Università degli Studi di Genova Roger Williams University

MASTER ERASMUS MUNDUS

EMARO ¨European Master In Advanced Robotics¨

2011/2012

Master Thesis

Presented by

Stefano Mafrica

On 26/10/2012

Advance Modeling of a Skid-Steering Mobile Robot for Remote Telepresence

---

President:	Prof. Giorgio Cannata	(*University of Genoa*)
Evaluators:	Prof. Antonio Sgorbissa	(*University of Genoa*)
	Prof. Silvio Sabatini	(*University of Genoa*)
	Prof. Armando Tacchella	(*University of Genoa*)
Supervisors:	Prof. Matthew Stein	(*Roger Williams University*)
	Prof. Fulvio Mastrogiovanni	(*University of Genoa*)

Abstract

The robot employed, in this project, for telepresence applications is a Pioneer 3-AT Robot, which is a four-wheeled skid-steering mobile robot (SSMR). For SSMRs, the non-holonomic constraint of zero lateral velocity considered for standard differential-driven mobile robots, such as unicycles or car-like robots [1], [2], [3], does not hold and the wheel slip must be taken into account. In particular, the tire/ground dynamics occurring during skidding can produce large amplitude vibrations, which can eventually lead to the robot instability. Although such large vibrations never arise when using only the Pioneer 3-AT Robot platform, they significantly characterize the extended Pioneer 3-AT Robot employed in this project, due to the inertia of its vertical structure. These vibrations lead to an erratic motion during sharp turns, in particular when the robot is swiveling in place, which causes unacceptable disruption of the video stream and destabilizes the structure. As the user needs full maneuverability in small environment, the robot motion must be fully controlled. In order to provide a stabilization control strategy for the robot jerky motion, an advance modeling of SSMRs reproducing the real robot vibrations must be first provided.

In this thesis, a novel three-dimensional dynamic model of SSMRs, including a spring-damper tire model, is presented. Some experimental data acquired from three 1-axis accelerometers and a force sensor are presented and analyzed to characterize the nature of the robot vibrations and the tire reaction forces. On the basis of the experimental data, a spring-mass-damper model separately for tire lateral, longitudinal and vertical reaction force is provided. A dynamic friction model, based on the work proposed in [4], [5], is also provided to include the contribution of the wheel longitudinal slip into the reaction force model. Finally, after identifying the robot geometric and dynamic parameters, the Simulink model implementing the propose three-dimensional skid-steering model is presented and validated by qualitatively comparing the results of the simulation to the data acquired from the accelerometers and the force sensor.

Nothing great in the World has been accomplished
without passion and friends.
G. W. F. Hegel

Acknowledgments

I have been indebted in the preparation of this thesis to my supervisor at Roger Williams University, Professor Matthew Stein, whose patience and kindness, as well as his academic experience, have been invaluable to me. I am extremely grateful to my supervisor at University of Genoa, Professor Fulvio Mastrogiovanni, who helped me in the completion of this thesis in Genoa, and whose proficiency and helpfulness have made this work more complete and fulfilling.
I want also to heartily thank my great friend and colleague Mariella Buttaci, who has accompanied me in this unique experience abroad and without whom this thesis would probably not have existed.

Contents

List of Figures

List of Tables

Introduction

The term telepresence was coined in 1980 in an article [6] published in the science fiction magazine *Omni* by Marvin Minsky, the co-founder of Massachusetts Institute of Technology's AI laboratory. In this article, he painted a picture of people suiting up in sensor-motor jackets to work at their jobs thousands of miles away. He called the remote control tools ¨telepresences¨, which emphasized the idea of remotely ¨being there¨ in such a high fidelity manner that it would seem as though the experience was ¨in person¨. By using his own words, we can fully describe the concept of telepresence as:

> *«You don a comfortable jacket lined with sensors and muscle-like motors. Each motion of your arm, hand, and fingers is reproduced at another place by mobile, mechanical hands. Light, dexterous, and strong, these hands have their own sensors through which you see and feel what is happening. Using this instrument, you can ¨work¨ in another room, in another city, in another country, or on another planet. Your remote presence possesses the strength of a giant or the delicacy of a surgeon. Heat or pain is translated into informative but tolerable sensation. Your dangerous job becomes safe and pleasant.»*

Robotics has entered the telepresence space not only as manipulators in sealed nuclear facilities as envisioned by Minsky [6], but more in general as remotely teleoperated Air, Underwater and Ground Vehicles used for reconnaissance, surveillance or explorations of unknown environments, like the Mars Rovers for space explorations, as well as hazardous duties, like mine rescue and survey or bomb disposal.
Contemporary commercial telepresence robots can be described as embodied video conferencing on wheels. These new telepresence robots provide a physical presence and independent mobility in addition to communication, unlike other video conferencing technologies. Over the last decade, companies such as Anybots, HeadThere, InTouch Health, iRobot, Ro-

boDynamics, VGo Communications, and Willow Garage have produced these robots with the intention for them to be employed in a wide variety of situations ranging from ad-hoc conversations at the office, to patient rounds at medical facilities, to interactive exhibitions as in museums [7] (Figure 1).

(a) (b) (c)

Figure 1: (a) A Mars Rover for space explorations. (b) The Vgo conference robot. (c) The TOURBOT robot at the Byzantine and Christian Museum, Athens.

The deployment of robotic systems able to operate in populated environments is heavily based on advances in a number of enabling technologies that facilitate safe, reliable and effective operation of mobile robots and the execution of assigned tasks. The set of enabling technologies needed to pursue a specific application varies according to the tasks implied by the application as well as the environment in which the robot(s) operate. Still, there exists a set of generic technologies that are essential to most of the applications. Examples are technologies that give rise to standard navigation competences, such as mapping, localization, path planning, and obstacle avoidance [7]. Furthermore, as these robots usually operate in polulated unstructured environments and are controlled through the Web, the following issues can be considered as characterizing for a remote telepresence robot:

- efficient and capable human-robot interface;
- efficient motion command generation;
- localized sensor data acquisition and processing;
- reliable autonomous navigation;
- robust real-time control implementation.

The first aspect involves only the remote user workstation and it represents the human-side for the telepresence through the teleoperated robot (Figure 2). The second one involves

both the user and the robot side, as the high level motion commands are generated from the remote workstation by the user through the interface, like the desired position on the map or the camera motion, while the low level motion commands, like the path planning between the current and the desired position of the robot, are generated directly on-board. Finally, the last three aspects involve only the robot-side and assume particular importance when the telepresence application requires high autonomy.

It is worth to notice that, although all interfaces provide tools to perceive the remote environment to make decisions and to generate commands, an autonomous navigation is always needed because of the limited user perception of the environment, due to limitations of sensor data and control, and unpredictable delays in communication, due to varying network conditions. For this reason, acquiring and processing the sensor data locally allows the robot to take immediate action, thereby providing a tighter closed loop control of the robot. However, the importance of the operator interface does not diminish as level of autonomy increases. Even if a robot is capable of operating autonomously, it still needs to convey to the operator how and what it did during task execution. This is particularly important when the robot encounters problems or fails to complete a task. Thus, as robots become more autonomous, interfaces are used less for control and more for monitoring and diagnosis.

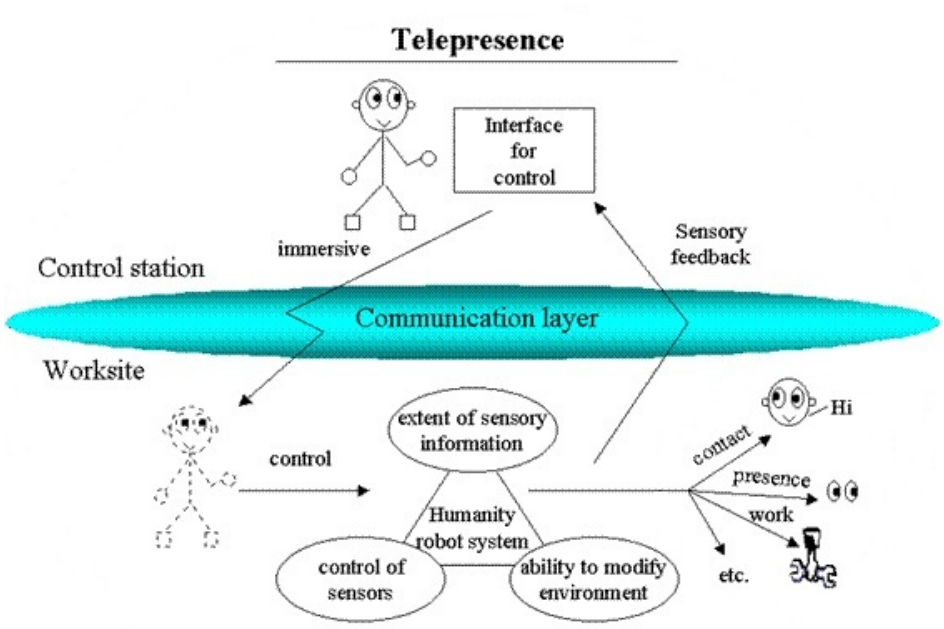


Figure 2: Flowchart representing the telepresence paradigm.

The robot employed, in this project, for telepresence applications is a Pioneer 3-AT Robot, which is a four-wheeled skid-steering mobile robot (SSMR). A SSMR is a mobile robot with no steering mechanism, in which the motion direction is provided by turning the left-side and right-side wheels at different velocities. The absence of a steering system makes the robot mechanically robust and simple for terrain and outdoor environment navigation. However, the non-holonomic constraint of zero lateral velocity considered for standard differential-driven mobile robots, such as unicycles or car-like robots [1], [2], [3], does not hold and the wheel slip must be taken into account. The wheel slip plays an important role in robot dynamics, as the wheel/ground interactions directly provide traction and braking forces that affect the motion stability and maneuverability, and they greatly depend on the wheel slip. Moreover, under certain configurations of the robot dynamic properties, for instance the position of its center of mass (CoM) and the moment of inertia along its principal axes, the tire/-ground dynamics occurring during skidding can produce large amplitude vibrations, which can eventually lead to the robot instability. Although such large vibrations never arise when using only the platform of the Pioneer 3-AT Robot, they suddenly increase when changing its inertia by adding a vertical structure on it, as it happens for the extended Pioneer 3-AT Robot employed in this project. These vibrations lead to an erratic motion during sharp turns, in particular when the robot is swiveling in place, which causes unacceptable disruption of the video stream and destabilizes the structure. Because of the robot destabilization and the video disruption, the remote user can loose the control of the robot for a moment. As the future aim of this project is the telepresence during laboratory class, the user needs full maneuverability in small environment, therefore the robot jerky motion must be fully controlled. In order to provide a stabilization control strategy for the robot jerky motion, an advance modeling of SSMRs, which reproduces the real robot vibrations, must be provided. In this thesis, we develop a three-dimensional dynamic model of SSMRs including a spring-damper tire model, which allows to reproduce the real robot jerky motion in a simulation environment. In particular, a complete description of the mobile robot employed in our project, together with its limitations and issues encountered until now [8], [9], is first provided in Chapter 1. Then, an introduction to the ¨State of the Art¨ of SSMRs modeling is presented in Chapter 2. In this section, a kinematic and dynamic model of a four-wheeled skid-steering mobile robot is presented to characterize the skid-steering properties. In particular, the concept of wheel slipping is presented and elaborated in order to characterize the wheel/ground

interaction at the kinematic level. A wheel/ground friction model, based on the wheel longitudinal slip, is incorporated into the robot dynamic model for both the longitudinal and lateral friction forces [10], [4], [5], [11], [12]. After presenting the kinematic and dynamic model proposed in literature until nowadays, a novel three-dimensional generalized dynamic model for SSMRs is provided by including a three-dimensional non-holonomic constraint and by considering the tire reaction forces as unknown functions.

In Chapter 3, some experimental data acquired from three 1-axis accelerometers and a force sensor are presented and analyzed to characterize the nature of the robot vibrations and the tire reaction forces. Consequently, a spring-mass-damper model separately for tire lateral, longitudinal and vertical reaction force is further discussed in Chapter 4. A dynamic friction model, based on the work proposed in [4], [5], is also provided in this chapter to include the contribution of the wheel longitudinal slip into the reaction force model.

A complete identification of the robot geometric and dynamic parameters is provided in provided in Chapter 5, including the identification of the Roll, Pitch and Yaw dynamics. Finally, the Simulink model of the three-dimensional skid-steering motion reproducing the real system behavior is presented in Chapter 6. In this chapter, also the results of the simulation are presented and compared to the data acquired from the accelerometers and the force sensor, in order to validate the proposed model.

Chapter 1

Hardware and Software

In this section, a description of the hardware used for the development of this thesis is provided. First, we introduce the wheeled mobile robot, the software and the communication protocol employed in this project, with their advantages and disadvantages. Then, we present the ¨offboard¨ sensors we used for the analysis and identification of the robot dynamics.

1.1 The Pioneer 3-AT Robot

Our mobile robot consists of a 4-wheeled skid-steering mobile robot called Pioneer 3-AT (Figure 1.1(a)) available for purchase from Mobile Robots Inc, Amherst, New Hampshire [8]. Factory-supplied capabilities of the robot include individual wheel control and position reporting, plus automatic power monitoring and stall detection. Thus, ¨out of the box¨ one gets a physically rugged four-wheeled platform with good position control of its wheels, but all sensing and navigation are the user's responsibility. The P3-AT is driven by a micro-controller based on a 32-bit Renesas SH2-7144 RISC micro-processor, and integrates an advanced embedded robot control software relying on the client-server paradigm via the Advanced Robot Control and Operations Software (ARCOS). The micro-controller communicates through the serial connection with a user-supplied server computer, and handles the imparted commands at the low level. The low level commands are provided by controlling the angular velocity of the two motors through a servo-control based on the sensory information provided by the wheel's encoders. The two same-side wheels can be, therefore, controlled only in velocity and they spin at the same angular velocity due to the motor belt. ARCOS has also a connection watchdog that halts the robot's motion if communication be-

tween itself and the server is disrupted for a set time interval. As soon as the connection activity is restored ARCOS resumes robot's motion. Our Pioneer 3-AT robot contains three, hot-swappable, seven ampere-hour, 12 volts direct-current (VDC), sealed lead/acid batteries (total of 252 watt-hours) capable of running the robot continuously with a range between two and three hours, even if due to usage at the end battery life decreased of $20\% - 30\%$.
The platform has been extended for interior, human filled, unstructured spaces by adding the following hardware components (Figure 1.1(b)):

- *Lenovo PowerPC laptop* with extended battery (almost four hours expiration)
- lightweight extruded aluminum framing to position the laptop at waist height relative to standing humans
- *Logitech USB camera* with Pan Tilt Zoom capabilities ($pan \in [-90^o, +90^o]$ and $tilt \in [-30^o, +30^o]$) positioned at chest height relative to standing humans
- horizontal wheels and bumpers designed to allow safe operation in the target environment.

The laptop is connected to the robot via an USB serial connection whose end is plugged into the RS-232 port. The laptop runs the client side of the ARCOS that waits and forwards commands to the robot via serial communication. The laptop also runs custom developed software that waits and forwards commands to control the camera parameters.
In the previous thesis [9], an iPhone running the AlterEgo application was employed for the remote user workstation. AlterEgo is an Objective-C/C application to interpret gestures performed by an operator. It sends a TCP packets through the Internet to reach the client side of ARCOS which is waiting for that packet. These packets contain commands for either the mobile robot or for the pan-tilt camera. After error checking, the laptop computer will forward the commands either to the mobile robot or to the camera for implementation.
¨Safe operation¨ in the target environment is defined by the inability of the robot to damage itself or its environment during operation. It specifically does not imply that the robot never contacts objects in its environment other than the floor. The robot has a parallel port whose pins trigger some behaviors, and in particular we use those pins to temporarily stopping robot motion. The robot is equipped with a front and a rear bumper. When the robot frontally or posteriorly hits an object straight on the robot temporarily stopped. Conversely,

four horizontal side wheels are also installed at the platform level such that when the robot obliquely encounters a wall they rotated against wall's surface allowing the robot to continue its motion smoothly. For high obstacles a pressure sensitive switch is installed for the entire aluminum frame. Thus, if the robot encounters a door handle, for example, the robot will stop. During testing, this system has been found to be largely effective but not perfect. There were a few objects in the environment that could defeat these features and cause the robot to overturn itself, potentially incurring damage. In addition, some difficulties have been encountered with the safety features being activated during normal operation due to the jerky motion of the skid-steering. Furthermore, jerky motion shakes the camera off the Velcro and it disrupts the image and adds delay while the operator waits for the camera view to settle down. This jerky motion is often due to the fact that telepresence operators tend to swivel the robot in place because that corresponds to ¨looking around¨, which is a very natural action for humans but it is not an easy one for the robot. Thus, the main aim of our future work will be reducing the robot jerky motion and large-scale vibrations during sharp turns and especially while swiveling in place.

(a) (b)

Figure 1.1: (a) The *Pioneer 3-AT* robot; (b) the extended *Pioneer 3-AT* robot.

1.2 The sensors

For this project, we employed three `SCA620-CF8H1A 1-AXIS ANALOG ACCELEROMETERS` [13]. The main characteristics are: power supply $V_{dd} = +5\,V$, voltage output $V_{out} \in [-0.3, V_{dd}+0.3]\,V$, acceleration range $a \in [-1.7g, 1.7g]$, sensitivity error $e_a = [-3, +3]\%$ and low-pass bandwidth, in laboratory conditions, at $80\,Hz$ [13].

In order to characterize the robot vibrations as well as identify some dynamic parameters, each accelerometer was welded to one face of a small cube of wood, so that the accelerometer's acceleration coincides with the cube's acceleration. They were also fixed in such a way that their axes intersect in one point, the origin of the cube frame, and their positive directions satisfy the right-hand rule (Figure 1.2).

The data are acquired from the accelerometers as Voltage signals by using the NI-DAQmx device, as data acquisition device (DAQ), and LabVIEW software, both produced by National Instruments Co.

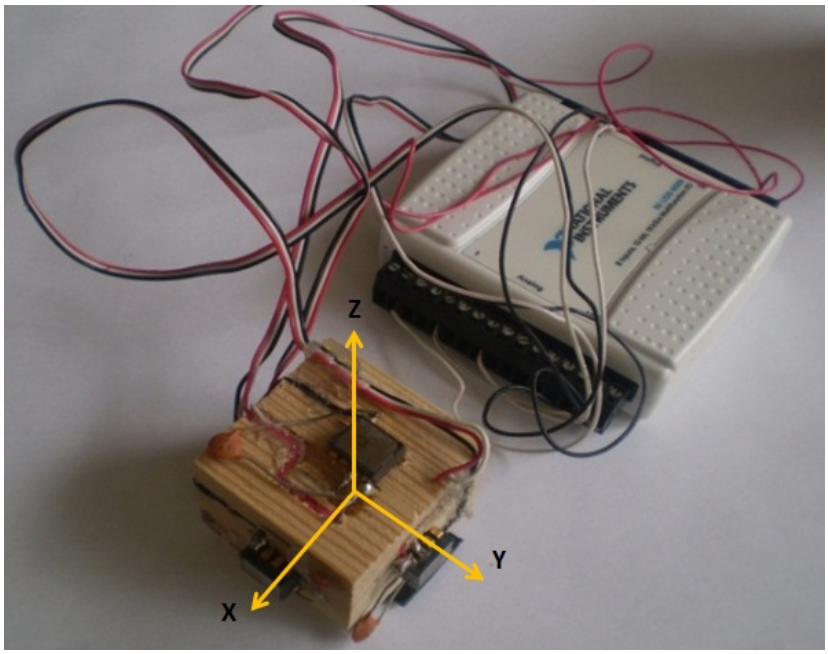


Figure 1.2: Picture of the wood-cube with the three 1-axis analog accelerometers and their axes.

Also, the `eXpert 2653 ADMET` universal testing machine [14] in the mechanics laboratory at Roger Williams University was utilized to perform tests for the identification of the tire reaction forces. The testing machine is capable of performing compression, tension, and flexure tests, thanks to its axial alignment, stiffness and cross-head guidance. The main system specifications are: load capacity $50\,kN$, speed range $[0.0025, 508]\,\frac{mm}{min}$ and position

control resolution 0.14 μm. It is also equipped with a force sensor with a sensitivity of 1 mN, and two closed loop servo controllers, both feature 8 kHz servo update frequency and programmable log rates to 1 kHz [14].

The data are acquired by using the `MTESTQuattro` software, provided by the same company.

Figure 1.3: Picture of the eXpert 2653 ADMET universal testing machine.

Chapter 2

Modeling of Skid-Steering Mobile Robots

Skid-steering mobile robots (SSMRs) have been widely used in many applications, such as terrain navigation and exploration, waste management, defense, security, and household services. For a skid-steered robot, there is no steering mechanism and motion direction is provided by turning the left-side and right-side wheels at different velocities. The absence of a steering system makes the robot mechanically robust and simple for terrain or outdoor environment navigation. Due to the varying tire/ground interactions and over-constrained contact, it is quite challenging to obtain accurate kinematic and dynamic models, and therefore robust tracking control scheme, for such mobile robots. Although there is a great deal of research on dynamic modeling and tracking control of differential-driven mobile robots that are under the non-holonomic constraint of zero lateral velocity, such as unicycles or car-like robots [1], [2], [3], the counterpart research on skid-steered mobile robots is less frequently reported. However, because of the similarity between skid-steering of tracked and wheeled vehicles, the method of modeling the track/ground interaction for tracked vehicles can be utilized for skid-steered wheeled robots.

Wheel slip plays a critical role in both the kinematic and dynamic modeling of skid-steered mobile robots. The slip information, in fact, provides a connection between the wheel rotation velocity and the linear and angular motion of the robot platform. The wheel slip also plays an important role in robot dynamics, as the wheel/ground interactions directly provide traction and braking forces that affect the motion stability and maneuverability and they greatly depend on the wheel slip. Thus, understanding the slip information is not only important for trajectory control and localization applications, such as dead-reckoning, but it is even more important for stabilization control applications when the roll and pitch motion are

not negligible and robot vibrations are not admissible. For this reason, the state of the art regarding skid-steering kinematic and dynamic modeling will be first presented, introducing the wheel slip, and then the dynamic model will be improved introducing the roll and pitch motion.

2.1 State of the Art

The aim of this section is to present the ¨state of the art¨ of SSMRs kinematic and dynamic modeling up to nowadays. The presented summary will be mainly relying on the works of Song *et al.*, Kozlowsky *et al.*, Defoort *et al.*, Jiang *et al.*, and it will used as base for the generalized modeling developed in the next section.
In literature, the description for the kinematic and dynamic model of 4-wheels skid-steering mobile robots is usually provided for trajectory tracking control and localization applications. For this reason, without loss of generality, the following **Assumptions** are commonly considered:

1. The center of mass of the robot is located at the geometric center of the body frame[1].
2. There is point contact between the wheel and the ground.
3. The contact rolling resistance force is negligible[2].
4. Each side's two wheels rotate at the same speed.
5. The normal forces at the wheel/ground contact points are equally distributed among four wheels during motion.
6. The robot is running on a flat ground surface and four wheels are always in contact with the ground surface.

2.1.1 Kinematic Model

In [10], [4], [5], [11], [12], [15], because of **Assumption** 6, the authors consider the vehicle allowed to move only on a two dimensional plane with inertial coordinate frame (X_g, Y_g), as depicted in Figure 2.1(a). Thereby, they consider, without loss of generality, the generalized coordinate vector and the body velocity vector respectively defined as $\boldsymbol{q} = [X \quad Y \quad \theta]^T$

[1] Similar results could be obtained if the mass center of the robot were located somewhere other than the robot's geometric center.
[2] Since we only consider wheel/ground point contact, the ground resistance force is negligible.

and $\boldsymbol{V} = [v_x \quad v_y \quad \omega_z]^T$, with v_x, v_y, ω_z determining respectively the longitudinal, lateral and angular velocity of the vehicle.

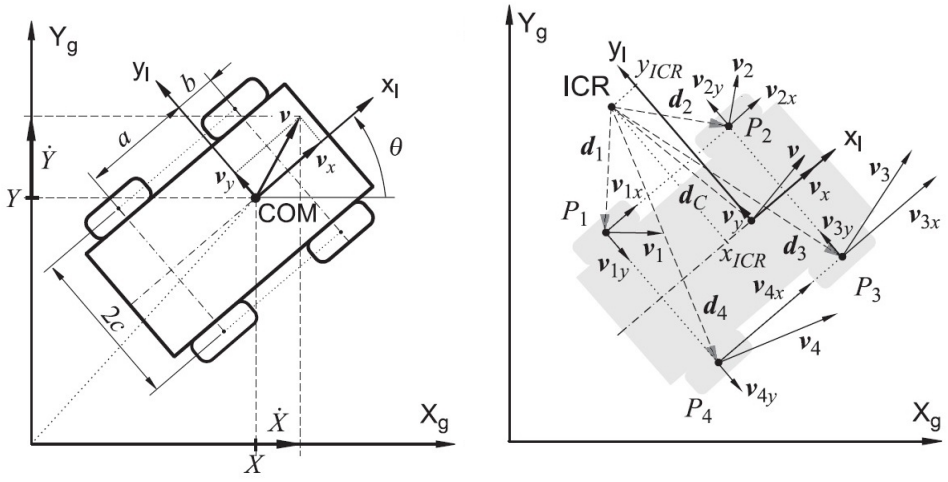


Figure 2.1: (a) SSMR kinematics; (b) Wheel/ground contact point velocities without slipping.

By looking at Figure 2.1(a), it is easy to derive the following kinematic equation giving the velocity constraint between the generalized velocity vector $\dot{\boldsymbol{q}} = [\dot{X} \quad \dot{Y} \quad \dot{\theta}]^T$[3] and the body velocity vector $\boldsymbol{V}$:

$$\dot{\boldsymbol{q}} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \boldsymbol{V} = {}^{g}R_l \boldsymbol{V} \tag{2.1}$$

where ${}^{g}R_l$ is the rotation matrix projecting the local frame (x_l, y_l) onto the inertial frame (X_g, Y_g) in the case of planar motion.

Let $ICR_G = (x_{ICR_G}, y_{ICR_G}, 0)$ denote the instantaneous center of rotation (ICR) of the entire robot expressed on the local frame, which represents the point which the vehicle's CoM rotates around. Accordingly to classical kinematics, in the condition of non longitudinal slipping, we can write the following equations:

$$\begin{aligned} \boldsymbol{v} &= \boldsymbol{d}_C \times \boldsymbol{\omega} \\ \boldsymbol{v}_i &= \boldsymbol{d}_i \times \boldsymbol{\omega} \end{aligned} \tag{2.2}$$

where $\boldsymbol{v} = [v_x \quad v_y \quad v_z]^T$ and $\boldsymbol{\omega} = [\omega_x \quad \omega_y \quad \omega_z]^T$ are respectively the linear and angular

[3]Note that the angular velocity on the inertial frame coincides to $\dot{\theta}$ because we are under the assumption of planar motion

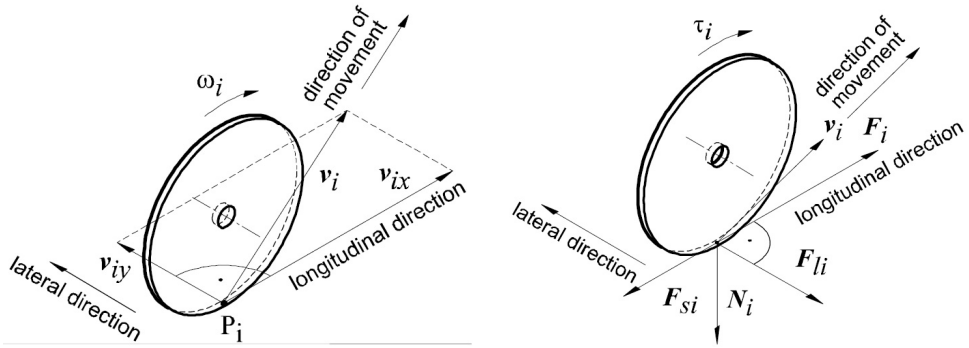
velocity vector for the extended three dimensional coordinate frame, and $\boldsymbol{v}_i = [v_{ix} \quad v_{iy} \quad v_{iz}]^T$ is the three dimensional vector of the wheel center velocity (Figure 2.2(a)). The vectors $\boldsymbol{d}_C = [d_{Cx} \quad d_{Cy} \quad d_{Cz}]^T = -[x_{ICR_G} \quad y_{ICR_G} \quad z_{ICR_G}]^T$ and $\boldsymbol{d}_i = [d_{ix} \quad d_{iy} \quad d_{iz}]^T$, instead, represent respectively the vector from the ICR_G to the CoM and to the i^{th} wheel/ground contact point P_i, expressed in the local frame.

By considering only planar motion, i.e. $d_{Cz}, d_{iz}, v_z, v_{iz}, \omega_x, \omega_y = 0$, equations (2.2) lead to the following relation:

$$\omega_z = -\frac{v_{ix}}{d_{iy}} = -\frac{v_x}{d_{Cy}} = \frac{v_{iy}}{d_{ix}} = \frac{v_y}{d_{Cx}}, \qquad i = 1, \ldots, 4 \tag{2.3}$$

Accordingly to the geometrical description in Figure 2.1, the coordinates of vectors $\boldsymbol{d}_i$ are defined as follows:

$$\begin{aligned} d_{1x} &= d_{4x} = d_{Cx} - a = -x_{ICR_G} - a \\ d_{2x} &= d_{3x} = d_{Cx} + b = -x_{ICR_G} + b \\ d_{1y} &= d_{2y} = d_{Cy} + c = -y_{ICR_G} + c \\ d_{3x} &= d_{4x} = d_{Cy} - c = -y_{ICR_G} - c \end{aligned} \tag{2.4}$$



(a) (b)

Figure 2.2: (a) Wheel linear and angular velocities; (b) Wheel forces and torques.

By combining equation (2.3) and (2.4), the following relations among the wheel velocities are obtained:

$$\begin{aligned} v_L &\triangleq v_{1x} = v_{2x}, \qquad & v_R &\triangleq v_{3x} = v_{4x} \\ v_F &\triangleq v_{2y} = v_{3y}, \qquad & v_B &\triangleq v_{1y} = v_{4y} \end{aligned} \tag{2.5}$$

where v_L, v_R denote respectively the longitudinal coordinates of left and right wheel velocities, while v_F, v_B are the lateral coordinates of front and rear wheels velocities. These velocities can be expressed in matrix form by combing equations (2.5) and (2.3):

$$\begin{bmatrix} v_L \\ v_R \\ v_F \\ v_B \end{bmatrix} = \begin{bmatrix} 1 & -c \\ 1 & c \\ 0 & -x_{ICR_G} + b \\ 0 & -x_{ICR_G} - a \end{bmatrix} \begin{bmatrix} v_x \\ \omega_z \end{bmatrix} \tag{2.6}$$

Although the above relations are often used to obtain the kinematic model of SSMR, the assumption of non longitudinal slipping is highly restrictive for the accuracy of the model as the slipping between the wheels and the ground usually is not negligible.
We define the longitudinal wheel slip λ_i at each wheel as the ratio between the difference of the wheel velocity and its center velocity, and the wheel velocity, namely:

$$\lambda_i = \frac{r\omega_i - v_{ix}}{r\omega_i} = -\frac{\Delta v_{ix}}{r\omega_i}, \qquad i = 1, \ldots, 4 \tag{2.7}$$

where r is the so called effective radius of the wheels, ω_i is the angular velocity of the i^{th} wheel and $\Delta v_{ix} = v_{ix} - r\omega_i$ is also called the longitudinal slip velocity of the i^{th} wheel/ground contact point P_i. We note that $\lambda_1 = \lambda_2 = \lambda_L$ and $\lambda_3 = \lambda_4 = \lambda_R$ as $\omega_1 = \omega_2 = \omega_L$ and $\omega_3 = \omega_4 = \omega_R$, due to **Assumption** 4. It is also observed that under the above definition, $\lambda_i \in [0, 1]$ if the i^{th} wheel is under traction ($\Delta v_{ix} < 0$), and $\lambda_i \in (-\inf, 0]$ if the wheel is under braking ($\Delta v_{ix} > 0$), which is undesirable for uniformly modeling the wheel/ground friction under traction and braking cases. To avoid such a problem, using the same treatment as in [4], we restrict the magnitude of λ_i to a maximum magnitude of 1, i.e. we take $\lambda_i = -1$ if $\lambda_i < -1$ under braking.
Let $ICR_l = (x_{ICR_l}, y_{ICR_l}, 0)$ and $ICR_r = (x_{ICR_r}, y_{ICR_r}, 0)$ denote the instantaneous center of rotation expressed in the local frame of the left-side and right-side wheel/ground contact points, respectively. It is known that ICR_l, ICR_r and ICR_G lie on a line parallel to the y-axis [4], [5], [10], [16] (Figure 2.3). From the aforementioned discussion and equation (2.3), we define the longitudinal ICR location s as follows:

$$s = x_{ICR_l} = x_{ICR_r} = x_{ICR_G} = -\frac{v_y}{\omega_z} \tag{2.8}$$

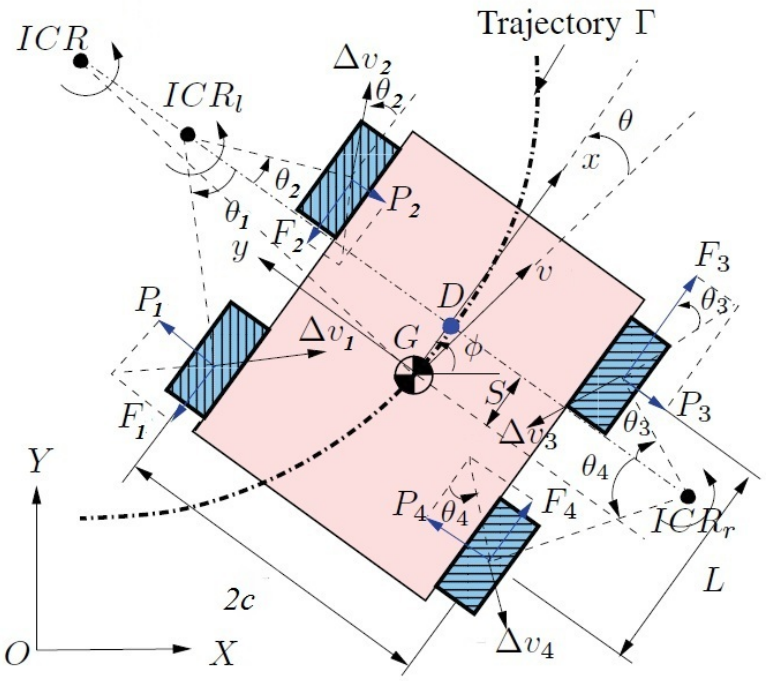


Figure 2.3: Wheel/ground contact point velocities with slipping.

According to the geometrical description in Figure 2.3 and to the given definition of ICR_l, ICR_r, we can write the longitudinal slip velocities as:

$$\begin{aligned} \Delta v_{Lx} &= \Delta v_{1x} = \Delta v_{2x} = (y_{ICR_l} - c)\omega_z \\ \Delta v_{Rx} &= \Delta v_{3x} = \Delta v_{4x} = (y_{ICR_r} + c)\omega_z \end{aligned} \tag{2.9}$$

By combining equation (2.9) and the expressions of v_L, v_R from equation (2.6), we obtain the following definitions for the y-coordinates of the *ICRs*:

$$y_{ICR_l} = \frac{v_x - r\omega_L}{\omega_z}, \qquad y_{ICR_r} = \frac{v_x - r\omega_R}{\omega_z}, \qquad y_{ICR_G} = \frac{v_x}{\omega_z} \tag{2.10}$$

It must be noticed that the coordinate y_{ICR_G} can reach infinite values if the vehicle angular velocity $\omega_z = 0$, namely when the robot moves along a perfect straight line under $\omega_L = \omega_R$. Conversely, the other coordinates y_{ICR_l}, y_{ICR_r} and the s value are bounded. This observation can be explained by the fact that both the numerators and the denominators in (2.8) and (2.10) are infinitesimals of the same order when the $\omega_z \to 0$. Therefore, the values for y_{ICR_l}, y_{ICR_r} and s are well defined at a finite distance when the vehicle is not rotating.

Let ω_L, ω_R be the control inputs at the kinematic level, from the definition of the longitudinal wheel slip in (2.7) we know that $v_{Lx} = r(1 - \lambda_L)\omega_L$ and $v_{Rx} = r(1 - \lambda_R)\omega_R$. By substituting

these expressions for v_{Lx}, v_{Rx} in equation (2.6) and adding/subtracting the first two scalar equations, we obtain the following relation between the wheel angular velocities and the longitudinal and angular velocity of the robot in its matrix form:

$$\boldsymbol{\eta} = \begin{bmatrix} v_x \\ \omega_z \end{bmatrix} = \frac{r}{2} \begin{bmatrix} 1-\lambda_L & 1-\lambda_R \\ -\frac{1-\lambda_L}{c} & \frac{1-\lambda_R}{c} \end{bmatrix} \begin{bmatrix} \omega_L \\ \omega_R \end{bmatrix} \tag{2.11}$$

In order to complete kinematic model of a SSMR, additional velocity constraints should be considered. We can easily obtain the velocity constraint on the local frame from equation (2.8) as:

$$v_y + s\omega_z = 0 \tag{2.12}$$

which can be rewritten with respect to the inertial frame, by using equation (2.1), as:

$$\begin{bmatrix} \sin\theta & \cos(\theta) & 0 \end{bmatrix} \dot{\boldsymbol{q}} + s\dot{\theta} = \begin{bmatrix} \sin\theta & \cos(\theta) & s \end{bmatrix} \dot{\boldsymbol{q}} = A(\boldsymbol{q})\dot{\boldsymbol{q}} = 0 \tag{2.13}$$

It is worth to notice that constraint (2.13) represents a non-holonomic constraint since it is not integrable, therefore our system can be defined as non-holonomic system similarly to all conventional wheel mobile robot.
Furthermore, since $\dot{\boldsymbol{q}}$ belongs to the null space of $A(\boldsymbol{q})$, we can also write:

$$\dot{\boldsymbol{q}} = S(\boldsymbol{q})\boldsymbol{\eta}$$

where the expression of the matrix $S(\boldsymbol{q})$ can be obtained from equation (2.1) by including the constraint (2.12) as follows:

$$\dot{\boldsymbol{q}} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_x \\ -s\omega_z \\ \omega_z \end{bmatrix} = \begin{bmatrix} \cos(\theta) & s\sin(\theta) \\ \sin(\theta) & -s\cos(\theta) \\ 0 & 1 \end{bmatrix} \boldsymbol{\eta} = S(\boldsymbol{q})\boldsymbol{\eta} \tag{2.14}$$

Equation (2.14) provides the constraint on velocity between the generalized velocity vector $\dot{\boldsymbol{q}}$ and the control input $\boldsymbol{\eta}$ and represents our kinematic model of SSMR.
The constraint on acceleration between $\dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}$ and $\boldsymbol{V}, \dot{\boldsymbol{V}}$ can be obtained by differentiating with respect to time equation (2.14):

$$\ddot{\boldsymbol{q}} = \dot{S}(\boldsymbol{q})\boldsymbol{V} + S(\boldsymbol{q})\dot{\boldsymbol{V}}. \tag{2.15}$$

Finally, by combining equation (2.13) with (2.14), it is straightforward to obtain the following condition, which will be useful in the treatment of the dynamic model of a SSMR:

$$S^T(\boldsymbol{q})A^T(\boldsymbol{q}) = 0. \tag{2.16}$$

2.1.2 Dynamic Model

In this subsection, we present the dynamic model of SSMRs presented in [10], [4], [5], [11], [11], [15]. Because of interaction between the wheels and the ground, the dynamic properties of SSMRs play a very important role. It should be noted that, if the robot is changing its orientation, reactive friction forces are usually much higher than forces resulting from inertia. As a consequence, even for relatively low velocities, dynamic properties of SSMRs influence motion much more than for other vehicles for which non-slipping and pure rolling assumption may be satisfied. However, in this section only simplified dynamics of SSMRs, which will be useful for trajectory control purpose, are introduced. In order to simplify this model, we assume that the mass distribution of the vehicle is almost homogeneous, the kinetic energy of the wheels and drives can be neglected, and detailed description of tire which can be found, for example, in [17] are omitted.
The dynamic equation of a SSMR can be obtained using Euler-Lagrange principle with Lagrange multipliers to include non-holonomic constraint. The Euler-Lagrange equation is defined as:

$$\Gamma = \frac{d}{dt}\frac{\partial L}{\partial \dot{\boldsymbol{q}}} - \frac{\partial L}{\partial \boldsymbol{q}} \tag{2.17}$$

where Γ denotes the active torques and forces vector without considering any external force and $L = E - U$ is the Lagrangian of the system defined as the sum of kinetic and potential energy. Due to **Assumption** 6, the gravitational potential energy is $U = 0$, therefore in our case the Lagrangian coincides with the kinetic energy of the robot[4]

$$L = E = \frac{1}{2}(m\boldsymbol{V}^T\boldsymbol{V} + I\omega_z^2) = \frac{1}{2}[m(\dot{X}^2 + \dot{Y}^2) + I\dot{\theta}^2]. \tag{2.18}$$

where m and I represent respectively the mass and the moment of inertia of the robot about the CoM. It must be noticed that **Assumption** 6 limits the motion to flat grounds and it has been made only to simplify the computations, although it is not difficult to generalize

[4]Note that the kinetic energy of the wheels is neglected for a sake of simplicity.

the model by setting $U = mgZ$.

By applying the Euler-Lagrange equation (2.17) to the Lagrangian defined in (2.18), we obtain:

$$\Gamma = \begin{bmatrix} m & 0 & 0 \\ 0 & m & 0 \\ 0 & 0 & I \end{bmatrix} \ddot{\boldsymbol{q}} = M\ddot{\boldsymbol{q}} \tag{2.19}$$

Furthermore, the vector Γ can be calculated by taking into account that the active force $\boldsymbol{F}_{a_i}$ exerted on the i^{th} wheel/ground contact point, expressed on the local frame, is related to the wheel torque τ_i by the following relation:

$$\boldsymbol{F}_{a_i} = \begin{bmatrix} \frac{\tau_i}{r} \\ 0 \end{bmatrix} \tag{2.20}$$

To obtain the total active force and torque, firstly we compute the resultant force on the local frame, by summing all the active forces $\boldsymbol{F}_{a_i} \quad i = 1, \ldots, 4$, and the resultant torque, by summing all the torques provided by each force $\boldsymbol{F}_{a_i}$ with respect to the CoM. Then, we project the resultant force and torque on the inertial frame, obtaining the following expression of Γ:

$$\Gamma = \begin{bmatrix} F_x \\ F_y \\ \tau_z \end{bmatrix} = {}^{g}R_l \begin{bmatrix} \sum_{i=1}^{4} \boldsymbol{F}_{a_i} \\ \sum_{i=1}^{4} \tau_{a_i} \end{bmatrix} = \frac{1}{r} \begin{bmatrix} \cos\theta \sum_{i=1}^{4} \tau_i \\ \sin\theta \sum_{i=1}^{4} \tau_i \\ c(-\tau_1 - \tau_2 + \tau_3 + \tau_4) \end{bmatrix} \tag{2.21}$$

Due to **Assumption** 4, we can simplify the notation of (2.21) by considering the following control input at dynamic level:

$$\boldsymbol{\tau} = \begin{bmatrix} \tau_L \\ \tau_R \end{bmatrix} = \begin{bmatrix} \tau_1 + \tau_2 \\ \tau_3 + \tau_4 \end{bmatrix} \tag{2.22}$$

leading to the following expression of Γ:

$$\Gamma = \frac{1}{r} \begin{bmatrix} \cos\theta & \cos\theta \\ \sin\theta & \sin\theta \\ -c & c \end{bmatrix} \boldsymbol{\tau} = B(\boldsymbol{q})\boldsymbol{\tau} \tag{2.23}$$

In order to complete the dynamic model of a SSMR, we must add the reaction forces and torques to the right term of the dynamic equation in (2.19) and include the non-holonomic

constraint (2.13) by using the Lagrange multipliers.

Let N, N_i be the normal reaction force exerted from the ground surface to the whole vehicle and to each wheel/ground contact point, respectively. According to the geometric description depicted in Figure 2.3 and thank to the symmetry along the longitudinal axis, the following relations can be easily deduced:

$$\begin{aligned} N_1 = N_4 &= \frac{b}{2(a+b)}N = \frac{b}{2(a+b)}mg \\ N_2 = N_3 &= \frac{a}{2(a+b)}N = \frac{a}{2(a+b)}mg \end{aligned} \tag{2.24}$$

Let F_i, P_i denote respectively the longitudinal and lateral reactive force vector at the i^{th} wheel/ground contact point expressed on the local frame (Figure 2.2,2.3). These forces can be thought only as friction forces according to [10]. Friction is difficult to model because of the high non-linearity related to the many variables that contribute to it. For sake of simplicity, the friction can be modeled as a superposition of viscous and Coulomb frictions as:

$$\boldsymbol{F}_{fi} = -\mu_{ci} sign(\Delta \boldsymbol{v}_i) N_i - \mu_{vi} \Delta \boldsymbol{v}_i \qquad i = 1, \ldots, 4 \tag{2.25}$$

where $\boldsymbol{F}_{fi} = [F_i \quad P_i]^T$ is the total friction force on the i^{th} wheel/ground contact point, μ_{ci}, μ_{vi} represent respectively the Coulomb and viscous friction coefficients and $\Delta \boldsymbol{v}_i$ is the wheel slip velocity (Figure 2.3).

As SSMRs are vehicles which usually operate at low velocity, in particular during turns, we can assume that $\mu_{ci} N_i >> |\mu_{vi} \Delta \boldsymbol{v}_i|$. Thereby, the term $-\mu_{vi} \Delta \boldsymbol{v}_i$ in (2.25) can be neglected, leading to simpler model:

$$\boldsymbol{F}_{fi} = -\mu_{ci} sign(\Delta \boldsymbol{v}_i) N_i = \begin{bmatrix} -\mu_{cix} sign(\Delta v_{ix}) N_i \\ -\mu_{ciy} sign(\Delta v_{iy}) N_i \end{bmatrix} \qquad i = 1, \ldots, 4 \tag{2.26}$$

We note that equation (2.26) can't be directly employed for two reasons: first it requires the knowledge of Δv_{iy} which is not directly known, second it does not represent a smooth function since the function $sign(x)$ is discontinuous, and therefore not differentiable, in $x = 0$. In order to overcome the first issue, we note that F_i, P_i are dependent on each other and, in particular, they form a friction force circle, namely $F_i = |\boldsymbol{F}_{fi}| \cos \theta_i$, $P_i = |\boldsymbol{F}_{fi}| \sin \theta_i$, where θ_i is the slip angle at the i^{th} wheel/ground contact point (Figure 2.3). Thereby, we can rewrite the lateral friction force as:

$$P_i = F_i \tan\theta_i \qquad i = 1, \ldots, 4 \tag{2.27}$$

The terms $\tan\theta_i$ can be calculated with respect to the *ICRs* positions according to the geometrical description depicted in Figure 2.3, obtaining:

$$\begin{aligned} \tan\theta_1 &= \frac{a - x_{ICR_l}}{y_{ICR_l} - c} \\ \tan\theta_2 &= \frac{-b - x_{ICR_l}}{y_{ICR_l} - c} \\ \tan\theta_3 &= \frac{-b - x_{ICR_r}}{-y_{ICR_r} - c} \\ \tan\theta_4 &= \frac{a - x_{ICR_r}}{-y_{ICR_r} - c} \end{aligned} \tag{2.28}$$

It must be noticed that the above definitions of $\tan\theta_i$ are always meaningful since it can be proved that the conditions $y_{ICR_l} \neq c, y_{ICR_r} \neq -c$ are always verified [16].
The issue due to the non-smoothness of the term $\mu_{cix} sign(\Delta v_{ix})$ in equation (2.25) is overcome by replacing it with a suitable function $\mu(\lambda_i)$, which represents a reasonable approximation of the *sign*() function. Here we present two approximating functions for $\mu(\lambda_i)$ [9], [4] with their advantages and disadvantages:

$$\mu(\lambda_i) = \mu_{cix}\frac{2}{\pi}\arctan(k_s \Delta v_{ix}) = -\mu_{cix}\frac{2}{\pi}\arctan(k_s r\omega_i \lambda_i) \tag{2.29a}$$

$$\mu(\lambda_i) = \begin{cases} -k_s\lambda_m + \frac{k_s\lambda_m - \mu_s}{1+\lambda_m}(\lambda_i + \lambda_m), & \text{for } \lambda_i \in (-1, -\lambda_m] \\ k_s\lambda_i, & \text{for } \lambda_i \in (-\lambda_m, \lambda_m) \\ k_s\lambda_m - \frac{k_s\lambda_m - \mu_s}{1-\lambda_m}(\lambda_i - \lambda_m), & \text{for } \lambda_i \in [\lambda_m, 1) \end{cases} \tag{2.29b}$$

where k_s is the friction stiffness coefficient, λ_m is the longitudinal slip corresponding to the maximum wheel/ground friction coefficient, and μ_s is the longitudinal wheel/ground sliding friction coefficient.
The function defined in (2.29a) derives directly from the conventional smooth approximation of the *sign*() function [9]. Its advantage is due to the dependency of only two identified parameter (μ_{cix}, k_s), while the disadvantages are the non-linearity of the arctan() function and the absence of an instability region due to the difference between the static and kinetic friction coefficient.
Conversely, the function defined in (2.29b) represents a linear piece-wise function, which

approximates the $\mu - \lambda$ curve obtained by fitting experimental data as it is depicted in Figure 2.4 [4]. As provided in [4], the function in (2.29b) includes the instability region, namely for $\lambda_m < \lambda_i < 1$, due to the difference between the static and kinetic friction coefficient, i.e. μ_p, μ_s. In such a region, in fact, the system described by the skid-steering motion presents a positive feedback, because of $\alpha_s < 0$, which leads to an undesired increase of the robot velocity [4]. However, the system is not globally unstable thanks to the constraint $|lambda_i| < 1$. In [4], the authors also assume that the sliding friction coefficient is a fraction of the peak friction coefficient μ_p, i.e. $\mu_s = \alpha_s \mu_p = \alpha_s k_s \lambda_m$, where $0 \leq \alpha_s \leq 1$. With such a simplification, we can rewrite equation (2.29b) as:

$$\mu(\lambda_i) = k_s[\sigma_1(\lambda_m) + \sigma_2(\lambda_m)\lambda_i] \tag{2.30}$$

where

$$\begin{aligned}
\sigma_1(\lambda_m) &= \begin{cases} -\frac{1+\alpha_s\lambda_m}{1+\lambda_m}\lambda_m, & \text{for } \lambda_i \in [-1, -\lambda_m] \\ 0, & \text{for } \lambda_i \in (-\lambda_m, \lambda_m) \\ \frac{1-\alpha_s\lambda_m}{1-\lambda_m}\lambda_m, & \text{for } \lambda_i \in [\lambda_m, 1] \end{cases} \\
\sigma_2(\lambda_m) &= \begin{cases} \frac{1-\alpha_s}{1+\lambda_m}\lambda_m, & \text{for } \lambda_i \in [-1, -\lambda_m] \\ 1, & \text{for } \lambda_i \in (-\lambda_m, \lambda_m) \\ -\frac{1-\alpha_s}{1-\lambda_m}\lambda_m, & \text{for } \lambda_i \in [\lambda_m, 1]. \end{cases}
\end{aligned} \tag{2.31}$$

It must be noticed that, although the function in (2.30) is a continuous function, it is not a smooth function since it is not differentiable in $\lambda_i = \lambda_m$. Moreover, differently from (2.29a), it requires the identification of the three parameters $(\lambda_m, k_s, \alpha_s)$.

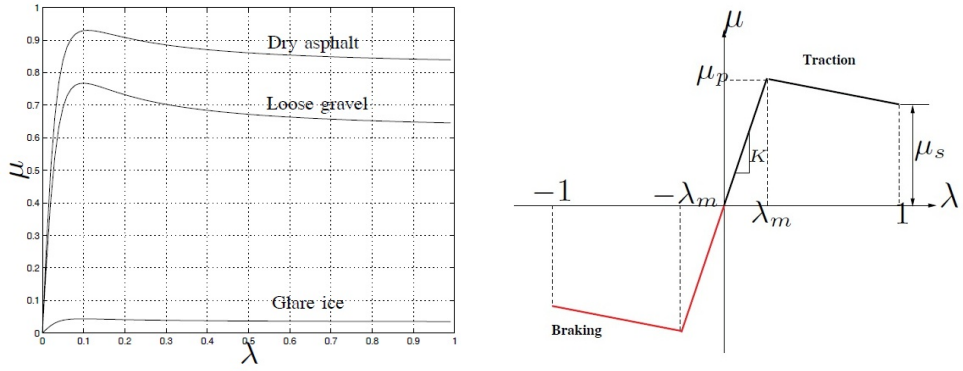
Under the aforementioned discussion, we can rewrite the longitudinal friction force F_i as:

$$F_i = -\mu(\lambda_i)N_i \qquad i = 1, \ldots, 4 \tag{2.32}$$

where $N_i \quad i = 1, \ldots, 4$ are defined in (2.24).

We can write the reaction forces and torques vector expressed in the inertial frame as:

$$\boldsymbol{R}(\boldsymbol{q}) = \begin{bmatrix} F_{rx} \\ F_{ry} \\ \tau_r \end{bmatrix} = {}^gR_l \begin{bmatrix} \sum_{i=1}^4 F_i \\ \sum_{i=1}^4 P_i \\ M_r \end{bmatrix} = \begin{bmatrix} \cos\theta \sum_{i=1}^4 F_i - \sin\theta \sum_{i=1}^4 P_i \\ \sin\theta \sum_{i=1}^4 F_i + \cos\theta \sum_{i=1}^4 P_i \\ M_r \end{bmatrix} \tag{2.33}$$



(a) (b)

Figure 2.4: (a) Relationship between the wheel/ground friction coefficient μ and the longitudinal slip λ under various road conditions; (b) A linear piece-wise approximation of the $\mu - \lambda$ relationship.

where $M_r = c(-F_1 - F_2 + F_3 + F_4) + b(P_2 + P_3) - a(P_1 + P_4)$ is the reaction torque and the reaction forces F_i, P_i are provided by (2.32) and (2.27).

Finally, we complete the dynamic model defined by (2.19) and (2.23), by adding the reaction forces and torques vector (2.33) and incorporating the non-holonomic constraint (2.13) by using the Lagrange multiplier ζ:

$$M\ddot{\boldsymbol{q}} + \boldsymbol{R}(\boldsymbol{q}) = B(\boldsymbol{q})\boldsymbol{\tau} + \boldsymbol{A}^T(\boldsymbol{q})\zeta. \tag{2.34}$$

To eliminate the unknown ζ, we multiply equation (2.34) by the matrix $S^T(\boldsymbol{q})$ defined in (2.14) and we use the constraint (2.16), obtaining the following dynamic model in the generalized coordinates $\boldsymbol{q}$:

$$\bar{M}\ddot{\boldsymbol{q}} + \bar{\boldsymbol{R}} = \bar{B}\boldsymbol{\tau} \tag{2.35}$$

where $\bar{M} = S^T M$, $\bar{\boldsymbol{R}} = S^T \boldsymbol{R}$ and $\bar{B} = S^T B$.

For control purpose, it is convenient to express the dynamic model of SSMR in terms of $\boldsymbol{\eta}, \dot{\boldsymbol{\eta}}$. Thus, by employing the kinematic equations (2.14) and (2.15) in (2.35), we obtained the dynamic model:

$$\bar{\bar{M}}\dot{\boldsymbol{\eta}} + \bar{\bar{C}}\boldsymbol{\eta} + \bar{\bar{\boldsymbol{R}}} = \bar{\bar{B}}\boldsymbol{\tau} \tag{2.36}$$

where

$$\bar{\bar{M}} = S^T M S = m \begin{bmatrix} 1 & 0 \\ 0 & x_{ICR}^2 + I \end{bmatrix}, \qquad \bar{\bar{C}} = S^T M \dot{S} = m x_{ICR} \begin{bmatrix} 0 & \dot{\theta} \\ -\dot{\theta} & \dot{x}_{ICR} \end{bmatrix},$$

$$\bar{\bar{R}} = \bar{R} = \begin{bmatrix} F_{rx}(\dot{q}) \\ x_{ICR} F_{ry}(\dot{q}) + M_r(\dot{q}) \end{bmatrix}, \qquad \bar{\bar{B}} = \bar{B} = \frac{1}{r} \begin{bmatrix} 1 & 1 \\ -c & c \end{bmatrix}.$$

2.2 Generalized Modeling

The simplified kinematic and dynamic models presented in the previous section hold if and only if the robot is strictly moving on a two dimensional plane, that is only two linear velocities parallel to the plane and the angular velocity around an axis perpendicular to the plane are considered. This assumption is generally a good approximation of the real robot motion when the roll and pitch motion, i.e. the rotation around the x and y axis, are negligible, which is generally true, as long as the robot stability is guaranteed, for trajectory tracking control and robot localization applications. The robot stability can be guaranteed by having, for instance, the height of CoM much smaller than the distance between the wheel/-ground contact points, or the moment of inertia along the x, y much lower than along the z axis, or a very stiff robot structure and wheel/ground interaction. However, the conditions on the robot motion stability do not depend only on the geometric and dynamic properties of the robot structure, but they also depend on the kinematic and dynamic constraints of robot motion, which might come from limits on robot velocities and interactions with the environment. Although such constraints play an important role in vehicle dynamics because of high velocities and slipping conditions [17], they are usually negligible for steered mobile robots (SMRs) [3], [18], [19]. In fact, for steered mobile robots non-slipping and pure rolling conditions are mostly satisfied, therefore, for relatively low velocities, reactive friction forces are usually much lower than forces resulting from inertia. Conversely, for SSMRs, when the robot is changing its orientation, reactive friction forces are usually much higher than forces resulting from inertia. As a consequence, even for relatively low velocities, dynamic properties of SSMRs can cause jerky motion and large amplitude vibrations, which influence robot motion and stability much more than for SMRs.

In [10], [4], [5], [11], [11], [15], the SSMR modeling is provided for trajectory tracking control and localization applications, therefore the dynamic properties due to the complex wheel/ground interaction are handled only by introducing the wheel slip, while no conse-

quence on robot vibrations is considered. Conversely, for remote telepresence applications, jerky motion and vibrations are usually not negligible as the robot is often required to perform sharp movements, like swiveling in place, and the remote user continuously needs full control of the robot and clear vision of the environment.

In order to provide a dynamic model reproducing the real robot jerky motion and large amplitude vibrations, we need to include the roll and pitch motion and a more accurate model of reaction forces. In particular, an accurate model of the reaction forces will be deeply discussed in Chapter 3 and 6.1, while a generalized 3D dynamic model of skid-steering motion is provided in this section, considering the reaction forces as unknown functions. As the robot is allowed to freely move in a 3D space, **Assumption** 1,5,6 considered in the previous section will not hold in the following, while **Assumption** 2,3,4 still hold.

We first consider a 4-wheels skid-steering vehicle moving freely in the 3D space, as depicted in Figure 2.5.

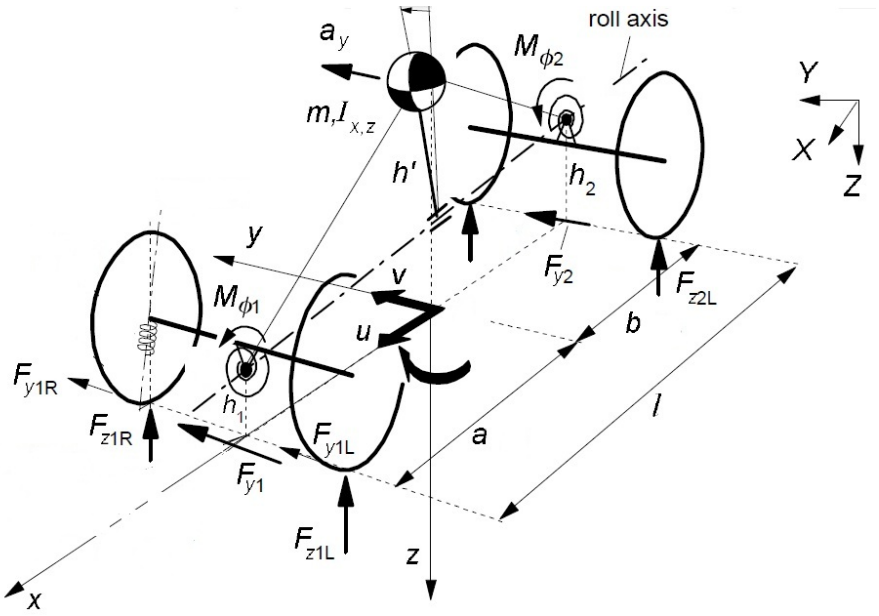


Figure 2.5: Three dimensional SSMR kinematics.

To describe motion of the robot it is convenient to define an local frame attached to it with origin in its CoM. We assume that $\boldsymbol{q} = [\boldsymbol{X}^T \quad \boldsymbol{\Theta}^T]^T = [X \quad Y \quad Z \quad \phi \quad \psi \quad \theta]^T$ denotes the generalized coordinate vector, where $\boldsymbol{X}, \boldsymbol{\Theta}$ determine respectively the CoM position and the orientation of the local frame with respect to the inertial frame, using the Roll (rotation around the z-axis about ϕ), Pitch (rotation around the y-axis about ψ) and Yaw (rotation around the x-axis about θ) angles as representation of the orientation. Let

$\boldsymbol{V} = [\boldsymbol{v}^T \quad \boldsymbol{\omega}^T]^T = [v_x \quad v_y \quad v_z \quad \omega_x \quad \omega_y \quad \omega_z]^T$ be a kinematic screw vector, namely the twist, where $\boldsymbol{v}, \boldsymbol{\omega}$ represent respectively the vector of the linear and angular velocity of the robot expressed in the local frame. The rotation matrix projecting the local frame onto the inertial frame is defined as:

$$
{}^gR_l(\phi,\psi,\theta) = \begin{bmatrix} \cos\phi\cos\psi & -\sin\phi\cos\theta + \cos\phi\sin\psi\sin\theta & \sin\phi\sin\theta + \cos\phi\sin\psi\cos\theta \\ \sin\phi\cos\psi & \cos\phi\cos\theta + \sin\phi\sin\psi\sin\theta & -\cos\phi\sin\theta + \sin\phi\sin\psi\cos\theta \\ -\sin\psi & \cos\psi\sin\theta & \cos\psi\cos\theta \end{bmatrix} \tag{2.37}
$$

With abuse of notation, we denote with $\dot{\boldsymbol{q}} = [{}^g\boldsymbol{v}^T \quad {}^g\boldsymbol{\omega}^T]^T = [{}^gv_x \quad {}^gv_y \quad {}^gv_z \quad {}^g\omega_x \quad {}^g\omega_y \quad {}^g\omega_z]^T$ the twist projected onto the inertial frame, i.e. the vector of the linear and angular velocity of the robot expressed in the inertial frame. It is well know from the kinematics of a rigid body that $\dot{\boldsymbol{q}} \neq \frac{d\boldsymbol{q}}{dt}$. In particular, the relationship between the vector $\boldsymbol{q}$ and the time derivative of $\boldsymbol{q}$ can be written as follows [18], [2], [19]:

$$
\frac{d\boldsymbol{q}}{dt} = \Omega\dot{\boldsymbol{q}} = \begin{bmatrix} \Omega_p & 0_3 \\ 0_3 & \Omega_r \end{bmatrix} \dot{\boldsymbol{q}} \tag{2.38}
$$

where Ω_p, Ω_r are defined as:

$$
\Omega_p = I_3, \qquad \Omega_r(\phi,\psi,\theta) = \begin{bmatrix} \cos\phi\tan\psi & \sin\phi\tan\psi & 1 \\ -\sin\phi & \cos\phi & 0 \\ \frac{\cos\phi}{\cos\psi} & \frac{\sin\phi}{\cos\psi} & 0 \end{bmatrix} \tag{2.39}
$$

By using the Plücker transform notation for spatial velocity vectors [18], [19], the kinematic equation in (2.1) can be replaced by the following equation, representing the kinematic model of a free rigid-body motion:

$$
\dot{\boldsymbol{q}} = J(\boldsymbol{q})\boldsymbol{V} = \begin{bmatrix} {}^gR_l & \hat{\boldsymbol{X}}{}^gR_l \\ 0_3 & {}^gR_l \end{bmatrix} \boldsymbol{V} \tag{2.40}
$$

where $J(\boldsymbol{q})$ represents the geometric Jacobian projecting the twist $\boldsymbol{V}$ to the inertial frame, and $\hat{\boldsymbol{X}}$ represents the skew-symmetric matrix of the position vector $\boldsymbol{X}$ and is defined as:

$$
\hat{\boldsymbol{X}} = \begin{bmatrix} 0 & -Z & Y \\ Z & 0 & -X \\ -Y & X & 0 \end{bmatrix} \tag{2.41}
$$

By combining equation (2.38) with (2.40), we can also wrote the following relation:

$$\frac{d\boldsymbol{q}}{dt} = \boldsymbol{\Omega} J(\boldsymbol{q})\boldsymbol{V} = J_a(\boldsymbol{q})\boldsymbol{V} \tag{2.42}$$

where $J_a(\boldsymbol{q})$ represents the analytical Jacobian of the system, considering $\boldsymbol{q}$ and $\boldsymbol{V}$ as state vector and control input respectively.

Furthermore, the kinematic relation given in (2.3) does not hold anymore in the general case where $v_z, \omega_x, \omega_y, d_{Cz} \neq 0$. In this case, in fact, the first equation in (2.2), combined with the constraint $\boldsymbol{d}_C \cdot \boldsymbol{v} = 0$, leads to the following relations:

$$\begin{aligned}
\omega_x |\boldsymbol{p}_{ICR}|^2 &= -y_{ICR} v_z + z_{ICR} v_y \\
\omega_x |\boldsymbol{p}_{ICR}|^2 &= x_{ICR} v_z - z_{ICR} v_x \\
\omega_x |\boldsymbol{p}_{ICR}|^2 &= -x_{ICR} v_y + y_{ICR} v_x \\
x_{ICR} v_x + y_{ICR} v_y + z_{ICR} v_z &= 0
\end{aligned} \tag{2.43}$$

where $|\boldsymbol{p}_{ICR}|^2 = (x_{ICR}^2 + y_{ICR}^2 + z_{ICR}^2)$.

We can rewrite (2.43) in matrix form, leading to the following non-holonomic constraint expressed in the local frame:

$$\begin{bmatrix}
0 & -z_{ICR} & y_{ICR} & |\boldsymbol{p}_{ICR}|^2 & 0 & 0 \\
z_{ICR} & 0 & -x_{ICR} & 0 & |\boldsymbol{p}_{ICR}|^2 & 0 \\
-y_{ICR} & x_{ICR} & 0 & 0 & 0 & |\boldsymbol{p}_{ICR}|^2 \\
x_{ICR} & y_{ICR} & z_{ICR} & 0 & 0 & 0
\end{bmatrix} \boldsymbol{V} = D(\boldsymbol{d}_C)\boldsymbol{V} = \boldsymbol{0} \tag{2.44}$$

By combining equation (2.40) and (2.44), we obtain the non-holonomic constraint expressed in the inertial frame:

$$D(\boldsymbol{d}_C)\boldsymbol{V} = D(\boldsymbol{d}_C) J^{-1}(\boldsymbol{q})\dot{\boldsymbol{q}} = A(\boldsymbol{q})\dot{\boldsymbol{q}} = \boldsymbol{0} \tag{2.45}$$

Similarly to the planar case, we notice from (2.45) that $\dot{\boldsymbol{q}}$ belongs to the null space of $A(\boldsymbol{q})$, therefore, by solving the equations in (2.44) with respect to $v_y, v_z, \omega_x, \omega_y$ and substituting them in $\boldsymbol{V}$, we can rewrite the kinematic model in (2.40) as follows:

$$\dot{\boldsymbol{q}} = J(\boldsymbol{q}) T(\boldsymbol{d}_C)\boldsymbol{\eta} = S(\boldsymbol{q})\boldsymbol{\eta} \tag{2.46}$$

where $T(\boldsymbol{d}_C)$ is the matrix derived from the substitution of $v_y, v_z, \omega_x, \omega_y$ and it is defined as follows:

$$\boldsymbol{V} = T(\boldsymbol{d}_C)\boldsymbol{\eta} = \begin{bmatrix} 1 & 0 \\ \frac{y_{ICR}}{x_{ICR}} & \frac{x_{ICR}^2+y_{ICR}^2+z_{ICR}^2}{x_{ICR}} \\ -\frac{x_{ICR}^2+y_{ICR}^2}{x_{ICR}z_{ICR}} & \frac{x_{ICR}^2y_{ICR}+y_{ICR}^3+y_{ICR}z_{ICR}^2}{x_{ICR}z_{ICR}} \\ \frac{y_{ICR}}{x_{ICR}z_{ICR}} & -\frac{y_{ICR}^2+z_{ICR}^2}{x_{ICR}z_{ICR}} \\ -\frac{x_{ICR}^2+y_{ICR}^2-z_{ICR}^2}{x_{ICR}^2z_{ICR}+y_{ICR}^2z_{ICR}+z_{ICR}^3} & \frac{x_{ICR}^2y_{ICR}+y_{ICR}^3+y_{ICR}z_{ICR}^2}{x^2*z+y^2*z+z^3} \\ 0 & 1 \end{bmatrix} \boldsymbol{\eta} \tag{2.47}$$

The vector $\boldsymbol{\eta} = [v_x \quad \omega_z]^T$, using the same notation for the planar case, represent the control input at kinematic level, therefore equation (2.46) represents the generalized kinematic model for SSMRs.

Finally, by combining equation (2.45) with (2.46), it is straightforward to obtain the same relation as (2.16), but with the new connotation of the matrices $A(\boldsymbol{q}), S(\boldsymbol{q})$. Similarly as it was done for the dynamic model in the planar case, such relation will be useful in the treatment of the generalized dynamic model of a SSMR expressed in the inertial and local frame.

It can be also proved that, although the relations in (2.3) does not hold for the 3D case and the non-holonomic constraint changes, the relations on wheel velocities defined in (2.5) still hold and a relation similar to (2.11) can be found. For this reason, we can consider the same control input $\boldsymbol{\eta}$ at kinematic level.

The generalized dynamic equation of SSMRs can still be obtained using Euler-Lagrange principle with Lagrange multipliers to include non-holonomic constraint. As **Assumption** 6 does not hold in the general case and all linear and angular velocities are allowed, the Lagrangian of the system is defined as:

$$L = E - U = \frac{1}{2}(m\dot{X}^T\dot{X} + \boldsymbol{\Omega}^T\boldsymbol{I}_g\boldsymbol{\Omega}) - m\boldsymbol{X}^T\boldsymbol{g} = \frac{1}{2}\dot{\boldsymbol{q}}^T\boldsymbol{M}_g\dot{\boldsymbol{q}} + mgZ \tag{2.48}$$

where $\boldsymbol{g} = [0 \quad 0 \quad -g]^T$ is the gravitational acceleration vector, with $g = 9.8\ \frac{m}{s^2}$, and $\boldsymbol{I}_g \in \mathbb{R}^3$, $\boldsymbol{M}_g \in \mathbb{R}^6$ are respectively the inertia matrix and the generalized inertia tensor calculated with respect to the inertial frame.

By applying the Euler-Lagrange equation in (2.17) to the Lagrangian in (2.48), we obtain:

$$\Gamma = \begin{bmatrix} mI_3 & 0_3 \\ 0_3 & \boldsymbol{I}_g \end{bmatrix} \ddot{\boldsymbol{q}} + mg \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \boldsymbol{M}_g \ddot{\boldsymbol{q}} + \boldsymbol{G} \tag{2.49}$$

where I_3 indicates a three dimensional identity matrix and $\boldsymbol{G} = -m[\boldsymbol{g}^T \quad \boldsymbol{0}_3^T]^T$ is a six-dimension vector representing the contribution of gravity.

As **Assumption** 4 is still considered valid, also because our robot has only one motor for each side's two wheels, we will be considering the vector $\boldsymbol{\tau}$ defined in (2.22) as control input at dynamic level. Thereby, similarly to the planar case, we can calculate the vector Γ by considering the vector of the active forces and torques exerted on the i^{th} wheel/-ground contact point expressed on the local frame as $\boldsymbol{F}_{ia} = [\frac{\tau_i}{r} \quad 0 \quad 0 \quad 0 \quad 0 \quad 0]^T$. To obtain the total active force and torque applied to the CoM and expressed in the inertial frame, first we compute the resultant active force and torque vector on the local frame, $\boldsymbol{F}_a = [F_{ax} \quad F_{ay} \quad F_{az} \quad M_{ax} \quad M_{ay} \quad M_{az}]^T$, and then we multiply it by the Plücker transform for spatial force vectors [18], [19]. In particular, as we are considering the frame centered at the i^{th} contact point parallel to the local frame, we can calculate the active force and torque vector in the local frame as:

$$\boldsymbol{F}_a = \sum_{i=1}^{4} \begin{bmatrix} I_3 & 0_3 \\ \hat{p}_i & I_3 \end{bmatrix} \boldsymbol{F}_{ia} = \begin{bmatrix} \frac{\tau_L+\tau_R}{r} \\ 0 \\ 0 \\ 0 \\ \frac{-h(\tau_L+\tau_R)}{r} \\ \frac{-c\tau_L+d\tau_R}{r} \end{bmatrix} \tag{2.50}$$

Thereby, we can express Γ also as follows:

$$\Gamma = (J^{-1})^T(\boldsymbol{q})\boldsymbol{F}_a = (J^{-1})^T(\boldsymbol{q})T_f\boldsymbol{\tau} = B(\boldsymbol{q})\boldsymbol{\tau} \tag{2.51}$$

where

$$(J^{-1})^T(\boldsymbol{q}) = \begin{bmatrix} {}^gR_l & 0_3 \\ \hat{\boldsymbol{X}}\,{}^gR_l & {}^gR_l \end{bmatrix}, \qquad T_f = \begin{bmatrix} 1 & 1 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ -h & -h \\ -c & d \end{bmatrix}. \tag{2.52}$$

In order to complete the generalized dynamic model of SSMRs, we must add the reaction forces and torques in equation (2.49) and include the non-holonomic constraint (2.45) by using the Lagrange multipliers. As previously said, in this section we are only considering the reaction forces at the i^{th} contact point as unknown functions f_{ix}, f_{iy}, f_{iz}. Thereby, similarly to case of the active forces and torques, we can write the reaction force and torque vector expressed in the inertial frame, by first calculating it in the local frame, $\boldsymbol{F}_r = [f_x \quad f_y \quad f_z \quad \tau_x \quad \tau_y \quad \tau_z]^T$, and then multiplying by the Plücker transform. By considering the vector of reaction forces and torques at each contact point defined as $\boldsymbol{F}_i = [f_{ix} \quad f_{iy} \quad f_{iz} \quad 0 \quad 0 \quad 0]^T$, we can calculate the reaction force and torque vector expressed in the local frame as:

$$\boldsymbol{F}_r = \sum_{i=1}^{4} \begin{bmatrix} I_3 & 0_3 \\ \hat{p}_i & I_3 \end{bmatrix} \boldsymbol{F}_i = \begin{bmatrix} \sum_{i=1}^4 f_{ix} \\ \sum_{i=1}^4 f_{iy} \\ \sum_{i=1}^4 f_{iz} \\ \sum_{i=1}^4 (y_i f_{iz} - z_i f_{iy}) \\ \sum_{i=1}^4 (-x_i f_{iz} + z_i f_{ix}) \\ \sum_{i=1}^4 (x_i f_{iy} - y_i f_{ix}) \end{bmatrix} \tag{2.53}$$

Thereby, the reaction force and torque vector expressed in the inertial frame can be calculated as follows:

$$\boldsymbol{R}(\boldsymbol{q}) = (J^{-1})^T(\boldsymbol{q})\boldsymbol{F}_r \tag{2.54}$$

As it will be described in Chapter 4, the reaction forces f_{ix}, f_{iy}, f_{iz} exerted on the i^{th} contact point depend respectively by the wheel center velocities v_{ix}, v_{iy}, v_{iz}. The relation between the wheel center velocity $\boldsymbol{v}_i$ and the robot linear and angular velocity can be written as follows:

$$\boldsymbol{v}_i = \begin{bmatrix} v_{ix} \\ v_{iy} \\ v_{iz} \end{bmatrix} = \boldsymbol{v} - \hat{p}_i \boldsymbol{\omega} = \begin{bmatrix} v_x - y_i\omega_z + z_i\omega_y \\ v_y + x_i\omega_z - z_i\omega_x \\ v_z - x_i\omega_y + y_i\omega_x \end{bmatrix} \tag{2.55}$$

where $\hat{p}_i$ is the skew-symmetric matrix of the vector p_i.

Finally, we can write the generalized dynamic model expressed in the inertial frame by combining (2.49) and (2.51), adding the reaction forces and torques vector (2.54) and incorporating the non-holonomic constraint (2.45) by using the vector of Lagrange multipliers $\boldsymbol{\zeta}$:

$$\boldsymbol{M}_g \ddot{\boldsymbol{q}} + \boldsymbol{R}(\boldsymbol{q}) + \boldsymbol{G} = B(\boldsymbol{q})\boldsymbol{\tau} + A^T(\boldsymbol{q})\boldsymbol{\zeta}. \tag{2.56}$$

To eliminate the unknown $\boldsymbol{\zeta}$, we multiply equation (2.56) by the matrix $S^T(\boldsymbol{q})$ defined in (2.40) and then we use the constraint (2.16), obtaining the following dynamic model in the generalized coordinates $\boldsymbol{q}$:

$$\bar{M}\ddot{\boldsymbol{q}} + \bar{\boldsymbol{R}} + \bar{\boldsymbol{G}} = \bar{B}\boldsymbol{\tau} \tag{2.57}$$

where $\bar{M} = S^T \boldsymbol{M}_g$, $\bar{\boldsymbol{R}} = S^T \boldsymbol{R}$, $\bar{\boldsymbol{G}} = S^T \boldsymbol{G}$ and $\bar{B} = S^T B$.

For control purpose, it is convenient to express the dynamic model of SSMR in terms of $\boldsymbol{\eta}, \dot{\boldsymbol{\eta}}$. Thus, by including the kinematic equation (2.40) and its time derivative in (2.57), we obtained the dynamic model:

$$\bar{\bar{M}}\dot{\boldsymbol{\eta}} + \bar{\bar{C}}\boldsymbol{\eta} + \bar{\boldsymbol{R}} + \bar{\boldsymbol{G}} = \bar{B}\boldsymbol{\tau} \tag{2.58}$$

where $\bar{\bar{M}} = S^T \boldsymbol{M}_g S$ and $\bar{\bar{C}} = S^T M \dot{S}$.

It is worth to notice that the dynamic model in (2.58) depends on $\boldsymbol{q}$, and more precisely on the value of $\boldsymbol{\Theta}$, only for the matrix $\bar{\boldsymbol{G}}$. In fact, we can rewrite the matrices in (2.58) by employing relations (2.46),(2.51),(2.54) as follows:

$$\begin{aligned}
\bar{\bar{M}} &= S^T \boldsymbol{M}_g S = T^T J^T (J^{-1})^T \boldsymbol{M} J^{-1} J T = T^T (J^{-1} J)^T \boldsymbol{M} J^{-1} J T = T^T \boldsymbol{M} T \\
\bar{\bar{C}} &= S^T \boldsymbol{M}_g \dot{S} = T^T J^T (J^{-1})^T \boldsymbol{M} J^{-1} (J\dot{T} + \dot{J}T) = T^T \boldsymbol{M}\dot{T} + T^T \boldsymbol{M} J^{-1} \dot{J} T \\
\bar{\boldsymbol{R}} &= S^T \boldsymbol{R} = T^T J^T (J^{-1})^T \boldsymbol{F}_r = T^T \boldsymbol{F}_r \\
\bar{\boldsymbol{G}} &= S^T \boldsymbol{G} = T^T J^T \boldsymbol{G} = T^T [-m({}^l R_g \boldsymbol{g})^T \quad \boldsymbol{0}_3^T]^T \\
\bar{B} &= S^T B = T^T J^T (J^{-1})^T T_f \boldsymbol{\tau} = T^T T_f \boldsymbol{\tau}.
\end{aligned} \tag{2.59}$$

where we used the transformation law of the inertia tensor $\boldsymbol{M}_g = (J^{-1})^T \boldsymbol{M} J^{-1}$ [19], with $\boldsymbol{M}$ representing the inertia tensor expressed in the local frame.
Moreover, it can be proved that the term $J^{-1}\dot{J}$, that appears in the expression of $\bar{\bar{C}}$, does not depend on $\boldsymbol{q}, \dot{\boldsymbol{q}}$ but on $\dot{\boldsymbol{V}}$, therefore on $\dot{\boldsymbol{\eta}}$ and $\dot{\boldsymbol{p}}_{ICR}$, as it can be rewritten as:

$$J^{-1}\dot{J} = \hat{\boldsymbol{V}} = \begin{bmatrix} \boldsymbol{0} & \hat{\boldsymbol{v}} \\ \hat{\boldsymbol{v}} & \hat{\boldsymbol{\omega}} \end{bmatrix} \tag{2.60}$$

A simple proof can be obtained by considering $\boldsymbol{\eta} = \boldsymbol{V}$, therefore $T = I$, and rewriting equation (2.58) with the resultant of all forces and torques on the right side:

$$\boldsymbol{M}\dot{\boldsymbol{V}} + \boldsymbol{M}J^{-1}\dot{J}\boldsymbol{V} = \boldsymbol{F}_{tot}$$

By using Newton-Euler equation, we can also write the $\boldsymbol{F}_{tot}$ as:

$$\boldsymbol{F}_{tot} = \begin{bmatrix} m\dot{\boldsymbol{v}} + \boldsymbol{\omega} \times m\boldsymbol{v} \\ \boldsymbol{I}\dot{\omega} + \boldsymbol{\omega} \times \boldsymbol{I}\boldsymbol{\omega} \end{bmatrix} = \boldsymbol{M}\dot{\boldsymbol{V}} + \boldsymbol{M}\begin{bmatrix} \boldsymbol{0} & \hat{\boldsymbol{v}} \\ \hat{\boldsymbol{v}} & \hat{\boldsymbol{\omega}} \end{bmatrix}\boldsymbol{V}$$

Finally, by equating the two equations, we obtain the relation in (2.60) so that we can rewrite equation (2.58) as:

$$T^T \boldsymbol{M} T\dot{\boldsymbol{\eta}} + (T^T \boldsymbol{M}\dot{T} + T^T \boldsymbol{M}\hat{\boldsymbol{V}}T)\boldsymbol{\eta} + T^T \boldsymbol{F}_r + T^T J^{-1}\boldsymbol{G} = T^T T_f \boldsymbol{\tau}. \tag{2.61}$$

where the matrix $\hat{\boldsymbol{V}}$ is calculated by substituting the coordinates of $\boldsymbol{V}$ with the transformed coordinates $T\boldsymbol{\eta}$.
It is worth noticing that the dynamic model defined in (2.61) requires the knowledge of the coordinates of the ICR. Let us imagine that the robot linear and angular velocities can be measured, for instance by employing an Inertial Measurement Unit (IMU). Then, from the knowledge of $\boldsymbol{V}$, the coordinates of $\boldsymbol{p}_{ICR}$ can be calculated by solving the equations in (2.44). Although the dynamic model in (2.61) can be useful for control issues, it can not be directly employed for simulation purposes, because it is not possible to determine the coordinates of $\boldsymbol{p}_{ICR}$ without knowing all the values of $\boldsymbol{V}$, which can not be computed as the output of the model. For this reason, we rewrite the dynamic model by substituting in (2.57) the kinematic model in (2.40) and its time derivative:

$$T^T \boldsymbol{M} T\dot{\boldsymbol{V}} + T^T \boldsymbol{M}\hat{\boldsymbol{V}}\boldsymbol{V} + T^T \boldsymbol{F}_r + T^T J^{-1}\boldsymbol{G} = T^T T_f \boldsymbol{\tau}. \tag{2.62}$$

where we used the matrices transformation described in (2.59).

Furthermore, we notice that the transformation matrix T defined in (2.47) is always full column-rank as the first and the last rows are linear independent $\forall x_{ICR}, y_{ICR}, z_{ICR}$. Thereby, an optimal inverse solution of equation (2.47) can be always obtained by pre-multiplying it by the pseudo-inverse of T as follows:

$$\boldsymbol{\eta} = T^{\#}\boldsymbol{V} = (T^T T)^{-1} T^T \boldsymbol{V} \tag{2.63}$$

Similarly, we notice that the matrix T^T is full row-rank, therefore we can defined its pseudo-inverse as:

$$(T^T)^{\#} = (T T^T)^{-1} T \tag{2.64}$$

Thereby, by pre-multiply equation (2.62) by $(T^T)^{\#}$, the dynamic model of SSMRs can be rewritten considering $\boldsymbol{V}$ as control input:

$$\boldsymbol{M}\dot{\boldsymbol{V}} + \boldsymbol{M}\hat{\boldsymbol{V}}\boldsymbol{V} + J^{-1}\boldsymbol{G} = T_f\boldsymbol{\tau} + \boldsymbol{F}_r = \boldsymbol{F}. \tag{2.65}$$

which represents, as expected, the generalized dynamic model of a rigid-body freely moving in a three-dimensional space, with the term $\boldsymbol{M}\hat{\boldsymbol{V}}\boldsymbol{V}$ representing the contribution of the Coriolis forces.

It is worth noticing that the dynamic model in (2.61),(2.65) holds if and only if **Assumption 6** is satisfied. Conversely, if the wheels can lift from the ground surface, the impact model, relating the contact point velocity before and after the impact with the ground, must be added to the model.

Chapter 3

Experimental Data

In this chapter, a deep analysis regarding the robot vibrations and the identification of the tire/ground reaction force is provided to fully understanding the dynamic causes of the robot jerky motion. The measurements are provided by using the accelerometers and the testing system machine presented in Section 1.2. The data are acquired respectively by using `LabVIEW` and `MTESTQuattro` software, as already mentioned, and then loaded and analyzed in `Matlab`.
The data presented here will be useful in the next chapters for an accurate modeling of the reaction forces and the skid-steering motion. The data will be utilize also as benchmark for validating the results obtained from the simulation.

3.1 The accelerometer

An accelerometer is a device for detecting and measuring linear acceleration. It produces an output, usually electrical, which is proportional to the rate of acceleration. The basic principle is outlined in Figure 3.1(a), where the spring-mass-damper system is the heart of the accelerometer. Under the assumption that we are in free space and that the body is subject to an acceleration a oriented along the y-axis, because of the relativism principle we can consider the proof mass feeling the same acceleration a oriented along the x-axis, i.e. the opposite direction of the body. Thereby, the system can be described by the following equation:

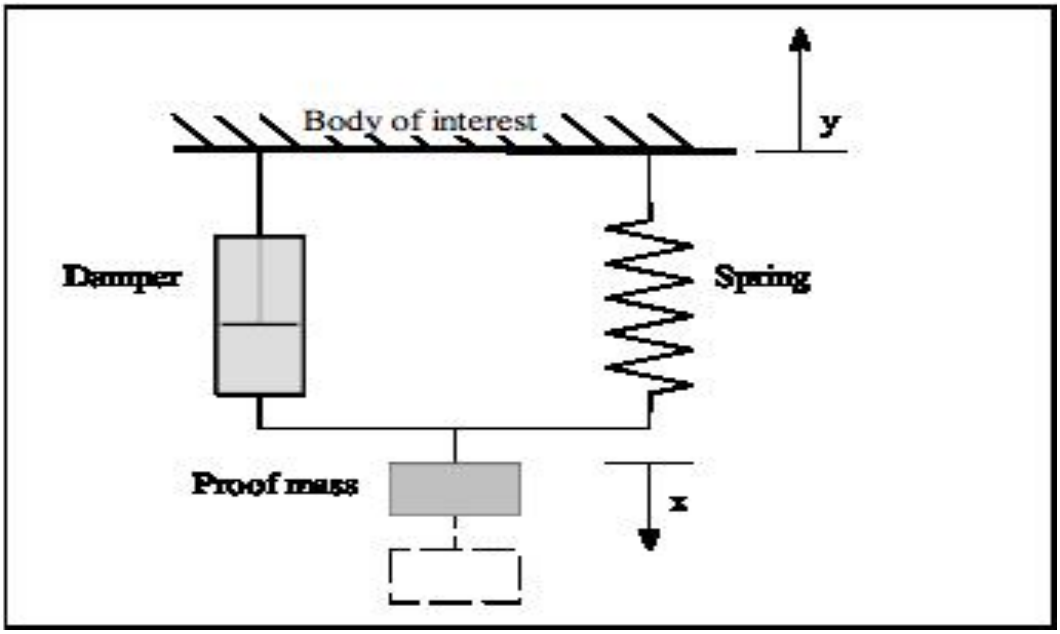
$$m_a \ddot{x}_a + D_a \dot{x}_a + K_a x_a = m_a a \tag{3.1}$$

where m_a, D_a, K_a are respectively the mass, the damping and elasticity coefficient of the spring.

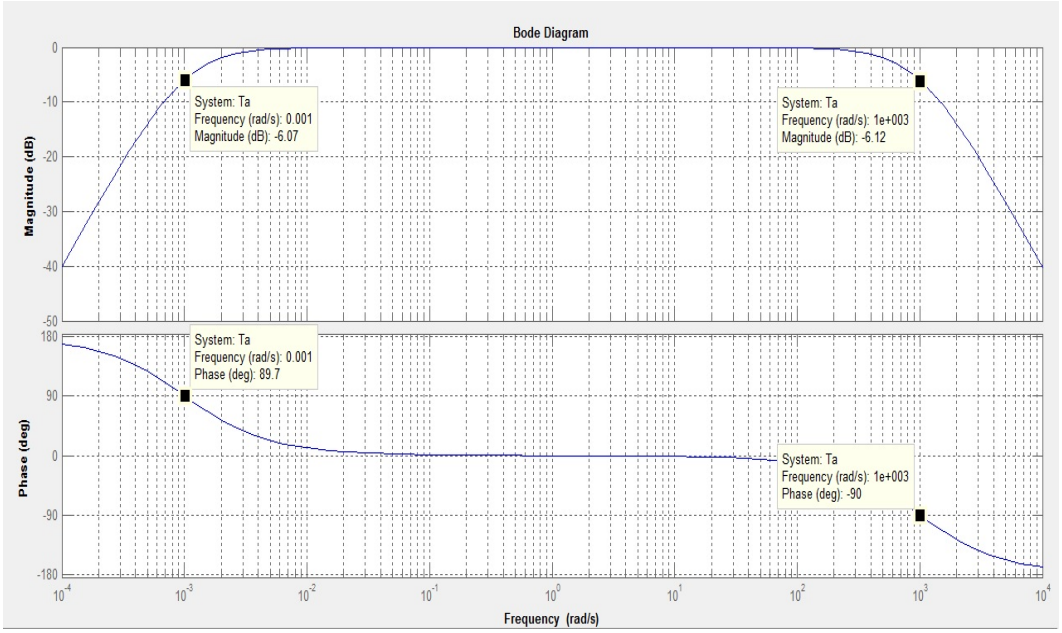
The device usually measures the displacement of the proof mass (x_a), for instance by measuring the voltage between the two faces of a capacitor, and then it computes the second derivative of the measured signal, producing the signal $\ddot{x}_a$ as output. To better understand the relation between the measured acceleration and the real body acceleration, we can compute the Laplace transform of the spring position $X_a(s)$ by solving equation (3.1) in the frequency domain, and its second time derivative by multiplying it for s^2:

$$s^2 X_a(s) = s^2 \frac{1}{s^2 + \frac{D_a}{m_a} s + \frac{K_a}{m_a}} A(s) = T_a(s) A(s) \tag{3.2}$$

where $X_a(s)$ and $A(s)$ are the Laplace transforms of x_a and a respectively, and $T_a(s)$ represents the transfer function of the system. We can also define the natural angular frequency of the system in (3.1) as $\omega_{n_a} = \sqrt{\frac{K_a}{m_a}}$, and the damping ratio as $\xi_a = \frac{D_a}{2\sqrt{K_a m_a}}$.
In order to avoid undesired oscillations in the measured signal x_a, the parameters m_a, D_a, K_a are always set such that the spring-mass-damper system defined by (3.1) is never underdamped ($\xi_a \geq 1$). Moreover, as the measured acceleration must be as close and as fast as possible to the body acceleration, such a system is usually critically damped ($\xi_a = 1$) and the natural angular frequency is relatively small, depending on the physical structure of the device. As outlined by the Bode diagram depicted in Figure 3.1(b), the transfer function $T_a(s)$ behaves like a second order derivative for $\omega << \omega_{n_a}$, cutting off the low frequency components of the real acceleration a, while it has nearly a unitary gain for $\omega >> \omega_{n_a}$, leaving unaltered the medium and high frequency components of a, except for a delay due to its phase. Moreover, as the exact differentiation is not physically possible, the acceleration $\ddot{x}_a$ is provided by employing electrical devices, like operational amplifiers, which introduce in $T_a(s)$ a double pole at ω_c, cutting off also the high frequency components of a. For this reason, we can consider that, for a very large range of frequencies (usually $\omega_{n_a} < 10^{-3}$ and $\omega_c > 10^3$) and with relatively small delay, the measured acceleration is equal to the real body acceleration, i.e. $\ddot{x}_a = a$. As a matter of notation, it is usually considered as body of interest the body containing the spring-mass-damper, i.e. the physical accelerometer, and as its positive direction the direction of the y-axis as depicted in Figure 3.1(a), i.e. the opposite direction of the spring axis which the position measurement is provided along. Thereby, a 1-axis accelerometer can be considered a black-box behaving as a band-pass filter measuring



(a)



(b)

Figure 3.1: (a) A spring-mass-damper 1-axis accelerometer scheme; (b) Bode diagram of the transfer function a spring-mass-damper 1-axis accelerometer system.

its acceleration along its axis.

3.2 Characterization of the Vibrations

The aim of this section is to provide a qualitative and quantitative idea about the vibrations which occur while the robot is turning ($\omega_L \neq \omega_R$). First of all, different qualitative tests has been done by running the robot at different angular velocities on different types of ground. The tests has been also done by modifying the robot's structure, therefore changing its dynamics configuration. In particular, the robot has been run with and without the aluminum frame and the laptop, with different types of tire and even with two cubes of wood replacing two wheels on the same side. Moreover, to check the importance of the robot inertia, a long plastic pale with two weights attached to its extremes was fixed to the robot along its principal axes. After several tests, we could clearly claim that the main cause of the robot vibrations is the complex behavior of the lateral reaction forces, due to the tire dynamics and the complex tire/ground interaction, and their interaction with each other. In fact, when we replace two wheels with cubes of wood, large vibrations never appear independently from the robot's structure. Conversely, when the four wheels are mounted on the robot, the vibrations do not appear while running the robot's platform, but they become relevant when the aluminum frame is fastened on it and they increase when increasing the robot weight and inertia along the x and y axis.

In order to have also a quantitative idea, the vibrations are measured by acquiring the data from the accelerometers fixing the wood-cube to the robot in different locations, while swiveling the robot in place with different angular velocities ($v_x^* = 0, \omega_z^* \neq 0$). The data are also acquired by running the robot on all the types of floor in the School of Engineering at RWU, i.e. carpet for the rooms, tile for the corridors and concrete for the laboratories.

The relation between the acceleration $\boldsymbol{a} = [a_x \quad a_y \quad a_z]^T$ measured from the three accelerometers and the robot acceleration $\dot{\boldsymbol{V}}$, when the wood-cube is fixed to the robot at the position $\boldsymbol{p}_a$ with respect to the robot CoM, can be obtained by computing the linear velocity of the wood-cube from the robot velocity, differentiating it with respect to time and adding the gravity contribution. The linear velocity $\boldsymbol{v}_a$ can be computed by applying the Plücker transform for spatial velocity vectors [18], [19] to the robot twist $\boldsymbol{V}$:

$$\boldsymbol{v}_a = {}^aR_l\boldsymbol{v} - \hat{p}_a{}^aR_l\boldsymbol{\omega} \tag{3.3}$$

where ${}^aR_l$ is the rotation matrix projecting the robot frame onto the wood-cube frame and $\hat{p}_a$ is the skew-symmetric matrix of $\boldsymbol{p}_a$.
As $\boldsymbol{p}_a$ and ${}^aR_l$ do not change during time, by differentiating equation (3.3) and taking into account the considerations made in Section 3.1, the acceleration $\boldsymbol{a}$ measured from the three accelerometers can be written as follows:

$$\boldsymbol{a} = {}^aR_l\dot{\boldsymbol{v}} - \hat{p}_a{}^aR_l\dot{\boldsymbol{\omega}} - {}^aR_g\boldsymbol{g} \tag{3.4}$$

where $\boldsymbol{g} = [0 \quad 0 \quad -g]^T$ is the gravity vector and ${}^aR_g = {}^aR_l{}^lR_g$ represents the rotation matrix projecting the inertial frame onto the wood-cube frame.
To simplify the relation (3.4), the wood-cube is always fixed to the robot such that its axes are parallel to the robot principal axes. In this way, we have ${}^aR_l = I_3$, therefore relation (3.4) can be rewritten:

$$\boldsymbol{a} = \dot{\boldsymbol{v}} - \hat{p}_a\dot{\boldsymbol{\omega}} - {}^lR_g\boldsymbol{g} \tag{3.5}$$

Although the accelerometers can have a high sensitivity and a large band-width, providing a very accurate and reliable measurement of their acceleration along their axis, it must be noticed that such acceleration might not coincide to (3.4). The measured accelerations mainly depend on the robot acceleration calculated on its CoM, but they also depend on the acceleration of the robot's part which they are fixed to, if such a part can move, and on how they are fixed to it. In such a case, in fact, when we differentiate equation (3.3) the matrices $\hat{p}_a, {}^aR_l$ are not constant, therefore equation (3.4) does not hold anymore. For this reason, the accelerometers should be always rigidly fixed to the robot so that the matrices $\hat{p}_a, {}^aR_l$ remain constant.
Taking into account the above considerations, the wood-cube is firmly fixed to robot's platform by using a strong duct scotch tape, as we could not fasten it by screws or glue to not damage the robot, approximatively at the position $\boldsymbol{p}_a \approx [-0.2 \quad 0 \quad 0]^T$ (Figure 3.2). This choice comes from the fact that the aluminum frame and the laptop are fastened to the robot's platform by screws and Velcro, therefore they can slightly move with respect to the robot frame. Furthermore, in order to measure the oscillations present in both the linear and angular accelerations, we want the accelerometers to be as far as possible from the robot

CoM, as the contribution of $\boldsymbol{\omega}$ in (3.5) is mainly given by the term $\hat{p}_a\dot{\boldsymbol{\omega}}$. By substituting the values of $\boldsymbol{p}_a$ in (3.5), we obtain the following relations:

$$\begin{aligned} a_x &= \dot{v}_x + g\sin\psi \\ a_y &= \dot{v}_y - p_{a_x}\dot{\omega}_z - g\cos\psi\sin\theta \\ a_z &= \dot{v}_z + p_{a_x}\dot{\omega}_y - g\cos\psi\cos\theta \end{aligned} \tag{3.6}$$

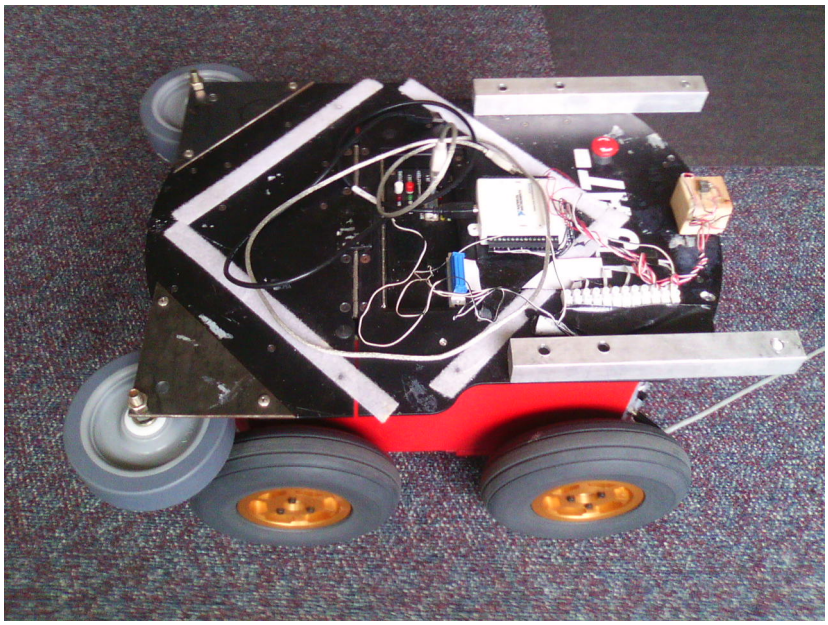


Figure 3.2: The wood-cube fixed to the robot's platform at the position $\boldsymbol{p}_a \approx [-0.2 \quad 0 \quad 0]^T$.

We notice that in (3.6) there are all the robot linear and angular accelerations along and around the three principal axes except for $\dot{\omega}_x$. However, as the robot oscillations around the x and y-axis are constraint by the tire vertical reaction forces, as result of the Roll, Pitch and Yaw motion the robot assumes significant values for the linear accelerations along the x and y-axis. In fact, as we will see in the next chapter, the tire reaction forces f_{ix}, f_{iy}, f_{iz} depend on the wheel center velocities, which depend on the robot angular velocity by relation (2.55). Thereby, the six differential equations defined in (2.65) are strongly coupled even when the Coriolis term is negligible, i.e. for relatively low velocities as in the case of swiveling in place. Moreover, we notice that, although the real robot angular acceleration can assume relatively high values because of the vibrations, the angles θ, ψ always assume relatively small values if the wheels are not lifting from the floor, which is in general true for $\omega_z^* \leq 64\,deg/s$. In such a condition, we can approximate the trigonometric functions by their first

order Taylor expansion, i.e. $\cos x \approx 1, \sin x \approx x$, so that we can rewrite (3.6) as follows:

$$\begin{aligned} a_x &= \dot{v}_x + g\psi \\ a_y &= \dot{v}_y - p_{a_x}\dot{\omega}_z - g\theta \\ a_z &= \dot{v}_z + p_{a_x}\dot{\omega}_y - g \end{aligned} \tag{3.7}$$

The data are acquired from the accelerometers at a sample rate of 1 kHz by using `LabVIEW` and then exported and saved in an Excel format, ready to be loaded and analyzed in `Matlab`. The Matlab functions that are used for the frequency analysis of the signals are defined in Appendix A. For every value of the control input ω_z^* contained in the variable `rangeW`, the main program first parses the data by calling the function `parseData()`, which loads the data from the Excel file, computes their Fast Fourier Transform (FFT), converts them from Voltage values to acceleration values and eventually plots them and their FFT. To improve the accuracy and reduce the computation time of the FFT function implemented in Matlab, the FFT should be computed on the closest lower power of 2 of samples [20]. In our case, as the data are acquired for 20 seconds, only the central 2^{14} over 20000 samples are considered for the FFT. Then, by calling the function `filterData()`, the program low-pass filters the data at 50 Hz by using a 10^{th} order Butterworth filter in order to cut off the noise due to alternate current which works at 60 Hz. Such a choice of the cut-off frequency is also motivated by the fact that, by taking a first look to the FFT of all the data, there never appear significant frequency components higher than 50 Hz, except for the alternate current frequency around 60 Hz. After low-pass filtering, the amplitude of the signal is calculated as difference between their maximum and minimum value, the FFT of the low-pass filtered data is computed and the three highest peaks in the FFT are found by calling the function `findPeakFreq()`. The `findPeakFreq()` function also band-pass filter the data by a narrow band, generally 2 Hz, centered on the maximum peak frequency. Finally, the original data are plotted together with the low-pass and band-pass filtered data, in order to check the reliability of the maximum peak frequency obtained, and all the amplitudes and peak frequencies are plotted with respect to the robot angular velocity.

Let us consider the data acquired while swiveling the robot in place at different desired angular velocity ω_z^*, separately on concrete, tile and carpet. In order to check the repeatability of the data, the accelerations are also acquired three times for each angular velocity ω_z^*. Because the wheels start lifting from the floor generally when $\omega_z^* > 64\,\frac{deg}{s}$, making the robot

continuously hopping on its wheels and eventually making the motors stalling, we will be considering only the data acquired for $\omega_z^* \leq 64 \frac{deg}{s}$.

Some example graphs of the low-pass filtered data with their FFT are provided in Appendix B. The graphs representing the average of the data amplitudes and the average of the three highest peak frequencies with respect to the angular velocity ω_z^* and the type of ground are also provided in Appendix B.

Although the data acquired during different trials present a sort of irregularity in their peak frequencies, due to the irregularity of the ground, there are some features which characterize them. In particular, by looking at the graphs depicted in Appendix B, the main features can be listed as follows:

- There are, usually, two main frequency components (generally the two highest peak in the FFT), which can be clearly identified for every angular velocity ω_z^* and for all the three types of ground.
- The two main components are, usually, one at ¨low¨ ($< 10\,Hz$) and the other at ¨high¨ frequency ($> 20\,Hz$), when swiveling the robot with relatively low angular velocities ($\omega_z^* < 30 \frac{deg}{s}$).
- The two main components are both at ¨low¨ frequency ($< 10\,Hz$), when swiveling the robot with relatively high angular velocities ($\omega_z^* > 30 \frac{deg}{s}$).
- When running the robot on carpet, the ¨low¨ frequency component is always the dominant component.
- When running the robot on concrete and tile, the ¨high¨ frequency component is, usually, the dominant component for relatively low angular velocities ($\omega_z^* < 30 \frac{deg}{s}$), while the ¨low¨ frequency component is, usually, the dominant component for relatively high angular velocities ($\omega_z^* > 30 \frac{deg}{s}$).
- The ¨high¨ frequency components seem to increase when increasing the robot angular velocity until $30 \frac{deg}{s}$.
- When running the robot on carpet, the amplitude of the oscillations increases almost linearly with respect to the angular velocity.
- When running the robot on concrete and tile, the amplitude of the oscillations is relatively small and constant for low angular velocities ($\omega_z^* < 30 \frac{deg}{s}$), while it becomes

suddenly larger and irregular for high angular velocities ($\omega_z^* > 30\ \frac{deg}{s}$).

First of all, it must be specified that the accelerometers are intentionally not mechanically isolated from the robot structural vibrations, therefore the frequency components found in the recorded signals may derive, especially in the high frequency spectrum, from structural vibrations which do not depend on the tire reaction forces and do not contribute to the large amplitude vibrations which destabilize the robot. However, as the accelerations in (3.6) depend on both the robot acceleration and on the Roll and Pitch angles, we might think that the high frequency component derives from the robot platform accelerations, produced by the tire reaction forces, while the low frequency component derives from the Roll and Pitch oscillations. The Roll and Pitch oscillations increase when, due to the type of ground and the robot inertial properties, the frequencies deriving from the tire reaction forces get closer to the robot structural resonance frequency. In such a condition, when the robot oscillations become relatively large, the high frequency components are mostly not affecting the signals anymore. This could also explain the difference in the oscillation amplitude when running on carpet with respect to concrete and tile. In fact, as the carpet is more rough and has a higher friction coefficient than concrete and tile, the frequency of the vibrations coming from the tire/ground interaction forces might get closer to the resonance frequency for lower angular velocities. Moreover, the fact that there are, usually, two low frequency components may derives from two different resonance frequencies for the Roll and Pitch motion.
In order to understand how the robot inertia contributes to large vibrations, the same tests were performed by using the robot platform without the laptop supporting structure. As the robot behavior and the acquired data qualitatively do not change between the concrete and the tile floor, we will be considering only the data acquired on concrete and carpet. Similarly to the previous case, some example graphs of the low-pass filtered data with their FFT are provided in Appendix B. The graphs representing the average of the data amplitudes and the average of the three highest peak frequencies with respect to the angular velocity ω_z^* and the type of ground are also provided in Appendix B.
By looking at the graphs depicted in Appendix B, we can claim that:

- Large amplitude vibrations never appears.
- Low frequency components nearly do not appear in the FFT of the data.
- The peak frequencies seem to periodically increase when increasing the robot angular

velocity (dotted lines in Figure 6.19, 6.20).

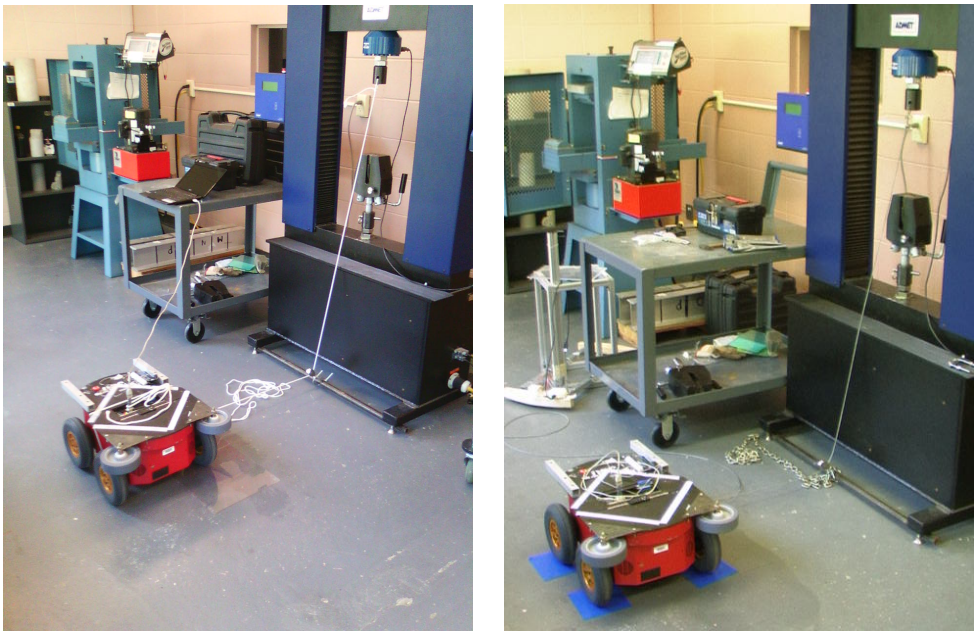
- The peak frequencies get less remarkable (data more irregular) when increasing the robot angular velocity.
- The rate of the increase of the peak frequencies (slopes of the dotted lines in Figure 6.19, 6.20) decreases when the robot angular velocity is increasing.

By comparing the peak frequencies obtained when swiveling the robot with and without the aluminum frame at relatively low angular velocities ($\omega_z^* < 30\,\frac{deg}{s}$), we can most likely deduce that, as it was previously supposed, such frequency components might derive from the tire reaction forces. However, we can not exclude yet that such frequency components derive from the robot structural vibrations.
Finally, to better identify the origin of the previously discussed frequency components, we fixed the wood-cube to one tire and we acquired and analyzed the data as the previous cases. However, in this case, the data are acquired only for $\omega_z^* < 20\,\frac{deg}{s}$, because of the unacceptable wire twisting when spinning the wheels with higher angular velocities. Some example graphs of the low-pass filtered data with their FFT are provided in Appendix B. The graphs representing the average of the data amplitudes and the average of the three highest peak frequencies with respect to the angular velocity ω_z^* and the type of ground are also provided in Appendix B. By looking at those graphs, we can clearly identify one or two frequency components, depending on the type of ground, which almost linearly increase when increasing the robot angular velocity. Such a result allows us to claim that the identified frequency components most likely derive from the tire reaction forces.

3.3 Characterization of the Tire Forces

In this section, the identification of the tire force is provided by presenting the experimental data acquired from the force sensor, while its mathematical model will be deeply discussed in the next chapter. In particular, the force is measured by using the `ADMET` testing system machine and the data are acquired using `MTESTQuattro` software for Windows, provided by the same company, and then exported and analyzed in `Matlab`.
In order to have a general idea about the tire longitudinal and lateral force, the robot was

first pulled backwards and sidewards by tying one side of a rope to the robot platform and the other side to the force sensor on the testing system machine. As the testing machine moves only vertically while the robot must be pulled horizontally, the rope passed through a pulley in between the robot and the sensor. It is worth noticing that, since the pulley has very low friction, the mass of the rope is relatively low and the angle between the sensor and the pulley is small, we can consider with good approximation that the vertical force measured by the sensor coincides with the horizontal force pulling the robot. Figure 3.3(a) shows the measurement setup previously described.



(a) (b)

Figure 3.3: Tire lateral force measurement setup: (a) Sidewards pull using a rope with four wheels touching on concrete; (b) Sidewards pull using a steel cable with the wheels on four oiled plastic pads.

To measure the tire longitudinal force, the robot's platform was pulled backwards at different constant velocities (v_x), first with the wheels stopped, and then by spinning the wheels at different angular velocities until the motors stalled. Figure 3.4 depicts a plot of the tire longitudinal force measured by the force sensor. The first part, i.e. approximatively from zero to five centimeters, corresponds to the static force measured with the wheels stopped, and the second part, i.e. nearly from five centimeters to the end, corresponds to the kinetic force measured with the wheel spinning. From the first part of the graph in Figure 3.4, it can be noticed that the force presents a saw-tooth shape, as if an elastic material was pulled and then suddenly released. By looking at the wheels while performing the test, we noticed that,

when the force was increasing, the wheels slightly turn because of the motor belt stretching. Immediately after reaching the maximum static force, the belt releases due to its elasticity and the wheels return to their initial position. Thereby, we can claim that the motor belt behaves like a spring-damper system. Moreover, as the testing machine is measuring the sum of the elastic reaction forces provided by the two belts, we can consider as we were measuring the resultant force of two springs connected in parallel. Thereby, if we consider two identical motor belts, the elasticity coefficient perceived by the force sensor can be written as:

$$K_{measured} = K_{belt1} + K_{belt2} = 2K_{belt} \tag{3.8}$$

As a consequence, we can estimate belt elasticity coefficient by measuring the slope of the ramps of the saw-tooth shape, and dividing it by two. In particular, the slopes vary from 5 $\frac{kN}{m}$ to 7 $\frac{kN}{m}$, therefore the belt elasticity coefficient can be estimated as $K_{belt} = 3\ \frac{kN}{m}$.
By looking at the second part of the graph in Figure 3.4, instead, it can be noticed that, although the force reaches higher values than static force, the force behavior is completely irregular and unpredictable due to the motor belt dynamics and especially to the complex wheel/ground interaction when the wheel are slipping. As the measured longitudinal force does not provide any significant information on the tire dynamics and slipping, in the following we will be focused on the tire lateral force, which plays an important role in skid-steering motion.

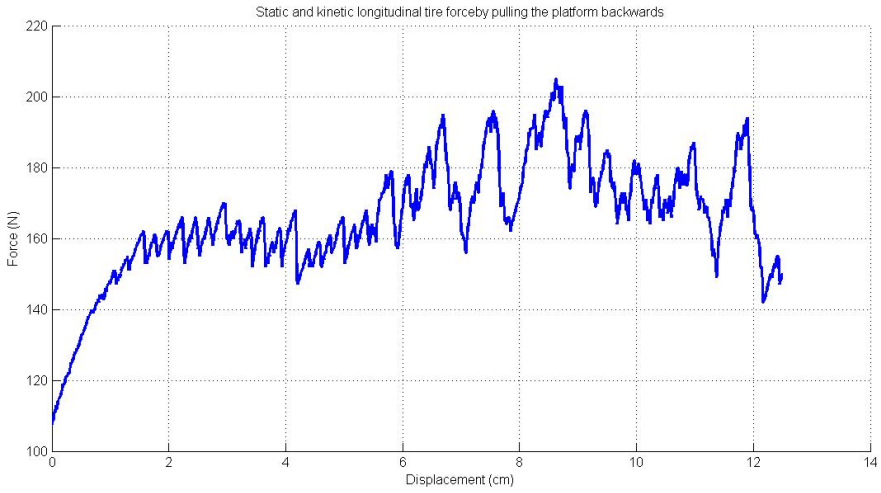
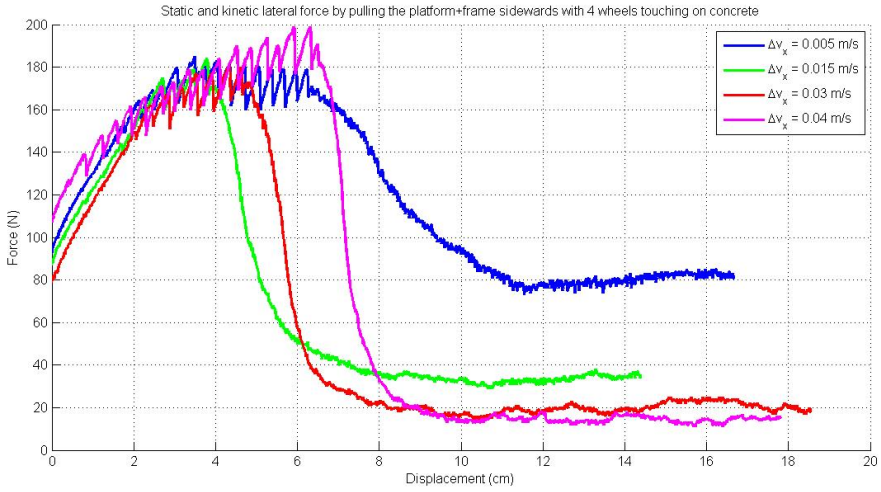
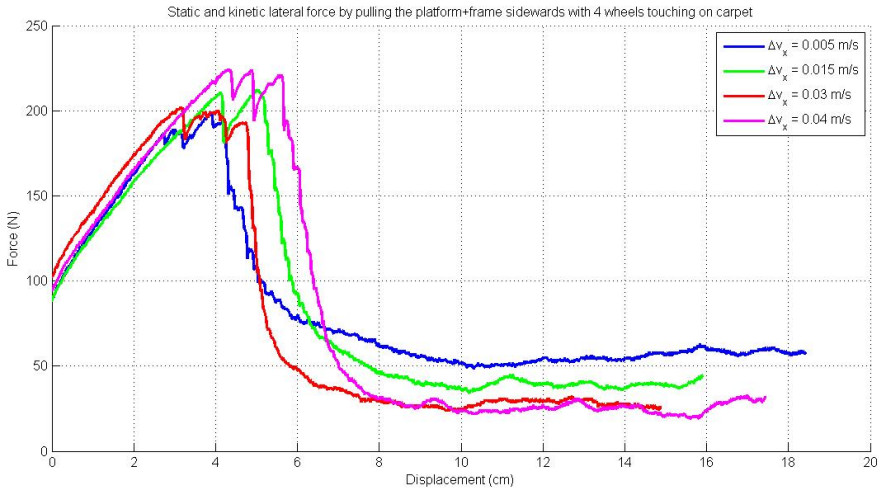


Figure 3.4: Static and kinetic longitudinal force by pulling the robot backwards with a rope at $v_x = 1\ \frac{mm}{s}$.

To measure the tire lateral force, the robot is pulled by the testing machine with different weight and at different constant lateral velocities (v_y) both on the carpet and concrete floor. First, the static lateral force is measured by having the wheels stopped, then also the kinetic lateral force is measured by spinning the wheels in place ($\Delta v_{ix} \neq 0$). The graphs in Figure 3.5 depict the static and kinetic force measured by the sensor with respect to the distance covered by the testing machine.



(a)



(b)

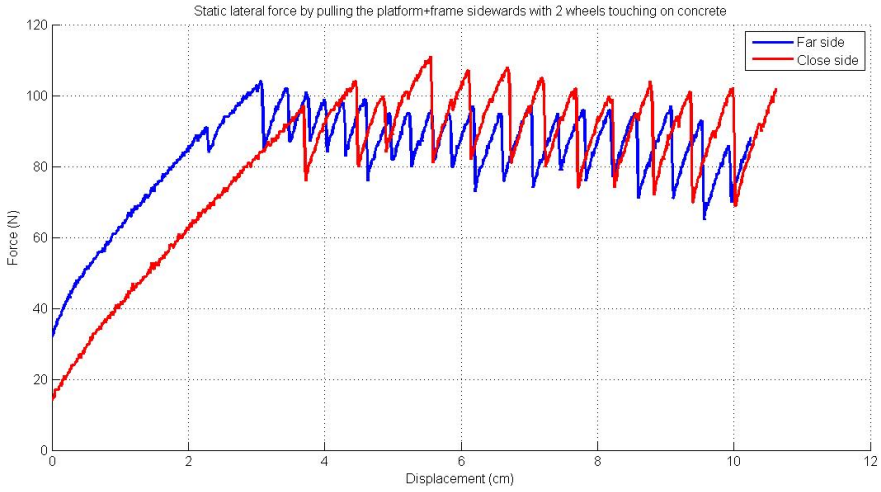
Figure 3.5: Static and kinetic lateral force by pulling the robot's platform sidewards with a rope at $v_y = 1\ \frac{mm}{s}$, with four wheels touching on: (a) Concrete; (b) Carpet.

By looking at the graphs in Figure 3.5, it can be noticed that, when the wheels are stopped (first part), the force presents a sort of saw-tooth shape while, when the wheels are spinning

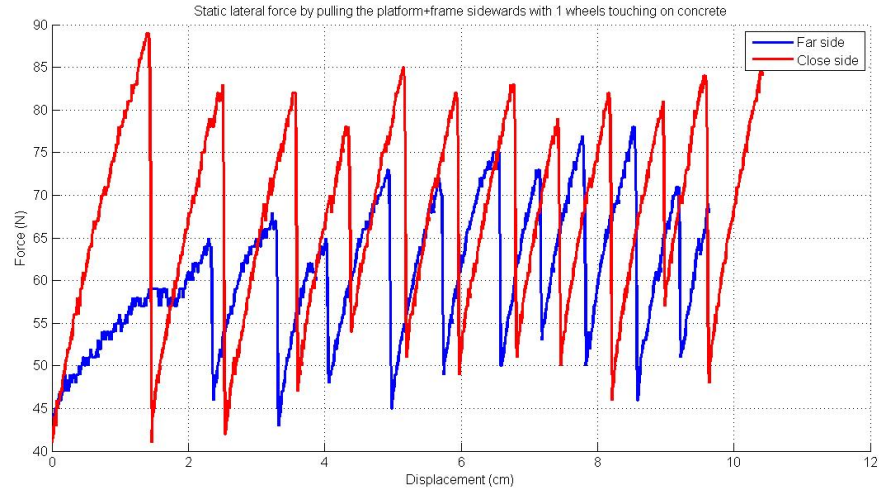
in place (second part), the force settles down nearly to a steady value. This steady value depends on the wheel slipping until a certain threshold after which it remains almost constant. In particular, the force decreases almost linearly for $\Delta v_{x_i} < 0.04 \frac{m}{s}$, while for $\Delta v_{x_i} \geq 0.04 \frac{m}{s}$ it remains almost constant around its minimum value between $20 - 30\,N$ depending on the type of the ground. By dividing for the weight of the robot ($gm_{platform} = 225\,N$), we obtain an estimation of the minimum kinetic friction coefficient for $v_y = 1 \frac{mm}{s}$, i.e. $\mu_{k_{min}} = 0.1, 0.12$. An estimation of the static friction coefficient, instead, can be obtained by considering the maximum force recorded in the first part, which varies between $180 - 200\,N$ for the concrete and $200 - 224\,N$ for the carpet, and dividing it for the weight of the robot. In particular, by computing the average value among all the coefficients estimated for each trial, we obtain a static friction coefficient $\mu_s = 0.86, 0.98$ respectively for the concrete floor and carpet. However, as the qualitative behavior for both the static and kinetic force does not change for different types of ground, only the data acquired on the concrete floor will be considered in the following. Another consideration is that the irregular structure of the carpet causes a significant variation on the force recorded during different tests, especially for the static force (see the first part of Figure 3.5(b)), providing not very accurate and reliable data.
In order to better understand and analyze the saw-tooth behavior resulting when the wheels are stopped, the robot was also pulled laterally but placing some oil and a piece of smooth plastic underneath two and three wheels, allowing respectively only two and one wheel touching with the ground while the others slide upon the floor nearly without friction. The graphs in Figure 3.6 depict the force measured in the two cases, using as touching wheels both the ones further and closer to the pulling side.

Figure 3.6 shows very well the saw-tooth shape of the force even when only one wheel is touching, meaning that such a behavior is not due to the complex interaction among the four wheels but it is due to some tire properties. By looking carefully at the wheels while pulling the robot, it can be noticed that such a behavior is due to the rubber of the tire and can be explained by thinking about the tire as a spring-mass-damper system. In fact, when the robot is pulled sidewards and the wheels are not slipping, the contact point of each wheel stays fixed to the ground, allowing the rubber of the tire to be stretched as a spring, until the spring force overcome the static friction force. Then, the contact point moves due to the tire-spring releasing and then it stops allowing the rubber to stretch again[1]. As the dynamics

[1] See Section 4 for further details.



(a)



(b)

Figure 3.6: Static lateral force by pulling the robot's platform sidewards with a rope at $v_y = 1\ \frac{mm}{s}$, sliding on concrete on: (a) Two wheels; (b) One wheel.

of the tire stretching is much slower than the releasing, the periodic motion results in a sort of saw-tooth wave as shown from the graphs.

Furthermore, from the graphs depicted in Figure 3.6 we can also extract the tire elasticity coefficient by calculating the slope of the ramps of the saw-tooth shape. In particular, the slope varies from 4 $\frac{kN}{m}$ to 6 $\frac{kN}{m}$, therefore the measured elasticity coefficient can be estimated as $K_{measured} = 5\ \frac{kN}{m}$. However, such a value of $K_{measured}$ is not reliable as the rope is extensible so that the force sensor is also measuring the rope elastic force and therefore the previously calculated coefficient includes the rope coefficient too. We can consider our system composed by the robot and the rope as two springs connected in series, so that the elasticity coefficient the sensor perceives is given by the formula:

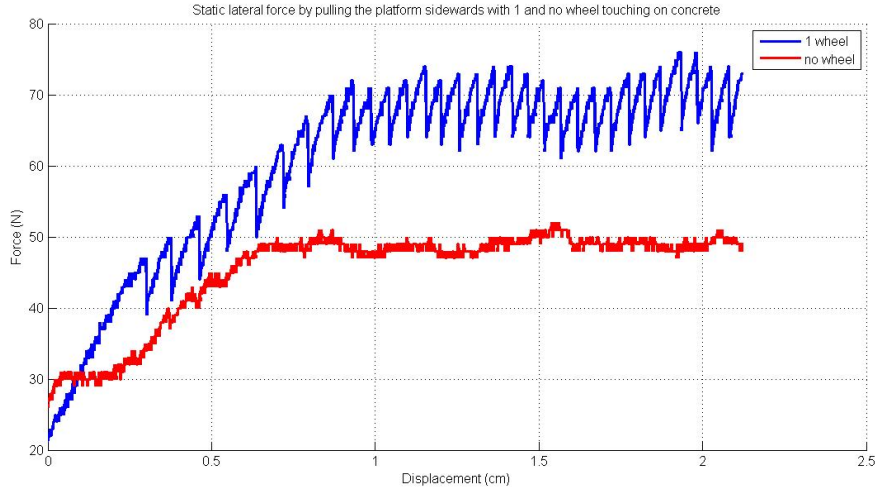
$$K_{measured} = \frac{K_{cable} K_{tire}}{K_{cable} + K_{tire}} \tag{3.9}$$

Thereby, when K_{cable} is comparable or smaller than K_{tire}, $K_{measured}$ is completely different from K_{tire} and can even be almost equal to K_{cable}.

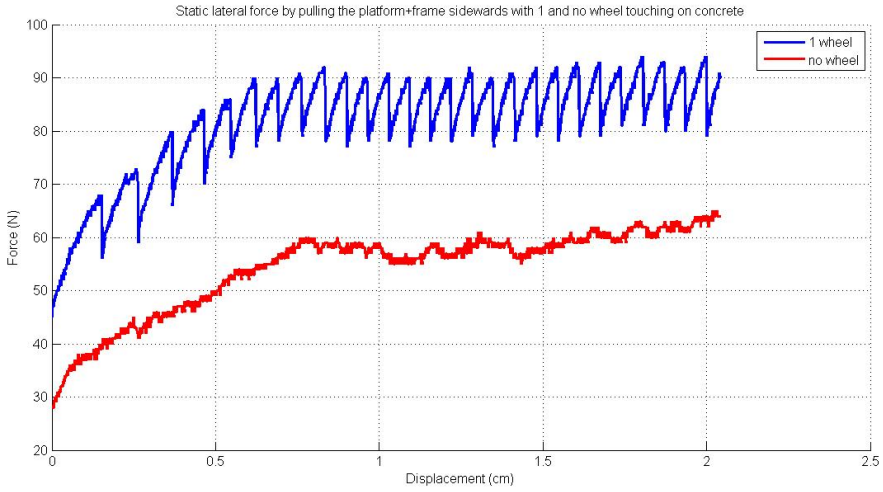
To overcome this problem, the rope was replaced with a steel cable which is not extensible. The force was then measured again for a single wheel touching the ground and also with no wheel touching (Figure 3.3(b)), in order to check whether or not the saw-tooth behavior was due only to the rope elasticity. Moreover, the force was measured by pulling only the robot platform with and without the aluminum frame to check if the weight or the inertia affect the measurements. Figure 3.7 shows the two cases with one and zero wheel touching.

By looking at Figure 3.7, it is easy to notice that, in both cases (with and without the aluminum frame), the saw-tooth shape is not present when all the four wheels are on the plastic pads with some oil underneath, meaning that such a behavior is not due to the rope elasticity or the robot inertia but only to the tire-spring dynamics. We can also see how the saw-tooth shape is more regular in these graphs than the ones in Figure 3.5, 3.6, thanks to the high elasticity of the cable. Moreover, as in this case $K_{cable} \gg K_{tire}$ and only one wheel is touching the ground, the value of $K_{measured}$ defined in (3.9) is almost equal to K_{tire}. In particular, the slope of the little ramps varies in the interval $20 - 25\ \frac{kN}{m}$ so that we can estimate the tire elasticity coefficient as $K_{tire} = 22.5\ \frac{kN}{m}$.

Although the data presented above clearly show the tire-spring behavior from a qualitative point of view, they do not provide reliable values for the identification of the tire elasticity coefficient. In fact, the measured force is somehow affected also from the other tire forces,



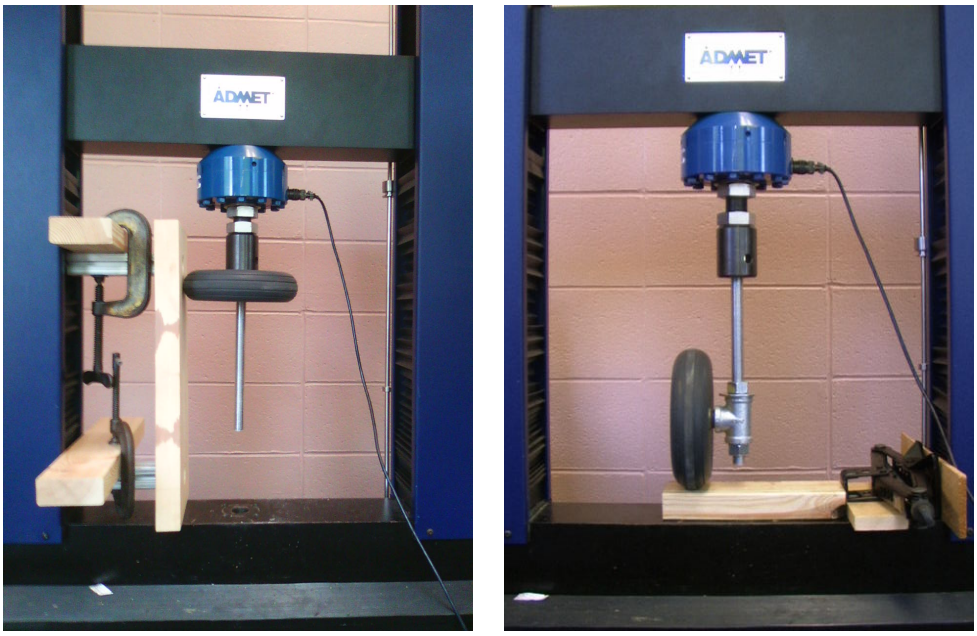
(a)



(b)

Figure 3.7: Static lateral force by pulling the robot sidewards with a steel cable at $v_y = 1\ \frac{mm}{s}$, sliding on concrete using: (a) only the platform; (b) the entire robot.

even if the three wheels are not touching the ground. Also, from those data is still unknown why and how the tire can release after its stretching overcome the static friction force. In order to , more data were acquired from the force sensor by fixing a single wheel to the sensor and dragging it up and downwards upon a piece of wood, as shown in Figure 3.8(a). The wood surface was also tilted by different angles so that the wheel can feel different normal reaction forces to the surface. Thereby, the static friction force either increases or decreases while dragging the wheel upon the piece of wood, depending on its slope.

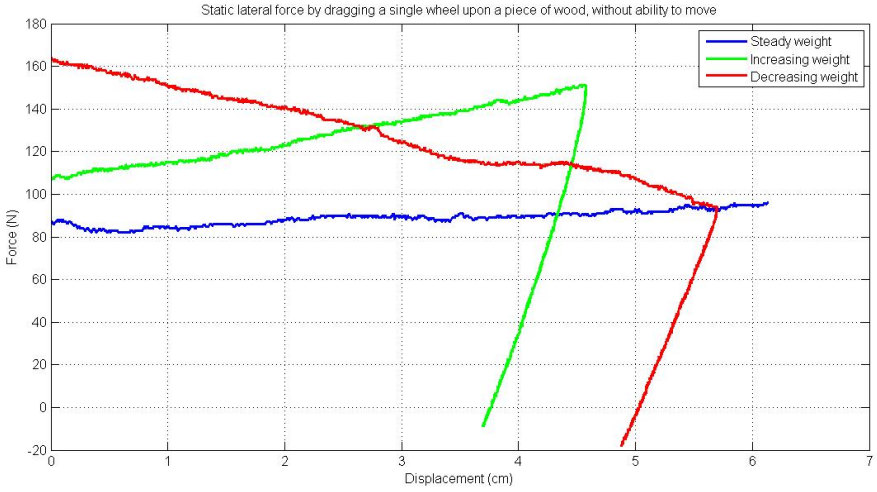


(a) (b)

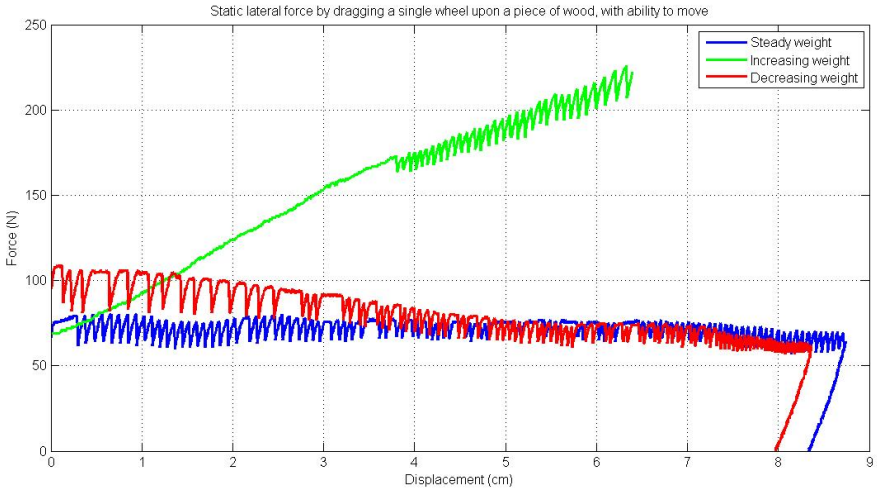
Figure 3.8: (a) Tire lateral force measurement by dragging a single wheel upon a piece of wood; (b) Tire vertical force measurement by dragging a single wheel upon a piece of wood.

First, the wheel was dragged on the wood surface after fixing it to the sensor in such a way it could not move in any direction, then it was fixed so that it could move along the direction perpendicular to the surface. Also, in both cases, the wheel was first pushed downwards, dragging it on the wood surface, and then was slowly pulled upwards until the measured force went to zero because of the tire releasing. As a consequence, by measuring the slope of the last ramp, the tire elasticity coefficient can be accurately identify. Some example graphs of the two experiments are depicted in Figure 3.9.

The graphs in Figure 3.9 show that the tire can release after stretching only when the wheel has the capability to move perpendicularly to the surface. Furthermore, by looking at the last part of the graphs, it can be noticed that the tire-spring dynamics is non-linear,



(a)

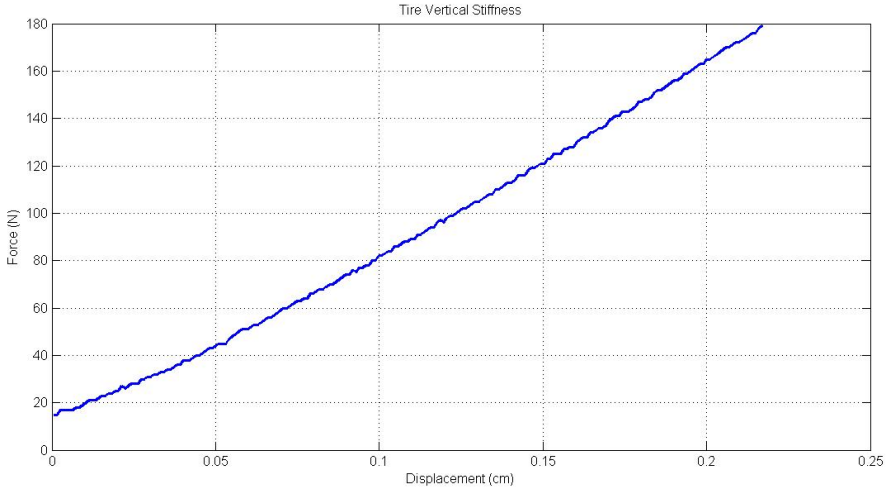


(b)

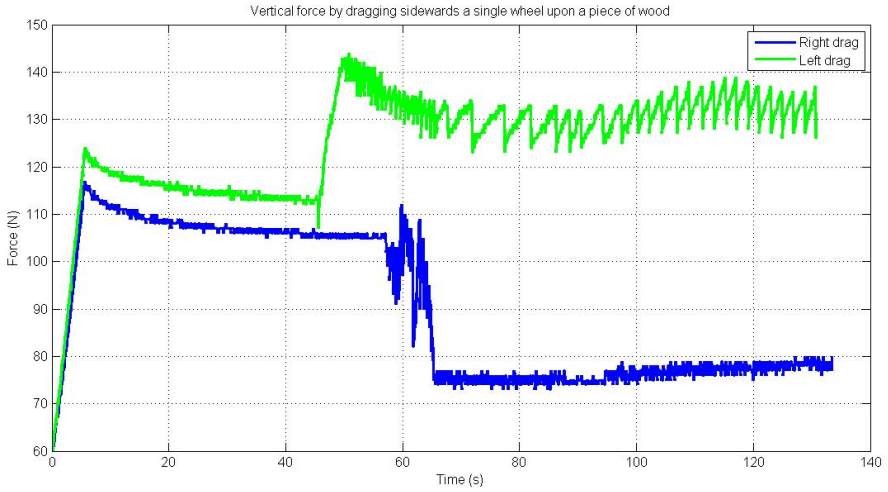
Figure 3.9: Static lateral force by dragging a single wheel upon a piece of wood: (a) without the ability to move perpendicularly to the drag surface; (b) with the ability to move perpendicularly to the drag surface.

as expected. However, by calculating the slope for relatively small intervals, the nonlinear dynamics can be reasonably approximated by two linear pieces, the first one from $0\,N$ to $90\,N$ with $K_{tire} = 15\,\frac{kN}{m}$ and the second one from $90\,N$ to $150\,N$ with $K_{tire} = 21\,\frac{kN}{m}$, therefore we can estimate the tire elasticity coefficient as $K_{tire} = 18\,\frac{kN}{m}$.
The last parameter we want to identify is the tire vertical force due to the tire-spring dynamics. In order to measure the tire vertical force, a single wheel was fixed to the force sensor by a steel tube with a 'L' shape (see Figure 3.8). The testing machine pushed the wheel downwards for few centimeters, so that a vertical load was applied on the wheel and the vertical force due to the tire compression could be measured with respect to distance (see Figure 3.10(a)). Then, the piece of wood was pulled sidewards, so that the wheel was dragged upon the wood surface and the vertical force was measured with respect to time (see Figure 3.10(b)).

The graph in Figure 3.10(a) shows that also the tire-spring dynamics during vertical compression is approximatively linear and by calculating its slope we obtain a tire vertical elasticity coefficient $K_{tire_z} = 80\,\frac{kN}{m}$.
The graph in Figure 3.10(b), instead, seems to confirm the results obtained for the graphs in Figure 3.9. When the wheel has no capability to move vertically, i.e. when wheel is dragged towards the right hand side, the force stays around a steady value meaning that the tire is not releasing. Conversely, when it has such a capability, i.e. when wheel is dragged towards the left hand side, the force presents a saw-tooth shape meaning that the tire always releases after stretching. The reason why the vertical force increases when the tire is stretching and decreases when releasing (left hand side), is probably due to 'L' shape tube since, when the tire stretches, the contact point moves a little closer to the sensor vertical axis and therefore the steel rod is pushed a little upwards increasing the measured force. Conversely, when the tire releases, the contact point moves a little further from the sensor vertical axis and therefore the steel rod is allowed to move a little downwards resulting in a decrease of the perceived force.



(a)



(b)

Figure 3.10: Tire vertical force by dragging a single wheel upon a piece of wood: (a) Right side, no ability to move perpendicularly to the drag surface; (b) Left side, ability to move perpendicularly to the drag surface.

Chapter 4

Modeling of the Tire Reaction Forces

In Section 3.3, different measurements of the tire lateral force were presented to fully understanding the dynamic causes of the robot jerky motion. To explain the saw-tooth behavior recorded by the force sensor, the tire spring-damper model for the static lateral force was introduced, without entering in details. The elasticity coefficients K_{tire_y}, K_{tire_z} was also identified for both the lateral and vertical motion.
In this section, we first provide a complete description of the tire spring-damper model for the tire lateral force. Then, a model for the three tire reaction forces along the three principal axes is defined to complete the SSMR dynamic model presented in Section 2.2.

4.1 Tire Lateral Force

Let us consider a spring-mass-damper system attached to a local frame (y, z) moving with respect to a fixed frame (Y, Z), as depicted in Figure 4.1.

Let m_t, K_y, D_y be respectively the mass and the elasticity and damping coefficient of the spring, and μ_{sy}, μ_{ky} respectively the static and kinetic friction coefficient at the contact point between the mass and the floor. Let us consider the local frame start moving at a velocity $v_{iy}(t)$ with respect to the fixed frame, expressed in the local frame. Supposing that the origin of the local frame is fixed to the zero position of the spring and its y-axis is parallel to the spring extension/compression direction, if the mass is not moving the spring displacement with respect to the local frame is defined as:

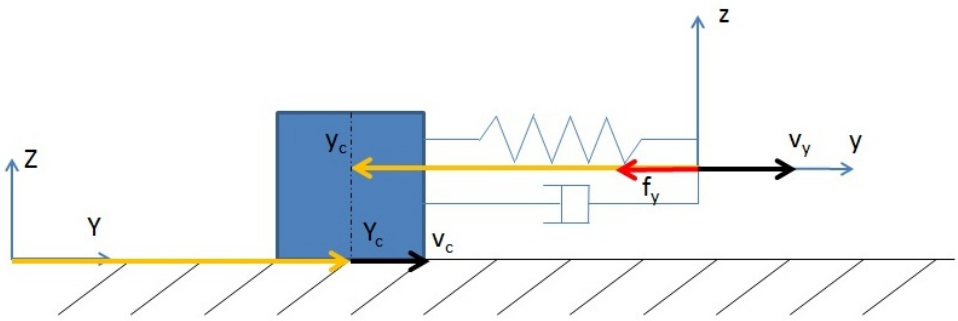


Figure 4.1: A scheme of a spring-mass-damper system with friction between the mass and the ground.

$$\Delta Y = Y_{ic} - Y_i = y_{ic} = -\int_0^t v_{iy}(\tau)d\tau \tag{4.1}$$

where y_{ic} denotes the position of the contact point with respect to the local frame.
The mass does not move until the resultant of all the forces applied to the contact point (F_{ic}) does not overcome the maximum static friction force $F_{s_{max}} = \mu_{sy}N_i$, i.e. until the condition $|F_{ic}| \leq F_{s_{max}}$ is satisfied, with N_i representing the normal reaction to the ground. Conversely, when such a condition is not anymore satisfied, the mass starts moving and the contact point position is described by the following equation:

$$m_t\ddot{y}_{ic} + D_y\dot{y}_{ic} + K_y y_{ic} = F_{ky} = -\mu_{ky}N_i sign(v_{ic}) \tag{4.2}$$

and relation (4.1) becomes:

$$\Delta Y = y_{ic} = \int_{t_k}^t (v_{ic}(\tau) - v_{iy}(\tau))d\tau \tag{4.3}$$

where $v_{ic} = \dot{Y}_{ic}$ represents the velocity of the contact point with respect to the fixed frame, expressed in the local frame.
It is worth noticing that relation (4.3) still holds if from t_k the mass is not moving, i.e. $v_{ic} = 0$. By considering v_{iy} as input of the system and substituting (4.3) in (4.2), we obtain the following equation:

$$m_t(\dot{v}_{ic}(t) - \dot{v}_{iy}(t)) + D_y(v_{ic}(t) - v_{iy}(t)) + K_y \int (v_{ic}(t) - v_{iy}(t))dt = -\mu_{ky}N_i sign(v_{ic}) \tag{4.4}$$

Let us suppose that the contact point starts moving with $v_{ic} > 0$, so that $sign(v_{ic}) = 1$. By solving equation (4.4) in the frequency domain, we obtain:

$$V_{ic}(s) = V_{iy}(s) - \frac{\mu_{ky} N}{m_t s^2 + D_y s + K_y} \tag{4.5}$$

where $V_{ic}(s)$ and $V_{iy}(s)$ are the Laplace transform of v_{ic} and v_{iy} respectively.

By calculating the Laplace inverse transform of equation (4.5), we obtain the following relation in the time domain:

$$v_{ic}(t) = v_{iy}(t) - \mu_{ky} N_i h(t) \tag{4.6}$$

where $h(t) = \mathfrak{L}^{-1} \frac{1}{m_t s^2 + D_y s + K_y}$.

We notice that, if the parameters m_t, D_y, K_y are defined such that the system is stable, the contribution of the friction in (4.6) initially increases, reaching its maximum, and then goes exponentially to zero, either oscillating or directly, depending on whether the system is under or over-damped. Thereby, if we suppose that $v_{ic}(t) > 0 \quad \forall t$, i.e. $v_{iy}(t) > \mu_{ky} N_i h(t) \quad \forall t$, the contact point will keep moving with a velocity $v_{ic}(t) \rightarrow v_{iy}(t)$. Conversely, if we have $v_{ic}(t) < 0$ for some values of t, equation (4.6) does not hold anymore, as the friction force changes its sign, and the nonlinearity of the $sign()$ function can affect the system stability. However, it can be proved, by substituting the function $sign(v_{ic})$ with $\arctan(k_s v_{ic})$, k_s ¨big¨, that the system is at least simply stable [21]. In particular, if $|v_{ic}|, |\dot{v}_{ic}|$ become relatively small, i.e. $|v_{ic}|, |\dot{v}_{ic}| < \frac{\mu_{ky} N_i}{k_s}$, we can consider as if the contact point stops. Thereby, the system suddenly switches to the static configuration, in which the friction coefficient switches from the kinetic to the static one, so that the contact point does not move until the elastic force does not overcome the static friction force, i.e. until the condition $|F_{ic}| \leq F_{s_{max}}$ is satisfied.

The system describing the dynamics of the contact point presents, therefore, a sort of hysteresis. More precisely, when the condition $|F_{ic}| \leq F_{s_{max}}$ is not anymore satisfied the dynamics switches from $v_{ic} = 0$ to (4.4), and when the condition $|v_{ic}|, |\dot{v}_{ic}| < \frac{\mu_{ky} N_i}{k_s}$ is satisfied it switches from (4.4) to $v_{ic} = 0$, and so on.

By considering the local frame in Figure 4.1 as the frame fixed to the i^{th} wheel and taking into account the aforementioned considerations, the lateral reaction force perceived from the i^{th} wheel can be written as:

$$f_{iy} = -D_y(v_{iy}(t) - v_{ic}(t)) - K_y \int (v_{iy}(t) - v_{ic}(t))dt$$

where the dynamics of v_{ic} is either $v_{ic} = 0$ or (4.4), depending whether the i^{th} contact point is moving or not, and the wheel center velocity v_{iy} is related to the robot linear and angular

velocity by equation (2.55).

The first condition, which switches the system dynamics from $v_{ic} = 0$ or (4.4), can be found by noticing that equation (4.4) still holds when the contact point is not moving if we substitute F_{ky} with F_{sy}. In fact, by isolating the term $m_t \dot{v}_{ic}$ in equation (4.4), we find the Newton's law applied to the contact point, which can be written as:

$$m_t \ddot{Y}_{ic} = F_{ic} + F_{sy} = F_{inertia} + F_{elastic} + F_{sy} = F_{tot} \tag{4.7}$$

where $F_{inertia} = m_t \dot{v}_{iy}$ is the force given by the inertia of the local frame, $F_{elastic} = D_y(v_{iy}(t) - v_{ic}(t)) + K_y \int (v_{iy}(t) - v_{ic}(t))dt$ is the elastic force given by the spring and F_{sy} is the static friction force. The static friction force can be considered as a non linear reaction force, which is always opposite to the sum of all the forces applied to the contact point until its maximum/minimum value $\pm F_{s_{max}}$. Thereby, we have $F_{tot} = 0$ until the condition $|F_{ic}| \leq F_{s_{max}}$ is satisfied, then the friction force becomes a kinetic friction force ($F_{sy} = F_{ky}$) and $F_{tot} = F_{ic} + F_{ky}$ until the contact point stops again.

The second condition, i.e. $|v_{ic}|, |\dot{v}_{ic}| < \frac{\mu_{ky} N_i}{k_s}$, is not easily implementable in a simulation environment and also requires the knowledge of the parameters k_s. In order to provide a condition based on the resultant force applied to the contact point, as for the first condition, we look at the steady value of equation (4.4). By substituting $v_{ic}, \dot{v}_{ic} = 0$ in (4.4), we find the equilibrium condition on the resultant force:

$$F_{ic} = -F_{ky} \tag{4.8}$$

We notice that such a condition is a necessary but not sufficient condition to guarantee that the contact point stops, i.e. $|v_{ic}|, |\dot{v}_{ic}| \approx 0$. The fact that the contact point either stops or keep moving after the condition in (4.8) is satisfied depends on the way the system reaches the equilibrium, which depends on whether the system in (4.4) is under or over-damped. We can consider (4.8) as a necessary and sufficient condition for the switching of the system dynamics if and only if the system described by equation (4.4) is over, critically or slightly under-damped, so that the equilibrium is reached nearly directly without oscillations and overshoots. In our case, by observing the tire behavior while dragging the robot sidewards and looking at the data presented in Section 3.3, we can claim that the dynamics describing the tire releasing after its stretching is always critically or over-damped as the contact point stops without any vibration of the tire. Such behavior may derive from the partic-

ular tire dynamic parameters (m_t, K_y, D_y), which might provide an over-damped response ($D_y > 2\sqrt{m_t K_y}$), but also from the kinetic friction force. In fact, the kinetic friction force depends on the normal reaction force N_i, which is not given only by the mass of the tire but it increases as the contact point gets closer to its equilibrium position, perceiving also the robot weight. Such increasing of the friction force as the tire goes to its equilibrium position would make the contact point stop without oscillating even if the system response was under-damped.

In conclusion, by taking into account all the above considerations, we can rewrite the equation describing the tire lateral reaction force and the corresponding conditions on the contact point, as follows:

$$
\begin{aligned}
&f_{iy}(t) = -D_y(v_{iy}(t) - v_{icy}(t)) - K_y \int (v_{iy}(t) - v_{icy}(t))dt \\
&m_t \dot{v}_{icy}(t) = F_{icy}(t) \\
&F_{icy}(t) = \begin{cases} 0, & \text{for } |F_{iy}(t)| < \mu_{sy} N_i \wedge v_{icy}(t - \Delta t) = 0 \\ F_{iy}(t), & \text{for } |F_{iy}(t)| > \mu_{ky} N_i \wedge v_{icy}(t - \Delta t) \neq 0 \end{cases} \\
&F_{iy}(t) = m_t \dot{v}_{iy}(t) + D_y(v_{iy}(t) - v_{icy}(t)) + K_y \int (v_{iy}(t) - v_{icy}(t))dt - \mu_{ky} N_i sign(v_{icy}(t))
\end{aligned}
\tag{4.9}
$$

where v_{iy} is related to the robot linear and angular velocity by equation (2.55).

In all the equations above, we have considered the static and kinetic friction coefficient as constants. This is true only when the wheels are not spinning. Conversely, if the wheels are spinning, the static and kinetic friction coefficient change with respect to the wheel angular velocity (ω_i), the longitudinal slip velocity ($\Delta v_{ix} = v_{ix} - r\omega_i$) and the wheel lateral velocity (v_{iy}). For a sake of simplicity, we can consider both of them proportional to a function $\mu_{iy}(\omega_i, \Delta v_{ix}, v_{iy})$, therefore we can write:

$$
\begin{aligned}
\mu_{sy} &= \mu_{s_{max}} \mu_{iy}(\omega_i, \Delta v_{ix}, v_{iy}) \\
\mu_{ky} &= \mu_{k_{max}} \mu_{iy}(\omega_i, \Delta v_{ix}, v_{iy})
\end{aligned}
\tag{4.10}
$$

where $\mu_{s_{max}}, \mu_{k_{max}}$ represent the maximum value for the static and kinetic friction coefficient respectively, i.e. the values of the coefficients when the wheels are not spinning. We notice that, as μ_{sy}, μ_{ky} can not be greater than their maximum value, the function μ_{iy} must satisfy the condition:

$$0 \leq \mu_{iy}(\omega_i, \Delta v_{ix}, v_{iy}) \leq 1$$

A qualitative representation of the function μ_{iy} can be obtained by first considering that, when the wheel slip is relatively high ($|\lambda_i| > \lambda_{max}$), i.e. the slip velocity is relatively high compared to the wheel spinning, the friction coefficient is proportional to the ratio between the wheel lateral velocity and the slip velocity. In fact, when v_{iy} is relatively high with respect to Δv_{ix}, we can consider as if the contact point is not moving along the longitudinal direction and therefore we can consider $\mu_{iy} \approx 1$. Conversely, when v_{iy} is relatively low with respect to Δv_{ix}, we can consider as if the contact point is moving along the longitudinal direction much faster than the wheel is moving along the lateral direction, therefore we can consider $\mu_{iy} \approx \mu_{min}$. The value μ_{min} depends on the wheel lateral velocity and can be zero only if $v_{iy} = 0$. An example of $\mu_{k_{min}}$ for $v_{iy} = 1 \frac{mm}{s}$ is provided in Section 3.3 (Figure 3.5). We notice that in such a case the tire lateral reaction force does not present the saw-tooth behavior as for the case $\Delta v_{ix} = 0$. This is due to the fact that, when v_{iy} is relatively low with respect to Δv_{ix}, the difference between μ_{sy} and μ_{ky}, therefore the difference between F_{sy} and F_{ky}, become so small that the tire stretching/releasing behavior is negligible, also because the saw-tooth would have such a high frequency that the robot structure would filter it out. In fact, if we consider the data acquired for the concrete floor, we have $\mu(0.36, -0.36, 0.001) = \frac{\mu_{k_{min}}}{\mu_{k_{max}}} \approx \frac{\mu_{s_{min}}}{\mu_{s_{max}}} \approx 0.12$, therefore we obtain $F_{s_{min}} - F_{k_{min}} = (\mu_{s_{max}} - \mu_{k_{max}})\mu(0.36, -0.36, 0.001)mg \approx 2\,N$, which is nearly the accuracy of the force sensor.

The proportionality to the ratio between the wheel lateral velocity and the slip velocity is not sufficient to describe μ_{iy} when the wheel slip is relatively low ($|\lambda_i| < \lambda_{max}$). In fact, if we consider $\Delta v_{ix} \approx 0$ but $\omega_i, v_{iy} \neq 0$, for instance when the robot is moving along a straight line with an external lateral force, by only considering $\mu_{iy} \propto \frac{v_{iy}}{\Delta v_{ix}}$ and constraining it to 1 as maximum value, we would obtain $\mu_{sy} = \mu_{s_{max}}, \mu_{ky} = \mu_{k_{max}}$, which is the same result obtained when the robot is not moving. We empirically verified that such a result is not true, as the tire lateral force decreases when the robot is moving along a straight line, and in particular is proportional to the ratio $\frac{v_{iy}}{\omega_i}$.

The aforementioned characteristic of the function μ_{iy} can be analytically expressed by considering the function proposed by Song *et al.* in [4], provided in (2.27),(2.30). First of all, we observe that, by combining (2.28) with (2.8),(2.10), the following relation holds:

$$\tan\theta_i = \frac{v_{iy}}{v_{ix} - r\omega_i} \tag{4.11}$$

Furthermore, we notice that the instable behavior, due to the difference between the static and kinetic friction coefficient, provided by the parameter α_s in (2.30),(2.31) [4], was already included in the previously discussed dynamics of f_{iy}, because of the tire stretching and releasing. Thereby, we can consider, for a sake of simplicity, $\alpha_s = 0$ so that we can rewrite (2.30),(2.31), after normalizing by $\mu_{s_{max}}$, as follows:

$$\mu_{ix}(\omega_i, \Delta v_{ix}) = \begin{cases} k_s|\lambda_i|, & \text{for } |\lambda_i| \leq \lambda_{max} \\ 1, & \text{for } |\lambda_i| > \lambda_{max} \end{cases} \tag{4.12a}$$

$$\mu_{iy}(\omega_i, \Delta v_{ix}, v_{iy}) = \tan\theta_i \mu_{ix}(\omega_i, \Delta v_{ix}) \tag{4.12b}$$

where $k_s = \frac{1}{\lambda_{max}}$. As a consequence, only the identification of the parameter λ_{max} is required, instead of the two parameters λ_m, α_s.
We observe that the value of μ_{iy} coincides with θ_i when $|\lambda_i| > \lambda_{max}$, while we obtain $\mu_{iy} = k_s \frac{v_{ix}}{r\omega_i}$ when $|\lambda_i| \leq \lambda_{max}$, which corresponds to the expected behavior as described above. It is worth noticing that, because of the saturation of μ_{ix}, μ_{iy} can assume values greater than 1, therefore the saturation to 1 must be guaranteed also for μ_{iy}.

4.2 Tire Longitudinal Force

We can consider the tire modeled as the spring-mass-damper system in Figure 4.1 also for the reaction force along the x-axis. However, as the tire is not dragged on the ground but it rolls, in this case, the contact point is not dragged by the wheel center velocity as for the lateral motion. Conversely, we can consider the longitudinal slip velocity (Δv_{ix}) as driving velocity for the contact point. Thereby, under the same assumptions done for the lateral reaction force, the longitudinal reaction force perceived from the local frame can be described by the following equation:

$$
\begin{aligned}
&f_{ix}(t) = -D_x(\Delta v_{ix}(t) - v_{icx}(t)) - K_x \int (\Delta v_{ix}(t) - v_{icx}(t))dt \\
&m_t \dot{v}_{icx}(t) = F_{icx}(t) \\
&F_{icx}(t) = \begin{cases} 0, & \text{for } |F_{ix}(t)| < \mu_{sx} N_i \wedge v_{icx}(t - \Delta t) = 0 \\ F_{ix}(t), & \text{for } |F_{ix}| > \mu_{kx} N_i \wedge v_{icx}(t - \Delta t) \neq 0 \end{cases} \\
&F_{ix}(t) = m_t \dot{v}_{ix}(t) + D_x(v_{ix}(t) - v_{icx}(t)) + K_x \int (v_{ix}(t) - v_{icx}(t))dt - \mu_{kx} N_i sign(v_{icx}(t))
\end{aligned}
\tag{4.13}
$$

with the same meaning of the variables and parameters as for the lateral reaction force, where μ_{kx} is defined in (4.12a).

It is worth noticing that the parameters D_x, K_x depend not only on the elasticity and damping properties of the tire along the longitudinal direction, but they also depend on the elasticity and damping properties of the motor belt. In particular, we can consider the spring-damper for the tire and the one for the belt as they were in series, therefore the relations between K_x, D_x and the elasticity and damping coefficient of the tire (K_{tx}, D_{tx}) and the motor belt (K_b, D_b) can be written as follows:

$$
K_x = \frac{K_{tx} K_b}{K_{tx} + K_b}, \quad D_x = \frac{D_{tx} D_b}{D_{tx} + D_b} \tag{4.14}
$$

As K_{tx}, D_{tx} do not depend on the tire inflating properties, like for the lateral and vertical deflection, but only on the tire's rubber stretching properties, we usually have $K_{tx} >> K_b$ and $D_{tx} >> D_b$. Thereby, by looking at relation (4.14) we obtain:

$$
K_x \approx K_b, \quad D_x \approx D_b
$$

4.3 Tire Vertical Force

Let us consider the robot rotating around the x-axis and y-axis. In such a case, it is easy to deduce that the normal reaction forces at each contact point do not exactly correspond to the robot's weight distribution as when the robot does not rotate. The normal reaction force N_i perceived at each wheel, in fact, depends on the tire vertical compression, which depends on the wheel vertical position and velocity.

Similarly to what we have done for the tire lateral force, let us model the tire as a spring-

damper system along the z-axis, with one side fixed to the ground and the other one attached to a local frame (x,y) moving with respect to a fixed frame (X,Y), as depicted in Figure 4.2.

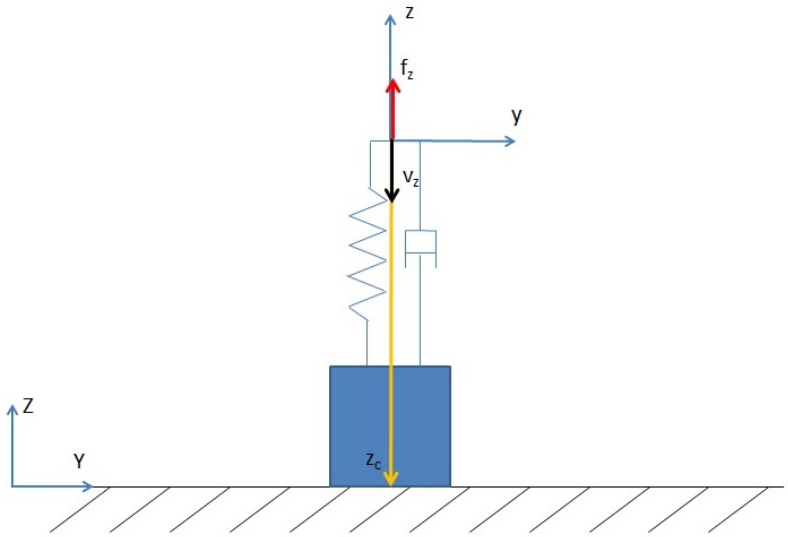


Figure 4.2: A scheme of a spring-mass-damper system moving along the vertical direction.

Let us consider the local frame start moving at a velocity $v_{iz}(t)$ with respect to the fixed frame, expressed in the local frame. As we consider the contact point not moving with respect to the fixed frame, it is straightforward define the reaction force perceived from the local frame as follows:

$$f_{iz} = N_i = -D_z v_{iz}(t) - K_z \int v_{iz}(t)dt \tag{4.15}$$

where the wheel center velocity v_{iz} is related to the robot linear and angular velocity by equation (2.55).

We notice that in (4.15) the gravitational force does not appear because the contribution of the gravity is already included in the robot dynamics determining the wheel vertical velocity v_{iz}. Moreover, although we considered the contact point not moving with respect to the fixed frame, it must noticed that, in reality, it is not fixed with the ground. Thereby, even if we will be considering in the following that the Roll and Pitch oscillations never make the wheel lift, the normal reaction force in (4.15) must be constraint to $N_i \geq 0$. Such a constraint will be represented, in the simulation, as a saturation on N_i, and therefore it introduces a nonlinearity which can cause the system instability.

Chapter 5

Identification of the Robot Parameters

In this chapter, the identification of all the geometric and dynamic parameters introduced in the previous chapters is provided. The measurements are provided by using standard laboratory instrumentation and the accelerometers presented in Section 1.2. The data are acquired respectively by using `LabVIEW` software as already mentioned, and then analyzed in `Matlab`.
The results presented here will be used in the next chapter for the simulation of the skid-steering motion, in order to likely reproduce the experimental data provided in Section 3.

5.1 Identification of the Geometric and Inertial Parameters

The mass of the robot, aluminum frame and laptop are measured by using an electronic scale with a precision of $10^{-2}\,g$, while their dimensions are measured with a standard measuring tape. In order to have an estimation of the planar position of the robot center of mass (CoM) with respect to the wheel/ground contact points, i.e. the x, y coordinates of the vectors $\boldsymbol{p}_i$, the robot is also weighted by putting on the scale only two wheels at the time and the other two wheels on few pieces of wood such that the robot is completely balanced, as depicted in Figure 5.1(a). It must be mentioned that the measurement procedure just described can not be as precise as the measurement of the global weight, since even a slope of few degrees between the scale and wood surface can result in a difference in few grams between the measured and real weight. However, as the setup is maintained constant for all the measurements, we can state that such a difference is nearly constant in every measurement, therefore the distribution of the weight on each wheel can still be obtained with enough precision with

respect to our purposes.

Moreover, to obtain the height of the CoM, i.e. the z coordinate of the vectors $\boldsymbol{p}_i$, the robot is tilted both on the left and right side until it reaches almost an equilibrium point, i.e. it does not fall on any side. Then, two straight lines perpendicular to the ground surface can be drawn starting from the wheel/ground contact point and the height of CoM with respect to the floor can be found purely by the geometric intersection of the two lines, as depicted in Figure 5.1(b). It is worth noticing that such estimation of p_{i_z}, which is constant for the four wheels, is reliable since we are considering the z axis of the robot frame perpendicular to the plane given by the four contact points.

(a) (b)

Figure 5.1: (a) Measuring the CoM position on x-y plane; (b) Measuring the CoM height.

Tab.5.1 list all the geometric parameters defined above, where the contact points are numbered clockwise starting from the right-rear wheel to the left-rear wheel.

$\boldsymbol{i}$	p_{x_i}	p_{y_i}	p_{z_i}
1	-0.15	0.18	-0.23
2	0.13	0.18	-0.23
3	0.13	-0.18	-0.23
4	-0.15	-0.18	-0.23

Table 5.1: Table of the geometric parameters.

To have a first estimation of the robot tensor of inertia with respect to its principal axes, the main components are measured and weighted, and their tensors of inertia are then calculated

with respect to their CoM for a frame parallel to the robot frame, by using standard formulas for regular geometric shapes. Then, the tensor of inertia of the robot can be computed by using the *Huygens-Steiner Theorem* and *Superimposition Principle* of moment of inertia for a rigid body, stating respectively:

$$I_O = I_{O'} + m\hat{OO'}\hat{OO'}^T$$

$$I = \sum_i I_i$$

where m, I are the mass and tensor of inertia of the body, $I_O, I_{O'}$ are the tensors of inertia calculated in two generic points O, O', $\hat{OO'}$ is the skew-symmetric matrix of the vector $\boldsymbol{OO'}$ and I_i is the tensor of inertia of the $i-th$ component of the body calculated with respect to the same frame.

By using the formulas above and the tensor of inertia of the robot's components, the tensor of inertia of the whole robot obtained is:

$$I = \begin{bmatrix} 1.13 & 0.001 & -0.04 \\ 0.001 & 1.52 & -0.002 \\ -0.04 & -0.002 & 0.91 \end{bmatrix}$$

To check the correctness of the above estimation and to have even a better estimation, the ideal robot tensor of inertia is calculated by using a Solidworks model of the robot, which was already made by Professor M. Stein for a previous project (Figure 5.2). Thereby, the tensor of inertia, automatically calculated by Solidworks, which will be considered in the following is:

$$I = \begin{bmatrix} 1.21 & 0.000355 & -0.0655 \\ 0.000355 & 1.41 & -0.00125 \\ -0.0655 & -0.00125 & 0.922 \end{bmatrix} \tag{5.1}$$

The data provided by the Solidworks model confirm also the approximated location of the robot CoM. In particular, the position of the CoM, with respect to an output coordinate system placed at one corner of the rear-right square tube with its axes parallel to the robot principal axes (Figure 5.2) (defined by default), is:

$${}^o\boldsymbol{p}_{CoM} = [0.087 \quad 0.139 \quad -0.041]^T$$

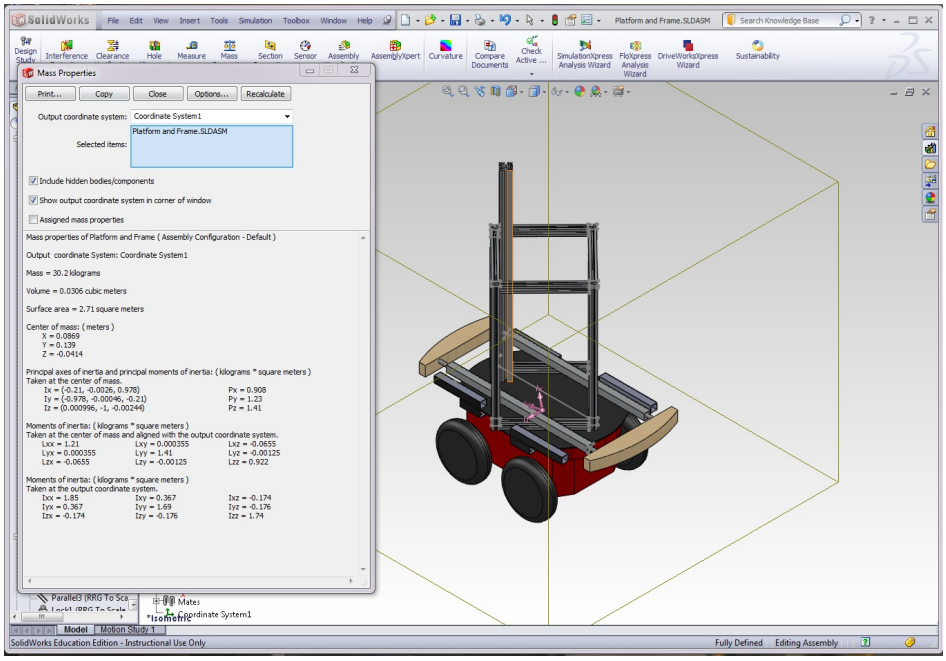


Figure 5.2: The Solidworks model of the robot.

The position of the rear-right contact point with respect to the output coordinate system is:

$$^{o}\boldsymbol{p}_4 = [-0.06 \quad -0.04 \quad -0.26]^T,$$

Thereby, the position of that contact point with respect to the robot frame is:

$$\boldsymbol{p}_4 = {}^{o}\boldsymbol{p}_4 - {}^{o}\boldsymbol{p}_{CoM} = [-0.147 \quad -0.179 \quad -0.221]^T,$$

which nearly coincides to the values in the 4^{th} row in Tab.5.1.

5.2 Identification of the Tire Dynamic Parameters

In Section 3.3 the identification of the lateral and vertical tire elasticity coefficient was presented by analyzing the different data acquired from the force sensor using the testing system machine.

In this section, the identification of the tire dynamic parameters is completed by analyzing the free response of the Roll, Pitch and Yaw motion, measured by the three 1-axis accelerometers.

Let us consider the robot performing separately pure rotations around the x,y,z-axis, respec-

tively defined as Roll, Pitch and Yaw motion. Consequently, the robot dynamics can be described by considering planar motion respectively on the $y-z$, $x-z$ and $x-y$ plane, as depicted in Figure 5.3.

Under the assumption that the wheels are not spinning, the contact points are not moving and some external torques M_z, M_y, M_x are applied to the robot CoM respectively for the three rotations, we can consider the Roll, Pitch and Yaw motion described by the following equations:

$$\begin{aligned} I_x\dot{\omega}_x &= \sum_{i=1}^{4}(y_i f_{iz} - z_i f_{iy}) + M_x \\ I_y\dot{\omega}_y &= \sum_{i=1}^{4}(-x_i f_{iz} + z_i f_{ix}) + M_y \\ I_z\dot{\omega}_z &= \sum_{i=1}^{4}(x_i f_{iy} - y_i f_{ix}) + M_z \end{aligned} \tag{5.2}$$

where x_i, y_i, z_i and f_{ix}, f_{ix}, f_{ix} are respectively the coordinates and the reaction forces of the i^{th} contact point. By substituting equations (4.13),(4.9),(4.15) in (5.2) and considering only $\omega_x, \omega_y, \omega_z \neq 0$ respectively for three rotations, we can rewrite the equations describing the Roll, Pitch and Yaw motion as follows:

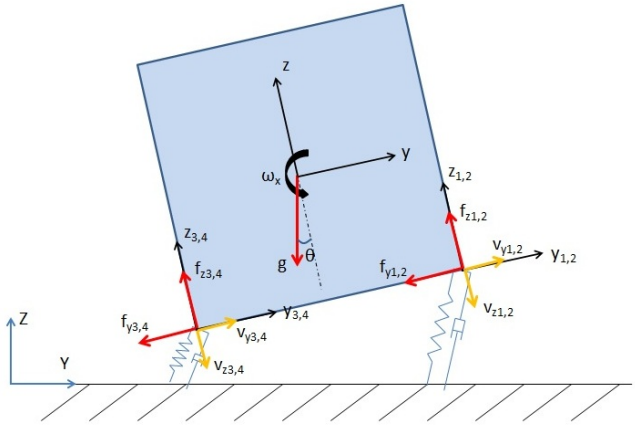
$$\begin{aligned} I_x\ddot{\omega}_x + (4h^2D_y + 4c^2D_z)\dot{\omega}_x + (4h^2K_y + 4c^2K_z)\omega_x &= M_x \\ I_y\ddot{\omega}_y + (4h^2D_x + 4a^2D_z)\dot{\omega}_y + (4h^2K_x + 4a^2K_z)\omega_y &= M_y \\ I_z\ddot{\omega}_z + (4c^2D_x + 4a^2D_y)\dot{\omega}_z + (4c^2K_x + 4a^2K_y)\omega_z &= M_z \end{aligned} \tag{5.3}$$

where, for a sake of simplicity, it was considered $b = a$.

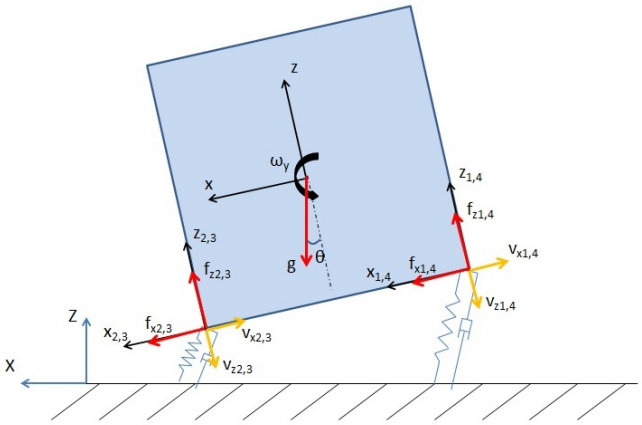
In order to identify the parameters in (5.3), we consider the free response of a second order system. We can write the free response of equation (5.3) by setting the inputs $M_x = M_y = M_z = 0$ and dividing by the moment of inertia I_x, I_y, I_z, obtaining the equation:

$$\ddot{\omega}_i + 2\xi_i\omega_{n_i}\dot{\omega}_i + \omega_{n_i}^2\omega_i = 0 \tag{5.4}$$

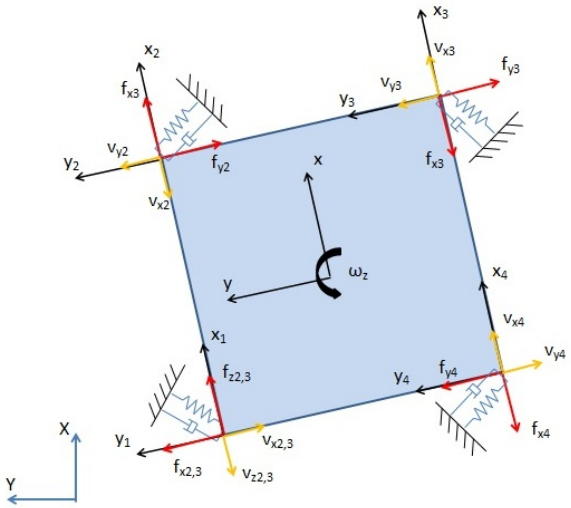
where ω_{n_i} and ξ_i, with $i = x, y, z$, correspond respectively to the natural angular frequencies and the damping ratios of the systems in (5.4), and they are related to the parameters K_i, D_i by the following formulas:



(a)



(b)



(c)

Figure 5.3: Scheme of: (a) Roll; (b) Pitch; (c) Yaw.

$$\begin{aligned}
\omega_{n_x} &= \sqrt{\frac{4h^2K_y + 4c^2K_z}{I_x}}, \qquad & \xi_x &= \frac{4h^2D_y + 4c^2D_z}{2\sqrt{I_x(4h^2K_y + 4c^2K_z)}} \\
\omega_{n_y} &= \sqrt{\frac{4h^2K_x + 4a^2K_z}{I_y}}, \qquad & \xi_y &= \frac{4h^2D_x + 4a^2D_z}{2\sqrt{I_y(4h^2K_x + 4a^2K_z)}} \\
\omega_{n_z} &= \sqrt{\frac{4c^2K_x + 4a^2K_y}{I_z}}, \qquad & \xi_z &= \frac{4c^2D_x + 4a^2D_y}{2\sqrt{I_z(4c^2K_x + 4a^2K_y)}}
\end{aligned} \tag{5.5}$$

It is worth noticing that, in this case, the three angular velocities coincide to the derivative of the Roll, Pitch and Yaw angles, therefore the equations in (5.4) can be rewritten by substituting the $\omega_x = \dot{\theta}, \omega_y = \dot{\psi}, \omega_z = \dot{\phi}$ and integrating it:

$$\begin{aligned}
\ddot{\theta} + 2\xi_x\omega_{n_x}\dot{\theta} + \omega_{n_x}^2\theta = 0 \\
\ddot{\psi} + 2\xi_y\omega_{n_y}\dot{\psi} + \omega_{n_y}^2\psi = 0 \\
\ddot{\phi} + 2\xi_z\omega_{n_z}\dot{\phi} + \omega_{n_z}^2\phi = 0
\end{aligned} \tag{5.6}$$

It is well known from automatic control theory [22] that, for under-damped systems ($\xi < 1$), the solution of (5.6) can be written as follows:

$$\begin{aligned}
\theta(t) &= A_x e^{-\xi_x\omega_{n_x}t}\sin\Big(\sqrt{1-\xi_x^2}\omega_{n_x}t + B_x\Big) \\
\psi(t) &= A_y e^{-\xi_y\omega_{n_y}t}\sin\Big(\sqrt{1-\xi_y^2}\omega_{n_y}t + B_y\Big) \\
\phi(t) &= A_z e^{-\xi_z\omega_{n_z}t}\sin\Big(\sqrt{1-\xi_z^2}\omega_{n_z}t + B_z\Big)
\end{aligned} \tag{5.7}$$

where the amplitudes A_i and the phases B_i depend on the initial conditions.

The parameters ω_{n_i}, ξ_i of equations in (5.6) are estimated by analyzing the data acquired from the accelerometers, fixing the wood-cube to the robot's platform approximatively at the position $\boldsymbol{p}_a \approx [-0.2 \quad 0 \quad 0]^T$ from the robot CoM and with its axes parallel to the robot principal axes (Figure 3.2). The initial condition on the Roll, Pitch and Yaw angle is given by tilting the robot about a small angle around the three principal axes, such that the wheels do not lift from the floor, and then releasing it. By considering the acceleration $\boldsymbol{a}$ measured from the accelerometers as defined in (3.6), it is convenient to consider a_y for the Roll and Yaw motion, and a_x for the Pitch motion. In fact, by considering in (3.6) only $\omega_z, \omega_y, \omega_x \neq 0$ respectively for each rotation, we obtain the following relations:

$$
\begin{aligned}
a_x &= g\sin(\psi) \approx g\psi \\
a_y &= -g\sin(\theta) \approx -g\theta \\
a_y &= -p_{a_x}\ddot{\phi} \\
&= -p_{a_x}A_z\omega_{n_z}^2 e^{-\xi_z\omega_{n_z}}\left[(2\xi_z^2-1)\sin\Big(\sqrt{1-\xi_z^2}\omega_{n_z}t+B_z\Big) - 2\xi_z\sqrt{1-\xi_z^2}\cos\Big(\sqrt{1-\xi_z^2}\omega_{n_z}t+B_z\Big)\right] \\
&= -p_{a_x}A_z\omega_{n_z}^2 e^{-\xi_z\omega_{n_z}}\sin\Big(\sqrt{1-\xi_z^2}\omega_{n_z}t+B_z+\Phi\Big)
\end{aligned}
\tag{5.8}
$$

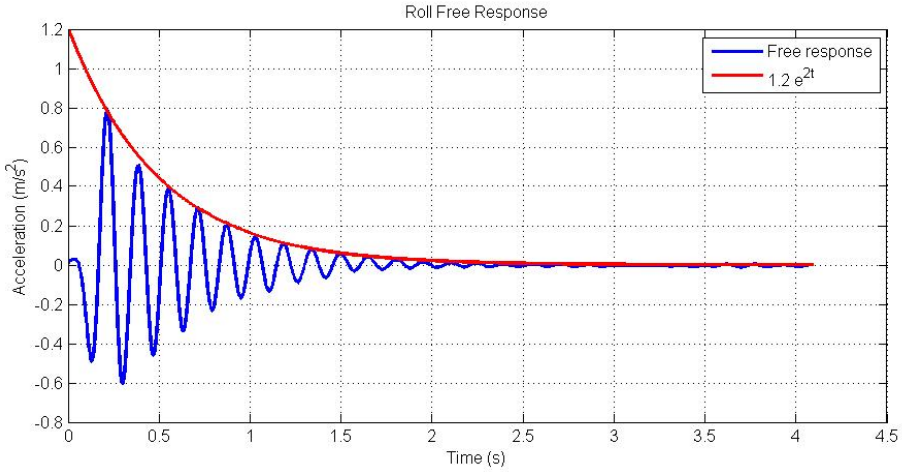
where g is the gravitational acceleration and the first equality come from the linear property of trigonometric functions such that: $a\sin x + b\cos x = \sqrt{a^2+b^2}\sin(x+\Phi)$, where $\tan\Phi = \frac{b}{a}$. The approximations come from the first order Taylor expansion of the sin() function, which makes sense for ¨small¨ values of θ, ψ.

By looking at the relations in (5.8), we can claim, with a good approximation, that the signals measured from the accelerometers differ from the free responses in (5.7) only by their amplitudes and eventually phases. Thereby. the Roll, Pitch and Yaw angular natural frequency ω_{i_n} and damping ratio ξ_i can be estimated by looking at the frequency of the oscillations measured from the accelerometers and the rate at which they decay. To this purpose, the data from the accelerometers are acquired eight times per each motion (Roll, Pitch and Yaw) and analyzed by using the Matlab functions defined in Appendix A. In particular, the data contain 2^{12} samples and are acquired with a sample frequency of 1 kHz, but only the first 2^{11} samples are analyzed (`tRange` = *'beginnig'*), in order to cut off possible disturbs present in the last part of the data where the system has already reached the equilibrium. Moreover, the data are lowpass-filtered at 20 Hz (`fRange` = $[1, 20]$) by using the function `filterData()` and the peak frequency is found by using the function `findPeakFreq()`.

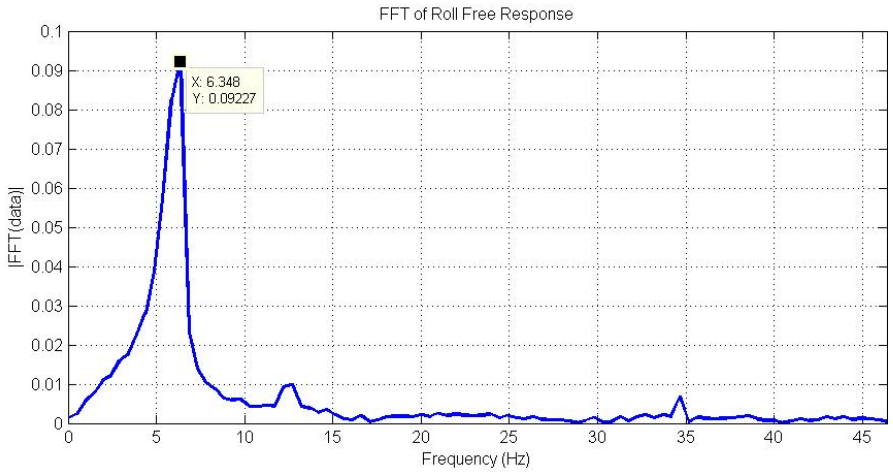
Figure 5.4, 5.5, 5.6 depict three examples of the acquired data and their FFT respectively for the Roll, Pitch and Yaw motion.

Tab.5.2 shows all the peak frequencies found for each recorded data. As the standard deviation is relatively small in all the three cases ($\frac{std(f_{peak})}{mean(f_{peak})} << 1$), the principal frequency component can be estimated by taking the mean of the peak frequencies.

Furthermore, the rate at which the oscillations decay can be estimated by looking at the exponential function that best envelops the data. After overlapping the lowpass-filtered data with the function $Ae^{-\lambda t}$ for different values of λ, we found that, with good approximation,

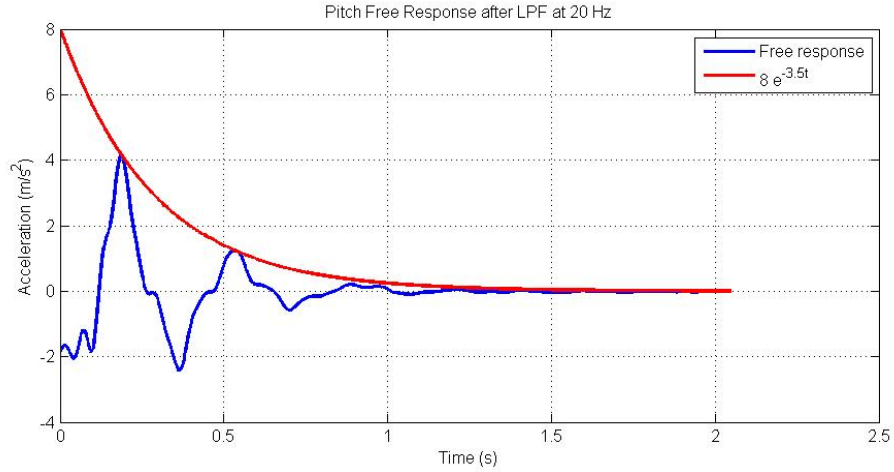


(a)

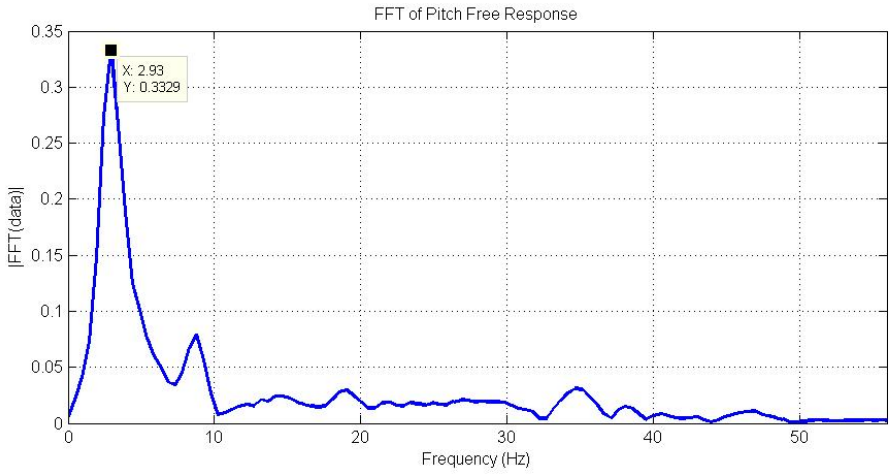


(b)

Figure 5.4: Example of acquired data for Roll motion: (a) Lowpass-filtered data and their envelop; (b) FFT of real data.

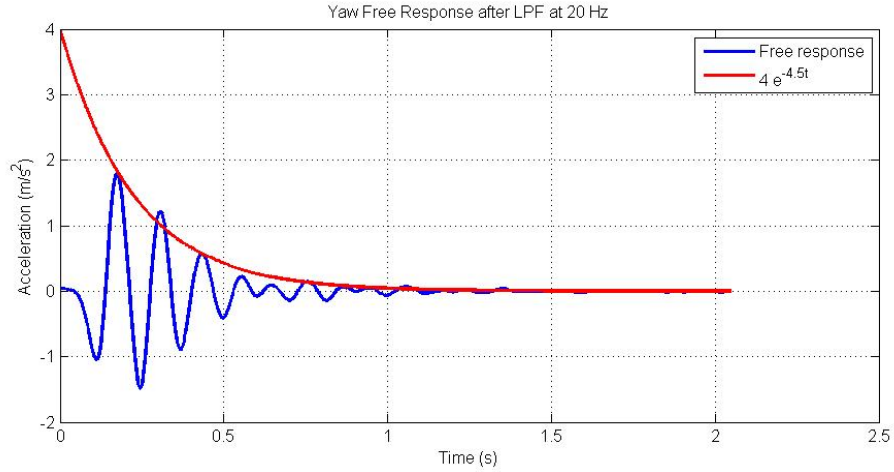


(a)

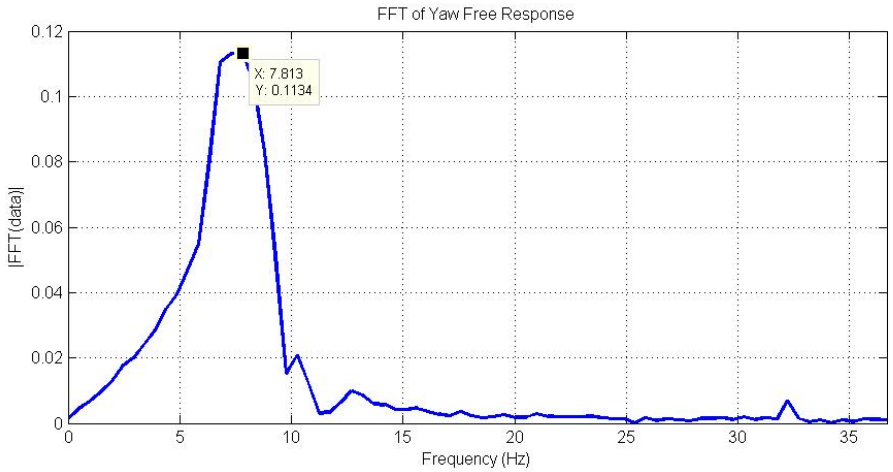


(b)

Figure 5.5: Example of acquired data for Pitch motion: (a) Lowpass-filtered data and their envelop; (b) FFT of real data.



(a)



(b)

Figure 5.6: Example of acquired data for Yaw motion: (a) Lowpass-filtered data and their envelop; (b) FFT of real data.

$\boldsymbol{data}$	f_{peak_x}	f_{peak_y}	f_{peak_z}
1	5.8594	2.9297	7.8125
2	5.8594	2.9297	7.3242
3	5.8594	2.9297	7.3242
4	6.3477	2.9297	6.8359
5	5.8594	2.9297	7.8125
6	6.3477	3.4180	7.8125
7	5.8594	2.9297	7.3242
8	6.3477	3.4180	7.8125
$\boldsymbol{mean}$	6.0425	3.0518	7.5073
$\boldsymbol{std}$	0.2527	0.2260	0.3633

Table 5.2: Peak frequencies of the Roll, Pitch and Yaw free response for random initial conditions.

the exponential function always envelops the data when λ assumes values relatively close to $\lambda = 2, 3.5, 4.5$ respectively for the Roll, Pitch and Yaw. In conclusion, we can consider the frequencies of the oscillations and the rates of decay estimated as:

$$\begin{aligned} f_{peak_x} &= 6.04, \qquad \lambda_x = 2 \\ f_{peak_y} &= 3.05, \qquad \lambda_y = 3.5 \\ f_{peak_z} &= 7.51, \qquad \lambda_z = 4.5 \end{aligned}$$

By looking at the free response functions defined in (5.7), the following relations between f_{peak_i}, λ_i and the corresponding ω_{n_i}, ξ_i can be easily derived:

$$\begin{aligned} 2\pi f_{peak_i} &= \sqrt{1-\xi_i^2}\,\omega_{n_i} \\ \lambda_i &= \xi_i \omega_{n_i} \end{aligned}$$

Thereby, the angular natural frequencies and the damping ratios are estimated as follows:

$$\begin{aligned} \omega_{i_n} &= \sqrt{4\pi^2 f_{peak_i}^2 + \lambda_i^2} \\ \xi_i &= \frac{\lambda_i}{\sqrt{4\pi^2 f_{peak_i}^2 + \lambda_i^2}} \end{aligned} \tag{5.9}$$

leading to the values:

$$\omega_{n_x} = 38.0031 \frac{rad}{s} \approx 38 \frac{rad}{s}, \qquad \xi_x = 0.0526 \approx 0.05$$
$$\omega_{n_y} = 19.4807 \frac{rad}{s} \approx 19.5 \frac{rad}{s}, \qquad \xi_y = 0.1797 \approx 0.18$$
$$\omega_{n_z} = 47.4008 \frac{rad}{s} \approx 47.4 \frac{rad}{s}, \qquad \xi_z = 0.0949 \approx 0.09$$

Finally, by substituting the results obtained above in (5.5) and considering the tire elasticity coefficients K_x, K_y, K_z identified in Section 3.3, the moments of inertia around the principal axes and the tire damping coefficients can be calculated as follows:

$$\begin{aligned}
I_x &= \frac{4h^2K_y + 4c^2K_z}{\omega_{n_x}^2} = 9.82, \qquad & D_x &= \frac{I_z\omega_{n_z}\xi_z - I_y\omega_{n_y}\xi_y}{2(c^2 - h^2)} = 1596 \\
I_y &= \frac{4h^2K_x + 4a^2K_z}{\omega_{n_y}^2} = 19.83, \qquad & D_y &= \frac{2\xi_z\sqrt{I_z(4c^2K_x + 4a^2K_y)} - 4c^2D_x}{4a^2} = 2284 \\
I_z &= \frac{4c^2K_x + 4a^2K_y}{\omega_{n_z}^2} = 0.97, \qquad & D_z &= \frac{2\xi_y\sqrt{I_y(4h^2K_x + 4a^2K_z)} - 4h^2D_x}{4a^2} = 4693
\end{aligned} \tag{5.10}$$

By comparing the values of the identified moment of inertia in (5.10) with the inertia tensor in (5.1), we notice that only I_z is close to value estimated using the Solidworks model, while the values of I_x, I_y are nearly one order greater than the estimated ones. This is probably due to the approximation we have done by considering pure Roll and Pitch motion, i.e. $v_x, v_y, v_z = 0$, as the robot was not constraint to move only around the two corresponding axes. In particular, while the initial condition for the Yaw motion is provided by slightly turning the robot around the z-axis, providing nearly the same initial tire stretching condition along the x, y-axis, the initial condition for the Roll and Pitch motion is provided by tilting the robot on only one side, providing different initial tire stretching conditions along the y, z-axis and x, z-axis respectively. Moreover, the difference between K_y, K_X and K_z makes the tire stretch more easily along the x, y-axis rather than the z-axis, therefore the robot is induced to move horizontally rather than vertically.

The equations describing the Roll and Pitch motion can be found by either considering respectively $v_y, v_z, \omega_x \neq 0$ and $v_x, v_z, \omega_y \neq 0$ and including the equations describing $\dot{v}_x, \dot{v}_y, \dot{v}_z$ in (5.2), or by simply substituting respectively $v_x, \omega_y, \omega_z = 0$ and $v_y, \omega_x, \omega_z = 0$ in (2.65). By considering the ¨small¨angles and therefore approximating $\sin\theta \approx \theta, \sin\psi \approx \psi, \cos\theta, \cos\psi = 0$, we obtain in both cases a fourth order linear system which can be represented by the following equations in the Laplace domain (impulse responses):

$$\begin{aligned}
&\Big[I_x m s^4 + 4(D_z c^2 + D_y h^2 + D_y I_x)s^3 + 4(K_z c^2 + K_y h^2 + K_y I_x + 4D_y D_z c^2)s^2 + \\
&\qquad +4(4D_y K_z c^2 + 4D_z K_y c^2 - D_y ghm)s + 4K_y(4K_z c^2 - ghm)\Big]\omega_x(s) = H_x \\
&\Big[I_y m s^4 + 4(D_z a^2 + D_x h^2 + D_x I_y)s^3 + 4(K_z a^2 + K_x h^2 + K_x I_y + 4D_x D_z a^2)s^2 + \\
&\qquad +4(4D_x K_z a^2 + 4D_z K_x a^2 - D_x ghm)s + 4K_x(4K_z a^2 - ghm)\Big]\omega_y(s) = H_y
\end{aligned} \tag{5.11}$$

where H_x, H_y are the energy of the impulse input respectively for the Roll and Pitch motion.

A fourth order equation can be always rewritten as product of two second order equation, therefore we can rewrite each equation in (5.11) as:

$$(s^2 + 2\xi_i'\omega_{n_i}' s + \omega_{n_i}'^2)(s^2 + 2\xi_i''\omega_{n_i}'' s + \omega_{n_i}''^2)\omega_i(s) = H_i \tag{5.12}$$

It can be proved that, if $\omega_{n_i}'' \gg \omega_{n_i}'$ and $\xi_i', \xi_i'' \gg \frac{\omega_{n_i}'}{\omega_{n_i}''}$, such a system can be approximated by a second order system represented by the second order equation containing ω_{n_i}', ξ_i'. If such a condition was verified in our case, the previously identified values for ω_{n_i}, ξ_i would coincide with ω_{n_i}', ξ_i', therefore, by equating the coefficients of (5.11) and (5.12) and solving the resulting equations, we can find the damping coefficients D_x, D_y, D_z. As we generally do not know whether the conditions $\omega_{n_i}'' \gg \omega_{n_i}'$ and $\xi_i', \xi_i'' \gg \frac{\omega_{n_i}'}{\omega_{n_i}''}$ are satisfied or not, a reasonable idea could be solving the equations given by equating the coefficients and, if there are admissible real solutions, looking at the values found for ω_{n_i}'', ξ_i''. Then, we check whether such conditions are satisfied or not in order to define the reliability of the result. In particular, we solved those equations in `Matlab` by using the symbolic toolbox and we found values for ω_{n_i}'' relatively close to the identified values for ω_{n_i}, therefore we can conclude that the damping coefficients can not be analytically identified by using the acquired data. However, an approximation of the damping coefficients can be found by looking at the free response obtained in the simulation environment after trying different set of values, as it will be provided in the next Chapter.

Chapter 6

Simulation

In this chapter, the `Simulink` implementing the simulation of the three-dimensional SSMR motion is first presented. The model includes the generalized dynamic model provided in Section 2.2 and the model of the tire reaction forces provided in Section 4. A complete description of the main blocks of the model is provided. Then, the results of the simulation, performed by using the identified robot parameters presented in the previous sections, are presented and further discussed to characterize the properties of the model. Finally, the obtained results are compared with the real data acquired with the accelerometers in order to check the adherence of the model to the real system. In particular, the following objectives will be considered as benchmark:

- To validate the tire reaction force model by qualitatively reproducing the real static and kinetic lateral force measured by the force sensor when pulling the robot platform sidewards (Section 3.3).
- To validate qualitatively the overall three-dimensional SSMR model by performing the two basic motion commands from an odometry point of view, i.e. moving along a straight line and swiveling in place.
- To reproduce qualitatively and quantitatively the robot jerky motion when swiveling in place, from a point of view of the amplitude and the frequency of the vibrations measured by the accelerometers (Section 3.2).

6.1 Simulink Model

The overall block scheme developed in `Simulink` for the simulation of three-dimensional SSMR motion is depicted in Figure 6.1. The block scheme can be divided in three main parts: the left hand side part represents the high level velocity controller, the central part represents the tire reaction force dynamics, and the right hand side part represents the three-dimensional rigid-body motion dynamics. In our case, as the robot is not equipped with any sensor measuring its velocity or the position, the controller provides the left and right-side angular velocities as feed-forward control input to each wheel, given the desired robot linear and angular velocity as reference signals, by inverting the kinematic relation in (2.11) and without considering the wheel slip. The reason why we do not consider the wheel slip is because we can not have any information on the wheel center velocities, due to the absence of velocity sensors.

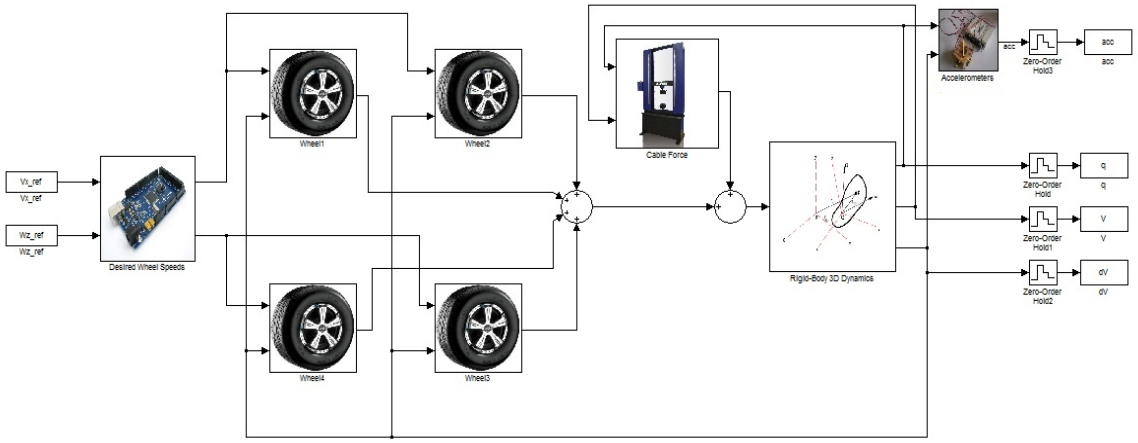


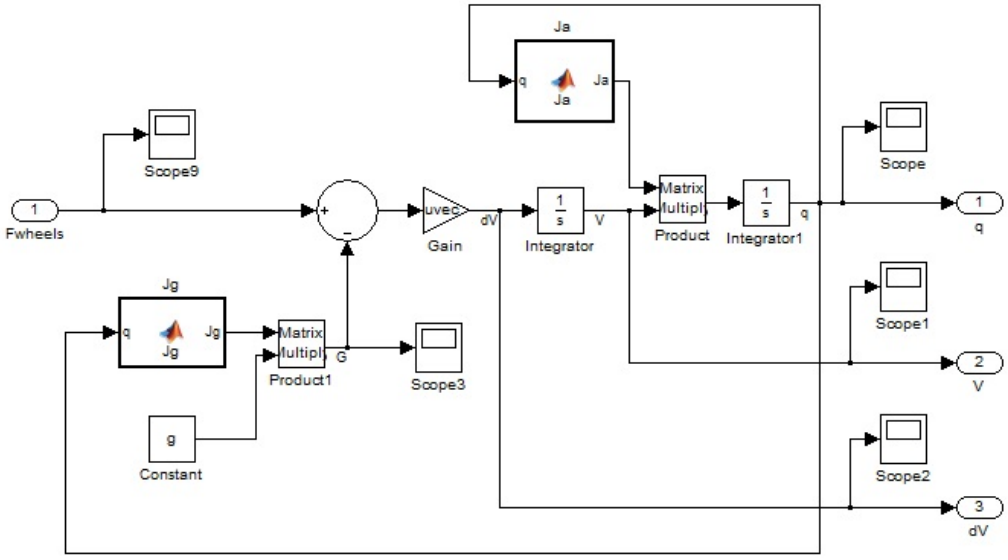
Figure 6.1: Block scheme of the Simulink model.

The block for the three-dimensional rigid-body motion dynamics simply implements the generalized dynamic model in (2.65) and provides the robot position $\boldsymbol{q}$, velocity $\boldsymbol{V}$ and acceleration $\dot{\boldsymbol{V}}$ given the resultant force and torque vector $\boldsymbol{F}$, as depicted in Figure 6.2(a). As the simulation is mainly performed for the robot swiveling in place, i.e. $v_x = 0, \omega_z \neq 0$, the product between the linear and angular velocities of the robot is relatively low with respect to its linear and angular accelerations, so that the Coriolis term $\boldsymbol{M}\hat{\boldsymbol{V}}\boldsymbol{V}$ is negligible. Thereby, in order to simplify the model and reduce the computational time, the Coriolis term was not included in the Simulink block.

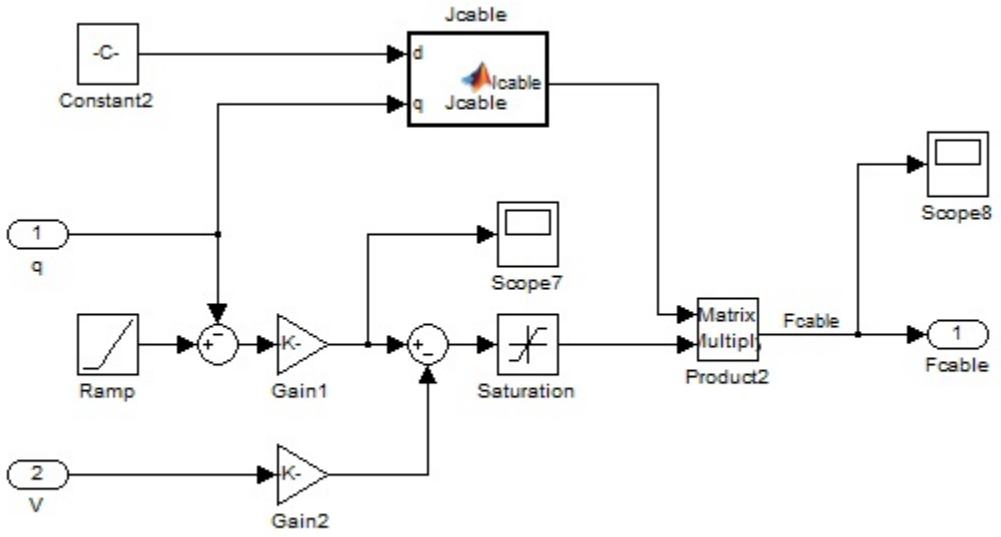
On the right hand side, there appear also two blocks, namely ¨Cable Force¨ and ¨Accelerometers¨ which provide respectively the external force by the testing machine when pulling the robot sidewards or backwards and the data from the three 1-axis accelerometers. The ¨Accelerometers¨ block simply implements (3.4), without considering any noise and band-pass filtering due to the accelerometers properties. The ¨Cable Force¨ block implements the non-linear spring-damped force produced by the steel cable when pulling it with the testing machine (see Figure 6.2(b)). The non-linearity is due to the double characteristic of a cable to provide a spring-damped force when it is pulled and a zero force when it is pushed, and it is implemented by using a saturation block. The displacement providing the cable spring force is given by the difference between the testing machine position, represented by the ramp block, and the robot position. The spring term of the force is, therefore, obtained by multiplying the cable displacement by the cable elasticity coefficient K_{cable}, and the damping term is obtained by multiplying the robot velocity by the cable damping coefficient D_{cable}. The overall force is then multiplied by J_{cable}, which transforms the cable force vector from the fixed frame to the body frame and is defined by means of the Plücker transformation of spatial force vectors given in (2.52), where the matrix $\hat{\boldsymbol{X}}$ is substituted by $\hat{p}_{cable}$, being $\boldsymbol{p}_{cable}$ the position of the application point of the cable force with respect to the robot CoM. As the steel cable is much more stiff then the tire and does not perceive relatively high frequency vibrations (see figures in Section 3.3), we set $K_{cable} \gg K_{tire}$ and $D_{cable} = 4\sqrt{mK_{cable}}$.

The tire reaction forces and torques expressed in the body frame are provided by the four ¨Wheel¨ blocks, giving the desired wheel angular velocity and the robot acceleration $\dot{\boldsymbol{V}}$ as inputs. As the four ¨Wheel¨ blocks are identical with each other, except for the skew-symmetric matrix of the contact point position $\hat{p}_i$ employed in the computation of the wheel center velocities and the toques exerted at the CoM. Thereby, the block is first masked, setting as dialog parameter the value for $\hat{p}_i$, and then put in a Simulink library model, so that every change made on the block is automatically updated for every wheel. The block scheme inside the ¨Wheel¨ block is depicted in Figure 6.3.

The block in yellow represents the wheel motor and the motor velocity servo-controller. We assume that the motor can always provide the angular velocity required by the motor velocity servo-controller and that it takes a certain time $t_{eq} = \frac{3}{K_w}$ to reach the desired velocity. In such conditions, the motor dynamics can be described by a first order system defined as follows:



(a)



(b)

Figure 6.2: (a) Block scheme inside the ¨3D Rigid-Body Dynamics¨ block; (b) Block scheme inside the ¨Cable Force¨ block.

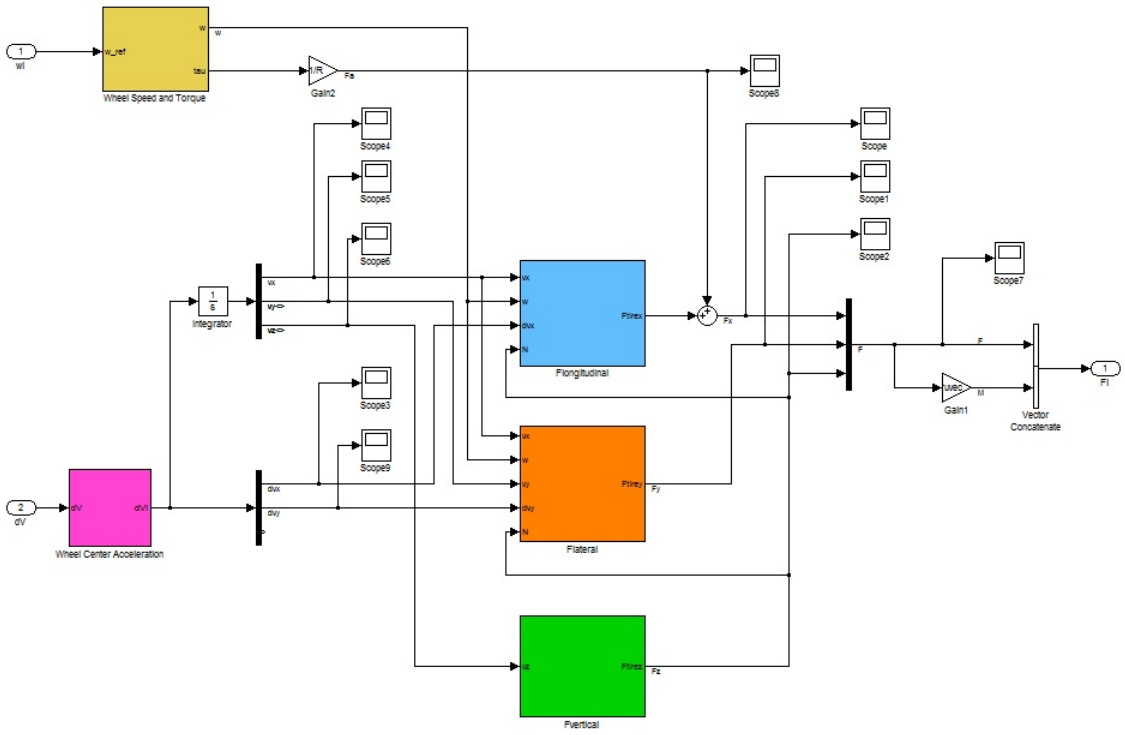


Figure 6.3: Block scheme inside the ¨Wheel¨ block.

$$\dot{\omega}_i = K_w(\omega_{i_{ref}} - \omega_i) \tag{6.1}$$

It is also well known that the motor dynamics can be described by the equation:

$$\tau_i = I_m \dot{\omega}_i \tag{6.2}$$

where I_m represents the motor inertia.
By combining equation (6.1) and (6.2), we obtain the following relation defining the torque control input:

$$\tau_i = I_m K_w(\omega_{i_{ref}} - \omega_i) \tag{6.3}$$

If we do not take into account any gear-transmission and electrical issue typical of DC motors, we have $K_w = \frac{1}{I_m}$, therefore the torque control torque is simply provided by the difference of the desired and measured wheel angular velocity. The block in yellow implements the motor model defined in (6.1) and provides both the wheel angular velocity and torque as output. It is worth noticing that such a model holds only if the motor can always provide the

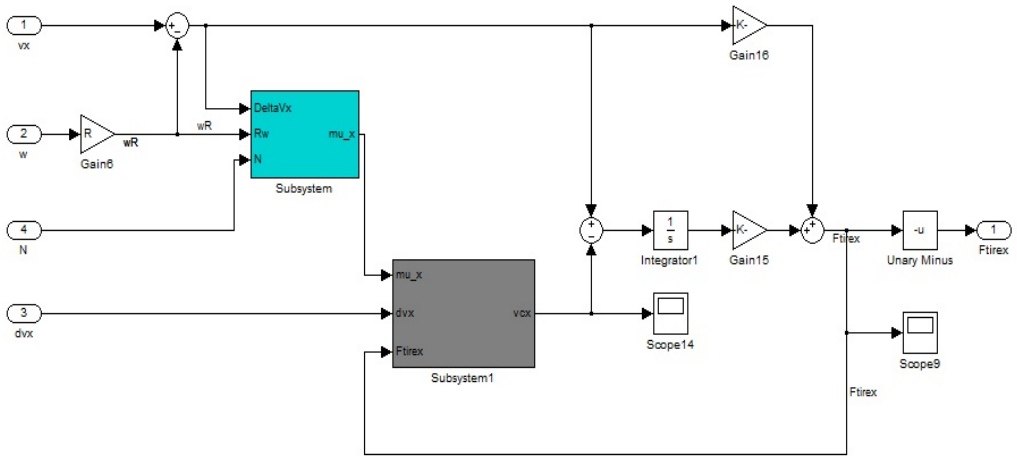
desired torque. Such an assumption can be considered verified, except for the fact that, when the motor can not provide the required torque, the robot safe mode makes the motor stall. Moreover, the model does not take into account any elasticity effect due to the motor belt. However, the model is still meaningful as the elasticity-damping effects of the motor belt are already take into account in the model for the tire longitudinal reaction force, as explained in Section 4.2.

The block in magenta computes the wheel center accelerations from the robot acceleration by employing (2.55), considering $\dot{\boldsymbol{V}}, \dot{\boldsymbol{V}}_i$ instead of $\boldsymbol{V}, \boldsymbol{V}_i$, then $\dot{\boldsymbol{V}}_i$ is integrated to provide $\boldsymbol{V}_i$.

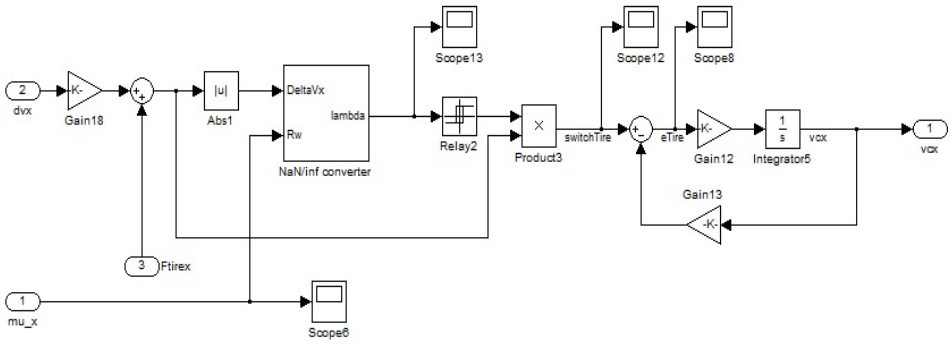
The blocks in light blue, red and green represent respectively the longitudinal, lateral and vertical reaction force block and implement (4.13), (4.9) and (4.15). In particular, while the block in green directly implements (4.15), the blocks for the longitudinal and lateral reaction forces present some differences with respect to (4.13), (4.9) due to simulation issues which are further discussed in the following description of the ¨Flongitudinal¨ block. However, we precise that, as such a description goes into details of the implementation of (4.13),(4.9), the reader, if not interested in, may go directly to Section 6.2 without any loss of comprehension. Figure 6.4(a) depicts the block scheme for the longitudinal reaction force. The same block scheme is provided for the lateral reaction force, by replacing the μ_{ix} with μ_{iy} and Δv_{ix} with v_{iy} as driving velocity.

The block in cyan implements the piece-wise linear function in (4.12a), representing the longitudinal dynamic friction coefficient when the wheels are slipping, and it multiplies it by the normal reaction force N_i. In such a way, the output of this block differs from the static and kinetic friction force only for the constant terms $\mu_{s_{max}}, \mu_{k_{max}}$, which is useful for the implementation of the hysteresis block determining the switch of the contact point dynamics, as it is explained in the following.

The block in gray provides the contact point velocity by implementing (4.13), as depicted in Figure 6.4. However, (4.13) presents an important issue from an implementation point of view, that is the complex behavior of the normal reaction force N_{ic} perceived by the contact point when moving. As already introduced in Section 4.1, the normal reaction force N_{ic} is not given only by the tire mass but it increases as the contact point gets closer to its equilibrium position, thereby perceiving also the robot weight. This increase of the normal reaction force perceived from the contact point, i.e. the increasing of the kinetic friction force, makes the contact point stops instead of following the wheel center velocity, under certain conditions



(a)



(b)

Figure 6.4: (a) Block scheme for the longitudinal reaction force; (b) Block scheme for the contact point dynamics.

as explained in Section 4.1. Such a dynamics of the contact point when the tire is releasing is so complex that it would require an ¨ad hoc¨ study, which is not provided in this thesis. In order to overcome this issue, we consider $N_{ic} = m_t g$ and we assume that the conditions on F_{icx} given in (4.13) still hold. However, when the value of F_{icx} switches from $F_{icx} \neq 0$ to $F_{icx} = 0$, the value for v_{icx} could not be zero as it should be. In such a case, as the contact point dynamics switches from $m_t \dot{v}_{icx} \neq 0$ to $m_t \dot{v}_{icx} = 0$, the velocity v_{icx} will stay stable to the value it had before switching, so that also the tire reaction force will keep its stable value instead of performing the expected stretching/releasing behavior. The solution to this issue can be found by simply rewriting (4.13) as follows:

$$
\begin{aligned}
& f_{ix} = -D_x(\Delta v_{ix} - v_{icx}) - K_x \int (\Delta v_{ix} - v_{icx}) dt \\
& m_t \dot{v}_{icx} + D_x v_{icx} + \mu_{kx} N_{ci} sign(v_{icx}) = F'_{icx} \\
& F'_{icx} = \begin{cases} 0, & \text{for } |F'_{ix}| < \mu_{sx} N_i \wedge f_i(t - \Delta t) = 0 \\ F'_{ix}, & \text{for } |F'_{ix}| > \mu_{kx} N_i \wedge f_i(t - \Delta t) \neq 0 \end{cases} \\
& F'_{ix} = m_t \dot{v}_{ix} + D_x v_{ix} + K_x \int (v_{ix} - v_{icx}) dt
\end{aligned}
\tag{6.4}
$$

where $f_i = 1, 0$ is a flag indicating whether the contact point is moving or not.

We notice that, in this case, when the value of F_{icx} switches from $F'_{icx} \neq 0$ to $F'_{icx} = 0$, the second equation in (6.4) represents an asymptotically stable system instead of a simply stable system, therefore v_{icx} will tend to zero independently from the value it had before switching. Briefly, as we do not know the dynamics of N_{ci} allowing the contact point to stop, we are imposing $v_{icx} = 0$ when $F'_{icx} = 0$. However, it must be noticed that the overall dynamics does not change from the theoretical one, except for the fact that the contact point does not stop when $F'_{icx} = 0$ as for the theoretical case, but it stops some time later. Moreover, as we it will be shown in the next Section, the system represented by the second equation in (6.4) is over-damped, i.e. $D_x > 2\sqrt{m_t K_x}$, and the corresponding time constant is relatively small, i.e. $\sqrt{\frac{m_t}{K_x}} \ll 0.1$. As a consequence, the contact point stops immediately after $F'_{icx} = 0$. It is worth noticing also that such approximation of the contact point dynamics does not take into account the case in which v_{ix} is relatively high, or N_{ci} relatively low, so that v_{icx} could never reach zero, as explained in Section 4.1. However, we are considering only the case $\omega_z < 64 \frac{deg}{s} \approx 1 \frac{rad}{s}$, that is $v_{ix} < 0.15 \frac{m}{s}$, and the acquired data from the force sensor always presents the saw-tooth shape when pulling the robot up to $0.1 \frac{m}{s}$, therefore we can assume

that v_{icx} can always reach zero when the tire is released. Furthermore, we can claim that, for values of v_{ix} such that the contact point would not stop, the frequency of the tire stretching and releasing f_t provided by (6.4) is so high that v_{icx} has not enough time to reach zero, i.e. $\sqrt{\frac{K_x}{m_t}} < f_t$. Thereby, even in such a condition, the approximation of contact point dynamics in (6.4) is meaningful.

The hysteresis of the system is provided by the Matlab ¨Relay¨ block, which sets its output, i.e. the flag f_i, either to 0 or 1 accordingly to the value of its input and output. As in the ¨Relay¨ block the condition on its input must be a constant value, we rewrite the conditions on $|F'_{icx}|$ in (6.4) as follows:

$$\frac{|F'_{icx}|}{\mu_{ix}N_i} < \mu_{s_{max}}$$
$$\frac{|F'_{icx}|}{\mu_{ix}N_i} > \mu_{k_{max}}$$

which are meaningful since we always have $\mu_{ix}N_i > 0$. Thereby, the switch of the F'_{icx} as input for the dynamics of v_{icx} is performed by having the ratio $\frac{|F'_{icx}|}{\mu_{ix}N_i}$ as input of the ¨Relay¨ block and multiplying its output (f_i) by F'_{icx}.

To avoid the simulation abort due to NaN or inf result of the division, the ¨NaN/inf converter¨ block is implemented. This block provides as output the value corresponding to the ratio if the absolute value of the result is less than inf[1], otherwise it provides either zero or 'realmax', i.e. the greatest representable value for 'double' variables, depending on whether the numerator is equal or different to zero. The ¨NaN/inf converter¨ block is also used for computing the value of λ_i and θ_i in the blocks for μ_{ix}, μ_{iy}.

Finally, because of zero-crossing issues during the simulation due to the $sign()$ function, even if approximated by a high gain ($> 10^6$) with the saturation to $-1, 1$, we neglected the term $\mu_{kx}N_{ci}sign(v_{icx})$. We notice that such approximation is meaningful since we have $\mu_{kx}N_{ci}sign(v_{icx}) \leq N_{ci} \ll D_x v_{icx}$ for $v_{icx} \gg 10^{-4}$, while for $v_{icx} < 10^{-4}$ the difference $v_{ix} - v_{icx}$, determining the tire stretching, does not significantly change independently from the exact value of v_{icx}.

The same considerations done for the longitudinal reaction force hold for the lateral reaction force, therefore the block for the lateral reaction force is obtained simply by replacing the block for μ_{ix} with the one for μ_{iy}, and Δv_{ix} with v_{iy} as driving velocity.

[1] Note that the condition $NaN < inf$ is always true.

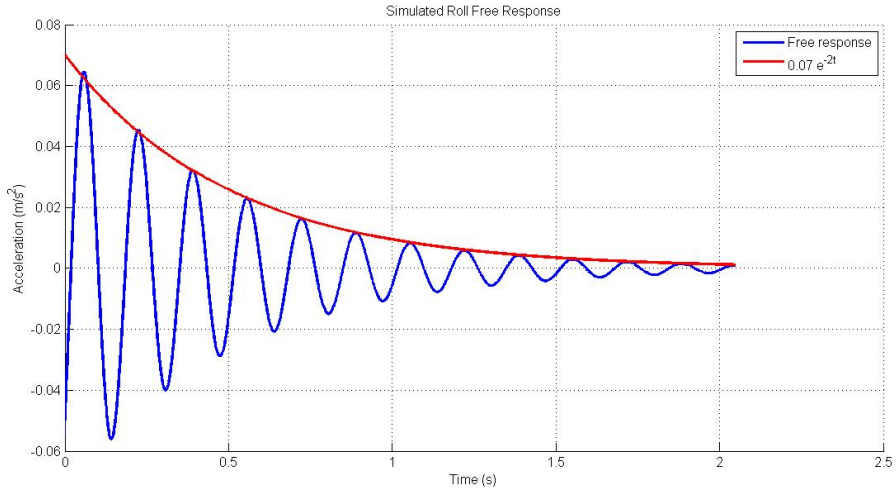
6.2 Simulation Results

In this section the results of the simulation are presented and further discussed to characterize the properties of the model, and they are compared with the real data acquired with the accelerometers in order to check the model reliability. Before running the simulation, we need to estimate the values for the damping coefficients D_x, D_y, D_z. As explained in Section 5.2, it was not possible to analytically compute the values for D_x, D_y, D_z from the identified values of the angular natural frequency and the damping ratio for the Roll, Pitch and Yaw motion. Conversely, an estimation of the tire damping coefficients can be obtained by setting D_x, D_y, D_z in the simulation to some random values and then changing those values so that the Roll and Pitch free responses match with the data presented in Section 5.2. We also recall the fact that, as the third equation in (5.3) holds for the acquired Yaw free response, the parameters D_x and D_y are dependent from each other by the second relation on the right side in (5.10). Thereby, we only need to set the values for D_z and either D_x or D_y.
The initial condition for the free response is provided by setting to different values the initial condition of the integrator for the vertical reaction force. In particular, for the Roll motion we set the same initial condition for the two left and right-side wheels, while for the Pitch angle we set the same condition for the two front and rear-side wheels. In order to provide a free response as close as possible to the acquired one, the initial conditions are set symmetrically with respect to the tire compression at equilibrium, that is for the Roll case $z_{init1} = z_{init2} = -\frac{mg}{4Ktire_z} - 0.001$ and $z_{init3} = z_{init4} = -\frac{mg}{4Ktire_z} + 0.001$. As the response is under-damped in the three cases (see Figure 5.4, 5.5), the damping coefficients are initially set to one fourth of the value corresponding to a critically damped response, i.e. $D_x = \frac{1}{2}\sqrt{mK_x}$, $D_y = \frac{1}{2}\sqrt{mK_y}$, $D_z = \frac{1}{2}\sqrt{mK_z}$, and then they are either increased or decreased depending on the simulation results. After several trials, by comparing the results from the ¨Accelerometers¨ block with Figure 5.4, 5.5, we obtained the following values:

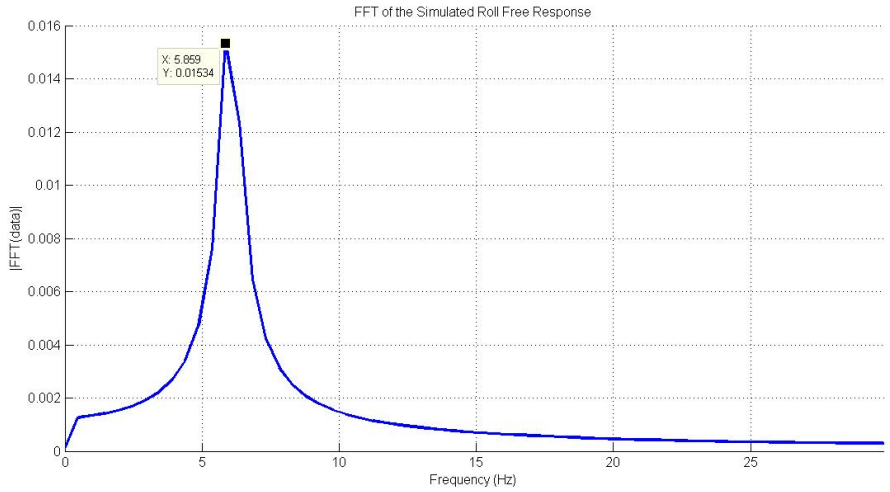
$$D_x = 50, \qquad D_y = 30, \qquad D_z = 70 \tag{6.5}$$

Figure 6.5 and 6.6 depict the simulation results respectively for the Roll and Pitch motion, obtained from the ¨Accelerometers¨ block by setting the values in 6.5.

It is worth noticing that, although this estimation is not accurate and could not be unique, it provides the needed information to reproduce the real Roll and Pitch oscillation, which is

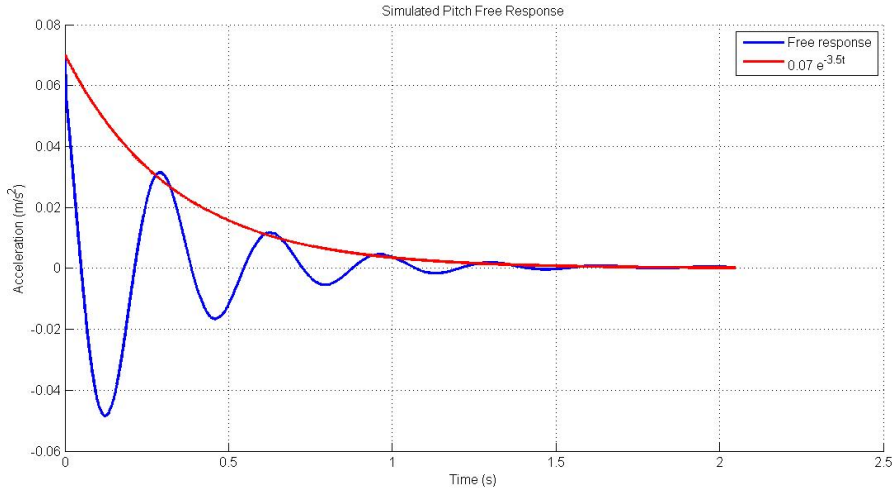


(a)

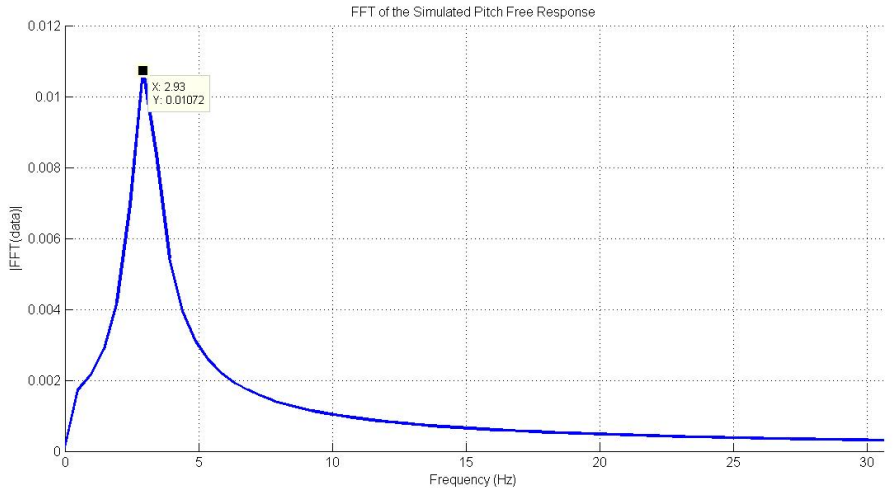


(b)

Figure 6.5: (a) Simulation result for Roll motion and its envelop; (b) FFT of the simulation result for Roll motion.



(a)



(b)

Figure 6.6: (a) Simulation result for Pitch motion and its envelop; (b) FFT of the simulation result for Pitch motion.

enough to qualitatively reproducing the real robot jerky motion. In fact, if we slightly change those values, the overall system behavior nearly does not change.

After the tire damping coefficients are estimated, we still need to identify/estimate two parameters to complete our model: the effective mass of the tire m_t, determining the contact point dynamics in (6.4), and the maximum wheel slip λ_{max}, determining the characteristic of the dynamic friction coefficient μ_x in (4.12a). Unfortunately, because of lack of adequate laboratory instrumentation and the presence of time constraints, we could neither identify nor estimate such parameters. However, we can provide an idea about their order of magnitude by qualitatively and quantitatively analyzing the simulation results obtained by setting them to some values from different ranges. Moreover, we notice that, as the mass of one wheel is $0.07\,kg$, it is reasonable to think that the effective tire mass should assume values of the order of $10^{-2}\,kg$. After changing different values in the range $[0.001, 0.07]$, we found out that the value of the tire mass nearly does not affect the system behavior. Thereby, we decided to set the tire mass as half of the wheel mass, i.e $m_t = 0.035\,kg$.

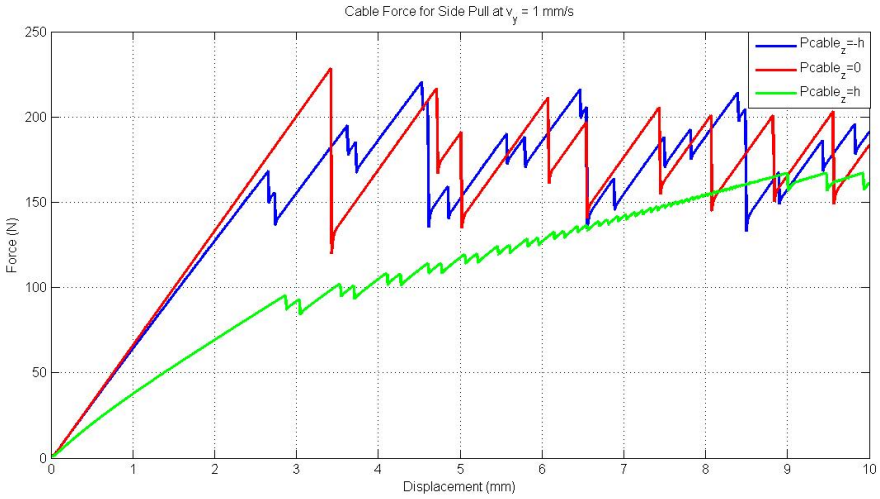
The Simulink model is first tested by running the simulation with the wheels stopped, i.e. $v_x, \omega_z = 0$, and the testing machine activated so that it provides a cable force pulling the robot along the y-axis, i.e. $K_{cable} = 10^6[0 \quad 1 \quad 0 \quad 0 \quad 0 \quad 0]^T$. In order to compare the results obtained in the simulation with the ones provided in Section 3.3, the simulations are done by moving the testing machine at $v_y = 0.001\,\frac{m}{s}$ and by setting the applications point $\boldsymbol{p}_{cable}$ at different heights.

We notice that, as the wheels are not spinning and no external force is provided along the x-axis, the value of λ_{max} is not affecting the system behavior. In fact, independently from λ_{max}, we have $\lambda_i = 0 < \lambda_{max}$, therefore $\mu_{iy} = \frac{v_{iy}}{r\omega_i} = 1$ because of the saturation block due to the constraint provided in Section 4.1. Thereby, we can set, for instance, $\lambda_{max} = 1$.

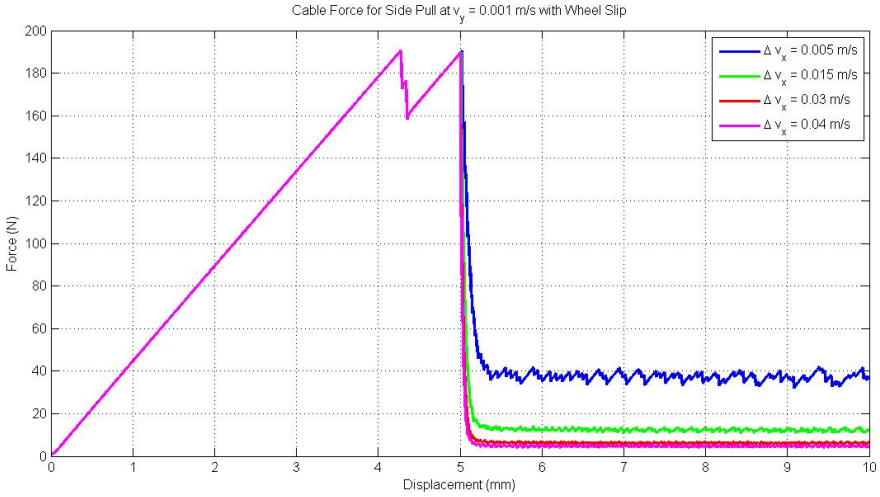
Figure 6.7(a) depicts the simulation results representing the force measured by the testing machine when pulling the robot platform sidewards, for different applications points $\boldsymbol{p}_{cable}$.

By comparing the graphs in Figure 6.7(a) with the graphs in Section 3.3 both from a qualitative and quantitative point of view, the following considerations can be done:

- The model qualitatively reproduces the real saw-tooth shape of the static lateral force as measured by the testing machine.
- The exact shape of the cable force depends on the height of the application point. The higher from the ground the application point is the more weight is perceived on right



(a)



(b)

Figure 6.7: (a) Simulation result for the cable force by pulling the robot platform sidewards at $v_y = 1\ \frac{mm}{s}$ and at different heights, when the wheels are stopped; (b) Simulation result for the cable force by pulling the robot platform sidewards at $v_y = 1\ \frac{mm}{s}$, when the wheels are spinning at different velocities.

hand side wheels (application point side) and vice versa, due to the robot tilting. This weight unbalancing makes two wheels releasing before than the other two, resulting in different shapes of the perceived force.

- The slopes of the ramps change depending on the application point height, because of the difference of the tire force at each wheel (see previous point). In particular, it may vary from $2K_{tire}$ to $4K_{tire}$, depending on whether the weight is distributed almost only on two wheels (two identical springs in parallel) or equally on all the four wheels (four identical springs in parallel).

By taking into account the two last points and the irregularity of the ground surface, we can not quantitatively evaluate the model reliability and accuracy, for instance by computing the error between the real and simulated results. However, we can claim so far that the model presented in Section 4 qualitatively represents a good approximation of the real the tire dynamics when the wheels are not spinning.

The same simulation is performed also by spinning the wheels at different angular velocities, after the maximum static lateral force is reached. In order to keep the robot in place, the same force provided by the wheel slipping is subtracted to the resultant of the longitudinal forces. The simulation is run for the different orders of magnitude of λ_{max}. In particular, as the constraint $|\lambda_i| < 1$ must be satisfied, we run the simulation for $\lambda_{max} \in [10^{-4}, 1]$. However, the results in the four different cases are identical, as expected. In fact, when $v_{ix} = 0, \omega_i \neq 0$ we always have $\lambda_i = 1 \geq \lambda_{max}$, i.e. $\mu_{ix} = 1$, therefore we always have $\mu_{iy} = \frac{v_y}{r\omega_i}$ independently from the value of λ_{max}.

The simulation results representing the cable force when pulling the robot platform sidewards, for different wheel slip velocities Δv_x, are depicted in Figure 6.7(b).

By comparing the graphs in Figure 6.7(b) with the graphs in Figure 3.5 both from a qualitative and quantitative point of view, we can conclude that, because of the irregularity of the ground surface, we can not quantitatively evaluate the model reliability and accuracy, for instance by computing the error between the real and simulated results. However, we can claim that the model presented in Section 4 qualitatively represents a good approximation of the real the tire dynamics also when the wheels are spinning.

After validating the proposed model for the tire reaction force, we also want to qualitatively validate the overall three-dimensional SSMR model, by performing different motion commands. In particular, we mostly focused on the two basic motion commands from an

odometry point of view, i.e. moving along a straight line and swiveling in place. The two types of motion are performed by providing different control velocities v_x, ω_z and setting λ_{max} to different orders of magnitude, i.e. $\lambda_{max} = 1, 10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}$. However, the value of λ_{max} does not affect the overall behavior when moving along a straight line, i.e. providing the control $v_x \neq 0, \omega_z = 0$, but it slightly affects only the wheel slipping when accelerating/decelerating, as depicted in Figure 6.8. Conversely, the value of λ_{max} significantly affects the system behavior when the robot is turning, especially during sharp turns, as it was expected. Figure 6.9 depicts two examples of the results obtained when swiveling in place, respectively for the two extreme cases $\lambda_{max} = 1, 10^{-4}$.

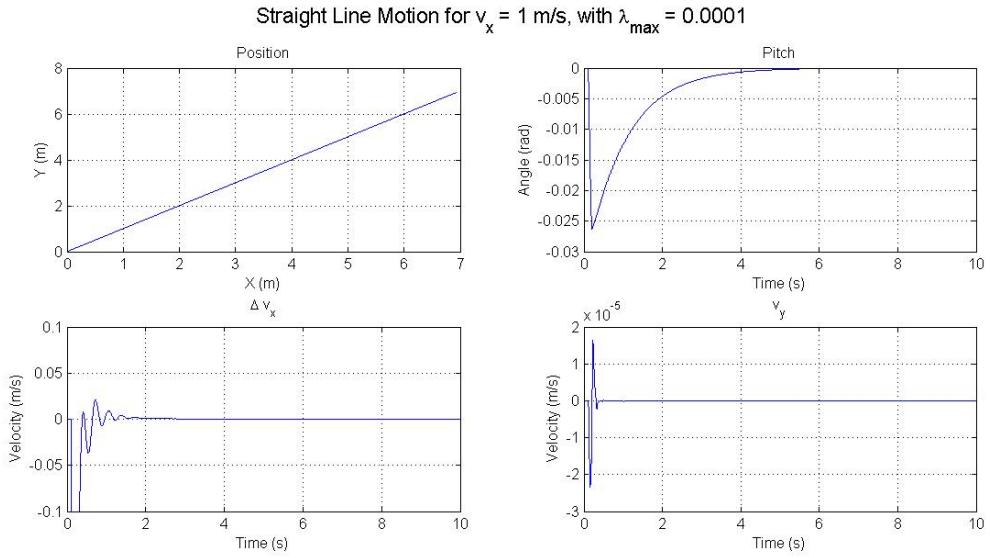
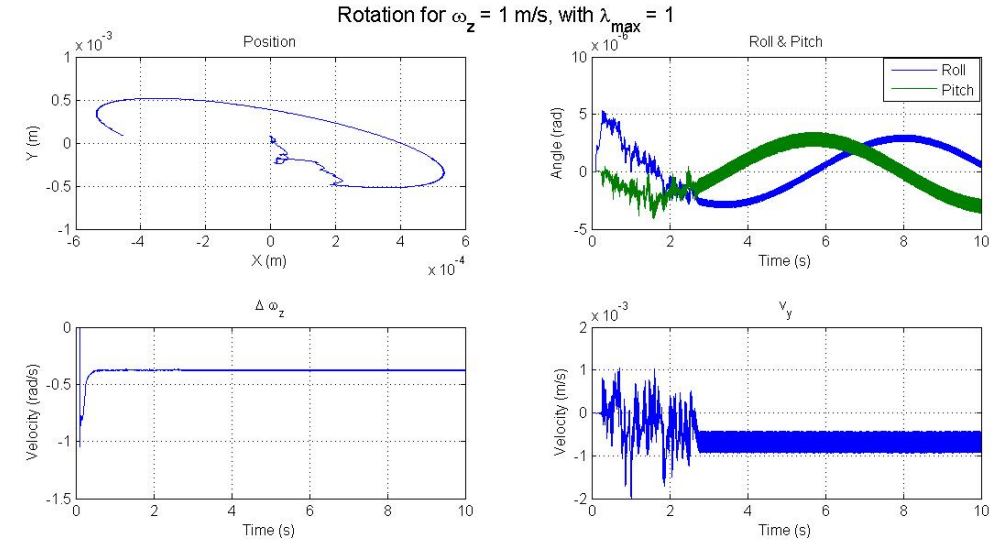


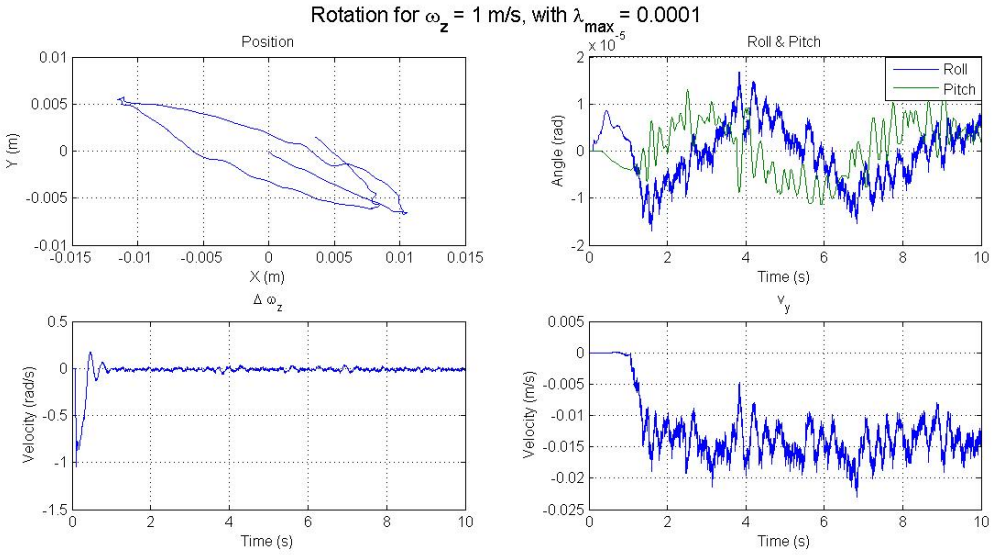
Figure 6.8: Simulation result for moving along a straight line at $v_x = 1\,\frac{m}{s}$ with $\lambda_{max} = 10^{-4}$.

Because of the absence of information about the real robot motion, i.e. its position and velocity, the results obtained during the simulation can not be evaluate from a quantitative point of view. However, by looking at the results depicted in Figure 6.8, 6.9 from a qualitative point of view, the following considerations con be done:

- The model qualitatively reproduces the real robot motion as it is coherent with the control commands provided to the model, both regarding the position and velocity errors.
- The model qualitatively reproduces the expected skid-steering motion as it provides a relatively small lateral velocity v_y when turning, due to the robot skidding properties.
- The model qualitatively reproduces the real robot jerky motion, providing both the



(a)



(b)

Figure 6.9: Simulation result for swiveling in place at $w_z = 60\,\frac{deg}{s}$ with: (a) $\lambda_{max} = 1$; (b) $\lambda_{max} = 10^{-4}$.

vibrations at relatively high and low frequency. However, the model does not quantitatively reproduce the expected large amplitude oscillations on the Roll and Pitch motion.

- The value of λ_{max} significantly affects the robot angular velocity, the lateral skid velocity and the frequency of the vibrations. In particular, the higher λ_{max} is the lower the angular velocity is (see the error $\Delta\omega_z$ in Figure 6.9(b)), the higher the lateral skid velocity is and the lower the frequency of the vibrations is.

By taking into account the last point, we can deduce that the value of λ_{max} strictly characterize the wheel/ground interaction and it might depend on both the ground and tire properties.

Furthermore, in order to qualitatively and quantitatively evaluate the model with respect to the amplitude and the frequency of the vibrations, we run the simulation for several values of ω_z between 4 $\frac{deg}{s}$ and 60 $\frac{deg}{s}$. The signals provided by the ¨Accelerometers ¨ block are sampled by a ¨Zero-Holder¨ block at 1 kHz and analyzed in the frequency domain by employing the Matlab functions defined in Appendix A, as it was done for the real data acquired from the accelerometers (see Section 3.2). Some example graphs of the low-pass filtered data with their FFT are provided in Appendix C. The graphs representing the average of the data amplitudes and the average of the three highest peak frequencies with respect to the angular velocity ω_z and different values of λ_{max} are also provided in Appendix C.

Although it was not possible to carefully characterize the model behavior and the contribution of the parameter λ_{max}, there are some features we can extract. In particular, by looking at the graphs depicted in Appendix C, the main features can be listed as follows:

- For $\lambda_{max} = 1$, there is only one frequency component which increases when increasing the robot angular velocity.

- For $\lambda_{max} > 1$, there are usually two frequency components, respectively at ¨low¨ and ¨high¨ frequency.

- For $\lambda_{max} = 10^{-1}, 10^{-2}$, the main peak frequency is the at same frequency for all the three accelerometer axes, i.e. a_x, a_y, a_z. Such a peak frequency seems to increase with the angular velocity up to 20 Hz, for relatively low angular velocities ($\omega_z < 20$), while it remains almost constant at ¨low¨ frequency for relatively high angular velocities ($\omega_z > 20$).

- For $\lambda_{max} = 10^{-3}, 10^{-4}$, the main peak frequency is the at same frequency for all the three accelerometer axes, i.e. a_x, a_y, a_z. Such a peak frequency seems to increase with the angular velocity up to 20 Hz, for relatively low angular velocities ($\omega_z < 20$), while it remains almost constant at ¨low¨ frequency for relatively high angular velocities ($\omega_z > 20$).

- For $\lambda_{max} = 1$, the amplitude of the vibrations linearly increases with the angular velocity for a_y, while it remains almost constant at relatively low value for a_x, a_z.

- For $\lambda_{max} = 10^{-1}, 10^{-2}$, the amplitude of the vibrations linearly increases and then remains almost constant, respectively at low ($\omega_z < 20$) and high ($\omega_z > 20$) angular velocity for a_y, a_x, while it remains almost constant at relatively low value for a_z.

- For $\lambda_{max} = 10^{-3}, 10^{-4}$, the amplitude of the vibrations remains almost constant at relatively high and low value respectively for a_y and $a_x a_z$.

- For all the tested cases, the amplitude of the vibrations is much smaller than the amplitudes of the real robot vibrations acquired from the accelerometers.

Finally, we can claim that the overall model qualitatively reproduces the frequencies but not the amplitudes of the real vibrations measured by the accelerometers. In particular, the value of λ_{max}, which allows the model to be ¨closer¨ to the real system behavior, seems to be of an order of magnitude of $10^{-1}, 10^{-2}$. However, by comparing the graphs in Appendix B and C and taking into account the considerations made in Section 3.2, we notice that the model does not quantitatively reproduce the amplitudes and frequencies of the real robot vibrations yet. The reasons of such a discrepancy might rely on the lack of an identification of λ_{max} or on the irregularities of the ground surface.
Thus, we conclude saying that, in order to validate the model also from a quantitative point of view, an accurate identification/estimation of the maximum wheel slip λ_{max} is needed. If after such identification the model is still inaccurate for our needs, we can either try to model the irregularities of the ground surface or propose another model.

Conclusions

In this thesis, a novel three-dimensional dynamic model of SSMRs, including a spring-damper tire model, was presented. Some experimental data acquired from three 1-axis accelerometers and a force sensor were presented and analyzed to characterize the nature of the robot vibrations and the tire reaction forces. Such a model allows to qualitatively reproduce the real robot jerky motion in a simulation environment. On the basis of the experimental data, a spring-mass-damper model separately for tire lateral, longitudinal and vertical reaction force was provided. A dynamic friction model, based on the work proposed in [4], [5], was also provided to include the contribution of the wheel longitudinal slip into the reaction force model. Finally, after identifying the robot geometric and dynamic parameters, the Simulink model of the three-dimensional skid-steering motion reproducing the real system behavior was presented and validated by qualitatively comparing the results of the simulation to the data acquired from the accelerometers and the force sensor.

The proposed model qualitatively reproduces the real behavior of the lateral force, both with and without wheel slip, as measured by the testing machine. It also reproduces the expected skid-steering motion, providing a relatively small lateral velocity v_y when turning, and the real robot jerky motion, providing both the vibrations at relatively high and low frequency. However, the model does not quantitatively reproduce the amplitudes and frequencies of the real robot vibrations yet. The reasons of such a discrepancy might rely on the lack of an identification of the maximum wheel slip λ_{max} and the irregularities of the ground surface.

Thus, the next step will be to provide an accurate identification/estimation of the maximum wheel slip λ_{max} and validate the model also from a quantitative point of view. Then, if the model, after such identification, will be considered enough accurate, some non-linear control techniques might be developed to stabilize the robot jerky motion. Conversely, if the model will be still inaccurate for our needs, we can either try to model the irregularities of the ground surface or propose another model for the tire reaction forces.

Appendixes

Appendix A

```
%% Main program for the frequency analysis of data acquired from accelerometers

%% Define parameters
step = 2;
rangeW = 2:step:20;
filterType = 'butter';
order = 10;
axis = 3;
plotOpt = 2;
accOpt = 1;
tRange = 'middle';
%tRange = 0.5:0.001:(0.5+2^11/1000);
fRange = [1,50];
nPeaks = 3;
conf = 'OnWheel-noFrame-concrete';

%% Declare cells and matrices
nData = length(rangeW);
data = cell(1,nData);
dataFFT = cell(1,nData);
dataLowFilt = cell(1,nData);
dataLowFiltFFT = cell(1,nData);
dataFilt = cell(1,nData);
time = cell(1,nData);
freq = cell(1,nData);
fPeak = zeros(nData,nPeaks,3);
peak = zeros(nData,nPeaks,3);
fMax = zeros(nData,3);
amplitude = zeros(nData,3);

%% Parse data and find peak frequencies
for i=rangeW
```

```
33
34      fileName = ['v0-w',num2str(i),'-t20-',conf,'.xlsx'];
35
36      [data{i/step},time{i/step},dataFFT{i/step},freq{i/step}] = ...
37          parseData(fileName,accOpt,0,tRange,fRange);
38  %     plotData(data{i/step},time{i/step},dataFFT{i/step},freq{i/step},...
39  %         fileName,plotOpt,fRange);
40
41      dataLowFilt{i/step} = filterData(data{i/step},'low',filterType,order,fRange);
42
43      amplitude(i/step,:) = (max(dataLowFilt{i/step})-min(dataLowFilt{i/step}))/2;
44
45      [dataLowFiltFFT{i/step},freq{i/step}] = computeFFT(dataLowFilt{i/step},tRange);
46      plotData(dataLowFilt{i/step},time{i/step},dataLowFiltFFT{i/step},freq{i/step},...
47          fileName,plotOpt,fRange);
48
49      [fPeak(i/step,:,:),peak(i/step,:,:),dataFilt{i/step}] = ...
50          findPeakFreq(dataLowFilt{i/step},dataLowFiltFFT{i/step},freq{i/step},...
51          filterType,order,fRange,nPeaks);
52      fMax(i/step,:) = squeeze(fPeak(i/step,1,:));
53
54      figure,hold on;
55      plot(time{i/step},data{i/step}(:,axis));
56      plot(time{i/step},dataLowFilt{i/step}(:,axis),'-g','LineWidth',2);
57      plot(time{i/step},dataFilt{i/step}(:,axis)+dataLowFiltFFT{i/step}(1,axis),...
58          '-r','LineWidth',2);
59
60      if accOpt == 1
61          ylabel('Acceleration (m/s^2)');
62      else
63          ylabel('Data (V)');
64      end
65      xlabel('Time (s)');
66      title(['Data "',fileName,'", fPeak = ',num2str(fMax(i/step,axis)),' Hz']);
67      legend('Real y-axis data','Lowpass-Filtered y-axis data','Bandpass-Filtered y-axis data');
68      grid on;
69
70  end
71
72  %% Plot peak frequencies and amplitudes with respect to the angular velocity
73  figure,plot(fMax,amplitude,'o','LineWidth',2);
74  title([conf,'; Max peak freq. VS Amplitude']);
75  xlabel('fPeaks (Hz)'),ylabel('Amplitude (V)');
76  legend('x-axis','y-axis','z-axis');
77  grid on;
78
79  figure,plot(rangeW,fMax,'-o','LineWidth',2);
```

```
title([conf,'; Angular vel. VS Max peak freq.']);
xlabel('Angular velocity (deg/s)'),ylabel('fPeaks (Hz)');
legend('x-axis','y-axis','z-axis');
grid on;

figure,plot(rangeW,amplitude,'-o','LineWidth',2);
title([conf,'; Angular Vel. VS Amplitude']);
xlabel('Angular velocity (deg/s)'),ylabel('Amplitude (V)');
legend('x-axis','y-axis','z-axis');
grid on;

diffPeak = squeeze(peak(:,1,:)-peak(:,2,:))./squeeze(peak(:,1,:));
figure,plot(rangeW,diffPeak,'-o','LineWidth',2);
title([conf,'; Angular vel. VS Diff. of peaks']);
xlabel('Angular velocity (deg/s)'),ylabel('(peak1-peak2)/peak1');
legend('x-axis','y-axis','z-axis');
grid on;

figure,plot(rangeW,squeeze(fPeak(:,:,1)),'-o','LineWidth',2);
title([conf,'; Angular vel. VS Peak freq. for x-axis']);
xlabel('Angular velocity (deg/s)'),ylabel('fPeaks (Hz)');
legend('1^{st} peak','2^{nd} peak','3^{rd} peak');
grid on;

figure,plot(rangeW,squeeze(fPeak(:,:,2)),'-o','LineWidth',2);
title([conf,'; Angular vel. VS Peak freq. for y-axis']);
xlabel('Angular velocity (deg/s)'),ylabel('fPeaks (Hz)');
legend('1^{st} peak','2^{nd} peak','3^{rd} peak');
grid on;

figure,plot(rangeW,squeeze(fPeak(:,:,3)),'-o','LineWidth',2);
title([conf,'; Angular vel. VS Peak freq. for z-axis']);
xlabel('Angular velocity (deg/s)'),ylabel('fPeaks (Hz)');
legend('1^{st} peak','2^{nd} peak','3^{rd} peak');
grid on;
```

```
function [data,time,dataFFT,freq] = parseData(fileName,accOpt,plotOpt,tRange,fRange)


% load data
for i=1:3
    dataLoaded = xlsread(fileName,4-i);
    dataRaw(:,i) = dataLoaded(11:end,2);
end
time = dataLoaded(11:end,1);

```

```
% compute fft of dataRaw
[dataRawFFT,freq] = computeFFT(dataRaw,tRange);

% eventually transform the data from voltage to acceleration
if accOpt == 1

    %dataFilt = 3.4*9.8/5*(dataRaw-repmat(abs(dataRawFFT(1,:)),rData,1));
    dataFilt = 3.4*9.8/5*(dataRaw-2.6);
    [dataFiltFFT,freq] = computeFFT(dataFilt,tRange);

    data = dataFilt;
    dataFFT = dataFiltFFT;

else
    data = dataRaw;
    dataFFT = dataRawFFT;
end

% eventually plot the data and its FFT
if plotOpt ~= 0
    plotData(data,time,dataFFT,freq,fileName,plotOpt,fRange);
end
```

```
function dataFilt = filterData(data,type,filterType,order,fRange)

[rData,cData] = size(data);
Fs = 1000;  % Sampling Frequency
N = order;  % Order

%% Design the filter

if strcmp(type,'low')

    Fc = fRange(2);   % Cutoff Frequency

    % Construct an FDESIGN object and call its 'filterType' method.
    h = fdesign.lowpass('N,F3dB', N, Fc, Fs);
    Hd = design(h,filterType);

elseif strcmp(type,'band')

    Fc1 = fRange(1);   % First Cutoff Frequency
    Fc2 = fRange(2);   % Second Cutoff Frequency

    % Construct an FDESIGN object and call its 'filterType' method.
    h = fdesign.bandpass('N,F3dB1,F3dB2', N, Fc1, Fc2, Fs);
```

```
    Hd = design(h,filterType);

elseif strcmp(type,'high')

    Fc = fRange(1);    % Cutoff Frequency

    % Construct an FDESIGN object and call its 'filterType' method.
    h = fdesign.lowpass('N,F3dB', N, Fc, Fs);
    Hd = design(h,filterType);

else
    error('Not valid filter type!');
end

%% Filter the data
dataFilt = zeros(rData,cData);

for i=1:cData
    dataFilt(:,i) = filtfilt(Hd.sosMatrix,Hd.ScaleValues,data(:,i));
end
```

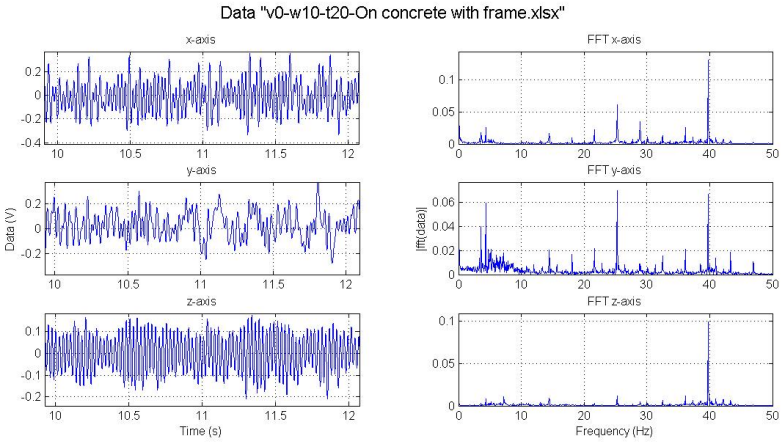
```
function [dataFFT,f] = computeFFT(data,tRange)

[rData,cData] = size(data);
Fs = 1000;

if ischar(tRange)

    NFFT = 2^nextpow2(rData)/2; % Previous power of 2 from length of data

    if strcmp(tRange,'beginning')
        data = data(1:NFFT,:);
    elseif strcmp(tRange,'middle')
        data = data(floor(end/2)-NFFT/2+1:floor(end/2)+NFFT/2,:);
    elseif strcmp(tRange,'end')
        data = data(end-NFFT:end,:);
    elseif strcmp(tRange,'all')
        data = data(1:end,:);
    else
        error('Not valid tRange!');
    end

elseif isnumeric(tRange) && (tRange(1)*Fs >= 1 && tRange(2)*Fs <= rData)

    tRange = uint16(tRange*Fs);
    data = data(tRange,:);
```

```
    NFFT = 2^nextpow2(length(tRange))/2; % Previous power of 2 from length of data

else
    error('Not valid tRange!');
end

% compute fft of data
L = length(data(:,1));
f = Fs/2*linspace(0,1,NFFT/2+1);
dataFFT = zeros(L,cData);

for i=1:cData
    dataFFT(:,i) = fft(data(:,i))/L;
end

dataFFT = [dataFFT(1,:); 2*dataFFT(2:NFFT/2+1,:)];
```

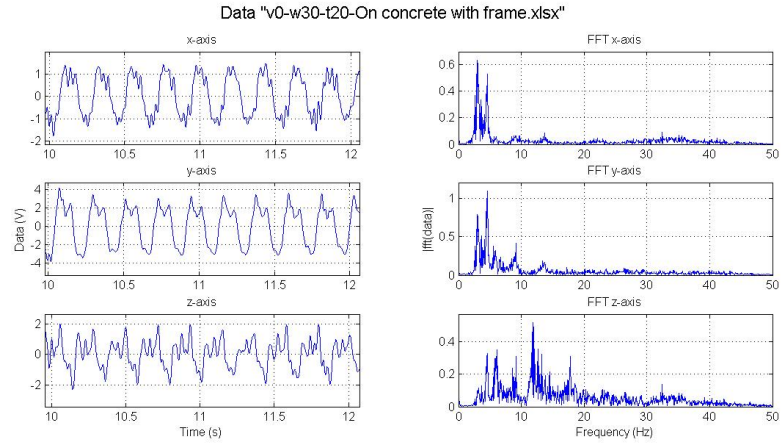
```
function [fPeak,peak,dataFilt] = findPeakFreq(data,dataFFT,freq,filterType,order,fRange,nPeaks)

[rData,cData] = size(data);

fStep = freq(2)-freq(1);
indStart = ceil(fRange(1)/fStep);
indStop = ceil(fRange(2)/fStep);
bandWd = 0.5;

% Find the peak in dataFFT
ind = zeros(nPeaks,cData);
peak = zeros(nPeaks,cData);
fPeak = zeros(nPeaks,cData);

for i=1:cData
    %[peak(i),ind(i)] = max(abs(dataFFT(indStart:indStop,i))); % find the peak
    [peak(:,i),ind(:,i)] = findpeaks(abs(dataFFT(indStart:indStop,i)),...
        'minpeakdistance',ceil(1/fStep),'npeaks',nPeaks,'sortstr','descend');
    fPeak(:,i) = freq(ind(:,i)+indStart-1); % find the frequency of the peak
end

%% Design the Bandpass filter and filter the data

% All frequency values are in Hz.
Fs = 1000;  % Sampling Frequency

N   = order;  % Order
Fc1 = fPeak(1,:)-bandWd;   % First Cutoff Frequency
Fc2 = fPeak(1,:)+bandWd;   % Second Cutoff Frequency
```

```
30
31 % Filter the data
32 dataFilt = zeros(rData,cData);
33
34 for i=1:cData
35     % Construct an FDESIGN object and call its 'filterType' method.
36     h = fdesign.bandpass('N,F3dB1,F3dB2', N, Fc1(i), Fc2(i), Fs);
37     Hd = design(h,filterType);
38     dataFilt(:,i) = filtfilt(Hd.sosMatrix,Hd.ScaleValues,data(:,i));
39 end
```

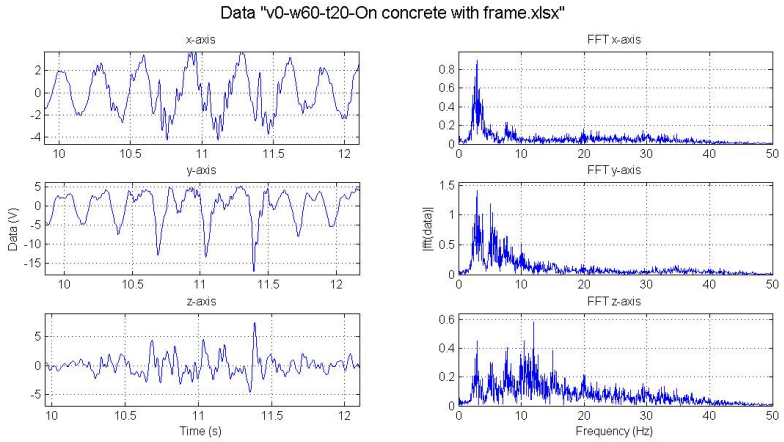
Appendix B



(a)

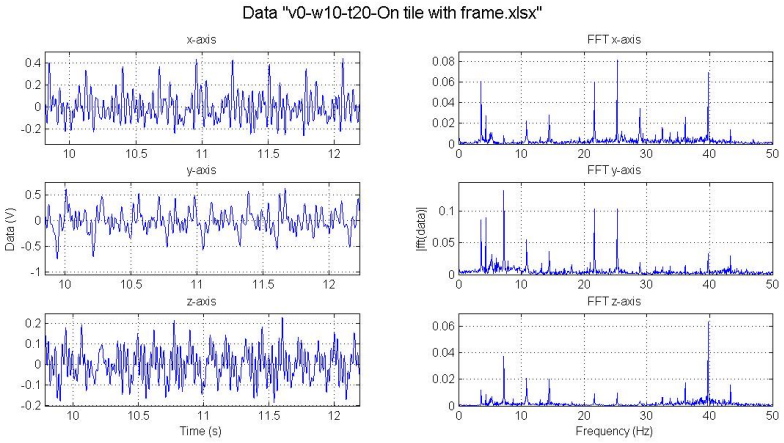


(b)

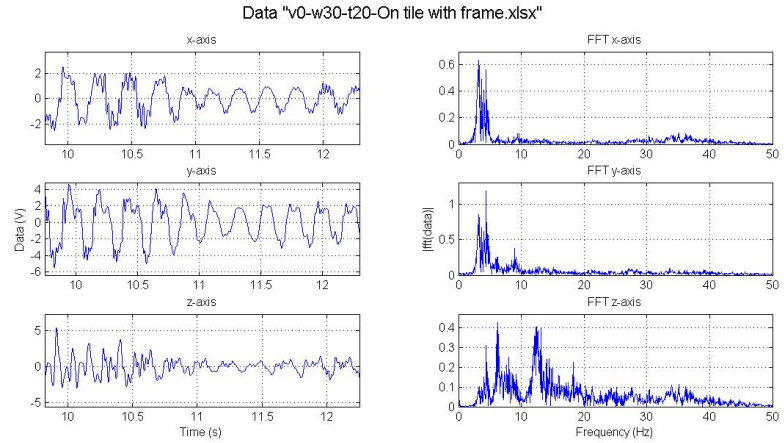


(c)

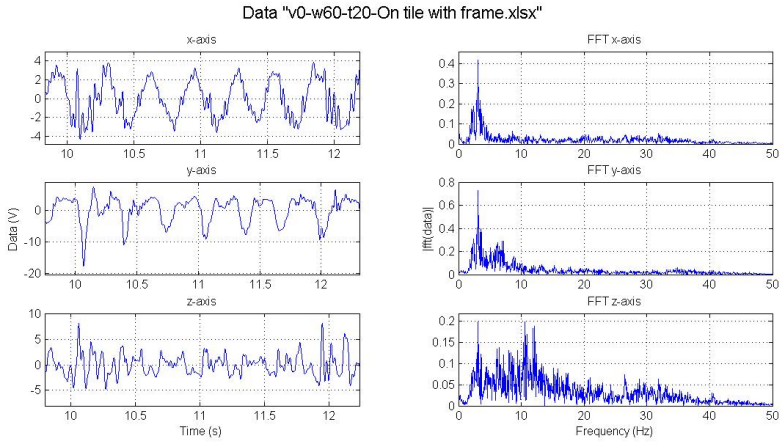
Figure 6.10: Low-pass filtered data from accelerometers and their FFT on concrete for: (a) $\omega_z^* = 10\,\frac{deg}{s}$; (b) $\omega_z^* = 30\,\frac{deg}{s}$; (c) $\omega_z^* = 60\,\frac{deg}{s}$.



(a)

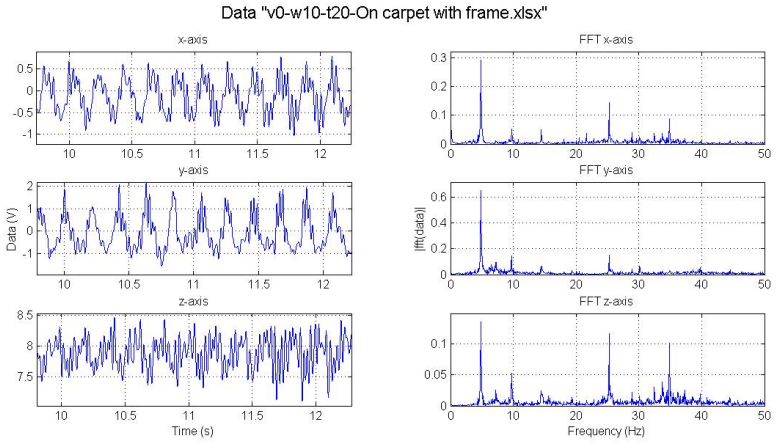


(b)

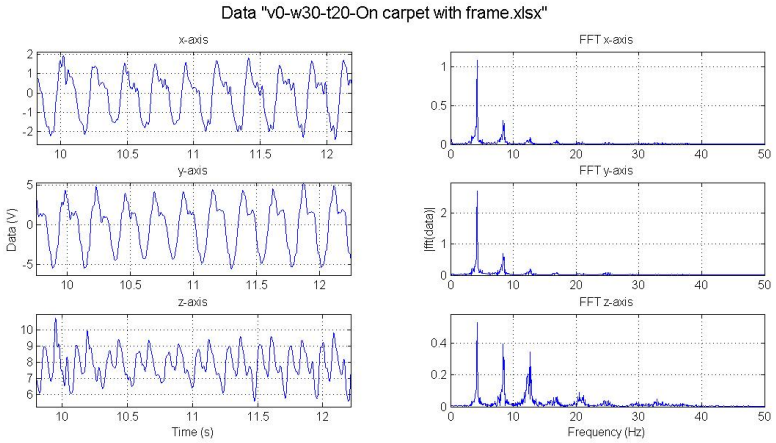


(c)

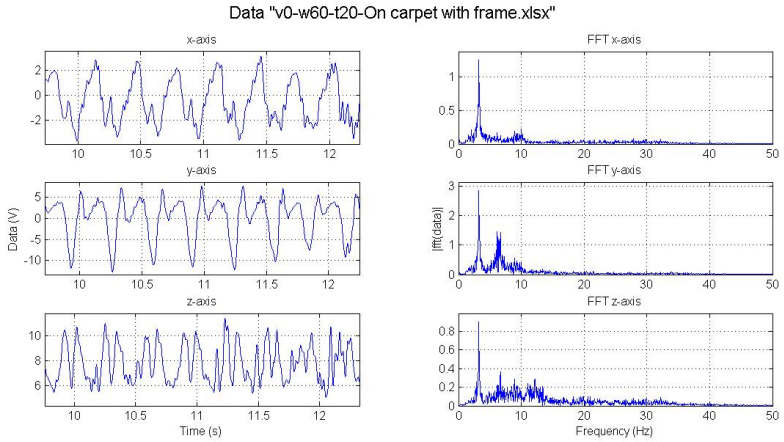
Figure 6.11: Low-pass filtered data from accelerometers and their FFT on tile for: (a) $\omega_z^* = 10\,\frac{deg}{s}$; (b) $\omega_z^* = 30\,\frac{deg}{s}$; (c) $\omega_z^* = 60\,\frac{deg}{s}$.



(a)

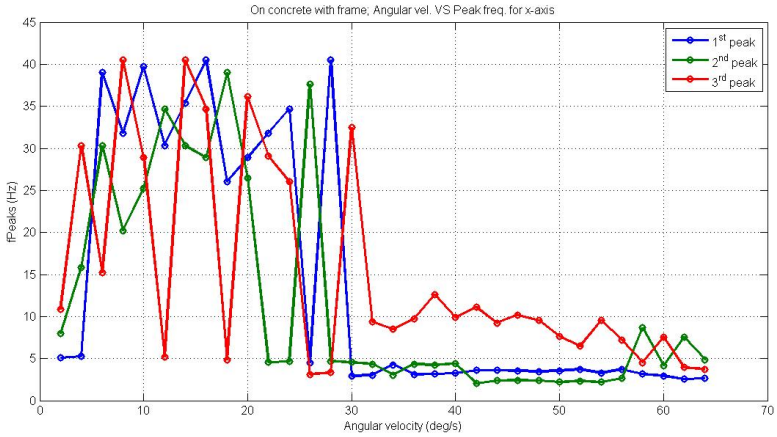


(b)

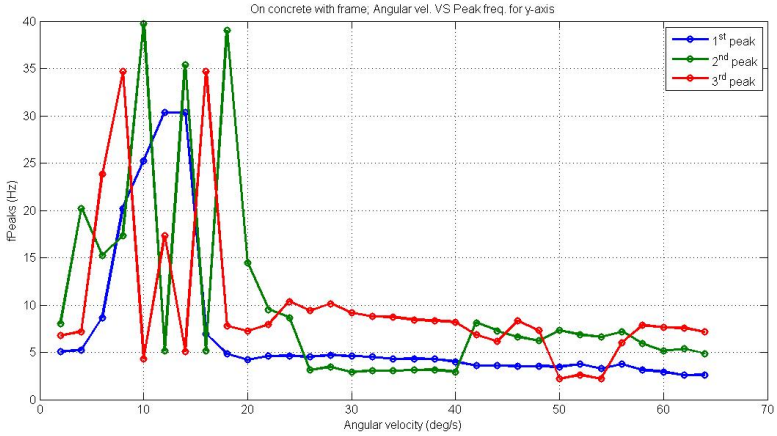


(c)

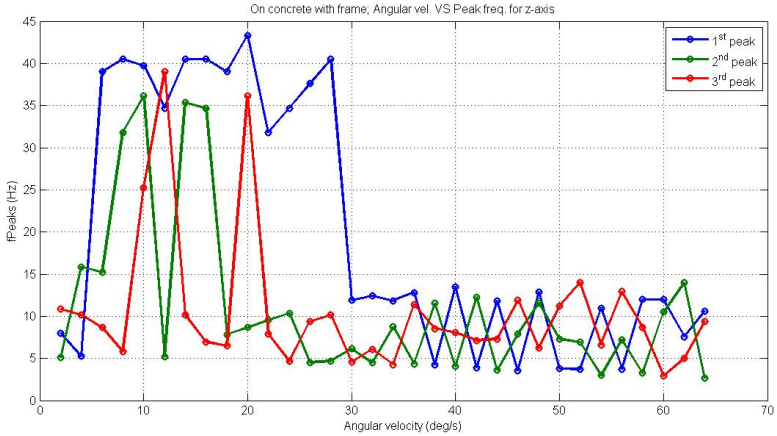
Figure 6.12: Low-pass filtered data from accelerometers and their FFT on carpet for: (a) $\omega_z^* = 10\,\frac{deg}{s}$; (b) $\omega_z^* = 30\,\frac{deg}{s}$; (c) $\omega_z^* = 60\,\frac{deg}{s}$.



(a)

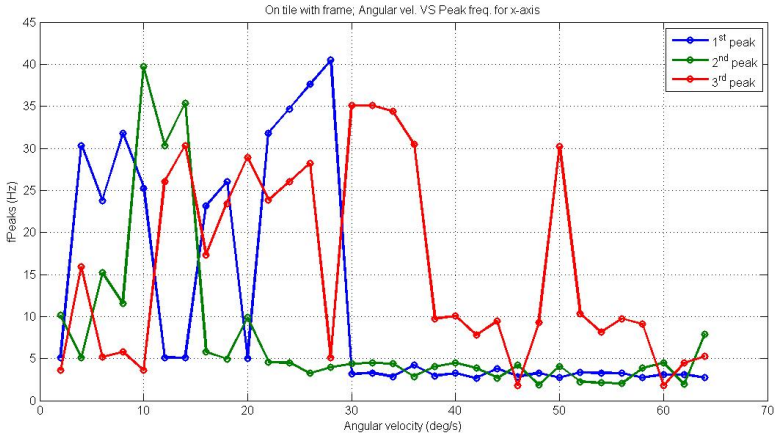


(b)

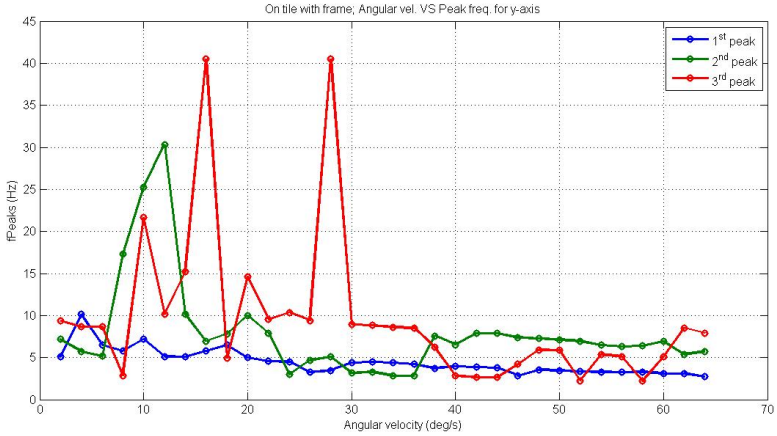


(c)

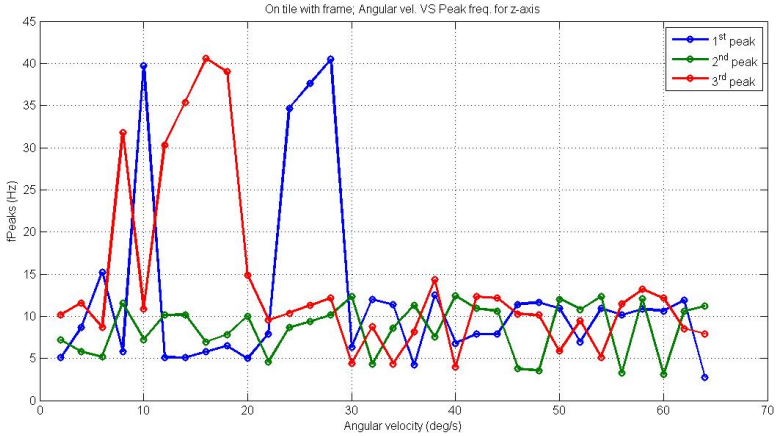
Figure 6.13: Peak frequencies on concrete for: (a) a_x; (b) a_y; (c) a_z.



(a)

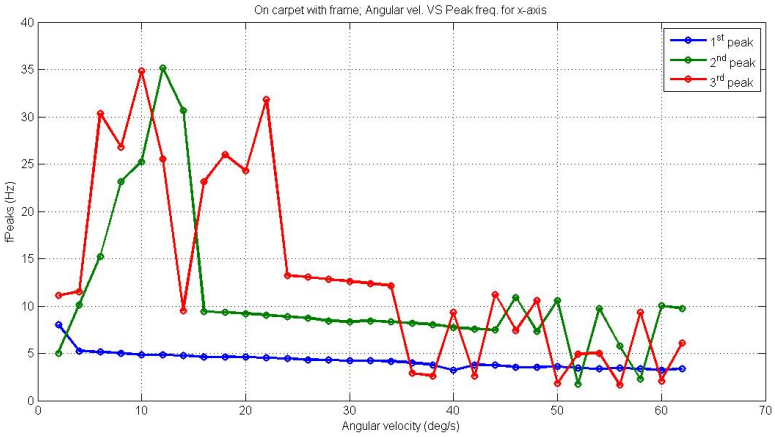


(b)

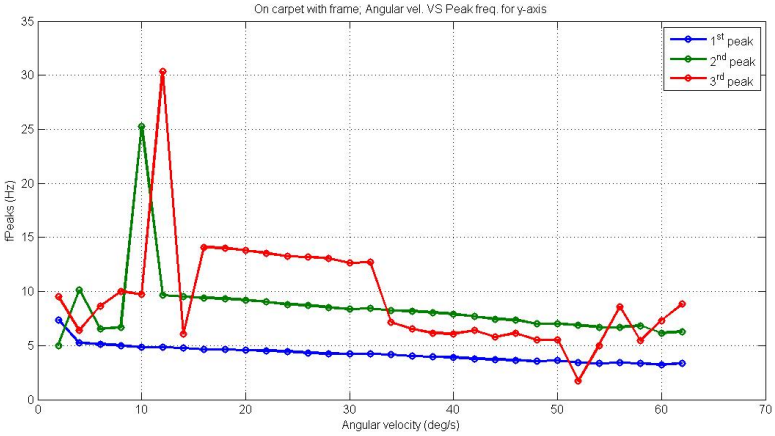


(c)

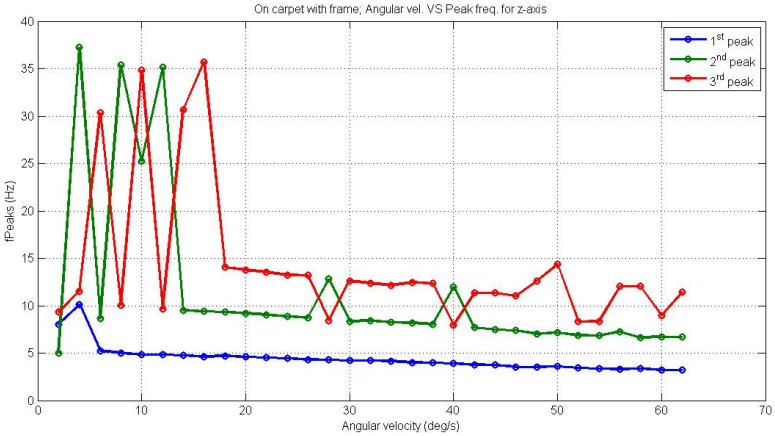
Figure 6.14: Peak frequencies on tile for: (a) a_x; (b) a_y; (c) a_z.



(a)

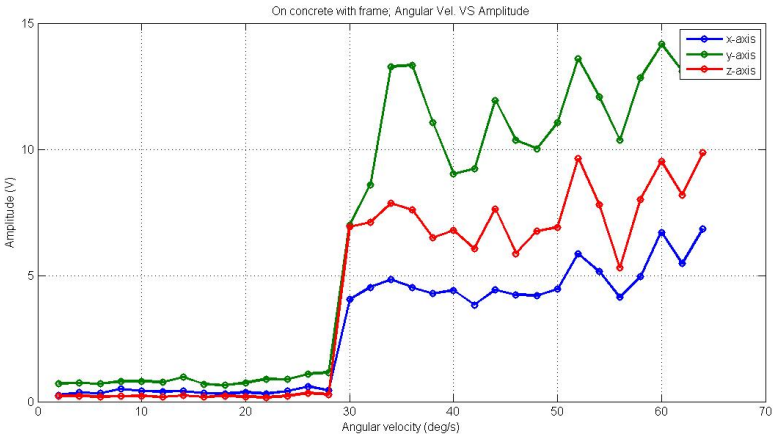


(b)

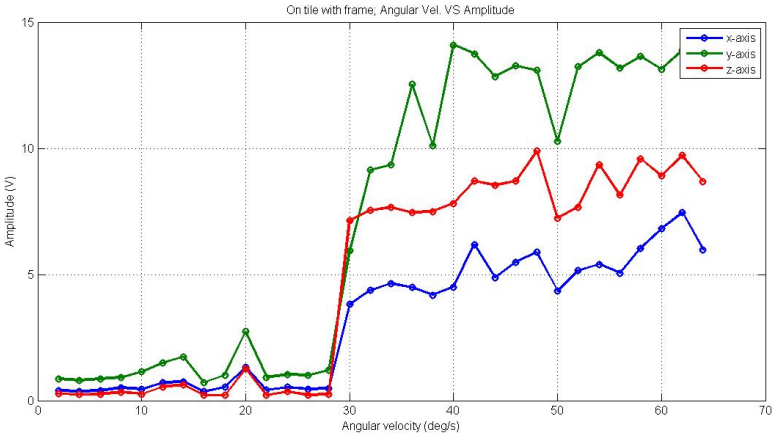


(c)

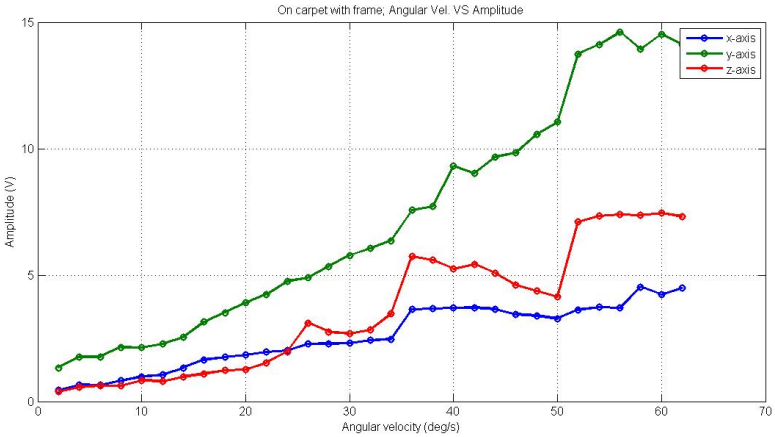
Figure 6.15: Peak frequencies on carpet for: (a) a_x; (b) a_y; (c) a_z.



(a)

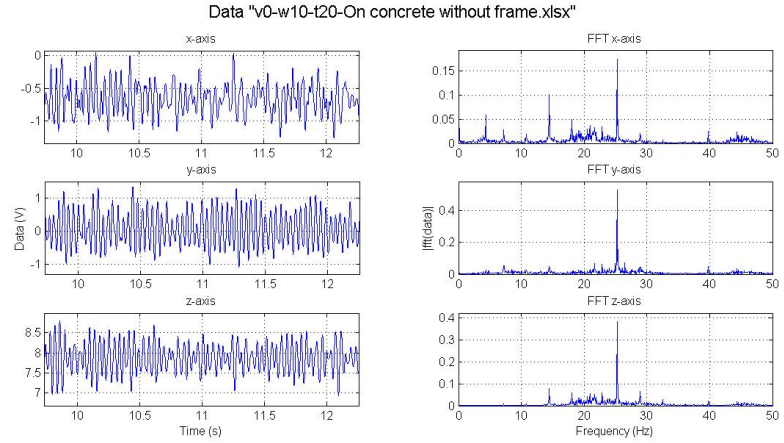


(b)

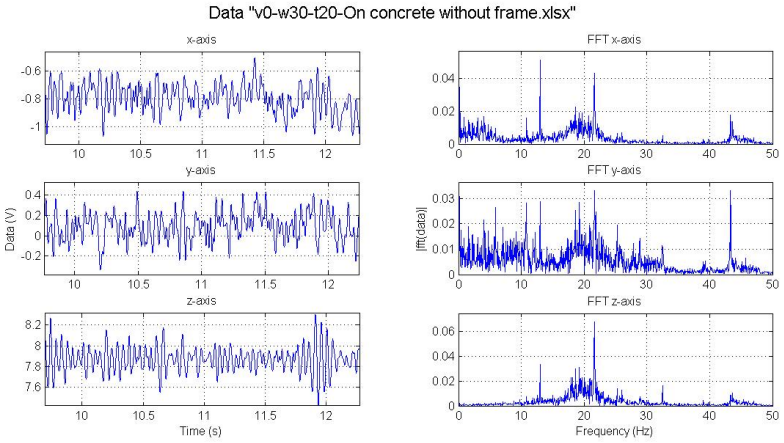


(c)

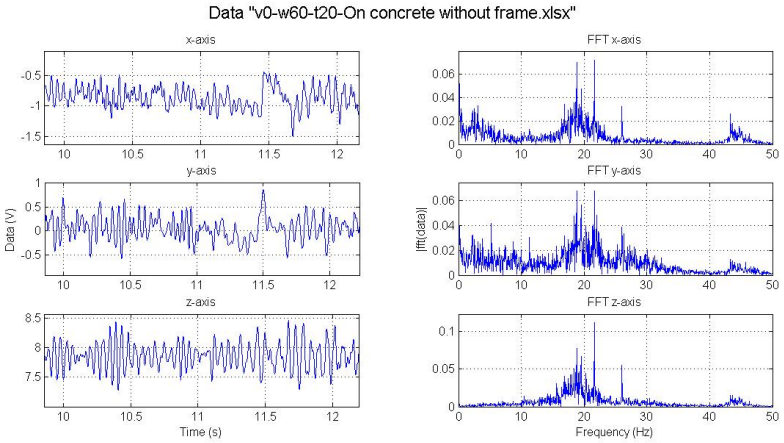
Figure 6.16: Amplitude of the data on: (a) Concrete; (b) Tile; (c) Carpet.



(a)

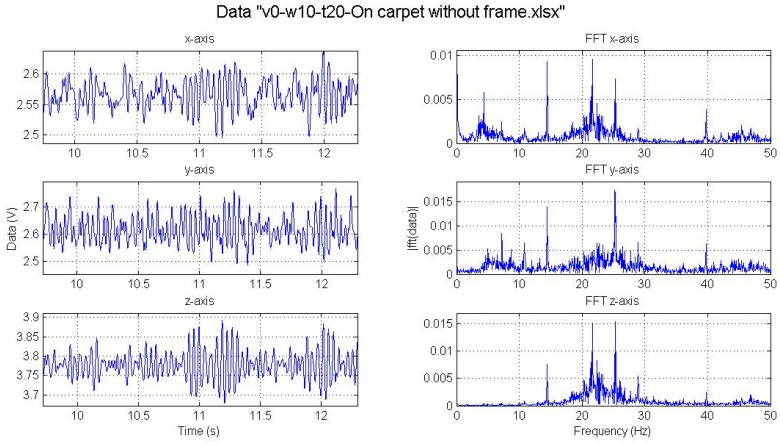


(b)

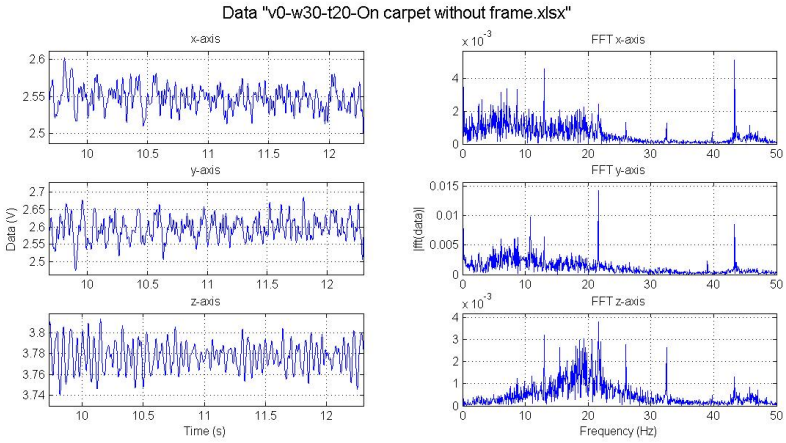


(c)

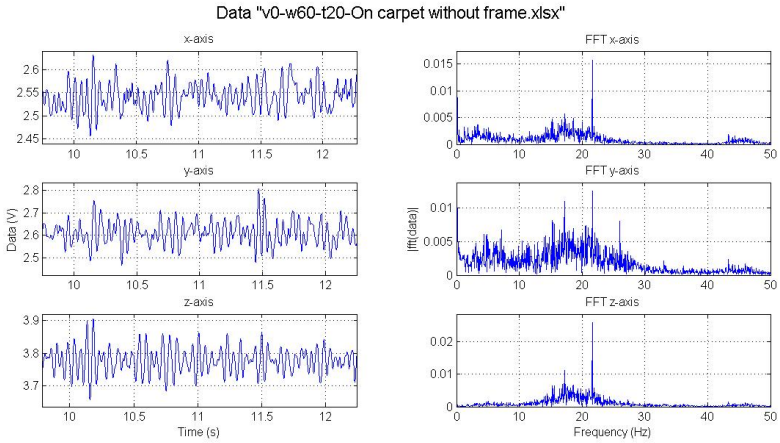
Figure 6.17: Low-pass filtered data from accelerometers and their FFT without the aluminum frame on concrete for: (a) $\omega_z^* = 10\ \frac{deg}{s}$; (b) $\omega_z^* = 30\ \frac{deg}{s}$; (c) $\omega_z^* = 60\ \frac{deg}{s}$.



(a)

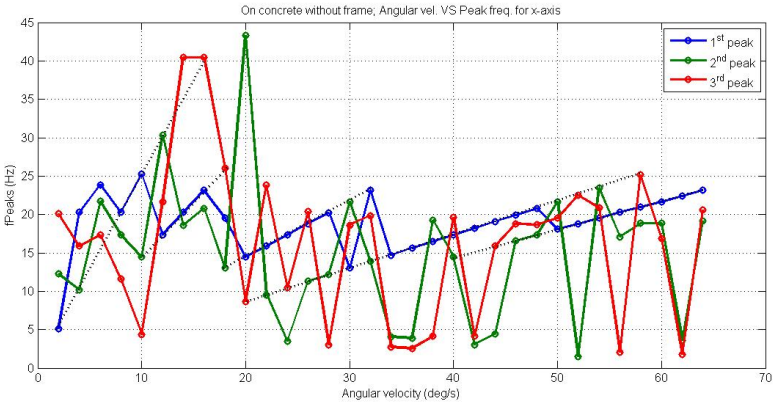


(b)

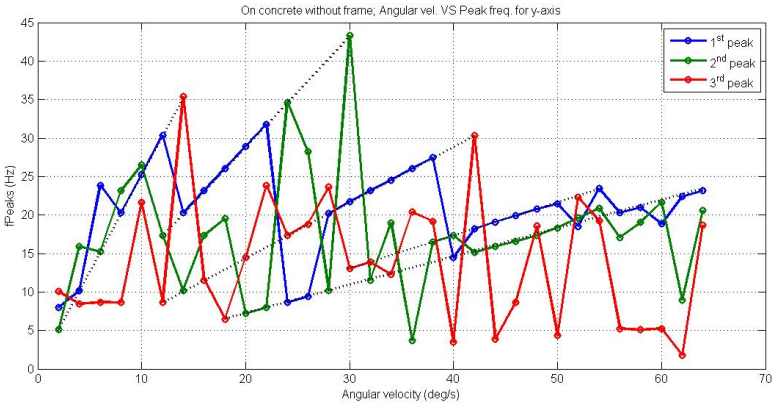


(c)

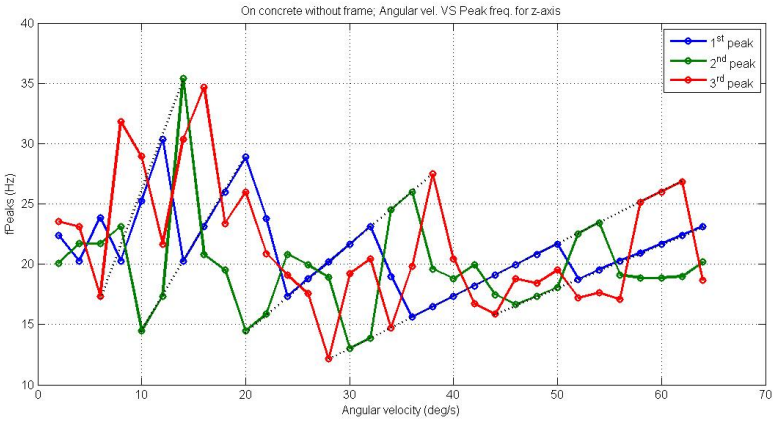
Figure 6.18: Low-pass filtered data from accelerometers and their FFT without the aluminum frame on carpet for: (a) $\omega_z^* = 10\ \frac{deg}{s}$; (b) $\omega_z^* = 30\ \frac{deg}{s}$; (c) $\omega_z^* = 60\ \frac{deg}{s}$.



(a)

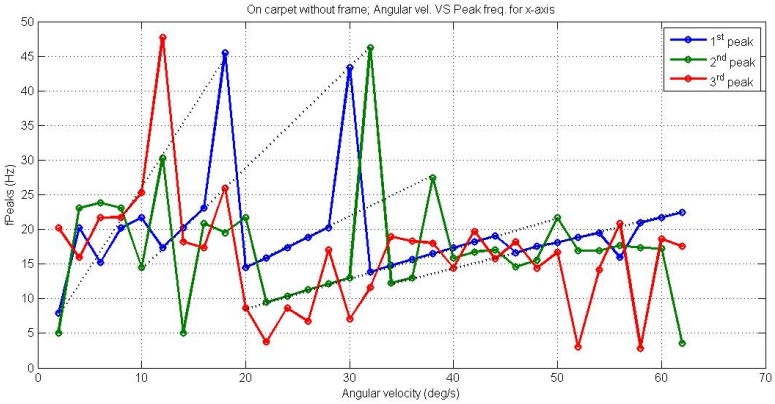


(b)

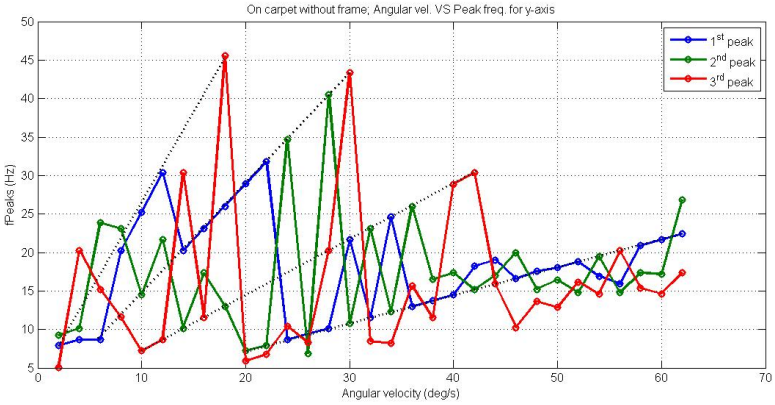


(c)

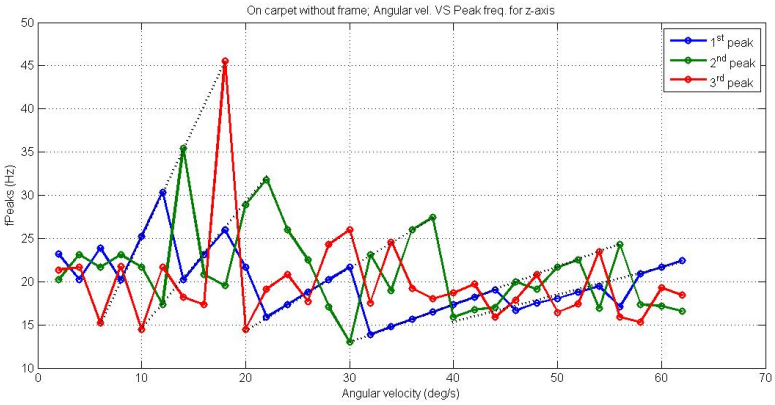
Figure 6.19: Peak frequencies without the aluminum frame on concrete for: (a) a_x; (b) a_y; (c) a_z.



(a)

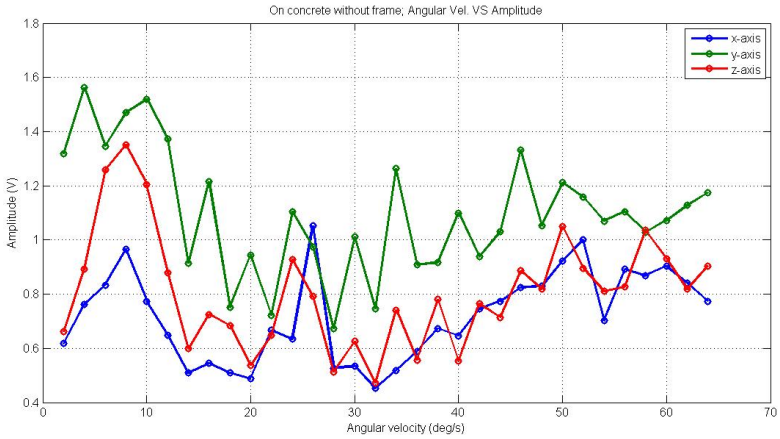


(b)

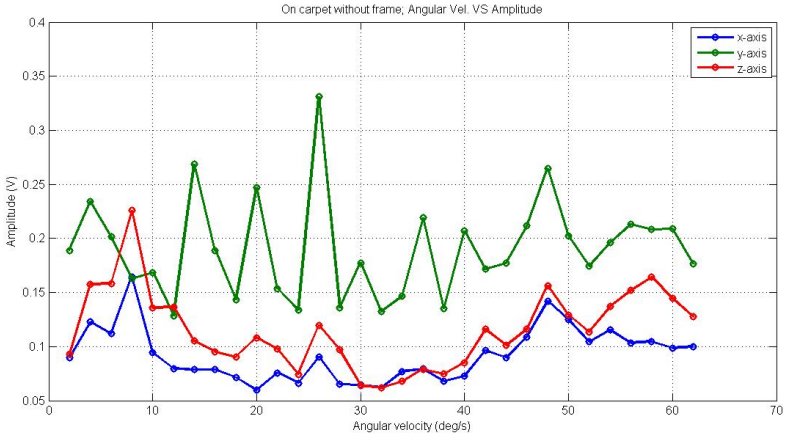


(c)

Figure 6.20: Peak frequencies without the aluminum frame on carpet for: (a) a_x; (b) a_y; (c) a_z.

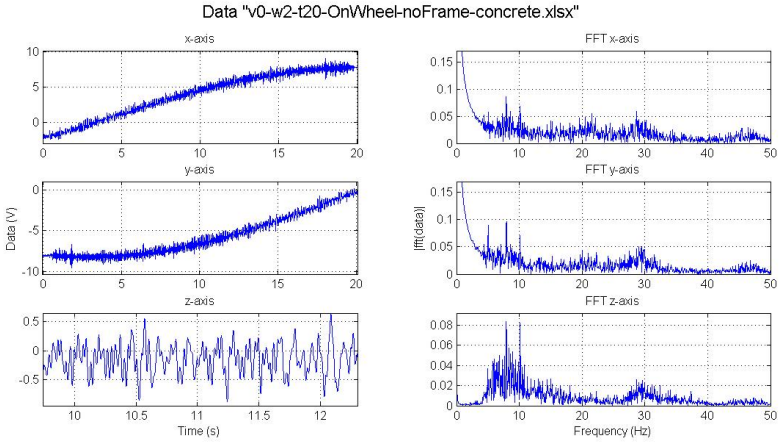


(a)

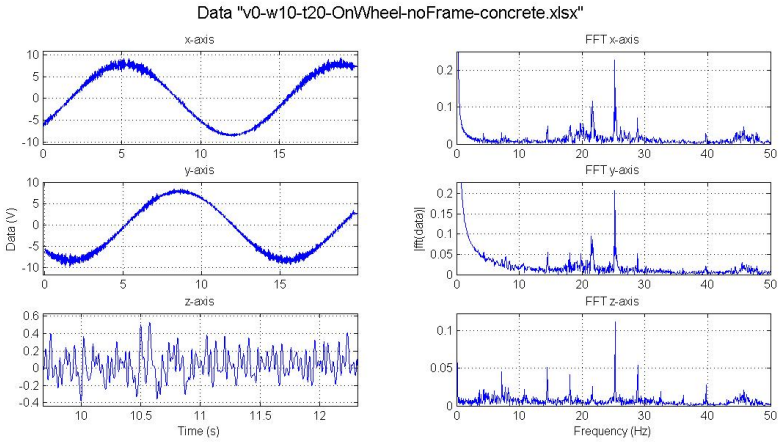


(b)

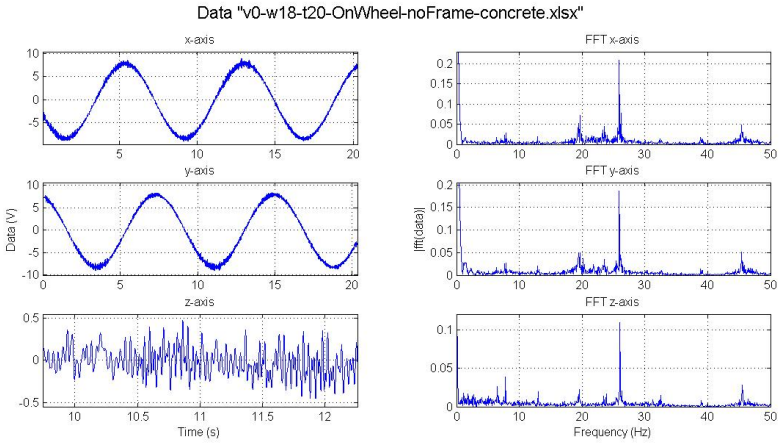
Figure 6.21: Amplitude of the data without the aluminum frame on: (a) Concrete; (b) Carpet.



(a)

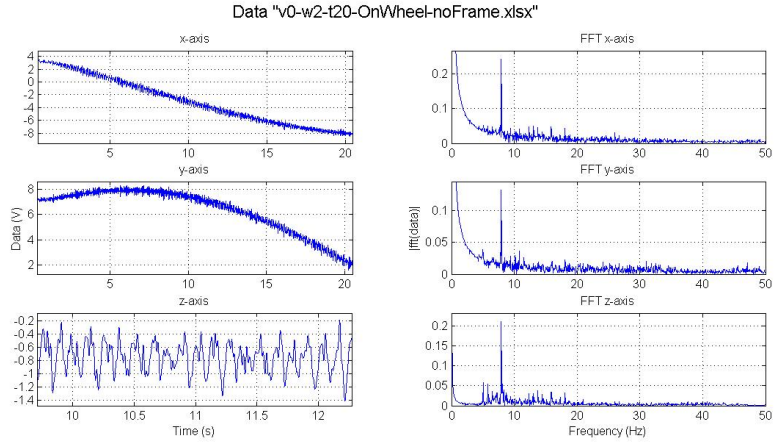


(b)

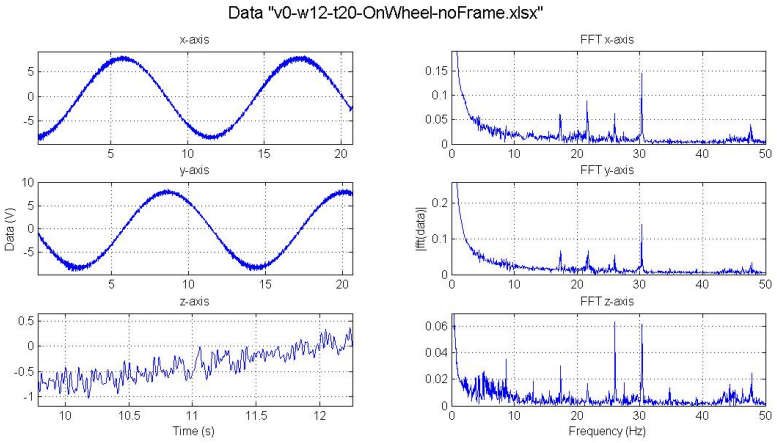


(c)

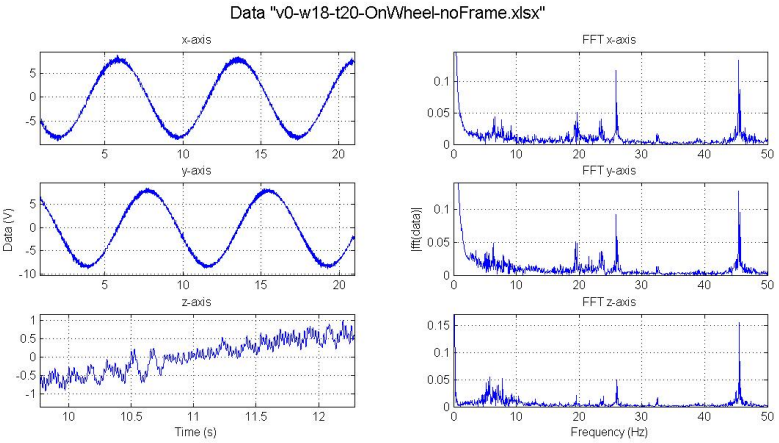
Figure 6.22: Low-pass filtered data from accelerometers and their FFT with the accelerometers on a wheel on concrete for: (a) $\omega_z^* = 2\ \frac{deg}{s}$; (b) $\omega_z^* = 10\ \frac{deg}{s}$; (c) $\omega_z^* = 18\ \frac{deg}{s}$.



(a)

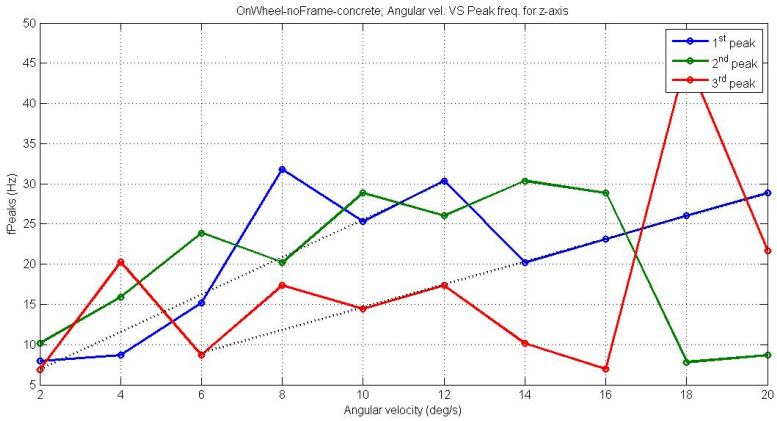


(b)

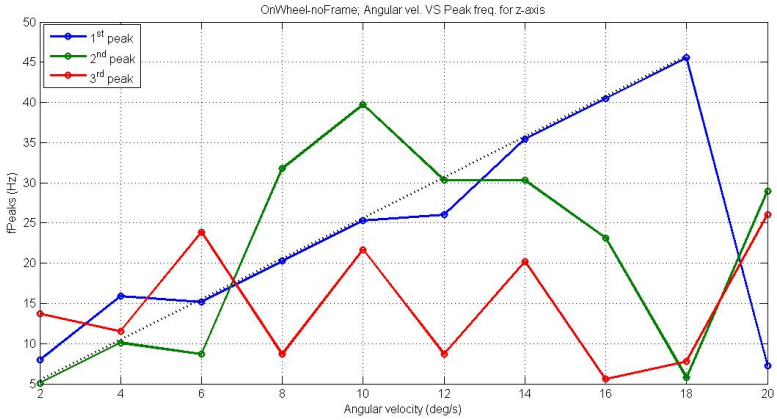


(c)

Figure 6.23: Low-pass filtered data from accelerometers and their FFT with the accelerometers on a wheel on carpet for: (a) $\omega_z^* = 2\ \frac{deg}{s}$; (b) $\omega_z^* = 10\ \frac{deg}{s}$; (c) $\omega_z^* = 18\ \frac{deg}{s}$.



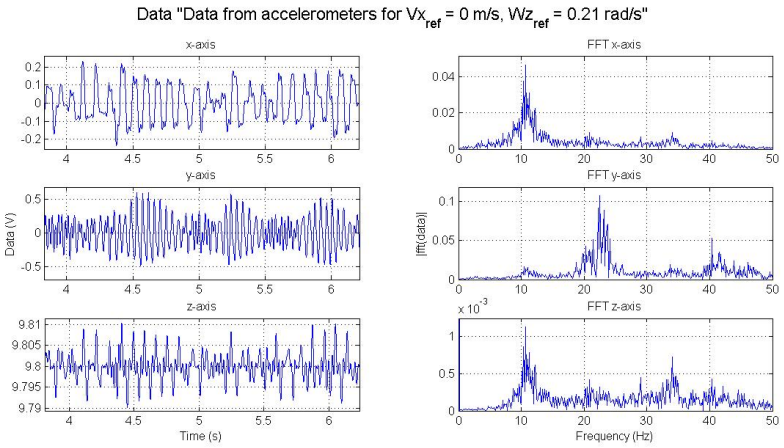
(a)



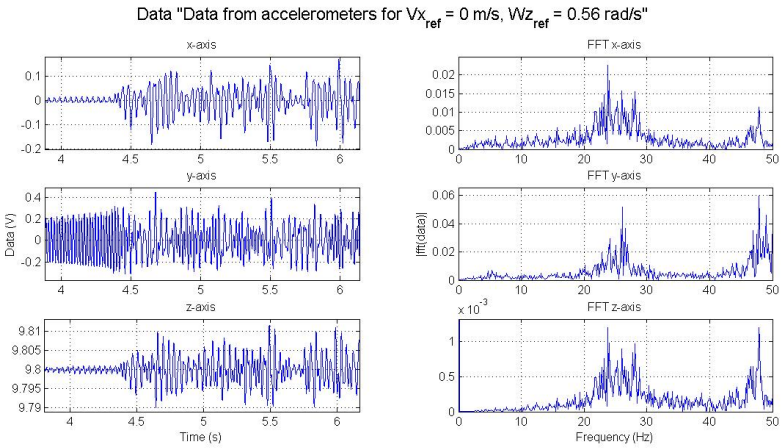
(b)

Figure 6.24: Peak frequencies with the accelerometers on a wheel on: (a) Concrete; (b) Carpet.

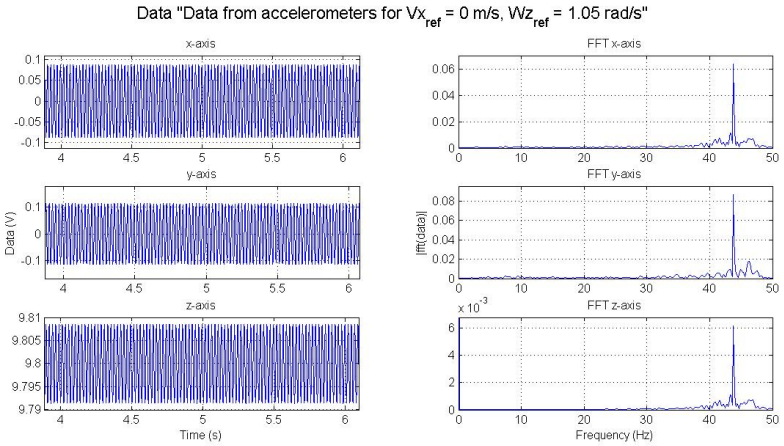
Appendix C



(a)

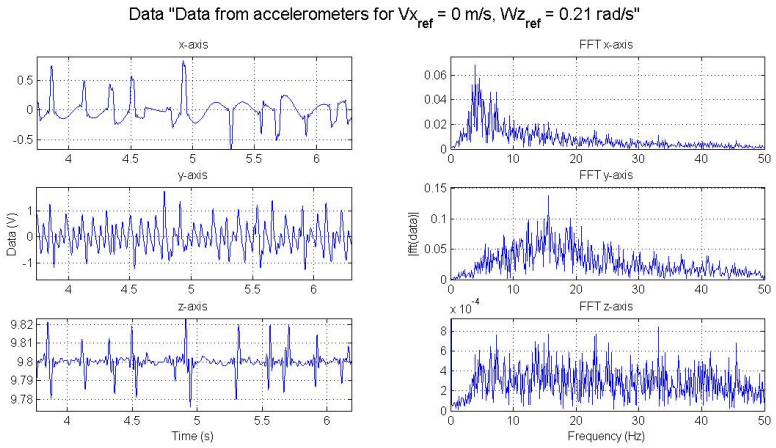


(b)

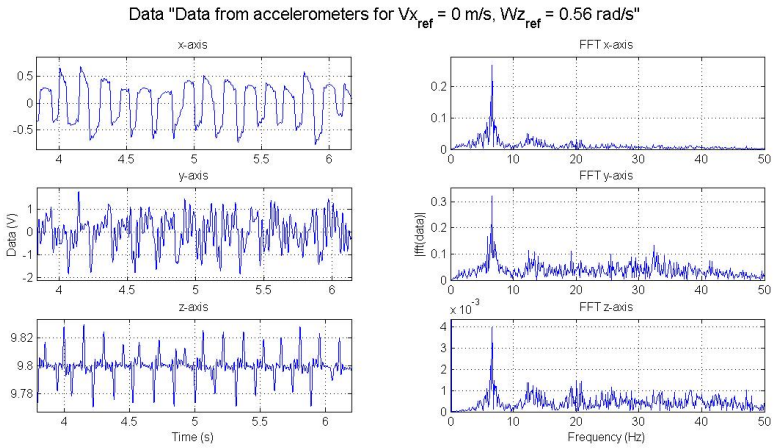


(c)

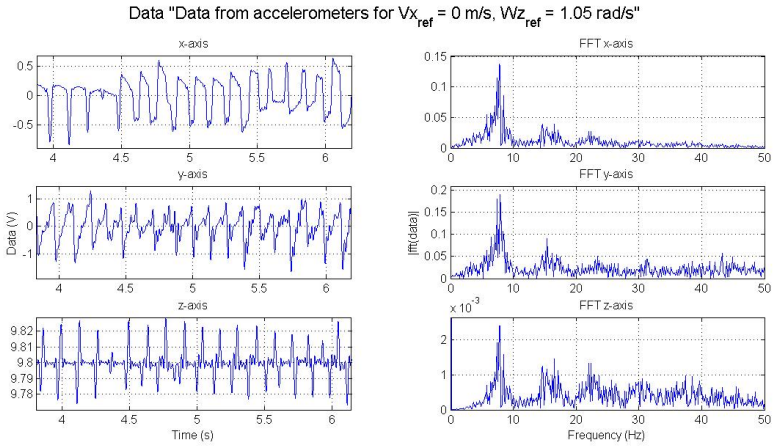
Figure 6.25: Simulation results from accelerometers and their FFT with $\lambda_{max} = 1$ for: (a) $\omega_z^* = 10\,\frac{deg}{s}$; (b) $\omega_z^* = 30\,\frac{deg}{s}$; (c) $\omega_z^* = 60\,\frac{deg}{s}$.



(a)

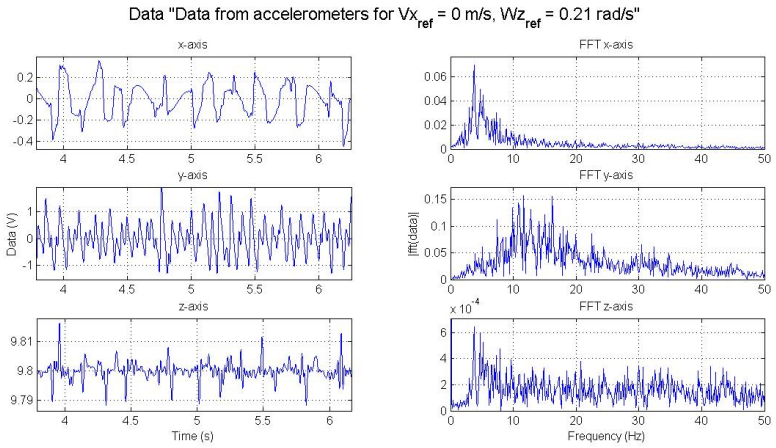


(b)

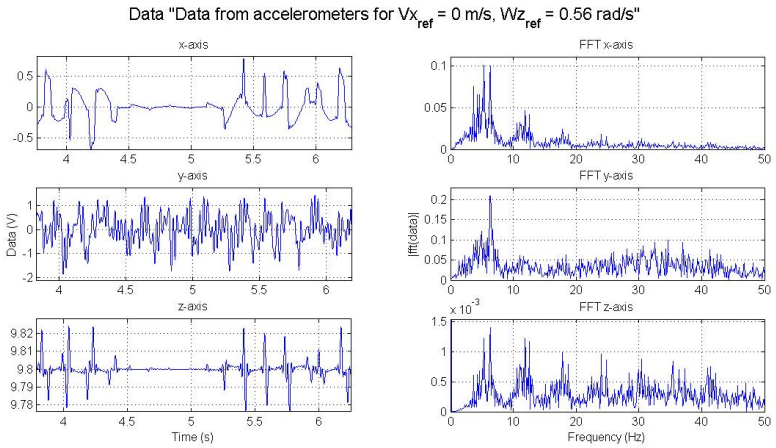


(c)

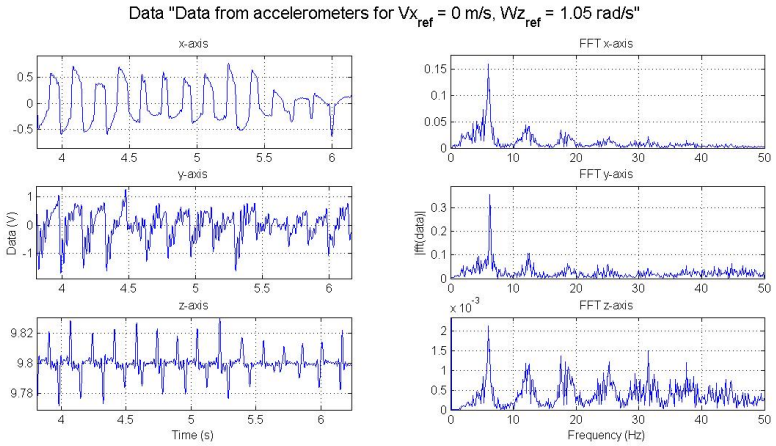
Figure 6.26: Simulation results from accelerometers and their FFT with $\lambda_{max} = 10^{-1}$ for: (a) $\omega_z^* = 10\,\frac{deg}{s}$; (b) $\omega_z^* = 30\,\frac{deg}{s}$; (c) $\omega_z^* = 60\,\frac{deg}{s}$.



(a)

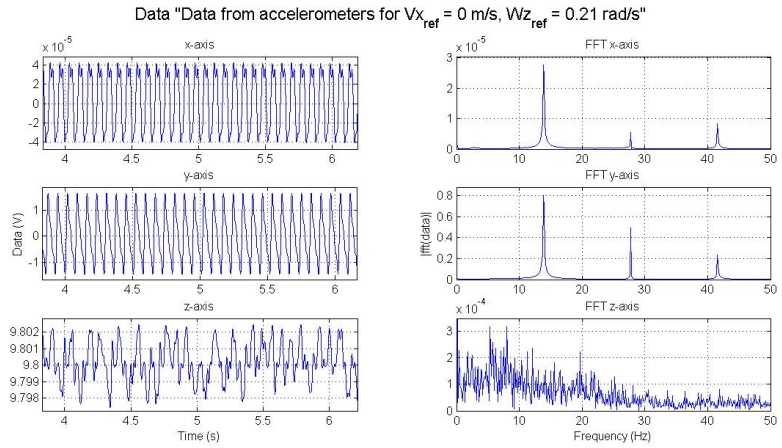


(b)

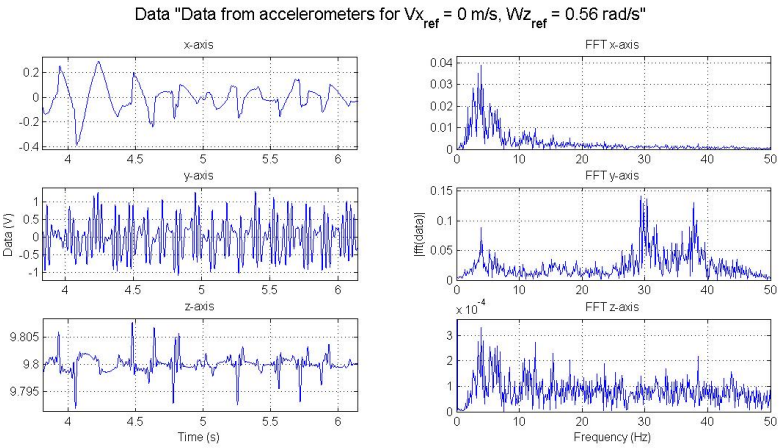


(c)

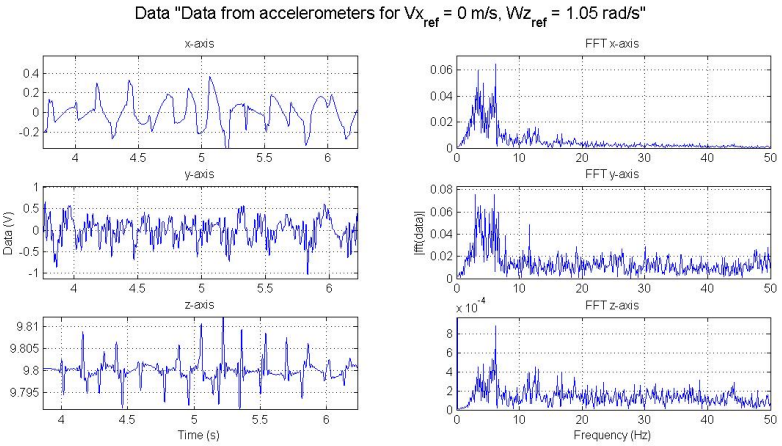
Figure 6.27: Simulation results from accelerometers and their FFT with $\lambda_{max} = 10^{-2}$ for: (a) $\omega_z^* = 10\,\frac{deg}{s}$; (b) $\omega_z^* = 30\,\frac{deg}{s}$; (c) $\omega_z^* = 60\,\frac{deg}{s}$.



(a)

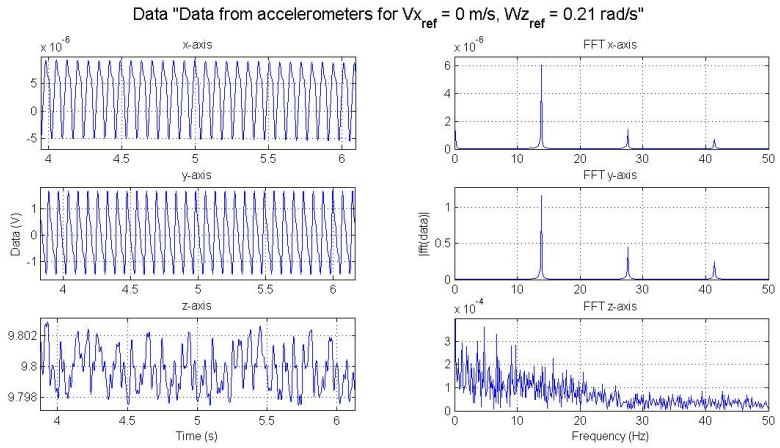


(b)

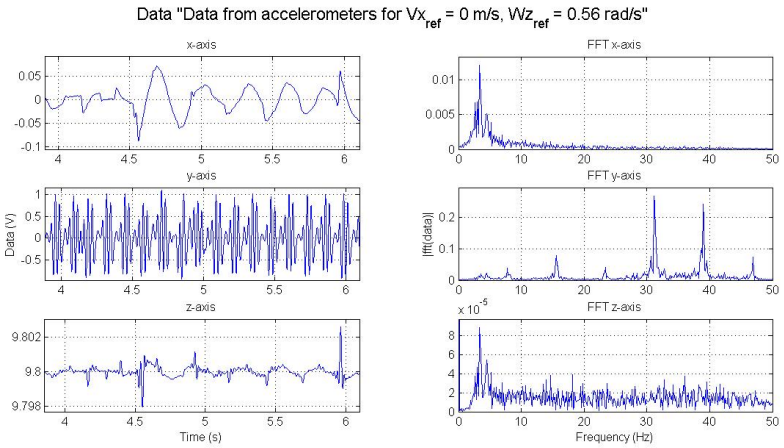


(c)

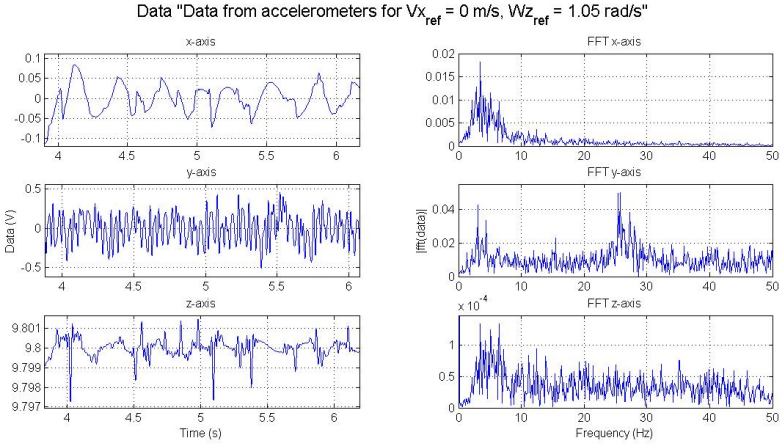
Figure 6.28: Simulation results from accelerometers and their FFT with $\lambda_{max} = 10^{-3}$ for: (a) $\omega_z^* = 10\,\frac{deg}{s}$; (b) $\omega_z^* = 30\,\frac{deg}{s}$; (c) $\omega_z^* = 60\,\frac{deg}{s}$.



(a)

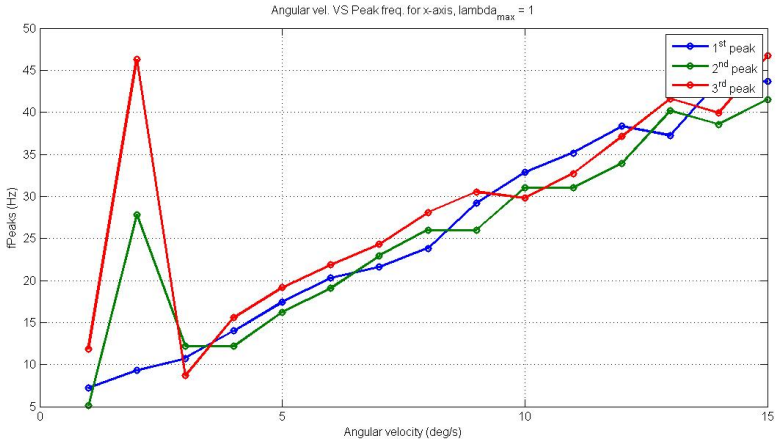


(b)

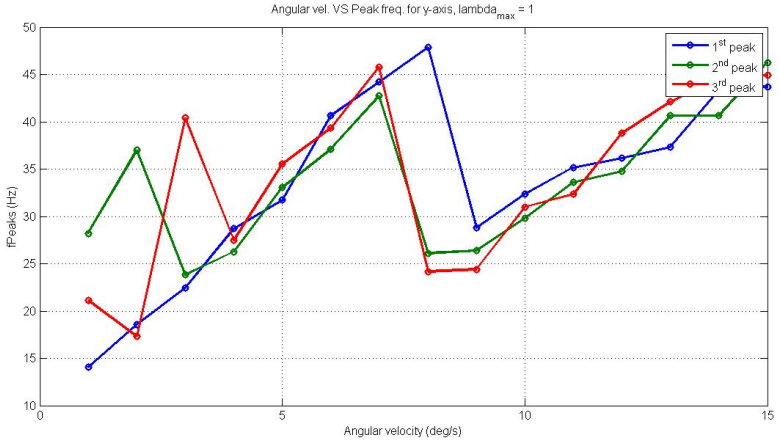


(c)

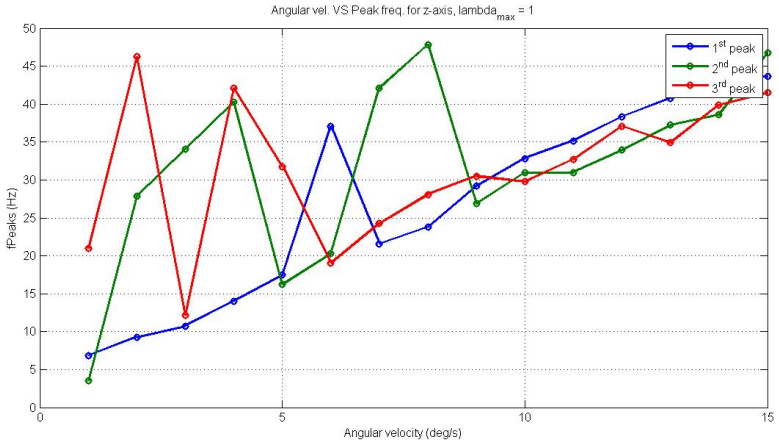
Figure 6.29: Simulation results from accelerometers and their FFT with $\lambda_{max} = 10^{-4}$ for: (a) $\omega_z^* = 10\,\frac{deg}{s}$; (b) $\omega_z^* = 30\,\frac{deg}{s}$; (c) $\omega_z^* = 60\,\frac{deg}{s}$.



(a)

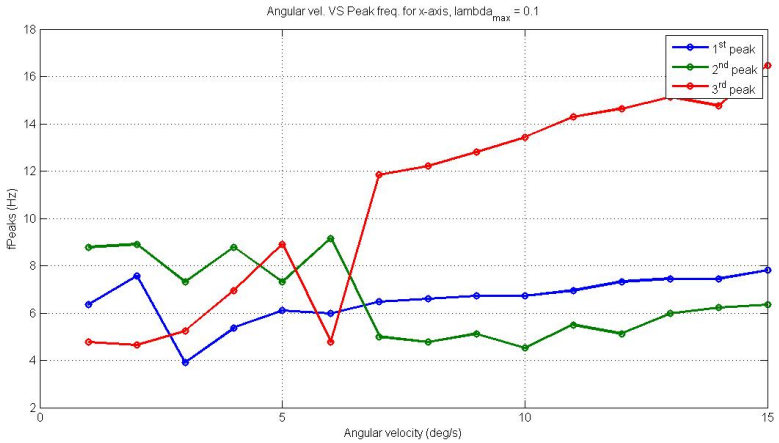


(b)

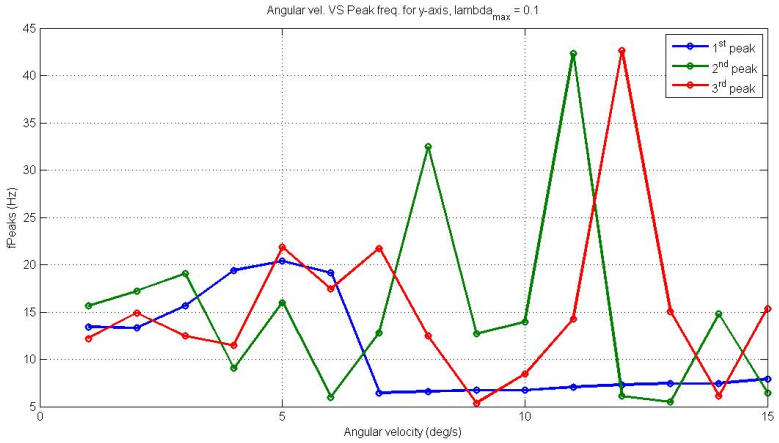


(c)

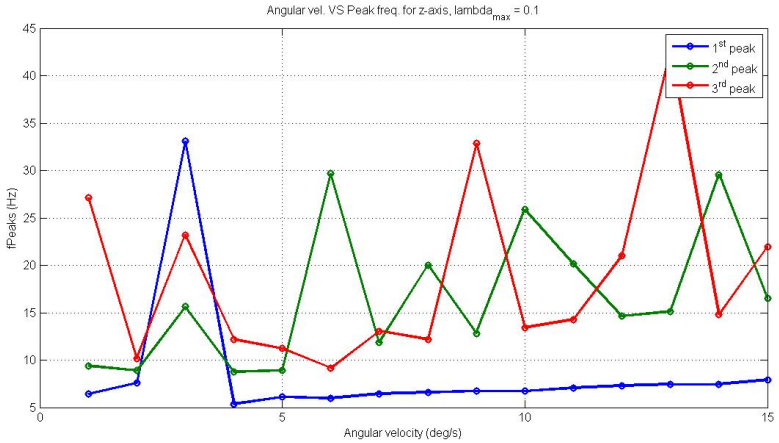
Figure 6.30: Peak frequencies of simulation results with $\lambda_{max} = 1$ for: (a) a_x; (b) a_y; (c) a_z.



(a)

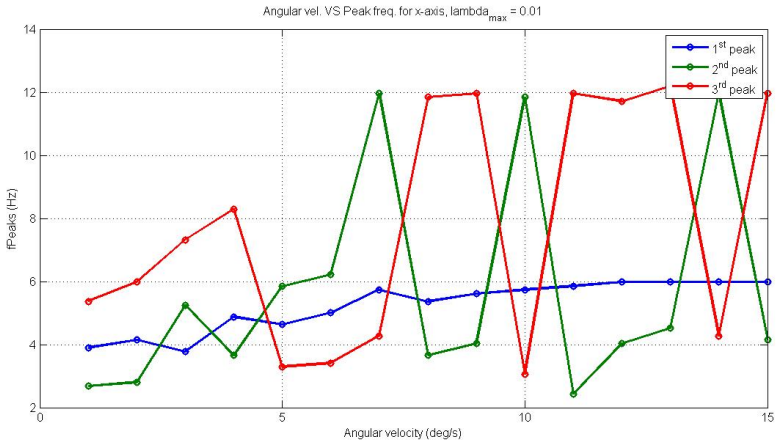


(b)

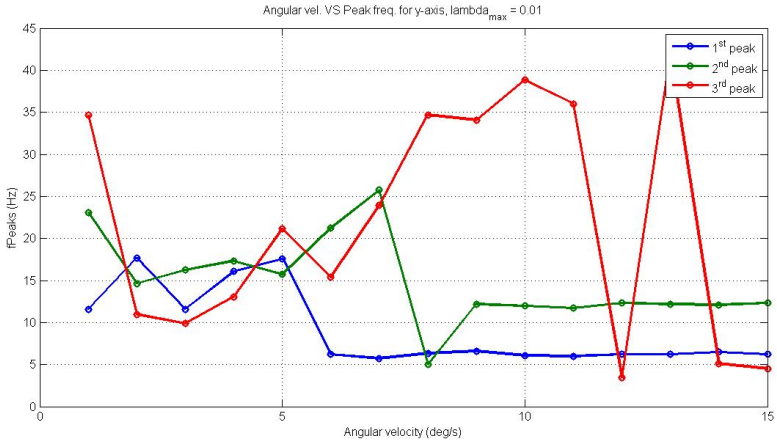


(c)

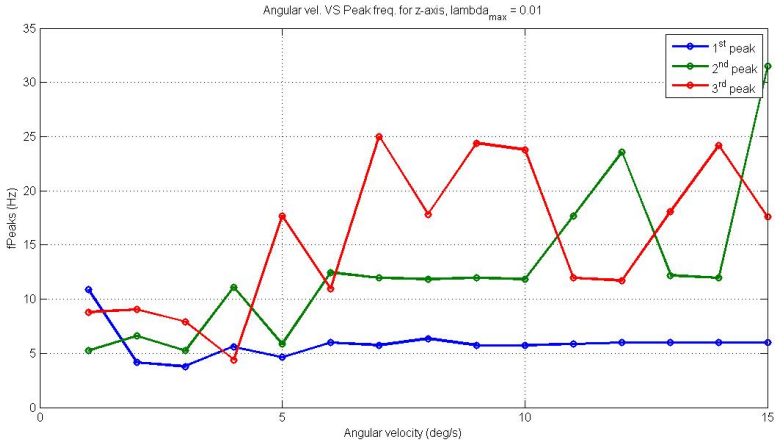
Figure 6.31: Peak frequencies of simulation results with $\lambda_{max} = 10^{-1}$ for: (a) a_x; (b) a_y; (c) a_z.



(a)

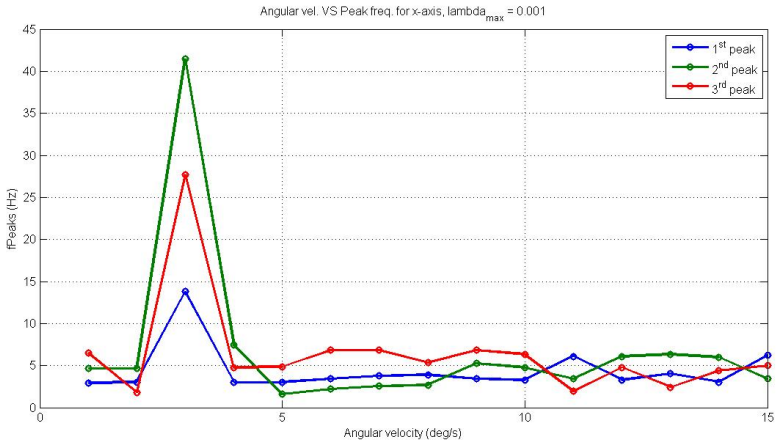


(b)

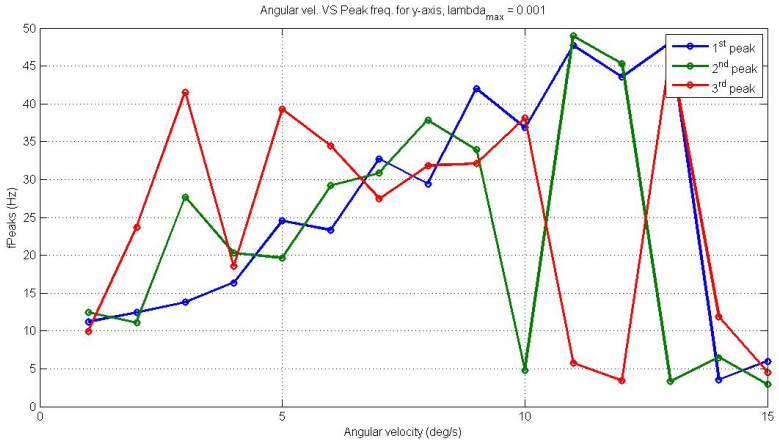


(c)

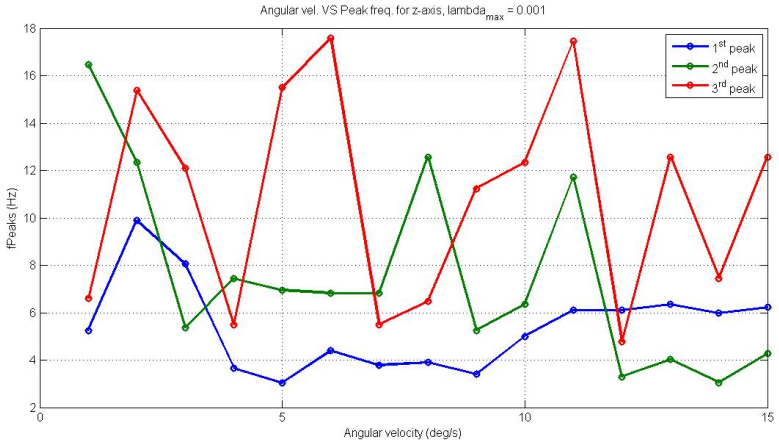
Figure 6.32: Peak frequencies of simulation results with $\lambda_{max} = 10^{-2}$ for: (a) a_x; (b) a_y; (c) a_z.



(a)

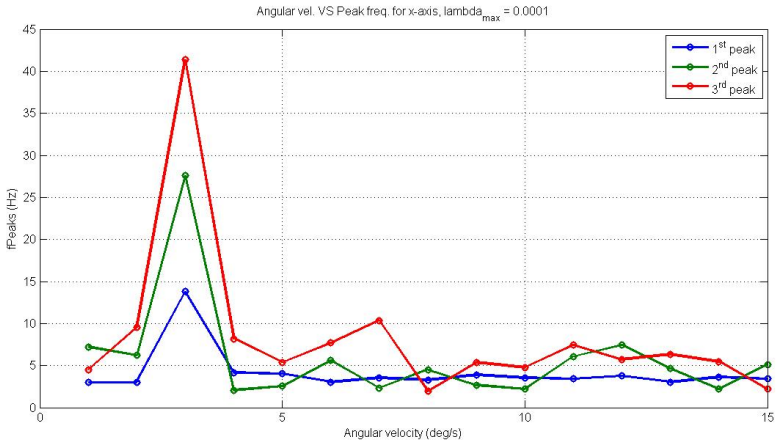


(b)

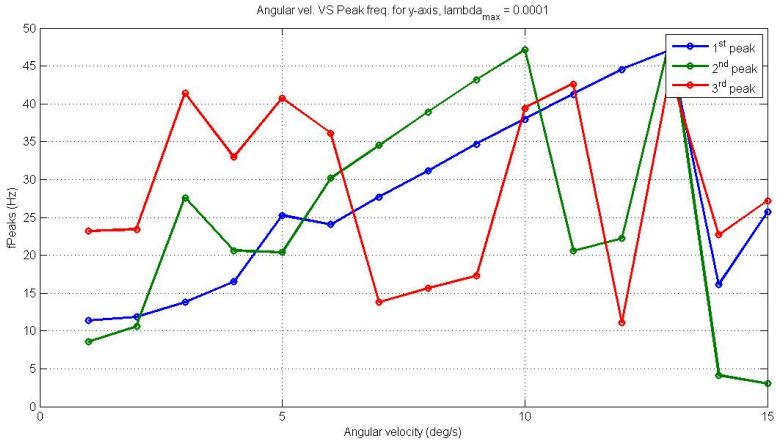


(c)

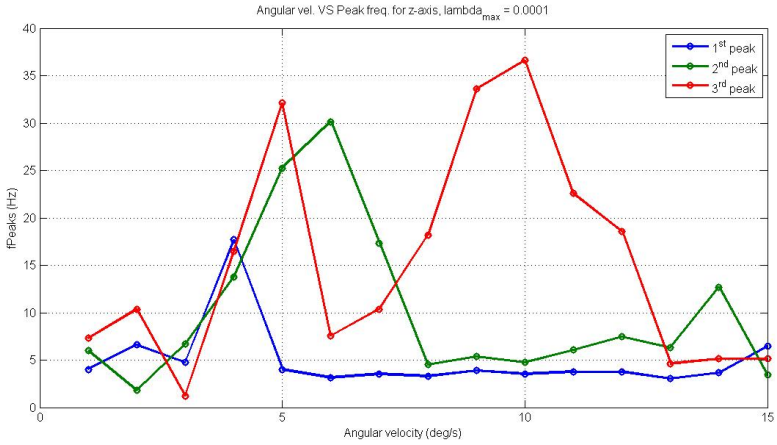
Figure 6.33: Peak frequencies of simulation results with $\lambda_{max} = 10^{-3}$ for: (a) a_x; (b) a_y; (c) a_z.



(a)

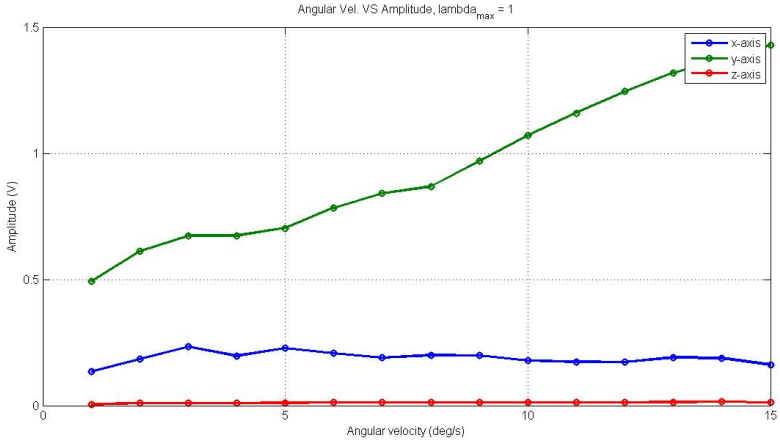


(b)

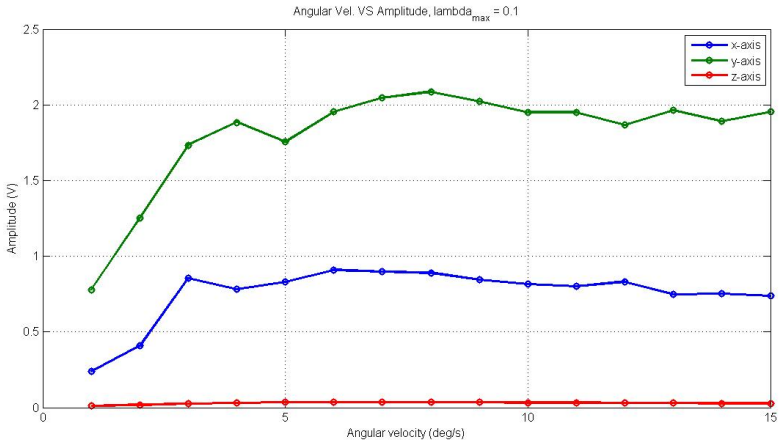


(c)

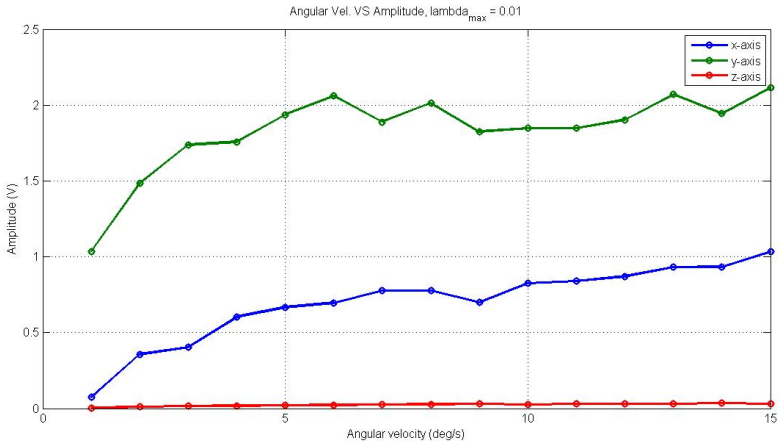
Figure 6.34: Peak frequencies of simulation results with $\lambda_{max} = 10^{-4}$ for: (a) a_x; (b) a_y; (c) a_z.



(a)

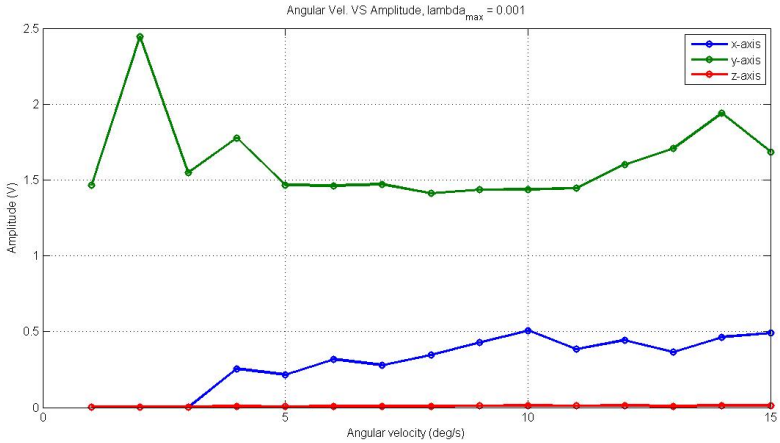


(b)

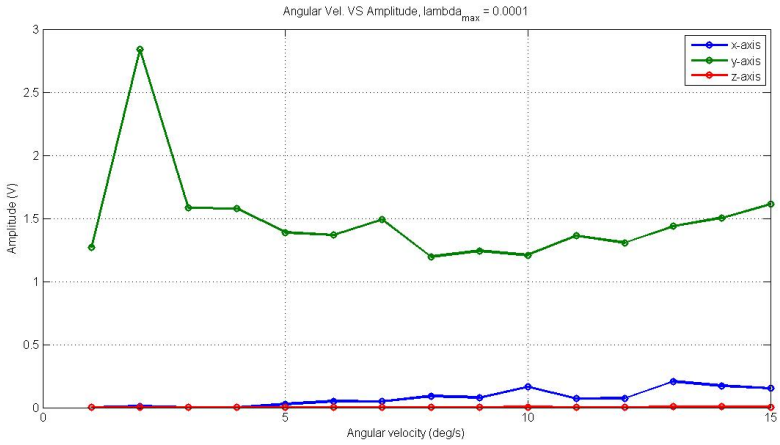


(c)

Figure 6.35: Amplitude of the simulation results with: (a) $\lambda_{max} = 1$; (b) $\lambda_{max} = 10^{-1}$; (c) $\lambda_{max} = 10^{-2}$.



(a)



(b)

Figure 6.36: Amplitude of the simulation results with: (a) $\lambda_{max} = 10^{-3}$; (b) $\lambda_{max} = 10^{-4}$.

Bibliography

[1] G. Oriolo, A. De Luca, and M. Vendittelli, "Wmr control via dynamic feedback linearization: Design, implementation, and experimental validation," *IEEE Transactions Contribution System Technology*, pp. 835–852, 2002.

[2] W. Khalil and E. Dombre, *Modelling, Identification and Control of Robots*. Kogan Page Science, 2002.

[3] C. Canudas, B. Siciliano, and G. Bastin, *Theory of Robot Control*. Springer, 1996.

[4] J. Yi, D. Song, J. Zhang, and Z. Goodwin, "Adaptive trajectory tracking control of skid-steered mobile robots," *2007 IEEE International Conference on Robotics and Automation*, pp. 2605–2610, 2007.

[5] J. Yi, D. Song, S. Jayasuriya, H. Wang, J. Zhang, and J. Liu, "Kinematic modeling and analysis of skid-steered mobile robots with applications to low-cost inertial-measurement-unit-based motion estimation," *IEEE Transactions on Robotics*, pp. 1087–1097, 2009.

[6] M. Minsky, "Telepresence," *Omni magazine*, 1980.

[7] W. Burgard, T. Panos, D. Hahnel, M. Moors, D. Schulz, H. Baltzakis, and A. Argyros, "Telepresence in populated exhibitions through web-operated mobile robots," *Mercury*, 2006.

[8] MobileRobots, *Pioneer 3-AT*. http://www.mobilerobots.com/Libraries/Downloads/Pioneer3AT-P3AT-RevA.sflb.ashx.

[9] P. Lauriano, "Intuitive control of a remote avatar using natural gestures," Master's thesis, Roger Williams University, 2011.

[10] L. Caracciolo, A. De Luca, and S. Iannitti, "Trajectory tracking control of a four-wheel differentially driven mobile robot," *EEE International Conference on Robotics and Automation*, pp. 2632–2638, 1999.

[11] K. Kozlowski and D. Pazderski, "Modeling and control of a 4-wheel skid-steering mobile robot: From theory to practice," *Technology*, 2008.

[12] K. Kozlowski, D. Pazderski, and W. Dixon, "Tracking and regulation control of a skid steering vehicle," *ANS Tenth International Topical Meeting on Robotics and Remote Systems*, pp. 369–376, 2004.

[13] VTI Technologies, *SCA620-CF8H1A 1-AXIS ANALOG ACCELEROMETERS*. http://www.muratamems.fi/sites/default/files/documents/sca620-cf8h1a_accelerometer_datasheet_82110200a.pdf.

[14] Admet, *eXpert 2600 Series Universal Testing Systems*. http://admet.com/products/universal-testing-machines/expert-2600/.

[15] M. Defoort, J. Palos, A. Kokosy, T. Floquet, and W. Perruquetti, "Performance based reactive navigation for nonholonomic mobile robots," *2009 IEEE International Conference on Robotics and Automation*, pp. 281–290, 2009.

[16] K. Weiss, "Skid-steering," 1971.

[17] H. Pacejka, *Tire and Vehicle Dynamics*. Oxford: Butterworth-Heinemann, 2002.

[18] L. Sciavicco and B. Siciliano, *Modelling and Control of Robot Manipulators*. Springer, 2007.

[19] B. Siciliano and O. Khatib, *Handbook of Robotics*. Springer, 2007.

[20] M. Frigo and S. Johnson, "The design and implementation of fftw3," *Proceedings of the IEEE*, pp. 216–231, 2005.

[21] B. Vrande, D. Campen, and A. Kraker, "An approximate analysis of dry-friction-induced stick-slip vibrations by a smoothing procedure," *Non Linear Dynamics*, pp. 155–169, 1999.

[22] B. Kuo and F. Golnaraghi, *Automatic Control Systems*. Wiley, 2009.