PONTIFICIA UNIVERSIDAD CATOLICA DE CHILE
SCHOOL OF ENGINEERING

DYNAMIC MODEL OF A SKID-STEER MOBILE MANIPULATOR USING SPATIAL VECTOR ALGEBRA AND EXPERIMENTAL VALIDATION WITH A COMPACT LOADER

SERGIO FRANCISCO AGUILERA MARINOVIC

Thesis submitted to the Office of Research and Graduate Studies
in partial fulfillment of the requirements for the degree of
Master of Science in Engineering

Advisor:
MIGUEL TORRES TORRITI

Santiago de Chile, November 2014

PONTIFICIA UNIVERSIDAD CATOLICA DE CHILE
SCHOOL OF ENGINEERING

DYNAMIC MODEL OF A SKID-STEER MOBILE MANIPULATOR USING SPATIAL VECTOR ALGEBRA AND EXPERIMENTAL VALIDATION WITH A COMPACT LOADER

SERGIO FRANCISCO AGUILERA MARINOVIC

Members of the Committee:
MIGUEL TORRES TORRITI
LUCIANO CHIANG SANCHEZ
FERNANDO AUAT CHEEIN
ALVARO SOTO ARRIAZA

Thesis submitted to the Office of Research and Graduate Studies
in partial fulfillment of the requirements for the degree of
Master of Science in Engineering

Santiago de Chile, November 2014

Gratefully to my family and fiancee

ACKNOWLEDGEMENTS

This work was supported by the National Commission for Science and Technology Research of Chile (CONICYT) under grants Fondecyt 1110343, Fondequip 120141 and CONICYT-PCHA/MagiterNacional/2014-22140151.

Special thanks to my advisor Miguel Torres, for all the time, support and guidance through my investigation and the completion of this work, as well as the learning that has given me personally and professionally.

Also thanks the members of the review committee for accepting to evaluate and contribute their comments and corrections to this work.

To my parents Sergio and Carmen for their unconditional love and support whom without them could not have been accomplished. And to my fiancee Magdalena, for her constant love, support and encouragement over the past years.

I extend my thanks to my friends in the Robotics and Automaton Laboratory (RAL) and specially to Jorge Reyes for the help on the data acquisition that validated my work.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS . . . iv

LIST OF FIGURES . . . vii

LIST OF TABLES . . . ix

ABSTRACT . . . x

RESUMEN . . . xi

1. INTRODUCTION . . . 1
   1.1. Motivation . . . 1
      1.1.1. Some examples . . . 1
   1.2. Problem Description . . . 2
   1.3. Objectives . . . 2
   1.4. Hypothesis . . . 2
   1.5. Existing Approaches . . . 2
   1.6. Summary of Contributions/Original Contributions . . . 4
   1.7. Thesis Outline . . . 5

2. MODEL OF A SKID-STEER MOBILE MANIPULATOR . . . 6
   2.1. Spatial Vector Algebra . . . 6
   2.2. General Model of a Skid-Steer Mobile Manipulator . . . 8
   2.3. Wheel-Ground Interaction . . . 18
      2.3.1. Terrain-Vehicle Interaction Forces . . . 18
      2.3.2. Wheel Contact Point Model . . . 23
   2.4. Skid-Steer Mobile Base Kinematics . . . 25
   2.5. Forward Dynamics Equations . . . 28
      2.5.1. Dynamic Model of a Floating Base . . . 29
      2.5.2. Dynamic Model of a Skid-Steer Mobile Manipulator . . . 35

3. SIMULATION . . . 48
3.1. SSMM Model Parameters . . . 48
3.2. Joints Friction, Viscosity and Control . . . 50
3.2.1. Response to step inputs . . . 50
3.2.2. Wheel torque model . . . 51
3.2.3. Arm torque model . . . 52
3.3. Simulation Flowchart . . . 52
4. SIMULATION AND EXPERIMENTAL VALIDATION . . . 54
4.1. Comparison of Experimental and Simulated Results . . . 55
4.2. Contact Points Simulation Results . . . 60
4.3. Skid-Steer Base Model Validation . . . 64
4.4. Base-Arm Interaction . . . 66
5. SPATIAL VECTOR ALGEBRA SYMBOLIC COMPUTATIONS PACKAGE . 68
6. CONCLUSION AND FUTURE RESEARCH . . . 74
6.1. Review of the Results and General Remarks . . . 74
6.2. Ongoing Research Topics . . . 75
References . . . 76
APPENDIXES . . . 82
Appendix . . . 83
APPENDIXES A. Additional resources . . . 83
A.1. Explicit Floating Base Equations . . . 83
A.2. Spatial Vector Algebra Library . . . 85
A.3. SSMM Matlab model . . . 88

LIST OF FIGURES

2.1 Skid-steer mobile manipulator. 10

2.2 Kinematic tree of the skid-steer mobile manipulator of fig. 2.1. 10

2.3 Contact point P_k and ground surface a a plane $\Pi_{x,y}$. 19

2.4 Projection of contact P_k onto the plane $\Pi_{x,y}$ along the line $L_{\perp k}$ orthogonal to $\Pi_{x,y}$ and the motion direction line L_{vk}. 21

2.5 Wheels contact points transition. 24

2.6 Skid-steer mobile base wheel dimensions. 26

2.7 Skid-steer mobile base model 26

2.8 Skid-steer mobile manipulator with external forces: gravity and ground reaction at the wheel contact points. 35

3.1 Simulated SSMM model. 50

3.2 Mobile base engine model. 51

3.3 Simulation flowchart. 53

4.1 Compact skid-steer loader at the experiment site with unloaded bucket (left) and loaded bucket (right). 54

4.2 Straight line motion experimental data compared to the simulated model with the SSMM without load. 55

4.3 On place turn motion experimental data compared to the simulated model with the SSMM without load. 56

4.4 Straight line motion experimental data compared to the simulated model with the SSMM with 400 kg load. 57

4.5 On place turn motion experimental data compared to the simulated model with the SSMM with 400 kg load. 57

4.6 Error between the simulation and the mean value of the experiments for straight line motion. 59
4.7 Error between the simulation and the mean value of the experiments for turn on place motion. 59
4.8 Evolution of the normal forces acting on the contact points of the rotating right-front wheel. 60
4.9 Evolution of the normal forces acting on the contact points of the wheels of the SSMM turning left. 61
4.10 Total normal force on the SSMM computed from the sum of the normal forces acting on all the active contact points of each of the four wheels. 62
4.11 Total normal forces FFT. 63
4.12 Longitudinal velocities of wheels. 65
4.13 Lateral velocities of the wheels. 65
4.14 Closed-loop response of the SSMM's arm when the mobile base follows a trapezoidal velocity profile. 67

LIST OF TABLES

2.1 Spatial cross product property table. 9

2.2 Geometric and inertial parameters for the SSMM. 11

2.3 Floating base children relative positions r_i. 11

2.4 Relationship between the spatial motion and force transformation matrices in terms ${}^{B}\mathbf{X}_{A}$. 13

2.5 SSMM model parameters using spatial vector algebra. 14

2.6 Motion and force variables for the SSMM. 17

2.7 Parameters and variables for the contact point model. 20

3.1 Cat® 262 Modeled Parameters in SI Units. 49

5.1 Maple implemented library. 69

ABSTRACT

The present work models the dynamics of general skid-steer mobile manipulators using the formalism and tools of the spatial vectors algebra. A unified and general model of a 6-DOF floating base with a N-DOF manipulator is proposed considering traction forces and also manipulator-vehicle and vehicle-ground interactions. This single model demonstrates the benefits of using the spatial vector algebra formulation, unlike other of the existing modeling approaches and simulation tools, thus opens the way to research on mechanically more complex robot designs and their controllers. The model built is validated using inertial measurements obtained during field tests with a compact skid-steer loader. It is to be noted that most of the existing models and simulations of mobile manipulators often consider two-wheeled differentially driven 3-DOF bases instead of skid-steering mobile bases because of the complexity of simulating wheels that skid while rolling. However, skid-steer traction is common in most of the industrial construction and mining machinery because of their simpler mechanics, high reliability, and better mobility in rough terrains. Hence, the development of physically accurate models of skid-steer manipulators is fundamental. Furthermore, a model of a 6-DOF mobile base is developed considering non-permanent contact points allowing to take into account the base interaction with the ground. The model was validated using a Cat® 262C compact-skid steer loader instead of a small mobile manipulator common in robotics research laboratory to highlight the usefulness of the presented model and the spatial vector algebra approach.

Keywords: Mobile Manipulator, Skid-Steer, Dynamic model, Experimental Validation, Spatial Vector Algebra.

RESUMEN

El trabajo presentado modela la dinámica de un manipulador móvil con base giro deslizante genérico utilizando el formalismo y herramientas del algebra de vectores espaciales. Se propone un modelo general y unificado de una base flotante de 6-DOF con un manipulador de N-DOF, considerando fuerzas de tracción, así como también la interacción manipulador-vehículo y vehículo-terreno. Este modelo demuestra los beneficios de utilizar la formulación de algebra de vectores espaciales sobre otros enfoques de modelamiento y herramientas de simulación existentes y abre el camino para la investigación de sistemas mecánicos más complejos y su control. Este modelo fue validado utilizando medidas inerciales obtenidas durante pruebas de terreno utilizando un cargador frontal con base giro deslizante. Se hace notar que la mayoría de los modelos y simulaciones existentes de manipuladores móviles en general consideran sistemas de dos ruedas con movimiento diferencial de 3-DOF en vez de modelos de bases giro deslizantes debido a la complejidad de simular ruedas que deslizan mientras giran. Sin embargo, la tracción de bases giro deslizante es la más utilizada por la maquinaria en la construcción industrial y minería debido a su sistema mecánico simple, alta confiabilidad y mejor movilidad en terrenos complicados. Es por esto que el desarrollo de un modelo físico preciso de un manipulador móvil con base giro deslizante es fundamental. Además, se desarrolló un modelo de una base móvil de 6-DOF considerando puntos de contactos no-permanentes que permiten tener tomar en consideración la interacción de la base con el terreno. Se escogió validar el modelo utilizando un cargador frontal con base giro deslizante Cat® 262C en vez de los robots pequeños, generalmente utilizados en los laboratorios de investigación para destacar la utilidad del modelo presentado y del enfoque que entrega el álgebra de vectores espaciales.

Palabras Claves: Manipulador Móvil, Base Giro Deslizante, Validación Experimental, Algebra de Vectores Espaciales.

1. INTRODUCTION

1.1. Motivation

Skid-steer mobile manipulators (SSMMs) are the integration of a robotic arm along with a skid-steer mobile base. These two elements combine the dexterity of a manipulator to interact with the environment and the mobility of the base to achieve and unlimited workspace. Furthermore, the skid-steer main advantages over other drive-mechanisms are their simpler mechanics, high reliability, and better mobility in rough terrains. The dynamics and control of manipulators have been investigated since the 70s with significant breakthroughs (Lee, 1989; Brock & Kemp, 2010), while the research concerning mobile bases dates back to the mid 90s with several of the main advances and contributions in dynamic and skidding models occurring during the last decade (Liu & Liu, 2009; Kozlowski, Krzysztof, Pazderski, & Dariusz, 2004) . Yet the integration of arms and mobile bases is still on early stages and extenive research in the dynamic modeling and control of mobile manipulators needs to be done to accomplish more difficult autonomous tasks.

1.1.1. Some examples

Mobile manipulators (MMs) applications seem attractive for both industrial (Helms, Schraft, & Hagele, 2002) and domestic (Simpkins & Simpkins, 2013) applications. In agriculture autonomous MM could be used to perform crop inspection and harvesting task (H. Tanner, Kyriakopoulos, & Krikelis, 2001; Auat Cheein & Carelli, 2013). In the aeronautics industry, coating removal or application to an airplane's fuselage (Baker, Draper, Pin, Primm, & Shekhar, 1996) is a task involving a large workspace which a single manipulator will be insufficient, but MM could easily cover. Compact loaders or load-haul-dump (LHD) can also be considered as mobile manipulators and are widely use in mining or construction sites. Thus making them autonomous would help to improve safety and productivity (Stentz, Bares, Singh, & Rowe, 1999). Domestic or indoor tasks could also benefit from mobile manipulators robots that could handle objects and solve domestic or office chores.

1.2. Problem Description

Despite the advantages of skid-steer mobile manipulator, additional complexity arises concerning the kinematic and the dynamic of the model (Kemp, Edsinger, & Torres-Jara, 2007). When a manipulator is attached to a mobile base, both bodies interact and the velocities, accelerations and forces that act on one have an impact in the other. Movements of the arm causes shifts in the robot center of mass (COM), while the interaction between the mobile base and the ground propagates to the arm. Furthermore, skidding effect inherent to skid-steer bases when turning is still under research (Yi et al., 2009; Mandow et al., 2007). Even though models that describe the behavior of these machines exist, these models generally focus on the kinematic aspects, consider planar motions and treat the vehicle-ground interaction as a permanent contact.

1.3. Objectives

The main objective is to obtain a general and unified dynamic model for skid-steer mobile manipulators that considers the base as a 6-DOF floating base, which can move freely in any direction and includes a vehicle-ground interaction forces with a non-permanent compliant contact model.

1.4. Hypothesis

A general and unified dynamic model for a SSMM robot that considers a floating base with non-permanent contact forces and the arm-base interaction dynamics can be obtained using the spatial vector algebra formulation of the recursive Newton-Euler approach proposed by (Featherstone, 2008). A model so obtained can be physically accurate and represent better the effects of the vehicle-ground interaction and base-arm interaction.

1.5. Existing Approaches

To the best of our knowledge, only the work by Liu et al. (Liu & Liu, 2009) treats the modeling of SSMMs with some detail. Liu et al. present a kinematic and dynamic model of SSMMs that takes into account traction and skidding forces, as well shifts in the COM of the robot due to changes in arm position. On the other hand, the model by Liu presents expressions in which the propagation of ground interaction forces to the arm, or the propagation of arm-accelerations back to the base are implicit. The model in (Liu & Liu, 2009) assumes the mobile base moves in a 2D plane, therefore is not a fully 6-DOF floating base with non-permanent ground contact interactions. While current work model presents explicit equations for the arm and base accelerations for a 1-DOF arm that can be extended to an N-link arm and solved in closed form provided that the computer has sufficient computational capacity. Due to the lack of works treating the modeling of SSMMs, we discuss next the existing research on mobile manipulators and skid-steer vehicles.

Different models for mobile manipulators have been developed over the last years, such as the ones proposed by (De Luca, Oriolo, & Robuffo Giordano, 2010; Tan, Xi, & Wang, 2003) that consider Ackermann steering geometry, (Q. Yu & Chen, 2002; White, Bhatt, Tang, & Krovi, 2009; Li, Ge, & Ming, 2007), which consider differential-drive schemes with caster wheels, or (Puga & Chiang, 2008), that considers a three wheeled base with differential-drive and a steering wheel. In general, these models separate the vehicle dynamics from that of the manipulator and do not consider the effects of the moving arm over the base trajectory. This is usually because the arm mass and velocity are assumed to be negligible, or because they focus on problems of redundancy resolution and trajectory planning. Thus for the physically accurate modeling of mobile manipulators it is desriable to establish unified dynamical models that simultaneously consider the coupled interaction between the base and the manipulator.

A skid-steer base was selected over other bases with Ackermann steering or differential-drive kinematics because it is a common type found in the industry, in such machines as compact loaders, LHDs and in many universities with the P3-AT and other robotics systems. The main reasons for the use of this kind of base are: i) the simpler mechanics as it only uses gears to adjust the the velocity and torque, while the Ackermann steering introduces geometric arrangements of linkages to achieve the desired angle of rotation,

ii) high-reliability and better mobility through rough terrains because all the wheel have torque.

Regarding skid-steer base models, the main difficulty is the modeling of the lateral skidding of the wheels, produced by the lateral centrifugal acceleration when turning. Some model the skid-steer as a differential drive base in which the lateral slippage may be neglected (Mandow et al., 2007), while others take into account only longitudinal slipping (Yi et al., 2009) or more detailed lateral skidding models (Kozlowski et al., 2004; Mohammadpour, Naraghi, & Gudarzi, 2010).

In general, a weakness of many of the published models is that they are only simulated for controller development purposes and very few of them experimentally validate their models. Some of these exceptions are found in (White et al., 2009; Mandow et al., 2007; W. Yu, Chuy, Collins, & Hollis, 2010), which conduct experimental verifications using small to medium size robots like like Pioneer P3-AT® used by several research groups. It is however desirable from an application perspective to validate the models also with large size and heavier industrial machinery.

1.6. Summary of Contributions/Original Contributions

The goal of the current research is to present a dynamical model for SSMMs built using spatial vector algebra that is general enough and jointly takes into account the coupled interaction between the mobile base and the manipulator, as well as the vehicle-ground interaction in a single and unified fashion. To this end, we rely on the modeling approach introduced by Featherstone using spatial vector algebra (Featherstone, 2008) for rigid body dynamics. The spatial vector formalism allows to model complex kinematic trees with compact equations and a high degree of generality. The model is built using the Spatial Toolbox for Matlab (*Spatial Vector Algebra Toolboox for Matlab*, 2014) and provides a nontrivial and enriching example that illustrates the capabilities of the modeling approach and the toolbox applied to a real world robot.

This paper also validates the model using data experimentally acquired from a compact skid-steer loader Cat® 262C Series 2, which is a good representative of similar machines employed in construction and mining. The model developed and measurements have been made publicly available at (*SSMM Model Files, Experimental Data, Simulation Videos and Spatial Vector Algebra Library*, 2014) for other researchers and students.

1.7. Thesis Outline

The thesis is organized as follows. In chapter 2 we present the dynamic modeling of a general SSMM using spatial vector algebra, is presented together with a model for contact points. The explicit equations for the SSMM direct dynamics derived step by step. chapter 3 describes the simulation details of the SSMM dynamic model developed in chapter 2. The experimental model validation methodology, as well as the comparison between the experimental and simulated results is presented chapter 4. chapter 5 presents another contribution of this thesis, which is the development of a library in maple that implements the spatial vector algebra operators and the articulated rigid body algorithm to obtain the direct dynamic equations of an articulated kinematic tree with floating base. This library can be very useful in future research about the dynamics of mechanical multibody systems because it allows to obtain explicit equations provided the complexity of the system is not beyond the computational power available to the user of the library. Finally, chapter 6 presents the conclusions of this work and discusses some aspects concerning ongoing research.

2. MODEL OF A SKID-STEER MOBILE MANIPULATOR

This chapter presents a complete and general model of the motion dynamics of skid-steer mobile manipulators. The model is derived using the spatial vector algebra formalism and the Articulated Rigib Body algorithm proposed by R. Featherstone (Featherstone, 2008) to obtain the forward dynamics equations. The model considers a skid-steer mobile base, an n-DOF manipulator, and also the ground-wheel interactions, as well as the base-manipulator interactions. First, a brief explanation about spatial vector algebra and reasons from choosing this modeling convention over others is presented.Folloginw the mathematical background of spatial vector algebra, the description of a general SSMM with an n-DOF arm using the spatial vector algebra formulation is introduced. Thirdly, a model for the ground-wheels interaction is proposed in terms of an approximated wheel with a finite number of contact points. The reaction forces acting on the contact points are also explained in detail. Fourth, even though this model considers the mobile base and manipulator as one entity, the kinematic considerations of a skid-steer mobile base are presented in order to fully understand the motion constraints that the model needs to fulfill. Finally, the articulated rigid body procesure (Featherstone, 2008) is employed to solve the direct dynamic of the mobile base without an arm and later with a 1-DOF. The equations for the floating mobile base are compared to the standard model of aircraft dynamics as a common example of a 6-DOF free floating platform approach.

2.1. Spatial Vector Algebra

The spatial vector algebra approach to modeling multibody mechanical systems offers a higher level of abstraction and more compact notation that results in fewer equations. The higher level of abstraction means among other things that the motion from one body to another is propagated by generic joint transformation, while the the accelerations due to velocity-product terms arising in rotating reference frames can be handled using the algebraic rules for spatial vector products, thus reducing error prone calculations of standard recursive Newton-Euler or Lagrange method for deriving the differential equations describing the motion dynamics of a robotic system. Other advantages of the spatial vector algebra approach over classic modeling schemes is that it provides unified framework to formulate the dynamic equations of both closed and open-loop kinematic trees, taking into account non-permanent contact points and thus allowing to model robots on floating bases, e.g. underwater vehicles, flying platforms, or ground vehicles on deformable terrains. Despite its higher level of abstraction, the spatial vector algebra formalism provides valuable insight into the dynamics and physical properties of multibody robotic systems. For example, the procedure to dervive motion equations preserves the intermediate information about the propagation of forces across joints and links, much like the recursive Newton-Euler approach. The approach also can be used to obtain the direct dynamics equations that are useful for controller design purposes. In terms of modeling efficiency, the spatial vector approach does not necessarily reduce the number of terms involved since in the end all constants, vectors, operators must be evaluated to their actual definitions in order to obtain the equations. In fact, for small serial kinematic chains with eight or less bodies, the Articulated Rigid Body algorithm using spatial vector algebra is computationally more expensive than the Composite Rigid Body Algorithm together with the recursive Newton-Euler algorithm to obtain the forward dynamic equations. However, for branched kinematic trees the advantages of the approach become more significant. An in depth discussion on the computational complexity of the Articulated Rigid Body algorithm using spatial vector algebra is found in (Featherstone, 2008).

Succinctly explained, spatial vectors are 6D vectors that describe the motion (or forces) of a rigid body using Plücker coordinates for rotation and translation (or couples and linear forces), i.e. a spatial velocity is a vector of the form $\mathbf{v} = \left[\omega^T \; v_O^T\right]^T$, where $\omega = [\omega_x \; \omega_y \; \omega_z]^T$ is the 3D vector describing the rotational velocity of the body about an axis passing through a point O, and $v_O = \left[v_{Ox} \; v_{Oy} \; v_{Oz}\right]^T$ is the 3D vector describing the velocity of a point O fixed to the body relative to some point O in space that coincides with O at a given instant. The Plücker coordinates date back to the 19^{th} century and Ball's screw theory and Von Mises' motor algebra. However, Featherstone introduced the concept of Plücker basis and defined one for a motion vector space and another for the a force vector space. This formalism combined together with a set of algebraic properties of spatial vectors that arise from the transformation rules for motion and force vectors expressed with respect to two Plücker coordinate systems A and B that can move with respect to each other provide a mathematical framework and tools that allow to express the motion of rigid bodies with fewer equations and a higher level of abstraction, but at the same time providing valuable insight into the dynamics and physical properties of multibody robotic systems. Instead of requiring two 3D equations of the form $f_O = m\dot{v}_O$ and $\tau_O = I_O\dot{\omega} + \omega \times I_O\omega$ to describe the motion of each body in the mechanism, spatial vector reduce the expressions to an equation of motion of the form $\mathbf{f} = \mathbf{I}\dot{\mathbf{v}} + \mathbf{v} \times^* \mathbf{I}\mathbf{v}$, where $\mathbf{f}$ is the spatial force vector containing the total moment and linear force acting on the body, and $\mathbf{I}$ is the spatial inertia matrix. In addition to this simplification, spatial velocity and force vectors are tightly related to the body's velocity and force vector fields, and have some nice algebraic properties. For example, the relative velocity $\mathbf{v}_{rel}$ between two bodies B_1 and B_2 with spatial velocities $\mathbf{v}_1$ and $\mathbf{v}_2$ is simply $\mathbf{v}_r = \mathbf{v}_2 - \mathbf{v}_1$; the total force on a body subject to spatial forces $\mathbf{f}_1$ and $\mathbf{f}_2$ is simply $\mathbf{f}_{tot} = \mathbf{f}_1 + \mathbf{f}_2$; similarly the total inertia of two bodies B_1 and B_2 connected together to form a composite rigid body is simply given the addition of their individual spatial inertias, i.e. $\mathbf{I}_{tot} = \mathbf{I}_1 + \mathbf{I}_2$. The spatial motion and spatial force vectors, together with their vector addition and multiplication rules define two dual vector spaces. The multiplication operation for the motion vector space is denoted by the cross product operator $\times$, while the equivalent cross product on the dual vector space is denoted by $\times^*$. The main properties connecting these two crosss products that are employed in this work are summarized in table 2.1. For a complete exposition of the spatial vector algebra and how it allows to reformulate the equation of motion of complex serial and closed-loop kinematic trees the reader is referred to (Featherstone, 2008).

2.2. General Model of a Skid-Steer Mobile Manipulator

To build the model it is convenient to first define the bodies and joints of the robot. To this end, the Cartesian coordinate frame $\mathcal{F}_0$ is first placed at a chosen fixed location in space to serve as virtual fixed base (i.e. global inertial frame) fig. 2.1. Next the mobile base

TABLE 2.1. Spatial cross product property table.

Motion Vector Product ($\times$)	Force Vector Product ($\times^*$)
$\mathbf{v}\times^* = -\mathbf{v}\times^T$	
$\mathbf{u}\times\mathbf{v} = -\mathbf{v}\times\mathbf{u}$	
$(\mathbf{X}\mathbf{v})\times = \mathbf{X}\mathbf{v}\times\mathbf{X}^{-1}$	$(\mathbf{X}\mathbf{v})\times^* = \mathbf{X}^*\mathbf{v}\times^*(\mathbf{X}^*)^{-1}$
$(\lambda\mathbf{v})\times = \lambda(\mathbf{v}\times)$	$(\lambda\mathbf{v})\times^* = \lambda(\mathbf{v}\times^*)$
$(\mathbf{u}\times\mathbf{v})\cdot = -\mathbf{v}\cdot\mathbf{u}\times^*$	$(\mathbf{u}\times^*\mathbf{f})\cdot = -\mathbf{f}\cdot\mathbf{u}\times$

is labeled as body 1 with coordinate frame $\mathcal{F}_1$. A convenient location for $\mathcal{F}_1$ is the robots center of mass. The mobile base (body 1) is treated as a body connected to the fixed base (body 0) by a six-degree-of-freedom (6-DOF) joint, i.e. the mobile base is a *floating base* allowed to move freely without any kinematic constraints save for non-permanent ground contact constraints. Attached to the mobile base are the wheels connected to the base by rotary joints. The wheels are bodies labeled 2, 3, 4 and 5 with corresponding Cartesian frames $\mathcal{F}_i$, $i = 2, 3, 4, 5$, as shown in fig. 2.1. Similarly, the robot arm is a series of bodies with coordinate frames $\mathcal{F}_i$, $i = 6, 7, 8, \ldots, N$, where N is the last body of the arm and represents also the total number of bodies of the robot. The connectivity graph for the N bodies of the robot is shown in fig. 2.2. The nodes of the graph represent each body, while the lines connecting the nodes represent the joints of the robot, such that joint i is the joint that connects body i to its parent. The SSMM is a kinematic tree, hence its connectivity graph is a topological tree. Adding more arms or wheels would add additional branches to the tree in fig. 2.2.

Having numbered the bodies and joints, the connectivity of the robot can be completely described by an array $\lambda \in \mathbb{Z}^{*N}$, such that $\lambda(i)$ (the i-th entry of the array), contains the body number of the parent of body i. From the connectivity graph of fig. 2.2 it should be clear that $\lambda = [0, 1, 1, 1, 1, 1, 6, 7, \ldots, N-1]$. The model geometric parameters are summarized in table 2.2. Parameters $a,\ b,\ c,\ d,\ e$ are common to skid-steer base model as used in (Kozlowski et al., 2004), l_i is generally used to describe manipulator length with σ the radius of the articulation, while r and w are used for wheels radius and width respectively. Additionally, our model includes the parameter h to describe the location

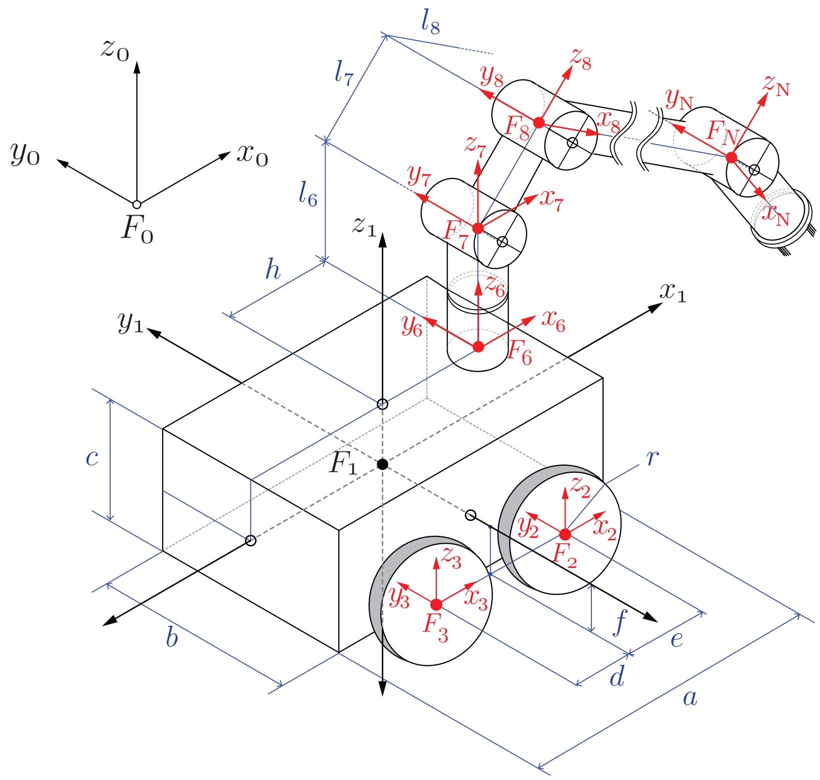


FIGURE 2.1. Skid-steer mobile manipulator.

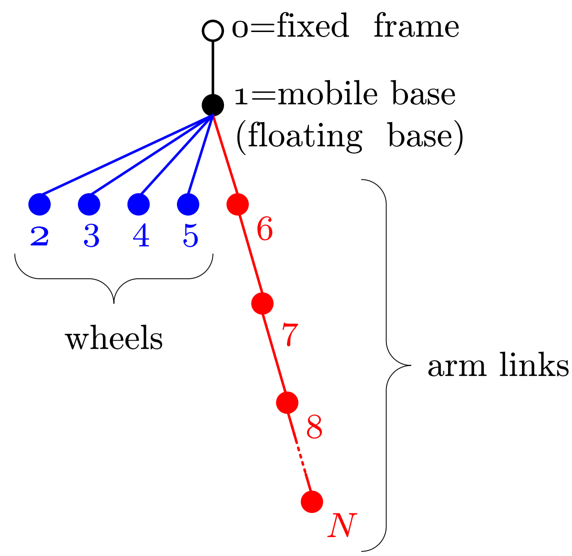


FIGURE 2.2. Kinematic tree of the skid-steer mobile manipulator of fig. 2.1.

of the manipulator along the longitudinal axis of the mobile base. The location of the children bodies $i = 2, 3, 4, 5$ corresponding to the wheels and the arm's base body $i = 6$, can be more easily described relative to the parent body using 3D position vectors $r_i = \left[r_{ix}\ r_{iy} r_{iz}\right]$, $i = 2, 3, 4, 5, 6$, in coordinates of the frame $\mathcal{F}_1$. The specific values for each body's position vector r_i are summarized in table 2.3.

TABLE 2.2. Geometric and inertial parameters for the SSMM.

Symbol	Description
Geometric parameters	
a	Base length.
b	Base width.
c	Base height.
d	Distance between the rear wheels and the base COM.
e	Distance between the front wheels and the base COM.
f	Distance between the axles plane and COM.
h	Distance the manipulator base and the base COM.
r	Wheels radius.
w	Wheels width.
l_i	Length of i-th link.
σ	Link cross-section radius.
$\mathcal{F}_0$	Cartesian coordinate inertial frame.
$\mathcal{F}_i$	Reference frame fixed to body i
Inertial parameters	
λ_i	Parent of body i.
I_i	Body i inertia matrix at body's COM.
m_i	Mass of body i.
S_i	Joint i motion subspace matrix.
g	Gravity acceleration constant.

TABLE 2.3. Floating base children relative positions r_i.

Body i	r_{ix}	r_{iy}	r_{iz}	Description
2	e	$-b/2$	$-f$	Front right wheel
3	$-d$	$-b/2$	$-f$	Rear right wheel
4	e	$b/2$	$-f$	Front left wheel
5	$-d$	$b/2$	$-f$	Rear left wheel
6	h	0	$c/2$	Manipulator first link

In addition to the connectivity of the robot bodies, to complete the geometric description of the robot it is necessary to define the geometric transformations relating the location of each joint relative to the reference frame of the body to which each joint is attached. Formally, this requires first to introduce pair of coordinate frames for each joint i that links body i to its parent $\lambda(i)$. One frame is labeled $\mathcal{F}_i$ and fixed to the body i, while the other is labeled $\mathcal{F}_{\lambda(i),i}$ is fixed to the parent body $\lambda(i)$. To minimize the number of paremeters required to describe the relative motion between $\mathcal{F}_i$ and $\mathcal{F}_{\lambda(i),i}$ it is convenient to locate the frames such that both frames coincide when the joint variables are zero, and have axes aligned following a set of rules like the widely employed Denavit and Hartenberg convention to constrain the possible frame locations (Featherstone, 2008). Other conventions than the D-H procedure to define coordinates frame are posible. In fact, the spatial vector algebra approach does not require frames to be defined according to the D-H procedure. However, using D-H procedure as part of the spatial vector algebra approach reduces the number of joint paremeters to a minimum of four with one of the parameters acting as joint variable. Noting that each body i contains a frame $\mathcal{F}_i$ and a variable number of frames $\mathcal{F}_{\lambda(j),j}$, for all j satisfying $\lambda(j) = i$ (with frames $\mathcal{F}_{\lambda(j),j}$ located at the joints of the children bodies j whose parent body is body $\lambda(j) = i$), it is convenient to select frame $\mathcal{F}_i$ as the coordinate system for body i in terms of which will be defined the spatial inertia of body i. Finally, a complete description of the robot geometry is obtained defining two transformations: $\mathbf{X}_T(i)$ and $\mathbf{X}_J(i)$. The transformation $\mathbf{X}_T(i)$ is the Plücker coordinate transform from body $\lambda(i)$ coordinates in frame $\mathcal{F}_{\lambda(i)}$ to the coordinates in frame $\mathcal{F}_{\lambda(i),i}$ located at joint i, but fixed to body $\lambda(i)$. The coordinate transform $\mathbf{X}_J(i)$ is the joint coordinate transform mapping coordinates from frame $\mathcal{F}_{\lambda(i),i}$ to coordinates in the frame $\mathcal{F}_i$ fixed to the children body i of parente body $\lambda(i)$. Therefore, with these transforms it is possible to construct the so-called *link-to-link* transform

$$ {}^{i}\mathbf{X}_{\lambda(i)} = \mathbf{X}_J(i)\mathbf{X}_T(i) \tag{2.1} $$

from coordinates in frame $\mathcal{F}_{\lambda(i)}$ of body $\lambda(i)$ to coordinates in the coordinates of frame $\mathcal{F}_i$ in body i. The complete set of transformations $\mathbf{X}_T(i)$, $i = 1, 2, \ldots, N$ describes the location of each joint-frame $\mathcal{F}_{i,j}$ relative to its corresponding body-frame $\mathcal{F}_i$ within body i. The body constant geometry data contained in $\mathbf{X}_T(i)$, $i = 1, 2, \ldots, N$, together with the variable joint coordinate transforms $\mathbf{X}_J(i)$, $i = 1, 2, \ldots, N$ and the connectivity data in the parent array λ, permit to completely describe the geometry of the robot and the position of its bodies with respect to the global reference frame $\mathcal{F}_0$. The recursive formula based on the link-to-link transformation (2.1):

$$^{i}\mathbf{X}_0 \quad = \quad {^{i}\mathbf{X}_{\lambda(i)}}\, {^{\lambda(i)}\mathbf{X}_0}, \quad \text{with } \lambda(i) \neq 0. \tag{2.2}$$

allows to compute the coordiante transform from the global reference frame $\mathcal{F}_0$ to the body coordinate frame $\mathcal{F}_i$, thus allowing to solve the forward kinematics. Equation (2.1) and (2.2) are also at the base of the recursive inverse kinematics and forward/inverse dynamics computations with algorithms whose detailed discussion can be found in (Featherstone, 2008). It should be stressed that the transforms (2.1) and (2.2) are the basic building block for the model of any mechanism, because in addition to transforming velocities and accelerations in the motion space, they can be used to implement the transformation of forces in the dual space, i.e. if $^{B}\mathbf{X}_A$ is the motion transform from coordinate frame A to coordinate frame B, then the force transform $^{B}\mathbf{X}_A^*$ can be expressed in terms of $^{B}\mathbf{X}_A$ as $^{B}\mathbf{X}_A^* = {^{B}\mathbf{X}_A^{-T}}$. A summary of the transformations, the dual relationship in terms of $^{B}\mathbf{X}_A$ and their meaning is included in table 2.4. Since all other transformations can be computed in terms of $^{B}\mathbf{X}_A$, building the model requires only to define transformations $^{B}\mathbf{X}_A$ for each body. The specific transformations required to build the SSMM model are summarized in Table 2.5 together with the joint variables.

TABLE 2.4. Relationship between the spatial motion and force transformation matrices in terms $^{B}\mathbf{X}_A$.

Transformation	Equivalent	Description
$^{B}\mathbf{X}_A$	$^{B}\mathbf{X}_A$	Motion transformation from frame A to B.
$^{A}\mathbf{X}_B$	$^{B}\mathbf{X}_A^{-1}$	Motion transformation from frame B to A.
$^{B}\mathbf{X}_A^*$	$^{B}\mathbf{X}_A^{-T}$	Force transformation from frame A to B.
$^{A}\mathbf{X}_B^*$	$^{B}\mathbf{X}_A^{T}$	Force transformation from frame B to A.

TABLE 2.5. SSMM model parameters using spatial vector algebra.

Joint-i	1	2,3,4,5	6	$7, 8, \ldots, N$
Type	Floating	Rotary $\mathbf{y}_i, i = 2, 3, 4, 5$	Rotary $\mathbf{z}_6$	Rotary $\mathbf{y}_i, i = 7, 8, \ldots$
Body-i	1	2,3,4,5	6	$7, 8, \ldots, N$
Description	Mobile Base	Wheels	First Arm Link	Remaining Arm Links
Parent Body $\lambda(i)$	0	1	1	6–$(N-1)$
Dimensions				
x	a	$2r$	ϵ	ϵ
y	b	w	ϵ	ϵ
z	c	$2r$	l_6	l_i
Joint-location Transforms $\mathbf{X}_T(i)$	$xlt([0, 0, 0])$	$xlt(r_i)$	$xlt(r_6)$	$xlt([0, 0, l_{i-1}])$
Joint Transforms $\mathbf{X}_J(i)$	$rotu(\cdot)xlt(\mathbf{E}_u^{-1}[p_x, p_y, p_z])$	$roty(q_i)$	$rotz(q_6)$	$roty(q_i)$
Mass	m_1	$m_2 = m_3 = m_4 = m_5$	m_6	m_i
COM in $\mathcal{F}_i$ coordinates	$[0, 0, 0]$	$[0, 0, 0]$	$[0, 0, l_6/2]$	$[0, 0, l_i/2]$
Inertia about COM	$\frac{m_1}{12}\begin{bmatrix} b^2+c^2 & 0 & 0 \\ 0 & a^2+c^2 & 0 \\ 0 & 0 & a^2+b^2 \end{bmatrix}$	$\frac{m_i}{12}\begin{bmatrix} 3r^2+w^2 & 0 & 0 \\ 0 & 6r^2 & 0 \\ 0 & 0 & 3r^2+w^2 \end{bmatrix}$	$\frac{m_6}{12}\begin{bmatrix} 3\sigma^2+l_6^2 & 0 & 0 \\ 0 & 3\sigma^2+l_6^2 & 0 \\ 0 & 0 & 6\sigma^2 \end{bmatrix}$	$\frac{m_i}{12}\begin{bmatrix} 3\sigma^2+l_i^2 & 0 & 0 \\ 0 & 3\sigma^2+l_i^2 & 0 \\ 0 & 0 & 6\sigma^2 \end{bmatrix}$

Summary of 3D arithmetic functions (cf. (Featherstone, 2008) for a complete list.)

$$\mathbf{E}_x(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c & s \\ 0 & -s & c \end{bmatrix}, \mathbf{E}_y(\theta) = \begin{bmatrix} c & 0 & -s \\ 0 & 1 & 0 \\ s & 0 & c \end{bmatrix}, \mathbf{E}_z(\theta) = \begin{bmatrix} c & s & 0 \\ -s & c & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{v}\times = \begin{bmatrix} 0 & -v_z & v_y \\ v_z & 0 & -v_x \\ -v_y & v_x & 0 \end{bmatrix}$$

with $c = \cos(\theta), s = \sin(\theta), \mathbf{v} = [v_x, v_y, v_z]$;

$$\mathbf{E}_u(\theta) = \begin{bmatrix} p_0^2 + p_1^2 - 1/2 & p_1p_2 + p_0p_3 & p_1p_3 - p_0p_2 \\ p_1p_2 - p_0p_3 & p_0^2 + p_2^2 - 1/2 & p_2p_3 + p_0p_1 \\ p_1p_3 + p_0p_2 & p_2p_3 - p_0p_1 & p_0^2 + p_3^2 - 1/2 \end{bmatrix} \text{(rotation about axis } u = [u_x, u_y, u_z]);$$

with Euler parameters (unit quaternion) $p_0 = \cos(\theta/2), p_1 = \sin(\theta/2)u_x$,

$p_2 = \sin(\theta/2)u_y, p_3 = \sin(\theta/2)u_z, p_0^2 + p_1^2 + p_2^2 + p_3^2 = 1.$

Summary of spatial vector arithmetic functions (cf. (Featherstone, 2008) for a complete list.)

$$xlt(\mathbf{r}) = \begin{bmatrix} \mathbf{1}_{3\times3} & \mathbf{0}_{3\times3} \\ -\mathbf{r}\times & \mathbf{1}_{3\times3} \end{bmatrix}, rot*(\theta) = \begin{bmatrix} \mathbf{E}_*(\theta) & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{E}_*(\theta) \end{bmatrix}, * = x, y, z, u$$

$${}^BX_A(\mathbf{r}, \mathbf{E}) = rot(\mathbf{E})xlt(\mathbf{r}) = \begin{bmatrix} \mathbf{E} & \mathbf{0}_{3\times3} \\ -\mathbf{E}\mathbf{r}\times & \mathbf{E} \end{bmatrix}, {}^BX_A^*(\mathbf{r}, \mathbf{E}) = rot(\mathbf{E})xlt(\mathbf{r})^{-T} = \begin{bmatrix} \mathbf{E} & -\mathbf{E}\mathbf{r}\times \\ \mathbf{0}_{3\times3} & \mathbf{E} \end{bmatrix},$$

It is possible to observe in Table 2.5 that the mobile base is a free-floating body with 6-DOF (three for orientation and three for translation) as defined by the world-to-base transform ${}^1\mathbf{X}_0 = \mathbf{X}_J(1)\mathbf{X}_T(1) = \mathbf{X}_J(1)$ because $\mathbf{X}_T(1) = \mathbb{I}_{6\times6}$ for the mobile base. However, to avoid the singularities of Euler Angles (roll, pitch, yaw), the rotations in $\mathbf{X}_J(1)$ are expressed in term of Euler Parameters (unit quaternion) involving a rotation axis (u_x, u_y, u_z)

and an angular amount θ that give rise to four unit quaternion parameters (p_0, p_1, p_2, p_3). Therefore, the description of the state of the first body involves a thirteen-dimensional vector:

$$\mathbf{x} = [\underbrace{p_0,\ p_1,\ p_2,\ p_3,}_{\substack{\text{Orientation}\\ \text{quaternion}}}\ \underbrace{p_x,\ p_y,\ p_z,}_{\substack{\text{Position}\\ \text{relative}\\ \text{to } \mathcal{F}_0}}\ \underbrace{\omega_x,\ \omega_y,\ \omega_z,}_{\substack{\text{Angular}\\ \text{velocity in}\\ \mathcal{F}_0\\ \text{coordinates}}}\ \underbrace{v_x,\ v_y,\ v_z}_{\substack{\text{Linear}\\ \text{velocity in}\\ \mathcal{F}_0\\ \text{coordinates}}}]$$

For the remaing 1-DOF joints for the wheels and the manipulator only the angular position q_i and angular velocity $\dot{q}_i$, $i = 2, 3, \ldots, N$, are required to complete the description of the state of the bodies. Thus for a four-wheeled SSMM with an M degrees of freedom manipulator, the full state vector would be given by $\mathbf{q}_{SSMM} = [\mathbf{x}|q_2\ q_3 \cdots q_{M+5}|\dot{q}_2\ \dot{q}_3 \cdots \dot{q}_{M+5}]$, requiring $13 + 4 \times 2 + M \times 2 = 13 + 2 \times (M + 4)$ joint variables. The complete list of motion and force variables is summarized in table 2.6. Throughout the thesis the notation v_i^j is employed to indicate that the vector v belongs to body i and its coordinates are expressed in the frame $\mathcal{F}_j$. When a variable of body i is expressed in the coordinates of the same body, i.e. when $j = i$, then the superscript is omitted to simplify the notation, e.g. $v_i^i = v_i$. The notation for the SSMM variables presented in table 2.6 corresponds to the typically employed notation for multibody mechanical systems, for which v_i^j and ω_i^j are the linear and angular velocities, and similarly, a_i^j, α_i^j are the linear and angular accelerations, while $q_i,\ \dot{q}_i$ and $\ddot{q}_i$ are used for the joints position, velocity and acceleration variables, respectively. Here the joint force is denoted by τ_i, which for purely rotational joints corresponds to the joint torque. To distinguish spatial vectors from common 3D vectors, the spatial vector variables are display in bold as $\mathbf{v}_i^j$, $\mathbf{a}_i^j$ and $\mathbf{f}_i^j$, while the latter employ non-bold fonts.

Considering the platform's orientation quaternion variables $p_0,\ p_1,\ p_2,\ p_3$ and position variables $p_x,\ p_y,\ p_z$, that together define the pose of the platform, the world-to-base transform ${}^1\mathbf{X}_0$ that maps spatial vectors with coordinates in the global frame $\mathcal{F}_0$ to spatial vectors with coordinates of the base frame $\mathcal{F}_1$, is given according to table 2.5 by:

$$ {}^1\mathbf{X}_0 = \begin{bmatrix} E & 0 \\ -Er\times & E \end{bmatrix}, \tag{2.3} $$

where the rotation matrix is

$$ E = \begin{bmatrix} 2\,{p_0}^2 + 2\,{p_1}^2 - 1 & 2\,p_1p_2 + 2\,p_0p_3 & 2\,p_3p_1 - 2\,p_0p_2 \\ 2\,p_1p_2 - 2\,p_0p_3 & 2\,{p_0}^2 + 2\,{p_2}^2 - 1 & 2\,p_2p_3 + 2\,p_0p_1 \\ 2\,p_3p_1 + 2\,p_0p_2 & 2\,p_2p_3 - 2\,p_0p_1 & 2\,{p_0}^2 + 2\,{p_3}^2 - 1 \end{bmatrix}, \tag{2.4} $$

and the matrix form of the cross-product is

$$ r\times = \begin{bmatrix} 0 & -p_z & p_y \\ p_z & 0 & -p_x \\ -p_y & p_x & 0 \end{bmatrix}. \tag{2.5} $$

Similarly, the link-to-link transformation that maps spatial motion vectors expressed in the coordinates of the base frame $\mathcal{F}_1$ to coordinates of the wheels or arm's inital link frames $\mathcal{F}_i$, $i = 2, 3, 4, 5, 6$, is given by

$$ {}^i\mathbf{X}_1 = xlt(r_i)roty(q_i) \tag{2.6} $$

$$ = \begin{bmatrix} \cos(q_i) & 0 & -\sin(q_i) & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 1\sin(q_i) & 0 & \cos(q_i) & 0 & 0 & 0 \\ -\sin(q_i)\,r_{iy} & \cos(q_i)\,r_{iz} + \sin(q_i)\,r_{ix} & -\cos(q_i)\,r_{iy} & \cos(q_i) & 0 & -\sin(q_i) \\ -r_{iz} & 0 & r_{ix} & 0 & 1 & 0 \\ \cos(q_i)\,r_{iy} & \sin(q_i)\,r_{iz} - \cos(q_i)\,r_{ix} & -\sin(q_i)\,r_{iy} & \sin(q_i) & 0 & \cos(q_i) \end{bmatrix}. \tag{2.7} $$

The motion transformations just defined in eqs. (2.3) and (2.7) are a basic part to build the model, together with their correspoding dual tranformations for spatial force vectors, which are computed from the motion transforms in (2.3) and (2.7).

TABLE 2.6. Motion and force variables for the SSMM.

Symbol	Definition	Description
3D space motion and force variables		
ω_i^j	$\left[\omega_{i\,x}^j\ \omega_{i\,y}^j\ \omega_{i\,z}^j\right]^T$	Angular velocity of the body i in frame $\mathcal{F}_j$.
v_i^j	$\left[v_{i\,x}^j\ v_{i\,y}^j\ v_{i\,z}^j\right]^T$	Linear velocity of the body i in frame $\mathcal{F}_j$.
α_i^j	$\left[\alpha_{i\,x}^j\ \alpha_{i\,y}^j\ \alpha_{i\,z}^j\right]^T$	Angular acceleration of the body i in frame $\mathcal{F}_j$.
a_i^j	$\left[a_{i\,x}^j\ a_{i\,y}^j\ a_{i\,z}^j\right]^T$	Linear acceleration of the body i in frame $\mathcal{F}_j$.
n_i^j	$\left[n_{i\,x}^j\ n_{i\,y}^j\ n_{i\,z}^j\right]^T$	Torque applied to the body i in frame $\mathcal{F}_j$.
f_i^j	$\left[f_{i\,x}^j\ f_{i\,y}^j\ f_{i\,z}^j\right]^T$	Force applied to the body i in frame $\mathcal{F}_j$.
q_i		Angular position of the joint i.
$\dot{q}_i$		Angular velocity of the joint i.
$\ddot{q}_i$		Angular acceleration of the joint i.
τ_i		Applied torque to the joint i.
	$[p_0\ p_1\ p_2\ p_3]^T$	Quaternion of the floating base orientation.
	$[p_x\ p_y\ p_z]^T$	Position of the floating base in $\mathcal{F}_0$.
Spatial space motion and force variables		
$\mathbf{v}_i^j$	$\begin{bmatrix}\omega_i^j \\ v_i^j\end{bmatrix}$	Spatial velocity of body i in frame $\mathcal{F}_j$.
$\mathbf{a}_i^j$	$\begin{bmatrix}\alpha_i^j \\ a_i^j\end{bmatrix}$	Spatial acceleration of body i in frame $\mathcal{F}_j$.
$\mathbf{f}_i^j$	$\begin{bmatrix}n_i^j \\ f_i^j\end{bmatrix}$	Spatial External force applied to the body i.
${}^B\mathbf{X}_A$		Frame motion transformation from body A to B.
${}^B\mathbf{X}_A^*$		Frame force transformation from body A to B.
$\mathbf{I}_i$		Spatial inertia of the body i in the frame $\mathcal{F}_i$
$\mathbf{I}_{A/B}$		Inertia propagated from body B to Body A.
$\mathbf{I}_i'$		Body i total inertia due to children bodies inertia.
$\mathbf{c}_i$		Joint i spatial acceleration due to velocity-product terms.

2.3. Wheel-Ground Interaction

The model presented in the previous section is a model for any floating base with an attached M-DOF manipulator and could be used for a ground vehicle, submarine or even a spacecraft provided that the interaction forces between the vehicle and its environment are appropriately specified. Thus to complete the dynamic model of the skid-steer mobile manipulator, the interaction between the ground and the vehicle must be defined. To this end, in addition of the ground surface geometry and its terramechanical specifications, a set of contact points (CPs) attached to the bodies that can come into contact with the ground, namely the wheels and the arm tool, must be defined together with the equations that describe the reaction forces. The next subsections explain the ground model and vehicle-ground interaction forces.

2.3.1. Terrain-Vehicle Interaction Forces

Computing the contact and collision forces between moving bodies can be computationally very expensive because accurate geomtric models of real objects can have infinite contact points, even when their geometry is relatively simple, as in the case of the contact of an ideal wheel described by the circle equation and flat ground described by the plane equation. This challenge has motivated significant research in compuational geometry algortihms to efficiently solve the intersection of bodies described by large number of geometric primitives (Mirtich & Mirtich, 1998). On the other hand, the study of different types of wheels or tracks, the modeling of ground deformation and the effect of terramechanical aspects on vehicle remains an open topic of research (Wong, 1989). However, what most of the different approaches proposed to model ground-wheel forces have in common is that contact forces consider a force decomposition into a normal and a traction force. In simple, terms the normal force is the projection of the vehicle weight onto the surface normal, while the traction force is directly related to the applied wheel torque and several complex effeccts that involve tangential and torsional restitution forces, ground internal deformation forces (Wong, 1989; Iagnemma & Dubowsky, 2004), in addition to the tangential Coloumb friction. The local deformation of bodies that are not really rigid causes the contact points to become contact areas. Despite this, a soft contact can be implemented as the contact between a point and a compliant surface in which the surface behaves as a first order massless dynamical system that includes not only the tangential Coulomb friction, but also generates damping and spring-like restitution forces that depend on the position and velocity of the contact point relative to the ground surface. For simplicity, the surface is described as a piecewise continuous concatenation of planes $\Pi_{x,y}$ with normal vector $\hat{n}_{x,y}$ and a distance to the world origin $\rho_{x,y}$ in the global coordinate frame $\mathcal{F}_0$. The ground friction, stiffness and damping coefficients that characterize the forces acting on the co ntact points are denoted by μ, K and D, respectively. Contact points (CPs) are points fixed to any of the robot's bodies of the form $P_k = [x_k \ y_k \ z_k]^T$, with coordinates expressed in the global reference frame $\mathcal{F}_0$, and with absolute velocity $v_k^0 = [v_{kx}^0 \ v_{ky}^0 \ v_{kz}^0]^T$ in $\mathcal{F}_0$ coordinates. The purpose of the CPs is to determine if a certain body is in contact with a surface and provides the location at which the contact forces act while the point is in contact. The list of parameters and variables of the contact model is presented in table 2.7.

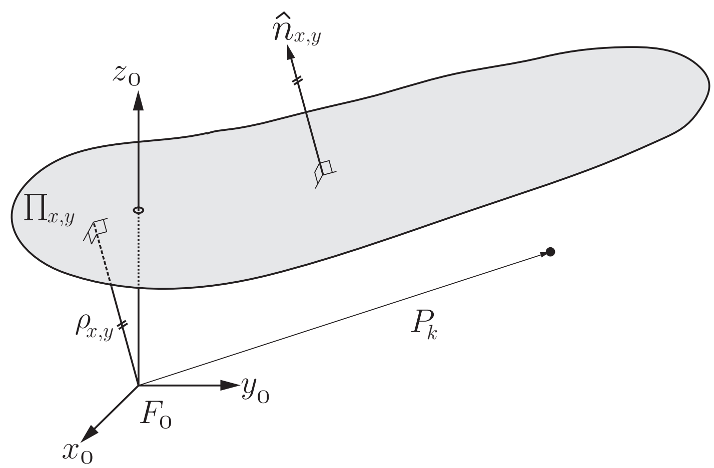


FIGURE 2.3. Contact point P_k and ground surface a a plane $\Pi_{x,y}$.

Considering the surface plane $\Pi(\hat{n}_{x,y}, \rho_{x,y})$ and CP P_k illustrated in fig. 2.3, the distance δ_k between $\Pi_{x,y}$ and P_k is given by

$$\delta_k = \hat{n}_{x,y}^T P_k - \rho_{x,y}. \tag{2.8}$$

TABLE 2.7. Parameters and variables for the contact point model.

Symbol	Description
Ground geometry and contact force parameters	
$\Pi_{x,y}$	Surface at the global position (x, y).
$\hat{n}_{x,y}$	Normal vector of the surface $\Pi_{x,y}$.
$\rho_{x.y}$	Distance between the surface $\Pi_{x,y}$ and the origin of frame $\mathcal{F}_0$.
K	Stiffness coefficient.
D	Damping coefficient.
μ	Friction coefficient.
Contact point variables	
P_k	Position of the contact point k in the global frame $\mathcal{F}_0$.
v_k^j	Linear velocity of the contact point k in frame $\mathcal{F}_j$.
$\hat{v}_k$	Normalized velocity of the contact point k in frame $\mathcal{F}_0$
δ_k	Distance between the contact point k and the surface $\Pi_{x,y}$.
$\dot{\delta}_k$	Penetration velocity of the contact point k into the surface $\Pi_{x,y}$.
ϵ_k	Projection of the displacement of the contact point k onto the tangenital plane of the surface at the point of contact.
N_k	Force applied to the contact point k normal to the surface $\Pi_{x,y}$.
f_{tk}	Force applied to the contact point k tangent to the surface $\Pi_{x,y}$.
$f_{stick\,k}$	Maximum tangent force that can be applied to the contact point k.
L_{vk}	Straight line passing through the contact point k along the direction $\hat{v}_k$.
$L_{\perp k}$	Straight line passing through the contact point k along the direction $\hat{n}_{x,y}$.

If $\delta_k > 0$, the CP is not in contact with the surface and no further analysis needs to be done. Otherwise, if $\delta_k \leq 0$, the the CP is in contact with the surface, and normal and friction forces have to be computed. In order to evaluate these forces, the velocity of the CP and two auxiliary lines are introduced as shown in fig. 2.4. The CP's velocity is a vector with magnitude and direction, while lines L_{vk} and $L_{\perp k}$ are two auxiliary lines that pass through point P_k with direction v_k^0 and $\hat{n}_{x,y}$, respectively.

The surface penetration distance δ_k and its rate of change, together with the surface stiffness coefficient K and damping coefficient D are employed to compute the ground contact normal reaction force as

$$N_k = \sqrt{-\delta_k}\left[-K\delta_k - D\dot{\delta}_k\right], \tag{2.9}$$

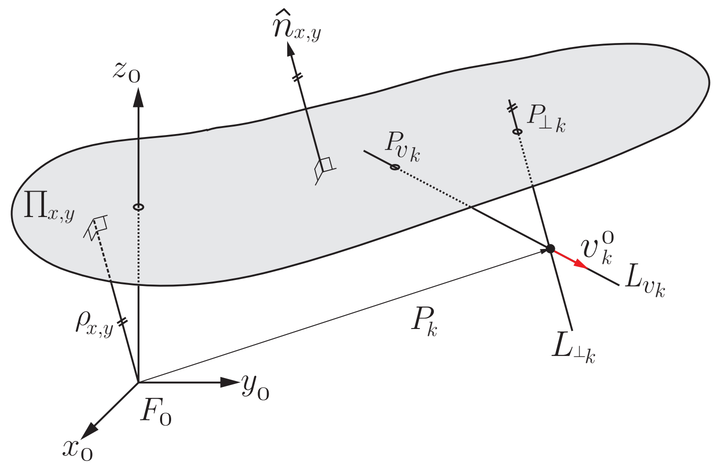


FIGURE 2.4. Projection of contact P_k onto the plane $\Pi_{x,y}$ along the line $L_{\perp k}$ orthogonal to $\Pi_{x,y}$ and the motion direction line $L_{v\,k}$.

where, by (2.8), the penetration velocity is given by

$$\begin{aligned} \tfrac{d}{dt}\left(\hat{n}_{x,y}^T P_k\right) &= \tfrac{d}{dt}\delta_k = \hat{n}_k^T \tfrac{d}{dt} P_k \\ \dot{\delta}_k &= \hat{n}_{x,y}^T v_k^0. \end{aligned} \tag{2.10}$$

Similarly, the tangential compliance of the surface produces a contact tangential reaction force $f_{t\,k}$ that satisfies a Coulomb friciton model with coefficient μ in which

$$f_{t\,k} = \begin{cases} \mu N_k, & |\mu N_k| < |f_{stick_k}| \\ f_{stick_k}, & |\mu N_k| \geq |f_{stick_k}| \end{cases}. \tag{2.11}$$

In this model, the tangential reaction force $f_{t\,k}$ is limited to a surface sticiking force f_{stick_k} for which the contact point does not slip unless it applies a force on the surface that exceeds the sticking force, i.e. as long as the contact point applies a force that is within the so-called friction cone. Analogous to the ground normal reaction force at the contact point, the tangential sticking force is given by

$$f_{stick\,k} = -K_t \epsilon_k - D_t v_{t\,k}^0 \tag{2.12}$$

where ϵ_k is the displacement of the surface from its equilibrium position in the tangent direction and $v_{t\,k}^0$ is the velocity of the surface's tangential deformation.

Two projections of the contact point P_k onto the surface $\Pi_{x,y}$ must be calculated in order to obtain the tangential displacement ϵ_k and its velocity v^0_{tk}. First, the point $P_{vk} = L_{vk}(P_k, \hat{v}_k) \cap \Pi(\hat{n}_{x,y}, \rho_{x,y})$ arising from the intersection between the surface $\Pi_{x,y}$ and a line $L_{vk}(P_k, \hat{v}_k)$ passing through the point P_k in the direction of motion of the contact point:

$$\hat{v}^0_k = \frac{v^0_k}{\|v^0_k\|}. \tag{2.13}$$

The point P_{vk} represents the location where the CP should be if it would not have penetrated the surface. The second point $P_{\perp k} = L_{\perp k}(P_k, \hat{n}_{x,y}) \cap \Pi(\hat{n}_{x,y}, \rho_{x,y})$ corresponds to the perpendicular projection of the point P_k onto the surface $\Pi_{x,y}$. Considering the equations for the plane $\Pi_{x,y}$ and projection lines L_{vk} and $L_\perp$:

$$\Pi_{x,y} : \{P \in \mathbb{R}^3 | \hat{n}^T_{x,y} P - \rho_{x,y} = 0\}, \tag{2.14}$$

$$L_{vk} : \{P \in \mathbb{R}^3 | P = \hat{v}^0_k t_v + P_k,\ \forall t_V \in \mathbb{R}\}, \tag{2.15}$$

$$L_{\perp k} : \{P \in \mathbb{R}^3 | P = \hat{n}_{x,y} t_\perp + P_k,\ \forall t_\perp \in \mathbb{R}\}, \tag{2.16}$$

substituting a point P of L_{vk} into the plane $\Pi_{x,y}$ equation yields

$$n^T_{x,y} \left[\hat{v}^0_k t_v + P_k\right] - \rho_{x,y} = 0 \ \Rightarrow\ t_v = \frac{\rho_{x,y} - \hat{n}^T_{x,y} P_k}{\hat{n}^T_{x,y} \hat{v}^0_k}, \tag{2.17}$$

thus

$$P_{vk} = \hat{v}_k \left[\frac{\rho_{x,y} - \hat{n}^T_{x,y} P_k}{\hat{n}^T_{x,y} \hat{v}_k}\right] + P_k. \tag{2.18}$$

Similarly, substituting a point P of $L_{\perp k}$ into the plane $\Pi_{x,y}$ equation yields

$$t_\perp = \frac{\rho_{x,y} - \hat{n}^T_{x,y} P_k}{\hat{n}^T_{x,y} \hat{n}_{x,y}} \tag{2.19}$$

and hence, the perpendicular projection of P_k onto plane $\Pi_{x,y}$ is given by

$$P_{\perp k} = \hat{n}_{x,y} \left[\frac{\rho_{x,y} - \hat{n}^T_{x,y} P_k}{\hat{n}^T_{x,y} \hat{n}_{x,y}}\right] + P_k \tag{2.20}$$

The difference between $P_{\perp k}$ and P_{vk} corresponds to the motion direction vector of point P_k projected onto the surface and provides information about the direction of the tangential ground deformation. Subtracting $P_{\perp k}$ from P_{vk}, yields:

$$\begin{aligned}\overline{P_{vk}P_{\perp k}} = P_{vk} - P_{\perp k} &= \hat{v}_k^0 \left[\frac{\rho_{x,y} - \hat{n}_{x,y}^T P_k}{\hat{n}_{x,y}^T \hat{v}_k^0}\right] + P_k - \hat{n}_{x,y}\left[\frac{\rho_{x,y} - \hat{n}_{x,y}^T P_k}{\hat{n}_{x,y}^T \hat{n}_{x,y}}\right] - P_k \\ &= \left[\frac{\hat{v}_k^0}{\hat{n}_{x,y}^T \hat{v}_k^0} - \frac{\hat{n}_{x,y}}{\hat{n}_{x,y}^T \hat{n}_{x,y}}\right]\left(\rho_{x,y} - \hat{n}_{x,y}^T P_k\right) \\ &= \left[\frac{\hat{v}_k^0 - \hat{n}_{x,y}\hat{n}_{x,y}^T \hat{v}_k^0}{\hat{n}_{x,y}^T \hat{v}_k^0}\right]\left(\rho_{x,y} - \hat{n}_{x,y}^T P_k\right) \end{aligned} \tag{2.21}$$

The tangential speed of the contact point P_k can now be calculated as the projection of v_k^0 onto the normalized motion direction vector $\hat{\sigma}_t = \overline{P_{vk}P_{\perp k}}/\|\overline{P_{vk}P_{\perp k}}\|$ according to

$$\|v_{tk}^0\| = \hat{\sigma}_t^T v_k^0, \text{ with } \hat{\sigma}_t = \frac{\hat{v}_k^0 - \hat{n}_{x,y}\hat{n}_{x,y}^T \hat{v}_k^0}{\|\hat{v}_k^0 - \hat{n}_{x,y}\hat{n}_{x,y}^T \hat{v}_k^0\|}. \tag{2.22}$$

Finally, the velocity $v_{tk}^0 = \|v_{tk}^0\|\hat{\sigma}_t$ of the tangential surface displacement is integrated while the CP is in contact to obtain the total tangential deformation ϵ_k at the contact point:

$$\epsilon_k = \int v_{tk}^0 dt \tag{2.23}$$

For numerical simulation, the previous integral is replaced by a summation using a simple rectangle approximation $\epsilon_k = \sum v_{tk}^0 \Delta t$.

2.3.2. Wheel Contact Point Model

As previously mentioned an ideal wheel has infinite contact points. However, for practical numerical simulation purposes the ideal circular wheel of radius r is approximated by a regular polygon inscribed in a circle of radius r with a total of N_{cp} contact points located at each one of the vertices on the wheel perimeter as shown in fig. 2.5, which for simplicity of expostion shows a wheel with $N_{cp} = 6$ in three different time instants at time $t = 0$, $t = \Delta t$ and $t = 2\Delta t$ from left to right. At each time instant the wheel has a translation and rotational velocity, and each CP has a position defined as P_k, starting from P_1 for the first point explicitly shown on each wheel and numbered in a clockwise direction.

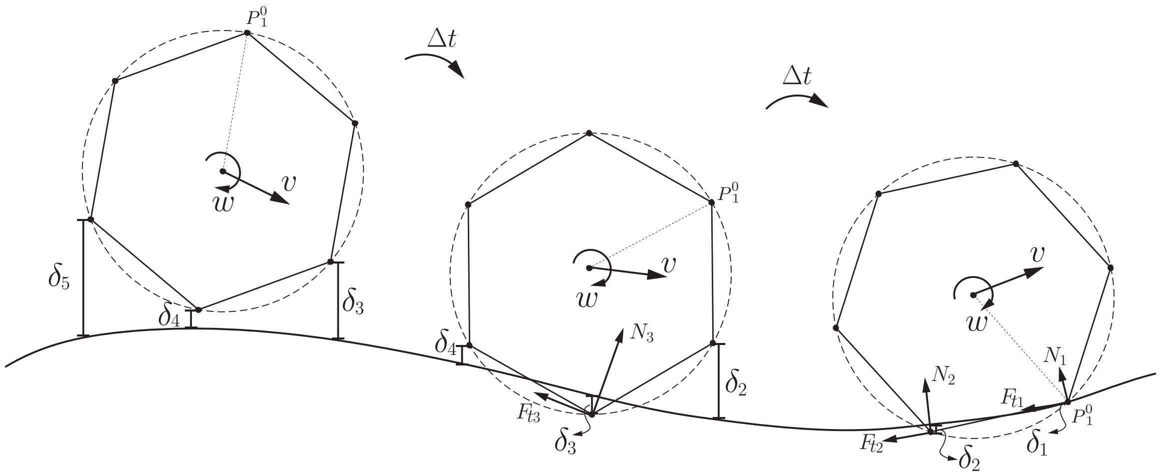


FIGURE 2.5. Wheels contact points transition.

At each time instant, the distance of the contact point to the surface is verified according to (2.8). For the first instant $t = 0$, the distances of the contact points to the ground surface satisfy $\delta_k > 0$, $k = 1, 2, \ldots, N_{cp}$, because none of the points is in contact and thus there is no reaction force of the ground acting on the wheel. At the second instant $t = \Delta t$, the wheel translates and rotates achieving a position in which the CP P_3 is in contact since $\delta_3 \leq 0$, thus the interaction between the wheel and the ground needs to be evaluated. Even though fig. 2.5 shows that the point has penetrated the surface, this situation is considered under the soft contact model as a deformation of the compliant surface due to the forces exerted on it by the wheel. According to (2.9) a normal force appears at the point P_3 as well as tangent forces that are related to the distance δ_3 and the velocity of point P_3. The linear velocity of the contact points on the terrain is not necessarily the same as the wheel's velocity. The velocity of the wheel's contact point is calculated as $v_3^0 = v - r\omega \times \hat{N}_3$, where v is the wheel's linear velocity, ω is its angular velocity, r the wheel's radius, and $\hat{N}_3$ is the unitary normal vector of the surface at the contact point (parallel to the normal force $\hat{N}_3$). In the last instant, two contact points (P_1 and P_2) are active, thus normal and tangent forces appear acting on each of them.

The accuracy of the model depends on the number of CPs employed to approximate the ideal circular wheel and the step time. To ensure a reasonable level of accuracy the step time has to be small enough so that contact point penetration of the ground is small from one simulation instant to the other. Due to the wheel approximation it is possible that several contact points in a wheel have penetrated the ground and thus several normal forces may appear, but their sum should be constant (up to numerical integration errors) when the system has no acceleration and if the surface has no slope changes. As will be shown later in section 4.2 for an SSMM on flat terrain, the total normal reaction force is equal to the weight of the system as expected.

2.4. Skid-Steer Mobile Base Kinematics

Unlike a typical differential-drive mobile base, the skid-steer mobile base (SSMB) is a slightly more complex base to model because of the skidding and slippage effects, which add some non-holonomic constraints. In this section, the main kinematic features of the skid-steer mobile base are revised focusing our attention on the 3-DOF planar kinematic motion model, before deriving a general 6-DOF floating-base dynamic model.

The main feature of the SSMB is that the applied force/torque for the left-side wheels is independent from the one applied to the right-side wheels. Thus it is possible to make the base move in a straight line if the applied torque is the same for both sides or make the base turn in-place if the applied torques have the same magnitude but opposite directions. Circular paths can also be accomplish by combining different applied torques to each of the sides.

The geometric description of the planar model for the skid-steer base is presented in Fig 2.6, which among its main features includes the longitudinal distances d and e of the wheels to the COM and the width of the base b corresponding to the distance between the wheels on each side, similar to the model proposed by (Kozlowski et al., 2004). Despite the simple geometric description, it should be sufficient to understand the kinematics of an SSMB. The kinematic model considers that each wheel is located at a point p_i, $i = 2, 3, 4, 5$, relative to the base frame. If the mobile base is turning, an instantaneous center of rotation (ICR) appears and each p_i has a corresponding velocity v_i^1 in the base frame as illustrated in fig. 2.7a and a distance d_i to the ICR as shown in fig. 2.7b. The velocities v_i^1 of each wheel can be used to compute the translation velocity v_1 of the COM in the base frame and its rotational velocity ω_1.

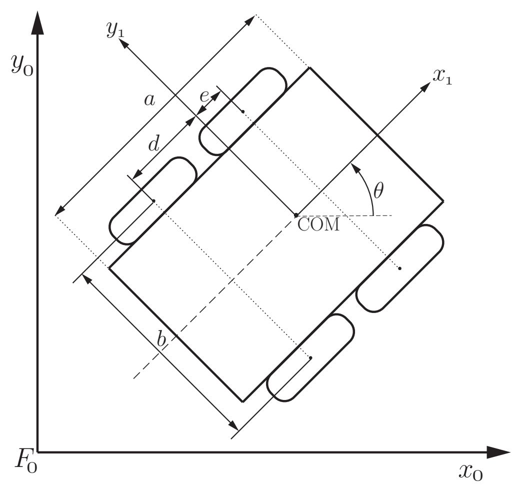
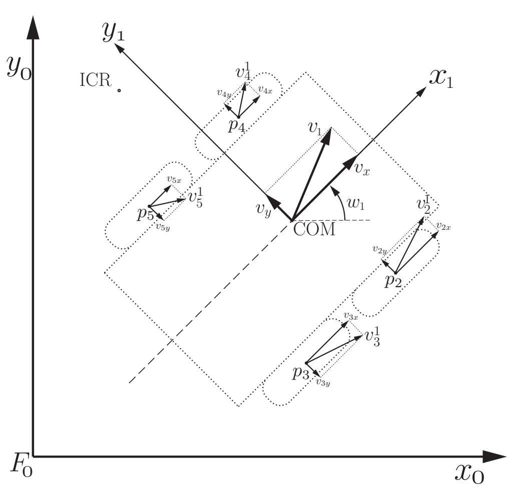
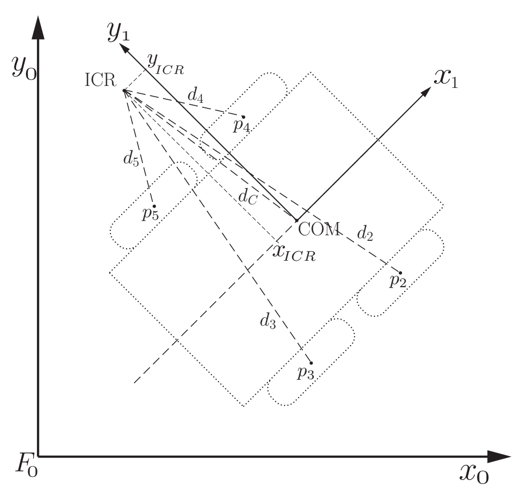


FIGURE 2.6. Skid-steer mobile base wheel dimensions.



(A) Wheel velocities.



(B) Turning geometry.

FIGURE 2.7. Skid-steer mobile base model

Since the wheels cannot separate from the base, the longitudinal velocity of the wheels on each side must be the same and thus must satisfy

$$\begin{aligned} v^1_{2x} &= v^1_{3x}, \\ v^1_{4x} &= v^1_{5x}, \end{aligned} \tag{2.24}$$

while the lateral velocity of the front wheels, as well as that of the rear wheels, has also to be the equal and satisfy

$$\begin{aligned} v^1_{2y} &= v^1_{4y}, \\ v^1_{3y} &= v^1_{5y}. \end{aligned} \tag{2.25}$$

Furthermore, since all bodies of the base rotate at the same angular speed about the ICR, considering the distances shown in fig. 2.7b between the wheels and the ICR, and that between the COM and the ICR, the following relationships must be also satisfied

$$\frac{\|v^1_i\|}{\|d_i\|} = \frac{\|v_1\|}{\|d_C\|} = \|\omega_1\|. \tag{2.26}$$

Combining (2.24), (2.25), (2.26) and the geometric dimensions of the base of fig. 2.6, the following velocities equalities are established

$$\begin{aligned} v^1_{2x} &= v^1_{3x} &&= v_{1x} - \tfrac{b}{2}\omega_1, \\ v^1_{4x} &= v^1_{5x} &&= v_{1x} + \tfrac{b}{2}\omega_1, \\ v^1_{2y} &= v^1_{5y} &&= (-x_{ICR} - d)\,\omega_1, \\ v^1_{3y} &= v^1_{4y} &&= (-x_{ICR} + e)\,\omega_1. \end{aligned} \tag{2.27}$$

These equations show the intrinsic relationship between the mobile base and how the velocities from the wheel reflect on the body and vice-versa. Finally, the nonholonomic constraint concerning the lateral velocity can be written as

$$v_{1y} + x_{ICR}\dot{\theta} = 0. \tag{2.28}$$

This constraint implies that the magnitude of the mobile base lateral velocity is directly related to the location of the ICR relative to the COM and if the ICR is not aligned with the COM along the longitudinal axes, then the base will exhibit lateral skidding.

2.5. Forward Dynamics Equations

This section presents the development of the model equations for the motion dynamics of a SSMM using the spatial vector formalism and the Articulated Body Algorithm (ABA) (Featherstone, 2008) for the calculation of the forward dynamics. Considering the complexity of the system whose state vector involves $13 + 2 \times (M + W)$ variables (13 for the base, $2 \times M$ for M arm joints and $2 \times W$ for W wheels), the dynamics for the base will be derived first without considering the wheels nor the arm. Since the base can translate and rotate in 3D space the base model corresponds to that of a floating base, whose motion is constrained later by the ground contact reaction forces. The dynamic equations of the unconstrained floating base derived using the spatial vector algebra approach and the Articulated Body Algortihm are compared to the well-known dynamic equations of a general aircraft or satellite derived using the traditional Newton-Euler force balance as an initial consistency check between both approaches. In the subsequent section an arm is added to the base, and the wheel contact forces are also included. Due to the size of the explicit equations, the arm considers only one degree of freedom. Obtaining symbolic expressions for the forward dynamics of a SSMM with an arm that has more degrees of freedoms should be possible provided the computer algebra software for symbolic computations can handle large expressions. In general this can prove to be a very difficult task, even for simpler systems as shown in previous work (Torres-Torriti & Michalska, 2005). Here we were able to compute closed-form explicit expressions only for an SSMM with a 1-DOF arm, since the spatial vector approach involves the inversion of several 6×6 joint-location and joint-tranform matrices as well as spatial 6×6 inertia matrices associated to each body. However, in the numerical application of the algorithm it is possible to apply it to more complex models, since each step evaluates matrix and vector operations that yield numercial values that do not require the complex symbolic simplification of expressions and symbol handling of the computer algebra software. Our implementation of a Maple package for the symbolic computation of the spatial vector algerbra and the ABA is later discussed in section 5.

2.5.1. Dynamic Model of a Floating Base

The dynamic model equations of a floating base are obtained in this section using Featherstone's ABA and the spatial vector approach proposed in (Featherstone, 2008). The spatial equation of motion of a body states that the net force acting on the body is equal to the change of momentum, i.e. $\mathbf{f} = \frac{d}{dt}(\mathbf{Iv}) = \mathbf{Ia} + \mathbf{v} \times^{*} \mathbf{Iv}$, where $\mathbf{v} \times^{*} \mathbf{Iv}$ is the velocity-product term that accounts for the Coriolis and centrifugal forces. Crudely stated, the body acceleration can be computed by subtracting the external forces to the velocity-product term and multiplying the difference by the inverse of the spatial inertial matrix to obtain $\mathbf{a} = -\mathbf{I}^{-1}(\mathbf{v} \times^{*} \mathbf{Iv} - \mathbf{f})$. The actual solution for a multibody system is undoubtedly more complex, but it also involves the computation of the velocity-product terms. In particular, the floating base model in the ABA implementation of Featherstone's Spatial Toolbox for Matlab (*Spatial Vector Algebra Toolboox for Matlab*, 2014, `http://royfeatherstone.org/spatial/index.html`) assumes that the spatial velocities of the floating base are expressed using coordinates referred to the global frame $\mathcal{F}_0$. However, in order to compare the resulting dynamic equations obtained for the floating base using the ABA to those of an aircraft or free flying object obtained in (Cook, 2007) using the standard Newton-Euler method, for which the force balance is typically carried out in body coordinates, it will be assumed here that the spatial velocity, acceleration and force variables of the base $\mathbf{v}_1^0$, $\mathbf{a}_1^0$ and $\mathbf{f}_1^0$ in the global frame coordinates frame have been expressed as a velocity, acceleration and force $\mathbf{v}_1^1 \equiv \mathbf{v}_1$, $\mathbf{a}_1^1 \equiv \mathbf{a}_1$ and $\mathbf{f}_1^1 \equiv \mathbf{f}_1$ in the body frame coordinates using the base-to-body transformation ${}^1X_0$ mapping coordinates from the global frame $\mathcal{F}_0$ to the base frame $\mathcal{F}_1$. Since the computation of the base-to-body transformation ${}^1X_0$ applied to $\mathbf{v}_0^1$ generates large expressions for $\mathbf{v}_1$, it will be convenient to assume that the base velocity is simply declared as a spatial velocity vector

$$\mathbf{v}_1 = \begin{bmatrix} \omega_{1x} \\ \omega_{1y} \\ \omega_{1z} \\ v_{1x} \\ v_{1y} \\ v_{1z} \end{bmatrix}, \tag{2.29}$$

whose angular and translational velocity components are referred to the floating body. For comparison, the spatial velocity of the floating base using the notation in (Cook, 2007) would have been written as $\mathbf{v}_1 = [p\ q\ r\ U\ V\ W]^T$.

The spatial inertial matrix of the floating base is defined in terms of the body inertia relative to the COM and its mass as:

$$\mathbf{I}_1 = \left[\begin{array}{c|c} I_1 & \mathbb{O}_{3\times3} \\ \hline \mathbb{O}_{3\times3} & m_1\mathbb{I}_{3\times3} \end{array}\right], \text{ with } I_1 = \begin{bmatrix} I_{xx1} & I_{xy_1} & I_{xz1} \\ I_{xy_1} & I_{yy_1} & I_{yz_1} \\ I_{xz1} & I_{yz_1} & I_{zz1} \end{bmatrix}. \tag{2.30}$$

The first step is to compute the velocity-product term that yields the Coriolis and centrifugal forces:

$$\mathbf{f}_{1c} = \mathbf{v}_1 \times^* \mathbf{I}_1 \mathbf{v}_1 = \left[n_{1cx}\, n_{1cy}\, n_{1cz}\, f_{1cx}\, f_{1cy}\, f_{1cz} \right]^T \tag{2.31}$$

$$= \begin{bmatrix} \left(-\omega_{1z} I_{xy\,1} + \omega_{1y} I_{xz\,1}\right)\omega_{1x} + \left(-\omega_{1z} I_{yy\,1} + \omega_{1y} I_{yz\,1}\right)\omega_{1y} + \left(-\omega_{1z} I_{yz\,1} + \omega_{1y} I_{zz\,1}\right)\omega_{1z} \\ \left(\omega_{1z} I_{xx\,1} - \omega_{1x} I_{xz\,1}\right)\omega_{1x} + \left(\omega_{1z} I_{xy\,1} - \omega_{1x} I_{yz\,1}\right)\omega_{1y} + \left(\omega_{1z} I_{xz\,1} - \omega_{1x} I_{zz\,1}\right)\omega_{1z} \\ \left(-\omega_{1y} I_{xx\,1} + \omega_{1x} I_{xy\,1}\right)\omega_{1x} + \left(-\omega_{1y} I_{xy\,1} + \omega_{1x} I_{yy\,1}\right)\omega_{1y} + \left(-\omega_{1y} I_{xz\,1} + \omega_{1x} I_{yz\,1}\right)\omega_{1z} \\ -\omega_{1z} m_1\, v_{1y} + \omega_{1y} m_1\, v_{1z} \\ \omega_{1z} m_1\, v_{1x} - \omega_{1x} m_1\, v_{1z} \\ -\omega_{1y} m_1\, v_{1x} + \omega_{1x} m_1\, v_{1y} \end{bmatrix}.$$

The vector $\mathbf{f}_{1ext}$ of external forces acting on the floating base 1 in body coordinates includes the gravitational force $\mathbf{f}^0_{1\,grav}$ expressed in the global frame $\mathcal{F}_0$, and other forces $\mathbf{f}_{1o}$, which can vary depending on the system. For an aerial or underwater vehicle, $\mathbf{f}_{1o}$ includes drag, lift or buoyancy and thrust forces, which are normally expressed in coordinates of the body frame $\mathcal{F}_1$, for ground vehicle, the other forces are typically ground normal and trangential traction reaction forces which are expressed in the global frame $\mathcal{F}_0$ as $\mathbf{f}^0_{1\,o}$. Therefore, before adding these forces to the platform's velocity they have to be transformed to coordinates of the body frame:

$$\mathbf{f}_{1ext} = {}^1\mathbf{X}^*_0(\mathbf{f}^0_{1\,grav} + \mathbf{f}^0_{1\,o}) = \left[n_{1extx}\, n_{1exty}\, n_{1extz}\, f_{1extx}\, f_{1exty}\, f_{1extz} \right]^T. \tag{2.32}$$

The inertial force acting on the floating body is given by

$$\mathbf{f}_1 = \mathbf{f}_{1c} - \mathbf{f}_{1ext}. \tag{2.33}$$

Hence, the inertial acceleration of the body is calculated as $\mathbf{a}_1 = -\mathbf{I}_1^{-1}\mathbf{f}_1 = \mathbf{a}_{1c} + \mathbf{a}_{1ext}$, with $\mathbf{a}_{1c} = -\mathbf{I}_1^{-1}\mathbf{f}_{1c}$ and $\mathbf{a}_{1ext} = \mathbf{I}_1^{-1}\mathbf{f}_{1ext}$. Calculating the acceleration due to the velocity-product terms yields

$$\mathbf{a}_{1c} = \begin{bmatrix} \left[\left({I_{yz\,1}}^2 - I_{zz\,1}\,I_{yy\,1}\right) n_{1x} + \left(I_{xy\,1}\,I_{zz\,1} - I_{yz\,1}\,I_{xz\,1}\right) n_{1y} + \left(I_{xz\,1}\,I_{yy\,1} - I_{xy\,1}\,I_{yz\,1}\right) n_{1z}\right] \Delta_I^{-1} \\ \left[\left(I_{xy\,1}\,I_{zz\,1} - I_{yz\,1}\,I_{xz\,1}\right) n_{1x} - \left(I_{zz\,1}\,I_{xx\,1} - {I_{xz\,1}}^2\right) n_{1y} + \left(I_{yz\,1}\,I_{xx\,1} - I_{xy\,1}\,I_{xz\,1}\right) n_{1z}\right] \Delta_I^{-1} \\ \left[\left(I_{xz\,1}\,I_{yy\,1} - I_{xy\,1}\,I_{yz\,1}\right) n_{1x} + \left(I_{yz\,1}\,I_{xx\,1} - I_{xy\,1}\,I_{xz\,1}\right) n_{1y} - \left(I_{yy\,1}\,I_{xx\,1} - {I_{xy\,1}}^2\right) n_{1z}\right] \Delta_I^{-1} \\ \omega_{1z}\,v_{1y} - \omega_{1y}\,v_{1z} \\ \omega_{1x}\,v_{1z} - \omega_{1z}\,v_{1x} \\ \omega_{1y}\,v_{1x} - \omega_{1x}\,v_{1y} \end{bmatrix} \tag{2.34}$$

$$\Delta_I = det(I) = I_{zz\,1}\,I_{xx\,1}\,I_{yy\,1} - I_{zz\,1}\,{I_{xy\,1}}^2 - {I_{xz\,1}}^2 I_{yy\,1} - {I_{yz\,1}}^2 I_{xx\,1} + 2\,I_{yz\,1}\,I_{xy\,1}\,I_{xz\,1} \tag{2.35}$$

Similarly, the acceleration due to external forces is calculated as:

$$\mathbf{a}_{1ext} = \left[\begin{array}{c|c} I_1^{-1} & \mathbb{O}_{3\times3} \\ \hline \mathbb{O}_{3\times3} & m_1^{-1}\mathbb{I}_{3\times3} \end{array}\right] \mathbf{f}_{1ext} \tag{2.36}$$

The complete expressions for the acceleration due to velocity-product terms and external forces are included in appendix A.1.

The inertial force $\mathbf{f}_1$ and inertial acceleration $\mathbf{a}_1$ computed using the spatial vector algebra approach were compared to the ones derived by the standard Newton-Euler method and determined to be equal. Model simplifications often consider the inertia off-diagonal cross-terms to be zero, i.e. $I_{xy} = I_{xz} = I_{yz} = 0$, thus reducing the expressions to

$$\mathbf{f}_1 = \begin{bmatrix} \omega_{1y} I_{zz\,1}\,\omega_{1z} - \omega_{1z} I_{yy\,1}\,\omega_{1y} \\ \omega_{1z} I_{xx\,1}\,\omega_{1x} - \omega_{1x} I_{zz\,1}\,\omega_{1z} \\ \omega_{1x} I_{yy\,1}\,\omega_{1y} - \omega_{1y} I_{xx\,1}\,\omega_{1x} \\ \omega_{1y}\,m_1\,v_{1z} - \omega_{1z}\,m_1\,v_{1y} \\ \omega_{1z}\,m_1\,v_{1x} - \omega_{1x}\,m_1\,v_{1z} \\ \omega_{1x}\,m_1\,v_{1y} - \omega_{1y}\,m_1\,v_{1x} \end{bmatrix} - \begin{bmatrix} n_{1ext x} \\ n_{1ext y} \\ n_{1ext z} \\ f_{1ext x} \\ f_{1ext y} \\ f_{1ext z} \end{bmatrix}, \tag{2.37}$$

and

$$\mathbf{a}_1 = \begin{bmatrix} \frac{\omega_{1z} I_{yy\,1}\,\omega_{1y} - \omega_{1y} I_{zz\,1}\,\omega_{1z}}{I_{xx\,1}} \\ \frac{\omega_{1x} I_{zz\,1}\,\omega_{1z} - \omega_{1z} I_{xx\,1}\,\omega_{1x}}{I_{yy\,1}} \\ \frac{\omega_{1y} I_{xx\,1}\,\omega_{1x} - \omega_{1x} I_{yy\,1}\,\omega_{1y}}{I_{zz\,1}} \\ \omega_{1z}\,v_{1y} - \omega_{1y}\,v_{1z} \\ \omega_{1x}\,v_{1z} - \omega_{1z}\,v_{1x} \\ \omega_{1y}\,v_{1x} - \omega_{1x}\,v_{1y} \end{bmatrix} + \begin{bmatrix} \frac{n_{1ext x}}{I_{xx\,1}} \\ \frac{n_{1ext y}}{I_{yy\,1}} \\ \frac{n_{1ext z}}{I_{zz\,1}} \\ \frac{f_{1ext x}}{m_1} \\ \frac{f_{1ext y}}{m_1} \\ \frac{f_{1ext z}}{m_1} \end{bmatrix}. \tag{2.38}$$

These equations correspond to those of any floating base like an aircraft, satellite, ship or ground vehicle provided that the external forces are adequately defined; see for example (Cook, 2007). In the specific case of a SSMM, the main external forces on the platform are the gravity force, the ground normal and tangential reaction forces at the contact points of the wheels on the terrain. These external forces are calculated as follows.

On each wheel acts a ground reaction force $f^1_{i\ w}$ that can be decomposed into a normal component N_i and a tangential force F_{ti}. In turn, the tangential reaction force has longitudinal and a lateral components f_{tx_i} and f_{ty_i}, respectively. If the ground under the robot is assumed to be locally flat, the ground normal at each wheel will be parallel to the $\mathbf{z}^1$-axis, while the longitudinal and lateral components will be parallel to the robot's base longitudinal and lateral axes, $\mathbf{x}^1$ and $\mathbf{y}^1$, as illustrated in figure 2.8. Hence, the ground reaction force at each wheel can be expressed as a linear force vector in the 3D space coordinates of frame $\mathcal{F}_1$:

$$f^1_{i\ w} = \begin{bmatrix} f_{tx_i} \\ f_{ty_i} \\ N_i \end{bmatrix}.$$

Thus the external spatial force will be given by

$$\mathbf{f}_{1ext} = \begin{bmatrix} n_{1extx} \\ n_{1exty} \\ n_{1extz} \\ \hline f_{1extx} \\ f_{1exty} \\ f_{1extz} \end{bmatrix} = \underbrace{\begin{bmatrix} \sum_{i=2}^{5}(r_i + r\hat{k}_1) \times f^1_{i\ w} \\ \hline \sum_{i=2}^{5} f^1_{i\ w} \end{bmatrix}}_{\mathbf{f}_{1o}} + \mathbf{f}_{1grav}, \tag{2.39}$$

where r_i, $i = 2,3,4,5$, is the position vector of each wheel relative to frame $\mathcal{F}_1$ (see table), r is the wheel radius, $\hat{k}_1 = [0\ 0\ 1]^T$ is the unit vector parallel to the $\mathbf{z}_1$-axis, and $\mathbf{f}_{1grav} = {}^1\mathbf{X}^*_0[0\ 0\ 0\ 0\ 0\ \ -m_1 g]^T$ is the gravity force.

Since the terrain model is often expressed in the global frame $\mathcal{F}_0$, for numerical simulation purposes it can be more convenient to compute the ground contact forces also in the global frame $\mathcal{F}_0$. If this is the case, the contact forces $\mathbf{f}^0_{i\ w}$, $i = 2,3,4,5$ must be transformed to forces expressed in the frame $\mathcal{F}_0$ before they can be added to the mobile base using the transformation:

$$\mathbf{f}^1_{i\ w} = {}^1\mathbf{X}^*_0\, \mathbf{f}^0_{i\ w}, \tag{2.40}$$

where ${}^j\mathbf{X}^*_0$ is the force transform from frame 0 to 1 (see (2.2) and table 2.4).

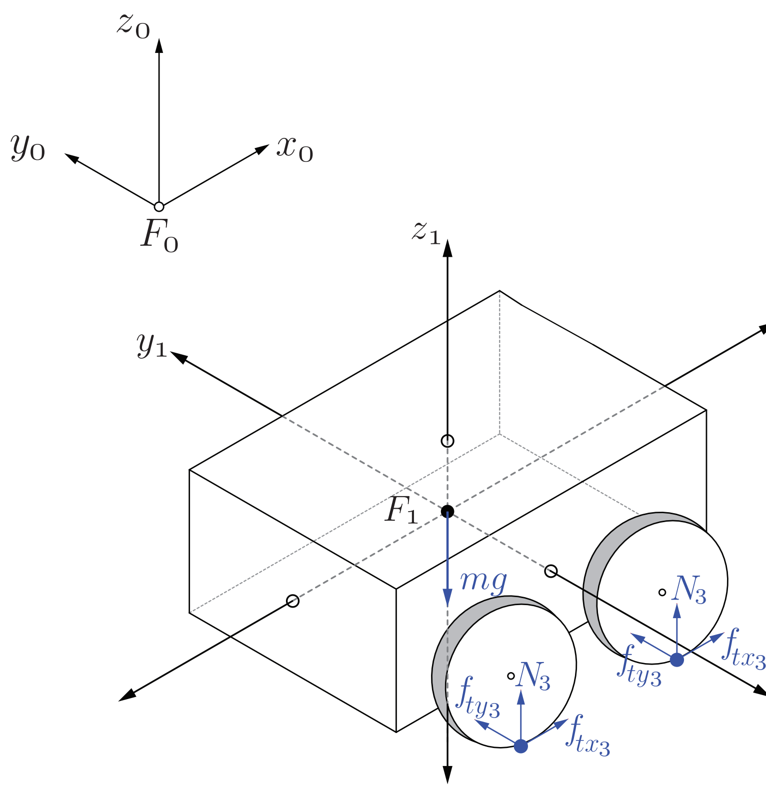


FIGURE 2.8. Skid-steer mobile manipulator with external forces: gravity and ground reaction at the wheel contact points.

2.5.2. Dynamic Model of a Skid-Steer Mobile Manipulator

The skid-steer mobile base is extended in this section with a robotic arm and its dynamic model equiations are obtained using once again the spatial vector algebra approach. Due to the large number of parameters associated to the motion and force link-to-link transformations, the model is developed here for a SSMM with only 1-DOF. The model also assumes the base inertia and mass is the lumped inertia of the body of the base and the wheels. Moreover, the off-diagonal inertia cross-terms are assumed to be zero and the inertia of the arm is considered to be non-zero only about its rotation axis. These simplifications are made in order to reduce the length of the expressions. However, a computer algebra system could be programmed to automate the development of the full equations for SSMMs with more complex arms as explained later in section 5.

Considering the simplifications mentioned, the spatial inertia matrices for the base and the arm are given by:

$$\mathbf{I}_1 = \begin{bmatrix} I_{xx1} & 0 & 0 & 0 & 0 & 0 \\ 0 & I_{yy1} & 0 & 0 & 0 & 0 \\ 0 & 0 & I_{zz1} & 0 & 0 & 0 \\ 0 & 0 & 0 & m_1 & 0 & 0 \\ 0 & 0 & 0 & 0 & m_1 & 0 \\ 0 & 0 & 0 & 0 & 0 & m_1 \end{bmatrix} \tag{2.41}$$

and

$$\mathbf{I}_6 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & I_{yy6} + \frac{1}{4}m_6 {l_6}^2 & 0 & 0 & 0 & -\frac{1}{2}m_6 l_6 \\ 0 & 0 & \frac{1}{4}m_6 {l_6}^2 & 0 & \frac{1}{2}m_6 l_6 & 0 \\ 0 & 0 & 0 & m_6 & 0 & 0 \\ 0 & 0 & \frac{1}{2}m_6 l_6 & 0 & m_6 & 0 \\ 0 & -\frac{1}{2}m_6 l_6 & 0 & 0 & 0 & m_6 \end{bmatrix}. \tag{2.42}$$

These spatial inertia matrices are calculated in terms of the body inertia I_i, the body mass m_i and the COM location $\vec{c}_i$ in body coordinates relative to the body-frame using the parameters in table 2.5 and the formula for the generalized version of the parallel axis theorem for spatial inertias:

$$\mathbf{I}_i \;=\; \begin{bmatrix} I_i + m_i \vec{c}_i \times \vec{c}_i \times^T & m_i \vec{c}_i \times \\ m_i \vec{c}_i \times^T & m_i \mathbb{I}_{3\times 3} \end{bmatrix}$$

It is to be noted that because the origin of the coordinate frame $\mathcal{F}_1$ of the floating base coincides with the COM location, i.e. $\vec{c}_1 = [0, 0, 0]$, the spatial inertia is easy to build, unlike the arm's COM, which is located at a distance $\frac{l_6}{2}$ from the arm frame $\mathcal{F}_6$, i.e. $\vec{c}_6 =$

$[0, 0, l_6/2]$, causing off-diagonal elements to appear in the expression for $\mathbf{I}_6$. In fact, under the assumption that the base is symmetric and its frame axes are aligned with the body's principal axes, the spatial inertia $\mathbf{I}_1$ is a diagonal matrix, unlike the inertia of the links of the arm $\mathbf{I}_i$, $i = 6, 5, \ldots, N$.

Just like in the case of the floating base without an arm, the velocity of the base $\mathbf{v}_1$ is declared in the body frame $\mathcal{F}_1$ according to (2.29) and an additional velocity variable $\dot{q}_6$ is introduced for the arm joint. The spatial velocity of the manipulator can now be calculated as the combined contributions of the effect of the base on the arm and the joint velocity as:

$$\mathbf{v}_6 = {}^6\mathbf{X}_1 \mathbf{v}_1 + S\dot{q}_6, \tag{2.43}$$

where ${}^6\mathbf{X}_1$ is the transformation matrix from the body frame to the manipulator frame given by

$${}^6\mathbf{X}_1 = \begin{bmatrix} \cos(q_6) & 0 & -\sin(q_6) & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ \sin(q_6) & 0 & \cos(q_6) & 0 & 0 & 0 \\ 0 & 0 & 0 & \cos(q_6) & 0 & -\sin(q_6) \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & \sin(q_6) & 0 & \cos(q_6) \end{bmatrix} \tag{2.44}$$

while S is the joint motion subspace matrix, which characterizes the motion constraint imposed by the joint. Since the model in fig. 2.1 considers that arm joints allow arms to rotate about their $\mathbf{y}$-axis, then $S_6 = [0\ 1\ 0\ 0\ 0\ 0]^T$. The velocity of the arm in the arm's frame $\mathcal{F}_6$ is thus

$$\mathbf{v}_6 = \begin{bmatrix} \cos(q_1)\omega_{1x} - \sin(q_1)\omega_{1z} \\ \omega_{1x} + \dot{q}_6 \\ \sin(q_1)\omega_{1x} + \cos(q_1)\omega_{1z} \\ \cos(q_1)v_{1x} - \sin(q_1)v_{1z} \\ v_{1y} \\ \cos(q_1)v_{1x} - \sin(q_1)v_{1z} \end{bmatrix}. \tag{2.45}$$

The force due to the velocity-product terms $\mathbf{f}_{ic} = \mathbf{v}_i \times^* \mathbf{I}_i\mathbf{v}_i$ can now be calculated using the spatial velocity and inertia of each body yielding

$$\mathbf{f}_{1c} = \begin{bmatrix} \omega_{1y} I_{zz\,1}\, \omega_{1z} - \omega_{1z} I_{yy\,1}\, \omega_{1y} \\ \omega_{1z} I_{xx\,1}\, \omega_{1x} - \omega_{1x} I_{zz\,1}\, \omega_{1z} \\ \omega_{1x} I_{yy\,1}\, \omega_{1y} - \omega_{1y} I_{xx\,1}\, \omega_{1x} \\ \omega_{1y}\, m_1\, v_{1z} - \omega_{1z}\, m_1\, v_{1y} \\ \omega_{1z}\, m_1\, v_{1x} - \omega_{1x}\, m_1\, v_{1z} \\ \omega_{1x}\, m_1\, v_{1y} - \omega_{1y}\, m_1\, v_{1x} \end{bmatrix} \tag{2.46}$$

and

$$
\mathbf{f}_{6c} = \begin{bmatrix}
\begin{array}{l}
-I_{yy6}\left(\sin(q_6)\,\omega_{1x}\omega_{1y} + \sin(q_6)\,\omega_{1x}\dot{q}_6 + \cos(q_6)\,\omega_{1z}\omega_{1y} + \cos(q_6)\,\omega_{1z}\dot{q}_6\right) \\[1ex]
- {}^1\!/_4\, m_6 l_6 \big(- l_6 \sin(q_6)\,{\omega_{1z}}^2 \cos(q_6) + 2\,\omega_{1y}\sin(q_6)\,v_{1z} + 2\,\dot{q}_6 \sin(q_6)\,v_{1z} \\
+ l_6 \cos(q_6)\,{\omega_{1x}}^2 \sin(q_6) - 2\,v_{1y}\sin(q_6)\,\omega_{1z} + 2\,v_{1y}\cos(q_6)\,\omega_{1x} \\
+ 2\,l_6\cos(q_6)^2\,\omega_{1x}\omega_{1z} - 2\,\omega_{1y}\cos(q_6)\,v_{1x} - 2\,\dot{q}_6\cos(q_6)\,v_{1x} - l_6\omega_{1x}\omega_{1z}\big) \\[1ex]
\quad - \sin(q_6)\,\omega_{1z} I_{yy6}\omega_{1y} - {}^1\!/_4\,\sin(q_6)\,\omega_{1z} m_6 {l_6}^2 \dot{q}_6 - \sin(q_6)\,\omega_{1z} I_{yy6}\dot{q}_6 \\
\quad - {}^1\!/_4\,\sin(q_6)\,\omega_{1z} m_6 {l_6}^2 \omega_{1y} + \cos(q_6)\,\omega_{1x} I_{yy6}\omega_{1y} + \cos(q_6)\,\omega_{1x} I_{yy6}\dot{q}_6 \\
\quad + {}^1\!/_4\,\cos(q_6)\,\omega_{1x} m_6 {l_6}^2 \omega_{1y} + {}^1\!/_4\,\cos(q_6)\,\omega_{1x} m_6 {l_6}^2 \dot{q}_6 - {}^1\!/_2\, m_6 l_6 \omega_{1x} v_{1z} \\
\quad + {}^1\!/_2\, m_6 l_6 \omega_{1z} v_{1x}
\end{array} \\[2ex]
\begin{array}{l}
{}^1\!/_2\, m_6 \Big(- 2\,l_6\sin(q_6)\,\omega_{1x}\cos(q_6)\,\omega_{1z} - 2\,v_{1y}\sin(q_6)\,\omega_{1x} + 2\,\omega_{1y}\sin(q_6)\,v_{1x} \\
+ 2\,\dot{q}_6\sin(q_6)\,v_{1x} - l_6\cos(q_6)^2\,{\omega_{1z}}^2 - 2\,v_{1y}\cos(q_6)\,\omega_{1z} + 2\,\dot{q}_6\cos(q_6)\,v_{1z} \\
+ 2\,\omega_{1y}\cos(q_6)\,v_{1z} - l_6{\omega_{1y}}^2 - 2\,l_6\omega_{1y}\dot{q}_6 - l_6\dot{q}_6^2 - l_6{\omega_{1x}}^2 + l_6{\omega_{1x}}^2\cos(q_6)^2\Big)
\end{array} \\[2ex]
\begin{array}{l}
{}^1\!/_2\, m_6 \Big(- l_6\sin(q_6)\,\omega_{1z}\dot{q}_6 - l_6\sin(q_6)\,\omega_{1z}\omega_{1y} + 2\,\omega_{1z}v_{1x} \\
- 2\,\omega_{1x}v_{1z} + l_6\cos(q_6)\,\omega_{1x}\omega_{1y} + l_6\cos(q_6)\,\omega_{1x}\dot{q}_6\Big)
\end{array} \\[2ex]
\begin{array}{l}
{}^1\!/_2\, m_6 \Big(- l_6\sin(q_6)\,{\omega_{1z}}^2\cos(q_6) + 2\,\omega_{1y}\sin(q_6)\,v_{1z} + 2\,\dot{q}_6\sin(q_6)\,v_{1z} \\
+ l_6\cos(q_6)\,{\omega_{1x}}^2\sin(q_6) - 2\,v_{1y}\sin(q_6)\,\omega_{1z} + 2\,v_{1y}\cos(q_6)\,\omega_{1x} \\
+ 2\,l_6\cos(q_6)^2\,\omega_{1x}\omega_{1z} - 2\,\omega_{1y}\cos(q_6)\,v_{1x} - 2\,\dot{q}_6\cos(q_6)\,v_{1x} - l_6\omega_{1x}\omega_{1z}\Big)
\end{array}
\end{bmatrix}. \tag{2.47}
$$

The expression for the force $\mathbf{f}_{6c}$ on the arm due to the velocity-product terms provides insight into the effect of the base motion on the arm force in addition to the Coriolis and centrifugal forces generated by the joint velocity $\dot{q}_6$.

The inertial force on the arm is

$$\mathbf{f}_6 = \mathbf{f}_{6c} - \mathbf{f}_{6ext} \tag{2.48}$$

where $\mathbf{f}_{6ext} = \left[n_{6extx}\; n_{6exty}\; n_{6extz}\; f_{6extx}\; f_{6exty}\; f_{6extz}\right]^T$ is the external force acting on the arm defined in a similar way to that in (2.39) including a contact and gravity force due to the payload. The inertial force on the arm can now be propagated back to the base and used to compute the inertial force on the base according to

$$\mathbf{f}_1 = \mathbf{f}_{1c} + {}^1\mathbf{X}_6^*\mathbf{f}_6 - \mathbf{f}_{1ext}. \tag{2.49}$$

where $\mathbf{f}_{1c}$ is the force due to the velocity-product terms and is $\mathbf{f}_{1ext}$ the external force on the base introduced in the previous section. If the arm has additional links, the inertial force of the i-th link has to be propagated back to its parent body $\lambda(i)$ and then to the grandparent following the kinematic tree till the base using the transformation matrix ${}^1\mathbf{X}_i^*$.

Before computing the inertial acceleration, it is necessary to compute the apparent inertia of the manipulator as seen by the base. The spatial inertia of the arm is propagated back to the base according to

$$\mathbf{I}_{1/6} = \mathbf{I}_6 - \mathbf{I}_6 S_6 \left(S_6^T \mathbf{I}_6 S_6\right)^{-1} S_6^T \mathbf{I}_6^T \tag{2.50}$$

and results in the spatial inertia matrix

$$\mathbf{I}_{1/6} = \begin{bmatrix} \Gamma\, l_6 \sin(q_6)^2 & 0 & \Gamma\, l_6 \cos(q_6) & 0 & 2\Gamma \sin(q_6) & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ \Gamma\, l_6 \cos(q_6) & 0 & \Gamma\, \cos(q_6)^2 l_6 & 0 & 2\Gamma\, \cos(q_6) & 0 \\ 0 & 0 & 0 & \frac{m_6\left(\cos(q_6)^2 m_6 {l_6}^2 + 4\, Iyy_6\right)}{4\, Iyy_6 + m_6 {l_6}^2} & 0 & -\frac{\cos(q_6) {m_6}^2 \sin(q_6) {l_6}^2}{4\, Iyy_6 + m_6 {l_6}^2} \\ 2\Gamma\, \sin(q_6) & 0 & 2\Gamma\, \cos(q_6) & 0 & m_6 & 0 \\ 0 & 0 & 0 & -\frac{\cos(q_6) {m_6}^2 \sin(q_6) {l_6}^2}{4\, Iyy_6 + m_6 {l_6}^2} & 0 & \frac{m_6\left(4\, Iyy_6 + \sin(q_6)^2 m_6 {l_6}^2\right)}{4\, Iyy_6 + m_6 {l_6}^2} \end{bmatrix} \tag{2.51}$$

where $\Gamma = \frac{1}{4}\, m_6 l_6$. Now the total spatial inertia of the floating base $\mathbf{I}_1'$ that includes the arm and the base own inertia can be calculated as

$$\mathbf{I}_1' = \mathbf{I}_1 - \mathbf{I}_{1/6}. \tag{2.52}$$

Using the inertial force on the base and the total spatial inertia of the base, the inertial acceleration of the base is simply given by

$$
\begin{aligned}
\mathbf{a}_1 &= (\mathbf{I}_1')^{-1}\mathbf{f}_1 \\
&= \left(\mathbf{I}_1'^{-1}\right) \cdot \left[\mathbf{f}_{1c} + {}^1\mathbf{X}_6^* \left(\mathbf{f}_{6c} - \mathbf{f}_{6ext}\right) - \mathbf{f}_{1ext}\right] \\
&= \underbrace{\left(\mathbf{I}_1'^{-1}\right) \mathbf{f}_{1c} + (\left(\mathbf{I}_1'^{-1}\right) {}^1\mathbf{X}_6^* \mathbf{f}_{6c}}_{\mathbf{a}_{1c}} - \underbrace{\left(\mathbf{I}_1'^{-1}\right) {}^1\mathbf{X}_6^* \mathbf{f}_{6ext}}_{\mathbf{a}_{1/6ext}} - \underbrace{\left(\mathbf{I}_1'^{-1}\right) \mathbf{f}_{1ext}}_{\mathbf{a}_{1ext}}
\end{aligned}
$$

The inertial acceleration of the base $\mathbf{a}_1$ can be split into an acceleration arising from the velocity-product terms $\mathbf{a}_{1c}$, and accelerations due to the external forces on the arm and the base $\mathbf{a}_{1/6ext}$ and $\mathbf{a}_{1ext}$, respectively. The angular and linear components of the acceleration due to the velocity-product terms about/along the axes $\mathbf{x}$, $\mathbf{y}$ and $\mathbf{z}$ of the base coordinate frame $\mathcal{F}_1$ are:

$$
\begin{aligned}
\alpha_{1cx} = \Big(& \cos(q_6)^2 {l_6}^2 m_1 m_6 \omega_{1z} I_{yy6} \omega_{1y} + \cos(q_6)^2 {l_6}^2 m_1 m_6 \omega_{1z} I_{yy6} \dot{q}_6 \\
& -2\, m_1 I_{zz1} m_6 {l_6}^2 \cos(q_6)^2 \omega_{1z} \dot{q}_6 - 2\, \omega_{1z} \omega_{1y} \cos(q_6)^2 m_1 I_{zz1} m_6 {l_6}^2 \\
& + \cos(q_6)^2 m_1 m_6 {l_6}^2 \omega_{1z} \omega_{1y} I_{yy1} - m_1 m_6 {l_6}^2 \cos(q_6) \sin(q_6) \omega_{1y} \omega_{1x} I_{xx1} \\
& + \cos(q_6) {l_6}^2 m_6 m_1 \sin(q_6) \omega_{1x} I_{yy6} \omega_{1y} + \cos(q_6) {l_6}^2 m_6 \sin(q_6) m_1 \omega_{1x} I_{yy6} \dot{q}_6 \\
& - \cos(q_6) {l_6}^2 m_6 m_1 I_{zz1} \sin(q_6) \omega_{1x} \omega_{1y} - 2\, \cos(q_6) {l_6}^2 m_6 m_1 I_{zz1} \sin(q_6) \omega_{1x} \dot{q}_6 \\
& + \cos(q_6) {l_6}^2 m_6 m_1 \sin(q_6) \omega_{1y} \omega_{1x} I_{yy1} + m_1 I_{zz1} \omega_{1z} m_6 {l_6}^2 \omega_{1y} \\
& +2\, m_1 I_{zz1} \omega_{1z} m_6 {l_6}^2 \dot{q}_6 + 4\, I_{zz1} m_6 \omega_{1z} I_{yy6} \omega_{1y} + 4\, I_{zz1} m_6 \omega_{1z} I_{yy6} \dot{q}_6 \\
& -4\, \omega_{1z} \omega_{1y} {I_{zz1}}^2 m_6 + 4\, \omega_{1z} \omega_{1y} I_{zz1} m_6 I_{yy1} - 4\, \omega_{1z} \omega_{1y} m_1 {I_{zz1}}^2 \\
& +4\, \omega_{1z} \omega_{1y} m_1 I_{zz1} I_{yy1} + 4\, m_1 I_{zz1} \omega_{1z} I_{yy6} \omega_{1y} + 4\, m_1 I_{zz1} \omega_{1z} I_{yy6} \dot{q}_6 \Big) \Big/ \\
\Big(& - \cos(q_6)^2 m_1 I_{zz1} m_6 {l_6}^2 + m_1 \cos(q_6)^2 m_6 {l_6}^2 I_{xx1} \\
& + m_1 I_{zz1} m_6 {l_6}^2 + 4\, I_{zz1} I_{xx1} m_6 + 4\, m_1 I_{xx1} I_{zz1} \Big)
\end{aligned}
$$

$$
\alpha_{1cy} = -\left(\omega_{1z} I_{xx1} \omega_{1x} - \omega_{1x} I_{zz1} \omega_{1z} + \tau_1\right) \Big/ \left(I_{yy1}\right)
$$

$$
\begin{aligned}
\alpha_{1cz} \quad = \quad & \Big(-2\,m_1 I_{xx1} \cos(q_6)\sin(q_6)\,\omega_{1z} m_6 {l_6}^2 \dot{q}_6 - m_1 I_{xx1}\cos(q_6)\sin(q_6)\,\omega_{1z} m_6 {l_6}^2 \omega_{1y} \\
& + m_1 m_6 {l_6}^2 \cos(q_6)\sin(q_6)\,\omega_{1z}\omega_{1y} I_{yy1} - m_1 m_6 {l_6}^2 \cos(q_6)\sin(q_6)\,\omega_{1z}\omega_{1y} I_{zz1} \\
& + \sin(q_6)\, m_1 m_6 {l_6}^2 \cos(q_6)\,\omega_{1z} I_{yy6}\dot{q}_6 + \sin(q_6)\, m_1 m_6 {l_6}^2 \cos(q_6)\,\omega_{1z} I_{yy6}\omega_{1y} \\
& + 4\,m_1 I_{xx1}\omega_{1x} I_{yy6}\omega_{1y} + 4\,m_1 I_{xx1}\omega_{1x} I_{yy6}\dot{q}_6 + m_1 m_6 {l_6}^2 \omega_{1x} I_{yy6}\omega_{1y} \\
& + m_1 m_6 {l_6}^2 \omega_{1x} I_{yy6}\dot{q}_6 + 2\,\omega_{1y}\omega_{1x} m_1 \cos(q_6)^2 m_6 {l_6}^2 I_{xx1} \\
& + 2\,m_1 I_{xx1}\cos(q_6)^2\,\omega_{1x} m_6 {l_6}^2 \dot{q}_6 - \omega_{1y}\omega_{1x} m_1 \cos(q_6)^2 m_6 {l_6}^2 I_{yy1} \\
& - 4\,\omega_{1y}\omega_{1x} m_1 {I_{xx1}}^2 - 4\,\omega_{1y}\omega_{1x} m_6 {I_{xx1}}^2 + 4\,\omega_{1y}\omega_{1x} m_1 I_{xx1} I_{yy1} \\
& - \omega_{1y}\omega_{1x} m_1 m_6 {l_6}^2 I_{xx1} + \omega_{1y}\omega_{1x} m_1 m_6 {l_6}^2 I_{yy1} + 4\,\omega_{1y}\omega_{1x} m_6 I_{xx1} I_{yy1} \\
& + 4\,m_6 I_{xx1}\omega_{1x} I_{yy6}\omega_{1y} + 4\,m_6 I_{xx1}\omega_{1x} I_{yy6}\dot{q}_6 - m_1 m_6 {l_6}^2 \cos(q_6)^2\,\omega_{1x} I_{yy6}\omega_{1y} \\
& - m_1 m_6 {l_6}^2 \cos(q_6)^2\,\omega_{1x} I_{yy6}\dot{q}_6 \Big) \Big/ \Big(\cos(q_6)^2\, m_1 I_{zz1} m_6 {l_6}^2 \\
& - m_1 \cos(q_6)^2\, m_6 {l_6}^2 I_{xx1} - m_1 I_{zz1} m_6 {l_6}^2 - 4\,I_{zz1} I_{xx1} m_6 - 4\,m_1 I_{xx1} I_{zz1} \Big)
\end{aligned}
$$

$$
\begin{aligned}
a_{1cx} \quad = \quad & -1/2 \Big(8\,I_{yy6} {m_1}^2 \omega_{1y} v_{1z} - 16\,I_{yy6} m_6 m_1 \omega_{1z} v_{1y} - 8\,I_{yy6} {m_6}^2 v_{1y}\omega_{1z} \\
& + 8\,I_{yy6} {m_6}^2 \omega_{1y} v_{1z} - 8\,I_{yy6} {m_1}^2 \omega_{1z} v_{1y} + 2\,{m_1}^2 m_6 {l_6}^2 \omega_{1y} v_{1z} \\
& + 2\,{m_6}^2 {l_6}^2 m_1 \omega_{1y} v_{1z} - 2\,{m_1}^2 m_6 {l_6}^2 \omega_{1z} v_{1y} + 16\,I_{yy6} m_6 m_1 \omega_{1y} v_{1z} \\
& - 2\,{m_6}^2 {l_6}^2 m_1 \omega_{1z} v_{1y} - 4\,\cos(q_6)\, I_{yy6} l_6 {m_6}^2 {\omega_{1z}}^2 - 4\,l_6 \sin(q_6)\,\tau_1 {m_6}^2 \\
& - 4\,\cos(q_6)\, I_{yy6} l_6 {m_6}^2 {\omega_{1y}}^2 - 8\,\cos(q_6)\, I_{yy6} l_6 {m_6}^2 \omega_{1y}\dot{q}_6 - 4\,\cos(q_6)\, I_{yy6} l_6 {m_6}^2 \dot{q}_6^2 \\
& - 4\,\cos(q_6)\, I_{yy6} l_6 m_6 m_1 \dot{q}_6^2 - 4\,\cos(q_6)\, I_{yy6} l_6 m_6 m_1 {\omega_{1z}}^2 - \cos(q_6)\, {l_6}^3 {m_6}^2 m_1 {\omega_{1x}}^2 \\
& - 4\,l_6 \sin(q_6)\, m_1 \tau_1 m_6 - 4\,I_{yy6} l_6 \sin(q_6)\, m_1 \omega_{1z}\omega_{1x} m_6 + \cos(q_6)^3\, {l_6}^3 {m_6}^2 m_1 {\omega_{1x}}^2 \\
& - 2\,\sin(q_6)\, m_1 \omega_{1z}\omega_{1x} {m_6}^2 {l_6}^3 \cos(q_6)^2 - 4\,\cos(q_6)\, I_{yy6} l_6 m_6 m_1 {\omega_{1y}}^2 \\
& - 8\,\cos(q_6)\, I_{yy6} l_6 m_6 m_1 \omega_{1y}\dot{q}_6 - 2\,\cos(q_6)\, {l_6}^3 {m_6}^2 m_1 \omega_{1y}\dot{q}_6 - \cos(q_6)\, {l_6}^3 {m_6}^2 m_1 \dot{q}_6^2 \\
& - \cos(q_6)\, {l_6}^3 {m_6}^2 m_1 {\omega_{1y}}^2 - 4\,I_{yy6} l_6 \sin(q_6)\,\omega_{1z}\omega_{1x} {m_6}^2 \\
& - \cos(q_6)^3\, {l_6}^3 {m_6}^2 m_1 {\omega_{1z}}^2 \Big) \Big/ \Big(4\,I_{yy6} {m_1}^2 + 4\,I_{yy6} {m_6}^2 + 8\,I_{yy6} m_1 m_6 \\
& + m_1 {m_6}^2 {l_6}^2 + {m_1}^2 m_6 {l_6}^2 \Big)
\end{aligned}
$$

$$
\begin{aligned}
a_{1cy} \quad = \quad & \Big(-2\, I_{zz1} m_6 l_6 \sin(q_6)\, \omega_{1z} I_{yy6} \omega_{1y} + 4\, I_{zz1} I_{xx1} m_6 l_6 \sin(q_6)\, \omega_{1z} \dot{q}_6 \\
& -2\, I_{zz1} m_6 l_6 \sin(q_6)\, \omega_{1z} \omega_{1y} I_{yy1} + 2\, I_{zz1} I_{xx1} m_6 l_6 \sin(q_6)\, \omega_{1z} \omega_{1y} \\
& +2\, {I_{zz1}}^2 m_6 l_6 \sin(q_6)\, \omega_{1z} \omega_{1y} - 2\, I_{zz1} m_6 l_6 \sin(q_6)\, \omega_{1z} I_{yy6} \dot{q}_6 \\
& +4\, m_1 I_{zz1} I_{xx1} \omega_{1x} v_{1z} - 4\, m_1 I_{zz1} I_{xx1} \omega_{1z} v_{1x} + 2\, \cos(q_6)\, m_6 l_6 I_{xx1} \omega_{1x} I_{yy6} \dot{q}_6 \\
& +2\, \cos(q_6)\, m_6 l_6 I_{xx1} \omega_{1x} I_{yy6} \omega_{1y} - 2\, \cos(q_6)\, m_6 l_6 {I_{xx1}}^2 \omega_{1y} \omega_{1x} \\
& -m_1 I_{zz1} m_6 {l_6}^2 \omega_{1z} v_{1x} + m_1 I_{zz1} m_6 {l_6}^2 \omega_{1x} v_{1z} - 4\, I_{zz1} I_{xx1} m_6 v_{1x} \omega_{1z} \\
& +4\, I_{zz1} I_{xx1} m_6 \omega_{1x} v_{1z} - m_1 \cos(q_6)^2\, m_6 {l_6}^2 I_{xx1} \omega_{1z} v_{1x} \\
& +m_1 I_{zz1} \cos(q_6)^2\, m_6 {l_6}^2 \omega_{1z} v_{1x} + m_1 \cos(q_6)^2\, m_6 {l_6}^2 I_{xx1} \omega_{1x} v_{1z} \\
& +2\, \cos(q_6)\, m_6 l_6 I_{xx1} \omega_{1y} \omega_{1x} I_{yy1} - m_1 I_{zz1} (\cos(q_6))^2\, m_6 {l_6}^2 \omega_{1x} v_{1z} \\
& -2\, I_{zz1} I_{xx1} m_6 l_6 \cos(q_6)\, \omega_{1x} \omega_{1y} - 4\, I_{zz1} I_{xx1} m_6 l_6 \cos(q_6)\, \omega_{1x} \dot{q}_6 \Big) \Big/ \\
& \Big(-\cos(q_6)^2\, m_1 I_{zz1} m_6 {l_6}^2 + m_1 \cos(q_6)^2\, m_6 {l_6}^2 I_{xx1} \\
& +m_1 I_{zz1} m_6 {l_6}^2 + 4\, I_{zz1} I_{xx1} m_6 + 4\, m_1 I_{xx1} I_{zz1} \Big)
\end{aligned}
$$

$$
\begin{aligned}
a_{1cz} \quad = \quad & 1/2 \Big(m_1 {m_6}^2 {l_6}^3 \cos(q_6)^2\, {\omega_{1x}}^2 \sin(q_6) - 2\, m_1 {m_6}^2 {l_6}^3 \omega_{1x} \cos(q_6)\, \omega_{1z} \\
& +4\, m_1 \cos(q_6)\, m_6 l_6 \tau_1 + 8\, I_{yy6} v_{1x} \omega_{1y} {m_1}^2 - 8\, I_{yy6} {m_1}^2 \omega_{1x} v_{1y} + 8\, I_{yy6} {m_6}^2 \omega_{1y} v_{1x} \\
& -2\, {m_6}^2 m_1 {l_6}^2 \omega_{1x} v_{1y} + 2\, {m_6}^2 m_1 {l_6}^2 \omega_{1y} v_{1x} + 16\, I_{yy6} m_6 m_1 \omega_{1y} v_{1x} \\
& -16\, I_{yy6} m_6 m_1 \omega_{1x} v_{1y} + 2\, m_1 {m_6}^2 {l_6}^3 \cos(q_6)^3\, \omega_{1x} \omega_{1z} + 2\, {m_1}^2 m_6 {l_6}^2 \omega_{1y} v_{1x} \\
& -2\, {m_1}^2 m_6 {l_6}^2 \omega_{1x} v_{1y} - 8\, I_{yy6} {m_6}^2 v_{1y} \omega_{1x} - 2\, m_1 \sin(q_6)\, {m_6}^2 {l_6}^3 \omega_{1y} \dot{q}_6 \\
& -m_1 \sin(q_6)\, {m_6}^2 {l_6}^3 \dot{q}_6^2 - m_1 \sin(q_6)\, {m_6}^2 {l_6}^3 {\omega_{1y}}^2 - \sin(q_6)\, m_1 {m_6}^2 {l_6}^3 {\omega_{1x}}^2 \\
& -m_1 \sin(q_6)\, {m_6}^2 {l_6}^3 (\cos(q_6))^2\, {\omega_{1z}}^2 - 4\, I_{yy6} {m_6}^2 \sin(q_6)\, l_6 \dot{q}_6^2 \\
& -8\, I_{yy6} {m_6}^2 \sin(q_6)\, l_6 \omega_{1y} \dot{q}_6 - 4\, I_{yy6} {m_6}^2 l_6 \omega_{1x} \cos(q_6)\, \omega_{1z} \\
& -4\, I_{yy6} m_1 \sin(q_6)\, m_6 l_6 \dot{q}_6^2 - 4\, I_{yy6} {m_6}^2 \sin(q_6)\, l_6 {\omega_{1y}}^2 \\
& -8\, I_{yy6} m_1 \sin(q_6)\, m_6 l_6 \omega_{1y} \dot{q}_6 - 4\, I_{yy6} \sin(q_6)\, m_1 m_6 l_6 {\omega_{1x}}^2 \\
& -4\, I_{yy6} \sin(q_6)\, {m_6}^2 l_6 {\omega_{1x}}^2 - 4\, I_{yy6} m_1 m_6 l_6 \omega_{1x} \cos(q_6)\, \omega_{1z} + 4\, {m_6}^2 \cos(q_6)\, l_6 \tau_1 \Big) \\
& \Big/ \Big(4\, I_{yy6} {m_1}^2 + 4\, I_{yy6} {m_6}^2 + 8\, I_{yy6} m_1 m_6 + m_1 {m_6}^2 {l_6}^2 + {m_1}^2 m_6 {l_6}^2 \Big)
\end{aligned}
$$

The angular and linear components of the base acceleration due to the external forces on the arm $\mathbf{a}_{1/6ext}$ in the coordinates of $\mathcal{F}_1$ are given by:

$$\begin{aligned}
\alpha_{1/6ext_x} = {} & \Big(4\cos(q_6)\, m_1 I_{zz\,1} n_{6ext\,x} + 4\cos(q_6)\, I_{zz\,1} n_{6ext\,x}\, m_6 \\
& + 4\, m_1 I_{zz\,1} \sin(q_6)\, n_{6ext\,z} - 2\, l_6 m_6 I_{zz\,1} \sin(q_6)\, f_{6ext_y} + 4\, m_6 \sin(q_6)\, I_{zz\,1} n_{6ext\,z} \\
& + \cos(q_6)\, {l_6}^2 m_6 m_1 n_{6ext\,x}\Big) \Big/ \Big(-\cos(q_6)^2\, m_1 I_{zz\,1} m_6 {l_6}^2 \\
& + m_1 \cos(q_6)^2\, m_6 {l_6}^2 I_{xx\,1} + m_1 I_{zz\,1} m_6 {l_6}^2 + 4\, I_{zz\,1} I_{xx\,1} m_6 + 4\, m_1 I_{xx\,1} I_{zz\,1}\Big)
\end{aligned}$$

$$\alpha_{1/6ext_y} = 0$$

$$\begin{aligned}
\alpha_{1/6ext_z} = {} & -\Big(m_1 m_6 {l_6}^2 \sin(q_6)\, n_{6ext\,x} + 2\cos(q_6)\, m_6 l_6 I_{xx\,1} f_{6ext_y} \\
& - 4\, m_1 I_{xx\,1} \cos(q_6)\, n_{6ext\,z} + 4\, I_{xx\,1} m_6 \sin(q_6)\, n_{6ext\,x} - 4\, I_{xx\,1} m_6 \cos(q_6)\, n_{6ext\,z} \\
& + 4\, m_1 I_{xx\,1} \sin(q_6)\, n_{6ext\,x}\Big) \Big/ \Big(-\cos(q_6)^2\, m_1 I_{zz\,1} m_6 {l_6}^2 \\
& + m_1 \cos(q_6)^2\, m_6 {l_6}^2 I_{xx\,1} + m_1 I_{zz\,1} m_6 {l_6}^2 + 4\, I_{zz\,1} I_{xx\,1} m_6 + 4\, m_1 I_{xx\,1} I_{zz\,1}\Big)
\end{aligned}$$

$$\begin{aligned}
a_{1/6ext_x} = {} & \Big({l_6}^2 m_1 m_6 \sin(q_6)\, f_{6ext_z} + 4\cos(q_6)\, I_{yy\,6} m_1 f_{6ext_x} + 4\, I_{yy\,6} m_6 \sin(q_6)\, f_{6ext_z} \\
& + 2\, l_6 m_6 m_1 \sin(q_6)\, n_{6ext\,y} + 2\, {m_6}^2 \sin(q_6)\, l_6 n_{6ext\,y} + 4\, I_{yy\,6} m_1 \sin(q_6)\, f_{6ext_z} \\
& + 4\cos(q_6)\, I_{yy\,6} m_6 f_{6ext_x} + {m_6}^2 \sin(q_6)\, f_{6ext_z} {l_6}^2 \\
& + \cos(q_6)\, m_1 m_6 {l_6}^2 f_{6ext_x}\Big) \Big/ \Big(4\, I_{yy\,6} {m_1}^2 + 4\, I_{yy\,6} {m_6}^2 \\
& + 8\, I_{yy\,6} m_1 m_6 + m_1 {m_6}^2 {l_6}^2 + {m_1}^2 m_6 {l_6}^2\Big)
\end{aligned}$$

$$\begin{aligned}
a_{1/6ext_y} = {} & \Big(-2\, I_{zz\,1} m_6 l_6 n_{6ext\,z} + I_{zz\,1} m_6 {l_6}^2 f_{6ext_y} + \cos(q_6)^2\, m_6 {l_6}^2 I_{xx\,1} f_{6ext_y} \\
& - 2\, I_{zz\,1} m_6 l_6 \sin(q_6) \cos(q_6)\, n_{6ext\,x} + 2\cos(q_6)^2\, I_{zz\,1} m_6 l_6 n_{6ext\,z} \\
& + 2\cos(q_6)\, m_6 l_6 I_{xx\,1} \sin(q_6)\, n_{6ext\,x} - 2\cos(q_6)^2\, m_6 l_6 I_{xx\,1} n_{6ext\,z} \\
& - I_{zz\,1} \cos(q_6)^2\, m_6 {l_6}^2 f_{6ext_y} + 4\, I_{zz\,1} I_{xx\,1} f_{6ext_y}\Big) \Big/ \Big(-\cos(q_6)^2\, m_1 I_{zz\,1} m_6 {l_6}^2 \\
& + m_1 \cos(q_6)^2\, m_6 {l_6}^2 I_{xx\,1} + m_1 I_{zz\,1} m_6 {l_6}^2 + 4\, I_{zz\,1} I_{xx\,1} m_6 + 4\, m_1 I_{xx\,1} I_{zz\,1}\Big)
\end{aligned}$$

$$
\begin{aligned}
a_{1/6ext_z} = & -\Big(-m_1 \cos(q_6) f_{6ext_z} m_6 {l_6}^2 - 2 {m_6}^2 \cos(q_6) l_6 n_{6ext_y} \\
& +4 I_{yy6} m_6 \sin(q_6) f_{6ext_x} - 4 I_{yy6} m_1 \cos(q_6) f_{6ext_z} + 4 I_{yy6} m_1 \sin(q_6) f_{6ext_x} \\
& -4 I_{yy6} m_6 \cos(q_6) f_{6ext_z} - {m_6}^2 \cos(q_6) f_{6ext_z} {l_6}^2 + m_1 \sin(q_6) f_{6ext_x} m_6 {l_6}^2 \\
& -2 m_1 \cos(q_6) m_6 l_6 n_{6ext_y} \Big) \Big/ \Big(4 I_{yy6} {m_1}^2 + 4 I_{yy6} {m_6}^2 \\
& +8 I_{yy6} m_1 m_6 + m_1 {m_6}^2 {l_6}^2 + {m_1}^2 m_6 {l_6}^2 \Big)
\end{aligned}
$$

Finally, the angular and linear components of the base acceleration due to the external forces on the base $\mathbf{a}_{1ext}$ in the coordinates of $\mathcal{F}_1$ are given by:

$$
\begin{aligned}
\alpha_{1ext_x} = & \Big(4 m_6 I_{zz1} n_{1ext_x} - \cos(q_6) {l_6}^2 m_6 m_1 \sin(q_6) n_{1ext_z} - 2 l_6 m_6 I_{zz1} \sin(q_6) f_{1ext_y} \\
& +4 m_1 I_{zz1} n_{1ext_x} + \cos(q_6)^2 {l_6}^2 m_6 m_1 n_{1ext_x} \Big) \Big/ \Big(-\cos(q_6)^2 m_1 I_{zz1} m_6 {l_6}^2 \\
& +m_1 \cos(q_6)^2 m_6 {l_6}^2 I_{xx1} + m_1 I_{zz1} m_6 {l_6}^2 + 4 I_{zz1} I_{xx1} m_6 + 4 m_1 I_{xx1} I_{zz1} \Big)
\end{aligned}
$$

$$
\alpha_{1ext_y} = n_{1ext_y} \big/ I_{yy1}
$$

$$
\begin{aligned}
\alpha_{1ext_z} = & \Big(4 I_{xx1} m_6 n_{1ext_z} + 4 m_1 I_{xx1} n_{1ext_z} + m_1 m_6 {l_6}^2 n_{1ext_z} \\
& -m_1 m_6 {l_6}^2 (\cos(q_6))^2 n_{1ext_z} - 2 \cos(q_6) m_6 l_6 I_{xx1} f_{1ext_y} \\
& -m_1 m_6 {l_6}^2 \sin(q_6) \cos(q_6) n_{1ext_x} \Big) \Big/ \Big(-\cos(q_6)^2 m_1 I_{zz1} m_6 {l_6}^2 \\
& +m_1 \cos(q_6)^2 m_6 {l_6}^2 I_{xx1} + m_1 I_{zz1} m_6 {l_6}^2 + 4 I_{zz1} I_{xx1} m_6 + 4 m_1 I_{xx1} I_{zz1} \Big)
\end{aligned}
$$

$$
\begin{aligned}
a_{1ext_x} = & -\Big(-{m_6}^2 f_{1ext_x} {l_6}^2 - \cos(q_6) {m_6}^2 \sin(q_6) {l_6}^2 f_{1ext_z} + \cos(q_6)^2 {m_6}^2 f_{1ext_x} {l_6}^2 \\
& -{l_6}^2 m_1 m_6 f_{1ext_x} - 4 I_{yy6} f_{1ext_x} m_1 - 4 I_{yy6} m_6 f_{1ext_x} \Big) \Big/ \Big(4 I_{yy6} {m_1}^2 + 4 I_{yy6} {m_6}^2 \\
& +8 I_{yy6} m_1 m_6 + m_1 {m_6}^2 {l_6}^2 + {m_1}^2 m_6 {l_6}^2 \Big)
\end{aligned}
$$

$$
\begin{aligned}
a_{1ext_y} = & -\Big(I_{zz1} \cos(q_6)^2 m_6 {l_6}^2 f_{1ext_y} + 2 \cos(q_6) m_6 l_6 I_{xx1} n_{1ext_z} \\
& -I_{zz1} m_6 {l_6}^2 f_{1ext_y} - \cos(q_6)^2 m_6 {l_6}^2 I_{xx1} f_{1ext_y} + 2 I_{zz1} m_6 l_6 \sin(q_6) n_{1ext_x} \\
& -4 I_{zz1} I_{xx1} f_{1ext_y} \Big) \Big/ \Big(-\cos(q_6)^2 m_1 I_{zz1} m_6 {l_6}^2 \\
& +m_1 \cos(q_6)^2 m_6 {l_6}^2 I_{xx1} + m_1 I_{zz1} m_6 {l_6}^2 + 4 I_{zz1} I_{xx1} m_6 + 4 m_1 I_{xx1} I_{zz1} \Big)
\end{aligned}
$$

$$
\begin{aligned}
a_{1ext\,z} \quad = \quad & \Big(m_1 f_{1ext\,z}\, m_6 {l_6}^2 + \left(\cos\left(q_6\right)\right)^2 {m_6}^2 {l_6}^2 f_{1ext\,z} + 4\, I_{yy\,6} m_6 f_{1ext\,z} + 4\, I_{yy\,6} f_{1ext\,z}\, m_1 \\
& + \cos\left(q_6\right) {m_6}^2 \sin\left(q_6\right) {l_6}^2 f_{1ext\,x}\Big) \Big/ \Big(4\, I_{yy\,6} {m_1}^2 + 4\, I_{yy\,6} {m_6}^2 \\
& + 8\, I_{yy\,6} m_1 m_6 + m_1 {m_6}^2 {l_6}^2 + {m_1}^2 m_6 {l_6}^2\Big)
\end{aligned}
$$

In the same way the velocity of the base was propagated to the arm in (2.43), the acceleration of the base is propagated to the arm and added to the arm acceleration due to the joint acceleration and the velocity-product term (corresponding to the spatial cross product of the arm velocity and joint velocity) to obtain the arm acceleration:

$$
\mathbf{a}_6 = {}^6\mathbf{X}_1 \mathbf{a}_1 + \underbrace{\begin{bmatrix} \left(-\sin(q_1)\omega_{1x} - \cos(q_1)\omega_{1z}\right)\dot{q}_6 \\ 0 \\ \left(\cos(q_1)\omega_{1x} - \sin(q_1)\omega_{1z}\right)\dot{q}_6 \\ \left(-\sin(q_1)v_{1x} - \cos(q_1)v_{1z}\right)\dot{q}_6 \\ 0 \\ \left(\cos(q_1)v_{1x} - \sin(q_1)v_{1z}\right)\dot{q}_6 \end{bmatrix}}_{\mathbf{c}_6 = \mathbf{v}_6 \times S\dot{q}_6} + S_6\ddot{q}_6. \tag{2.53}
$$

The arm's spatial acceleration can now be employed to calculate the arm-joint acceleration, which is given by

$$
\ddot{q}_6 = \left(\tau_6 - S_6^T \mathbf{f}_6 - \mathbf{I}_6 S_6 \left({}^6\mathbf{X}_1 \mathbf{a}_1 + \mathbf{c}_6\right)\right)\left(S_6^T \mathbf{I}_6 S_6\right)^{-1}. \tag{2.54}
$$

In an analogous way to the computation of the base acceleration, the joint acceleration $\ddot{q}_6$ can be split into a term associated to the velocity-product terms $\ddot{q}_{6c}$, i.e. does not include the external forces, and two other terms that include the external forces acting on the arm and the base, $\ddot{q}_{6ext}$ and $\ddot{q}_{6/1ext}$, respectively. These terms are:

$$\begin{aligned}
\ddot{q}_{6c} &= \Big(4\, I_{yy6} m_1 \omega_{1z} I_{xx1} \omega_{1x} - 4\, I_{yy6} m_1 \omega_{1x} I_{zz1} \omega_{1z} - m_1 \omega_{1x} I_{zz1} \omega_{1z} m_6 {l_6}^2 \\
&\quad - m_1 m_6 {l_6}^2 I_{yy1} \omega_{1z} \omega_{1x} + 2\, m_1 m_6 {l_6}^2 I_{yy1} \cos(q_1)^2 \omega_{1x} \omega_{1z} \\
&\quad - m_1 m_6 {l_6}^2 I_{yy1} \sin(q_1) {\omega_{1z}}^2 \cos(q_1) + m_1 \omega_{1z} I_{xx1} \omega_{1x} m_6 {l_6}^2 \\
&\quad + m_1 m_6 {l_6}^2 I_{yy1} \cos(q_1) {\omega_{1x}}^2 \sin(q_1) + 4\, m_1 \tau_6 I_{yy1} + 8\, I_{yy6} \tau_6 m_1 \\
&\quad + 2\, m_1 \tau_6 m_6 {l_6}^2 - 8\, I_{yy6} \omega_{1x} I_{zz1} \omega_{1z} m_6 + 16\, I_{yy6} m_6 \tau_6 + 8\, I_{yy6} \omega_{1z} I_{xx1} \omega_{1x} m_6 \\
&\quad + 8\, \tau_6 I_{yy1} m_6\Big) \Big/ \Big(\left(4\, m_1 I_{yy6} + m_1 m_6 {l_1}^2 + 8\, I_{yy6} m_6\right) I_{yy1}\Big), \\
\ddot{q}_{6ext} &= \left(4\, n_{6ext\,y}\, m_1 + 4\, n_{6ext\,y}\, m_6 + 2 m_6 l_6 f_{6ext\,z}\right) \Big/ \\
&\quad \left(4\, I_{yy6} m_1 + m_6 {l_6}^2 m_1 + 4\, m_6 I_{yy6}\right), \\
\ddot{q}_{6/1ext} &= \Big(-4\, m_1 I_{yy6} n_{1ext\,y} - 4\, I_{yy6} n_{1ext\,y}\, m_6 - n_{1ext\,y}\, m_6 {l_6}^2 m_1 \\
&\quad + 2\, m_6 l_6 I_{yy1} \cos(q_6) f_{1ext\,z} + 2\, m_6 l_6 I_{yy1} \sin(q_6) f_{1ext\,x}\Big) \Big/ \\
&\quad \Big(\left(4\, I_{yy6} m_1 + m_6 {l_6}^2 m_1 + 4\, m_6 I_{yy6}\right) I_{yy1}\Big).
\end{aligned}$$

These equations show that the joint acceleration $\ddot{q}_6$ does not only depend on the applied torque τ_6 supplied by the joint actuator, but also on the external forces that propagate from the base to the arm, as well as the load and possible contact forces on the arm.

The forward dynamics equations just obtained allow to simulate the model of a SSMM with 1-DOF arm and develop a motion and joint controllers. The simulation of the SSMM model implemented with the forward dynamics equations is presented in the next chapter, which also discusses the measurements obtained during the field tests and validate the model's physical compliance.

3. SIMULATION

To corroborate the dynamic model of the SSMM, a SSMM was simulated using the Spatial Toolbox (version 2) for Matlab (*Spatial Vector Algebra Toolboox for Matlab*, 2014). This toolbox provides a set of algorithms and functions for simulating the motion dynamics of multibody mechanical systems using the spatial vector algebra approach. The toolbox allows to build models in terms of easy to use data structures that only require one to fill in the parent array λ of the the kinematic structure, the position of the joints relative to the parent body (i.e. the parameters of the joint-location transform $\mathbf{X}_T$), the location of the COM of each body and the mass and inertia matrices of the bodies. The toolbox contains functions for computing the forward and inverse dynamics that can be called from Matlab scripts or from Simulink simulation models. The functions provided also allow to include user-defined contact and motion constraints. The results from the simulations can be visualized using functions that are part of the toolbox and that allow to create 3D graphical representations of the robot or multibody mechanical system.

The SSMM model built for the simulations corresponds to the one presented in chapter 2 with a 1-DOF manipulator to provide a realistic representation of a Cat® 262C Series 2 compact skid-steer loader that was robotized for teleoperation and autonomous navigation experiments. The simulation results of the Cat® 262C are later compared to the measurements obtained with the real Cat® 262C skid-steer loader.

In this chapter, the multiple considerations of the simulation are discussed as model parameters, internal friction and viscosity of the joints and the simulation scheme.

3.1. SSMM Model Parameters

The Cat® 262C will be modeled as a floating base with four wheels and a one-DOF manipulator. As shown on the model illustrated in fig. 2.1, the mobile base is assigned body number 1, the wheels are bodies $2,\ 3,\ 4,\ 5$ and the arm is the sixth body. The arm of the Cat® 262C is implemented in the simulation as a 1-DOF rotary joint with axis-y parallel to y_1. Extra bodies and joints can be easily added to represent the motion of the loader's bucket. However, to reduce the number of variables the experiments were carried out using a fixed bucket positon and therefore bodies $i = 7, 8, \ldots, N$ of the general SSMM model were not defined in the model script that can be obtained from (*SSMM Model Files, Experimental Data, Simulation Videos and Spatial Vector Algebra Library*, 2014, http://ral.ing.puc.cl/ssmm.htm). The specific values for the geometric and inertial parameters in Table 2.5 of the Cat® 262C model are summarized in Table 3.1. It is to be noted that the joint location parameter h of the loader arm corresponding to frame $\mathcal{F}_6$ in fig. 2.1 shown in Table 3.1 is negative. This is because the Cat 262C arm is positioned at the rear-end of the machine, opposite to the front location of the arm in the general SSMM model shown in fig. 2.1.

TABLE 3.1. Cat® 262 Modeled Parameters in SI Units.

Description		Mobile Base	Wheels	Manipulator
Dimensions	x	$a = 3$	$2r = 0.9$	$\epsilon = 0.15$
	y	$b = 1.6$	$w = 0.25$	$\epsilon = 0.15$
	z	$c = 1.2$	$2r = 0.9$	$l_6 = 3.3$
Mass		$m_1 = 2389$	$m_{2,3,4,5} = 47.7$	$m_6 = 1034$
Joint Location Parameters		$-$	$d = 0.2$, $e = 1$, $f = 0.5$	$h = -1.2$

Once each body is declared, the ground contact points are defined with respect to each body's frame $\mathcal{F}_i$ in 3D coordinates. More specifically, eight CPs are defined at each corner of the mobile base represented by a rectangular box, one CP is defined at the end of the manipulator arm where the tip of the bucket is located, and each of the four wheels has 32 CPs distributed around its perimeter. Such number of contact points on the wheels was choosen to obtain a more realistic simulation of the ground-wheel interaction. Fig. 3.1 shows the simulated SSMM model through the spatial vector algebra toolbox.

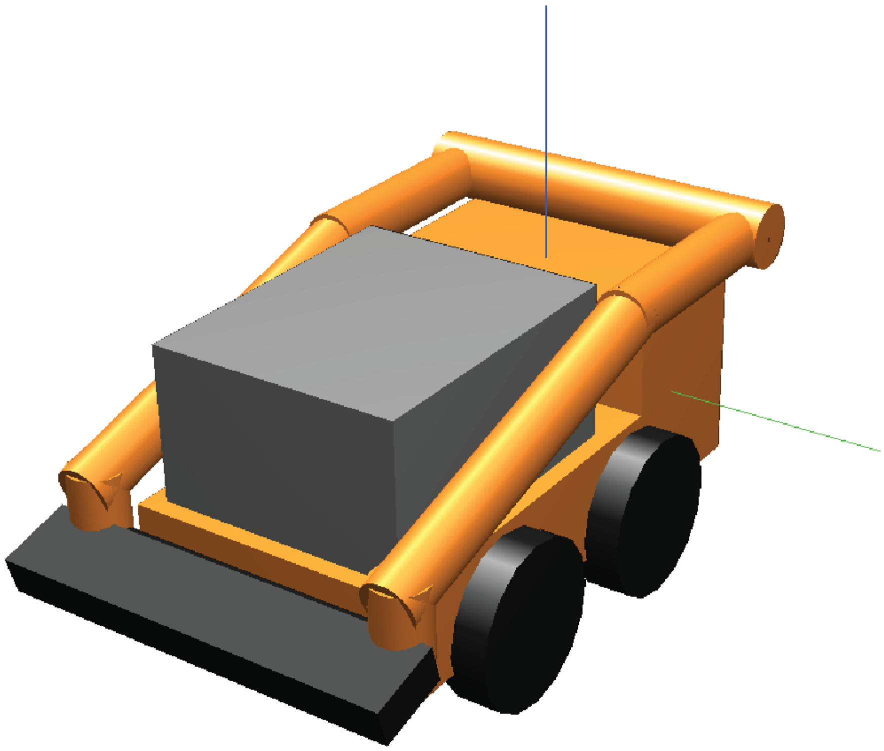


FIGURE 3.1. Simulated SSMM model.

3.2. Joints Friction, Viscosity and Control

3.2.1. Response to step inputs

When a step command is send to the Cat® 262C, the response is not instantaneous due to the response dynamics of the different mechanical subsystems (diesel engine, pump, hydraulic motors). A basic model of the engine is shown if fig. 3.2. The first input of the system is the accelerator throttle that set the power of the diesel engine. The engine moves an hydrostatic pump which supplies hydraulic pressure to the left and right hydraulic motors through the corresponding servovalves.

The diesel engine, the hydrostatic pump and the hydraulic motors each have an inertia and friction constants which add to the total response time and efficiency reductions of the power train. It was measured from the data collected in the experiments that the machine velocity profile is such that the acceleration period last 0.2 s, while the deceleration period

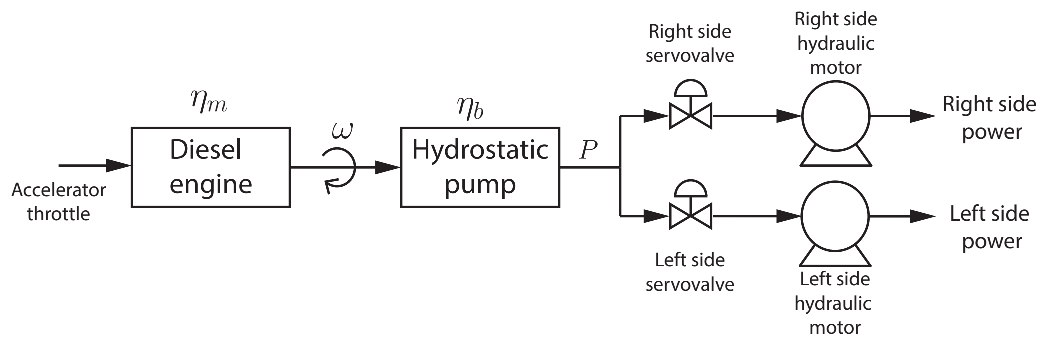


FIGURE 3.2. Mobile base engine model.

is 0.1 seconds. These times where included as a velocity ramping in the simulated model. These ramping periods are implemented on the machine for safety reasons and to reduce mechanical wear-off and damages that could be caused by sudden braking.

3.2.2. Wheel torque model

The net torque of each wheel assumes the standard viscous friction force proportional to the wheel velocity $\dot{q}_i$. Thus each wheel satisfies a first order equation of the form

$$J_i \ddot{q}_i = \tau_i - c\dot{q}_i, \qquad i = 2, 3, 4, 5 \tag{3.1}$$

where J_i is the moment of inertia of the wheel, τ_i the torque applied to the wheel and c is the viscous friction coefficient. The viscous friction coefficient can be obtained from (3.1) as

$$c = \frac{\tau_i - J_i \ddot{q}_i}{\dot{q}_i},$$

which should be valid also when the machine achieves a steady-state velocity $\dot{q}_{max}$ with steady-state torque τ_{max}. Since in steady-state equilibrium $\ddot{q}_i = 0$, then

$$c = \frac{\tau_{max}}{\dot{q}_{max}}.$$

From the machine design specifications it is possible to obtain P_{max}, which allows to calculate $c = \frac{P_{max}}{\dot{q}_{max}^2}$ since $\tau_{max} = \frac{P_{max}}{\dot{q}_{max}}$.

3.2.3. Arm torque model

Similar to the wheels, the arm also has a viscous friction force that is proportional to the arms velocity ($\dot{q}_6$) and its coefficient is calculated using (3.1) and experimental data acquired of the falling arm. The torque applied by the arm joint controller is assumed to follow a PID control law:

$$\tau(q_{6_e}) \;\; = \;\; K_p \cdot q_{6_e} + K_d \cdot \frac{d}{dt} q_{6e} + K_i \int q_{6_e} dt - c\dot{q}_6 \tag{3.2}$$

where $q_{6_e} = {q_6}^r - q_6$ is the error between the reference position ${q_6}^r$ and the measured joint position q_6. The controller proportional constant was set to

$$K_p \;\; = \;\; \left[\frac{l_6}{2} m_6 g + l_6 m_{load} g \right].$$

The chosen value for K_p is equal to the torque that should be applied by the arm at an horizontal equilibrium position while holding a load of size m_{load}. While the derivative and integral constants were set to $K_d \;=\; 0.5 K_p T, \quad K_i \;=\; \frac{K_p}{2T}$, where T is the dead time, and K_i and K_d are tuned according to the Ziegler-Nichols method. With this selection of controller parameters, it is ensure that the controller's response does not compromise the stability of the arm and dampens arm oscillation that affect the vehicle's displacement.

3.3. Simulation Flowchart

A simulation flowchartfor the SSMM model is shown in fig. 3.3. The simulation starts with the initial conditions for the joint states (position and velocities contained in x, q and $\dot{q}$), the joint applied and dissipative torques represented by τ_j and the external forces as CP forces and load force represented by F_{ext}. The first four terms ($x,\ q,\ \dot{q},\ \tau_i$) are needed to calculate the SSMM foward dynamics and the F_{ext} is needed to simulate the interaction with the environment.

Simultaneously, the user supplies the joystick setpoints (torque, right and left wheels throttles) that results in the applied torque to each wheel and the arm joint setpoint that provides the reference signal to the 1-DOF arm. The applied torque of the wheels and the arm joint torque minus the viscous friction torques feedback to the system's model. The ground contact points are calculated and supplied to for the next iteration of the model.

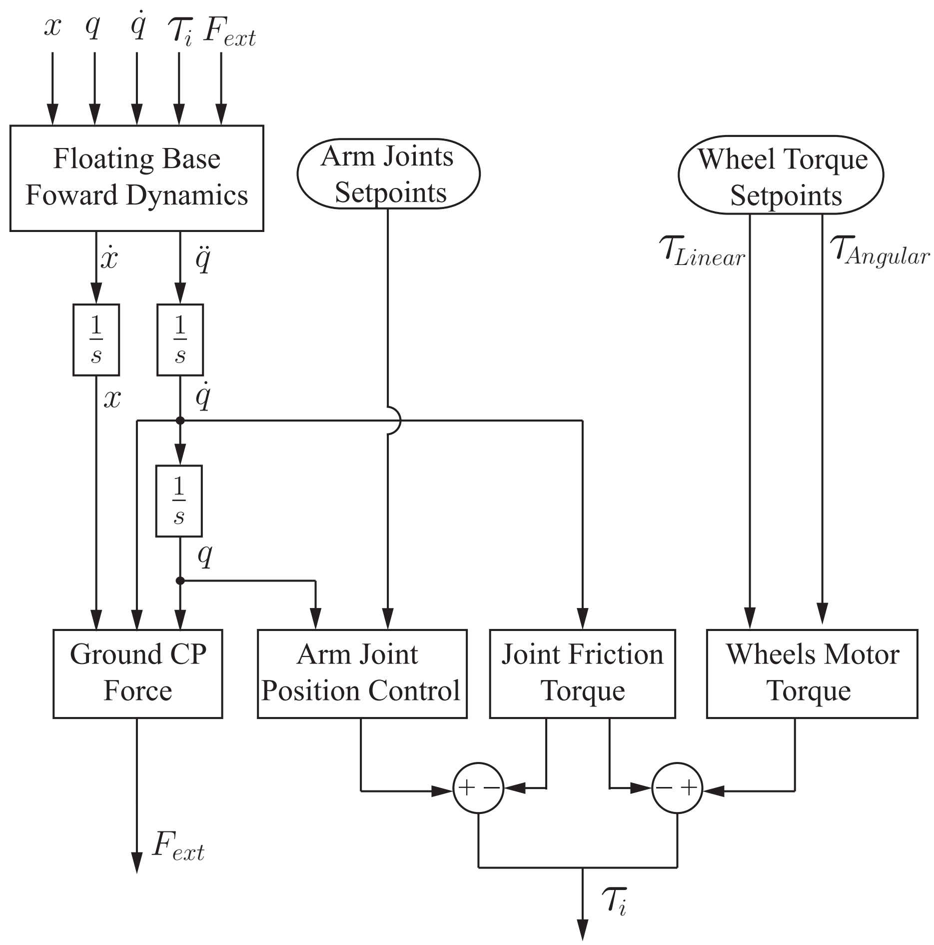


FIGURE 3.3. Simulation flowchart.

4. SIMULATION AND EXPERIMENTAL VALIDATION

In order to validate the SSMM model, motion experiments were conducted using the Cat® 262C skid-steer loader shown in fig. 4.1. The tests were carried out on asphalt pavement with the machine being remotely operated. The tests consisted of two driving maneuvers (i) straight line motion for approximately 4 m, and (ii) in-place 360° rotations carried out with and without load. The load applied to the bucket corresponds to 5×80 liter drums of water totaling 400 kg (approx.). This load is equivalent to roughly 11% of the unloaded machine weight and is sufficient to considerably affect the location of the loader's COM. The four set of experiments were repeated ten times each. The acceleration and heading data was acquired using a high precision Crossbow® IMU and gyroscope model RGA300CA. This device outputs linear accelerations in the X, Y and Z axes, angular velocity ω_z, roll and pitch angles ϕ and θ, respectively.

FIGURE 4.1. Compact skid-steer loader at the experiment site with unloaded bucket (left) and loaded bucket (right).

In this section three main results are shown, one comparing the simulated against the experimentally acquired data in order to evaluate the physical accuracy of the model. The other two results correspond to the simulation of the contact points and wheels velocities. These results are discussed in the context of theoretical background in the following subsections.

4.1. Comparison of Experimental and Simulated Results

The measurements for both straight line motion and in-place rotation show a reasonable agreement with the simulated values. This can be confirmed from fig. 4.2, which shows the machine position for the twice integrated acceleration measured in the straight line motion experiment without load (blue lines) and the simulated value for the machine longitudinal position (red line). Similarly, the in-place rotation experiment, whose results are shown in fig. 4.3, display great consistency with the simulated turning rate of the machine. From the straight line experiments and simulation of fig. 4.2 it is also possible to see that the both the measured and simuated velocities are consistent. In the case of the turning speed shown in fig. 4.3, the agreement between the measured and simulated acceleration and deceleration is clear. The mismatch in the final linear and angular displacements is due to the fact that setpoints were manually issued, and there is a ± 1 second difference in triggering the stopping command.

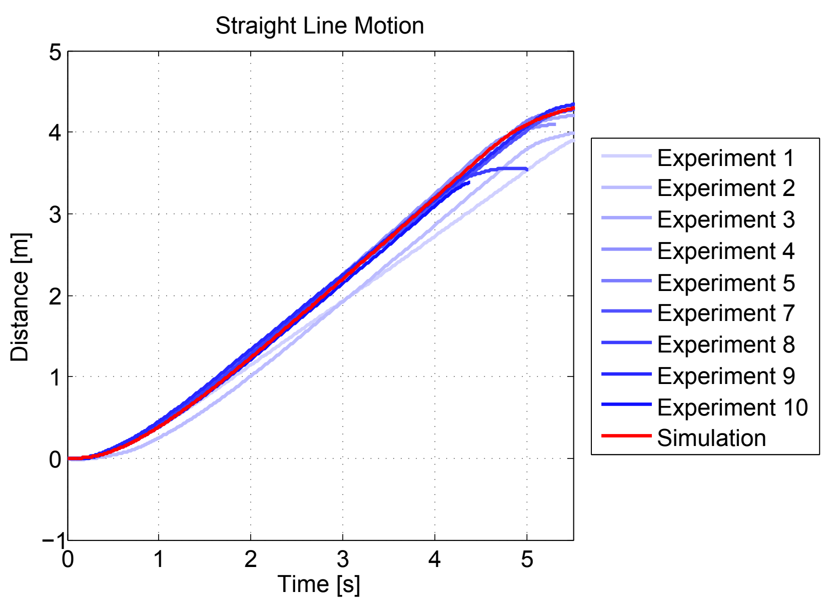


FIGURE 4.2. Straight line motion experimental data compared to the simulated model with the SSMM without load.

The experiments carried out with load show that the machine's acceleration was slightly reduced. This can be observed in the straight line motion experiment of fig. 4.4, which

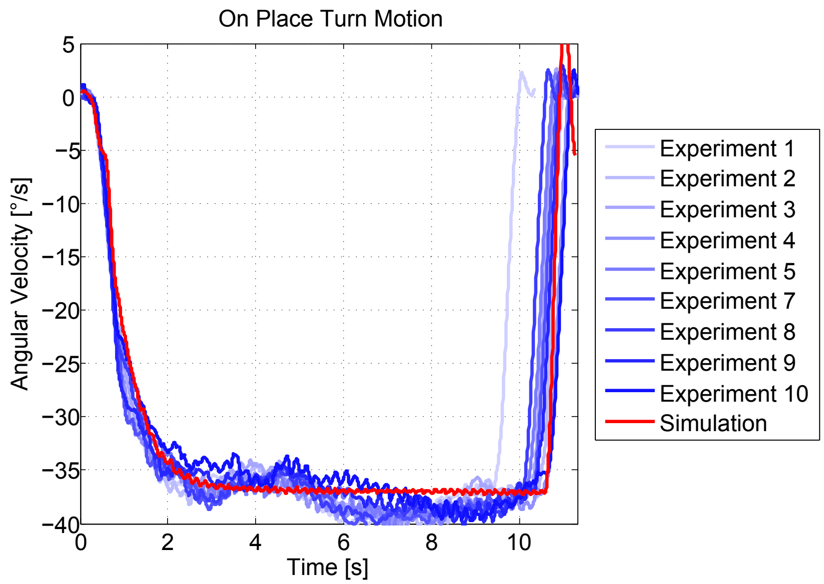


FIGURE 4.3. On place turn motion experimental data compared to the simulated model with the SSMM without load.

shows that the position curve has a slightly smaller slope when compared to the same curves in fig. 4.2. Similarly, the in-place turn with the loaded bucket took about 4 seconds more to complete the 360° turn, and the measured turning velocity decreased from roughly $37{\pm}2\ {}^\circ/s$ to $27{\pm}2\ {}^\circ/s$ as apparent from the comparison of figs. 4.3 and 4.5.

It is to be noted that the ripple in the simulated turning speed of the loaded and unloaded machine shown in figs. 4.3 and 4.5 is produced by the skidding, which is not constant because the active CPs change while the machine rotates. Also minor arm oscillations due to controller tuning induce small disturbances in the forces or the active CPs. These results confirm the power of the spatial vector algebra modeling approach in capturing the force interactions between bodies of the SSMM.

To quantify the error between the simulated values and the real measurements obtained during the straight line and in-place turn experiments, the measurements were averaged to obtain the mean error and its standard deviation. In the computation of the average trajectory, the worst four experiments were discarded because of noticeable discrepancies in the

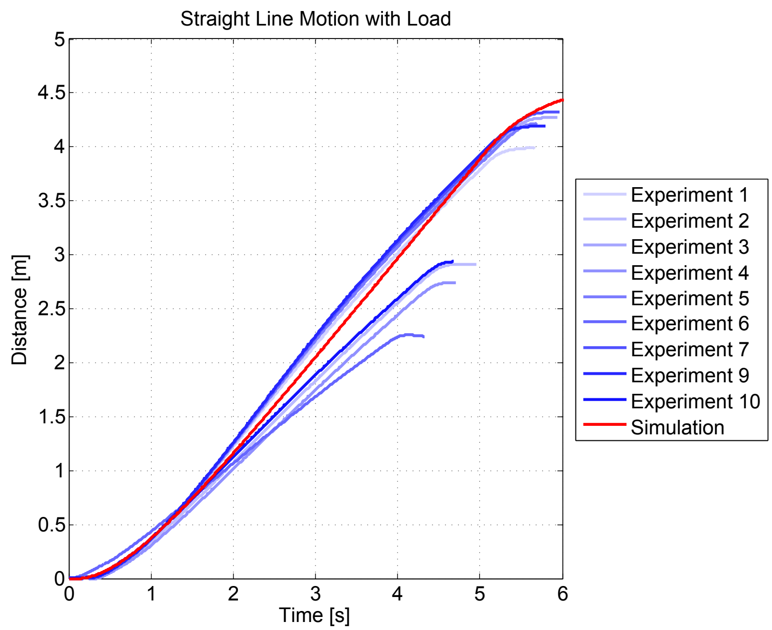


FIGURE 4.4. Straight line motion experimental data compared to the simulated model with the SSMM with 400 kg load.

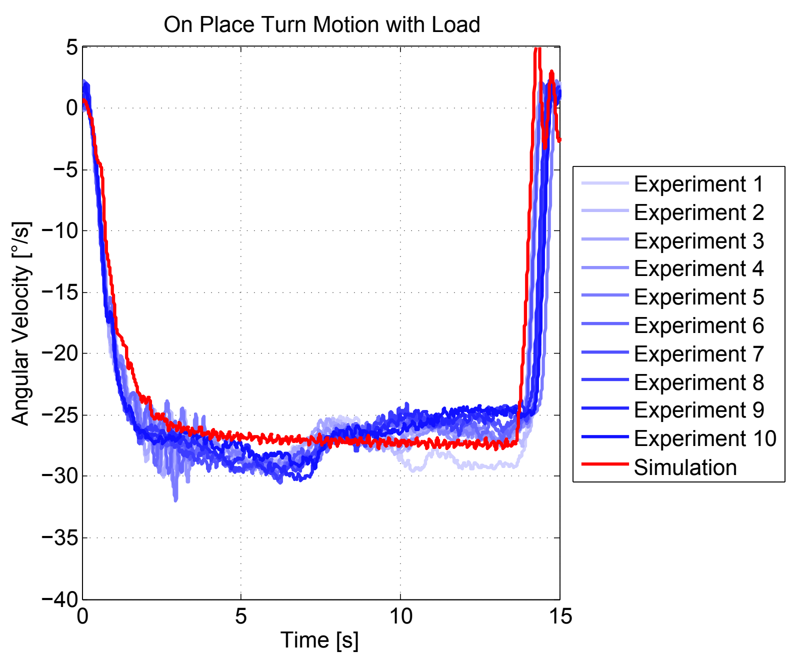


FIGURE 4.5. On place turn motion experimental data compared to the simulated model with the SSMM with 400 kg load.

trajectories length caused by the ± 0.75 s reation-time errors of manually issuing the stopping commands. Due to this reason, it was only after computing the position curves from the inertial measurements that the outliers could be identified and discarded. The figs. 4.6 and 4.7 present the mean error between the simulation and the measurements (blue line) bounded by a lower and upper curve corresponding to one-standard deviation (red line). Fig. 4.6 shows that the accumulative error increases from 0.01 m to 0.06 m roughly after 4 seconds of motion. This result shows that the error at any given time is below 1.25% of the total distance traveled. This accumulative error is expected because the model parameters, such as mass and COM location, are not perfectly known and any discrepancy between the simulated and measured acceleration is integrated in time to yield an increasing error. On the other hand, the polygonal approximation of the wheels can cause other errors in the traction model that affect the estimated displacement. Nonetheless, the relative error of 1-2% is reasonable considering that an IMU was employed to estimate the motion. This result provides significant evidence that the simulated instantaneous accelerations and forces are very close to the correct ones, since after four seconds of motion the standard deviation is less than $\sigma = \pm 0.03$ m for a 4 m long trajectory. Considering $n = 9$ valid experiments, the 95% confidence interval after four seconds is $\pm 1.96\sigma/\sqrt{n} = 0.02$ m, which in relative terms with respect to the mean value $\mu = 4$ m corresponds to a small confidence interval of $\pm 0.5\%$.

Regarding the average error between the simulation and the measurments for the in-place turn experiment, the results in fig. 4.7 show that the error is smaller than a 2-3°, with a significantly noisier curve because of the angular velocity measurement noise of the gyroscope as can be seen in the experiments shown in fig. 4.3. In this case it is possible to see that the mean error is approximately $-2\ /s$ and stays bounded unlike the longitudinal motion measurements. This is because the angular velocity measurment is not computed by integrating accelerations like in the case of the longitudinal motion experiments. The mean error between the simulated and measured angular velocities is approximately $6\ \%$ of the theoretical value. However, the error oscillates and the final angular value of the simulated turn fits well with the experimental observations as shown in fig. 4.3.

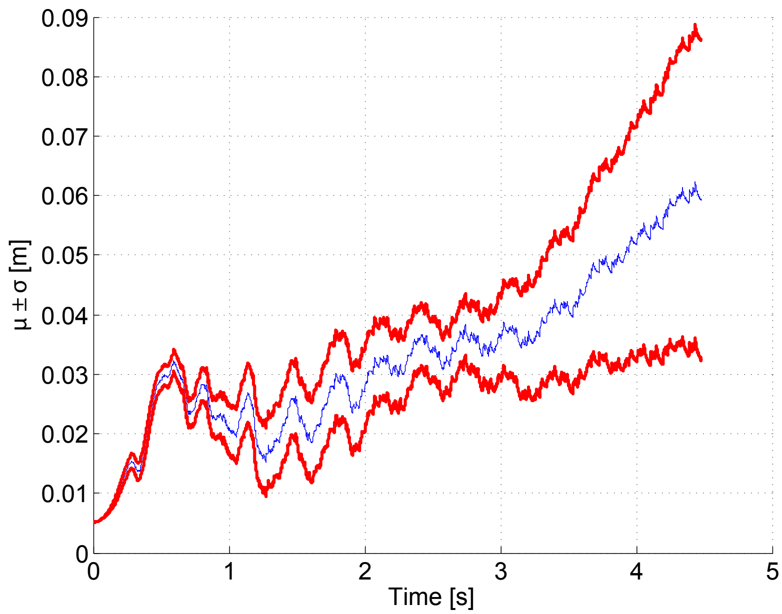


FIGURE 4.6. Error between the simulation and the mean value of the experiments for straight line motion.

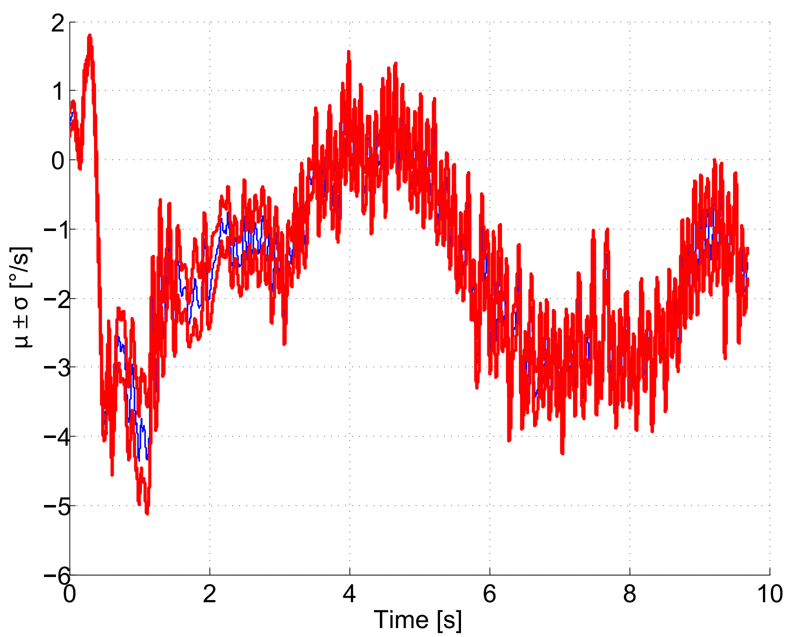


FIGURE 4.7. Error between the simulation and the mean value of the experiments for turn on place motion.

4.2. Contact Points Simulation Results

The simulation results for the normal forces acting on each of the active contact points of the n-sided polygonal approximation to the wheels of the SSMM are presented in fig. 4.8, which shows how each point becomes gradually active as the wheel turns and sinks in the compliant terrain surface when the machine is translating and turning along and arc of a circle. The curves in fig. 4.8 correspond to a segment of the graph of the normal forces acting on the contact points of the right-front wheel of the SSMM that is presented in fig. 4.9. It is possible to observe in fig. 4.9 that the envelope curve of the contact point forces has a maximum value which is different for the left/right and front/rear wheels. This is because the centrifugal acceleration forces of the turning SSMM causes a roll moment about its longitudinal axis $\mathbf{x}_1$ that makes the SSMM lean towards the outside of the turn, thus increasing the pressure on the external wheels. In the simulation, the SSMM was made to turn to the left, i.e. counter-clockwise about the $\mathbf{z}_1$ axis, therefore, the right-side wheels are subject to larger normal forces than the left wheels. Also due to the weight distribution of the SSMM, in which the COM is closer to the rear wheels, the rear wheels have a normal force that is almost twice the normal force of the front wheels.

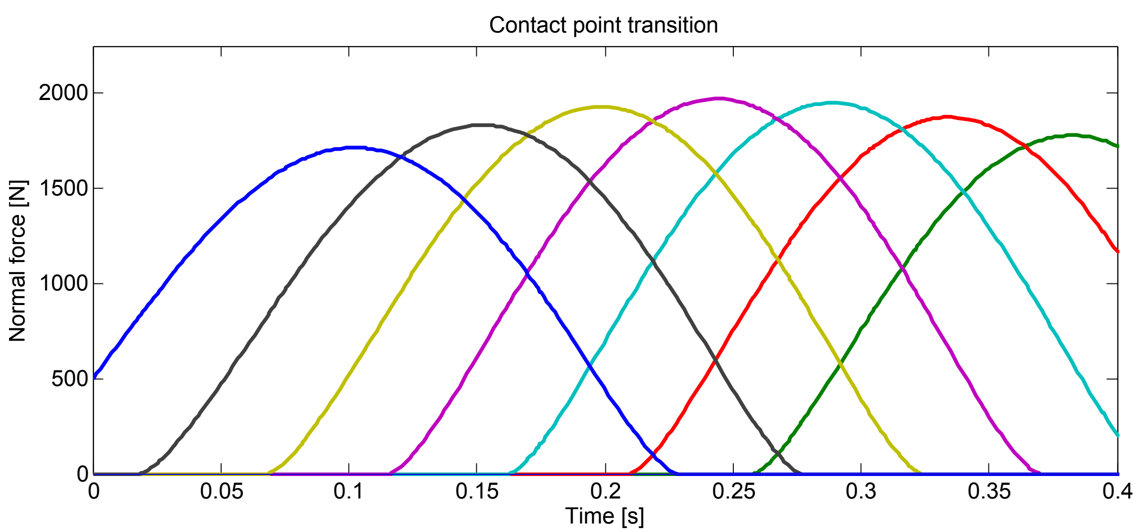


FIGURE 4.8. Evolution of the normal forces acting on the contact points of the rotating right-front wheel.

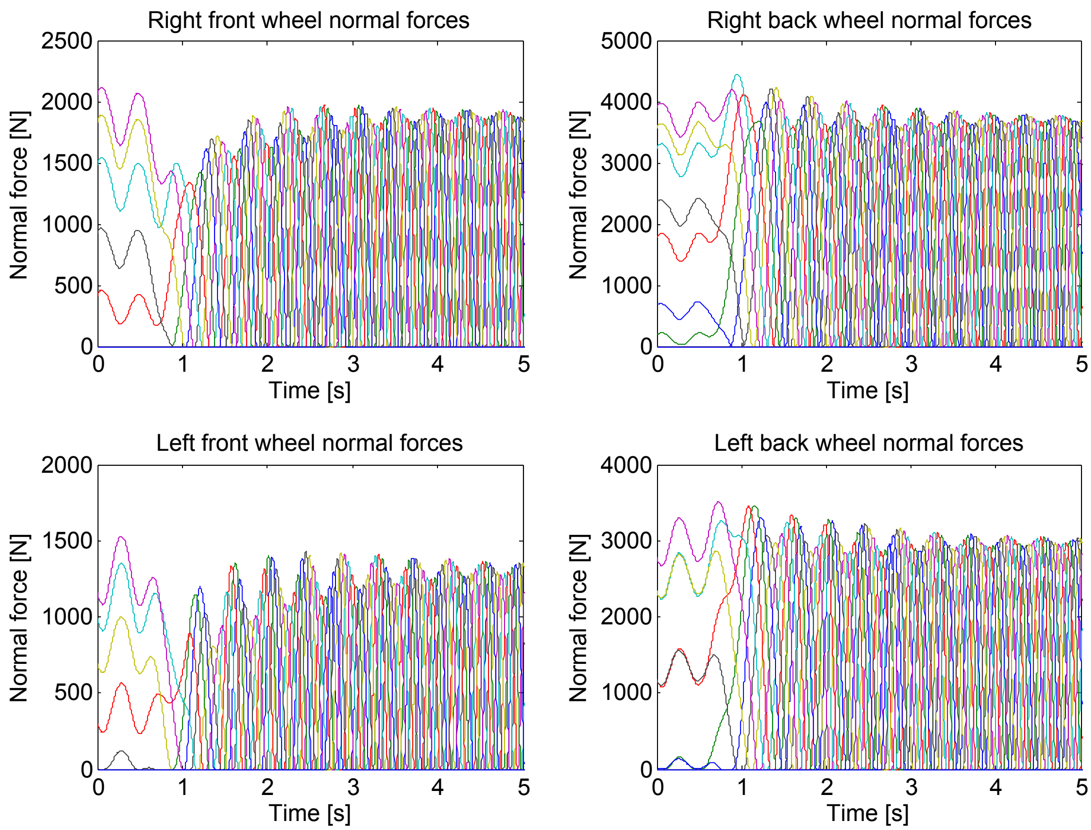


FIGURE 4.9. Evolution of the normal forces acting on the contact points of the wheels of the SSMM turning left.

If all the normal forces acting on the active contact points of each of the four wheels are added together, the curves for the total computed normal force shown in fig. 4.10 are obtained. These curves show the total resulting normal force when the wheels are approximated by n-sided polygons with $n = 32, 64, 128$. It is possible to see that with more contact points, the normal force oscillates less as the contact between the wheels and the terrain is produced in a more continuous way. Nonetheless, regardless of the number of contact points and the amplitude of the oscillations, the average total normal force is 35429.1 N. This number is consistent with the total weight $W_{tot} = m_{tot} \cdot g = 35451.4$ N computed using the total mass $m_{tot} = m_{body} + m_{arm} + 4m_{wheel} = 3613.8$ kg and the gravity acceleration constant $g = 9.81$ m/s^2. This confirms the validity of the contact point model as it provides a reasonable description of the ground-wheel interaction forces. The error between the theoretical normal force and the one obtained from the simulations is less than 0.3% for $n = 32$, and less than 0.1% for $n = 128$.

FIGURE 4.10. Total normal force on the SSMM computed from the sum of the normal forces acting on all the active contact points of each of the four wheels.

To understand better the results presented in fig. 4.10, the fast Fourier transform (FFT) was calculated for the normal forces arising for each of the polygonal approximations to the wheel. The frequency spectrum amplitude obtained is shown in fig. 4.11. Considering that for a given linear longitudinal velocity v, a wheel of radius r with p CPs will have a CP bumping frequency given by $(vp) / (2\pi r)$, then for the simulated SSMM, if v is approximately 1 m/s and $r = 0.45$ m, the bumping frequencies can be calculated to be 11.32 Hz, 22.64 Hz, and 45.28 Hz, for $n = 32, 64, 128$, respectively. As expected, the peak amplitudes of the FFT occur at frequencies 11.7 Hz, 23.4 Hz and 46.8 Hz, once again confirming the accuracy of the simulation. It is also possible to see in the FFT plot of fig. 4.11 that in all simulations there is a response with an oscillation frequency of 3 Hz. This low frequency oscillation is produced by the arm controller, which cannot filter the disturbances propagated from the wheels to the base up to the arm. This coupling between the arm and the base is propagated back to the wheels which experiment a variation in the normal forces due to the vibrations of the arm.

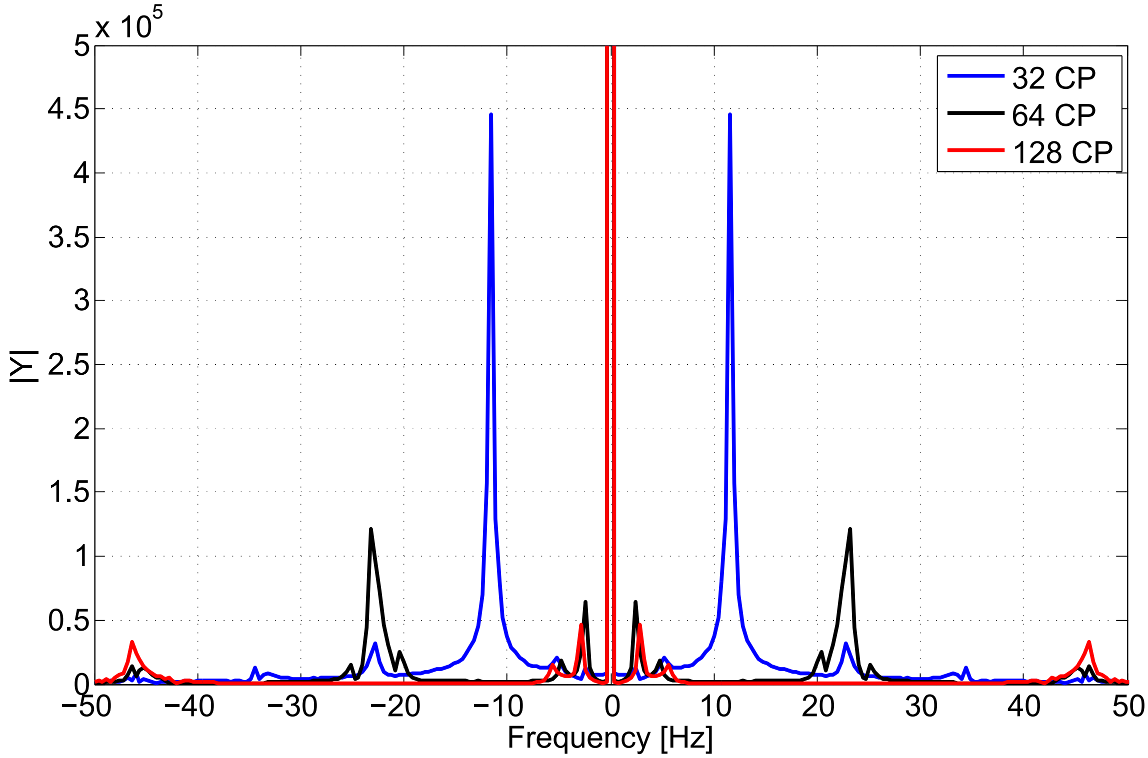


FIGURE 4.11. Total normal forces FFT.

4.3. Skid-Steer Base Model Validation

The relationship between the longitudingal and lateral velocities of the wheels' centers with respect to the COM's velocity expressed in coordinates of the floating base frame $\mathcal{F}_1$ were computed from the numerical simulation results and checked for consistency with respect to the expected behaviour described by (2.24) and (2.25) for the case the vehicle is turning along a semi-circle trajectory. The mobile base starts its motion from an initial rest position according to a trapezoidal velocity profile applied to the wheels in which the left-side wheels have a higher velocity reference set-point than those on the right to make the machine turn right.

According to (2.24), the longitudinal velocity of the wheels on a give side (right or left) must be the same. This is indeed the case as shown in the curves of fig. 4.12. It is also possible to notice that for the SSMM turning to the right from an initial rest position, the interior wheels to the curve, i.e. those on the right side, initially have a negative velocity as they skid due to the large torque applied on the left-side wheels. Once the torque on the right-side wheels is increased, these wheels gain traction and start rolling. Due to the initial difference between the velocities of the left-side and right-side, with $\omega_1 \gg v_1$, the SSMM turns almost in-place and the right wheels briefly move backwards as observed in the real machine.

On the other hand, according to (2.25), the lateral velocity of the front wheels must be the same, and likewise, the lateral velocity of the rear wheels must be equal to each other. This is verified in the simulation results for the wheels' lateral velocities presented in fig. 4.13. Since the COM is closer to the rear axle than to the front axle, the lateral velocity of the rear-wheels is significantly smaller than the lateral velocity of the front wheels.

Even though this results may seem trivial because the wheels in the real skid-steer loader are rigidly attached to the mobile base and cannot separate from it, the results confirm that the model built using spatial vector algebra approach is consistent with the theoretical kinematic constraints stated in (2.24) and (2.25).

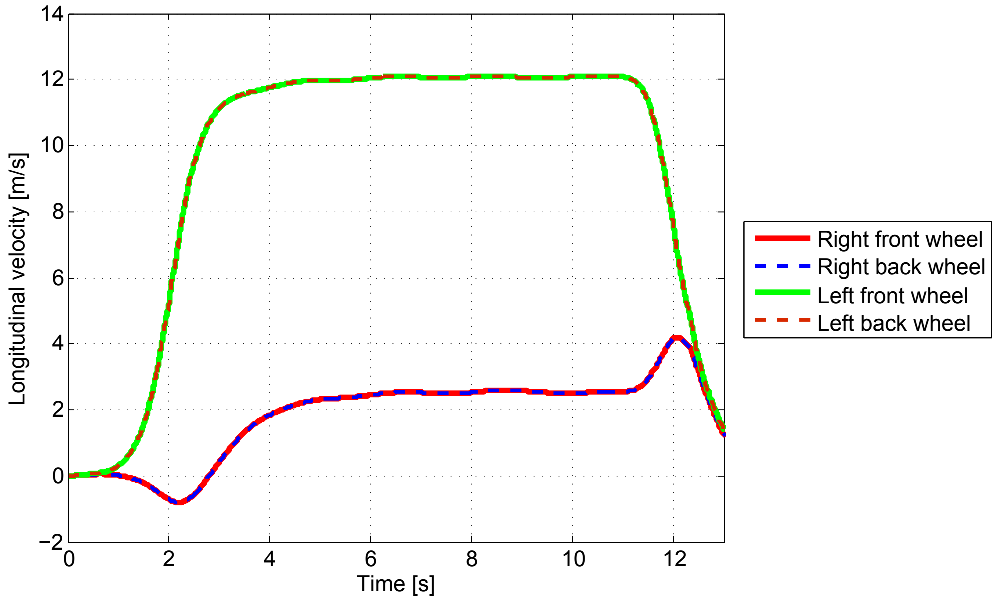


FIGURE 4.12. Longitudinal velocities of wheels.

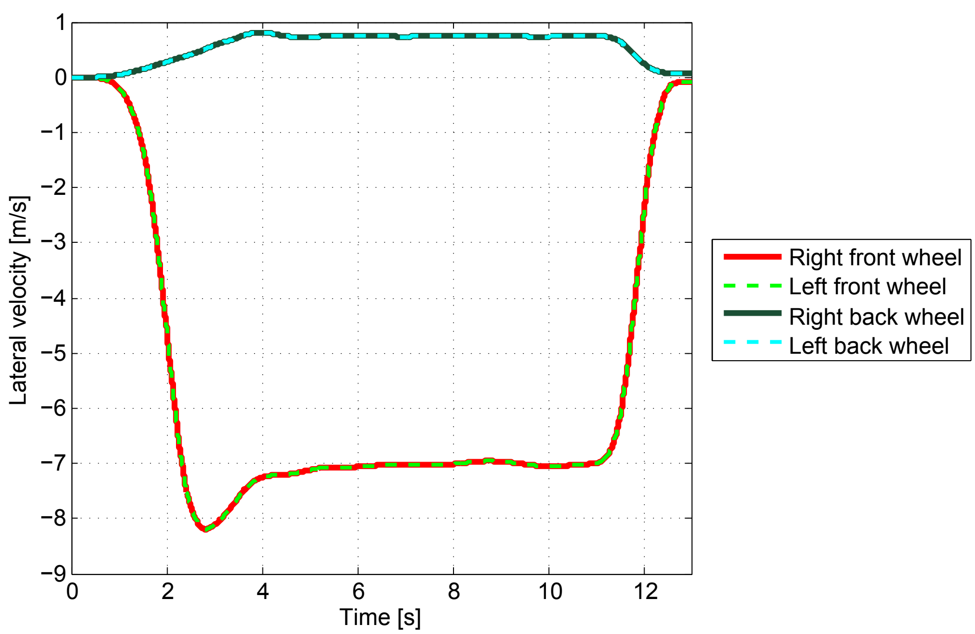


FIGURE 4.13. Lateral velocities of the wheels.

4.4. Base-Arm Interaction

The propagation of the forces acting on the base due to the wheel-ground interaction, more specifically the normal and tangential reaction forces at the contact points, produces disturbances on the SSMM's arm whose position is held close to a fixed reference value using a PID controller as explained in section 3.2.3. The closed-loop response of the arm when the base is commanded to move according to a trapezoidal velocity profile is shown in fig. 4.14. It is possible to observe that the arm initially goes down from a starting position of -14° to -14.5° when the base is accelerated until the speed of the base levels to a value of almost 1 m/s. The opposite effect occurs on the arm when the base decelerates as may be seen in the third graph of fig. 4.14. The transient perturbations occurring during acceleration and deceleration of the base are more notorious in the fourth graph of fig. 4.14, which shows the angular velocity. of the arm joint. It is also possible to observe that there is a permanent oscillation in the arm velocity which is caused by the periodicity of the normal forces acting on the contact points that become active as the wheels rotate. Considering that the mobile base is translating at a velocity of $v_x = 1$ m/s, the angular velocity of the wheels is $\omega = v_x/(2\pi r) = 1/(2\pi 0.45) = 0.35$ revolutions per second. Thus it takes the wheel 1/0.35=2.83 seconds to complete one turn, and since the wheel has 32 contact points, the contact interaction period is approximately 2.83/32=0.088 seconds. This contact interaction period is precisely the period of the oscillations that can be observed in fig. 4.14 during the time lapse at which the mobile base moves with constant velocity roughly between seconds 2 and 4. This behaviour can also be observed in the controller's response curve which applies a torque that attempts to compensate the arm position. It is also possible to confirm that the controller applies a torque which has an average value of 16.14 kNm, consistent with the theoretically expected torque which for a 3.3 m long arm that has a mass of 1034 kg and is held at -14.5° angle with respect to the horizon should approximately be $(3.3/2)\cos(-14.5\pi/180) \cdot 1034 \cdot 9.81 \approx 16.16$ kNm in steady-state. These allows us to conclude that the spatial vector algebra modeling approach provides physically accurate results and allows to take into account the arm and base interactions. This feature is very useful for the desing of controllers that could effectively reduce the disturbances produced by rough terrains on the arm.

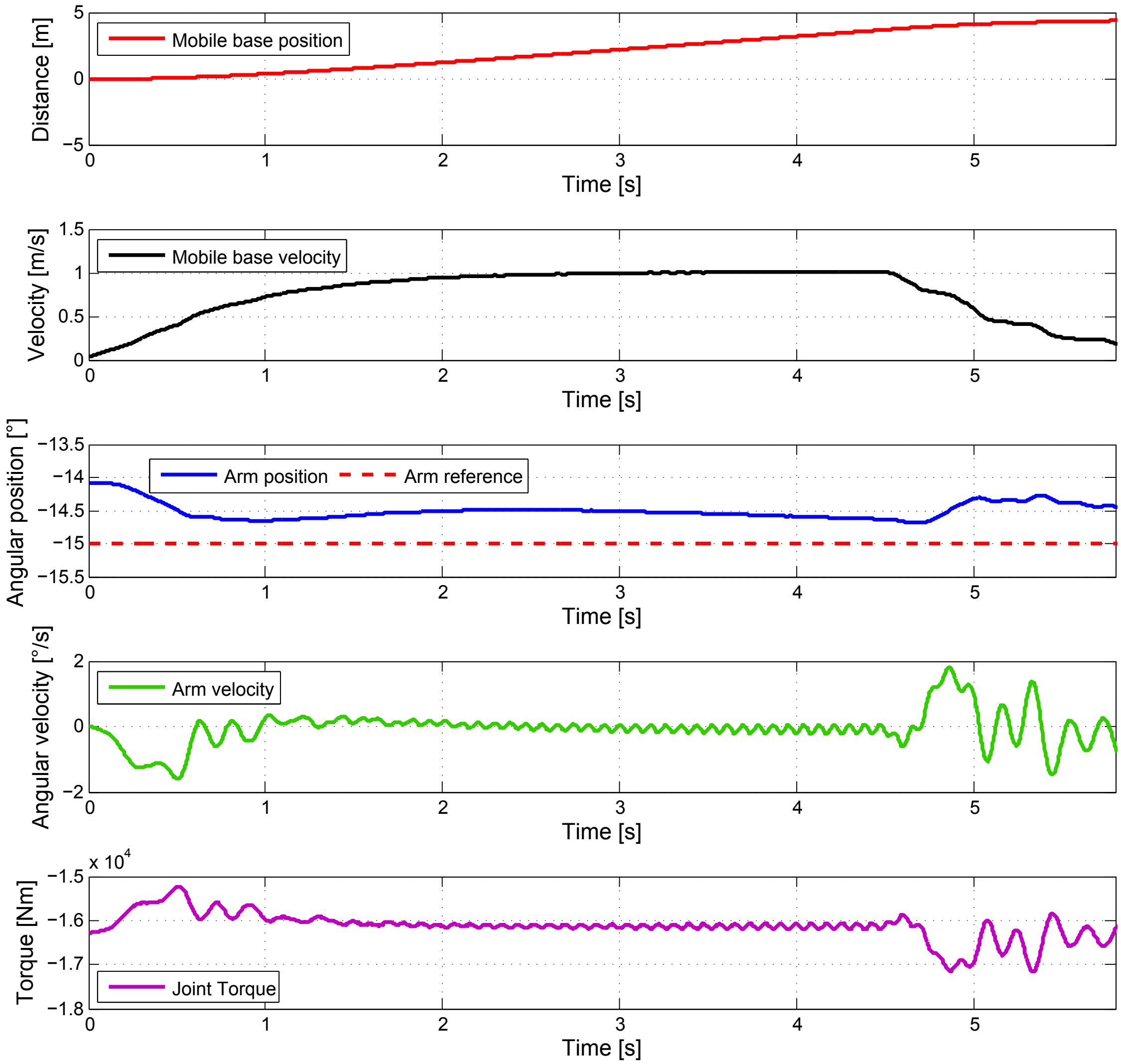


FIGURE 4.14. Closed-loop response of the SSMM's arm when the mobile base follows a trapezoidal velocity profile.

5. A MAPLE PACKAGE FOR SPATIAL VECTOR ALGEBRA SYMBOLIC COMPUTATIONS

An additional contribution of this research is the implementation of a Maple® package for spatial vector algebra computations and modeling of the motion dynamics equatios of multybody systems. This software tool was employed to check the derivation of the equations of motion for the SSMM that were presented in chapter 2. This software package for the symbolic manipulation and computation of spatial vector algebra expressions could also be useful to understand and learn how to work with spatial vector algebra. The complete code for this software tool is presented in the appendix A.2 and can be obtained from the website (*SSMM Model Files, Experimental Data, Simulation Videos and Spatial Vector Algebra Library*, 2014). A summary of the main functions of the software library for symbolic spatial vector algebra computations is presented in table 5.1 below.

An example showing how the developed software library for symbolic spatial vector algebra computations can be employed to obtain the model equations that were derived in 2 for the SSMM are presented next. For the sake of clarity, the example is divided into five sections: "Floating base model variables and parameters", "Basic body velocities", "External forces", "Inertial and force reactions" and "Body accelerations". The first section defines the model parameters, and is the main section that has to be updated or modified by the user to describe a different robot. In the example, the first section defines a floating base with 1-DOF manipulator. The remaining sections are rather general and can be executed to derive the forward dynamics equations of any system using Featherstone's Articulated Body Alogrithm provided the user has adequately described system in the first section and that the number of parameters does not exceed the amount of data/memory Maple® can handle. The variable $\mathbf{X}_{up}[i]$ is employed in the example code to define the motion transformation between the the parent body $\lambda(i)$ of body i, i.e. $\mathbf{X}_{up}[i]$ represents the motion transform ${}^{i}\mathbf{X}_{\lambda(i)}$ presented in table 2.5 of chapter 2.

TABLE 5.1. Maple implemented library.

Function	Inputs	Description
crm	$x \in \mathbb{R}^6$	Returns the cross-product operator for spatial motion vectors x.
crf	$x \in \mathbb{R}^6$	Returns the spatial cross-product operator for spatial forces x.
rq	$x \in \mathbb{R}^4$	Converts quaternions x to $3x3$ rotation matrices.
$skew$	$x \in \mathbb{R}^3$	Calculates the $3x3$ skew-symmetric cross-product matrix for vector x.
$plux$	$x \in \mathbb{R}^{3x3}$, $y \in \mathbb{R}^3$	Returns the Plücker coordinate transformation corresponding to a rotation matrix x and a translation y. If the motion transformation from frame A to B is parameterized by a rotation matrix ${}^B\mathbf{E}_A$ and a translation ${}^B\mathbf{r}_A$, this function returns ${}^B\mathbf{X}_A$. To make the code more compact, the notation $\mathbf{X}_{up}[i]$ is employed to define the motion transformation between the the parent body $\lambda(i)$ of body i, i.e. $\mathbf{X}_{up}[i]$ represents ${}^i\mathbf{X}_{\lambda(i)}$.
$rotx$ $roty$ $rotz$	$\theta \in \mathbb{R}$	Returns the spatial rotation matrix corresponding to a rotation θ radians about the x, y or z axis.
xtl	$x \in \mathbb{R}^3$	Returns the spatial translation matrix from A to B by a 3D vector x.
mcI	$m \in \mathbb{R}$, $c \in \mathbb{R}^3$, $I \in \mathbb{R}^{3,3}$	Returns the rigid body inertia computed in terms of the body mass m, the center of mass c, and the 3D inertia matrix I.

Direct Dynamics Algorithm

This file implements the algorithm for spatial vector algebra for direct dynamics presented in the book "Rigid Body Dynamics Algorithms" by Roy Featherstone.
The code implemented below corresponds to the Articulated Body Algorithm presented in Ch. 7, p. 132, table 7.1, applied to the Skid-Steer Mobile Manipulator with 1-DOF arm. The code can be modified and extended to implement the model of any other robot provided that the user has adequately defined the model parameters and that the number of parameters does not exceed the amount of data/memory that Maple can handle.

Remarks:
- The model presented below employs some simplifying assumptions like diagonal body inertia matrices and an arm location centered in the base in order to minimize the parameters required for the symbolic computations.
- The ABA algorithm for forward dynamics applied to the SSMM with 1-DOF arm has been writen in a general way. If additional DOFs for the arm or wheels are needed, the code only requires the addition of some extra parameters in the section "Floating base model variable and parameters".
- In order to properly execute the code, the "libname" variable containing the path to the library directory has to be updated.
- Any question concerning the algorithm implementation or further results may be sent to sfaguile@uc.cl.

```
> restart;
> libname := "C:/Users/Sergio Aguilera/Dropbox/Magister/MapleFuctions/lib", libname :
  with(LibrarySVA);
  with(LinearAlgebra) :
```

This file models a floating base with a 1-DOF manipulator attached to it.

Floating base model variable and parameters

First, the local variables are declared. Because of the the base is a 6-DOF floating base, six bodies connected by 1-DOF joints are employed to account for the 6-DOF. Following Roy Featherstone's approach, the six bodies with 1-DOF joints are combined together and their properties are stored in element 6 of the parameter vectors, while the manipulator is labeled as body 7.

```
> xfb := ⟨p[0], p[1], p[2], p[3], p[x], p[y], p[z], ω[x], ω[y], ω[z], v[x], v[y], v[z]⟩ :

      # xfb: is the state of the floating base in the global frame coordinates described by the
      orientation quaternion, its global position and spatial global velocity.
> vecq := ⟨q[7]⟩ :
   # q[1] is the joint position of the arm. For a rotary joint, it represent the rotation angle.
> vecqd := ⟨qd[7]⟩ :
  # qd[1] is the joint velocity.
> vecτ := ⟨τ[7]⟩ :
  # τ[7] : is the applied torque to the manipulator joint.
> vecm := ⟨m[6], m[7]⟩ :
  # m[6] and m[7] are the mass of the floating base and the manipulator, respectively.
> CoM[6] := ⟨0, 0, 0⟩ : CoM[7] := ⟨l[7]/2, 0, 0⟩ :
  #CoM is the center of mass location in the body frame coordinates.
```

```
> Icm[6] := MatrixScalarMultiply(Matrix(⟨Ixx[6], Iyy[6], Izz[6]⟩, shape = diagonal), 1) :
  Icm[7] := MatrixScalarMultiply(Matrix(⟨0, Iyy[7], 0⟩, shape = diagonal), 1) :
  #Icm is the 3D inertia of the body in body coordinates about its center of mass.
> for i from 6 by 1 to 7 do
   In[i] := mcI(vecm(i − 5), CoM[i], Icm[i]) :
   end do:
  #In[i] is the spatial inertia of the body i in body frame coordinates.
> #xtree[7] := xlt(⟨a7_1, a7_2, a7_3⟩) :
  xtree[7] := xlt(⟨0, 0, 0⟩) :
  #xtree is the translation matrix from body frame 6 to body frame 7.
> λ := ⟨0, 0, 0, 0, 0, 0, 6⟩ :
  #The i-the element of the lambda vector stores the index of the parent body of body i.
> qn := xfb[1 .. 4] :
  #qn is the orientation quaternion of the floating base.
> r := xfb[5 ..7] :
  #r is the global position of the floating base.
> vff := xfb[8 ..13] :
  #vff is the velocity of the floating base in the global frame.
> E := simplify(rq(qn)) :
  #E is the rotation matrix between the world frame and the floating base frame.
> Xup[6] := convert(plux(E, r), Matrix) :
  #Xup[6] is the transformation matrix from the global frame to the floating base frame. It
  depends on the orientation of the floating base and its location.
> Fext[6] := ⟨τ6extx, τ6exty, τ6extz, f6extx, f6exty, f6extz⟩ : Fext[7] := ⟨τ7extx, τ7exty, τ7extz,
      f7extx, f7exty, f7extz⟩ :
  #Fext[6] := ⟨0, 0, 0, 0, 0, 0⟩ : Fext[7] := ⟨0, 0, 0, 0, 0, 0⟩ :
  #Fext are the spatial forces corresponding to the external spatial forces.
> NB := 7 :
  #NB is the number of bodies (the 6-DOF floating base counts as 6 bodies and the 1-DOF
  manipulator counts as 1 body).
```

Basic body velocities

The direct dynamics algorithm takes as input the velocities of the system and the forces in the global frame. But for floating bases it is easier to work with velocities expressed in the floating base frame. Because of this, the velocity v[6] of the floating base expressed in the coordinates of the floating base frame is calculated from vff , but replaced by a vector of parameters that will be employed along the whole code.

```
> v[6] := simplify(MatrixVectorMultiply(Xup[6], vff)) :
  #Floating base velocity in the coordinate frame of the floating base.
> v[6] := ⟨P, Q, R, U, V, W⟩ :
  # A substitution is made to simplify the equations and mathematical procedures.
> pA[6] := MatrixVectorMultiply(MatrixMatrixMultiply(convert(crf(v[6]), Matrix), In[6]),
      v[6]) :
  #Coriolis and centrifugal forces due to the cross product between the velocity of the body and
  the inertia transformed velocity.
```

When the base velocities are calculated, the velocity of the base is propagated to the children, expressing the velocity in the coordinate frame of the children bodies and added to their joint velocity.
- v[i] is the velocity of the body i
- c[i] is the centrifugal force and Coriolis effects due to cross product between the joint velocity and the floating base velocity.
- pA[i] is the force of the body i, due to the velocity of body i and the inertia of the body.

```
> for i from 7 by 1 to NB do
  XJ := roty(vecq(i - 6)) :
   S[i] := ⟨0, 1, 0, 0, 0, 0⟩ :
   vJ := vecqd(i - 6)·S[i] :
   Xup[i] := MatrixMatrixMultiply(XJ, xtree[i]);
   v[i] := MatrixVectorMultiply(Xup[i], v[λ[i]]) + vJ;
   c[i] := MatrixVectorMultiply(crm(v[i]), vJ);
   pA[i] := MatrixVectorMultiply(MatrixMatrixMultiply(convert(crf(v[i]), Matrix), In[i]),
       v[i]);
   end do:
```

```
> v[7] : c[7] : pA[7] :
```

External forces

Once the forces due to the velocity of the bodies have been calculated, the external forces are added. The external forces are usually expressed in the global frame and have to be transform to the body frame.

-Xa is the transformation matrix from the global frame to the body i.

```
> Xa[6] := simplify(Xup[6]) :
> for i from 7 by 1 to NB do
   Xa[i] := simplify(MatrixMatrixMultiply(Xup[i], Xa[λ[i]])) :
   end do:
> for i from 6 by 1 to NB do
   #pA[i] := pA[i] - MatrixVectorMultiply(MatrixInverse(Transpose(Xa[i])), Fext[i]);
   pA[i] := pA[i] - Fext[i];
  end do:
```

Inertial and force reactions

Once each body forces are calculated in their own frame, these forces are propagated to their parents. In addition to the force, also the apparent inertia of the children is propagated back to the parent and added to the parent's own inertia.

- Ia is the apparent inertia transmited from i to lambda[i].
- pa is the force transmited from i to lambda[i].

```
> for i from NB by -1 to 7 do
  UU[i] := MatrixVectorMultiply(In[i], S[i]);
  d[i] := Multiply(Transpose(S[i]), UU[i]);
  u[i] := vecτ(i - 6) - Multiply(Transpose(S[i]), pA[i]);
```

$$Ia := In[i] - Multiply\left(\frac{UU[i]}{d[i]}, Transpose(UU[i])\right);$$

$$pa := pA[i] + MatrixVectorMultiply(Ia, c[i]) + \frac{UU[i]\cdot u[i]}{d[i]};$$

```
  In[λ[i]] := In[λ[i]] + MatrixMatrixMultiply(MatrixMatrixMultiply(Transpose(Xup[i]),
      Ia), Xup[i]);
  pA[λ[i]] := pA[λ[i]] + MatrixVectorMultiply(Transpose(Xup[i]), pa);

  end do:
```

Body accelerations

The last iteration of the algorithm calculates the accelerations of each body.
First, the acceleration of the floating base a[6] is calculated as the inverse of the updated inertia of the base postmultipled by the total applied force.
Similarly to velocities, body accelerations are propagated from each parent to its children.
Finally, the acceleration of the children is calculated addint the acceleration of the parent and the bodies own acceleration.

```
> a[6] := simplify(MatrixVectorMultiply(-MatrixInverse(In[6]), pA[6])) :
> for i from 7 by 1 to NB do
  a[i] := MatrixVectorMultiply(Xup[i], a[λ[i]]) + c[i];
```

$$qdd[i-6] := \frac{(u[i] - Multiply(Transpose(UU[i]), a[i]))}{d[i]};$$

```
  a[i] := a[i] + S[i]·qdd[i - 6];
  end do:
```

The accelerations of each body in its own frame can be easily transformed into an acceleration expressed in the global frame using the inverse of the transformation matrix Xa[i].

6. CONCLUSION AND FUTURE RESEARCH

6.1. Review of the Results and General Remarks

The forward dynamics equations for a generic skid-steer mobile manipulator considering the base as a 6-DOF floating base with non-permanent ground contacts was developed using the spatial vector algebra and the Articulated Body Algorithm. Unlike the existing models for mobile manipulators, the model developed provides explicit expressions for the arm-vehicle and the wheel-ground interactions. Due to the growing number of parameters when the arm includes several links, the model in this work is developed for an arm with only one degree-of-feedom. However, the approach can be extended to arms with several degrees-of-freedom, or mobile bases with more than one arm or wheels. The limitations to the symbolic modeling are in the number of parameters that can reasonably be dealt with in calculations by hand, or using a computer algebra system for symbolic manipulation of mathematical expressions.

Another contribution of this work is the development of a software library for symbolic spatial algebra computations in Maple®. This library can be employed to derive the dynamic model of traditional robot arms, mobile robot bases, an other multi-body systems whose topology may corresponds to a kinematic trees.

To validate the dynamic model of the SSMM, the model the simulation results were compared to measurements acquired from the inertial sensors installed on board of a Cat® 262C compact skid-steer loader. The accuracy between the model and the experimental data reassures the usefulness of the spatial vector formulation and the model built enriches the set of examples included with the Spatial Toolbox. Both the model and experimental data collected during the field tests have been made publicly available at (*SSMM Model Files, Experimental Data, Simulation Videos and Spatial Vector Algebra Library*, 2014) for the robotics community interested in studying the dynamics, motion control and mechanical design of SSMMs, among other related topics. Even though the model equations and the modeling approach has been validated for a specific machine, it should be easily adaptable to other machines given the generality of the approach and provided that the paramters in tables 2.5 and 3.1 are adequately set. In addition to providing a dynamic model for a compact skid-steer loader that can be employed to develop control strategies and trajectory tracking controllers that optimize the machine performance under terrain disturbances, this work shows that the spatial vector algebra formulation allows to obtain a unified model that takes into account the vehicle-terrain and arm-vehicle interaction in a single set of equations. This is difficult to do with traditional Newton-Euler o Lagrange methodologies without introducing problem specific terrain constraints that are more difficult to generalize to different mechanical systems or environments.

The work presented also improves the standard planar wheel-terrain contact model on a flat terrain and extends it so that the new model considers also the tangential wheel-ground traction and lateral skidding reaction forces arising under the assumption of a compliant deformable terrain. Furthermore, the terrain can be modeled as a piecewise continuous concatenation of planes.

6.2. Ongoing Research Topics

The motion dynamics equations derived for the SSMM are being employed in the design and development of motion controllers that can compensate or attenuate the disturbances propagated across the base to the arm due to terrain unevenness. Ongoing research efforts are also concerned with the development of state observers to estimate wheel slippage and identify changes in terrain type, which in turn are necessary to compute the optimal application of torque to a robot's wheels in order to reduce slippage. Another goal of our current research is to improve the capabilities of the developed software package for symbolic spatial algebra computations, in particular, the efforts are focused on the implementation of algorithms to model more complex closed-loop kinematic trees. The development of physically accurate models for automonous mobile manipulators is essential in the future development of autonomous robots, especially of load-haud and dump machines for the mining industry and crops and harvest handling robots.

References

Akhtarhusein A. Tayebali, B.-w. T., & Monismith, C. L. (1994). *Stiffness of Asphalt-Aggregate Mixes.*

Auat Cheein, F., & Carelli, R. (2013, September). Agricultural Robotics: Unmanned Robotic Service Units in Agricultural Tasks. *IEEE Industrial Electronics Magazine*, *7*(3), 48-58. Retrieved from http://dx.doi.org/10.1109/MIE.2013.2252957

Azad, M., & Featherstone, R. (2014, June). A New Nonlinear Model of Contact Normal Force. *IEEE Transaction on Robotics*, *30*(3), 736 - 740.

Baker, J., Draper, J., Pin, F., Primm, A., & Shekhar, S. (1996, May). Conceptual design of an aircraft automated coating removal system.. Retrieved from http://www.osti.gov/scitech/servlets/purl/234695

Brock, O., & Kemp, C. (2010). Guest editorial: special issue on autonomous mobile manipulation. *Autonomous Robots*, *28*(1), 1-3. Retrieved from http://dx.doi.org/10.1007/s10514-009-9147-6 doi: 10.1007/s10514-009-9147-6

Cheong, J. (2010). A consistent kinematic modeling method for mobile manipulators. In *Control Automation Robotics Vision (ICARCV), 2010 11th International Conference on* (p. 805-810). doi: 10.1109/ICARCV.2010.5707827

Cook, M. V. (2007). *Flight Dynamics Principles, Second Edition: A Linear Systems Approach to Aircraft Stability and Control* (2nd ed.). Butterworth-Heinemann. Retrieved from http://www.worldcat.org/isbn/0750669276

De Luca, A., Oriolo, G., & Robuffo Giordano, P. (2010, May). Kinematic Control of Nonholonomic Mobile Manipulators in the Presence of Steering Wheels. In *IEEE Int. Conf. on Robotics and Automation* (pp. 1792–1798). Anchorage, AS, USA.

Featherstone, R. (2008). *Rigid Body Dynamics Algorithms*.

Gacovski, Z. (2011). *Mobile Robots - Current Trends*. InTech. doi: 10.5772/2305

Helms, E., Schraft, R., & Hagele, M. (2002). rob@work: Robot assistant in industrial environments. In *Robot and Human Interactive Communication, 2002. Proceedings. 11th IEEE International Workshop on* (p. 399-404). doi: 10.1109/ROMAN.2002.1045655

Iagnemma, K., & Dubowsky, S. (2004). Rough Terrain Control. In *Mobile Robots in Rough Terrain* (pp. 81–96). Springer Berlin Heidelberg.

Kemp, C., Edsinger, A., & Torres-Jara, E. (2007, March). Challenges for robot manipulation in human environments [Grand Challenges of Robotics]. *Robotics Automation Magazine, IEEE*, *14*(1), 20-29. doi: 10.1109/MRA.2007.339604

Kozlowski, Krzysztof, Pazderski, & Dariusz. (2004). Modeling and control of a 4-wheel skid-steering mobile robot. *International Journal of Applied Mathematics and Computer Science*, *14*(4), 477-496. Retrieved from http://eudml.org/doc/207713

Lee, C. S. G. (1989). Introduction to Special Issue on Robot Manipulators: Algorithms and Architectures. *IEEE Transactions on Robotics and Automation*(5).

Li, Z., Ge, S. S., & Ming, A. (2007). Adaptive Robust Motion/Force Control of Holonomic-Constrained Nonholonomic Mobile Manipulators. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, *37*(3), 607-616. Retrieved from http://dblp.uni-trier.de/db/journals/tsmc/tsmcb37.html#LiGM07

Liu, Y., & Liu, G. (2009, November). Modeling of tracked mobile manipulators with consideration of track-terrain and vehicle-manipulator interactions. *Robot. Auton. Syst.*, *57*(11), 1065–1074. Retrieved from http://dx.doi.org/10.1016/j.robot.2009.07.007 doi: 10.1016/j.robot.2009.07.007

Luca, A. D., Oriolo, G., & Giordano, P. R. (2006). Kinematic Modeling and Redundancy Resolution for Nonholonomic Mobile Manipulators. In *ICRA* (p. 1867-1873). IEEE. Retrieved from http://dblp.uni-trier.de/db/conf/icra/icra2006.html#LucaOG06

Lynas, D., & Horberry, T. (2011). Human Factor Issues with Automated Mining Equipment. *The Ergonomics Open Journal*, *4*, 74-80. Retrieved from http://benthamscience.com/open/toergj/articles/V004/SI0061TOERGJ/74TOERGJ.pdf doi: 10.2174/1875934301104010081

Mandow, A., Martinez, J., Morales, J., Blanco, J.-L., Garcia-Cerezo, A., & Gonzalez, J. (2007). Experimental kinematics for wheeled skid-steer mobile robots. In *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on* (p. 1222-1227). doi: 10.1109/IROS.2007.4399139

Mirtich, B., & Mirtich, B. (1998). *Rigid Body Contact: Collision Detection to Force Computation* (Tech. Rep.). Mitsubishi Electric Research Laboratories.

Mohammadpour, E., Naraghi, M., & Gudarzi, M. (2010). Posture stabilization of Skid Steer Wheeled Mobile Robots. In *Robotics Automation and Mechatronics (RAM), 2010 IEEE Conference on* (p. 163-169). doi: 10.1109/RAMECH.2010.5513194

Petrov, P., de Lafontaine, J., Bigras, P., & Tetreault, M. (2000). Lateral control of a skid-steering mining vehicle. In *Intelligent Robots and Systems, 2000. (IROS 2000). Proceedings. 2000 IEEE/RSJ International Conference on* (Vol. 3, p. 1804-1809 vol.3). doi: 10.1109/IROS.2000.895233

Puga, J. P., & Chiang, L. E. (2008, 5). Optimal trajectory planning for a redundant mobile manipulator with non-holonomic constraints performing pushpull tasks. *Robotica*, *26*, 385–394. Retrieved from http://journals.cambridge.org/article_S0263574707004031 doi: 10.1017/S0263574707004031

Simpkins, C., & Simpkins, A. (2013, June). Towards Service Robots for Everyday Environments: Recent Advances in Designing Service Robots for Complex Tasks in Everyday Environments (E. Prassler et al.; 2012) [On the Shelf]. *Robotics Automation Magazine, IEEE*, *20*(2), 106-106. doi: 10.1109/MRA.2013.2255516

Spatial Vector Algebra Toolboox for Matlab. (2014, 1 February). http://royfeatherstone.org/spatial/index.html. Retrieved from http://royfeatherstone.org/spatial/index.html

SSMM Model Files, Experimental Data, Simulation Videos and Spatial Vector Algebra Library. (2014, 10 September). http://ral.ing.puc.cl/ssmm.htm.

Stentz, A., Bares, J., Singh, S., & Rowe, P. (1999, September). A Robotic Excavator for Autonomous Truck Loading. *Auton. Robots*, *7*(2), 175–186. Retrieved from http://dx.doi.org/10.1023/A:1008914201877 doi: 10.1023/A:1008914201877

Tan, J., Xi, N., & Wang, Y. (2003). Integrated task planning and control for mobile manipulators. *Int. J. Rob. Res*, 337–354.

Tanner, H., Kyriakopoulos, K., & Krikelis, N. (2001). Advanced agricultural robots: kinematics and dynamics of multiple mobile manipulators handling non-rigid material. *Computers and Electronics in Agriculture*, *31*(1), 91 - 105. Retrieved from http://www.sciencedirect.com/science/article/pii/S0168169900001769 doi: http://dx.doi.org/10.1016/S0168-1699(00)00176-9

Tanner, H. G., & Kyriakopoulos, K. J. (2001, 8). Mobile manipulator modeling with Kane's approach. *Robotica*, *19*, 675–690. Retrieved from http://journals.cambridge.org/article_S0263574701003381 doi: 10.1017/S0263574701003381

Torres-Torriti, M., & Michalska, H. (2005). A Software Package for Lie Algebraic Computations. *SIAM Review*, *47*(4), 722-745. Retrieved from http://link.aip.org/link/?SIR/47/722/1 doi: http://dx.doi.org/10.1137/S0036144502410427

White, G., Bhatt, R., Tang, C. P., & Krovi, V. (2009). Experimental Evaluation of Dynamic Redundancy Resolution in a Nonholonomic Wheeled Mobile Manipulator. *Mechatronics, IEEE/ASME Transactions on*, *14*(3), 349-357. doi: 10.1109/TMECH.2008.2008802

Wong, J. (1989). *Terramechanics and off-road vehicles*. Elsevier. Retrieved from http://books.google.cl/books?id=YHdTAAAAMAAJ

Wong, J. (2008). *Theory of Ground Vehicles*. John Wiley & Sons. Retrieved from http://books.google.cl/books?id=Blp2D1DteTYC

X. Zeng, J. R., & Rice, J. (2001). Stiffness and Damping Ratio of Rubber-Modified Asphalt Mixes: Potential Vibration Attenuation for High-Speed Railway Trackbeds. *Journal of Vibration and Control*. doi: 10.1177/107754630100700403

Yamamoto, Y., & Yun, X. (1992). Coordinating locomotion and manipulation of a mobile manipulator. In *Decision and Control, 1992., Proceedings of the 31st IEEE Conference on* (p. 2643-2648 vol.3). doi: 10.1109/CDC.1992.371337

Yamamoto, Y., & Yun, X. (1996). Effect of the dynamic interaction on coordinated control of mobile manipulators. *Robotics and Automation, IEEE Transactions on*, *12*(5), 816-824. doi: 10.1109/70.538986

Yi, J., Wang, H., Zhang, J., Song, D., Jayasuriya, S., & Liu, J. (2009). Kinematic Modeling and Analysis of Skid-Steered Mobile Robots With Applications to Low-Cost Inertial-Measurement-Unit-Based Motion Estimation. *Robotics, IEEE Transactions on*, *25*(5), 1087-1097. doi: 10.1109/TRO.2009.2026506

Yu, Q., & Chen, I.-M. (2002). A General Approach to the Dynamics of Nonholonomic Mobile Manipulator Systems. *Journal of Dynamic Systems, Measurement, and Control*, *124*, 512 - 521. doi: DOI:10.1115/1.1513178

Yu, W., Chuy, O., Collins, J., E.G., & Hollis, P. (2010). Analysis and Experimental Verification for Dynamic Modeling of A Skid-Steered Wheeled Vehicle. *Robotics, IEEE Transactions on*, *26*(2), 340-353. doi: 10.1109/TRO.2010.2042540

APPENDIXES

APPENDIXES A. ADDITIONAL RESOURCES

A.1. Explicit Floating Base Equations

$$
\mathbf{a}_{1c} = \begin{array}{l}
\Big[\left({I_{yz1}}^2 - I_{zz1}\, I_{yy1}\right)\left(\left(-\omega_{1z} I_{xy1} + \omega_{1y} I_{xz1}\right)\omega_{1x} + \left(-\omega_{1z} I_{yy1} + \omega_{1y} I_{yz1}\right)\omega_{1y} + \left(-\omega_{1z} I_{yz1} + \omega_{1y} I_{zz1}\right)\omega_{1z}\right) \\
\quad + \left(I_{xy1}\, I_{zz1} - I_{yz1}\, I_{xz1}\right)\left(\left(\omega_{1z} I_{xx1} - \omega_{1x} I_{xz1}\right)\omega_{1x} + \left(\omega_{1z} I_{xy1} - \omega_{1x} I_{yz1}\right)\omega_{1y} + \left(\omega_{1z} I_{xz1} - \omega_{1x} I_{zz1}\right)\omega_{1z}\right) \\
\quad + \left(I_{xz1}\, I_{yy1} - I_{xy1}\, I_{yz1}\right)\left(\left(-\omega_{1y} I_{xx1} + \omega_{1x} I_{xy1}\right)\omega_{1x} + \left(-\omega_{1y} I_{xy1} + \omega_{1x} I_{yy1}\right)\omega_{1y} + \left(-\omega_{1y} I_{xz1} + \omega_{1x} I_{yz1}\right)\omega_{1z}\right)\Big]\Delta_I^{-1} \\
\Big[\left(I_{xy1}\, I_{zz1} - I_{yz1}\, I_{xz1}\right)\left(\left(-\omega_{1z} I_{xy1} + \omega_{1y} I_{xz1}\right)\omega_{1x} + \left(-\omega_{1z} I_{yy1} + \omega_{1y} I_{yz1}\right)\omega_{1y} + \left(-\omega_{1z} I_{yz1} + \omega_{1y} I_{zz1}\right)\omega_{1z}\right) \\
\quad - \left(I_{zz1}\, I_{xx1} - {I_{xz1}}^2\right)\left(\left(\omega_{1z} I_{xx1} - \omega_{1x} I_{xz1}\right)\omega_{1x} + \left(\omega_{1z} I_{xy1} - \omega_{1x} I_{yz1}\right)\omega_{1y} + \left(\omega_{1z} I_{xz1} - \omega_{1x} I_{zz1}\right)\omega_{1z}\right) \\
\quad + \left(I_{yz1}\, I_{xx1} - I_{xy1}\, I_{xz1}\right)\left(\left(-\omega_{1y} I_{xx1} + \omega_{1x} I_{xy1}\right)\omega_{1x} + \left(-\omega_{1y} I_{xy1} + \omega_{1x} I_{yy1}\right)\omega_{1y} + \left(-\omega_{1y} I_{xz1} + \omega_{1x} I_{yz1}\right)\omega_{1z}\right)\Big]\Delta_I^{-1} \\
\quad\Big[- \left(-I_{xz1}\, I_{yy1} + I_{xy1}\, I_{yz1}\right)\left(\left(-\omega_{1z} I_{xy1} + \omega_{1y} I_{xz1}\right)\omega_{1x} + \left(-\omega_{1z} I_{yy1} + \omega_{1y} I_{yz1}\right)\omega_{1y} + \left(-\omega_{1z} I_{yz1} + \omega_{1y} I_{zz1}\right)\omega_{1z}\right) \\
\quad\quad + \left(I_{yz1}\, I_{xx1} - I_{xy1}\, I_{xz1}\right)\left(\left(\omega_{1z} I_{xx1} - \omega_{1x} I_{xz1}\right)\omega_{1x} + \left(\omega_{1z} I_{xy1} - \omega_{1x} I_{yz1}\right)\omega_{1y} + \left(\omega_{1z} I_{xz1} - \omega_{1x} I_{zz1}\right)\omega_{1z}\right) \\
\quad\quad - \left(I_{yy1}\, I_{xx1} - {I_{xy1}}^2\right)\left(\left(-\omega_{1y} I_{xx1} + \omega_{1x} I_{xy1}\right)\omega_{1x} + \left(-\omega_{1y} I_{xy1} + \omega_{1x} I_{yy1}\right)\omega_{1y} + \left(-\omega_{1y} I_{xz1} + \omega_{1x} I_{yz1}\right)\omega_{1z}\right)\Big]\Delta_I^{-1} \\
\qquad\qquad\qquad\qquad \omega_{1z}\, v_{1y} - \omega_{1y}\, v_{1z} \\
\qquad\qquad\qquad\qquad \omega_{1x}\, v_{1z} - \omega_{1z}\, v_{1x} \\
\qquad\qquad\qquad\qquad \omega_{1y}\, v_{1x} - \omega_{1x}\, v_{1y}
\end{array}
\tag{A.1}
$$

$$\mathbf{a}_{1ext} = \begin{bmatrix} -\left[n_{1ext\,x}\, I_{zz\,1}\, I_{yy\,1} - n_{1ext\,x}\, {I_{yz\,1}}^2 - n_{1ext\,y}\, I_{xy\,1}\, I_{zz\,1} + n_{1ext\,y}\, I_{yz\,1}\, I_{xz\,1} - n_{1ext\,z}\, I_{xz\,1}\, I_{yy\,1} + n_{1ext\,z}\, I_{xy\,1}\, I_{yz\,1}\right] \Delta_I^{-1} \\ \left[n_{1ext\,x}\, I_{xy\,1}\, I_{zz\,1} - n_{1ext\,x}\, I_{yz\,1}\, I_{xz\,1} - n_{1ext\,y}\, I_{zz\,1}\, I_{xx\,1} + n_{1ext\,y}\, {I_{xz\,1}}^2 + n_{1ext\,z}\, I_{yz\,1}\, I_{xx\,1} - n_{1ext\,z}\, I_{xy\,1}\, I_{xz\,1}\right] \Delta_I^{-1} \\ -\left[-n_{1ext\,x}\, I_{xz\,1}\, I_{yy\,1} + n_{1ext\,x}\, I_{xy\,1}\, I_{yz\,1} - n_{1ext\,y}\, I_{yz\,1}\, I_{xx\,1} + n_{1ext\,y}\, I_{xy\,1}\, I_{xz\,1} + n_{1ext\,z}\, I_{yy\,1}\, I_{xx\,1} - n_{1ext\,z}\, {I_{xy\,1}}^2\right] \Delta_I^{-1} \\ -f_{1ext\,x} m_1^{-1} \\ -f_{1ext\,y} m_1^{-1} \\ -f_{1ext\,z} m_1^{-1} \end{bmatrix} \tag{A.2}$$

$$\Delta_I = det(I) = I_{zz\,1}\, I_{xx\,1}\, I_{yy\,1} - I_{zz\,1}\, {I_{xy\,1}}^2 - {I_{xz\,1}}^2 I_{yy\,1} - {I_{yz\,1}}^2 I_{xx\,1} + 2\, I_{yz\,1}\, I_{xy\,1}\, I_{xz\,1} \tag{A.3}$$

A.2. Spatial Vector Algebra Library

Spatial Vector Algebra Library

```
> restart :
> # ---------------------------------------------------------------
  # Simple Package
  # Sergio Aguilera Marinovic (c) 07. AUG.2014
  # Previous version: 20.JUN.2014, 27. APR.2014
  # Last Updated: 07.AUG.2014
  # ---------------------------------------------------------------
  Libreria1 := module( )
  export crm, crf, rq, plux, roty, rotx, rotz, mcI, skew, xlt;
  local startup, shutdown;
  option package, load = startup, unload = shutdown,
        `Copyright (c) 07.VIII.2014 Sergio Aguilera Marinovic`;
  startup := proc( )
    with(LinearAlgebra) :
    printf("Startup procedure of SimplePack load the package 'LinearAlgebra'\n");
  end proc:
  shutdown := proc( )
    printf("Shutdown procedure of SimplePack does nothing!\n");
  end proc:


      # The funtion "crm" describes the special cross product for motion, defined and used by the
      spatial vector algebra.
  crm := proc(x)
    ⟨⟨0, x[3], -x[2], 0, x[6], -x[5]⟩|⟨-x[3], 0, x[1], -x[6], 0, x[4]⟩|⟨x[2], -x[1], 0, x[5],
     -x[4], 0⟩|⟨0, 0, 0, 0, x[3], -x[2]⟩|⟨0, 0, 0, -x[3], 0, x[1]⟩|⟨0, 0, 0, x[2], -x[1], 0⟩⟩;
  end proc:


      # The funtion "crf" describes the special cross product for force, defined and used by the
      spatial vector algebra.
  crf := proc(x)
    local a;
    a := crm(x);
    evalm(-transpose(a));
  end proc:

  # The funtion "rq" transform the quaternion into a 3x3 rotation matrix.
  rq := proc(x)
    local q0s, q1s, q2s, q3s, q01, q02, q03, q12, q13, q23, a, q, nq;
    #q := x / VectorNorm(x, 2, conjugate = false) :
    #The vector q must be normalize, but we assume that it comes normalize.
    q := x;
    q0s := q[1]*q[1];
    q1s := q[2]*q[2];
    q2s := q[3]*q[3];
    q3s := q[4]*q[4];
    q01 := q[1]*q[2];
```

```
q02 := q[1]*q[3];
q03 := q[1]*q[4];
q12 := q[2]*q[3];
q13 := q[4]*q[2];
q23 := q[3]*q[4];
a := ⟨⟨2· q0s + 2· q1s − 1, 2 · q12 -2 · q03, 2 · q13 +2 · q02 ⟩|⟨ 2 ·q12 +2 · q03, 2 · q0s
   +2 ·q2s − 1, 2 ·q23 -2 · q01⟩|⟨2 ·q13 -2 · q02, 2 · q23 + 2 ·q01, 2 · q0s +2 ·q3s − 1⟩⟩;
end proc:

#The funtion "skew" is the skew simetric matrix, defined by the given global location "i2".
skew := proc(i2)
⟨⟨0, i2[3],-i2[2]⟩|⟨-i2[3], 0, i2[1]⟩|⟨i2[2],-i2[1], 0⟩⟩;
end proc:


   # The funtion "plux" compose the Plucker transformation with the given rotation matrix
   "i1" and the global posicion "i2".
plux :=proc(i1, i2)
local a, b;
b := skew(i2);
a := MatrixMatrixMultiply( ( -i1 ), b ) ;
⟨⟨i1|⟨⟨0, 0, 0⟩|⟨0, 0, 0⟩|⟨0, 0, 0⟩⟩⟩, ⟨a|i1⟩⟩;
end proc:


   # The funtion "rotx" assemble the spatial rotation matrix along the x-axis with the given
   angle "θ".
rotx :=proc(θ)
⟨⟨1, 0, 0, 0, 0, 0⟩|⟨0, cos(θ), -sin(θ), 0, 0, 0⟩|⟨0, sin(θ), cos(θ), 0, 0, 0⟩|⟨0, 0, 0, 1, 0, 0⟩
   |⟨0, 0, 0, 0, cos(θ), -sin(θ)⟩|⟨0, 0, 0, 0, sin(θ), cos(θ)⟩⟩;
end proc:


   # The funtion "roty" assemble the spatial rotation matrix along the y-axis with the given
   angle "θ".
roty :=proc(θ)
⟨⟨cos(θ), 0, sin(θ), 0, 0, 0⟩|⟨0, 1, 0, 0, 0, 0⟩|⟨-sin(θ), 0, cos(θ), 0, 0, 0⟩|⟨0, 0, 0, cos(θ),
   0, sin(θ)⟩|⟨0, 0, 0, 0, 1, 0⟩|⟨0, 0, 0, -sin(θ), 0, cos(θ)⟩⟩;
end proc:


   # The funtion "rotz" assemble the spatial rotation matrix along the z-axis with the given
   angle "θ".
rotz :=proc(θ)
⟨⟨cos(θ), -sin(θ), 0, 0, 0, 0⟩|⟨sin(θ), cos(θ), 0, 0, 0, 0⟩|⟨0, 0, 1, 0, 0, 0⟩|⟨0, 0, 0, cos(θ),
   -sin(θ), 0⟩|⟨0, 0, 0, sin(θ), cos(θ), 0⟩|⟨0, 0, 0, 0, 0, 1⟩⟩;
end proc:

# The funtion "mcI" assemble the rigid body inertia matrix by the given mass "m", the center of
   mass "c" and 3x3 inertia matrix "I1"
```

```
mcI := proc(m, c, I1)
 local C, Caux, Iaux;
  C := skew(c);
  Caux := Transpose(C);
  Iaux := MatrixScalarMultiply(MatrixMatrixMultiply(C, Caux), m);
 ⟨⟨MatrixAdd(I1, Iaux), MatrixScalarMultiply(Caux, m)⟩|⟨MatrixScalarMultiply(C, m),
     MatrixScalarMultiply(Matrix(3, shape = identity), m)⟩⟩
 end proc:

# The funtion "xlt" assemble calculates the transform matrix from the frame A to B for spatial
     motion vectors, where "r" is the distance between the frames
xlt := proc(r)
 ⟨⟨1, 0, 0, 0, -r[3], r[2]⟩|⟨0, 1, 0, r[3], 0, -r[1]⟩|⟨0, 0, 1, -r[2], r[1], 0⟩|⟨0, 0, 0, 1, 0, 0⟩|⟨0,
     0, 0, 0, 1, 0⟩|⟨0, 0, 0, 0, 0, 1⟩⟩
 end proc


 end module:

> # Create/save the package.
libname := "C:/Users/Sergio Aguilera/Dropbox/Magister/MapleFuctions/lib", libname :
march('create', libname[1], 100) :
savelib('Libreria1') :
   # saves to the first path in libname unless savelibname is
   defined
```

A.3. SSMM Matlab model

```
% Skid-Steer Mobile Manipulator Model and Simulation Using the
% Spatial Toolbox for Rigid Body Dynamics
%
% Copyright (c) 2014.02.01
% Sergio Aguilera-Marinovic (sfaguile@uc.cl)
% Miguel Torres-Torriti (mtorrest@ing.puc.cl)
% Robotics and Automation Laboratory
% Pontificia Universidad Catolica de Chile
% http://ral.ing.puc.cl/ssmm.htm
%
% Version 2.0 - 2014.08.07
%
% Description
% This Matlab script shows how to build the model for an SSMM using the
% Spatial Toolbox for rigid body dynamics modeling developed by Roy
% Featherstone available at http://royfeatherstone.org.
%
% The model considers a floating base with four wheels and a simplfied
% 1-DOF arm. It is possible to easily add more degrees of freedom to the
% arm by copying the data structure for the 1-DOF arm and updating the
% parent link data where appropriate. The reason of implementing 1-DOF is
% to keep the code as simple and illustrative as possible, but more
% important, to replicate the dynamics of a Caterpillar CAT262C skid-steer
% compact loader.
%
```

```
% The model simulation results have been compared with IMU measuremnts
% obtained from experiments with a real CAT262C machine. The data from
% the experiments is available at http://ral.ing.puc.cl/ssmm.htm.
%
%
% Instructions
% 1. Prerequisites in addition to a standard installation of Matlab and
% Simulink is two download and setup the Spatial Toolbox version 2 by
% Roy Featherstone available at http://royfeatherstone.org.
%
% 2. Initialize the Spatial Toolbox with the command 'startup.m', which
% adds its installation path to Matlab's environment list of paths.
%
% 3. Change the directory to the location of the SSMM script files
% (this model and simulation files).
%
% 4. Run the SSMM_model.m file, it will create and store the SSMM model
% in variable 'model'. At the end of these code, few examples are
% presented
%
% 6. Run the Simulink simulation file SSMM_sim.slx. The output of the
% simulation is stored in the variable xout, which is a 23x1xN
% array structure. To reduce the singleton dimension you may execute
% res = squeeze(xout), which will store the results in variable res
% 23xN array.
%
% The 23 model state variables are the following:
% 1. The floating base varaibles:
%
```

```
% x = [x1 x2 x3 x4 x5 x6 x7 x8 x9 x10 x11 x12 x13]';
% |__________| |_______| |________| |__________|
% | | | |->Linear Velocity in
% | | | F1 Coordinates
% | | |
% | | |->Angular Velocity in
% | | F1 Coordinates
% | |
% | |->Position relative to F_0
% |
% |->Orientation Quaternion
%
% 2. The 1-DOF joint positions:
%
% q = [q1 q2 q3 q4 q5]'
% |_________| |
% | |->Arm joint position
% |
% |->Wheel joint positions
%
% 3. The 1-DOF joint velocities:
%
% qd = [qd1 qd2 qd3 qd4 qd5]'
% |______________| |
% | |->Arm joint velocity
% |
% |->Wheel joint velocities
%
% The full state vector with results of the simulation is stored in
% the variable xout and contains the previous vectors, as follows:
%
% xout = [x q qd]'
%
% 7. At the end of the simulation you may execute:
```

```
% res = squeeze(xout)
% plot(tout,res(11,:)) % plots the longitudinal velocity of the mobile
% % base
% plot(tout,res(5,:)) % plots the longitudinal displacement of the
% % mobile base
% plot(tout,res(18,:)) % plots the arm position
%
% The following command renders a 3D representation of the
% model and its motion using showmotion provided with the Spatial
% Toolbox:
%
% showmotion(model,tout,[fbanim(xout);squeeze(xout(14:18,:,:))])
%

% ============================ SSMM Model ============================

% The model script stars here...
%
clear all; clc; % Clear the workspace and erase the command window.

%--------------------- Reference Frame F0 ----------------------
% Draw the reference frame F_0 axes X_0, Y_0 and Z_0 using the "appearance"
% attribute to provide a visual reference in space
model.appearance.base = ...
  { 'colour', [0.9 0 0], 'line', [0 0 0; 2 0 0], ...
    'colour', [0 0.9 0], 'line', [0 0 0; 0 2 0], ...
    'colour', [0 0.3 0.9], 'line', [0 0 0; 0 0 2] };


%--------------------- SSMM Model ----------------------
%
```

```
% Store all model parameters in a variable called "model".
% model.NB: is the number of bodies in the model.
% Variable model.NB is initialized in zero and incremented whenever a new
% body is added to the model... This allows to easily expand any model!
model.NB = 0;


%---------------------Floating Base---------------------
%
% To create a floting base, a body (with frame F1) is added and "connected"
% to the reference frame F0 by any 1-DOF joint (e.g. rotary or prismatic),
% which will be later replaced by a full 6-DOF joint using the function
% floatbase. The 1-DOF joint is simply a temporary placeholder for the
% 6-DOF joint.
i=1; % This is the first body, and it's index is one.
model.NB=model.NB+1; % model.NB is updated with the new body.

model.jtype{i} = 'R'; % Any type of joint may be selected

model.parent(i) = 0 ; % The floating base parent is the fixed frame,
                  % i.e. \lambda(1) = 0

% The initial link-to-link transform from F_0 to F_1 is the identity,
% because at the initial state both frames are align.
model.Xtree{i} = xlt([0 0 0]);

mass(i) = 2389; % The mass of the CAT 262C is for about 1150 kg

```

```
% For modeling simplicity, the center of mass (COM) of the base is defined
% at the origin of frame F1.
CoM(i,:) = [0 0 0];

% The rotational inertia for the SSMM is approximated to the rotational
% inertia of a cube of size [3 1.6 1.2] with uniform density and total mass
% equal to mass(i).
Icm(i:i+2,:) = mass(i)*[1.6^2+1.2^2 0 0;...
                        0 3^2+1.2^2 0;...
                        0 0 3^2+1.6^2]/12;

% The mass, COM and Icm are employed to build the rigid-body's spatial
% inertia, which is stored in the model parameter "model.I{1}".
model.I{i} = mcI(mass(i), CoM(i,:), Icm(i:i+2,:));

% Define the floating base appearance to display/visualize the simulation
% results.
model.appearance.body{1} = {'colour', [250 137 45]/255, 'box', [-1.2 -0.8 -0.6; 1.8 0.8 0]...
                        'colour', [250 137 45]/255, 'box', [-1.2 -0.8 0; -0.3 0.8 0.6]...
                        'colour', [0.5 0.5 0.5], 'box', [ 0 -0.7 0; 1.7 0.7 0.85] };


%--------------------- Wheels ----------------------
% Wheels are the bodies 2, 3, 4 and 5.
for i = 2:5
model.NB=model.NB+1; % Update the body number for each wheel
```

```
% Wheels are modeled as rotary joints that rotate about the Y-axis of
% frames F2, F3, F4, and F5 (children of parent frame F1).
model.jtype{i} = 'Ry';

% The wheels' parent body is the floating base, i.e. \lambda(i) = 1
model.parent(i) = 1 ;

R = 0.45; % The wheels radius is .45 m
T = 0.25; % The thickness of the wheels is .25 m
density = 300; % THe density of each wheel is 300 kg/m^3

% As shown in the paper, each wheel has a different location and the
% link-to-link transform is only a translation of the wheel frame with
% to the body frame without rotation.
if(i==2)model.Xtree{i} = xlt([ 1 -0.8+T/4 -0.5]);end
if(i==3)model.Xtree{i} = xlt([-0.2 -0.8+T/4 -0.5]);end
if(i==4)model.Xtree{i} = xlt([ 1 0.8-T/4 -0.5]);end
if(i==5)model.Xtree{i} = xlt([-0.2 0.8-T/4 -0.5]);end


% The mass of the wheels equals to the volumen of the wheel times the
% density
mass(i) = pi*R^2*T*density;

% Because of the wheel clindrical shape the COM is located at the origin
% of their frame which also coincides with the geometrical centroid of the
% wheel.
CoM(i,:) = [0 0 0];

% The wheels are modeled as solid cylinders of radius of .45 m and
```

```matlab
% thickness of .25 m for the purpose of calculating and approximation to
% their rotational inertia.
Ibig = mass(i)*R^2/2;
Ismall = mass(i)*(3*R^2 +T^2)/12;

% The spatial rigid-body inertia is calculated for each wheel using the
% mass, COM and inertia tensor.
model.I{i} = mcI(mass(i),CoM(i,:),diag([Ismall Ibig Ismall]));

% Define the wheels' visual representation attributes. Also, a red dot is
% added to each wheel as a visual reference, to appreciate the turn of the
% wheels
model.appearance.body{i} = { 'colour', [0.1 0.1 0.1],...
                        'facets', 32,...
                        'cyl', [0 -T/2 0; 0 T/2 0], R,...
                        'colour', [0.8 0.1 0.1],...
                        'cyl', [0 -T/2-2e-3 -0.3; 0 T/2+2e-3 -0.3],0.05 };
end


%--------------------- Manipulator ----------------------
% This is a 1-DOF simplified arm, so only 1 joint will be defined.
% However, the next block of code can be copied to create additional joints
% and add DOFs as needed.

i=6; % The arm is the body number six.

model.NB=model.NB+1; % The model's body count is updated.
```

```
% For the CAT 262C skid-steer loader bucket arm movement is simplfied in
% this model to a rotary joint that rotates about the Y-axis of frame F6.
% While the actual machine has hydraulic (linear) pistons that actuate on
% the bucket arm, the pistons are hinged and the arm is in fact attached
% to rotary joint on a hinged mechanism with multiple bars. This mechanism
% allows the pivoting point to move a little forward/backward when the
% the arm is fully up/down. This feature has been left out of the model
% to keep it simple and pedagogical, but can be modeled defining the
% additional linking bodies and passive (i.e. non-actuated) joints to
% create a closed-chain type of four-bar mechanism.
model.jtype{i} = 'Ry';

% The manipulator parent is also the floating base, so the sixth element of
% the parent array is set to 1 (the floating base), i.e. \lambda(6) = 1.
model.parent(i) = 1;

% The link-to-link transform for the manipulator arm corresponds to a
% translation of frame F6 by [-1.2 0 0.6] relative to frame F1. This
% translation positions the arm at the top-rear of the CAT 262C.
% The initial joint position with q6=0 is defined to be the position where
% the arm's end-effector (bucket) is almost touching the ground.
% To this end, a 15 rotation about the Y-axes of F6 is applied.
model.Xtree{i} = roty(15*pi/180)*xlt([-1.2 0 0.6]);
```

```
l_arm(i) = 3.3; % The arm's length
mass(i) = 1034; % The arm's mass
r = 0.15; % The arm is modeled as a cylinder of radius r

% The center of mass of the arm is located at the mid-point of the
    arm's
% length.
CoM(i,:) = l_arm(i)*[0.5,0,0];


% The inertia matrix of the arm is approximated to that of a solid
    cylinder
% (like in the case of the wheels), and is given by
Icm(i+3:i+5,:) = mass(i)*[ r^2*6 0 0;...
                      0 r^2*3+l_arm(i)^2 0;...
                      0 0 r^2*3+l_arm(i)^2]/12;

% The spatial rigid-body inertia of the arm link is calculated using
    the
% mass, the COM and the inertia matrix:
model.I{i} = mcI(mass(i), CoM(i,:), Icm(i+3:i+5,:));

% Finally, define the arm's visual representation attributes to
    ressemble
% that of a compact skid-steer loader.
rotation = roty(-15*pi/180);
rot = rotation(1:3,1:3);
model.appearance.body{i} = ...
     { 'colour', [0 0 1],...
       'cyl', [0 0 0; 3.3 0 0], 0.15};
% 'colour', [250 137 45]/255,...
% 'cyl', [0 0.9 0; 1.3 0.9 0]*rot, 0.15, ...
% 'cyl', [1.2 0.9 0; 3.25 0.9 -0.6]*rot, 0.15, ...
```

```
% 'cyl', [3.15 0.9 -0.5; 3.15 0.9 -0.9]*rot, 0.15, ...
% ...
% 'cyl', [0 -0.9 0; 1.3 -0.9 0]*rot, 0.15, ...
% 'cyl', [1.2 -0.9 0; 3.25 -0.9 -0.6]*rot, 0.15, ...
% 'cyl', [3.15 -0.9 -0.5; 3.15 -0.9 -0.9]*rot, 0.15, ...
% ...
% 'cyl', [0 -1.05 0; 0 1.05 0]*rot, 0.18, ...
% 'colour', [0.1 0.1 0.1],...
% 'cyl', [3.15 -1 -0.9; 3.15 1 -0.9]*rot, 0.1,...
% 'box', [3.15 -1.1 -0.95;3.6 1.1 -0.85]*rot};

% To see the idealized (cylindrical) model arm, instead of an arm that
% visually ressembles that of the compact skid-steer loader, comment the
% previous appearance attributes and uncomment the folowing line:
% model.appearance.body{i} = { 'cyl', [0 0 0; 3.3 0 0], 0.15};


%--------------------- Additional Model Parameters ----------------------
% The default gravity is zero, so it must be defined as:
model.gravity = [0 0 -9.8];

% Once each body in the model has been defined, the first body must be
% turned into a floating base:
model = floatbase(model);
% By doing this, the body 1 has now 6 DOFs and can be tought as if it would
% be formed by the composition of 6 bodies each having a 1-DOF joint. So
% the first six joint variables belong to body 1, while the wheels which
```

```
% to be associated to bodies/frames 2, 3, 4, 5 are now bodies/frames 7, 8,
% 9, 10. Similarly, the arm body 6 is now boy 11, and the model variable
% model.NB is now valued model.NB+5.

%--------------------- Contact Points (CPs) ----------------------
% The contact points are defined as point that cannot penetrate the ground
% plane defined as the plane z=0 in the frame F0. Each contact point
% contains information about the body to which it belongs and its location
% in the body's reference frame.

% Floating Base CPs --------------------------------------------
CP_Base =[-1.2 1.8 -1.2 1.8 -1.2 1.8 -1.2 1.8;... X parameter of each CP
       -0.8 -0.8 0.8 0.8 -0.8 -0.8 0.8 0.8;... Y parameter of each CP
       -0.6 -0.6 -0.6 -0.6 0.6 0.6 0.6 0.6]; %Z parameter of each CP

% The body number of each CP
CP_Base_Body_Labels = 6*ones(1,length(CP_Base));

% Total number of CPs for the floating base
CP_Base_Num = length(CP_Base_Body_Labels);


% Wheels' CPs---------------------------------------------------
% Because of the wheels' simmetry, all CPs are located equidistant one
% from another about the perimeter of each wheel.
npt_1 = 32; % 32 CPs per wheel are been modeled

```

```
% The position for each CP on a wheel is calculated next.
ang = (0:npt_1-1) * 2*pi / npt_1;
Y = ones(1,npt_1) * T/2;
X = sin(ang) * R;
Z = cos(ang) * R;

CP_Wheel = [ X;...
           Y;...
           Z ];

% A contact point at the center of each wheel is added just to extract the
% position and velocity of each of the wheels.
CP_Wheel = [CP_Wheel [0;0;0]];

% Define the corresponding body for the wheels' CPs
CP_Wheel_1_Body_Labels = 7*ones(1,length(CP_Wheel(1,:)));
CP_Wheel_2_Body_Labels = 8*ones(1,length(CP_Wheel(1,:)));
CP_Wheel_3_Body_Labels = 9*ones(1,length(CP_Wheel(1,:)));
CP_Wheel_4_Body_Labels = 10*ones(1,length(CP_Wheel(1,:)));

% The CPs of all four wheels are store in a single variable:
CP_Wheels = [CP_Wheel CP_Wheel CP_Wheel CP_Wheel];

% All the corresponding body label for the wheels' CPs are also stored
% in a single variable:
%Wheels_Parent = [Cuerpo_wheel_1 Cuerpo_wheel_2 Cuerpo_wheel_3 Cuerpo_wheel_4];
CP_Wheels_Body_Labels = [CP_Wheel_1_Body_Labels CP_Wheel_2_Body_Labels...
                    CP_Wheel_3_Body_Labels CP_Wheel_4_Body_Labels];

% Total number of wheel contact points
```

```
CP_Wheels_Num = length(CP_Wheel_1_Body_Labels)*4;


% Manipulator CP -----------------------------------------------
% Only one CP is defined at end of the arm where the end-effector (
    bucket
% of the skid-steer loader) is located
CP_Arm = [l_arm(6);...
        0;...
        0 ];

% The arm contact point belogs to body 11
%Arm_Parent
CP_Arm_Body_Labels = 11*ones(1,length(CP_Arm(1,:)));

% Total number of arm contact points
CP_Arm_Num = length(CP_Arm_Body_Labels);

%------------------------- Model Format ------------------------
% All the contact points and parents position previously defined
% are put on the Spatial Toolbox format as shown below
model.gc.point = [CP_Base CP_Wheels CP_Arm];
model.gc.body = [CP_Base_Body_Labels CP_Wheels_Body_Labels
    CP_Arm_Body_Labels];

% The simulation in Simulink needs some auxiliar variables that
    define
% the starting index of the wheels's CPs within the general CP array
% (model.gc.point) for each wheel separately.
CP_Wheel_1_Index = length(CP_Base_Body_Labels)+1;
CP_Wheel_2_Index = length(CP_Base_Body_Labels)+length(
    CP_Wheels_Body_Labels)/4+1;
CP_Wheel_3_Index = length(CP_Base_Body_Labels)+length(
    CP_Wheels_Body_Labels)*2/4+1;
```

```
CP_Wheel_4_Index = length(CP_Base_Body_Labels)+length(CP_Wheels_Body_Labels)*3/4+1;
CP_Wheels_Final_Index = length(CP_Base_Body_Labels)+length(CP_Wheels_Body_Labels)*4/4;
% Auxiliar variables are declared to store the total number of CPs in the
% simulation considering the external forces and without including the
% external forces:
CP_Num = CP_Base_Num + CP_Arm_Num + CP_Wheels_Num;
CP_Num_aux = CP_Base_Num + CP_Arm_Num + CP_Wheels_Num;
%-------------------- Initialization -------------------------------
% Finally, the initial condition is declared:
x_init = [1 0 0 0 0 0 0.95 0 0 0 0 0 0]';
% |_______| |_________| |______| |_____|
% | | | |->Linear Velocity in F_1
% | | | Coordinates
% | | |
% | | |->Angular Velocity in F_1 coordinates
% | |
% | |->Position relative to F_0
% |
% |->Orientation Quaternion
%
% q_init(1:4) contains the wheels' initial position and q_init(5) contains
% the arm's initial position
q_init = [0 0 0 0 -30*pi/180]';
```

```
% qd_init(1:4) contains the wheels' initial velocities and q_init(5)
    contains
% the arm's initial velocity
qd_init = [0 0 0 0 0]';



% Example 1: Straight motion without load
Instants = [-7 0.5 0.9 5.1 5.3 6.5]+7;
Values = [0 0 1 1 0 0];
Accelerator = 0.3;
Linear = 1;
Turn = 0;
Extra_weight = [0;0;0];

% % Example 2: On-place turn motion without load
% Instants = [-1 0.5 0.9 10.8 11 11.5]+1;
% Values = [0 0 1 1 0 0];
% Accelerator = 0.3;
% Linear = 0;
% Turn = 1;
% Extra_weight = [0;0;0];

% % Example 3: Straight motion with load
% Instants = [-1 0.5 0.7 5.5 5.7 6.5]+1;
% Values = [0 0 1 1 0 0];
% Accelerator = 0.3;
% Linear = 1;
% Turn = 0;
% Extra_weight = [0;0;400];

% Example 4: In-place turn motion with load
% Instants = [-3 0.5 0.7 13.8 14.8 15.5]+3;
% Values = [0 0 1 1 0 0];
```

```
% Accelerator = 0.3;
% Linear = 0;
% Turn = 1;
% Extra_weight = [0;0;400];

% % Example 5: Circular motion
% Instants = [-5 0.5 0.9 10.8 11 11.5]+5;
% Values = [0 0 1 1 0 0];
% Accelerator = 0.4;
% Linear = 0.3;
% Turn = 0.7;
% Extra_weight = [0;0;0];
```