DYNAMIC MODELING AND CONTROL OF NONHOLONOMIC WHEELED MOBILE ROBOT SUBJECTED TO WHEEL SLIP

By

Shahrul Naim Sidek

Dissertation

Submitted to the Faculty of the

Graduate School of Vanderbilt University

in partial fulfillment of the requirements

for the degree of

DOCTOR OF PHILOSOPHY

in

Electrical Engineering

December, 2008

Nashville, Tennessee

Approved:

Prof. Nilanjan Sarkar

Prof. George E. Cook

Prof. D. Mitch Wilkes

Prof. Carol A. Rubin

Prof. Akram Aldroubi

In the memory of my Father

ACKNOWLEDGEMENTS

These past few years have been a time of challenge, and growth for my research as well as myself. Much has changed since I first walked into the compound of Vanderbilt School of Engineering five years ago. One thing that has remained consistent is the support of many people and my passion for the applicable research. I would like to take this opportunity to give my sincere gratitude to those who have not only helped to make this dissertation possible but who have also supported me in various area of my life.

I dedicate my heartfelt thanks to Prof. Sarkar, my teacher and mentor, whose guidance, support, time, and patience are invaluable. He has shown me, by example, how a good scientist should be.

My appreciation also goes to my Dissertation Committee, Prof. Cook, Prof. Wilkes, Prof. Rubin and Prof. Aldroubi for their times and interests in this dissertation. I must express my gratitude to my friends, Uttama, Yu, Karla, Furui, Jadav, Dr. Halder, Dr. Erol, Vishnu and Vikash who together provide a conducive research environment.

Nobody has been more important to me in the pursuit of this work than the members of my family. I would like to thank my mother who taught me that constant perseverance would allow me to achieve anything in life. Most importantly, I wish to thank my loving and supportive better half, Azweeda, and my children, Ammaar and Nawfal Naim who provide me unending love, inspiration and motivation.

I also gratefully acknowledge ONR grant N00014-03-1-0052 and N00014-06-1-0146 that partially supported this work.

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

global position system (GPS)

inertial measuring unit (IMU)

wheeled mobile robot (WMR)

center of mass (COM)

degree of freedom (DOF)

proportional-integral-derivative (PID)

laser measurement sensor (LMS)

graphical user interface (GUI)

CHAPTER I

INTRODUCTION

Wheeled mobile robot - definition, current applications, and future potentials

A wheeled mobile robot (WMR) is defined as a wheeled vehicle that can move autonomously without assistance from external human operator. The WMR is equipped with a set of motorized actuators and an array of sensors, which help it to carry out useful work. In order to govern its motion, usually, there is an on-board computer to command the motors to drive, based on reference inputs and the signals gathered by the sensors.

Unlike the majority of industrial robots that can only move about a fixed frame in a specific workspace, the WMR has a distinct feature of moving around freely within its predefined workspace to fulfill a desired task. The mobility of WMR makes it suitable for a variety of applications in structured as well as in unstructured environments. For examples, *Spirit*, the NASA's Mars rovers (URL 1.1) have successfully demonstrated its ability to achieve the mission goals in exploring and running experiments on the red planet. In military and high-risk hazardous environments, AB Precision Ltd (URL 1.2) has developed *Cyclops*, a miniature remotely operated vehicle that has been in use in many military and law enforcement organizations worldwide. It provides distinct advantages over human operators to complete critical missions in a safe manner. *Whiskers*, developed by Angelus Research Corp. (URL 1.3) is a programmable intelligent mobile robot that has been an impetus to many, to learn more about robots. The list goes

as the WMR can also be found in other field of applications such as in mining, transportation, entertainment and so on.

The ever increasing demand and applications of WMRs justify the research needs and potentials of this very fascinating topic. We should expect WMR in the future to have stronger autonomous capabilities and higher agility, be able to self-learn and reliable for continuous operation regardless of time and environment.

Research on WMR - modeling, planning, control etc.

In general, the research on WMR can be divided into several components namely the modeling of the WMR, the planning and the navigation strategies, the localization techniques, the communication system and the mobility (i.e., control task) (URL 1.4). The relationship between all these components is shown in Fig. (1.1).

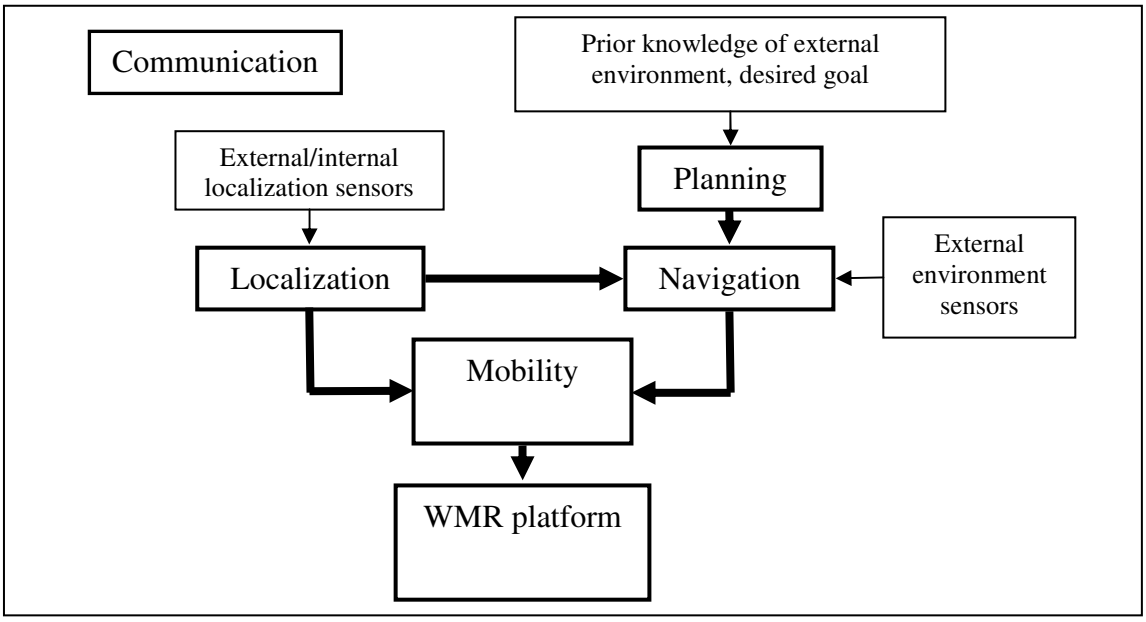


Fig. 1.1: The relationship between components in the autonomous control application of WMR

The research in mobility of the WMR is related to understand the physical mechanics of the robot platform, the model of the interaction between the robot and its environment as well as the overall effect of control algorithm on the WMR. In localization, the research objective is to estimate the location, attitude, velocity and acceleration of the WMR. Navigation is concerned with the acquisition of and response to external sensed information to execute the mission. Meanwhile, research in planning is related to behaviors, trajectories or waypoints generation for the robot mission. Lastly, the goal of communication research is to provide the link between WMR and any remaining elements in the whole system, including system operators or other WMRs.

Research – problem statements

As the demands on WMR increases, it becomes necessary to improve the performance of the WMR. While the WMR performance has steadily improved over the years for conventional applications (e.g., slow speed maneuvering in a structured environment), it remains a challenge to operate a WMR at high speed in an unstructured environment. However, this is an important area of application where successful WMR deployment could be beneficial (e.g., in battlefield). The primary objective of the dissertation is to address the above challenge. We argue that one solution to improve the performance is through better modeling of the system. In particular, for model-based control approaches, which are the most widely used techniques to control WMR, the ability to develop a realistic model will greatly benefit the development of advanced controllers. While the modeling of WMR has been extensively studied from an ideal perspective in which the wheel rolls without slip and the WMR does not move laterally

instantaneously, there is little research that models wheel slip and consider the effect of traction force on the motion of the WMR. In this dissertation we first address the issue of improving the WMR model by incorporating wheel slips and traction forces into the WMR dynamics so that the model can reliably and accurately predict WMR navigation performance on different surfaces with the presence of wheel slips. As a result, the consideration of the above factors in modeling the WMR promises a better performance of the system through the design of model-based controllers. Specifically, as a secondary objective of this research, we design a dynamic path following controller that is guided by the conceptual idea of the driving behavior of a human driver. Based on the driving cues of the expert, such as a rally-car driver, who utilizes slip to garner the full potential of the vehicle traction force, we design a dynamic velocity planner that regulates the forward velocity of the WMR to optimize the traction force.

Research potential

The consideration of wheel slip and traction force in modeling the WMR is shown to increases the accuracy and consistency of the model in this research. In particular, as a direct benefit of having a precise model, a variety of control algorithms based on the model-based approaches can be designed and potentially provide better control over the system. In addition, the model developed in this research has a potential to enhance the performance of the WMR simulator performance currently available such as *Player/Stage* (URL 1.5), *Microsoft Robotics Studio* (URL 1.6), *COSIMIR* (URL 1.7), *The Webots Simulator* (URL 1.8), *and OpenSim* (URL 1.9). These simulators are used to provide realistic simulation environment to develop planning and control methodologies for

WMRs. These simulation platforms, however, do not provide mechanisms to model wheel slip and thus may not be able to describe the robot behavior correctly and effectively where wheel slip is a critical factor (URL 1.10). Instead these simulation platforms offer high-level planning strategies to offset the effect of slip when the WMR is deployed in real-world environment (Peasgood, 2008).

Dissertation organization

The dissertation is organized as follows: in Chapter II, we present the survey of existing literatures that is relevant to our research work. This is followed by the discussion on the framework of our modeling approach in Chapter III. We discuss traction force thoroughly in Chapter IV. In the following Chapter V, the new dynamic model of WMR is verified through a series of experiments. In addition, the design of dynamic planner with experimental results is presented in this chapter.

We describe the formulation of the dynamic path following controller that is motivated by the way a human drives a car in Chapter VI. Simulation results are presented in Chapter VII to demonstrate the efficacy of our proposed model and control technique on surfaces with different friction coefficients. Chapter VIII concludes this dissertation with a list of contributions including some suggestions of possible future work.

CHAPTER II

LITERITURE SURVEY

This dissertation covers a broad range of research areas such as mechanics and control of mobile robots, wheel slip phenomenon, traction force at wheel-surface contact point, and mobile robot simulator. Thus, it requires broad literature survey encompassing multidisciplinary areas. For the theoretical part of this dissertation, we assume some basic knowledge of kinematics, dynamics, control theory, and mathematics while for the application part we require some idea on robotic systems (i.e., mobile robot hardware and software). There are many textbooks available that discuss the background knowledge used in the theoretical part of the dissertation. For instance, the readers who are interested in a good introduction on kinematics, dynamics and control of robotic systems can refer to (Craig 2004).

While the majority of WMR models presented in the literature do not consider slip, there are a few recent articles that discuss the slip phenomenon. Our review focuses on these WMR models, which include slip as a part of their models. We then discuss several control techniques pertaining to WMR navigation problems. The relationship between slip and traction force for a variety of surfaces and a few state-of-the-art techniques and devices to measure the slip are discussed afterwards.

The WMR is useful in many applications mentioned in Chapter I. Majority of these WMR platforms use standard wheels over omni-directional wheels due to the

inherent mechanical simplicity. These WMRs are called nonholonomic mobile robots because of the velocity constraints imposed due to the structure of the wheels.

A car is an example of a four-wheel vehicle system that shares many similarities with a WMR system due to the same wheel structure. Nevertheless, we found many works especially written by the people from the vehicle community, which do not consider constraint equations in their model of the vehicle dynamic as found in (Lagerberg & Egardt 2007, Verma, Vecchio, & Fathy 2008, Kyung-Ho 2007, Der-Cheng & Wen-Ching 2006). This is mainly due to model simplification. The modeling of a vehicle system such as a car system has to consider the modeling of the power and drive trains of the system, which already results in a relatively complex model.

On the other hand, a review from WMR literatures indicates that conventional modeling of a WMR assumes nonholonomic, no-slip constraints at the contact point between the wheel and the ground surface (Sarkar 1993, Conceicao et al. 2007, Dongbin et. al 2007, Eghtesad & Necsulescu 2006, Liyong & Wei 2007, Salerno & Angeles 2007). As a matter of fact, the work in this research is fundamentally the extension of the work done by (Sarkar 1993), who pioneered the investigation related to the nonholonomy of WMR systems. The constraints in these works are developed based on the ideal case where the longitudinal motion of the wheel is contributed by pure rolling and as such the longitudinal slip at the contact point between the wheel and the surface is always zero. The lateral slip at the contact point is also assumed to be zero. Thus, the issue of traction force is not relevance in those works. Such assumptions are legitimate if the WMR moves slowly, on a regular surface where slip is negligible. However, as the technology becomes more sophisticated and robust, the application domains of WMR expand,

requiring the WMR to navigate at a faster pace and on unstructured environment such as slippery and irregular surfaces. In such cases, it is important to note that the pure rolling/no-slip assumptions cannot be satisfied. In fact, slip will always exist as the wheel continues to roll on a surface and thus it becomes important to consider the effect of slip on WMR motion.

As mentioned before, there are a few recent works on WMR that consider slip at various levels. We limit our discussion only to these works since they are directly relevant to our research. In (Tarokh & McDermott 2005, Dixon, Dawson, & Zergeroglu 2000, Volpe 1999) the author presented a comprehensive methodology to develop a generalized kinematics model of an articulated rover, which includes the side, the rolling and the turning slips. They proposed an inverse kinematic based control approach to minimize the effect of slips during navigation. In (Dixon, Dawson, & Zergeroglu 2000), slip was treated as a small, measurable, bounded perturbation to the robot kinematic model. The goal is to develop a robust controller that can function well in the presence of slip.

(Motte & Campion 2000) is one of the earliest papers that consider slip in the WMR dynamic model. The authors modeled the slip as a small constraint violation, which introduced a sliding effect into the original dynamics model through the singular perturbation formulation. (Lin et al. 2007) represented a novel way to model the constraint violation due to wheel slip using an anti-slip factor. By using anti-slip control, the simulation results promised stability for the desired robot trajectory. (Balakrisna and Ghosal 1995) indirectly included traction force in the system model by measuring the magnitude of slip. However, the slip was assumed to be very small, thus, was omitted in

the system dynamic equation. (Jung and Hsia 2005, Stonier et al. 2007) applied similar concept but specific to lateral traction on a bicycle model and on an omni-directional WMR respectively. (Stonier et al. 2007) introduced the notion of slip space to analyze the dynamics of slip of an omni-directional WMR.

Slip is necessary to generate traction force at the contact point between the wheel and the surface that is responsible for the motion of the WMR. The optimized usage of the available traction force between the wheel and the surface can contribute to better maneuverability of the WMR as well as less wear and tear of the wheel. This can be done especially through the control of the slip velocity which in turn has a direct relation to the traction force. On the flipped side, excessive slip may generate instability in motion and thus should be handled carefully. For this research we define excessive slip to be the slip that exceeds a predefined magnitude of slip based on a desirable performance. We argue that in order to enhance the previous works on nonholonomic WMR, which neglect the effect of slip phenomenon, slip and traction force, must be considered in the new dynamic system model.

Based on the previous discussion on the WMR modeling technique, (Motte & Campion 2000, Stonier et. al. 2007) developed a model-based controller, which was based on the pure rolling/no-slip condition. It is noted that this method works if the slip ratio is considerably small and covers the linear part of the traction curve. In (Stonier et. al. 2007) the author extended this work by comparing their model-based controller to a PID controller and (Motte & Campion 2000) applied a slow manifold technique to handle the violation of constraint equation due to the bigger but still linear slip. In (Jung & Hsia 2005), the author discussed the lateral force control using the force and position

controllers that include slip factor. (Zhu et. al. 2006) introduced a robust controller for trajectory tracking application by augmenting the WMR kinematics model with slip in the form of a transverse function and the stability is confirmed though *Lie* group operation. (Ishigami, Nagatani, & Yoshida 2007) reported a simpler way to design a navigation controller, where by having prior knowledge of terrain characteristic including the slip, they generated a candidate path for the mobile robot to follow. Note that the objectives of the above-mentioned controllers are either to minimize the effect of slip or to keep the slip bounded. None of the works actively sought to optimize the magnitude of slip and thus the amount of traction force to improve the navigation performance.

It is clear that, in order to model slip and traction force in the dynamics we need to measure slip and know the relationship between slip and traction force for a given surface. In this research, we assume that such information is available to us to design the controller. There are various research groups that have been working in this area and we leverage their results for our research. In what follows, we briefly discuss the current state-of-the-art techniques and technologies in slip measurement and slip-traction relationship mapping.

A variety of techniques have been reported to classify terrain types. Among the most common are the uses of frequency modulated sonar signal (Politis, Probert, Smith 2001) and laser signal (Vandapel et. al. 2004). These techniques basically work by capturing any distinct signature from the reflected signal which later can be associated to a particular type of surface. (Weiss, Frohlich, & Zell 2006, Brooks & Iagnemma 2005) proposed to mount an accelerometer unit on the WMR body to capture vibration signal generated when the robot was in the move. The processed vibration signal then can be

used to classify the surface types. Basically, once the surface type is known, we can proceed to use the relevant traction curve.

Traction curve is a function of slip velocity. The analytical mapping between slip velocity and traction force however is difficult to be formulated due to a variety of factors such as wheel temperature, thread pattern, camber angle and so on. Nevertheless the general behaviors of this relationship, in particular, for rubber tire, have been reported in (Germann, Wurtenberger, & Daiss 1994). (Li & Wang 2006) provided an excellent review of current trends in modeling the traction forces using different methodologies namely empirical, semi-empirical and analytical methods. Amongst methods discussed were piecewise linear model, *Buckhardt* model, *Rill* model, *Dahl* model, *Lugre* model and *Pacejka* model or better known as *magic formula*. In order to measure the slip, different combination of sensors and data processing techniques can be used and have been reported in the literatures. (Ward & Iagnemma 2007, Ray 1997) adopted *Kalman* filtering technique to filter the data collected from the wheel encoder, global positioning system (GPS) and inertial measuring unit (IMU). (Seyr & Jakubek 2006) presented a purely proprioceptive navigation strategy using gyro, accelerometers and wheel encoders. The states (i.e., slip accelerations) were estimated using the extended *Kalman* filter. This data represents the states of the WMR components. (Helmick et. al. 2004, Helmick et. al. 2005) used a similar approach to analyze the data from stereo imagery unit and IMU before the results were compared to the kinematics estimator. (Angelova et. al. 2006) predicted the amount of slip by learning from the previous examples of imagery data, recorded using stereo imagery unit.

CHAPTER III

DYNAMIC MODELING OF A WHEELED MOBILE ROBOT

The motion of a mechanical system is related through a set of dynamic equations, and the forces or torques the system is subjected to. The dynamics of a mechanical system has been discusses in a numerous scholarly literatures related to engineering mechanics and analytical mechanics. The study of this subject is important due to the problem of the control of the system, which is in contact with its environment. Indeed, for a model-based controller, the synthesis of the controller of such system depends heavily on the mathematical model of the physical structure of the system, which is intrinsically nonlinear. Thus, the goal here is to develop a model that can describe the dynamics of the system as close to the real system so that we can have a better chance to develop a controller for the system that is effective in real-world situation. In the case of a WMR, the contact with the environment occurs at the contact point between the wheel outer surface and the ground surface. The interaction between these two surfaces has a significant influence over the dynamic motion of the system, and hence, need to be properly modeled.

In general, there are two major methods for deriving the dynamic equations of mechanical systems namely Newton's method that is directly related to Newton's 2nd law and Lagrange's method that has its root in the classical work of d'Alembert and Lagrange on analytical mechanics. The main difference between the two methods is in dealing with constraint equations. While Newton's method treats each rigid body separately and

explicitly model the constraints through the reactive force required to enforce them, Lagrange's provides systematic procedures for eliminating the constraints from the dynamic equations, typically yielding simpler system equations. Thus, it is not surprising that the majority of the conventional WMR models we found in the literature were developed using Lagrange's as a method of choice.

In this chapter, we first describe system constraints and formulate the dynamic equation of the conventional WMR using Lagrange's method. While the parts of the generalized formulation are not new and were previously explored by several researchers (Conceicao et al. 2007, Dongbin et. al 2007, Eghtesad & Necsulescu 2006, Liyong & Wei 2007, Salerno & Angeles 2007), the formulation is needed and serves as an important platform to combine the slip dynamics with that of the WMR. We then develop a detailed model for a two-WMR (i.e., two differential driving wheels) that is one of the most commonly available WMRs using the generalized formulation. Following that is the detailed discussion on the formulation of the dynamics model of a two-WMR with the inclusion of slips. This model is first developed using Lagrange's, allowing us to present the model in the standard form. In order to ensure model consistency, we present the derivation of dynamic equations using Newton's method. We list down all the variables with their definitions to help in the modeling process of the WMR system in the following table.

Table 3.1: Variables used in the model formulation and their definitions

$\{X, Y\}$	The coordinate system for the inertial frame
$\{x, y\}$	The coordinate system for the WMR reference frame

P_o	the origin of the WMR reference frame with coordinate $\{x_o, y_o\}$
P_c	the center of mass of the WMR with coordinate $\{x_c, y_c\}$
P_l	the virtual reference point (the look-ahead point) attached to the WMR with coordinate $\{x_l, y_l\}$
φ	the angular displacement of WMR
θ_1, θ_2	the angular displacement of the driving wheels
ξ_1, ξ_2	the longitudinal displacement of the driving wheels due to slips
η_1, η_2	the lateral displacement of the driving wheels due to slips
m_r	the effective mass of the WMR without the driving wheels and the motors
m_w	the effective mass of the driving wheels and the motors
I_{rz}	the moment of inertia of the WMR without the driving wheels and the motors taken at the center of mass about the vertical axis.
I_{wz}	the moment of inertia of the driving wheels and the motors about the vertical axis
I_{wy}	the moment of inertia of the driving wheels and the motors about the wheel axis (the WMR lateral axis)
l	the distance between the point P_c and P_l
d	the distance between the point P_o and P_c
b	the distance between the driving wheel and the origin of axis of symmetry of the robot reference frame
a	the length of the WMR platform parallel to it *x*-axis
h	the height of the WMR platform in the direction of the *z*-axis
r	the radius of the driving wheels

System constraints

Mechanical systems can be classified into linear and nonlinear systems. Nonlinear systems can be further classified into holonomic and nonholonomic systems. A wheeled mobile robot is an example of mechanical system that falls under the latter category. Holonomy and nonholonomy are fundamental concepts that describe the constraints of the systems, which play the essential part in governing the motion of those systems.

In the following we define some terms related to the discussion. These definitions are taken from (Rosenberg 1977, Sarkar 1993).

Lagrangian coordinates: Set of coordinates, Q, (not necessarily a minimal set) that are required to distinctively specify the configuration of the system.

If the number of Lagrangian coordinates is more than the number of degree of freedom (DOF) of a system, N, then we may assign N of the Lagrangian coordinates as *primary coordinates*. The remaining coordinates are called *secondary coordinates*. In classical mechanics, the primary coordinates are called *generalized coordinates.*

Catastatic and acatastatic constraints: The general form of equality constraints considered in classical mechanics is given as,

$$\sum_{s=1}^{n} A_{rs} dq_s + A_r dt = 0 \quad , r \in \{1,2, \dots, m\}$$

in which $q = [q_1 \ q_2 \ \dots \ q_n]^T$ are the generalized coordinates of the dynamic systems, t is the time, and A_{rs} and A_r are at least one piecewise differentiable function of q. These m linear differential forms are called *Pfaffian* form. If A_r is zero, and the set of constraint is called catastatic and the resulting dynamic system is called catastatic system. Otherwise the constraint is called acatastatic.

Holonomic constraint: Any constraint of the form,

$$f_c(q,t) = 0 \tag{3.1}$$

where q is the generalized coordinate of the system and t is the time.

Nonholonomic constraint: Any constraint that cannot be reduced (i.e., non-integrable) to form Eq. (3.1)

In general cases of mechanical systems, nonholonomic constraint can be written as,

$$f_c(q,\dot{q},t) = 0 \tag{3.2}$$

A holonomic constraint as in Eq. (3.1) reduces the degree of freedom (DOF) of the system. For instance if there is m number of holonomic constraints with n number of generalized coordinates q_j, where $j \in \{1,2,\dots,n\}$, the number of independent coordinates are $n-m$, which is the DOF of the system. Thus, $n-m$ number of coordinates is needed to describe the system configuration and $n-m$ number of inputs is required to drive the system.

On the other hand, for nonholonomic constrained system, there are two types of DOF, namely the DOF in the small (for infinitesimal displacements), which is $n-m$ and DOF in the large (for finite displacements). The DOF in the large is the same as the minimum number of independent coordinates required to specify the configuration of the system. Nonholonomic system has fewer DOF in the small than that in the large.

For a nonholonomic WMR system, Eq. (3.2) can be further simplified to,

$$f_c(q,\dot{q}) = 0 \tag{3.3}$$

to form a nonholomic kinematic constraint of the system. Examples of such nonholonomic constraint equalities are the rolling of the wheels and the side wise motion of the WMR. Eq. (3.3) represents a velocity-level constraint of the system at the given configuration. The non-integrable nature of the constraint does not necessarily reduce the

number of generalized coordinates. Consequently, there arises possibility to steer the system using less number of inputs (Kurfess 2005, Melchiorri & Tornambe 1996).

Dynamic model of a general nonholonomic WMR

Let us consider a nonholonomic system whose vector of generalized coordinates, and vector of longitudinal and angular velocities are defined as $q \in \Re^{n\times 1}$ and $\dot{q} \in \Re^{n\times 1}$, respectively. If the system is subjected to a set of constraint forces given by Q_k, where $k = 1,2,\dots,n-m$, then there will be m nonholonomic constraint equations that must be explicitly satisfied by the system. We can write the constraint equations in *Pfaffian* form (Eq. (3.3)) as follows,

$$A(q)\dot{q} = 0,\ c \in \{1,2,\dots,m\} \tag{3.4}$$

where $A(q) = \frac{\partial f_c}{\partial q} \in \Re^{m\times n}$ is the *Jacobian* of Eq. (3.3) and is called the constraint matrix. It is full rank matrix everywhere. By using the Lagrange's method, when the WMR is subjected to the nonholonomic kinematic constraints of the form Eq. (3.4), the *Lagrangian* equation of motion can be written as follows,

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_j}\right) - \frac{\partial L}{\partial q_j} = u_j + A(q)^T\lambda_k,\ j \in \{1,2,\dots,n\}, k \in \{1,2,\dots,n-m\} \tag{3.5}$$

where, $L = T - P$ is the *Lagrangian* function. T is the total kinetic energy and P is the total potential energy of the system. u_j is the generalized force corresponding to the generalized coordinate q_j. λ_k is a vector of *Lagrange* multiplier that accounts for the constraint induced force $A(q)^T\lambda_k$. By solving the *Lagrangian*, we can formulate the general dynamic equation of the system as follows,

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} = B(q)u + A^T\lambda \tag{3.6}$$

where $M(q) \in \Re^{n\times n}$ is called the inertia matrix of the system. $C(q,\dot{q}) \in \Re^{n\times 1}$ is the centrifugal and Coriolis matrix. $B(q) \in \Re^{n\times(n-m)}$ and $u \in \Re^{(n-m)\times 1}$ are the input transformation matrix and input vector, respectively. $\lambda \in \Re^{m\times 1}$ is a vector of *Lagrange* multipliers and $A(q) \in \Re^{m\times n}$ is a constraint matrix that is *adjoined* to the dynamic equation. Several properties of the dynamic equation, Eq. (3.6) can be observed (Das & Kar 2006),

P1: The inertia matrix $M(q)$ is symmetric and positive definite.

P2: The matrix $\left(\dot{M}-2C\right)$ is skew symmetric resulting in the following characteristic, $q^T\left(\dot{M}-2C\right)q = 0$ for all $q \in \Re^n$

For the WMR shown in Fig. 3,1, the generalized system coordinates are given as,

$$q = [x_c, y_c, \varphi, \theta_1, \theta_2, \theta_3, \dots, \theta_{n-1}, \theta_n]^T \tag{3.7}$$

where $(x_c,\ y_c)$ is the coordinate of the reference point on the WMR platform, φ is the platform orientation with respect to an inertial frame $\{x_I, y_I\}$ and $\theta_i, i = 1,2,\dots,n$ are the wheel angular displacements.

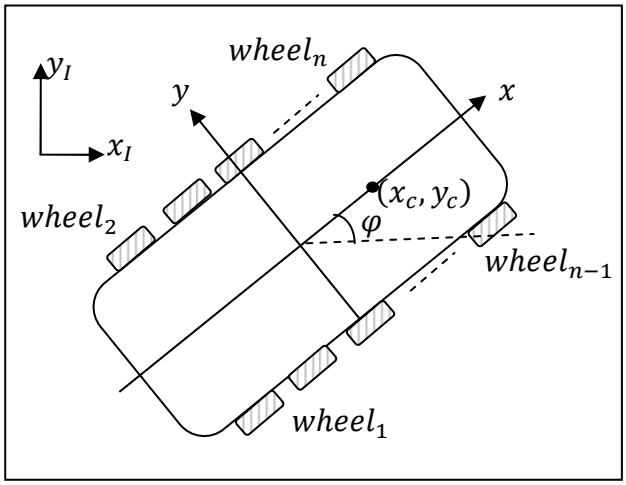


Fig. 3.1: Generalized nonholonomic WMR platform

In many conventional WMR systems, the constraint equations in Eq. (3.4) are defined under the assumption that the wheel rolls without longitudinal slip and there is no lateral slip. That means, the longitudinal velocity of the WMR is governed by the linear velocity of the wheel that is solely defined using the wheel angular velocity and there is no velocity along the lateral direction. Hence, if slip occurs, these assumptions are clearly violated.

In this dissertation, we argue that the modeling of the WMR that is based on the assumptions of wheel pure rolling and zero lateral slipping (for WMR with unicycle type of wheel) in a real practical situation is rather unrealistic. Our solution is to relax the assumptions by introducing new states due to wheel slip.

As shown in Fig. 3.2, we introduce ζ_i and η_i to represent the longitudinal slip displacement and the lateral slip displacement, respectively, for the *i*-th wheel of the WMR. The wheels are rigidly connected to the WMR body as shown in Fig. 3.1.

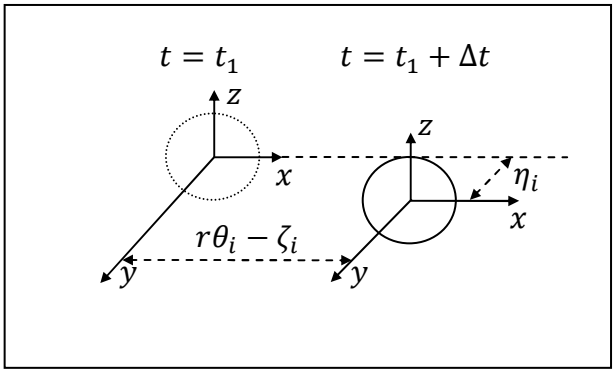


Fig. 3.2: System motion contributed by wheel's rolling and both longitudinal and lateral slips on a planar surface

We can write a new state, ρ_i to represent the total longitudinal displacement of the *i*-th wheel as,

$$\rho_i = r\theta_i - \zeta_i \tag{3.8}$$

The new coordinate system allows us to describe the motion of the system in the presence of slip. We then define $S(q) \in \Re^{n\times(n-m)}$ to be a full rank matrix formed by a set of smooth and linearly independent vector fields, spanning the null space of $A(q)$. Thus, the result of multiplication of these matrices can be written as follows,

$$S(q)^T A(q)^T = 0 \tag{3.9}$$

It is then possible to find a set of vector of time functions, that is, for all t,

$$\nu(t) \in \Re^{(n-m)\times 1} \tag{3.10}$$

so as,

$$\dot{q}(t) = S(q)\nu(t) \tag{3.11}$$

We can use matrix $S(q)$ from Eq. (3.9) to eliminate the *Lagrange* multipliers in general dynamic equation, Eq. (3.6). We further differentiate Eq. (3.11), to get the state acceleration as follows,

$$\ddot{q}(t) = \dot{S}(q)\nu(t) + S(q)\dot{\nu}(t) \tag{3.12}$$

By placing Eq. (3.11) and Eq. (3.12) into the dynamic equation, Eq. (3.6), we obtain,

$$S^T MS\dot{\nu} + S^T M\dot{S}\nu + S^T CS\nu = S^T Bu \tag{3.13}$$

Eq. (3.13) describes the dynamics of the nonholonomic WMR system in a new set of local coordinates ν, such that matrix $S(q)$ projects the velocities, ν in the WMR base coordinate to velocities in Cartesian coordinate, $\dot{q}$. Therefore the properties of the original dynamics hold for the new set of coordinates that is,

P3: The matrix $(S^T MS)$ is symmetric and positive definite

P4: The matrix$(S^T\dot{M}S) - 2(S^TM\dot{S})$ is skew-symmetric.

We can rearrange Eq. (3.13) to form,

$$\dot{v}(t) = (S^TMS)^{-1}S^T\left(-M\dot{S}v - Cv - Bu\right) \tag{3.14}$$

$\dot{v}(t)$ is the acceleration of vector time function defined in Eq. (3.10). Based on the equation Eq. (3.14), we can develop a suitable model-based controller for the WMR system.

Detailed modeling of a two-wheeled nonholonomic mobile robot

Now we develop a detailed model for a two-wheeled (i.e., two driving wheels) WMR that is one of the most commonly available WMRs using the above generalized formulation.

Ideal model: A WMR without wheel slip

The WMR shown in Fig. 3.3 is a standard platform of a nonholonomic two-wheeled mobile robot.

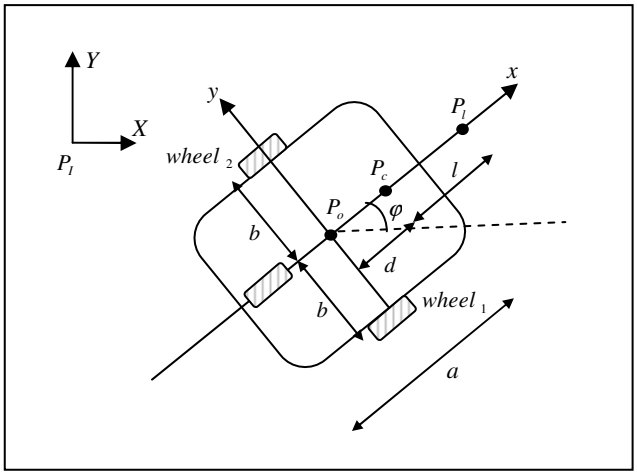


Fig. 3.3: A two-wheeled nonholonomic mobile robot platform

It has two differential-driving wheels and a castor wheel used for balancing. The differential driving wheels are from unicycle type as shown in Fig. 3.4. It has two degrees of freedom; the rotation around the motorized wheel axle and the contact point.

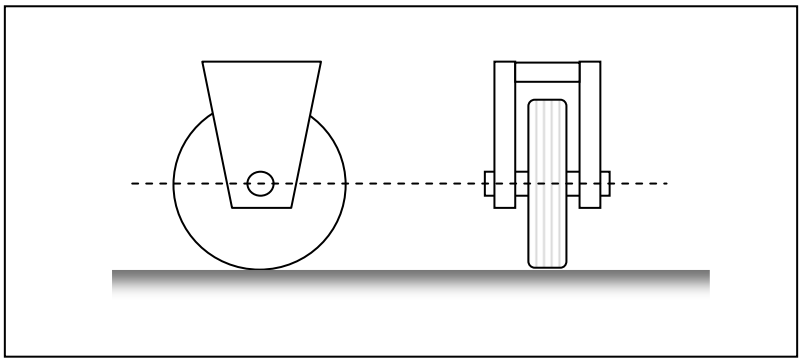


Fig. 3.4: A standard unicycle wheel

While the castor wheel as shown in Fig. 3.5 is located at the back of the WMR (i.e. can be located anywhere) and has three degrees of freedom which are the rotation around the wheel axle, the contact point and the castor wheel. In our wheel modeling, we assume all the wheels are deformable and rigidly held to the WMR platform.

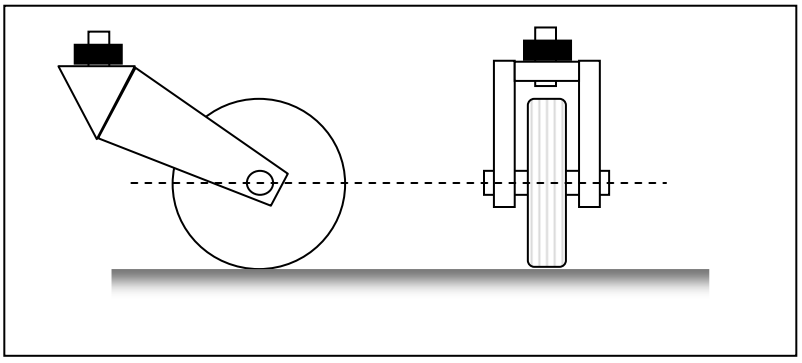


Fig. 3.5: A standard castor wheel

The two-driving wheels are powered by DC motors and have the same wheel radius, *r*. Point P_o is the origin of WMR axis, which is located at the intersection of the longitudinal *x*-axis and the lateral *y*-axis. The center of mass (COM) is located at point P_c. This point cannot be located at the intersection point of axis of symmetry of the platform, P_o to ensure no singularity in the control solution (Sarkar 1992, R. Zulli et. al. 1995). P_l is defined as a look-ahead point located on the *x*-axis of the WMR platform. *b* is the distance measured from the center of the WMR to the center of the wheel along the *y*-axis of the WMR reference frame. Here we assume the wheel model is represented by a thin, solid disk having a single point contact with the terrain surface. *d* denotes the distance between point P_o and point P_c along the *x*-axis. The distance of the look-ahead point is *l* from point P_c, which is also along the *x*-axis. The origin of the inertial frame $\{X,Y\}$ is shown as P_i and as such allows the pose of the WMR to be completely specified through the following vector of generalized coordinates,

$$q = [x_c, y_c, \varphi, \theta_1, \theta_2]^T \tag{3.15}$$

where x_c and y_c are the coordinate of the COM. φ represents the orientation of the WMR frame from the inertial frame and $[\theta_1, \theta_2]$ is the angular displacement vector for the WMR driving $wheel_1$ and $wheel_2$, respectively. Due to the nonholonomy of the system and by following the ideal no-slip assumption, the rolling constraints for both wheels are written as,

$$r\dot{\theta}_1 = \dot{x}_c \cos\varphi + \dot{y}_c \sin\varphi + b\dot{\varphi}$$

$$r\dot{\theta}_2 = \dot{x}_c \cos\varphi + \dot{y}_c \sin\varphi - b\dot{\varphi} \tag{3.16}$$

Eqn. (3.16) describes the longitudinal velocity of the WMR's center of mass that is constrained by the longitudinal velocity of the wheels generated due to pure rotation. By using the same premise, the knife-edge constraint can be written as follows,

$$0 = \dot{y}_c \cos\varphi - x_c \sin\varphi - d\dot{\varphi} \tag{3.17}$$

where the lateral velocity measured along the turning axis of the WMR is constrained to zero velocity. In order to derive the dynamic equation of the system using *Lagrangian* formalism, the WMR platform can be partitioned into three parts namely the body of the platform and its two wheels (i.e., $wheel_1, wheel_2$). The expression of the kinetic energy of the WMR body is given as,

$$T_r = \frac{1}{2} m_r (\dot{x}^2 + \dot{y}^2) + \frac{1}{2} I_{rz} \dot{\varphi}^2 \tag{3.18}$$

and the expression of the kinetic energy for both $wheel_1$ and $wheel_2$ are given respectively as,

$$T_{w1} = \frac{1}{2} m_w (\dot{x} + b\dot{\varphi}\cos\varphi + d\,\dot{\varphi}\sin\varphi)^2 + \frac{1}{2} m_w (\dot{y} + b\dot{\varphi}\sin\varphi - d\,\dot{\varphi}\cos\varphi)^2 + \frac{1}{2} I_{wz} \dot{\varphi}^2 + \frac{1}{2} I_{wy} \dot{\theta}_1^{\,2} \tag{3.19}$$

$$T_{w2} = \frac{1}{2} m_w (\dot{x} - b\dot{\varphi}\cos\varphi + d\,\dot{\varphi}\sin\varphi)^2 + \frac{1}{2} m_w (\dot{y} - b\dot{\varphi}\sin\varphi - d\,\dot{\varphi}\cos\varphi)^2 + \frac{1}{2} I_{wz} \dot{\varphi}^2 + \frac{1}{2} I_{wy} \dot{\theta}_2^{\,2} \tag{3.20}$$

By using the constraint equations (Eq. 3.16 and 3.17) and energy equations (Eq. 3.18-3.20), we can develop the dynamic equation for the WMR system without wheel slip as in the form of Eq. (3.6) or Eq. (3.14). The details of the derivation can be found in (Sarkar, Xiaoping & Kumar 1993).

In the next paragraph, we discuss the focus of this research, where the WMR is now subjected to wheel slip. For such a condition we basically relax the constraint equations and develop a new dynamic model of the WMR.

Non-ideal case: A WMR with wheel slips

In this research we want to investigate the navigation problem of a nonholonomic WMR when the ideal no-slip assumption does not hold true. In order to model that condition, we need to include slip into the dynamics of the system. We start by introducing the new set of generalized coordinates vector after no-slip condition is relaxed as follows,

$$q = [x_c, y_c, \varphi, \eta_1, \eta_2, \rho_1, \rho_2, \theta_1, \theta_2]^T \tag{3.21}$$

Note that, the slip-contributed states (i.e., ρ_i and η_i) can be easily expanded to accommodate a WMR with more fixed wheels. Using the new generalized coordinate vector, we can formulate the rolling constraints of the WMR with two fixed driving wheels in the following form,

$$\dot{\rho}_1 = \dot{x}_c \cos\varphi + \dot{y}_c \sin\varphi + b\dot{\varphi} \tag{3.22}$$

$$\dot{\rho}_2 = \dot{x}_c \cos\varphi + \dot{y}_c \sin\varphi - b\dot{\varphi} \tag{3.23}$$

where $\dot{\rho}_i = r\dot{\theta}_i - \dot{\zeta}_i$. The constraint equations, Eq. (3.22) and Eq. (3.23) relaxes the assumption of no-slip by allowing the longitudinal velocity at the wheel hub to be the summation of the longitudinal velocity generated by the wheel angular velocity, and the longitudinal slip velocity.

The same basis can be applied to develop the knife-edge/lateral constraints. Note that lateral slip in each wheel of a WMR is independent if the wheels are connected to the body of the WMR with mechanisms that allow relative motion (e.g., connected using springs and dampers). However, in our case as shown in Fig. 3.3, the two wheels of the WMR are rigidly connected to the body of the WMR and thus cannot have two different

lateral slips as can be seen from the following equations where both $\dot{\eta}_1$ and $\dot{\eta}_2$ have the same expressions,

$$\dot{\eta}_1 = \dot{y}_c \cos\varphi - x_c \sin\varphi - d\dot{\varphi} \tag{3.24}$$

$$\dot{\eta}_2 = \dot{y}_c \cos\varphi - x_c \sin\varphi - d\dot{\varphi} \tag{3.25}$$

where lateral slip is allowed to occur along the turning axis of the WMR during cornering. In this research, we analyze the effect of both slips, particularly, to investigate the agility of WMR navigation to negotiate sharp cornering. Now, after we rearrange the coordinate system, the new constraints defined above can be rewritten in the form of Eq. (3.4), where $A(q) \in \Re^{4\times 9}$ is given as follows,

$$A(q) = \begin{bmatrix} \cos\varphi & \sin\varphi & b & 0 & 0 & -1 & 0 & 0 & 0 \\ \cos\varphi & \sin\varphi & -b & 0 & 0 & 0 & -1 & 0 & 0 \\ -\sin\varphi & \cos\varphi & -d & -1 & 0 & 0 & 0 & 0 & 0 \\ -\sin\varphi & \cos\varphi & -d & 0 & -1 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{3.26}$$

which is a full rank matrix. We can find matrix $S(q) \in \Re^{9\times 5}$ from $\mathcal{N}(A(q))$ to fulfill the requirement of Eq.(3.9) as,

$$S(q) = \begin{bmatrix} -\sin\varphi & \frac{(b\cos\varphi - d\sin\varphi)}{2b} & \frac{(b\cos\varphi + d\sin + d\sin\varphi)}{2b} & 0 & 0 \\ \cos\varphi & \frac{(d\cos\varphi + b\sin\varphi)}{2b} & \frac{(-d\cos\varphi + b\sin\varphi)}{2b} & 0 & 0 \\ 0 & \frac{1}{2b} & -\frac{1}{2b} & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.27}$$

In order to formulate the inertia matrix, we define the kinetic energy of the WMR as follows,

$$T_r = \frac{1}{2} m_r (\dot{x}^2 + \dot{y}^2) + \frac{1}{2} I_{rz} \dot{\varphi}^2 \tag{3.28}$$

$$T_{w1} = \frac{1}{2} m_w \left(\dot{\rho}_1{}^2\right) + \frac{1}{2} m_w \left(\dot{\eta}_1{}^2\right) + \frac{1}{2} I_{wz} \dot{\varphi}^2 + \frac{1}{2} I_{wy} \dot{\theta}_1{}^2 \tag{3.29}$$

$$T_{w2} = \frac{1}{2} m_w \left(\dot{\rho}_2{}^2\right) + \frac{1}{2} m_w \left(\dot{\eta}_2{}^2\right) + \frac{1}{2} I_{wz} \dot{\varphi}^2 + \frac{1}{2} I_{wy} \dot{\theta}_2{}^2 \tag{3.30}$$

where T_r is the kinetic energy of the WMR body and, T_{w1} and T_{w2} are the kinetic energies for $wheel_1$ and $wheel_2$, respectively. We found the inertia matrix, $M(q) \in \Re^{9\times 9}$ to be,

$$M = \begin{bmatrix} m_r & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & m_r & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & I_{rz} + 2I_{wz} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & m_w & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & m_w & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & m_w & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & m_w & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & I_{wy} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & I_{wy} \end{bmatrix} \tag{3.31}$$

which is positive definite and symmetric. We later introduce a vector of lateral traction force, f_{lat_i} and longitudinal traction force, f_{lon_i} as

$$F(\dot{q}) = [0,0,0, f_{lat1}, f_{lat2}, f_{lon1}, f_{lon2}, -rf_{lon1}, -rf_{lon2}]^T \tag{3.32}$$

where each individual element of the traction force vector is calculated from the magnitude of the respective slips. The dynamic equation of WMR system can now be represented as,

$$M(q)\ddot{q} = B(q)\tau + F(\dot{q}) + A^T\lambda \tag{3.33}$$

where the transformation matrix, $B(q)$ takes the form of,

$$B(q) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}^T \tag{3.34}$$

The input τ is the torques to the driving wheels, given in the vector form,

$$\tau = \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} \tag{3.35}$$

and the *Lagrange* multiplier,

$$\lambda = [\lambda_1 \quad \lambda_2 \quad \lambda_3 \quad \lambda_4]^T \tag{3.36}$$

We expand the dynamic equation, Eq. (3.33), to obtain the following set of equations,

$$m_r\ddot{x} = \cos\varphi\,(\lambda_1 + \lambda_2) - \sin\varphi\,(\lambda_3 + \lambda_4) \tag{3.37}$$

$$m_r\ddot{y} = \sin\varphi\,(\lambda_1 + \lambda_2) + \cos\varphi\,(\lambda_3 + \lambda_4) \tag{3.38}$$

$$(I_{rz} + 2I_{wz})\ddot{\varphi} = \mathrm{b}(\lambda_1 - \lambda_2) - \mathrm{d}(\lambda_3 + \lambda_4) \tag{3.39}$$

$$m_w\ddot{\eta}_1 = f_{lat1} - \lambda_3 \tag{3.40}$$

$$m_w\ddot{\eta}_2 = f_{lat2} - \lambda_4 \tag{3.41}$$

$$(I_{wy} + m_w r^2)\ddot{\theta}_1 = \tau_1 - rf_{lon1} \tag{3.44}$$

$$(I_{wy} + m_w r^2)\ddot{\theta}_2 = \tau_2 - rf_{lon2} \tag{3.45}$$

From these set of equations, we note that the last two equations (Eq. (3.44) and Eq. (3.45)) are independent of *Lagrange* multipliers, which are due to the constraint equations. This allow us to separate these two equations from the dynamic equation, Eq.(3.33), and eliminate the last two column of matrix $S(q)$, in Eq.(3.27). By following the procedure described in Eq. 3.4-3.14, we present the dynamics of the WMR in the following form,

$$\dot{v} = \begin{bmatrix} \ddot{\eta}_2 \\ \ddot{\rho}_1 \\ \ddot{\rho}_2 \end{bmatrix} = (S^T M S)^{-1} S^T\left(-M\dot{S}v\right) + (S^T M S)^{-1} S^T F = \varsigma + \kappa F \tag{3.46}$$

$$I_{wy}\ddot{\theta}_1 = \tau_1 - rf_{lon2} \tag{3.47}$$

$$I_{wy}\ddot{\theta}_2 = \tau_2 - rf_{lon2} \tag{3.48}$$

Derivation of the dynamic equation using Newton's method

Newton's method is another main formalism to derive the governing equation pertaining to the dynamics of a mechanical system. The *Newton* equations relate forces and torques to the linear and rotational accelerations of the body masses. In the following, we use Newton's method to rederive the dynamic equation of WMR to ensure the consistency of the model previously developed using Lagrange's method.

Using *free-body* approach, we isolate the WMR into three parts namely, the body of the WMR and its two wheels (i.e.,$wheel_1, wheel_2$), and account for all the forces and torques acting on the joints of the parts. Fig. 3.6 depicts the acting forces and torques on the parts using free body diagram.

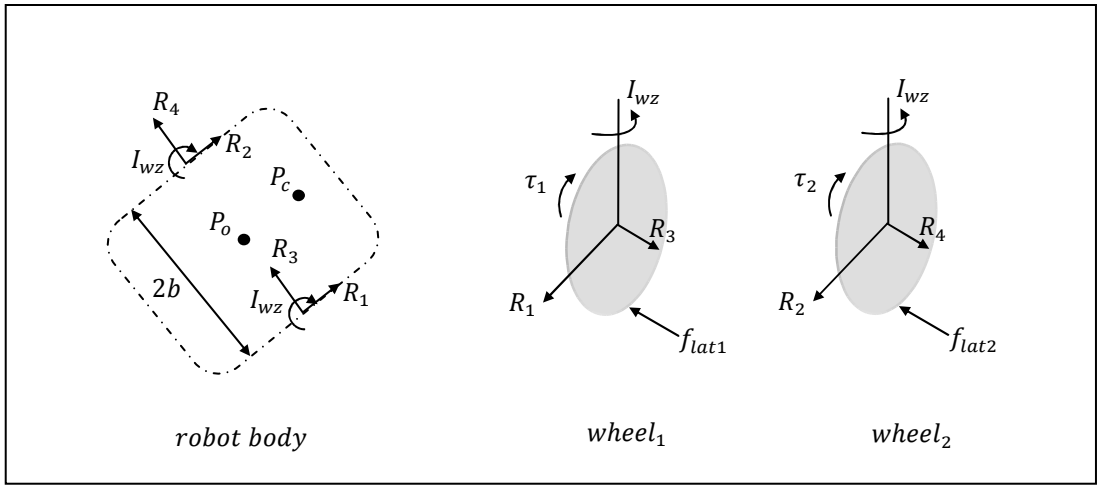


Fig. 3.6: Free body diagram of a wheeled-mobile robot

By letting R_i to be the reactive forces acting on the respective joints, the Newton equations for the body of the WMR are given as follows,

$$m_r\ddot{x} = \cos\varphi\,(R_1 + R_2) - \sin\varphi\,(R_3 + R_4) \tag{3.49}$$

$$m_r\ddot{y} = \sin\varphi\,(R_1 + R_2) + \cos\varphi\,(R_3 + R_4) \tag{3.50}$$

$$I_{rz}\ddot{\varphi} = \mathrm{b}(R_1 - R_2) - \mathrm{d}(R_3 + R_4) - 2I_{wz}\ddot{\varphi} \tag{3.51}$$

and for $wheel_1$ and $wheel_2$, the equations can be written as,

$$m_w\ddot{\eta}_1 = f_{lat1} - R_3 \tag{3.52}$$

$$m_w\ddot{\eta}_2 = f_{lat2} - R_4 \tag{3.53}$$

$$(I_{wy} + m_w r^2)\ddot{\theta}_1 = \tau_1 - rR_1 \tag{3.56}$$

$$(I_{wy} + m_w r^2)\ddot{\theta}_2 = \tau_2 - rR_2 \tag{3.57}$$

We observe the set of equations (Eq. 3.49-3.57) derived using Newton's method have the same form as the equations derived using Lagrange's method (Eq. 3.37-3.45) if we let the relationship between Lagrange's multipliers and the reactive forces to be,

$$R_i = \lambda_i \tag{3.58}$$

This shows the consistency between the two models developed using Newton's and Lagrange's methods. In the following chapter, we discuss the model of traction force and its relationship to wheel slips.

CHAPTER IV

WHEEL SLIP AND TRACTION FORCE

Increasing requirements on ride safety and comfort of vehicles inspired many research areas such as advanced vehicle control system, online system monitoring and wheel-surface (wheel-ground surface) traction force modeling. The latter involves the study on the mechanism to convert motor torque to traction force. The analysis of wheel-surface interaction can provide an insight into the understanding of vehicle dynamics so as to improve ride and trajectory performance. This is particularly important for an autonomous system like the WMR, where the control performance based on the model-based controller depends heavily on its dynamic model.

The main task of wheel-surface traction force modeling and monitoring is to determine the relationship between traction forces and slip velocity. However, the relationship is difficult to analyze due to the following three challenges. First, traction force is generated through tire deformation, tire adhesion as well as tire wear and tear, and it is influenced by several factors including wheel-surface conditions, tire pressure, and load etc. When there is a continuous interaction between the tire and the surface, the result is the elastic deformation at a molecular level of the outer layer of the tires called asperities shown in Fig. (4.1). The load of the WMR causes these asperities to penetrate the surface asperities. When this happens, it yields a resistance force or a traction force. Deformation of the tire provides most of the traction force. Adhesion is a property of the rubber that causes it to stick to other materials at the contact point. It is caused by the

molecular bonds between the rubber thread and the surface. The strength of the bonds relies on the temperature and the amount of slip occurs at the contact point. In addition to tire deformation and tire adhesion, the wear and tear of the wheel also contributes to the traction force.

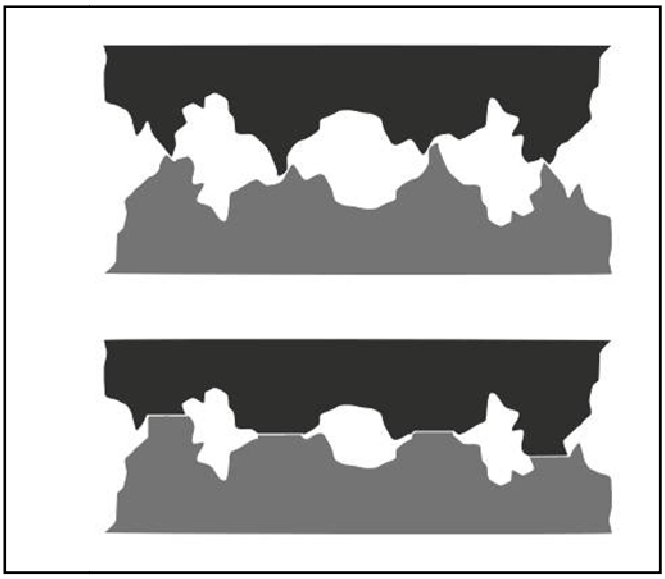


Fig. 4.1: Asperities deformation of two surfaces before load is applied (top) and after load is applied (bottom)

Second, the nonlinear and dynamic properties of traction force such as the viscous and hysteresis phenomenon are difficult to describe analytically. On the other hand, most empirical traction force models are hard to rationalize by physical laws.

Third, the complexity of the traction force model can affect the performance of the systems. For instance, for a system application, which is time critical, the model of the traction force should be able to be employed in real-time.

In this chapter, we describe the choice of traction force model used in this research. Our requirement is to have a traction model that is made up of a continuous, differentiable function.

Traction force models

The model of traction force can generally be classified into two types namely empirical (or semi-empirical) and analytical models. The former are developed based on curve-fitting techniques and can accurately capture the nonlinear characteristics of the traction force. However, most of these models lack physical interpretation, and cannot directly reflect the effects of some dynamic factors like tire hysteresis, humidity of the surface and tire pressure. Meanwhile, most analytical model of traction force is composed of differential equation. For example, *Brush* model can model the dynamic factors mentioned above (Li & Wang 2006). However, these models lack the empirical (semi-empirical) accuracy and as a result, the choice of the traction models greatly depends on the type of system applications. For a WMR that is equipped with standard unicycle wheels as described in Chapter III, the traction force acting on the longitudinal and lateral directions can be modeled separately, resulting in two independent traction force models, longitudinal and lateral traction models. Table 4.1 lists some of the most useful traction models found in the literatures.

Table 4.1: Traction force models and their brief descriptions

MODEL NAME	PROPERTIES	FEATURES
Piecewise Linear (longitudinal)	Empirical	Easy to identify but cannot accurately fit curves
Burckhardt model (longitudinal)	Semi-empirical	Can accurately fit curve
Rill model (longitudinal)	Semi-empirical	Easy to identify

Delft model or Magic formula (longitudinal and lateral)	Semi-empirical	Can accurately fit curves and model different factors
Dahl model (longitudinal)	Analytical	Can describe Coulomb friction and produce smooth transition around zero velocity
Bliman-Sorine (longitudinal)	Analytical	An improvement over Dahl's with an additional Stribeck effect
LuGre model (longitudinal)	Analytical	An improvement over Bliman-Sorine's with additional combination of pre-sliding and sliding factors
Linear Proportional (lateral)	Empirical	Cannot accurately reflect saturation properties but easy to identify
Nonlinear Proportional (lateral)	Semi-empirical	Can accurately fir curves
Bicycle (lateral)	Analytical	Does not reflect the traction force directly

The *Delft* model or famously known as *Magic formula* model is an elegant, semi-empirical model based on curve fitting. It has been widely accepted in the industry and academia (Politis et al. 2001, Li and Wang 2006) to generalize the model of both longitudinal and lateral traction forces. It was introduced by (Bakker, Nyborg & Pacejka 1987) and since then has been revised several times. The advantages of this model over the others stem from its accuracy, simplicity and ability to be interpreted. Due to these reasons, in this research we employ the *Magic formula* model of traction force. Moreover the model is composed of a continuous, differentiable function.

Slip (wheel slip) is a major component in the *Magic formula* model of traction force. It is an indirect measure of the fraction of the contact point that is sliding when the

velocity of the tire with respect to the surface at the contact patch is nonzero. In this dissertation, we define longitudinal slip as a slip that happened along the mean plane of the moving wheel and side slip as a slip that happens in lateral direction of a moving wheel. In the following we describe both types of slips under the framework of longitudinal and lateral traction force, respectively.

Longitudinal slip and longitudinal traction force

Longitudinal slip, $\dot{\zeta}$, also known as circumferential or tangential slip, happens along the mean plane of the wheel and is responsible for the generation of the longitudinal traction force. It is defined as a difference between the resultant linear velocity due to the angular velocity of the wheel, $r\dot{\theta}$, and the instantaneous velocity of the hub centerline of the wheel, V_{wx} with respect to the ground during acceleration and deceleration phases of motion. The term slip ratio, *sr*, is generally used to represent longitudinal slip and is defined as follows,

$$sr_i = \frac{r\dot{\theta}_i - V_{wx\_i}}{max(r\dot{\theta}_i, V_{wx\_i})} = \frac{\dot{\xi}_i}{max(r\dot{\theta}_i, V_{wx\_i})} \tag{4.1}$$

where i denotes the i-th wheel. From Eq. (4.1), a free rolling wheel, where $V_{wx} = 0$ has the slip ratio, $sr = 1$ and for the wheel that rolls without slip, $V_{wx} = r\dot{\theta}$, the slip ratio, $sr = 0$. In this research we assume the wheel is rolling with slip. Thus, $V_{wx} = r\dot{\theta} - \dot{\xi}$ is an expression to represent the total longitudinal velocity of the WMR at the wheel hub centerline. In this research, the relationship between the longitudinal slip and the longitudinal traction force, f_{lon} in the *Magic formula* model of traction is presented as,

$$f_{lon} = Dsin\left(a_0 tan^{-1}\left(SB + E(tan^{-1}(SB) - SB)\right)\right) \tag{4.2}$$

where all the variables are given and described below,

$$\mu_{xp} = a_1 F_z + a_2 \tag{4.3}$$

$$D = \mu_{xp} F_z \tag{4.4}$$

$$Ba_0 D = (a_3 F_z^2 + a_4 F_z) e^{-a5Fz} \tag{4.5}$$

$$E = a_6 F_z^2 + a_7 F_z + a_8 \tag{4.6}$$

$$S = 100sr + a_9 F_z + a_{10} \tag{4.7}$$

F_z is the weight of the WMR in kN and the eleven empirical numbers, a_i , $i \in \{1,2,\dots,11\}$, are used to characterize a particular tire. Eq. (4.3) is a linear function of weight that estimates the peak of longitudinal friction coefficient. In Eq. (4.4), we see D as a linear function of F_z, where μ_{xp} can be treated as a regular coefficient of traction function and a_2 in Eq. (4.3) can be seen as a direct measurement of the degree of tire stickiness. B in Eq. (4.5) scales the other independent variable, $a_0 D$ and is known as a stiffness factor. Eq. (4.6) represents E as a factor known as the curvature factor. It determines the shape around the peak of the curve. The last variable, S in Eq. (4.7) is a function of slip ratio, sr which can be measured directly from the system. Fig. 4.2 shows some examples of longitudinal traction force or longitudinal traction curve for rubber wheel for different type of surfaces. A region on the left of the peak force starts linearly and is known as a stable region. Increasing the slip ratio passing the peak force will significantly decreases the traction force and thus the whole region on the right of the peak force should be avoided as it represents instability.

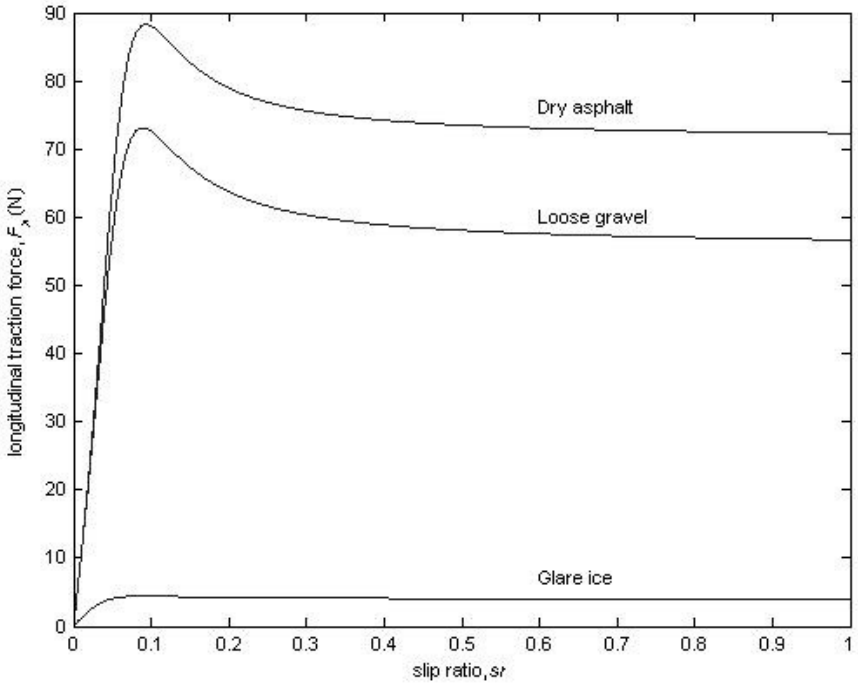


Fig. 4.2: Some examples of traction curves for a variety of surface types

Lateral slip and lateral traction force

Lateral traction force, f_{lat}, is generated as a result of slip angle, *sa*, during wheel cornering. It is sometimes known as cornering force. Slip angle is defined as the angle between the instantaneous velocity of the WMR and the instantaneous linear velocity of the wheel. In this research this term is defined as follows,

$$sa_i = tan^{-1}\left(\frac{\dot{\eta}_i}{\dot{\rho}_i}\right) \tag{4.8}$$

where $\dot{\eta}_i$ is the lateral slip of the *i*-th wheel and $\rho_i = r\theta_i - \zeta_i$ is total longitudinal displacement of the *i*-th wheel. The *Magic formula* model to define lateral force is almost identical to its longitudinal counterpart but requires significantly different interpretation.

Besides *sa*, there are fifteen more empirical numbers, b_i , $i \in \{1,2,\dots,15\}$ that help to construct the lateral traction model, which can be written as follows,

$$F_z = Dsin\left(b_0 tan^{-1}\left(SB + E(tan^{-1}(SB) - SB)\right)\right) + s_v \tag{4.9}$$

where the variables and their brief descriptions are given below.

$$\mu_{yp} = b_1 F_z + b_2 \tag{4.10}$$

$$D = \mu_{yp} F_z \tag{4.11}$$

$$Ba_0 D = b_3 sin\left(2tan^{-1}(F_z/b_4)\right)(1 - b_5|\gamma|) \tag{4.12}$$

$$S = sa + b_8\gamma + b_9 F_z + b_{10} \tag{4.13}$$

$$E = b_6 F_z + b_7 \tag{4.14}$$

$$S_v = \left((b_{11}F_z + b_{12})\gamma + b_{13}\right)F_z + b_{14} \tag{4.15}$$

Eq. (4.10) defines the peak of lateral friction coefficient. The variable, *D*, in Eq. (4.11) defines the product of the friction coefficient and the normal force, F_z. In general, the term b_2 in the equation must be significantly larger than $b_1 F_z$ to maintain the Newtonian behavior of the traction force. The variable *B* given in Eq. (4.12) has different interpretation where γ is the camber angle of the wheel measured in degree. The fourth variable in Eq. (4.13) and the last variable in Eq. (4.14) can be measured in a straight forward manner. In addition to the above-mentioned variables, the lateral traction form of *Magic formula* requires an additive correction term for *ply steer* and *conicity*, (Eq. (4.15)). Fig. (4.3) shows a plot of lateral traction force or also know as traction curve for different surfaces with different friction coefficient (Pasterkamp 1997).

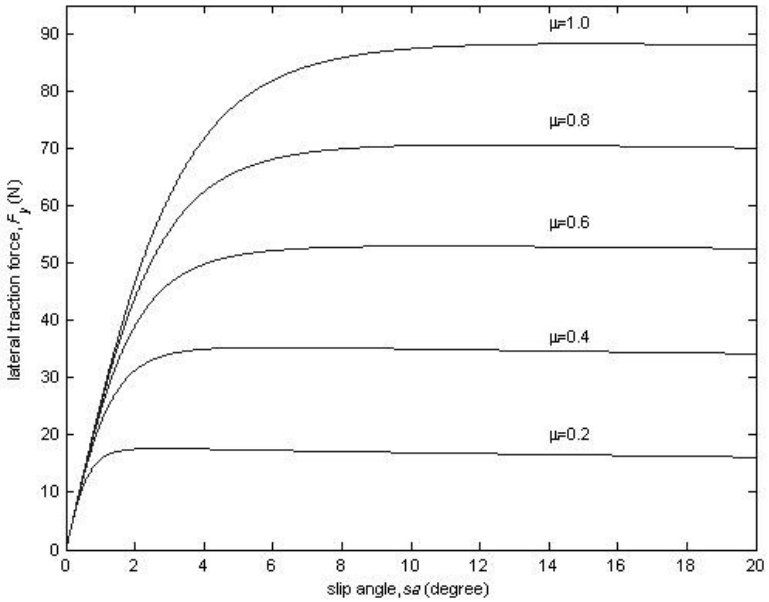


Fig. 4.3: Some examples of lateral traction curves for a variety of surfaces with different friction coefficients

CHAPTER V

WMR MODEL VERIFICATION

Designing new planners and controllers for the WMR through experimentation can be hazardous as well as costly in terms of time and resources. Realistic simulation can be an attractive alternative to the real experiments. If we are able to develop a realistic WMR model, predictions regarding the output of the real experiments could be made from the simulation study. It allows systematic analysis of the WMR dynamic behavior and provides fast and flexible development of new planners and controllers for the WMR. (Nehmzow 2003) quotes the computer simulation as,

"model which is amenable to manipulations which would be impossible, too expensive or impractical to perform on the entity it portrays. The operation of the model can be studied and, from it, properties concerning the behavior of the actual system or its subsystems can be inferred."

The basic requirement of a realistic WMR simulation implies the existence of a reliable model of the system. The majority of the on-shelf wheeled mobile robot platforms available, such as *Robulab*, *Roburoc* (URL 5.1), *Trilobot Research Robot* (URL 5.2), and *Pioneer*, *AmigoBot* and *PowerBot* (URL 5.3) come with their own simulator software (i.e. *MobileSim for Pioneer robot*). Additionally, a number of more general purpose WMR simulators have also been developed based on the open-source platform, which allows wider access to multiple WMR platforms such as *Stage* and *OpenSim*. While majority of the WMR simulators have become quite useful for general WMR

applications, they do not model wheel slip and thus may not be able to describe the robot behavior correctly and effectively where wheel slip is a critical factor (URL 1.10). At low speed, these WMR models may be valid but when the slip is significant the navigation and control algorithms develop based on these models may result in undesirable performance.

In this chapter, we present the verification of the WMR model developed in Chapter III through a series of experimental studies. In particular, we want to investigate the dynamics of the WMR while taking a sharp turn at high speed and when asked to move along a straight line on slippery surface.

WMR motion task: sharp cornering at high speed

One of the reasons to have high speed navigation for a WMR is to achieve service efficiencies. However, there are fundamental difficulties when we want to increase the speed of a WMR. (**Chung, Kim, & Choi. May 2006**) classified the difficulties into three categories: i) unexpected dynamic changes of the environment likes the abrupt appearance of obstacles; ii) the control and computational limitations due to the system response for real time applications; and iii) the dynamic and mechanical limitations.

In this chapter, we first discuss the dynamics of the WMR during sharp cornering at high speed when the lateral velocities are generated and the wheel side slip, $\dot{\eta}$, could become large enough to impact the overall performance of the system (Travis, Bevly 2005). Here the term high speed is a relative concept and is defined with respect to the surface on which the WMR is traversing. For example, for the WMR under study we

define high speed to be 0.8*m/s* when the WMR is traversing on the slippery surface (it may not be the case if the surface has higher traction such as a dry pavement).

Simulation parameters

In the model simulation, our objective is to observe the dynamics of the WMR that is asked to follow an *L*-shape path (i.e., a sharp corner) in an open-loop manner with torque as the input to both wheels. The simulated model is based on *Pioneer P3DX* (two wheeled mobile robot) manufactured by *MobileRobot Inc.* shown in Fig. 5.1 and can be represented schematically as in Fig. 3.3.

Fig. 5.1: *Pioneer P3DX*, the two wheeled mobile robot platform

We employ Eq. (3.33) to model the dynamics of the *Pioneer PD3X* WMR and set the WMR parameters (refer Fig. 3.3) as follows: $b = 0.24m$; $d = 0.05m$; $r = 0.095$; $m_r = 17kg$; $m_w = 0.5kg$; $I_{rz} = 0.537kgm^2$; $I_{wy} = 0.0023kgm^2$; $I_{rz} = 0.0011kgm^2$. The respective moment of inertia values are obtained by assuming the WMR body to take a solid cuboid shape of *height* x *width* x *length* = 0.245*m* x 0.4*m* x 0.45*m* and each wheel to be of the form of a thin, solid disk of radius, *r* and mass, m_w.

Experimental setup

The goal of the experiment is to replicate the results obtained from the simulation studies as best as possible in order to verify the WMR analytical model developed for this research. For a particular *Pioneer P3DX* WMR platform, the lack of lateral velocity sensing unit requires us to select a proper sensor to measure the quantity. We opt for an accelerometer, *MDS302,* manufactured by *Mechworks System Inc.* as shown in Fig. 5.2.

Fig. 5.2: Accelerometer, *MDS302*

The accelerometer can read up to ±2*g* acceleration, which is suitable for our WMR application. We found that on a planar surface, the choice of using accelerometer is sufficient to measure the lateral slip in order to validate our dynamic model. We also realize that, by using direct integration method to find lateral slip measurement from accelerometer signal is prone to 'drift' problem. However, with proper adjustment of the offset value we are able to minimize this problem. The availability of an extra serial port on the *Pioneer P3DX* allows this external accelerometer to be directly tethered and positioned on the system as shown in Fig. 5.1.

We program the accelerometer to run along the program of the *Pioneer P3DX* in synchronous mode where several tasks are done in multithread with proper prioritization.

(the sample program can be found in the Appendix). The data from each task are updated for every 100*ms* of robot command cycle. As a result, the processing of accelerometer signal (i.e., filtering, data conversion) into velocity during each command cycle limits the sampling rate of the accelerometer data to 10*samples/s* from each axis. The faster sampling rate used, causes delay and overflow in the system buffer and the slower sampling rate, lessens the signal resolution. We reduce the effect of noise by applying a built-in 5^{th} order Butterworth filter with cut off frequency of 50*Hz*.

Sharp cornering motion through open loop control: results and discussion

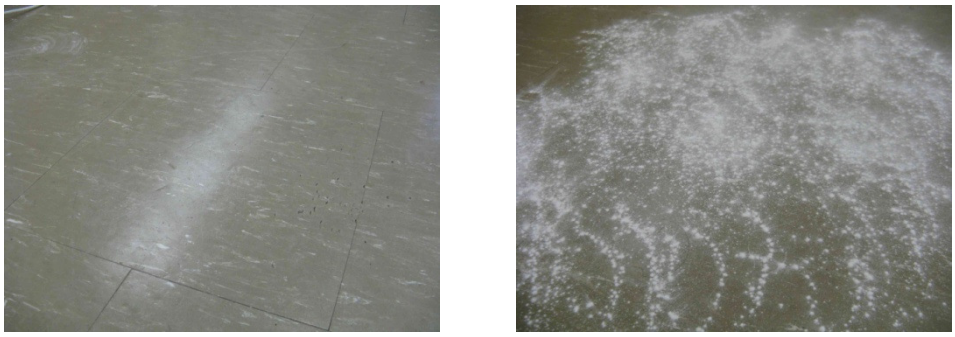
We run the simulation studies and the experiments to investigate the *Pioneer P3DX* dynamics using the open loop control. The idea to apply the open loop control is to provide a common platform for the simulated dynamic model of the *Pioneer P3DX* to be compared to the real *Pioneer P3DX* and to isolate any input that exists in the feedforward/feedback control approach. With regard to this point, we consider the closed loop response of the low-level built-in PID controller in the *Pioneer P3DX* to have insignificant impact especially on the overall, qualitative performance of the system.

In our studies, the *Pioneer P3DX* is commanded to move on two surfaces:

Surface 1: a clean tiled surface (non-slippery)

Surface 2: a powder-layered tiled surface (slippery)

In order to identify the friction coefficient between the wheel (i.e., tire) of the *Pioneer P3DX* and these surfaces, we conduct a simple experiment to measure the friction coefficient of these surfaces. Fig. 5.3 shows the surfaces we conduct the experiment on and Fig. 5.4 displays the type of tire the *Pioneer P3DX* is equipped with.



(a) (b)

Fig.5.3: Type of surfaces, (a): clean tiled surface (non-slippery), (b): powder-layered tiled surface (slippery)

(a) (b)

Fig.5.4: (a): Wheel of *Pioneer P3DX* (b): Tire thread

In Fig. 5.5, we show the experiment set-up. The electronic scale, acting as a dynamometer is hooked up to one end of the robot.

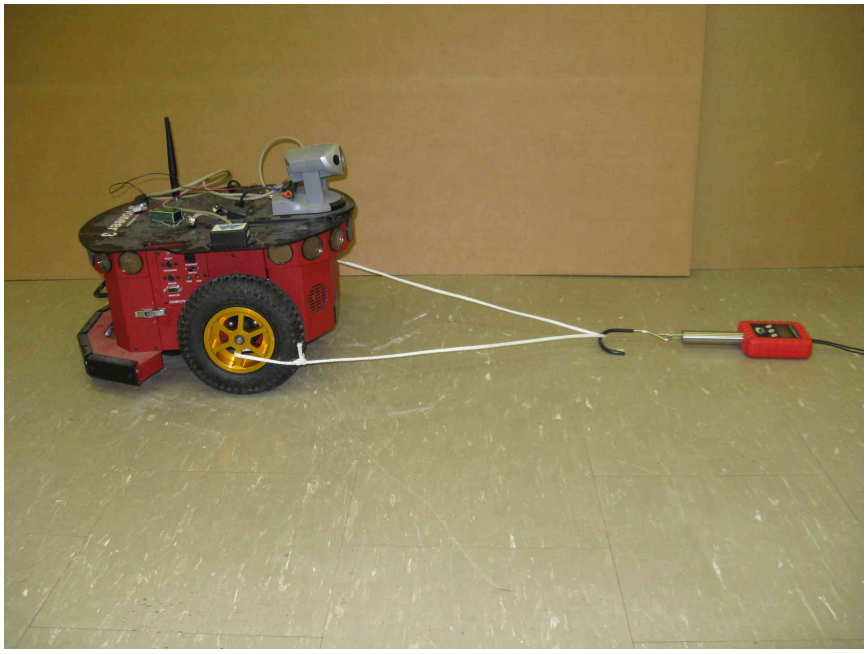


Fig. 5.5: Experimental setup to measure friction coefficient, μ_d

We pull the robot using the dynamometer and once the wheels start to slip, we continuously pull the dynamometer at constant speed and record the measurement. We run the experiment at random locations around the surface area so as to capture as much variations in the value of friction coefficient for that surface. Table 5.1 and 5.2 show the measurement of longitudinal and lateral traction forces data respectively, where for each surface we take 10 samples of data.

Table 5.1: Data of longitudinal traction force measurement

Longitudinal traction force on non-slippery surface (*kgf*)	Longitudinal traction force on slippery surface (*kgf*)
11.45	4.90
11.19	5.03
10.43	4.60
9.19	4.79
10.96	5.60

9.20	5.00
12.50	5.17
11.79	5.30
10.62	4.80
12.60	5.40
F_{ave} =10.993	F_{ave}=5.095

Table 5.2: Data of lateral traction force measurement

Lateral traction force on non-slippery surface (*kgf*)	Lateral traction force on slippery surface (*kgf*)
6.27	3.82
6.50	3.89
6.80	4.02
6.02	4.32
7.90	4.19
7.63	3.97
6.59	3.85
7.66	4.50
7.95	4.31
7.38	4.22
F_{ave} =6.941	F_{ave}=4.109

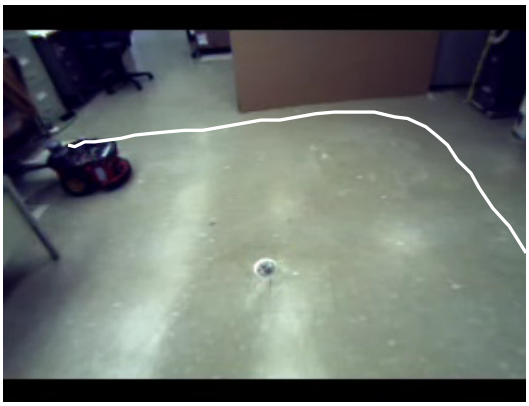
The *Pioneer P3DX* weighs 18kg and to estimate the value of longitudinal and lateral friction coefficients, μ_{dlon} and μ_{dlat} , we use the following equation,

$$F_{ave} = \mu_d m_{WMR} g \tag{5.1}$$

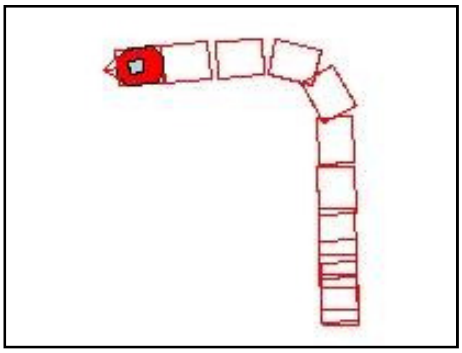
where m_{WMR} is the weight of the WMR, g is the gravitational acceleration and μ_d is μ_{dlon} for longitudinal case and μ_{dlat} for lateral case. We find μ_{dlon} and μ_{dlat} for the non-slippery surface to be 0.6107 and 0.3856 respectively, and for the slippery surface to be 0.2811 and 0.2283.

In order to study the effect of slip in sharp cornering motion, we command the WMR to start turning at a reference point (i.e., the point that connects two straight segments to form an *L*-shaped path). Fig. 5.6(a) shows the resultant trajectory of the

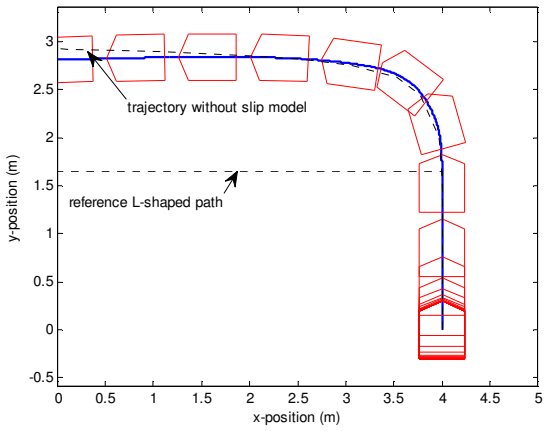
Pioneer P3DX on the clean, tiled surface (non-slippery surface). We find the trajectory is stable with negligible lateral slip and the *Pioneer P3DX* is able to respond to sharp cornering command satisfactory by moving parallel to the second straight segment of the path with insignificant deviation. Similar trajectory is also produced from the standard *Pioneer* simulator (i.e., *MobileSim 3.0*) shown in Fig. 5.6(b). (Note: The *MobileSim* produces small deviation from the *L*-shaped path, due to the strict 100*ms* robot command cycle, the program has to obey. However this is not the case for the real system where the command cycle can be from 90*ms* to 110*ms* resulted in the consistent *L*-shape trajectory for non-slippery surface). In Fig. 5.6(c), we present the result of the *Pioneer P3DX* simulation based on the new dynamic model where the surface is not slippery (i.e., the surface provides high traction). The trajectory of this model matches the real trajectory of the WMR when taking a sharp corner. Here, the magnitude of the lateral slip is shown to be relatively small as discussed later. In this experiment, we set the longitudinal friction coefficient μ_{dlon} and the lateral friction coefficient μ_{dlat} to be 0.6107 and 0.3856 respectively.



(a)



(b)

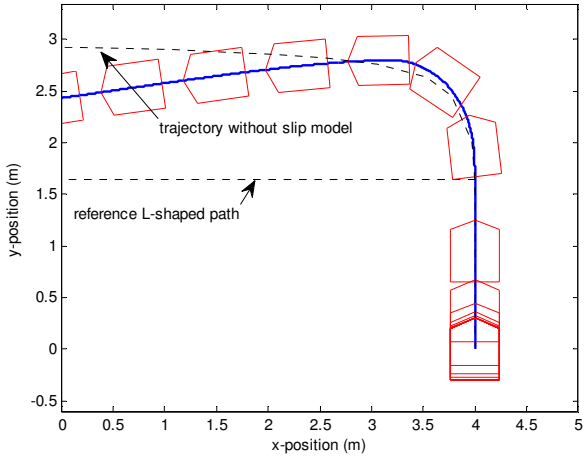


(c)

Fig.5.6: Trajectories of Pioneer P3DX on surface with negligible slip (a): experiment, (b): standard simulator model, (c): new dynamic model

For the second surface, we make it more slippery (by layering the tiled surface with powder). We observe in the experiment, that the WMR is not moving in parallel to the second straight segment of the *L*-shape path as shown in Fig. 5.7(a). While *MobileSim* and similar other WMR simulators cannot incorporate the surface slip-traction information and thus cannot predict the deviation (Fig. 5.6(b)), our dynamics model successfully captured the observed phenomenon due to slip-traction relationship as can be seen from Fig. 5.7(b). In this experiment, we set the longitudinal friction coefficient μ_{dlon} and the lateral friction coefficient μ_{dlat} to be 0.2811 and 0.2283 respectively.

(a)



(b)

Fig. 5.7: Trajectories of Pioneer P3DX on surface with significant slip (a): experiment, (b): simulation

We observe that the magnitude of the lateral slip from the real experiment for the slippery case is estimated to be $0.18m/s$ and for non-slippery case is $0.06m/s$ as shown in Fig. 5.8(top). More interestingly, based on the WMR simulated trajectories (Fig. 5.6(c) for non-slippery case and Fig. 5.7(b) for slippery case), the resultant lateral slip profiles produce a comparable match to the experimental lateral slip profiles as shown in Fig. 5.8(bottom). In the case of slippery surface, the magnitude of lateral slip is $0.20m/s$ and

for the non-slippery case, it is around $0.0668m/s$. We attribute the small mismatch in the magnitude of lateral slip between the experiment and the simulation to several factors, namely, our approximation of system parameters (i.e., moment of inertia, mass) as well as the exact value of friction coefficient for the type of surfaces under study, which may be varying throughout the course of path.

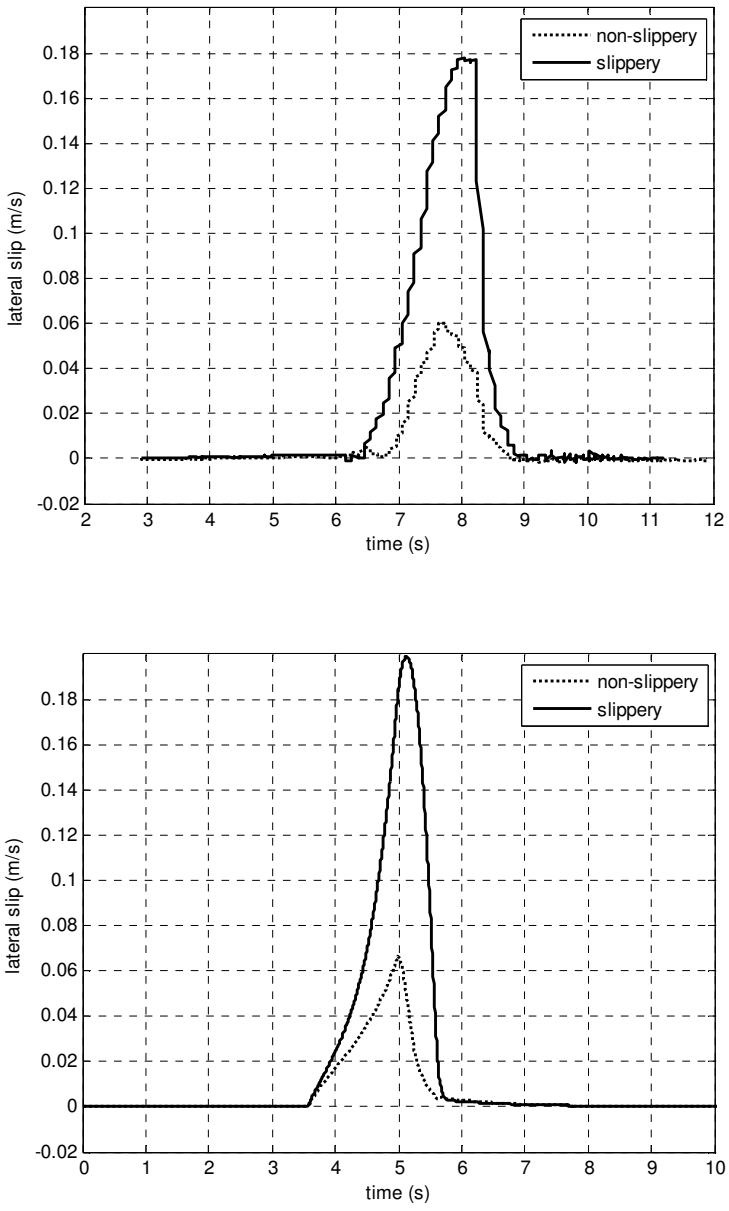


Fig. 5.8: Lateral slip profiles (top): Experiment, (bottom): Simulation

The results obtained from the simulation studies and the experiments are important to indicate that the presented modeling technique is able to capture the lateral slip phenomenon during WMR cornering motion at high speed. As such, it is natural for us for the next step to develop a controller that is capable of minimizing the effect of lateral slip especially on a slippery surface to avoid any instability during WMR autonomous navigation tasks. In the following discussion, we develop a dynamic velocity planner to regulate the velocity of the *Pioneer P3DX* to minimize the effect of lateral slip.

Dynamic planner through feed-forward control

For majority of on-shelve WMRs like *Pioneer P3DX,* the lack of access to the system inputs (i.e., torques given to the wheels) means the WMR can only be controlled at velocity level, thus voids the objective of developing the dynamic model of the WMR. Nevertheless, the lateral slip information obtained from the accelerometer in this study can be made useful to govern the desired forward velocity of the *Pioneer P3DX*, particularly during cornering at high speed.

Here we develop and implement the dynamic velocity planner for the *Pioneer P3DX* (also applicable to general WMR model). Our objective is to minimize the effect of lateral slip when the robot is to corner sharply at high speed ($0.8m/s$) on slippery surface. We model the desired control input using a function of lateral traction force as follows,

$$J(F_y) = \frac{F_{ymax}F_y}{(F_{ymax}-F_y)^{\mu}} \tag{5.2}$$

where μ is the decay factor and F_{ymax} is the peak lateral traction force and is dependent on the type of surfaces. F_y is the instantaneous lateral traction force related to the slip angle as defined in Eq. (4.9). Now we define function,

$$K(F_y) = \frac{1}{1+|J(F_y)|} \tag{5.3}$$

which behaves as a weighted coefficient to the desired forward velocity. From Eq. (5.3), we see that the weight approaches the value of 1.0 as the traction force gradually moves away from the pre-specified allowable maximum traction, F_{ymax}. It converges to zero as it approachesF_{ymax}.The rate of convergence can be set using the decay factor, μ. For such a WMR that can reach a maximum velocity of V_{max}, we set our desired forward velocity to be,

$$h_{desired} = K(F_y)V_{max} \tag{5.4}$$

The above dynamic planner essentially regulates the desired forward velocity that may be provided for the WMR navigation task such as path-following. It indirectly limits the magnitude of lateral slip in such a way that the WMR can operate only within the maximum allowable traction force.

We apply the dynamic velocity planner on the *Pioneer P3DX* to respond to a sharp corner using the same series of command as in the experiment we did to verify the lateral slip model and set the decay factor, μ to be 0.4. As shown in Fig. 5.9, the magnitude of the lateral slip on the slippery surface is successfully brought down to the lateral slip profile of non-slippery surface. In addition, the trajectory of the *Pioneer P3DX* after the application of dynamic planner on slippery surface is shown in Fig. 5.10. From this figure, we observe, the angle after the *Pioneer P3DX* takes the corner is kept to be the same as in the case of non-slippery trajectory and the *Pioneer P3DX* stops early due

to the regulation of the forward velocity. As compared to the trajectory of the *Pioneer P3DX* without the dynamic planner (Fig. 5.7(a)), due to lateral slip, the robot deviates from the straight path after the sharp cornering motion.

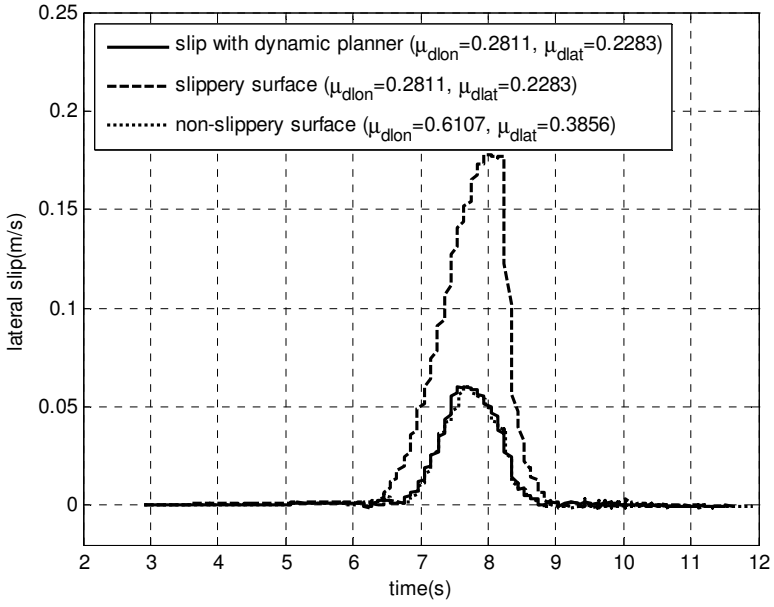


Fig. 5.9: Lateral slip before and after the application of dynamic planner

Fig. 5.10: Pioneer P3DX trajectory on slippery surface after the application of dynamic planner

We anticipate that in the future there will be more WMRs that are able to fulfill the demand of high speed applications by allowing direct regulation of motor torques where the dynamic model will be significant to study the performance of the WMR especially when the WMR is subjected to lateral slip. The model we develop in this dissertation can provide a basis to develop a better model-based controller where its performance can be investigated in the mobile robot simulator under various surface conditions.

WMR motion task: moving along a straight line

Experimental setup

The objective of the experiment is to command the WMR to move along a straight line and to investigate and verify the longitudinal slip dynamics from the developed model equation. Due to the inability of the accelerometer sensor to capture the longitudinal slip reliably, we seek to choose a laser range device to estimate the velocity of the WMR. When the WMR is moving forward in the straight line, the total velocity of the WMR is the same as the velocity of the wheel hub, thus enable us to use Eq. (4.1) to measure the longitudinal slip. As shown in Fig. 5.11, is the *Pioneer P3DX* equipped with the laser measurement sensor (LMS) 200 manufactured by the *SICK AG.*

Fig.5.11: Pioneer P3DX with mounted laser measurement sensor

We run the experiment on a slippery surface (i.e., liquid soap-layered tiled surface) as shown in Fig. 5.12 with friction coefficient, μ_d estimated at 0.1512 for both longitudinal and lateral versions. This type of surface is more slippery than the previous, powder-layered tiled surface, so as, to enable more efficient study on the effect of longitudinal slip towards the overall dynamics of the WMR system.

Fig. 5.12: Tiled surface layered by liquid soap

Simulation and experimental results

In Fig. 5.13, we present the WMR velocity profiles captured using motor encoder sensor and LMS unit for both slippery and non slippery cases. While the velocity profile of the WMR on non-slippery case is quite consistent throughout the time, the velocity profile captured by LMS unit on slippery case (Fig 5.13 (top)), especially from the time of $1s$ to $1.8s$ is lower than that of the signal recorded by the motor encoder. This represents significant longitudinal slip that occurs while the WMR is moving forward at the rate of $0.45m/s^2$ to reach the top speed of $0.84m/s$.

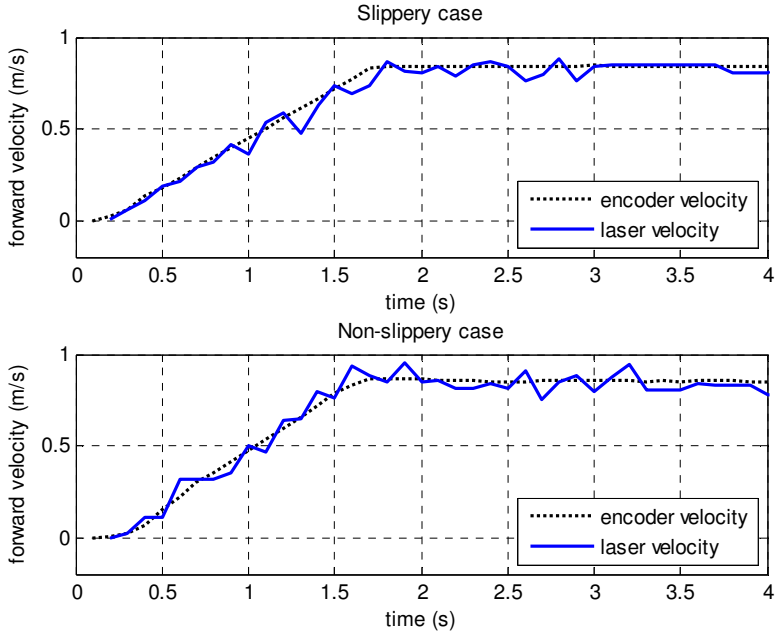


Fig. 5.13: WMR velocity profiles on (top): Slippery case, (bottom): Non-slippery case

The position data captured from the LMS unit is then differentiated and compared to the data recorded by the wheel encoders to produce the longitudinal slip profiles. This

result is then compared to the longitudinal slip profiles produced by the simulated model and shown in Fig. 5.14.

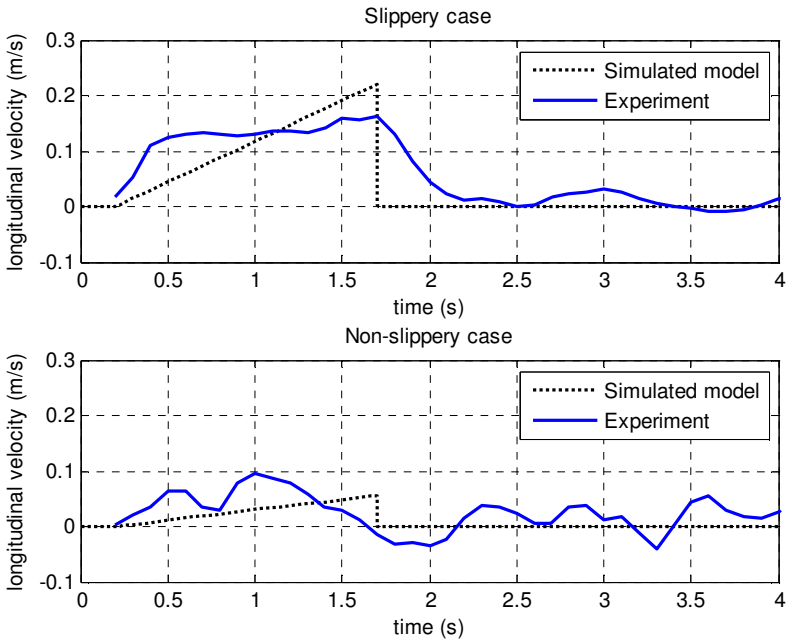


Fig. 5.14: Longitudinal slip profiles obtained from experiment and simulated model

In the case of slippery surface with friction coefficient, $\mu_{dlon} = 0.1512$, our proposed dynamic model is able to reflect the occurrence of longitudinal slip quite satisfactory. (Fig. 14(top)). Meanwhile, for non-slippery surface with friction coefficient, $\mu_{dlon} = 0.6107$, the simulated model shows a smaller size of longitudinal slip profile. The result from LMS unit however is insufficient to satisfactorily characterize the slip of this size.

The qualitative performance of the two longitudinal slip profiles is useful in verifying the new WMR model developed in this research. We contribute the quantitative inaccuracy between the real and simulated WMR due to several factors. First is the inconsistency of the surface friction coefficient for the surface under study. The liquid-

soap layered surface has low friction coefficient and this value varies quite dramatically through-out the surface area, which in turn, affects the overall performance of the WMR. In addition, besides having other system uncertainties as described in the case of lateral slip, the model of traction force used in the simulation is described using a constant friction coefficient, thus, may not realistically represent the traction force for the surface under study.

CHAPTER VI

CONTROL OF NONHOLONOMIC WMR

We established in Chapter III that, the nonholonomic nature of a WMR is related to the model of contact point between the wheel and the surface. Due to the natural occurrence of wheel slip when the robot is in motion, we develop a new WMR dynamic model that considers slips at the wheel-surface contact point. Considering the assumptions of the ideal rolling constraints and the ideal lateral constraints, which assume slip to be nonexist, we relax the constraints by introducing new states to represent longitudinal and lateral slips. This implies the presence of a new non-integrable set of first order constraints Eq. (3.22-3.25) on the configuration variables. These constraints reduce the instantaneous motions that the robot can perform and lead to some challenging problems in the synthesis of feedback controller in nonholonomic motion planning.

State space representation

We refer to the WMR model with constraints, Eq. (3.33). The state space representation of the WMR system can be written in the form,

$$\dot{x} = \begin{bmatrix} Sv \\ (S^T M S)^{-1}\left(-S^T M \dot{S} v + S^T F\right) \\ \frac{-r f_{lon}}{I_{wy}} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ I \end{bmatrix} \tau \tag{6.1}$$

where we choose $x = [x_c, y_c, \varphi, \eta_1, \eta_2, \rho_1, \rho_2, \theta_1, \theta_2, \dot{\eta}_2, \dot{\rho}_1, \dot{\rho}_2, \dot{\theta}_1, \dot{\theta}_2]$ as the state space variables. The system in Eq. (5.1) can be further simplified to reduce to,

$$\dot{x} = f(x) + g(x)u \tag{6.2}$$

where u is the control input to the system. (i.e., torques to the driving wheels)

Motion tasks

The design of a suitable feedback controller for a WMR depends on the motion tasks the robot is to perform. In general they are three possible motions than can be broadly classified as namely *point-to-point motion* (the robot must reach a desired goal configuration starting from a given initial configuration), *path following* (the robot must reach and follow a geometric path in the Cartesian space starting from a given initial configuration) and *trajectory tracking* (the robot must reach and follow a trajectory in Cartesian space (i.e., a geometric path with associated timing law) starting from a given initial configuration).

In this research work we design a path following controller that allows the WMR to navigate in the presence of slip. The goal here is to ensure that the WMR can follow a specific path with a desired forward velocity while negotiating slip. However, if the WMR determines that it is becoming unstable while trying to achieve the desired velocity due to slip then it will autonomously reduce the forward velocity exploiting the maximum allowable traction forces from the slip-traction properties to follow the path under the given condition. In this paper, by navigation performance we mean the ability of the WMR to follow a given path with a given forward velocity. However, between these two tasks, we assign a higher priority on staying on the path over achieving the desired forward velocity if wheel slip causes instability.

Output equations and feedback linearization

The idea of employing feedback linearization is to transform the nonlinear system into an equivalent linear system. There are two notions related to the feedback linearization techniques namely, *input-state linearization* and *input output linearization*. The former is defined when the state equations can be completely linearized and the latter means the input-output map is linearized but the state equations may be only partially linearized. We give the mathematical definition of both linearization techniques as below,

Input state linearization: The nonlinear system in the form of $\dot{\mathbf{x}} = f(x) + g(x)u$ with $f(x)$ and $g(x)$ being smooth vector field on $\Re^n$, is said to be input-state linearizable if there exists a region Ω in $\Re^n$, a diffeomorphism $T: \Omega \to \Re^n$, and a nonlinear feedback control law $u = \alpha(x) + \beta(x)v$, such that the new state variables $\mathrm{z} = T(x)$ and the new input, v satisfy a linear time invariant relation $\dot{\mathbf{z}} = A\mathbf{z} + Bv$

where $A = \begin{bmatrix} 0 & 1 & 0 & . & . & 0 \\ 0 & 0 & 1 & . & . & 0 \\ . & . & 0 & . & . & 0 \\ . & . & . & . & . & 0 \\ 0 & 0 & . & . & . & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$ and $B = [0 \quad 0 \quad 0 \quad . \quad . \quad 1]^T$

Input output linearization: The nonlinear system in the form of, $\dot{\mathbf{x}} = f(x) + g(x)u$

where $\mathbf{x}$ is the state and u is the input, with output equation, $y = h(x)$ is said to be input-output linearizable if it is possible to generate a linear differential relation between the output y and a new input, v satisfy a linear time invariant relation $\dot{\mathbf{z}} = A\mathbf{z} + Bv$

where $A = \begin{bmatrix} 0 & 1 & 0 & . & . & 0 \\ 0 & 0 & 1 & . & . & 0 \\ . & . & 0 & . & . & 0 \\ . & . & . & . & . & 0 \\ 0 & 0 & . & . & . & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$ and $B = [0 \quad 0 \quad 0 \quad . \quad . \quad 1]^T$

In this dissertation, based on the dynamic model previously developed, we approach the problem of WMR navigation under the path following formulation. Referring to Fig. 3.3, the coordinates of the look-ahead point P_l are given by,

$$x_l = x_c + l\cos\varphi$$
$$y_l = y_c + l\sin\varphi \tag{6.3}$$

By following the conventional wisdom in which one drives a car, we can establish the following two driving objectives:

1. The WMR has to pursue a given prescribed path as closely as possible, and

2. The WMR has to travel the path with a given desired forward velocity.

Based on the above objectives, we can establish the output equations where the first equation relates the shortest distance between the WMR (a reference point on the WMR platform) and the desired path. The second equation is to describe the WMR forward velocity. Let the output equation be represented by a vector *y*, where,

$$y = h = [h_1(q) \;\; h_2(v)] \tag{6.4}$$

where $h_1(q)$ is a measure of the first objective and $h_2(v)$ is a measure of the second objective. Since any set of paths can be constructed through a combination of circular and straight-line segments (Dubins 1957), we develop explicit equations for $h_1(q)$ for both circular and straight-line paths. For a circular path $h_2(v)$ can be formulated as follows,

$$h_1(q) = h_1(x_c, y_c, \varphi) = \sqrt{\left(x_l - x_f\right)^2 + \left(y_l - y_f\right)^2} - R \tag{6.5}$$

where $P_f = (x_f, y_f)$ is the instantaneous center of circular path with respect to inertial frame and R is the instantaneous radius of the circular path. Points $P_l = (x_l, y_l)$ (i.e., look-ahead point) and $P_c = (x_c, y_c)$ (i.e., center of mass) are related through Eq. (6.3)

As for a straight-line path, the output equation becomes,

$$h_1(x_c, y_c, \varphi) = \frac{C_1 x_l + C_2 y_l + C_3}{\sqrt{C_1^2 + C_2^2}} \tag{6.6}$$

where all $C_i, i = 1,2,3$ are constants used to describe the line. From Eq. (6.5) and Eq. (6.6), we see the shortest distance between the look-ahead point and the path can be taken as the absolute value of h_1. After the introduction of longitudinal slip, the forward velocity of the WMR can be written as follows,

$$h_2(v) = \dot{x}_c \cos\varphi + \dot{y}_c \sin\varphi = \frac{r}{2}(\dot{\rho}_1 + \dot{\rho}_2) \tag{6.7}$$

where, $\dot{\rho}_1$ and $\dot{\rho}_2$ are $v_2$and v_3 respectively.

Now, we proceed to develop a nonlinear controller based on the feedback linearization technique. The decoupling matrix for feedback linearization for the above output equations are differentiated until the input terms appear in the output equations such that,

$$\dot{y}_1 = \frac{\partial h_1}{\partial q}\dot{q} = J_{h1} S v$$

$$\ddot{y}_1 = \frac{\partial (J_{h1} S)}{\partial q}\dot{q} v + J_{h1} S \dot{v} \tag{6.8}$$

$$\dot{y}_2 = J_{h2} \dot{v} \tag{6.9}$$

and usually we can set $\dot{v} = u$ where u is the input to the control system. As an example, for the straight-line path,

$$J_{h1} = \frac{1}{\sqrt{C_1^2 + C_2^2}} [C_1 \quad C_2 \quad C_2 l\cos\varphi - C_1 l \sin\varphi \quad 0 \quad 0 \quad 0 \quad 0 \quad 0]$$

and

$J_{h2} = [0 \quad 0 \quad 0 \quad 0 \quad 0 \quad r/2 \quad r/2]$.

$J_{h1} = \frac{\partial h_i}{\partial q}$ is known as *Jacobian* matrix and we can use them to compute the decoupling matrix, Φ as follows,

$$\Phi = \begin{bmatrix} J_{h1}S \\ J_{h2} \end{bmatrix} \tag{6.10}$$

As oppose to *input-state feedback linearization*, we utilize the decoupling matrix to establish the *input-output feedback linearization* as shown below,

$$\ddot{y} = \begin{bmatrix} \ddot{y}_1 \\ \dot{y}_2 \end{bmatrix} = \dot{\Phi}\nu + \Phi u \tag{6.11}$$

It has been shown in (Rosenberg 1977, Sarkar 1993) that for a nonholomic system, the system is not *input-state linearizable* if one or more constraints are nonholonomic.

If we let $u = \dot{v}$ and represent Eq. (6.11) in the form of $\ddot{y} = U + V\dot{v}$, with Eq. (3.46) we can find $\ddot{y}$ in the new following form,

$$\ddot{y} = U + V\varsigma + WF \tag{6.12}$$

where $W = V\kappa$.

It is clear from Eq. (6.11) that we cannot solve the problem to model the input torque, τ. Thus, we differentiate the output equations again as follows,

$$\dddot{y} = \begin{bmatrix} \dddot{y}_1 \\ \ddot{y}_2 \end{bmatrix} = \ddot{\Phi}\nu + 2\dot{\Phi}\dot{\nu} + \Phi\ddot{\nu} \tag{6.13}$$

where

$$\ddot{v} = (S^T M S)^{-1}(-(\dot{S}^T M S + 2S^T M\dot{S})\dot{\nu} - (\dot{S}^T M\dot{S} + S^T M\ddot{S})\nu + \dot{S}^T F + S^T\dot{F}) \tag{6.14}$$

Here we assume the derivative of traction function, $\dot{F}$, would have the torque components (as we do by utilizing the *Magic formula*), so as, the input torque can be directly

determined from the inverse dynamic relationship. For example if we let $\ddot{y} = P + Q\tau$, the input torque can found as,

$$\tau = Q^{-1}(u_d - P) \tag{6.15}$$

where Q^{-1} is taken to be the inverse of Q. If we let the error term,

$$e = h_{i\_desired} - h_{i\_actual}, \tag{6.16}$$

then the desired control u_d can be formulated as follows,

$$u_d = \ddot{y}_{desired} + K_a\ddot{e} + K_v\dot{e} + K_p e \tag{6.17}$$

where K_a, K_v and K_p are constant gains for the linear outer feedback loop chosen to ensure the convergence of the control error.

Dynamic planner with path following controller

We implement the dynamic planner with the path following controller, which is subjected to wheel slip, as discussed in Chapter V. Besides setting our desired forward velocity to be,

$$h_{2\_desired} = K(F_y)V_{max} \tag{6.18}$$

the desired absolute distance to be within a predefined boundary can be defined as,

$$h_{1\_desired} = absolute\ distance < |\varepsilon| \tag{6.19}$$

where ε can be taken as a very small number.

The above dynamic planner essentially regulates the desired forward velocity that is provided to the path-following controller as a reference input based on slip-traction relationship. In other words, it indirectly limits the slip such that the WMR can operate within the maximum allowable traction force. The complete WMR system with dynamic

path following controller is depicted in Fig. 6.1 (i.e., the system in dotted block is the WMR dynamical model).

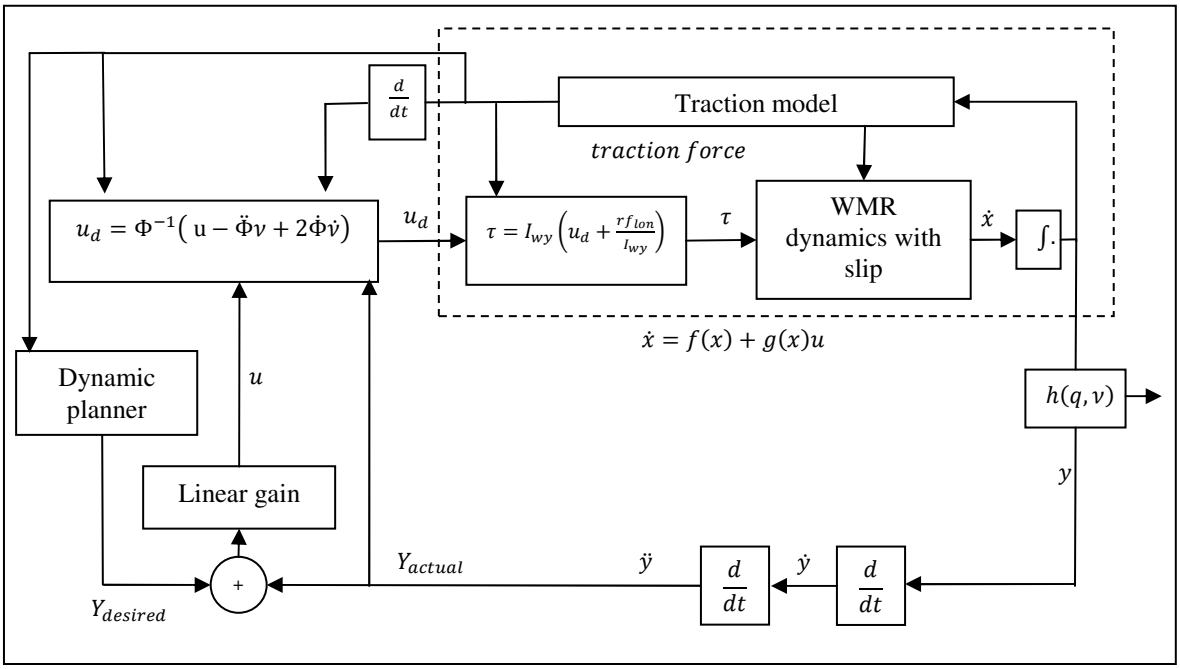


Fig.6.1: Block diagram of the WMR control system

CHAPTER VII

SIMULATION RESULTS AND DISCUSSION

We first present simulation results to demonstrate the validity of the WMR model that include slip dynamics and the efficacy of the dynamic path following controller that utilizes slip-traction properties. We also present preliminary experimental results to show the effectiveness of the developed simulation environment in analyzing and translating the simulation results in experiments. For the simulation task we chose the WMR parameters (refer Fig.4) as follows: b=0.24m; d=0.05m; r=0.095m; m_r=17kg; m_w=0.5kg; I_{rz}=0.537kgm^2; I_{wy}=0.0023kgm^2; I_{wz}=0.0011kgm^2. The look-ahead point is at 0.5m away from point P_c along the x-axis of the WMR body. The gains for the linear feedback loop are designed in such a way that we can get a critically damped output response where K_{p1}=60, K_{v1}=5, K_{a1}=5, K_{v2}=50, and K_{a2}=0.1. The decay factor, μ in Eq.(5.2), regulates traction performance. In this simulation we choose $\mu = 4.0$. We apply our proposed approach to the WMR navigation that is subjected to both lateral and longitudinal slips. As for the desired path the WMR has to follow, it is composed of two straight-line segments (i.e., segment AB and segment BC), connected at a right angle to resemble a sharp corner. The idea to have such a shape of path is to observe the effect of slip when the WMR needs to navigate through sharp cornering, simulating a harsh yet realistic navigation scenario (e.g., in target chasing scenario or in avoiding dynamic obstacles etc.). Additionally, we are interested in investigating how the presented controller allows autonomous navigation on different surfaces having varying traction properties.

Therefore we conduct simulation studies to show how the WMR performs during cornering on two different surfaces under three different scenarios (i.e., *Case I, II*, and *III*).

For the first two cases we specify a fixed reference velocity for the WMR and let the controller (Eq.(6.15)) attempt to achieve this velocity on both surfaces. Here we do not let the dynamic planner (Eq.(6.18) and Eq.(6.19)) play any role so that we can observe the navigation performance of the WMR under different velocities without any dynamic adjustment of the reference input (i.e., the forward velocity). As we will see, that navigation performance depends on the desired velocity, which varies for different surfaces. Then in *Case III* we demonstrate how the navigation performance can be improved by the introduction of the dynamic planner in conjunction with the controller such that the WMR can autonomously adjust its velocity to avoid excessive slipping leading to instability. For each of these cases, we set the initial states in such a way that the WMR can reach the desired forward velocity before encountering the first turn. The two surfaces that we use in this analysis are characterized by two different values of friction coefficients. The Surface 1 (i.e., non slippery surface) has a coefficient of friction $u_2 = 0.7$ represents a wet pavement surface and the Surface 2 (i.e., slippery surface), on the other hand, has a coefficient of friction $u_1 = 0.3$, which represents a snowy pavement surface (Muller et al. 2003, Wang et al. 2004). Fig. 7.1 shows the traction curve properties for the above-mentioned two surfaces. The maximum allowable torque for each driving motor is set to 1.24 *N-m* and the absolute distance, ε in Eq. (38) is set to be zero.

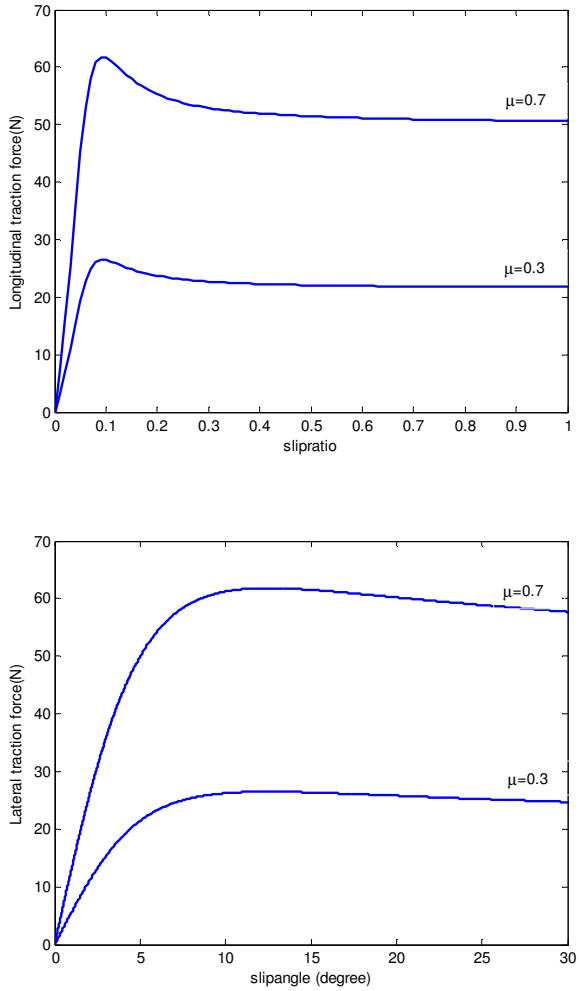


Fig.7.1: Traction force vs slipratio (top)/slipangle(bottom) relationship for two surfaces with different friction coefficients (a) 0.7 (b) 0.3

***Case I**: Effect of longitudinal and lateral slips on navigation performance on both Surface 1(non slippery) and Surface 2 (slippery) when the desired forward velocity is 1m/s.*

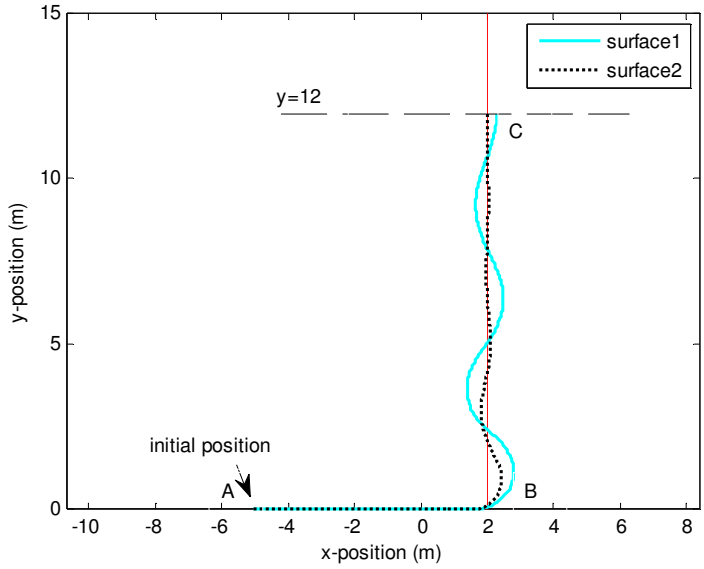


Fig.7.2: WMR's path following trajectory for Case I

The WMR navigates on both surfaces in a stable manner as shown in Fig. 7.2. However, note that the WMR deviates from the desired path more on Surface 1 in order to generate enough traction to turn, which resulted in a longer time to reach the desired destination at y=12*m* (in this case, it took 1.0493*s* longer for WMR on Surface 1). From Fig. 7.3, we observe that the response showed by the WMR to reach the desired forward velocity is faster and has smaller overshoot on Surface 2 than on Surface 1.

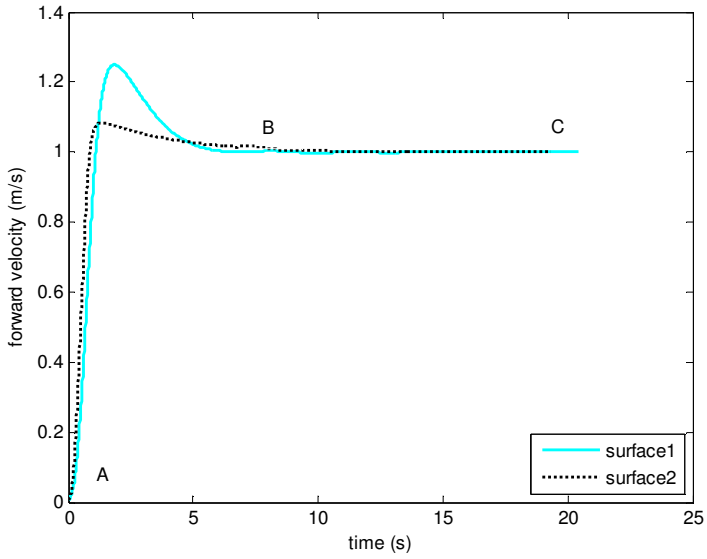


Fig.7.3: Forward velocity profile for Case I

This comes not as a surprise as can be seen in Fig. 7.4 (segment AB). The WMR on Surface 2 has to accelerate faster to gain enough traction to propel the WMR forward. This requires a larger magnitude of longitudinal slip and a faster rate of slip than that on Surface 1 to generate enough traction force. For Surface 1, the WMR has to accelerate a little longer than that on Surface 2 to cause the wheel to slip with small magnitude but larger traction force.

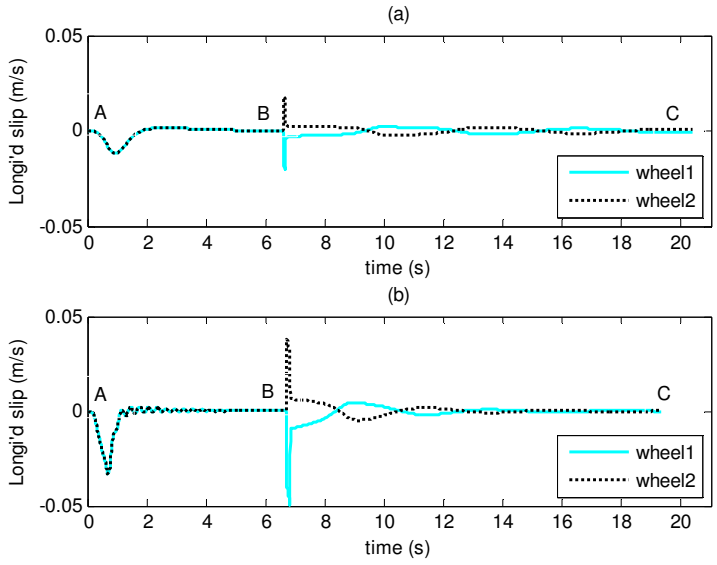


Fig.7.4: Longitudinal slip on (a) Surface 1 (b) Surface 2 for Case I

As expected, there is no lateral slip on either surface on segment AB (Fig. 7.5). However, as the WMR begins turning, the wheels start slipping laterally (Fig. 7.5, segment BC). On Surface 1, due to its higher traction properties, the WMR slips less laterally as compared to Surface 2, and thus follows a bigger radius of turning.

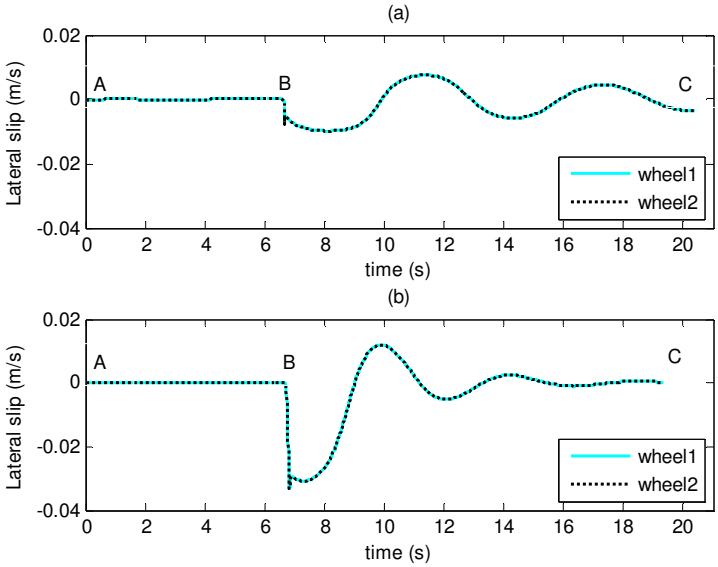


Fig.7.5: Lateral slip on (a) Surface 1 (b) Surface 2 for Case I

On Surface 2 due to its low traction properties, the WMR can slip more laterally, which in turn allows the WMR to quickly steer towards the desired path. This phenomenon resembles car racing where expert drivers utilize the traction properties of the race course to slip laterally to improve steering performance. Similarly a relatively larger magnitude of longitudinal slip is observed on Surface 2 as compared to Surface 1 (Fig. 7.4, segment BC). We observe that slips on both surfaces reduce over time indicating that the WMR is capable of following the desired path at the desired forward velocity on both surfaces. This suggests the possibility of further increasing the desired forward velocity of the WMR for the given path, which we study in *Case II*.

***Case II**: Effect of longitudinal and lateral slips on navigation performance on both Surface 1(non slippery) and Surface 2 (slippery) when the desired forward velocity is 2m/s.*

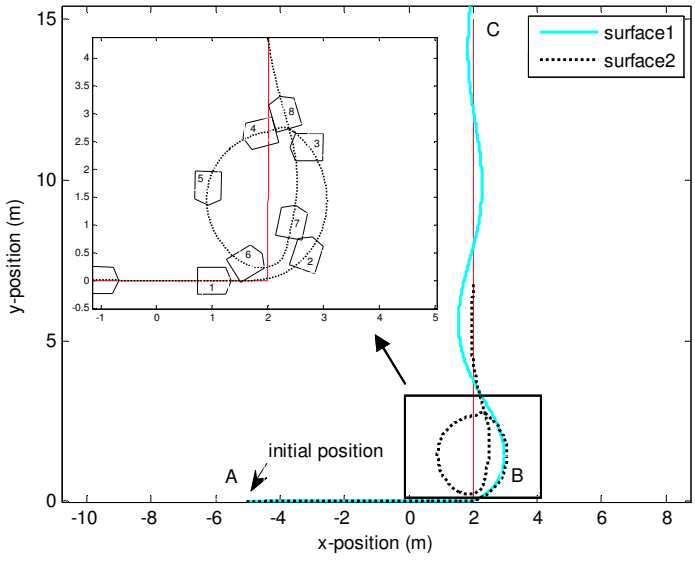


Fig.7.6: WMR's path following trajectory for Case II

As seen from *Case I*, the WMR is capable of traversing the given path in a stable manner at 1*m/s* forward velocity. Now in *Case II* we increase the desired forward velocity of the WMR navigation to 2*m/s*. The objective is to investigate how slips on different surfaces influence WMR navigation as the forward velocity is increased. Such an investigation will provide insight on the maximum allowable forward velocity of a WMR for a given surface and will help develop advanced planners that can accommodate slip characteristics during path planning for an autonomous controller. As shown in Fig. 7.6, the WMR is able to follow the path on Surface 1 at this forward velocity. But more importantly, the slip on Surface 2 is so much that the WMR becomes initially unstable (inset of Fig. 7.6) and deviates from the given path. Here the WMR over steers right after it takes the first corner to follow the path on Surface 2. However, in this case, the controller is eventually able to bring the WMR back to follow the desired path. It should be clear from this result that any further increase in the forward velocity for the given surface will adversely affect the performance of the WMR to follow the desired path. As can be seen from the traction curve properties (Fig. 7.1), excessive slip beyond the peak of the traction curve (which depends on the nature of the surface) reduces that available traction force and leads to instability. Thus, a controller with a fixed reference forward velocity and fixed gains may not be suitable in such a scenario. In order to address this issue, we propose a new dynamic planner that can modify the reference input as discussed in *Case III*. In Fig. 7.7, we show the forward velocity profile of the WMR on both surfaces. The velocity of the WMR on Surface 2 drops whenever the WMR heading angle is not parallel to its motion path.

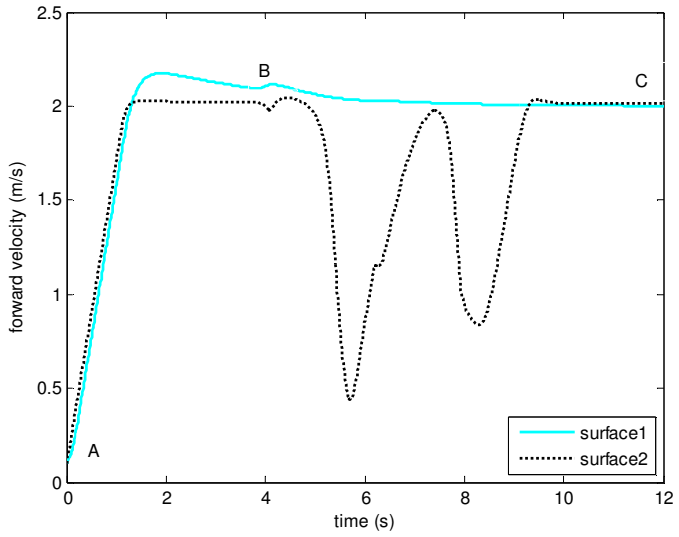


Fig.7.7: Forward velocity profile for case II

From Fig. 7.8, when we compare to *Case I,* we observe that the magnitude of the longitudinal slips are larger for both surfaces when the WMR is starting to move forward (segment AB). On the second straight segment, BC, while the WMR can navigate on Surface 1 in a stable manner, for Surface 2 the longitudinal slip profile shows instability (does not complement each other).

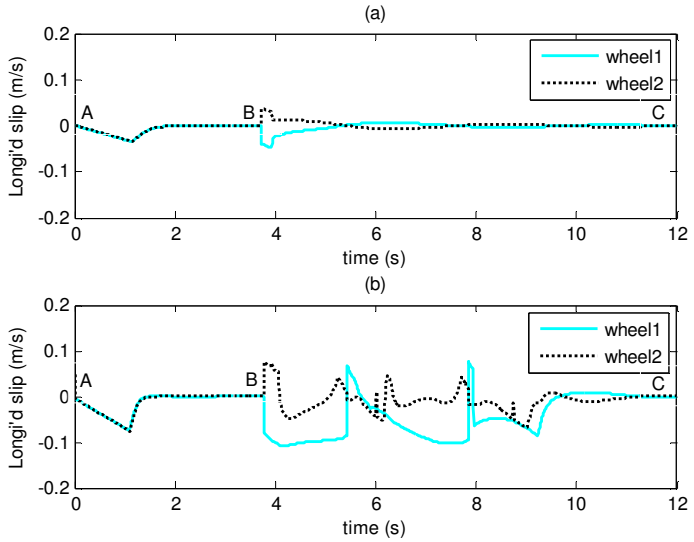


Fig.7.8: Longitudinal slip on (a) Surface 1 (b) Surface 2 for Case II

The navigation instability can also be learned from the lateral slip profile as shown in Fig. 7.9 as the magnitude of slip on Surface 2 is relatively huge when compared to Surface 1 and is not reduced after the 6th second.

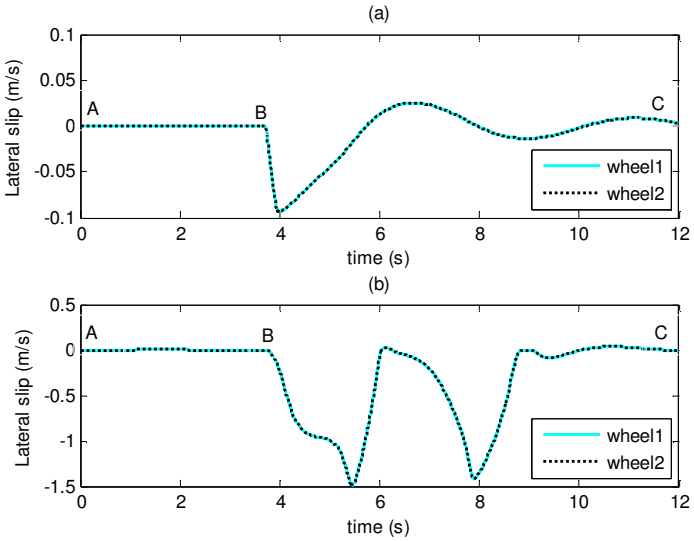


Fig.7.9: Lateral slip on (a) Surface 1 (b) Surface 2 for Case II

***Case III**: Effect of a planning strategy to control the longitudinal and lateral slips on both Surface 1 (non slippery) and Surface 2 (slippery) by autonomous adaptation of the desired forward velocity (max 2m/s)*

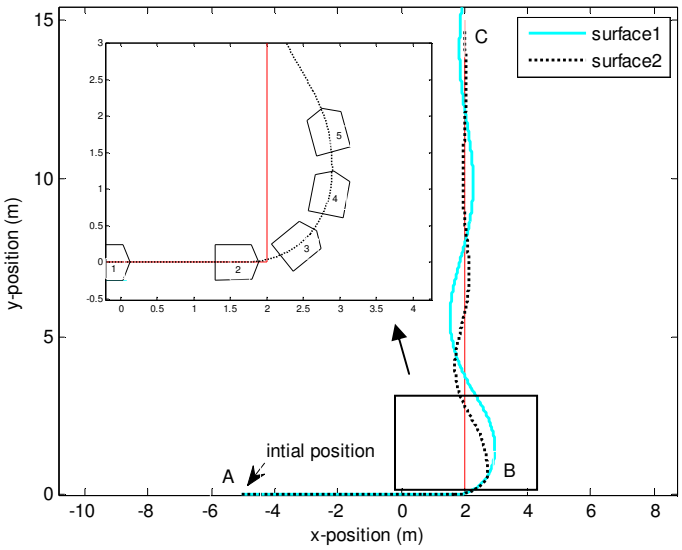


Fig.7.10: WMR's path following trajectory for Case III

In this case, we show that by using the path following controller with a dynamic planner developed in this paper, we can adaptively change the desired forward velocity based on the available traction force (which is a function of slip) and thus can stabilize the WMR during path-following navigation at high-speed. In particular, we have seen in *Case II* that the WMR becomes initially unstable on Surface 2 when the desired forward velocity is set at 2*m/s*. Here we show that by dynamically changing the desired forward velocity during cornering, the WMR can be made to follow the path on Surface 2. Fig.7.10 shows that the WMR can take the sharp turn in a stable manner.

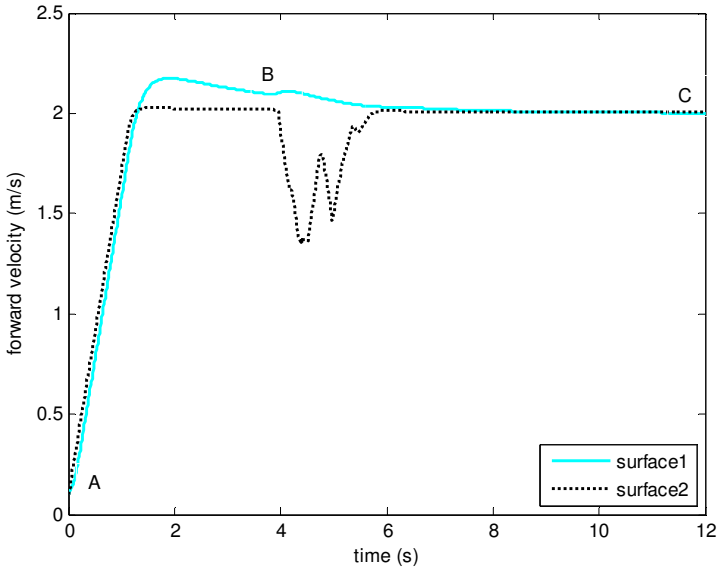


Fig. 7.11: Forward velocity profile for Case III

Fig. 7.11 shows a comparison of the desired forward velocity profiles before after the dynamic planner is applied. It is clear that by using the dynamic path following controller, the desired forward velocity on Surface 2 is regulated to make the navigation stable. Form the inset in Fig. 7.10, the drop in the forward velocity value on Surface 2 minimizes the deviation of the WMR's heading from its motion path. On the other hand, we observe the desired velocity on Surface 1 is maintained as in *Case II*. In Fig. 7.12, we see that the longitudinal slip for wheel 2 on Surface 2 is relatively large than that of wheel 1. This is required to maintain enough traction to stabilize the WMR motion during cornering. The velocity stabilization is also reflected on lateral slip profiles as shown in Fig. 7.13 where on the Surface 2, the magnitude of the slip is reduced gradually especially during the first cornering.

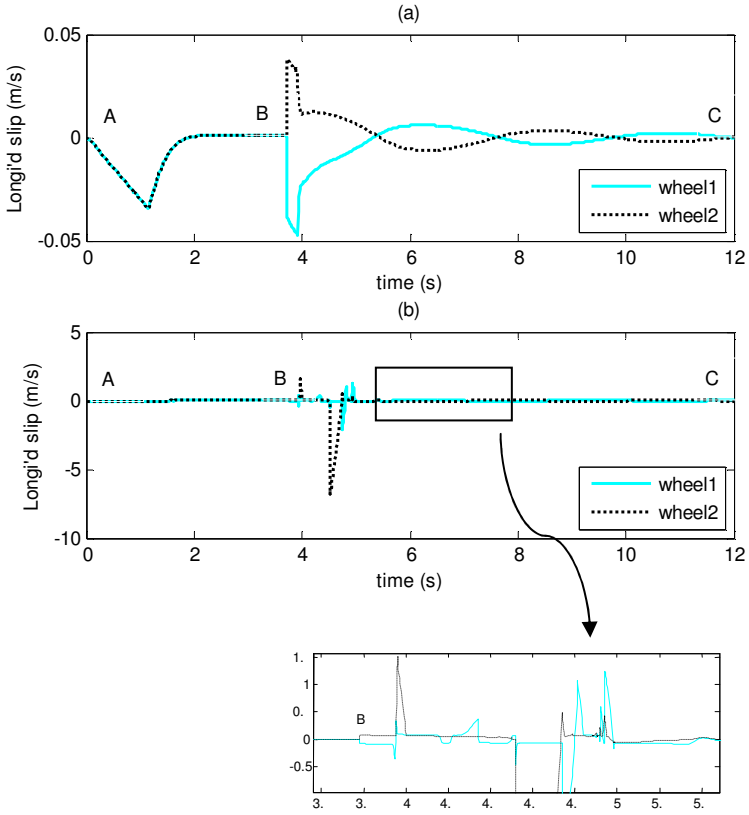


Fig.7.12: Longitudinal slip on (a) Surface 1 (b) Surface 2 for Case III

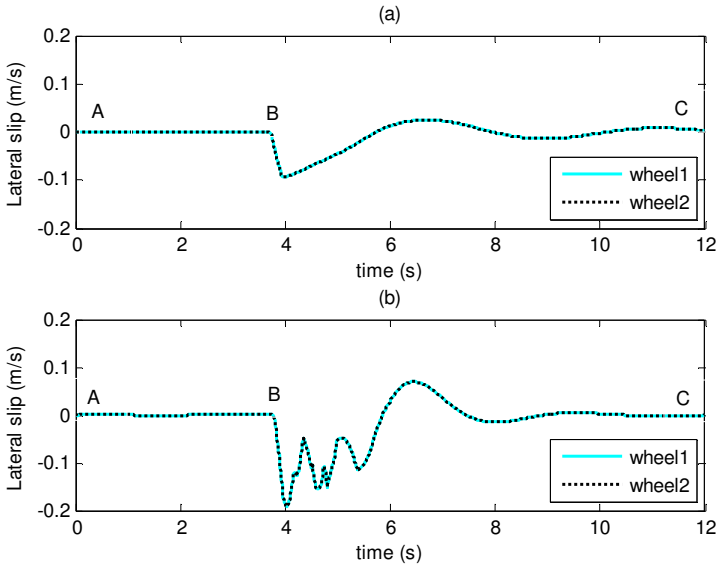


Fig.7.13: Lateral slip on (a) Surface 1 (b) Surface 2 for Case III

CHAPTER VIII

CONTRIBUTIONS

In this dissertation, we investigate how the WMR can be modeled and used to navigate in the presence of slip. We summarize the significance of the work in this chapter.

First, the presented dynamic model is a new contribution in the literature. We model the WMR in such a way that the wheel and ground contact point allows both longitudinal and lateral slips, which in turn represents a more realistic WMR navigation scenario. In order to include the slip dynamics into the WMR dynamics we augment the generalized coordinates of the WMR by including the slip coordinates and develop an integrated dynamic model that can simulate WMR motion in the presence of slip. The validity of the model has been investigated through a set of preliminary experiments. The results are encouraging.

Second, we design a path following controller and a dynamic planner for the WMR that can improve the performance of navigation and avoid instability in the presence of slip. In particular, this control technique allows the WMR to exploit slip to improve maneuverability. The presented approach exploits slip-traction properties to generate as much velocity as possible without causing instability of the WMR. As a result, such an approach has the potential to enhance WMR navigation performance in real world situation when the tasks demand sudden changes in motion at high-speeds. It will help the WMR to autonomously adjust its speed appropriately based on the

environmental conditions and thus will help achieve smarter navigation capabilities in the future.

Third, the use of the new model in a mobile robot simulator can improve the simulator performance where researchers can expect more realistic behavior of the WMR. As a consequence, they will be able to design, test and develop new planners and controllers that can be readily applicable to real-world scenarios.

While the work presented has a number of advantages, it is not without its limitations. First, the model is developed for planar navigation. Second, we assume that we have a priori knowledge of the traction properties of a given surface. Third, the controller presented here requires slip information, which we assume to be available all the time. While the extension of the dynamic model for non-planar navigation is feasible based on the presented methodology, the other two limitations require further discussion. In real applications, in order to model slip and traction force in the dynamics we need to measure the slip and identify the surface characteristics. Slip measurement and surface identification are separate research topics by themselves and are beyond the scope of this dissertation.

Appendix

Programming the serial port

The device `/dev/ttyS*` is used to hook up the client terminal to the onboard Linux box and needed to be configured properly after the machine starts up. All the configurations of the device can be done within a program and stored in a structure called `struct termios`. This structure is predefined in `<asm/termbits.h>` file and shown as follows,

```
#define NCCS 19
struct termios {
      tcflag_t c_iflag; // input mode flags
      tcflag_t c_oflag; //output mode flags
      tcflag_t c_cflag; //control mode flags
      tcflag_t c_lflag; //local mode flags
      cc_t c_line; //line dicepline
      cc_t c_cc[NCCS]; //control characters
      }
```

The input mode flags `c_iflag` is defined to handle all input processing to allow the characters sent from the device (i.e., accelerometer) to be processed before they can be properly read. Similarly, the output mode flags `c_oflag` is defined to handles the output processing. The setting of the port such as the baudrate, bits per character etc. can be done in the control mode flags, `c_cflag`. The local mode flags, `c_lflag` determine if characters are echoed or signals are sent to the program. Finally the `c_cc` defines the control characters such as EOF, stop and so on. The default values for the control characters are also defined in `termios.h`. The serial programming between the accelerometer, *MDS302* and the Linux box is given below to accomplish the research task. The comments given are not comprehensive but sufficient to understand the reasons behind all the settings.

```
#include "stdio.h" //Standard i/o definitions
#include "fcntl.h" //File control definitions
#include "errno.h" //Error number definitions
#include "termios.h" //POSIX terminal control definitions

struct termios oldtio, newtio;
int mainfd = 0

int main(void)
{
//~~~Open serial port (i.e., 'configure port()')~~~
//~~~Returns the file descriptor on success
mainfd=configure_serial();
printf("Succeed to open the port - %d", mainfd);

…
→ Our program goes here (i.e., initialize accelerometer, initialize
Pioneer ←)
…

//~~~close the port~~~
tcsetattr(mainfd, TCSANOW, &oldtio);
printf(" \n I'm closing the port now \n");
close(mainfd);
}

int configure_serial(void)
{
int fd; //file descriptor for the port

//~~~Open the port with options, prints error if fd=-1 and exit
fd=open("/dev/ttyS3", O_RDWR | O_NOCTTY | O_NDELAY |O_NONBLOCK);
if (fd<0)
      {
      fprintf(stderr," open_port: unable to open port - %s
      \n",strerror(errno));
      exit(1);
      }

//~~~Initialize new terminal handler
bzero(&newtio, sizeof(newtio));
fcntl(fd, F_SETFL, 0);//FNDELAY);/*configure port reading, needed for
raw data, will return 0 if no char read into the serial buffer*/

//~~~POSIX control mode flags
newtio.c_cflag |= (B115200| CLOCAL | CREAD); //Set baud rate to
115.2kbps, set local line and enable receiver
newtio.c_cflag &= ~CRTSCTS; //disable hardware flow control

//~~~mask the character size to 8 bits, no parity (8N1)
newtio.c_cflag &= ~PARENB; //disable parity bit
newtio.c_cflag &= ~CSTOPB; //select 1 stop bit
newtio.c_cflag &= ~CSIZE; //disable bit mask for data bits
newtio.c_cflag |= CS8; //select 8 data bits
```

```
//~~~POSIX input mode flags
newtio.c_iflag &= ~(IXON|IXOFF|IXANY); //disable software flow control
(outgoing and incoming), does not allow any character to start flow
again
newtio.c_iflag |=ICRNL; //map CR to NL

//~~~POSIX output mode flags
newtio.c_oflag &= ~OPOST; // set for raw output;

//~~~POSIX local mode flags
newtio.c_lflag &= ~(ICANON | ECHO |ECHOE | ISIG); //Enable data to be
processed as raw input
newtio.c_lflag = 0;

tcsetattr(fd, TCSANOW, &newtio); //set the new options for the port now
return fd;
}
```

Programming the accelerometer

The accelerometer, *MDS302* is manufactured by *Mechworks Systems Inc*. It is a *3-axis* digital accelerometer that can measure acceleration in *x*, *y* and *z* directions. It comes with built-in filter and amplifier circuitaries and is powered through 9V battery. These features help for the application that requires measuring the signal in real-time and lacks of access to the power outlet. In addition, the accelerometer has a sensing range of +/-2*g*, and communicates through a serial interface, *RS-232*, that takes a character format as described below,

Baud rate: 115.2kbps
Parity: None
Data bits: 8
Stop bits: 1

The following table outlines the signals for the *female DB-9* of *RS-232* connector.

Signal	Pin	Name	Direction
Rx	2	Received data	In to PC
Tx	3	Transmitted data	Out from PC
Gnd	5	Ground	Not available

Data transmission – PC (i.e., onboard Linux box) to Accelerometer

The accelerometer sensor is assigned by a unique address. (i.e., default address is 0). As part of the data transmission sequence, the address is the first byte sent by the PC to the accelerometer. The accelerometer then read this first byte of a sequence of bytes and if the address matches, the sequence continues otherwise is ignored. Every setting to initialize and to control the communication between the PC and the accelerometer is defined and saved in the control byte array. Each array is composed of two bytes; the data byte (carry the setting and the function) and the index byte (contain index value of the

data byte). In order to end a particular data transmission, the end byte is sent to end the sequence. The order of the control byte transmission is given as follow,

address → index1 → data1 → index2 → data2 → … → … → end byte

In the following we list down the description of each control byte,

1. Mode Byte – Determines the mode the accelerometer is to perform in.
2. Function Byte – Determines the function the accelerometer is to perform.
3. Channel Selection Byte – Determines which channel(s) the accelerometer is to send
4. Sampling Unit Byte – Determines the unit at which the data sampling is to perform.
5. Sampling Rate Upper Byte – Determines the sampling rate of the analog to digital.
6. Sampling Rate Middle Byte
7. Sampling Rate Lower Byte
8. Collect Data Size Upper Byte – Determines the total number of samples to be collected.
9. Collect Data Size Middle Byte
10. Collect Data Size Lower Byte
11. Bandwidth Upper Byte – Determines the bandwidth of the low-pass filter (in Hz).
12. Bandwidth Lower Byte
13. Gain Upper Byte – Determines the gain factor of the digital signal conditioner.
14. Gain Lower Byte
15. Address Byte – Determines the address of the accelerometer unit.

There are some special bytes that do not composed of data and index bytes, instead, to transmit this byte, we just need to transmit the bytes in the following form,

address byte → special byte

As given below is the list of these special bytes with their descriptions,

1. End Byte (0xFF) – Signals to the accelerometer the end of the transmission sequence.
2. Go Byte (0xFE) – Used as part of the calibration function to signal to the accelerometer package that the prescribed task has been completed.
3. Stop Byte (0xFD) – Signals to the accelerometer package to stop its current action.
4. Reset Byte (0xFB) – Signals to the accelerometer package to reset its Address Byte to the default value of 0.
5. Connect Byte (0xFA) – Signals to the accelerometer package to send back the prescribed connect information.
6. Send Available Data Byte (0xF9) – Signals to the accelerometer package to send back all collected data.
7. Assert Calibration Values Byte (0xF8) – Signals to the accelerometer package to rewrite the saved calibration values with those sent.

Data transmission – Accelerometer to PC

The data transmitted from the accelerometer to the PC is governed by the commands sent from PC. Due to the master-slave relationship, the accelerometer won't transmit any unsolicited data. In addition, each command produces a fixed number of bytes based on the current setting and the byte structure differs for each command.

The accelerometer unit *MDS302* comes with its own standard software to process the data called *MechManager 3.1*. It has an excellent Graphical User Interface (GUI) that can visualize the data in a number of formats. Due to our requirement to access the data

in more flexible manner, we write our own code to initialize and control the communication process between the accelerometer and the PC. In the following, we present the program written in *C*-language.

```
//~~~~~The main program~~~~~
int main(int argc, char **argv)
{
//~~~We have to initialize the serial port first

//→ Here is where the program of serial port goes ←

//~~~Initialize special and control bytes
unsigned char specialbyte[4];
unsigned char ctrbyte[36];
unsigned char
accpara[27]={0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0};
specialbyte[0]=0;                          //Address byte
specialbyte[1]=0xFD;                       //Stop all acc. activity
specialbyte[2]=0;                          //Address byte
specialbyte[3]=0xFA;                       //Connect to acc.

ctrbyte[0]=0x00;                           //Address bytes
ctrbyte[1]=0x80;                           //Mode byte
ctrbyte[2]=0x01;                           //x01:continuous, x02:collect
ctrbyte[3]=0x81;                           //Fct. byte
ctrbyte[4]=0;
ctrbyte[5]=0x82;                           //Channel select byte
ctrbyte[6]=0x03;                           //Select channel x and y
ctrbyte[7]=0x83;                           //Sample unit byte
ctrbyte[8]=0x01;                           //x01: in seconds
ctrbyte[9]=0x84;                           //Sample rate upper byte
ctrbyte[10]=0;
ctrbyte[11]=0x85;                          //Sample rate middle byte
ctrbyte[12]=0;
ctrbyte[13]=0x86;                          //Sample rate lower byte
ctrbyte[14]=0x32;                          //100Hz:0x64, 50Hz:0x32
ctrbyte[15]=0x87;                          //Unused
ctrbyte[16]=0;
ctrbyte[17]=0x88;                          //Collect data size upp. byte
ctrbyte[18]=0;
ctrbyte[19]=0x89;                          //Collect data size mid byte
ctrbyte[20]=0x07;
ctrbyte[21]=0x8A;                          //Collect data size low. byte
ctrbyte[22]=0x68;                          //This is set in collect mode
ctrbyte[23]=0x8B;                          //Bandwidth upp. byte
ctrbyte[24]=0;
ctrbyte[25]=0x8C;                          //Bandwidth low. byte
ctrbyte[26]=0x32;                          //BW x64:100Hz, x32:50Hz
ctrbyte[27]=0x8D;                          //Gain upp. byte
ctrbyte[28]=0x03;
ctrbyte[29]=0x8E;                          //Gain low. byte
```

```
ctrbyte[30]=0x04;                          //Gain=1
ctrbyte[31]=0x8F;                          //not used
ctrbyte[32]=0;
ctrbyte[33]=0x90;                          //not used
ctrbyte[34]=0;
ctrbyte[35]=0xFF;                          //End byte

//~~~Start communication by sending connect function
write(mainfd,&specialbyte[0],1);
write(mainfd,&specialbyte[1],1);      //stop all accelerometer activity
usleep(500000);     //required to give time to accelerometer to respond
write(mainfd,&specialbyte[2],1);
write(mainfd,&specialbyte[3],1);      //connect to accelerometer
usleep(500000);

//~~~Read the respond from accelerometer using port descriptor.
//~~~Accelerometer will report its current configuration
read(mainfd, &accpara, 27);
int j;
for (j=0; j<=26; j++){printf("%d, %X\n",j, accpara[j]);}
printf("~~~~~~~~~~~~~~~ \n");

//~~~Transmit control bytes
write(mainfd,&ctrbyte[1],1);
write(mainfd,&ctrbyte[2],1);
write(mainfd,&ctrbyte[3],1);
write(mainfd,&ctrbyte[4],1);
write(mainfd,&ctrbyte[5],1);
write(mainfd,&ctrbyte[6],1);
write(mainfd,&ctrbyte[7],1);
write(mainfd,&ctrbyte[8],1);
write(mainfd,&ctrbyte[9],1);
write(mainfd,&ctrbyte[10],1);
write(mainfd,&ctrbyte[11],1);
write(mainfd,&ctrbyte[12],1);
write(mainfd,&ctrbyte[13],1);
write(mainfd,&ctrbyte[14],1);
write(mainfd,&ctrbyte[15],1);
write(mainfd,&ctrbyte[16],1);
write(mainfd,&ctrbyte[17],1);
write(mainfd,&ctrbyte[18],1);
write(mainfd,&ctrbyte[19],1);
write(mainfd,&ctrbyte[20],1);
write(mainfd,&ctrbyte[21],1);
write(mainfd,&ctrbyte[22],1);
write(mainfd,&ctrbyte[23],1);
write(mainfd,&ctrbyte[24],1);
write(mainfd,&ctrbyte[25],1);
write(mainfd,&ctrbyte[26],1);
write(mainfd,&ctrbyte[27],1);
write(mainfd,&ctrbyte[28],1);
write(mainfd,&ctrbyte[29],1);
write(mainfd,&ctrbyte[30],1);
write(mainfd,&ctrbyte[31],1);
write(mainfd,&ctrbyte[32],1);
write(mainfd,&ctrbyte[33],1);
write(mainfd,&ctrbyte[34],1);
```

```
write(mainfd,&ctrbyte[35], 1);
usleep(500000);

//~~~We have to initialize the Pioneer here
//→ Here is where the program goes ←

//~~~ Now ready to read and process the data
//~~~ You can do the process in separate thread (i.e., DataCollect)

//~~~ Once done, stop all accelerometer activity
write(mainfd,&specialbyte[0],1);
write(mainfd,&specialbyte[1],1);
usleep(500000);
printf("END \n");

//~~~We have to close the Aria properly
//→ Here is where the codes goes ←

}

//~~~Data collect function
//~~~Here is where the data polling start
//~~~You have to make sure your data is kept being polled to avoid
//~~~accumulation in the accelerometer memory buffer. This is made to
//~~~ensure your data is the current one, which is important for real-
//~~~time application

void DataCollect(void)
{
double slipangleLeftrad,slipangleRightrad, slipangleLeftdeg,
slipangleRightdeg;
double slipangleLeftradave,slipangleRightradave, slipangleLeftdegave,
slipangleRightdegave;
double velwheelRight, velwheelLeft;

//~~~Data obtained from acc. calibration (get this from GUI)
double gvolx=1.7145;
double gvoly=1.534;
double sensitivityx=181;
double sensitivityy=181;

//~~~Initilized all variable to process data
int ptr, gain =1;
double zeroGx = (gvolx/3.3)*4095;
double zeroGy = (gvoly/3.3)*4095;

int constant=4095;
int invaccelx[buffersize/5], invaccely[buffersize/5];
double theresultx[buffersize/5],velx[buffersize/5],
theresulty[buffersize/5],vely[buffersize/5];
double  resultmx[buffersize/5],  resultmy[buffersize/5];
int upnumx[buffersize/5], lownumx[buffersize/5];
int upnumy[buffersize/5], lownumy[buffersize/5];
int finalnumx[buffersize/5], finalnumy[buffersize/5];

unsigned char count[buffersize/5];
```

```
unsigned char sbyte[2];
sbyte[0]=0;
sbyte[1]=0xF9;  //Send available data
int readresult;

//~~~Request data from sensor
write(mainfd,&sbyte[0], 1);
write(mainfd,&sbyte[1], 1);
usleep(30000);
readresult=read(mainfd, &data_in,buffersize);//read data into PC buffer

//~~~Now for this application, this function is called for every 100ms
//~~~so, must make sure the number of data to be processes is
//~~~suitable, so that the processing time won't exceed 100ms
int k;
nt kk=0;
for (k=0; k<buffersize-5; k=k+w)
{
count[kk]=data_in[k];
if(count[kk]>0xBF )
{} //again precaution for invalid data //break from loop and wait for
read again
else if (count[kk]<0x80)
{}
else if(readresult==0)
{}
else {
w=5;
upnumx[kk]=data_in[k+1];                        //upper byte for x data
lownumx[kk]=data_in[k+2];                       //lower byte for x data
upnumy[kk]=data_in[k+3];                        //upper byte for y data
lownumy[kk]=data_in[k+4];                       //lower byte for y data

//~~~Merge two 6 bits data
upnumx[kk] = ((upnumx[kk]&0x003F)<<6);// & 0x3FA0;
lownumx[kk] = lownumx[kk] & 0x003F;
upnumy[kk] = ((upnumy[kk]&0x003F)<<6);// & 0x3FA0;
lownumy[kk] = lownumy[kk] & 0x003F;
finalnumx[kk] = upnumx[kk]|lownumx[kk];
finalnumy[kk] = upnumy[kk]|lownumy[kk];

//~~~Resultant data is in Volt unit.
//~~~Data conversion to mm/s2 or m/s2
invaccelx[kk] = abs(constant-(int)finalnumx[kk]);//Supposed to be int
invaccely[kk] = abs(constant-(int)finalnumy[kk]);//Supposed to be int

theresultx[kk] = (invaccelx[kk] - zeroGx)*gain +zeroGx;
theresulty[kk] = (invaccely[kk] - zeroGy)*gain +zeroGy;

if(theresultx[kk] > 4095)                       //to set threshold value
theresultx[kk] = 4095;
if(theresultx[kk] < 0)
theresultx[kk] = 0;
if(theresulty[kk] > 4095)
theresulty[kk] = 4095;
if(theresulty[kk] < 0)
theresulty[kk] = 0;
```

```
//~~~Formula in mm/s2
//resultmm=((theresult*3.3)/4095-gvol)*1e6*(9.81/sensitivity);

//~~~Formula in m/s2
resultmx[kk]=((theresultx[kk]*3.3)/4095-gvolx)*1e3*(9.81/sensitivityx);
resultmy[kk]=((theresulty[kk]*3.3)/4095-gvoly)*1e3*(9.81/sensitivityy);

//~~~To find velocity value, use integration method - sampling freq.
// This is not required.
//if (resultmy[kk]<0.005 & resultmy[kk]>-0.005) resultmy[kk]=0;
//if (resultmx[kk]<0.005 & resultmx[kk]>-0.005) resultmx[kk]=0;
if (start.mSecSince() < 6150)
resultmy[kk]=resultmy[kk]/10;

velx[kk]=(resultmx[kk]/50)+veltempx;  //divide by sampling rate
veltempx=velx[kk];
vely[kk]=(resultmy[kk]/50)+veltempy;  //divide by sampling rate
if (start.mSecSince()>8240 && resultmy[kk]< 0)
{
veltempy=fabs(vely[kk]/1.2);
}
else
veltempy=vely[kk]; //don't have to put fabs since profile is ok

//printf("Accx = %e , Velx= %e, Accy= %e , Vely= %e\n",resultmx[kk],
velx[kk],resultmy[kk], vely[kk]);

velwheelRight=robot.getRightVel(); //i assume here, traction will be
the same for right as well
velwheelLeft=robot.getLeftVel();
if (velwheelRight==0)
{slipangleRightrad=0;
//slipangleRightradave=0;
}
else
{
 slipangleRightrad=atan(vely[kk]/fabs(velwheelRight*1e-3)); //only on
the right wheel
//slipangleRightrad=atan(vely[kk]/fab(velx[kk])); //only on the right
wheel
//slipangleRightrad=atan(vely[kk]/fabs(velwheelRight));
//slipangleRightradave=atan(velyaverage/fabs(velwheelRight));
}

if (velwheelLeft==0)
{slipangleLeftrad=0;
}
else
{slipangleLeftrad=atan(vely[kk]/fabs(velwheelLeft*1e-3));
}
//~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
slipangleRightdeg=((slipangleRightrad/1)*180)/Pi;
slipangleRight=slipangleRightdeg; //last value from loop will be used
globally

slipangleLeftdeg=(slipangleLeftrad*180)/Pi;
```

```
slipangleLeft=slipangleLeftdeg; //last value from loop will be used
globally

w=5;

fprintf(pAccelero," %e, %e, %e, %e ,  %ld, %e, %e, %e\n", velx[kk],
resultmx[kk], vely[kk],resultmy[kk], start.mSecSince(), hf,
slipangleRight, fytot);
printf("%d, %x, %e, %e, %e, %e ,  %ld\n",kk,count[kk],
velx[kk],resultmx[kk],  vely[kk],resultmy[kk],start.mSecSince());

//printf("Count = %x , Vely= %e ,Slipangle = %e, Coef = %e , time =
%ld\n", count[kk], vely[kk],slipangle,start.mSecSince());

kk++;

}//close brace for 'if'
} //close brace for 'for'
 }
```

Programming *Pioneer P3DX* using ARIA

Advanced Robotics Interface Applications (ARIA) is a very useful interface to *Pioneer P3DX* used for experimental and simulation works of this research. It runs under *Linux* or *Win32* operating system and is written in *C++* language based entirely on object oriented paradigm as an Application Programming Interface (API). It communicates with the WMR via a client/server relationship using TCP/IP connection and run in multi-threaded allowing us to create a dedicated thread to process the data from the accelerometer in real time. We utilize ARIA action-based control (i.e., high level control) features in our programming to control the actions of our WMR. These built-in actions are all resolved strictly based on priority of the threads which can be preset beforehand. We found this capability works best for our application since we have several threads that need to be run simultaneously with different level of priorities. In the following, we present the codes to initialize the WMR with some predefined actions.

```
//~~~The main program
int main(int argc, char **argv)
{

//~~~Robot initialization
//~~~One way to do multithread is by using ArGlobarFunctor

ArGlobalFunctor DataCollectCB(&DataCollect); //Thread to collect data
ArGlobalFunctor DataPrintCB(&DataPrint);    //Thread to do printing job
ArGlobalFunctor AlterSpeedAS(&AlterSpeed);   //Thread to regulate speed
ArGlobalFunctor TractionForceTF(&TractionForce); //Thread to measure
traction force

ArTime start;                              //time struct
ArKeyHandler keyHandler;                   //keyboard handler

Aria::init();                              //mandatory initialization

//define the connection
ArSimpleConnector connector(&argc, argv);
   if (!connector.parseArgs() || argc > 1){
       connector.logOptions();
       exit(1);
```

```
       }

//open the connection
   if (!connector.connectRobot(&robot)){
       printf("Could not connect to robot... exiting\n");
       Aria::shutdown();
       return 1;
       }
Aria::setKeyHandler(&keyHandler);    //let global Aria stuff knows
about it
robot.attachKeyHandler(&keyHandler);   //toss it to robot


//define actions
robot.setTransVelMax(1600); //Max is 2200
robot.setTransAccel(300);   //If not set, value from the flash will be
used (this is the last value used)
robot.setTransDecel(300);


// Add the tasks to robot based on priority, every action is called
every 100ms

robot.addUserTask("TractionForce",90,&TractionForceTF);
robot.addUserTask("DataCollect",85,&DataCollectCB);
robot.addUserTask("AlterSpeed",80,&AlterSpeedAS);
robot.addUserTask("DataPrint",75,&DataPrintCB); //the synchronous tasks
get called every robot cycle, every 100ms by default

start.setToNow();           //Timer start
robot.comInt(ArCommands::ENABLE, 1);   //Turn ON the motor
robot.run(true);            //Start the robot, TRUE means that if lose
connection 'ru' will stops
robot.waitForRunExit();

//~~~now exit
Aria::exit(0);
return 0;
}
```

REFERENCES

Angelova A., Matthies L., Helmick D.M., Sibley G., Perona P. (2006), Learning to predict slip for ground robots*, Proc. of IEEE Int. Conf. on Robotics and Automation*, 3324-3331

Baffet G., Charara A., Stephant J. (2006), Sideslip angle, lateral tire force and road fricion estimation in simulations and experiments, *Proc. of IEEE Int. Conf. on Control Applications,* 903-908

Bakker E., Nyborg L., Pacejka H.B. (1987), Tire modeling for the use of the vehicle dynamics studies, *SAE paper*, 870421

Balakrishna R., Ghosal A. (1995), Modeling of slip for wheeled mobile robot, *IEEE Trans. on Robotics and Automation*, **11**(**1**), 126-132

Brooks C.A., Iagnemma K. (2005), Vibration-based terrain classification for planetary exploration rovers, *IEEE Trans. on Robotics and Automation,* **21**(**6**), 1185 – 1191

Campion G., Bastin G., & D'Andrea-Novel B, (Feb. 1996), Structural Properties and Classification of Kinematic and Dynamic Models of Wheel Mobile Robots", *IEEE Trans. on Robotics and Automation*, **12**(**1**), 47-62

Chung W., Kim S., & Choi J. (May 2006), High speed navigation of a mobile robot based on experiences**,** *Proc. of the 2006 JSME Annual Conf. on Robotics and Mechatronics***,** 799-802

Conceicao A.S., Oliveira H.P., Sousa e Silva A., Oliveira D., Moreira A.P. (June 2007). A nonlinear model predictive control of an omni-directional mobile robot, *IEEE Int. Symposium on Industrial Electronics ISIE07* 2161 – 2166

Craig J. J (2004), Introduction to Robotics: Mechanics and Control, Prentice Hall; 3rd ed.

D'Souza A.F. & Garg V.K. (1984), Advanced dynamics; Modeling and analysis, Prentice Hall

Das T, Kar I.N. (May 2006), Design and implementation of an adaptive fuzzy logic-based controller for wheeled mobile robots, *IEEE Trans. on Control System Technology*, **14**(**3**), 501-510

Der-Chen L. & Wen-Ching C. (Oct. 2006), Control design for vehicle's lateral dynamics, *IEEE Int. Conf. on Systems, Man and Cybernetics, ICSMC '06*, **3**, 2081 – 2086

Dixon W.E., Dawson D.M., and Zergeroglu E. (2000), Robust control of a mobile robot system with kinematic disturbance, *IEEE Int. Conference on Control Applications*, 437-442

Dubins L.E. (1957), On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents, *American Journal of Mathematics*, **79**, 497-516

Eghtesad M., Necsulescu D.S. (2006), Study of the internal dynamics of an autonomous mobile robot, *Robotics and Autonomous Systems,* **54**, 342-349

Fierro R., Lewis F.L. (Dec. 1995), Control of a nonholonomic mobile robot: backstepping kinematics into dynamics", Proc. of the 34th Conf. on Decision and Control, 3805-3810

Fierro R. & Lewis F.L. (July 1998), Control of a nonholonomic mobile robot using neural networks, *IEEE Trans. On Neural Network*, **9**(**4**), 589-600

Germann M., Wurtenberger A. & Daiss A. (Aug. 1994), Monitoring of the friction coefficient between tyre and road surface, *Proc. of the Third IEEE Conf. on Control Applications*, **1**, 613 – 618, 24-26

Helmick D.M., Cheng Y., Clouse D.S., Bajracharya M., Matthies L.H. (2005), Slip compensation for a Mars rover, *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS05*, 2806 – 2813

Helmick D.M., Cheng Y.., Clouse D.S., Matthies L.H., Roumeliotis S.I. (2005), Path following using visual odometry for a Mars rover in high-slip environments, *Proc. IEEE Aerospace Conference*, **2**, 2772 – 789

Ishigami G., Nagatani K., Yoshida K. (2007), Path planning for planetary exploration rovers and its evaluation based on wheel slip dynamics, *Proc. of IEEE Int. Conf. on Robotics and Automation*

Jung S., Hsia T.C. (2005), Explicit lateral force control of an autonomous mobile robot with slip", *EEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS05*, 388 – 393

Kurfess T.R. (2005), Robotics and automation handbook, CRC Press

Kyung-Ho B. (Sept. 2007), Development of dynamics modeling in the vehicle simulator for road safety analysis, *Annual Conference SICE07*, 649 – 653

Lagerberg A. and Egardt B. (May 2007), Backlash estimation with application to automotive powertrains, *IEEE Trans. on Control Systems Technology*, **15**(**3**), 483 – 493

Li L., Wang F.Y. (2006), Integrated longitudinal and lateral tire/road friction modeling and monitoring for vehicle motion control, *IEEE Trans. on Intelligent Transportation Sys*, **7**(**1**),1-19

Lin W.-S., Chang L.-H., Yang P.-C. (Jan. 2007**),** Adaptive critic anti-slip control of wheeled autonomous robot, *Control Theory & Applications, IET*, **1**(**1**), 51 - 57

Liyong Y., Wei X. (June 2007), An adaptive tracking method for non-holonomic wheeled mobile robots, *Control Conference, CCC07. Chinese*, 801 – 805

Melchiorri C., Tornambe A. (1996), Modeling and Control of Mechanisms and robots, World Scientific Publishing

Motte I. & Campion G.A. (2000), A Slow manifold approach for the control of mobile robots not satisfying the kinematic constraints", *IEEE Trans. on Robotics and Automation*, **16**(**6**), 875-880

Nehmzow, U (2003), Mobile robotics, a practical introduction, *Springer*, 2nd Ed.

Pasterkamp W.R. (1997), The tyre as sensor to estimate friction, Delft University Press

Peasgood, M. (April 2008), A complete and scalable strategy for coordinating multiples robots within roadmaps, *IEEE Transaction on Robotics*, **24**(**2**), 283-292

Politis Z., Probert Smith P.J. (2001), Classification of textured surfaces for robot navigation using continuous transmission frequency-modulated sonar signatures, *Int. Journal of Robotics Research*, **20**(**2**), 107-128

Ray L.R. (1997), Nonlinear tire force estimation and road friction identification: simulation and experiments*, Journal of Automatica*, **33**(**10**), 1819-1833

Salerno A, Angeles, J. (Feb 2007), A new family of two-wheeled mobile robots: modeling and controllability robotics, *IEEE Transactions on Robotics and Automation,* **23**(**1**), 169 – 173

Sarkar N. (August 1993), Control of mechanical systems with rolling contact: Applications to robotics, *PhD Dissertation*, University of Pennsylvania

Sarkar, N., Yun, X., Kumar, V. (Dec. 1993), Dynamic path following: a new control algorithm for mobile robots, Proc. of the 32nd IEEE Conference on Decision and Control, **3**, 2670 – 2675

Sarkar N.,Yun X. and Kumar V. (Feb., 1994), Control of mechanical systems with rolling constraints: Application to the dynamic control of mobile robots, in: Int. *Journal of Robotics Research*, **31**, 55-69

Seyr M., Jakubek S. (Dec. 2006), Proprioceptive navigation, slip Estimation and slip control for autonomous wheeled mobile robot, *IEEE Conf. on Robotics, Automation and Mechatronics*, 1-6

Stonier D., Se-Hyoung C., Sung-Lok C., Kuppuswamy N.S., Jong-Hwan K. (April 2007), Nonlinear slip dynamics for an omniwheel mobile robot platform, *IEEE Int. Conf. on Robotics and Automation,* 2367 – 2372

Tarokh M., McDermott G.J. (2005), Kinematics modeling and analyses of articulated rover, *IEEE Trans. on Robotics*, **21**(**4**), 539-553

Travis W., Bevly D.M. (Sept. 2005), Navigation errors introduced by ground vehicle dynamics, *18th International Technical Meeting of the Satellite Division*, 302-310

URL 1.1: *marsrovers.nasa.gov/*

URL 1.1: www.abprecision.co.uk/

URL 1.3: *www.angelusresearch.com*

URL 1.4: www.prod.sandia.gov/cgi-bin/techlib/access-control.pl/2001/013685.pdf

URL 1.5: playerstage.sourceforge.net

URL 1.6: msdn.microsoft.com/en-us/robotics/default.aspx

URL 1.7: www.cosimir.com

URL 1.8: www.cyberbotics.com

URL 1.9: opensimulator.sourceforge.net

URL 1.10: www.acm.org/crossroads/wikifiles/13-3-R/13-3-2.pdf, 2006

URL 5.1 www.robosoft.com

URL 5.2 www.arrickrobotics.com

URL 5.3 www.activrobots.com

Vandapel N., Huber D.F., Kapuriac A., Hebert M. (2004), Natural terrain classification using 3-D ladar data, *Proc. of IEEE Int. Conf. on Robotics and Automation*, **26**, 5117 – 5122

Verma R., Vecchio D., and Fathy H. (Feb. 2008), Development of a scaled vehicle with longitudinal dynamics of an HMMWV for an ITS testbed, ***IEEE/ASME Trans. on*** *Mechatronics*, **13**(**1**), 46 – 57

Volpe R. (1999), Navigation results from desert field tests of the Rocky 7 Mars rover prototype, *Int. Journal of Robotics Research*, **18**(**7**), pp. 669-683

Ward C.C., Iagnemma K. (2007), Model-based wheel slip detection for outdoor mobile robots", *IEEE Int. Conf. on Robotics and Automation,* 2724 – 2729

Weiss C., Frohlich H., Zell A. (2006), Vibration-based terrain classification using support vector machines", *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 4429 – 4434

Zhou D., Yi J. and Deng X. (Nov. 2007), Motion regulation of redundantly actuated omni-directional wheeled mobile robots with internal force control, *IEEE/RSJ International Conference Intelligent Robots and Systems, IROS07,* 3919 – 3924

Zhu X., Dong G., Hu D., Cai Z. (2006), Robust tracking control of wheeled mobile robots not satisfying nonholonomic constraints, *Proc. of the 6th Int. Conf. on Intelligent Systems Design and Applications, ISDA06*

Zulli R., Fierro R., Cote G., Lewis F.L. (1995), Motion planning and control for non-holonomic mobile robot, *IEEE Proc. of the Int. Symposium on Intelligent Control*, 551-557