{-2l\dot{u}_1 + d\dot{u}_2}{d}\right)sq_1\}N_3. \tag{4.27}$$

The linear acceleration was derived using the same method as above, having the form

$$^{i}a^{j} = \frac{^{i}d^{i}v^{j}}{dt} \tag{4.28}$$

where,

j = A point on a body and

i = The reference frame

and is expressed as

$$^{N}a^{R^*} = u_1(-lu_1 + du_2)(-sq_1 N_3 + cq_1 N_1) + (-l\dot{u}_1 + d\dot{u}_2)(cq_1 N_3 + sq_1 N_1), \tag{4.29}$$

$$
\begin{aligned}
{}^{N}a^{E^*} = {} & u_1(-lu_1 + du_2)(-sq_1N_3 + cq_1N_1) \\
& +(-l\dot{u}_1 + d\dot{u}_2)(cq_1N_3 + sq_1N_1) \\
& +lu_1^2(cq_1N_1 - sq_1N_3) + l\dot{u}_1(sq_1N_1 - cq_1N_3),
\end{aligned} \tag{4.30}
$$

and

$$
\begin{aligned}
{}^{N}a^{B^*} = {} & u_1(-lu_1 + du_2)(-sq_1N_3 + cq_1N_1) \\
& +(-l\dot{u}_1 + d\dot{u}_2)(cq_1N_3 + sq_1N_1) \\
& -lu_1^2(cq_1N_1 - sq_1N_3) - l\dot{u}_1(sq_1N_1 - cq_1N_3).
\end{aligned} \tag{4.31}
$$

With the kinematics completed, the next step is to determine $\frac{{}^{i}d^{i}H^{j/j^*}}{dt}$ for all bodies, which includes the right wheel, left wheel, and frame, where

$\frac{{}^{i}d^{i}H^{j/j^*}}{dt}$ = time derivative of body j's angular momentum.

Instead of taking the time derivative of each bodies angular momentum, a convient formula will be used (not shown here, see Kane and Levinson, 1985). To use this equation the following information will be needed; angular velocity of each body, angular acceleration of each body, and the inertia matrix for each body. The inertia matrix for each body has the following form, where

$$
I^*_{R1,R2,R3} = \begin{bmatrix} 2I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & I \end{bmatrix} \quad \text{is for bodies E and B, and} \tag{4.32}
$$

$$I^{*}_{R1,R2,R3} = \begin{bmatrix} I_1 & 0 & 0 \\ 0 & I_2 & 0 \\ 0 & 0 & I_3 \end{bmatrix} \quad \text{is for body R.} \tag{4.33}$$

In order to use these inertia matrixes in the formula given by Kane and Levinson, the ω's and α's derived early need to be expressed in the R frame, where

$$^{N}\omega^{R} = u_1 R_2, \tag{4.34}$$

$$^{N}\omega^{E} = u_1 R_2 + u_2 R_1, \tag{4.35}$$

$$^{N}\omega^{B} = u_1 R_2 + (\frac{-2lu_1 + du_2}{d}) R_1, \tag{4.36}$$

$$^{N}\alpha^{R} = \dot{u}_1 R_2, \tag{4.37}$$

$$^{N}\alpha^{E} = \dot{u}_2 R_1 + \dot{u}_1 R_2 - u_1 u_2 R_3, \quad \text{and} \tag{4.38}$$

$$^{N}\alpha^{B} = \dot{u}_1 R_2 + \left(\frac{-2l\dot{u}_1 + d\dot{u}_2}{d}\right)(R_1 - u_1 R_3). \tag{4.39}$$

Using the above equations, both the active forces and inertia forces will be determined. The active and inertia forces are forces applied to the system and forces that are generated by accelerations, respectively. The active force equation has the following form

$$F_r = \sum_{i=1}^{P_i} F_i \cdot {}^{N} v_{u_r}^{P_i} + \sum_{j=1}^{P_j} R_j \cdot {}^{N} v_{u_r}^{B_j^*} + \sum_{k=1}^{P_k} T_k \cdot {}^{N} \omega_{u_r}^{B_k}, \tag{4.40}$$

where,

F_i = the ith force acting on the ith particle,

${}^{N}v_{u_r}^{P_i}$ = the partial velocity of the ith particle,

R_j = the resultant force acting on the jth body,

${}^{N}v_{u_r}^{B_j}$ = the center of mass partial velocity of the jth body,

T_k = the kth torque acting on the kth body, and

${}^{N}\omega_{u_r}^{B_k}$ = the partial angular velocity of the kth body.

Using Eqn. (4.40), the following active force equations become

$$F_1 = -2T_B \frac{l}{d} \text{ and} \tag{4.41}$$

$$F_2 = T_B + T_E \tag{4.42}$$

where,

T_B = the torque applied by the motor to the right wheel and

T_E = the torque applied by the motor to the left wheel.

The inertia force equation is expressed as

$$F_r^* = \sum_{i=1}^{P_i} -m_i {}^{N}a^{P_i} \cdot {}^{N}v_{u_r}^{P_i} + \sum_{j=1}^{P_j} -m_j {}^{N}a^{B_j} \cdot {}^{N}v_{u_r}^{B_j^*} + \sum_{k=1}^{P_k} -\frac{{}^{N}d^{N}H^{B/B_k^*}}{dt} \cdot {}^{N}\omega_{u_r}^{B_k} \tag{4.43}$$

where,

m_i = the mass of the ith particle,

${}^{N}a^{P_i}$ = the acceleration of the ith particle,

${}^{N}v_{u_r}^{P_i}$ = the partial velocity of the ith particle,

m_j = the mass of the jth body,

${}^{N}a^{B_j^*}$ = the center of mass acceleration of the jth body,

${}^{N}v_{u_r}^{B_j^*}$ = the partial velocity of the jth body,

$\frac{{}^{N}d^{N}H^{B/B_k^*}}{dt}$ = the time derivative of the kth bodies angular momentum, and

${}^{N}\omega_{u_r}^{B_k}$ = the partial angular velocity of the kth body.

Using Eqn. (4.43), the inertia force equations were produced and the following equations result;

$$F_1^* = m_R(-l\dot{u}_1 + d\dot{u}_2)l + m\{2l(-l\dot{u}_1 + d\dot{u}_2) - 2l^2\dot{u}_1\}$$
$$-I_2\dot{u}_1 - I\dot{u}_1 + \left\{4I\left(\frac{-2I\dot{u}_1 + d\dot{u}_2}{d}\right)\frac{l}{d} - I\dot{u}_1\right\} \quad (4.44)$$

and

$$F_2^* = -m_R(-l\dot{u}_1 + d\dot{u}_2)d - 2m(-l\dot{u}_1 + d\dot{u}_2)d$$
$$-2I\dot{u}_2 - 2I\left(\frac{-2l\dot{u}_1 + d\dot{u}_2}{d}\right) \quad (4.45)$$

where,

m = the mass of right wheel and left wheel and

m_R = the mass of the TMRs' frame.

Next combining Eqns. (4.41) and (4.42), the active force equations, with Eqns. (4.44) and (4.45), the inertia force equations, will result in Kane's Equations which have the following form

$$F_r + F_r^* = 0. \quad \text{(Kane's Equation)}$$

Kane's Equations, which are the dynamical equations of motion for the TMR model in Fig.(4.1), are

$$m_R(-l\dot{u}_1 + d\dot{u}_2)l + m\{2l(-l\dot{u}_1 + d\dot{u}_2) - 2l^2\dot{u}_1\} - I_2\dot{u}_1$$

$$-I\dot{u}_1 + \left\{4I\left(\frac{-2I\dot{u}_1 + d\dot{u}_2}{d}\right)\frac{l}{d} - I\dot{u}_1\right\} - 2T_B\frac{l}{d} = 0 \tag{4.46}$$

and

$$-m_R(-l\dot{u}_1 + d\dot{u}_2)d - 2m(-l\dot{u}_1 + d\dot{u}_2)d - 2I\dot{u}_2$$

$$-2I\left(\frac{-2l\dot{u}_1 + d\dot{u}_2}{d}\right) + T_B + T_E = 0. \tag{4.47}$$

4.3 State-space Representation

The purpose of this section is to convert the dynamical equations of motion into state-space form. In order to solve for all of the generalized coordinates and speeds, both the dynamical equations and the constraint equations need to be solved. Therefore, to solve for all the unknowns simultaneously, first the two dynamical equations will need to be solved analytically. To solve for $\dot{u}_1$ and $\dot{u}_2$, the following equations

$$0 = a\dot{\tilde{z}} + b\tilde{u} \tag{4.48}$$

and

$$\dot{\tilde{z}} = -a^{-1}b\tilde{u} \tag{4.49}$$

will be used, where,

$$\dot{\tilde{z}} = \begin{bmatrix} \dot{u}_1 \\ \dot{u}_2 \end{bmatrix} \text{ (state vector)}, \tag{4.50}$$

$$a = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \text{ (system matrix)}, \tag{4.51}$$

$$b = \begin{bmatrix} -2l/d & 0 \\ 1 & 1 \end{bmatrix} \text{ (input matrix), and} \tag{4.52}$$

$$\tilde{u} = \begin{bmatrix} T_B \\ T_E \end{bmatrix} \text{ (input vector).} \tag{4.53}$$

The system matrix coefficients are expressed as

$$a_{11} = -(m_R + 4m)l^2 - I_2 - 2I\left(1 + 4\frac{l^2}{d^2}\right), \tag{4.54}$$

$$a_{12} = (m_R + 2m)dl + 4I\frac{l}{d}, \tag{4.55}$$

$$a_{21} = (m_R + 2m)dl + 4I\frac{l}{d}, \text{ and} \tag{4.56}$$

$$a_{22} = (m_R + 2m)d^2 - 4I. \tag{4.57}$$

Therefore, the equation needed to solve for $\dot{u}1$ and $\dot{u}2$ can be expressed as

$$\dot{\tilde{z}} = \begin{bmatrix} \dot{u}_1 \\ \dot{u}_2 \end{bmatrix} = \begin{bmatrix} a_{22} & -a_{21} \\ -a_{12} & a_{11} \end{bmatrix} \begin{bmatrix} -2lT_B/d \\ T_B + T_E \end{bmatrix} \begin{bmatrix} 1 \\ \det(a) \end{bmatrix} \tag{4.58}$$

where,

$$\det(a) = a_{11}a_{22} - a_{21}a_{12}. \tag{4.59}$$

Now, using $\dot{u}1$ and $\dot{u}2$, with the constraint Eqns. (4.13), (4.14), and (4.15), all the generalized coordinates and speeds can be calculated with the following state-space equations;

$$\dot{\tilde{x}} = A\tilde{x} + b\tilde{u} \quad \text{(state equation)} \tag{4.60}$$

and

$$y = c\tilde{x}, \quad \text{(output equation)} \tag{4.61}$$

where,

$$\dot{\tilde{x}} = \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \\ \dot{q}_4 \\ \dot{q}_5 \\ \dot{u}_1 \\ \dot{u}_2 \end{bmatrix} \text{ (time derivative of state vector),} \tag{4.62}$$

$$\tilde{x} = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \\ q_5 \\ u_1 \\ u_2 \end{bmatrix} \text{ (state vector),} \tag{4.63}$$

$$A = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & -2l/d & 1 \\ 0 & 0 & 0 & 0 & 0 & -lcq1 & dcq1 \\ 0 & 0 & 0 & 0 & 0 & -lsq1 & dsq1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \text{ (system matrix),} \tag{4.64}$$

$$b = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ b_{61} & b_{62} \\ b_{71} & b_{72} \end{bmatrix} \text{ (input matrix),} \tag{4.65}$$

$$\tilde{u} = \begin{bmatrix} T_B \\ T_E \end{bmatrix} \text{ (input vector), and} \tag{4.66}$$

$$c = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \text{(output matrix).} \tag{4.67}$$

The input matrix coefficients are expressed as

$$b_{61} = (-\{\ (m_R + 2m)d^2 - 4I\ \}\frac{2l}{d} - (m_R + 2m)dl + 4I\frac{l}{d}) / \det(a), \tag{4.68}$$

$$b_{62} = (-(m_R + 2m)dl + 4I\frac{l}{d}) / \det(a), \tag{4.69}$$

$$b_{71} = (\{\ (m_R + 2m)dl + 4I\frac{l}{d}\ \}\frac{2l}{d} - (m_R + 4m)l^2 - I_2 - 2I\left(1 + 4\frac{l^2}{d^2}\right)) / \det(a), \text{ and} \tag{4.70}$$

$$b_{72} = (-(m_R + 4m)l^2 - I_2 - 2I\left(1 + 4\frac{l^2}{d^2}\right)) / \det(a). \tag{4.71}$$

These equations can now be used to solve for the generalized coordinates and speeds, $\tilde{x}$, given that the initial conditions, $\tilde{x}_o$, are given or selected.

4.4 Conclusion

In this chapter, equations of motion were developed for the TMR platform model. The first part of the chapter covered the derivation of the equations of motion of the TMR model, using an analytical method based on Kane's method. In the second part, these equations were converted into state-space form, which will facilitate later simulations.

The TMR platform model will be used in the next chapter to simulate the path following abilities of the TMR by using an optimal control algorithm.

CHAPTER 5 - CONTROL MODEL

5.1 Introduction

The purpose of this chapter is to develop a control model, using Simulink software, and then simulate the path tracking ability of the TMR using this model. Initially a literature review of control algorithms will be completed, which focuses on algorithms for autonomous and semi-autonomous vehicles. After choosing an algorithm, each control block will be developed in detail. The control blocks will entail information on the algorithm chosen along with the equations of motion derived in chapter 4. The chapter will then conclude with tests verifying the control algorithms path tracking abilities.

5.2 Control Algorithms

The purpose of this section is to investigate many different control algorithms and then choose one to use in simulating the path tracking or reference following ability of the TMR model. The control algorithm literature review focused on general locomotion control methods for autonomous and semi-autonomous vehicles, which were needed to follow specified paths. The main difficulties in controlling an autonomous/semi-autonomous vehicle lie in the fact that vehicles usually have three degrees of freedom in position and orientation, while having only two degrees of freedom for motion control. It has also been found that modeling the vehicles dynamics is not of prime importance when developing control strategies for autonomous and semi-autonomous guided vehicles, due to the low speeds and accelerations at which they operate. In this section, a description of a variety of algorithms will be presented.

The path following or control algorithm for a mobile robot, directs the vehicle by following a reference position, trying to keep parameters like position error and velocity under predefined limits. The trajectory described by the algorithm for the vehicle to follow must be smooth because of stability restraints of the robot (Salichs, 1991).

Kanayama (1988) proposed a posture control loop that tracks the flow of reference postures from the path planner by generating the control variables (velocity and angular velocity) for the velocity control loop. The current posture is updated by dead reckoning, and an error vector is computed from the current and reference postures. This transformation generates the distances the vehicle should travel in the forward, lateral, and angular directions. These distances are then proportional integral derivative (PID) filtered to generate the outputs (velocity and angular velocity). This control algorithm seems to work very well.

Salichs (1991) presents three different control algorithms based on classical, fuzzy, and neural approaches, respectively, all of which were used to optimize and smooth the trajectory of the mobile robot. In the classical algorithm, the minimum radius of curvature is a function of the optimum velocity, and this velocity is computed in order to minimize the difference between the reference and the minimum distance. The optimum velocity is restricted between two limiting values, maximum and minimum, which is provided as inputs to the algorithm. The errors were PID filtered, and the vehicle speed was computed to minimize the error.

In the fuzzy control algorithm, there are a defined set of rules and fuzzy variables of triangular type representing the knowledge of a human driver to solve the tracking problem. The input is the position error, and depending on its value, the output variable curvature is derived. The other output variable which is controlled using fuzzy techniques is the velocity of the mobile robot. The results of this algorithm are very close to those produced by the classical method.

The last method presented by Salichs is that of the Neural Network. The neural network used is that of perceptron, which has three layers, the Input layer, the Hidden layer, and the Output layer. The Input layer acts as a linear discriminator, and divides the input space in different regions by means of hyperplanes. The Hidden layer acts as a

logical and function between the outputs of the input layer nodes. The Output layer acts as a logical or function between the outputs of the hidden layer.

Based on the three different control algorithms described above, classical, fuzzy, and neural, the most work and success has been with the use of classical control (Kanayama, 1988; Petrov, 1991; Salichs, 1991; Hongo, 1985; Kanayama, 1985). Although there is a large amount of literature available on these methods, the simulation to follow will be based on Kanayama (1988) control algorithms due to its simplicity and accuracy.

5.3 Control Model

The general configuration for the control model that controls the TMR is based on classical control and uses basically two main loops, an outer position loop and an inner velocity loop. This configuration is shown in Fig. (5.1). This figure shows three main blocks which are needed to properly control most systems. They are the position control block, velocity control block, and the dynamic model block. In addition, feedback loops are needed for both the position and velocity state variables, if closed loop control is used. The position reference is derived from the path in which the mobile robot is needed to follow.

Although Fig. (5.1) shows the basic components of a general control block diagram, the TMR's control system is somewhat different. The major difference is that this figure is based on a single-input single-output (SISO) configuration, where as the TMR control system will be based on a multi-input multi-output (MIMO) configuration. The MIMO results since the TMR has two BLDC motors that are controlled independently, yet they have a coupling effect on the TMR's trajectory or motion. In addition, to control the BLDC motors, there are four basic state variables that need to be monitored and adjusted. These state variables are the angular position and velocity of each motor. The control of the BLDC motors is accomplished in the velocity control loop. The position control loop is also based on the MIMO configuration, calculating the corresponding error in the x-

position, y-position, and orientation and then using some type of algorithm to calculate the reference velocities.

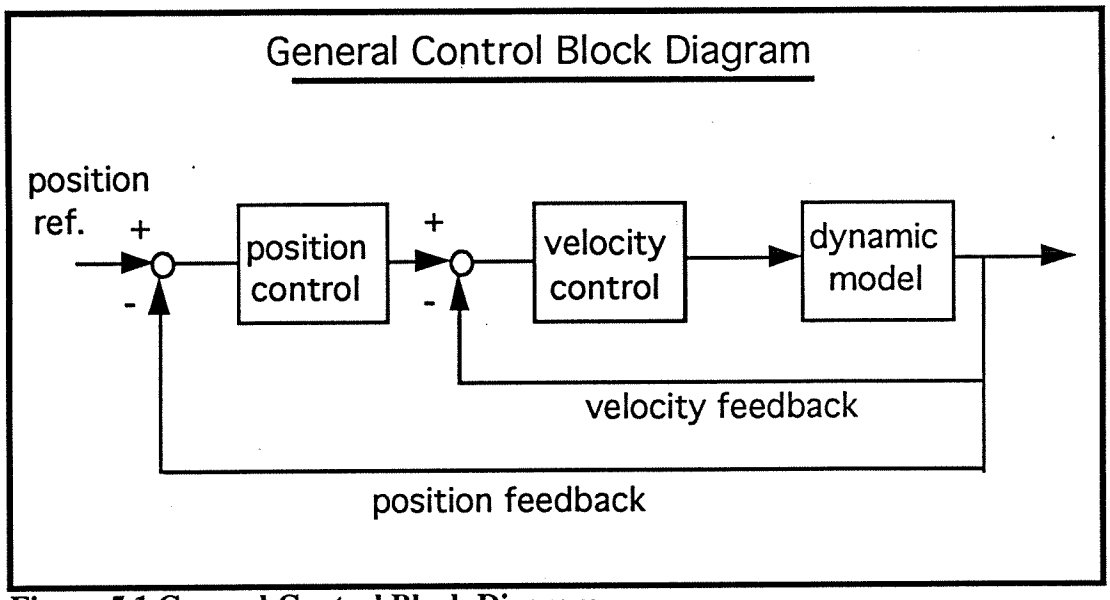


Figure 5.1 General Control Block Diagram

Description of Control Model

The following Figs. (5.2), (5.3), and (5.4), depict the basic control block diagrams used in simulating control of the TMR. The first Fig. (5.2), contains the complete control block diagram, while the remaining figures are sub-control block diagrams within diagram (5.2). Each of the block diagrams were completed using Simulink software.

Fig. (5.2) is the complete control block diagram used in simulating the path following abilities of the TMR. There are many different components which make up this system, such as the Position Reference, PID error manipulator, Error Control block (see Fig. 5.3), Velocity Control block (see Fig 5.4), and additional feedback lines.

The Position Reference arrangement allows almost any type of path to be generated, ranging from simple step and ramp paths to complex sinusoidal motions. Shown in Fig. (5.2), there are three generation blocks, one for the x-position, y-position, and θ-orientation, in which all are needed to completely describe the reference path. The

θ-orientation, is shown here as a step function, meaning that the orientation will remain constant throughout the path. The x-position and y-position are shown as ramps, which will make the position move linearly with time. This combination of references will create a path which starts at an (x,y) position of (0,0) and moves linearly at a 45 degree angle, relative to the x-axis.

The PID error manipulator, shown in Fig. (5.2), takes the errors, x-position, y-position, and θ-orientation and then filters them with proportional integral derivative (PID) control algorithms. Although this method was chosen, any control algorithm which can be expressed analytically can be used. This PID algorithm, which was developed by Kanayama (1988), calculates the errors, which will be discussed below, and then filters these errors to produce the corresponding velocity and angular velocity for the TMR. These equations are expressed as

$$v = k_1 e_x + k_2 \int e_x dt + k_3 \frac{de_x}{dt} \tag{5.1}$$

and

$$\omega = k_4 e_y + k_5 \int e_y dt + k_6 \frac{de_y}{dt} + k_7 e_\theta + k_8 \int e_\theta dt + k_9 \frac{de_\theta}{dt} \tag{5.2}$$

where,

v = the center of mass velocity of the TMR and

ω = the angular velocity of the TMR.

The gains $k_i (i = 1,2,\cdots,9)$ can be calculated by trial and error or some type of analytical method. Lastly, v and ω can be used to calculate the reference velocities of each wheel on the TMR using the following equations:

$$v_r = v + L\omega \tag{5.3}$$

$$v_l = v - L\omega \tag{5.4}$$

where,

$2L$ = wheel base.

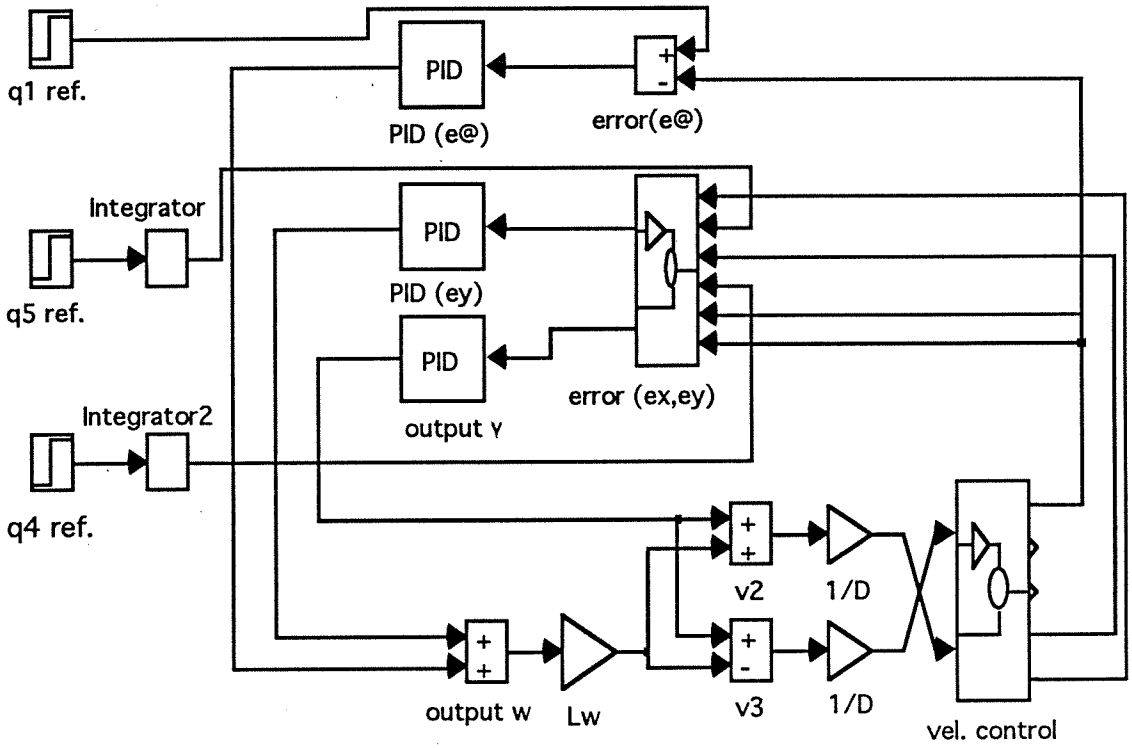


Figure 5.2 Control System Block

Velocity Control Block

The velocity control block, Fig. (5.3), is a sub-block of the complete control block Fig. (5.2). This control block is a MIMO control system, as can be seen in Fig. (4.3), which has two inputs and five outputs. The inputs are the reference velocities calculated from the error manipulator block and the five outputs consist of the following: two feedback wheel velocities, and three position state-variables used for the error

generation block. The major aspects in this control block are the state-equations and the PID controllers. The PID controllers are used to filter the velocity errors and then produce scaled torque values which are used as inputs to the state-equations. The state-equations, which represent the dynamics of the TMR, can be found in the "State-variable Representation" section in chapter 4.

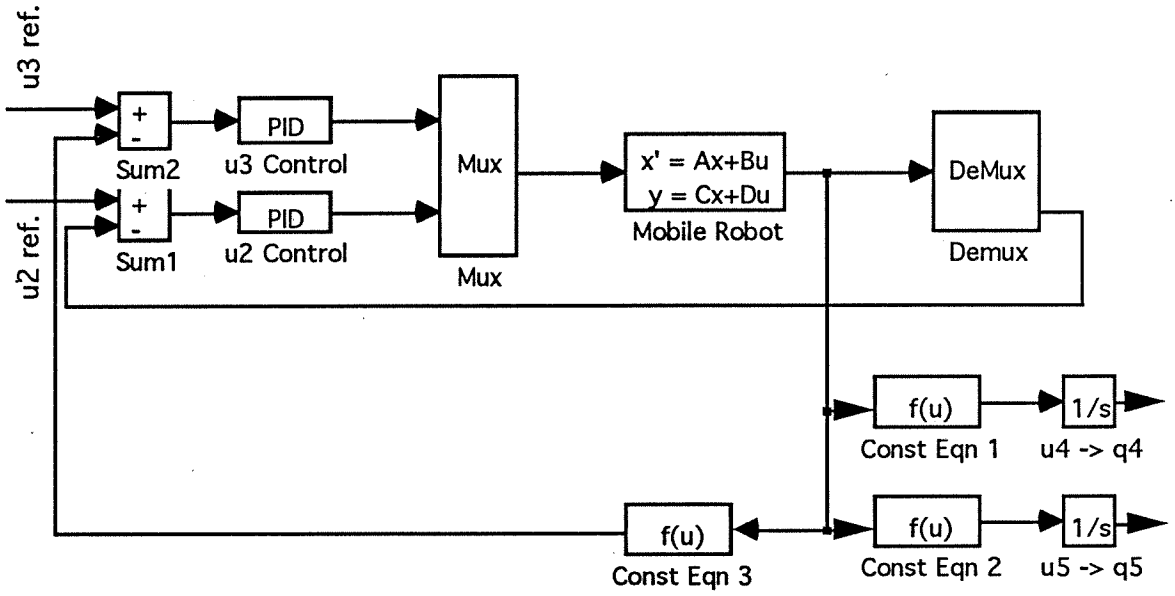


Figure 5.3 Velocity Control Block

Error Block

The error block used here is based on the method of Kanayama (1988), which uses dead reckoning as the sensor; however any other practical method may be used. The error is computed from the difference between the current position and the reference position. The block diagram shown in Fig. (5.4), calculates the following equations

$$e_x = (y_r - y_c) * \sin(\theta_c) + (x_r - x_c) * \cos(\theta_c), \qquad (5.5)$$

$$e_y = (y_r - y_c)*\cos(\theta_c) - (x_r - x_c)*\sin(\theta_c), \text{ and} \tag{5.6}$$

$$e_\theta = \theta_r - \theta_c, \tag{5.7}$$

where,

e_x = error in the x-position,

e_y = error in the y-position, and

e_θ = error in the orientation

for use in the complete control system simulation. The following notation

y_r = q5 ref., x_r = q4 ref., θ_r = q1 ref.,

y_c = q5, x_c = q4, and θ_c = q1

are presented to help clarify Fig. (5.4).

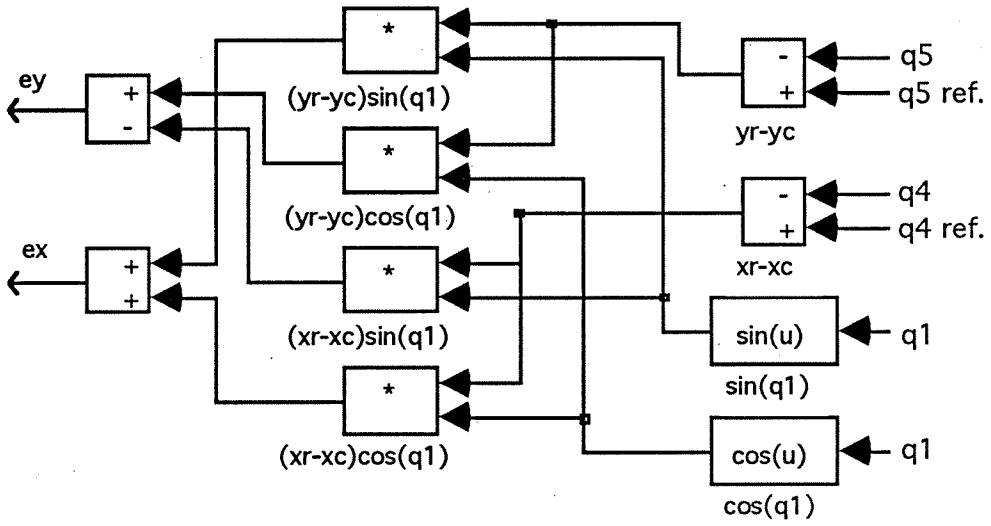


Figure 5.4 Error Block

5.4 Control Block Simulations

The purpose of this section is to verify the control algorithm chosen early, by simulating the TMR's path tracking abilities. The simulations will be completed using the block diagrams shown in Figs. (5.2), (5.3), and (5.4) and Simulink software. Three basic tests were completed, they are as follows; ramp, ramp with unit offset, and a unit step reference. In each test the TMR was started at the origin (0,0).

The first test, graph (5.5), uses a ramp as the reference path. This graph shows the TMR following the reference path very accurately, basically with no error. One can conclude that when the TMR has no initial error, the control algorithm keeps the error minimized. This initial test also shows that the TMR model is stable under these straight line trajectories. The second test has a ramp reference with a unit offset, which gives the TMR an initial error. The reference and actual path of the TMR can be seen in Figs. (5.6) and (5.7). Fig. (5.7) best shows the trajectory of the TMR as it approaches the reference, where initially the TMR has a large angular velocity which gradually decreases as it approaches the reference path. The third test used a unit step as the reference path. The trajectory shown in Fig. (5.8) is very similar to a second order system, yet shows the same characteristics as in the second test, where the angular velocity is largest at the beginning and then gradually decreases.

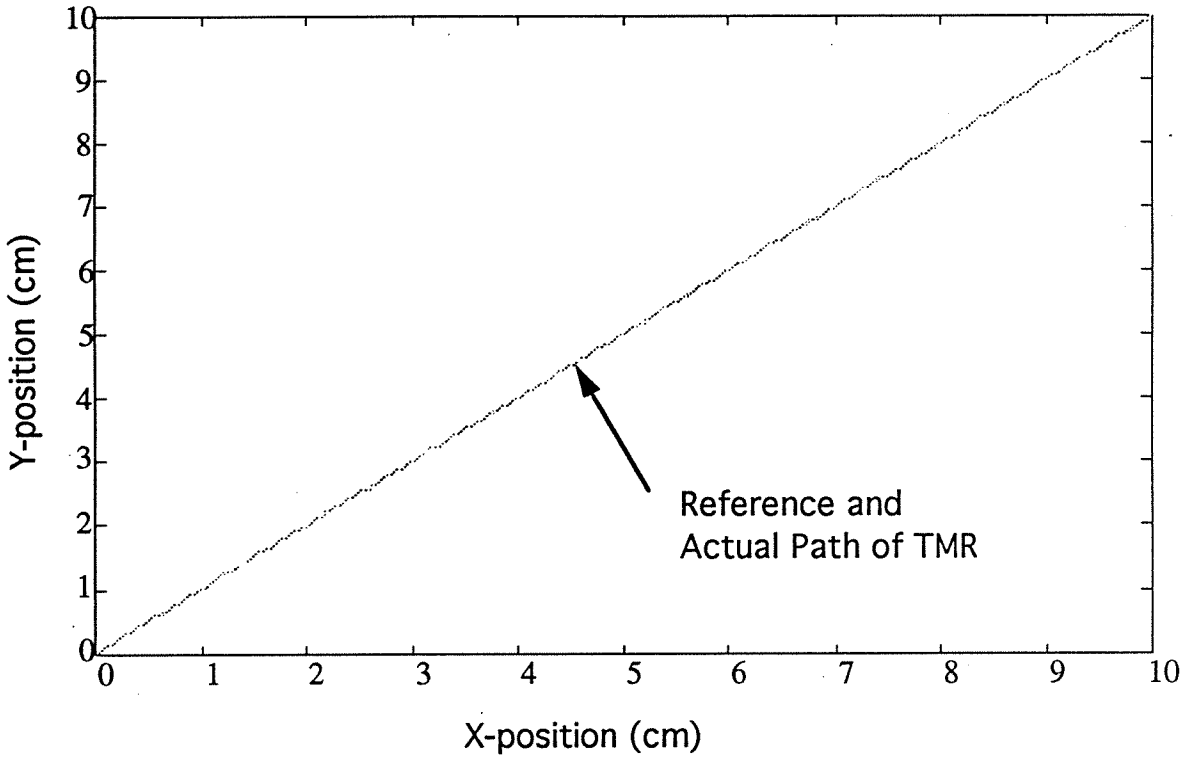


Figure 5.5 Graph of TMR trajectory with ramp reference, no initial error.

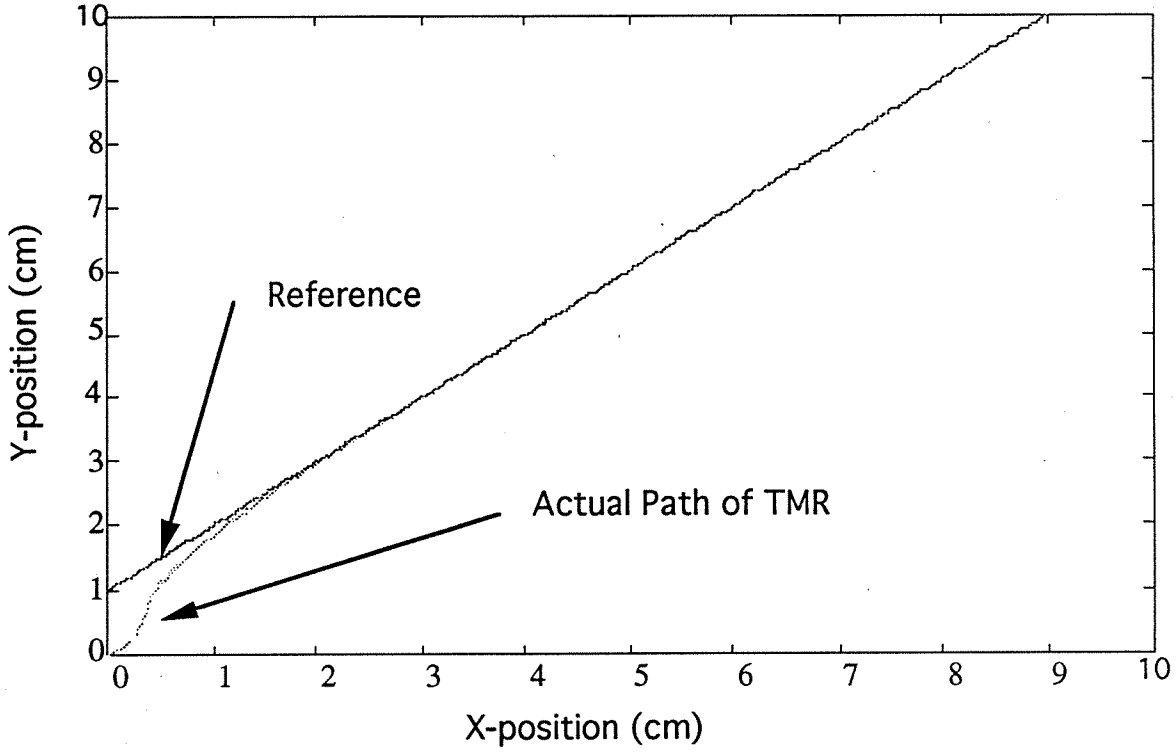


Figure 5.6 Graph of TMR trajectory with ramp reference and unit offset.

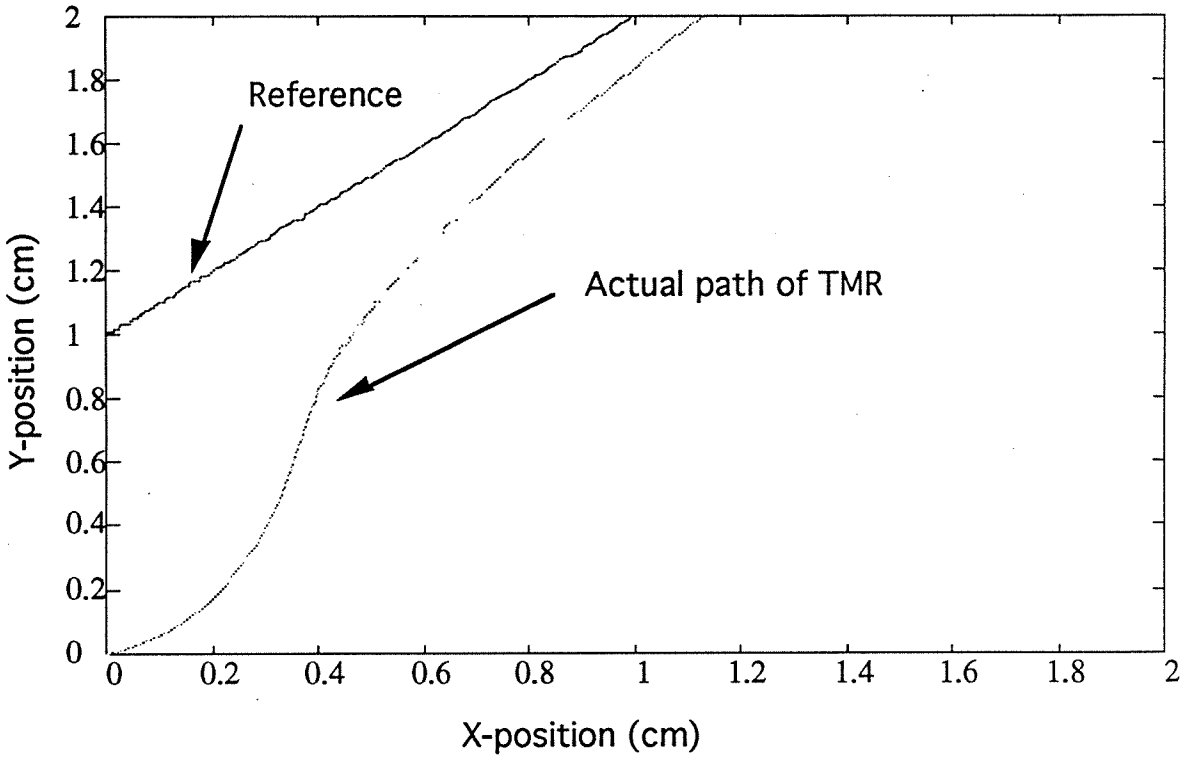


Figure 5.7 Enlarged view of Figure 5.6

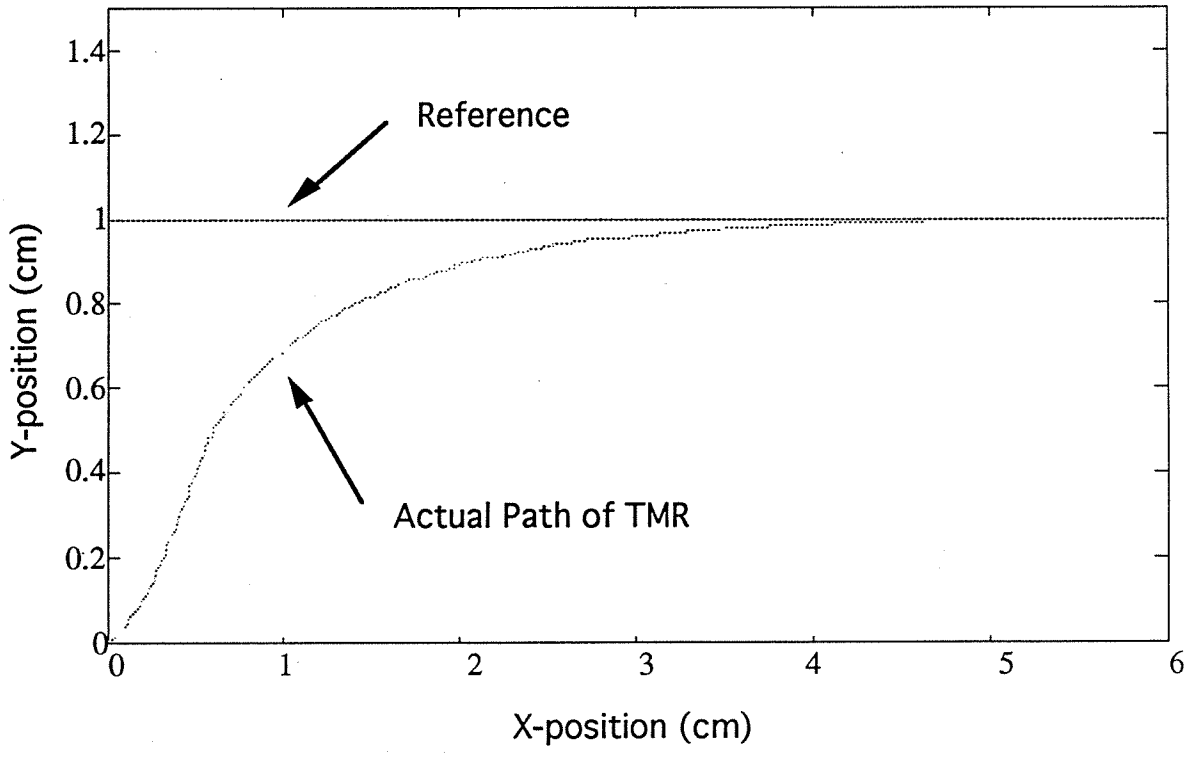


Figure 5.8 Graph of TMR trajectory with a step reference

Overall the simulations seem to be stable and prove the validity of the control algorithm. Each test showed that the TMR followed the reference path and minimized the positional error. The initial conditions are of prime importance when considering the tracking ability, due to possibilities in large orientation errors causing the control algorithm to be unstable. The control gains are also of prime importance when examining the tracking ability. The gains will determine the overall rise time and stability of the system.

5.5 Conclusion

In this chapter a control model was developed using Simulink software, and then simulated to prove validity of a control algorithm developed by Kanayama (1988). Initially a literature review of control algorithms was completed, which focused on algorithms for autonomous and semi-autonomous vehicles. Although there are mainly three different types of control algorithms, classical, fuzzy, and neural, the control algorithm based on Kanayama was chosen for its accuracy and simplicity. After choosing an algorithm, each control block was developed in detail. The general block diagram consisted of a position reference, PID error manipulator, Error Control block (see Fig. 5.3), Velocity Control block (see Fig. 5.4), and additional feedback lines. The control blocks entailed information on the algorithm chosen along with the equations of motion derived in chapter 4. The chapter then concluded with tests verifying the control algorithms' path tracking abilities. Three basic tests were completed for the simulations, they are as follows; ramp, ramp with unit offset, and a unit step reference. Overall the control algorithm seem to work well, however the tracking ability is very dependent on control gains.

CHAPTER 6 - PHYSICAL MODEL TESTING

6.1 Introduction

The purpose of this chapter is to develop a scaled down physical model of the Tethered Mobile Robot and then complete tests which will enable us to examine the controllers, kinematics, and path trajectories covered in the previous chapters. The testing will be separated into two sections, open loop and closed loop testing, respectively. The open loop test will compare the general trajectory for a given set of test conditions to the theoretical trajectory for those same conditions. The closed loop test will show test results of the TMR following a defined path given error data simulated by the local sensor. In the closed-loop test, the tracking system (linkage) will be ignored, and supplemented with dead reckoning as the means of positional information. This should not cause any significant errors, due to the short duration of each test.

6.2 Physical Model Development

The purpose of this section is to describe the general configuration of the physical TMR model. The TMR wheel configuration chosen is similar to a tricycle, yet reversed, having two driven wheels in the front and one passive castor in the rear (see Fig. 6.2). The main frame of the TMR prototype is made of aluminum and is approximately 7.1 inches (18 cm) wide by 9.8 inches (25 cm) long. Attached to the frame is two D.C. motors and a passive castor. Attached to the motors are two wheels, each wheel being aluminum with very tractive type tires with a radius of approximately 1.2 inches (2.95 cm). The track width of the front is approximately 8.3 inches (21 cm) and the castor is located 5.3 inches (13.5 cm) behind the front axle. Stated otherwise, this configuration has two diametrically opposed drive wheels and a single free rolling castor. This general configuration will allow two dimension motion, therefore any path in a plane may be traced. The general parameters describing the system can be found in table (6.1).

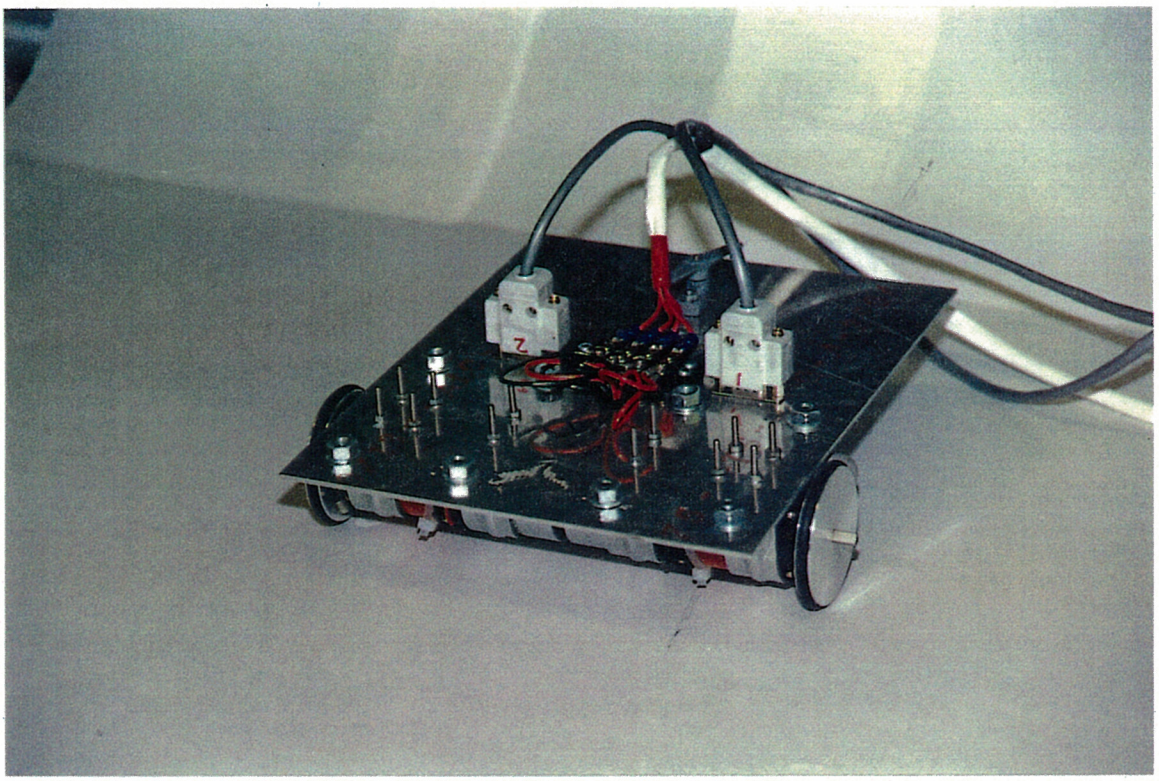


Figure 6.1 Front view of mobile platform

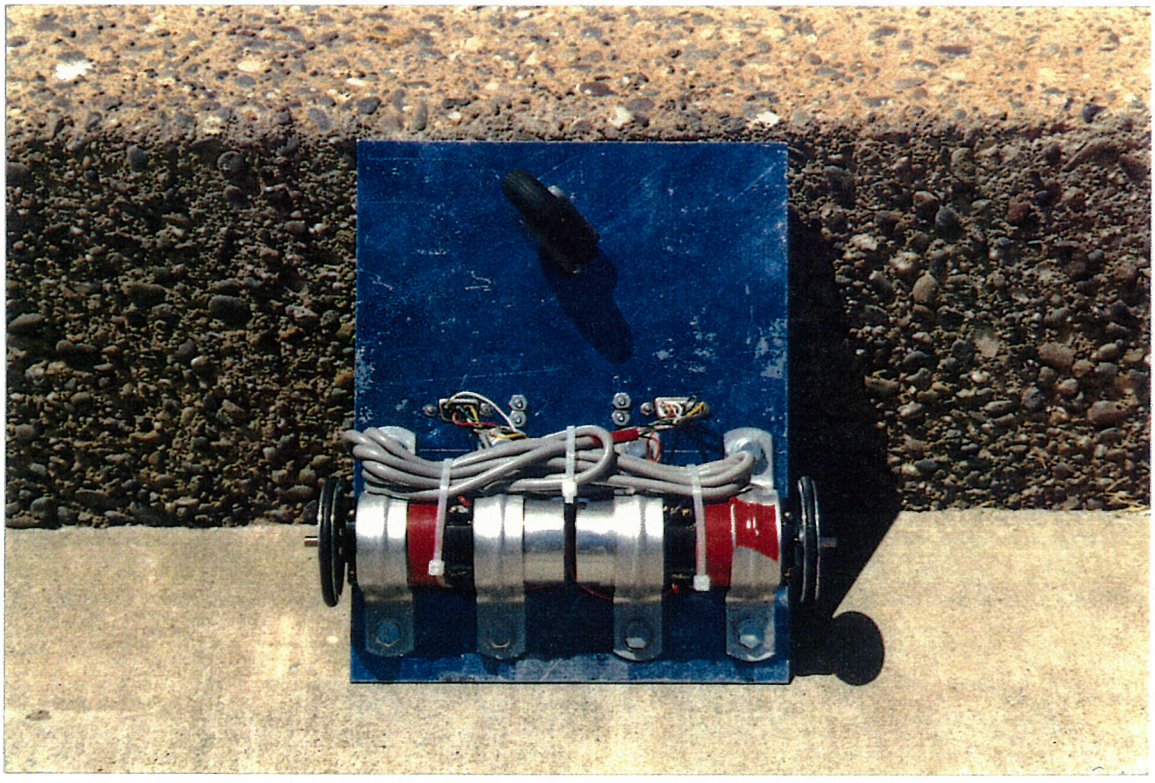


Figure 6.2 Underside of mobile platform

PLATFORM PARAMETERS	VALUE
Total platform weight:	3.85 lbs. (1.75 Kg)
Front weight:	3.08 lbs. (1.4 Kg)
Rear weight:	.77 lbs. (.35 Kg)
Track width:	8.34 in. (21.2 cm)
Wheel radius:	1.16 in. (2.95 cm)
Wheel base:	5.31 in. (13.5 cm)
Platform inertia:	.47 lbs.*ft.^{2} (.02 Kg*m^2)
MOTOR PARAMETERS	VALUE
Rated power:	.04 hp. (30 W)
Rated torque:	1.06 lbs.*in. (1.23 Kg*cm)
Rated speed:	2400 rpm
Rotor inertia:	.0015 oz.*cm*sec^2 (.12 g*cm*sec^2)
Weight:	12.5 oz. (390 g)
ENCODER PARAMETERS	VALUES
Resolution: A,B	500 pulses/rev
Z	1 pulse/rev

Table 6.1 Parameters for TMR prototype

6.3 Open Loop Test

The main purpose of the open-loop experiment is to determine how well the TMRs' trajectory, under constant velocity, follows the theoretical path derived using the same velocities. In order to quantify this test, the experiments will be graphed against the

theoretical results. First, a description of the general configuration used in testing the open-loop configuration for the TMR will be given. In order for the TMR to be controlled, a flexible control system discussed earlier will be used. The control system hardware consists of the following: a "IBM" based PC; two Flexible Servo Controllers; power transistor circuit; two D.C. motors; two optical encoders; and several connection cables (e.g. RS-232C).

The basic operation of the system is as follows (see Fig. 6.1). The basic kinematics of the mobile platform were derived and written in a Quick-Basic program code. Also written in Quick-basic is a communication program which allows the "IBM" PC to talk with the Flexible Servo Controllers. Computer code is included in Appendix A and a flow chart can be seen in Fig. (6.3). Given the mobile platforms center of mass velocity and angular velocity, the kinematic program calculates the corresponding wheel angular velocities and has those values sent by the communication program to the first Flexible Servo Controller. The first Flexible Servo Controller then sends the proper information through optical fiber cables to the second Flexible Servo Controller with only a slight delay. With both Flexible Servo Controllers initialized, the controllers send signals with the use of power transistors in the form of pulse width modulation (PWM) to control the motor angular velocities. The Flexible Servo Controllers then control and monitor very accurately the angular velocity of the motors with the use of optical encoders. The motors mechanical time constant, which is the most dominant, is approximately 9.2 milliseconds.

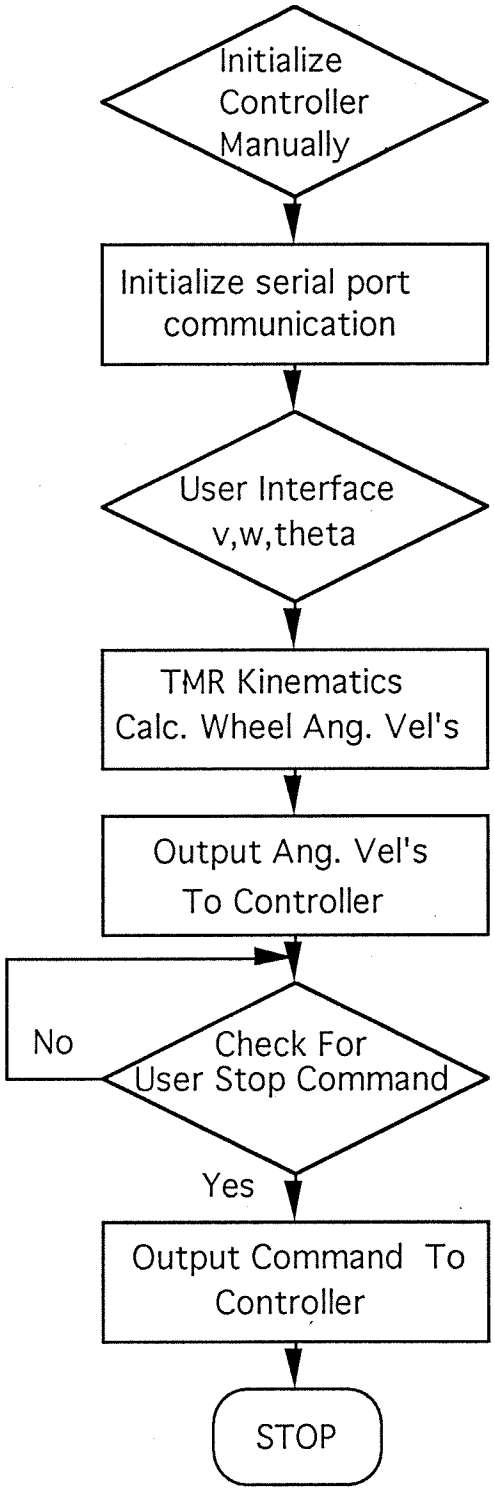


Figure 6.3 Flow Chart for Open-Loop Source Code

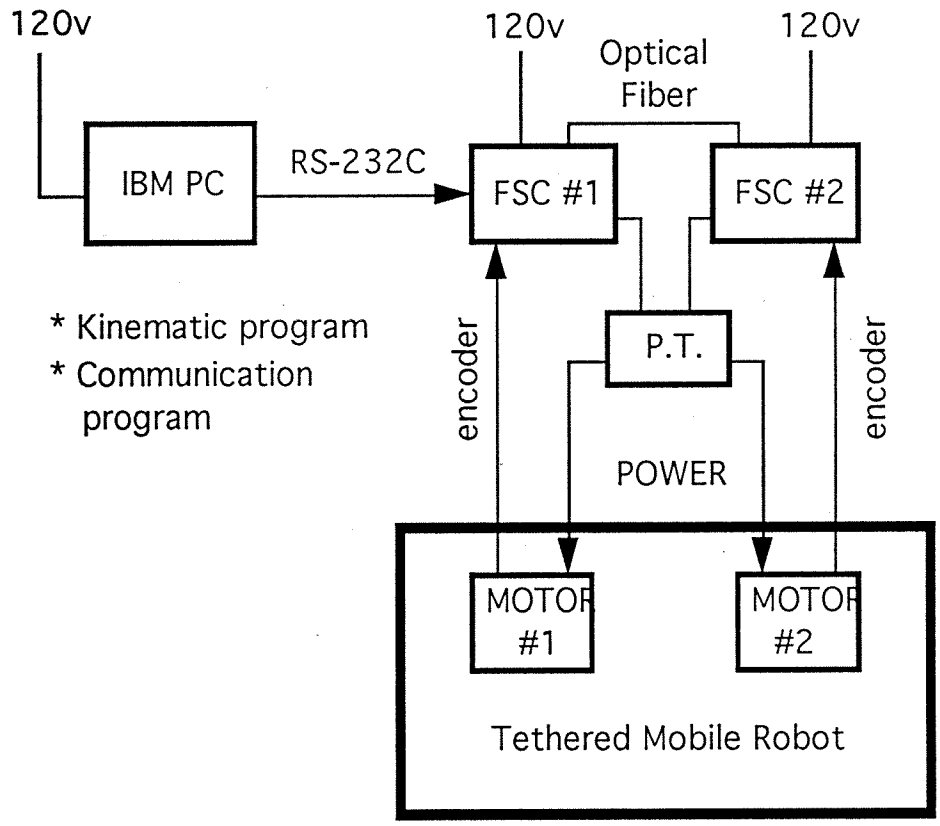


Figure 6.4 Block Diagram of Open-Loop Experimental Set-Up

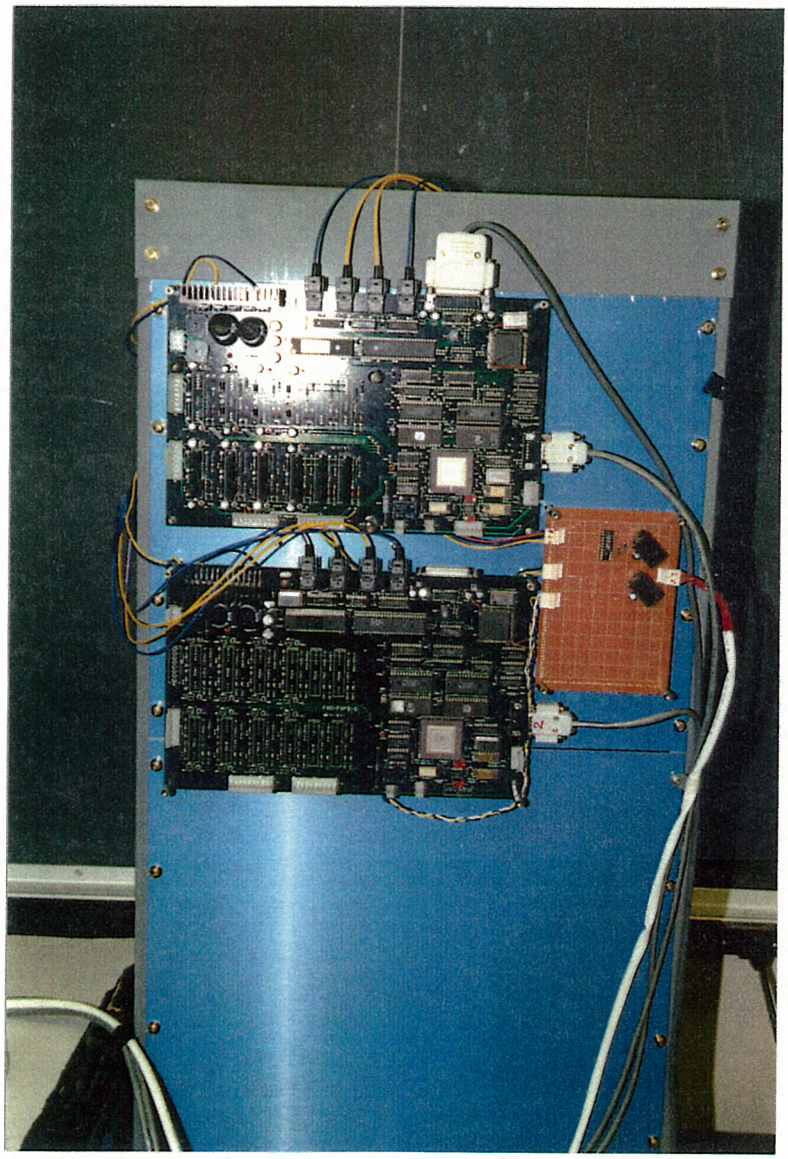


Figure 6.5 Flexible Servo Controllers

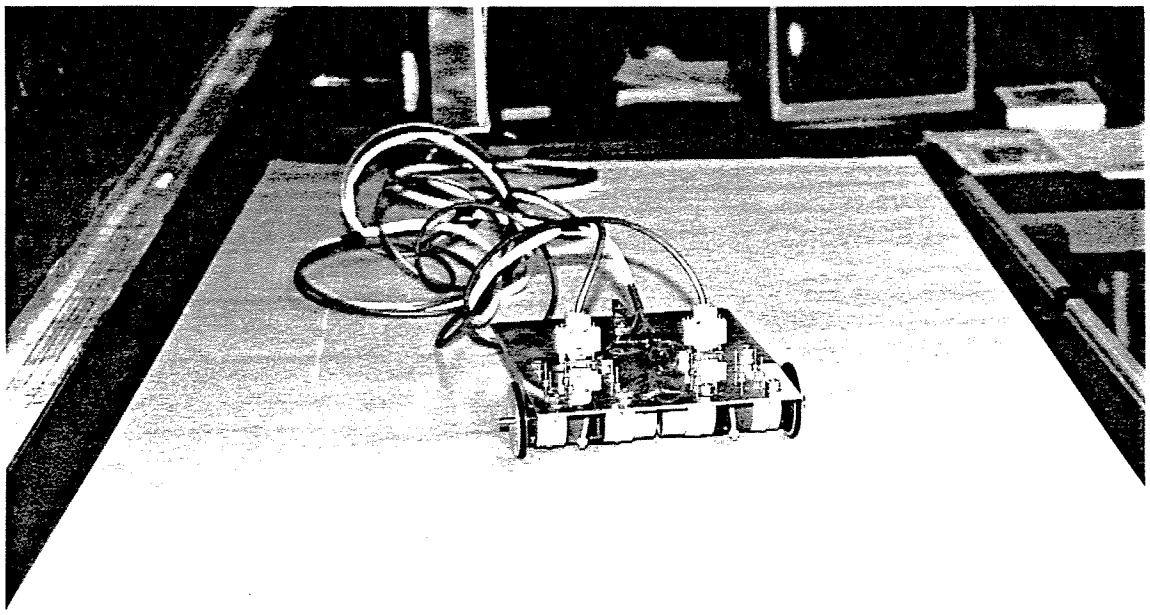


Figure 6.6 Mobile platform on test table

Test Procedure

With the physical model completed and operating properly, three controlled open loop tests were completed. As can be seen in Fig. (6.6), the tests were run on a table approximately 3 ft. (.91 m) wide by 9 ft. (2.74 m) long. A large sheet of paper was fixed to the table and marked with a X and Y axis so that the trajectory could be followed accurately. Inserting a pen centered between the front driven wheels allowed the path of each test to be measured fairly accurately. The TMR prototype was aligned as accurately as possible using a T-square to set the initial conditions. Fig. (6.7) is a photo of the experimental set-up, which included the Flexible Servo Controllers, mobile platform, and other accessories. With the tests completed, the data was then compared with theoretical data to determine the general accuracy to the TMR prototype under open-loop conditions.

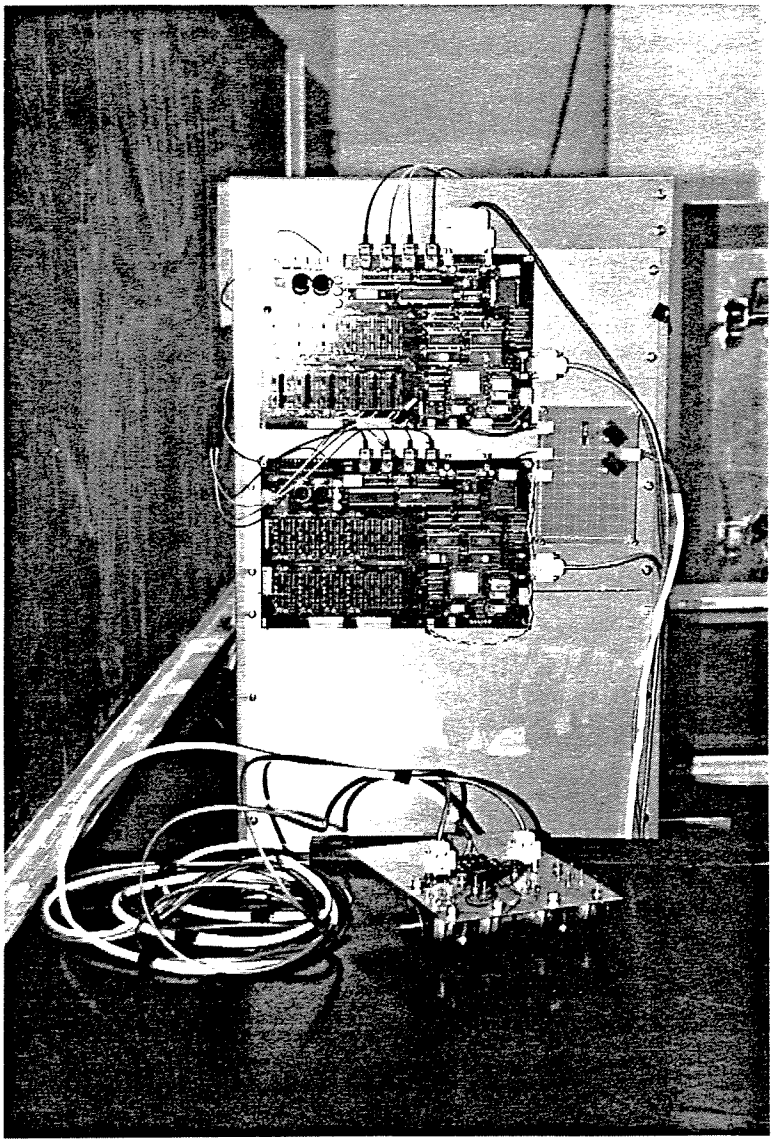


Figure 6.7 Photo of experimental set-up

Next, the three test results, with their corresponding test conditions, are given in numerical and graphical form:

x(cm)	y(cm)
0	21.8
2	21.8
4	21.7
6	21.6
8	21.4
10	21.3
12	20.9
14	20.4
16	20.2
18	19.6
20	19.0
22	18.4
24	17.7
26	16.8
28	15.9
30	15.0
32	13.9
34	12.7
36	11.4
38	10.1
40	8.7
42	6.8
44	4.9
46	3.2
48	1.0
50	-1.0

Table 6.2 Test data for low velocity test. The following test conditions were used: u_2 =34rpm, u_3 =48rpm, w =.2rad/s, and v =.02m/s.

x(cm)	y(cm)
0	14.9
2	14.8
4	14.8
6	14.6
8	14.3
10	14.0
12	13.3
14	12.7
16	12.0
18	11.3
20	10.3
22	9.3
24	7.9
26	6.6
28	5.0
30	3.2
32	1.3
34	-1.3
36	-3.8
38	-6.9

Table 6.3 Test data for large angular velocity test. The following test conditions were used: u_2 =30rpm, u_3 =51rpm, w =.3rad/s, and v =.02m/s.

x(cm)	y(cm)
0	18.8
4	18.8
8	18.8
12	18.8
16	18.7
20	18.5
24	18.2
28	18.0
32	17.6
36	17.1
40	16.3
44	16.0
48	15.5
52	14.7
56	14.0
60	13.2
64	12.3
68	11.4

Table 6.4 Test data for large velocity test. The following test conditions were used: u_2 =30rpm, u_3 =51rpm, w =.3rad/s, and v =.02m/s.

where,

u_2 = angular velocity of the right wheel,

u_3 = angular velocity of the left wheel,

w = angular velocity of TMR, and

v = linear velocity of TMR.

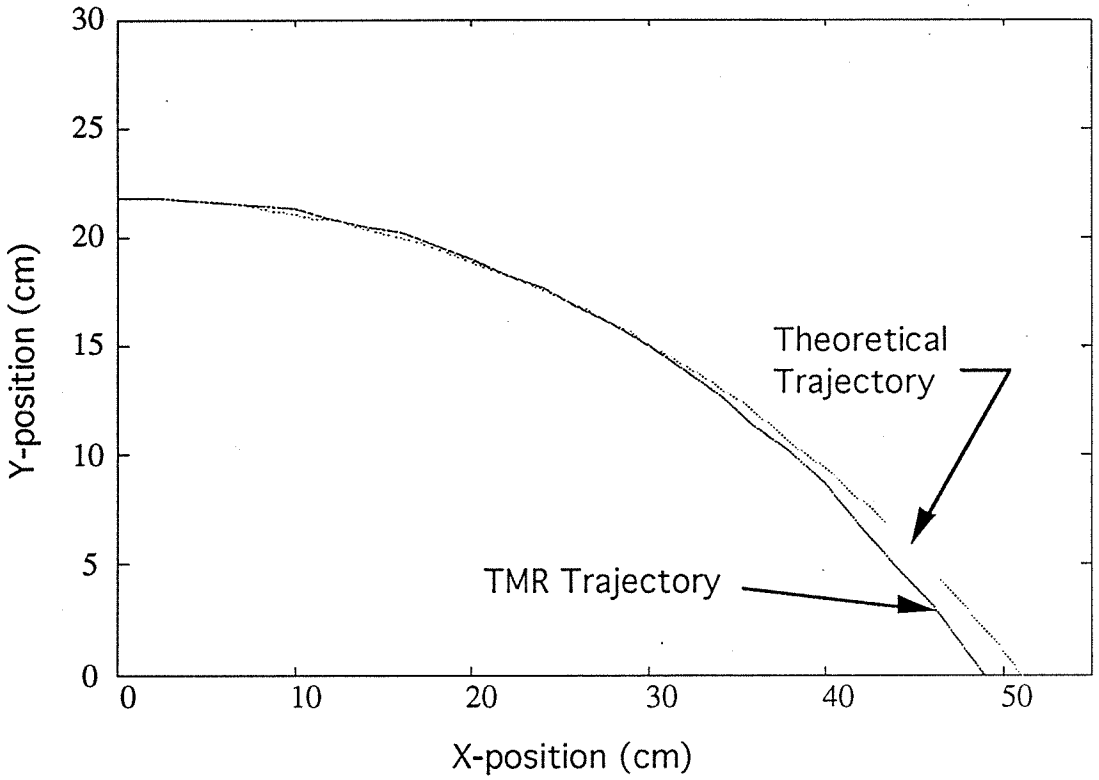


Figure 6.8 Low Velocity Test . Test conditions are u_2 =34rpm, u_3 =48rpm, w =.2rad/s, and v =.02m/s.

Summary of Results

The main purpose of the open-loop experiment was to determine how well the TMRs' trajectory, under constant velocity, followed the theoretical path derived using the same velocities. The theoretical path was determined by the corresponding wheel velocities, and resulted in a circular arc with the radius related to the wheel velocities.

With the completion of the three tests, the complied data was put in graphical form and plotted with the corresponding theoretical data. In graphical form, a better understanding

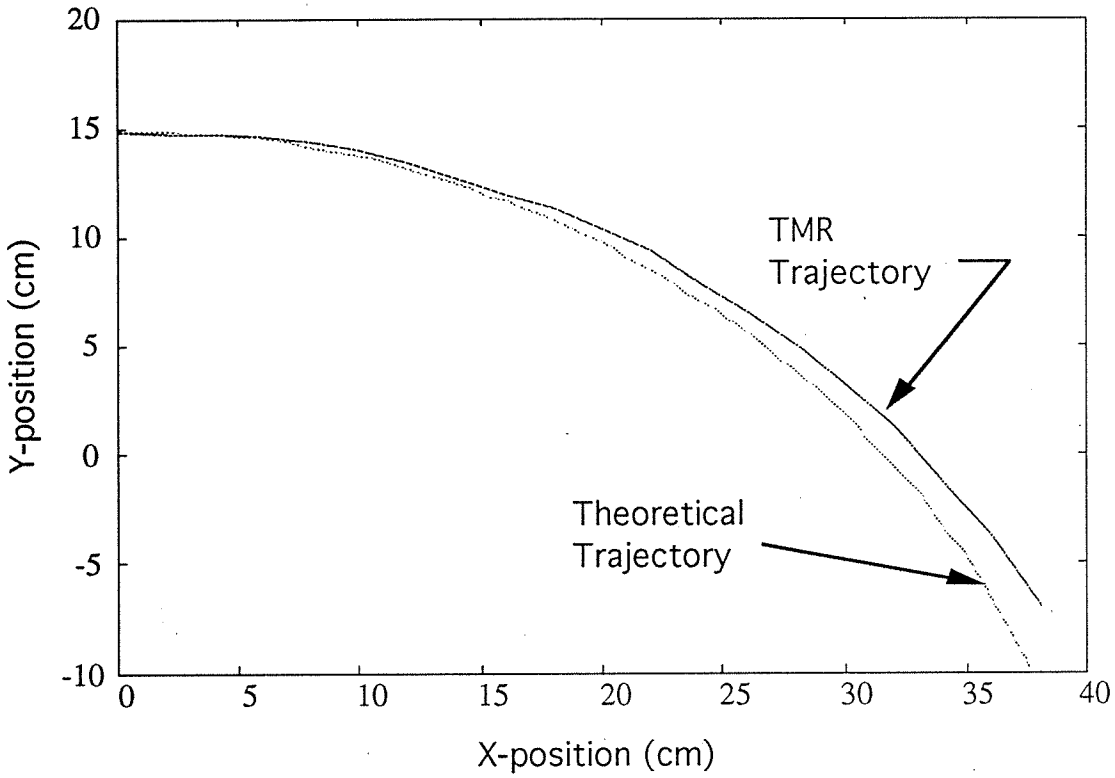


Figure 6.9 High Angular Velocity Test. Test conditions are u_2 =30rpm, u_3 =51rpm, w =.3rad/s, and v =.02m/s.

of the data can be gained while comparing tests. During the data collection stage, it was sometimes difficult to see the exact trajectory, and therefore small errors were possibly introduced. However, looking at the graphs, these errors seem to be small due to the continuity of the trajectories. Possible errors in the tests could be accounted for as follows: initial angular position not exactly zero; encoder and power cables caused torque on platform; wheel slip; measurement errors. These errors may be probable due to the fact that each test had a different trajectory relative to the theoretical. Notice that in Fig. (6.8), the TMR trajectory followed to the inside of the theoretical trajectory, yet the

opposite occurred in Fig. (6.9). Overall the tests seem to follow the theoretical solutions closely for the first two tests Figs. (6.8) and (6.9), but the third test, Fig. (6.10) was not as accurate.

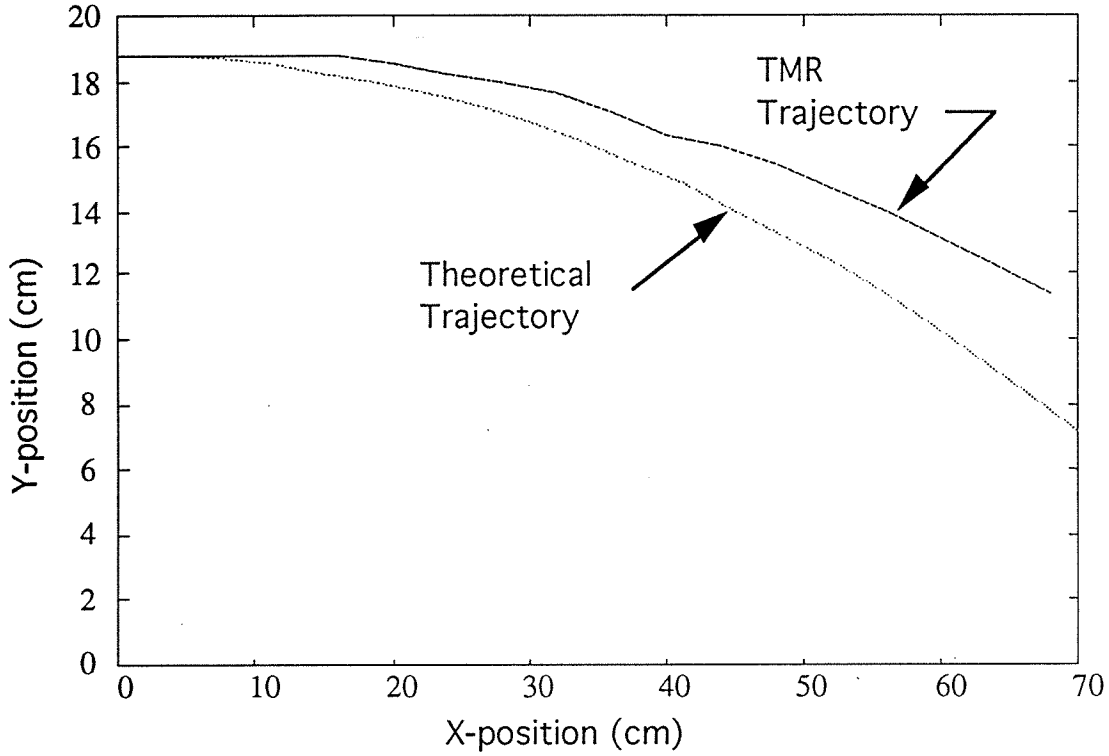


Figure 6.10 High Velocity Test. Test conditions are u_2 =30rpm, u_3 =51rpm, w =.3rad/s, and v =.02m/s.

6.4 Closed Loop Test

The main purpose of the closed-loop experiment is to validate that the TMR can be controlled to follow a defined path. Optimization of the path following will not be addressed in this thesis. In order for the TMR to be controlled, the flexible control system described earlier is used. The control system hardware consists of the following: a "IBM" based PC; two Flexible Servo Controllers; power transistor circuit; two D.C. motors; two optical encoders; and many connection cables (e.g. RS-232C).

The basic operation of the system is as follows (see Fig. 6.12). The basic kinematics and control algorithm for the TMR were derived and written in a Microsoft C program code, also written in Microsoft C is a communication program which allows the "IBM" PC to talk with the Flexible Servo Controllers and vice-versa. Appendix B includes the computer code and a flow chart is shown in Fig. (6.11). Given the proper initialization for the computer code, the TMR is placed on the table with the proper initial conditions. To simulate the local sensor generating an offset error, an error table was produced and stored in the "IBM" PC. Once initialized and started, the Flexible Servo Controllers will read the encoders and send this information to the "IBM" PC, which will calculate positional information. Therefore, with the simulated local sensor information and encoder information being read periodically by the "IBM" PC, the control algorithm can calculate a velocity and angular velocity, relative to the error, for the TMR. These velocities are then converted to wheel velocities and sent to the Flexible Servo Controllers. The Flexible Servo Controllers control the wheel velocities very accurately, having a sampling time approximately ten times that of the positional control loop.

Test Procedure

The simulated test will use a table of data points describing the path, and these data points will be sampled in order to generate the reference. In order to compare the actual path of the TMR to the reference path, the encoder values of each wheel will be converted into a linear position. The TMR prototype will be aligned as accurately as possible using a T-square to set the initial conditions. The simulated test will help in debugging the system, before using the local sensor to close the control loop. Besides these minor changes, the closed-loop test procedure will follow the format of the open-loop test.

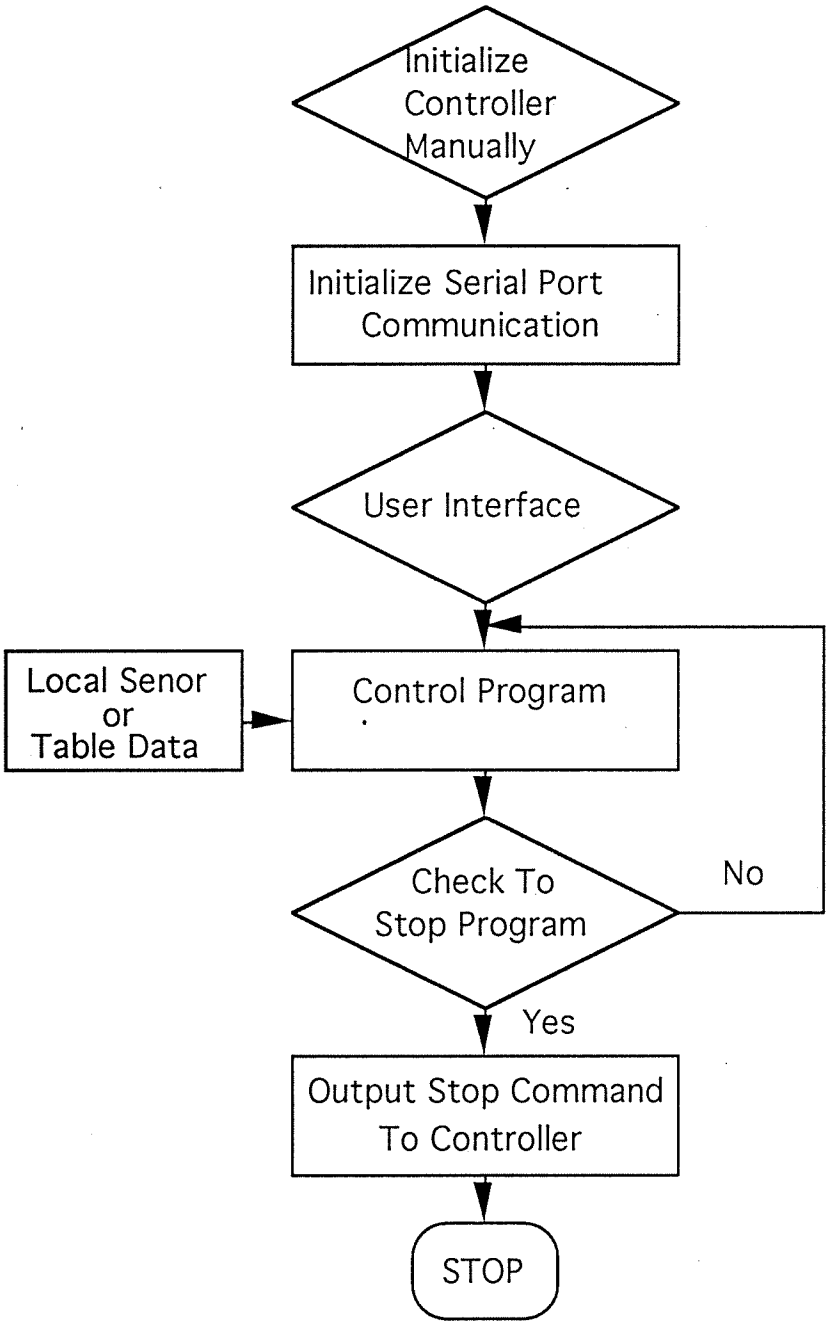


Figure 6.11 Flow Chart for Closed-Loop Source Code

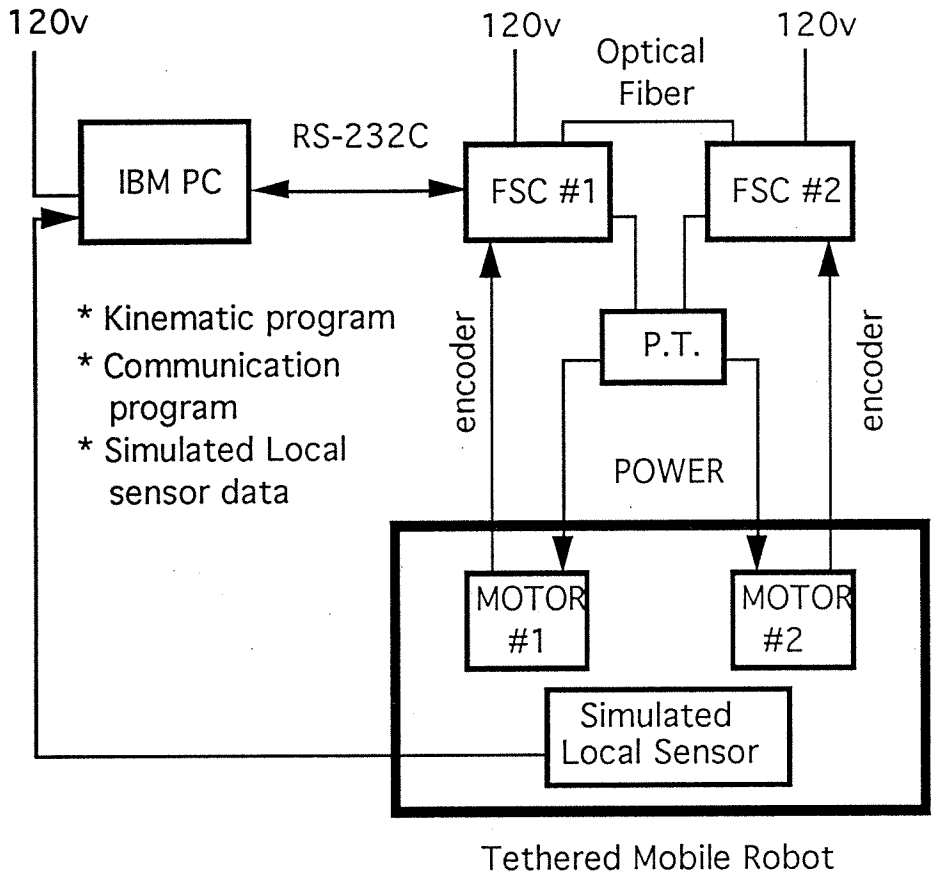


Figure 6.12 Block Diagram of Closed-Loop Experimental Set-Up

Summary of Results

The main purpose of the closed-loop experiment is to validate that the TMR can be controlled to follow a defined path. The experiment used a table of data points representing an error to simulate the use of a local sensor. The simulated tests were also used in debugging the system, therefore reducing problems when using the local sensor to close the control loop.

With the completion of the simulated tests, the complied data was put in graphical form and plotted with the corresponding reference data. Shown in Fig. (6.13) is one of

the simulated tests, which used a sinusoidal reference. Overall the TMR tracked the reference very well, showing only minor errors while turning sharp corners. These errors could possibly be reduced by adjusting the control gains or using a better algorithm.

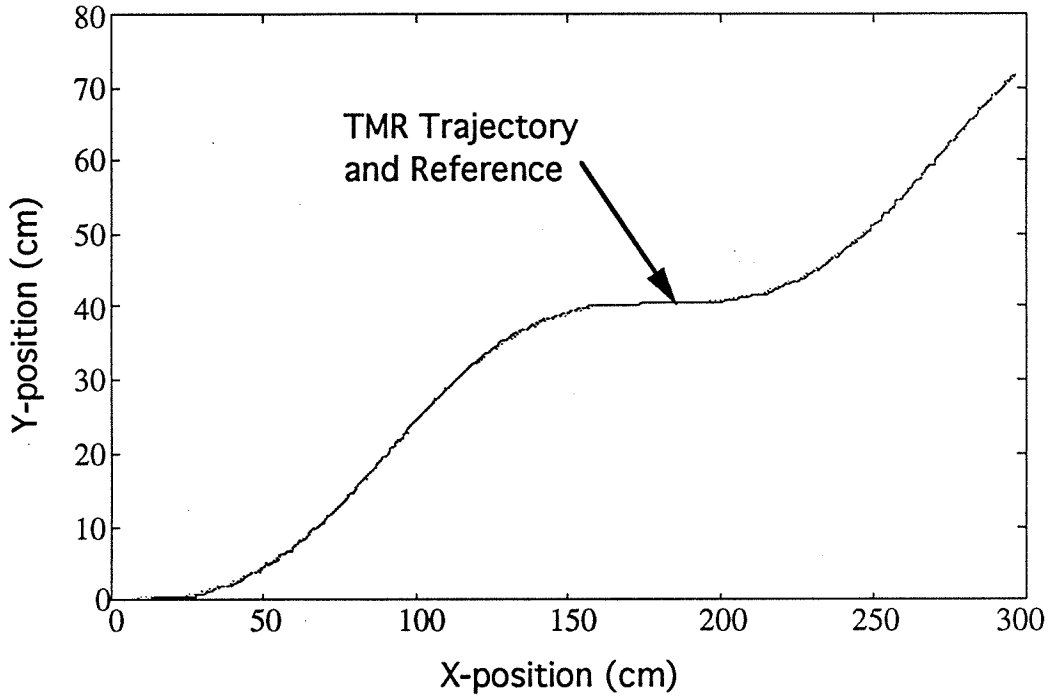


Figure 6.13 Closed-Loop Test Using Table Data as the Reference. Reference Function equals Offset=sin(t*PI/400).

6.5 Conclusion

In this chapter a scaled down physical model of the Tethered Mobile Robot was developed and tests were completed which enable the controllers, kinematics, and path trajectories covered in the previous chapters to be examined. The testing was separated into two sections, open loop and closed loop testing, respectively. The open loop test compared the general trajectory for a given set of test conditions to the theoretical trajectory for those same conditions. These tests showed that the TMR followed the theoretical trajectory fairly accurately, however possible errors were suggested. The closed loop test showed results of the TMR following a defined path given simulated

error data from the local sensor. As mentioned earlier, the closed-loop test was performed to prove that the TMR could be controlled to follow a defined path, yet optimization was not performed. In the closed-loop test, the tracking system (linkage) was ignored, and supplemented with dead reckoning as the means of positional information. Overall the tests proved that the TMR can be controlled using closed-loop control.

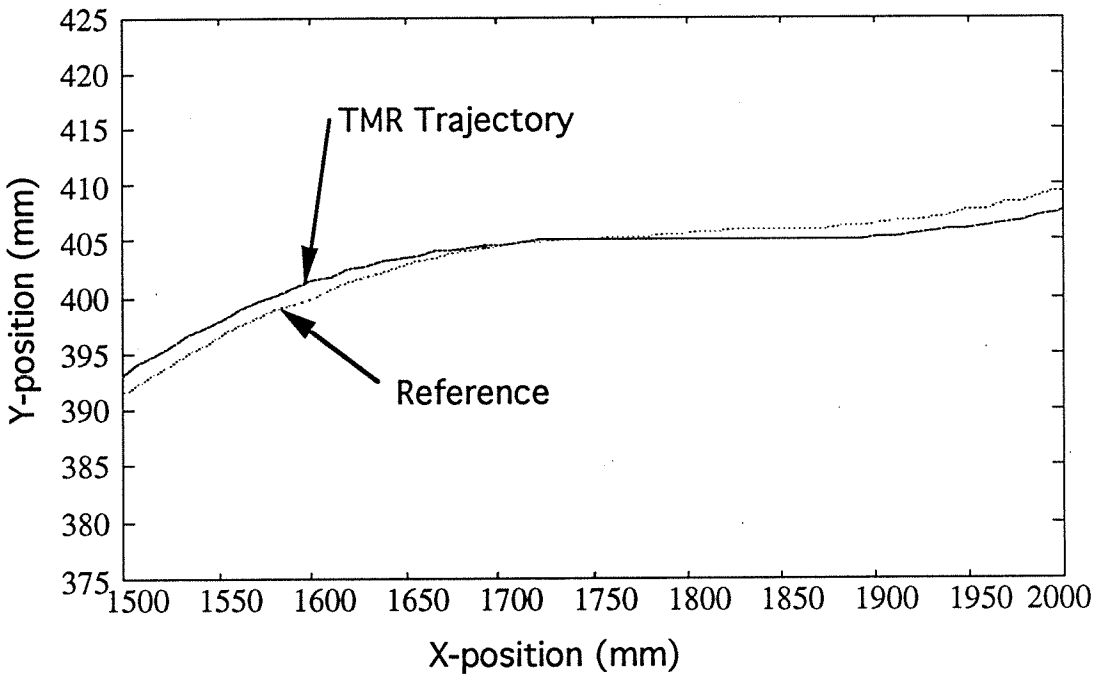


Figure 6.14 Enlarged view of Figure 6.13.

CHAPTER 7 - CONCLUSIONS AND RECOMMENDATIONS

7.1 Conclusions

The purpose of this thesis was the conceptual development of a self-propelled robot which works in close proximity to a support vehicle for purposes of power, etc., and allowing for the measurement of these robot's position relative to the support vehicle with high accuracy. Such an approach has the potential for use in a wide variety of applications. The development involved a detailed literature search of each area, development of the general TMR configuration, development of the workspace, development of a dynamic model, and lastly the development of a small scale prototype which was used for testing/verification of the control equations.

To begin the development, in chapter 1, a general description of the problem was presented, outlining the need for a new method to overcome the inherent disadvantages of the use of conventional robots for highway maintenance operations. Additionally, a literature search was performed to investigate existing tethered mobile robot technology and other related topics. Based on an intensive literature review, it is the author's opinion that this concept has not been presented previously. The literature most closely related concerns autonomous mobile robots, semi-autonomous mobile robots and other mobile robot technologies. Lastly, the objective and problem statement was presented which detailed the development procedure.

Next, in chapter 2, a general description of the TMR's conceptual configuration was presented. This included areas such as the wheel configuration, tracking configuration, control configuration, and local sensor configuration. In each section a literature review preceded the components discussion. After presenting the conceptual configuration of these specific areas, the TMR's integrated conceptual layout was given. Many different wheeled configurations were reviewed and investigated for use as the

propelling method for the TMR, ranging from conventional to omnidirectional to ball wheels. The tracking configuration was needed in order to determine the TMR's spatial position, which included possible methods such as dead reckoning, vision systems, infrared systems, etc. The control system configuration included various components, that is, the robot controller, the actuators which drive the TMR wheels, the actuator controllers, and the corresponding control algorithms and software. The local sensor was investigated as a method to detect cracks in pavement surfaces, so that a reference error can be produced which is used to help control the TMR. Lastly, the TMR's integrated conceptual layout presented the chosen configurations, which included a wheel configuration using two conventional wheels, a tracking configuration using a mechanical linkage, a control configuration based on microelectronics and used BLDC motors as wheel actuators, and the local sensor chosen was a laser range finding sensor based on the principle of triangulation.

In chapter 3, the general linkage and workspace analysis used in designing the TMR system was presented. Additionally, the general workspace configuration was addressed which detailed the actual layout. Two workspace areas were decided upon in order to have a versatile and usable system, one which allowed the system to address a full lane width and the other which is located within the width confines of the support vehicle. A numerical optimization method was implemented to size the linkage. Lastly, an error analysis was used on the linkage configuration, generating an error equation which can be used for a given set of parameters. This error can then be used in determining the proper encoder resolution and link length accuracy's needed.

With the completion of conceptual aspects of this thesis, in chapter 4, the TMR was modeled in order to facilitate in the design of control algorithms. With the modelling completed, the dynamical equations of motion were developed using an analytical method based on Kane's method. The last part of this chapter was used to convert the

equations of motion into state-space form. In state-space form, the equations can be used with computer software for simulation purposes.

In chapter 5 a control model was developed using Simulink software. Before the model was developed however, a literature review focusing on control algorithms such as classical, fuzzy, and neural was completed. A classical control algorithm based on the method of Kanayama was chosen due to its simplicity and accuracy. The control model developed was based on a multi-input multi-output (MIMO) scheme, and included many different control blocks, such as PID error manipulator block, error control block, and velocity control block. Lastly, there were three basic simulations completed which verified the chosen algorithm; ramp, ramp with unit offset, and a unit step reference. Overall the simulations seem to be stable and proved the validity of the control algorithm.

Chapter 6 focused on the development and testing of a scaled down prototype TMR. This scaled down prototype was used in testing of controllers, kinematics, and path trajectories covered in the previous chapters, but was not meant to be an exact TMR replica. The testing was separated into two sections, open loop and closed loop testing, respectively. The open loop test compared the general trajectory for a given set of test conditions to the theoretical trajectory for those same conditions. The closed loop test showed the TMR prototype following a defined path given simulated error data. For each test, block diagrams showing experimental set-ups and flow charts describing the computer source code was presented.

7.2 Recommendations

As a final section of this thesis, recommendations concerning the TMR's development and operation will be made. Although the initial portion of this thesis was concerned with the conceptual development , many practical questions arose.

The mechanical linkage chosen for the tracking method must be designed in a fashion which will not bind or lock when the TMR is traversing across the workspace.

These problems could arise when routing material and power cables along the linkage. Possible solutions would be to use swivel connection joints for the cables at the joints of the linkage or use some type of boom to support the cables overhead. A second consideration must be made concerning vibration. This will most likely be a problem during crack routing, and could be solved by inserting damping devices at each connection joint of the linkage.

The conventional wheels and configuration chosen may need addition modifications after initial testing. The castor arrangement may need to be modified due to sealant problems or road surface irregularity. Moving the castors to the front of the cart could eliminate possible sealant problems in addition to making a more stable configuration. Also if road surface irregularities are a problem, a single castor in the front can replace the two current rear castors. Sealant affixation problems on the drive wheels may cause possible vibration and other unwanted disturbances. This may be eliminated by using some type of scraping system which rubs on the wheels continuously.

The control aspect of the TMR must be examined much further. In this thesis, general control algorithms were investigated and a control algorithm by Kanayama was selected for initial testing. The testing proved that the TMR could be controlled to follow a path, however possible stability problems and disturbances should be investigated further. In addition, parameters must be decided upon which governs the accuracy at which the control algorithm must control the TMR.

Lastly, additional areas which must be considered for system development, which were not addressed in this thesis, are areas of initial condition path planning, stowage, path selection (for multi-crack workareas), etc. These areas in addition to the conceptual configuration presented will complete the TMR system conceptual development.

REFERENCES

Alexander, J.C. and Maddocks, J.H. (1989) "On the Kinematics of Wheeled Mobile Robots," *IJORR*, Vol. 8, No. 5.

Banta, L.E. (1988) "A Self Tuning Navigation Algorithm," IEEE International Conference On Robotics and Automation, pp. 1313-1314.

Bennett, D. (1992) Information Concerning Routing Process, *personal communication*, University of California, Davis.

Blais, F., Rioux, M., Domey, J. (1991) "Optical Range Image Acquisition for the Navigation of a Mobile Robot," IEEE International Conference On Robotics and Automation, Vol. 3, pp. 2574-2580.

Carlisle, B. (1983) "An Omni-directional Mobile Robot," *Developments in Robotics*, pp. 79-87.

Cox, I.J. and Wilfong, G.T. (1990) "Autonomous Robot Vehicles," AT&T.

Dainis, A., Juberts, M. (1985) "Accurate Remote Measurement of Robot Trajectory Motion," IEEE International Conference On Robotics and Automation, pp. 92-99.

De Vegte, J.V. (1990) Feedback Control Systems, Prentice-Hall, Inc.

Dorf, R.C. (1988) International Encyclopedia of Robotics, John Wiley & Sons, Inc., Vol. 1.

Dorf, R.C. (1990) Concise International Encyclopedia of Robotics, John Wiley & Sons, Inc.

Feng, D. and Krogh, B.H. (1991) "Dynamic Steering Control of Conventionally Steered Mobile Robots," *Journal of Robotics Systems*, Vol. 8(5), pp. 699-721.

Gosselin, C.M. and Guillot, M. (1991) "The Synthesis of Manipulators with Prescribed Workspace", Journal of Mechanical Design, Vol. 113, pp. 451-455.

Gupta, K.C. and Roth, B. (1982) "Design Considerations for Manipulator Workspace", Transactions of the ASME, Vol. 104, pp. 704-711.

Gupta, K.C. (1986) "On the Nature of Robot Workspace", The International Journal of Robotic Research, Vol. 5, No. 2, pp. 112-121.

Helmers, C. (1983) "Ein Heldenleben, (Or, A Hero's Life, With Apologies to R. Strauss)," *Robotics Age*, Vol. 5, No.2, pp. 7-16.

Hollis, R. (1977) "Newt: A Mobile, Cognitive Robot," *Byte*, Vol. 2, No. 6, pp. 30-45.

Hongo, T., Arakawa, H., Sugimto, G., Tange, K., and Yamamoto, Y. (1985) "An Automatic Guidance System Of A Self-Controlled Vehicle", IEEE International Conf. on Industrial Electronics, Control & Instrumentation, pp. 535-540.

Jing, L., Kirschke, K., Quinlan, T., Schultheis, E., Smith, M., Velinsky, S. (1990) "On the Automatic Sealing of Cracks in Pavement", Manufacturing Automation and Productivity Program Technical Report #90-03-01 University of California, Davis.

Jo, D. and Haug, E.J. (1989) " Workspace Analysis of Multibody Mechanical Systems Using Continuation Methods", Journal of Mechanisms, Transmissions, and Automation in Design, Vol. 111, pp. 581-589.

Kanayama, Y., Yuta, S., Yajima, T., and Shimmura, S. (1985) "An Implementation of Mitchi -- A Locomotion Command For Intelligent Mobile Robot", International Conference on Advanced Robotics, pp. 127-134.

Kanayama, Y., Nilipour, A., and Lelm, C. (1988) "A Locomation Control Method For Autonomous Vehicles", IEEE , pp. 1315-1317.

Kane, K.R. and Levinson, L.A. (1985) "Dynamics: Theory and Applications", McGraw-Hill, Inc.

Killoughi, S.M. and Pin, F.G. (1991) "Design and Control of a Fully Omnidirectional and Holonomic Wheeled Platform for Robotic Vehicles", *IEEE ICRA*.

Kompass, E.J. and Williams, T.J. (1989) " Total Control Systems Availability, "Purdue Research Foundation, West Lafayette, Indiana.

Krulewich, D.A. and Velinsky, S.A. (1992) "Development of a High Resolution Sensing System for Automated Crack Sealing Machinery", Interim Report of SHRP H-107A, Strategic Highway Research Program, National Research Council, Washington, D.C.

McGillem, C.D. and Rappaport, T.S. (1988) "Infra-red Location System for Navigation of Autonomous Vehicles," Proceedings IEEE Conference on Robotics and Automation, New York, NY, pp. 1236-1238.

Moravec, H.P. (1983) "The Stanford Cart and The CMU Rover," The Robotics Institute, Carnegie-Mellon University, Pittsburgh, PA 15213.

Muir, P.F. and Neuman, C.P. (1986) "Kinematic Modeling of Wheeled Mobile Robots", *Technical Report No. CMU-RI-TR-86-12*, The Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, 15213.

Muir, P.F. and Neuman, C.P. (1987) "Kinematic Modeling of Wheeled Mobile Robots", *Journal of Robotic Systems*, Vol. 4(2), pp. 281-340.

Muir, P.F. and Neuman, C.P. (1990) "Resolved Motion Rate and Resolved Acceleration Servo-Control of Wheeled Mobile Robots," *IEEE Conf. Robotics and Auto.*

Petrov, P.P. (1991) "Robust Trajectory Tracking Algorithms For A Wheeled Mobile Robot," IEEE International Conf. on Industrial Electronics, Control & Instrumentation, pp. 1071-1074.

Reister, D.B. (1991) "A New Wheel Control System For The Omnidirectional Hermies-III Robot," Report No. DE91011517.

Salichs, M.A., Puente, E.A., Gachet, D., and Moreno (1991) "Trajectory Tracking for a Mobile Robot - An Application to Contour following," IEEE International Conf. on Industrial Electronics, Control & Instrumentation, pp. 1067-1070.

Scott, D. (1989) "Easy Roller," *Pop. Sci.*, pp. 31.

Smith, D.E. and Starkey, J.M. (1991) "Overview of Vehicles Models, Dynamics, and Control Applied to Automated Vehicles, "Advances in Automotive Technologies, ASME, DE-Vol. 40.

Sugimoto, G. et al (1988) "Practical Course Follow Performance of an AGV without Fixed Guideways." Proceedings of the USA-Japan Symposium on Flexible Automation - Crossing Bridges: Advances in Flexible Automation and Robotics, Minneapolis, MN, pp. 651- 655.

Wang, C.M. (1988) "Location Estimation and Uncertainty Analysis for Mobile Robot," IEEE International Conference On Robotics and Automation, pp. 1230-1235.

Wong, J.Y. (1978) Theory of Ground Vehicles, John Wiley & Sons, Inc.

Yamazaki, K. and Numazawa, S. (1990) "High Speed and High Accuracy Control of a Direct Drive Robot," The Winter Annual Meetings of ASME on Automation of Manufacturing Processes, DSC-Vol.22, pp.83-88.

Yamazaki, K., Schepper, F de., Kamiyama, M. (1987) "Development of Flexible Actuator Controller for Advanced Machine Tool and Robot Control", Anuals of the CIRP, Vol. 36(1), pp. 285-288.

Yamazaki, K., Schepper, F de., Imai, M. and Satou, J. (1988) "Application of ASIC-Technology to Mechatronics Control: Development of the Flexible Servo Peripheral Chip," Annals of the CIRP, Vol. 37(1), pp. 329-332.

Zelinsky, A. (1991) "Mobile Robot Map Making Using Sonar," Journal of Robotic Systems, Vol. 8, No. 5, pp. 557-578.

APPENDIX A - OPEN LOOP SOURCE CODE

```
REM ************************************************************************
REM ** program for open loop experiment
REM ** FILE NAME: Control.out
REM ** Language: Quick-Basic
REM **
REM ************************************************************************

DECLARE SUB Send3Controller (a%)
DECLARE SUB control ()
DECLARE SUB Send2Controller (a%)

REM ******* this section sets-up the communication for the serial port **************

OPEN "COM1:4800,N,8,1,ds" FOR RANDOM AS #1

500 b$ = INKEY$
      IF b$ = "y" THEN CALL control
      IF (b$ <> "") AND (b$ <> "y") THEN PRINT #1, b$; 505 IF b$ = CHR$(27)
            THEN 2000
510 IF EOF(1) THEN 500
590 a$ = INPUT$(1, #1)
600 IF a$ = CHR$(10) THEN 620       'skip LF
610 PRINT a$;
620 IF LOC(1) > 0 THEN 590
700 GOTO 500
2000 END


SUB control

REM ************* SYSTEM PARAMETERS *******************************
REM ** WHEEL RADIUS(m)
D =  .0295
REM ** (CART TRACK WIDTH)/2 (m)
LA = .106

REM ************** USER INTERFACE *********************************
CLS
PRINT "              INITIAL CONDITIONS"
INPUT "INPUT ANGULAR ORIENTATION (deg)", ANGPOS
PRINT
INPUT "INPUT CENTER OF MASS VELOCITY (m/s)", VELCM
PRINT
INPUT "INPUT ANGULAR VELOCITY (rad/s)", ANGVEL
PRINT
CLS

REM ************* CALCULATIONS ************************************
REM **
Q2dot = (VELCM - (LA * ANGVEL) / 6.283185) * 60 / D
Q3dot = (120 * VELCM / D) - Q2dot
ANGVEL = (D / (2 * LA)) * (Q3dot - Q2dot) * (6.283185 / 60) VELCM = (D / 120) *
(Q2dot + Q3dot)
Q2dot% = -Q2dot: Q3dot% = Q3dot
```

```
REM ************ OUTPUT TO SERIAL PORT ****************************

PRINT #1, "H"
IF Q2dot% < 0 THEN
   Q2dot% = ABS(Q2dot%)
   CALL Send3Controller(Q2dot%)
ELSE
   CALL Send2Controller(Q2dot%)
END IF
PRINT #1, CHR$(13);
IF Q3dot% < 0 THEN
   Q3dot% = ABS(Q3dot%)
   CALL Send3Controller(Q3dot%)
ELSE
   CALL Send2Controller(Q3dot%)
END IF
PRINT #1, CHR$(13);
aa$ = INPUT$(1, #1)
PRINT aa$
Q2dot% = -Q2dot%

REM ************ SCREEN OUTPUT ************************************
LOCATE 2, 20
PRINT "HIT ANY KEY TO STOP CART"
LOCATE 5, 10
PRINT "Q2DOT%(rpm)  Q3DOT%(rpm)  ANGVEL(rad/s)  VELCM(m/s)" DO
LOCATE 7, 10
PRINT USING "####"; Q2dot%
LOCATE 7, 24
PRINT USING "####"; Q3dot%
LOCATE 7, 38
PRINT USING "#.##"; ANGVEL
LOCATE 7, 50
PRINT USING "#.##"; VELCM
LOOP UNTIL INKEY$ <> ""

REM ********** set wheel angular velocities equal to zero ***********************

Q2dot% = 0
Q3dot% = 0

CALL Send2Controller(Q2dot%)
PRINT #1, CHR$(13);
CALL Send2Controller(Q3dot%)
PRINT #1, CHR$(13);

Q2dot% = 9999: Q3dot% = 9999
CALL Send2Controller(Q2dot%)
PRINT #1, CHR$(13);
CALL Send2Controller(Q3dot%)
PRINT #1, CHR$(13);


REM ****** do loop which waits for ANY_KEY to be pressed ********************
```

```
DO
LOOP UNTIL INKEY$ <> ""

END SUB

REM ****** subroutine to calculate scaled output values to controller ***************

SUB Send2Controller (a%)
digit1% = 0: digit2% = 0: digit3% = 0: digit4% = 0
digit1% = INT(a% / 1000)
digit2% = INT((a% - digit1% * 1000) / 100)
digit3% = INT((a% - digit1% * 1000 - digit2% * 100) / 10) digit4% = a% - digit1% *
1000 - digit2% * 100 - digit3% * 10 sdg1$ = CHR$(digit1% + 48)
sdg2$ = CHR$(digit2% + 48)
sdg3$ = CHR$(digit3% + 48)
sdg4$ = CHR$(digit4% + 48)
'IF digit1% < 0 OR digit2% < 0 OR digit3% < 0 OR digit4% < 0 THEN '   PRINT #1,
CHR$(45);
'END IF
PRINT #1, sdg1$; sdg2$; sdg3$; sdg4$;
END SUB

REM ****** subroutine to calculate scaled output values to controller ***************

SUB Send3Controller (a%)
digit1% = 0: digit2% = 0: digit3% = 0: digit4% = 0
digit1% = INT(a% / 1000)
digit2% = INT((a% - digit1% * 1000) / 100)
digit3% = INT((a% - digit1% * 1000 - digit2% * 100) / 10) digit4% = a% - digit1% * 1000 -
digit2% * 100 - digit3% * 10 sdg1$ = CHR$(digit1% + 48)
sdg2$ = CHR$(digit2% + 48)
sdg3$ = CHR$(digit3% + 48)
sdg4$ = CHR$(digit4% + 48)
PRINT #1, CHR$(45);
PRINT #1, sdg1$; sdg2$; sdg3$; sdg4$;
END SUB
```

APPENDIX B - CLOSED LOOP SOURCE CODE

Figs. (B.1) and (B.2) are given to help better understand the computer source code used for the closed-loop tests. Fig. (B.1) shows the TMR platform following a crack, the parameters "Offset", "d", and "@c" are used in the control equations. Fig. (B.2) is a block diagram of the control equations used in the computer source code.

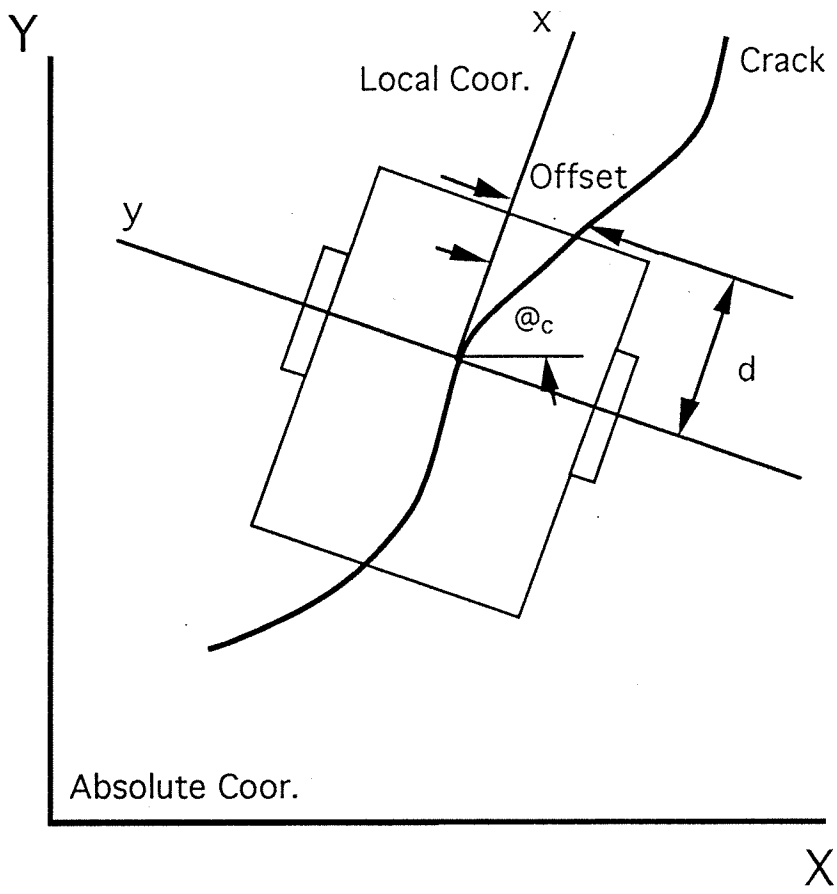


Figure B.1 TMR with parameters "Offset", "d", and "@c" used in control equations.

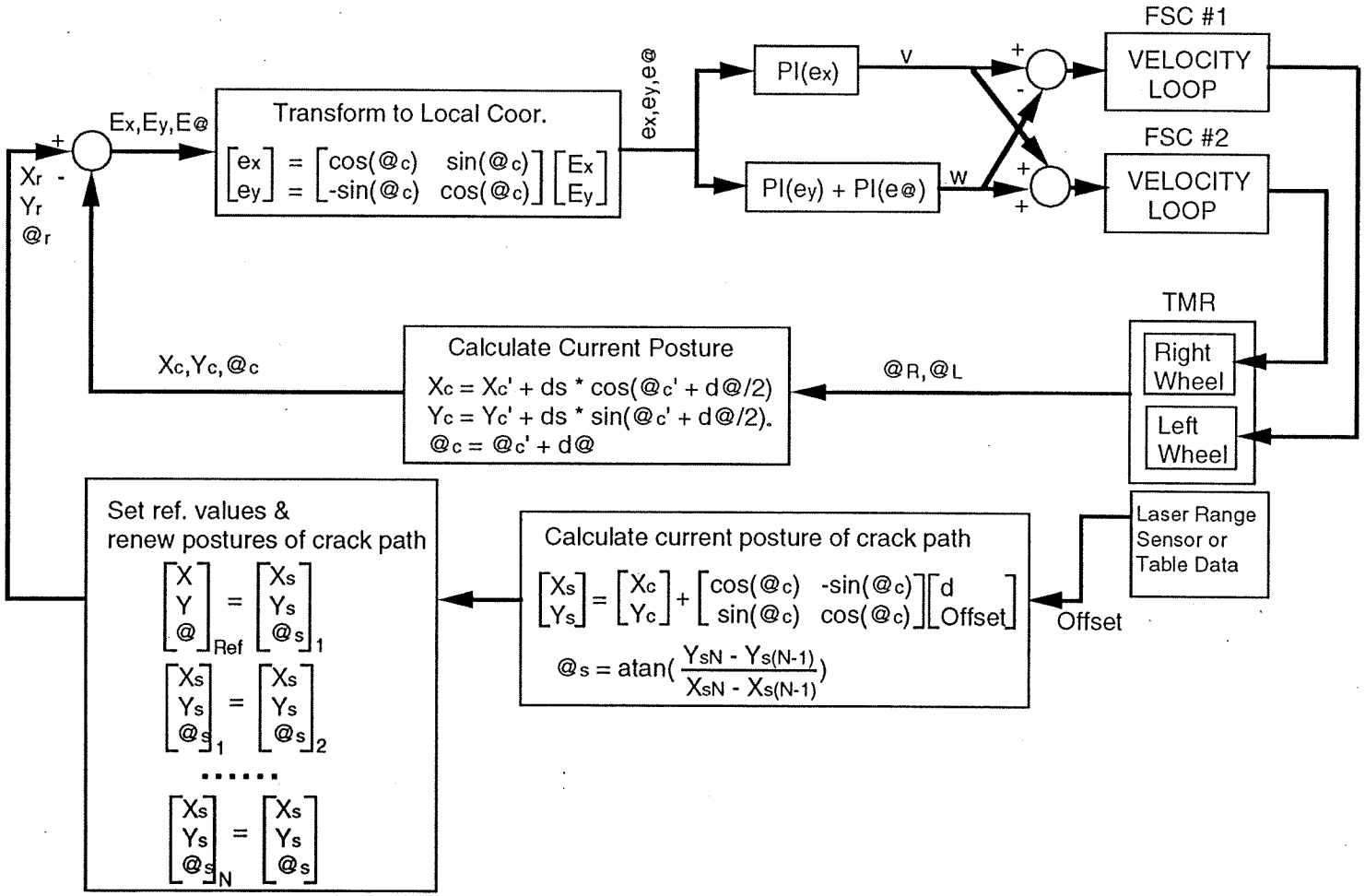


Figure B.2 Block diagram of control equations

The variables used in the block diagram have the following meaning:

ex,ey,e@c = Local coordinate error
Ex,Ey,E@c = Absolute coordinate error
@c = Angle between absolute and local coordinates
d@c = Angle difference between one time step in local coordinates
Xs,Ys = Absolute position
Xc,Yc = Current position
Xc',Yc' = Previous current position

```
/*********************************************************/
/*          This program is for the closed loop test
 *          Written by: Hong, Winters
 *                                                       */


#include <stdio.h>
#include <conio.h>
#include <dos.h>
#include <bios.h>
#include <stdlib.h>
#include <time.h>
#include <math.h>

float WheelRadius=29.5, CartLength=106.0, Conv2Radian=2*3.141592/2000;

#define ESC_KEY 0x1b
#define CR              13
#define LF  10
#define TRUE            1
#define FALSE           0
#define COM1            0
#define COM2            1
#define BUFLEN  0x4000                  /* data buffer length*/

#define DATA            0x3f8           /* Data buffer register*/
#define IER             0x3f9           /* Interrupt Enable Register*/
#define IIR             0x3fa           /* Interrupt ID Register*/
#define LCR             0x3fb           /* Line Control Register*/
#define MCR             0x3fc           /* Modem Control Register*/
#define LSR             0x3fd           /* Line Status Register*/
#define MSR             0x3fe           /* Modem Status Register*/

#define ONMSK           0xe7
#define OFFMSK  0x18
#define PICMSK  0x21                    /* 8259 Mask Register*/
#define PICEOI  0x20    /* 8259 EOI Command*/
#define IRQ4            0x0c            /* COM1 interrupt vector*/
#define IRQ3            0x0b            /* COM2 interrupt vector*/

#define DATA7           0x02            /* 7 data bits*/
#define DATA8           0x03            /* 8 data bits*/
```

```
#define STOP1       0x00            /* 1 stop bit*/
#define STOP2       0x04            /* 2 stop bit*/

#define NOPA        0x00            /* No parity*/
#define ODDP        0x08            /* Odd parity*/
#define EVENP       0x18            /* Even parity*/

#define B1200       0x80            /* 1200 Baud Rate*/
#define B2400       0xa0            /* 2400*/
#define B4800       0xc0            /* 4800*/
#define B9600       0xe0            /* 9600*/

FILE        *fp1,*file_pointer;

unsigned char ibuffer[BUFLEN];      /* input data buffer*/
unsigned int  ifront = 0, irear = 0;  /* data pointers*/
float       ThetaC=0.0,Xc=0.0,Yc=0.0;
float       Xref , Yref, ThetaRef;
float       Kxp = .6, Kyp = .2, Ktp = 15, Kxi = .03, Kyi = .01, Kti =5;
float       D = 70.0;
int         P_Limit = 100, N_Limit = -100, N=5;

void interrupt far intsr_sc(void);

void (interrupt far *oldvect)();        /* save original vector */

void interrupt far intsr_sc()
/*          Interrupt Service Routine*/
/*          This routine is interrupted whenever Data Ready*/
/*          and reads data from data port and save it to buffer*/
{
            irear %= BUFLEN;
            ibuffer[irear++] = (char) inp(DATA);
            outp(PICEOI,0x20);
}

int rsready()
/*          check if the buffer has an input data*/
{
            return((ifront != irear) ? TRUE : FALSE);
}

unsigned char rsin()
/*          read one cahracter from buffer*/
{
            ifront %= BUFLEN;
            return (ibuffer[ifront++]);
}

unsigned char rgetc()
/*          read one char if the buffer has input char.'s*/
/*          return NUL if there is not input char after wating*/
/*          for some time           */
{
```

```
int tzcount = 0;
int timeout = 0;

        while(!rsready()) {
                tzcount ++;
                if (tzcount > 10000) {
                        timeout ++;
                        break;
                }
        }
        if (timeout) return(0);
        else return(rsin());
}

void rsout(unsigned char ch)
/*      send one char through communication line*/
{
        do
        {
                /*      .....       */
        } while( !(inp(LSR) & 0x20) );  /* wait until ready to send */ outp(DATA,ch);
        /* send a char to data port */
}

#define DELAY 1000
void sendline(unsigned char *s)
/*      send a string to end with NUL char*/
{
        int i=0;
        while(*s){
                rsout(*s++);
                for( i ; i<DELAY ; i++);
        }
        i=0;
        rsout(CR);
        for ( i; i<DELAY ;i++);
}

void init_serial_port(void)
/*      initialize serial port                  */
{
        unsigned data;
        data = ( DATA8 | STOP1 | NOPA | B4800 );
        _bios_serialcom(_COM_INIT,COM1,data);
}

void init_port()
/*      initialize communication port and interrupt enable register */
{
        oldvect = _dos_getvect(IRQ4);       /* save old vector    */ _dos_setvect(IRQ4,
        intsr_sc);        /* set a new vector   */

                init_serial_port();
```

```
                outp(MCR,0x0b);
        outp(IER,0x01);  /* set bit 0 of interrupt enable register  */ /* so, interrupt can
                                happen when data ready*/
        outp(PICMSK, (inp(PICMSK) & (ONMSK)));
                                /* set INT mask bit of 8259 PIC        */ }

void close_port()
/*      restore the original vector and INT mask bit          */
{
        outp(MCR,0x00);      /* disconnect phone line      */ outp(IER,0x00);      /*
        interrupt disable         */
outp(PICMSK, (inp(PICMSK) | OFFMSK)); /* restore INT mask bit of PIC */
_dos_setvect(IRQ4,oldvect);  /* restore the original vector */
}

/* this portion of the code is for control*/
void pcontrol()
{
        unsigned char ch, s1[9], s2[9],c1[9],c2[9];
        float d_increment1=0, d_increment2=0, d_theta=0, d_s=0,\
Ex=0,Ey=0,Et=0;
        float Xe=0, Ye=0,V=0,W=0,Vr=0,Vl=0,Exi=0,Eyi=0,Eti=0,WR2,\
WR2C;
        float sinT, cosT, d_inc1 = 0, d_inc2 = 0;
        float offset, Xs[10],Ys[10],Ts[10];
        int iii=0, jjj=0, Vrc, Vlc,i;
        time_t ticksnow;
        unsigned tused;

        printf("..... Control loop \n");
        printf("                    Hit any key to escape \n");
        fprintf(fp1,"d_increment1 d_increment2 Xref Yref ThetaRef \

Xc Yc ThetaC Ex Ey Et V W Vlc Vrc\n");

        for (i=0;i<10;i++)
        {
                Xs[i]=0.0;
                Ys[i]=0.0;
                Ts[i]=0.0;
        }

        WR2 = WheelRadius/2.0;
        WR2C = WR2/CartLength;

        do
        {
                rsout('P'); rsout(CR);
                while( (ch=rgetc()) != ' ' )
                {
                        s1[iii] = ch;
                    iii++;
                }
                while( (ch=rgetc()) !=
```

```
CR ) {
                s2[jjj] = ch;
                jjj++;
}
ch = rgetc();
d_increment1 = (float) atof(s1);
d_increment2 = (float) atof(s2);

if (((d_increment1>0) && (d_inc1>0)) ||
((d_increment1<0) && (d_inc1<0))) {
                if( abs(d_increment1-d_inc1) > 300.0 )
 d_increment1 = d_inc1;
}
else
{
                if ((abs(d_increment1)+abs(d_inc1)) > 300.0)\

 d_increment1=d_inc1; }

if (((d_increment2>0) && (d_inc2>0)) || \
((d_increment2<0) && (d_inc2<0))) {
                if( abs(d_increment2-d_inc2) > 300.0 )\
d_increment2 = d_inc2;
}
else
{
                if ((abs(d_increment2)+abs(d_inc2)) > 300.0)\

 d_increment2=d_inc2; }

d_inc1 = d_increment1;
d_inc2 = d_increment2;


iii=0; jjj=0;
if (rsready()) ch = rsin();

d_increment1 = -d_increment1*Conv2Radian;
d_increment2 = d_increment2*Conv2Radian;

d_theta = (d_increment2 - d_increment1)*WR2C;
d_s = (d_increment1+d_increment2)*WR2;

sinT = (float) sin(ThetaC+d_theta/2.0); cosT =
(float) cos(ThetaC+d_theta/2.0);

Xc=Xc+d_s*cosT;
Yc=Yc+d_s*sinT;
ThetaC = ThetaC+d_theta;

fscanf(file_pointer,"%f\n", &offset); printf(" offset :
%f\n",offset);

sinT = (float) sin(ThetaC);
```

```
cosT = (float)
cos(ThetaC);

for (i=0;i<N-1;i++)
{
        Xs[i] =
        Xs[i+1];
        Ys[i] =
        Ys[i+1];
        Ts[i] =
        Ts[i+1];
}
Xs[N-1] = Xc + D*cosT - offset*sinT;
Ys[N-1] = Yc + D*sinT + offset*cosT;
Ts[N-1] = ThetaC + (float) atan( offset/D );

Xref = Xs[0];
Yref = Ys[0];
ThetaRef = Ts[0];

fprintf(fp1,"%2.3f %2.3f %5.2f %5.2f %5.2f %5.2f \
%5.2f %5.2f ",d_increment1,d_increment2,Xref,Yref,ThetaRef,Xc,Yc,ThetaC);

Xe = Xref - Xc;
Ye = Yref - Yc;


Ex = Xe*cosT + Ye*sinT;
Ey = Ye*cosT - Xe*sinT;
Et = ThetaRef - ThetaC;

Exi = Exi + Ex;
Eyi = Eyi + Ey;
Eti = Eti + Et;

V = Kxp*Ex + Kxi*Exi;
W = Kyp*Ey + Kyi*Eyi + Ktp*Et + Kti*Eti;

Vr =  V + W;
Vl = -V + W;

Vrc = (int) Vr; Vlc = (int) Vl;

if (Vrc > P_Limit) Vrc = P_Limit;
if (Vrc < N_Limit) Vrc = N_Limit;
if (Vlc > P_Limit) Vlc = P_Limit;
if (Vlc < N_Limit) Vlc = N_Limit;

        rsout('S');
rsout(CR);
        if (rsready()) ch =
rsin();
```

```
                                        itoa(Vlc,c1,10);
                                        itoa(Vrc,c2,10);
                                          sendline(c1);
                                if(rsready()) ch = rsin();
                                sendline(c2);
                                if (rsready()) ch = rsin(); ticksnow
                                = clock();
                                tused = (unsigned) ticksnow;
fprintf(fp1,"%5.2f %5.2f %5.2f %5.2f %5.2f %s %s %u \n",Ex,Ey,Et,V,W\

,c1,c2,tused);

        } while(!kbhit());


}

menu()
{
        printf("*****************************************\
********\n");
        printf("*               \n");
}

void main()
{
unsigned char ch, in;

        fp1 = fopen("tmr.out","w");
        file_pointer=fopen("output.tmr","r");
        init_port();
        menu();
        while(1) {
                    if (rsready()) putch(rsin());
                if (_bios_keybrd(_KEYBRD_READY)) {
                        ch = _bios_keybrd(_KEYBRD_READ) &
0xff;
                        if (ch == ESC_KEY) pcontrol();
                        else rsout(ch);
                        if (ch == 45) break; /* Alt-X then exit...*/ }
        }
        close_port();


}
```