NATIONAL TECHNICAL UNIVERSITY OF ATHENS
SCHOOL OF MECHANICAL ENGINEERING
Department of Mechanical Design and Control Systems
Control Systems Lab

Diploma thesis

MODELLING AND SIMULATION OF A NON-HOLONOMIC WHEELED-LEGGED EXCAVATOR BASED ON MOBILE

Dimitrios Koklas

Supervising Professors: Evangelos Papadopoulos (NTUA), Andres Kecskemethy (UDE)

ATHENS 2019

Abstract

Mobile robots have tremendous variety of capabilities and mobility. To achieve their mobility, robots must be designed in such way that they are able to adapt to their current environment. The vast majority of Earth's surface is covered by uneven, slippery or muddy terrains. The wheeled-legged robots have huge adaptability advantages because they can change their locomotion method according to the current terrain thanks to their design. Therefore, they can traverse both hard and even terrains by alternating between legged and wheeled locomotion accordingly. Firstly, this thesis, presents the kinematics of a model of a wheeled-legged excavator both on the object-oriented programming package, MOBILE, and analytically. The non-holonomic constraints that govern the robot and how these affect its steering is studied and analysed, simulated on MOBILE, and verified mathematically. Finally, the dynamics of the system are presented and simulated on MOBILE.

Περίληψη

Τα ρομπότ που έχουν τη δυνατότητα κίνησης στο χώρο παρουσιάζουν τεράστια ποικιλομορφία δυνατοτήτων. Για να επιτευχθεί η κίνηση τους στο χώρο πρέπει να είναι σχεδιασμένα έτσι ώστε να προσαρμόζονται στο εκάστοτε περιβάλλον. Η πλειονότητα της επιφάνειας της Γης καλύπτεται από ανομοιόμορφα, ολισθηρά και λασπώδη εδάφη. Τα ρομπότ που συνδυάζουν πόδια και ρόδες έχουν σημαντικό προσαρμοστικό πλεονέκτημα, έναντι άλλων ρομπότ, καθώς μπορούν, λόγω του σχεδιασμού τους, να εναλλάσουν τη μέθοδο κίνησής τους ανάλογα με το έδαφος στο οποίο βρίσκονται. Συνεπώς, μπορούν να κινηθούν σε ανομοιόμορφα εδάφη, όπως αυτά που αναφέρθηκαν παραπάνω, αλλά και σε ομοιόμορφα εναλλάσοντας το τρόπο κίνησης τους μεταξύ ποδιών και τροχών ανάλογα. Αρχικά, αυτή η διπλωματική εργασία, παρουσιάζει την κινηματική ενός μοντέλου εκσκαφέα που χρησιμοποιεί πόδια και ρόδες, τόσο στο πακέτο αντικειμενοστρεφούς προγραμματισμού, MOBILE, αλλά και αναλυτικά. Στη συνέχεια αναλύονται οι μη-ολονομικοί περιορισμοί που διέπουν το σύστημα και το πώς αυτοί επηρεάζουν την κατεύθυνση του οχήματος, με προσομοιώσεις στο MOBILE και μαθηματική επαλήθευση. Τέλος, παρουσιάζεται η δυναμική του συστήματος προσομοιωμένη στο MOBILE.

Ευχαριστίες

Στην οικογένειά μου και στην κοπέλα μου Δήμητρα
που με στήριξαν σε αυτήν την προσπάθεια.

Table of Contents

Abstract **3**

Περίληψη **5**

Table of Contents **7**

List of figures **8**

List of tables **13**

1 Introduction **16**

1.1 Motivation 16

1.2 Related work and state of the art 17

1.3 Thesis structure 24

2 Elements of MOBILE **26**

2.1 Introduction to MOBILE 26

2.2 The concept of kinetostatic transmission element 28

3 Kinematics **31**

3.1 Basic kinematic model and MOBILE implementation 31

3.2 Inverse kinematics 34

3.3 MOBILE iterative method for inverse kinematics 35

3.4 Analytical verification of the inverse kinematics 48

4 Non-holonomic constraints **79**

4.1 Calculation of the dependent velocities on MOBILE 81

4.2 Analytical verification of MOBILE's dependent velocities values 84

5 Steering behaviour **103**

5.1 For dependent variables $\dot{y}_C$ and $\dot{\varphi}_{yaw}$ 103

5.2 For dependent variables θ_{l5}, θ_{r5} 111

6 Dynamics **119**

6.1 Calculation of the dependent accelerations on MOBILE 119

6.2 Adding mass properties to the robot 121

6.3 Inverse dynamics of the mechanism 128

6.4 Simulation results 132

7 Conclusions and Future Work **146**

7.1 Conclusions 146

7.2 Future Work 147

8 Bibliography **148**

List of figures

Figure 1-1: The MAMMOTH rover: An RGB-D sensor used to collect point clouds of the surrounding terrain is mounted on the top of the rover's mast [22]. 17

Figure 1-2: Sherpa stepping onto a high obstacle. The manipulator was used to support the rover while lifting the wheel onto the obstacle [5]. 17

Figure 1-3: Hylos robot [1], [10]. 18

Figure 1-4: Overview of the Justin mobile platform system [9]. 18

Figure 1-5: DRC-Hubo+ is a versatile humanoid that won first place at the DARPA Robotics Challenge in 2015. The robot can use tools, open doors, and even drive a vehicle. It can transform itself from a walking biped to a wheeled machine [19]. 19

Figure 1-6: Momaro is a mobile robot that can carry out sensing and manipulation tasks in disaster sites and other harsh environments. It was designed to compete in the DARPA Robotics Challenge, coming in fourth place [16]. 19

Figure 1-7: The legged-wheeled centaur-like robot, CENTAURO [15], [17], [18]. 20

Figure 1-8: Tri-ATHLETE robot, a six-legged robot developed at the NASA's Jet Propulsion Laboratory in southern California [11]. 20

Figure 1-9: The hybrid quadruped/wheeled robot that is developed by the Control Systems Lab of the National Technical University of Athens in assosiation with the University of Duisburg-Essen. 21

Figure 1-10: Hyundai's Ultimate Mobility Vehicle concept, Elevate [13]. 22

Figure 1-11: Scaled proof of concept for Hyundai's Elevate [13]. 22

Figure 1-12: Boston Dynamic's Handle [3], [4]. 23

Figure 1-13: Kaiser's S12 ALLROAD. 24

Figure 1-14: Menzi Muck A91. 24

Figure 2-1: Objects in multibody systems. 28

Figure 2-2: Elementary kinetostatic transmission element (from Kecskemethy (1993)). 29

Figure 3-1: Kinetostatic skeleton of the hybrid excavator. 31

Figure 3-2: The three prismatic and three revolute joints that model the cabin's centre position and rotation with respect to the ground-fixed frame K0. 32

Figure 3-3: The vehicle's cabin attached to the position and rotation joints ,that connect the cabin-fixed and the ground-fixed frames, made earlier. 33

Figure 3-4: The excavator's front legs attached to the cabin. 33

Figure 3-5: The kinetostatic skeleton implemented on MOBILE. 34

Figure 3-6: Forward and inverse kinematics relationship. 34

Figure 3-7: Tree type and closed-loop systems illustration. 35

Figure 3-8: The three prismatic joints in series that will model the left leg's position with respect to the ground-fixed frame K_0. 38

Figure 3-9: Scalar measurements idea. 39

Figure 3-10: Geometrical entities involved in the measurements between points and planes. 39

Figure 3-11: Basic geometric types of scalar measurements. 39

Figure 3-12: The characteristic measurements for the left's leg loop closure. 40

Figure 3-13: The inverse kinematics of the left leg are complete and by changing the independent variables we can observe the dependent values needed for that change. 41

Figure 3-14: The three prismatic joints in series that will model the right leg's position with respect to the ground-fixed frame K_0. 41
Figure 3-15: The characteristic measurements for the right's leg loop closure. 42
Figure 3-16: The inverse kinematics of the right leg are complete and by changing the independent variables we can observe the dependent values needed for that change. 43
Figure 3-17: The prismatic joint that models the left wheel's height (z) with respect to the ground-fixed frame K_0. 43
Figure 3-18: The characteristic measurements for the left wheel's loop closure. 44
Figure 3-19: The inverse kinematics of the left wheel are complete and by changing the independent variables we can observe the dependent values needed for that change. 45
Figure 3-20: The prismatic joint that models the right wheel's height (z) with respect to the ground-fixed frame K_0. 45
Figure 3-21: The characteristic measurements for the right wheel's loop closure. 46
Figure 3-22: The inverse kinematics of the right wheel are complete and by changing the independent variables we can observe the dependent values needed for that change. 47
Figure 3-23: The variables of the kinetostatic skeleton of the hybrid excavator. 48
Figure 3-24: Frames distribution to the components regarding the left leg tip distance from the inertia frame K0. 48
Figure 3-25: 2D diagram of the left leg tip distance from the inertia frame K0. 49
Figure 3-26: 2D diagram describing the geometry of the wheel levers and wheels. 49
Figure 3-27: Vehicle's cabin geometry. 50
Figure 3-28: The robot's geometrical values. 50
Figure 3-29: MOBILE's ψ_{l2}, ψ_{l3} and ψ_{l4} values for x_{Fl} range [1.80, 4.30]. 54
Figure 3-30: MATLAB calculated values for x_{Fl} from MOBILE's output ψ_{l2}, ψ_{l3} and ψ_{l4} values. x_{Fl} to ψ_{l2} diagram. 54
Figure 3-31: xFl, yFl and zFl errors calculated as (xFl-xFlcalculated), (yFl-yFlcalculated) and (zFl-zFlcalculated) accordingly when the varying value is xFl. 55
Figure 3-32: x_{Fl}, y_{Fl} and z_{Fl} errors calculated as (x_{Fl}-$x_{Flcalculated}$), (y_{Fl}-$y_{Flcalculated}$) and (z_{Fl}-$z_{Flcalculated}$) accordingly when the varying value is y_{Fl}. 56
Figure 3-33: x_{Fl}, y_{Fl} and z_{Fl} errors calculated as (x_{Fl}-$x_{Flcalculated}$), (y_{Fl}-$y_{Flcalculated}$) and (z_{Fl}-$z_{Flcalculated}$) accordingly when the varying value is z_{Fl}. 57
Figure 3-34: x_{Fl} error calculated as (x_{Fl}-$x_{Flcalculated}$) when the varying value is ψ_{l1}. 58
Figure 3-35: Frames distribution to the components regarding the right leg tip distance from the inertia frame K0. 59
Figure 3-36: 2D diagram of the right leg tip distance from the inertia frame K0. 59
Figure 3-37: x_{Fr}, y_{Fr} and z_{Fr} errors calculated as (x_{Fr}-$x_{Frcalculated}$), (y_{Fr}-$y_{Frcalculated}$) and (z_{Fr}-$z_{Frcalculated}$) when the varying value is x_{Fr} 62
Figure 3-38: x_{Fr}, y_{Fr} and z_{Fr} errors calculated as (x_{Fr}-$x_{Frcalculated}$), (y_{Fr}-$y_{Frcalculated}$) and (z_{Fr}-$z_{Frcalculated}$) when the varying value is y_{Fr} 63
Figure 3-39: x_{Fr}, y_{Fr} and z_{Fr} errors calculated as (x_{Fr}-$x_{Frcalculated}$), (y_{Fr}-$y_{Frcalculated}$) and (z_{Fr}-$z_{Frcalculated}$) when the varying value is z_{Fr} 64
Figure 3-40: x_{Fr}, y_{Fr} and z_{Fr} errors calculated as (x_{Fr}-$x_{Frcalculated}$), (y_{Fr}-$y_{Frcalculated}$) and (z_{Fr}-$z_{Frcalculated}$) when the varying value is ψ_{r1}. 65
Figure 3-41: Frames distribution to the components regarding the rear left wheel end frame distance from the inertia frame K0. 66

Figure 3-42: 2D diagram of the left rear wheel end frame distance from the inertia frame K0. 67
Figure 3-43: x_{Wl}, y_{Wl} and z_{Wl} errors calculated as (x_{Wl}-$x_{Wlcalculated}$), (y_{Wl}-$y_{Wlcalculated}$) and (z_{Wl}-$z_{Wlcalculated}$) when the varying value is z_{Wl}. 70
Figure 3-44: x_{Wl}, y_{Wl} and z_{Wl} errors calculated as (x_{Wl}-$x_{Wlcalculated}$), (y_{Wl}-$y_{Wlcalculated}$) and (z_{Wl}-$z_{Wlcalculated}$) when the varying value is θ_{l5}. 71
Figure 3-45: Frames distribution to the components regarding the rear right wheel end frame distance from the inertia frame K0. 72
Figure 3-46: 2D diagram of the right rear wheel end frame distance from the inertia frame K0. 73
Figure 3-47: x_{Wr}, y_{Wr} and z_{Wr} errors calculated as (x_{Wr}-$x_{Wrcalculated}$), (y_{Wr}-$y_{Wrcalculated}$) and (z_{Wr}-$z_{Wrcalculated}$) when the varying value is z_{Wr}. 76
Figure 3-48: x_{Wr}, y_{Wr} and z_{Wr} errors calculated as (x_{Wr}-$x_{Wrcalculated}$), (y_{Wr}-$y_{Wrcalculated}$) and (z_{Wr}-$z_{Wrcalculated}$) when the varying value is θ_{r5}. 77
Figure 4-1: Robot's variables 80
Figure 4-2: The dependent velocity variables, β_1 and β_2, due to the nonholonomic constraints φ_1 and φ_2. 81
Figure 4-3: Kinetostatic transmission element connecting the dependent variables β and the wheels' tip velocity φ. 83
Figure 4-4: The y unit vector of K_{Wl}, y_{Wl}. The wheel's velocity must be zero in that direction for the nonholonomic constraint to be satisfied. 86
Figure 4-5: Verification of the constraint equation, $\dot{r}_{Wl}{}^{T} \cdot \hat{y}_{Wl} = 0$, with the MOBILE variable values for $\dot{\varphi}_{roll} = 5\,\mathrm{deg}/s$. 89
Figure 4-6: Verification of the constraint equation, $\dot{r}_{Wl}{}^{T} \cdot \hat{y}_{Wl} = 0$, with the MOBILE variable values for $\dot{\theta}_{l5} = 0.5\,\mathrm{deg}/s$. 90
Figure 4-7: Verification of the constraint equation, $\dot{r}_{Wl}{}^{T} \cdot \hat{y}_{Wl} = 0$, with the MOBILE variable values for $\dot{\theta}_{r5} = 0.5\,\mathrm{deg}/s$ 92
Figure 4-8: The y unit vector of K_{Wr}, y_{Wr}. The wheel's velocity must be zero in that direction for the nonholonomic constraint to be satisfied. 94
Figure 4-9: Verification of the constraint equation, $\dot{r}_{Wr}{}^{T} \cdot \hat{y}_{Wr} = 0$, with the MOBILE variable values for $\dot{\varphi}_{roll} = 5\,\mathrm{deg}/s$. 97
Figure 4-10: Verification of the constraint equation, $\dot{r}_{Wr}{}^{T} \cdot \hat{y}_{Wr} = 0$, with the MOBILE variable values for $\dot{\theta}_{l_5} = 0.5\,\mathrm{deg}/s$. 98
Figure 4-11: Verification of the constraint equation, $\dot{r}_{Wr}{}^{T} \cdot \hat{y}_{Wr} = 0$, with the MOBILE variable values for $\dot{\theta}_{r_5} = 0.5\,\mathrm{deg}/s$. 100
Figure 4-12: Verification of the constraint equations, $\dot{r}_{Wl}{}^{T} \cdot \hat{y}_{Wl} = 0$ (left) and $\dot{r}_{Wr}{}^{T} \cdot \hat{y}_{Wr} = 0$ (right) with the MOBILE variable values for the complete verification test. 101
Figure 5-1: Effect of cabin roll angle to cabin yaw angle for positive (left) and negative (right) roll values, for dependent variables ẏC and φ̇yaw. 104
Figure 5-2: Effect of cabin roll angle to cabin y distance with respect to K0 for positive (left) and negative (right) roll values, for dependent variables ẏC and φ̇yaw. 105
Figure 5-3: Mechanism's pose for positive (left) and negative (right) roll value, for dependent variables ẏC and φ̇yaw. 105
Figure 5-4: Influence of positive (left) and negative (right) left wheel lever straddle angles to cabin yaw rotation for dependent variables ẏC and φ̇yaw. 107

Figure 5-5: Influence of positive (left) and negative (right) left wheel lever straddle angles to cabin y distance with respect to K0 for dependent variables ẏC and φ̇yaw. 107
Figure 5-6: Mechanism's pose for positive θ_{l5} value (left) and negative θ_{l5} value (right), for dependent variables ẏC and φ̇yaw. 108
Figure 5-7: Influence of positive (left) and negative (right) right wheel lever straddle angles to cabin yaw rotation for dependent variables ẏC and φ̇yaw. 109
Figure 5-8: Influence of positive (left) and negative (right) right wheel lever straddle angles to cabin y distance with respect to K0 for dependent variables ẏC and φ̇yaw. 110
Figure 5-9: Mechanism's pose for positive θ_{r5} values (left) and negative θ_{r5} value (right), for dependent variables ẏC and φ̇yaw. 110
Figure 5-10: Positive (left) and negative (right) cabin roll angles influence to left wheel lever straddle angle, for dependent variables θ_{l5}, θ_{r5}. 112
Figure 5-11: Positive (left) and negative (right) cabin roll angles influence to right wheel lever straddle angle, for dependent variables θ_{l5}, θ_{r5}. 112
Figure 5-12: Mechanism's pose for positive (left) and negative (right) cabin roll value, for dependent variables θ_{l5}, θ_{r5}. 113
Figure 5-13: Positive (left) and negative (right) cabin yaw angles influence to left wheel lever straddle angle, for dependent variables θ_{l5}, θ_{r5}. 114
Figure 5-14: Positive (left) and negative (right) cabin yaw angles influence to right wheel lever straddle angle, for dependent variables θ_{l5}, θ_{r5}. 115
Figure 5-15: Mechanism's pose for positive (left) and negative (right) cabin yaw value, for dependent variables θ_{l5}, θ_{r5}. 115
Figure 5-16: Positive (left) and negative (right) yC values influence to left wheel lever straddle angle, for dependent variables θ_{l5}, θ_{r5}. 117
Figure 5-17: Positive (left) and negative (left) yC values influence to right wheel lever straddle angle, for dependent variables θ_{l5}, θ_{r5}. 117
Figure 5-18: Mechanism's pose for positive (left) and negative (right) yC value, for dependent variables θ_{l5}, θ_{r5}. 118
Figure 6-1: The non-holonomic acceleration constraints φ_1, φ_2 and the dependent accelerations β_1 and β_2 because of them. 120
Figure 6-2: Model of a mass element on MOBILE. 122
Figure 6-3: Cytec Thornel Mat VMA Carbon Fiber properties. 122
Figure 6-4: Solidworks model of the cabin. The pink system of coordinates denotes the center of mass while the blue one is the output coordinate system for Solidworks' mass properties calculations. 123
Figure 6-5: Mass properties of the cabin. 123
Figure 6-6: Rigid body modelling the cabin in MOBILE. 124
Figure 6-7: Solidworks model of thigh for both legs. The pink system of coordinates denotes the center of mass while the blue one is the output coordinate system for Solidworks' mass properties calculations. 124
Figure 6-8: Mass properties of the thighs of both legs. 125
Figure 6-9: Rigid bodies modelling the left and right leg thighs in MOBILE. 125
Figure 6-10: Solidworks model of the rigid body between the knee and the ankle for both legs. The pink system of coordinates denotes the center of mass while the blue one the output coordinate system for Solidworks' mass properties calculations. 126
Figure 6-11: Mass properties of the rigid bodies between the knees and the ankles of both legs .. 126
Figure 6-12: Rigid bodies modelling the left and right leg parts between the knees and the ankles in MOBILE. 127

Figure 6-13: Solidworks model for each of the wheel lever and wheel system. The pink system of coordinates denotes the center of mass while the blue one is the output coordinate system for Solidworks' mass properties calculations. 127
Figure 6-14: Mass properties for each wheel lever and wheel system of the mechanism. 128
Figure 6-15: Rigid bodies modelling the left and wheel lever and wheel systems in MOBILE. All the rigid bodies of the system are now visual. 128
Figure 6-16: Model of the inverse dynamics of a multibody system. 129
Figure 6-17: 1st Simulation start (left) and end (right) robot pose. 133
Figure 6-18: Robot's variables. 133
Figure 6-19: Dimensions of the robot. The whole system is symmetrical. 133
Figure 6-20: ψl2, ψl3, ψl4 and ψr2 torques needed for a cabin x movement of 1.482 m using the first weight method. 134
Figure 6-21: ψr3, ψr4, θl6 and θr6 torques needed for a cabin x movement of 1.482 m using the first weight method. 135
Figure 6-22: ψl2, ψl3, ψl4 and ψr2 torques needed for a cabin x movement of 1.482 m using the second weight method. 136
Figure 6-23: ψr3, ψr4, θl6 and θr6 torques needed for a cabin x movement of 1.482 m using the second weight method. 136
Figure 6-24: Static torque of ψ_2 to check the torque magnitude of the simulation results. 137
Figure 6-25: 2nd Simulation start (left) and end (right) robot pose. 138
Figure 6-26: ψl2, ψl3, ψl4 and ψr2 torques needed for a cabin roll movement of -5.41º using the first weight method. 139
Figure 6-27: ψr3, ψr4, θl6 and θr6 torques needed for a cabin roll movement of -5.41º using the first weight method. 140
Figure 6-28: ψl2, ψl3, ψl4 and ψr2 torques needed for a cabin roll movement of -5.41º using the second weight method. 140
Figure 6-29: ψr3, ψr4, θl6 and θr6 torques needed for a cabin roll movement of -5.41º using the second weight method. 141
Figure 6-30: 3rd Simulation start (upper) and end (down) robot pose. 142
Figure 6-31: ψl2, ψl3, ψl4 and ψr2 torques needed for a cabin z movement of -1.743 m using the first weight method. 143
Figure 6-32: ψr3, ψr4, θl6 and θr6 torques needed for a cabin z movement of -1.743 m using the first weight method. 144
Figure 6-33: ψl2, ψl3, ψl4 and ψr2 torques needed for a cabin z movement of -1.743 m using the second weight method. 144
Figure 6-34: ψr3, ψr4, θl6 and θr6 torques needed for a cabin z movement of -1.743 m using the first weight method. 145

List of tables

Table 3-1: Excavator’s geometrical values. 50

Table 3-2: The values of the independent variables affecting the left foreleg’s position for the 1st verification. 53

Table 3-3: The values of the independent variables affecting the left foreleg’s position for the 1st verification. 55

Table 3-4: The values of the variables affecting the left foreleg's position for the third verification. 57

Table 3-5: The values of the variables affecting the left foreleg's position for the fourth verification. 58

Table 3-6: The values of the variables affecting the right foreleg's position for the first verification. 62

Table 3-7: The values of the variables affecting the right foreleg's position for the second verification. 63

Table 3-8: The values of the variables affecting the right foreleg's position for the third verification. 64

Table 3-9: The values of the variables affecting the right foreleg's position for the fourth verification. 65

Table 3-10: The values of the variables affecting the rear left wheel’s position for the first verification. 70

Table 3-11: The values of the variables affecting the rear left wheel’s position for the second verification. 71

Table 3-12: The values of the variables affecting the rear right wheel’s position for the first verification. 76

Table 3-13: The values of the variables affecting the rear right wheel’s position for the second verification. 77

Table 4-1: Initial conditions and fixed values for the first verification test of the left wheel. 88

Table 4-2: Initial conditions and fixed values for the second verification test of the left wheel. 89

Table 4-3: Initial conditions and fixed values for the third verification test of the left wheel. 91

Table 4-4: Initial conditions and fixed values for the first verification test of the right wheel. 96

Table 4-5: Initial conditions and fixed values for the second verification test of the right wheel. 97

Table 4-6: Initial conditions and fixed values for the third verification test of the right wheel. 99

Table 4-7: Initial conditions and fixed values for the complete verification test of both wheels. 100

Table 5-1: Initial conditions and fixed values for the steering influence of cabin roll simulation, for dependent variables ẏC and φ̇yaw. 103

Table 5-2: Initial conditions and fixed values for the steering influence of left wheel lever straddle angle simulation, for dependent variables ẏC and φ̇yaw. 106

Table 5-3: Initial conditions and fixed values for the steering influence of right wheel lever straddle angle simulation, for dependent variables ẏC and φ̇yaw. 108

Table 5-4: Initial conditions and fixed values for the steering influence of cabin roll angle simulation, for dependent variables θ_{l5}, θ_{r5}. 111

Table 5-5: Initial conditions and fixed values for the steering influence of cabin yaw angle simulation, for dependent variables θ_{l5}, θ_{r5}. 113

Table 5-6: Initial conditions and fixed values for the steering influence of yC simulation, for dependent variables θl5, θr5. 116

Table 6-1: Total weight and approximate total length, width and height of the robot. 132

Table 6-2: Initial conditions, geometry variables and mass properties of the robot for the 1st inverse dynamics simulation. 133
Table 6-3: Initial conditions, geometry variables and mass properties of the robot for the 2nd inverse dynamics simulation. 138
Table 6-4: Initial conditions, geometry variables and mass properties of the robot for the 3rd inverse dynamics simulation. 142

1 Introduction

1.1 Motivation

Mobile robots are able to move around in their environment instead of being fixed to one specific location. Generally, mobile robots can be roughly classified into three categories, air-based robots, water-based robots and land-based robots. The latter can be divided according to the way they move into legged, wheeled and tracked robots. The most common categories are legged robots and wheeled robots, each category has its own advantages and disadvantages.

Wheels are by far the most popular method of providing robot mobility. In contrast with legged robots, wheeled robots, have much simpler and cheaper design, production, programming process and control. They can also achieve higher velocities and travel larger distances. Nevertheless, wheeled robots have two important restrictions which are their big disadvantages in comparison with legged robots. This is their need for flat surface with adequate friction coefficient to operate and that their obstacle passing ability is heavily restricted by their wheels' radius.

On the other hand, legged robots are not subject to wheeled robots' limitations as they can navigate in extremely rough terrains and overcome large obstacles. That comes with a cost as, compared to wheeled robots, they have complicated design and control, they are more expensive and move much slower.

Over the last years, wheel-legged hybrid mobile robots have become a focus of research because they can integrate many advantages of both categories while compensating the disadvantages of each other when employed in various applications like:

- Search and rescue missions
- Working in hazardous environments
- Assisting people in their homes and/or work places
- Assisting impaired people
- Biomechanics field contribution
- Delivering goods
- Supplying isolated human teams
- Agriculture/Mining/Construction
- Firefighting

The aim of this thesis is to provide a better understanding of a non-holonomic excavator combining wheels and legs, through kinematic and dynamic modelling and simulation, making a step towards creating a working platform for the study of these robots' behaviour with the goal of tackling the above applications in a more efficient manner than a wheeled or legged robot would.

1.2 Related work and state of the art

In the last years, there has been a significant work on wheeled-legged locomotion. This section aims to present a brief overview of the early ideas and the latest approaches, regarding hybrid wheel-legged robots.

Most of these hybrid robots, such as MAMMOTH [22], SHERPA [5], Hylos [1], [10] and Justin [9], which can be seen in the following figures, Figure 1-1, Figure 1-2, Figure 1-3 and Figure 1-4 are Actively Articulated Wheeled Mobile Robots (AAWMRs). They are capable of articulating their limbs to actively conform their terrain. Therefore, they use their legs as an active suspension system while driving instead of a locomotion alternative to the wheels.

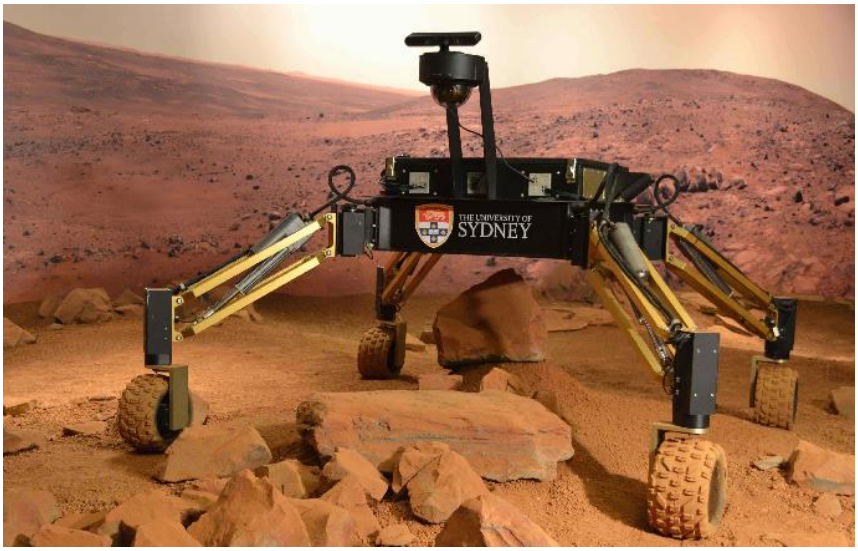

Figure 1-1: The MAMMOTH rover: An RGB-D sensor used to collect point clouds of the surrounding terrain is mounted on the top of the rover's mast [22].

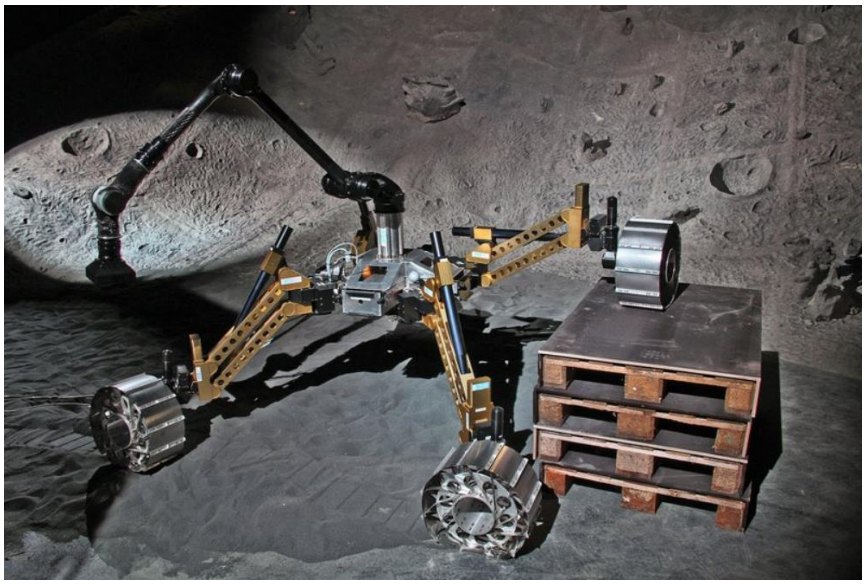

Figure 1-2: Sherpa stepping onto a high obstacle. The manipulator was used to support the rover while lifting the wheel onto the obstacle [5].

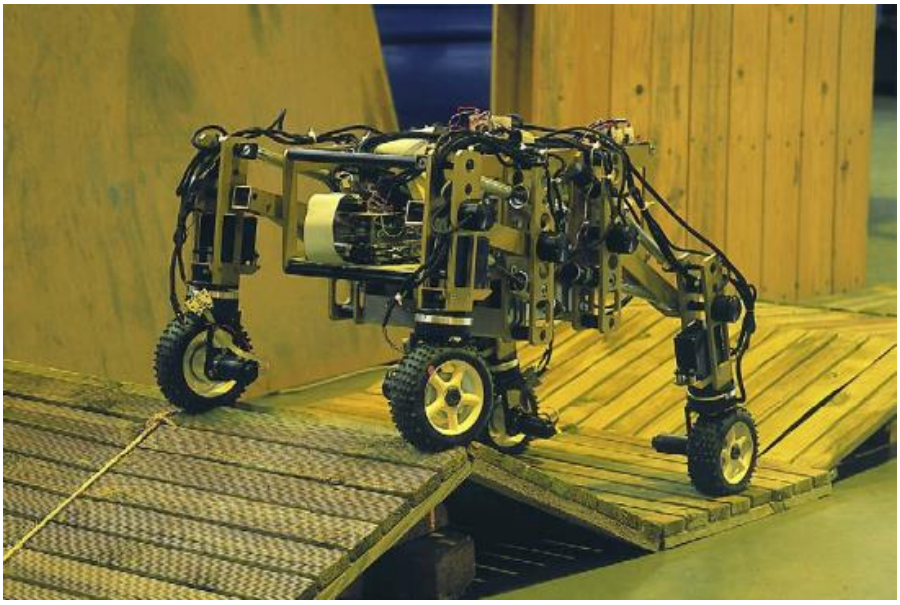

Figure 1-3: Hylos robot [1], [10].

Figure 1-4: Overview of the Justin mobile platform system [9].

On the contrary, the next robots use their legs as an alternative to the wheels locomotion method.

DRC-HUBO+ [19], as seen in Figure 1-5, is a wheeled humanoid robot which can select two types of mobility by transforming the posture of its legs. It can travel on flat land using wheels attached to the knees, it can walk and traverse rubble and stairs using its two legs. The legs are not used for locomotion or balance while its driving.

Figure 1-5: DRC-Hubo+ is a versatile humanoid that won first place at the DARPA Robotics Challenge in 2015. The robot can use tools, open doors, and even drive a vehicle. It can transform itself from a walking biped to a wheeled machine [19].

Momaro's [16] locomotion concept is based on four compliant legs which end in pairs of directly driven steerable wheels. This allows for omnidirectional driving on rugged terrain. To overcome larger obstacles and to climb stairs, individual legs are lifted and the robot makes steps. Therefore, it can drive and walk without changing its configuration. Momaro can be seen in Figure 1-6. Similar to Momaro is H2020 project CENTAURO [15], [17], [18] which can be seen in Figure 1-7.

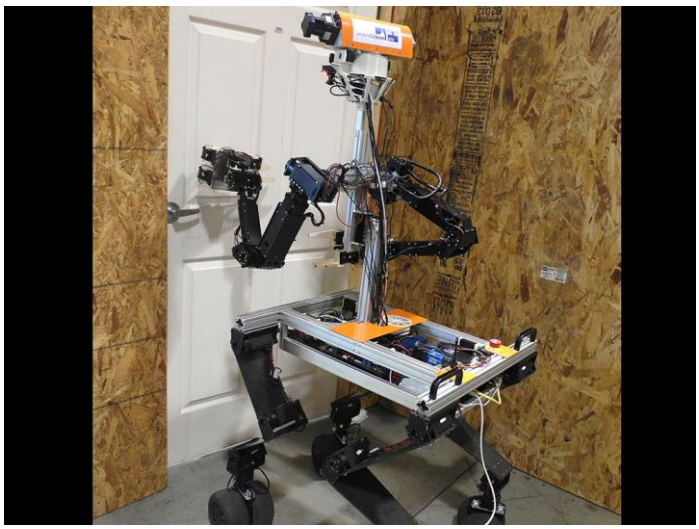

Figure 1-6: Momaro is a mobile robot that can carry out sensing and manipulation tasks in disaster sites and other harsh environments. It was designed to compete in the DARPA Robotics Challenge, coming in fourth place [16].

Figure 1-7: The legged-wheeled centaur-like robot, CENTAURO [15], [17], [18].

The Tri-ATHLETE [11] vehicle system, which can be seen in Figure 1-8, is a new form of two cooperative robotic vehicles that can act individually or physically connected through a structural pallet to transport and manipulate cargo. The basis of the Tri-ATHLETE (All Terrain Hex Limed Extra Terrestrial Explorer) robot is the wheel-on-limb vehicle concept. Tri-ATHLETE vehicle is the second generation of a wheel-on-limp vehicle being developed to support the return of humans to the lunar surface. This hybrid mobility platform enables the vehicle to traverse at high speeds across benign terrain, as well as enabling walking, by locking the wheels and using them as feet, on extreme terrain.

Figure 1-8: Tri-ATHLETE robot, a six-legged robot developed at the NASA's Jet Propulsion Laboratory in southern California [11].

The hybrid quadruped/wheeled robot that is developed by the Control Systems Lab of the National Technical University of Athens in association with the University of Duisburg-Essen [28] can be seen in Figure 1-9. This diploma thesis by Elias Zournatzis, addresses the methods followed towards the analysis, design and manufacturing of this hybrid robot. The design of the proposed robotic system is an attempt to get acquainted

with the field of hybrid robots as well as to gain insight into the incorporation of additive manufacturing methods in the implementation of low-cost robotic systems for research and educational purposes.

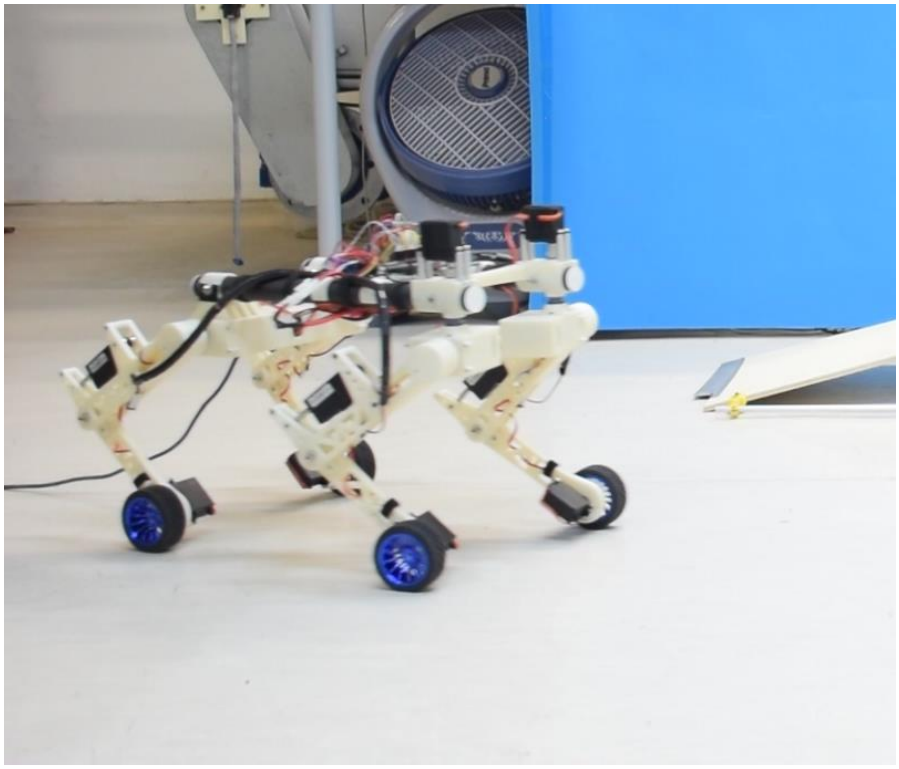

Figure 1-9: The hybrid quadruped/wheeled robot that is developed by the Control Systems Lab of the National Technical University of Athens in assosiation with the University of Duisburg-Essen.

Hyundai recently unveiled a walking car concept, called Elevate [13], to efficiently, rapidly and resiliently assist in disaster situations. It can be seen in Figure 1-10. Elevate's robotic leg architecture has five degrees of freedom plus wheel hub propulsion motors. This design is capable of both mammalian and reptilian walking gaits, allowing it to move in any direction. The legs can also fold up into a stowed drive-mode, where power to the joints is cut, and the use of an integrated passive suspension system maximizes battery efficiency. The non-back drivable motors enable the legs to lock in any position and this allows Elevate to drive at highway speeds. But Elevate is still in concept stage, instead Hyundai is working in the above at 1/8th scale which can be seen in Figure 1-11.

Figure 1-10: Hyundai's Ultimate Mobility Vehicle concept, Elevate [13].

Figure 1-11: Scaled proof of concept for Hyundai's Elevate [13].

Most of the robots using actuated wheels are not taking into account the dynamic model of the whole-body, including the wheels, preventing them from performing dynamic locomotion during walking and driving. So far Handle [3], [4], which can be seen in Figure 1-12, is the only solution that demonstrates dynamic motions to overcome high obstacles while showing adaptability against terrain irregularities. Handle is a robot that uses legs and wheels to provide highly agile and small-footprint material handling solutions for logistics. Using an active counterbalancing system, Handle can pick up and move cases weighing up to 15 kg. It can tackle pallet building, depalletizing and truck

unloading tasks. Due to the non-existent publications on Handle, there is no knowledge about its locomotion framework.

Figure 1-12: Boston Dynamic's Handle [3], [4].

Except for robots, excavators or heavy machines exist that rely on both legs and wheels. A walking excavator or spider excavator is a special type of all-terrain excavator. The use of a classic excavator requires fairly flat and/or specific terrains. The walking excavators can overcome these restrictions and provide a stable and horizontal position to the cab regardless of the terrain. Like the regular excavator, it consists of a boom, stick, bucket and cab on rotating platform known as the "house". However, its house sits atop an undercarriage consisting of leg extensions with or without wheels. All extensions can move and (sometimes with the help of the boom) the excavator can overcome obstacles and make walking motions in uneven and extreme terrains, hence the name "walking excavators".

In 1966, Edwin Ernst Menzi (1897-1984) and Joseph Kaiser (1928-1993) together invented the walking excavator for work on mountain slopes. Subsequently, Kaiser AG, Schaanwald, Liechtenstein, and Menzi Muck AG, Kriessern, Switzerland, developed excavators separately [29].

Another company is Euromach. Euromach, Montichiari (BS), Italy, appeared in 1977. Over 2000 excavators Euromach were built since that date. All these machines are sold in France by the multi-brand Company Camuc, Alby sur Cheran, HBI Group, which also sells among other Kaiser walking excavators. In the next figures below, Figure 1-13 and Figure 1-14, two characteristic walking excavator models can be seen.

Figure 1-13: Kaiser's S12 ALLROAD.

Figure 1-14: Menzi Muck A91.

1.3 Thesis structure

The work is organized as follows: Chapter 2 contains an introduction to the object-oriented programming package, MOBILE, and to the basic theory behind it. Chapter 3 describes how the excavator's model was implemented in MOBILE and how the inverse kinematics were done in it. It also contains the mathematical verification for MOBILE's inverse kinematics. Chapter 4 is about the non-holonomic constraints, how they were added to the model and how they were verified mathematically. Chapter 5 describes the different steering capabilities that arise because of the non-holonomic constraints.

Chapter 6 describes the whole procedure followed towards the completion of the inverse dynamics of the mechanism. Chapter 7 concludes this thesis and proposes future work. Chapter 8 contains the bibliography.

2 Elements of MOBILE

2.1 Introduction to MOBILE

MOBILE is an object-oriented programming package designed for the modelling of multibody systems. Its main features are

- Intuitive representation of mechanical entities as objects capable of transmitting motion and force across the system.
- Direct modelling of mechanical systems as executable programs, allowing the user to embed the resulting modules in existing libraries.
- Open, building-block system design, making it possible to extend the provided library in any direction.
- Scalable approach, treating all mechanical systems in a unified manner.
- Responsibility-driven client-server implementation, simplifying the task of invoking the required functions and of implementing own customized modules.
- Portable and efficient implementation, based on the object-oriented programming language C++.
- Built-in interfaces for three-dimensional graphic libraries for animation with direct user feed-back. User interaction includes click-and-drag features for on-line kinematics, statics and dynamics (this last feature may depend on system complexity and computer resources).

MOBILE addresses the following topics and more

- Basic mathematical objects and related operators for calculations in spatial dynamics: scalars, vectors, matrices, orthogonal transformations, elementary transformations, inertia tensors.
- Elementary building blocks for multibody systems: reference frames, angular and linear variables, elementary joints (prismatic and revolute), rigid links, elementary measurements mapping spatial motion to scalar quantities and tuples thereof, objects for creating composite chains of transmission elements.
- Elementary force elements (spring/damper, gravitation, mass, etc).
- Objects for the resolution of constraint equations, either in closed-form or iteratively.
- Objects for the generation of the equations of motion.
- Objects for the numerical integration of the dynamical equations.

One of the main features of MOBILE is that it allows the user to model systems as executable programs that can be used as building blocks for other environments. This is achieved by representing each real-world component by a dedicated object that is capable of performing some well-defined set of actions upon request. The objects of MOBILE are roughly organized in three categories:

a) **basic mathematical objects,** which provide the algebraic resources for performing the typical multibody calculations,
b) **kinetostatic state objects,** which are used to store and retrieve kinematic or load-related information at specific locations of the multibody system.
c) **kinetostatic transmission elements,** which transmit the information stored with the kinetostatic state objects from one location of the system to the other.

Each transmission element supplies, in analogy to its real-world counterpart, two basic operations:

i. the **transmission of motion** and
ii. the **transmission of forces**

In MOBILE, these two operations are realized as virtual functions, “doMotion()” and “doForce()”, respectively, that are shared by all kinetostatic transmission elements. Kinetostatic state objects serve as input and output variables for the various types of kinetostatic transmission elements. There exist two basic types of kinetostatic state objects:

a) **spatial kinetostatic state objects**, or reference frames, which can be imagined as interconnection junctures between pairs of kinetostatical transmission elements, and
b) **scalar kinetostatic state objects,** which represent actuator or sensor data used to drive the motors of the joints or to store scalar data extracted from the system by measurements.

The overall picture of the approach is illustrated in Figure 2-1 from MOBILE manual. Prior to system assembly, reference systems are “floating” in space and possess no mutual relationship. Scalar variables resemble “wires” waiting to be plugged into appropriate places of the kinetostatic transmission elements in order to generate the desired motion. After assembly, the reference systems become attached at specific points of the transmission elements, interconnecting them by pairs, while the scalar variables accomplish the task of inducing motion at selected joints of the system. The assembly of a mechanical system thus consists in connecting the inputs and outputs of the kinetostatic transmission elements in appropriate order such that the resulting chains resemble the original system.

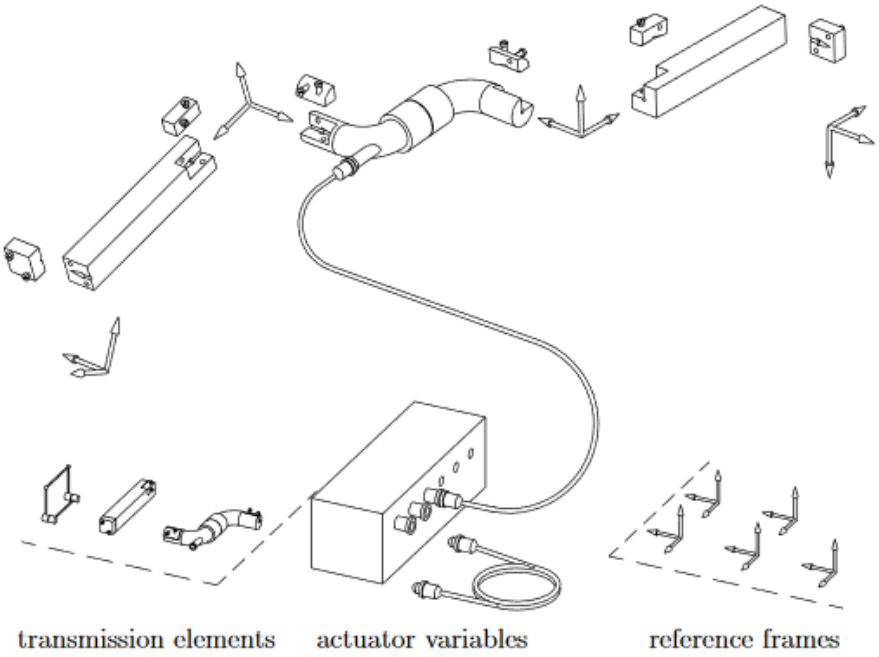

Figure 2-1: Objects in multibody systems.

The modelling of mechanical systems by kinetostatic transmission elements mirrors the client-server paradigm of object-oriented programming. In this setting, objects represent individuals that are endowed with specific “responsibilities”. These responsibilities are chosen in such a way that the correct functioning of the overall society is warranted. However, the particular manner in which each object fulfils its responsibility is left as a matter of taste. In MOBILE, the responsibilities of the mechanical elements are to provide the aforementioned virtual transmission functions. For these functions, it does not matter how an object realizes its task. What matters is only that it does realize it.

2.2 The concept of kinetostatic transmission element

A mechanical system can be regarded as a concatenation of kinetostatic transmission elements mapping motion and forces from one set of input state objects to a set of output state objects. State objects are, in this context, spatial reference frames and/or scalar variables, collecting positions and orientations as well as associated velocities, accelerations and generalized forces. Let the input and output state objects of an ideal kinetostatic transmission element be collected in vectors q_{in} and q_{out} respectively. This allows for the element to be represented as the block diagram shown in Figure 2-2.

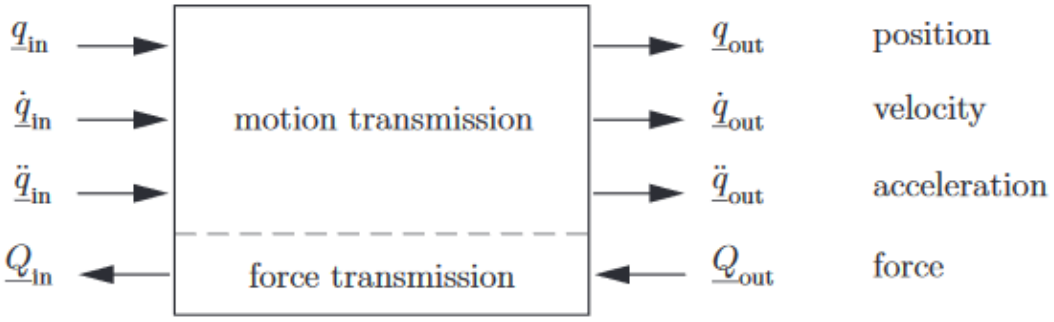

Figure 2-2: Elementary kinetostatic transmission element (from Kecskemethy (1993)).

The motion transmission behaviour described by the element comprises three sub-operations:
Position:

$$q_{out} = \varphi(q_{in}) \tag{2.1}$$

Velocity:

$$\dot{q}_{out} = J_{\varphi}\dot{q}_{in} \tag{2.2}$$

Acceleration:

$$\ddot{q}_{out} = J_{\varphi}\ddot{q}_{in} + \dot{J}_{\varphi}\dot{q}_{in} \tag{2.3}$$

$J_{\varphi} = \partial\varphi / \partial q_{in}$ represents the Jacobian of the transmission element. Furthermore, since the transmission element is ideal, the virtual work at the input and output of the element should be the same, i.e.

$$\delta q^{T}{}_{in} Q_{in} = \delta q^{T}{}_{out} Q_{out} \tag{2.4}$$

Substituting $\delta q_{out} = J_{\varphi}\delta q_{in}$ and noting that this condition must hold for all $\delta q_{in} \in \Re^{n}$, yields the force transmission function:

$$Q_{in} = J_{\varphi}{}^{\mathrm{T}} Q_{out} \tag{2.5}$$

As shown by Equation (2.5), the force transmission takes place in opposite direction to the motion transmission and can be computed using transposed velocity Jacobian. This relationship is known as the “duality of velocity and force” and holds independently of the complexity and nature of the motion transmission function $\varphi(q_{in})$.

One of the most powerful features of the concept described earlier is that a concatenation of kinetostatic transmission elements itself can be regarded as one global kinetostatic transmission element mapping global input state objects into global output state objects, which allows for different levels of abstraction and analysis. In the case of a whole mechanism, the global input state objects correspond to the generalized coordinates of the system and the output state objects can be set to be e.g. the end-effector frame as well as the frames on which the external and inertia forces act. This allows for an efficient

computation of Jacobian matrices as well as the projection of applied forces and inertial properties on the generalized coordinates.

3 Kinematics

Kinematics is the branch of mechanics that deals with the motion of bodies without consideration of the forces or moments that cause the motion. Robot kinematics applies geometry to the study of the movement of mutli-degree of freedom kinematic chains that form the structure of robotic systems. The emphasis on geometry means that the links of the robot are modelled as rigid bodies and its joints are assumed to provide pure rotation or translation. Robot kinematics studies the relationship between the dimensions and connectivity of kinematic chains and the position, velocity and acceleration of each of the links in the robotic system.

3.1 Basic kinematic model and MOBILE implementation

The kinematic model of the system comprises of the cabin, the two fore legs (including four revolute joints each, three at the hip called ψ_1, ψ_2 and ψ_3 and one at the knee called ψ_4) and the two rear wheel levers (including three revolute joints each, two at the hip called θ_6 and θ_5 and one for parallel wheel guidance at the end of the lever called θ_4). In addition to these physical entities, there are also some virtual mechanisms: a virtual serial chain of three prismatic and three revolute joints connecting the cabin-fixed reference system K_C (x_C, y_C, z_C, φ_{yaw}, φ_{pitch} and φ_{roll}) to the ground-fixed coordinate frame K_0, two virtual mechanisms (each constituting of two joints, θ_2 and θ_3) describing the wheel-ground contact with cornering and traction effects neglected because the wheels are regarded as ideally slippery and finally two virtual mechanisms for positioning the tips of the feet (each constituting of three joints, x_F, y_F and z_F), see Figure 3-1.

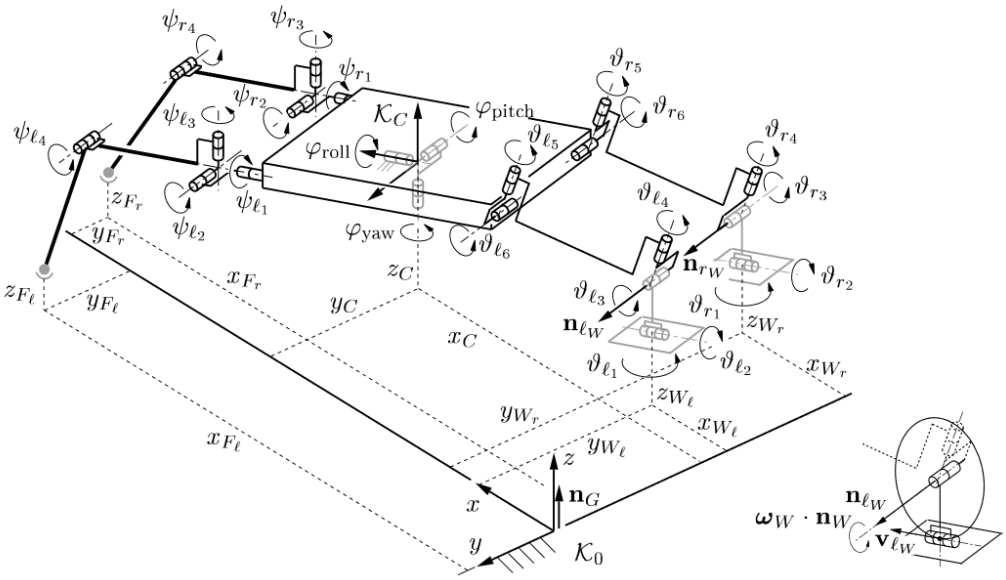

Figure 3-1: Kinetostatic skeleton of the hybrid excavator.

The above kinematic model from [25] was then implemented into MOBILE as a concatenation of frames, links and joints. Firstly, the virtual serial chain of the three prismatic joints (x_C, y_C and z_C in green colour) and three revolute joints (φ_{yaw}, φ_{pitch} and φ_{roll} in yellow colour) were made as seen in Figure 3-2 in order for the cabin-fixed frame K_C to be connected with the ground-fixed frame K_0 (red colour). The ground level is visualised as a white grid.

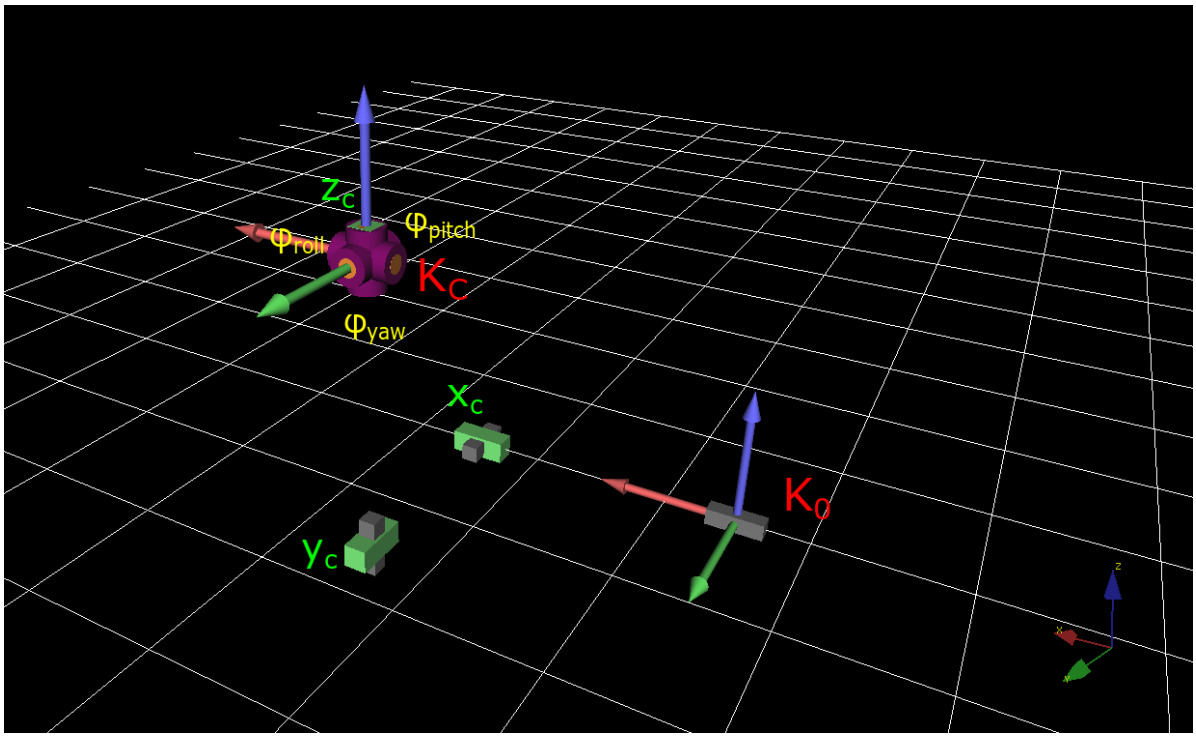

Figure 3-2: The three prismatic and three revolute joints that model the cabin's centre position and rotation with respect to the ground-fixed frame K0.

Secondly, the vehicle's cabin was constructed. It consists of four links that connect the cabin-fixed frame K_C with the two fore legs and the two rear wheel levers. These links can be seen in Figure 3-3, where their end frames can also be seen. The cabin only consists of those links that are necessary for motion and force transmission, any other additional link except them would result errors in the calculations of forces. This is the reason why the cabin does not have the form of a real cabin but only consists of those links necessary for motion and force transmission.

The next step was the addition of the mechanism's fore legs. The legs are symmetrical to the x-z plane that is collinear to the centre of the cabin (frame K_C). Each leg consists of three revolute joints (ψ_{l1}, ψ_{l2} and ψ_{l3} for the left leg and ψ_{r1}, ψ_{r2} and ψ_{r3} for the right in x-y-z direction accordingly) in series at the hip. These joints are connected to the knee by a link. At the end of that link there is another revolute joint (ψ_{l4} for the left leg and ψ_{r4} for the right in y direction). The result is shown in Figure 3-4 (x_C, y_C and z_C values are 0 in Figure 3-4).

Finally, the wheels' system was added which is also symmetrical to the x-z plane that is collinear to the centre of the cabin (frame K_C). The wheel levers comprise of two revolute joints at the hip (θ_{l6} and θ_{l5} for the left lever and θ_{r6} and θ_{r5} for the right, in y and z direction

accordingly) connected with a link. At the end of that link there is another revolute joint (θ_{l4} for the left lever and θ_{r4} for the right, in z direction) and then follows the wheel's virtual mechanism. It consists of a revolute joint aligned along the intersection of the contact plane and the wheel plane allowing for camber inclination of the wheel (θ_{l2} for the left wheel and θ_{r2} for the right, in x direction) and a revolute joint collinear to the wheel hub (θ_{l3} for the left wheel and θ_{r3} for the right, in y direction), connected with a link with the previous joint, describing the orientation of the rear wheel lever with respect to the ground normal n_G (see Figure 3-1), which connects with the end of the wheels' lever. The complete kinetostatic model is visualized in Figure 3-5 where the virtual serial chain is hidden.

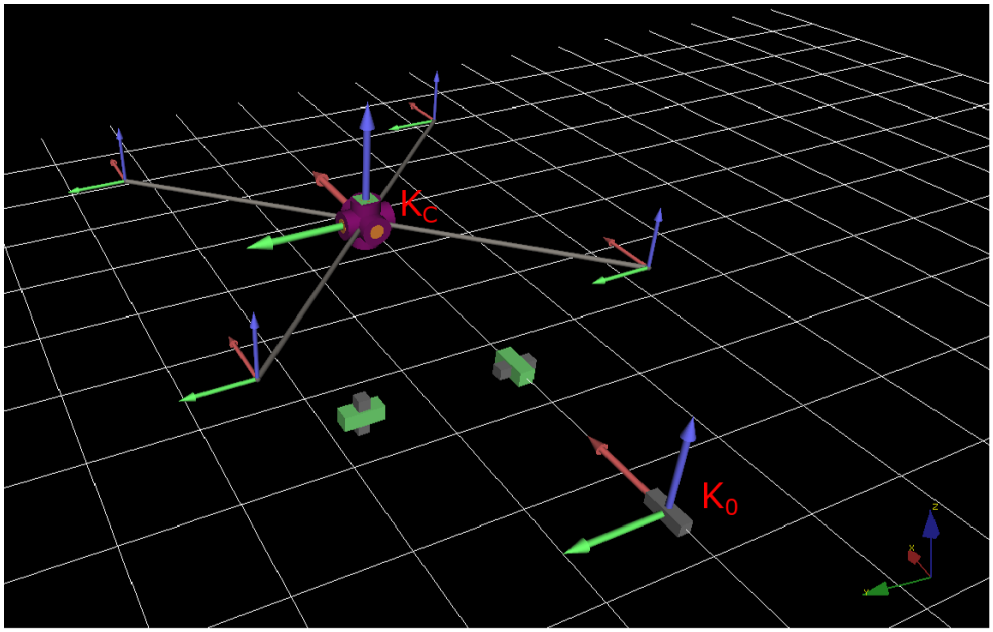

Figure 3-3: The vehicle's cabin attached to the position and rotation joints ,that connect the cabin-fixed and the ground-fixed frames, made earlier.

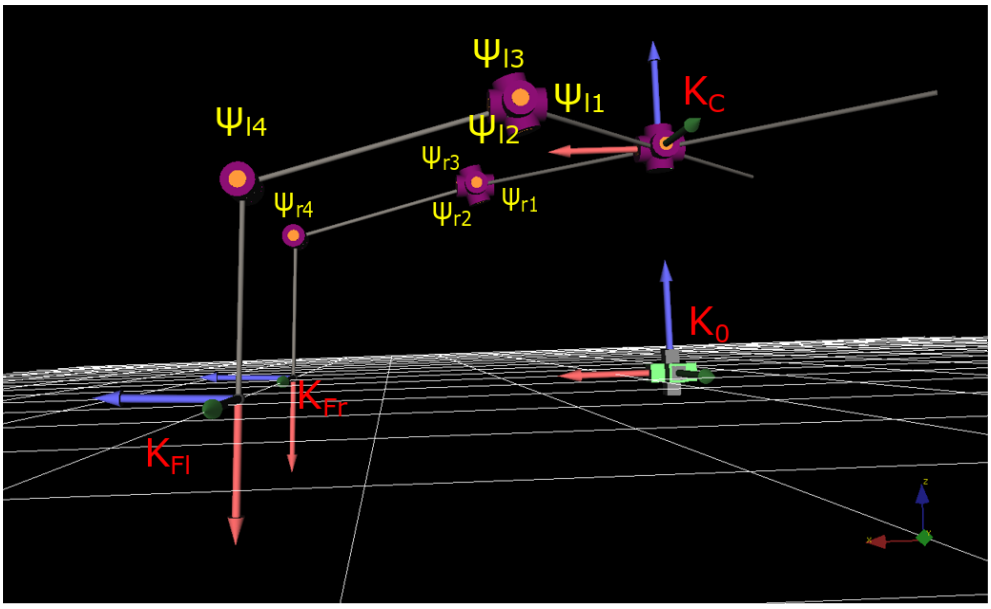

Figure 3-4: The excavator's front legs attached to the cabin.

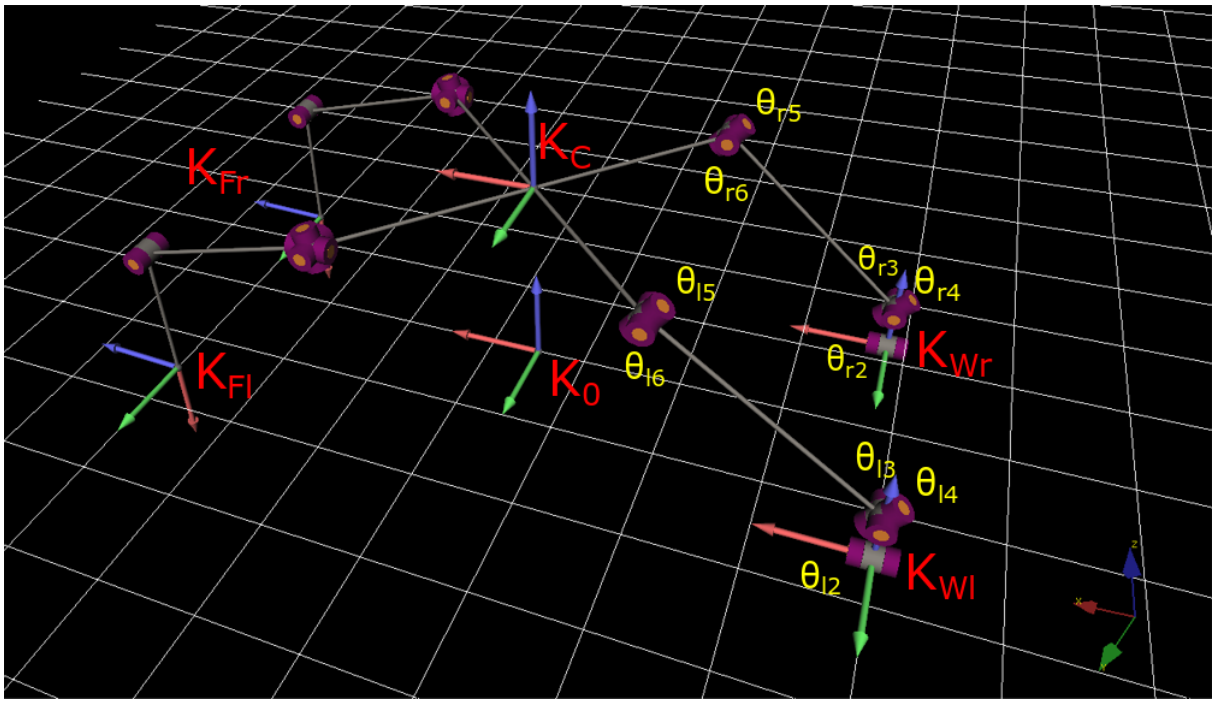

Figure 3-5: The kinetostatic skeleton implemented on MOBILE.

3.2 Inverse kinematics

The robot open chain kinematics can be divided into forward kinematics and inverse kinematics. The forward kinematics problem is straightforward and there is no complexity deriving the equations. Forward kinematics uses the joint parameters to compute the configuration of the kinematical chain. Hence, there is always a forward kinematics solution of a manipulator.

Inverse kinematics is a much more difficult problem than forward kinematics. They make use of the kinematics equations to determine the joint parameters that provide a desired pose for each of the robot's end-effectors. The solution of the inverse kinematics problem is computationally expensive as singularities and nonlinearities make the problem more difficult to solve. The relationship between forward and inverse kinematics is illustrated in Figure 3-6.

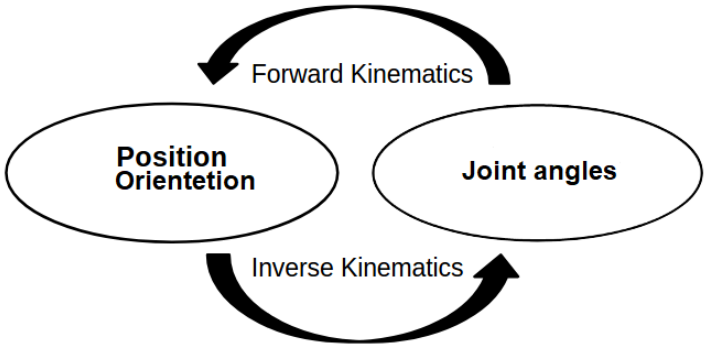

Figure 3-6: Forward and inverse kinematics relationship.

Two main solution approaches for the inverse kinematics problem is the analytical and numerical method. An analytical solution to an inverse kinematics problem is a closed-form expression that takes the end-effector pose as input and gives joint positions as output. Analytical inverse kinematics solvers can be significantly faster than numerical solvers and provide more than one solution for a given end-effector pose. In our case both methods will be employed; an iterative method will be used by MOBILE for inverse kinematics while an analytical solution will be used for verification.

Several approaches exist today for both methods, each having its advantages and disadvantages depending on the objectives of the simulation. For example, users seeking a high degree of efficiency need to access closed-form solutions where possible in order to avoid redundant computations, while users requiring a rapid yet maybe not so efficient modelling are satisfied with iterative solution procedures.

3.3 MOBILE iterative method for inverse kinematics

In MOBILE we apply inverse kinematics by creating closed loops. The basic procedure for tackling multibody loops in MOBILE is to first dissect the originally closed loop into serial chains and then to bring again the loose ends of the serial chains together by requiring the fulfilment of appropriate closure conditions.

Multibody systems can feature two fundamental types of structure: (i) *tree-type* structure or (ii) *single* or *multiple loop* structure (see Figure 3-7).

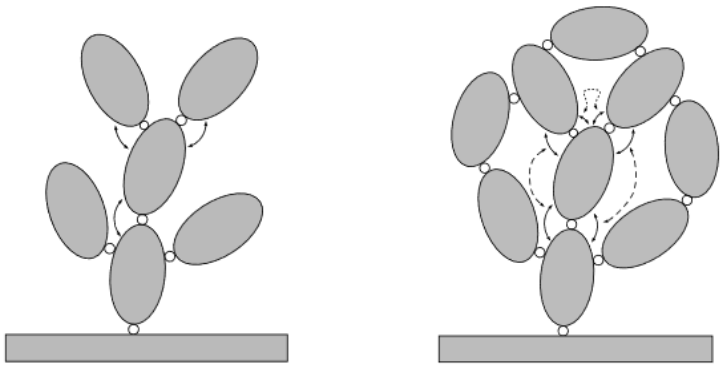

Figure 3-7: Tree type and closed-loop systems illustration.

In systems featuring tree-type structure, there is only one path between any component and the inertial frame. Thus, the relative motions between any two pairs of neighbouring bodies are independent, and it is possible to process the kinetostatics of the elements on a component by component basis. A user concerned with the modelling of such a system just needs to concatenate its components in an order that is compatible with its topological structure, i.e., starting at the inertial system and ending at the tips of the branches.

When the bodies of the multibody system form closed loops, the relative motions within the loop become dependent; a change of relative motion at one place induces a change of

relative motion at another place. Such dependencies make it impossible to proceed joint by joint or body by body as in the tree-type structure case. Instead, one has to formulate and solve the so-called *constraint equations* or *closure conditions* that hold the branches of the loop together.

In MOBILE, the closure of loops is accomplished as a two-stage process:

- In the first stage, a set of "**characteristic measurements**" is defined whose vanishing indicates the closure of the loop. These measurements, also called "chords" in MOBILE, are typically generalized distances between geometric elements such as points, planes and lines. MOBILE provides a whole family of classes for making such measurements, which are derived from the (abstract) super-ancestor class "MoChord". The objects instantiated from these classes can be used as any other kinetostatic transmission element to propagate motions and forces. The measurement objects of MOBILE are characterized by three basic attributes. Firstly, the geometric type, determined by the type of geometric elements involved in the measurement (point, plane, line or reference frame). Measurements generating tensorial quantities are hereby denoted by *spatial measurements*, while measurements producing scalar outputs are termed *scalar measurements*. Secondly, the topological type, which is determined by the number of frames involved in the measurement, as well as the type of motion (relative or absolute) regarded in the measurement. Finally, the activity type, which characterizes the behaviour (static or self-reconfiguring) of the measurement with respect to the motion of the involved frames. There are spatial measurements, like measuring the relative displacement between the origins of two frames and the relative orientation between two frames, and scalar measurements, like the quadratic and linear distance (these will be explained later) between the origins of two frames.
- In the next stage, one or more objects termed "**solvers**" are defined that are set to determine the dependent relative motions within the loop such that the measurements vanish. MOBILE supplies two classes for this purpose, which are both derived from the (abstract) super-class "MoSolver". One solves the constraint equations by iterative, Newton-based procedures and is called *implicit solver*. The implicit solvers can resolve any number of constraint equations iteratively for a set of unknowns. This is the universal, generally applicable method. The other takes a scalar equation and solves it in closed form for an unknown joint variable and is called *explicit solver*. Explicit solvers can resolve a scalar constraint equation explicitly in terms of one unknown. This method only works for special types of measurements and loop architectures. In both cases, the resulting solver objects behave again like kinetostatic transmission elements, supplying a motion and force transmission function. The motion transmission function consists of establishing (and carrying out) the motion of the dependent chain such that the loop stays closed. The force transmission function involves

the computation of the constraint forces within the loops and their propagation within the dependent chain such that static equilibrium is achieved. Constraint solving objects in MOBILE can process the kinetostatics of one or more closed loops. In order to keep the loops closed, a solver object needs three pieces of information: (i) the measurements whose vanishing will signal the closure of the loops, (ii) the dependent variables whose variation will lead to the closure of the loop and (iii) the dependent chain that will reconfigure the cut frames involved in the measurements after perturbing the dependent variables.

Therefore, the user has to carry out the following steps:

1. Decide where to cut the loop apart.
2. Decide which of the joint variable(s) of the loop are to be treated as dependent variable(s), and put these together in an object of type “MoVariableList” in case there are more than one unknowns. The other variables and motions are regarded as independent variables or kinematic inputs of the loop.
3. Create one or more object(s) modelling the dependent chains of the dissected loop. Each dependent chain is typically an object of type MoMapChain containing the kinetostatics from the dependent variables to the cut frames.
4. Create one or more object(s) derived from type “MoChord” that describe the loop closure condition(s).
5. Create an object of type “MoSolver”, passing to it the dependent chain(s), the list of dependent variable(s), and the object representing the closure condition(s).

After carrying out these steps, the user can employ the resulting object of type MoSolver as a simple kinetostatic transmission element representing the kinetostatics of the closed loop(s). The doMotion function of the solver generates the motion of the dependent chains so that they follow the input motion while keeping the loop closed; the doForce function computes the forces at the cut frames and within the dependent chains so that static equilibrium is achieved. Usage of this object is then fully equivalent to the usage of any other kinetostatic transmission element such as an elementary joint or a rigid link, i.e., solver objects can be used again as constituents of chains of kinetostatic transmission elements or even “super loops” exhibiting in their branches other loops.

3.3.1 MOBILE inverse kinematics for the left leg

Firstly, following the steps mentioned before, we have to create the closed loop for the left leg in order to compute the inverse kinematics. For that purpose, three prismatic joints (S_1 – z direction, S_2 – x direction, S_3 – y direction) are created in series begging from the ground-fixed inertia frame K_0 (see Figure 3-8). These joints will be used later to describe the left leg’s xyz movement with respect to K_0.

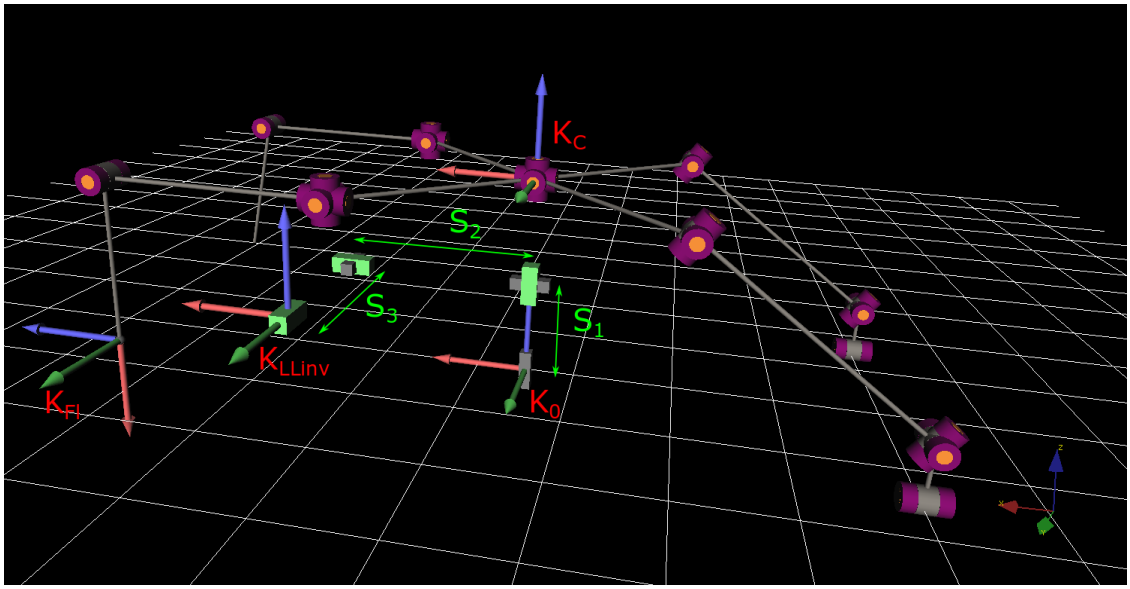

Figure 3-8: The three prismatic joints in series that will model the left leg's position with respect to the ground-fixed frame K_0.

Afterwards, we need to define the loop closure conditions, the characteristic measurements, for the left leg, between the left leg's end effector frame (K_{Fl}) and the end frame of the prismatic joints in series (K_{LLinv}) both seen in Figure 3-8. Scalar measurements will be used in this situation.

Scalar measurements are generated by projections from spatial frames to real numbers. The basic idea of this projection is illustrated in Figure 3-9 in its most simple form. The measurement object takes the motion of two frames, termed the target frame K_E and the base frame K_B, and produces a scalar quantity that depends only on the relative motion between both frames. The scalar measurements can be for example the quadratic distance between the origins of two frames, the linear distance between the origins of two frames, the cosine of the angle between two coordinate planes, the distance from a point to a plane where the point is located at the origin of frame K_E and the plane is coplanar to a coordinate plane of frame K_B and the shortest distance from a plane to a point where the plane is now a coordinate plane of the frame K_E and the point is the origin of frame K_B .

Figure 3-10 illustrates the two geometrical types of measurements most used in multibody analysis. Figure 3-11 summarizes the scalar geometric measurements and the underlying measurement expressions at position level. In these expressions, $\mathbf{r_B}$ and $\mathbf{r_E}$ denote the distance vectors between the reference frames K_B and K_E and the inertial system accordingly, and R_B and R_E are the corresponding transformation matrices from the reference frames to the inertial system. The vector $\mathbf{u_B}$ is a unit vector normal to the plane involved in the measurement. In MOBILE, only coordinate planes are allowed in measurements. Thus, unit vectors can only have one of the three values, **xAxis**, **yAxis** and **zAxis**.

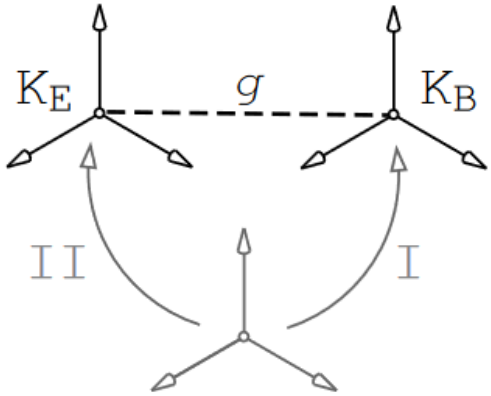

Figure 3-9: Scalar measurements idea.

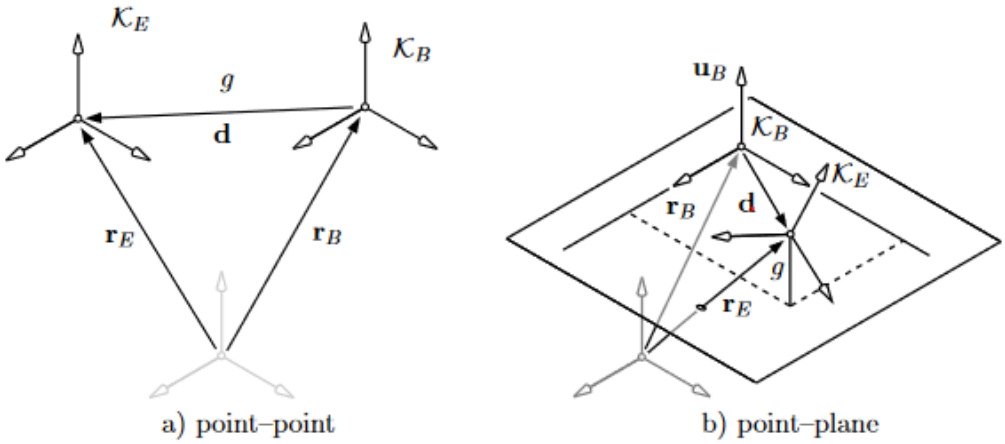

Figure 3-10: Geometrical entities involved in the measurements between points and planes.

MoChord...	geom. entity $\mathcal{K}_B$	geom. entity $\mathcal{K}_E$	expression
PointPlane	*coordinate plane*	*origin*	$(\mathbf{R}_E\,\mathbf{r}_E - \mathbf{R}_B\,\mathbf{r}_B)\,\mathbf{R}_B\,\mathbf{u}_B$
PointPointQuadratic	*origin*	*origin*	$\|\|\,\mathbf{R}_E\,\mathbf{r}_E - \mathbf{R}_B\,\mathbf{r}_B\,\|\|^2$
PointPointLinear	*origin*	*origin*	$\|\|\,\mathbf{R}_E\,\mathbf{r}_E - \mathbf{R}_B\,\mathbf{r}_B\,\|\|$
PlanePoint	*coordinate plane*	*origin*	$(\mathbf{R}_B\,\mathbf{r}_B - \mathbf{R}_E\,\mathbf{r}_E)\,\mathbf{R}_E\,\mathbf{u}_E$
PlanePlane	*origin*	*origin*	$(\mathbf{R}_B\,\mathbf{u}_B)\cdot(\mathbf{R}_E\,\mathbf{u}_E)$

Figure 3-11: Basic geometric types of scalar measurements.

In the left leg's case three "distance from a point to plane" measurements will be used. The measurements expression at position level is, as seen in Figure 3-11:

$$\left(R_E r_E - R_B r_B\right) R_B u_B \tag{3.1}$$

The measurements (Equation (3.1)) that the solver will have to bring to zero for the left leg are the distance of the left leg's end effector frame, K_{Fl}, with the three coplanar planes of K_{LLinv} in x,y and z direction as seen in Figure 3-12.

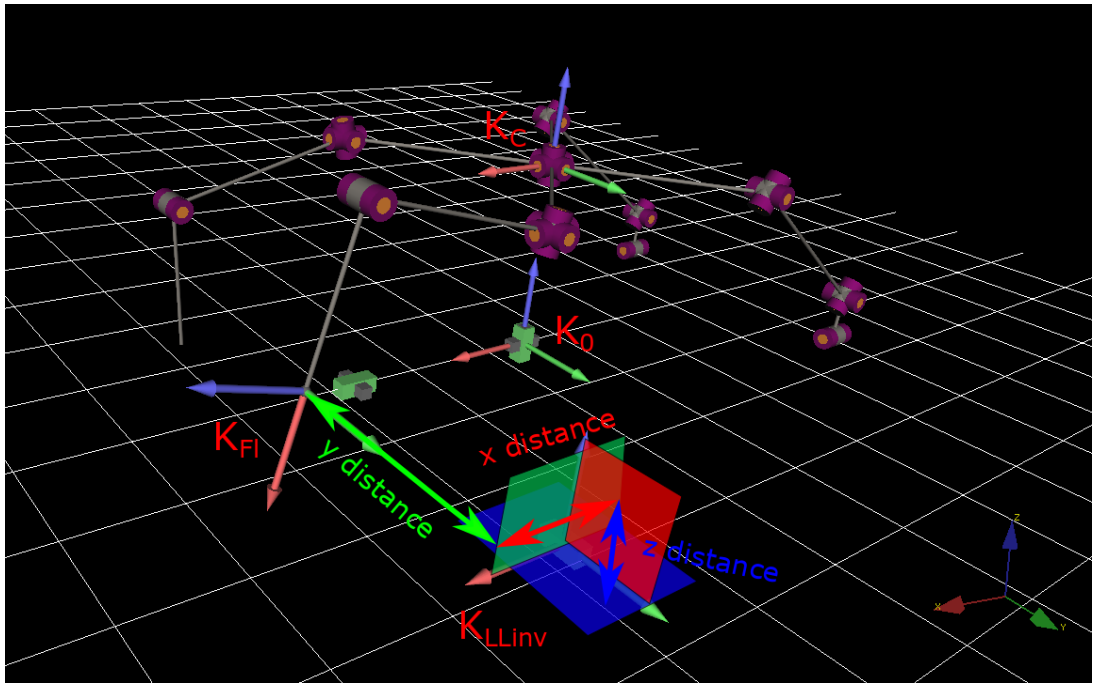

Figure 3-12: The characteristic measurements for the left's leg loop closure.

Next an object for constraint solving needs to be added. An implicit solver is going to be used for our problem.

Out of all the variables affecting the left leg's position (x_C, y_C, z_C, φ_{yaw}, φ_{pitch}, φ_{roll}, ψ_{l1}, ψ_{l2}, ψ_{l3}, ψ_{l4}, x_{Fl}, y_{Fl} and z_{Fl}, as seen in Figure 3-1) we select ψ_{l2}, ψ_{l3} and ψ_{l4} to be treated as dependent and the remaining as generalized (input) variables. We have to consider that the number of dependent variables must be the same with the constraint equations (characteristic measurements) for the solver to be able to solve the inverse kinematics. This is satisfied in our case as we have three constraints and three dependent variables, ψ_{l2}, ψ_{l3} and ψ_{l4}. Thus, having decided for the dependent variables, the closing conditions and the solver, the inverse kinematics of the left leg are complete. By changing the cabin's pose (x_C, y_C, z_C, φ_{yaw}, φ_{pitch}, φ_{roll}), the left leg's roll angle (ψ_{l1}) and the left leg's tip position (x_{Fl}, y_{Fl}, z_{Fl}), which are the independent variables, we can observe the dependent variables values (ψ_{l2}, ψ_{l3}, ψ_{l4}) needed for that change (see Figure 3-13).

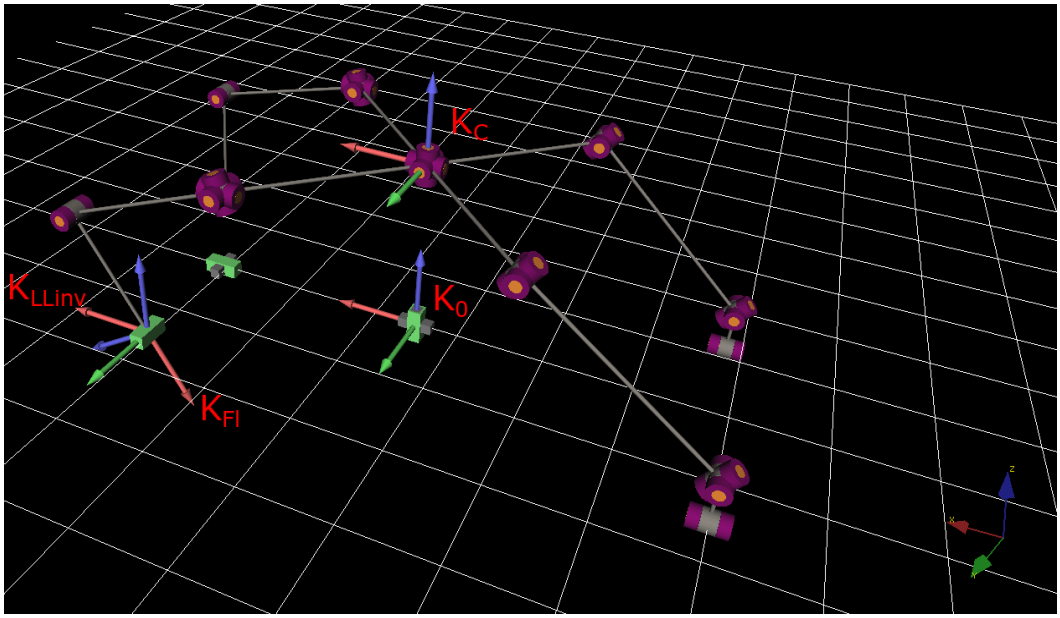

Figure 3-13: The inverse kinematics of the left leg are complete and by changing the independent variables we can observe the dependent values needed for that change.

3.3.2 MOBILE inverse kinematics for the right leg

As on the left leg, we firstly create the closed loop for the right leg in order to compute the inverse kinematics. For that purpose, three prismatic joints (S_4 – z direction, S_5 – x direction, S_6 – y direction) are created in series starting from the ground-fixed inertia frame K_0 (see Figure 3-14). These joints will be used later to describe the right leg's xyz movement with respect to K_0.

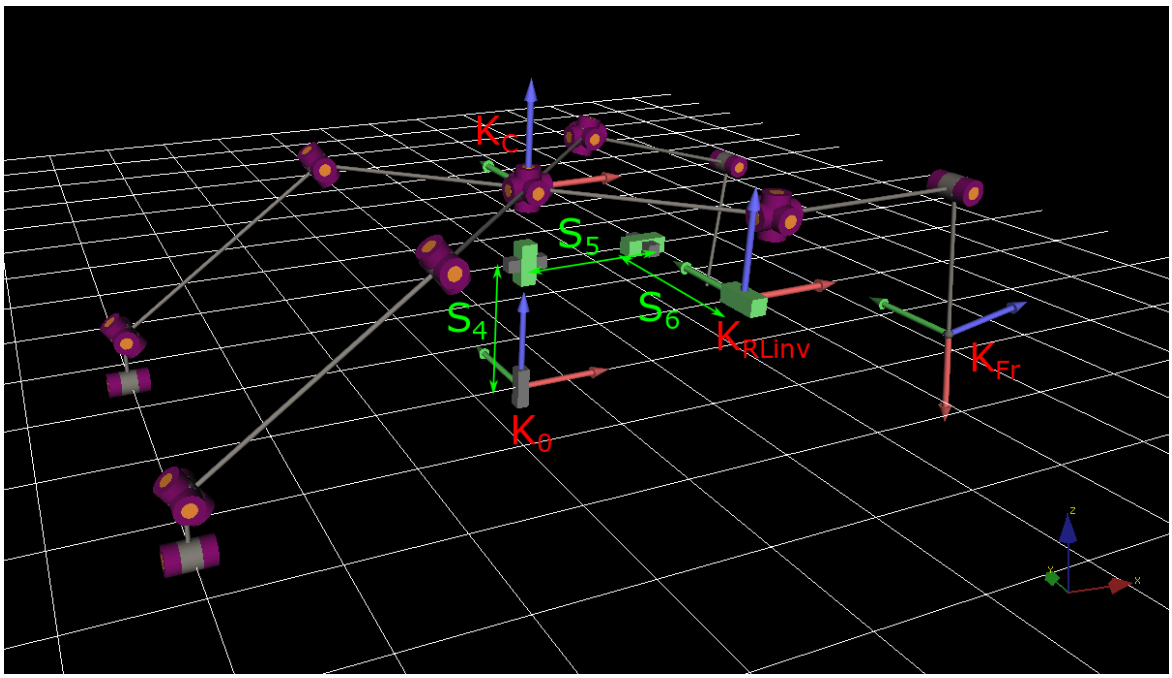

Figure 3-14: The three prismatic joints in series that will model the right leg's position with respect to the ground-fixed frame K_0.

Next, we need to define the loop closure conditions, the characteristic measurements, for the right leg, between the right leg's end effector frame (K_{Fr}) and the end frame of the prismatic joints in series (K_{RLinv}). Scalar measurements will be used, as before, measuring the distance of the right leg's end effector frame with the three coplanar planes of K_{RLinv} in x,y and z direction as seen in Figure 3-15. These are the measurements that the solver needs to satisfy (bring to zero).

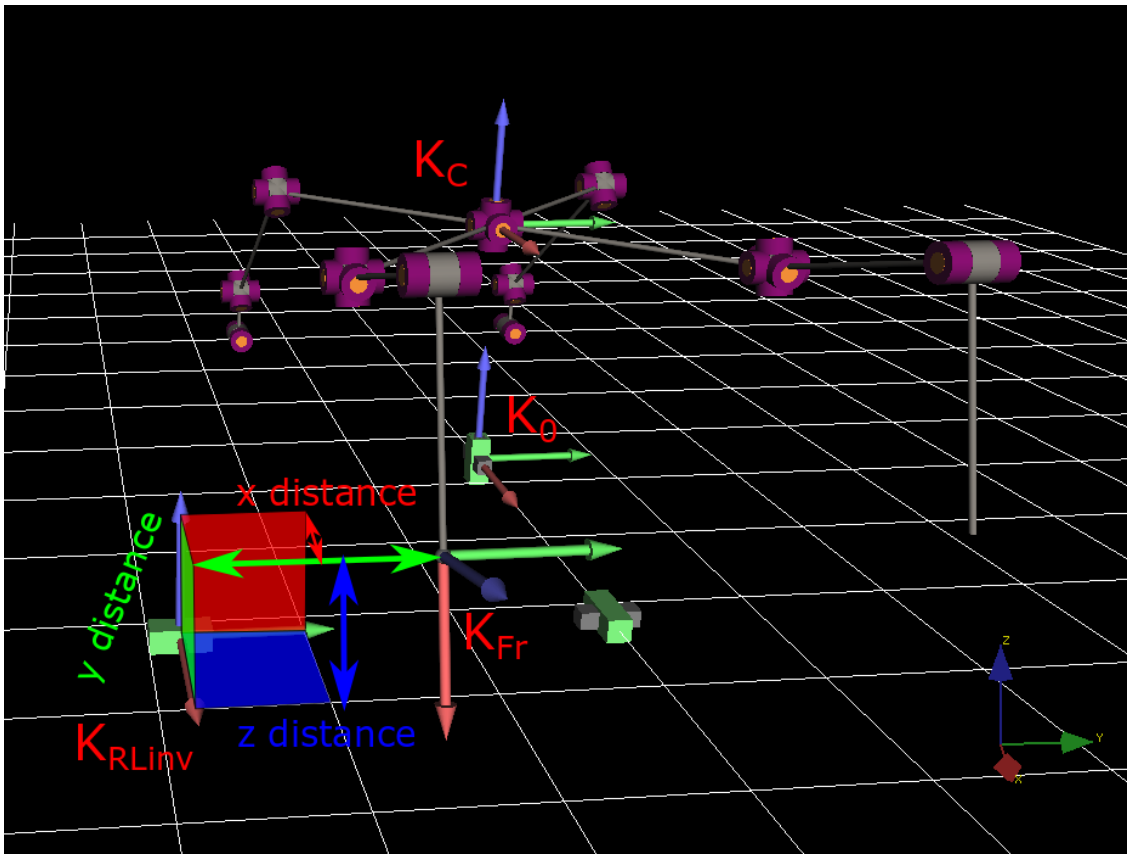

Figure 3-15: The characteristic measurements for the right's leg loop closure.

Finally, an object for constraint solving needs to be added. An implicit solver is going to be used like on the left leg.

Out of all the variables affecting the right leg's position (x_C, y_C, z_C, φ_{yaw}, φ_{pitch}, φ_{roll}, ψ_{r1}, ψ_{r2}, ψ_{r3}, ψ_{r4}, x_{Fr}, y_{Fr} and z_{Fr}, as seen in Figure 3-1) we select ψ_{r2}, ψ_{r3} and ψ_{r4} to be treated as dependent and the remaining as generalized (input) variables. Again, we have to consider that the number of dependent variables must be the same with the constraint equations (characteristic measurements) for the solver to be able to solve the inverse kinematics which happens in this case too as we have three constraints and three dependent variables. Thus, having decided for the dependent variables, the closing conditions and the solver, the inverse kinematics of the right leg are complete. By changing the cabin's pose (x_C, y_C, z_C, φ_{yaw}, φ_{pitch}, φ_{roll}), the right leg's roll angle (ψ_{r1}) and the right leg's tip position (x_{Fr}, y_{Fr}, z_{Fr}), which are the independent variables, we can observe the dependent variables values (ψ_{r2}, ψ_{r3}, ψ_{r4}) needed for this change (see Figure 3-16).

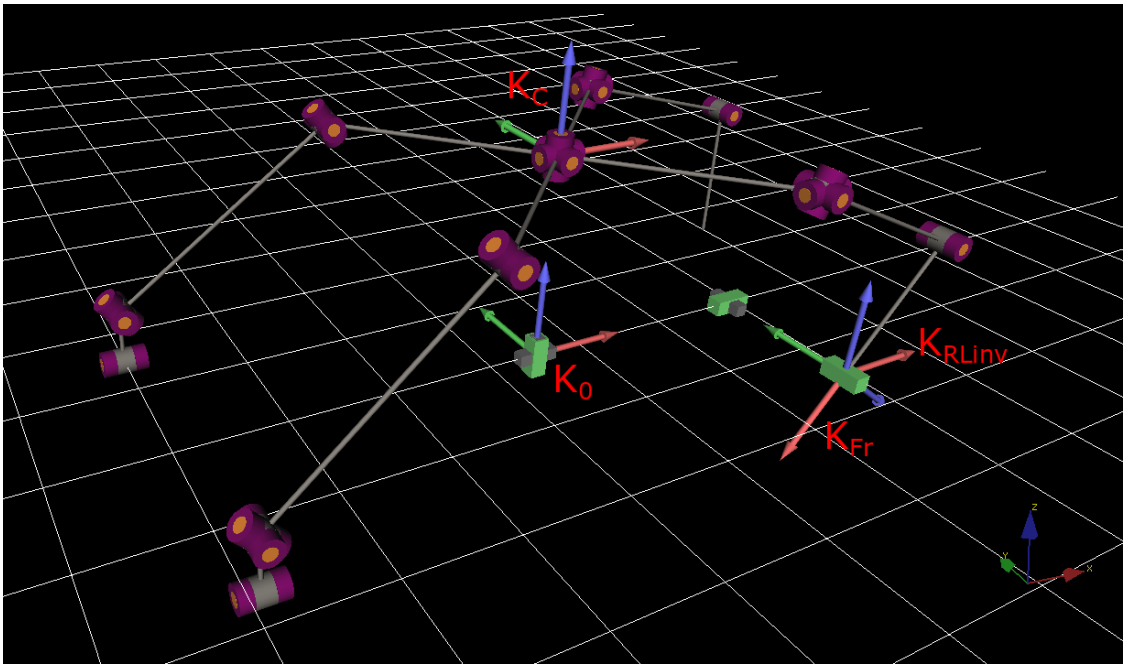

Figure 3-16: The inverse kinematics of the right leg are complete and by changing the independent variables we can observe the dependent values needed for that change.

3.3.3 MOBILE inverse kinematics for the left wheel

Moving on to the mechanism's left wheel, we also have to create a closed loop in order to do the inverse kinematics. For that purpose, we create a prismatic joint (S_7 – z direction) beginning from the ground-fixed inertia frame K_0 (see Figure 3-17). That joint will be used later as the left wheel's z movement with respect to K_0. We don't need prismatic joints on x and y direction like we did with the legs because the wheels' exact x-y position in of no concern.

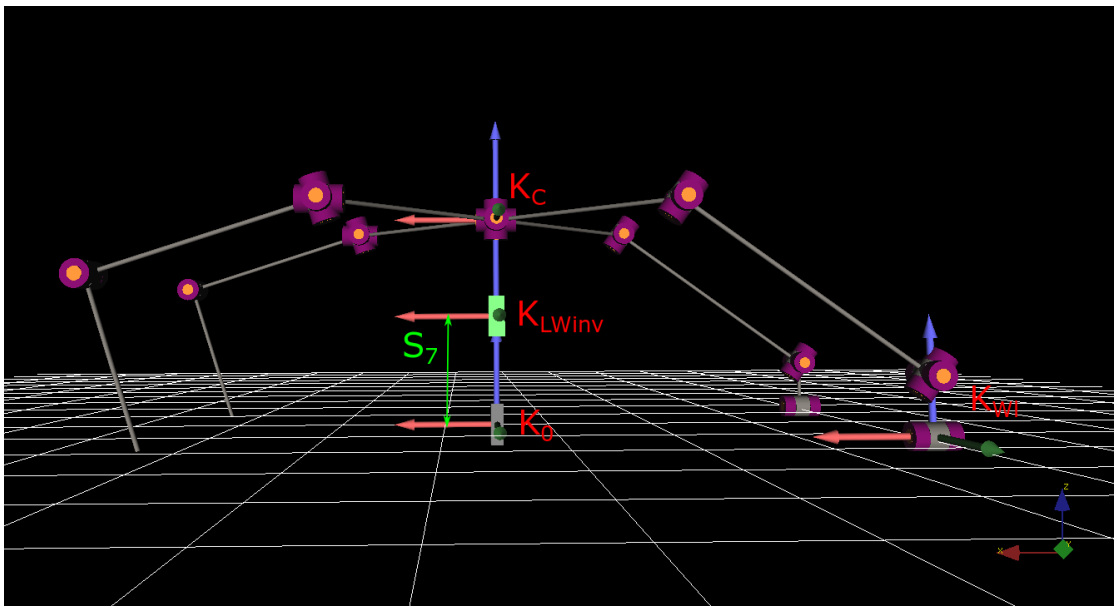

Figure 3-17: The prismatic joint that models the left wheel's height (z) with respect to the ground-fixed frame K_0.

Next, we define the loop closure conditions, the characteristic measurements, for the left wheel, between the left wheel's end effector frame (K_{Wl}) and the end frame of the z - prismatic joint (K_{LWinv}). Scalar measurements will be used, as before. The first measurement is the distance between the left wheel's end effector frame and the coplanar plane of K_{LWinv} in z direction. The second measurement is the cosine between the x axis of the left wheel's end effector frame (K_{Wl}) and the z axis of z-prismatic joint end frame (K_{LWinv}). The third measurement is the cosine between the y axis of the left wheel's end effector frame (K_{Wl}) and the z axis of z-prismatic joint end frame (K_{LWinv}). The vanishing of the last two measurements that indicates the closure of the loop means that the cosine will be zero, which means that x and y axis of K_{Wl} will be perpendicular to the z axis of K_{LWinv} (see Figure 3-18). With these three constraint equations we obtain a rolling behaviour for the wheel end effector frame throughout the x-y plane of the S_7 prismatic joint which gives us the flat rolling motion of the wheel on the z level of S_7 joint later on.

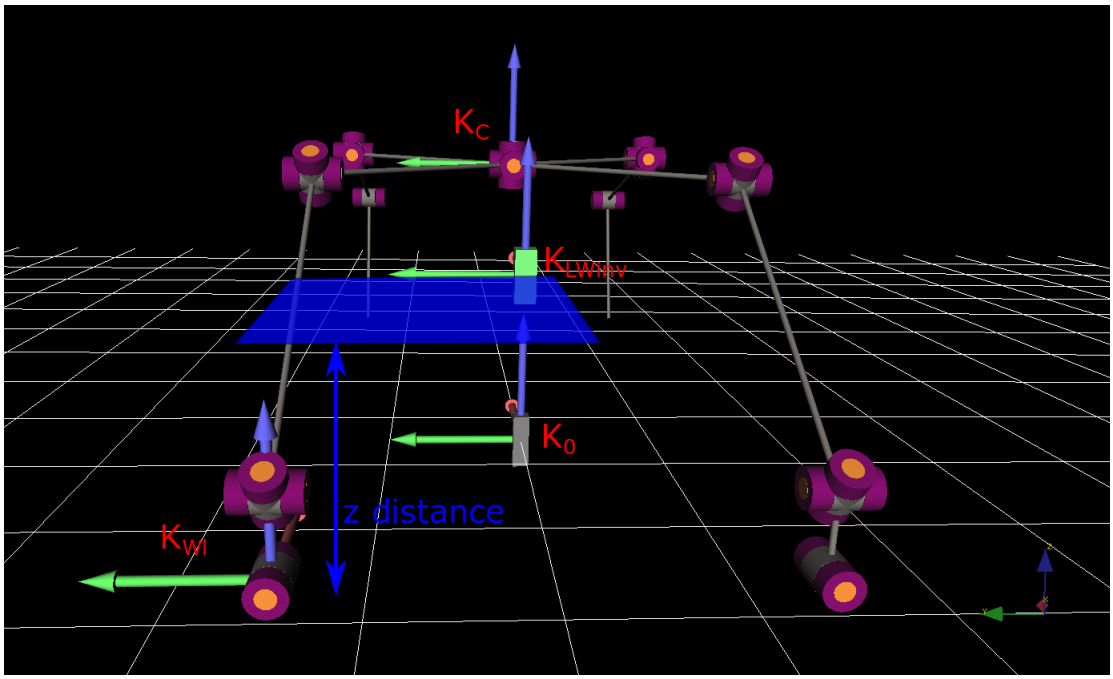

Figure 3-18: The characteristic measurements for the left wheel's loop closure.

An implicit solver is going to be used as on the legs. Out of all the variables affecting the left wheel's position (x_C, y_C, z_C, φ_{yaw}, φ_{pitch}, φ_{roll}, θ_{l6}, θ_{l5}, θ_{l4}, θ_{l3}, θ_{l2} and z_{Wl}, as seen in Figure 3-1) we select θ_{l6}, θ_{l4}, θ_{l3} and θ_{l2} to be treated as dependent and the remaining as generalized (input) variables. In order to ensure parallel wheel axes to the chassis for arbitrary straddle angle θ_{l5}, a constraint $\theta_{l4} = -\theta_{l5}$ simulating a parallel guidance mechanism is added to the overall closure conditions so the dependent variables become three (θ_{l6}, θ_{l3} and θ_{l2}). Like before, the number of dependent variables is the same with the number of constraint equations. Both of them are three, so the solver is able to calculate the inverse kinematics. Thus, having decided for the dependent variables, the closing conditions and the solver, the inverse kinematics of the left wheel are complete. By changing the cabin's pose (x_C, y_C, z_C,

φ_{yaw}, φ_{pitch}, φ_{roll}), the left wheel's lever straddle angle (θ_{l5}) and the left wheel's tip height (z_{Wl}), which are the independent variables, we can observe the dependent variables' values (θ_{l6}, θ_{l3}, θ_{l2}) needed for this change (see Figure 3-19).

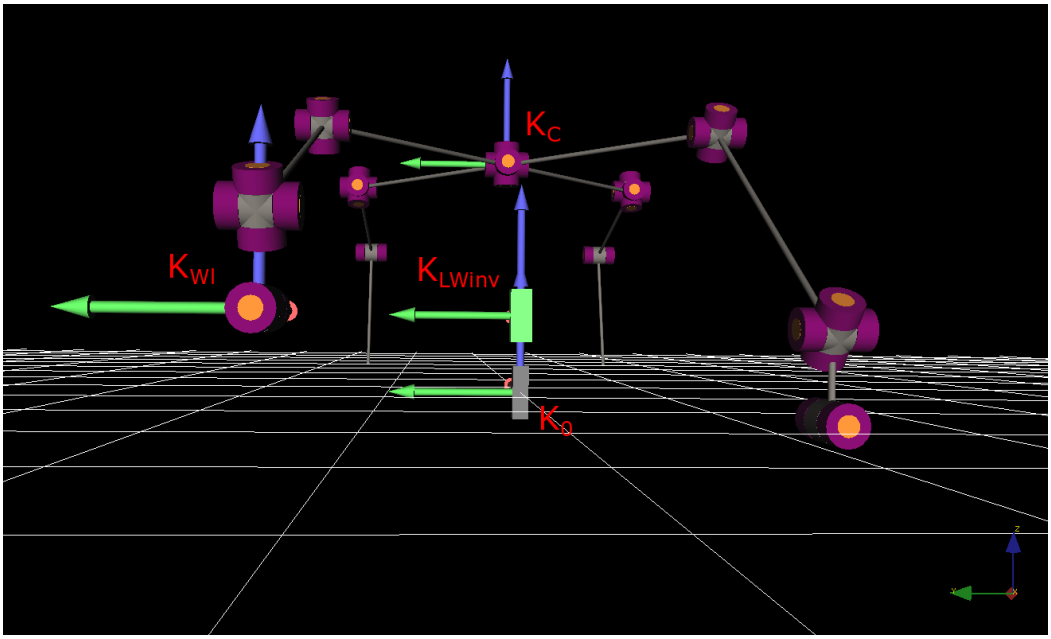

Figure 3-19: The inverse kinematics of the left wheel are complete and by changing the independent variables we can observe the dependent values needed for that change.

3.3.4 MOBILE inverse kinematics for the right wheel

Finally, we move to the mechanism's right wheel, creating a closed loop in order to do the inverse kinematics. For that purpose, we create a prismatic joint (S_8 – z direction) beginning from the ground-fixed inertia frame K_0 (see Figure 3-20). This joint will be used later as the right wheel's z movement with respect to K_0. We again don't need prismatic joints on x and y direction like on the legs because the wheels' exact x-y position in of no concern.

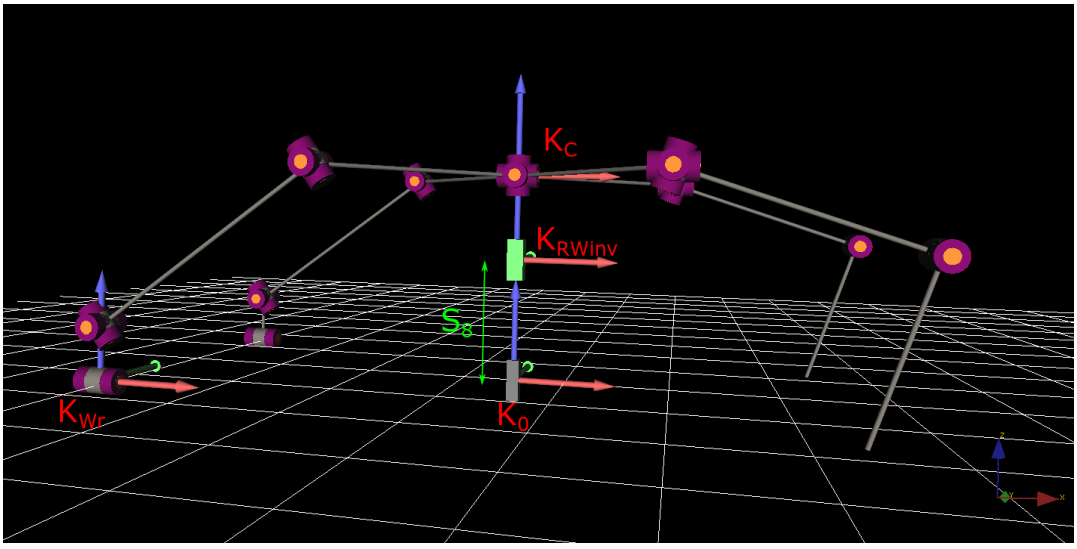

Figure 3-20: The prismatic joint that models the right wheel's height (z) with respect to the ground-fixed frame K_0.

Next, we define the loop closure conditions, the characteristic measurements, for the right wheel, between the right wheel's end effector frame (K_{Wr}) and the end frame of the z - prismatic joint (K_{RWinv}). Scalar measurements will be used, as before. The first measurement is the distance between the right wheel's end effector frame and the coplanar plane of K_{RWinv} in z direction. The second measurement is the cosine between the x axis of the right wheel's end effector frame (K_{Wr}) and the z axis of z-prismatic joint end frame (K_{RWinv}). The third measurement is the cosine between the y axis of the right wheel's end effector frame (K_{Wr}) and the z axis of z-prismatic joint end frame (K_{RWinv}) (see Figure 3-21). The vanishing of the last two measurements that indicates the closure of the loop means that the cosine will be zero, which means that x and y axis of K_{Wr} will be perpendicular to the z axis of K_{RWinv}. That is how we obtain a rolling behaviour for the wheel end effector frame throughout the x-y plane of the S_8 prismatic joint which gives us the flat rolling motion of the wheel on the z level of S_8 later on.

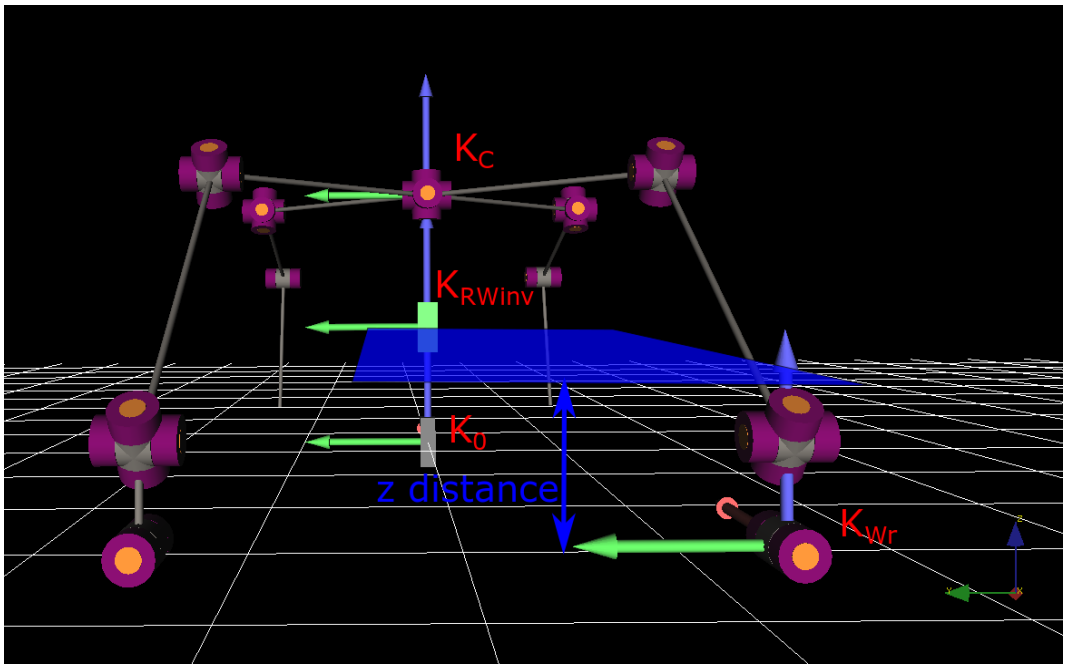

Figure 3-21: The characteristic measurements for the right wheel's loop closure.

An implicit solver is going to be used as before. Out of all the variables affecting the right wheel's position (x_C, y_C, z_C, φ_{yaw}, φ_{pitch}, φ_{roll}, θ_{r6}, θ_{r5}, θ_{r4}, θ_{r3}, θ_{r2} and z_{Wr}, as seen in Figure 3-1) we select θ_{r6}, θ_{r4}, θ_{r3} and θ_{r2} to be treated as dependent and the remaining as generalized (input) variables. In order to ensure parallel wheel axes to the chassis for arbitrary straddle angle θ_{r5}, a constraint $\theta_{r4} = -\theta_{r5}$ simulating a parallel guidance mechanism is added to the overall closure conditions so the dependent variables become three (θ_{r6}, θ_{r3} and θ_{r2}) as on the left wheel. Like before, we make sure the number of dependent variables is the same with the number of constraint equations. Both of them are three, so the solver is able to calculate the inverse kinematics. Thus, having decided for the dependent variables, the closing conditions and the solver, the inverse kinematics of the right wheel are complete. By changing the cabin's pose (x_C, y_C, z_C, φ_{yaw}, φ_{pitch}, φ_{roll}), the right wheel's lever straddle angle

(θ_{r5}) and the right wheel's tip height (z_{Wr}), which are the independent variables, we can observe the dependent variables' values (θ_{r6}, θ_{r3}, θ_{r2}) needed for this change (see Figure 3-22).

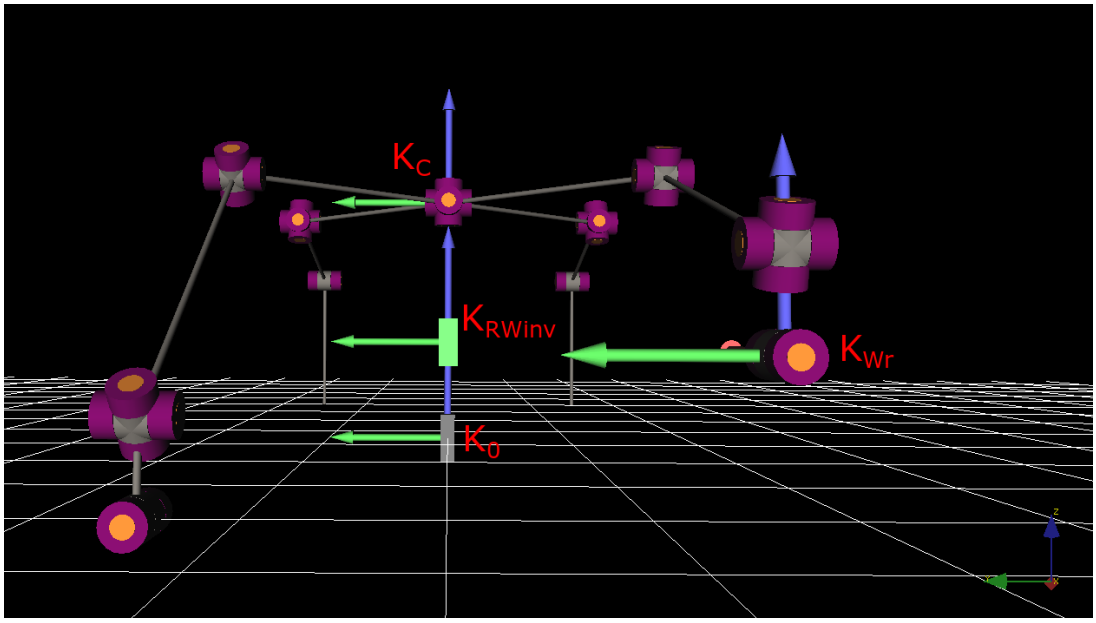

Figure 3-22: The inverse kinematics of the right wheel are complete and by changing the independent variables we can observe the dependent values needed for that change.

Complete kinematic model

Thus, the input coordinates of the mechanism are regarded to be the full cabin pose (x_C, y_C, z_C, φ_{yaw}, φ_{pitch} and φ_{roll}), the three feet tip coordinates for both legs (x_{Fl}, y_{Fl}, z_{Fl} and x_{Fr}, y_{Fr}, z_{Fr}), the legs' roll angles (ψ_{l1} and ψ_{r1}), the wheel levers' straddle angles (θ_{l5} and θ_{r5}) and the height of the wheel's contact point for both wheels (z_{Wl} and z_{Wr}). Altogether, the position of the system is described by the generalized (input) variables q and dependent variables β:

$$\begin{aligned} q &= \left[x_C, y_C, z_C, \varphi_{yaw}, \varphi_{pitch}, \varphi_{roll}, x_{Fl}, y_{Fl}, z_{Fl}, x_{Fr}, y_{Fr}, z_{Fr}, z_{Wl}, z_{Wr}, \psi_{l1}, \psi_{r1}, \theta_{l5}, \theta_{r5} \right]^T \\ \beta &= \left[\psi_{l2}, \psi_{l3}, \psi_{l4}, \psi_{r2}, \psi_{r3}, \psi_{r4}, \theta_{l1}, \theta_{l2}, \theta_{l3}, \theta_{l4}, \theta_{l6}, \theta_{r1}, \theta_{r2}, \theta_{r3}, \theta_{r4}, \theta_{r6}, x_{Wl}, y_{Wl}, x_{Wr}, y_{Wr} \right]^T \end{aligned} \tag{3.2}$$

All of these variables can be seen in Figure 3-1 which is repeated here for better visualization in Figure 3-23. There is a total of 18 degrees of freedom. The other 20 variables can be determined (as said, iteratively or analytically) from the generalized coordinates. Together with the first and second time derivatives, the dependencies are thus:

$$\begin{aligned} \beta &= \beta(q), \\ \dot{\beta} &= J\dot{q}, \\ \ddot{\beta} &= J\ddot{q} + \dot{J}\dot{q} \end{aligned} \tag{3.3}$$

The ground is considered to be flat throughout this work.

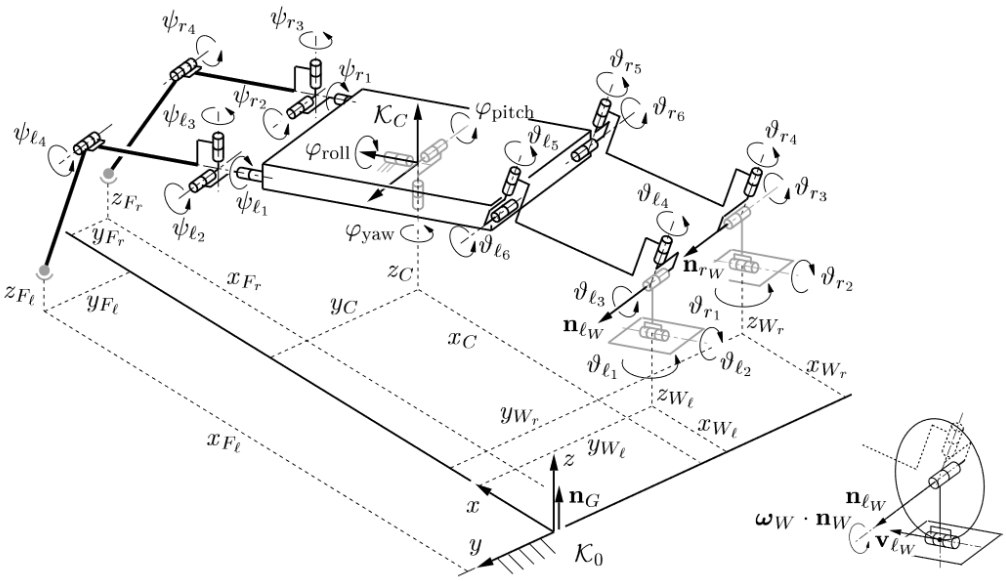

Figure 3-23: The variables of the kinetostatic skeleton of the hybrid excavator.

3.4 Analytical verification of the inverse kinematics

In this chapter MOBILE's iterative inverse kinematics are going to be compared with the analytical inverse kinematics of the mechanism. The closed form expressions of the analytical method are solved in MATLAB.

3.4.1 Analytical verification for the left leg

We begin by constructing the closed form equation that expresses the left leg's position with respect to the ground-fixed frame K_0. This distance can be calculated from the following Equation (3.4):

$$ {}^{0}_{0}r_{Fl} = {}^{0}T_{4} \cdot {}^{4}T_{8} \cdot {}^{8}T_{10} \cdot {}^{10}_{10}r_{Fl} \tag{3.4} $$

In Figure 3-24 the left leg's frame distribution can be seen, regarding Equation (3.4)

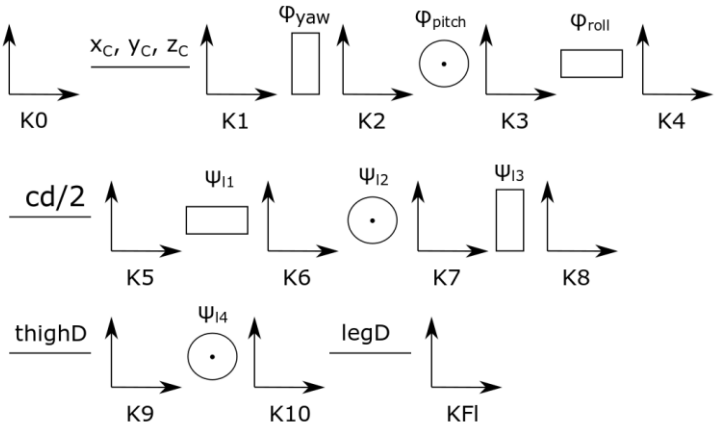

Figure 3-24: Frames distribution to the components regarding the left leg tip distance from the inertia frame K0.

In Figure 3-25 the 2D diagram of the left leg can be seen with all the joints and links that construct it and in Figure 3-27 the cabin's geometry can be seen.

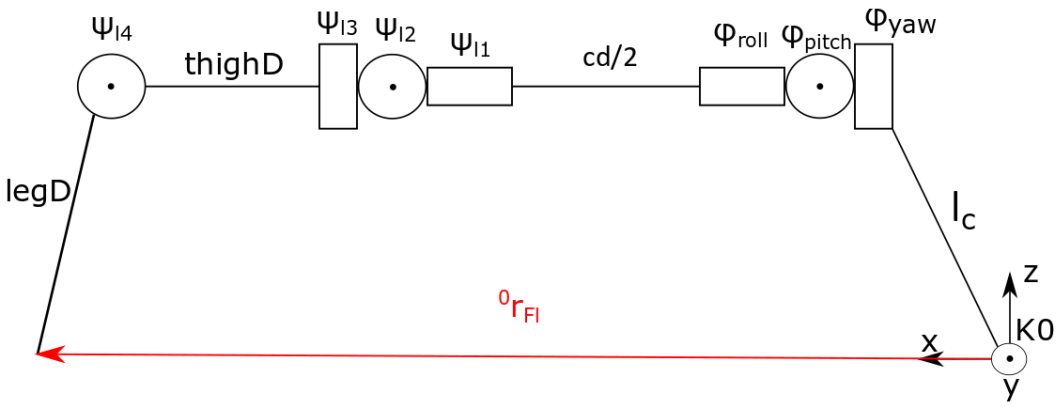

Figure 3-25: 2D diagram of the left leg tip distance from the inertia frame K0.

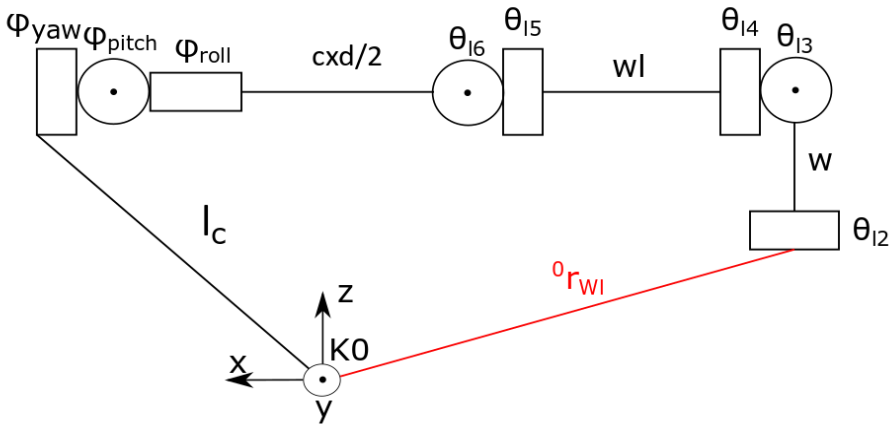

Figure 3-26: 2D diagram describing the geometry of the wheel levers and wheels.

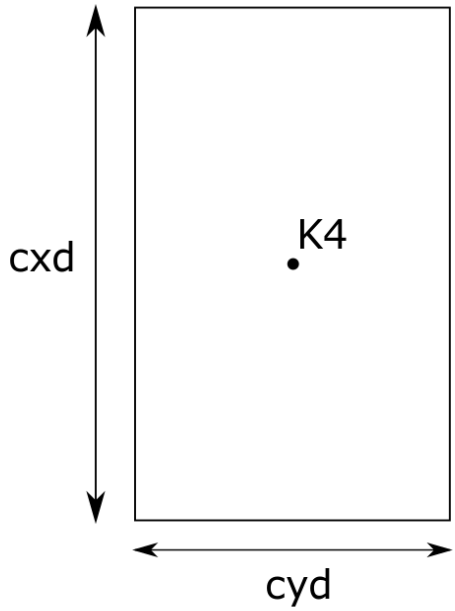

Figure 3-27: Vehicle's cabin geometry.

In Table 3-1, the geometrical values for the mechanism are listed. If not stated otherwise, these are the values used throughout the thesis. These values can all be visualized in Figure 3-28.

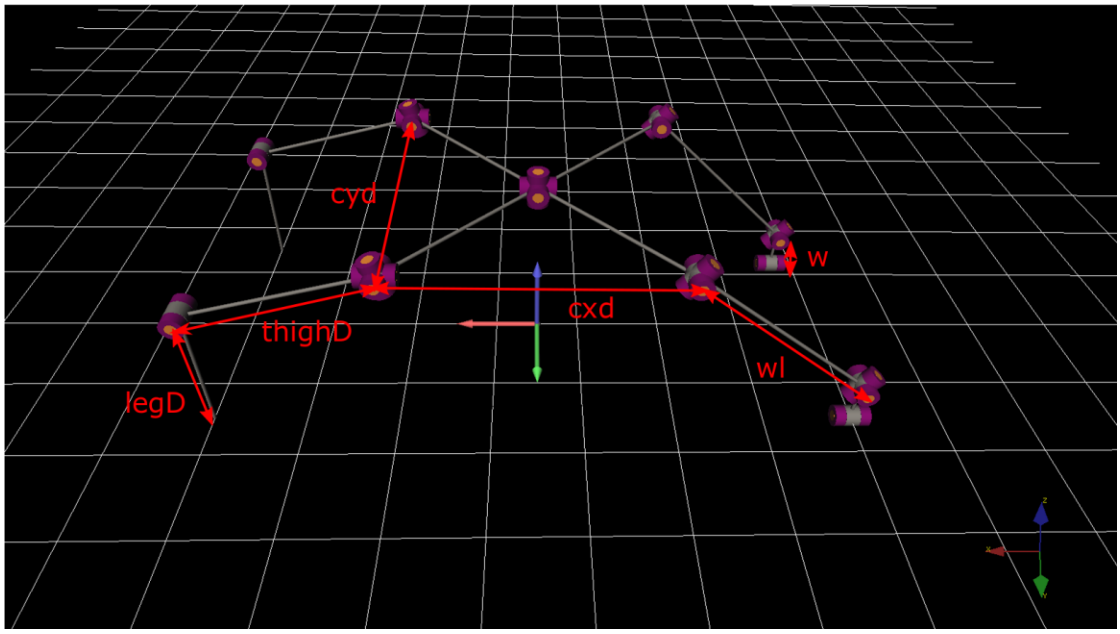

Figure 3-28: The robot's geometrical values.

Table 3-1: Excavator's geometrical values.

cxd (m)	cyd (m)	thighD (m)	legD (m)	wl (m)	w (m)
3.0	3.0	2.0	1.5	2.5	0.5

The homogeneous transformation matrices stated in Eq. can be expressed therefore as:

$$
{}^{0}R_4 = rot_z(yaw)\cdot rot_y(pitch)\cdot rot_x(roll) =
$$

$$
\begin{bmatrix} c_{\varphi_{yaw}} & -s_{\varphi_{yaw}} & 0 \\ s_{\varphi_{yaw}} & c_{\varphi_{yaw}} & 0 \\ 0 & 0 & 1 \end{bmatrix}\cdot\begin{bmatrix} c_{\varphi_{pitch}} & 0 & s_{\varphi_{pitch}} \\ 0 & 1 & 0 \\ -s_{\varphi_{pitch}} & 0 & c_{\varphi_{pitch}} \end{bmatrix}\cdot\begin{bmatrix} 1 & 0 & 0 \\ 0 & c_{\varphi_{roll}} & -s_{\varphi_{roll}} \\ 0 & s_{\varphi_{roll}} & c_{\varphi_{roll}} \end{bmatrix}\Rightarrow
$$

$$
{}^{0}R_4 = \begin{bmatrix} c_{\varphi_{yaw}}c_{\varphi_{pitch}} & -s_{\varphi_{yaw}}c_{\varphi_{roll}} + c_{\varphi_{yaw}}s_{\varphi_{pitch}}s_{\varphi_{roll}} & s_{\varphi_{roll}}s_{\varphi_{yaw}} + c_{\varphi_{roll}}s_{\varphi_{pitch}}c_{\varphi_{yaw}} \\ s_{\varphi_{yaw}}c_{\varphi_{pitch}} & c_{\varphi_{yaw}}c_{\varphi_{roll}} + s_{\varphi_{yaw}}s_{\varphi_{pitch}}s_{\varphi_{roll}} & -s_{\varphi_{roll}}c_{\varphi_{yaw}} + c_{\varphi_{roll}}s_{\varphi_{pitch}}s_{\varphi_{yaw}} \\ -s_{\varphi_{pitch}} & c_{\varphi_{pitch}}s_{\varphi_{roll}} & c_{\varphi_{roll}}c_{\varphi_{pitch}} \end{bmatrix}\Rightarrow \tag{3.5}
$$

$$
{}^{0}T_4 = \begin{bmatrix} c_{\varphi_{yaw}}c_{\varphi_{pitch}} & -s_{\varphi_{yaw}}c_{\varphi_{roll}} + c_{\varphi_{yaw}}s_{\varphi_{pitch}}s_{\varphi_{roll}} & s_{\varphi_{roll}}s_{\varphi_{yaw}} + c_{\varphi_{roll}}s_{\varphi_{pitch}}c_{\varphi_{yaw}} & x_c \\ s_{\varphi_{yaw}}c_{\varphi_{pitch}} & c_{\varphi_{yaw}}c_{\varphi_{roll}} + s_{\varphi_{yaw}}s_{\varphi_{pitch}}s_{\varphi_{roll}} & -s_{\varphi_{roll}}c_{\varphi_{yaw}} + c_{\varphi_{roll}}s_{\varphi_{pitch}}s_{\varphi_{yaw}} & y_c \\ -s_{\varphi_{pitch}} & c_{\varphi_{pitch}}s_{\varphi_{roll}} & c_{\varphi_{roll}}c_{\varphi_{pitch}} & z_c \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

$$
{}^{4}R_8 = rot_x(\psi_{l1})\cdot rot_y(\psi_{l2})\cdot rot_z(\psi_{l3}) =
$$

$$
\begin{bmatrix} 1 & 0 & 0 \\ 0 & c_{\psi_{l1}} & -s_{\psi_{l1}} \\ 0 & s_{\psi_{l1}} & c_{\psi_{l1}} \end{bmatrix}\cdot\begin{bmatrix} c_{\psi_{l2}} & 0 & s_{\psi_{l2}} \\ 0 & 1 & 0 \\ -s_{\psi_{l2}} & 0 & c_{\psi_{l2}} \end{bmatrix}\cdot\begin{bmatrix} c_{\psi_{l3}} & -s_{\psi_{l3}} & 0 \\ s_{\psi_{l3}} & c_{\psi_{l3}} & 0 \\ 0 & 0 & 1 \end{bmatrix}\Rightarrow
$$

$$
{}^{4}R_8 = \begin{bmatrix} c_{\psi_{l2}}c_{\psi_{l3}} & -c_{\psi_{l2}}s_{\psi_{l3}} & s_{\psi_{l2}} \\ s_{\psi_{l1}}s_{\psi_{l2}}c_{\psi_{l3}} + c_{\psi_{l1}}s_{\psi_{l3}} & -s_{\psi_{l1}}s_{\psi_{l2}}s_{\psi_{l3}} + c_{\psi_{l1}}c_{\psi_{l3}} & -s_{\psi_{l1}}c_{\psi_{l2}} \\ -s_{\psi_{l2}}c_{\psi_{l1}}c_{\psi_{l3}} + s_{\psi_{l1}}s_{\psi_{l3}} & s_{\psi_{l2}}s_{\psi_{l3}}c_{\psi_{l1}} + s_{\psi_{l1}}c_{\psi_{l3}} & c_{\psi_{l1}}c_{\psi_{l2}} \end{bmatrix}\Rightarrow \tag{3.6}
$$

$$
{}^{4}T_8 = \begin{bmatrix} c_{\psi_{l2}}c_{\psi_{l3}} & -c_{\psi_{l2}}s_{\psi_{l3}} & s_{\psi_{l2}} & cxd/2 \\ s_{\psi_{l1}}s_{\psi_{l2}}c_{\psi_{l3}} + c_{\psi_{l1}}s_{\psi_{l3}} & -s_{\psi_{l1}}s_{\psi_{l2}}s_{\psi_{l3}} + c_{\psi_{l1}}c_{\psi_{l3}} & -s_{\psi_{l1}}c_{\psi_{l2}} & cyd/2 \\ -s_{\psi_{l2}}c_{\psi_{l1}}c_{\psi_{l3}} + s_{\psi_{l1}}s_{\psi_{l3}} & s_{\psi_{l2}}s_{\psi_{l3}}c_{\psi_{l1}} + s_{\psi_{l1}}c_{\psi_{l3}} & c_{\psi_{l1}}c_{\psi_{l2}} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

And the final transformation matrix along with the distance vector from K_{10} to K_{Fl} with respect to K_{10} are:

$$
{}^{8}R_{10} = rot_y(\psi_{l4}) = \begin{bmatrix} c_{\psi_{l4}} & 0 & s_{\psi_{l4}} \\ 0 & 1 & 0 \\ -s_{\psi_{l4}} & 0 & c_{\psi_{l4}} \end{bmatrix}\Rightarrow
$$

$$
{}^{8}T_{10}\,{}^{10}r_{Fl} = \begin{bmatrix} c_{\psi_{l4}} & 0 & s_{\psi_{l4}} & thighD \\ 0 & 1 & 0 & 0 \\ -s_{\psi_{l4}} & 0 & c_{\psi_{l4}} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} legD \\ 0 \\ 0 \\ 1 \end{bmatrix} \tag{3.7}
$$

And so, Equation (3.4). becomes:

$$\begin{bmatrix} x_{Fl} \\ y_{Fl} \\ z_{Fl} \\ 1 \end{bmatrix} = \begin{bmatrix} c_{\varphi_{yaw}} c_{\varphi_{pitch}} & -s_{\varphi_{yaw}} c_{\varphi_{roll}} + c_{\varphi_{yaw}} s_{\varphi_{pitch}} s_{\varphi_{roll}} & s_{\varphi_{roll}} s_{\varphi_{yaw}} + c_{\varphi_{roll}} s_{\varphi_{pitch}} c_{\varphi_{yaw}} & x_c \\ s_{\varphi_{yaw}} c_{\varphi_{pitch}} & c_{\varphi_{yaw}} c_{\varphi_{roll}} + s_{\varphi_{yaw}} s_{\varphi_{pitch}} s_{\varphi_{roll}} & -s_{\varphi_{roll}} c_{\varphi_{yaw}} + c_{\varphi_{roll}} s_{\varphi_{pitch}} s_{\varphi_{yaw}} & y_c \\ -s_{\varphi_{pitch}} & c_{\varphi_{pitch}} s_{\varphi_{roll}} & c_{\varphi_{roll}} c_{\varphi_{pitch}} & z_c \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

$$\begin{bmatrix} c_{\psi_{l2}} c_{\psi_{l3}} & -c_{\psi_{l2}} s_{\psi_{l3}} & s_{\psi_{l2}} & cxd/2 \\ s_{\psi_{l1}} s_{\psi_{l2}} c_{\psi_{l3}} + c_{\psi_{l1}} s_{\psi_{l3}} & -s_{\psi_{l1}} s_{\psi_{l2}} s_{\psi_{l3}} + c_{\psi_{l1}} c_{\psi_{l3}} & -s_{\psi_{l1}} c_{\psi_{l2}} & cyd/2 \\ -s_{\psi_{l2}} c_{\psi_{l1}} c_{\psi_{l3}} + s_{\psi_{l1}} s_{\psi_{l3}} & s_{\psi_{l2}} s_{\psi_{l3}} c_{\psi_{l1}} + s_{\psi_{l1}} c_{\psi_{l3}} & c_{\psi_{l1}} c_{\psi_{l2}} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (3.8)$$

$$\begin{bmatrix} c_{\psi_{l4}} & 0 & s_{\psi_{l4}} & thighD \\ 0 & 1 & 0 & 0 \\ -s_{\psi_{l4}} & 0 & c_{\psi_{l4}} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} . \begin{bmatrix} legD \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

Therefore, the x, y and z position of the left foreleg with respect to K_0 can be calculated from the next three equations, Equation (3.9), Equation (3.10) and Equation (3.11):

$$\begin{aligned} & x_{Flcalculated} = x_c - \left(c_{\varphi_{roll}} \cdot s_{\varphi_{yaw}} - c_{\varphi_{yaw}} \cdot s_{\varphi_{pitch}} \cdot s_{\varphi_{roll}} \right) \cdot \\ & \left(\frac{cyd}{2} + \left(thighD + c_{\psi_{l4}} \cdot legD \right) \cdot \left(c_{\psi_{l1}} \cdot s_{\psi_{l3}} + c_{\psi_{l3}} \cdot s_{\psi_{l1}} \cdot s_{\psi_{l2}} \right) + c_{\psi_{l2}} \cdot legD \cdot s_{\psi_{l1}} \cdot s_{\psi_{l4}} \right) + \\ & \left(\left(thighD + c_{\psi_{l4}} \cdot legD \right) \cdot \left(s_{\psi_{l1}} \cdot s_{\psi_{l3}} - c_{\psi_{l1}} \cdot c_{\psi_{l3}} \cdot s_{\psi_{l2}} \right) - c_{\psi_{l1}} \cdot c_{\psi_{l2}} \cdot legD \cdot s_{\psi_{l4}} \right) \cdot \\ & \left(s_{\varphi_{roll}} \cdot s_{\varphi_{yaw}} + c_{\varphi_{roll}} \cdot c_{\varphi_{yaw}} \cdot s_{\varphi_{pitch}} \right) + \\ & c_{\varphi_{pitch}} \cdot c_{\varphi_{yaw}} \cdot \left(\frac{cxd}{2} + c_{\psi_{l2}} \cdot c_{\psi_{l3}} \cdot \left(thighD + c_{\psi_{l4}} \cdot legD \right) - legD \cdot s_{\psi_{l2}} \cdot s_{\psi_{l4}} \right) \end{aligned} \quad (3.9)$$

$$\begin{aligned} & y_{Flcalculated} = y_c + \left(c_{\varphi_{roll}} \cdot c_{\varphi_{yaw}} + s_{\varphi_{yaw}} \cdot s_{\varphi_{pitch}} \cdot s_{\varphi_{roll}} \right) \cdot \\ & \left(\frac{cyd}{2} + \left(thighD + c_{\psi_{l4}} \cdot legD \right) \cdot \left(c_{\psi_{l1}} \cdot s_{\psi_{l3}} + c_{\psi_{l3}} \cdot s_{\psi_{l1}} \cdot s_{\psi_{l2}} \right) + c_{\psi_{l2}} \cdot legD \cdot s_{\psi_{l1}} \cdot s_{\psi_{l4}} \right) - \\ & \left(\left(thighD + c_{\psi_{l4}} \cdot legD \right) \cdot \left(s_{\psi_{l1}} \cdot s_{\psi_{l3}} - c_{\psi_{l1}} \cdot c_{\psi_{l3}} \cdot s_{\psi_{l2}} \right) - c_{\psi_{l1}} \cdot c_{\psi_{l2}} \cdot legD \cdot s_{\psi_{l4}} \right) \cdot \\ & \left(s_{\varphi_{roll}} \cdot c_{\varphi_{yaw}} - c_{\varphi_{roll}} \cdot s_{\varphi_{yaw}} \cdot s_{\varphi_{pitch}} \right) + \\ & c_{\varphi_{pitch}} \cdot s_{\varphi_{yaw}} \cdot \left(\frac{cxd}{2} + c_{\psi_{l2}} \cdot c_{\psi_{l3}} \cdot \left(thighD + c_{\psi_{l4}} \cdot legD \right) - legD \cdot s_{\psi_{l2}} \cdot s_{\psi_{l4}} \right) \end{aligned} \quad (3.10)$$

$$z_{Flcalculated} = z_c - s_{\varphi_{pitch}} \cdot \begin{pmatrix} \frac{cxd}{2} + c_{\psi_{l2}} \cdot c_{\psi_{l3}} \cdot \left(thighD + c_{\psi_{l4}} \cdot legD \right) \\ -legD \cdot s_{\psi_{l2}} \cdot s_{\psi_{l4}} \end{pmatrix} +$$
$$c_{\varphi_{pitch}} \cdot s_{\varphi_{roll}} \begin{pmatrix} \frac{cyd}{2} + \left(thighD + c_{\psi_{l4}} \cdot legD \right) \cdot \left(c_{\psi_{l1}} \cdot s_{\psi_{l3}} + c_{\psi_{l3}} \cdot s_{\psi_{l1}} \cdot s_{\psi_{l2}} \right) \\ +c_{\psi_{l2}} \cdot legD \cdot s_{\psi_{l1}} \cdot s_{\psi_{l4}} \end{pmatrix} + \quad (3.11)$$
$$c_{\varphi_{pitch}} \cdot c_{\varphi_{roll}} \cdot \begin{pmatrix} \left(thighD + c_{\psi_{l4}} \cdot legD \right) \cdot \left(s_{\psi_{l1}} \cdot s_{\psi_{l3}} - c_{\psi_{l1}} \cdot c_{\psi_{l3}} \cdot s_{\psi_{l2}} \right) \\ -c_{\psi_{l1}} \cdot c_{\psi_{l2}} \cdot legD \cdot s_{\psi_{l4}} \end{pmatrix}$$

The next step is to compare and validate the MOBILE iterative inverse kinematics values with the calculated values from the above equations. That is done following the next procedure. We input the independent variables (x_{Fl}, y_{Fl}, z_{Fl} and ψ_{l1}) in a width of values and we store the output values of the dependent variables (ψ_{l2}, ψ_{l3} and ψ_{l4}). Afterwards, we input the values of these dependent variables to the above equations in MATLAB and we get as output the position of the left foreleg tip in the xyz space with respect to K_0, numerically. Afterwards, we compare the x_{Fl}, y_{Fl}, z_{Fl} values that we chose to input with the numerically calculated values $x_{Flcalculated}$, $y_{Flcalculated}$, $z_{Flcalculated}$ and we check the divergence between them.

1st Verification test

We input to MOBILE the next range of x_{Fl} values, [1.80m, 4.30m] with a step of 0.01m. For every x_{Fl} value we get a value for each one of the dependent variables and store them. While doing that, we keep the rest independent variables, except x_{Fl}, that affect the left foreleg's position at a fixed value. The values of the variables that affect the left foreleg's position can be seen in Table 3-2:

Table 3-2: The values of the independent variables affecting the left foreleg's position for the 1st verification.

x_{Fl} (m)	y_{Fl} (m)	z_{Fl} (m)	ψ_{l1} (deg)	x_C (m)	y_C (m)	z_C (m)	ϕ_{yaw} (deg)	ϕ_{pitch} (deg)	ϕ_{roll} (deg)
[1.80, 4.30]	1.50	0	0º	0	0	2	0º	0º	0º

As a result, we get the following diagrams:

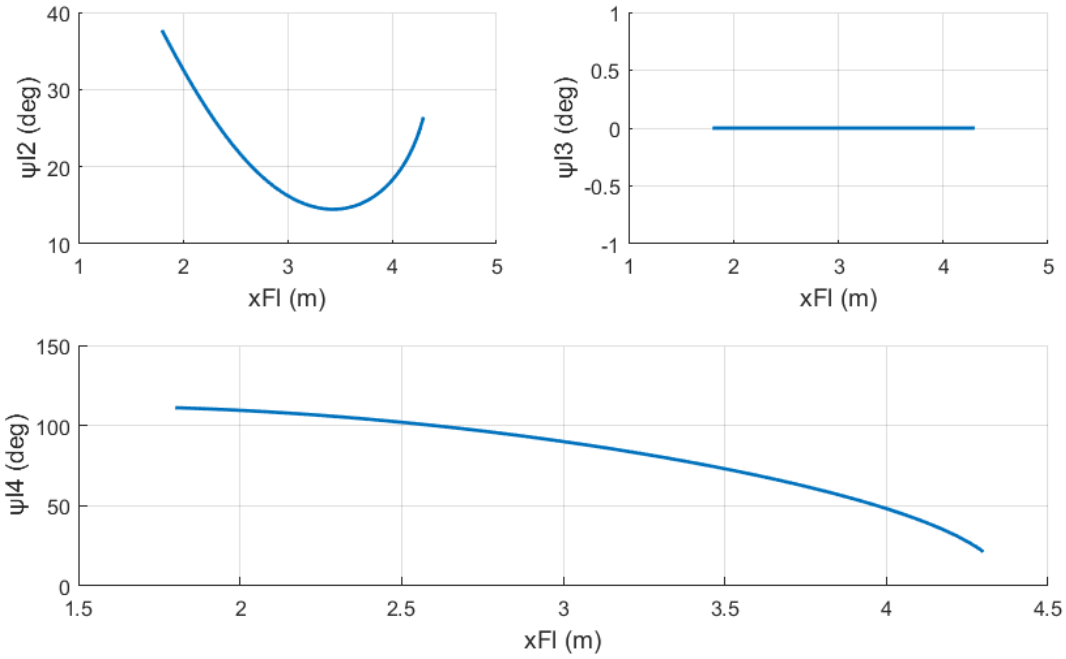

Figure 3-29: MOBILE's ψ_{l2}, ψ_{l3} and ψ_{l4} values for x_{Fl} range [1.80, 4.30].

Then, we will input the ψ_{l2}, ψ_{l3} and ψ_{l4} values that we got as an output from MOBILE, to Equation (3.4) and calculate the $x_{Flcalculated}$, $y_{Flcalculated}$, $z_{Flcalculated}$ values using MATLAB. After the calculations we get the following diagrams in Figure 3-30:

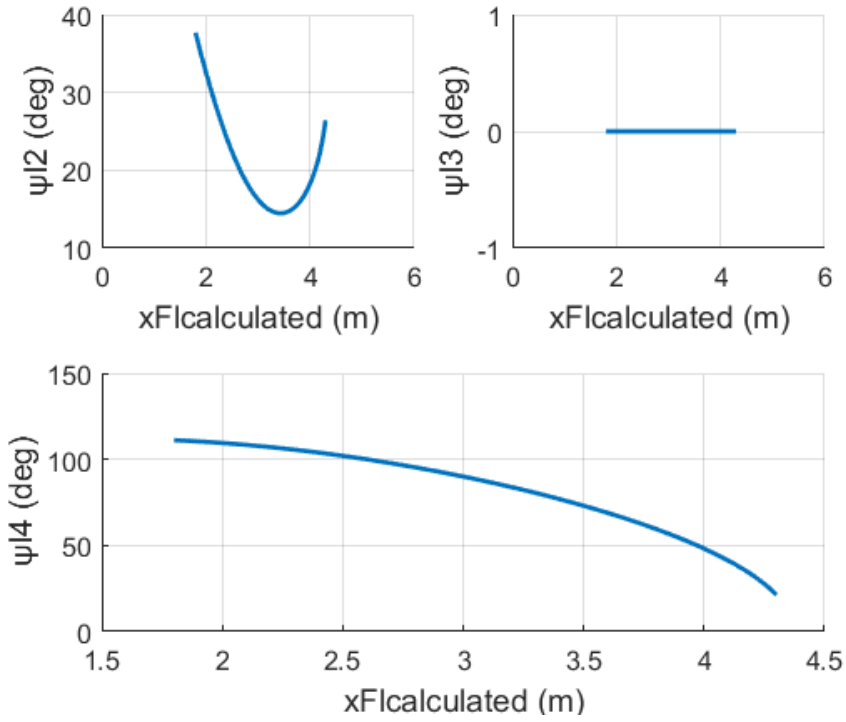

Figure 3-30: MATLAB calculated values for x_{Fl} from MOBILE's output ψ_{l2}, ψ_{l3} and ψ_{l4} values. x_{Fl} to ψ_{l2} diagram.

Now having the calculated left foreleg's tip position values, we can compare them with the initial left foreleg's tip position values we inputted to MOBILE and get the next error diagrams on Figure 3-31:

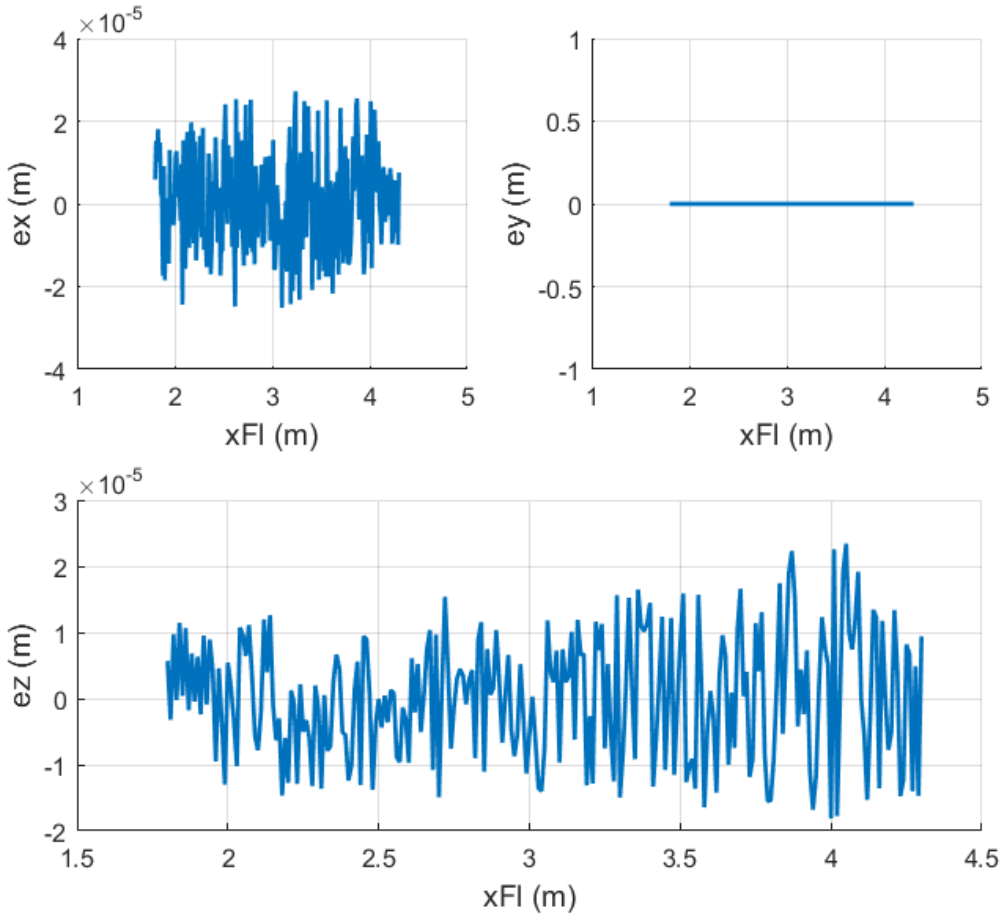

Figure 3-31: xFl, yFl and zFl errors calculated as (xFl-xFlcalculated), (yFl-yFlcalculated) and (zFl-zFlcalculated) accordingly when the varying value is xFl.

We observe that the errors are of the order of 10^{-5} m, which is acceptably small.

2nd Verification test

For the 2nd test, much like the previous one, we input to MOBILE the next range of y_{Fl} values, [-0.5m, 3.50m] with a step of 0.01m. For every y_{Fl} value we get a value for each one of the dependent variables and store them. While doing that, we keep the rest independent variables, except y_{Fl}, that affect the left foreleg's position at a fixed value. The values of the variables that affect the left foreleg's position can be seen in Table 3-3:

Table 3-3: The values of the independent variables affecting the left foreleg's position for the 1st verification.

x_{Fl} (m)	y_{Fl} (m)	z_{Fl} (m)	ψ_{l1} (deg)	x_C (m)	y_C (m)	z_C (m)	ϕ_{yaw} (deg)	ϕ_{pitch} (deg)	ϕ_{roll} (deg)

3.5	[-0.50, 3.5]	0	0°	0	0	2	0°	0°	0°

Following the same procedure as before, we use the ψ_{l2}, ψ_{l3} and ψ_{l4} values we get from MOBILE to calculate in MATLAB $x_{Flcalculated}$ $y_{Flcalculated}$ $z_{Flcalculated}$ and then compare these with the values we inputted to MOBILE in the first place. Below the error diagrams can be seen in Figure 3-32.

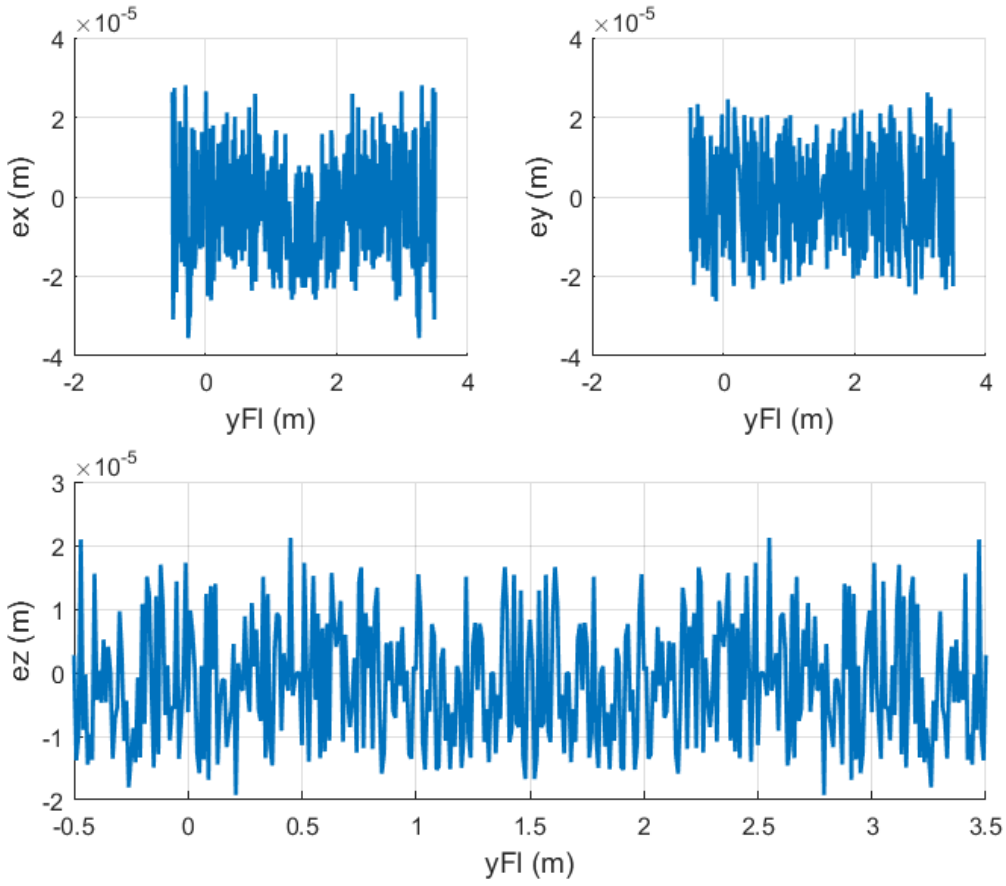

Figure 3-32: x_{Fl}, y_{Fl} and z_{Fl} errors calculated as (x_{Fl}-$x_{Flcalculated}$), (y_{Fl}-$y_{Flcalculated}$) and (z_{Fl}-$z_{Flcalculated}$) accordingly when the varying value is y_{Fl}.

We see that the errors are of the order of 10^{-5} m, which is acceptably small.

3rd Verification test

Moving on to the third kinematics verification test, we input to MOBILE the next range of z_{Fl} values, [-0.5m, 3.50m] with a step of 0.01m. For every z_{Fl} value we get a value for each one of the dependent variables and store them. While doing that, we keep the rest independent variables, except z_{Fl}, that affect the left foreleg's position at a fixed value. The values of the variables that affect the left foreleg's position for the third test can be seen in Table 3-4:

Table 3-4: The values of the variables affecting the left foreleg's position for the third verification.

x_{Fl} (m)	y_{Fl} (m)	z_{Fl} (m)	ψ_{l1} (deg)	x_C (m)	y_C (m)	z_C (m)	ϕ_{yaw} (deg)	ϕ_{pitch} (deg)	ϕ_{roll} (deg)
3.5	1.50	[-0.50, 3.50]	0°	0	0	2	0°	0°	0°

We again use the stored MOBILE values of the dependent variables ψ_{l2}, ψ_{l3} and ψ_{l4} to calculate the analytic values of the left foreleg's tip position $x_{Flcalculated}$ $y_{Flcalculated}$ and $z_{Flcalculated}$ and compare them with the ones we inputted to MOBILE. The error diagrams can be seen in Figure 3-33.

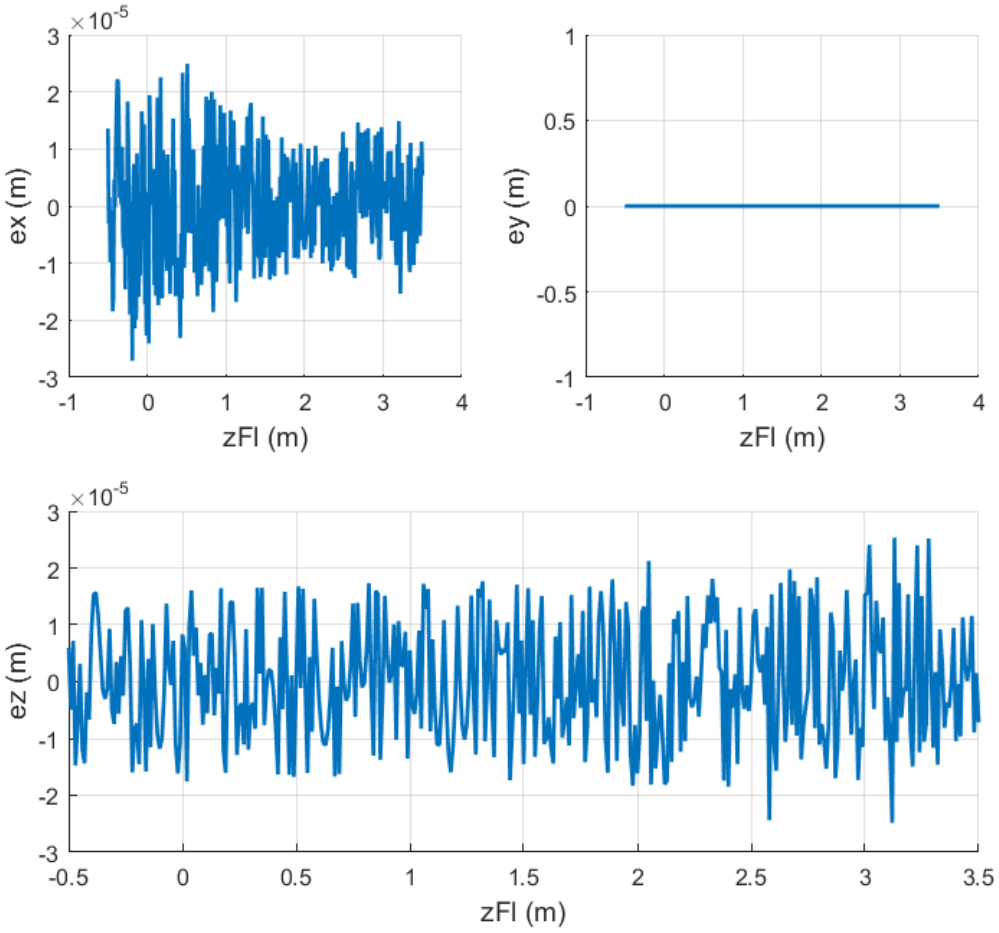

Figure 3-33: x_{Fl}, y_{Fl} and z_{Fl} errors calculated as (x_{Fl}-$x_{Flcalculated}$), (y_{Fl}-$y_{Flcalculated}$) and (z_{Fl}-$z_{Flcalculated}$) accordingly when the varying value is z_{Fl}.

We observe that the errors are of the order of 10^{-5} m, which is acceptably small.

4th Verification test

For the last verification test, we input to MOBILE the next range of ψ_{l1} values, [-30° , 30°] with a step of 0.1°. For every ψ_{l1} value we get a value for each one of the dependent variables and store them. While doing that, we keep the rest independent variables, except

ψ_{l1}, that affect the left foreleg's position at a fixed value. The values of the variables that affect the left foreleg's position for the fourth test can be seen in Table 3-5:

Table 3-5: The values of the variables affecting the left foreleg's position for the fourth verification.

x_{Fl} (m)	y_{Fl} (m)	z_{Fl} (m)	ψ_{l1} (deg)	x_C (m)	y_C (m)	z_C (m)	ϕ_{yaw} (deg)	ϕ_{pitch} (deg)	ϕ_{roll} (deg)
3.5	1.50	0	[-30º, 30º]	0	0	2	0º	0º	0º

Next, the error diagrams can be seen in Figure 3-34, made following the procedure described in the previous verifications.

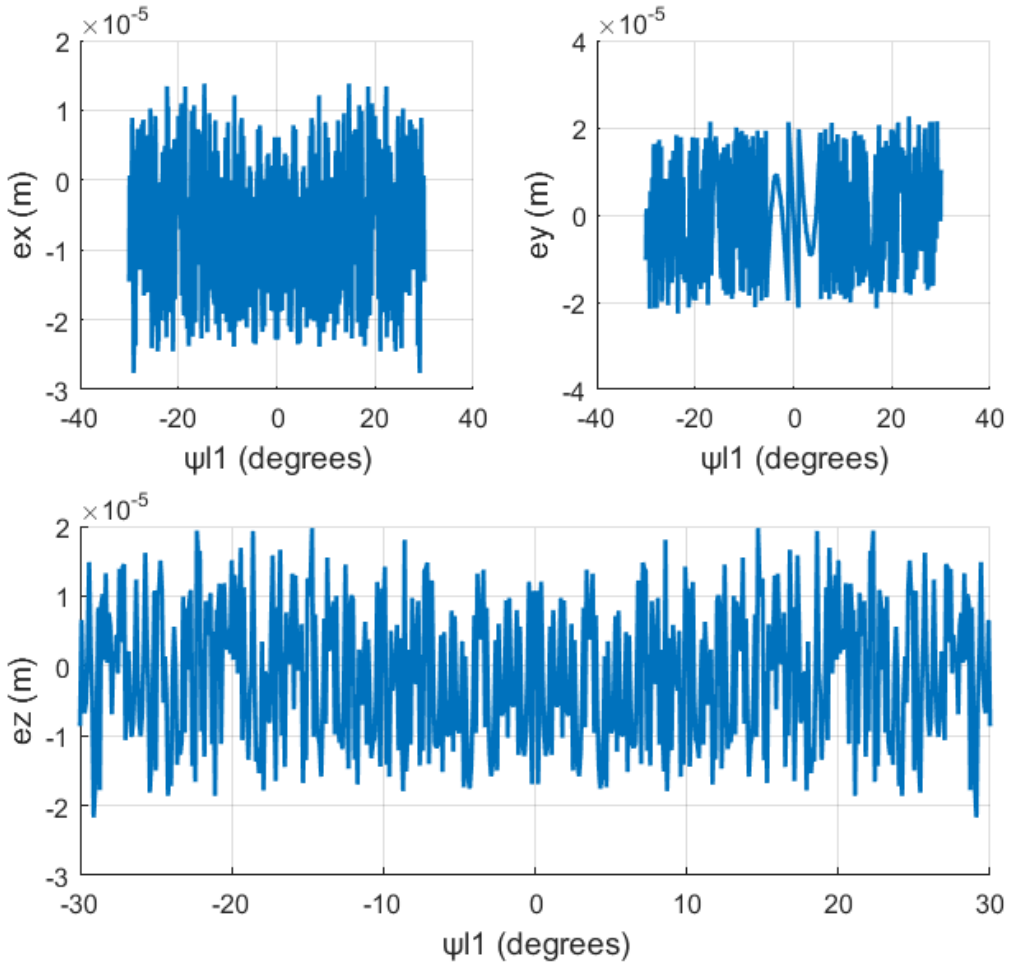

Figure 3-34: x_{Fl} error calculated as (x_{Fl}-$x_{Flcalculated}$) when the varying value is ψ_{l1}.

We observe that the errors are of the order of 10^{-5} m, which is acceptably small.

3.4.2 Analytical verification for the right leg

We proceed by constructing the closed form equation that expresses the right leg's position in respect with the ground-fixed frame K_0. This distance can be calculated from the following Equation (3.12):

$$ {}^{0}_{0}r_{Fr} = {}^{0}T_{4} \cdot {}^{4}T_{8} \cdot {}^{8}T_{10} \cdot {}^{10}_{10}r_{Fr} \tag{3.12} $$

In Figure 3-35 the right leg's frame distribution can be seen, regarding Equation (3.12).

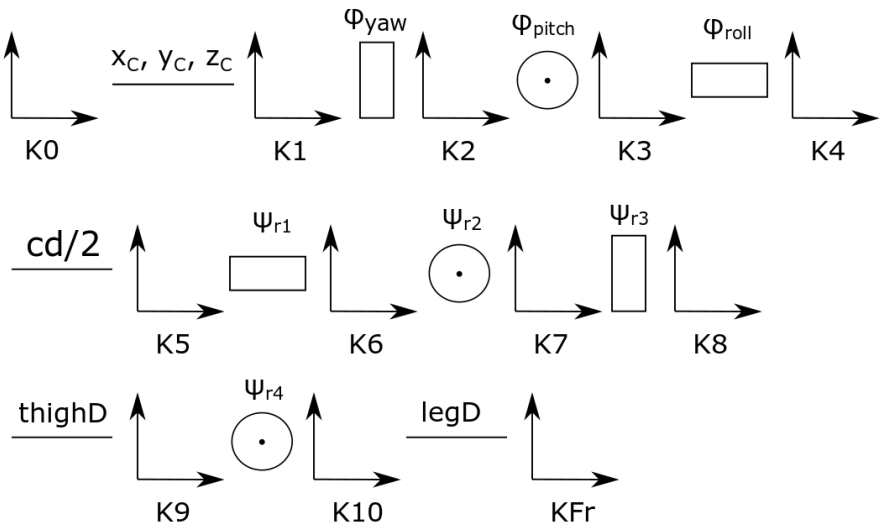

Figure 3-35: Frames distribution to the components regarding the right leg tip distance from the inertia frame K0.

In Figure 3-36 the 2D diagram of the right leg can be seen with all the joints and links that construct it. The cabin's geometry is the same as described in the case of the left leg and can be seen in Figure 3-27, in Figure 3-28 the robot's geometrical variables can be seen in MOBILE environment.

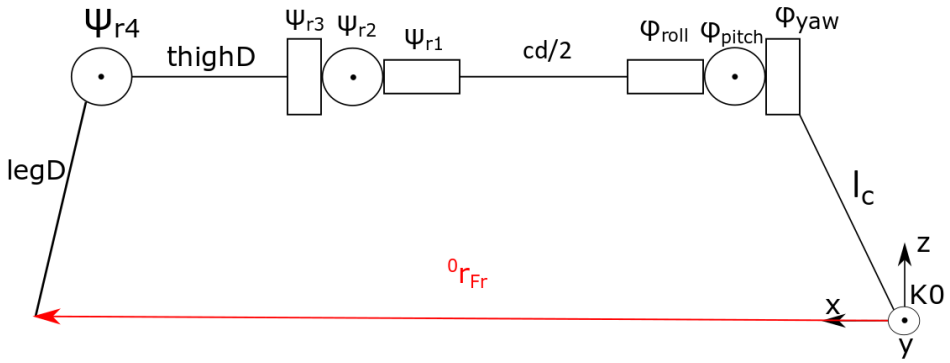

Figure 3-36: 2D diagram of the right leg tip distance from the inertia frame K0.

The homogeneous transformation matrices stated in Equation (3.12) can be expressed therefore as:

$$
{}^{0}R_4 = rot_z(yaw)\cdot rot_y(pitch)\cdot rot_x(roll) =
$$

$$
\begin{bmatrix} c_{yaw} & -s_{yaw} & 0 \\ s_{yaw} & c_{yaw} & 0 \\ 0 & 0 & 1 \end{bmatrix}\cdot\begin{bmatrix} c_{pitch} & 0 & s_{pitch} \\ 0 & 1 & 0 \\ -s_{pitch} & 0 & c_{pitch} \end{bmatrix}\cdot\begin{bmatrix} 1 & 0 & 0 \\ 0 & c_{roll} & -s_{roll} \\ 0 & s_{roll} & c_{roll} \end{bmatrix}\Rightarrow
$$

$$
{}^{0}R_4 = \begin{bmatrix} c_{yaw}c_{pitch} & -s_{yaw}c_{roll}+c_{yaw}s_{pitch}s_{roll} & s_{roll}s_{yaw}+c_{roll}s_{pitch}c_{yaw} \\ s_{yaw}c_{pitch} & c_{yaw}c_{roll}+s_{yaw}s_{pitch}s_{roll} & -s_{roll}c_{yaw}+c_{roll}s_{pitch}s_{yaw} \\ -s_{pitch} & c_{pitch}s_{roll} & c_{roll}c_{pitch} \end{bmatrix}\Rightarrow \tag{3.13}
$$

$$
{}^{0}T_4 = \begin{bmatrix} c_{yaw}c_{pitch} & -s_{yaw}c_{roll}+c_{yaw}s_{pitch}s_{roll} & s_{roll}s_{yaw}+c_{roll}s_{pitch}c_{yaw} & x_c \\ s_{yaw}c_{pitch} & c_{yaw}c_{roll}+s_{yaw}s_{pitch}s_{roll} & -s_{roll}c_{yaw}+c_{roll}s_{pitch}s_{yaw} & y_c \\ -s_{pitch} & c_{pitch}s_{roll} & c_{roll}c_{pitch} & z_c \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

$$
{}^{4}R_8 = rot_x(\psi_{r1})\cdot rot_y(\psi_{r2})\cdot rot_z(\psi_{r3}) =
$$

$$
\begin{bmatrix} 1 & 0 & 0 \\ 0 & c_{\psi_{r1}} & -s_{\psi_{r1}} \\ 0 & s_{\psi_{r1}} & c_{\psi_{r1}} \end{bmatrix}\cdot\begin{bmatrix} c_{\psi_{r2}} & 0 & s_{\psi_{r2}} \\ 0 & 1 & 0 \\ -s_{\psi_{r2}} & 0 & c_{\psi_{r2}} \end{bmatrix}\cdot\begin{bmatrix} c_{\psi_{r3}} & -s_{\psi_{r3}} & 0 \\ s_{\psi_{r3}} & c_{\psi_{r3}} & 0 \\ 0 & 0 & 1 \end{bmatrix}\Rightarrow
$$

$$
{}^{4}R_8 = \begin{bmatrix} c_{\psi_{r2}}c_{\psi_{r3}} & -c_{\psi_{r2}}s_{\psi_{r3}} & s_{\psi_{r2}} \\ s_{\psi_{r1}}s_{\psi_{r2}}c_{\psi_{r3}}+c_{\psi_{r1}}s_{\psi_{r3}} & -s_{\psi_{r1}}s_{\psi_{r2}}s_{\psi_{r3}}+c_{\psi_{r1}}c_{\psi_{r3}} & -s_{\psi_{r1}}c_{\psi_{r2}} \\ -s_{\psi_{r2}}c_{\psi_{r1}}c_{\psi_{r3}}+s_{\psi_{r1}}s_{\psi_{r3}} & s_{\psi_{r2}}s_{\psi_{r3}}c_{\psi_{r1}}+s_{\psi_{r1}}c_{\psi_{r3}} & c_{\psi_{r1}}c_{\psi_{r2}} \end{bmatrix}\Rightarrow \tag{3.14}
$$

$$
{}^{4}T_8 = \begin{bmatrix} c_{\psi_{r2}}c_{\psi_{r3}} & -c_{\psi_{r2}}s_{\psi_{r3}} & s_{\psi_{r2}} & cxd/2 \\ s_{\psi_{r1}}s_{\psi_{r2}}c_{\psi_{r3}}+c_{\psi_{r1}}s_{\psi_{r3}} & -s_{\psi_{r1}}s_{\psi_{r2}}s_{\psi_{r3}}+c_{\psi_{r1}}c_{\psi_{r3}} & -s_{\psi_{r1}}c_{\psi_{r2}} & -cyd/2 \\ -s_{\psi_{r2}}c_{\psi_{r1}}c_{\psi_{r3}}+s_{\psi_{r1}}s_{\psi_{r3}} & s_{\psi_{r2}}s_{\psi_{r3}}c_{\psi_{r1}}+s_{\psi_{r1}}c_{\psi_{r3}} & c_{\psi_{r1}}c_{\psi_{r2}} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

The final transformation matrix along with the distance vector from K_{10} to K_{Fr} with respect to K_{10} are:

$$
{}^{8}R_{10} = rot_y(\psi_{r4}) = \begin{bmatrix} c_{\psi_{r4}} & 0 & s_{\psi_{r4}} \\ 0 & 1 & 0 \\ -s_{\psi_{r4}} & 0 & c_{\psi_{r4}} \end{bmatrix}\Rightarrow
$$

$$
{}^{8}T_{10} = \begin{bmatrix} c_{\psi_{r4}} & 0 & s_{\psi_{r4}} & thighD \\ 0 & 1 & 0 & 0 \\ -s_{\psi_{r4}} & 0 & c_{\psi_{r4}} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.15}
$$

$$
{}^{10}_{10}r_{Fr} = \begin{bmatrix} legD \\ 0 \\ 0 \\ 1 \end{bmatrix}
$$

Then, Equation (3.12) becomes:

$$\begin{bmatrix} x_{Fr} \\ y_{Fr} \\ z_{Fr} \\ 1 \end{bmatrix} = \begin{bmatrix} c_{yaw}c_{pitch} & -s_{yaw}c_{roll} + c_{yaw}s_{pitch}s_{roll} & s_{roll}s_{yaw} + c_{roll}s_{pitch}c_{yaw} & x_c \\ s_{yaw}c_{pitch} & c_{yaw}c_{roll} + s_{yaw}s_{pitch}s_{roll} & -s_{roll}c_{yaw} + c_{roll}s_{pitch}s_{yaw} & y_c \\ -s_{pitch} & c_{pitch}s_{roll} & c_{roll}c_{pitch} & z_c \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$
$$\begin{bmatrix} c_{\psi_{r2}}c_{\psi_{r3}} & -c_{\psi_{r2}}s_{\psi_{r3}} & s_{\psi_{r2}} & cxd/2 \\ s_{\psi_{r1}}s_{\psi_{r2}}c_{\psi_{r3}} + c_{\psi_{r1}}s_{\psi_{r3}} & -s_{\psi_{r1}}s_{\psi_{r2}}s_{\psi_{r3}} + c_{\psi_{r1}}c_{\psi_{r3}} & -s_{\psi_{r1}}c_{\psi_{r2}} & -cyd/2 \\ -s_{\psi_{r2}}c_{\psi_{r1}}c_{\psi_{r3}} + s_{\psi_{r1}}s_{\psi_{r3}} & s_{\psi_{r2}}s_{\psi_{r3}}c_{\psi_{r1}} + s_{\psi_{r1}}c_{\psi_{r3}} & c_{\psi_{r1}}c_{\psi_{r2}} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (3.16)$$
$$\begin{bmatrix} c_{\psi_{r4}} & 0 & s_{\psi_{r4}} & thighD \\ 0 & 1 & 0 & 0 \\ -s_{\psi_{r4}} & 0 & c_{\psi_{r4}} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.\begin{bmatrix} legD \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

Therefore, the position of the right foreleg can be calculated from the next equations:

$$\begin{aligned} x_{Frcalculated} = x_c - &\left(c_{\varphi_{roll}} \cdot s_{\varphi_{yaw}} - c_{\varphi_{yaw}} \cdot s_{\varphi_{pitch}} \cdot s_{\varphi_{roll}}\right) \cdot \\ &\left(-\frac{cyd}{2} + \left(thighD + c_{\psi_{r4}} \cdot legD\right) \cdot \left(c_{\psi_{r1}} \cdot s_{\psi_{r3}} + c_{\psi_{r3}} \cdot s_{\psi_{r1}} \cdot s_{\psi_{r2}}\right) + \right. \\ &\left. + c_{\psi_{r2}} \cdot legD \cdot s_{\psi_{r1}} \cdot s_{\psi_{r4}}\right) + \\ &\left(\left(thighD + c_{\psi_{r4}} \cdot legD\right) \cdot \right. \\ &\left.\left(s_{\psi_{r1}} \cdot s_{\psi_{r3}} - c_{\psi_{r1}} \cdot c_{\psi_{r3}} \cdot s_{\psi_{r2}}\right) - c_{\psi_{r1}} \cdot c_{\psi_{r2}} \cdot legD \cdot s_{\psi_{r4}}\right) \cdot \left(s_{\varphi_{roll}} \cdot s_{\varphi_{yaw}} + c_{\varphi_{roll}} \cdot c_{\varphi_{yaw}} \cdot s_{\varphi_{pitch}}\right) + \\ &c_{\varphi_{pitch}} \cdot c_{\varphi_{yaw}} \cdot \left(\frac{cxd}{2} + c_{\psi_{r2}} \cdot c_{\psi_{r3}} \cdot \left(thighD + c_{\psi_{r4}} \cdot legD\right) - legD \cdot s_{\psi_{r2}} \cdot s_{\psi_{r4}}\right) \end{aligned} \quad (3.17)$$

$$\begin{aligned} y_{Frcalculated} = y_c + &\left(c_{\varphi_{roll}} \cdot c_{\varphi_{yaw}} + s_{\varphi_{yaw}} \cdot s_{\varphi_{pitch}} \cdot s_{\varphi_{roll}}\right) \cdot \left(\begin{array}{l} \left(thighD + c_{\psi_{r4}} \cdot legD\right) \cdot \\ \left(c_{\psi_{r1}} \cdot s_{\psi_{r3}} + c_{\psi_{r3}} \cdot s_{\psi_{r1}} \cdot s_{\psi_{r2}}\right) - \\ \frac{cyd}{2} + c_{\psi_{r2}} \cdot legD \cdot s_{\psi_{r1}} \cdot s_{\psi_{r4}} \end{array}\right) - \\ &\left(\begin{array}{l} \left(thighD + c_{\psi_{r4}} \cdot legD\right) \cdot \left(s_{\psi_{r1}} \cdot s_{\psi_{r3}} - c_{\psi_{r1}} \cdot c_{\psi_{r3}} \cdot s_{\psi_{r2}}\right) - \\ c_{\psi_{r1}} \cdot c_{\psi_{r2}} \cdot legD \cdot s_{\psi_{r4}} \end{array}\right) \cdot \left(s_{\varphi_{roll}} \cdot c_{\varphi_{yaw}} - c_{\varphi_{roll}} \cdot s_{\varphi_{yaw}} \cdot s_{\varphi_{pitch}}\right) + \\ &c_{\varphi_{pitch}} \cdot s_{\varphi_{yaw}} \cdot \left(\frac{cxd}{2} + c_{\psi_{r2}} \cdot c_{\psi_{r3}} \cdot \left(thighD + c_{\psi_{r4}} \cdot legD\right) - legD \cdot s_{\psi_{r2}} \cdot s_{\psi_{r4}}\right) \end{aligned} \quad (3.18)$$

$$\begin{aligned} z_{Frcalculated} = z_c - &s_{\varphi_{pitch}} \cdot \left(\begin{array}{l} \frac{cxd}{2} + c_{\psi_{r2}} \cdot c_{\psi_{r3}} \cdot \left(thighD + c_{\psi_{r4}} \cdot legD\right) - \\ legD \cdot s_{\psi_{r2}} \cdot s_{\psi_{r4}} \end{array}\right) + \\ &c_{\varphi_{pitch}} \cdot s_{\varphi_{roll}} \left(\begin{array}{l} \left(thighD + c_{\psi_{r4}} \cdot legD\right) \cdot \left(c_{\psi_{r1}} \cdot s_{\psi_{r3}} + c_{\psi_{r3}} \cdot s_{\psi_{r1}} \cdot s_{\psi_{r2}}\right) - \\ \frac{cyd}{2} + c_{\psi_{r2}} \cdot legD \cdot s_{\psi_{r1}} \cdot s_{\psi_{r4}} \end{array}\right) + \\ &c_{\varphi_{pitch}} \cdot c_{\varphi_{roll}} \cdot \left(\begin{array}{l} \left(thighD + c_{\psi_{r4}} \cdot legD\right) \cdot \left(s_{\psi_{r1}} \cdot s_{\psi_{r3}} - c_{\psi_{r1}} \cdot c_{\psi_{r3}} \cdot s_{\psi_{r2}}\right) - \\ c_{\psi_{r1}} \cdot c_{\psi_{r2}} \cdot legD \cdot s_{\psi_{r4}} \end{array}\right) \end{aligned} \quad (3.19)$$

Afterwards we proceed with the validation of the MOBILE iterative inverse kinematics values for the right leg, using the above equations, Equation (3.17), Equation (3.18) and Equation (3.19), with four verifications following the exact same procedure that was followed on the left leg.

1st Verification test

We input to MOBILE the next range of x_{Fr} values, [1.80 m, 4.30 m] with a step of 0.01 m. For every x_{Fr} value we get a value for each one of the dependent variables and store the results. While doing that, we keep the rest independent variables, except x_{Fr}, that affect the right foreleg's position at a fixed value. The values of the variables that affect the right foreleg's position can be seen in Table 3-6:

Table 3-6: The values of the variables affecting the right foreleg's position for the first verification.

x_{Fr} (m)	y_{Fr} (m)	z_{Fr} (m)	ψ_{r1} (deg)	x_C (m)	y_C (m)	z_C (m)	ϕ_{yaw} (deg)	ϕ_{pitch} (deg)	ϕ_{roll} (deg)
[1.80, 4.30]	-1.50	0	0°	0	0	2	0°	0°	0°

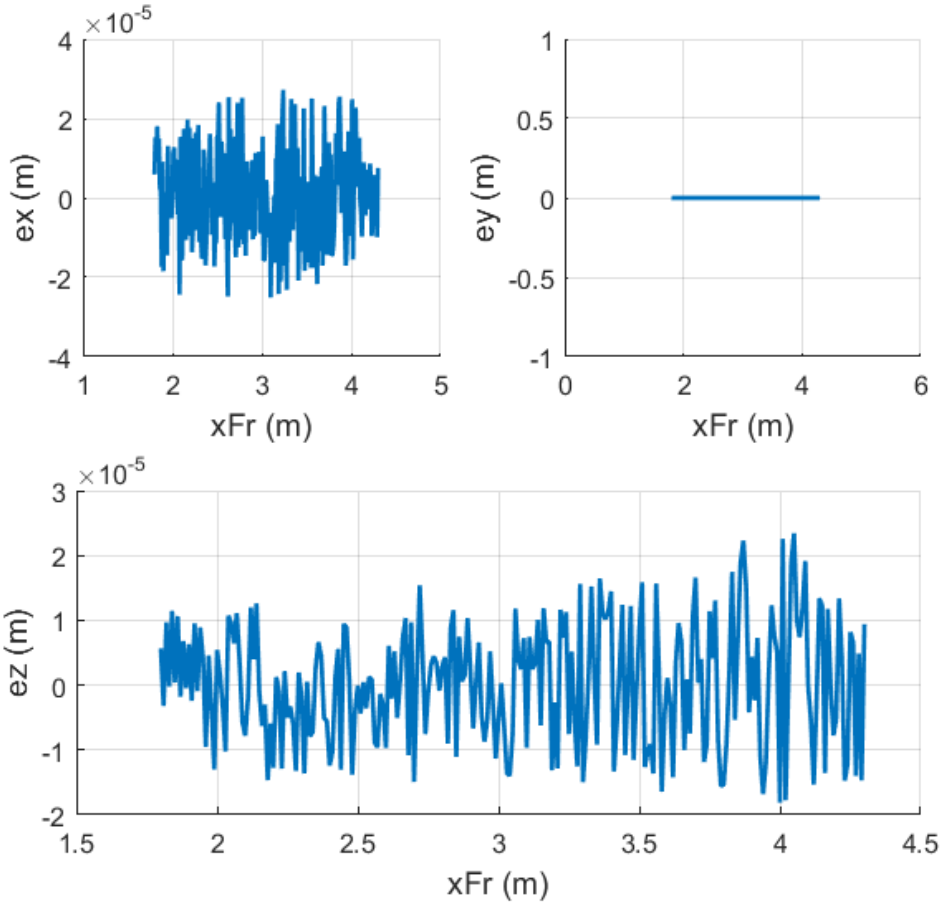

Figure 3-37: x_{Fr}, y_{Fr} and z_{Fr} errors calculated as (x_{Fr}-$x_{Frcalculated}$), (y_{Fr}-$y_{Frcalculated}$) and (z_{Fr}-$z_{Frcalculated}$) when the varying value is x_{Fr}.

We use the values of the dependent variables ψ_{r2}, ψ_{r3} and ψ_{r4} we got from MOBILE, and the rest that are on a fixed value to get $x_{Frcalculated}$ $y_{Frcalculated}$ and $z_{Frcalculated}$ and receive the above error diagrams in Figure 3-37, between the MATLAB analytically calculated position of the right foreleg's tip and the one inputted to MOBILE.

We observe that the errors are of the order of 10^{-5} m, which is acceptably small.

2nd Verification test

For the second verification test we input to MOBILE the next range of y_{Fr} values, [0.50 m, -3.50 m] with a step of 0.01 m. For every y_{Fr} value we get a value for each one of the dependent variables and store the results. While doing that, we keep the rest independent variables, except y_{Fr}, that affect the right foreleg's position at a fixed value. The values of the variables that affect the right foreleg's position can be seen in Table 3-7:

Table 3-7: The values of the variables affecting the right foreleg's position for the second verification.

x_{Fr} (m)	y_{Fr} (m)	z_{Fr} (m)	ψ_{r1} (deg)	x_C (m)	y_C (m)	z_C (m)	ϕ_{yaw} (deg)	ϕ_{pitch} (deg)	ϕ_{roll} (deg)
3.5	[0.50, -3.50]	0	0°	0	0	2	0°	0°	0°

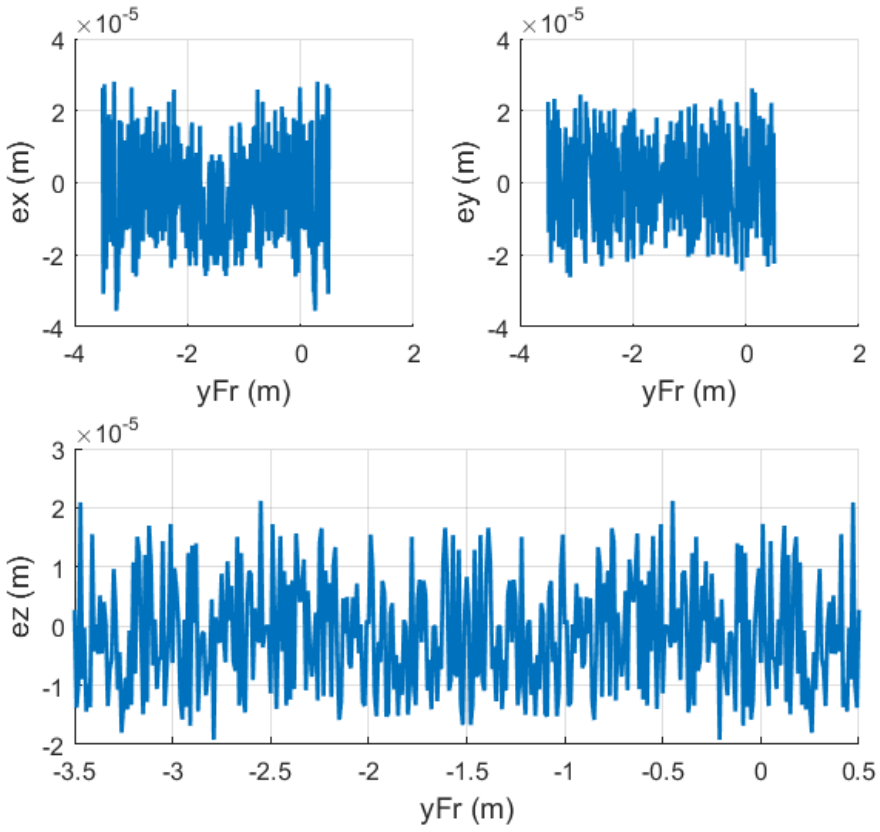

Figure 3-38: x_{Fr}, y_{Fr} and z_{Fr} errors calculated as (x_{Fr}-$x_{Frcalculated}$), (y_{Fr}-$y_{Frcalculated}$) and (z_{Fr}-$z_{Frcalculated}$) when the varying value is y_{Fr}.

In a similar way like before, we use the values of the dependent variables ψ_{r2}, ψ_{r3} and ψ_{r4} we got from MOBILE, and the rest that are on a fixed value to calculate analytically $x_{Frcalculated}$, $y_{Frcalculated}$ and $z_{Frcalculated}$ on MATLAB and get the above error diagrams in Figure 3-38.

We observe that the errors are of the order of 10^{-5} m, which is acceptably small.

3rd Verification test

For the third verification test we input to MOBILE the next range of z_{Fr} values, [-0.50 m, 3.50 m] with a step of 0.01 m. For every z_{Fr} value we get a value for each one of the dependent variables and store the results. While doing that, we keep the rest independent variables, except z_{Fr}, that affect the right foreleg's position at a fixed value. The values of the variables that affect the right foreleg's position can be seen in Table 3-8:

Table 3-8: The values of the variables affecting the right foreleg's position for the third verification.

x_{Fr} (m)	y_{Fr} (m)	z_{Fl} (m)	ψ_{r1} (deg)	x_C (m)	y_C (m)	z_C (m)	ϕ_{yaw} (deg)	ϕ_{pitch} (deg)	ϕ_{roll} (deg)
3.5	-1.50	[-0.50, 3.50]	0°	0	0	2	0°	0°	0°

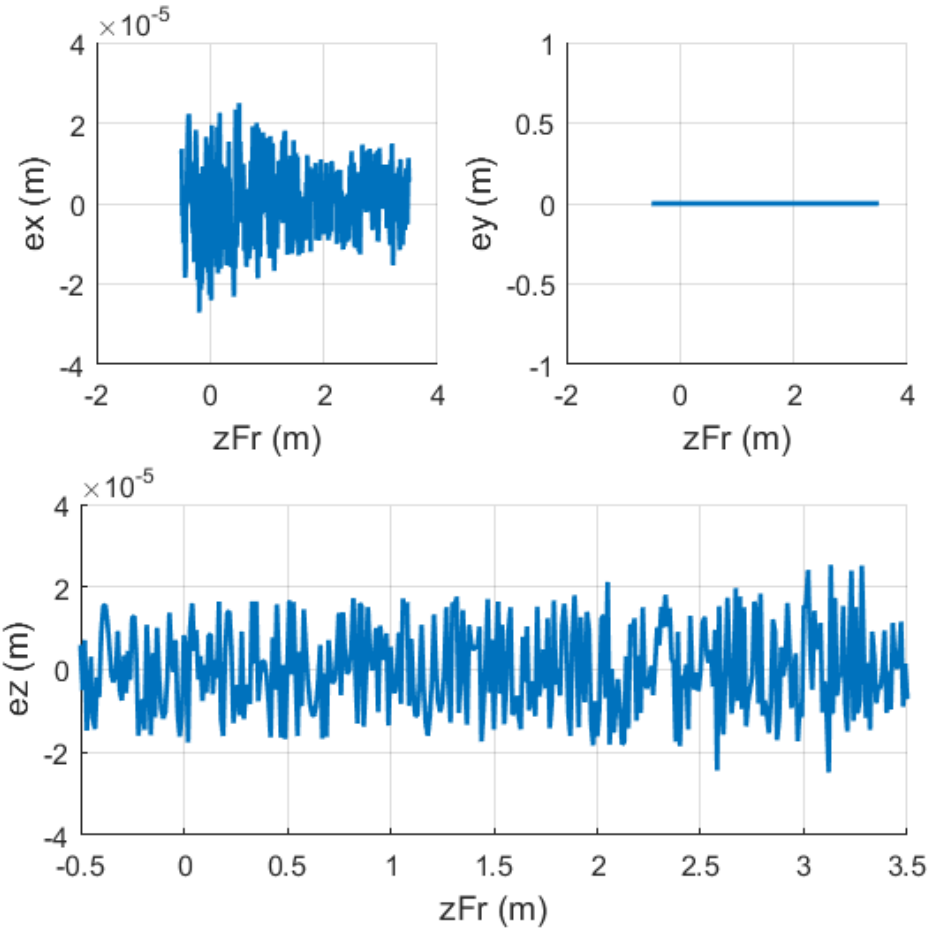

Figure 3-39: x_{Fr}, y_{Fr} and z_{Fr} errors calculated as (x_{Fr}-$x_{Frcalculated}$), (y_{Fr}-$y_{Frcalculated}$) and (z_{Fr}-$z_{Frcalculated}$) when the varying value is z_{Fr}.

We use the values of the dependent variables ψ_{r2}, ψ_{r3} and ψ_{r4} we got from MOBILE, and the rest that are on a fixed value to calculate analytically $x_{Frcalculated}$, $y_{Frcalculated}$ and $z_{Frcalculated}$ on MATLAB and get the above error diagrams for the third verification test in Figure 3-39:

We observe that the errors are of the order of 10^{-5} m, which is acceptably small.

4th Verification test

For the fourth verification test we input to MOBILE the next range of ψ_{r1} values, [-30º , 30º] with a step of 0.1º. For every ψ_{r1} value we get a value for each one of the dependent variables and store the results. While doing that, we keep the rest independent variables, except ψ_{r1}, that affect the right foreleg's position at a fixed value. The values of the variables that affect the right foreleg's position can be seen in Table 3-9:

Table 3-9: The values of the variables affecting the right foreleg's position for the fourth verification.

x_{Fr} (m)	y_{Fr} (m)	z_{Fr} (m)	ψ_{r1} (deg)	x_C (m)	y_C (m)	z_C (m)	ϕ_{yaw} (deg)	ϕ_{pitch} (deg)	ϕ_{roll} (deg)
3.5	-1.50	0	[-30º, 30º]	0	0	2	0º	0º	0º

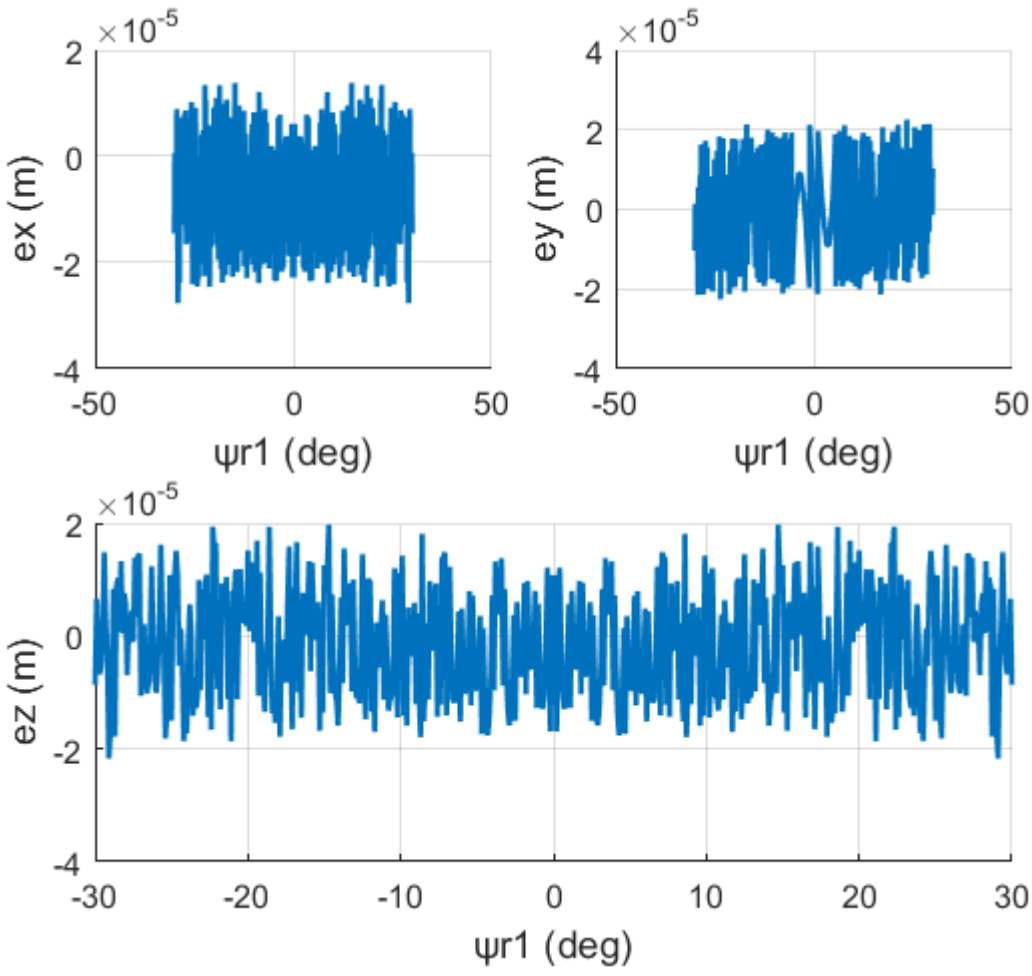

Figure 3-40: x_{Fr}, y_{Fr} and z_{Fr} errors calculated as (x_{Fr}-$x_{Frcalculated}$), (y_{Fr}-$y_{Frcalculated}$) and (z_{Fr}-$z_{Frcalculated}$) when the varying value is ψ_{r1}.

We use the values of the dependent variables ψ_{r2}, ψ_{r3} and ψ_{r4} we got from MOBILE, and the rest that are on a fixed value to calculate analytically $x_{Frcalculated}$, $y_{Frcalculated}$ and $z_{Frcalculated}$ on MATLAB and get the above error diagrams for the fourth verification test in Figure 3-40:

We observe that the errors are of the order of 10^{-5} m, which is acceptably small.

3.4.3 Analytical verification for the left wheel lever and wheel

Firstly, we construct the closed form equation that expresses the left wheel's tip position in respect with the ground-fixed frame K_0. This distance can be calculated from Equation (3.20) :

$$ {}^{0}_{0}r_{Wl} = {}^{0}T_{4} \cdot {}^{4}T_{7} \cdot {}^{7}T_{10} \cdot {}^{10}T_{12} \cdot {}^{12}_{12}r_{Wl} \tag{3.20} $$

In Figure 3-41, the left wheel's frame distribution can be seen, regarding Equation (3.20):

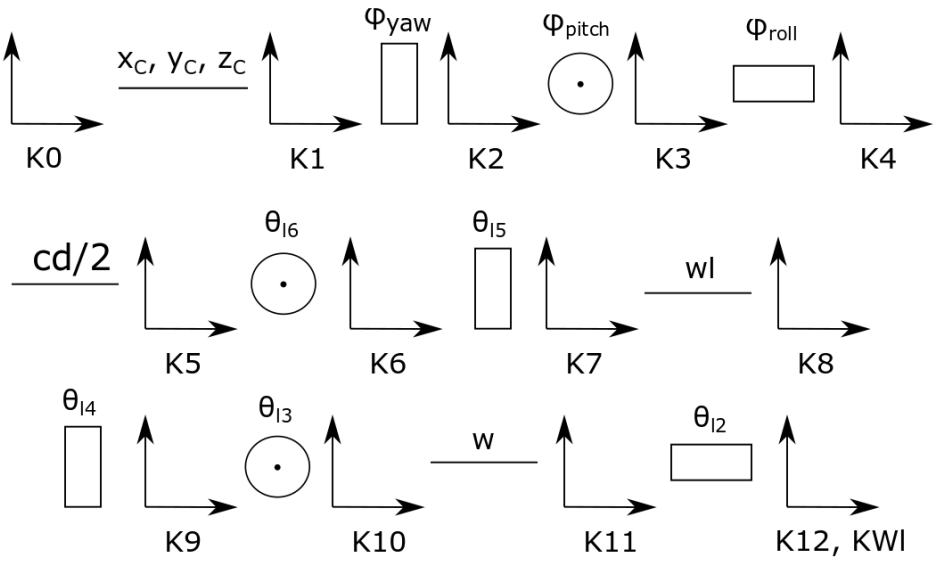

Figure 3-41: Frames distribution to the components regarding the rear left wheel end frame distance from the inertia frame K0.

In Figure 3-42 the 2D diagram of the left wheel lever and wheel can be seen with all the joints and links that construct them. The cabin's geometry is the same as described in the previous parts and can be reminded in Figure 3-27, in Figure 3-28 the robot's geometrical variables can be reminded in MOBILE environment.

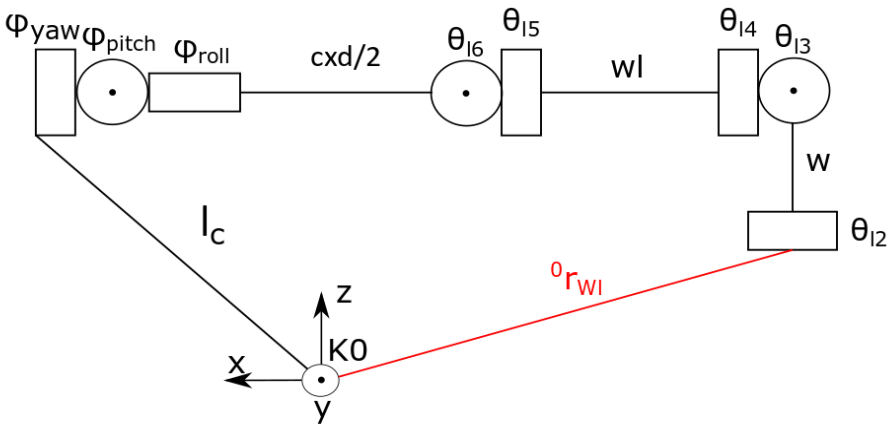

Figure 3-42: 2D diagram of the left rear wheel end frame distance from the inertia frame K0.

The homogeneous transformation matrices stated in Equation (3.20) can be expressed therefore as:

$$
{}^{0}R_4 = rot_z(yaw)\cdot rot_y(pitch)\cdot rot_x(roll) =
$$

$$
\begin{bmatrix} c_{yaw} & -s_{yaw} & 0 \\ s_{yaw} & c_{yaw} & 0 \\ 0 & 0 & 1 \end{bmatrix}\cdot\begin{bmatrix} c_{pitch} & 0 & s_{pitch} \\ 0 & 1 & 0 \\ -s_{pitch} & 0 & c_{pitch} \end{bmatrix}\cdot\begin{bmatrix} 1 & 0 & 0 \\ 0 & c_{roll} & -s_{roll} \\ 0 & s_{roll} & c_{roll} \end{bmatrix}\Rightarrow
$$

$$
{}^{0}R_4 = \begin{bmatrix} c_{yaw}c_{pitch} & -s_{yaw}c_{roll}+c_{yaw}s_{pitch}s_{roll} & s_{roll}s_{yaw}+c_{roll}s_{pitch}c_{yaw} \\ s_{yaw}c_{pitch} & c_{yaw}c_{roll}+s_{yaw}s_{pitch}s_{roll} & -s_{roll}c_{yaw}+c_{roll}s_{pitch}s_{yaw} \\ -s_{pitch} & c_{pitch}s_{roll} & c_{roll}c_{pitch} \end{bmatrix}\Rightarrow \tag{3.21}
$$

$$
{}^{0}T_4 = \begin{bmatrix} c_{yaw}c_{pitch} & -s_{yaw}c_{roll}+c_{yaw}s_{pitch}s_{roll} & s_{roll}s_{yaw}+c_{roll}s_{pitch}c_{yaw} & x_c \\ s_{yaw}c_{pitch} & c_{yaw}c_{roll}+s_{yaw}s_{pitch}s_{roll} & -s_{roll}c_{yaw}+c_{roll}s_{pitch}s_{yaw} & y_c \\ -s_{pitch} & c_{pitch}s_{roll} & c_{roll}c_{pitch} & z_c \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

$$
{}^{4}R_7 = rot_y(\theta_{l6})\cdot rot_z(\theta_{l5}) = \begin{bmatrix} c_{\theta_{l6}} & 0 & s_{\theta_{l6}} \\ 0 & 1 & 0 \\ -s_{\theta_{l6}} & 0 & c_{\theta_{l6}} \end{bmatrix}\cdot\begin{bmatrix} c_{\theta_{l5}} & -s_{\theta_{l5}} & 0 \\ s_{\theta_{l5}} & c_{\theta_{l5}} & 0 \\ 0 & 0 & 1 \end{bmatrix}\Rightarrow
$$

$$
{}^{4}R_7 = \begin{bmatrix} c_{\theta_{l5}}c_{\theta_{l6}} & -c_{\theta_{l6}}s_{\theta_{l5}} & s_{\theta_{l6}} \\ s_{\theta_{l5}} & c_{\theta_{l5}} & 0 \\ -c_{\theta_{l5}}s_{\theta_{l6}} & s_{\theta_{l5}}s_{\theta_{l6}} & c_{\theta_{l6}} \end{bmatrix}\Rightarrow \tag{3.22}
$$

$$
{}^{4}T_7 = \begin{bmatrix} c_{\theta_{l5}}c_{\theta_{l6}} & -c_{\theta_{l6}}s_{\theta_{l5}} & s_{\theta_{l6}} & -cxd/2 \\ s_{\theta_{l5}} & c_{\theta_{l5}} & 0 & cyd/2 \\ -c_{\theta_{l5}}s_{\theta_{l6}} & s_{\theta_{l5}}s_{\theta_{l6}} & c_{\theta_{l6}} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

$$
{}^{7}R_{10} = rot_z(\theta_{l4}) \cdot rot_y(\theta_{l3}) = \begin{bmatrix} c_{\theta_{l4}} & -s_{\theta_{l4}} & 0 \\ s_{\theta_{l4}} & c_{\theta_{l4}} & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} c_{\theta_{l3}} & 0 & s_{\theta_{l3}} \\ 0 & 1 & 0 \\ -s_{\theta_{l3}} & 0 & c_{\theta_{l3}} \end{bmatrix} \Rightarrow
$$

$$
{}^{7}R_{10} = \begin{bmatrix} c_{\theta_{l3}} c_{\theta_{l4}} & -s_{\theta_{l4}} & c_{\theta_{l4}} s_{\theta_{l3}} \\ c_{\theta_{l3}} s_{\theta_{l4}} & c_{\theta_{l4}} & s_{\theta_{l3}} s_{\theta_{l4}} \\ -s_{\theta_{l3}} & 0 & c_{\theta_{l3}} \end{bmatrix} \Rightarrow \tag{3.23}
$$

$$
{}^{7}T_{10} = \begin{bmatrix} c_{\theta_{l3}} c_{\theta_{l4}} & -s_{\theta_{l4}} & c_{\theta_{l4}} s_{\theta_{l3}} & -wl \\ c_{\theta_{l3}} s_{\theta_{l4}} & c_{\theta_{l4}} & s_{\theta_{l3}} s_{\theta_{l4}} & 0 \\ -s_{\theta_{l3}} & 0 & c_{\theta_{l3}} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

The final transformation matrix along with the distance vector from K_{12} to K_{Wl} with respect to K_{12} are:

$$
{}^{10}R_{12} = rot_x\left(\theta_{l2}\right) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c_{\theta_{l2}} & -s_{\theta_{l2}} \\ 0 & s_{\theta_{l2}} & c_{\theta_{l2}} \end{bmatrix}
$$

$$
{}^{10}T_{12} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c_{\theta_{l2}} & -s_{\theta_{l2}} & 0 \\ 0 & s_{\theta_{l2}} & c_{\theta_{l2}} & -w \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.24}
$$

$$
{}^{12}_{12}r_{Wl} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}
$$

Combining the above expressions, we get the following system of equations:

$$
\begin{bmatrix} x_{Wl} \\ y_{Wl} \\ z_{Wl} \\ 1 \end{bmatrix} = \begin{bmatrix} c_{yaw}c_{pitch} & -s_{yaw}c_{roll} + c_{yaw}s_{pitch}s_{roll} & s_{roll}s_{yaw} + c_{roll}s_{pitch}c_{yaw} & x_c \\ s_{yaw}c_{pitch} & c_{yaw}c_{roll} + s_{yaw}s_{pitch}s_{roll} & -s_{roll}c_{yaw} + c_{roll}s_{pitch}s_{yaw} & y_c \\ -s_{pitch} & c_{pitch}s_{roll} & c_{roll}c_{pitch} & z_c \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot
$$

$$
\begin{bmatrix} c_{\theta_{l5}} c_{\theta_{l6}} & -c_{\theta_{l6}} s_{\theta_{l5}} & s_{\theta_{l6}} & -cxd/2 \\ s_{\theta_{l5}} & c_{\theta_{l5}} & 0 & cyd/2 \\ -c_{\theta_{l5}} s_{\theta_{l6}} & s_{\theta_{l5}} s_{\theta_{l6}} & c_{\theta_{l6}} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \tag{3.25}
$$

$$
\begin{bmatrix} c_{\theta_{l3}} c_{\theta_{l4}} & -s_{\theta_{l4}} & c_{\theta_{l4}} s_{\theta_{l3}} & -wl \\ c_{\theta_{l3}} s_{\theta_{l4}} & c_{\theta_{l4}} & s_{\theta_{l3}} s_{\theta_{l4}} & 0 \\ -s_{\theta_{l3}} & 0 & c_{\theta_{l3}} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c_{\theta_{l2}} & -s_{\theta_{l2}} & 0 \\ 0 & s_{\theta_{l2}} & c_{\theta_{l2}} & -w \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}
$$

Therefore, the three equations expressing the left wheel's tip position with respect to K_0 are:

$$
\begin{aligned}
x_{Wl} = x_C - \frac{\left(cyd\left(c_{roll}s_{yaw} - c_{yaw}s_{pitch}s_{roll}\right)\right)}{2} \\
+w\begin{pmatrix}
c_{\theta_{l4}}s_{\theta_{l3}}\begin{pmatrix}
s_{\theta_{l5}}\left(c_{roll}s_{yaw} - c_{yaw}s_{pitch}s_{roll}\right) + \\
c_{\theta_{l5}}s_{\theta_{l6}}\left(s_{yaw}s_{roll} + c_{roll}c_{yaw}s_{pitch}\right) - c_{pitch}c_{\theta_{l5}}c_{\theta_{l6}}c_{yaw}
\end{pmatrix} - \\
c_{\theta_{l3}}\left(c_{\theta_{l6}}\left(s_{roll}s_{yaw} + c_{roll}c_{yaw}s_{pitch}\right) + c_{yaw}c_{pitch}s_{\theta_{l6}}\right) + \\
s_{\theta_{l3}}s_{\theta_{l4}}\begin{pmatrix}
c_{\theta_{l5}}\left(c_{roll}s_{yaw} - c_{yaw}s_{pitch}s_{roll}\right) - \\
s_{\theta_{l5}}s_{\theta_{l6}}\left(s_{roll}s_{yaw} + c_{roll}c_{yaw}s_{pitch}\right) + c_{pitch}c_{\theta_{l6}}c_{yaw}s_{\theta_{l5}}
\end{pmatrix}
\end{pmatrix} \\
+wl\begin{pmatrix}
s_{\theta_{l5}}\left(c_{roll}s_{yaw} - c_{yaw}s_{pitch}s_{roll}\right) + \\
c_{\theta_{l5}}s_{\theta_{l6}}\left(s_{roll}s_{yaw} + c_{roll}c_{yaw}s_{pitch}\right) - c_{pitch}c_{\theta_{l5}}c_{\theta_{l6}}c_{yaw}
\end{pmatrix} - \frac{c_{pitch}c_{yaw}cxd}{2}
\end{aligned}
\tag{3.26}
$$

$$
\begin{aligned}
y_{Wl} = y_C + \frac{\left(cyd\left(c_{roll}c_{yaw} + s_{yaw}s_{pitch}s_{roll}\right)\right)}{2} \\
+w\begin{pmatrix}
c_{\theta_{l3}}\begin{pmatrix}
c_{\theta_{l6}}\left(s_{roll}c_{yaw} - s_{yaw}s_{pitch}c_{roll}\right) - \\
c_{pitch}s_{\theta_{l6}}s_{yaw}
\end{pmatrix} - \\
c_{\theta_{l4}}s_{\theta_{l3}}\begin{pmatrix}
s_{\theta_{l5}}\left(c_{roll}c_{yaw} + s_{roll}s_{yaw}s_{pitch}\right) + s_{\theta_{l6}}c_{\theta_{l5}}\begin{pmatrix}
c_{yaw}s_{roll} - \\
c_{roll}s_{yaw}s_{pitch}
\end{pmatrix} + \\
c_{pitch}c_{\theta_{l5}}c_{\theta_{l6}}s_{yaw}
\end{pmatrix} + \\
s_{\theta_{l3}}s_{\theta_{l4}}\begin{pmatrix}
s_{\theta_{l5}}s_{\theta_{l6}}\left(s_{roll}c_{yaw} - s_{yaw}s_{pitch}c_{roll}\right) - \\
c_{\theta_{l5}}\left(c_{roll}c_{yaw} + s_{roll}s_{yaw}s_{pitch}\right) + c_{pitch}c_{\theta_{l6}}s_{yaw}s_{\theta_{l5}}
\end{pmatrix}
\end{pmatrix} \\
-wl\begin{pmatrix}
s_{\theta_{l5}}\left(c_{roll}c_{yaw} + s_{yaw}s_{pitch}s_{roll}\right) + \\
c_{\theta_{l5}}s_{\theta_{l6}}\left(s_{roll}c_{yaw} - c_{roll}s_{yaw}s_{pitch}\right) + c_{pitch}c_{\theta_{l5}}c_{\theta_{l6}}s_{yaw}
\end{pmatrix} - \frac{c_{pitch}s_{yaw}cxd}{2}
\end{aligned}
\tag{3.27}
$$

$$
\begin{aligned}
z_{Wl} = z_C + \frac{\left(s_{pitch}cxd\right)}{2} + wl\begin{pmatrix}
c_{\theta_{l5}}c_{\theta_{l6}}s_{pitch} - \\
c_{pitch}s_{roll}s_{\theta_{l5}} + \\
c_{pitch}c_{roll}s_{\theta_{l6}}c_{\theta_{l5}}
\end{pmatrix} + \\
w\begin{pmatrix}
c_{\theta_{l3}}\left(s_{pitch}s_{\theta_{l6}} - c_{pitch}c_{roll}c_{\theta_{l6}}\right) + \\
c_{\theta_{l4}}s_{\theta_{l3}}\left(c_{\theta_{l5}}c_{\theta_{l6}}s_{pitch} - c_{pitch}s_{roll}s_{\theta_{l5}} + c_{pitch}c_{roll}c_{\theta_{l5}}s_{\theta_{l6}}\right) - \\
s_{\theta_{l3}}s_{\theta_{l4}}\left(c_{pitch}c_{\theta_{l5}}s_{roll} + c_{\theta_{l6}}s_{pitch}s_{\theta_{l5}} + c_{pitch}c_{roll}s_{\theta_{l5}}s_{\theta_{l6}}\right)
\end{pmatrix} \\
+\frac{c_{pitch}s_{roll}cyd}{2}
\end{aligned}
\tag{3.28}
$$

Following a similar procedure as with the forelegs, the next step is to validate the MOBILE iterative inverse kinematics values with the analytic equations, Equation (3.26), Equation (3.27) and Equation (3.28). We do that by inputting the independent variables of the left wheel lever and wheel system (z_{Wl} and θ_{l5}) on MOBILE in a width of values and then store the according output values of the dependent variables (θ_{l6}, θ_{l3} and θ_{l2}). Also, we measure the x_{Wl} and y_{Wl} values for these widths, using MOBILE's measurements capabilities and we

store them too. After that, we input the dependent values (θ_{l6}, θ_{l3} and θ_{l2}) we stored in the above analytic equations on MATLAB, which gives us the calculated values of the rear left wheel tip position. Finally, we compare these calculated values to the z_{Wl} values that we inputted on MOBILE and x_{Wl}, y_{Wl} that we measured from MOBILE and observe their divergence.

1st Verification test

For the first verification we input to MOBILE the next range of z_{Wl} values, [0.90m, -0.70m] with a step of 0.01 m. For every z_{Wl} value we get a value for each one of the dependent variables and store the results. While doing that, we keep the other independent variable, θ_{l5} at a fixed value. The values of the variables that affect the rear left wheel's tip position can be seen in Table 3-10:

Table 3-10: The values of the variables affecting the rear left wheel's position for the first verification.

z_{Wl} (m)	θ_{l5} (deg)	x_C (m)	y_C (m)	z_C (m)	ϕ_{yaw} (deg)	ϕ_{pitch} (deg)	ϕ_{roll} (deg)
[0.90, -0.70]	0°	0	0	2	0°	0°	0°

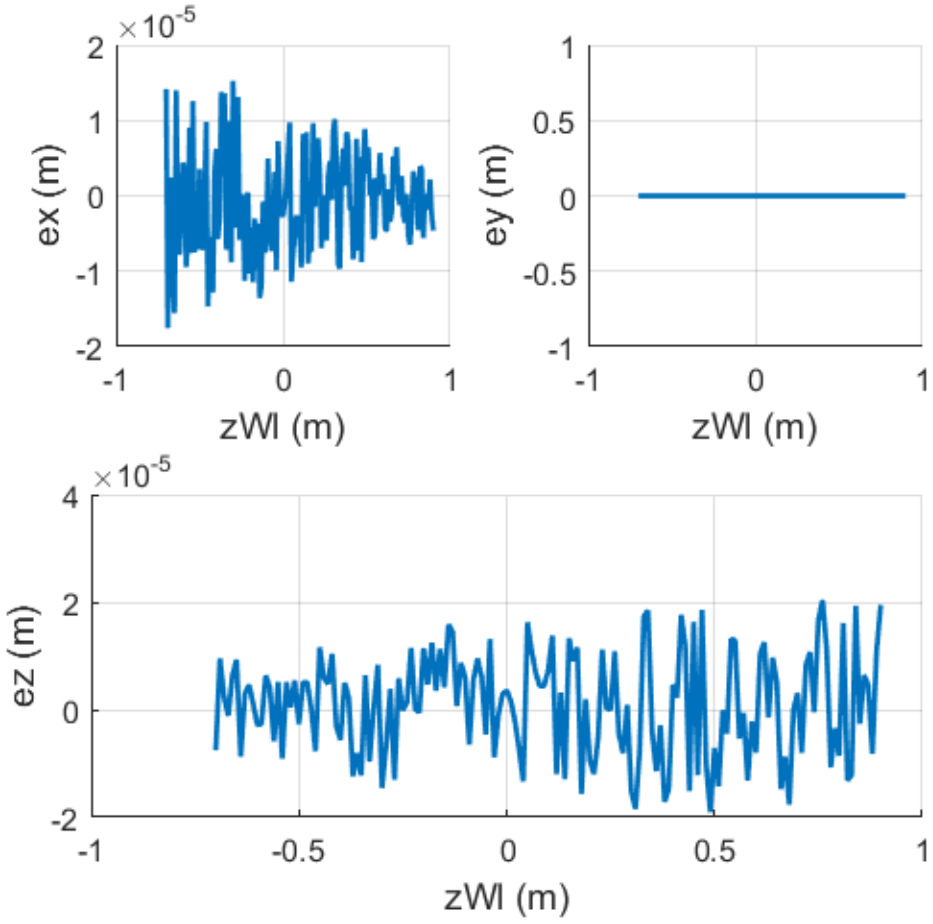

Figure 3-43: x_{Wl}, y_{Wl} and z_{Wl} errors calculated as (x_{Wl}-$x_{Wlcalculated}$), (y_{Wl}-$y_{Wlcalculated}$) and (z_{Wl}-$z_{Wlcalculated}$) when the varying value is z_{Wl}.

We use the values of the dependent variables θ_{l6}, θ_{l3}, θ_{l2}, x_{Wl} and y_{Wl} we get from MOBILE, and the rest that are on a fixed value to calculate $x_{Wlcalculated}$, $y_{Wlcalculated}$ and $z_{Wlcalculated}$ on MATLAB. Figure 3-43 represents the error diagrams between the calculated and desired position for the first verification test of the left wheel system.

We observe that the errors are of the order of 10^{-5} m, which is acceptably small.

2nd Verification test

For the second verification we input to MOBILE the next range of θ_{l5} values, [-40.0º, 15.0º] with a step of 0.1º. For every θ_{l5} value we get a value for each one of the dependent variables and store the results. While doing that, we keep the other independent variable, z_{Wl} at a fixed value. The values of the variables that affect the rear left wheel's tip position can be seen in Table 3-11:

Table 3-11: The values of the variables affecting the rear left wheel's position for the second verification.

z_{Wl} (m)	θ_{l5} (deg)	x_C (m)	y_C (m)	z_C (m)	ϕ_{yaw} (deg)	ϕ_{pitch} (deg)	ϕ_{roll} (deg)
0	[-40.0º, 15.0º]	0	0	2	0º	0º	0º

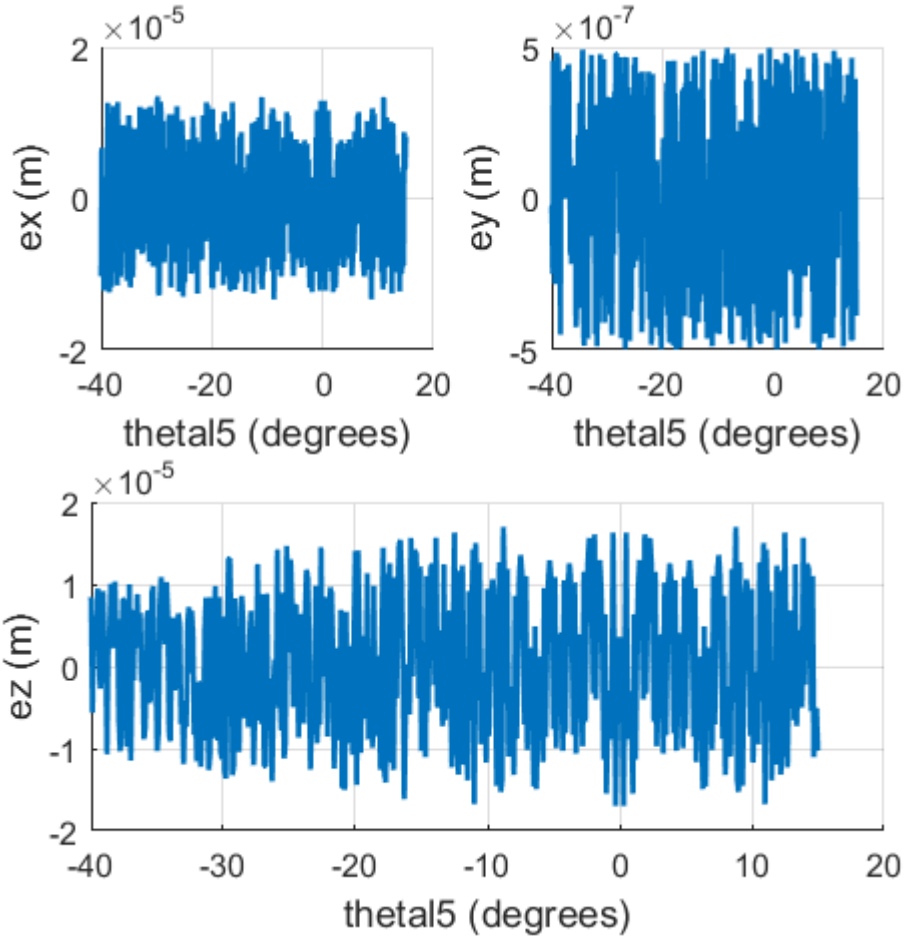

Figure 3-44: x_{Wl}, y_{Wl} and z_{Wl} errors calculated as (x_{Wl}-$x_{Wlcalculated}$), (y_{Wl}-$y_{Wlcalculated}$) and (z_{Wl}-$z_{Wlcalculated}$) when the varying value is θ_{l5}.

Similar to the previous verification, we use the values of the dependent variables θ_{l6}, θ_{l3}, θ_{l2}, x_{Wl} and y_{Wl} we get from MOBILE, and the rest that are on a fixed value to calculate $x_{Wlcalculated}$, $y_{Wlcalculated}$ and $z_{Wlcalculated}$ on MATLAB. Figure 3-44 represents the error diagrams between the calculated and desired position for the second verification simulation of the left wheel system.

We observe that the errors are of the order of 10^{-5} m or even smaller, which is acceptably small.

3.4.4 Analytical verification for the right wheel lever and wheel

Firstly, we construct the closed form equation that expresses the right wheel's tip position in respect with the ground-fixed frame K_0. That distance can be calculated from Equation (3.29) :

$$ {}^{0}_{0}r_{Wr} = {}^{0}T_{4} \cdot {}^{4}T_{7} \cdot {}^{7}T_{10} \cdot {}^{10}T_{12} \cdot {}^{12}_{12}r_{Wr} \tag{3.29} $$

In Figure 3-45 the right wheel's frame distribution can be seen, regarding Equation (3.29).

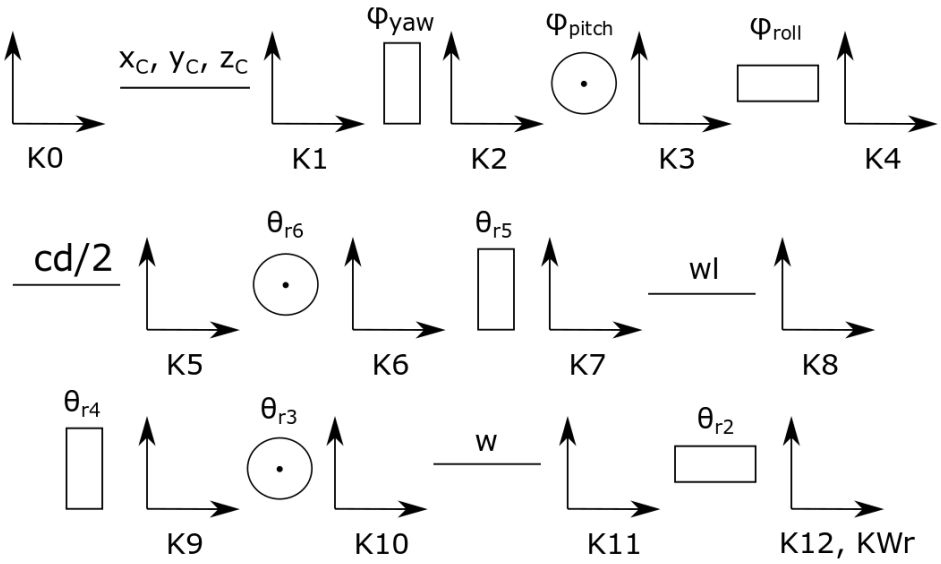

Figure 3-45: Frames distribution to the components regarding the rear right wheel end frame distance from the inertia frame K0.

In Figure 3-46 the 2D diagram of the right wheel lever and wheel can be seen with all the joints and links that construct them. The cabin's geometry is the same as before and can be reminded in Figure 3-27, in Figure 3-28 the robot's geometrical variables can be reminded in MOBILE environment.

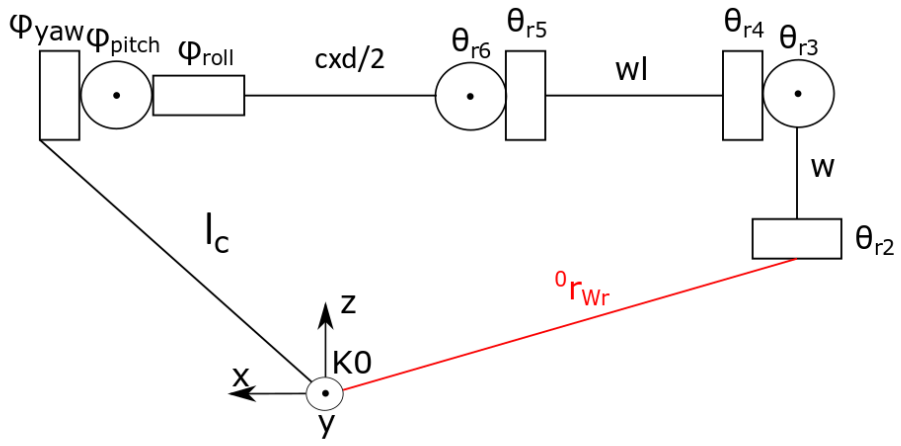

Figure 3-46: 2D diagram of the right rear wheel end frame distance from the inertia frame K0.

The homogeneous transformation matrices stated in Equation (3.29) can be expressed therefore as:

$$
{}^{0}R_4 = rot_z(yaw)\cdot rot_y(pitch)\cdot rot_x(roll) =
$$

$$
\begin{bmatrix} c_{yaw} & -s_{yaw} & 0 \\ s_{yaw} & c_{yaw} & 0 \\ 0 & 0 & 1 \end{bmatrix}\cdot\begin{bmatrix} c_{pitch} & 0 & s_{pitch} \\ 0 & 1 & 0 \\ -s_{pitch} & 0 & c_{pitch} \end{bmatrix}\cdot\begin{bmatrix} 1 & 0 & 0 \\ 0 & c_{roll} & -s_{roll} \\ 0 & s_{roll} & c_{roll} \end{bmatrix}\Rightarrow
$$

$$
{}^{0}R_4 = \begin{bmatrix} c_{yaw}c_{pitch} & -s_{yaw}c_{roll}+c_{yaw}s_{pitch}s_{roll} & s_{roll}s_{yaw}+c_{roll}s_{pitch}c_{yaw} \\ s_{yaw}c_{pitch} & c_{yaw}c_{roll}+s_{yaw}s_{pitch}s_{roll} & -s_{roll}c_{yaw}+c_{roll}s_{pitch}s_{yaw} \\ -s_{pitch} & c_{pitch}s_{roll} & c_{roll}c_{pitch} \end{bmatrix}\Rightarrow \tag{3.30}
$$

$$
{}^{0}T_4 = \begin{bmatrix} c_{yaw}c_{pitch} & -s_{yaw}c_{roll}+c_{yaw}s_{pitch}s_{roll} & s_{roll}s_{yaw}+c_{roll}s_{pitch}c_{yaw} & x_c \\ s_{yaw}c_{pitch} & c_{yaw}c_{roll}+s_{yaw}s_{pitch}s_{roll} & -s_{roll}c_{yaw}+c_{roll}s_{pitch}s_{yaw} & y_c \\ -s_{pitch} & c_{pitch}s_{roll} & c_{roll}c_{pitch} & z_c \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

$$
{}^{4}R_7 = rot_y(\theta_{r6})\cdot rot_z(\theta_{r5}) = \begin{bmatrix} c_{\theta_{r6}} & 0 & s_{\theta_{r6}} \\ 0 & 1 & 0 \\ -s_{\theta_{r6}} & 0 & c_{\theta_{r6}} \end{bmatrix}\cdot\begin{bmatrix} c_{\theta_{r5}} & -s_{\theta_{r5}} & 0 \\ s_{\theta_{r5}} & c_{\theta_{r5}} & 0 \\ 0 & 0 & 1 \end{bmatrix}\Rightarrow
$$

$$
{}^{4}R_7 = \begin{bmatrix} c_{\theta_{r5}}c_{\theta_{r6}} & -c_{\theta_{r6}}s_{\theta_{r5}} & s_{\theta_{r6}} \\ s_{\theta_{r5}} & c_{\theta_{r5}} & 0 \\ -c_{\theta_{r5}}s_{\theta_{r6}} & s_{\theta_{r5}}s_{\theta_{r6}} & c_{\theta_{r6}} \end{bmatrix}\Rightarrow \tag{3.31}
$$

$$
{}^{4}T_7 = \begin{bmatrix} c_{\theta_{r5}}c_{\theta_{r6}} & -c_{\theta_{r6}}s_{\theta_{r5}} & s_{\theta_{r6}} & -cxd/2 \\ s_{\theta_{r5}} & c_{\theta_{r5}} & 0 & -cyd/2 \\ -c_{\theta_{r5}}s_{\theta_{r6}} & s_{\theta_{r5}}s_{\theta_{r6}} & c_{\theta_{r6}} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

$$
{}^{7}R_{10} = rot_z(\theta_{r4}) \cdot rot_y(\theta_{r3}) = \begin{bmatrix} c_{\theta_{r4}} & -s_{\theta_{r4}} & 0 \\ s_{\theta_{r4}} & c_{\theta_{r4}} & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} c_{\theta_{r3}} & 0 & s_{\theta_{r3}} \\ 0 & 1 & 0 \\ -s_{\theta_{r3}} & 0 & c_{\theta_{r3}} \end{bmatrix} \Rightarrow
$$

$$
{}^{7}R_{10} = \begin{bmatrix} c_{\theta_{r3}} c_{\theta_{r4}} & -s_{\theta_{r4}} & c_{\theta_{r4}} s_{\theta_{r3}} \\ c_{\theta_{r3}} s_{\theta_{r4}} & c_{\theta_{r4}} & s_{\theta_{r3}} s_{\theta_{r4}} \\ -s_{\theta_{r3}} & 0 & c_{\theta_{r3}} \end{bmatrix} \Rightarrow \tag{3.32}
$$

$$
{}^{7}T_{10} = \begin{bmatrix} c_{\theta_{r3}} c_{\theta_{r4}} & -s_{\theta_{r4}} & c_{\theta_{r4}} s_{\theta_{r3}} & -wl \\ c_{\theta_{r3}} s_{\theta_{r4}} & c_{\theta_{r4}} & s_{\theta_{r3}} s_{\theta_{r4}} & 0 \\ -s_{\theta_{r3}} & 0 & c_{\theta_{r3}} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

The final transformation matrix along with the distance vector from K_{12} to K_{Wr} with respect to K_{12} are:

$$
{}^{10}R_{12} = rot_x(\theta_{r2}) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c_{\theta_{r2}} & -s_{\theta_{r2}} \\ 0 & s_{\theta_{r2}} & c_{\theta_{r2}} \end{bmatrix}
$$

$$
{}^{10}T_{12} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c_{\theta_{r2}} & -s_{\theta_{r2}} & 0 \\ 0 & s_{\theta_{r2}} & c_{\theta_{r2}} & -w \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.33}
$$

$$
{}^{12}_{12}r_{Wr} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}
$$

Combining the above expressions, we get the following system of equations:

$$
\begin{bmatrix} x_{Wr} \\ y_{Wr} \\ z_{Wr} \\ 1 \end{bmatrix} = \begin{bmatrix} c_{yaw}c_{pitch} & -s_{yaw}c_{roll} + c_{yaw}s_{pitch}s_{roll} & s_{roll}s_{yaw} + c_{roll}s_{pitch}c_{yaw} & x_c \\ s_{yaw}c_{pitch} & c_{yaw}c_{roll} + s_{yaw}s_{pitch}s_{roll} & -s_{roll}c_{yaw} + c_{roll}s_{pitch}s_{yaw} & y_c \\ -s_{pitch} & c_{pitch}s_{roll} & c_{roll}c_{pitch} & z_c \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot
$$

$$
\begin{bmatrix} c_{\theta_{r5}} c_{\theta_{r6}} & -c_{\theta_{r6}} s_{\theta_{r5}} & s_{\theta_{r6}} & -cxd/2 \\ s_{\theta_{r5}} & c_{\theta_{r5}} & 0 & -cyd/2 \\ -c_{\theta_{r5}} s_{\theta_{r6}} & s_{\theta_{r5}} s_{\theta_{r6}} & c_{\theta_{r6}} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \tag{3.34}
$$

$$
\begin{bmatrix} c_{\theta_{r3}} c_{\theta_{r4}} & -s_{\theta_{r4}} & c_{\theta_{r4}} s_{\theta_{r3}} & -wl \\ c_{\theta_{r3}} s_{\theta_{r4}} & c_{\theta_{r4}} & s_{\theta_{r3}} s_{\theta_{r4}} & 0 \\ -s_{\theta_{r3}} & 0 & c_{\theta_{r3}} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c_{\theta_{r2}} & -s_{\theta_{r2}} & 0 \\ 0 & s_{\theta_{r2}} & c_{\theta_{r2}} & -w \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}
$$

Therefore, the three equations expressing the right wheel's tip position with respect to K_0 are:

$$x_{Wr} = x_C + \left(c_{roll}s_{yaw} - c_{yaw}s_{pitch}s_{roll}\right)\left(\begin{matrix}\frac{cyd}{2} + s_{\theta_{r5}}\left(wl + c_{\theta_{r4}}s_{\theta_{r3}}w\right) \\ +c_{\theta_{r5}}s_{\theta_{r3}}s_{\theta_{r4}}w\end{matrix}\right) -$$
$$\left(s_{roll}s_{yaw} + c_{roll}c_{yaw}s_{pitch}\right)\left(\begin{matrix}c_{\theta_{r3}}c_{\theta_{r6}}w - c_{\theta_{r5}}s_{\theta_{r6}}\left(wl + c_{\theta_{r4}}s_{\theta_{r3}}w\right) \\ +s_{\theta_{r3}}s_{\theta_{r4}}s_{\theta_{r5}}s_{\theta_{r6}}w\end{matrix}\right) - \quad (3.35)$$
$$c_{pitch}c_{yaw}\left(\begin{matrix}\frac{cxd}{2} + c_{\theta_{r5}}c_{\theta_{r6}}\left(wl + c_{\theta_{r4}}s_{\theta_{r3}}w\right) + c_{\theta_{r3}}s_{\theta_{r6}}w \\ -c_{\theta_{r6}}s_{\theta_{r3}}s_{\theta_{r4}}s_{\theta_{r5}}w\end{matrix}\right)$$

$$y_{Wr} = y_C - \left(c_{roll}c_{yaw} + s_{yaw}s_{pitch}s_{roll}\right)\left(\begin{matrix}\frac{cyd}{2} + s_{\theta_{r5}}\left(wl + c_{\theta_{r4}}s_{\theta_{r3}}w\right) \\ +c_{\theta_{r5}}s_{\theta_{r3}}s_{\theta_{r4}}w\end{matrix}\right) +$$
$$\left(s_{roll}c_{yaw} - c_{roll}s_{yaw}s_{pitch}\right)\left(\begin{matrix}c_{\theta_{r3}}c_{\theta_{r6}}w - c_{\theta_{r5}}s_{\theta_{r6}}\left(wl + c_{\theta_{r4}}s_{\theta_{r3}}w\right) \\ +s_{\theta_{r3}}s_{\theta_{r4}}s_{\theta_{r5}}s_{\theta_{r6}}w\end{matrix}\right) - \quad (3.36)$$
$$c_{pitch}s_{yaw}\left(\begin{matrix}\frac{cxd}{2} + c_{\theta_{r5}}c_{\theta_{r6}}\left(wl + c_{\theta_{r4}}s_{\theta_{r3}}w\right) + c_{\theta_{r3}}s_{\theta_{r6}}w \\ -c_{\theta_{r6}}s_{\theta_{r3}}s_{\theta_{r4}}s_{\theta_{r5}}w\end{matrix}\right)$$

$$z_{Wr} = z_C + s_{pitch}\left(\begin{matrix}\frac{cxd}{2} + c_{\theta_{r5}}c_{\theta_{r6}}\left(wl + c_{\theta_{r4}}s_{\theta_{r3}}w\right) + c_{\theta_{r3}}s_{\theta_{r6}}w \\ -c_{\theta_{r6}}s_{\theta_{r3}}s_{\theta_{r4}}s_{\theta_{r5}}w\end{matrix}\right) -$$
$$c_{roll}c_{pitch}\left(c_{\theta_{r3}}c_{\theta_{r6}}w - c_{\theta_{r5}}s_{\theta_{r6}}\left(wl + c_{\theta_{r4}}s_{\theta_{r3}}w\right) + s_{\theta_{r3}}s_{\theta_{r4}}s_{\theta_{r5}}s_{\theta_{r6}}w\right) - \quad (3.37)$$
$$c_{pitch}s_{roll}\left(\frac{cyd}{2} + s_{\theta_{r5}}\left(wl + c_{\theta_{r4}}s_{\theta_{r3}}w\right) + c_{\theta_{r5}}s_{\theta_{r3}}s_{\theta_{r4}}w\right)$$

Following the same procedure as with the left wheel, the next step is to validate the MOBILE iterative inverse kinematics values with the above analytic equations, Equation (3.38), Equation (3.39) and Equation (3.40). We do that by inputting the independent variables of the right wheel lever and wheel system (z_{Wr} and θ_{r5}) on MOBILE in a width of values and then store the according output values of the dependent variables (θ_{r6}, θ_{r3} and θ_{r2}). Also, we measure the x_{Wr} and y_{Wr} values for these widths, using MOBILE's measurement capabilities and we store them too.

After that, we input the dependent values (θ_{r6}, θ_{r3} and θ_{r2}) we stored in the above analytic equations on MATLAB, which gives us the calculated values of the rear right wheel tip position. Finally, we compare these calculated values to the z_{Wr} values that we inputted on MOBILE and x_{Wr}, y_{Wr} that we measured from MOBILE and observe their divergence.

1st Verification test

For the first verification we input to MOBILE the next range of z_{Wr} values, [0.90m, -0.70m] with a step of 0.01 m. For every z_{Wr} value we get a value for each one of the dependent variables and store the results. While doing that, we keep the other independent variable, θ_{r5} at a fixed value. The values of the variables that affect the rear right wheel's tip position can be seen in Table 3-12:

Table 3-12: The values of the variables affecting the rear right wheel's position for the first verification.

z_{Wr} (m)	θ_{r5} (deg)	x_C (m)	y_C (m)	z_C (m)	ϕ_{yaw} (deg)	ϕ_{pitch} (deg)	ϕ_{roll} (deg)
[0.90, -0.70]	0°	0	0	2	0°	0°	0°

Next, we use the values of the dependent variables θ_{r6}, θ_{r3}, θ_{r2}, x_{Wr} and y_{Wr} we get from MOBILE, and the rest that are on a fixed value to calculate $x_{Wrcalculated}$, $y_{Wrcalculated}$ and $z_{Wrcalculated}$ on MATLAB. Figure 3-47 represents the error diagrams between the calculated and desired position for the first verification of the right wheel system.

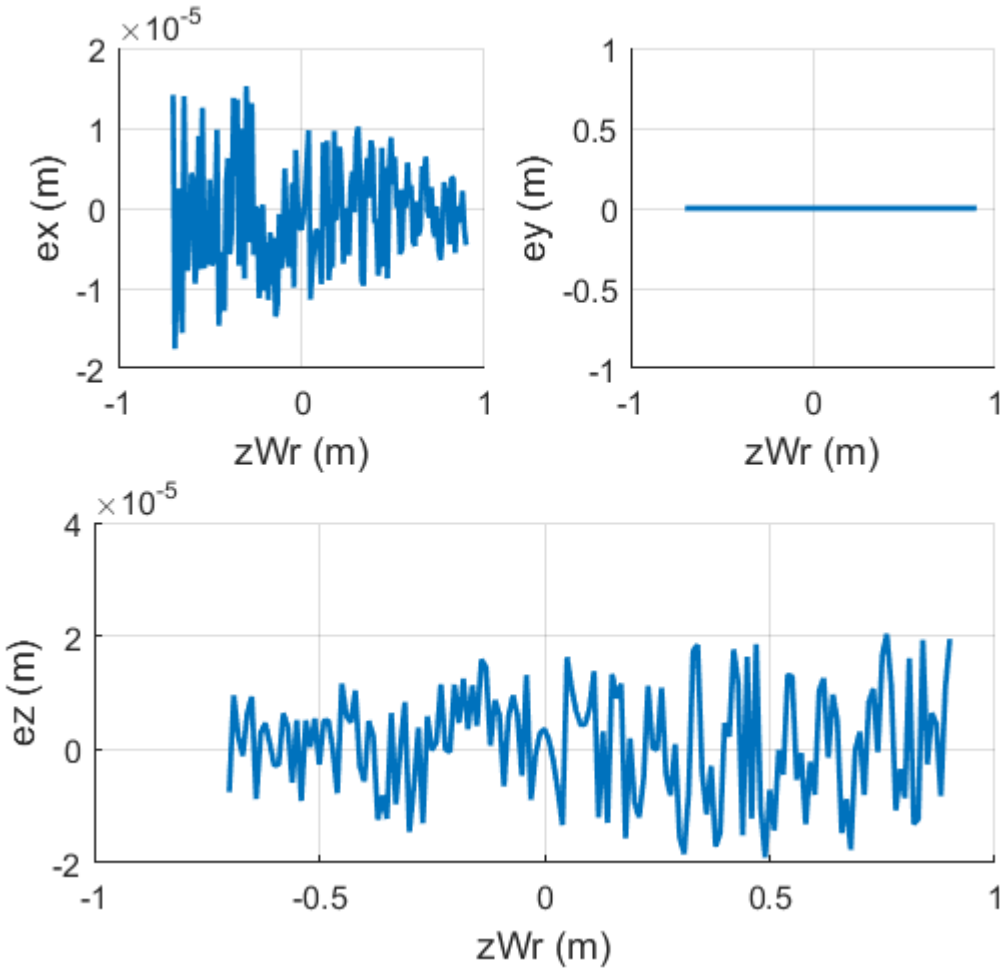

Figure 3-47: x_{Wr}, y_{Wr} and z_{Wr} errors calculated as (x_{Wr}-$x_{Wrcalculated}$), (y_{Wr}-$y_{Wrcalculated}$) and (z_{Wr}-$z_{Wrcalculated}$) when the varying value is z_{Wr}.

We observe that the errors are of the order of 10^{-5} m, which is acceptably small.

2nd Verification test

For the second verification we input to MOBILE the next range of θ_{r5} values, [-40.0°, 15.0°] with a step of 0.1°. For every θ_{r5} value we get a value for each one of the dependent variables and store the results. While doing that, we keep the other independent variable, z_{Wr} at a fixed value. The values of the variables that affect the rear right wheel's tip position can be seen in Table 3-13:

Table 3-13: The values of the variables affecting the rear right wheel's position for the second verification.

z_{Wr} (m)	θ_{r5} (deg)	x_C (m)	y_C (m)	z_C (m)	ϕ_{yaw} (deg)	ϕ_{pitch} (deg)	ϕ_{roll} (deg)
0	[-40.0°, 15.0°]	0	0	2	0°	0°	0°

Then, like the previous verification, we use the values of the dependent variables θ_{r6}, θ_{r3}, θ_{r2}, x_{Wr} and y_{Wr} we get from MOBILE, and the rest that are on a fixed value to calculate $x_{Wrcalculated}$, $y_{Wrcalculated}$ and $z_{Wrcalculated}$ on MATLAB. The next figures represent the error diagrams between the calculated and desired position for the second verification of the right wheel system:

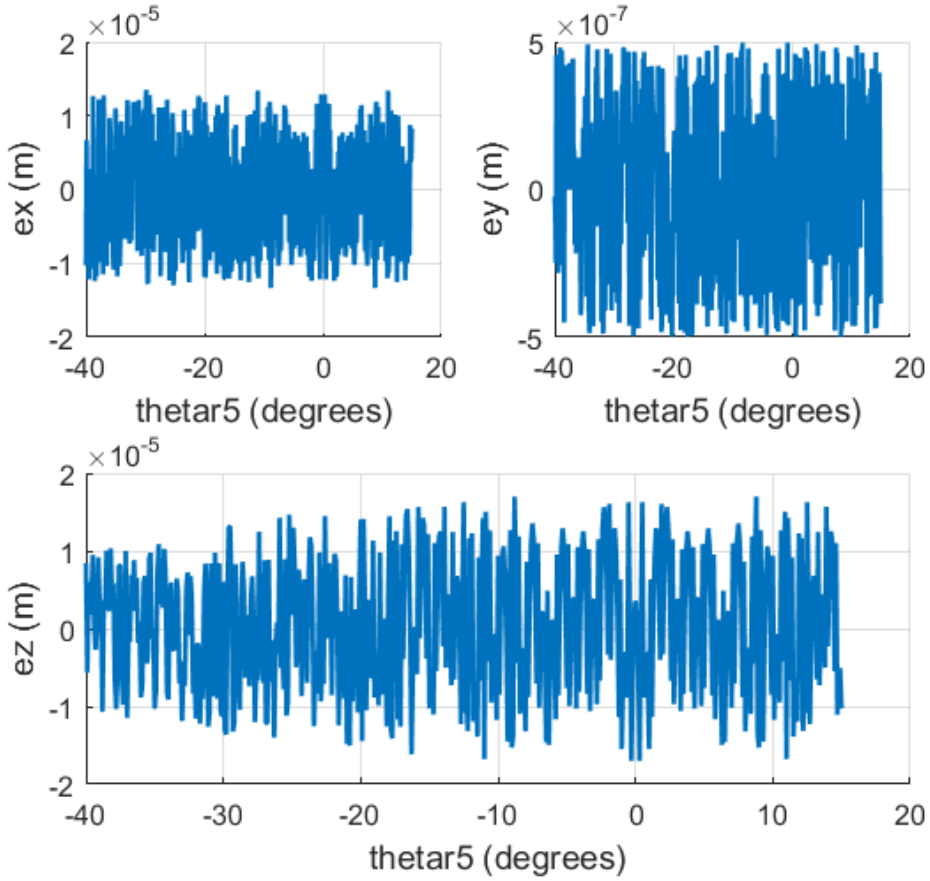

Figure 3-48: x_{Wr}, y_{Wr} and z_{Wr} errors calculated as (x_{Wr}-$x_{Wrcalculated}$), (y_{Wr}-$y_{Wrcalculated}$) and (z_{Wr}-$z_{Wrcalculated}$) when the varying value is θ_{r5}.

We observe that the errors are of the order of 10^{-5} m or even smaller, which is acceptably small.

All the errors between the MOBILE values and the expected values from the analytical equations are of 10^{-5}m order or even smaller. Compared to the dimensions of the model (see Figure 3-25, Figure 3-26, Figure 3-27 and Table 3-1) these errors are insignificant and therefore we can conclude that MOBILE's iterative method for the inverse kinematics is acceptably accurate and also easier to apply than the closed-form solution.

4 Non-holonomic constraints

These constraints arise in systems such as multifingered robot hands and wheeled mobile robots, in which rolling contact is involved, as well as in systems in which angular momentum is conserved.

Non-holonomic behaviour in robotic systems is particularly interesting, because it implies that the mechanism can be completely controlled with a reduced number of actuators. On the other hand, both planning and control are much more difficult than in conventional holonomic systems, and require special techniques.

In many interesting cases, the system motion is subject to constraints that may arise from the structure itself of the mechanism, or from the way in which it is actuated and controlled. Various classifications of such constraints can be devised. For example, constraints may be expressed as equalities or inequalities (respectively, bilateral or unilateral constraints) and they may explicitly depend on time or not (rheonomic or scleronomic constraints). Motion restrictions that may be put in the next form are called holonomic constraints.

$$\begin{aligned} &h_i(q) = 0, \\ &i = 1, ..., k < n \end{aligned} \tag{4.1}$$

A system whose constraints, if any, are all holonomic, is called a holonomic system. Holonomic constraints are typically introduced by mechanical interconnections between the various bodies of the system. For example, prismatic and revolute joints commonly used in robotic manipulators are a source of such constraints. If we consider a fixed-base kinematic chain composed of n rigid links connected by elementary joints, the composite configuration space of the system is $(\mathbb{R}^3 \times SO(3))^n$. Since each joint imposes five constraints, the number of degrees of freedom is 6n-5n=n.

System constraints whose expression involves generalized coordinates and velocities in the form of Equation (4.2) are referred to as kinematic constraints.

$$\begin{aligned} &a_i(q, \dot{q}) = 0, \\ &i = 1, ..., k < n \end{aligned} \tag{4.2}$$

These will limit the admissible motions of the system by restricting the set of generalized velocities that can be attained at a given configuration. Of course, the holonomic constraints (4.1) imply the existence of kinematic constraints expressed as:

$$\begin{aligned} &\frac{\partial h_i}{\partial q} \dot{q} = 0, \\ &i = 1, ..., k \end{aligned} \tag{4.3}$$

However, the converse is not necessarily true: it may happen that the kinematic constraints (4.2) are not integrable, i.e., they cannot be put in the form (4.1). In this case, the constraints

and the mechanical system itself are called nonholonomic. For a mechanical system with n generalized coordinates and k nonholonomic constraints, although the generalized velocities at each point are confined to an (n-k) dimensional subspace, accessibility of the entire configuration space is preserved.

Now regarding the robot under study, in addition to the holonomic subsystem described in the “Complete kinematic model” chapter (see Figure 4-1 for the kinematic model), the interaction of the wheels with the ground gives rise to two nonholonomic, i.e., non-integrable kinematic constraints:

- Velocity of immaterial contact point (cornering constraint):

$$\mathbf{v_W} \cdot \mathbf{n_W} = 0 \tag{4.4}$$

- wheel rate (traction constraint, no slip condition):

$$\boldsymbol{\omega_W} \cdot \mathbf{n_W} - \|v_W\| / r_W = 0 \tag{4.5}$$

where r_w is the wheel radius and $\|\cdot\|$ represents the Euclidean norm. The terms in Equation (4.4) and Equation (4.5) can be seen in Figure 4-1. Equation (4.4) restricts the values of the generalized speeds $\dot{q}$, while Equation (4.5) affects only the wheel rotation. As the wheel rotation angle does not appear in the 38 variables, only the first nonholonomic constraint per wheel needs to be regarded. Moreover, as the velocity v_W of the immaterial wheel contact point depends only on the generalized coordinates of cabin and wheel levers, the cornering nonholonomic constraints affect the next subset of input velocities:

$$\dot{q}_I = [\dot{x}_C, \dot{y}_C, \dot{z}_C, \dot{\varphi}_{yaw}, \dot{\varphi}_{pitch}, \dot{\varphi}_{roll}, \dot{\theta}_{l5}, \dot{\theta}_{r5}]^T \tag{4.6}$$

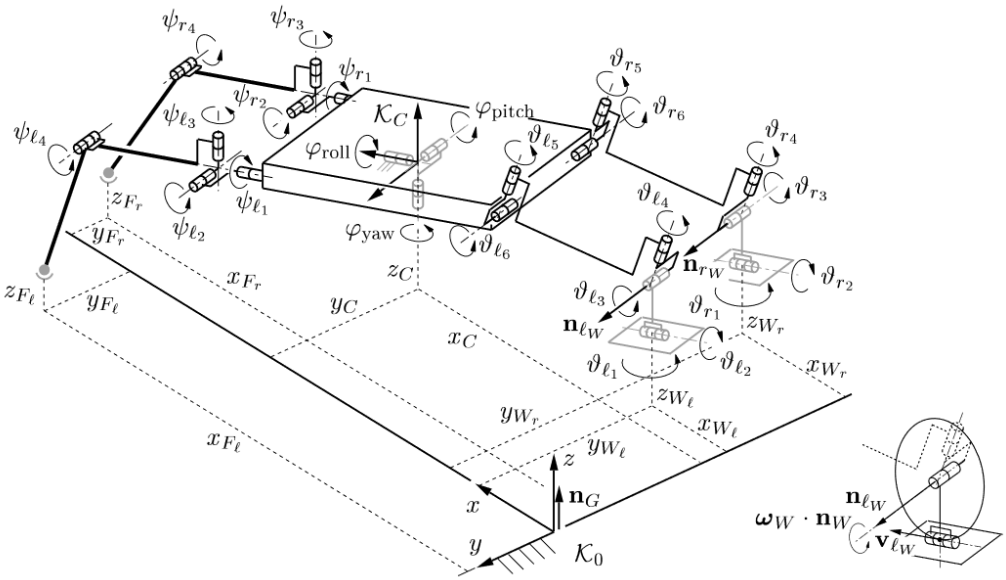

Figure 4-1: Robot’s variables.

Of these, the operator can command a restricted subset of six independent commanded velocities because of the two nonholonomic constraints that confine two generalized velocities. These other two generalized velocities result as linear functions of these

velocities. Put together, one obtains a set of ordinary differential equations $\dot{q}_I = A(q_I)\dot{\pi}(t)$ which upon being integrated (numerically) yield the values of generalized coordinates q_I. Once these variables are known, they can be used in conjunction with the other generalized variables to compute the complete configuration of the vehicle at any point.

4.1 Calculation of the dependent velocities on MOBILE

4.1.1 Velocity constraints

Let's assume for now that the dependent variables $\dot{\beta}$ are the sway of the cabin (side to side motion along the y axis of the cabin-fixed coordinate frame K_C) and the yaw of the cabin (rotation of the cabin about the vertical axis z of the cabin-fixed coordinate frame K_C).

$$\dot{\beta} = \begin{bmatrix} \dot{\beta}_1 : sway \\ \dot{\beta}_2 : yaw \end{bmatrix} \tag{4.7}$$

The constraints that the wheels add, as mentioned before, are that the wheels cannot move in the y direction of the wheel-tip-fixed coordinate frame, therefore:

$$\dot{\varphi} = \begin{bmatrix} \dot{\varphi}_1 \\ \dot{\varphi}_2 \end{bmatrix} \triangleq 0 \tag{4.8}$$

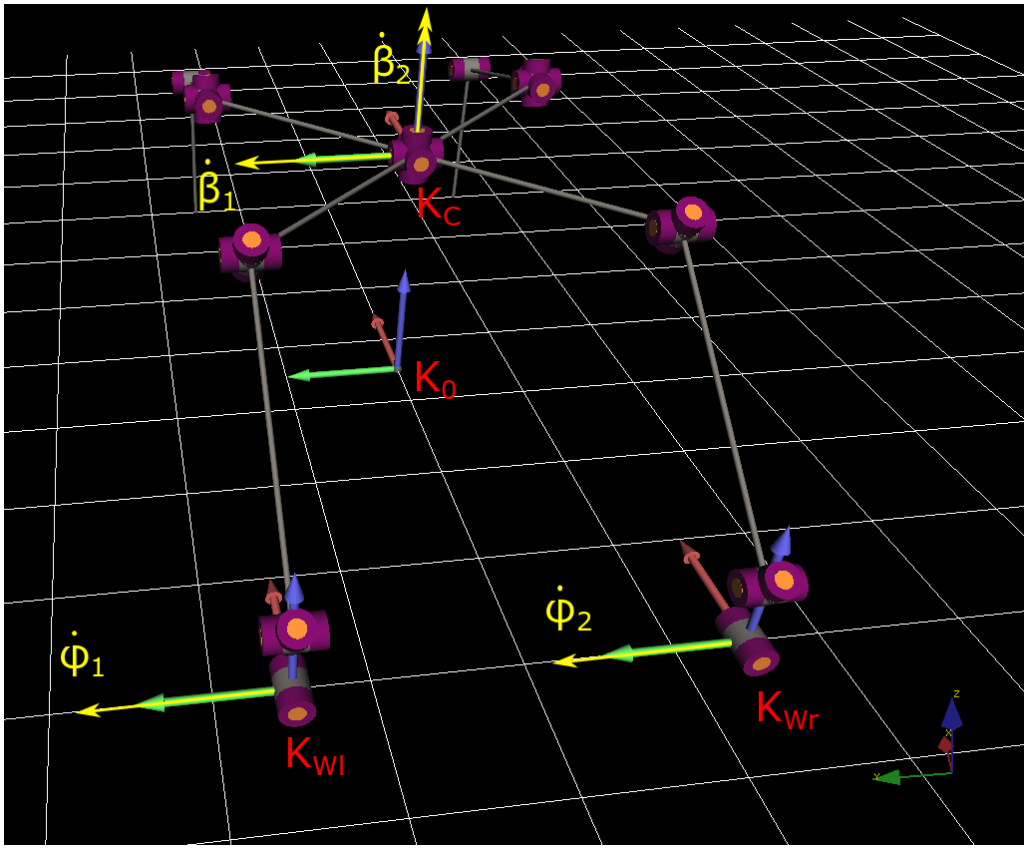

Figure 4-2: The dependent velocity variables, β_1 and β_2, due to the nonholonomic constraints φ_1 and φ_2.

The Equation (4.9) describes the wheel tip velocity $\dot{\varphi}$:

$$\dot{\varphi} = J_\beta \dot{\beta} + \hat{\dot{\varphi}} = 0 \tag{4.9}$$

In this equation $\hat{\dot{\varphi}}$ is the partial wheel tip velocity due to inputs (independent variables $\dot{q}_I$) and $J_\beta \dot{\beta}$ is the partial wheel tip velocity due to the dependent variables. Thus, we can calculate the dependent variables' values from the following Equation (4.10):

$$\dot{\beta} = -J_\beta{}^{-1} \hat{\dot{\varphi}} \tag{4.10}$$

Using the computational capabilities of MOBILE, we will calculate the $\hat{\dot{\varphi}}$ and the J_β values.

4.1.2 Force-based determination of J_β

Computations in object-oriented multibody libraries are typically based on a Jacobian-free formulation, since this simplifies considerably the implementation of new elements and the expansion of the library. In order to compute the Jacobians, the following derivative-free methods proposed by Kecskemethy and Hiller (1994) are available. Regarding the general kinetostatic transmission element depicted in Figure 2-2, the velocity Jacobian J_φ can be computed using (a) the element's velocity transmission or (b) the element's force transmission.

a) Velocity-based determination of Jacobians

Setting all velocity components at the input of the transmission element besides the jth-one equal to zero, and the jth-one equal to one, yields an output velocity vector which is identical to the jth-column of the Jacobian:

$$J_\varphi = \dot{q}_{in} \Big|_{\dot{q}_{in}^{(i)} = \begin{cases} 1, for & i=j \\ 0, otherwise \end{cases}} \tag{4.11}$$

b) Force-based determination of Jacobians

Setting all force components at the output of the transmission element besides the jth-one equal to zero, and the jth-one equal to one, yields a vector of generalized forces at the input of the transmission element which is identical to the jth-column of the transposed Jacobian, thus to its jth-row:

$$J_\varphi = Q_{in} \Big[_{Q_{out}^{(j)} = \begin{cases} 1, for & i=j \\ 0, otherwise \end{cases}} \tag{4.12}$$

In our problem we want to calculate the Jacobian (J_β) that connects the velocity dependent variables $\dot{\beta}$ with the wheels' tip velocity $\dot{\varphi}$ (see Figure 4-3). Then the kinetostatic element becomes:

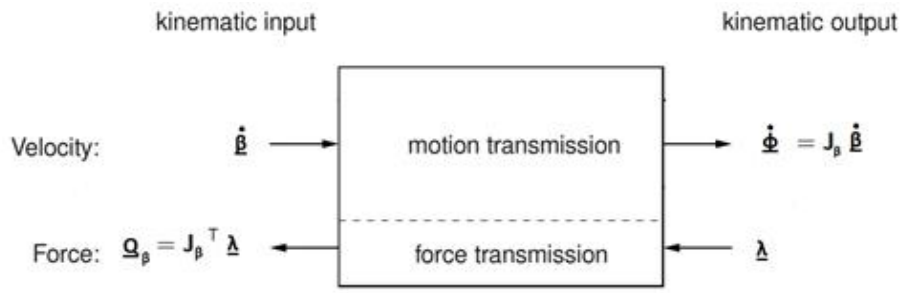

Figure 4-3: Kinetostatic transmission element connecting the dependent variables β and the wheels' tip velocity φ.

Where λ is the force at the wheel tip y direction. Firstly, we input $\lambda = \begin{bmatrix} 1 & 0 \end{bmatrix}^T$ in order to find the first row of the Jacobian that refers to the left wheel and then we input $\lambda = \begin{bmatrix} 0 & 1 \end{bmatrix}^T$ to find the second row of the Jacobian that refers to the right wheel. After inputting each λ value we do a force transmission in MOBILE and read the Q_β values (force or torque of the dependent variables) which are the force/torque values required for static equilibrium if at the output the force acting on the wheels' tip y direction is λ(N). By following this procedure each time, the position of the mechanism changes we can calculate the required Jacobian in each position. We now only miss $\dot{\hat{\varphi}}$ to be able to calculate the dependent velocities values in every time step.

4.1.3 Calculation of partial wheels' tip velocity due to independent variables

From Equation (4.10), we see that the last thing missing in order to calculate the values of the dependent velocities is the calculation of $\dot{\hat{\varphi}}$. That term expresses the effect that the independent velocities $\dot{q}_I$ have to the wheels' tip velocity on local y direction. In order to compute that term, we instantaneously set the dependent velocities to zero so the Equation (4.9) becomes:

$$\dot{\varphi} = \dot{\hat{\varphi}}$$

And then we do a velocity transmission in MOBILE and read the $\dot{\varphi}$ values.

4.1.4 Calculation of the dependent velocities

Now having both the Jacobian J_β and the $\dot{\hat{\varphi}}$ values we can calculate the values of the dependent velocities, by doing the above in every time step, from the following equations derived from Equation (4.10):

$$\dot{\beta} = -J_{\beta}^{-1}\hat{\dot{\varphi}} \Rightarrow$$

$$\begin{bmatrix} \dot{\beta}_1 \\ \dot{\beta}_2 \end{bmatrix} = -\begin{bmatrix} J_{\beta_{11}} & J_{\beta_{12}} \\ J_{\beta_{21}} & J_{\beta_{22}} \end{bmatrix}^{-1} \hat{\dot{\varphi}} = -\frac{1}{J_{\beta_{11}}J_{\beta_{22}} - J_{\beta_{12}}J_{\beta_{21}}} \begin{bmatrix} J_{\beta_{22}} & -J_{\beta_{12}} \\ -J_{\beta_{21}} & J_{\beta_{11}} \end{bmatrix} \begin{bmatrix} \hat{\dot{\varphi}}_1 \\ \hat{\dot{\varphi}}_2 \end{bmatrix} \Rightarrow$$

$$\dot{\beta}_1 = \frac{\hat{\dot{\varphi}}_1 J_{\beta_{22}} - \hat{\dot{\varphi}}_2 J_{\beta_{12}}}{J_{\beta_{11}}J_{\beta_{22}} - J_{\beta_{12}}J_{\beta_{21}}} \qquad (4.13)$$

$$\dot{\beta}_2 = \frac{\hat{\dot{\varphi}}_2 J_{\beta_{11}} - \hat{\dot{\varphi}}_1 J_{\beta_{21}}}{J_{\beta_{11}}J_{\beta_{22}} - J_{\beta_{12}}J_{\beta_{21}}}$$

We observe that it is possible for the denominators of $\dot{\beta}_1$ and $\dot{\beta}_2$ to be zero. That happens when both wheel axes are collinear. In that occasion the two non-holonomic constraints of the wheels collapse into a corresponding single one at the midpoint of the common axis. In this thesis we only analyse the more complex behaviour of non-collinear wheel axes, e.g., when the vehicle is in operation in uneven terrain.

4.2 Analytical verification of MOBILE's dependent velocities values

In this chapter the dependent velocities values, calculated following the procedure of the previous chapter, will be checked analytically in terms of satisfying the nonholonomic constraints.

4.2.1 Left wheel verification

We have the same notations for the left wheel variables and distances names as described on the inverse kinematics chapter (see Figure 3-42) and also the same frames distribution (see Figure 3-41) and vehicle geometry. The equations that describe the left wheel's tip position with respect to the ground-fixed coordinate frame K_0 were derived in the inverse kinematics chapter and are the following:

$$
\begin{bmatrix} x_{Wl} \\ y_{Wl} \\ z_{Wl} \end{bmatrix} = \begin{bmatrix}
x_C - \frac{\left(cyd\left(c_{roll}s_{yaw} - c_{yaw}s_{pitch}s_{roll}\right)\right)}{2} \\
+w\begin{pmatrix} c_{\theta_{l4}}s_{\theta_{l3}}\begin{pmatrix} s_{\theta_{l5}}\left(c_{roll}s_{yaw} - c_{yaw}s_{pitch}s_{roll}\right) + \\ c_{\theta_{l5}}s_{\theta_{l6}}\left(s_{yaw}s_{roll} + c_{roll}c_{yaw}s_{pitch}\right) - c_{pitch}c_{\theta_{l5}}c_{\theta_{l6}}c_{yaw} \end{pmatrix} - \\ c_{\theta_{l3}}\left(c_{\theta_{l6}}\left(s_{roll}s_{yaw} + c_{roll}c_{yaw}s_{pitch}\right) + c_{yaw}c_{pitch}s_{\theta_{l6}}\right) + \\ s_{\theta_{l3}}s_{\theta_{l4}}\begin{pmatrix} c_{\theta_{l5}}\left(c_{roll}s_{yaw} - c_{yaw}s_{pitch}s_{roll}\right) - \\ s_{\theta_{l5}}s_{\theta_{l6}}\left(s_{roll}s_{yaw} + c_{roll}c_{yaw}s_{pitch}\right) + c_{pitch}c_{\theta_{l6}}c_{yaw}s_{\theta_{l5}} \end{pmatrix} \end{pmatrix} \\
+wl\begin{pmatrix} s_{\theta_{l5}}\left(c_{roll}s_{yaw} - c_{yaw}s_{pitch}s_{roll}\right) \\ +c_{\theta_{l5}}s_{\theta_{l6}}\left(s_{roll}s_{yaw} + c_{roll}c_{yaw}s_{pitch}\right) \\ -c_{pitch}c_{\theta_{l5}}c_{\theta_{l6}}c_{yaw} \end{pmatrix} - \frac{c_{pitch}c_{yaw}cxd}{2} \\
y_C + \frac{\left(cyd\left(c_{roll}c_{yaw} + s_{yaw}s_{pitch}s_{roll}\right)\right)}{2} \\
+w\begin{pmatrix} c_{\theta_{l3}}\begin{pmatrix} c_{\theta_{l6}}\left(s_{roll}c_{yaw} - s_{yaw}s_{pitch}c_{roll}\right) - \\ c_{pitch}s_{\theta_{l6}}s_{yaw} \end{pmatrix} - \\ c_{\theta_{l4}}s_{\theta_{l3}}\begin{pmatrix} s_{\theta_{l5}}\left(c_{roll}c_{yaw} + s_{roll}s_{yaw}s_{pitch}\right) + s_{\theta_{l6}}c_{\theta_{l5}}\begin{pmatrix} c_{yaw}s_{roll} - \\ c_{roll}s_{yaw}s_{pitch} \end{pmatrix} + \\ c_{pitch}c_{\theta_{l5}}c_{\theta_{l6}}s_{yaw} \end{pmatrix} + \\ s_{\theta_{l3}}s_{\theta_{l4}}\begin{pmatrix} s_{\theta_{l5}}s_{\theta_{l6}}\left(s_{roll}c_{yaw} - s_{yaw}s_{pitch}c_{roll}\right) - \\ c_{\theta_{l5}}\left(c_{roll}c_{yaw} + s_{roll}s_{yaw}s_{pitch}\right) + c_{pitch}c_{\theta_{l6}}s_{yaw}s_{\theta_{l5}} \end{pmatrix} \end{pmatrix} \\
-wl\begin{pmatrix} s_{\theta_{l5}}\left(c_{roll}c_{yaw} + s_{yaw}s_{pitch}s_{roll}\right) \\ +c_{\theta_{l5}}s_{\theta_{l6}}\left(s_{roll}c_{yaw} - c_{roll}s_{yaw}s_{pitch}\right) \\ +c_{pitch}c_{\theta_{l5}}c_{\theta_{l6}}s_{yaw} \end{pmatrix} - \frac{c_{pitch}s_{yaw}cxd}{2} \\
z_C + \frac{\left(s_{pitch}cxd\right)}{2} + wl\left(c_{\theta_{l5}}c_{\theta_{l6}}s_{pitch} - c_{pitch}s_{roll}s_{\theta_{l5}} + c_{pitch}c_{roll}s_{\theta_{l6}}c_{\theta_{l5}}\right) \\
+w\begin{pmatrix} c_{\theta_{l3}}\left(s_{pitch}s_{\theta_{l6}} - c_{pitch}c_{roll}c_{\theta_{l6}}\right) + \\ c_{\theta_{l4}}s_{\theta_{l3}}\left(c_{\theta_{l5}}c_{\theta_{l6}}s_{pitch} - c_{pitch}s_{roll}s_{\theta_{l5}} + c_{pitch}c_{roll}c_{\theta_{l5}}s_{\theta_{l6}}\right) - \\ s_{\theta_{l3}}s_{\theta_{l4}}\left(c_{pitch}c_{\theta_{l5}}s_{roll} + c_{\theta_{l6}}s_{pitch}s_{\theta_{l5}} + c_{pitch}c_{roll}s_{\theta_{l5}}s_{\theta_{l6}}\right) \end{pmatrix} + \frac{c_{pitch}s_{roll}cyd}{2}
\end{bmatrix} \tag{4.14}
$$

By deriving the Equations (3.26), (3.27) and (3.28) summarized above, we get the velocity of the left wheel end frame with respect to the ground-fixed coordinate frame K_0:

$$
\dot{r}_{Wl} = \begin{bmatrix} \dot{x}_{Wl} \\ \dot{y}_{Wl} \\ \dot{z}_{Wl} \end{bmatrix} \tag{4.15}
$$

For the velocity constraint to be satisfied, the y component of velocity on the wheel tip-fixed coordinate frame (K_{Wl}) must be zero. To calculate that velocity we multiply $\dot{r}_{Wl}$ with the orientation of the y unit vector of K_{Wl}, $\hat{y}_{Wl}$ (see Figure 4-4), with respect to the ground-fixed frame K_0. We get that orientation vector from the second column of the rotation matrix ${}^{0}R_{12}$ that connects K_0 and K_{12}(=K_{Wl}) and has been calculated on the inverse kinematics chapter.

$$
\begin{aligned}
&{}^{0}R_{12} = {}^{0}R_{4} \cdot {}^{4}R_{7} \cdot {}^{7}R_{10} \cdot {}^{10}R_{12} = \\
&= rot_z(\varphi_{yaw}) \cdot rot_y(\varphi_{pitch}) \cdot rot_x(\varphi_{roll}) \cdot rot_y(\theta_{l6}) \cdot rot_z(\theta_{l5}) \cdot rot_z(\theta_{l4}) \cdot rot_y(\theta_{l3}) \cdot rot_x\left(\theta_{l2}\right)
\end{aligned}
\tag{4.16}
$$

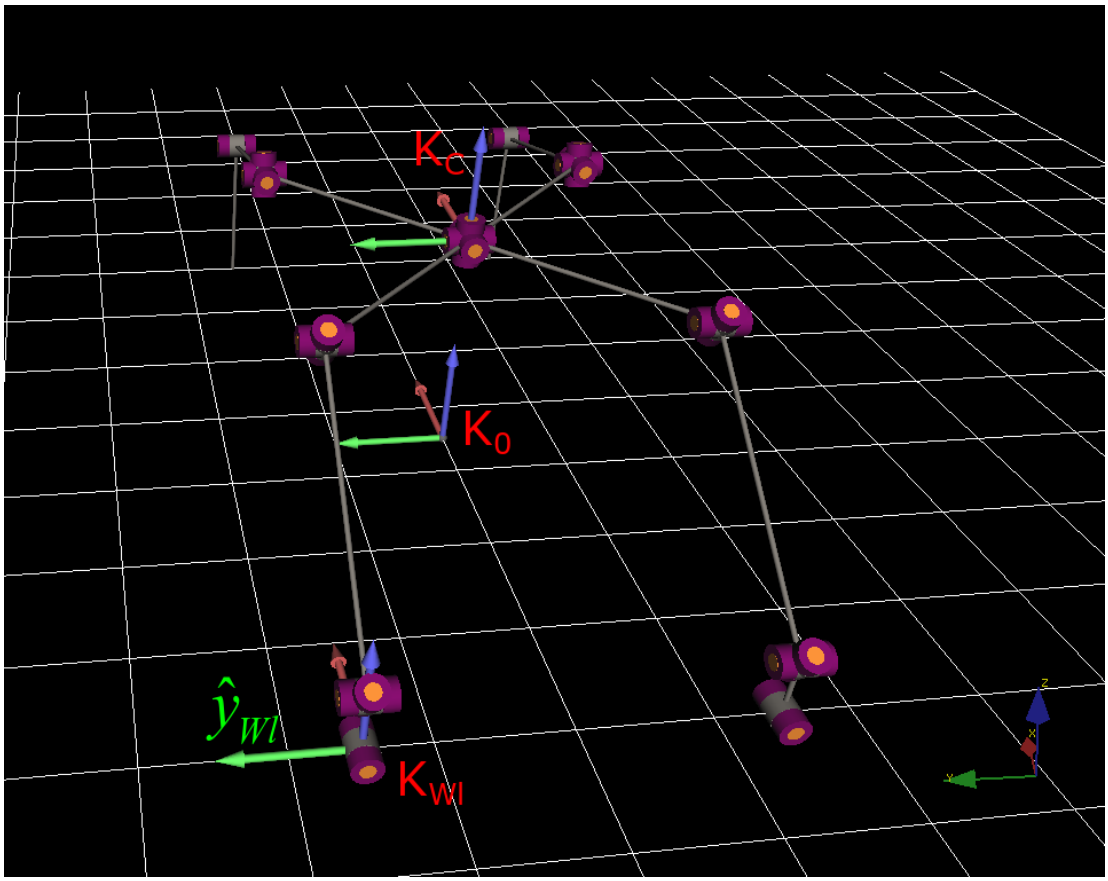

Figure 4-4: The y unit vector of K_{Wl}, y_{Wl}. The wheel's velocity must be zero in that direction for the nonholonomic constraint to be satisfied.

$$\hat{y}_{Wl} = \begin{bmatrix} -s_{\theta_{l2}}\left(c_{\theta_{l4}}s_{\theta_{l3}}\left(\begin{array}{l} s_{\theta_{l5}}\left(\begin{array}{l} c_{roll}s_{yaw} \\ -c_{yaw}s_{pitch}s_{roll} \end{array}\right) \\ +c_{\theta_{l5}}s_{\theta_{l6}}\left(\begin{array}{l} s_{roll}s_{yaw} \\ +c_{roll}c_{yaw}s_{pitch} \end{array}\right) \\ -c_{pitch}c_{\theta_{l5}}c_{\theta_{l6}}c_{yaw} \end{array}\right) - c_{\theta_{l3}}\left(\begin{array}{l} c_{\theta_{l6}}\left(\begin{array}{l} s_{roll}s_{yaw} \\ +c_{roll}c_{yaw}s_{pitch} \end{array}\right) \\ +c_{pitch}c_{yaw}s_{\theta_{l6}} \end{array}\right) + \right. \\ \left. s_{\theta_{l3}}s_{\theta_{l4}}\left(\begin{array}{l} c_{\theta_{l5}}\left(\begin{array}{l} c_{roll}s_{yaw} \\ -c_{yaw}s_{pitch}s_{roll} \end{array}\right) \\ -s_{\theta_{l5}}s_{\theta_{l6}}\left(\begin{array}{l} s_{roll}s_{yaw} \\ +c_{roll}c_{yaw}s_{pitch} \end{array}\right) \\ +c_{pitch}c_{\theta_{l6}}c_{yaw}s_{\theta_{l5}} \end{array}\right)\right) - c_{\theta_{l2}}\left(\begin{array}{l} c_{\theta_{l4}}\left(\begin{array}{l} c_{\theta_{l5}}\left(\begin{array}{l} c_{roll}s_{yaw} \\ -c_{yaw}s_{pitch}s_{roll} \end{array}\right) \\ -s_{\theta_{l5}}s_{\theta_{l6}}\left(\begin{array}{l} s_{roll}s_{yaw} \\ +c_{roll}c_{yaw}s_{pitch} \end{array}\right) \\ +c_{pitch}c_{\theta_{l6}}c_{yaw}s_{\theta_{l5}} \end{array}\right) \\ -s_{\theta_{l4}}\left(\begin{array}{l} s_{\theta_{l5}}\left(\begin{array}{l} c_{roll}s_{yaw} \\ -c_{yaw}s_{pitch}s_{roll} \end{array}\right) + \\ c_{\theta_{l5}}s_{\theta_{l6}}\left(\begin{array}{l} s_{roll}s_{yaw} \\ +c_{roll}c_{yaw}s_{pitch} \end{array}\right) - \\ c_{pitch}c_{\theta_{l5}}c_{\theta_{l6}}c_{yaw} \end{array}\right) \end{array}\right) \\ -s_{\theta_{l2}}\left(\begin{array}{l} c_{\theta_{l3}}\left(\begin{array}{l} c_{\theta_{l6}}\left(\begin{array}{l} c_{yaw}s_{roll} \\ -c_{roll}s_{pitch}s_{yaw} \end{array}\right) \\ -c_{pitch}s_{\theta_{l6}}s_{yaw} \end{array}\right) \\ -c_{\theta_{l4}}s_{\theta_{l3}}\left(\begin{array}{l} s_{\theta_{l5}}\left(\begin{array}{l} c_{roll}c_{yaw} \\ +s_{roll}s_{yaw}s_{pitch} \end{array}\right) \\ +c_{\theta_{l5}}s_{\theta_{l6}}\left(\begin{array}{l} c_{yaw}s_{roll} \\ -c_{roll}s_{pitch}s_{yaw} \end{array}\right) \\ +c_{pitch}c_{\theta_{l5}}c_{\theta_{l6}}s_{yaw} \end{array}\right) \\ +s_{\theta_{l3}}s_{\theta_{l4}}\left(\begin{array}{l} s_{\theta_{l5}}s_{\theta_{l6}}\left(\begin{array}{l} c_{yaw}s_{roll} \\ -c_{roll}s_{pitch}s_{yaw} \end{array}\right) \\ -c_{\theta_{l5}}\left(\begin{array}{l} c_{roll}c_{yaw} \\ +s_{roll}s_{yaw}s_{pitch} \end{array}\right) \\ +c_{pitch}c_{\theta_{l6}}s_{yaw}s_{\theta_{l5}} \end{array}\right) \end{array}\right) - c_{\theta_{l2}}\left(\begin{array}{l} s_{\theta_{l4}}\left(\begin{array}{l} s_{\theta_{l5}}\left(\begin{array}{l} c_{roll}c_{yaw} \\ +s_{yaw}s_{pitch}s_{roll} \end{array}\right) \\ +c_{\theta_{l5}}c_{\theta_{l6}}\left(\begin{array}{l} s_{roll}c_{yaw} \\ -c_{roll}s_{yaw}s_{pitch} \end{array}\right) \\ +c_{pitch}c_{\theta_{l5}}s_{yaw}c_{\theta_{l6}} \end{array}\right) \\ +c_{\theta_{l4}}\left(\begin{array}{l} s_{\theta_{l5}}s_{\theta_{l6}}\left(\begin{array}{l} c_{yaw}s_{roll} \\ -c_{roll}s_{pitch}s_{yaw} \end{array}\right) \\ -c_{\theta_{l5}}\left(\begin{array}{l} c_{roll}c_{yaw} \\ +s_{roll}s_{yaw}s_{pitch} \end{array}\right) \\ +c_{pitch}s_{\theta_{l5}}c_{\theta_{l6}}s_{yaw} \end{array}\right) \end{array}\right) \\ c_{\theta_{l2}}\left(c_{\theta_{l4}}\left(\begin{array}{l} c_{pitch}c_{\theta_{l5}}s_{roll} \\ +c_{\theta_{l6}}s_{pitch}s_{\theta_{l5}} \\ +c_{pitch}c_{roll}s_{\theta_{l5}}s_{\theta_{l6}} \end{array}\right) + s_{\theta_5}\left(\begin{array}{l} c_{\theta_{l5}}c_{\theta_{l6}}s_{pitch} \\ -c_{pitch}s_{roll}s_{\theta_{l5}} \\ +c_{pitch}c_{roll}c_{\theta_{l5}}s_{\theta_{l6}} \end{array}\right)\right) \\ -s_{\theta_{l2}}\left(c_{\theta_{l3}}\left(\begin{array}{l} s_{pitch}s_{\theta_{l6}} \\ -c_{pitch}c_{roll}c_{\theta_{l6}} \end{array}\right) + c_{\theta_{l4}}s_{\theta_{l3}}\left(\begin{array}{l} c_{\theta_{l5}}c_{\theta_{l6}}s_{pitch} \\ -c_{pitch}s_{roll}s_{\theta_{l5}} \\ +c_{pitch}c_{roll}c_{\theta_{l5}}s_{\theta_{l6}} \end{array}\right) - s_{\theta_{l3}}s_{\theta_{l4}}\left(\begin{array}{l} c_{pitch}c_{\theta_{l5}}s_{roll} \\ +c_{\theta_{l6}}s_{pitch}s_{\theta_{l5}} \\ +c_{pitch}c_{roll}s_{\theta_{l5}}s_{\theta_{l6}} \end{array}\right)\right) \end{bmatrix} \tag{4.17}$$

Thus, the constraint we must test if it is satisfied by MOBILE is:

$$\dot{r}_{Wl}{}^{T} \cdot \hat{y}_{Wl} = 0 \tag{4.18}$$

Using Equation (4.18), we run some verification tests in order to ensure that the velocity values that MOBILE calculates for the dependent variables, with the procedure explained in the previous chapter, comply with the nonholonomic constraints. For the verification tests we input to MOBILE a velocity in variables that affect the cornering nonholonomic constraints the most (these are the cabin roll angle and the left and right wheel straddle angle), then we

read and store all the values involved in the above constraint equation for a period of time and then we check in MATLAB if the MOBILE values satisfy the constraint.

1st Verification test

For the first verification test we input to MOBILE a cabin roll velocity, $\dot{\varphi}_{roll} = 5\,\mathrm{deg}/s$. The chosen step, for acceptable accuracy, of the MOBILE integrator is 0.001 s. Then we observe this scenario for 2001 velocity integrations, that means approximately for 2 s. At every integration (step in time), we store the values of all the variables involved in the constraint equation and then we input these values to this equation in MATLAB and check if they satisfy it. In Table 4-1 the initial conditions and fixed values for the first verification test of the left wheel can be seen.

Table 4-1: Initial conditions and fixed values for the first verification test of the left wheel.

x_C (m)	0.0	dx_C (m/s)	0.0	cxd (m)	3.0
y_C (m)	0.0	-	-	cyd (m)	3.0
z_C (m)	2.0	dz_C (m/s)	0.0	w (m)	0.5
x_{Fl} (m)	3.5	dx_{Fl} (m/s)	0.0	wl (m)	2.5
y_{Fl} (m)	1.5	dy_{Fl} (m/s)	0.0	thighD (m)	2.0
z_{Fl} (m)	0.0	dz_{Fl} (m/s)	0.0	legD (m)	1.5
x_{Fr} (m)	3.5	dx_{Fr} (m/s)	0.0		
y_{Fr} (m)	-1.5	dy_{Fr} (m/s)	0.0		
z_{Fr} (m)	0.0	dz_{Fr} (m/s)	0.0		
z_{Wl} (m)	0.0	dz_{Wl} (m/s)	0.0		
z_{Wr} (m)	0.05	dz_{Wr} (m/s)	0.0		
φ_{yaw} (deg)	0.0	-	-		
φ_{pitch} (deg)	0.0	$d\varphi_{pitch}$ (deg/s)	0.0		
φ_{roll} (deg)	0.0	$d\varphi_{roll}$ (deg/s)	5.0		
ψ_{l1} (deg)	0.0	$d\psi_{l1}$ (deg/s)	0.0		
ψ_{r1} (deg)	0.0	$d\psi_{r1}$ (deg/s)	0.0		
θ_{l5} (deg)	0.0	$d\theta_{l5}$ (deg/s)	0.0		
θ_{r5} (deg)	0.0	$d\theta_{r5}$ (deg/s)	0.0		

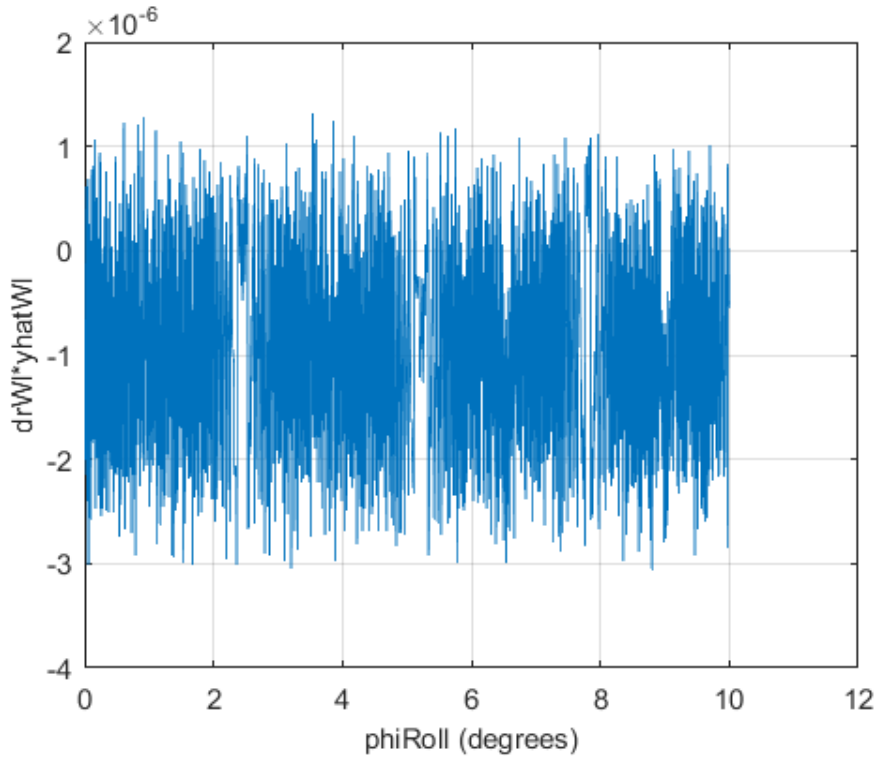

Figure 4-5: Verification of the constraint equation, $\dot{r}_{Wl}{}^{T} \cdot \hat{y}_{Wl} = 0$, with the MOBILE variable values for $\dot{\varphi}_{roll} = 5\,\mathrm{deg}/s$.

We see from Figure 4-5 that the MOBILE variable values satisfy the constraint equation for the left wheel as 10^{-6}-10^{-5} values can be considered as zero.

2nd Verification test

For the second verification test we input to MOBILE a left wheel straddle angle, $\dot{\theta}_{l5} = 0.5\,\mathrm{deg}/s$. The chosen step, for acceptable accuracy, of the MOBILE integrator is 0.001 s, as before. Then we observe this scenario for 451 velocity integrations, that means approximately for 0.451 s. At every integration (step in time), we store the values of all the variables involved in the constraint equation and then we input these values to this equation in MATLAB and check if they satisfy it, exactly like the first test. In Table 4-2 we can see the initial conditions and fixed values for the second verification test of the left wheel.

Table 4-2: Initial conditions and fixed values for the second verification test of the left wheel.

x_C (m)	0.0	dx_C (m/s)	0.0	cxd (m)	3.0
y_C (m)	0.0	-	-	cyd (m)	3.0
z_C (m)	2.0	dz_C (m/s)	0.0	w (m)	0.5
x_{Fl} (m)	3.5	dx_{Fl} (m/s)	0.0	wl (m)	2.5
y_{Fl} (m)	1.5	dy_{Fl} (m/s)	0.0	thighD (m)	2.0
z_{Fl} (m)	0.0	dz_{Fl} (m/s)	0.0	legD (m)	1.5

x_{Fr} (m)	3.5	dx_{Fr} (m/s)	0.0		
y_{Fr} (m)	-1.5	dy_{Fr} (m/s)	0.0		
z_{Fr} (m)	0.0	dz_{Fr} (m/s)	0.0		
z_{Wl} (m)	0.0	dz_{Wl} (m/s)	0.0		
z_{Wr} (m)	0.05	dz_{Wr} (m/s)	0.0		
φ_{yaw} (deg)	0.0	-	-		
φ_{pitch} (deg)	0.0	$d\varphi_{pitch}$ (deg/s)	0.0		
φ_{roll} (deg)	0.0	$d\varphi_{roll}$ (deg/s)	0.0		
ψ_{l1} (deg)	0.0	$d\psi_{l1}$ (deg/s)	0.0		
ψ_{r1} (deg)	0.0	$d\psi_{r1}$ (deg/s)	0.0		
θ_{l5} (deg)	0.0	$d\theta_{l5}$ (deg/s)	0.5		
θ_{r5} (deg)	0.0	$d\theta_{r5}$ (deg/s)	0.0		

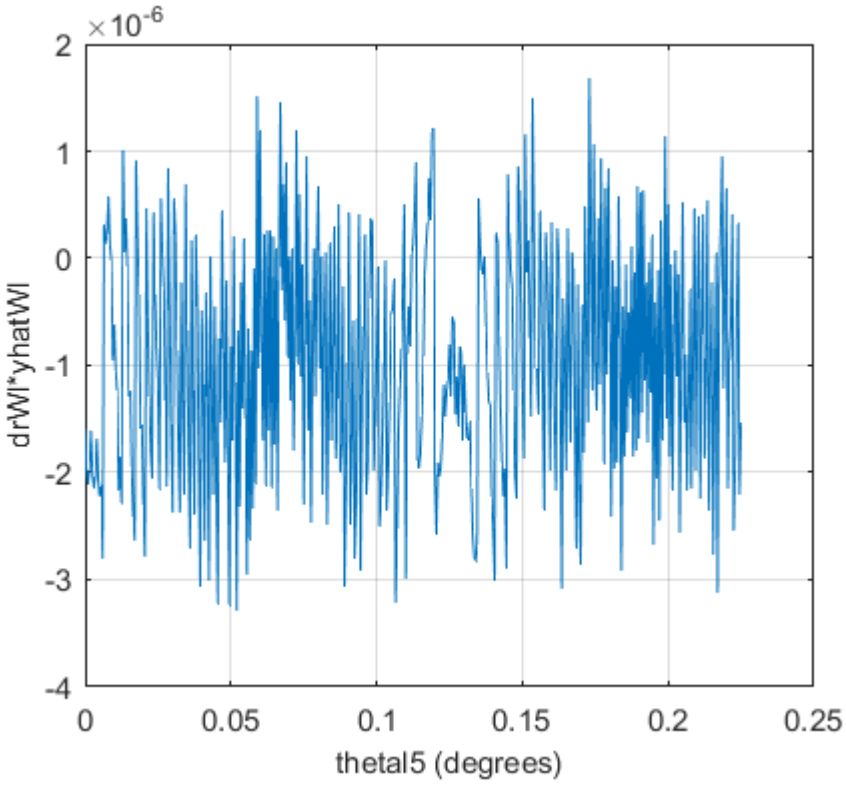

Figure 4-6: Verification of the constraint equation, $\dot{r}_{Wl}^{\ T} \cdot \hat{y}_{Wl} = 0$, with the MOBILE variable values for $\dot{\theta}_{l5} = 0.5\,\mathrm{deg}/s$.

We see from Figure 4-6 that the MOBILE variable values satisfy the constraint equation for the left wheel as 10^{-6}-10^{-5} values can be considered as zero.

3rd Verification test

For the third verification test we input to MOBILE a right wheel straddle angle, $\dot{\theta}_{r5} = 0.5\,\mathrm{deg}/s$. The chosen step, for acceptable accuracy, of the MOBILE integrator is 0.001 s, as before. Then we observe this scenario for 451 velocity integrations, that

means approximately for 0.451 s. At every integration (step in time), we store the values of all the variables involved in the constraint equation and then we input these values to this equation in MATLAB and check if they satisfy it, exactly like the previous tests. In Table 4-3 we can see the initial conditions and fixed values for the third verification test of the left wheel.

Table 4-3: Initial conditions and fixed values for the third verification test of the left wheel.

x_C (m)	0.0	dx_C (m/s)	0.0	cxd (m)	3.0
y_C (m)	0.0	-	-	cyd (m)	3.0
z_C (m)	2.0	dz_C (m/s)	0.0	w (m)	0.5
x_{Fl} (m)	3.5	dx_{Fl} (m/s)	0.0	wl (m)	2.5
y_{Fl} (m)	1.5	dy_{Fl} (m/s)	0.0	thighD (m)	2.0
z_{Fl} (m)	0.0	dz_{Fl} (m/s)	0.0	legD (m)	1.5
x_{Fr} (m)	3.5	dx_{Fr} (m/s)	0.0		
y_{Fr} (m)	-1.5	dy_{Fr} (m/s)	0.0		
z_{Fr} (m)	0.0	dz_{Fr} (m/s)	0.0		
z_{Wl} (m)	0.0	dz_{Wl} (m/s)	0.0		
z_{Wr} (m)	0.05	dz_{Wr} (m/s)	0.0		
φ_{yaw} (deg)	0.0	-	-		
φ_{pitch} (deg)	0.0	$d\varphi_{pitch}$ (deg/s)	0.0		
φ_{roll} (deg)	0.0	$d\varphi_{roll}$ (deg/s)	0.0		
ψ_{l1} (deg)	0.0	$d\psi_{l1}$ (deg/s)	0.0		
ψ_{r1} (deg)	0.0	$d\psi_{r1}$ (deg/s)	0.0		
θ_{l5} (deg)	0.0	$d\theta_{l5}$ (deg/s)	0.0		
θ_{r5} (deg)	0.0	$d\theta_{r5}$ (deg/s)	0.5		

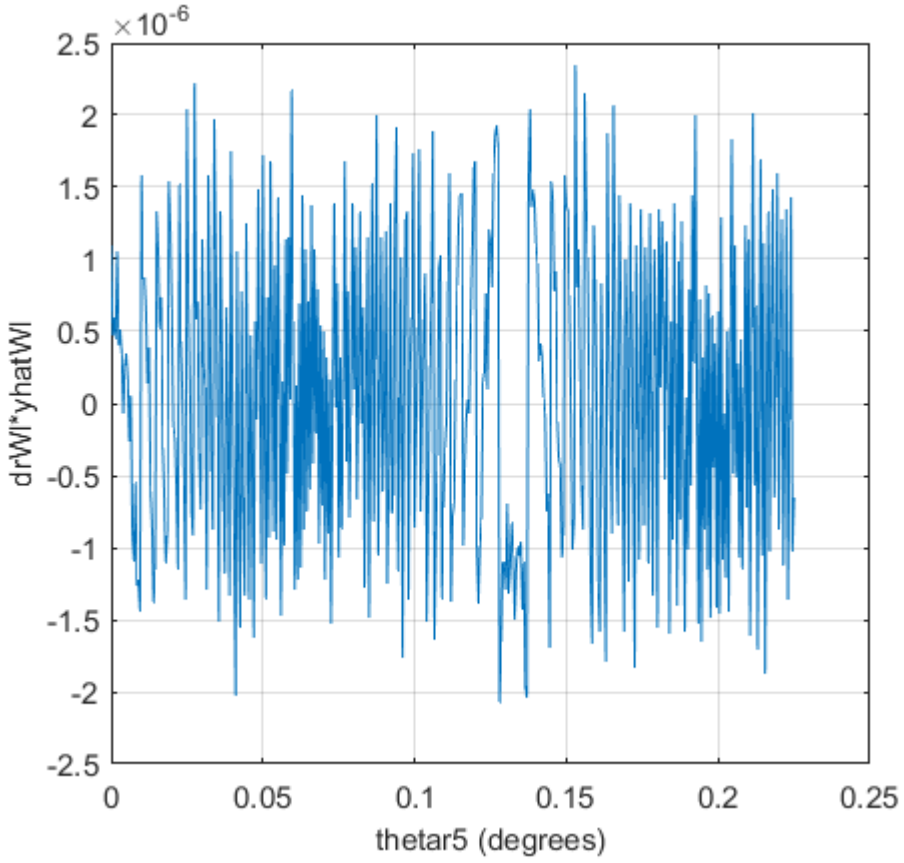

Figure 4-7: Verification of the constraint equation, $\dot{r}_{Wl}{}^{T} \cdot \hat{y}_{Wl} = 0$, with the MOBILE variable values for $\dot{\theta}_{r5} = 0.5\text{deg}/s$.

We see from Figure 4-7 that the MOBILE variable values satisfy the constraint equation for the left wheel as 10^{-6}-10^{-5} values can be considered as zero.

4.2.2 Right wheel verification

We have the same notations for the right wheel variables and distances names as described on the inverse kinematics chapter (see Figure 3-46) and also the same frames distribution (see Figure 3-45) and vehicle geometry. The equations that describe the right wheel's tip position with respect to the ground-fixed coordinate frame K_0 were derived in the inverse kinematics chapter and are the following:

$$\begin{bmatrix} x_{Wr} \\ y_{Wr} \\ z_{Wr} \end{bmatrix} = \begin{bmatrix} x_C + \left(c_{roll}s_{yaw} - c_{yaw}s_{pitch}s_{roll}\right)\left(\frac{cyd}{2} + s_{\theta_{r5}}\left(wl + c_{\theta_{r4}}s_{\theta_{r3}}w\right) + c_{\theta_{r5}}s_{\theta_{r3}}s_{\theta_{r4}}w\right) - \\ \left(s_{roll}s_{yaw} + c_{roll}c_{yaw}s_{pitch}\right)\left(c_{\theta_{r3}}c_{\theta_{r6}}w - c_{\theta_{r5}}s_{\theta_{r6}}\left(wl + c_{\theta_{r4}}s_{\theta_{r3}}w\right) + s_{\theta_{r3}}s_{\theta_{r4}}s_{\theta_{r5}}s_{\theta_{r6}}w\right) - \\ c_{pitch}c_{yaw}\left(\frac{cxd}{2} + c_{\theta_{r5}}c_{\theta_{r6}}\left(wl + c_{\theta_{r4}}s_{\theta_{r3}}w\right) + c_{\theta_{r3}}s_{\theta_{r6}}w - c_{\theta_{r6}}s_{\theta_{r3}}s_{\theta_{r4}}s_{\theta_{r5}}w\right) \\ y_C - \left(c_{roll}c_{yaw} + s_{yaw}s_{pitch}s_{roll}\right)\left(\frac{cyd}{2} + s_{\theta_{r5}}\left(wl + c_{\theta_{r4}}s_{\theta_{r3}}w\right) + c_{\theta_{r5}}s_{\theta_{r3}}s_{\theta_{r4}}w\right) + \\ \left(s_{roll}c_{yaw} - c_{roll}s_{yaw}s_{pitch}\right)\left(c_{\theta_{r3}}c_{\theta_{r6}}w - c_{\theta_{r5}}s_{\theta_{r6}}\left(wl + c_{\theta_{r4}}s_{\theta_{r3}}w\right) + s_{\theta_{r3}}s_{\theta_{r4}}s_{\theta_{r5}}s_{\theta_{r6}}w\right) - \\ c_{pitch}s_{yaw}\left(\frac{cxd}{2} + c_{\theta_{r5}}c_{\theta_{r6}}\left(wl + c_{\theta_{r4}}s_{\theta_{r3}}w\right) + c_{\theta_{r3}}s_{\theta_{r6}}w - c_{\theta_{r6}}s_{\theta_{r3}}s_{\theta_{r4}}s_{\theta_{r5}}w\right) \\ z_C + s_{pitch}\left(\frac{cxd}{2} + c_{\theta_{r5}}c_{\theta_{r6}}\left(wl + c_{\theta_{r4}}s_{\theta_{r3}}w\right) + c_{\theta_{r3}}s_{\theta_{r6}}w - c_{\theta_{r6}}s_{\theta_{r3}}s_{\theta_{r4}}s_{\theta_{r5}}w\right) - \\ c_{roll}c_{pitch}\left(c_{\theta_{r3}}c_{\theta_{r6}}w - c_{\theta_{r5}}s_{\theta_{r6}}\left(wl + c_{\theta_{r4}}s_{\theta_{r3}}w\right) + s_{\theta_{r3}}s_{\theta_{r4}}s_{\theta_{r5}}s_{\theta_{r6}}w\right) - \\ c_{pitch}s_{roll}\left(\frac{cyd}{2} + s_{\theta_{r5}}\left(wl + c_{\theta_{r4}}s_{\theta_{r3}}w\right) + c_{\theta_{r5}}s_{\theta_{r3}}s_{\theta_{r4}}w\right) \end{bmatrix} \quad (4.19)$$

By deriving Equations (4.19), we get the velocity of the right wheel end frame with respect to the ground-fixed coordinate frame K_0:

$$\dot{r}_{Wr} = \begin{bmatrix} \dot{x}_{Wr} \\ \dot{y}_{Wr} \\ \dot{z}_{Wr} \end{bmatrix} \quad (4.20)$$

For the velocity constraint to be satisfied, the y component of velocity on the wheel tip-fixed coordinate frame (K_{Wr}) must be zero. To calculate that velocity we multiply $\dot{r}_{Wr}$ with the orientation of the y unit vector of K_{Wr}, $\hat{y}_{Wr}$ (see Figure 4-8), with respect to the ground-fixed frame K_0. We get that orientation vector from the second column of the rotation matrix ${}^0R_{12}$ that connects K_0 and K_{12}(=K_{Wr}) and has been already calculated on the inverse kinematics chapter.

$$\begin{aligned} {}^0R_{12} &= {}^0R_4 \cdot {}^4R_7 \cdot {}^7R_{10} \cdot {}^{10}R_{12} = \\ &= rot_z(\varphi_{yaw}) \cdot rot_y(\varphi_{pitch}) \cdot rot_x(\varphi_{roll}) \cdot rot_y(\theta_{r6}) \cdot rot_z(\theta_{r5}) \cdot rot_z(\theta_{r4}) \cdot rot_y(\theta_{r3}) \cdot rot_x\left(\theta_{r2}\right) \end{aligned} \quad (4.21)$$

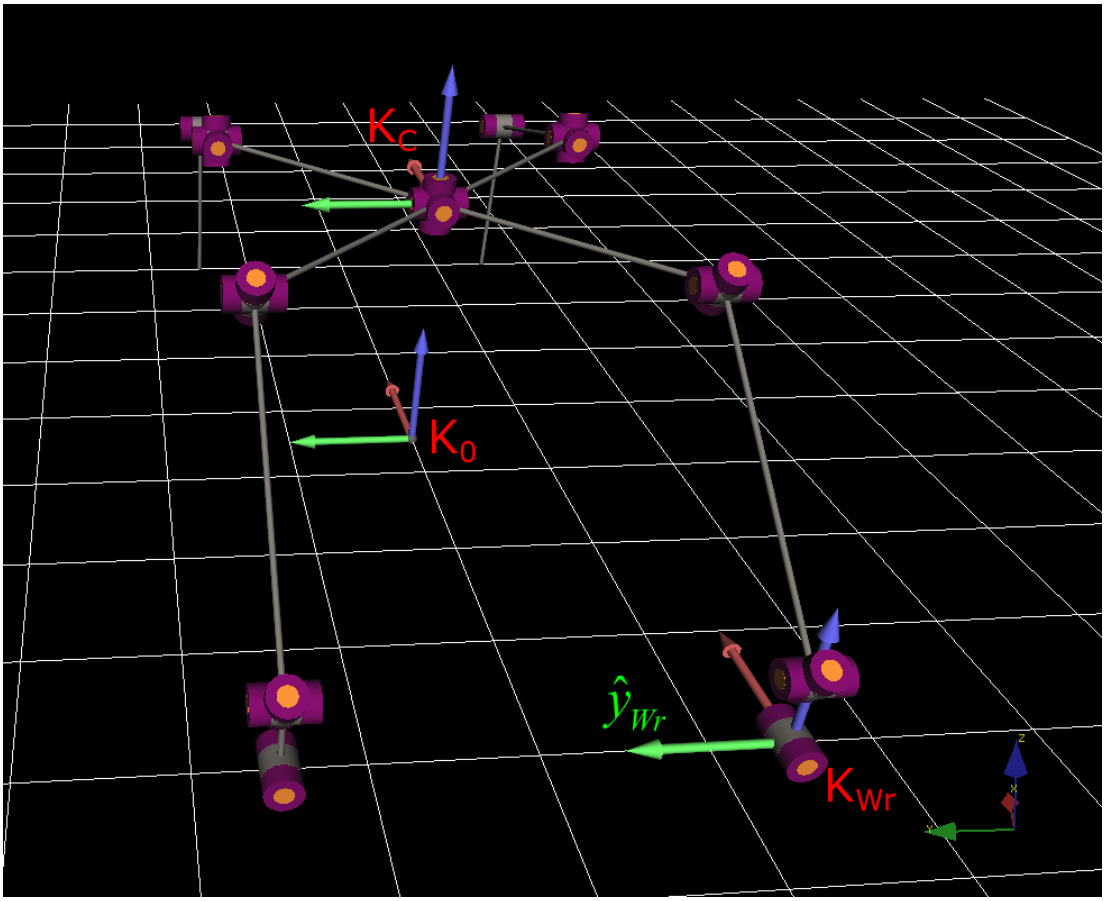

Figure 4-8: The y unit vector of K_{wr}, y_{wr}. The wheel's velocity must be zero in that direction for the nonholonomic constraint to be satisfied.

$$\hat{y}_{Wr} = \begin{bmatrix} -S_{\theta_{r2}}\left(\begin{array}{l} C_{\theta_{r4}}S_{\theta_{r3}}\left(\begin{array}{l} S_{\theta_{r5}}\left(\begin{array}{l} C_{roll}S_{yaw} \\ -C_{yaw}S_{pitch}S_{roll} \end{array}\right) \\ +C_{\theta_{r5}}S_{\theta_{r6}}\left(\begin{array}{l} S_{roll}S_{yaw} \\ +C_{roll}C_{yaw}S_{pitch} \end{array}\right) \\ -C_{pitch}C_{\theta_{r5}}C_{\theta_{r6}}C_{yaw} \end{array}\right) - C_{\theta_{r3}}\left(\begin{array}{l} C_{\theta_{r6}}\left(\begin{array}{l} S_{roll}S_{yaw} \\ +C_{roll}C_{yaw}S_{pitch} \end{array}\right) \\ +C_{pitch}C_{yaw}S_{\theta_{r6}} \end{array}\right) \\ +S_{\theta_{r3}}S_{\theta_{r4}}\left(\begin{array}{l} C_{\theta_{r5}}\left(\begin{array}{l} C_{roll}S_{yaw} \\ -C_{yaw}S_{pitch}S_{roll} \end{array}\right) \\ -S_{\theta_{r5}}S_{\theta_{r6}}\left(\begin{array}{l} S_{roll}S_{yaw} \\ +C_{roll}C_{yaw}S_{pitch} \end{array}\right) \\ +C_{pitch}C_{\theta_{r6}}C_{yaw}S_{\theta_{r5}} \end{array}\right) \end{array}\right) \\ -C_{\theta_{r2}}\left(\begin{array}{l} C_{\theta_{r4}}\left(\begin{array}{l} C_{\theta_{r5}}\left(\begin{array}{l} C_{roll}S_{yaw} \\ -C_{yaw}S_{pitch}S_{roll} \end{array}\right) \\ -S_{\theta_{r5}}S_{\theta_{r6}}\left(\begin{array}{l} S_{roll}S_{yaw} \\ +C_{roll}C_{yaw}S_{pitch} \end{array}\right) \\ +C_{pitch}C_{\theta_{r6}}C_{yaw}S_{\theta_{r5}} \end{array}\right) - S_{\theta_{r4}}\left(\begin{array}{l} S_{\theta_{r5}}\left(\begin{array}{l} C_{roll}S_{yaw} \\ -C_{yaw}S_{pitch}S_{roll} \end{array}\right) + \\ C_{\theta_{r5}}S_{\theta_{r6}}\left(\begin{array}{l} S_{roll}S_{yaw} \\ +C_{roll}C_{yaw}S_{pitch} \end{array}\right) - \\ C_{pitch}C_{\theta_{r5}}C_{\theta_{r6}}C_{yaw} \end{array}\right) \end{array}\right) \\ -S_{\theta_{r2}}\left(\begin{array}{l} C_{\theta_{r3}}\left(\begin{array}{l} C_{\theta_{r6}}\left(\begin{array}{l} C_{yaw}S_{roll} \\ -C_{roll}S_{pitch}S_{yaw} \end{array}\right) \\ -C_{pitch}S_{\theta_{r6}}S_{yaw} \end{array}\right) - C_{\theta_{r4}}S_{\theta_{r3}}\left(\begin{array}{l} S_{\theta_{r5}}\left(\begin{array}{l} C_{roll}C_{yaw} \\ +S_{roll}S_{yaw}S_{pitch} \end{array}\right) \\ +C_{\theta_{r5}}S_{\theta_{r6}}\left(\begin{array}{l} C_{yaw}S_{roll} \\ -C_{roll}S_{pitch}S_{yaw} \end{array}\right) + \\ +C_{pitch}C_{\theta_{r5}}C_{\theta_{r6}}S_{yaw} \end{array}\right) + \\ S_{\theta_{r3}}S_{\theta_{r4}}\left(\begin{array}{l} S_{\theta_{r5}}S_{\theta_{r6}}\left(\begin{array}{l} C_{yaw}S_{roll} \\ -C_{roll}S_{pitch}S_{yaw} \end{array}\right) \\ -C_{\theta_{r5}}\left(\begin{array}{l} C_{roll}C_{yaw} \\ +S_{roll}S_{yaw}S_{pitch} \end{array}\right) \\ +C_{pitch}C_{\theta_{r6}}S_{yaw}S_{\theta_{r5}} \end{array}\right) \end{array}\right) \\ -C_{\theta_{r2}}\left(\begin{array}{l} S_{\theta_{r4}}\left(\begin{array}{l} S_{\theta_{r5}}\left(\begin{array}{l} C_{roll}C_{yaw} \\ +S_{yaw}S_{pitch}S_{roll} \end{array}\right) \\ +C_{\theta_{r5}}C_{\theta_{r6}}\left(\begin{array}{l} S_{roll}C_{yaw} \\ -C_{roll}S_{yaw}S_{pitch} \end{array}\right) \\ +C_{pitch}C_{\theta_{r5}}S_{yaw}C_{\theta_{r6}} \end{array}\right) + C_{\theta_{r4}}\left(\begin{array}{l} S_{\theta_{r5}}S_{\theta_{r6}}\left(\begin{array}{l} C_{yaw}S_{roll} \\ -C_{roll}S_{pitch}S_{yaw} \end{array}\right) \\ -C_{\theta_{r5}}\left(\begin{array}{l} C_{roll}C_{yaw} \\ +S_{roll}S_{yaw}S_{pitch} \end{array}\right) \\ +C_{pitch}S_{\theta_{r5}}C_{\theta_{r6}}S_{yaw} \end{array}\right) \end{array}\right) \\ C_{\theta_{r2}}\left(\begin{array}{l} C_{\theta_{r4}}\left(\begin{array}{l} C_{pitch}C_{\theta_{r5}}S_{roll} \\ +C_{\theta_{r6}}S_{pitch}S_{\theta_{r5}} \\ +C_{pitch}C_{roll}S_{\theta_{r5}}S_{\theta_{r6}} \end{array}\right) + S_{\theta_{r5}}\left(\begin{array}{l} C_{\theta_{r5}}C_{\theta_{r6}}S_{pitch} \\ -C_{pitch}S_{roll}S_{\theta_{r5}} \\ +C_{pitch}C_{roll}C_{\theta_{r5}}S_{\theta_{r6}} \end{array}\right) \end{array}\right) - \\ S_{\theta_{r2}}\left(\begin{array}{l} C_{\theta_{r3}}\left(\begin{array}{l} S_{pitch}S_{\theta_{r6}} \\ -C_{pitch}C_{roll}C_{\theta_{r6}} \end{array}\right) + C_{\theta_{r4}}S_{\theta_{r3}}\left(\begin{array}{l} C_{\theta_{r5}}C_{\theta_{r6}}S_{pitch} \\ -C_{pitch}S_{roll}S_{\theta_{r5}} \\ +C_{pitch}C_{roll}C_{\theta_{r5}}S_{\theta_{r6}} \end{array}\right) - S_{\theta_{r3}}S_{\theta_{r4}}\left(\begin{array}{l} C_{pitch}C_{\theta_{r5}}S_{roll} \\ +C_{\theta_{r6}}S_{pitch}S_{\theta_{r5}} \\ +C_{pitch}C_{roll}S_{\theta_{r5}}S_{\theta_{r6}} \end{array}\right) \end{array}\right) \end{bmatrix} \tag{4.22}$$

Thus, the constraint we must test if it is satisfied by MOBILE is:

$$\dot{r}_{Wr}^{\ T} \cdot \hat{y}_{Wr} = 0 \tag{4.23}$$

Using Equation (4.23), we run some verification tests in order to ensure that the velocity values that MOBILE calculates for the dependent variables, with the procedure explained in the previous chapter, comply with the nonholonomic constraints. For the verification tests we input to MOBILE a velocity in variables that affect the cornering nonholonomic constraints the most (like on the left wheel these are the cabin roll angle and the left and right wheel straddle angle), then we read and store all the values involved in the above constraint equation for a period of time and then we check in MATLAB if the stored values satisfy the constraint, exactly like the left wheel.

1st Verification test

For the first verification test we input to MOBILE a cabin roll velocity, $\dot{\varphi}_{roll} = 5\,\mathrm{deg}/s$. The chosen step, for acceptable accuracy, of the MOBILE integrator is 0.001 s. Then we observe this scenario for 2001 velocity integrations, that means approximately for 2 s. At every integration (step in time), we store the values of all the variables involved in the constraint equation and then we input these values to this equation in MATLAB and check if they satisfy it. In Table 4-4 we can see the initial conditions and fixed values for the first verification test of the right wheel.

Table 4-4: Initial conditions and fixed values for the first verification test of the right wheel.

x_C (m)	0.0	dx_C (m/s)	0.0	cxd (m)	3.0
y_C (m)	0.0	-	-	cyd (m)	3.0
z_C (m)	2.0	dz_C (m/s)	0.0	w (m)	0.5
x_{Fl} (m)	3.5	dx_{Fl} (m/s)	0.0	wl (m)	2.5
y_{Fl} (m)	1.5	dy_{Fl} (m/s)	0.0	thighD (m)	2.0
z_{Fl} (m)	0.0	dz_{Fl} (m/s)	0.0	legD (m)	1.5
x_{Fr} (m)	3.5	dx_{Fr} (m/s)	0.0		
y_{Fr} (m)	-1.5	dy_{Fr} (m/s)	0.0		
z_{Fr} (m)	0.0	dz_{Fr} (m/s)	0.0		
z_{Wl} (m)	0.0	dz_{Wl} (m/s)	0.0		
z_{Wr} (m)	0.05	dz_{Wr} (m/s)	0.0		
φ_{yaw} (deg)	0.0	-	-		
φ_{pitch} (deg)	0.0	$d\varphi_{pitch}$ (deg/s)	0.0		
φ_{roll} (deg)	0.0	$d\varphi_{roll}$ (deg/s)	5.0		
ψ_{l1} (deg)	0.0	$d\psi_{l1}$ (deg/s)	0.0		
ψ_{r1} (deg)	0.0	$d\psi_{r1}$ (deg/s)	0.0		
θ_{l5} (deg)	0.0	$d\theta_{l5}$ (deg/s)	0.0		
θ_{r5} (deg)	0.0	$d\theta_{r5}$ (deg/s)	0.0		

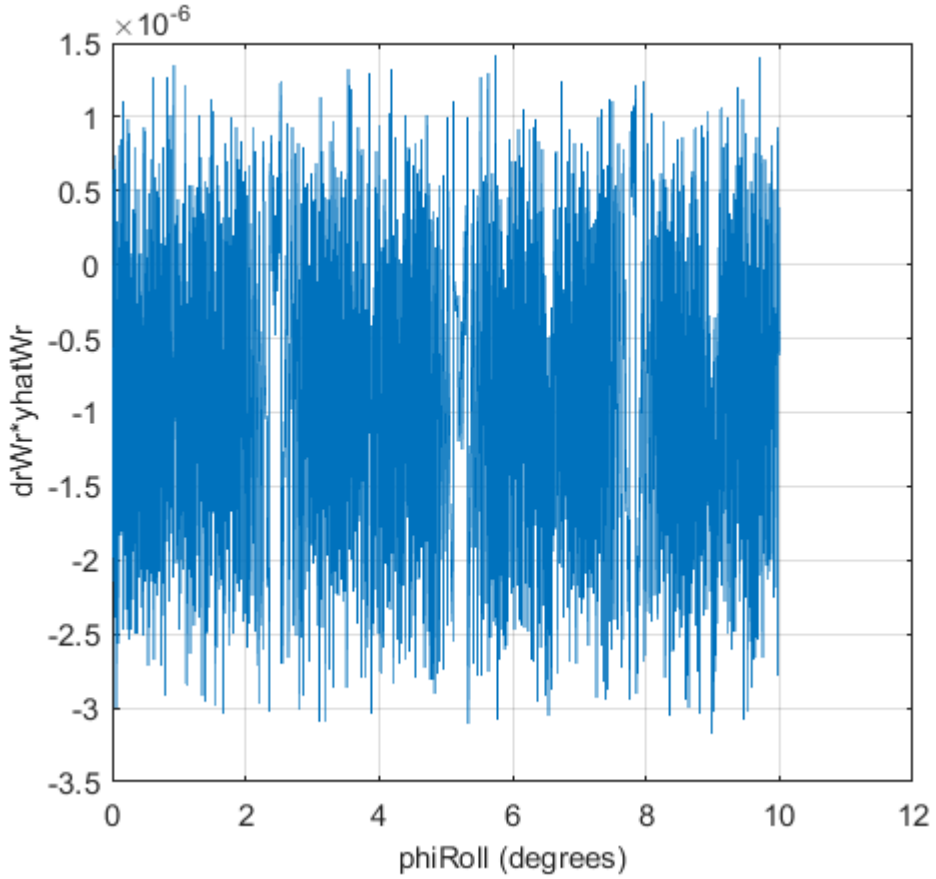

Figure 4-9: Verification of the constraint equation, $\dot{r}_{Wr}{}^{T} \cdot \hat{y}_{Wr} = 0$, with the MOBILE variable values for $\dot{\varphi}_{roll} = 5\,\mathrm{deg}/s$.

We see from Figure 4-9 that the MOBILE variable values satisfy the constraint equation for the right wheel as 10^{-6}-10^{-5} values can be considered as zero.

2nd Verification test

For the second verification test we input to MOBILE a left wheel straddle angle, $\dot{\theta}_{l5} = 0.5\,\mathrm{deg}/s$. The chosen step, for acceptable accuracy, of the MOBILE integrator is 0.001 s, as before. Then we observe this scenario for 451 velocity integrations, that means approximately for 0.451 s. At every integration (step in time), we store the values of all the variables involved in the constraint equation and then we input these values to this equation in MATLAB and check if they satisfy it, exactly like the first test. In Table 4-5 we can see the initial conditions and fixed values for the second verification test of the right wheel.

Table 4-5: Initial conditions and fixed values for the second verification test of the right wheel.

x_C (m)	0.0	dx_C (m/s)	0.0	cxd (m)	3.0
y_C (m)	0.0	-	-	cyd (m)	3.0
z_C (m)	2.0	dz_C (m/s)	0.0	w (m)	0.5

x_{Fl} (m)	3.5	dx_{Fl} (m/s)	0.0	wl (m)	2.5
y_{Fl} (m)	1.5	dy_{Fl} (m/s)	0.0	thighD (m)	2.0
z_{Fl} (m)	0.0	dz_{Fl} (m/s)	0.0	legD (m)	1.5
x_{Fr} (m)	3.5	dx_{Fr} (m/s)	0.0		
y_{Fr} (m)	-1.5	dy_{Fr} (m/s)	0.0		
z_{Fr} (m)	0.0	dz_{Fr} (m/s)	0.0		
z_{Wl} (m)	0.0	dz_{Wl} (m/s)	0.0		
z_{Wr} (m)	0.05	dz_{Wr} (m/s)	0.0		
φ_{yaw} (deg)	0.0	-	-		
φ_{pitch} (deg)	0.0	$d\varphi_{pitch}$ (deg/s)	0.0		
φ_{roll} (deg)	0.0	$d\varphi_{roll}$ (deg/s)	0.0		
ψ_{l1} (deg)	0.0	$d\psi_{l1}$ (deg/s)	0.0		
ψ_{r1} (deg)	0.0	$d\psi_{r1}$ (deg/s)	0.0		
θ_{l5} (deg)	0.0	$d\theta_{l5}$ (deg/s)	0.5		
θ_{r5} (deg)	0.0	$d\theta_{r5}$ (deg/s)	0.0		

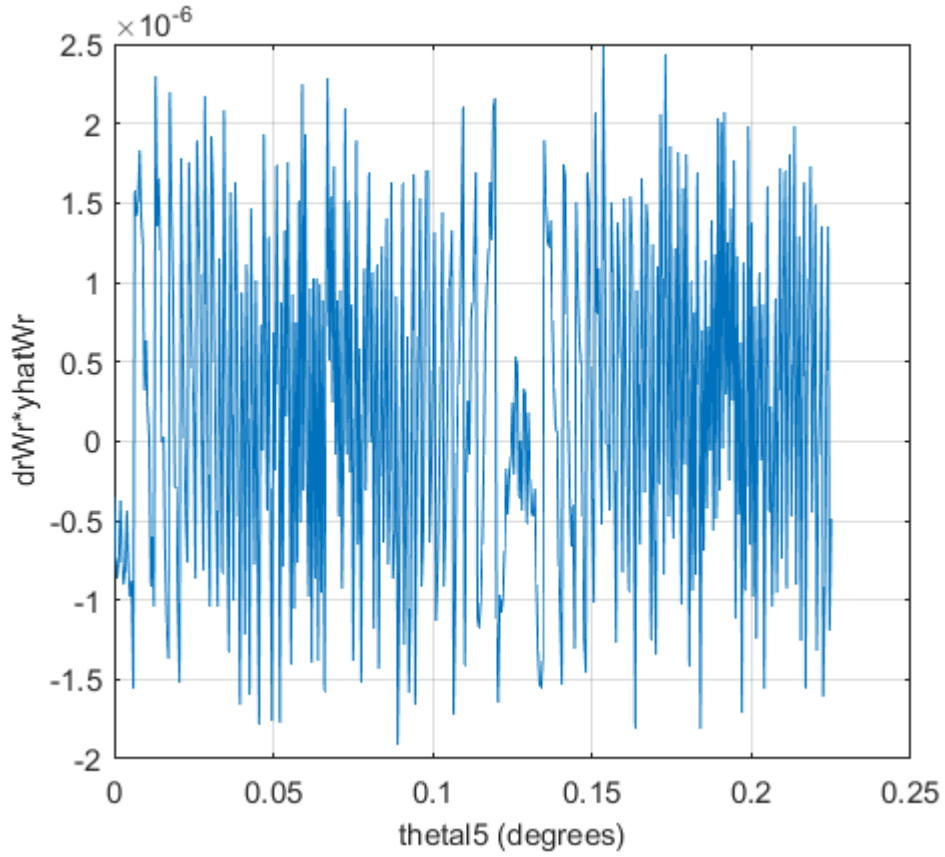

Figure 4-10: Verification of the constraint equation, $\dot{r}_{Wr}^{\ T} \cdot \hat{y}_{Wr} = 0$, with the MOBILE variable values for $\dot{\theta}_{l_5} = 0.5 \deg/s$.

We see from Figure 4-10 that the MOBILE variable values satisfy the constraint equation for the right wheel as 10^{-6}-10^{-5} values can be considered as zero.

3rd Verification test

For the third verification test we input to MOBILE a right wheel straddle angle, $\dot{\theta}_{r5} = 0.5\,\mathrm{deg}/s$. The chosen step, for acceptable accuracy, of the MOBILE integrator is 0.001 s, as before. Then we observe this scenario for 451 velocity integrations, that means approximately for 0.451 s. At every integration (step in time), we store the values of all the variables involved in the constraint equation and then we input these values to this equation in MATLAB and check if they satisfy it, exactly like the previous tests. In Table 4-6 we can see the initial conditions and fixed values for the third verification test of the right wheel.

Table 4-6: Initial conditions and fixed values for the third verification test of the right wheel.

x_C (m)	0.0	dx_C (m/s)	0.0	cxd (m)	3.0
y_C (m)	0.0	-	-	cyd (m)	3.0
z_C (m)	2.0	dz_C (m/s)	0.0	w (m)	0.5
x_{Fl} (m)	3.5	dx_{Fl} (m/s)	0.0	wl (m)	2.5
y_{Fl} (m)	1.5	dy_{Fl} (m/s)	0.0	thighD (m)	2.0
z_{Fl} (m)	0.0	dz_{Fl} (m/s)	0.0	legD (m)	1.5
x_{Fr} (m)	3.5	dx_{Fr} (m/s)	0.0		
y_{Fr} (m)	-1.5	dy_{Fr} (m/s)	0.0		
z_{Fr} (m)	0.0	dz_{Fr} (m/s)	0.0		
z_{Wl} (m)	0.0	dz_{Wl} (m/s)	0.0		
z_{Wr} (m)	0.05	dz_{Wr} (m/s)	0.0		
φ_{yaw} (deg)	0.0	-	-		
φ_{pitch} (deg)	0.0	$d\varphi_{pitch}$ (deg/s)	0.0		
φ_{roll} (deg)	0.0	$d\varphi_{roll}$ (deg/s)	0.0		
ψ_{l1} (deg)	0.0	$d\psi_{l1}$ (deg/s)	0.0		
ψ_{r1} (deg)	0.0	$d\psi_{r1}$ (deg/s)	0.0		
θ_{l5} (deg)	0.0	$d\theta_{l5}$ (deg/s)	0.0		
θ_{r5} (deg)	0.0	$d\theta_{r5}$ (deg/s)	0.5		

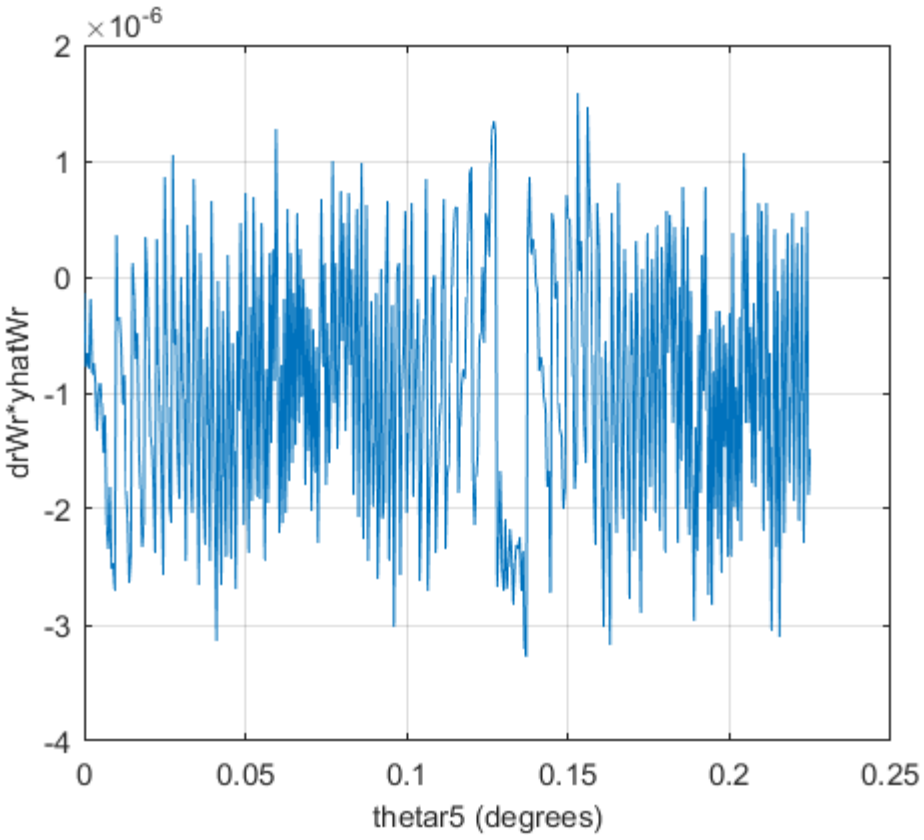

Figure 4-11: Verification of the constraint equation, $\dot{r}_{Wr}^{\ T} \cdot \hat{y}_{Wr} = 0$, with the MOBILE variable values for $\dot{\theta}_{r_5} = 0.5\,\text{deg}/\,s$.

We see from Figure 4-11 that the MOBILE variable values satisfy the constraint equation for the right wheel as 10^{-6}-10^{-5} values can be considered as zero.

4.2.3 Complete verification test for both wheels simultaneously

For the complete verification test we input to MOBILE a right wheel straddle angle velocity, $\dot{\theta}_{r5} = -0.5\,\text{deg}/\,s$, a left wheel straddle angle velocity, $\dot{\theta}_{l5} = 0.5\,\text{deg}/\,s$ and a cabin roll velocity, $\dot{\varphi}_{roll} = 10.0\,\text{deg}/\,s$. The chosen step, for acceptable accuracy, of the MOBILE integrator is 0.001 s, as before. Then we observe this scenario for 2001 velocity integrations, that means approximately for 2 s. At every integration (step in time), we store the values of all the variables involved in the constraint equation and then we input these values to this equation in MATLAB and check if they satisfy it, exactly like the previous tests. In Table 4-7 we can see the initial conditions and fixed values for the complete verification test of the non-holonomic constraints for both wheels.

Table 4-7: Initial conditions and fixed values for the complete verification test of both wheels.

x_C (m)	0.0	dx_C (m/s)	0.0	cxd (m)	3.0
y_C (m)	0.0	-	-	cyd (m)	3.0
z_C (m)	2.0	dz_C (m/s)	0.0	w (m)	0.5

x_{Fl} (m)	3.5	dx_{Fl} (m/s)	0.0	wl (m)	2.5
y_{Fl} (m)	1.5	dy_{Fl} (m/s)	0.0	thighD (m)	2.0
z_{Fl} (m)	0.0	dz_{Fl} (m/s)	0.0	legD (m)	1.5
x_{Fr} (m)	3.5	dx_{Fr} (m/s)	0.0		
y_{Fr} (m)	-1.5	dy_{Fr} (m/s)	0.0		
z_{Fr} (m)	0.0	dz_{Fr} (m/s)	0.0		
z_{Wl} (m)	0.0	dz_{Wl} (m/s)	0.0		
z_{Wr} (m)	0.05	dz_{Wr} (m/s)	0.0		
φ_{yaw} (deg)	0.0	-	-		
φ_{pitch} (deg)	0.0	$d\varphi_{pitch}$ (deg/s)	0.0		
φ_{roll} (deg)	0.0	$d\varphi_{roll}$ (deg/s)	10.0		
ψ_{l1} (deg)	0.0	$d\psi_{l1}$ (deg/s)	0.0		
ψ_{r1} (deg)	0.0	$d\psi_{r1}$ (deg/s)	0.0		
θ_{l5} (deg)	0.0	$d\theta_{l5}$ (deg/s)	0.5		
θ_{r5} (deg)	0.0	$d\theta_{r5}$ (deg/s)	-0.5		

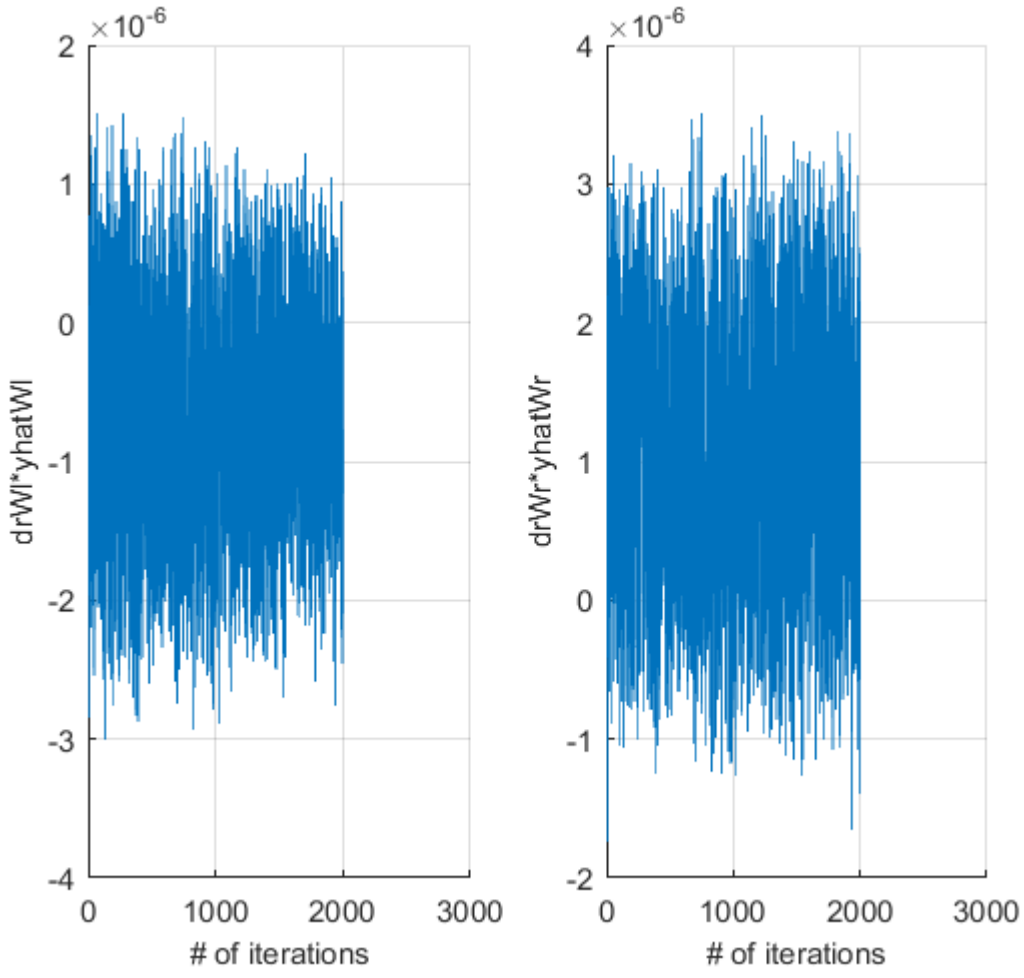

Figure 4-12: Verification of the constraint equations, $\dot{r}_{Wl}{}^{T} \cdot \hat{y}_{Wl} = 0$ (left) and $\dot{r}_{Wr}{}^{T} \cdot \hat{y}_{Wr} = 0$ (right) with the MOBILE variable values for the complete verification test.

We see from Figure 4-12 that the MOBILE variable values satisfy the constraint equation for both wheels as 10^{-6}-10^{-5} values can be considered as zero.

We observe that the MOBILE dependent velocity values are satisfying the nonholonomic constraints, as they verify the analytical constraint equations. Hence, we can conclude that MOBILE's iterative method for the satisfaction of the non-holonomic constraints is acceptably accurate, except from easier to apply than an analytical way.

5 Steering behaviour

In this chapter, the steering behaviour of the excavator is analysed as a result of the non-holonomic constraints that characterize the mechanism. As stated in Chapter 4, Equation (4.6), the cornering non-holonomic constraints of the system affect the next subset of input velocities:

$$\dot{q}_I = [\dot{x}_C, \dot{y}_C, \dot{z}_C, \dot{\varphi}_{yaw}, \dot{\varphi}_{pitch}, \dot{\varphi}_{roll}, \dot{\theta}_{l5}, \dot{\theta}_{r5}]^T$$

Out of these velocities, six are controllable and two are dependent because of the two wheels that insert the constraints. Throughout the analysis until now $\dot{y}_C, \dot{\varphi}_{yaw}$ where considered to be the dependent velocities. Depending on which two variables we consider to be dependent, different relationships arise between the above velocities.

In this chapter two characteristic choices for the dependent velocities are going to be checked in terms of the different steering abilities they give to the mechanism or how the cabin yaw angle can be influenced.

5.1 For dependent variables $\dot{y}_C$ and $\dot{\varphi}_{yaw}$

For this choice of dependent variables only φ_{roll}, θ_{l5} and θ_{r5} can influence the cabin yaw angle φ_{yaw} by bringing the non-holonomic constraints to the test and these are going to be examined.

5.1.1 Steering influence of cabin roll angle φ_{roll}

In order to examine the steering influence of the cabin roll angle φ_{roll}, we input to MOBILE a velocity, $\dot{\varphi}_{roll} = 5.0\,\mathrm{deg}/s$ and then a velocity $\dot{\varphi}_{roll} = -5.0\,\mathrm{deg}/s$. The chosen step, for acceptable accuracy, of the MOBILE integrator is 0.001 s, as before. Then we observe these scenarios for 2001 velocity integrations, that means approximately for 2 s. At every integration (step in time), we store the values of φ_{roll}, y_C and φ_{yaw} for each velocity and express their relationship in the following diagrams. In Table 5-1 we can see the initial conditions and fixed values for the cabin roll angle steering test for both velocities.

Table 5-1: Initial conditions and fixed values for the steering influence of cabin roll simulation, for dependent variables $\dot{y}_C$ and $\dot{\varphi}_{yaw}$.

x_C (m)	0.0	dx_C (m/s)	0.0	cxd (m)	3.0
y_C (m)	0.0	-	-	cyd (m)	3.0
z_C (m)	2.0	dz_C (m/s)	0.0	w (m)	0.5
x_{Fl} (m)	3.5	dx_{Fl} (m/s)	0.0	wl (m)	2.5
y_{Fl} (m)	1.5	dy_{Fl} (m/s)	0.0	thighD (m)	2.0
z_{Fl} (m)	0.0	dz_{Fl} (m/s)	0.0	legD (m)	1.5
x_{Fr} (m)	3.5	dx_{Fr} (m/s)	0.0		

y_{Fr} (m)	-1.5	dy_{Fr} (m/s)	0.0		
z_{Fr} (m)	0.0	dz_{Fr} (m/s)	0.0		
z_{Wl} (m)	0.0	dz_{Wl} (m/s)	0.0		
z_{Wr} (m)	0.05	dz_{Wr} (m/s)	0.0		
φ_{yaw} (deg)	0.0	-	-		
φ_{pitch} (deg)	0.0	$d\varphi_{pitch}$ (deg/s)	0.0		
φ_{roll} (deg)	0.0	$d\varphi_{roll}$ (deg/s)	5.0/-5.0		
ψ_{l1} (deg)	0.0	$d\psi_{l1}$ (deg/s)	0.0		
ψ_{r1} (deg)	0.0	$d\psi_{r1}$ (deg/s)	0.0		
θ_{l5} (deg)	0.0	$d\theta_{l5}$ (deg/s)	0.0		
θ_{r5} (deg)	0.0	$d\theta_{r5}$ (deg/s)	0.0		

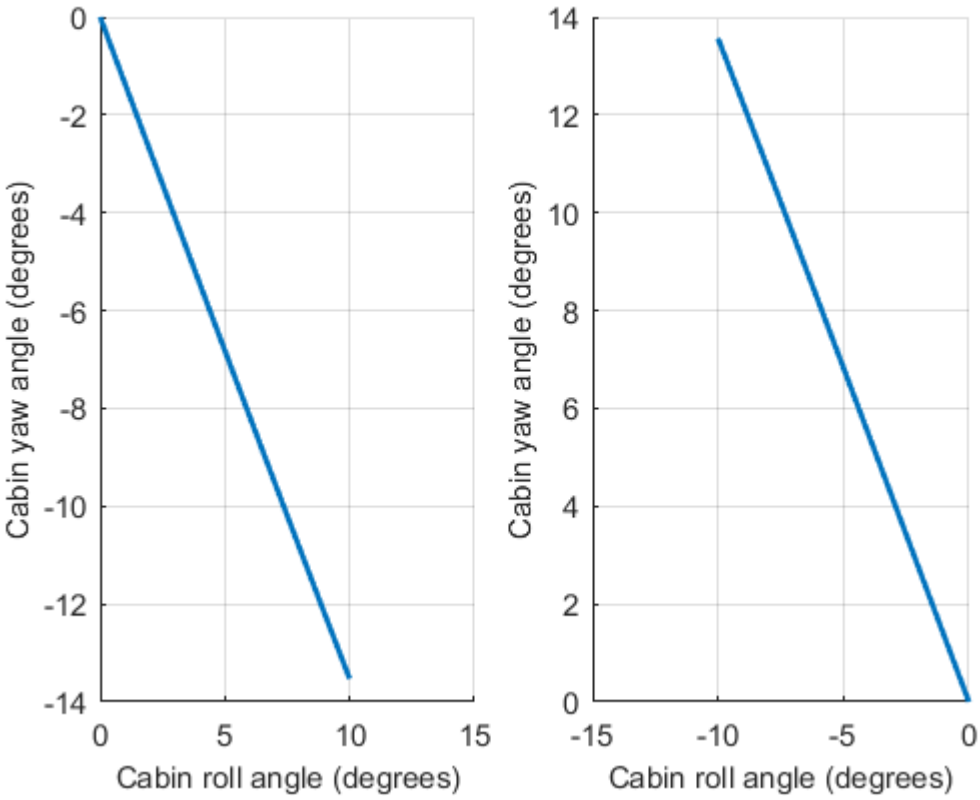

Figure 5-1: Effect of cabin roll angle to cabin yaw angle for positive (left) and negative (right) roll values, for dependent variables ẏC and φ̇yaw.

We observe that as we increase the cabin roll angle, the yaw angle and y distance of the cabin with respect to K0 decrease. The exact opposite happens as we decrease the cabin roll angle. This means that the vehicle turns towards the roll direction and also moves along the y axis towards the roll direction. That is a very interesting behaviour that looks similar to the steering of an airplane.

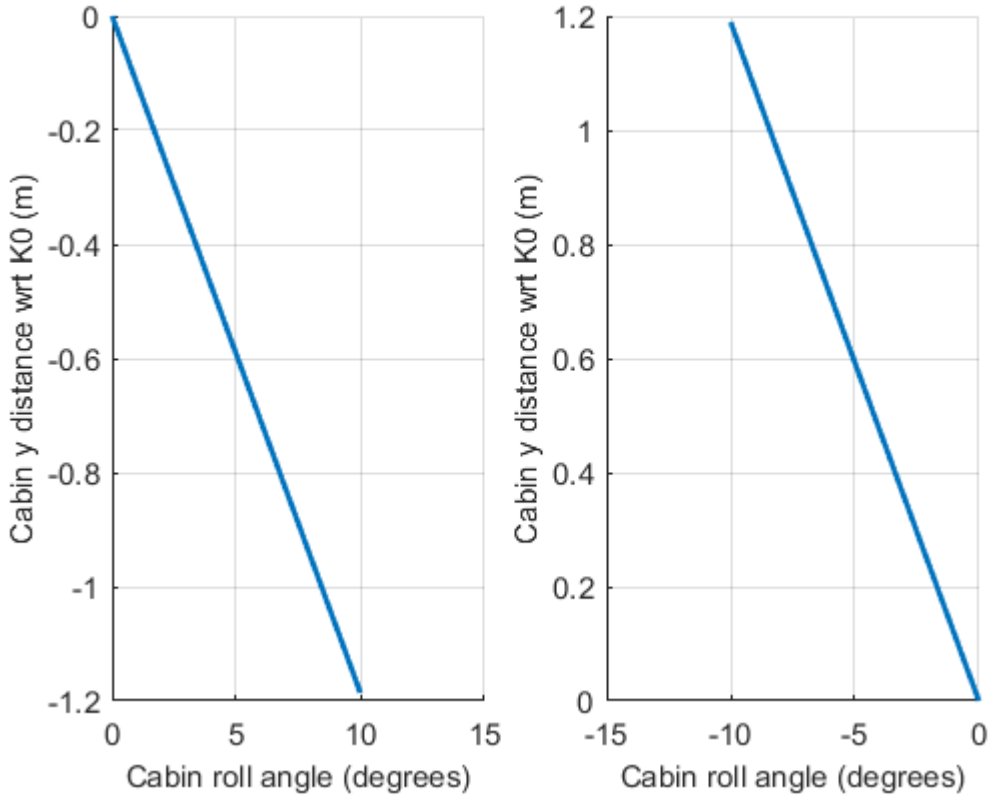

Figure 5-2: Effect of cabin roll angle to cabin y distance with respect to K0 for positive (left) and negative (right) roll values, for dependent variables ẏC and φ̇yaw.

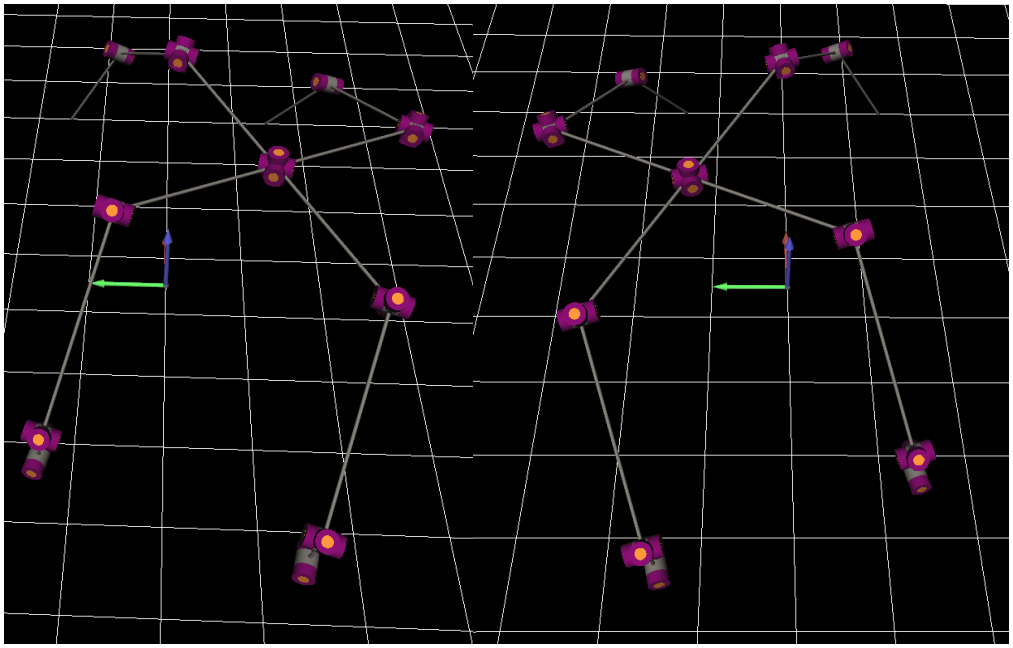

Figure 5-3: Mechanism's pose for positive (left) and negative (right) roll value, for dependent variables ẏC and φ̇yaw.

5.1.2 Steering influence of left wheel lever straddle angle θ_{l5}

In order to examine the steering influence of the left wheel lever straddle angle θ_{l5}, we input to MOBILE a velocity, $\dot{\theta}_{l5} = 0.5\,\mathrm{deg}/s$ and then a velocity $\dot{\theta}_{l5} = -0.5\,\mathrm{deg}/s$. The chosen step, for acceptable accuracy, of the MOBILE integrator is 0.001 s, as before. Then we observe these scenarios for 451 velocity integrations, that means approximately for 0.451 s. At every integration (step in time), we store the values of θ_{l5}, y_C and φ_{yaw} for each velocity and express their relationship in the following diagrams. In Table 5-2 we can see the initial conditions and fixed values for the left wheel lever straddle angle steering test for both velocities.

Table 5-2: Initial conditions and fixed values for the steering influence of left wheel lever straddle angle simulation, for dependent variables ẏC and φ̇yaw.

x_C (m)	0.0	dx_C (m/s)	0.0	cxd (m)	3.0
y_C (m)	0.0	-	-	cyd (m)	3.0
z_C (m)	2.0	dz_C (m/s)	0.0	w (m)	0.5
x_{Fl} (m)	3.5	dx_{Fl} (m/s)	0.0	wl (m)	2.5
y_{Fl} (m)	1.5	dy_{Fl} (m/s)	0.0	thighD (m)	2.0
z_{Fl} (m)	0.0	dz_{Fl} (m/s)	0.0	legD (m)	1.5
x_{Fr} (m)	3.5	dx_{Fr} (m/s)	0.0		
y_{Fr} (m)	-1.5	dy_{Fr} (m/s)	0.0		
z_{Fr} (m)	0.0	dz_{Fr} (m/s)	0.0		
z_{Wl} (m)	0.0	dz_{Wl} (m/s)	0.0		
z_{Wr} (m)	0.05	dz_{Wr} (m/s)	0.0		
φ_{yaw} (deg)	0.0	-	-		
φ_{pitch} (deg)	0.0	$d\varphi_{pitch}$ (deg/s)	0.0		
φ_{roll} (deg)	0.0	$d\varphi_{roll}$ (deg/s)	0.0		
ψ_{l1} (deg)	0.0	$d\psi_{l1}$ (deg/s)	0.0		
ψ_{r1} (deg)	0.0	$d\psi_{r1}$ (deg/s)	0.0		
θ_{l5} (deg)	0.0	$d\theta_{l5}$ (deg/s)	0.5/-0.5		
θ_{r5} (deg)	0.0	$d\theta_{r5}$ (deg/s)	0.0		

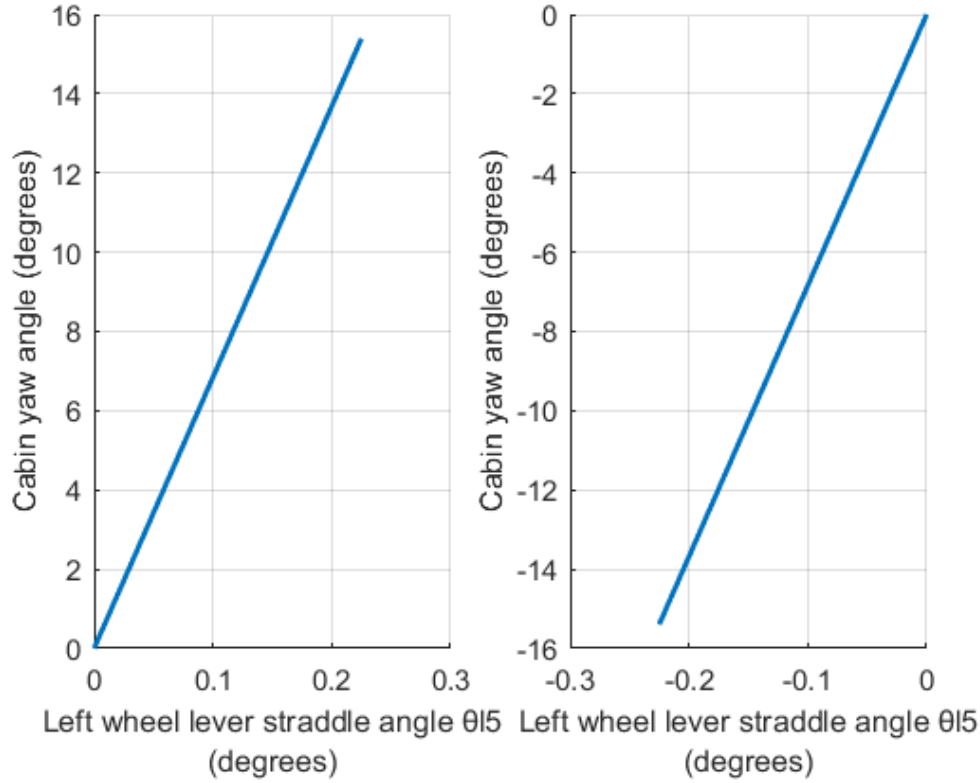

Figure 5-4: Influence of positive (left) and negative (right) left wheel lever straddle angles to cabin yaw rotation for dependent variables ẏC and φ̇yaw.

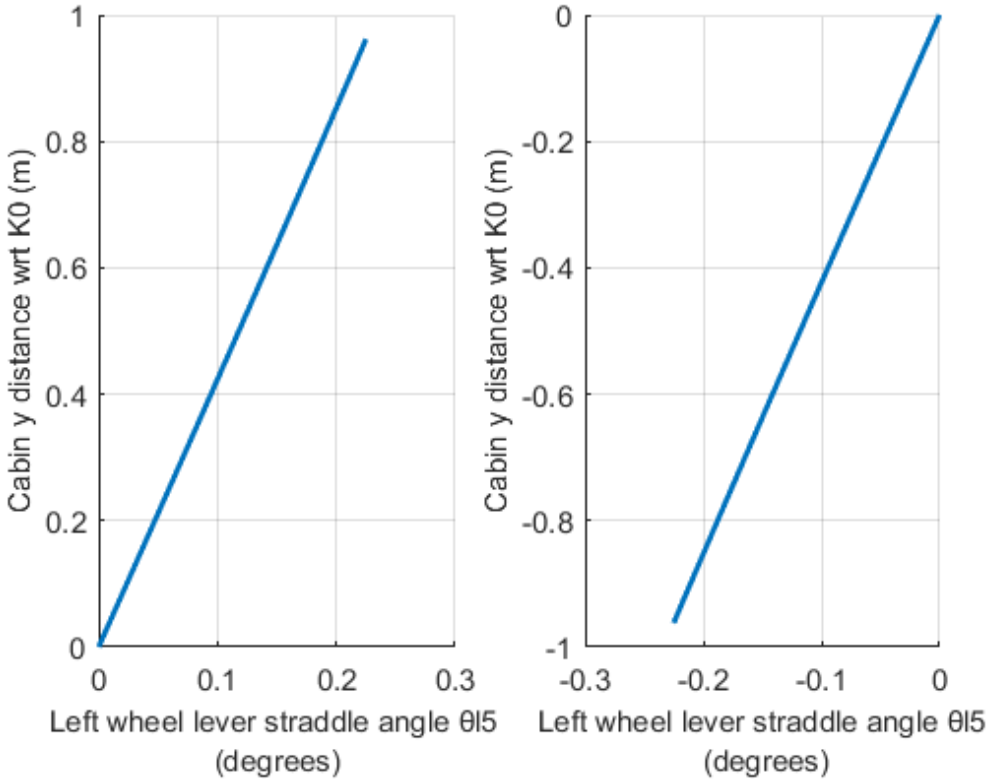

Figure 5-5: Influence of positive (left) and negative (right) left wheel lever straddle angles to cabin y distance with respect to K0 for dependent variables ẏC and φ̇yaw.

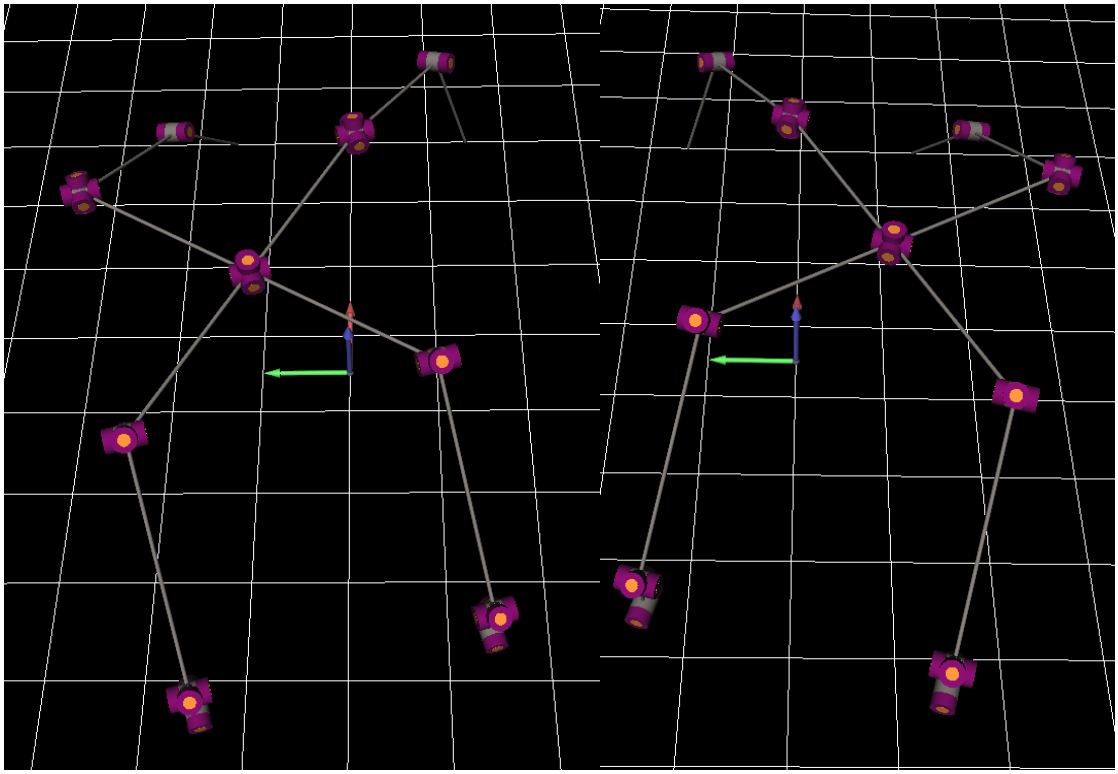

Figure 5-6: Mechanism's pose for positive θ_{l5} value (left) and negative θ_{l5} value (right), for dependent variables ẏC and φẏaw.

In this situation, we observe a similar behaviour to the previous one. For positive θ_{l5} velocity values we get the mechanism steer to the left with y_C and φ_{yaw} velocity values combining to achieve the airplane like steering. The exact opposite happens for negative θ_{l5} velocity values. This time the mechanism steers without changing its roll angle, as it is an independent variable and we don't change it.

5.1.3 Steering influence of right wheel lever straddle angle θ_{r5}

In order to examine the steering influence of the right wheel lever straddle angle θ_{r5}, we input to MOBILE a velocity, $\dot{\theta}_{r5} = 0.5\,\mathrm{deg}/s$ and then a velocity $\dot{\theta}_{r5} = -0.5\,\mathrm{deg}/s$. The chosen step, for acceptable accuracy, of the MOBILE integrator is 0.001 s, as before. Then we observe these scenarios for 451 velocity integrations, that means approximately for 0.451 s. At every integration (step in time), we store the values of θ_{r5}, y_C and φ_{yaw} for each velocity and express their relationship in the following diagrams. In Table 5-3 we can see the initial conditions and fixed values for the left wheel lever straddle angle steering test for both velocities.

Table 5-3: Initial conditions and fixed values for the steering influence of right wheel lever straddle angle simulation, for dependent variables ẏC and φẏaw.

x_C (m)	0.0	dx_C (m/s)	0.0	cxd (m)	3.0
y_C (m)	0.0	-	-	cyd (m)	3.0
z_C (m)	2.0	dz_C (m/s)	0.0	w (m)	0.5
x_{Fl} (m)	3.5	dx_{Fl} (m/s)	0.0	wl (m)	2.5

y_{Fl} (m)	1.5	dy_{Fl} (m/s)	0.0	thighD (m)	2.0
z_{Fl} (m)	0.0	dz_{Fl} (m/s)	0.0	legD (m)	1.5
x_{Fr} (m)	3.5	dx_{Fr} (m/s)	0.0		
y_{Fr} (m)	-1.5	dy_{Fr} (m/s)	0.0		
z_{Fr} (m)	0.0	dz_{Fr} (m/s)	0.0		
z_{Wl} (m)	0.0	dz_{Wl} (m/s)	0.0		
z_{Wr} (m)	0.05	dz_{Wr} (m/s)	0.0		
φ_{yaw} (deg)	0.0	-	-		
φ_{pitch} (deg)	0.0	$d\varphi_{pitch}$ (deg/s)	0.0		
φ_{roll} (deg)	0.0	$d\varphi_{roll}$ (deg/s)	0.0		
ψ_{l1} (deg)	0.0	$d\psi_{l1}$ (deg/s)	0.0		
ψ_{r1} (deg)	0.0	$d\psi_{r1}$ (deg/s)	0.0		
θ_{l5} (deg)	0.0	$d\theta_{l5}$ (deg/s)	0.0		
θ_{r5} (deg)	0.0	$d\theta_{r5}$ (deg/s)	0.5/-0.5		

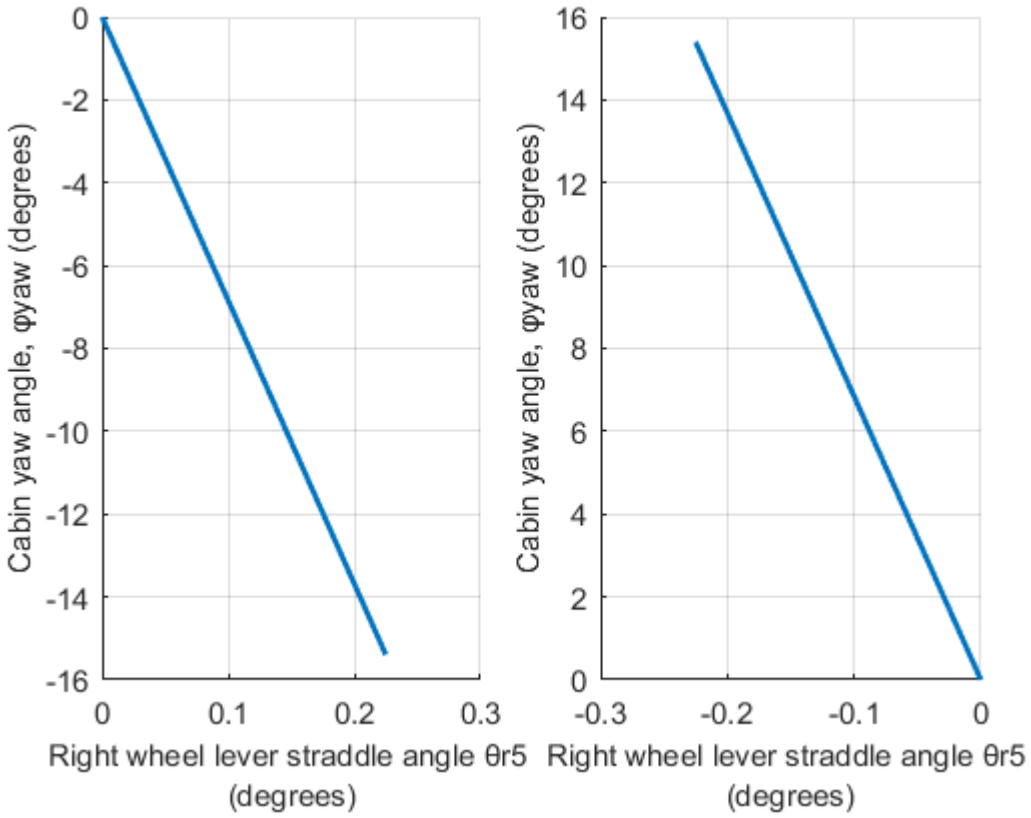

Figure 5-7: Influence of positive (left) and negative (right) right wheel lever straddle angles to cabin yaw rotation for dependent variables ẏC and φ̇yaw.

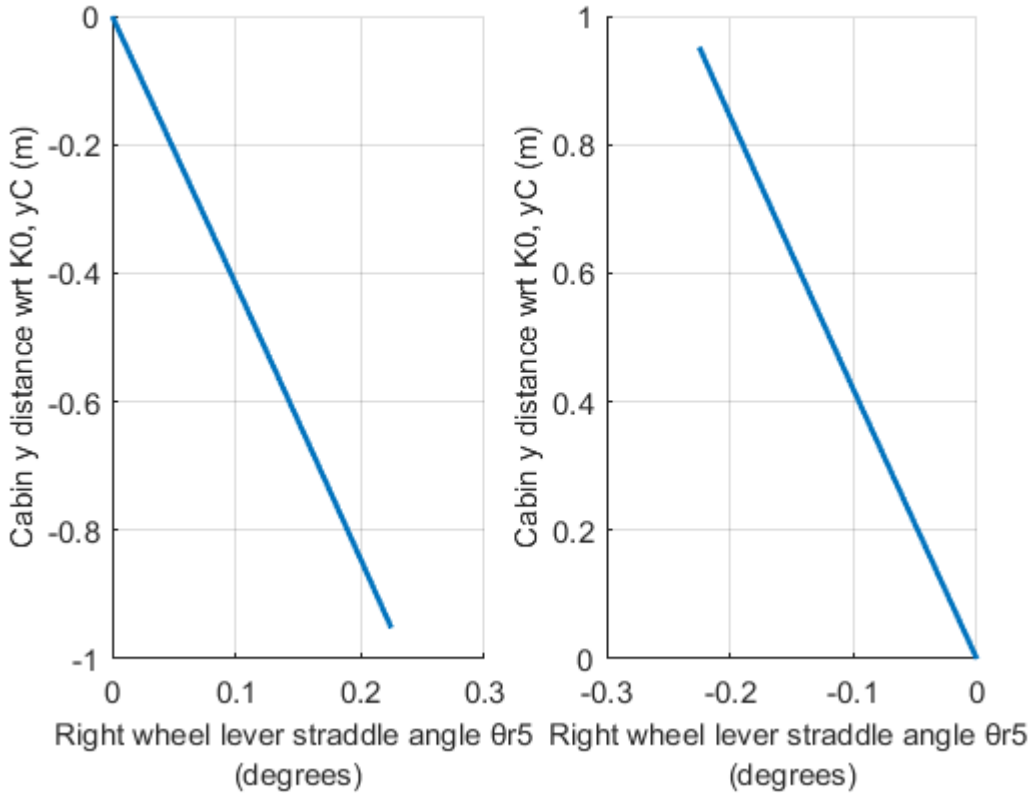

Figure 5-8: Influence of positive (left) and negative (right) right wheel lever straddle angles to cabin y distance with respect to K0 for dependent variables ẏC and φ̇yaw.

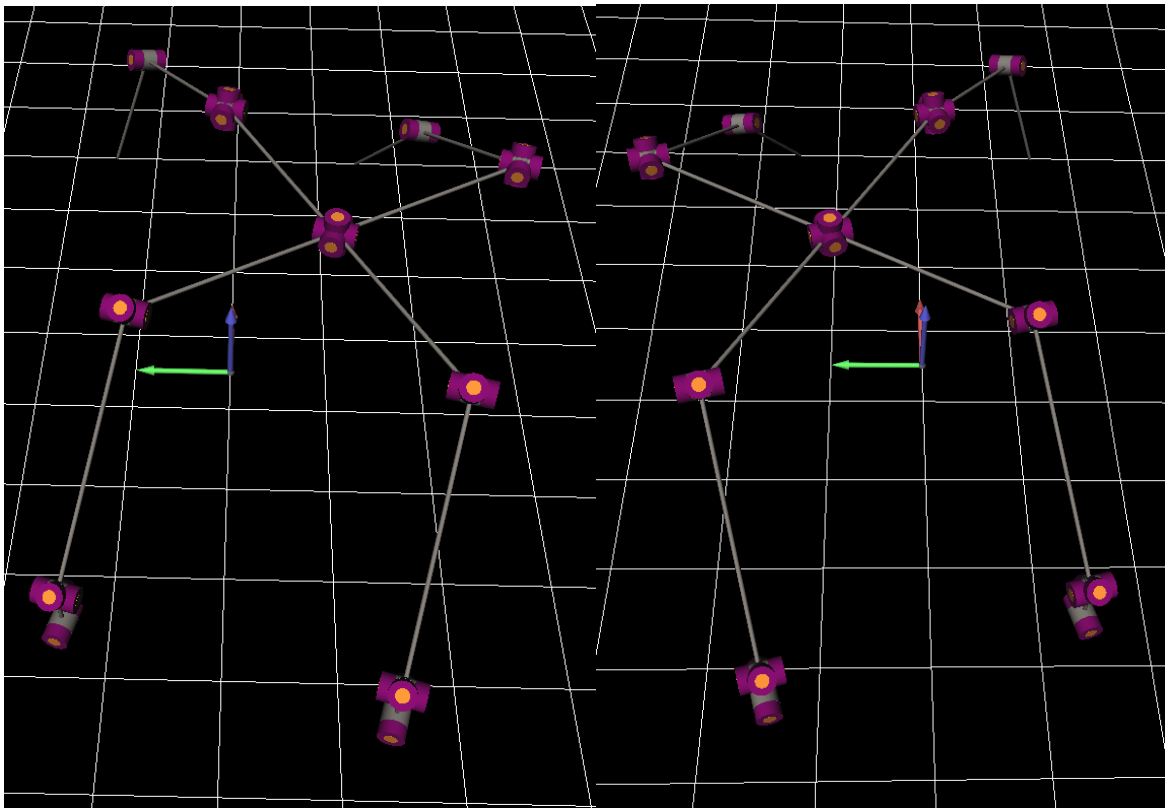

Figure 5-9: Mechanism's pose for positive θ_{r5} values (left) and negative θ_{r5} value (right), for dependent variables ẏC and φ̇yaw.

In this case, we observe that the results are exactly symmetrically opposite from the θ_{l5} case of Chapter 5.1.2.

5.2 For dependent variables θ_{l5}, θ_{r5}

For this choice of dependent variables only φ_{roll}, φ_{yaw} and y_C can influence the dependent variables by bringing the non-holonomic constraints to the test and these are going to be examined.

5.2.1 Steering influence of cabin roll angle φ_{roll}

In order to examine the steering influence of the cabin roll angle φ_{roll} for this case, we input to MOBILE a velocity, $\dot{\varphi}_{roll} = 5.0\,\mathrm{deg}/s$ and then a velocity $\dot{\varphi}_{roll} = -5.0\,\mathrm{deg}/s$. The chosen step, for acceptable accuracy, of the MOBILE integrator is 0.001 s, as before. Then we observe these scenarios for 4001 velocity integrations, that means approximately for 4 s. At every integration (step in time), we store the values of φ_{roll}, θ_{l5} and θ_{r5} for each velocity and express their relationship in the following diagrams. In Table 5-4 we can see the initial conditions and fixed values for the cabin roll angle steering test for both velocities.

Table 5-4: Initial conditions and fixed values for the steering influence of cabin roll angle simulation, for dependent variables θ_{l5}, θ_{r5}.

x_C (m)	0.0	dx_C (m/s)	0.0	cxd (m)	3.0
y_C (m)	0.0	dy_C (m/s)	0.0	cyd (m)	3.0
z_C (m)	2.0	dz_C (m/s)	0.0	w (m)	0.5
x_{Fl} (m)	3.5	dx_{Fl} (m/s)	0.0	wl (m)	2.5
y_{Fl} (m)	1.5	dy_{Fl} (m/s)	0.0	thighD (m)	2.0
z_{Fl} (m)	0.0	dz_{Fl} (m/s)	0.0	legD (m)	1.5
x_{Fr} (m)	3.5	dx_{Fr} (m/s)	0.0		
y_{Fr} (m)	-1.5	dy_{Fr} (m/s)	0.0		
z_{Fr} (m)	0.0	dz_{Fr} (m/s)	0.0		
z_{Wl} (m)	0.0	dz_{Wl} (m/s)	0.0		
z_{Wr} (m)	0.05	dz_{Wr} (m/s)	0.0		
φ_{yaw} (deg)	0.0	$d\varphi_{yaw}$ (deg/s)	0.0		
φ_{pitch} (deg)	0.0	$d\varphi_{pitch}$ (deg/s)	0.0		
φ_{roll} (deg)	0.0	$d\varphi_{roll}$ (deg/s)	5.0/-5.0		
ψ_{l1} (deg)	0.0	$d\psi_{l1}$ (deg/s)	0.0		
ψ_{r1} (deg)	0.0	$d\psi_{r1}$ (deg/s)	0.0		
θ_{l5} (deg)	0.0	-	-		
θ_{r5} (deg)	0.0	-	-		

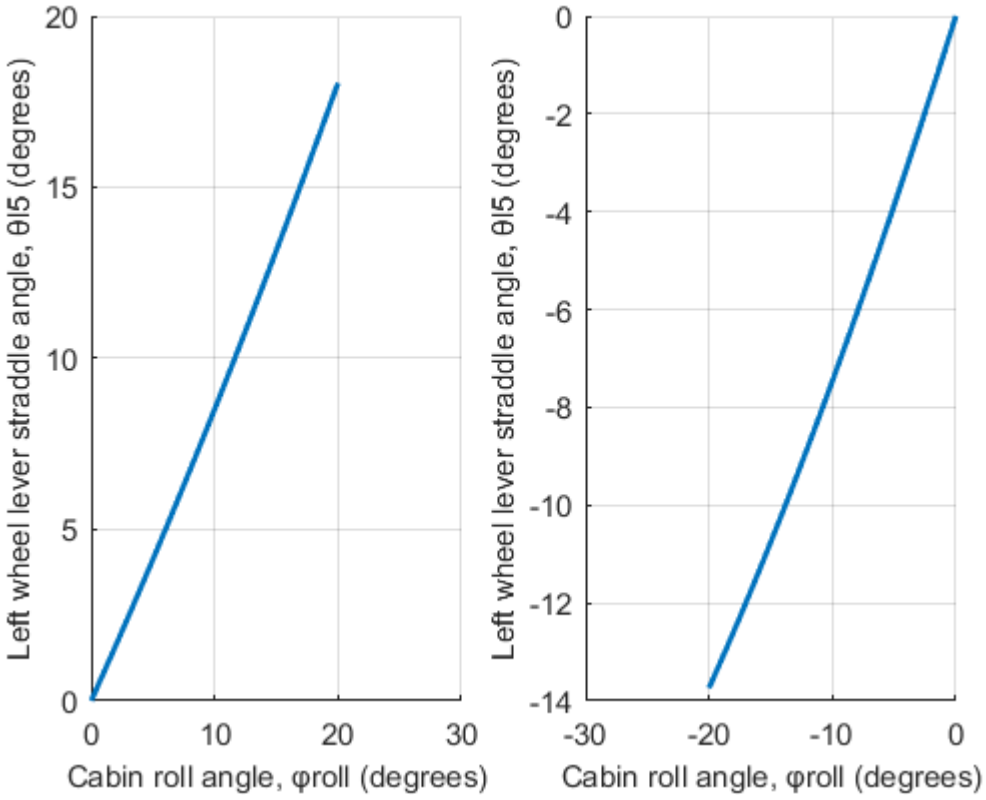

Figure 5-10: Positive (left) and negative (right) cabin roll angles influence to left wheel lever straddle angle, for dependent variables θ_{l5}, θ_{r5}.

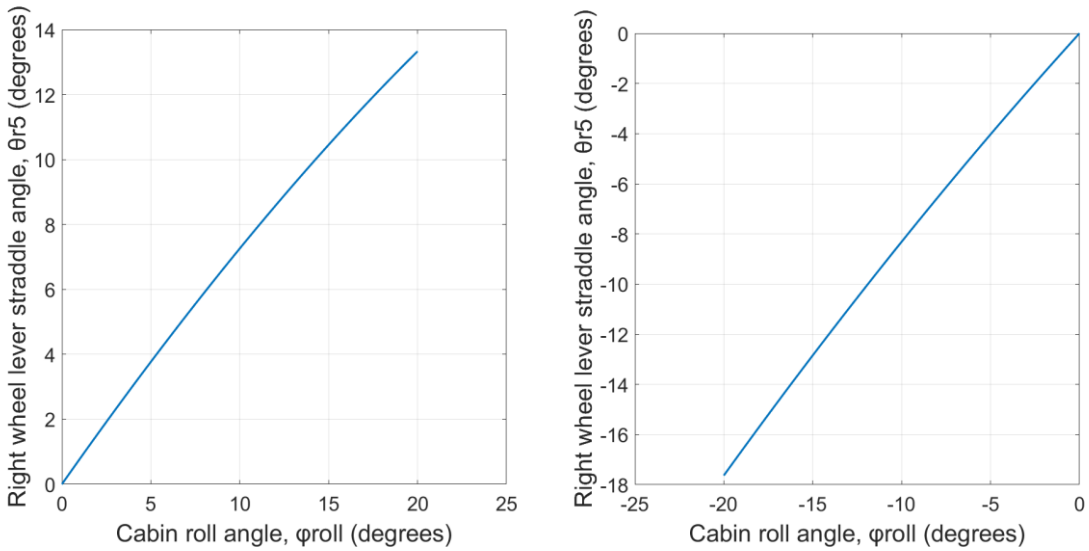

Figure 5-11: Positive (left) and negative (right) cabin roll angles influence to right wheel lever straddle angle, for dependent variables θ_{l5}, θ_{r5}.

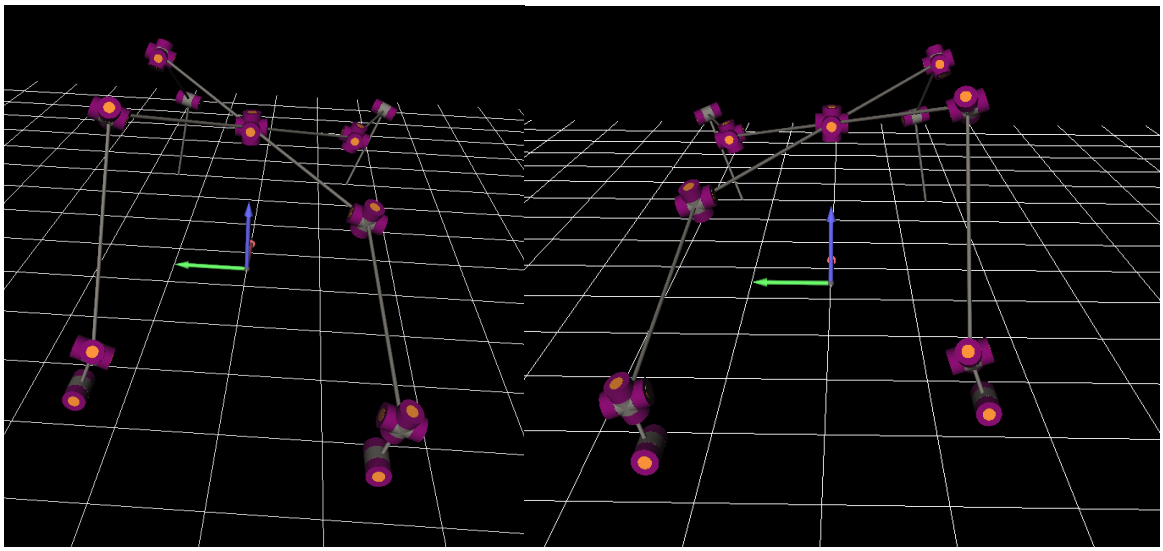

Figure 5-12: Mechanism's pose for positive (left) and negative (right) cabin roll value, for dependent variables θ_{l5}, θ_{r5}.

In this case of dependent variables, we observe that we can alternate the cabin's roll values without changing cabin's yaw angle or y_C like before.

5.2.2 Steering influence of cabin yaw angle φ_{yaw}

In order to examine the steering influence of the cabin yaw angle φ_{yaw} for this case, we input to MOBILE a velocity, $\dot{\varphi}_{yaw} = 5.0\,\mathrm{deg}/s$ and then a velocity $\dot{\varphi}_{yaw} = -5.0\,\mathrm{deg}/s$. The chosen step, for acceptable accuracy, of the MOBILE integrator is 0.001 s, as before. Then we observe these scenarios for 4001 velocity integrations, that means approximately for 4 s. At every integration (step in time), we store the values of **φ_{yaw}, θ_{l5} and θ_{r5}** for each velocity and express their relationship in the following diagrams. In Table 5-5 we can see the initial conditions and fixed values for the cabin roll angle steering test for both velocities.

Table 5-5: Initial conditions and fixed values for the steering influence of cabin yaw angle simulation, for dependent variables θ_{l5}, θ_{r5}.

x_C (m)	0.0	dx_C (m/s)	0.0	cxd (m)	3.0
y_C (m)	0.0	dy_C (m/s)	0.0	cyd (m)	3.0
z_C (m)	2.0	dz_C (m/s)	0.0	w (m)	0.5
x_{Fl} (m)	3.5	dx_{Fl} (m/s)	0.0	wl (m)	2.5
y_{Fl} (m)	1.5	dy_{Fl} (m/s)	0.0	thighD (m)	2.0
z_{Fl} (m)	0.0	dz_{Fl} (m/s)	0.0	legD (m)	1.5
x_{Fr} (m)	3.5	dx_{Fr} (m/s)	0.0		
y_{Fr} (m)	-1.5	dy_{Fr} (m/s)	0.0		
z_{Fr} (m)	0.0	dz_{Fr} (m/s)	0.0		
z_{Wl} (m)	0.0	dz_{Wl} (m/s)	0.0		
z_{Wr} (m)	0.05	dz_{Wr} (m/s)	0.0		
φ_{yaw} (deg)	0.0	$d\varphi_{yaw}$ (deg/s)	5.0/-5.0		

φ_{pitch} (deg)	0.0	$d\varphi_{pitch}$ (deg/s)	0.0		
φ_{roll} (deg)	0.0	$d\varphi_{roll}$ (deg/s)	0.0		
ψ_{l1} (deg)	0.0	$d\psi_{l1}$ (deg/s)	0.0		
ψ_{r1} (deg)	0.0	$d\psi_{r1}$ (deg/s)	0.0		
θ_{l5} (deg)	0.0	-	-		
θ_{r5} (deg)	0.0	-	-		

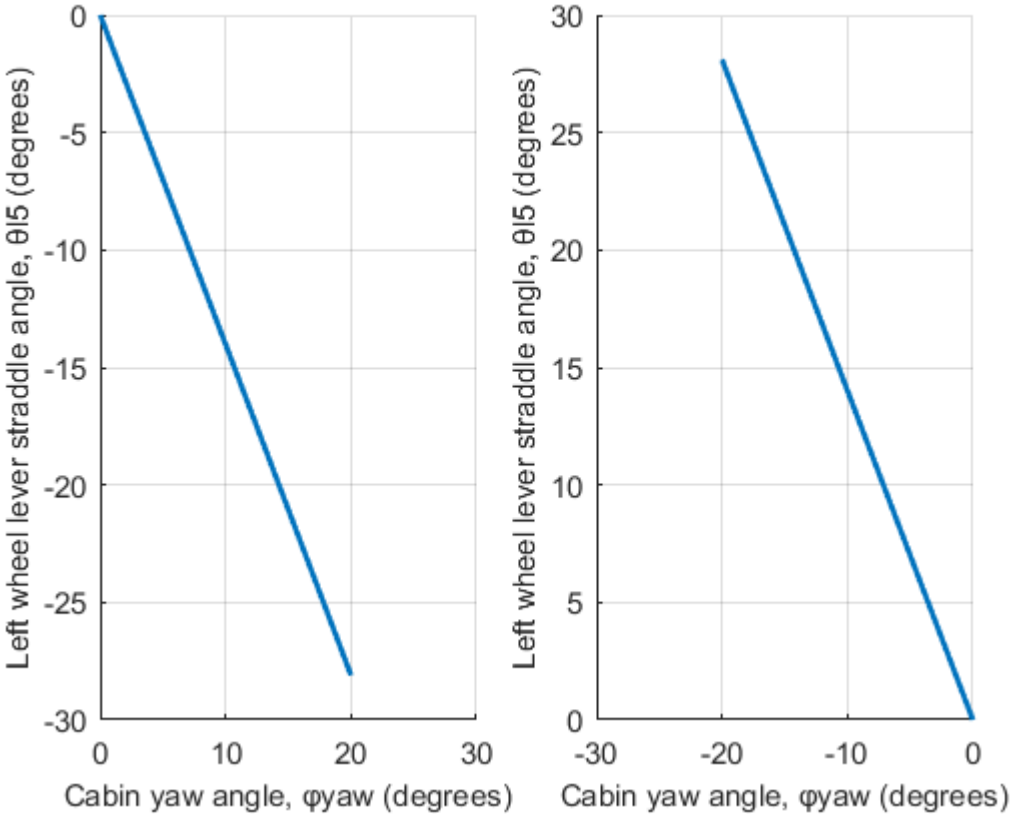

Figure 5-13: Positive (left) and negative (right) cabin yaw angles influence to left wheel lever straddle angle, for dependent variables θ_{l5}, θ_{r5}.

In this case, we note that having as dependent variables θ_{l5} and θ_{r5} we can steer the vehicle by changing its cabin's yaw angle without any other cabin movement.

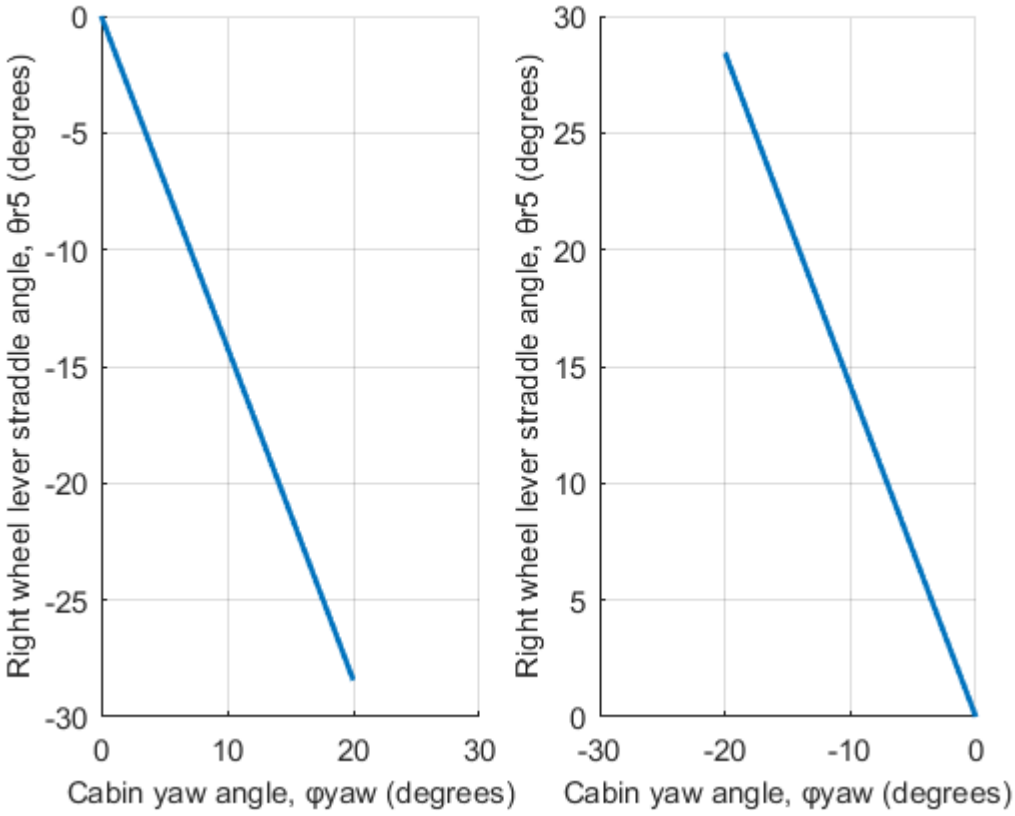

Figure 5-14: Positive (left) and negative (right) cabin yaw angles influence to right wheel lever straddle angle, for dependent variables θ_{l5}, θ_{r5}.

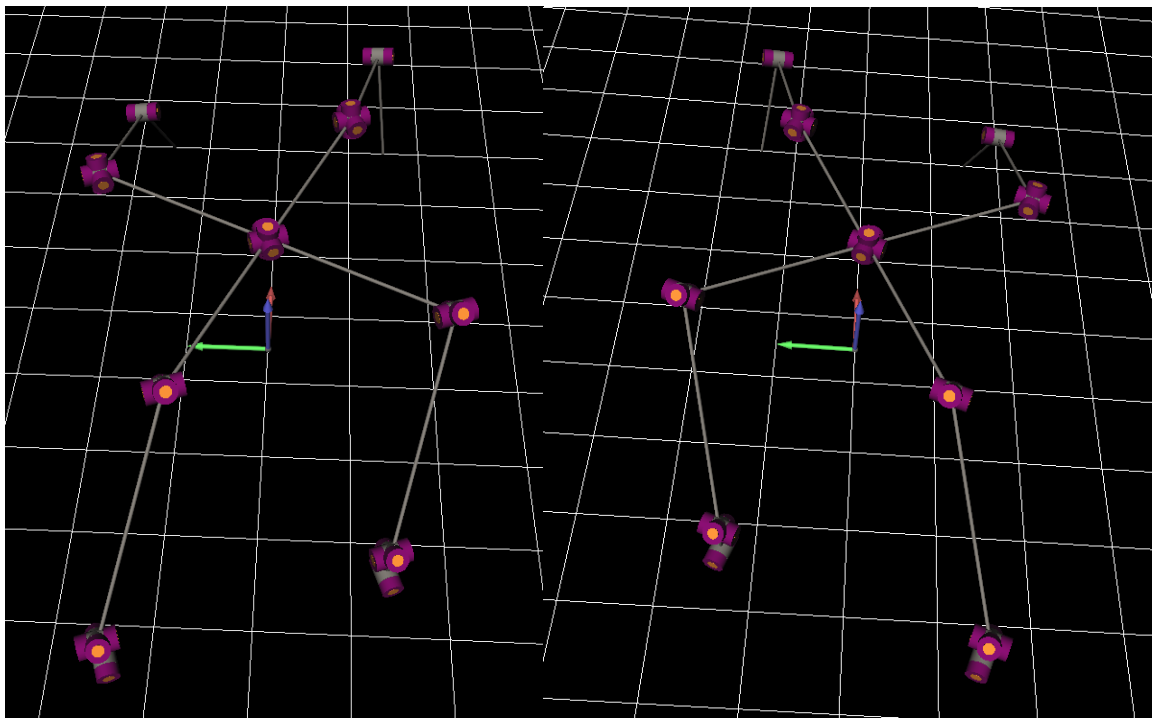

Figure 5-15: Mechanism's pose for positive (left) and negative (right) cabin yaw value, for dependent variables θ_{l5}, θ_{r5}.

5.2.3 Steering influence of cabin y distance with respect to K0, yC

In order to examine the steering influence of the cabin y distance with respect to the ground fixed frame K0, y_C**,** for this case, we input to MOBILE a velocity, $\dot{y}_C = 0.5\,\mathrm{deg}/s$ and then a velocity $\dot{y}_C = -0.5\,\mathrm{deg}/s$. The chosen step, for acceptable accuracy, of the MOBILE integrator is 0.001 s, as before. Then we observe these scenarios for 2001 velocity integrations, that means approximately for 2 s. At every integration (step in time), we store the values of y_C, θ_{l5} and θ_{r5} for each velocity and express their relationship in the following diagrams on. In Table 5-6 we can see the initial conditions and fixed values for the cabin y movement test for both velocities.

Table 5-6: Initial conditions and fixed values for the steering influence of yC simulation, for dependent variables θl5, θr5.

x_C (m)	0.0	dx_C (m/s)	0.0	cxd (m)	3.0
y_C (m)	0.0	dy_C (m/s)	0.5/-0.5	cyd (m)	3.0
z_C (m)	2.0	dz_C (m/s)	0.0	w (m)	0.5
x_{Fl} (m)	3.5	dx_{Fl} (m/s)	0.0	wl (m)	2.5
y_{Fl} (m)	1.5	dy_{Fl} (m/s)	0.0	thighD (m)	2.0
z_{Fl} (m)	0.0	dz_{Fl} (m/s)	0.0	legD (m)	1.5
x_{Fr} (m)	3.5	dx_{Fr} (m/s)	0.0		
y_{Fr} (m)	-1.5	dy_{Fr} (m/s)	0.0		
z_{Fr} (m)	0.0	dz_{Fr} (m/s)	0.0		
z_{Wl} (m)	0.0	dz_{Wl} (m/s)	0.0		
z_{Wr} (m)	0.05	dz_{Wr} (m/s)	0.0		
φ_{yaw} (deg)	0.0	$d\varphi_{yaw}$ (deg/s)	0.0		
φ_{pitch} (deg)	0.0	$d\varphi_{pitch}$ (deg/s)	0.0		
φ_{roll} (deg)	0.0	$d\varphi_{roll}$ (deg/s)	0.0		
ψ_{l1} (deg)	0.0	$d\psi_{l1}$ (deg/s)	0.0		
ψ_{r1} (deg)	0.0	$d\psi_{r1}$ (deg/s)	0.0		
θ_{l5} (deg)	0.0	-	-		
θ_{r5} (deg)	0.0	-	-		

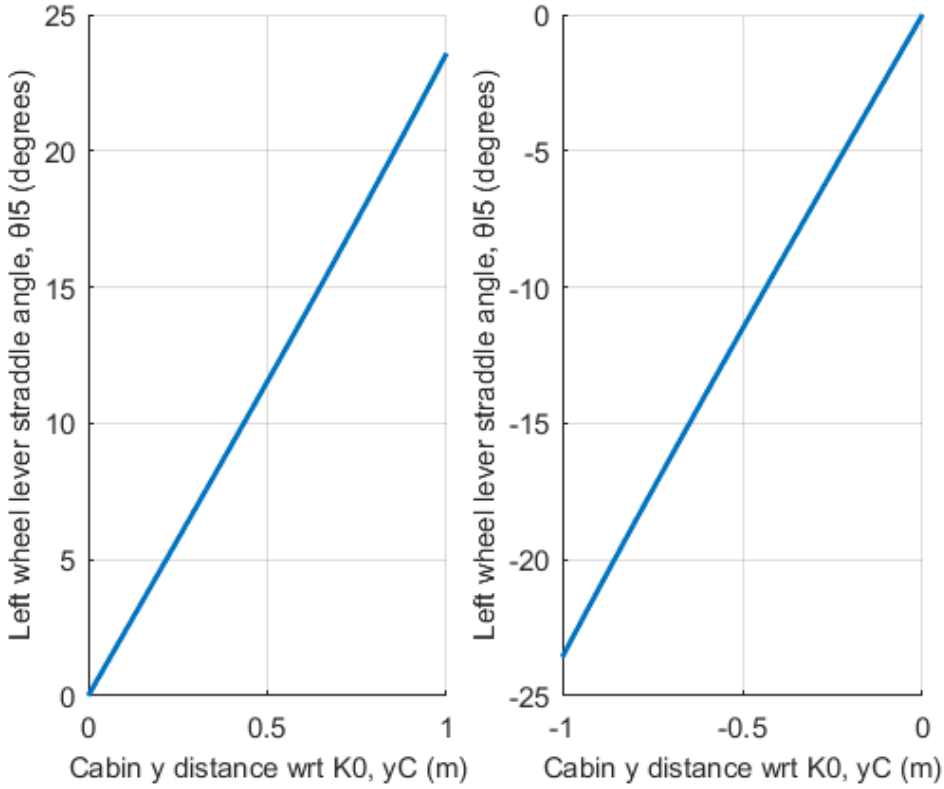

Figure 5-16: Positive (left) and negative (right) yC values influence to left wheel lever straddle angle, for dependent variables θ_{l5}, θ_{r5}.

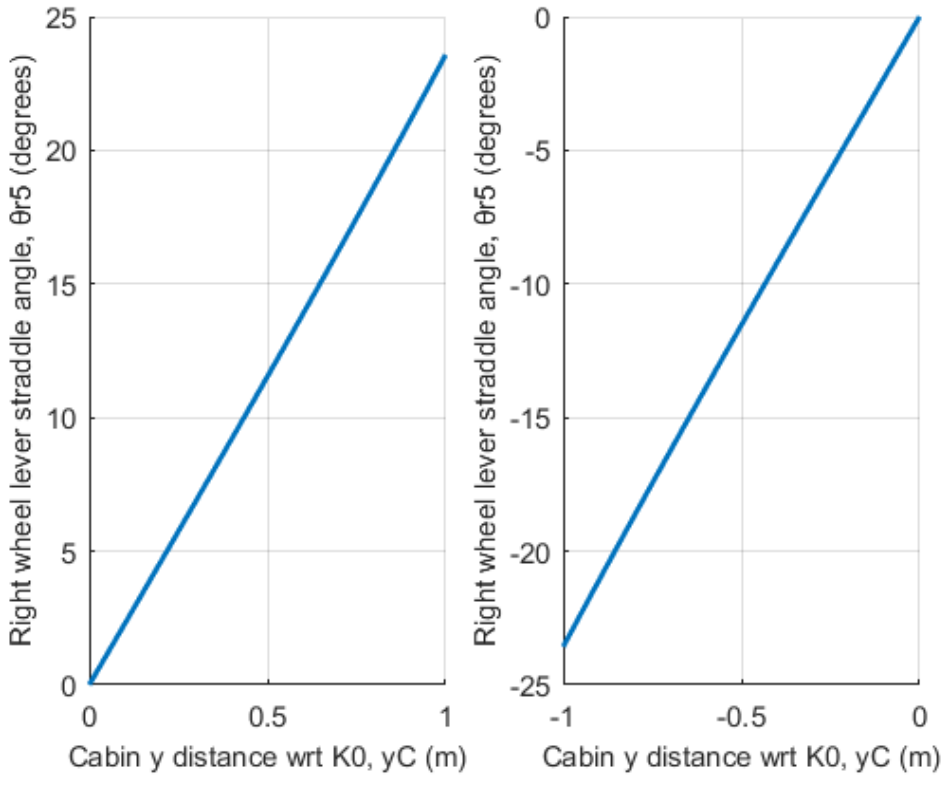

Figure 5-17: Positive (left) and negative (left) yC values influence to right wheel lever straddle angle, for dependent variables θ_{l5}, θ_{r5}.

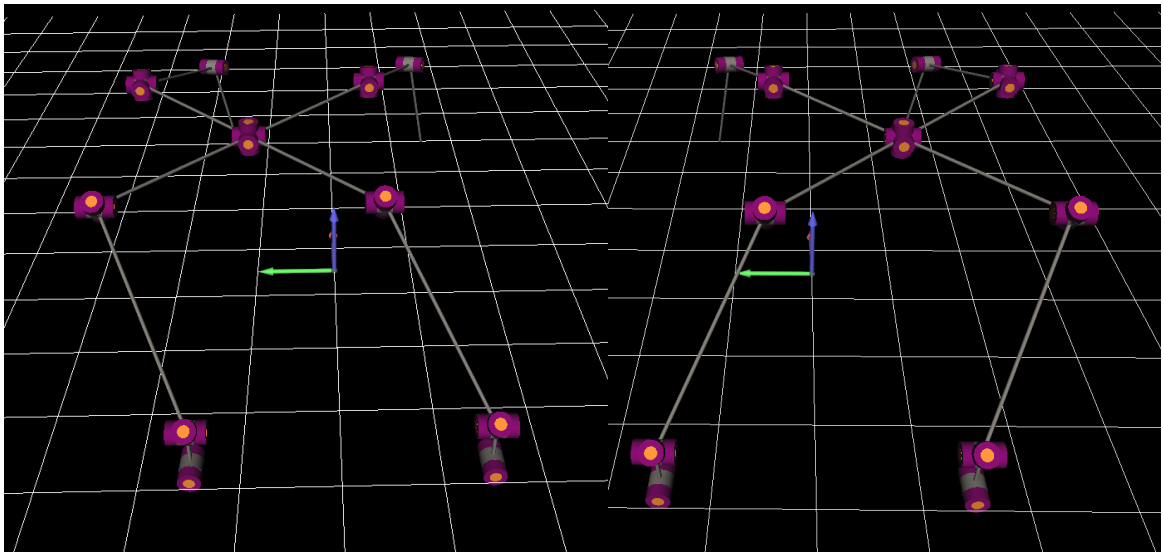

Figure 5-18: Mechanism's pose for positive (left) and negative (right) yC value, for dependent variables θ_{l5}, θ_{r5}.

Finally, in this case where the dependent variables are the wheel lever straddle angles the cabin can move in the y direction (green arrow in the above Figure). Scenario that wasn't possible with the previous choice of dependent variables.

We observe that by selecting different pairs of dependent velocities, we can achieve different behaviours between the velocity set of Equation (4.6):

$$\dot{q}_I = [\dot{x}_C, \dot{y}_C, \dot{z}_C, \dot{\varphi}_{yaw}, \dot{\varphi}_{pitch}, \dot{\varphi}_{roll}, \dot{\theta}_{l5}, \dot{\theta}_{r5}]^T$$

6 Dynamics

Robot dynamics is concerned with the relationship between the forces acting on a robot mechanism and the accelerations they produce. The two main problems in robot dynamics are forward dynamics and inverse dynamics.

In the forward dynamics problem, the motion is calculated from known applied control or reaction forces and/or torques. Forward dynamics is used primarily for simulation, for example to test the response of a robot to a control strategy.

On the other hand, in the inverse dynamics problem the required input forces and/or torques are reconstructed from the current or desired movements and from known external forces. Humans can perform very complicated and precise movements. Before the body moves, the brain calculates the necessary movement of each muscle involved and tells the muscles what to do as the body moves. In the case of a robot arm, the "muscles" are the electric motors which must apply the desired torque at a given moment. Each motor must be supplied with just the right amount of electric current, at just the right time. Inverse dynamics are used for various applications, including: on-line control of robot motions and forces, trajectory design and optimization, design of robot mechanisms actuator sizing, and as a component in some forward-dynamics algorithms.

Setting up the kinematics is the most difficult part; as soon as this has been done, both inverse and forward dynamics calculations can be done very easily by matrix multiplication and matrix inversion, respectively. In this chapter, we will go through how we can construct the equations of motion for the hybrid excavator model using MOBILE and solving them to get the inverse dynamics using MATLAB.

6.1 Calculation of the dependent accelerations on MOBILE

Firstly, we have to implement the non-holonomic constraints at the acceleration level. The two velocities $\dot{\boldsymbol{\beta}}$ (see Equation (6.1)) that were chosen to be dependent are the sway of the cabin (side to side motion along the y axis of the cabin-fixed coordinate frame K_C) and the yaw of the cabin (rotation of the cabin about the vertical axis z of the cabin-fixed coordinate frame K_C) as seen in Figure 6-1. Therefore, the accelerations of these two variables are dependent too.

$$\dot{\beta} = \begin{bmatrix} \dot{\beta}_1 : sway \\ \dot{\beta}_2 : yaw \end{bmatrix} \tag{6.1}$$

The constraints that the wheels add, as mentioned before, are that the wheels cannot move in the y direction of the wheel-tip-fixed coordinate frame, therefore at velocity level:

$$\dot{\varphi} = \begin{bmatrix} \dot{\varphi}_1 \\ \dot{\varphi}_2 \end{bmatrix} \triangleq 0 \tag{6.2}$$

The equation that describes the constraints that the wheels add (see Section 4.1.1), is Equation (6.3):

$$\dot{\varphi} = J_\beta \dot{\beta} + \hat{\dot{\varphi}} = 0 \tag{6.3}$$

In this equation $\hat{\dot{\varphi}}$ is the partial wheel tip velocity due to inputs $\dot{q}_I$ (subset of independent variables affected by the non-holonomic constraints, see Equation (6.4)) and $J_\beta \dot{\beta}$ is the partial wheel tip velocity due to the dependent variables.

$$\dot{q}_I = [\dot{x}_C, \dot{y}_C, \dot{z}_C, \dot{\varphi}_{yaw}, \dot{\varphi}_{pitch}, \dot{\varphi}_{roll}, \dot{\theta}_{l5}, \dot{\theta}_{r5}]^T \tag{6.4}$$

By differentiating this equation, we get Equation (6.5), that yields the wheel tip acceleration $\ddot{\varphi}$ in the y direction of the wheel-tip-fixed coordinate frame, which must be zero in order for the non-holonomic constraints to be satisfied (see Figure 6-1):

$$\ddot{\varphi} = J_\beta \ddot{\beta} + \dot{J}_\beta \dot{\beta} + \hat{\ddot{\varphi}} = 0 \tag{6.5}$$

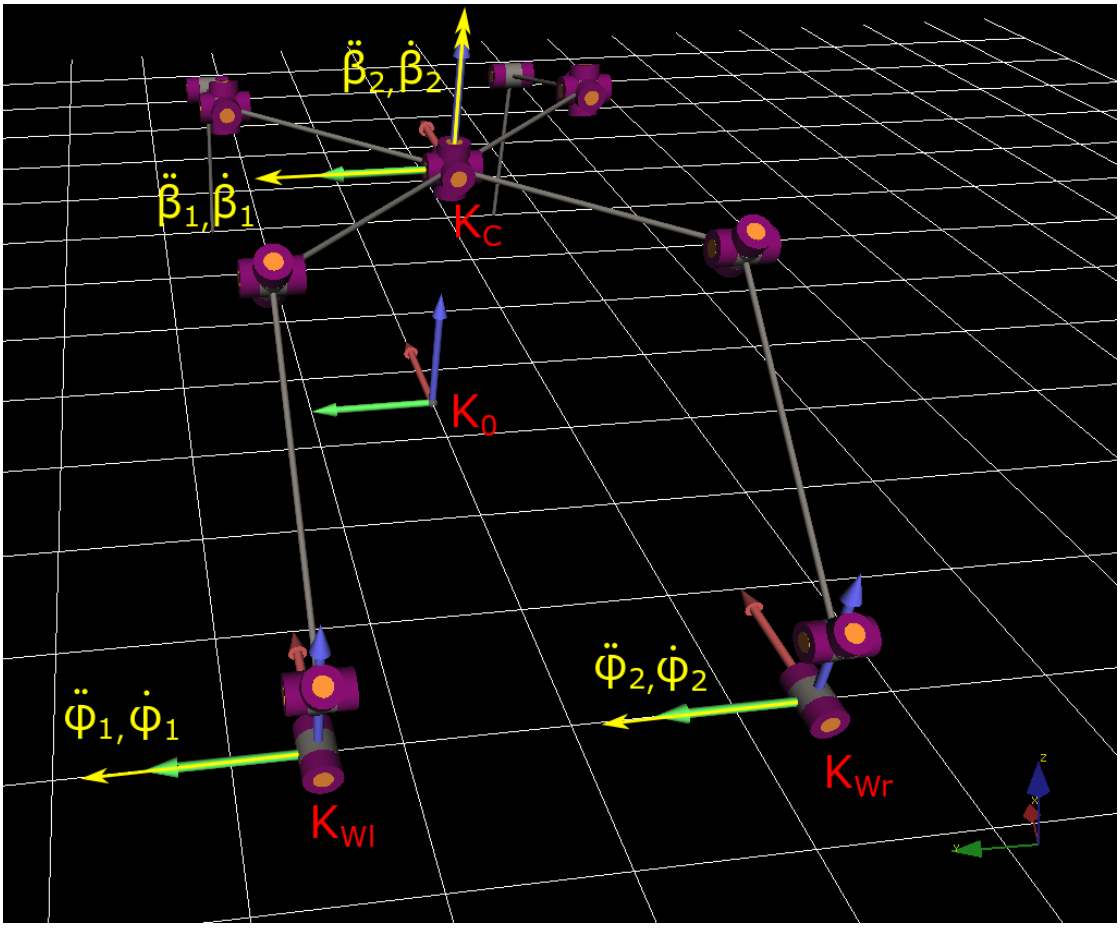

Figure 6-1: The non-holonomic acceleration constraints φ_1, φ_2 and the dependent accelerations β_1 and β_2 because of them.

From Equation (6.5) we only need to calculate $\hat{\ddot{\varphi}}$ and $\dot{J}_\beta$, as the J_β Jacobian and the $\dot{\beta}$ velocities were calculated in Chapter 4. Capitalizing on MOBILE's capabilities, we don't need to compute these missing terms separately, conversely, we can compute them in a unified

manner. In order to do that, we set instantaneously the values of the dependent accelerations, $\ddot{\beta}$, to zero and we do an acceleration transmission, reading $\ddot{\varphi}$ which is now equal to:

$$\ddot{\varphi} = \dot{J}_{\beta}\dot{\beta} + \hat{\ddot{\varphi}} = \varphi' \tag{6.6}$$

Having the value of φ' we can now calculate the dependent accelerations, by doing the above in every time step, as:

$$\begin{aligned}
&\ddot{\beta} = -J_{\beta}^{-1}\varphi' = -\begin{bmatrix} J_{\beta_{11}} & J_{\beta_{12}} \\ J_{\beta_{21}} & J_{\beta_{22}} \end{bmatrix}^{-1} \varphi' = \\
&-\frac{1}{J_{\beta_{11}}J_{\beta_{22}} - J_{\beta_{12}}J_{\beta_{21}}}\begin{bmatrix} J_{\beta_{22}} & -J_{\beta_{12}} \\ -J_{\beta_{21}} & J_{\beta_{11}} \end{bmatrix}\begin{bmatrix} \varphi_1' \\ \varphi_2' \end{bmatrix} \Rightarrow \\
&\ddot{\beta}_1 = \frac{\varphi_1' J_{\beta_{22}} - \varphi_2' J_{\beta_{12}}}{J_{\beta_{11}}J_{\beta_{22}} - J_{\beta_{12}}J_{\beta_{21}}} \\
&\ddot{\beta}_2 = \frac{\varphi_2' J_{\beta_{11}} - \varphi_1' J_{\beta_{21}}}{J_{\beta_{11}}J_{\beta_{22}} - J_{\beta_{12}}J_{\beta_{21}}}
\end{aligned} \tag{6.7}$$

6.2 Adding mass properties to the robot

In MOBILE, mass elements model the inertia properties of a rigid body, i.e., its mass m and its inertia tensor Θ_s. The inertia tensor is assumed to be defined with respect to the center of gravity of the body. The center of gravity itself can be offset from the origin of a reference frame K by a vector Δs, as seen in Figure 6-2. The linear and angular velocity vectors (**u** and **ω**) as well as the force and torque (**f** and **τ**) are collected in the vectors **t** and **w** accordingly as seen in Equation (6.8) All tensorial quantities are always assumed to be decomposed with respect to the actual frame K as seen in Figure 6-2.

Inertial forces in MOBILE are modelled as d' Alembert's forces. Under a general motion of a frame K, the d' Alembert's forces exerted by the mass upon the origin of that frame K are computed from the linear acceleration **α**, angular velocity **ω** and angular acceleration **ω̇** of frame K. They can be seen in Equation (6.9):

$$\underline{t} = \begin{bmatrix} \omega \\ u \end{bmatrix} \; and \; \; w = \begin{bmatrix} \tau \\ f \end{bmatrix} \tag{6.8}$$

$$\begin{aligned}
f &= -m\left[a + \dot{\omega} \times \Delta s + \omega \times \left(\omega \times \Delta s\right)\right], \\
\tau &= -\left[\Theta_s \dot{\omega} + \omega \times \Theta_s \omega\right] + \Delta s \times f
\end{aligned} \tag{6.9}$$

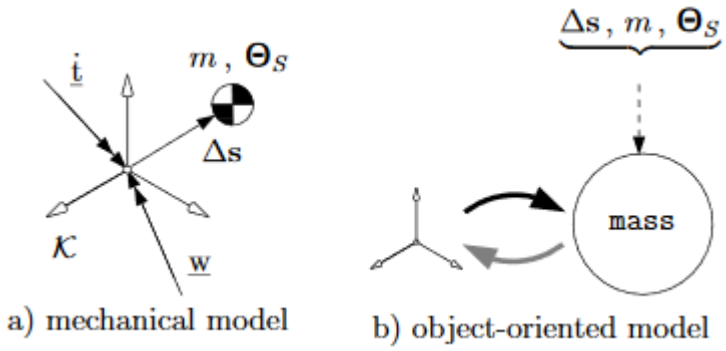

Figure 6-2: Model of a mass element on MOBILE.

The material used to calculate the mass properties is Cytec Thornel® Mat VMA Carbon Fiber and the some of its properties can be seen in Figure 6-3.

Physical Properties	Metric	English	Comments
Density	2.00 g/cc	0.0723 lb/in³	Filament Property.
Apparent Bulk Density	0.0360 g/cc	0.00130 lb/in³	
Specific Surface Area	0.40 m²/g	0.40 m²/g	Filament Property

Mechanical Properties	Metric	English	Comments
Tensile Strength, Ultimate	1400 MPa	203000 psi	Filament Property. Mat strength is 120 N/m
Modulus of Elasticity	170 GPa	24700 ksi	Filament property, in tension

Thermal Properties	Metric	English	Comments
Thermal Conductivity	0.410 W/m-K	2.85 BTU-in/hr-ft²-°F	across thickness of mat

Electrical Properties	Metric	English	Comments
Electrical Resistivity	0.00120 ohm-cm	0.00120 ohm-cm	Filament
	0.700 ohm-cm	0.700 ohm-cm	Mat

Figure 6-3: Cytec Thornel Mat VMA Carbon Fiber properties.

6.2.1 Cabin mass properties

The cabin of the excavator was modelled in Solidworks as a 3mx3mx0.5m solid rectangular cuboid of the before-mentioned material (see Figure 6-4). Then the mass properties of that model were extracted and can be seen in Figure 6-5. The center of mass and the output coordinate system used by Solidworks for the mass properties are the same.

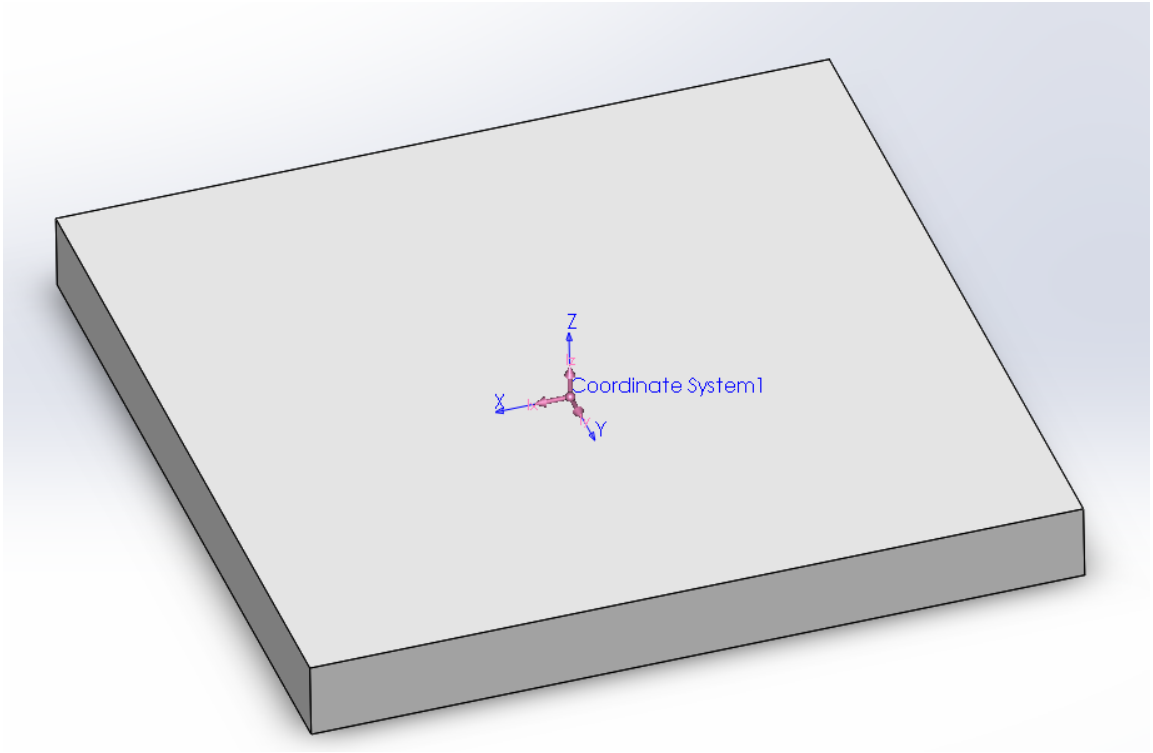

Figure 6-4: Solidworks model of the cabin. The pink system of coordinates denotes the center of mass while the blue one is the output coordinate system for Solidworks' mass properties calculations.

Density = 2000.000 kilograms per cubic meter

Mass = 5400.000 kilograms

Volume = 2.700 cubic meters

Surface area = 21.600 square meters

Center of mass: (meters)
X = 0.000
Y = 0.000
Z = 0.000

Principal axes of inertia and principal moments of inertia: (kilograms * square meters)
Taken at the center of mass.

Ix = (1.000, 0.000, 0.000)	Px = 4090.500
Iy = (0.000, 1.000, 0.000)	Py = 4090.500
Iz = (0.000, 0.000, 1.000)	Pz = 8100.000

Moments of inertia: (kilograms * square meters)
Taken at the center of mass and aligned with the output coordinate system.

Lxx = 4090.500	Lxy = 0.000	Lxz = 0.000
Lyx = 0.000	Lyy = 4090.500	Lyz = 0.000
Lzx = 0.000	Lzy = 0.000	Lzz = 8100.000

Figure 6-5: Mass properties of the cabin.

Next, we insert a mass object with the above mass and inertia matrix to MOBILE in order to model cabin's mass, called M1 (see Figure 6-6). The rigid body's center of mass is located at the center of the cabin's kinematical chain. It should be mentioned here that the center of mass is located higher that than in reality, but for the purposes of this work, it's considered to be at the center of the cabin's kinematical chain.

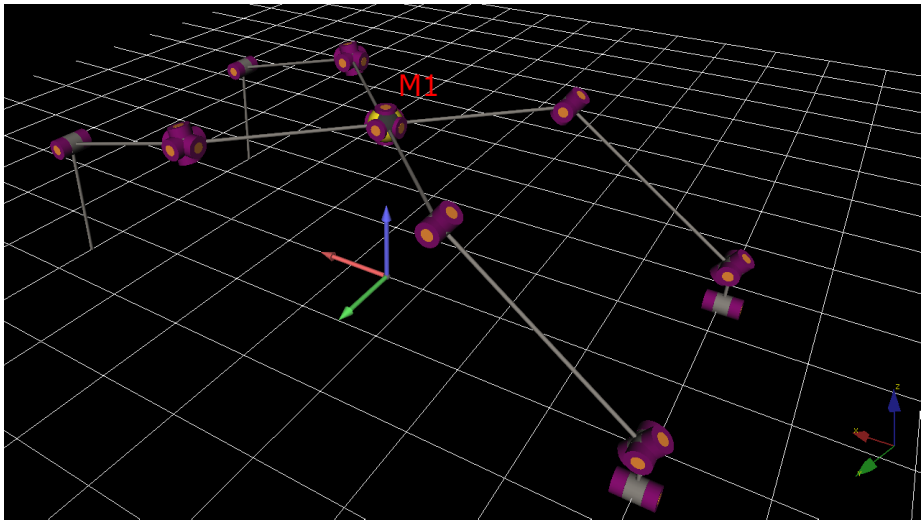

Figure 6-6: Rigid body modelling the cabin in MOBILE.

6.2.2 Left and right leg mass properties

The thigh part of both legs was modelled in Solidworks as a solid cylinder with 0.2m diameter and 2.0m height of the before-mentioned material (see Figure 6-7). Then the mass properties of that model were extracted and can be seen in Figure 6-8. The center of mass and the output coordinate system used by Solidworks for the mass properties are the same.

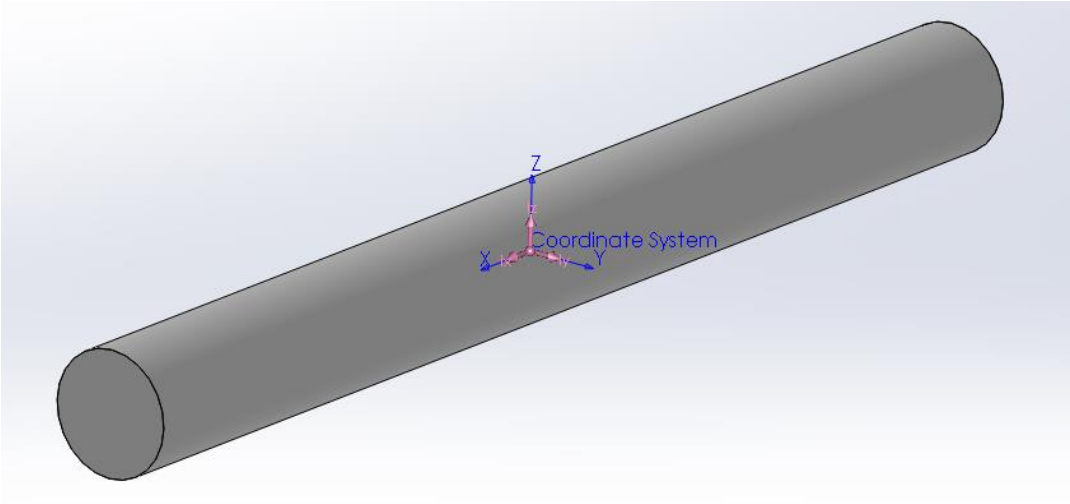

Figure 6-7: Solidworks model of thigh for both legs. The pink system of coordinates denotes the center of mass while the blue one is the output coordinate system for Solidworks' mass properties calculations.

Density = 2000.00 kilograms per cubic meter

Mass = 125.66 kilograms

Volume = 0.06 cubic meters

Surface area = 1.32 square meters

Center of mass: (meters)
X = 0.00
Y = 0.00
Z = 0.00

Principal axes of inertia and principal moments of inertia: (kilograms * square meters)
Taken at the center of mass.

Ix = (1.00, 0.00, 0.00)	Px = 0.63
Iy = (0.00, 1.00, 0.00)	Py = 42.20
Iz = (0.00, 0.00, 1.00)	Pz = 42.20

Moments of inertia: (kilograms * square meters)
Taken at the center of mass and aligned with the output coordinate system.

Lxx = 0.63	Lxy = 0.00	Lxz = 0.00
Lyx = 0.00	Lyy = 42.20	Lyz = 0.00
Lzx = 0.00	Lzy = 0.00	Lzz = 42.20

Figure 6-8: Mass properties of the thighs of both legs.

Next, we insert two mass objects with the above mass and inertia matrix to MOBILE in order to model left and right leg thigh mass as M3 and M2 accordingly (see Figure 6-9). The rigid bodies' center of mass is located at the midpoint between the hip and the knee of the kinematical chain for both legs.

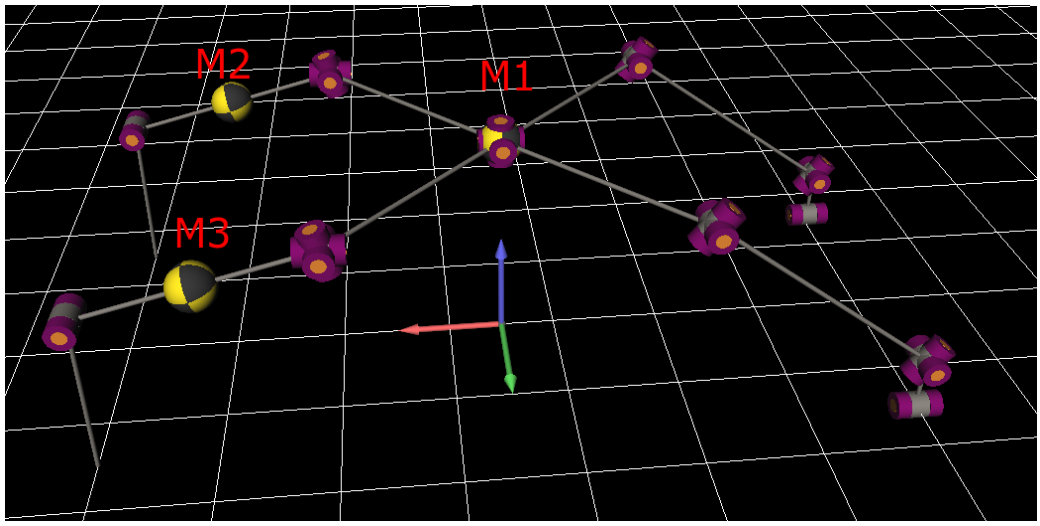

Figure 6-9: Rigid bodies modelling the left and right leg thighs in MOBILE.

The part between the knee and the ankle for both legs was modelled in Solidworks as a solid cylinder with 0.2m diameter and 1.5m height of the before-mentioned material (see Figure 6-10). Then the mass properties of that model were extracted and can be seen in Figure 6-11. The center of mass and the output coordinate system used by Solidworks for the mass properties are the same.

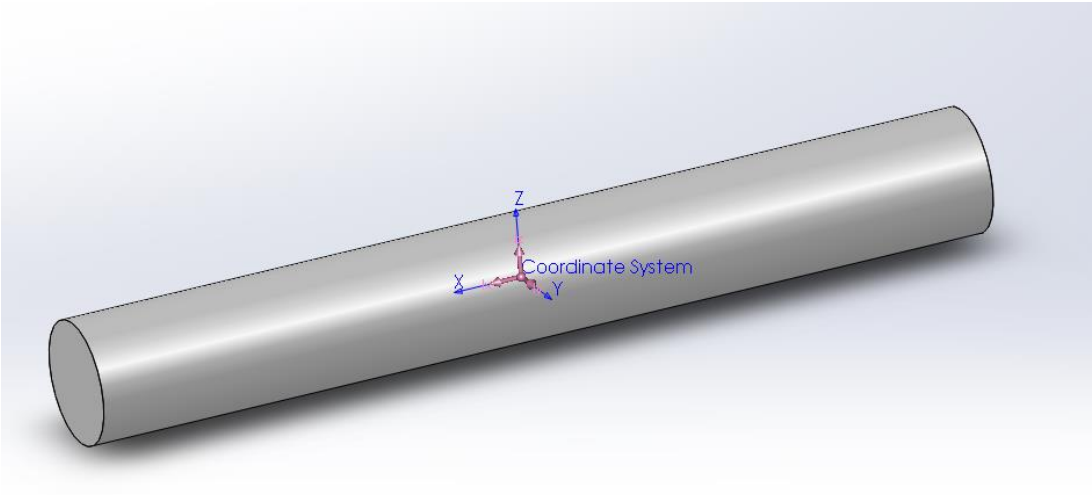

Figure 6-10: Solidworks model of the rigid body between the knee and the ankle for both legs. The pink system of coordinates denotes the center of mass while the blue one the output coordinate system for Solidworks' mass properties calculations.

Density = 2000.00 kilograms per cubic meter

Mass = 94.25 kilograms

Volume = 0.05 cubic meters

Surface area = 1.01 square meters

Center of mass: (meters)
X = 0.00
Y = 0.00
Z = 0.00

Principal axes of inertia and principal moments of inertia: (kilograms * square meters)
Taken at the center of mass.

Ix = (1.00, 0.00, 0.00)	Px = 0.47
Iy = (0.00, 1.00, 0.00)	Py = 17.91
Iz = (0.00, 0.00, 1.00)	Pz = 17.91

Moments of inertia: (kilograms * square meters)
Taken at the center of mass and aligned with the output coordinate system.

Lxx = 0.47	Lxy = 0.00	Lxz = 0.00
Lyx = 0.00	Lyy = 17.91	Lyz = 0.00
Lzx = 0.00	Lzy = 0.00	Lzz = 17.91

Figure 6-11: Mass properties of the rigid bodies between the knees and the ankles of both legs

Next, we insert two mass objects with the above mass and inertia matrix to MOBILE in order to model left and right leg rigid bodies between the knees and the ankles, as M5 and M4 accordingly (see Figure 6-12). The rigid bodies' center of mass is located at the midpoint between the knee and the ankle of the kinematical chain for both legs.

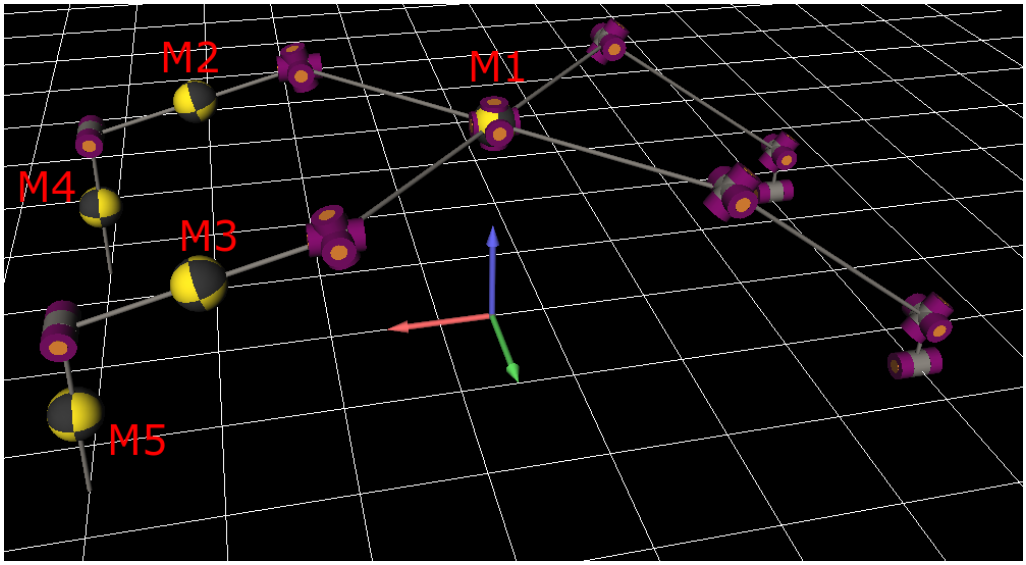

Figure 6-12: Rigid bodies modelling the left and right leg parts between the knees and the ankles in MOBILE.

6.2.3 Left and right wheel lever and wheel mass properties

Because the wheel's mass is so much smaller that the other masses of the mechanism the whole lever wheel and wheel system was modelled as one rigid body at the wheel lever. The wheel lever was modelled at Solidworks as a solid cylinder with 0.2m diameter and 2.5m height of the before-mentioned material (see Figure 6-13). Then the mass properties of that model were extracted and can be seen in Figure 6-14. The center of mass and the output coordinate system used by Solidworks for the mass properties are the same.

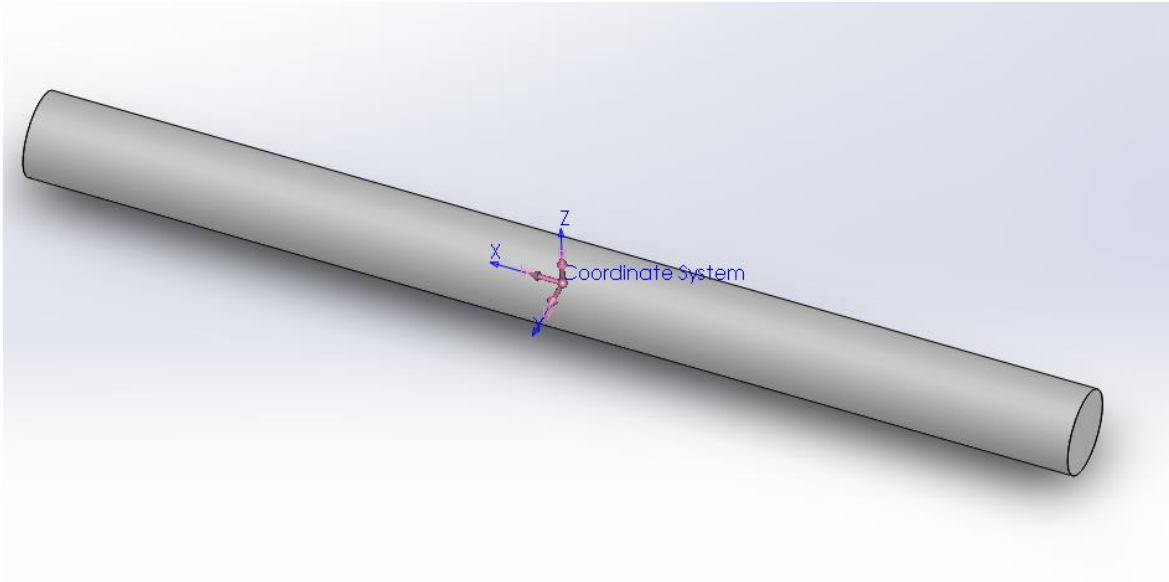

Figure 6-13: Solidworks model for each of the wheel lever and wheel system. The pink system of coordinates denotes the center of mass while the blue one is the output coordinate system for Solidworks' mass properties calculations.

```
Density = 2000.00 kilograms per cubic meter

Mass = 157.08 kilograms

Volume = 0.08 cubic meters

Surface area = 1.63 square meters

Center of mass: ( meters )
    X = 0.00
    Y = 0.00
    Z = 0.00

Principal axes of inertia and principal moments of inertia: ( kilograms * square meters )
Taken at the center of mass.
     Ix = ( 1.00,  0.00,  0.00)     Px = 0.79
     Iy = ( 0.00,  1.00,  0.00)     Py = 82.21
     Iz = ( 0.00,  0.00,  1.00)     Pz = 82.21

Moments of inertia: ( kilograms * square meters )
Taken at the center of mass and aligned with the output coordinate system.
     Lxx = 0.79          Lxy = 0.00          Lxz = 0.00
     Lyx = 0.00          Lyy = 82.21         Lyz = 0.00
     Lzx = 0.00          Lzy = 0.00          Lzz = 82.21
```

Figure 6-14: Mass properties for each wheel lever and wheel system of the mechanism.

Next, we insert two mass objects with the above mass and inertia matrix to MOBILE in order to model left and right wheel lever and wheel system mass, as M6 and M7 accordingly (see Figure 6-15). The rigid bodies' center of mass is located at the midpoint of the wheel lever kinematical chain for both wheel systems.

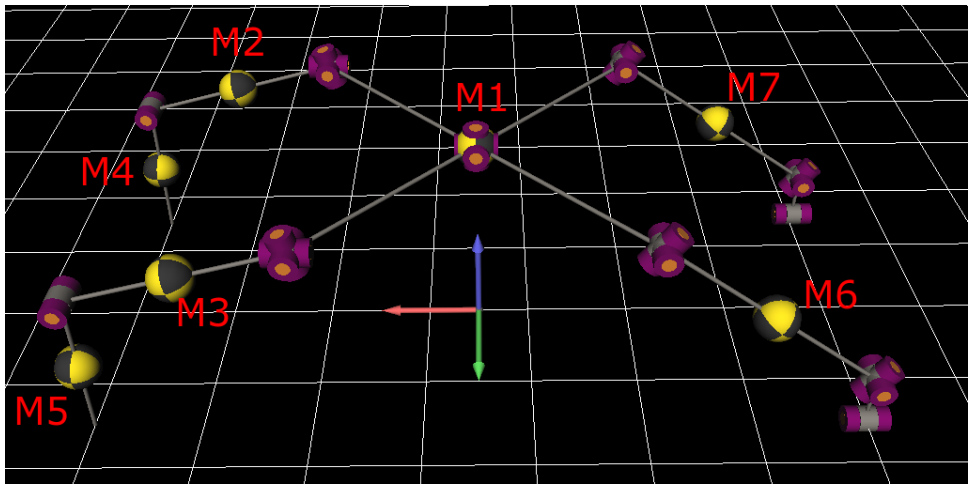

Figure 6-15: Rigid bodies modelling the left and wheel lever and wheel systems in MOBILE. All the rigid bodies of the system are now visual.

6.3 Inverse dynamics of the mechanism

6.3.1 Equations of motion

The inverse dynamics of the mechanism, with its n degrees of freedom and its closed kinematical topology, can be regarded as a function mapping the generalized coordinates q and their time derivatives to a set of generalized forces Q at the input of the kinetostatical skeleton. The equations of motion for the hybrid excavator can be written as,

$$-M\ddot{\underline{q}} - \underline{b}(q,\dot{q}) + Q_q^{External} + J_\beta^T \underline{\tau}_\beta = \underline{Q}_q \tag{6.10}$$

where M is the nxn mass matrix of the mechanism, b is the n-dimensional vector containing the centripetal and Coriolis terms, $Q_q^{External}$ is an n-dimensional vector containing the projection of the gravitational forces and external forces on the generalized coordinates, $J_\beta^T \tau_\beta$ is the contribution of the dependent variables to the generalized forces and Q_q is an n-dimensional vector collecting the generalized forces, these are usually referred to as "residual forces".

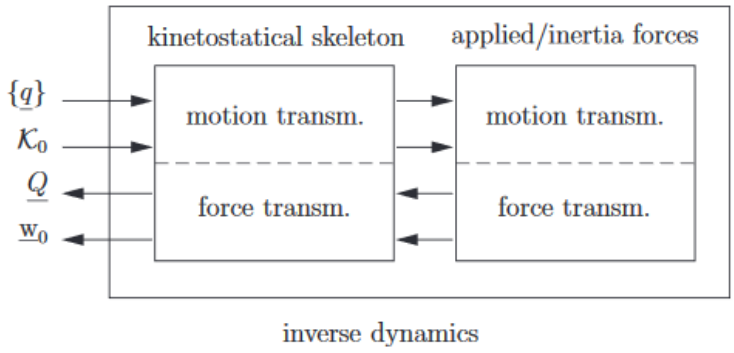

Figure 6-16: Model of the inverse dynamics of a multibody system.

The generalized variables q of Equation (6.10) are the degrees of freedom that the operator will be able to command regarding the vehicle's movement. These are selected to be the next four virtual joint variables (these can be seen in Figure 3-1):

$$q = \left[z_C, x_C, \varphi_{roll}, \varphi_{pitch} \right]^T \tag{6.11}$$

The dependent variables β of Equation (6.10) are selected to be the kinematically dependent real joint variables of the robot (these can also be seen on Figure 3-1):

$$\beta = \left[\psi_{l2}, \psi_{l3}, \psi_{l4}, \psi_{r2}, \psi_{r3}, \psi_{r4}, \theta_{l6}, \theta_{r6} \right]^T \tag{6.12}$$

6.3.2 Jacobian calculation

To solve the equations of motion we will need the Jacobian that connects $\dot{q}$ and $\dot{\beta}$, the derivatives of the above variables. We will do that by using MOBILE's kinematical capabilities. As mentioned in Section 4.1.2, there are two options, a force-based Jacobian determination and a velocity-based Jacobian determination. It is important to be able to select between these depending on the situation as choosing the better can simplify a lot the calculations and the computational cost.

In the force-based method you construct the Jacobian matrix row by row while in the velocity-based method you construct the Jacobian matrix column by column. Hence, the choice depends on which of the above is smaller, the rows or the columns of the output Jacobian matrix. If the matrix is square there is no difference.

In our case the Jacobian connects our variables as shown in Equation (6.13):

$$\dot{\beta} = J_{\beta,8x4}\dot{q} \tag{6.13}$$

As a result, this Jacobian has 8 rows and 4 columns and therefore the velocity-based method would be a better choice. As stated in Section 4.1.2, that is done by setting all velocity components at the input of the transmission element besides the jth-one equal to zero, and the jth-one equal to one. This yields an output velocity vector which is identical to the jth-column of the Jacobian. By this procedure we calculate the required Jacobian at every time step.

6.3.3 Calculation of torques at dependent joints

The next step is to calculate the torques of the dependent joints β, so that the residual forces Q_q are zero. This is necessary because the joint variables q are virtual and consequently their forces must be zero. As a result of the above, Equation (6.10) becomes:

$$\begin{aligned} &-M\underline{\ddot{q}} - \underline{b}(q,\dot{q}) + Q_q^{External} + J_\beta^{\mathrm{T}}\underline{\tau}_\beta = 0 \Rightarrow \\ &J_\beta^{\mathrm{T}}\underline{\tau}_\beta = M\underline{\ddot{q}} + \underline{b}(q,\dot{q}) - Q_q^{External} \Rightarrow \\ &J_\beta^{\mathrm{T}}\underline{\tau}_\beta = \underline{Q}^* \\ &\textit{where,} \\ &\underline{Q}^* = M\underline{\ddot{q}} + \underline{b}(q,\dot{q}) - Q_q^{External} \end{aligned} \tag{6.14}$$

Hence, in order to calculate the required τ_β we need to calculate the Q^* term. We will do that by using MOBILE's capabilities. From Equation (6.10) we see that if we set τ_β to zero, Q^* is then equal to Q_q. Therefore, by setting the dependent forces τ_β to zero instantaneously at every time step and then reading the generalized forces Q_q we have $-Q^*$ at every time step.

Finally, we are ready to solve the system (6.14). Let's consider the next system of linear equations:

$$y = Hx \tag{6.15}$$

Frequently, when matrix H has more columns than rows with linearly independent rows the system is "underdetermined". In this case, it is common to seek a solution x with minimum norm. In other words, we would like to solve the following optimization problem:

$$\min_x \|x\|_2^2 \tag{6.16}$$

such that y=Hx. Minimization with constraints can be done with Langrange multipliers. Following we have the definition of the Langrangian:

$$L(x,\mu) = \|x\|_2^2 + \mu^{\mathrm{T}}(y - Hx) \tag{6.17}$$

Next, we take the derivatives of the Lagrangian:

$$\begin{aligned} &\frac{\partial}{\partial x}L(x) = 2x - H^T\mu \\ &\frac{\partial}{\partial \mu}L(x) = y - Hx \end{aligned} \tag{6.18}$$

Setting the derivatives to zero we get:

$$x = \frac{1}{2} H^T \mu \tag{6.19}$$

$$y = Hx \tag{6.20}$$

Plugging (6.19) into (6.20) gives:

$$y = \frac{1}{2} H H^T \mu \tag{6.21}$$

Assuming that HH^T is invertible, we have:

$$\mu = 2\left(\mathrm{HH^T}\right)^{-1} y \tag{6.22}$$

Plugging (6.22) into (6.19) gives the least squares solution:

$$x = H^T \left(HH^T\right)^{-1} y \tag{6.23}$$

Next we verify that x in this formula does in fact satisfy y=Hx:

$$Hx = H\left|H^T \left(HH^T\right)^{-1} y\right| = \left(HH^T\right)\left(HH^T\right)^{-1} y = y \tag{6.24}$$

Hence,

$$\min_x \|x\|_2^2 \rightarrow s.t. \rightarrow y = Hx \Rightarrow x = H^T \left(HH^T\right)^{-1} y \tag{6.25}$$

In some situations, like ours, it is desirable to minimize the weighted sum, i.e.,

$$\sum\nolimits_n w_n x_n^2 \tag{6.26}$$

where w_n are positive weights. This corresponds to minimizing,

$$\left\|W^{1/2} x\right\|_2^2 \tag{6.27}$$

where W is the diagonal matrix,

$$\left[W\right]_{n,n} = w_n \tag{6.28}$$

The derivation of the solution is similar, and gives:

$$\min_x \left\|W^{1/2} x\right\|_2^2 \rightarrow s.t. \rightarrow y = Hx \Rightarrow x = W^{-1} H^T \left(HW^{-1} H^T\right)^{-1} y \tag{6.29}$$

In our case we are using the weighted problem because we want to be able to determine the involvement of each actuated joint to the mechanism's movements. To apply the above to our problem we just replace y with Q^*, x with τ_β and H with J_β^T. Then the problem and the solution in our case is:

$$\min_x \left\|W^{1/2} \tau_\beta\right\|_2^2 \rightarrow s.t. \rightarrow Q^* = J_\beta^T \tau_\beta \Rightarrow \tau_\beta = W^{-1} J_\beta \left(J_\beta^T W^{-1} J_\beta\right)^{-1} Q^* \tag{6.30}$$

The above Equation (6.30) is then solved for each time step in MATLAB.

There are a lot of ways to approach the choice of the weights, two of them are the next:

- Divide the actuators as strong and weak, hence deciding which actuators will take bigger part in the movements of the mechanism. Small weights should be put to the strong actuators and large weights to the weak actuators.
- By the contribution of each actuator force to Q^*. If the contribution is big the weight should be small and vice versa. So,

$$W_i = \frac{1}{\left\| ith\ row\ of\ J_\beta \right\|} \tag{6.31}$$

6.4 Simulation results

6.4.1 1st Inverse dynamics simulation

For the first simulation we impose to the robot a forward velocity of 3 m/s ($\dot{x}_C = 3m/s$). In Figure 6-18 the variables of the system are reminded, in Figure 6-17 the start and end position of the robot can be seen, in Figure 6-19 the dimensions of the robot are visualized and in Table 6-2 the initial conditions, geometry and mass values of the system can be seen.

We begin by using the strong-weak actuator method for weights. The weight choice can be seen in Equation (6.32). These weights from upper left to downright they affect the torques of the actuated joints in Equation (6.33) accordingly.

$$W = \begin{bmatrix} 0.01 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 100 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.01 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 100 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.01 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.01 \end{bmatrix} \tag{6.32}$$

$$\beta = \left[\psi_{l2}, \psi_{l3}, \psi_{l4}, \psi_{r2}, \psi_{r3}, \psi_{r4}, \theta_{l6}, \theta_{r6}\right]^T \tag{6.33}$$

Hence, joints $\psi_{l2}, \psi_{r2}, \theta_{l6}, \theta_{r6}$ were chosen to handle most of the loads while ψ_{l3}, ψ_{r3} the least. Joints ψ_{l4}, ψ_{r4} are between the before mentioned. The robot begins its motion from $x_C = 0m$ and stops at $x_C = 1.482m$.

In Table 6-1, the total weight of the robot can be seen along with the approximate total length, the approximate total width and the approximate total height (these varie in relation to the legs' and wheel levers' extension). These apply to all the next dynamic simulations.

Table 6-1: Total weight and approximate total length, width and height of the robot.

Total weight (kg)	Total length (m)	Total width (m)	Total height (m)
6153.98	~7	~3	~(0, 2.6]

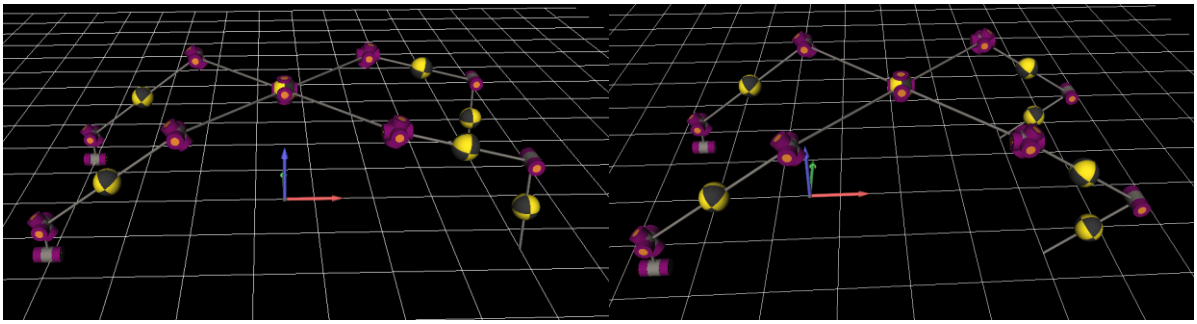

Figure 6-17: 1st Simulation start (left) and end (right) robot pose.

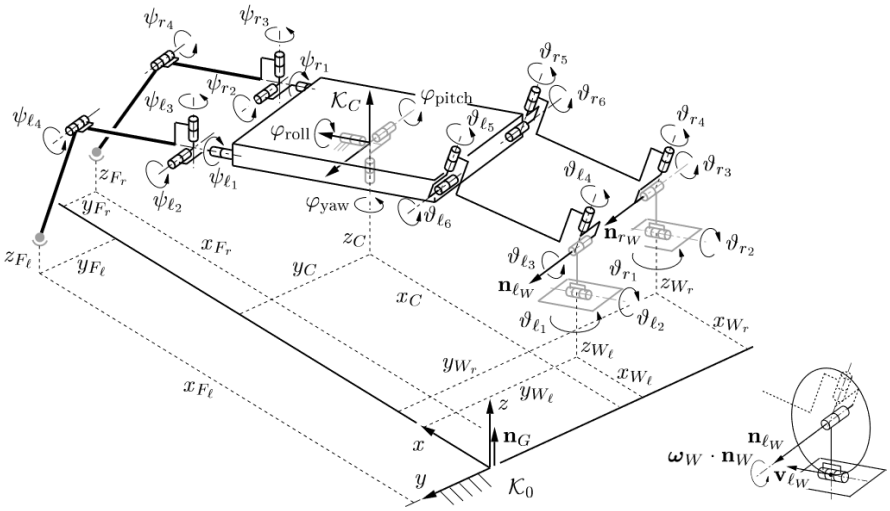

Figure 6-18: Robot's variables.

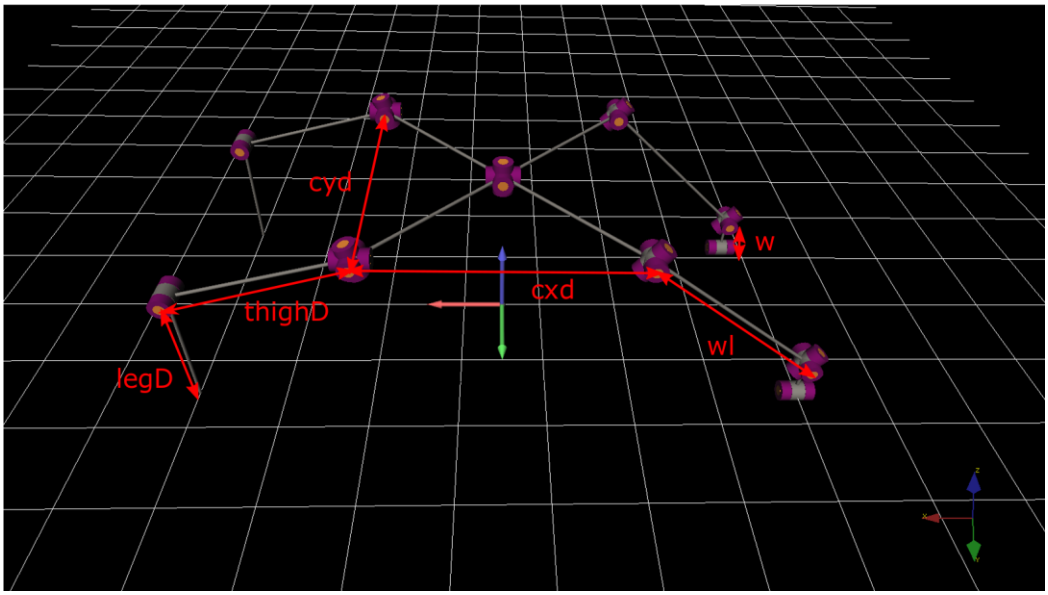

Figure 6-19: Dimensions of the robot. The whole system is symmetrical.

Table 6-2: Initial conditions, geometry variables and mass properties of the robot for the 1st inverse dynamics simulation.

x_C (m)	0.0	dx_C (m/s)	3.0	cxd (m)	3.0
y_C (m)	0.0	-	-	cyd (m)	3.0

z_C (m)	2.0	dz_C (m/s)	0.0	w (m)	0.5
x_{Fl} (m)	3.5	dx_{Fl} (m/s)	0.0	wl (m)	2.5
y_{Fl} (m)	1.5	dy_{Fl} (m/s)	0.0	thighD (m)	2.0
z_{Fl} (m)	0.0	dz_{Fl} (m/s)	0.0	legD (m)	1.5
x_{Fr} (m)	3.5	dx_{Fr} (m/s)	0.0	M1 (kg)	5400
y_{Fr} (m)	-1.5	dy_{Fr} (m/s)	0.0	M2 (kg)	125.66
z_{Fr} (m)	0.0	dz_{Fr} (m/s)	0.0	M3 (kg)	125.66
z_{Wl} (m)	0.0	dz_{Wl} (m/s)	0.0	M4 (kg)	94.25
z_{Wr} (m)	0.05	dz_{Wr} (m/s)	0.0	M5 (kg)	94.25
φ_{yaw} (deg)	0.0	-	-	M6 (kg)	157.08
φ_{pitch} (deg)	0.0	$d\varphi_{pitch}$ (deg/s)	0.0	M7 (kg)	157.08
φ_{roll} (deg)	0.0	$d\varphi_{roll}$ (deg/s)	0.0		
ψ_{l1} (deg)	0.0	$d\psi_{l1}$ (deg/s)	0.0		
ψ_{r1} (deg)	0.0	$d\psi_{r1}$ (deg/s)	0.0		
θ_{l5} (deg)	0.0	$d\theta_{l5}$ (deg/s)	0.0		
θ_{r5} (deg)	0.0	$d\theta_{r5}$ (deg/s)	0.0		

The masses M1 to M7 are visualized in Figure 6-15 and the mass properties are shown in Figure 6-5, Figure 6-8, Figure 6-11 and Figure 6-14 accordingly. Also, the acceleration of gravity is set at approximately 9.81 m/s^2. In Figure 6-20 and Figure 6-21 the torques of the actuated joints β, needed for the cabin x movement with the first weight method, are shown.

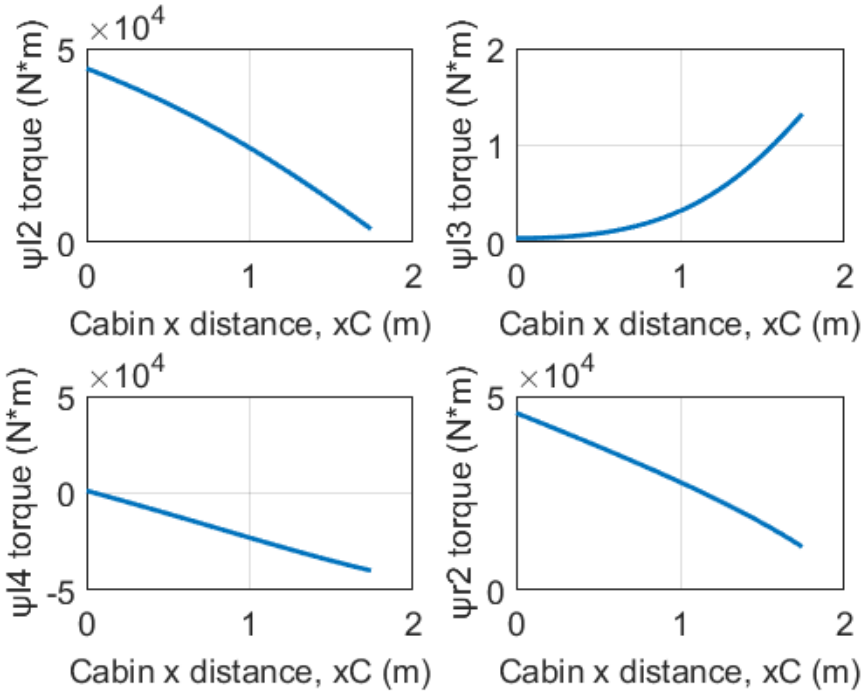

Figure 6-20: ψl2, ψl3, ψl4 and ψr2 torques needed for a cabin x movement of 1.482 m using the first weight method.

We observe from Figure 6-20 and Figure 6-21 that the torque distribution is how we planned it with the weights. Also, the direction of the torques seems right if we look the start and end pose of the robot (Figure 6-17).

Now we are going to repeat the above simulation but with different weights. The weight matrix is going to be calculated by the contribution of each actuator force to Q^* (see Equation (6.31), so it is changing for every time step. In Figure 6-22 and Figure 6-23 the torques of the actuated joints β, needed for the cabin x movement with the second weight method, are shown.

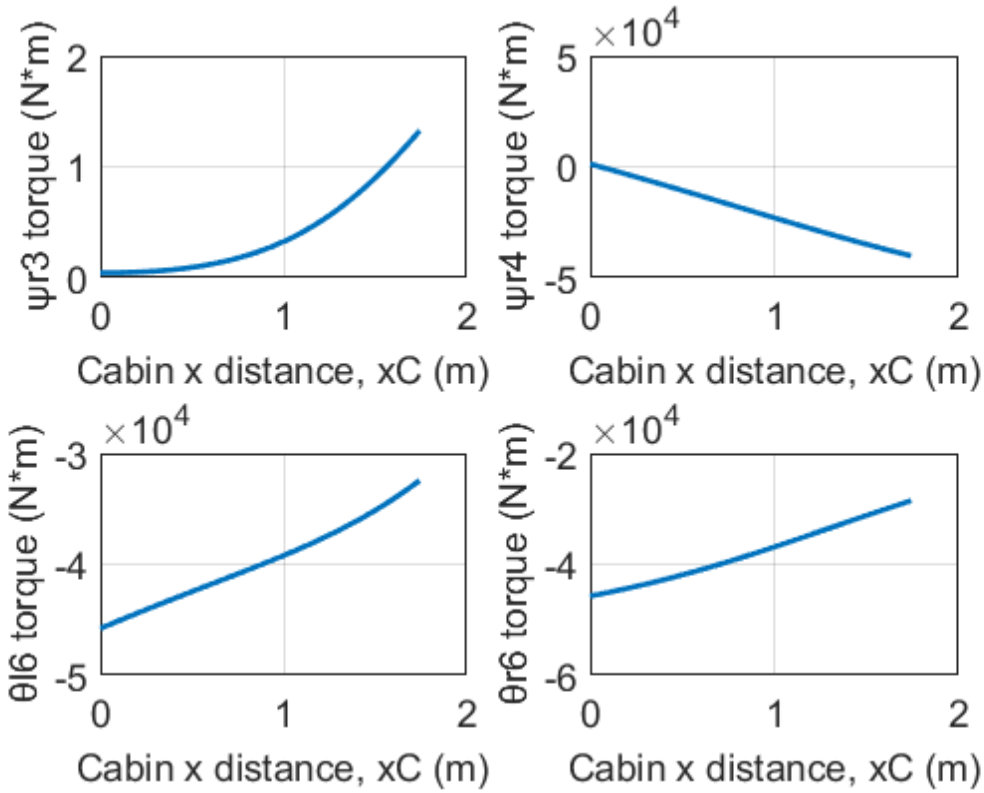

Figure 6-21: ψr3, ψr4, θl6 and θr6 torques needed for a cabin x movement of 1.482 m using the first weight method.

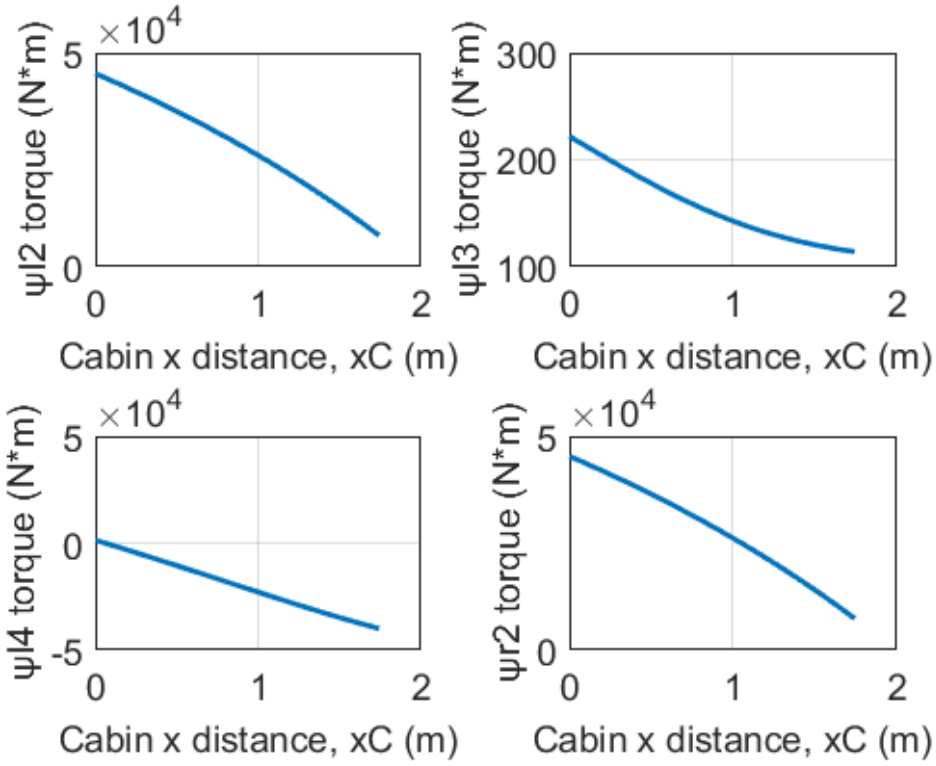

Figure 6-22: ψl2, ψl3, ψl4 and ψr2 torques needed for a cabin x movement of 1.482 m using the second weight method.

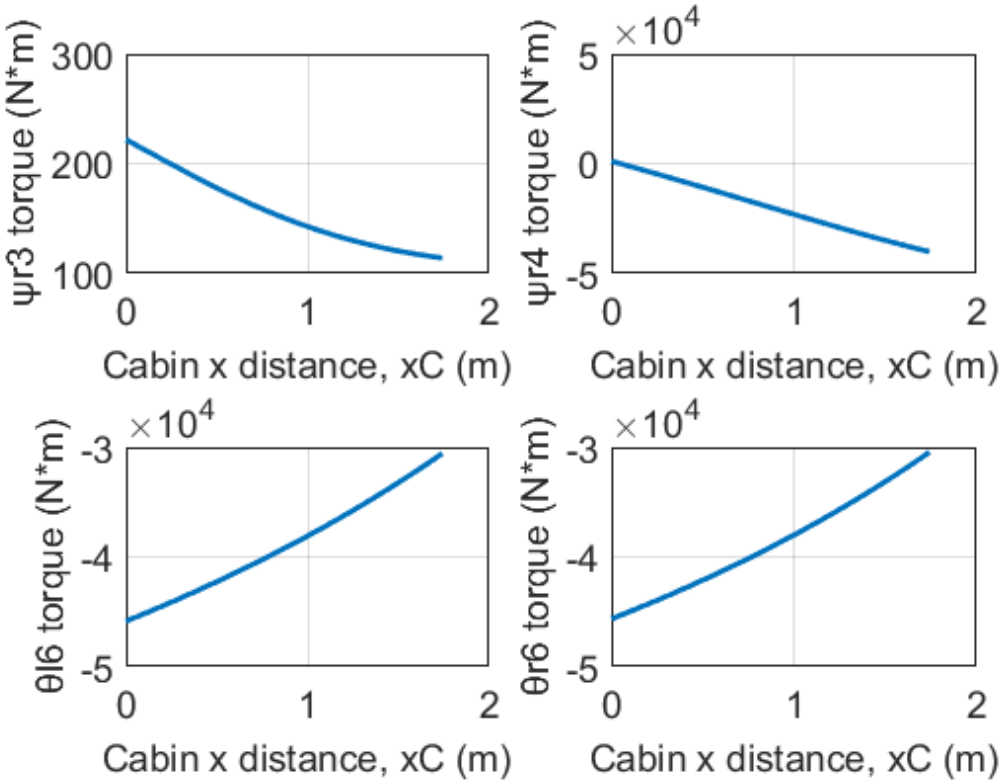

Figure 6-23: ψr3, ψr4, θl6 and θr6 torques needed for a cabin x movement of 1.482 m using the second weight method.

We observe that the two methods for the weights are quite close in terms of results except for the torques of ψ_{l3} and ψ_{r3}.

We also observe from Figure 6-20, Figure 6-21, Figure 6-22 and Figure 6-23 that the torque values are of magnitude 10^4 Nm. We are going to test this by comparing to the static ψ_2 torque of the 2D problem seen in Figure 6-24.

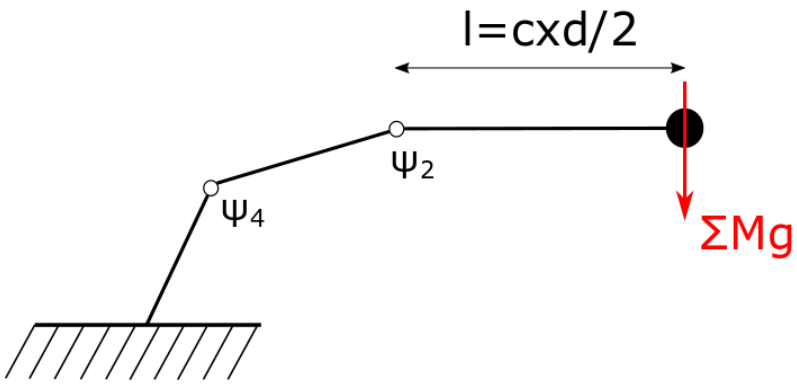

Figure 6-24: Static torque of ψ_2 to check the torque magnitude of the simulation results.

In Figure 6-24, ΣM is the total weight of the robot and g the gravitational acceleration. Hence, the static torque of joint ψ_2 is:

$$\tau_{\psi_2} = l \times \frac{mg}{2} = \frac{cxd}{2} \times \frac{mg}{2} \Rightarrow$$
$$\tau_{\psi_2} = \frac{3}{2} \times \frac{6153.98 \cdot 9.81}{2} = 45277.91\ N \cdot m = 4.53 \times 10^4\ N \cdot m \quad (6.34)$$

Thus, we can see from Equation (6.34), that the order of magnitude of the simulation torques are valid.

6.4.2 2nd Inverse dynamics simulation

For the second simulation we impose to the robot a cabin roll velocity of -10 deg/s ($\dot{\varphi}_{roll} = -10 \text{deg}/s$). In Figure 6-25 the start and end position of the robot can be seen, the variables of the system are reminded in Figure 6-18, the dimensions of the robot are the same as before and are visualized in Figure 6-19 and in Table 6-3 the initial conditions, geometry and mass values of the system can be seen.

Like before, we begin by using the strong-weak actuator method for weights. The weight choice can be seen in Equation (6.32). These weights from upper left to downright they affect the torques of the actuated joints in Equation (6.33) accordingly. Hence, joints $\psi_{l2}, \psi_{r2}, \theta_{l6}, \theta_{r6}$ were chosen to handle most of the loads while ψ_{l3}, ψ_{r3} the least. Joints ψ_{l4}, ψ_{r4} are between the before mentioned. The robot begins its motion from $\varphi_{roll} = 0^o$ and stops at $\varphi_{roll} = -5.41^o$.

The masses M1 to M7 are visualized in Figure 6-15 and the mass properties are shown in Figure 6-5, Figure 6-8, Figure 6-11 and Figure 6-14 accordingly. Also, the acceleration of gravity is set at approximately 9.81 m/s^2. In Figure 6-26 and Figure 6-27 and the torques of

the actuated joints β, needed for the cabin roll movement with the first weight method, are shown.

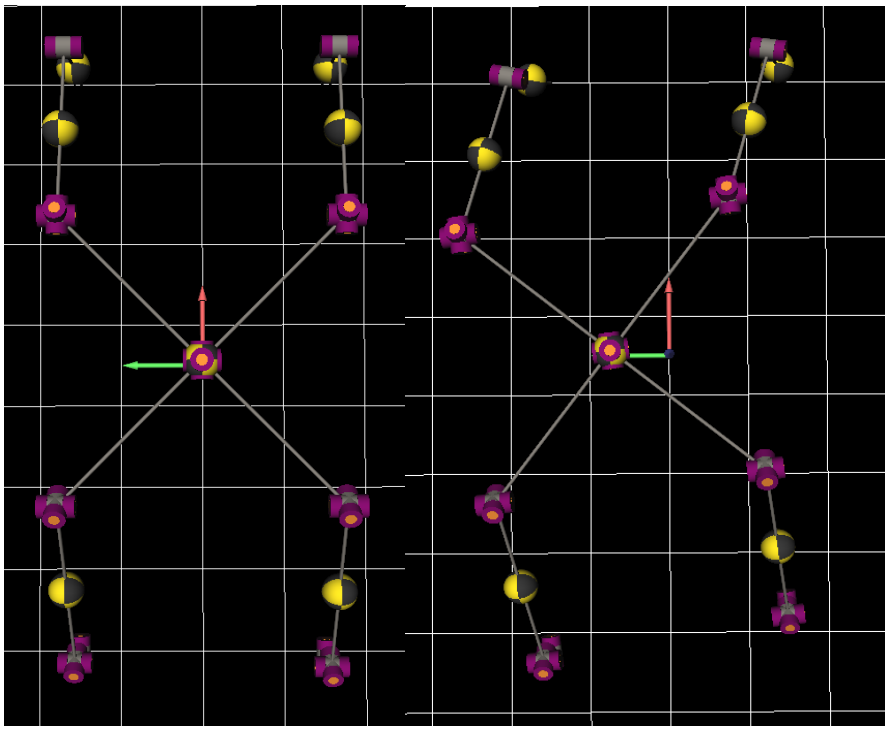

Figure 6-25: 2nd Simulation start (left) and end (right) robot pose.

Table 6-3: Initial conditions, geometry variables and mass properties of the robot for the 2nd inverse dynamics simulation.

x_C (m)	0.0	dx_C (m/s)	0.0	cxd (m)	3.0
y_C (m)	0.0	-	-	cyd (m)	3.0
z_C (m)	2.0	dz_C (m/s)	0.0	w (m)	0.5
x_{Fl} (m)	3.5	dx_{Fl} (m/s)	0.0	wl (m)	2.5
y_{Fl} (m)	1.5	dy_{Fl} (m/s)	0.0	thighD (m)	2.0
z_{Fl} (m)	0.0	dz_{Fl} (m/s)	0.0	legD (m)	1.5
x_{Fr} (m)	3.5	dx_{Fr} (m/s)	0.0	M1 (kg)	5400
y_{Fr} (m)	-1.5	dy_{Fr} (m/s)	0.0	M2 (kg)	125.66
z_{Fr} (m)	0.0	dz_{Fr} (m/s)	0.0	M3 (kg)	125.66
z_{Wl} (m)	0.0	dz_{Wl} (m/s)	0.0	M4 (kg)	94.25
z_{Wr} (m)	0.05	dz_{Wr} (m/s)	0.0	M5 (kg)	94.25
φ_{yaw} (deg)	0.0	-	-	M6 (kg)	157.08
φ_{pitch} (deg)	0.0	$d\varphi_{pitch}$ (deg/s)	0.0	M7 (kg)	157.08
φ_{roll} (deg)	0.0	$d\varphi_{roll}$ (deg/s)	-10.0		
ψ_{l1} (deg)	0.0	$d\psi_{l1}$ (deg/s)	0.0		
ψ_{r1} (deg)	0.0	$d\psi_{r1}$ (deg/s)	0.0		
θ_{l5} (deg)	0.0	$d\theta_{l5}$ (deg/s)	0.0		

θ_{r5} (deg)	0.0	$d\theta_{r5}$ (deg/s)	0.0		

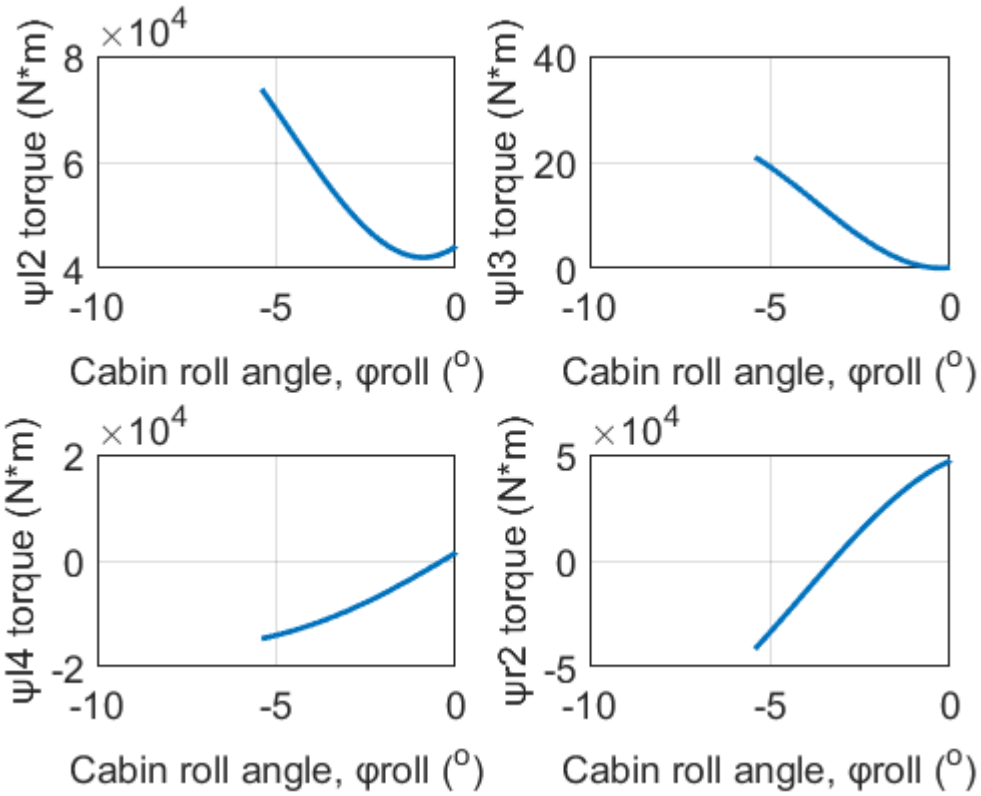

Figure 6-26: ψl2, ψl3, ψl4 and ψr2 torques needed for a cabin roll movement of -5.41° using the first weight method.

We observe from Figure 6-26 and Figure 6-27 that the torque distribution is how we planned it with the weights.

Now we are going to repeat the above simulation but with different weights. Like before, the weight matrix is going to be calculated by the contribution of each actuator force to Q^* (see Equation (6.31), so it is changing for every time step. In Figure 6-28 and Figure 6-29 the torques of the actuated joints β, needed for the cabin roll movement with the second weight method, are shown.

We observe that the two methods for the weights give quite different results and the second's method torque values seem to agree more with the movement of the robot.

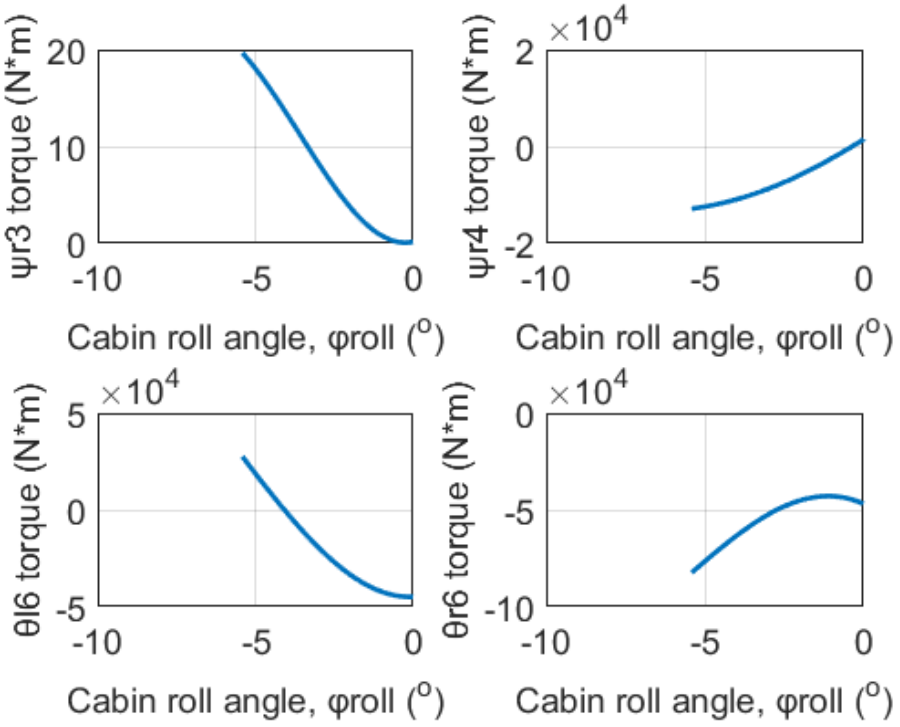

Figure 6-27: ψr3, ψr4, θl6 and θr6 torques needed for a cabin roll movement of -5.41° using the first weight method.

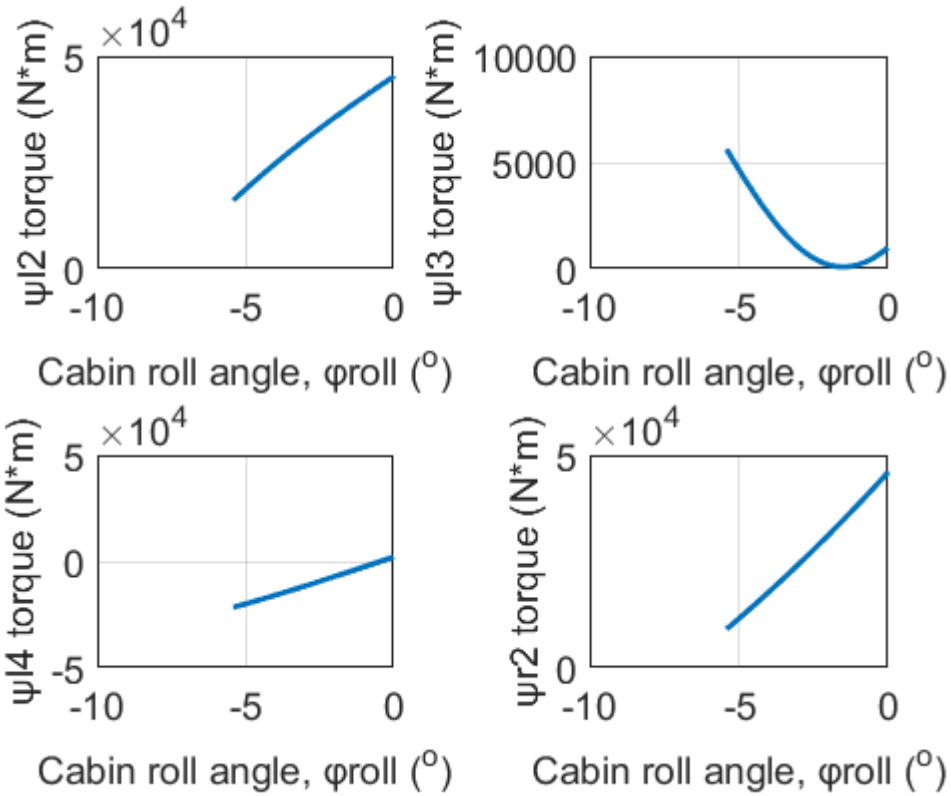

Figure 6-28: ψl2, ψl3, ψl4 and ψr2 torques needed for a cabin roll movement of -5.41° using the second weight method.

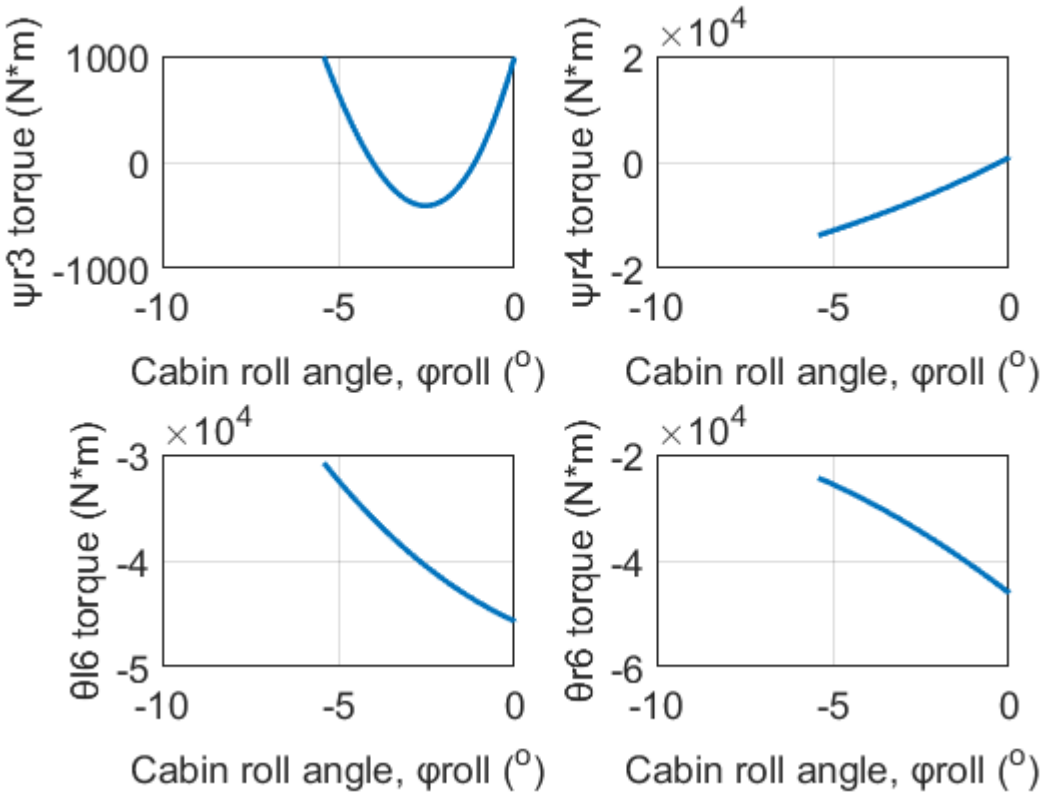

Figure 6-29: ψr3, ψr4, θl6 and θr6 torques needed for a cabin roll movement of -5.41° using the second weight method.

6.4.3 3rd Inverse dynamics simulation

For the third simulation we impose to the robot a cabin z velocity of -3 m/s ($\dot{z}_C = -3m/s$). In Figure 6-30 the start and end position of the robot can be seen, the variables of the system are reminded in Figure 6-18, the dimensions of the robot are the same as before and are visualized in Figure 6-19 and in Table 6-4 the initial conditions, geometry and mass values of the system can be seen.

Like before, we begin by using the strong-weak actuator method for weights. The weight choice can be seen in Equation (6.32). These weights from upper left to downright they affect the torques of the actuated joints in Equation (6.33) accordingly. Hence, joints $\psi_{l2}, \psi_{r2}, \theta_{l6}, \theta_{r6}$ were chosen to handle most of the loads while ψ_{l3}, ψ_{r3} the least. Joints ψ_{l4}, ψ_{r4} are between the before mentioned. The robot begins its motion from $z_C = 2.0m$ and stops at $z_C = 0.2570m$.

The masses M1 to M7 are visualized in Figure 6-15 and the mass properties are shown in Figure 6-5, Figure 6-8, Figure 6-11 and Figure 6-14 accordingly. Also, the acceleration of gravity is set at approximately 9.81 m/s^2. In Figure 6-31 and Figure 6-32 the torques of the actuated joints β, needed for cabin's z movement with the first weight method, are shown.

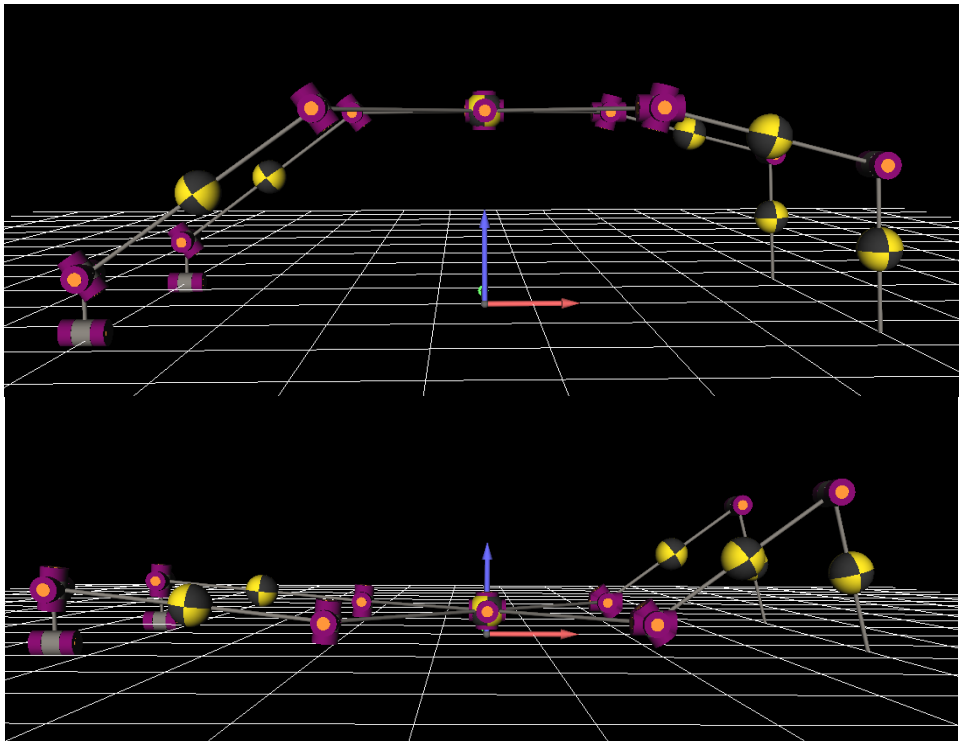

Figure 6-30: 3rd Simulation start (upper) and end (down) robot pose.

Table 6-4: Initial conditions, geometry variables and mass properties of the robot for the 3rd inverse dynamics simulation.

x_C (m)	0.0	dx_C (m/s)	0.0	cxd (m)	3.0
y_C (m)	0.0	-	-	cyd (m)	3.0
z_C (m)	2.0	dz_C (m/s)	-3.0	w (m)	0.5
x_{Fl} (m)	3.5	dx_{Fl} (m/s)	0.0	wl (m)	2.5
y_{Fl} (m)	1.5	dy_{Fl} (m/s)	0.0	thighD (m)	2.0
z_{Fl} (m)	0.0	dz_{Fl} (m/s)	0.0	legD (m)	1.5
x_{Fr} (m)	3.5	dx_{Fr} (m/s)	0.0	M1 (kg)	5400
y_{Fr} (m)	-1.5	dy_{Fr} (m/s)	0.0	M2 (kg)	125.66
z_{Fr} (m)	0.0	dz_{Fr} (m/s)	0.0	M3 (kg)	125.66
z_{Wl} (m)	0.0	dz_{Wl} (m/s)	0.0	M4 (kg)	94.25
z_{Wr} (m)	0.05	dz_{Wr} (m/s)	0.0	M5 (kg)	94.25
φ_{yaw} (deg)	0.0	-	-	M6 (kg)	157.08
φ_{pitch} (deg)	0.0	$d\varphi_{pitch}$ (deg/s)	0.0	M7 (kg)	157.08
φ_{roll} (deg)	0.0	$d\varphi_{roll}$ (deg/s)	0.0		
ψ_{l1} (deg)	0.0	$d\psi_{l1}$ (deg/s)	0.0		
ψ_{r1} (deg)	0.0	$d\psi_{r1}$ (deg/s)	0.0		

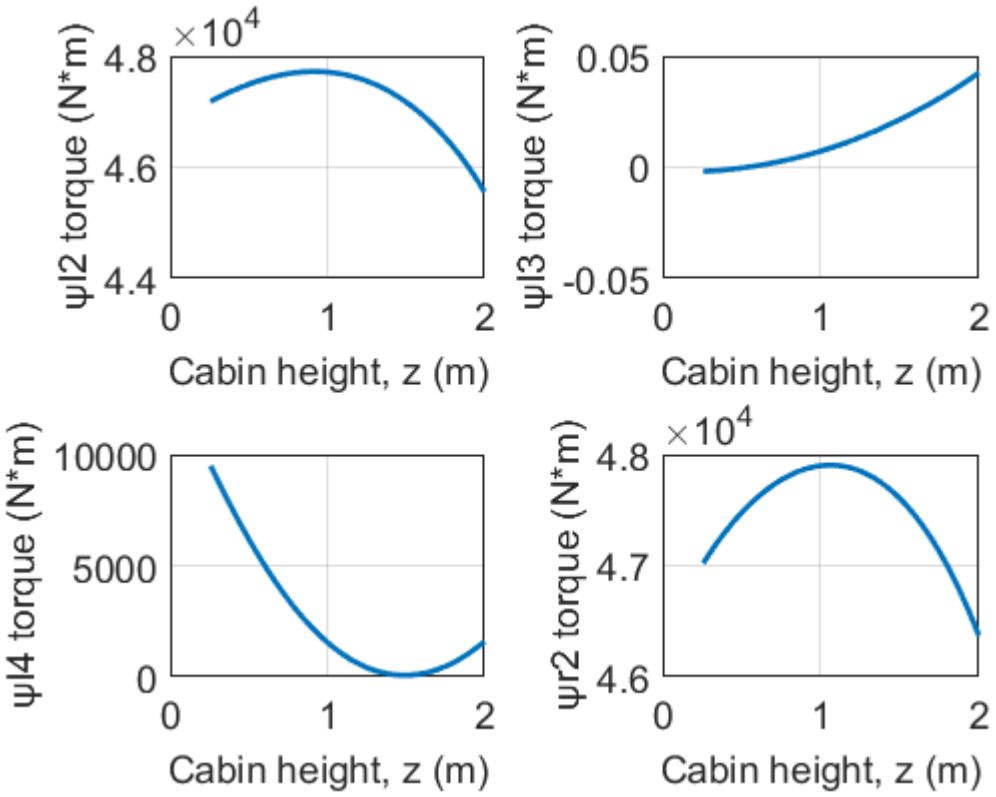

Figure 6-31: ψl2, ψl3, ψl4 and ψr2 torques needed for a cabin z movement of -1.743 m using the first weight method.

We observe from Figure 6-31 and Figure 6-32 that the torque distribution is how we planned it with the weights.

Now we are going to repeat the above simulation but with different weights. Like before, the weight matrix is going to be calculated by the contribution of each actuator force to Q^* (see Equation (6.31), so it is changing for every time step. In and the torques of the actuated joints β, needed for cabin's z movement with the second weight method, are shown.

We observe that the two methods for the weights are quite close in terms of results except for the torques of ψ_{l3} and ψ_{r3}.

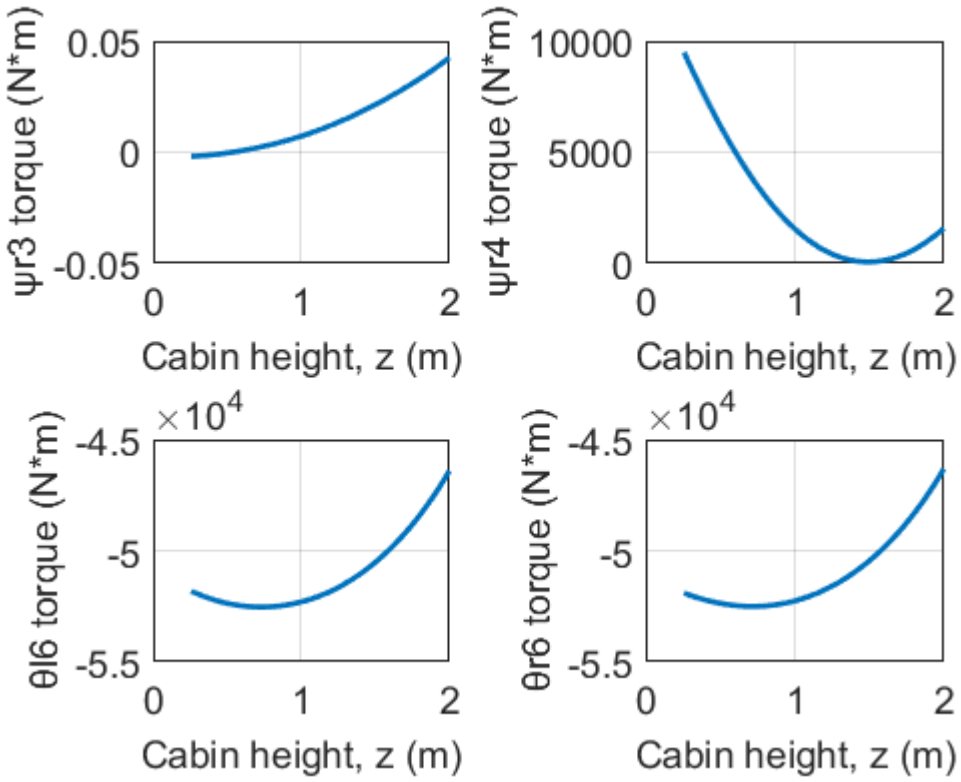

Figure 6-32: ψr3, ψr4, θl6 and θr6 torques needed for a cabin z movement of -1.743 m using the first weight method.

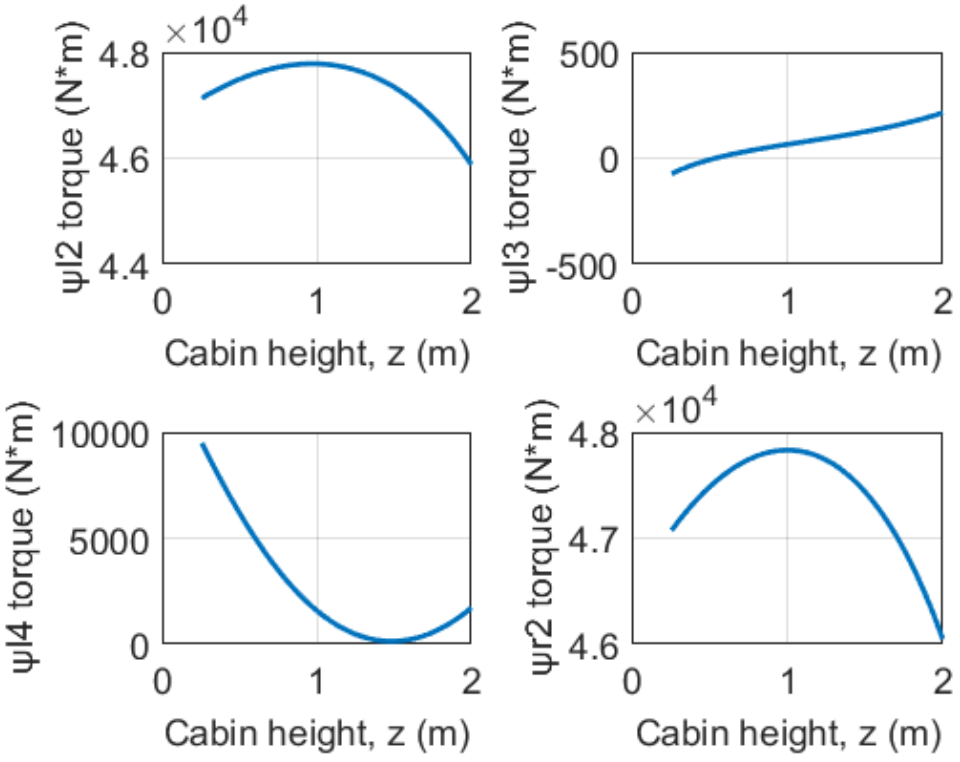

Figure 6-33: ψl2, ψl3, ψl4 and ψr2 torques needed for a cabin z movement of -1.743 m using the second weight method.

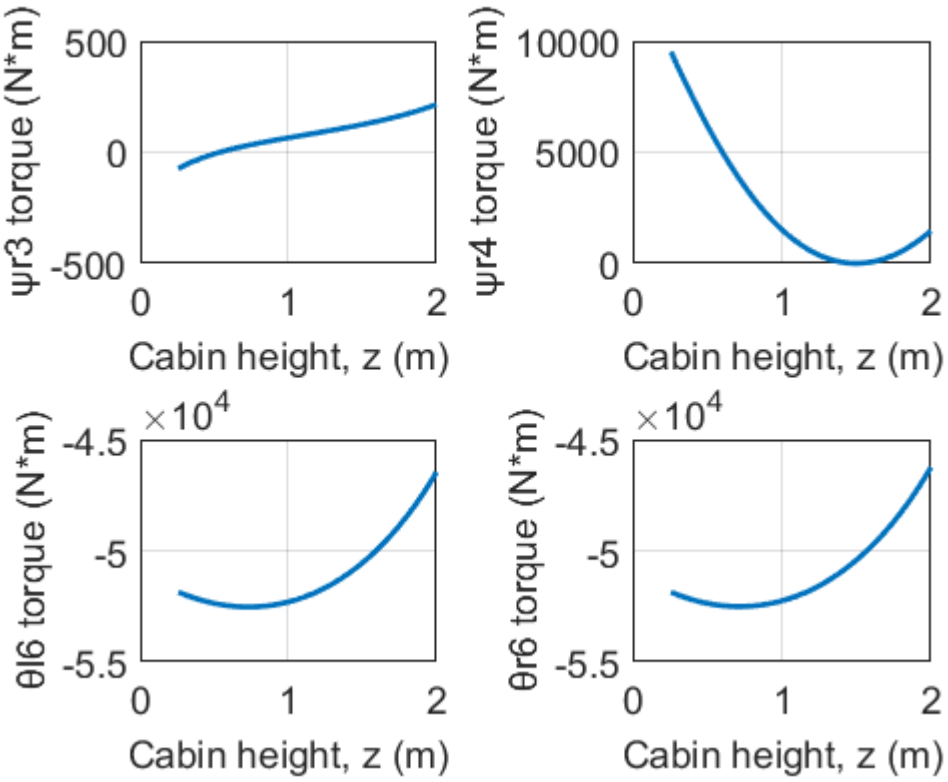

Figure 6-34: ψr3, ψr4, θl6 and θr6 torques needed for a cabin z movement of -1.743 m using the first weight method.

In this chapter we saw how mass properties and gravity were added to the MOBILE model and how we used MOBILE's iterative calculations capabilities along with MATLAB to be able to compute the robot's needed torques for certain movements. Two different choices for the weight matrix were made allowing us to decide how the torques are distributed to the robot. We conclude that there are differences in the actuator torques for the different weight matrices, which depend on the robot's movements, as the first method uses a fixed weight matrix while second method matrix changes in every time step.

7 Conclusions and Future Work

In this chapter the conclusions of the work described in this thesis are presented. Also recommendations for future work than can be done to further develop this project.

7.1 Conclusions

The aim of this research was to provide a better understanding of a non-holonomic excavator, which combines wheels and legs, through kinematic and dynamic modelling and simulation, making a step towards creating a working platform for the study of these hybrid robots' behaviour with the goal of tackling a big variety of applications in a more efficient manner than a wheeled or legged robot would.

Firstly, towards that goal, a basic kinematic model of the hybrid excavator was implemented in the object-oriented programming package, MOBILE. Next, the inverse kinematics of this model were built. This was done using MOBILE's iterative method by creating closed loops. This method was verified by the analytical equations describing the robot's kinematics. The verification process showed that MOBILE's iterative method values for the inverse kinematics are very close to the analytical solutions and therefore accurate and acceptable. The non-holonomic constraints were addressed and implemented into MOBILE model. The right implementation was verified analytically and the results showed that the MOBILE dependent velocities' values satisfy the non-holonomic constraints thus completing the kinematic model.

The robot's steering behaviour due to its non-holonomic constraints was studied for different choice of dependent variables. The results showed that different sets of dependent variables result different steering behaviours which were presented for the choices made. Finally, the dynamics of the robot was addressed. Mass properties and gravity were added to the MOBILE model. By using MOBILE's iterative capabilities and MATLAB, the inverse dynamics of the robot were completed thus enabling us to know the robot's actuator torques needed for certain movements. The simulations present the actuators' torques needed for some typical movements. Because we want to be able to decide the involvement of each actuated joint to the robot's movements, a weight matrix was used in the inverse dynamics solution. Two different choices for that matrix are presented and compared. The results show that there are differences in the actuator torques for different weight matrices and the extend of these differences depends on the robot's movements.

In conclusion, a "platform" is created for the kinematic and dynamic study of a non-holonomic wheeled-legged robot. It is essential to mention that this "platform" has been made parametric, meaning that every value regarding the robot can be easily altered to study different scenarios.

7.2 Future Work

There are many ways this thesis can further continue; some of these are going to be mentioned here.

Future research should examine ground contact impact and also implement control for the robot. It would be extremely interesting to compare all the MOBILE values with experimental results and check how well this simulation meets real life results. Future research also should address gait and stability analysis of the robot as it would be interesting to study which gate methods are best for moving and steering this wheeled-legged robot and why, and also to find the boundaries of its movements so it is always stable. Additionally, it would be useful to compare MOBILE methods for kinematics and dynamics with other similar programs in terms of computational speed and ease of setting up the models. Different combinations of geometries, masses and materials should be studied also to yield which is better for different goals. Also, different choices, than the ones studied, for dependent variables could be made to allow one to examine how these affect the mobility and performance of the robot. Finally, other weight matrices could be used with different criteria, like the stability of the robot, and study how these affect the dynamics of the robot.

8 Bibliography

[1] G. Besseron, Ch. Grand, F. Ben Amar, F. Plumet, and Ph. Bidaud, “Locomotion Modes of an Hybrid Wheel-Legged Robot,” *Climbing and Walking Robots*, Springer, Berlin, Heidelberg, 2005.

[2] Marko Bjelonic, C. Dario Bellicoso, Yvain de Viragh, Dhionis Sako, F. Dante Tresoldi, Fabian Jenelten and Marco Hutter, “Keep Rollin’ – Whole-Body Motion Control and Planning for Wheeled Quadrupedal Robots,” *IEEE robotics and automation letters*, January 2019.

[3] Boston Dynamics, Introducing handle, https://www.youtube.com/watch?v=-7xvqQeoA8c

[4] Boston Dynamics, Mobile Box Handling Robots for Logistics, https://www.bostondynamics.com/handle

[5] F. Cordes, C. Oekermann, A. Babu, D. Kuehn, T. Stark, F. Kirchner, and D. R. I. C. Bremen, “An Active Suspension System for a Planetary Rover,” in *Proceedings of the International Symposium on Artificial Intelligence, Robotics and Automation in Space (i-SAIRAS)*, 2014, pp. 17-19.

[6] Francisco Geu Flores, “An Object-Oriented Framework for Spatial Motion Planning of Multiboy Systems,” *Thesis*, University of Duisburg-Essen, December, 2012.

[7] Francisco Geu Flores, “Spatial Motion Planning of Multibody Systems,” *4th International Doctoral School of Electrical Engineering and Power Electronics*, Riga, Latvia, May 29th, 2015.

[8] Daniel Germann, Manfred Hiller, Dieter Schramm, “Design and Control of the Quadruped Walking Robot ALDURO,” *22nd International Symposium on Automation and Robotics, ISARC*, 2005.

[9] P. R. Giordano, M. Fuchs, A. Albu-Schaffer and G. Hirzinger, "On the kinematic modeling and control of a mobile platform equipped with steering wheels and movable legs," *2009 IEEE International Conference on Robotics and Automation*, Kobe, 2009, pp. 4080-4087.

[10] C. Grand, F. Benamar and F. Plumet, “Motion kinematics analysis of wheeled-legged rover over 3d surface with posture adaptation,” *Mechanism and Machine Theory,* 2010, vol. 45, no. 3, pp. 477-495.

[11] M. Heverly, J. Matthews, M. Frost, Ch. Mc-Quin, “Development of the Tri-ATHLETE Lunar Vehicle Prototype,” *Proceedings of the 40th Aerospace Mechanisms Symposium*, NASA Kennedy Space Center, May 12-14, 2010.

[12] M. Hiller, D. Germann, “Manoeuvrability of the Legged and Wheeled Vehicle ALDURO in Uneven Terrain with Consideration of Nonholonomic Constraints,” *Proceedings of ISOM*, Chemnitz, Germany, March 21-22, 2002.

[13] Hyundai Elevate concept video, https://newspress-hyundai.s3.amazonaws.com/videos%2Foriginal%2F35027-ElevateFullLengthFINAL.mp4, and Hyundai Cradle Walking Car Concept – Robot Demo | CES 2019, https://www.youtube.com/watch?v=hQ5Xib6sFp4

[14] Nils Brynedal Ignell, Niclas Rasmusson, and Johan Matsson, “An overview of legged and wheeled robotic locomotion.,” 2012.

[15] M. Kamedula, N. Kashiri, and N. G. Tsagarakis, “On the kinematics of wheeled motion control of a hybrid wheeled-legged centauro robot,” *in IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 2426-2433.

[16] T. Klamt and S. Behnke, “Anytime hybrid driving-stepping locomotion planning,” *in International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 4444-4451.

[17] T. Klamt, D. Rodriguez, M. Schwarz, C. Lenz, D. Pavlichenko, D. Droeschel, and S. Behnke, “Supervised autonomous locomotion and manipulation for disaster response with a centaur-like robot,” *in IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 1-8.

[18] A. Laurenzi, E. M. Hoffman, and N. G. Tsagarakis, “Quadrupedal walking motion and footstep placement through linear model predictive control,” *in IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 2267-2273.

[19] J. Lim, I. Lee, I. Shim, H. Jung, H. M. Joe, H. Bae, O. Sim, J. Oh, T. Jung, S. Shin, et al., “Robot system of DRC-HUBO+ and control strategy of team kaist in darpa robotics challenge finals,” *Journal of Field Robotics*, vol. 34, no. 4, 2017, pp. 802-829.

[20] Shun Hoe Lim, Jason Teo, “Recent Advances on Locomotion Mechanisms of Hybrid Mobile Robots,” *WSEAS transactions on systems*, vol. 14, 2015.

[21] Meng Ning, Bilun Xue, Zefeng Ma, Changhong Zhu, Zihao Liu, Cuncai Zhang, Yao Wang, and QiuJu Zhang, “Design, Analysis, and Experiment for Rescue Robot with Wheel-Legged Structure,” *Mathematical Problems in Engineering*, November 6, 2017.

[22] W. Reid, F. J. Pérez-Grau, A. H. Göktoğan and S. Sukkarieh, "Actively articulated suspension for a wheel-on-leg rover operating on a Martian analog surface," *2016 IEEE International Conference on Robotics and Automation (ICRA)*, Stockholm, 2016, pp. 5596-5602.

[23] Max Schwarz, Tobias Rodehutskors, Michael Schreiber, and Sven Behnke, “Hybrid Driving-Stepping Locomotion with the Wheeled-legged Robot Momaro,” *IEEE International Conference on Robotics and Automation (ICRA)*, Stockholm, Sweden, May 2016.

[24] Bruno Siciliano, Lorenzo Sciavicco, Luigi Villani, Giuseppe Oriolo, “*Ρομποτική*,” 2009.

[25] Klaus Six, Andres Kecskemethy, “Steering properties of a combined wheeled and legged striding excavator,” *Proceedings of the Tenth World Congress on the Theory of Machines and Mechanisms*, Oulu, Finland, June 20-24, 1999, p. 135-140.

[26] J. Szrek and P. Wojtowicz, “Idea of wheel-legged robot and its control system design,” *Bulletin of the polish academy of sciences, technical sciences*, vol. 58, no. 1, 2010.

[27] Change Zheng, Jinhao Liu, Tony E Grift, Zichao Zhang, Tao Sheng, Jiangwenjie Zhou, Yuliang Ma and Ming Yin, “Design and analysis of a wheel-legged hybrid locomotion mechanism,” *Advances in Mechanical Engineering*, Vol. 7(11) 1-10, 2015.

[28] Elias Zournatzis, “Design of a small-scale hybrid quadruped/wheeled robot with low-cost methods,” *Thesis*, National Technical University of Athens, April, 2019.

[29] “*Walking Excavator and Mobility*”, *Unusual off-road locomotion website*, September 2010.