Exploring Human-in-the-Loop Mechatronic Systems: Tools for Complex Dynamic Simulator Design and Sizing

A thesis presented for the degree of
Doctor of Philosophy

Dipartimento di Ingegneria Elettronica, Informatica ed Elettrica
Università degli Studi di Pavia
Italy

Supervisor

Prof. Hermes Giberti

Candidate

Andrea Spini

Student Identification Number

495540

Ringraziamenti

Contents

Abstract **11**

Introduction **12**

1 Driving simulator **16**
- 1.1 History of simulation 16
- 1.2 Classification of Driving Simulators 22
- 1.3 Field of application 23
  - 1.3.1 Research 23
  - 1.3.2 Training 24
- 1.4 Set up of a driving simulator 24
- 1.5 Sizing issue 27

2 Architecture analysis: kinematics **32**
- 2.1 Architecture choice 32
- 2.2 Parallel Kinematic Manipulators introduction 34
  - 2.2.1 Definitions 35
  - 2.2.2 Generalities on Parallel Robots 36
  - 2.2.3 3DoF Parallel Kinematic Manipulators 37
- 2.3 Tripod's kinematic analysis 39
  - 2.3.1 Reference systems 39
  - 2.3.2 Rotation Matrix 41
  - 2.3.3 Kinematic configuration 47
  - 2.3.4 Geometric parametrization 49
  - 2.3.5 Definition of Parasitic Motions 50
  - 2.3.6 Kinemati Crosstalk modelization 51
  - 2.3.7 Inverse Kinematic 51
  - 2.3.8 Kinetostatic analysis 55
  - 2.3.9 Constraints 56
- 2.4 Cartesian's kinematic analysis 60

2.4.1 Reference systems 61

3 **Architecture analysis: dynamics** **64**
3.1 Tripod's dynamic analysis 64
3.1.1 Dynamic model 65
3.2 Cartesian dynamic analysis 76
3.3 Performance definition tools 83
3.3.1 Workspace 83
3.3.2 Single DOf Performances 90
3.3.3 Kinematic crosstalk workspace 92

4 **Test case: complete system** **93**
4.1 Upper stage: Tripod 94
4.2 Lower stage: Cartesian 101

5 **Automotive dynamic simulation** **104**
5.1 Source signals' origin 104
5.1.1 Shared memory 105
5.1.2 Custom plug-in interface 110
5.2 Motion Cueing Algorithm - MCA 110
5.2.1 Algorithm development 111
5.2.2 Cueing parametrization 111

6 **Statistical method for sizing** **114**
6.1 Motor-Transmission selection method 115
6.1.1 Uncertain loading condition 118
6.2 Sizing tool 119
6.2.1 Platform motion: characterization and prametrization 120
6.2.2 Monte Carlo Method 121
6.3 Procedure's customization 128
6.3.1 Customization of logarithmic graphs 128
6.3.2 PDF customization 129

7 **Real Case** **130**
7.1 Dataset generation 130
7.1.1 Vehicles, Cueings, and Tracks 131
7.2 Statistical method customization 135
7.3 MCM results 139
7.4 Mechanical Sizing with custom MCM 153

7.5 Monte Carlo Method: customized vs generic 165

8 Mastering Tools: Unveiling Practical Applications 168

8.1 Hexalift 168

8.1.1 Requirements 168

8.2 Static and Dynamic comparison 168

8.2.1 Performance comparison 168

8.2.2 Exploring platform variations 168

8.2.3 Workspace investigation 168

9 Conclusions 169

List of Figures

1.1 Antoinette fight simulator. 17
1.2 Gough platform, 1947. 17
1.3 Stewart platform, 1965. 18
1.4 IFAS 1 DoF simulator 19
1.5 MARS Driving simulator 19
1.6 Daimler-Benz simulator, 1985. 20
1.7 Virttex simulator, 2001. 20
1.8 BMW simulator, 2003. 21
1.9 University of Iowa's driving simulator NADS-I, 2003. 21
1.10 Toyota's Driving Simulator, 2007. 22
1.11 Flow chart of a driving simulator with its hardware and software components 25
1.12 Servo-system example 28
1.13 Exampel of MLB, denoted as $g(x)$, of a given function $f(x)$. 30

2.1 3 DoFs PKM + 3DoFs cartesian robot 34
2.2 Delta robot (Claved R. 1986). 37
2.3 RPZ different configurations 38
2.4 Different reference systems: Local RS (Yellow), Top-Disk RS (green) and Global RS (Purple). 40
2.5 From global to an intermediate reference system due to Yaw rotation . 42
2.6 From a first intermediate reference (r.s.) system to a second intermediate r.s. due to Pitch rotation 42
2.7 From a first intermediate reference system to a second intermediate r.s. due to Pitch rotation 43
2.8 Yaw rotation seen from $x - y_{PLANE}$. 43
2.9 Pitch rotation seen from $x - z_{PLANE}$. 44
2.10 Roll rotation seen from $y - z_{PLANE}$. 44
2.11 consecutive rotations. 46
2.12 Example of $\underline{\text{P}}$RS configuration for a RPZ PKM 48

2.13 Custom configuration of the RPZ involved in the presented work . . . 48
2.14 Actuation planes for the considered configuration 49
2.15 Angles of the auction planes. 50
2.16 Stylized model of the machine with vectorial closures drawn 52
2.17 Actuator stroke 57
2.18 Cardan mobility range 58
2.19 Stylized Cartesian coupled with tripod 61
2.20 Cartesian (Lower Stage) **Global R.S.** and Cartesian (Lower Stage) **Local R.S.** 62

3.1 Representation of the masses for the top-disk subsystem (platform subsystem) 66
3.2 Upper subsystem joints' forces definition in *Global RS* $x - y_{plane}$ . . . 71
3.3 Forces definition in *Actuation RS* $x - z_{plane}$ 73
3.4 Inertial Resulting Forces from cartesian motion to the CoG of the upper stage 77
3.5 Inertial Resulting Forces from cartesian motion to the CoG of the upper stage: view from $x - y_{plane}$ 78
3.6 Coupled cartesian and tripod system with an example of yaw rotation. The position of the slider on 1^{st} actuation system is underlined 80
3.7 *Roll-Pitch-Z* maximum workspace for the considered platform 85
3.8 *Roll-Pitch-Z* maximum workspace for the considered platform with maximum z displacement for given roll and pitch rotation. 86
3.9 *Roll-Z* maximum workspace with $Pitch = 0$ $[deg]$. 87
3.10 *Pitch-Z* maximum workspace with $Roll = 0$ $[deg]$. 88
3.11 Maximum cartesian workspace 89
3.12 Example of logarithmic graph 91

4.1 Complete system 94
4.2 Configuration A 95
4.3 Configuration B 95
4.4 Configuration C 95
4.5 Configuration D 95
4.6 Final configuration: linear guide rotated around y-axis of the actuation R.S: by an angle of $\mu = 15[deg]$ 96
4.7 View of the universal joint of the upper stage's link 97
4.8 Upper stage link 98
4.9 Guide assembly 99

4.10 Topdisk 99
4.11 Upper stage assembly: 3DoF PKM 100
4.12 Complete system lateral view 101
4.13 Lower stage: cartesian 102
4.14 Section of the section devoted to the implementation of the yaw 102

5.1 Use of the commercial video game Assetto Corsa in the process previously shown in the figure 105
5.2 Generic use of a motion cueing algorithm 110

6.1 Example of motor-trasmission-load system 115
6.2 PKM TOW where the planes defining the limits of the pitch sinusoid are shown in red 124
6.3 PKM TOW where the planes defining the limits of the pitch and roll sinusoid are shown as red and green surfaces. 126
6.4 PKM TOW where the planes defining the limits of the pitch, roll and z sinusoid are shown as red, green and yellow surfaces. 127

7.1 131
7.2 Formula RSS3 V6: chassy with representation of CoG, driver position, TCP positions 132
7.3 Ferrari 488 gt3: chassy with representation of CoG, driver position, TCP positions 132
7.4 Abarth 500: chassy with representation of CoG, driver position, TCP positions 132
7.5 Audi a1s1: chassy with representation of CoG, driver position, TCP positions 133
7.6 Maserati Levante: chassy with representation of CoG, driver position, TCP positions 133
7.7 Praga r1: chassy with representation of CoG, driver position, TCP positions 133
7.8 Dummy logarithmic graphs obtained from AC simulations. Precisely, the blue line represents the limit in position, the red line represents the limit in speed, and the green line represents the limit in acceleration. The locus of the points represented as purple dots is what is called the dummy log graph. 138
7.9 Histograms representing the samples of the 6 DOFs describing the platform movements defined by the story generator 141

7.10 Histograms representing the load factors generated through the story generator on all six kinematic axes of the considered robot. 142
7.11 Merged histograms 143
7.12 β dispersions over the 50 subsets defined by the histogram 7.11a . . . 144
7.13 β_{95} for the tripod 144
7.14 β_{95} of the tripod versus the accelerating factor of the involved motors . 145
7.15 Admissibile τ range (τ_{min}, τ_{max}) for the chosen motor on the upper stage (tripod). 146
7.16 Charcteristic quantities at tripod actuation after the introduction of the chosen motors 147
7.17 β_{95} of the the Y cartesian axis versus the accelerating factor of the involved motors 148
7.18 Charcteristic quantities at tripod actuation after the introduction of the chosen motors 148
7.19 β_{95} of the Yaw cartesian axis versus the accelerating factor of the involved motors 149
7.20 Admissibile τ range (τ_{min}, τ_{max}) for the chosen motor for Yaw actuation. 150
7.21 Histogram for the max Torques [Nm] generated in each generated history at the actuation of Yaw-axis 150
7.22 Logarithmic graphs for each platform DoFs obtained considering the motors in tab 7.5 152
7.23 Requested motor torque and speed for not admissible motion 154
7.24 Distribution of max force directed as tripod's link 156
7.25 Distribution of max lateral force in tripod's actuation planes 156
7.26 Distribution of max force directed as z-axis of Actuation R.S. rotated by μ 157
7.27 Distribution of max force directed as x-axis of Actuation R.S. rotated by μ 157
7.28 Distribution of max force directed as x-axis of Actuation R.S. 158
7.29 Distribution of max force directed as z-axis of Actuation R.S. 158
7.30 Distribution of M_Z around the z-axis of Actuaion R.S. 159
7.31 Distribution of M_V around the z-axis of Actuaion R.S. rotated by μ . . 159
7.32 Distribution of M_Q around the x-axis of Actuaion R.S. rotated by μ . . 160
7.33 Distribution of M_X around the x-axis of Actuaion R.S. 160
7.34 Distribution of max force directed as x- actuation axis of the cartesian robot 161

7.35 Distribution of max force directed as y-actuation axis of the cartesian robot 161
7.36 Distribution of max force discharged from the upper system to the lower one aligned as the x-axis of the Global Lower R.S. 162
7.37 Distribution of max force discharged from the upper system to the lower one aligned as the y-axis of the Global Lower R.S. 162
7.38 Distribution of max force discharged from the upper system to the lower one aligned as the z-axis of the Global Lower R.S. 163
7.39 Distribution of M_x (around x-axis of the Global Lower R.S.) discharged from the upper system to the lower one 163
7.40 Distribution of M_y (around y-axis of the Global Lower R.S.) discharged from the upper system to the lower one 164
7.41 Distribution of M_z (around z-axis of the Global Lower R.S.) discharged from the upper system to the lower one 164
7.42 Tripod Load factor (β) histogram: Customized MCM vs Generic MCM 166
7.43 X-Y Load factor (β) histogram: Customized MCM vs Generic MCM . 166
7.44 Yaw Load factor (β) histogram: Customized MCM vs Generic MCM 167

List of Tables

2.1 tab: RPZ reference performances 33
2.2 tab: Surge sway and yaw reference performances 33
2.3 Summary of benefits and drawbacks for PKMs 37
2.4 Nomenclature for kinematic analysis 50

3.1 Maximum Z displacement function of Roll-Pitch ranges 86
3.2 Maximum Z displacement function of Roll ranges with Pitch = 0 [deg] 87
3.3 Maximum Z displacement function of Pitch ranges with Roll = 0 [deg] 88
3.4 Cartesian robot's maximum displacement 90

4.1 Information of the top disk showed in figure 4.10 100

5.1 static AC available data 106
5.2 Physics AC available data 107
5.3 Graphics AC available data 109
5.4 Cueing algorithm parameters that can vary and on which platform signals depend 112

7.1 Cuieng parameters of the five sets involved in platform DOFs generation 134
7.2 Correlation values involved in the customized MCM procedure (α= roll, β = pitch, γ = yaw). 136
7.3 Position, velocity and acceleration limit coming from the analysis of the AC simulation. 139
7.4 β_{95} , the value considered to exclude engines that cannot be used based on limited performance 143
7.5 Summary of the torque and speed characteristics of the motors chosen for axes actuation. 151
7.6 Platform DoFs performances 152

Abstract

The primary focus of this PhD research was to develop valuable tools for designing complex mechanical systems, particularly Human-In-the-Loop (HiL) mechatronic solutions. These structures integrate the human operator as an integral part of the system, adding complexity to platform handling. Consequently, critical quantities essential for mechanical sizing and implementation definition become challenging to determine. This thesis introduces a tool, along with additional resources, designed to evaluate and design dynamic platforms, specifically targeting the challenges encountered during the critical phase of sizing. To achieve this, a real-world test case is incorporated.

Introduction

The primary objective of this PhD research activity was to study and develop valuable tools for the design of complex mechanical systems. Specifically, the focus of this study was on mechatronic solutions known as Human-In-the-Loop (HiL) systems. HiL systems, characterized by the inclusion of human operators, find applications across a wide spectrum of fields, ranging from flight simulators and military training to medical training and sports equipment testing, including medical-rehabilitation garments.
In the automotive sector, HiL systems play a pivotal role in the research and development of new technologies. These particular structures are thus characterized by an environment whereby the operator becomes an integral part of the complete system. Generally in automotive applications, it is referred to as cockpit. The mechanical platform is another critical element, enabling the simulation of various realistic scenarios in alignment with the intended goals. To achieve this, dedicated hardware and software components are required to facilitate real-time simulation.

A paramount objective of HiL systems in the automotive field is to faithfully replicate a driver's experiences in real-world conditions. This necessitates not only immersive auditory and visual elements but also the generation of a realistic inertial force field for the operator. Such complexity underscores the need for these systems to be exceptionally high-performing, meeting stringent specifications while ensuring safety.

Typically, HiL systems leverage parallel kinematic robots due to their impressive dynamic capabilities and payload capacity. In certain instances, other kinematic structures, such as Cartesian systems, may be used to address specific limitations of parallel kinematic machines, resulting in hybrid structures. Some advanced dynamic simulators even employ kinematic structures with redundant degrees of freedom to separate tasks, where multiple implementations of the same degree of freedom act on different dynamics.

In the literature, it is possible to see how different tools have been used for the analysis and design of robots involved in dynamic simulation. In the two papers in succession

described in [1] and [2], it is possible to see how the use of kinematic and dynamic models of the structures under consideration is of considerable importance. In [3] the definition of the structure of a PKM of the kinematic point of view is addressed, using different approaches. Various procedures have been developed over the years for defining the most suitable structure according to project specifications, like [4] where a particular kinematic and dynamic analysis has been carried out in order to achieve specific results. In [5], an approach utilizing indices for the evaluation of a robot's dynamic performance is introduced. Another example is the use of special algorithms, such as the genetic algorithm, for the kinematic optimization of these particular machines. So, it is clear how having a wide use for these machines various avenues for optimal design have been considered.
Regarding the sizing process, the commonly traveled paths are to use experience, thus relying on the designer's prior knowledge, or to take a limit case and consider it sufficiently close to the critical performance required for such a machine. Accordingly, based on the choices made, size the structure and define the actuation system.

As mentioned above, due to the involvement of humans, who exhibit non-deterministic behavior, and the complex kinematics of the considered structures the actuation requirements for these machines can be challenging to define accurately. This is further compounded by the unique characteristics of the automotive application, particularly the need to create an immersive force field for the operator. As a result, determining the performance criteria necessary to achieve the intended objectives can be complex.

To address these challenges, specialized tools are essential to assist designers. One of the main problems addressed in this study is the sizing process for these specific HiL systems. In addition, other tools introduced include single Degree of Freedom (DoF) logarithmic graphs (combined with workspace definition, a common tool) and the definition and evaluation of kinematic crosstalk, which involves unwanted movements in specific kinematic solutions. These tools collectively contribute to the development and optimization of HiL systems for various applications, particularly in the automotive industry.
Given the introduction above, the thesis structure is here shown.

Chapter 1

In this first chapter, the Stat-of-Art of the mechanical stuctures involved in driving simulation field is presented. Then, an overview of the addressed sizing issue is introduced.

Chapter 2

The study being presented was applied to a real test case. Specifically, it was applied to the design of a compact dynamic simulator. Thus, this chapter covers the kinematic description of the mechanical system involved.

Chapter 3

Recalling what was described in the previous chapter, a detailed description of the dynamic model is made in the third chapter. In addition, special attention is paid to some useful tools for defining machine performance and evaluating kinematics.

Chapter 4

Having provided the kinematic and dynamic descriptions of the structure, this chapter aims to briefly address some implementation choices for the considered platforms thus providing a clearer picture of the specific one to which the main object of the thesis, namely the implementation definition tool, is applied.

Chapter 5

Given the application of the considered platform. In this chapter, a crucial step for simulation is addressed: the origin of the motions (or input signals) to be replicated in simulation. Then the use that is made of the commercial video game (Assetto Corsa) is explained and finally, a digression on the involved cueing algorithm is presented.

Chapter 6

At this stage of the thesis, The statistical method used to define the critical conditions to be used in sizing will be accurately described. The operational methodology and the mandatory elements will be listed and explained.

Chapter 7

This chapter closes the design description by including the results obtained through the previously described method. So, it shows the choice of actuation system at the design stage of the considered machine using the presented method.

Chapter 8

The concluding chapter serves the purpose of demonstrating the versatility of the tools utilized in this thesis. These tools can be applied in various scenarios, including the comparison of different potential implementation solutions for the same machine or the evaluation of two distinct machines in terms of both dynamic and static performance, specifically the workspace.

To illustrate this, we will briefly introduce a particular structure, a Parallel Kinematic Manipulator (PKM) robot with six Degrees of Freedom (DoFs). This example will serve as a practical case study to showcase the wide-ranging applications of the tools discussed throughout this thesis.

Chapter 1
Driving simulator

This chapter is intended to provide all the necessary ingredients for an overview of dynamic simulation.
The discussion begins with a concise overview of the history of simulation, progressing to an examination of the classification of driving simulators and their different applications. Additionally, the use of parallel kinematics in this domain will be explored. Finally, the chapter will delve into the components comprising a driving simulator.

1.1 History of simulation

The earliest recorded instance of motion simulation can be traced back to the Antoinette flight simulator. This simulator consisted of a basic seat equipped with wing controls attached to a universal joint. To enhance the fidelity and effectiveness of training sessions, the teachers manually simulated wind forces acting on the structure, providing the environmental inputs for the simulated vehicle. This early innovation aimed to create a more realistic training experience and improve the quality of training sessions. Two additional significant milestones deserve mention in the history of motion simulation. In 1947, E. Gough achieved a pivotal breakthrough with the introduction of a revolutionary concept: a 6-degree-of-freedom parallel manipulator featuring six linear actuators [6] (refer to Fig. 1.2).

Figure 1.1: Antoinette fight simulator.

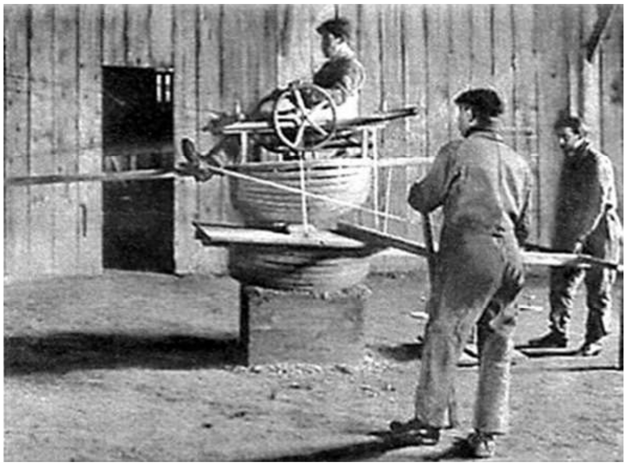
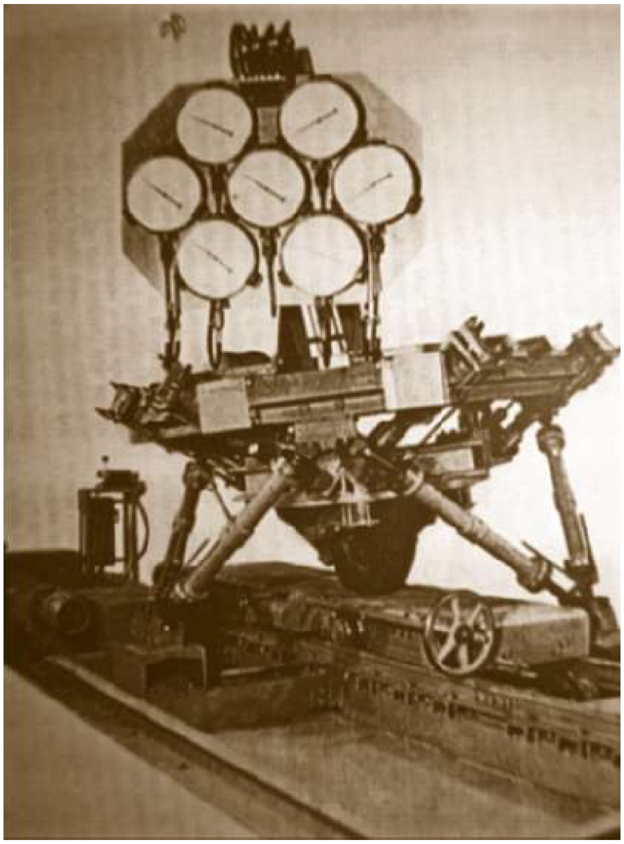
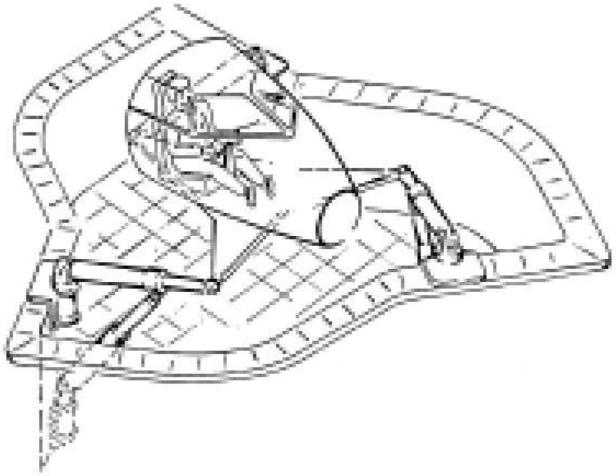


Figure 1.2: Gough platform, 1947.



In 1965, the renowned Stewart platform was introduced by D. Stewart [7]. Stewart presented a modified version of Gough's platform, specifically noting its potential application as a flight simulator, thus marking its entry into the field of motion simulation.

Figure 1.3: Stewart platform, 1965.



The reasons for the success of the so-called Stewart platform were that it combined a relatively compact design with relatively large allowable payloads. In the later years following the introduction of this platform, driving simulators have seen growing deployment in many research fields and the large variety of applications and purposes. Due to this variety, a high number of different simulators have been developed. The most important design solutions developed over the years will be now briefly introduced. It will allow to better understand which design ways have been undertaken over 50 years. The first driving simulator was built by Volkswagen [8] (the early 1970s).
It concerned a car on a 3DOF motion system (Roll, Pitch, Yaw). A single, flatscreen was mounted in front of the driver sitting. In 1984 IFAS [9] produced a 1 DOF with the visual projected in a box in front of the driver. The actuated DOF was a hydraulically driven y-sled (Fig. 1.4). IFAS extended their simulator with 6DOFs in 2004 with the MARS Driving Simulator. A y-sled is present in the new design, but has a hexapod mounted on top of it (Fig. 1.5).

Figure 1.4: IFAS 1 DoF simulator

Figure 1.5: MARS Driving simulator

In 1985 Daimler-Benz introduced the first 6DOF driving simulator [10]. A truck or car cabin could be fitted inside the moving dome on which a 180° image was projected (1.6).

Figure 1.6: Daimler-Benz simulator, 1985.

Figure 1.7: Virttex simulator, 2001.

In 1993, the simulator was improved by extending the motion system in the lateral direction (sway) [11]. Ford introduces the Virttex in 1994, a dome on a hydraulic hexapod. The Virttex was renewed in 2001 [12] (1.7). BMW developed a hydraulic hexapod standing at 4 meters tall, accompanied by a small screen mounted on the motion platform alongside a full-size car. This setup was completely rebuilt in 2003 [13]. The platform was equipped with a dome, and to enhance the immersive experience, the driver now enters the simulator through a tunnel, simulating the sensation of being inside a car rather than a simulator (refer to Fig. 1.8).

Figure 1.8: BMW simulator, 2003.

In 2004, Lander Simulation & Training Solutions introduced TUTOR, a comprehensive bus and truck simulator designed for professional driver training, aimed at enhancing driving behavior and skill development. In 2003, the University of Iowa unveiled the North American Driving Simulator (NADS-1), a sophisticated 9-degree-of-freedom (9DOF) simulator. The NADS-1 comprises an xy-table upon which a hexapod moves, with a turntable mounted on top to provide yaw acceleration. A dome housing a full-size car rotates atop the turntable (refer to Fig. 1.9).

Figure 1.9: University of Iowa's driving simulator NADS-I, 2003.

In 2007, Toyota introduced the Toyota Driving Simulator (refer to Fig. 1.10), which surpasses the NADS-1 in size. A notable distinction from the University of Iowa's simulator lies in the configuration of the turntable: in the Toyota Driving Simulator, the car yaws within the dome, whereas in the NADS-1, the entire dome, with the car inside, undergoes yaw movement. This simulator is intended for conducting driving tests that are too risky to perform in real-world settings, such as assessing the effects of drowsiness, fatigue, intoxication, illness, and distraction..

Figure 1.10: Toyota's Driving Simulator, 2007.

It is worth pointing out that many other simulators have been developed, which are not listed here.

1.2 Classification of Driving Simulators

The previous section has shown the wide variety of design solutions for motion simulators due to different application fields and purposes. A first classification based on the quality of simulation [14] can be carried out:

- *Low-level simulator*: In a fixed base simulator setup, the driver occupies a stationary car seat firmly anchored to the ground. Similarly, the screen displaying the simulated environment remains stationary. To optimize the viewing experience, the screen is typically designed with a wide viewing angle, achieved either through multiple screens or, ideally, a single large convex screen. Steering wheels and pedals in such setups often feature force feedback technology.

- *Mid-level simulator*: This setup involves a car that is accelerated along one degree of freedom (DoF). The screen may remain fixed or move in tandem with the car's acceleration. Force feedback is implemented in the steering wheel. Commonly, the actuated DoF includes a y-sled, x-sled, or a yaw-table.

- *High-level driving simulator*: The simulator operates by actuating the payload in at least 6 degrees of freedom (DOFs), with the potential for additional DOFs to accelerate the payload, thereby introducing redundant DOFs. This capability allows for extended planar excursions, among other functionalities. In contrast to fixed screens, some simulators utilize head-mounted displays (HMDs), which are commonly employed in various simulation research applications.

Excluding the *Low-level simulator*, the remaining options typically necessitate complex robotic systems to accomplish the desired tasks.

1.3 Field of application

Building upon the classification outlined earlier, it is pertinent to discuss the diverse application fields of this technology. Driving simulators, beyond their entertainment value, are seeing growing utilization in research across various domains. They are also becoming instrumental in driver training, offering a platform for imparting safe driving techniques and addressing specific driving challenges.

1.3.1 Research

Driving simulators serve a multitude of purposes within research facilities, spanning a broad spectrum of applications. These include:

- Study of driver behavior
- Human-machine interface (HMI) research
- Investigation of the impact of fatigue and drugs on driving
- Assessment of environmental concerns related to driving
- Exploration of road and landscape design considerations
- Tunnel design research
- Analysis of driver reactions in various scenarios
- Evaluation of drivers with reduced functionality
- Integration testing of new subsystems in vehicles [15].

In addition to their role in driver training research, driving simulators offer a safe and ethical means to investigate driver behavior under circumstances where conducting experiments on public roads would be illegal or unethical. For example, studying driver distraction poses significant risks and ethical concerns when conducted on public roads.

Vehicle manufacturers leverage driving simulators primarily for interior design evaluation and testing Advanced Driver Assistance Systems (ADAS). With the increasing prevalence of in-vehicle information systems (IVIS) such as satellite navigation, mobile phones, and DVD players, simulators play a crucial role in assessing their safety and functionality. Moreover, simulators are valuable tools for exploring the effects of noise and vibrations on driver performance. They also contribute to road planning efforts by aiding in the determination of optimal placements for road signs and contributing to aesthetic design considerations.

1.3.2 Training

Driving simulators are increasingly being utilized worldwide for driver training purposes. Numerous studies have demonstrated their efficacy and practicality as educational tools for teaching safe driving techniques to drivers of all skill levels. They offer a cost-effective alternative in scenarios where the use of real objects would be financially impractical. One notable advantage is the reduction of risk, as drivers can be exposed to complex scenarios that may pose hazards in real-world environments. Additionally, the ability to closely monitor students during practice sessions and provide real-time guidance is another valuable benefit. Various types of driving simulators, including train simulators, truck simulators, bus simulators, and car simulators, are employed as training tools to cater to different driving contexts.

1.4 Set up of a driving simulator

Despite the diverse applications of driving simulators, they can generally be introduced with a standardized structure to facilitate understanding. Each simulator is then distinguished by factors such as the budget allocated for its construction, project specifications, and, of course, the intended area of use.

Figure 1.11: Flow chart of a driving simulator with its hardware and software components

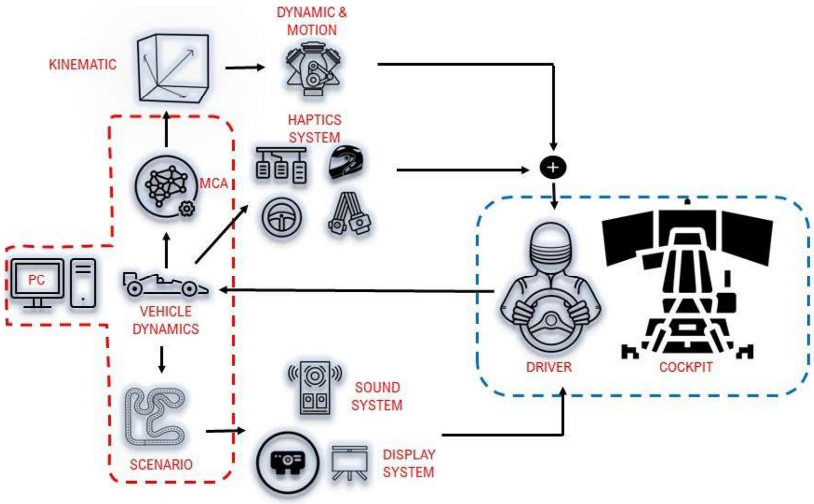


Figure 1.11 presents a simple yet effective diagram illustrating the main components of a driving simulator and their interconnections. Dynamic simulators, encompassing not only driving simulators but also other types, are intricate systems that entail not only mechanical components and kinematic structures but also software components of equal importance and complexity. Therefore, the principal subsystems of a driving simulator are depicted not only in terms of hardware but also in terms of software.
The main hardware subsystems are:

- the motion system, a robotic platform and usually a parallel kinematic manipulator
- the haptic interfaces such as actuated steering wheel, pedalboard, actuated helmet, and actuated seatbelts
- sound system
- visual system
- cockpit, or driver's cabin
- real-time computer.

Instead, the software mainly involved are:

- vehicle dynamics
- Motion Cueing Algorithm (MCA)

- kinematic and dynamic model of the considered robot

The cornerstone of the driving simulator experience is the motion system. It is a dedicated machine that produces tangible sensations for the driver. In particular, this system is tasked with generating a force field that encompasses inertial forces, aiming to replicate the most realistic sensations possible, thereby fostering a highly immersive experience for the pilot. The driver's input, combined with vehicle dynamics within the given scenario, generates crucial reference signals pivotal in dynamic simulation. These signals, once processed with appropriate algorithms, dictate the movements of the platform, consequently shaping the force field it produces. The scenario within the virtual simulation incorporates environmental and vehicle dynamics information, providing essential data for generating visual and auditory cues. While some simulators utilize multiple projectors to create a seamless image displayed on curved screens, employing image warping and blending processes prior to projection, next-generation simulators leverage virtual reality headsets for an even more immersive experience.
Additionally, haptic feedback, such as steering torque, active seat belts, and active helmets, is employed to convey cues derived from the vehicle dynamics. Thus, by retracing the diagram shown in Figure 1.11, the driver inside the cockpit, with the steering wheel and pedals, generates inputs for the vehicle dynamics. These inputs, combined with the scenario, define the vehicle's response: new positions, speeds, and accelerations. Visual and auditory cues are determined by the scenario. Instead, signals from the vehicle dynamics are combined using the kinematic model and a complex algorithm called the Motion Cueing Algorithm (MCA), responsible for transforming vehicle dynamics into platform dynamics. The platform dynamics return specific sensations to the rider. The pilot responds to these sensations by manipulating the peripherals (pedals and steering wheel), thus restarting the cycle. This cycle broadly describes the operation of a driving simulator. One of the most important aspects described above is the Motion Cueing Algorithm. The MCA analyzes the vehicle's response and calculates the appropriate movements of the motion system, considering the system's kinematics. These calculated movements are then translated into actuator commands by the machine's kinematics and executed by the motion system, providing the human driver with accurate cues. Thus, the importance of this algorithm in the driving simulation field is evident. It is this algorithm that defines the platform's movements, responsible for the sensations perceived by the driver. There are several approaches to motion cueing algorithms. The four main approaches for developing motion cueing are: classical, optimal, adaptive, and predictive. In [16], the development of the first three approaches is described in the context of flight simulators. The predictive approach, represented by the Motion Predictive Control Algorithm (MPC), can be considered the newest

approach.

1.5 Sizing issue

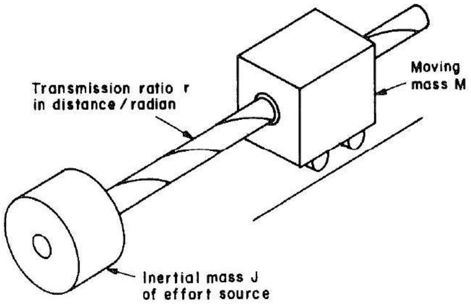
It becomes evident how platform movement is contingent upon three key factors: the presence of the pilot (as each pilot reacts differently to stimuli generated by the platform), the intricate kinematics of the robot, and finally, the cueing algorithm. The interdependence of these three factors complicates the definition of reference quantities for sizing purposes. Each factor introduces uncertainty into determining the critical movement for the platform under consideration, further complicating the sizing phase. In the context of a driving simulator, as mentioned earlier, uncertainty also arises from the application itself. Since the system is designed to react with motion to unpredictable inputs—in our case, driver requests to the vehicle—there isn't a specific and predefined task the machine can be sized for with certainty. Consequently, it's challenging to definitively establish the critical condition to consider during the sizing process. Moreover, boundary conditions in driving simulation, such as vehicle type, track layout, tire models, and similar factors, can further influence resulting motion even with identical inputs. This leads to a collection of individual dynamic requirements, the potential coupling of which is contingent on multiple and often unpredictable factors.

The most common approach to addressing this complex problem involves defining a specific task. Typically, in the sizing process, the starting and known element is precisely the task, often cyclical, that the platform must replicate. Another approach is through experience, where the identification of the most demanding task may be determined by experts and technicians familiar with the technology, both from the designers' and customers' perspectives. In this scenario, experience helps establish broad specifications (reference tasks) that are incorporated into the sizing procedure. Therefore, the task used in sizing is generally a known element, rather than an unknown variable.

During the years, the problem of the actuation sizing has attracted the attention of researchers.

In general, the term *Actuator* refers to the component of a generic machine responsible for generating motion, and the precise definition of these components is of critical importance in the overall design process of a mechanical system. The efficiency and performance of the system depend heavily on this definition. Therefore, "*Actuation sizing*", in a general mechanical design process, refers to the procedure involved in defining and selecting the correct collection of devices for motion generation and control. This procedure entails considering factors such as load requirements, speed, torque, precision, efficiency, safety, and reliability. The importance of actuation sizing cannot be overstated. Undersizing the actuation system would render it impossible to achieve the targeted performance, resulting in insufficient power or force generation, poor system performance, compromised functionality, and potential failure. Conversely, oversizing may not directly impact performance but can lead to issues such as unnecessary weight, increased costs, reduced energy efficiency, and inefficient resource utilization. Optimal actuation sizing is crucial to strike the right balance between performance, reliability, and cost-effectiveness. However, defining this procedure can be challenging due to the lack of necessary information. Precise dynamic performance can only be determined once all machine components have been precisely defined. The mechanical systems involved here are robots, falling within the field of mechatronics, which focuses on designing and manufacturing systems incorporating both mechanical and electronic components [17]. Typically, the actuation system comprises an electric motor and its related transmission. In 1984, [18] addressed the choice of the actuator and its transmission. Specifically, Pask and Seering addressed this issue by emphasizing the importance of maximizing system output acceleration (a) for many high-performance applications. They considered a drive system with inertia (J) and torque (T). After defining the transmission ratio (r), the equation of motion for the system shown in Figure 1.12 can be expressed as Equation 1.1.

Figure 1.12: Model of the servo-system considered in [18].



$$T = (\frac{J}{r} + mr)a. \tag{1.1}$$

The procedure, aimed at maximizing acceleration, seeks to minimize the derivative of the bracket term in Equation 1.1 to achieve the maximum acceleration for a given torque T. Throughout the remainder of the article, the effects of various decisions that may be made during the design stage are considered, including changes in rotary inertia stages, alterations in driven mass, and the incorporation of a generalized impedance at both load and source. The conclusion reached is that, in the absence of dissipative forces, for the system depicted in Figure 1.12, the transmission ratio $r = \sqrt{J/M}$ will yield maximum acceleration for the system.

By Van de Streate et al. in 1998, with the paper [19], a partially automated selection procedure was introduced and the problem of the servo-drive definition is split in two parts:

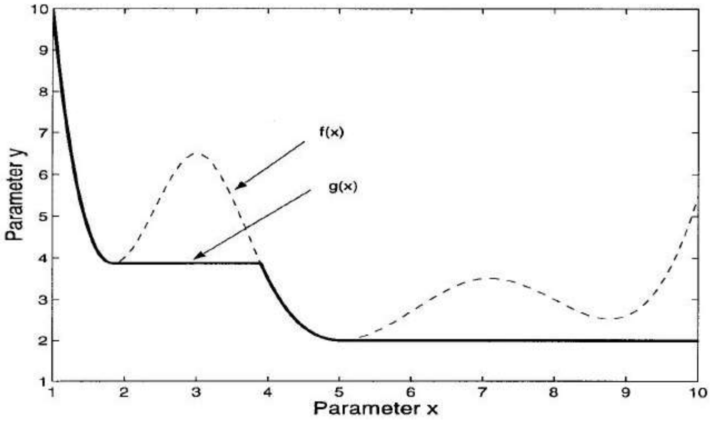
1. "*finds motors that are capapble of driving a given load*";
2. "*from this subset of feasible motors, choose the motor that satisfies best an additional criterion (price, weight, volume, etc.)*" .

The authors delve into the first step, presenting an approach for eliminating all motors incapable of driving a given load. They highlight the improvements brought about by this method, including its applicability to loads of any nature, the separation of load and motor considerations, and the graphical representation of dynamic feasibility.

The criterion introduced by [19] involves the following steps:

1. Calculate or simulate the speed, acceleration, and torque required to drive the load.
2. Choose the norms to evaluate the torque (often the 2-norm or ∞-norm).
3. Plot speed and torque for a simulated motion using generic motor inertia and multiple transmission ratios.
4. Eliminate motors whose representation falls below the "Maximum Lower Bound" (MLB) of the load curve (see Figure 1.13).
5. Finally, for the remaining motors, determine the admissible transmission ratio.

Figure 1.13: Exampel of MLB, denoted as $g(x)$, of a given function $f(x)$.



The graphical aspect of this method revolves around the concept of the Maximum Lower Bound (MLB) of a function. In the context of the problem discussed, the MLB delineates a single line on the $\omega - T$ graph, serving as a threshold that distinguishes between admissible and inadmissible motors.

In 1999, [20] described the aforementioned approach as a solution for driver selection, focusing on the most crucial servo motors categorized into three groups: DC permanent-magnet (DCPM) motors, brushless DC motors (BLDC), and AC permanent magnet synchronous motors (ACPM) and induction (IND) motors. Consequently, the procedure outlined in [20] involves a feasibility study akin to that in [19]. This feasibility analysis encompasses considerations such as peak torque limit, voltage limit, and continuous limit. The utilization of this procedure hinges on a solid foundation: a comprehensive understanding of the speed, acceleration, and torque requirements of the machine's operational cycle under consideration. As noted in [20], this approach "yields more insight into the requirements of typical load cases and into the relative potential of different drive systems.".

In his 2005 article [21], Cusimano G. proposes a generalized approach to the servo-reducer selection process, considering various factors that can significantly influence the selection process. These factors include resistant torques, additional masses on the motor and load, complex reference tasks, and specific characteristics of the motor's operating ranges. This method builds upon a previous approach described by Cusimano himself in [22]. Notably, in [21], a brushless motor is assumed. The updated method in [21] incorporates considerations for a resistant torque acting on the load (M_L), particularly a constant torque $M_{L,m}$ during motion and a constant torque $M_{L,r}$ during a fraction of the arrest time. Furthermore, optimization of the actuation system

and transmission is based on two quantities related to the motor and the load: kinetic energies and powers. This method ingeniously employs the k-p diagram to represent parameters crucial for the final decision. In contrast, in [23], Cusimano G. delves into the introduction of a new procedure, providing a detailed analysis of the methods presented in previous articles ([21] and [22]). Emphasis is placed on the motor-reducer selection process in cases where the thermal limit is more stringent than the limit on maximum torque.
This concise overview illustrates the evolution of approaches to defining implementation systems over the years. Initially, the problem was addressed with simplifications [17], but gradually more comprehensive considerations were incorporated, taking into account various influences on the decision-making process [23, 21]. A notable aspect of these methods is the consideration of a well-defined and known operational cycle of the involved machine. This ensures a precise understanding of the handling requirements. Through the literature cited above, we observe the evolution in approaches to dimensioning problems in applications involving electric motors and reducers, progressing from simplified methodologies to more sophisticated and comprehensive techniques.

This thesis aims to delve into the problem previously discussed and outlined in the summary above. Specifically, it focuses on addressing the challenges associated with complex mechanical systems whose movements are undefined and of complex definition, making the selection of the actuation system a complex task. To illustrate the work undertaken, a real-world test case is utilized, providing clarity to the reader. This test case embodies an application characterized by the aforementioned complexities, particularly in defining the most demanding task essential for sizing. Specifically, the focus is on a Human-in-the-Loop (HIL) system: a parallel kinematics (hybrid) robot utilized in dynamic simulation.

Chapter 2

Architecture analysis: kinematics

This chapter delves into the kinematic description of the mechanical structure of the complete Human-in-the-Loop (HIL) system introduced in the previous chapter.

The chapter begins with a concise introduction to the overall architecture under consideration. Following this, there is a brief exploration of parallel kinematics robots. Although brief, this digression is essential for comprehending the structure under consideration and its broader implications. A deeper understanding of parallel kinematic robots is instrumental in grasping the inherent challenges associated with defining specifications for sizing, which constitutes the primary focus of this study.

2.1 Architecture choice

Over the years, as dynamic simulators have evolved and extensive research has been conducted on simulating perceived motion, various design configurations have emerged for creating simulated environments for vehicles. Central to achieving high-quality simulation is the utilization of a system capable of accommodating all six degrees of freedom (6 DoFs). Parallel kinematic manipulators (PKMs) are a common feature found in the majority of dynamic simulators. This prevalence can be attributed to the necessity of reproducing degrees of freedom (DoFs) that demand high dynamics, frequencies, accuracy, and have small workspaces. The characteristics of parallel kinematics machines align more favorably with these specific requirements, making them a preferred choice. In high-end driving simulators, a commonly employed solution to address lateral and longitudinal, low-frequency acceleration challenges is the use of a Cartesian rail system. While effective in resolving such issues, this approach results in increased size and bulkiness of the simulator, elevated production costs, and limited dynamic performance. Of particular note is the significant footprint required to accommodate low-frequency acceleration requirements. For the structure presented, the reference payload is set at 300 kg ($m_{payload} = 300[kg]$).

From the literature [24],[25],[26] it is possible to inspect the typical excursions of the DoFs needed to guarantee an optimal feeling during the driving simulation. The typical performance values of the RPZ driving simulators available on the market are reported in Table 2.1.

	Heave (z)	Roll (α)	Pitch (β)
Displacement	$\pm 0.15[m]$	$\pm 10[deg]$	$\pm 10[deg]$
Velocity	$\pm 0.6[m/s]$	$\pm 100[deg/s]$	$\pm 100[deg/s]$
Acceleration	$\pm 1[g]$	$\pm 200[deg/s^2]$	$\pm 200[deg/s^2]$

Table 2.1: tab: RPZ reference performances

Defining the typical excursion for the remaining degrees of freedom (surge, sway, and yaw) poses more difficulty. In advanced simulator systems, these degrees of freedom are often redundant, meaning they are produced by two separate systems within the same machine. One system handles high frequencies, while the other addresses low frequencies, eliminating the need to limit the frequency range for these DoFs. Alternatively, in solutions aiming to reproduce these degrees of freedom solely in the high-frequency domain, a robot with a parallel kinematics structure utilizing a 6 DoF PKM is employed. Reference values for surge, sway, and yaw can be found in Table 2.2.

	Surge (x)	Sway (y)	Yaw (γ)
Displacement	$\pm 0.1[m]$	$\pm 0.1[m]$	$\pm 10[deg]$
Velocity	$\pm 1[m/s]$	$\pm 1[m/s]$	$\pm 10[deg/s]$
Acceleration	$\pm 1[g]$	$\pm 1[g]$	$\pm 200[deg/s^2]$

Table 2.2: tab: Surge sway and yaw reference performances

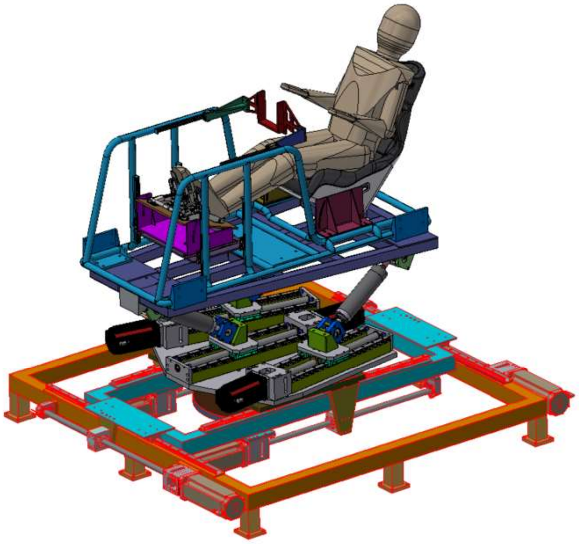
The particular solution presented in this study represents a middle ground between the two families in terms of reproducing these DoFs.

This knowledge combined with the necessity to design a compact and modular simulator led to the definition of a particular mechanism consisting of 2 separate machines:

- *the upper stage*, a 3 DoFs PKM (Roll, Pitch and Heave)
- *the lower stage*, a 3 DoFs cartesian robot for reproducing surge, sway and yaw (X,Y, and Yaw).

The upper stage, comprising the RPZ mechanism, facilitates the reproduction of all local movements of the vehicle chassis relative to the ground, which is a primary objective in driving simulation. These "chassis movements" result from suspension dynamics and can be induced by ground roughness, as well as load transfers during braking or steering. Typically, these movements occur in the high-frequency domain, making the use of a parallel kinematics system suitable for such applications. On the other hand, the lower stage, designed for compactness, focuses on reproducing only the high-frequency dynamics of the relative degrees of freedom (DoFs). By employing a Cartesian system for surge, sway, and yaw, with a larger available stroke dedicated solely to each individual DoF, a greater low-frequency component can be replicated. Importantly, this setup allows the workspace to remain unrestricted by the position assumed by the robot in these three dimensions, which is not feasible with a 6 DoF PKM. To enhance the reproduction of low-frequency accelerations, the presented system can be integrated with haptic systems such as active seat belts and active helmets. This solution offers the advantage of shorter strokes, resulting in a reduced footprint and lower costs.

Figure 2.1: 3 DoFs PKM + 3DoFs cartesian robot



2.2 Parallel Kinematic Manipulators introduction

As mentioned earlier, motion simulators, particularly driving simulators, often consist of complex robot platforms. Robots can be classified based on various criteria, including their degrees of freedom, kinematic structure, workspace geometry, and motion characteristics. One primary classification scheme categorizes robots according to their kinematic configuration:

- *Serial Robot* This configuration features an open kinematic chain composed of rigid links connected in series via 1 degree of freedom (1DOF) actuated articulations, which can be rotations or translations.
- *Parallel Robot* In contrast, this configuration involves a closed-loop kinematic chain with multiple convergent kinematic chains. The joints in parallel robots may have multiple degrees of freedom and can be either active or passive.

Parallel manipulators are typically preferred for motion and force transmission rather than dexterous manipulation tasks. They excel in applications requiring high stiffness, speed, and payload capability. Additionally, parallel manipulators offer advantages such as compact structure, low moving inertia, and cost-effectiveness compared to serial robots. Over the past two decades, parallel manipulators have been the subject of extensive research and continue to attract significant attention from both academic institutions and industry.

To provide a clearer understanding of parallel kinematic robots, this section will briefly introduce them.

2.2.1 Definitions

ISO 8373 is the International Standard that defines terms used in relation to robots and robotics devices operating in both industrial and non-industrial environments. The current definition of an industrial robot, according to ISO 8373 [27], is as follows: "An industrial robot is an automatically controlled, reprogrammable, multipurpose manipulator, programmable in three or more axes, which can be either fixed in place or mobile for use in industrial automation applications." Below are some of the definitions compiled in ISO 8373:

- **End Effector (EE)**. It is a peripheral device that attaches to a robot's wrist, allowing the robot to interact with its task
- **End Effector Pose**. It is the position and orientation of the EE: 3 position coordinates (*x, y, z*) and 3 orientation coordinates (*Roll, Pitch and Yaw*).
- **Tool Center Point (TCP)**. It is the point in relation to which all robot positionings are defined.

- **Home Position**. It is the pose of the robot when the TCP is in the center of its workspace, with null roll and pitch.
- **Accuracy**. It is the difference between the required pose and the average pose obtained following a number of repetitions.
- **Repeatibility**. It measures the dispersion of the obtained poses compared to their average.
- **Workspace**. The workspace of a robot is the set of all positions that its TCP can reach. This depends on a number of factors including the dimensions of the links.
- **Space of Movements**. Space achievable by every part of the robot.

2.2.2 Generalities on Parallel Robots

A parallel kinematic robot is generally made up of:

- **Space of Movements**: one mobile (top-disc) and the other fixed (base).
- **Links**: connecting elements between the base and the top-disc.
- **Joints**: that connect the links to the bases.

The joints, differently from the Serial robots, can have more degrees of freedom and can be active or passive. Usually, only the 1DOF joint is actuated for reasons of simplicity. Moreover, it is preferable to actuate one joint per link in order to obtain a modular solution in the construction of the manipulator.
The following joints can be used:

- 1DOF: Rotational (R) and Prismatic (P).
- 2DOF: Universal (U) and Cylindrical (C).
- 3DOF: Spherical (S).

In the following table, Table 2.3, are summerized the benefits and drawbacks of Parallel Kinematic Manipulators (PKMs). These can be considered the main characteristics of these specific platforms.

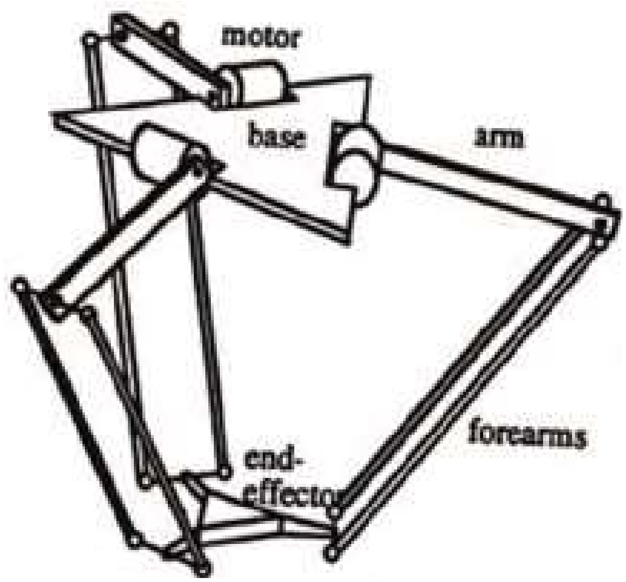
Advantages	Disadvantages
High Dynamic Performances	Difficult to obtain inclination of the EE
High Accuracy and Repeatability	Risk of collision between links
High Stiffness	Reduced Workspace
Modular Configuration	High Anisotropy
Large Load Carrying Capacity	Singularities

Table 2.3: Summary of benefits and drawbacks for PKMs

2.2.3 3DoF Parallel Kinematic Manipulators

In recent years, mechanisms with fewer degrees of freedom (DoFs) have garnered attention from researchers due to their reduced manufacturing costs and smaller footprints compared to 6-DoF Parallel Kinematic Manipulators (PKMs). Extensive studies on 3-DoF parallel robots have been conducted, with the 3-DoF Delta robot (see Fig. 2.2) emerging as the most renowned. It has found widespread application across various industries, particularly in contexts such as packaging and pick-and-place activities. The Delta robot, first investigated by R. Clavel in 1986 [28], has been extensively studied and implemented in industrial settings.

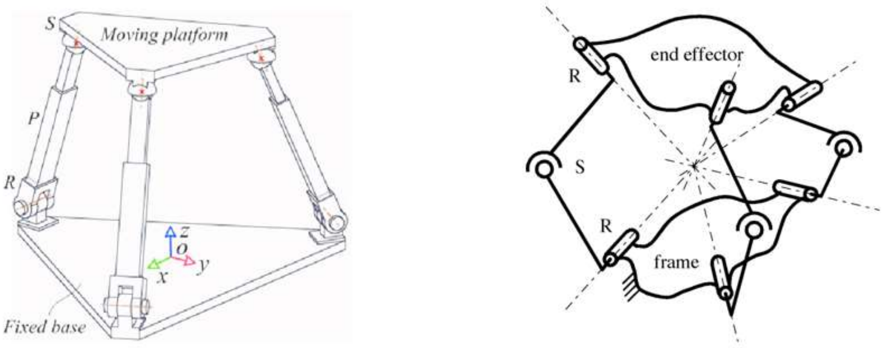
Figure 2.2: Delta robot (Claved R. 1986).



Over the years, various layouts have been studied and patented for driving simulation, with the Roll-Pitch-Heave (RPZ) configuration being particularly relevant. The utilization of a 3 degrees of freedom (3DoFs) parallel kinematic manipulator (PKM) in driving simulation presents several benefits, including compactness and reduced manufacturing costs. However, a notable drawback of such PKMs is their potential limitation to executing roto-translations instead of pure rotations, depending on the joint arrangement. This means that to achieve a specific rotation (e.g., roll or pitch), the end-effector may also undergo undesired movements in the horizontal plane (x and y directions) or rotation around the z-axis (yaw). These unintended movements, known as "parasitic motions," cannot be directly controlled and are distinct from the actual degrees of freedom (roll, pitch, and z) of the system. Therefore, it becomes crucial for 3DoFs PKMs to study the behavior of these parasitic motions and develop an analytical model for their calculation. Understanding these parasitic motions is essential for comprehensively grasping the kinematics of the robot and assessing its overall performance. In chapter 3, it will be introduced the analytical dynamic model and define Parasitic motions in detail.

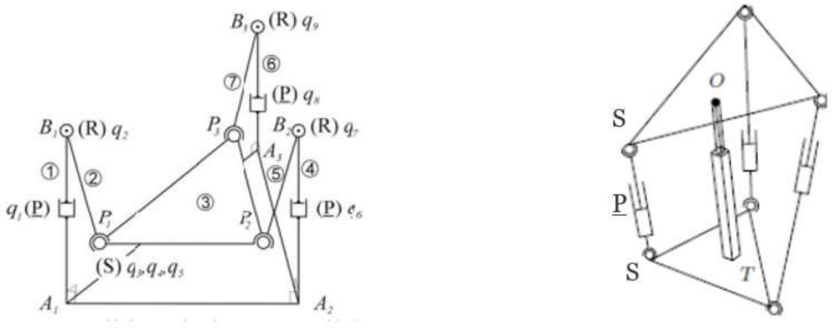
Figure 2.3 shows different configurations for a RPZ platform taken from literature.

Figure 2.3: RPZ different configurations



(a) RPS configuration

(b) RSR configuration



(c) PRS configuration

(d) SPS configuration

2.3 Tripod's kinematic analysis

This section provides the necessary analytical tools for conducting a comprehensive kinematic analysis of a generic Parallel Kinematics Machine (PKM) designed to perform roll, pitch, and heave movements, commonly referred to as a Tripod or RPZ (Roll-Pitch-Z). An initial RPZ configuration will be selected from existing literature, and a kinematic analysis will be conducted using a generic geometric parametrization. Furthermore, a methodological approach will be developed to facilitate the comparison of different geometric parametrizations and enable geometric optimization. Through this analysis, readers will gain insight into the relationships between the coordinates of the end-effector **X** in the workspace and the coordinates of the actuated joints **q**. The end-effector operates in a 3-dimensional space, necessitating the definition of both a point's position and the body's orientation. Two sets of kinematic equations will be explored: direct and inverse kinematics. Direct kinematics equations determine the workspace coordinates based on the joint variables, while inverse kinematics equations find the joint variables given the workspace coordinates. It's worth noting that for parallel architectures, inverse kinematics are generally more straightforward than direct kinematics. This is because inverse kinematics have a unique analytical solution, whereas direct kinematics often require numerical methods to solve a non-linear set of equations with a non-unique solution.

$$\mathbf{q} = \left\{\begin{matrix} q_1 \\ q_2 \\ q_3 \\ q_4 \\ q_5 \\ q_6 \end{matrix}\right\} \begin{matrix} \xrightarrow{Direct\ kinematic} \\ \xleftarrow[Inverse\ kinematic]{} \end{matrix} \left\{\begin{matrix} x \\ y \\ z \\ \alpha \\ \beta \\ \gamma \end{matrix}\right\} = \mathbf{X}. \tag{2.1}$$

2.3.1 Reference systems

In this section, we will introduce the various reference systems (R.S.) that will be used in the study of the kinematics and dynamics of the Tripod. Each reference system is chosen based on its suitability for specific aspects of the analysis.

- **Global R.S. (Grounded R.S.)**: This global reference system has its origin positioned at the center of the xy workspace, with the z-axis pointing upward. The x-y plane of this reference system belongs to the plane from which the positions of the actuation are defined. This system remains fixed for the tripod.

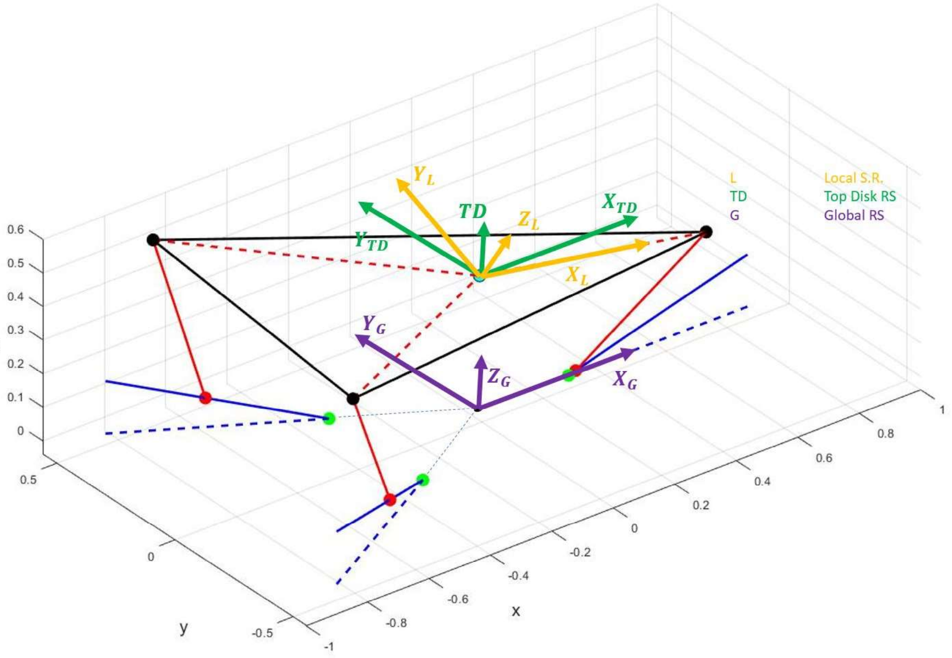
- **Actuation R.S.**: This local reference system is employed for solving the inverse kinematics of the robot. It shares the same origin as the Grounded R.S., with the z-axis pointing upward and the x-axis lying in the plane of the kinematic chain. Each kinematic chain has its own Actuation R.S.

- **Top-Disc R.S.**: The origin of this reference system is located at the center of the top-disc in its home position. It remains fixed and does not follow the motion of the top-disc. It can be used to describe the absolute motion of the end effector of the robot. It is similar to the Global R.S. The main difference is the z-coordinate.

- **Rotation-Disc R.S. (Local R.S.)**: Similar to the Top-Disc R.S., this reference system also has its origin at the center of the top-disc. However, it accounts for the roll and pitch movement of the Top-Disc. In this reference system, any point belonging to the TCP (Tool Center Point) remains constant.

The following figure (Figure 2.4) helps in the understanding of the definitions just introduced.

Figure 2.4: Different reference systems: Local RS (Yellow), Top-Disk RS (green) and Global RS (Purple).



In the kinematic analysis, the vectors will be indicated with a subscript specifying the Reference System in which they are written:

- G: is for Global R.S.
- A: is for Actuation R.S.
- T: is for Top-Disc R.S.
- L: is for Local R.S.

2.3.2 Rotation Matrix

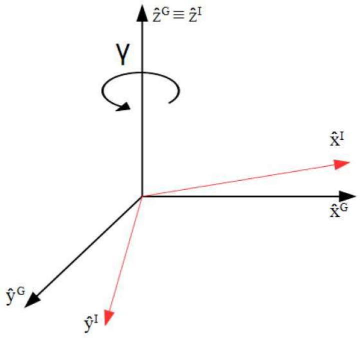
The coordinates of the Tool Center Point (TCP) expressed in the global reference system $\mathbf{x}_{tcp}^{G}$, combined with the rotational coordinates of the platform completely define the coordinates of the robot's end effector:

$$X = \begin{Bmatrix} \mathbf{x}_{tcp}^{G} \\ \Theta \end{Bmatrix} = \begin{Bmatrix} x_{tcp}^{G} \\ y_{tcp}^{G} \\ z_{tcp}^{G} \\ \alpha \\ \beta \\ \gamma \end{Bmatrix} \tag{2.2}$$

where $\Theta = \{\alpha, \beta, \gamma\} = \{Roll, Pitch, Yaw\}$. To characterize the orientation of the Top-disc, a set of appropriate rotational coordinates is required. Cardanic angles have been selected as the chosen coordinate system. These angles involve three consecutive rotations, applied in a specified sequence, that align the directions of the global reference axes with the directions of the local reference axes. The specified sequence, known as "Nautical angles," uniquely defines the orientation of the local reference frame relative to the global reference frame using three angles: α (roll), β (pitch), and $\gamma(yaw)$. In this section the most generic case is involved. Specifically, all three possible rotations are considered despite the fact that in the particular configuration involved in this thesis only two of the three rotations are possible.
Thus, the cited three consecutive rotations can be described as follows:

1. Yaw rotation (γ). It defines a rotation around the z-axis of the global reference system identified by $\hat{z}^{G}$. As shown by figure 2.5 , this rotation expresses the transition from the global reference to an intermediate reference "I":
$$\{x^{G}, y^{G}, z^{G}\} \longrightarrow \{x^{I}, y^{I}, z^{I}\}$$

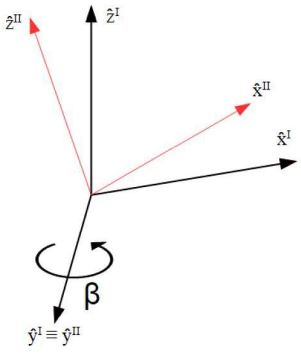
Figure 2.5: From global to an intermediate reference system due to Yaw rotation



2. Pitch rotation (β). This rotation is performed around the new y-axis y^I and it leads toward a second intermediate reference system (Figur 2.6):

$$\{x^I, y^I, z^I\} \longrightarrow \{x^{II}, y^{II}, z^{II}\}$$

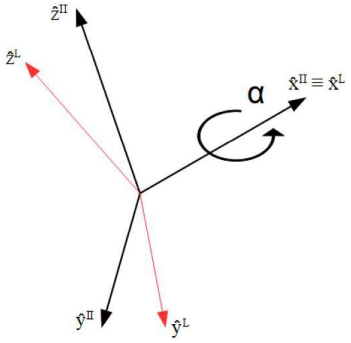
Figure 2.6: From a first intermediate reference (r.s.) system to a second intermediate r.s. due to Pitch rotation



3. Roll rotation (α). This last rotation is performed around x^{II} and it defines the rotation between the second intermediate reference system to the local reference system (Figure 2.7).

$$\{x^{II}, y^{II}, z^{II}\} \longrightarrow \{x^L, y^L, z^L\}$$

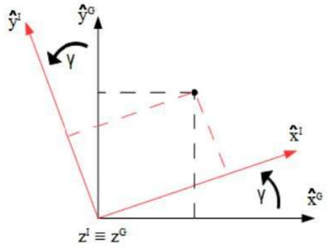
Figure 2.7: From a first intermediate reference system to a second intermediate r.s. due to Pitch rotation



By using these rotations is easily possible to define some reference parameters in a reference system or in another one. Considering a generic vector $\mathbf{d}^G$, defined in the global reference frame, the corresponding $\mathbf{d}^L$ defined in the local reference system can be obtained with the rotation matrix $[R_{LG}]$. In the presented thesis this matrix will also be referred to by the following notation $[R_1] = [R_{LG}]$. This matrix is defined by considering the just-mentioned sequential rotations. The following procedure will illustrate how $[R_{LG}]$ can be defined:

1. Yaw rotation (γ). Referring to figure 2.5 and looking at the $x - y_{PLANE}$

Figure 2.8: Yaw rotation seen from $x - y_{PLANE}$.



$$\mathbf{d}^I = [R_{I-G}]\mathbf{d}^G \tag{2.3}$$

where $[R_{I-G}]$ is defined as

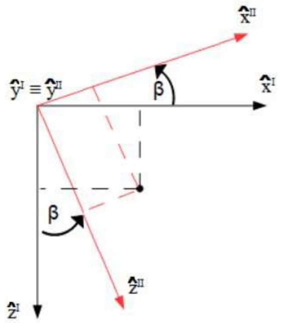
$$[R_{I-G}] = \begin{bmatrix} \cos(\gamma) & \sin(\gamma) & 0 \\ -\sin(\gamma) & \cos(\gamma) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.4}$$

and

$$\mathbf{d}^G = [R_{I-G}]^T\mathbf{d}^I \tag{2.5}$$

2. Pitch rotation (β). Referring to figure 2.6 and looking at the $x - z_{PLANE}$

Figure 2.9: Pitch rotation seen from $x-z_{PLANE}$.



$$\mathbf{d}^{II} = [R_{II-I}]\mathbf{d}^{I} \tag{2.6}$$

where $[R_{II-I}]$ is defined as

$$[R_{II-I}] = \begin{bmatrix} \cos(\beta) & 0 & -\sin(\beta) \\ 0 & 1 & 0 \\ \sin(\beta) & 0 & \cos(\beta) \end{bmatrix} \tag{2.7}$$

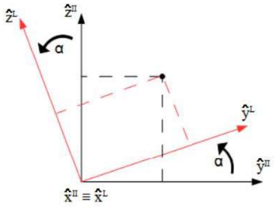
and

$$\mathbf{d}^{I} = [R_{II-I}]^T \mathbf{d}^{I} I \tag{2.8}$$

3. Roll rotation (α). Referring to figure 2.7 and looking at the $y-z_{PLANE}$

$$\mathbf{d}^{L} = [R_{L-II}]\mathbf{d}^{II} \tag{2.9}$$

Figure 2.10: Roll rotation seen from $y-z_{PLANE}$.



where $[R_{L-II}]$ is defined as

$$[R_{L-II}] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha) & \sin(\alpha) \\ 0 & -\sin(\alpha) & \cos(\alpha) \end{bmatrix} \tag{2.10}$$

and

$$\mathbf{d}^{II} = [R_{L-II}]^T \mathbf{d}^{L}. \tag{2.11}$$

Once the three consecutive rotations are defined it is possible to obtain the relationship between $\mathbf{d}^G$ and $\mathbf{d}^L$:

$$\mathbf{d}^{L} = [R_{L-II}]\mathbf{d}^{II} = [R_{L-II}][R_{II-I}]\mathbf{d}^{I} = [R_{L-II}][R_{II-I}][R_{I-G}]\mathbf{d}^{G} \tag{2.12}$$

and the overall rotation matrix is:

$$[R_{LG}] = [R_{L-II}][R_{II-I}][R_{I-G}] \tag{2.13}$$

$$[R_{LG}] =$$

$$\begin{bmatrix} c(\gamma)c(\beta) & s(\gamma)c(\beta) & -s(\beta) \\ c(\gamma)s(\beta)s(\alpha)-s(\gamma)c(\alpha) & s(\gamma)s(\beta)s(\alpha)-c(\gamma)c(\alpha) & c(\beta)s(\alpha) \\ c(\gamma)s(\beta)c(\alpha)+s(\gamma)s(\alpha) & s(\gamma)s(\beta)c(\alpha)-c(\gamma)s(\alpha) & c(\beta)c(\alpha) \end{bmatrix} \tag{2.14}$$

To obtain the rotation matrix required for the reverse transformation (from local to global),

$$\mathbf{d}^G = [R_{GL}]\mathbf{d}^L$$

it is necessary to compute the inverse of the previously calculated rotation matrix. Since the planar rotation matrices are defined as orthogonal, their inverse matrix is equivalent to their transpose. Additionally, when a matrix is obtained as a product of orthogonal matrices, it remains orthogonal. Therefore, the matrix $[R_{LG}]$ is also orthogonal, leading to the following relationship:

$$[R_{GL}] = [R_{LG}]^{-1} = [R_{LG}]^T = \tag{2.15}$$

$$[R_1] = [R_{GL}] = [R_{I-G}]^T [R_{II-I}]^T [R_{L-II}]^T = [R_{G-I}][R_{I-II}][R_{II-L}] \tag{2.16}$$

Before the next section, there is still a velocity analysis that needs to be conducted. To describe the absolute velocity of a specific point on the moving platform, such as its center of mass "*cm*", Rival's theorem for rigid bodies can be applied:

$$v_{cm}^{TD} = v_{tcp}^{TD} + \omega^{TD} \times (cm - TCP). \tag{2.17}$$

In matrical form it becomes:

$$v_{cm}^{TD} = v_{tcp}^{TD} + [R_{GL}]\omega^G [\tilde{x}_{cm-tcp}^L]\omega^L \tag{2.18}$$

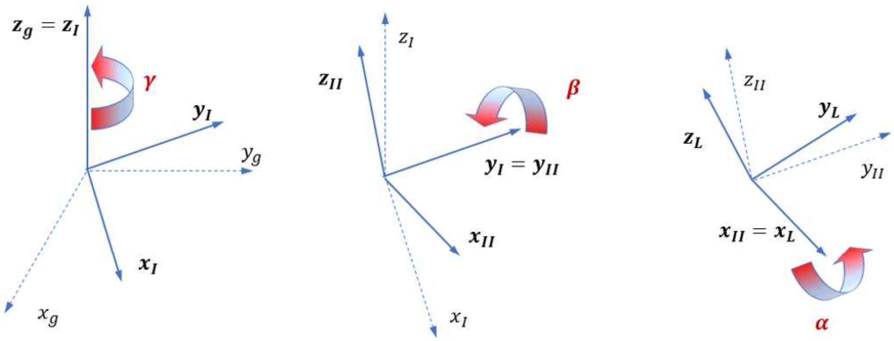
where $[\tilde{x}_{cm-tcp}^L]$ represents a skew-symmetric matrix. It depends on the local coordinates of *cm*:

$$[\tilde{x}_{cm-tcp}^L] = \begin{bmatrix} 0 & z_{cm}^L & -y_{cm}^L \\ -z_{cm}^L & 0 & x_{cm}^L \\ y_{cm}^L & -x_{cm}^L & 0 \end{bmatrix}. \tag{2.19}$$

This skew matrix allows to switch from angular to linear speed in equation 2.18.
The angular velocity vector is defined as the ratio of the infinitesimal rotation caused by infinitesimal changes in the Cardan angles ($d\alpha$, $d\beta$, $d\gamma$) over the infinitesimal time interval dt in which the considered rotation occurs. Based on the definition of the Cardan angles, the angular velocity vector can be expressed as the sum of the following three components:

- along $z^{TD} \equiv z^{I}$ a component with $\dot{\gamma}$ intensity
- along $y^{I} \equiv z^{II}$ a component with $\dot{\beta}$ intensity;
- along $x^{I} \equiv x^{L}$ a component with $\dot{\alpha}$ intensity.

Figure 2.11: consecutive rotations.



In the *Top-Disk RS* it is possible to write:

$$\omega^{TD} = \begin{Bmatrix} 0 \\ 0 \\ \dot{\gamma} \end{Bmatrix} = [R_{TD-I}] \left[\begin{Bmatrix} 0 \\ \dot{\beta} \\ 0 \end{Bmatrix} + [R_{I-II}] \begin{Bmatrix} \dot{\alpha} \\ 0 \\ 0 \end{Bmatrix} \right]. \tag{2.20}$$

The previous equation 2.20 can be rewritten as follows

$$\omega^{TD} = \begin{bmatrix} \cos(\beta)\cos(\gamma) & -\sin(\gamma) & 0 \\ \cos(\beta)\sin(\gamma) & \cos(\gamma) & 0 \\ -\sin(\beta) & 0 & 1 \end{bmatrix} \begin{Bmatrix} \dot{\alpha} \\ \dot{\beta} \\ \dot{\gamma} \end{Bmatrix} \tag{2.21}$$

$$\omega^{TD} = [A^{G}]\dot{\Theta} \tag{2.22}$$

with $\dot{\Theta}$ the vector collecting the time derivatives of the three Cardan angles and $[A_G]$ a matrix transforming the time derivatives of the Cardan angles in the components of the angular speed vector along the *Top Disk RS*.A similar procedure can be followed to de

nefine the components of the angular speed vector along the local reference:

$$\omega^{L} = \begin{Bmatrix} \dot{\alpha} \\ 0 \end{Bmatrix} = [R_{L-II}] \left[\begin{Bmatrix} 0 \\ \dot{\beta} \\ 0 \end{Bmatrix} + [R_{II-I}] \begin{Bmatrix} 0 \\ 0 \\ \dot{\gamma} \end{Bmatrix} \right]. \tag{2.23}$$

Thus, as above:

$$\omega^{TD} = \begin{bmatrix} 1 & 0 & -\sin(\beta) \\ 0 & \cos(\alpha) & \cos(\beta)\sin(\alpha) \\ 0 & -\sin(\alpha) & \cos(\alpha)\cos(\beta) \end{bmatrix} \begin{Bmatrix} \dot{\alpha} \\ \dot{\beta} \\ \dot{\gamma} \end{Bmatrix} \tag{2.24}$$

$$\omega^{L} = [A^{L}]\dot{\Theta}. \tag{2.25}$$

Substituting in Eq. 2.18

$$v^{TD}_{cm} = v^{TD}_{tcp} + [R_{GL}]\omega^{G}[\tilde{x}^{L}_{cm-tcp}][A_{L}]\dot{\Theta} \tag{2.26}$$

2.3.3 Kinematic configuration

The kinematic configuration of the Tripod has been carefully selected to meet specific criteria such as compactness, lightness, and high dynamic performance. Additionally, the cost of the Motion Simulator is a crucial consideration, emphasizing the need for affordability. Therefore, the kinematic configuration must be simple, minimizing the number of passive joints and links.

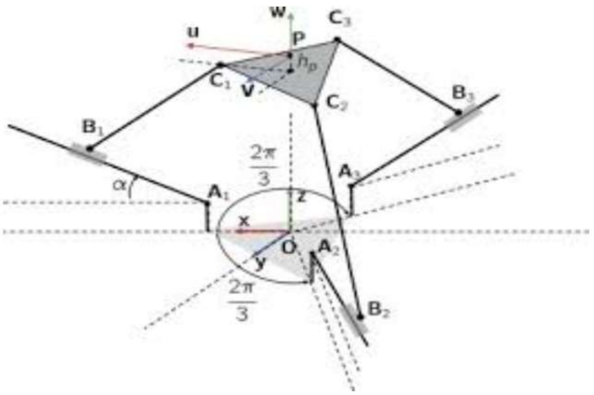
Among the various possible kinematic configurations, the RPS [1] (See Figure 2.3 in Section 2.2.3) configuration is the most suitable choice that effectively fulfills the aforementioned requirements.
The chosen configuration, RPS, offers simplicity; however, it presents a challenge due to the additional mass introduced by the moving actuators, which can limit the dynamic performance of the system. An alternative configuration, known as PRS, can be adopted to address this issue. In the PRS configuration, the actuators are positioned on the ground, resulting in grounded prismatic joints. This modification allows for a simpler kinematic structure while effectively resolving the problem of the moving actuators and their impact on system performance.

An illustrative example of the PRS configuration can be observed in Figure 2.12. This particular configuration consists of three links, each connected to a slider via a Revolute joint and to the top-disc through a Spherical joint. The sliders are actuated by Prismatic joints. Although this configuration offers the advantage of enabling significant rotations with a small-sized top-disc, it exhibits some drawbacks. The presence of protruding sliders extending beyond the projection of the top-disc results in a bulky structure. Additionally, this configuration imposes substantial forces on the actuators, which may pose challenges in terms of actuator capacity.

---

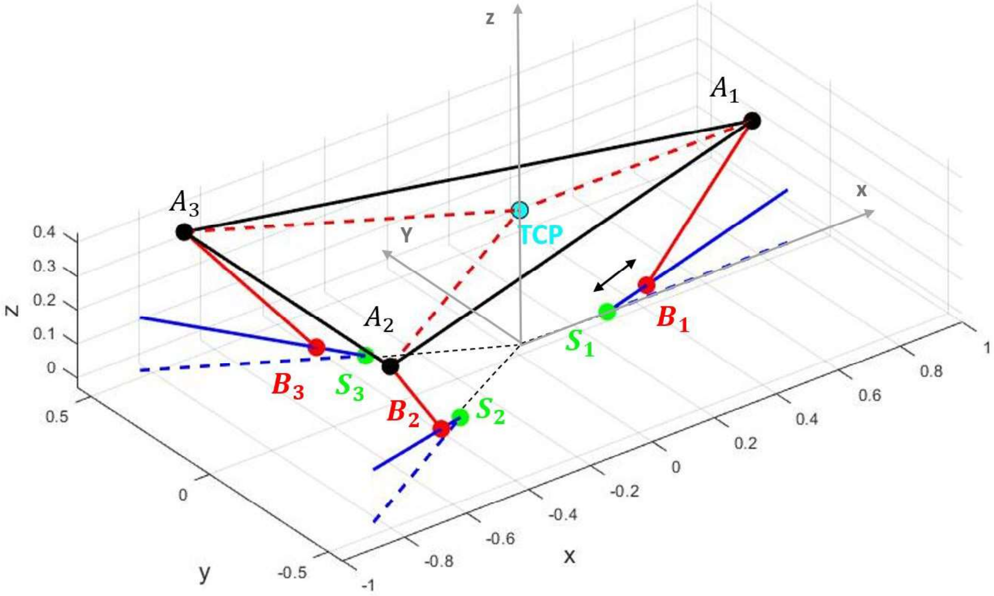
[1] R: Rotational, P: Prismatic, S: Spherical. The underlined letter refers to the actuated joint.

Figure 2.12: Example of PRS configuration for a RPZ PKM



In the presented thesis, the configuration adopted is shown in figure 2.13. This configuration mitigates the disadvantage of large forces discharged on the actuators, allows for a larger top disk however allows lower rotations but still acceptable. Analyzing the configuration presented, it can be seen that it is composed by three identical kinematic chains which lay in the three actuation planes (see section 2.3.4 and figure 2.14). Each kinematic chain is made up of: a linear guide (the blue line), on which a slider moves; a point S which identifies the starting point of the actuator; a link (the red line) which connects the center of the revolute joint (B), which is fixed on the slider, to the center of the spherical joint (A).

Figure 2.13: Custom configuration of the RPZ involved in the presented work

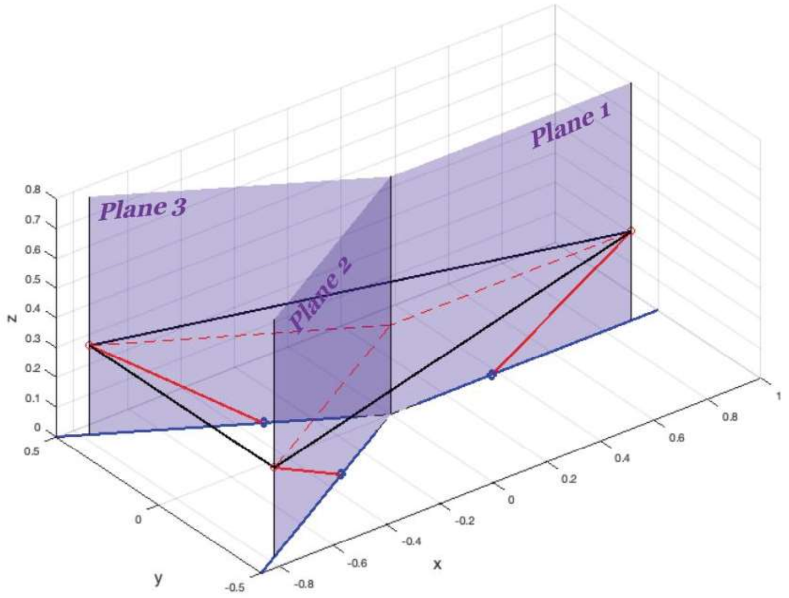


2.3.4 Geometric parametrization

Given the tools defined in the previous section, the kinematic analysis that will be performed will be fully parameterized: all the required dimensions will be considered as undefined parameters. Thanks to this parameterization, it is quick and easy to analyze new kinematic configurations. This is a major advantage, especially during the design and optimization of a future machine.

The configuration here considered is characterized by three actuation planes. The links are constrained to lay in these planes. In the study of other possible configurations by referring to these actuation planes, it could be possible to obtain other geometric configurations by adding a y-component in the Actuation R.S. to the points S_2 and S_3 to allow the second and third kinematic chains to lay in planes parallel to the actuation planes of the initial configuration.

Figure 2.14: Actuation planes for the considered configuration

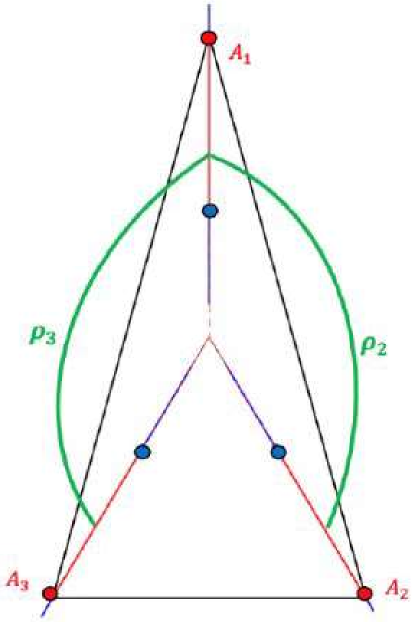


In the following table (Table 2.4) are listed the symbols and the relative descriptions of the fundamental quantities (See fig. 2.13) The first actuation plane coincides with the $x - z_{PLANE}$ of the Global reference system. The remaining two planes are obtained by rotations around the global z-axis. The parameter that defines this rotation ρ. Specifically, They are denoted as ρ_2 and ρ_3 as shown in figure 2.15. The starting points of the actuation coordinates S have a y-component in the actuation planes equal to zero, and a z-component that can have any value but for graphical reasons will be considered equal to zero.

Symbol	Description
TCP	Tool Center Point (Center of the Top-Disk)
A_i	Position of the spherical joint of the i^{th} actuation plane ($i = 1, 2, 3;$)
B_i	Position of the revolute joint in the i^{th} actuation plane
S_i	Starting point of the actuation coordinate in the i^{th} actuation plane
ρ_i	Angle of the i^{th} actuation plane with respect to the $x - z_{PLANE}$
d_i	Distance between TCP and A_i
L	Length of the links
q_i	Actuation coordinate (i^{th} actuation plane)
ψ_i	Angle between the link and the base
μ_i	Angle between the actuation guide and the horizontal in the i^{th} actuation plane
θ_x	Angle between the link and the top-disk in the $y - z_{PLANE}$ (Actuation R.S.)
θ_y	Angle between the link and the top-disk in the $x - z_{PLANE}$ (Actuation R.S.)

Table 2.4: Nomenclature for kinematic analysis

Figure 2.15: Angles of the auction planes.



2.3.5 Definition of Parasitic Motions

[...]

2.3.6 Kinemati Crosstalk modelization

[...]

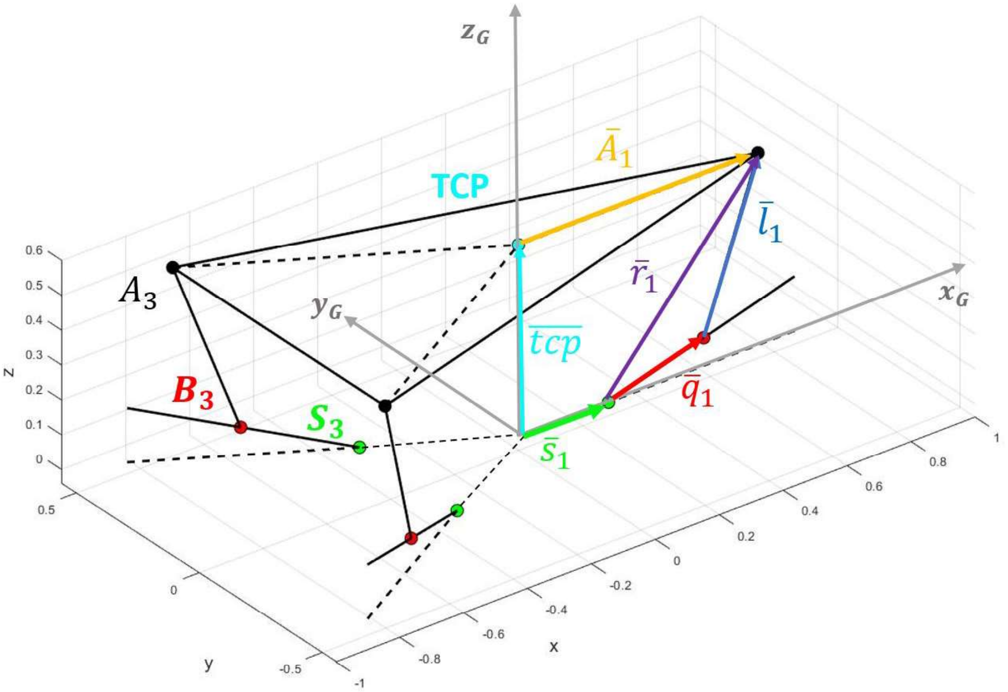
2.3.7 Inverse Kinematic

Thanks to Inverse Kinematics, we can establish the relationship between the workspace coordinates and the joint coordinates. The former represents the platform's pose (α,β,z), while the latter represents the actuation coordinates, namely the position of the three sliders with respect to the guide (**q**).
In order to solve the inverse kinematic it is necessary to define two vectorial closing equation.

In the Home Position, the TCP (Tool Center Point) lies on the z-axis of the *Global R.S.*, centered at point O. However, in positions other than the home position, the TCP not only moves along the z-direction (one of the 3 Degrees of Freedom of the system) but also along the x and y directions, as explained in the previous section, due to the presence of undesired Degrees of Freedom.

Once TCP position has been determined by solving the system of equations [...] in *Global R.S.*, it is also possible to solve the first of the two necessary vectorial closing equation required to solve the inverse kinematics.

Figure 2.16: Stylized model of the machine with vectorial closures drawn



Referring to Fig. 2.16 the aforementioned first 3D vectorial equation for the *i-th* chain can be written as :

$$\mathbf{r}_i = \mathbf{tcp} + [R_1]\mathbf{A}_i^L - \mathbf{s}_i \tag{2.27}$$

Using the second vectorial closure it is possible to write :

$$\mathbf{l}_i = \mathbf{r}_i - q_i\mathbf{u}_i. \tag{2.28}$$

By squaring Eq. 2.28 can be obtained

$$l_i^2 = \mathbf{l}_i^T\mathbf{l}_i = (\mathbf{r}_i q_i \mathbf{u}_i)^T(\mathbf{r}_i q_i \mathbf{u}_i) = \mathbf{r}_i^T\mathbf{r}_i - 2\mathbf{r}_i^T\mathbf{u}_i q_i + q_i^2. \tag{2.29}$$

If sufficient attention is paid to Eq. 2.29 it can be rewritten as a second order equation in q_i:

$$q_i^2 - 2\mathbf{r}_i^T\mathbf{u}_i q_i + \mathbf{r}_i^T\mathbf{r}_i - l_i^2 = 0 \tag{2.30}$$

which has the following solution:

$$q_i = \mathbf{r}_i^T\mathbf{u}_i - \sqrt{\mathbf{r}_i^T(\mathbf{u}_i\mathbf{u}_i^T - [I])\mathbf{r}_i + l_i^2} \tag{2.31}$$

. The other solution of equation 2.31 is neglected because of the specific configuration of the guides.

Thus, the solving procedure of the tripod's inverse kinematic can be summerized in the following steps (repeated for the three chains):

- definition of the vector $\mathbf{r}_i$. It can be performed the kinematic crosstalk problem is already solved and the DoFs of the platform are known thus, the matrix $[R_1]$ is completely defined.
- definition of the actuation coordinate q_i through equation 2.31.

Given the relationship between the platform DoFs and the actuation coordinates q_i, in order to describe the relation between the velocity of the tcp and the ones of the joints coordinates, it is neccessary to define the Jacobian matrix referred to as $[J]$. Because of the chosen conventions, what will be defined is the inverse of the Jacobian matrix $[J_{inv}]$since it represents the transformation from TCP's velocities and the joints' velocities.

$$\dot{\mathbf{q}} = \begin{Bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \\ \dot{q}_4 \\ \dot{q}_5 \\ \dot{q}_6 \end{Bmatrix} \begin{array}{c} \xrightarrow{[J]} \\ \xleftarrow[{[J_{inv}]}]{} \end{array} \begin{Bmatrix} x \\ y \\ z \\ \alpha \\ \beta \\ \gamma \end{Bmatrix} = \dot{\mathbf{X}}. \tag{2.32}$$

Due to the specific platform considered, a 3 DoF platform, not all 6 DoFs showed in the equation 2.32 are controllable. This, since a numerical definiton of J_{inv}, is here considered it will results in some zero component of the Inverse Jacobian Matrix J_{inv} (or Jacobian matrix$[J]$). From Eq. 2.32

$$\dot{\mathbf{q}} = [J_{inv}]\dot{\mathbf{X}}, \tag{2.33}$$

and deriving this equation it is possible to obtain the following, useful for joints' accelerations:

$$\ddot{\mathbf{q}} = [J_{inv}]\ddot{\mathbf{X}} + [\dot{J}_{inv}]\dot{\mathbf{X}}. \tag{2.34}$$

Two different Inverse Jacobian matrices are here presented based on where they will be used:

$$[J_{inv}^{3x3}] = \begin{bmatrix} j_{inv}^{1,1} & j_{inv}^{1,2} & j_{inv}^{1,3} \\ j_{inv}^{2,1} & j_{inv}^{2,2} & j_{inv}^{2,3} \\ j_{inv}^{3,1} & j_{inv}^{3,2} & j_{inv}^{3,3} \end{bmatrix}. \tag{2.35}$$

$$[J_{inv}^{3x6}] = \begin{bmatrix} 0 & 0 & j_{inv}^{1,3} & j_{inv}^{1,1} & j_{inv}^{1,2} & 0 \\ 0 & 0 & j_{inv}^{2,3} & j_{inv}^{2,1} & j_{inv}^{2,2} & 0 \\ 0 & 0 & j_{inv}^{3,3} & j_{inv}^{3,1} & j_{inv}^{3,2} & 0 \end{bmatrix}. \tag{2.36}$$

In chapter 3, their usage will be clarified.

Given that the PKM here considered is a 3 DoFs PKM Eq. 2.33 applied in this particular case:

$$\dot{\mathbf{q}} = \begin{Bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{Bmatrix} \underset{[J_{inv}^{3x3}]}{\overset{[J^{3x3}]}{\rightleftarrows}} \begin{Bmatrix} \alpha \\ \beta \\ z_{tcp} \end{Bmatrix} = \dot{\mathbf{X}}. \tag{2.37}$$

It is worth pointing out that also the order in the tcp coordinate has changed. From the common name of this particular configuration of PKM, RPZ, the coordinate vector of the tcp is $[\alpha, \beta, z_{tcp}]$.

The inverse kinematics for the considered PKM already introduced can be used to define the inverse jacobian matrix. The inverse kinematics relationship can be summarized as:

$$\mathbf{q} = f(\mathbf{X}) \tag{2.38}$$

and specifically, it is represented by the function $f(\mathbf{X})$. Therefore, the inverse of the Jacobian matrix,$[J_{inv}]$,can be obtained by deriving the previous relation (eq. 2.38) w.r.t. real DoF of the considered PKM:

$$\begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} = \begin{bmatrix} \frac{\delta q_1(X)}{\delta\alpha} & \frac{\delta q_1(X)}{\delta\beta} & \frac{\delta q_1(X)}{\delta z_{tcp}} \\ \frac{\delta q_2(X)}{\delta\alpha} & \frac{\delta q_2(X)}{\delta\beta} & \frac{\delta q_2(X)}{\delta z_{tcp}} \\ \frac{\delta q_3(X)}{\delta\alpha} & \frac{\delta q_3(X)}{\delta\beta} & \frac{\delta q_3(X)}{\delta z_{tcp}} \end{bmatrix} \begin{bmatrix} \dot{\alpha} \\ \dot{\beta} \\ \dot{z}_{tcp} \end{bmatrix} = [J_{inv}^{3x3}] \begin{bmatrix} \dot{\alpha} \\ \dot{\beta} \\ \dot{z}_{tcp} \end{bmatrix} \tag{2.39}$$

thus

$$[J_{inv}^{3x3}] = \begin{bmatrix} \frac{\delta q_1(X)}{\delta\alpha} & \frac{\delta q_1(X)}{\delta\beta} & \frac{\delta q_1(X)}{\delta z_{tcp}} \\ \frac{\delta q_2(X)}{\delta\alpha} & \frac{\delta q_2(X)}{\delta\beta} & \frac{\delta q_2(X)}{\delta z_{tcp}} \\ \frac{\delta q_3(X)}{\delta\alpha} & \frac{\delta q_3(X)}{\delta\beta} & \frac{\delta q_3(X)}{\delta z_{tcp}} \end{bmatrix} = \begin{bmatrix} j_{inv}^{1,1} & j_{inv}^{1,2} & j_{inv}^{1,3} \\ j_{inv}^{2,1} & j_{inv}^{2,2} & j_{inv}^{2,3} \\ j_{inv}^{3,1} & j_{inv}^{3,2} & j_{inv}^{3,3} \end{bmatrix} \tag{2.40}$$

The Jacobian matrix defined as the one shown in equation 2.3.7, is referred to as the numerical Jacobian. This computation is performed by introducing a small variation (δX) to a real degree of freedom (DoF) of the TCP and then evaluating the corresponding variation in the actuation coordinates. The final step employs an approximation method, such as the two-point backward approximation, to estimate the derivative:

- A small perturbation of the i^{th} DoF of the TCP is introduced;
- The inverse kinematic is solved for $\mathbf{X}$ and $\mathbf{X}+\delta\mathbf{X}$, where $\delta\mathbf{X}$ contains only δx_i). Referring to , the first row is obtained by introducing only $\delta\alpha$ and so on.
- The respective positions in the joint space are obtained $\mathbf{q}(\mathbf{X}+\delta\mathbf{X})$ and $\mathbf{q}(\mathbf{X})$.
- An approximation method is employed to calculate the derivative. In the following notation, in order to separate the considered degree of freedom (DoF) at the TCP into the joint space coordinates and the corresponding platform DoF the former will be referred to with the index "j"

$$j_{inv}^{j,i} = \frac{q_j(\mathbf{X}+\delta\mathbf{X}) - q_j(\mathbf{X})}{\delta x_i} \tag{2.41}$$

- This procedure must be repeated for each one of the DoF at the TCP.

Once the evaluation of each term is performed, the required inverse Jacobian matrix is completely defined, the velocity and the acceleration in the joints' coordinates space can be defined ($\dot{\mathbf{q}}$, $\ddot{\mathbf{q}}$).

2.3.8 Kinetostatic analysis

The Kinostatic analysis yields the actuation force $\mathbf{f}_Q$ required to balance the external forces applied to the TCP $\mathbf{f}_{e,tcp}$ of the considered machine. Applying the virtual work principle it is possible to obtain and to describe this relationship:

$$\delta\mathbf{X}^T\mathbf{f}_{e,tcp} + \delta\mathbf{q}^T\mathbf{f}_Q = 0 \tag{2.42}$$

Considering that the virtual change in the workspace coordinates is connected to the virtual change in the joint coordinates through this relationship:

$$\delta\mathbf{X} = [J]\delta\mathbf{q} \tag{2.43}$$

it is then possible to reformulate the virtual work principle (Eq. 2.42) as

$$\delta\mathbf{q}^T([J]^T\mathbf{f}_{e,tcp} + \mathbf{f}_Q) = 0. \tag{2.44}$$

Looking at this equation (Eq. 2.44) and knowing that this relation should be valid for each virtual variation, it is possible to delete them and to obtain:

$$[J]^T \mathbf{f}_{e,tcp} + \mathbf{f}_Q = 0 \tag{2.45}$$

$$\mathbf{f}_Q = -[J]^T \mathbf{f}_{e,tcp}. \tag{2.46}$$

It is worth noticing, from Eq. 2.46, that external forces applied to the TCP are considere negative by convention.
It is interesting also to analyse how the actuation forces act on the TCP:

$$\mathbf{f}_Q = [J]^T \mathbf{f}_{tcp} \tag{2.47}$$

where $\mathbf{f}_{tcp}$ represents the results of the actuation forces on the TCP. In this case the Jacobian matrix becomes the so-colled transmission ratio.

2.3.9 Constraints

In order label a certain pose as reachable, a set of condition must be satisfied. These constraints can be classified in 3 categories. The limits introduced in the following list depend on the application (i.e. on the encumbrance of each physical component).

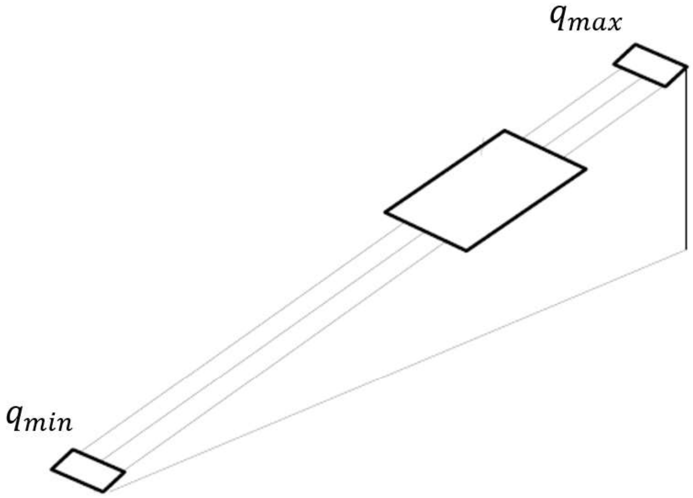
- **Kinematic constraints** The kinematic constraints state for those conditions associated with the physical ability to reach a pose, not considering surfaces' collisions and the involved forces. Primarily, they are related to the feasibility of closing individual kinematic chains. This involves verifying whether a given pose, along with a set of closure vectors' lengths or ranges, allows for a valid solution. Thus, these conditions involve the verification of the existence of a unique admissible solution to the Inverse Kinematics problem.

- **Geometric constraints** These conditions pertain to the potential occurrence of collisions between the robot's components and external objects. Additionally, the availability of mounting space is considered as a constraint within this category.
- **Kinetostatic constraints** Conditions related to the Jacobian matrix, and thus to the transmission ratio.

Kinematic constraints

- **Actuator stroke** The coordinate of the actuated joint (qi) must be constrained within a specific interval. The specified minimum and maximum values have an impact on the size of the robot and can determine the required angular range between its parts. The constraint on the actuator stroke will depend on the actual component which is used to realize the actuator but also on the encumbrance of the slider which moves on it. The actuation coordinates can vary between q_{min} and q_{max}, which is the maximum achievable distance between points S and B (see Fig. 2.16).

Figure 2.17: Actuator stroke



$$q_{min} \leq q_i \leq q_{max} \tag{2.48}$$

- **Joint mobility range** The motion of the link must be feasible in terms of mobility of both base and platform passive joints.
  Some preliminary steps are required to derive the mentioned joint mobility angles. It is necessary to derive the unitary vector describing the direction of the link:

$$\mathbf{n}_i = \frac{\mathbf{l}_i}{l_i} = \frac{\mathbf{r}_i - q_i\mathbf{u}_i}{l_i} \tag{2.49}$$

By considering the revolute joint at the center of their range, when the robot is in its home position (*hm*), it is possible to use the same formula (Eq. 2.49) to compute the unitary vector in *home position*:

$$\mathbf{n}_{i,hm} = \frac{\mathbf{l}_{i,hm}}{l_{i,hm}} = \frac{\mathbf{r}_{i,hm} - q_i\mathbf{u}_i}{l_i} \tag{2.50}$$

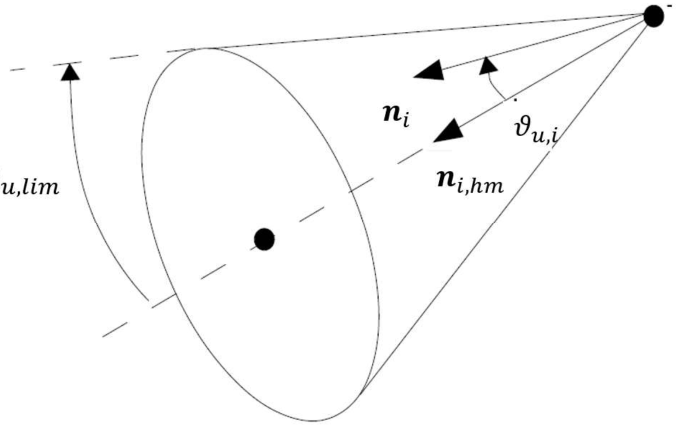
At this point it is possible to compute the rotation angle of each of the joint and compare it with their mobility range. For the platform is is necessary to introduce that in terms of constructive solution, the spherical joints are replaced by a universal joint. By switching from a spherical joint to a universal joint, one degree of freedom is inevitably lost. To regain this degree of freedom, the chosen approach is to allow for unrestricted relative rotation of the two ends of the link along its axis. Globally, when referring to the platform joint it will always be used the term spherical. Therefore, it is necessary to introduce its limit's angle. For the platform joint whose angle is defined as θ_u ("u stays for Universal"):

$$\theta_{u,i} = a\cos(([R_1(\alpha,\beta,\gamma_{kc})](-\mathbf{n}_i))^T \times \mathbf{n}_{i,hm}) \tag{2.51}$$

thus

$$\theta_{u,i} \leq \theta_{c,lim}. \tag{2.52}$$

Figure 2.18: Cardan mobility range



Instead for the base passive joint (revolute joint) whose angle is defined as θ_R (R stays for Revolute):

$$\theta_{R,i} = \psi_i - \psi_{i,hm} \tag{2.53}$$

where

$$\psi_i = a\cos(\mathbf{n}_i{}^T \times \mathbf{u}_i) \tag{2.54}$$

and $\psi_{i,hm}$ using the same equation (Eq. 2.54) applied to the *home position*:

$$\psi_{i,hm} = a\cos(\mathbf{n}_{i,hm}{}^T \times \mathbf{u}_i) \tag{2.55}$$

The angle performed by each revolute joint will have an upper and lower limit which will depends on how the joint is manufactured:

$$\theta_{R,min} \leq \theta_{R,i} \leq \theta_{R,imax}. \tag{2.56}$$

Geometric constraints

In order to define the angles for the evaluation of geometric constraints, it is useful to introduce the unit vector perpendicular to the top-disc $\mathbf{n}_{TD}^{G}$ (*Grounded R.S.*) which can be obtained by rotating a vertical unit vector by means of the rotation matrix R_1:

$$\mathbf{n}_{TD}^{G} = [R_1] \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \tag{2.57}$$

It is then possible to rotate the vector from the *Grounded R.S.* to the *Actuation R.S.* by means of the matrix $[R_2]$

$$\mathbf{n}_{TD}^{A} = [R_2]\mathbf{n}_{TD}^{G} \tag{2.58}$$

The components of $\mathbf{n}_{TD}^{A}$ are the direction cosines, i.e., the cosines of angles that the vector forms with the coordinate axes:

$$\mathbf{n}_{TD}^{A} = \begin{bmatrix} n_x \\ n_y \\ n_z \end{bmatrix} = \begin{bmatrix} \cos(\delta) \\ \cos(\xi) \\ \cos(\eta). \end{bmatrix} \tag{2.59}$$

- **Collision link-platform** The angles of interest are: δ which is the angle between the top-disc and x-axis ξ which is the angle between the top-disc and y-axis. These angles can be easily obtained from the Eq. 2.59 as:

  $$\delta = acos(n_x) \tag{2.60}$$

  $$\xi = acos(n_y) \tag{2.61}$$

  It is now possible to calculate the angle between the top-disc and the link in the xz-plane:

  $$\theta_y = \frac{3}{2}\pi[\delta + (\pi - \psi)] = \frac{\pi}{2} + \psi - \delta \tag{2.62}$$

$$\theta_x = \xi. \tag{2.63}$$

Finally, the costraints are:

$$\theta_{x,min} < \theta_y a < \theta_{x,max}. \tag{2.64}$$

$$\theta_{y,min} < \theta_y < \theta_{y,max}. \tag{2.65}$$

- **Collision link-slider** A constraint on the angle ψ is necessary to avoid impacts between link and slider:

$$\psi_{min} < \psi < \psi_{max}. \tag{2.66}$$

Kinetostatic constraints

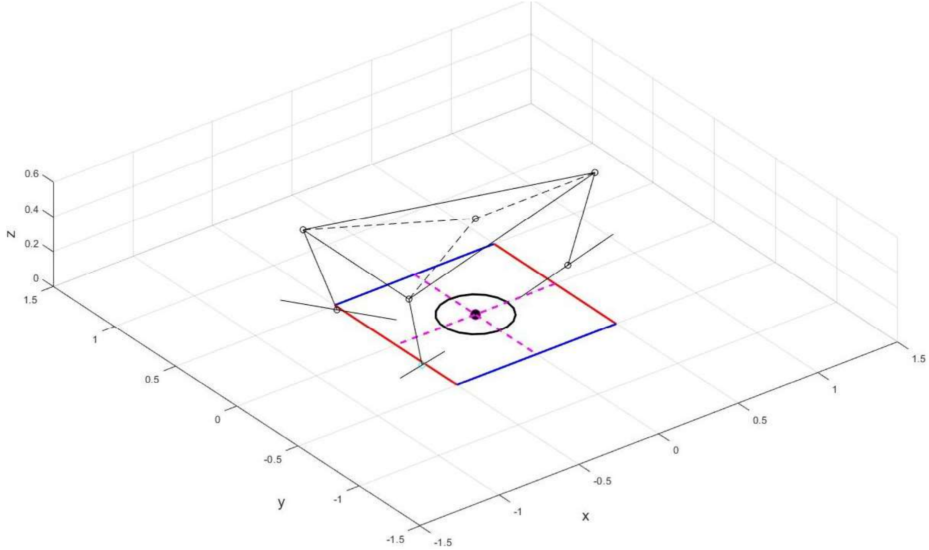
- **Singularity** To ensure the avoidance of singularity, it is necessary for the determinant of the Jacobian matrix to differ from zero by a minimum tolerance. This condition overlaps with another geometry check that explicitly identifies singular configurations. However, it allows for the straightforward exclusion of all singular configurations from the workspace.

2.4 Cartesian's kinematic analysis

Now, let's shift the focus to the lower stage. As it will be noticed, the discussion regarding the lower stage will be relatively shorter, particularly when it comes to defining the kinematics, in comparison to what has been previously discussed for the tripod.
This particular robot is designed to provide motion along the x and y axes, as well as rotation around the yaw axis. These three degrees of freedom (x, y, and yaw (γ)) are the primary capabilities of this robot.

The cartesian robot, considered at this stage, in order to reproduce the degrees of freedom is composed of 4 linear axes coupled two by two for x and y; instead, a separate implementation is used for the yaw.

Figure 2.19: Stylized Cartesian coupled with tripod



Given the very simple kinematics involved, in this case there is no need for a true kinematic analysis in order to bind the Cartesian degrees of freedom and the coordinates of the joint space.

$$\mathbf{q}_{cart} = \begin{bmatrix} q_{cart,1} \\ q_{cart,2} \\ q_{cart,3} \end{bmatrix} = \begin{bmatrix} x \\ y \\ \gamma \end{bmatrix} = \mathbf{X}_{cart} \tag{2.67}$$

On the other hand, the crucial aspect will be defining the interaction between the Cartesian and the tripod systems in terms of forces and moment. The reference systems used will be briefly introduced below. Then the aforementioned interactions between the two systems will be analyzed

2.4.1 Reference systems

As for the introduced upper stage (3 DoFs RPZ PKM), also for the lower stage, the definition of reference systems and their rotation matrices is vital (refers to figure 2.20):

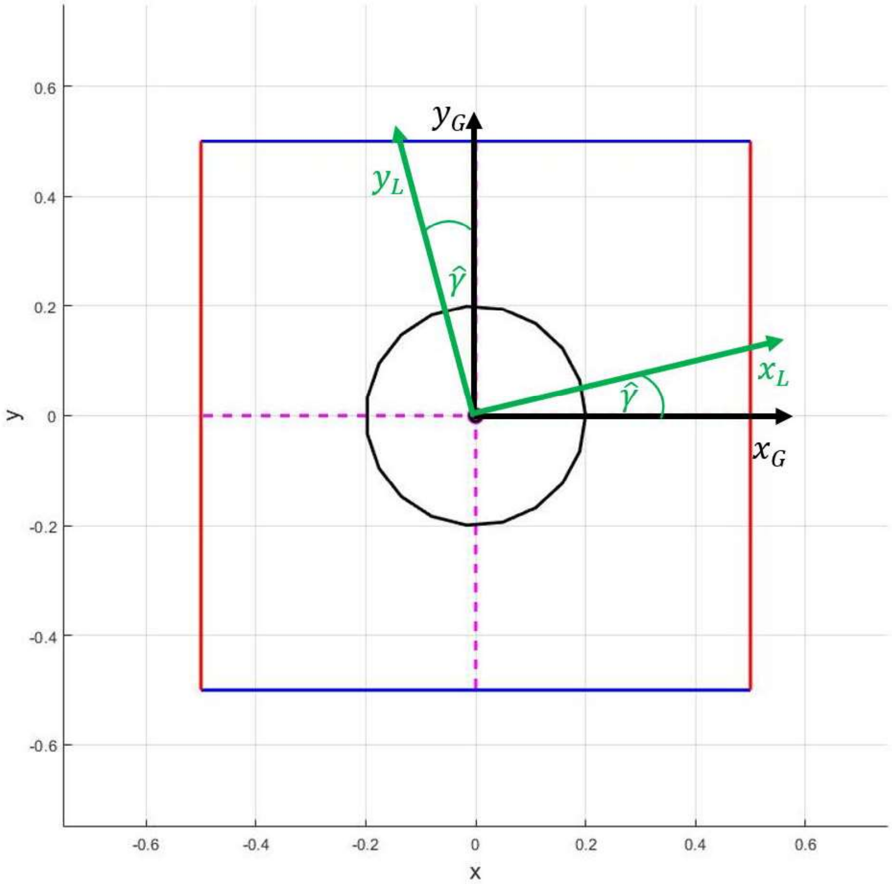
- **Global R.S.** The cartesian global reference system has its origin positioned in the center of the cartesian x-y workspace. It matches also the intersection point between the cartesian x-y workspace and the axis around which the yaw rotation is applied to the robot. The z-axis points upward. The main difference between this *Global R.S.* and the one introduced in section 2.3.1 is just the yaw angle. The *Global R.S.* of section 2.3.1 is considered unrotated by the yaw angle. To differentiate between the two , reference is henceforth made to the stage to which they belong:

Cartesian (Lower Stage) **Global R.S.** $X^{G,LS}$

Tripod (Upper Stage) **Global R.S.** $X^{G,US}$

- **Local R.S.**: similar to the Local R.S: defined in section refsec: ref syst, the one here defined it has the come origin of *Cartesian Global R.S.* and it accounts for the yaw rotation $X^{L,LS}$. If the two stages (Cartesian and tripod) are mounted on each other respectively, here is that the local reference system of the Cartesian corresponds to the global reference system of the tripod (unless there is an offset on the z-axis).

Figure 2.20: Cartesian (Lower Stage) **Global R.S.** and Cartesian (Lower Stage) **Local R.S.**



In figure 2.20 the "*G,LS R.S.*" is denoted by the black arrows, instead the local one "*L,LS R.S.*" is in green. In addition, again from the same figures it is possible to see the representation of the axes actuated in x (in blue) and the axes actuated in y (red). The black circle wants to represent the torque motor that was intended to be used as the actuation for the degree of freedom yaw.

Once the kinematic and dynamics quantities are defined in the *Tripod Global R.S.* ≡ *Cartesian Local R.S.* , in order to transform them into quantities in the *Cartesian Global R.S.* the following rotation matrix is required :

$$[R_{yaw}] = \begin{bmatrix} \cos(\gamma) & -\sin(\gamma) & 0 \\ \sin(\gamma) & \cos(\gamma) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.68}$$

Chapter 3

Architecture analysis: dynamics

3.1 Tripod's dynamic analysis

Another fundamental tool required for the scope of this thesis is the dynamic analysis of the PKM. This analysis involves determining the dynamic performance of the robot, sizing the actuators and components of the actuation chain, and evaluating the structural load. These needs are not specific to the 3 DoF PKM under consideration, but rather general requirements.

Typically, a common approach at this stage is to utilize multi-body software for modeling the machine. However, since the dynamic behavior of the PKM cannot be predetermined due to interactive driving requirements and the significant variation in performance throughout its workspace, extensive analysis is necessary to properly size all components and optimize the system. While multi-body software is often used to determine loads on the structure and actuators, an analytical approach may be more efficient for running numerous simulations and easily modifying robot configuration and geometry. Moreover, this analytical model will be beneficial for future control purposes, as it takes advantage of the comprehensive parameterization, reduces computational time, and allows for immediate and automated changes in geometry and pose.

In the following procedure, the robot will be divided into two subsystems: the top disk subsystem and the actuated joints subsystem. The top disk subsystem comprises the top disk, the payload (e.g., the cockpit), and three masses representing a portion of the moving link and the joints at the top disk. The actuated joints subsystem includes the slider, the remaining portion of the moving links, and the actuation structure (motors, screws, pulleys, etc.). Since the weight of each leg is significantly lower than the payload and top disk, and their rotational inertia is negligible compared to the carried load, representing them with two masses is an acceptable and conservative simplification that facilitates the analytical formulation and provides a safety margin for actuation sizing.

To derive the complete formulation of the dynamic motion equations, the principle of kinetic energy will be employed. The objective is to compute all forces acting at the actuation level as well as the loads and reaction forces transmitted through links and passive joints.

Another key aspect of this section is the Jacobian matrix, which establishes the connection between the velocity vector at the TCP and the velocity vector in the lower joint space.

The problem discussed below is generic, as mentioned earlier, but the equation development will be focused on the 3 DoF PKM. With the necessary precautions, the presented model can be adapted for use with any PKM.

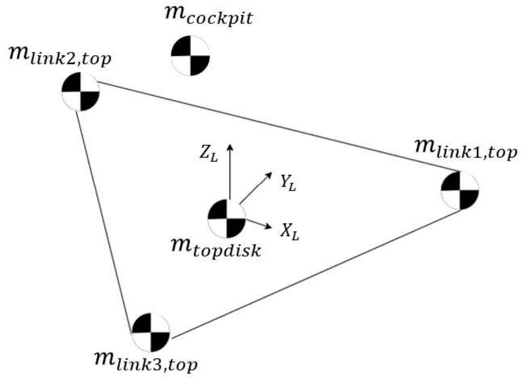
3.1.1 Dynamic model

The first step performed for the analysis introduced above is is the division of the robot in two different subsystems: the top-disk subsystem (platform subsystem) and the actuated joints subsystem (base subsystem). This subdivision is based on considering each link as two concentrate masses each of which is defined as:

$$m_{link_i,top} = \frac{m_{link_i}}{2} \tag{3.1}$$

where m_{link_i} represents the total mass of each link.
The upper subsystem is shown in figure 3.1 where the masses of links, cockpit and topdisk presented.

Figure 3.1: Representation of the masses for the top-disk subsystem (platform subsystem) .



In order to define the force at actuator level in the actuation direction, the kinetic energy theorem was invoked:

$$W_m + W_r = \frac{dEc}{dt} \tag{3.2}$$

where respectively W_m includes the work generated from the actuation section and W_r represents the work done by the resistant forces including gravity. The right-hand side (*rhs*) of equation 3.2 it is the derivative of kinetic energy with respect to time.
The complete model is characterized by a kinetic energy as :

$$E_c = \frac{1}{2}\dot{\mathbf{q}}^T[M_q]\dot{\mathbf{q}} + \frac{1}{2}\dot{\mathbf{X}}^T[M_x]\dot{\mathbf{X}} \tag{3.3}$$

thus, the *rhs* of Eq. 3.2 can derived

$$\frac{dEc}{dt} = \ddot{\mathbf{q}}^T[M_q]\dot{\mathbf{q}} + \ddot{\mathbf{X}}^T[M_x]\dot{\mathbf{x}} + \frac{1}{2}\dot{\mathbf{X}}^T[\dot{M}_x]\dot{\mathbf{X}} \tag{3.4}$$

Two matrices are introduced in equation 3.4:

> $[M_q]$ is the $[6x6]$ matrix that includes all the equivalent translating masses at the actuator (transmission components are included).
>
> $[M_x]$ is the mass matrix of the equivalent upper subsystem

$$[M_x] = \begin{bmatrix} [m_{tot}] & [m_{tot}][R_1]skew(X^L_{CoG,top})[A_l] \\ [A_l]^T(X^L_{CoG,top})^T[R_1]^T[m_{tot}] & [A_l]^T[I_{eq,top}][A_l] \end{bmatrix} \tag{3.5}$$

with

$$[I_{eq,top}] = [I_{link,top}] + [I_{topdisk}] + [I_{cockpit,tcp}] \tag{3.6}$$

$$[m_{tot}] = m_{tot}[I^{3x3}] = (m_{topdisk} + m_{cockpit} + \sum_{i=1}^{3} m_{link_i,top})tot[I^{3x3}]. \tag{3.7}$$

Eq. 3.6 refers to the Inertia tensor of the upper subsystem (whose elements are indicated by subscript "*top*"), In fact, as shown in figure 3.1 is composed by top-disk, cockpit and half of each link.

Recalling equation 2.33 and equation 2.34, the already modified *rhs* of 3.2 into equation 3.4 can be again define as:

$$\frac{dEc}{dt} = \left[\mathbf{X}_6^T\left([\dot{J}^{3x6}_{inv}]^T[M_q][J^{3x6}_{inv}] + \frac{1}{2}[\dot{M}_x]\right) + \mathbf{X}_6^T\left([J^{3x6}_{inv}]^T[M_q][J^{3x6}_{inv}] + [M_x]\right)\right]\mathbf{X}_6. \tag{3.8}$$

where

$\mathbf{X}_6$ represents the pose vector of the origin of the local reference frame at the TCP expressed w.r.t. the *grounded RD* (or *global RD*) frame thefore $\mathbf{X}_6 = [x_{kc}, y_{kc}, z_{tcp}, \alpha, \beta, \gamma_{kc},]$. The parasitic motion are not proper DoF of the PKM but since they present the pose vector must be completed in this way.

Finally, the initial equation 3.2 becomes:

$$\begin{gathered} \mathbf{F}_q^T[J^{3x6}_{inv}]\dot{\mathbf{X}}_6 + \mathbf{F}^*_{TD,ext}{}^T\dot{\mathbf{X}}_6 - g[M_{q,t}]^T[U_3][J^{3x6}_{inv}]\dot{\mathbf{X}}_6 = \\ \left[\dot{\mathbf{X}}_6^T\left([\dot{J}^{3x6}_{inv}]^T[M_q][J^{3x6}_{inv}] + \frac{1}{2}[\dot{M}_x]\right) + \ddot{\mathbf{X}}_6^T\left([J^{3x6}_{inv}]^T[M_q][J^{3x6}_{inv}] + [M_x]\right)\right]\dot{\mathbf{X}}_6 \end{gathered} \tag{3.9}$$

From the previous equation 3.9 can be extracted the force on the actuation side $\mathbf{F}_q$ by simplifying the $\mathbf{X}_6$ term and by multypling each side for the pseudo-inverse matrix of $[J^{3x6}_{inv}]$ equivalent to $[J^{6x3}] = [J^{3x6}_{inv}]^{-1}$

$$\begin{aligned} \mathbf{F}_q^T = -\mathbf{F}^*_{TD,ext}{}^T[J^{3x6}_{inv}]^{-1} + g[M_{q,t}]^T[U_3] + \dot{\mathbf{X}}_6^T\left([\dot{J}^{3x6}_{inv}]^T[M_q] + \frac{1}{2}[\dot{M}_x][J^{3x6}_{inv}]^{-1}\right) \\ + \ddot{\mathbf{X}}_6^T\left([J^{3x6}_{inv}]^T[M_q] + [M_x][J^{3x6}_{inv}]^{-1}\right) \end{aligned} \tag{3.10}$$

where

$\mathbf{F}_q$ represent the resulting forces on actuator direction for a given dynamics and load of the tcp.

$\mathbf{F}^*_{TD,ext}$ contains all the external forces and torque (also gravitational contribution) at the tcp

$$\mathbf{F}^*_{TD,ext} = \mathbf{F}_{ext} - [D_m]\mathbf{F}_E \tag{3.11}$$

$$\mathbf{F}^T_{ext} = \begin{Bmatrix} F_{x,ext} \\ F_{y,ext} \\ F_{z,ext} \\ T_{\alpha,ext} \\ T_{\beta,ext} \\ F_{\gamma,ext} \end{Bmatrix} \quad (3.12) \qquad \mathbf{F}^T_E = \begin{Bmatrix} 0 \\ 0 \\ m_{tot} \cdot g \\ 0 \\ 0 \\ 0 \end{Bmatrix} \quad (3.13)$$

$[D_m]$ is a particular matrix that take into account the position of the CoG of the equivalent system shown in figure 3.1. In order to transform forces and torques applied at the CoG to the TCP. Hence, with $\mathbf{F}_{ext}$ are denoted the external forces applied directly at the TCP (they do not require $[D_m]$) and with $\mathbf{F}_E$ the gravitational contribution applied at the CoG (it requires $[D_m]$)

$$[D_m] = \begin{bmatrix} [I^{3x3}] & [S] \\ [S]^T & [I^{3x3}] \end{bmatrix} \tag{3.14}$$

with

$$[S] = skew([R_1]X^L_{CoG,top}). \tag{3.15}$$

The following list contains the description of the quantities already introduce in the above equations but not yet fully described:

$X^L_{CoG,top}$ is the vector definition of upper subsystem CoG in the local frame.

$[M_{q,t}]$ is a $[6x6]$ diagonal matrix whose terms are the effctive translating mass at the actuator (the half of the link mass, the bottom passive joint and the other mechanical components) without the equivalent terms coming from rotating elements like motor, screw, pulleys and eventually reduction stages. They are here excluded because this term is used for the gravitational contribution.

$[U_3]$ is a diagonal $[3x3]$ matrix that contains the actuation direction orientation w.r.t the global horizontal plane. Since for this application, each actuation direction is parallel to the horizontal plane, the movement of the equivalent translating masses associated with each actuator does not cause any potential energy variation, therefore this term is null.

$[\dot{M}_x]$ is the derivate of $[M_x]$ obatined numerically.

A short technical digression about the Jacobian matrix is nedeed before continuing with the remaining part of this section. In the previous chapter, and more precisely at the end of section 2.3.7, two different notations for two different inver Jacobian matrix were introduced. From equation 3.8 to Eq. 3.10 the particular $[J^{3x6}_{inv}]$ is employed. This matrix is involved when a relation with a 6-dimensions vector, as $\mathbf{X}_6$ is required. The single components of $[J^{3x6}_{inv}]$ are rearranged given the components of $\mathbf{X}_6$ (the position in $[J^{3x6}_{inv}]$ depends on the derivative and the corresponding position of that variable in $\mathbf{X}_6$). Using Eq. 2.33, the definition of the inverse Jacobian matrix $[J^{3x3}_{inv}]$ in Eq. eq: numeric J inv and by considering the aforementioned relation with $\mathbf{X}_6 = [x, y, z, \alpha, \beta, \gamma]$, it is possible to understand how the $[J^{3x6}_{inv}]$ can be used:

$$\begin{bmatrix} q_1 \\ q_2 \\ q_3 \end{bmatrix} = [\dot{J}^{3x6}_{inv}] \begin{bmatrix} x_{kc} \\ y_{kc} \\ z \\ \alpha \\ \beta \\ \gamma_{kc} \end{bmatrix}. \tag{3.16}$$

At this point, It should be remembered that $[x_{kc}, y_{kc}, \gamma_{kc}]$ are not DoFs of the platform but undesired motions defined as *kinematic crosstalk*.

In the Eq. 3.10 the Jacobian matrix do not take into account the terms involving parasitic motion, due to the fact that those loads transfer into reaction forces. Those components will be treated in detail in the next section.

Constraint reactions

The vector force 3.10 represents the resulting forces in the direction of the actuators for a given dynamics of the TCP. It is used when the objective of the analysis is to determine the required actuation force.

When applying the kinetic energy theorem and introducing the Jacobian matrix to write the balance with the actuation coordinates **q**, certain forces disappear from the equation as they do not contribute to the generation of work. Therefore, when the constraint reactions are required a step back is necessary. In this situation, it is useful to start from the resulting forces at the top subsystem (the one in figure 3.1). The Jacobian matrix and the mass matrix $[M_q]$ are not necessary since the balance is written on the equivalent rigid body of the figure 3.1. Equation 3.17 shows how only contributions of external forces/torques and the inertial one of the top-equivalent subsystem are included:

$$\mathbf{F}_{tcp} = -\mathbf{F}^{*}_{TD,e}{}^{T} + \frac{1}{2}\dot{\mathbf{X}}^{T}\,[\dot{M}_x] + \ddot{\mathbf{X}}^{T}\,[M_x]. \tag{3.17}$$

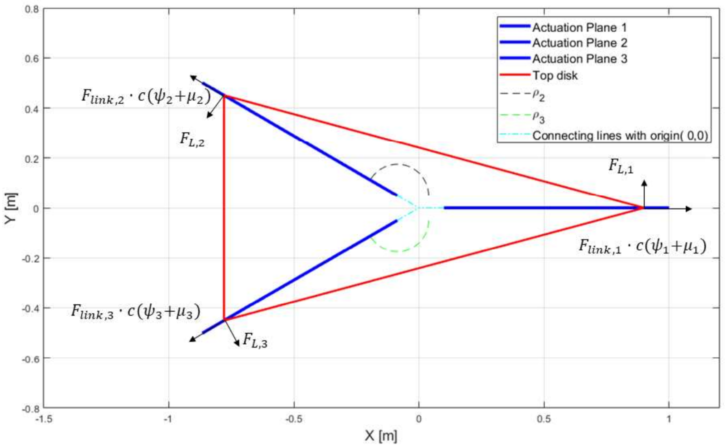
Therefore, the forces defined here in Eq. 3.17 can be considere the resultants of all forces seen from the TCP reference frame.

At this point, two more steps must be introduced in order to completely define the set of reaction forces on the structure:

1. the resultant at TCP will be used to compute the reaction forces at the three spherical joints of the platform (upper subsystem)
2. the resultant forces at spherical joints must be transported to the lower revolute/translating joints (lower subsystem).

In the x-y plane (*Global RS*) as in figure, the forces acting are composed of the link force component (considered positive in traction) and the lateral force. This is due to the mechanical constraint of the spherical joints, which are unable to transfer any torque and are bound to stay within the actuation plane. As a result, they generate the lateral reaction as a constraint and can only transfer load in the direction of the link within the actuation plane, resulting in axial load on the leg. Figure 3.2 shows joints' forces at the top-disk in *Global RS* $x - y_{plane}$.

Figure 3.2: Upper subsystem joints' forces definition in *Global RS* $x-y_{plane}$



In order to switch from the tcp forces to the spherical joint forces, here referred to as $\mathbf{F}_{s,2d}$, a new matrix must be defined $[K_{s,tcp}]$:

$$[K_{s,tcp}]\mathbf{F}_{s,2d} = \mathbf{F}_{tcp}. \tag{3.18}$$

This matrix includes all the terms necessary to switch from forces at the TCP to the forces at the top spherical joints. Referring to the figures 3.2 it is easy to identify the x-y projections of F_{link} and F_L. Respectively, the force along the link e the lateral force at each spherical joing (positive in the direction of y-axis of *actuation plane*). Thus, in the x-y plane two orthogonal forces are take into account for each joint: $F_{link_i} \cdot \cos(\psi_i + \mu_i)$ and $F_{L_i}$where $\psi + \mu$ is the angle between the link and the x-axis in the actuation plane as shown in figure 3.3 Ṙespectively, the X-contributions (*Top-Disk RS*)) are:

$$F_{x,link_i} = F_{link_i} \cdot \cos(\psi_i + \mu_i) \cdot \cos(\rho_i) \tag{3.19}$$

$$F_{x,L_i} = -\sin(\rho_i) \cdot F_{L_i}. \tag{3.20}$$

. From these two equations (Eq. 3.19 and Eq. 3.20), the first row of the aforementioned $[K_{s,tcp}]$ matrix can be defined as follow:

$$K[1,J] = \begin{cases} \cos(\psi_i+\mu_i)\cdot\cos(\rho_i) \quad with\ j=1+2(i-1) \quad for\ link-i^{th} force \\ -\sin(\rho_i) \quad with\ j=2+2(i-1) \quad for\ lateral-i^{th} force \end{cases} \tag{3.21}$$

for $j = 1,2,3,4,5,6$.

Each row of $[K_{s,tcp}]$ is composed by the repetition of two main components, dependent in pairs on the same joint index (i), therefore only the generic formulation dependent by index (I,j) will be explained.

The vector force $\mathbf{F}_{s,2d}$ is defined as

$$\mathbf{F}_{s,2d} = \begin{Bmatrix} F_{link_1} \\ F_{L_1} \\ F_{link_2} \\ F_{L_2} \\ F_{link_3} \\ F_{L_3} \end{Bmatrix}. \tag{3.22}$$

the index $i = 1,2,3$ is considered two times for each value it can assume. Given i = 1, the i^{th} upper joint is taken into account and respectively, the link-directed force comes first and the lateral force is second.

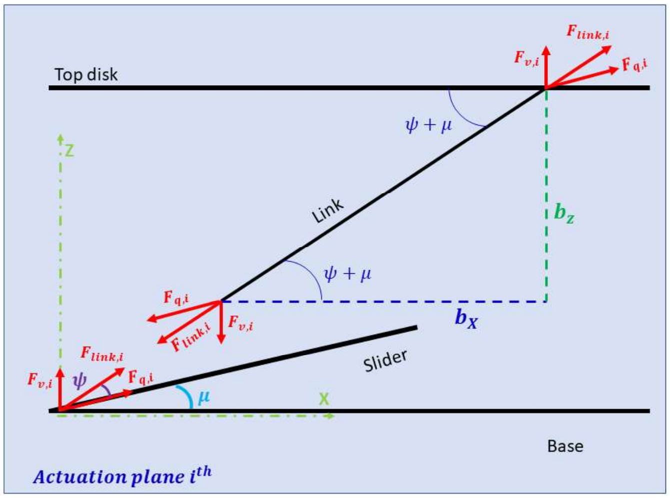
Then, the first row of $[K_{s,tcp}]$ is

$$K[1,:] = \begin{Bmatrix} \cos(\psi_1+\mu_1)\cot\cos(\rho_1) \\ -\sin(\rho_1) \\ \cos(\psi_2+\mu_2)\cot\cos(\rho_2) \\ -\sin(\rho_2) \\ \cos(\psi_3+\mu_3)\cot\cos(\rho_3) \\ -\sin(\rho_3) \end{Bmatrix}^T = \begin{Bmatrix} F_{x,link_1} \\ F_{x,L_1} \\ F_{x,link_2} \\ F_{x,L_2} \\ F_{x,link_3} \\ F_{x,L_3} \end{Bmatrix}^T. \tag{3.23}$$

Using the same notation it is possible to introduce the y-contributions that will compose the second row of $[K_{s,tcp}]$ (take as reference figure 3.2):

$$K[2,J] = \begin{cases} \cos(\psi_i+\mu_i)\cdot\sin(\rho_i) \quad with\ j=1+2(i-1) \quad for\ link-i^{th} force \\ \cos(\rho_i) \quad with\ j=2+2(i-1) \quad for\ lateral-i^{th} force \end{cases} \tag{3.24}$$

Figure 3.3: Forces definition in *Actuation RS* $x - z_{plane}$



Looking at figure 3.2 it is clear that the only $F_{x,link_i}$ produces a component along z (F_{v_i}), so the third row is

$$K[3,J] = \begin{cases} \sin(\psi_i + \mu_i) \quad with\ j = 1 + 2(i-1) \quad for\ link - i^{th} force \\ 0 \quad with\ j = 2 + 2(i-1) \quad for\ lateral - i^{th} force \end{cases} \tag{3.25}$$

The last three rows represent the components generating M_x, M_y, M_z (or T_x, T_y, T_z). In order to define them it is necessary to know the position of the spherical joint in the *Top Disk R.S.* A_i^{TD}. From the definition of A_i^{TD} it is possible to compute

$$r_i = \sqrt{A_i^{TD}(1)^2 + A_i^{TD}(2)^2} \tag{3.26}$$

which is the distance from spherical joints to TCP projected in x-y plane. Therefore:

$$K[4,J] = \begin{cases} -c(\psi_i + \mu_i) \cdot s(\rho_i) \cdot A_i^{TD}(3) + s(\psi_i + \mu_i) \cdot A_i^{TD}(2) \\ \qquad \text{with j=1+2(i-1) for link-}i^{th}\text{ force} \\ -c(\rho_i) \cdot A_i^{TD}(3) \\ \qquad \text{with j=2+2(i-1) for lateral-}i^{th}\text{ force} \end{cases} \tag{3.27}$$

$$K[5,J] = \begin{cases} c(\psi_i + \mu_i) \cdot c(\rho_i) \cdot A_i^{TD}(3) - s(\psi_i + \mu_i) \cdot A_i^{TD}(1) \\ \qquad \text{with j=1+2(i-1) for link-}i^{th}\text{ force} \\ -s(\rho_i) \cdot A_i^{TD}(3) \\ \qquad \text{with j=2+2(i-1) for lateral-}i^{th}\text{ force} \end{cases} . \tag{3.28}$$

Finally, it is possible to close the matrix considering the M_z generating forces

$$K[6,J] = \begin{cases} c(\psi_i + \mu_i) \cdot \sin(\rho_i) \cdot A_i^{TD}(2) + c(\psi_i + \mu_i) \cdot s(\rho_i) \cdot A_i^{TD}(1) \\ \qquad \text{with j=1+2(i-1) for link-}i^{th}\text{ force} \\ s(\rho_i) \cdot A_i^{TD}(2) + c(\rho_i) \cdot A_i^{TD}(1) \\ \qquad \text{with j=2+2(i-1) for lateral-}i^{th}\text{ force} \end{cases} . \tag{3.29}$$

It is now possible to solve equation 3.18 and defined the vector force $\mathbf{F}_{s,2d}$

$$\mathbf{F}_{s,2d} = [K_{s,tcp}]^{-1}\mathbf{F}_{tcp}. \tag{3.30}$$

. At this point, the two support reactions $F_{v,i}$ and $F_{q,i}$ (as shown in Figure 3.3) can be determined by considering the angle between the link and the X_G axis. Subsequently, the remaining operations can make use of the 2D representation within each actuation plane:

$$F_{q_i} = F_{link_i} \cdot \cos(\psi) \tag{3.31}$$

$$F_{v_i} = F_{link_i} \cdot \sin(\psi) \tag{3.32}$$

. The constraints reaction are then defined as:

$$\begin{aligned} \mathbf{F}_{q,tcp} = &\quad [F_{q_1}, F_{q_2}, F_{q_3}] \\ \mathbf{F}_v = &\quad [F_{v_1}, F_{v_2}, F_{v_3}] \\ \mathbf{F}_L = &\quad [F_{L_1}, F_{L_2}, F_{L_3}] \end{aligned} \tag{3.33}$$

The subscript "tcp" has been added to F_q to underline the origin of the contribution to the actuation forces. Recalling equation 3.10, the complete resultant on the actuation direction can be found by adding the inertial contribution relative to the equivalent translating mass on each actuator:

$$\mathbf{F}_q = \mathbf{F}_{q,tcp}{}^T + \dot{\mathbf{X}}_6^T [\dot{J}_{inv}^{3x6}][M_q] + \ddot{\mathbf{X}}_6^T [J_{inv}^{3x6}][M_q] \tag{3.34}$$

Lateral forces acting at the upper joint result in reaction torques at the lower joint due to the rotational constraint that allows rotation around a single axis (unlike the spherical joint). This constraint determines the fixed direction of the actuation planes. Once the forces are defined as shown in Figures 3.2, the corresponding moments at the lower joint can be evaluated:

$$M_{v_i} = F_{L_i} \cdot b_q \tag{3.35}$$

$$M_{q_i} = -F_{L_i} \cdot b_v. \tag{3.36}$$

where

$$b_{q_i} = ||\mathbf{l}_i||_2 \cdot \cos(\psi_i) \tag{3.37}$$

$$b_{v_i} = ||\mathbf{l}_i||_2 \cdot \sin(\psi_i). \tag{3.38}$$

Finally, the support reactions are now fully determined. With the desired dynamics of the end-effector specified, this analytical formulation provides an effective means to calculate all the relevant loads required for sizing the joints and structural components accurately. After the definition of the required force on the slider ($\mathbf{F}_q$), it must be defined the requested torque at the motor. Assuming to use a ball-screw linear unit, reduction stages and pulleys, the resulting transmission ration can be used to evaluate the requested torque and the corresponding angular velocity:

$$\mathbf{C}_m = \mathbf{F}_q^T \cdot \tau \tag{3.39}$$

where tau depends on the type of transmission. If the actuation installed includes a screw ball (τ_v) and an electric motor with a corresponding pair of pulleys between motor and screw shaft (τ_t):

$$\tau = \tau_v \cdot \tau_t \tag{3.40}$$

In the aforementioned example the respective angular velocity and angular acceleration at the screw (ω_v) and at the motor (ω_m) are:

$$\begin{aligned} \omega_v &= \dot{q} \cdot \tau_V \\ \dot{\omega}_v &= \ddot{q} \cdot \tau_V \\ \omega_m &= \omega_v \cdot \tau_t \\ \dot{\omega}_m &= \ddot{q} \cdot \tau_t \end{aligned} \tag{3.41}$$

As aforementioned, the dynamic model presented here has been developed as a smart tool for evaluating dynamic performance and for the sizing procedure of the actuation section. Arrived at this point, all the essential steps have been introduced.
Before concluding this section, it is useful to pay attention to the following note.

The equivalent translating mass $[M_q]$, which includes components such as motors, screws, pulleys, and the lower part of the links, was involved in the formulation of $\mathbf{F}_q$ (equation 3.10). However, it should be noted that the specific information contained within $[M_q]$ may be determined as an outcome of the sizing stages.

When selecting actuation components, it may be advantageous to define a partial resultant force along the actuation direction that considers only the force components generated by well-established moving masses

$$\mathbf{F}_r = -\mathbf{F}_e^T\,[D_m]\,[J_{inv}^{3x6}]^{-1} + \mathbf{F}_{ext}^T\,[J_{inv}^{3x6}]^{-1} + \dot{\mathbf{X}}_6^T\left([\dot{J}_{inv}^{3x6}]^T\,[M_{q,cr}] + \frac{1}{2}[\dot{M}_x]\,[J_{inv}^{3x6}]^{-1}\right) \\ +\ddot{\mathbf{X}}_6^T\left([J_{inv}^{3x6}]^T\,[M_{q,cr}] + [M_x]\,[J_{inv}^{3x6}]^{-1}\right). \tag{3.42}$$

This force represents the resultant force that needs to be generated on the slider to move it when disconnected from the actuation chain. It depends on the geometric masses of the link and joint as well as the imposed robot dynamics. The inertial contribution of the motor is not taken into account. The matrix $[M_{q,cr}]$ represents an equivalent mass at the slider without the transmission section. Now, similar to $\mathbf{F}_q$, it is possible to determine the torque required to generate $\mathbf{F}_r$

$$\mathbf{C}_r = \mathbf{F}_r^T \cdot \tau. \tag{3.43}$$

This torque and the resultant transmission ratio τ can be used to design actuation parameters optimization.

3.2 Cartesian dynamic analysis

Having fully described the dynamics of the upper stage, the PKM, it is necessary to do the same with the lower stage, the Cartesian. As with the kinematics, the description of the dynamics for this subsystem of the complete machine is simpler. So, such a description requires less space and requires less difficulty in understanding. In This description, one must take into account the reference systems introduced earlier, particularly in the section 2.4.

The dynamics of the lower system alone, as mentioned, is simple to describe. However, we must pay close attention to the effect of the upper system on the lower system. The dynamics of the upper system has a significant influence on the dynamics of the Cartesian.

Inertial Resulting Forces

One of the first step that must be considered in order to completely define the dynamic model of the overall system is to consider the inertial load generated on the Toll Center Point (TCP) of the tripod due to the cartesian motion.

Figure 3.4: Inertial Resulting Forces from cartesian motion to the CoG of the upper stage

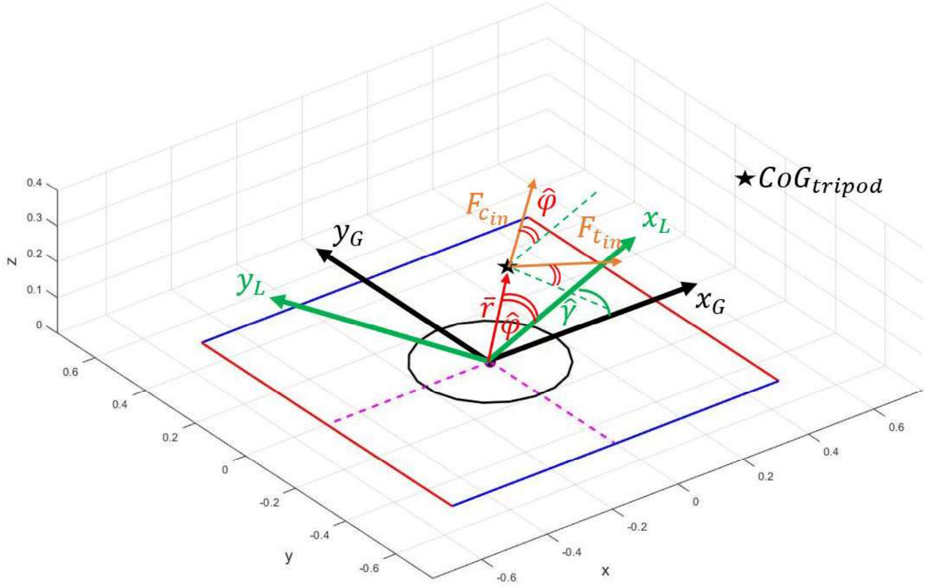
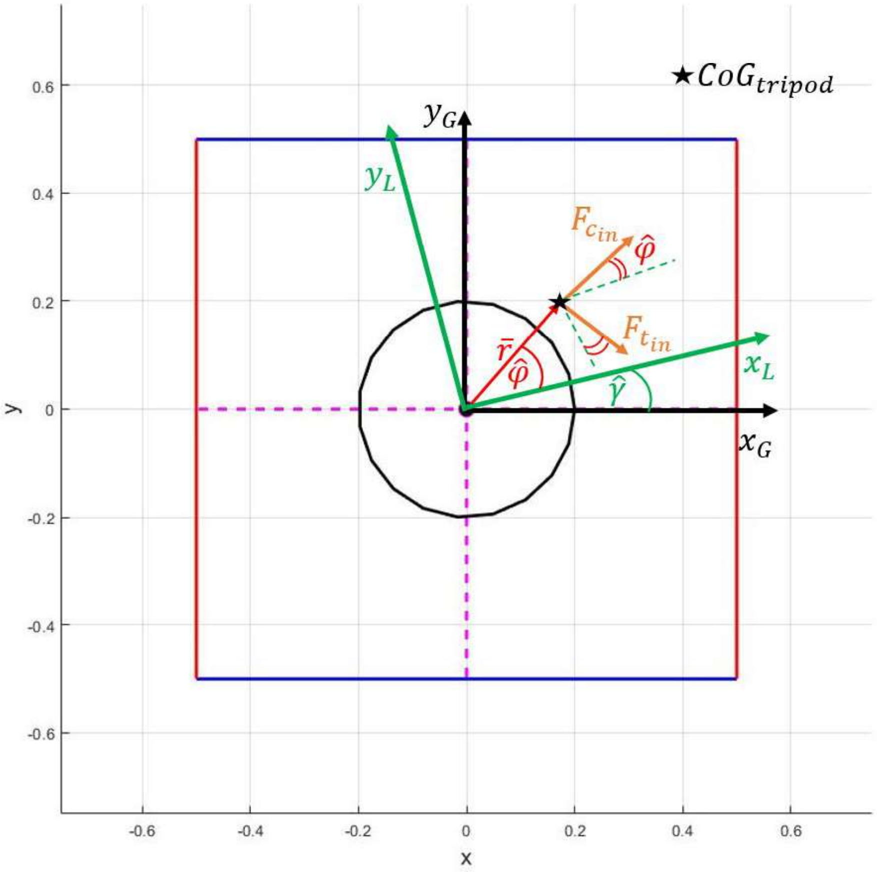


Figure 3.5: Inertial Resulting Forces from cartesian motion to the CoG of the upper stage: view from $x - y_{plane}$



Looking at figure 3.4 and figure 3.5, the vector $\mathbf{r}$ is the position of the Center of Gravity (CoG) of the upper stage in *Cartesian Local R.S.*. It is possible to define the angle ϕ using the following equation

$$\phi = \frac{a\sin(\mathbf{r}(2))}{||\mathbf{r}||_2}. \tag{3.44}$$

Considering the Cartesian degrees of freedom already determined in terms of positions $\mathbf{X}_{cart}$, velocities $\dot{\mathbf{X}}_{cart}$ and accelerations $\ddot{\mathbf{X}}_{cart}$, it is possible to proceed to a more detailed analysis.

Centripetal and tangential acceleration due to yaw rotation can be determined as follows

$$a_c = \dot{\gamma}^2 \cdot ||\mathbf{r}||_2 \tag{3.45}$$

$$a_t = \ddot{\gamma} \cdot ||\mathbf{r}||_2 \tag{3.46}$$

Thus, the corresponding force can be evaluated (see figure3.5)

$$F_c = m_{tot,us} \cdot a_c \tag{3.47}$$

$$F_t = m_{tot,us} \cdot a_t. \tag{3.48}$$

where $m_{tot,us}$ represents the total mass of the upper stage.
To transform these quantities in the *Tripod Global R.S.* It can be used the matrix

$$[R_{psi}] = \begin{bmatrix} \cos(\psi) & \sin(\psi) \\ \sin(\psi) & -\cos(\psi) \end{bmatrix} \tag{3.49}$$

then inertial forces due to yaw rotation $\mathbf{F}^{G,US}_{in,yaw}$

$$\mathbf{F}^{G,US}_{in,yaw} = [R_{psi}] \begin{bmatrix} F_c \\ F_t \end{bmatrix}. \tag{3.50}$$

Not only inertial forces due to yaw should be taken into account but also those due to relative motions in x and y. Referring to equation 2.68 and considering only the first 2 rows and columns ($[R^{2x2}_{yaw}]$), the cited contribution are

$$\mathbf{F}^{G,US}_{in,x-y} = [R^{2x2}_{yaw}]^T \begin{bmatrix} -m_{tot,us} \cdot \ddot{x}_{cart} \\ -m_{tot,us} \cdot \ddot{y}_{cart} \end{bmatrix}. \tag{3.51}$$

Thus, the overall inertial force due to cartesian motions are:

$$\mathbf{F}^{G,US}_{in} = \mathbf{F}^{G,US}_{in,yaw} + \mathbf{F}^{G,US}_{in,x-y}. \tag{3.52}$$

Lastly, the generated moments must be considered:

$$M_x = \mathbf{F}^{G,US}_{in}(2) \cdot \mathbf{b}(3) \tag{3.53}$$

$$M_y = -\mathbf{F}^{G,US}_{in}(1) \cdot \mathbf{b}(3) \tag{3.54}$$

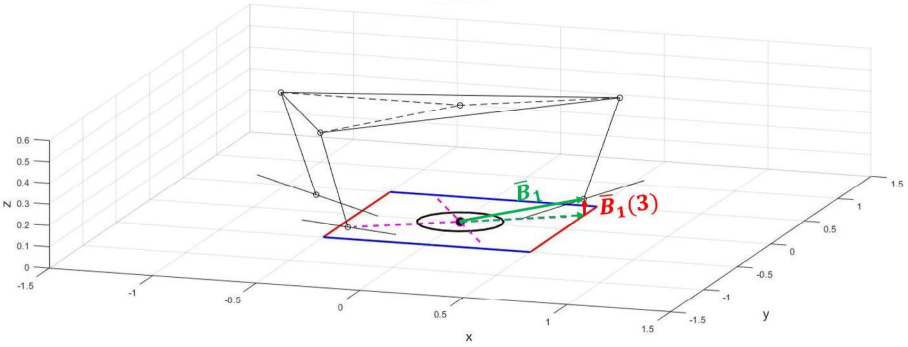
$$M_z = \mathbf{F}^{G,US}_{in}(2) \cdot \mathbf{b}(1) - \mathbf{F}^{G,US}_{in}(1) \cdot \mathbf{b}(2) \tag{3.55}$$

where $\mathbf{b}$ is the CoG position of the upper stage in the *Tripod Global R.S.*.
To incorporate these factors into the tripod dynamics, they are included as components of the vector of external forces acting on the tripod defined in the equation 3.12, which in turn is used in equation 3.11 (See sec. 3.1.1).

Exerted forces from upper systems

Another step is to define the forces exerted by the upper structure to the lower system. Obviously, this paragraph will mention the resultants defined in section 3.1.1.

Figure 3.6: Coupled cartesian and tripod system with an example of yaw rotation. The position of the slider on 1^{st} actuation system is underlined



Considering the origin position of the cartesian-tripod interface as $\mathbf{ori}_{int}$, the relative position between the latter and point $B^{G,US}$ (of each actuation line) can be defined as:

$$\mathbf{B}_{i,rel} = \mathbf{B}_i^{G,US} - \mathbf{ori}_{int}^{G,US}. \tag{3.56}$$

Recalling the resultant on the actuation direction defined in equation 3.34, and also the forces defined 3.32, it is possible to define the overall resultant forces oriented as axes "x" and "z" of each actuation plane (as it can be seen from the following equations, they are defined in the *Actuation R.S.*). In order to do that, it is useful to define a temporary vector:

$$\mathbf{F}_{qv} = \begin{bmatrix} \mathbf{F}_q^T \\ \mathbf{F}_v^T \end{bmatrix} \tag{3.57}$$

it is also necessary to define the following matrix

$$[R_\mu] = \begin{bmatrix} \cos(\mu) & -\sin(\mu) \\ \sin(\mu) & \cos(\mu) \end{bmatrix} \tag{3.58}$$

so

$$\mathbf{F}_{xz}^A = [R_\mu] \cdot \mathbf{F}_{qv} \tag{3.59}$$

where

$$\mathbf{F}_x^A = \mathbf{F}_{xz}^A(1,:)^T \tag{3.60}$$

$$\mathbf{F}_z^A = \mathbf{F}_{xz}^A(2,:)^T. \tag{3.61}$$

The following step is required for the projection of forces defined in Eq. 3.59 and the $\mathbf{F}_L$ vector force defined by equation 3.30 as shown in 3.22. Specifically, the focus is on the projection of these resultants in $x - y_{plane}$ in each actuation. A particular matrix is required

$$[R_{\pi-\rho_i}] = \begin{bmatrix} \cos(\pi-\rho_i) & -\sin(\pi-\rho_i) \\ \sin(\pi-\rho_i) & \cos(\pi-\rho_i) \end{bmatrix} \tag{3.62}$$

since, it is worth remembering, that each actuation plane has it's own ρ angle which defined the rotation with respect the $x - axis$ of the *Tripod global R.S.* (or "G, US"). Using this matrix (Eq. 3.62) is it possible to define

$$\mathbf{F}_{x_i,y_i}^{G,US} = \begin{bmatrix} F_{x_i}^{G,US} \\ F_{y_i}^{G,US} \end{bmatrix} = [R_{\pi-\rho_i}] \begin{bmatrix} \mathbf{F}_x^A(i) \\ \mathbf{F}_L^A(i) \end{bmatrix} \tag{3.63}$$

. Following the same procedure as the one summarized in equation 3.59, the two moments M_x and M_z can be defined. Here, the definition in Eq. 3.35 and Eq. 3.36 must be take into account

$$\mathbf{M}_{qv} = \begin{bmatrix} \mathbf{M}_q^T \\ \mathbf{M}_v^T \end{bmatrix} \tag{3.64}$$

where

$$\mathbf{M}_v = \begin{bmatrix} M_{v_1} \\ M_{v_2} \\ M_{v_3} \end{bmatrix} \tag{3.65}$$

$$\mathbf{M}_q = \begin{bmatrix} M_{q_1} \\ M_{q_2} \\ M_{q_3} \end{bmatrix}. \tag{3.66}$$

Thus

$$\mathbf{M}_{xz}^A = [R_\mu] \cdot \mathbf{M}_{qv} \tag{3.67}$$

where

$$\mathbf{M}_x^A = \mathbf{M}_{xz}^A(1,:)^T \tag{3.68}$$

$$\mathbf{M}_z^A = \mathbf{M}_{xz}^A(2,:)^T. \tag{3.69}$$

Once these moments are defined, the corresponding ones in *Tripod global R.S.* (or "G, US") can be evaluate:

$$M_{x_i}^{G,US} = -F_{y_i}^{G,US} \cdot \mathbf{B}_{i,rel}(3) + F_{z_i}^A \cdot \mathbf{B}_{i,rel}(2) - \mathbf{M}_{x,i}^A \cdot \cos(\pi-\rho_i) \tag{3.70}$$

$$M_{y_i}^{G,US} = F_{x_i}^{G,US} \cdot \mathbf{B}_{i,rel}(3) - F_{z_i}^A \cdot \mathbf{B}_{i,rel}(1) + \mathbf{M}_{x,i}^A \cdot \sin(\pi-\rho_i) \tag{3.71}$$

$$M_{z_i}^{G,US} = -F_{x_i}^{G,US} \cdot \mathbf{B}_{i,rel}(2) + F_{y_i}^A \cdot \mathbf{B}_{i,rel}(1) + M_{z,i}^A. \tag{3.72}$$

Lastly, the projections of $\mathbf{F}^{G,US}_{x_i,y_i}$ (Eq. 3.63), $\mathbf{M}^{G,US}_{x}$ (vector formed by the components introduced in Eq. 3.70) and $\mathbf{M}^{G,US}_{y}$ (Eq. 3.71) can be performed using the already defined $[R^{2x2}_{yaw}]$ from Eq. 2.68. Appending $\mathbf{M}^{G,US}_{x}$ and $\mathbf{M}^{G,US}_{y}$ into

$$\mathbf{M}^{G,US}_{xy} = \begin{bmatrix} \mathbf{M}^{G,US^T}_{x} \\ \mathbf{M}^{G,US^T}_{y} \end{bmatrix} \tag{3.73}$$

the required operations are

$$\mathbf{M}^{G,LS}_{xy} = [R^{2x2}_{yaw}]\mathbf{M}^{G,US}_{xy} \tag{3.74}$$

$$\mathbf{F}^{G,LS}_{xy} = [R^{2x2}_{yaw}]\mathbf{F}^{G,US}_{xy}. \tag{3.75}$$

Then

$$F^{G,LS}_{x,trp} = \sum_{i=1}^{3} \mathbf{F}^{G,LS}_{xy}(1,i) \tag{3.76}$$

$$F^{G,LS}_{y,trp} = \sum_{i=1}^{3} \mathbf{F}^{G,LS}_{xy}(2,i) \tag{3.77}$$

$$F^{G,LS}_{z,trp} = \sum_{i=1}^{3} F^{A}_{z_i} \tag{3.78}$$

$$M^{G,LS}_{x,trp} = \sum_{i=1}^{3} \mathbf{M}^{G,LS}_{xy}(1,i) \tag{3.79}$$

$$M^{G,LS}_{y,trp} = \sum_{i=1}^{3} \mathbf{M}^{G,LS}_{xy}(2,i) \tag{3.80}$$

$$M^{G,LS}_{z,trp} = \sum_{i=1}^{3} \mathbf{M}^{A}_{z_i}. \tag{3.81}$$

The subscript "*trp*" has been added so as to highlight the origin of these forces, that is, from the tripod.

Inverse dynamics

In the previous sections, all the necessary steps were carried out to make it possible to determine the forces (or torques) at lower stage level (cartesian robot). So what will be introduced here is precisely the evaluation of the actuation forces (or torques) for the lower system.

To definitely complete the definition of the forces, the inertial contribution of the cartesian robot and its actuation must be considered

- *X axis*

$$F^{G,LS}_{x,in} = -\ddot{x}_{cart}(M_{cart_{tot}} + J_{m,cart} \cdot \tau^2_{v,cart} \cdot \tau^2_{t,cart}) \tag{3.82}$$

$$F^{G,LS}_{x,tot} = -F^{G,LS}_{x,in} - F^{G,LS}_{x,trp} \tag{3.83}$$

- *Y axis*

$$F^{G,LS}_{y,in} = -\ddot{y}_{cart}(M_{cart_{\frac{1}{2}}} + J_{m,cart} \cdot \tau^2_{v,cart} \cdot \tau^2_{t,cart}) \tag{3.84}$$

$$F_{y,tot}^{G,LS} = -F_{y,in}^{G,LS} - F_{y,trp}^{G,LS} \tag{3.85}$$

- *Yaw axis*

$$T_{yaw,in}^{G,LS} = -\ddot{\gamma}(I_{z,trp} + J_{m,yaw}) \tag{3.86}$$

$$T_{yaw,tot}^{G,LS} = -T_{yaw,in}^{G,LS} - M_{z,trp}^{G,LS} \tag{3.87}$$

As with the tripod in 3.42, a force acting on the implementation can be defined for the lower stage that considers only the fully defined components. This will then be involved in the calculation of the load factor 6.11 for the Cartesian axes. Precisely, the forces defined with the equations 3.83-3.85- 3.87 without the motor-related components can be considered.

3.3 Performance definition tools

Given a complete description of the platforms considered, both from a kinematic and dynamic point of view, it is useful and necessary to introduce some fundamental tools for characterizing the performance of a dynamic simulator. For complex systems such as those under consideration in this study, defining performance is a complicated matter. This difficulty is greatly increased in machines with parallel kinematics. These machines are built to replicate motion that is not defined a priori in a space defined by the degrees of freedom of interest. So, performance for these structures must be evaluated precisely by the ability to generate motion, velocity, and acceleration for each DoF. It should be remembered, however, that because of the way the kinematics of these robots are defined, the available travel, or the ability to generate velocity or acceleration), on a specific DoF depends on the specific request made on the remaining DoFs. So, the definition of performance turns out to be a fairly complex procedure. Thus, the kinematic and dynamic description of the considered structure should be accompanied by tools that are able to define its performance. These tools become essential in various steps of the design process: from kinematic optimization to comparison between different existing solutions. So, a detailed understanding of these tools is essential, and that is the purpose of this section.
Specifically, this section introduces the following toolss: the workspace and the logarithmic graph, the last one a performance evaluation tool on the individual DoF.

3.3.1 Workspace

This section shows the workspace of the considered platform.
The workspace of a manipulator is the volume defined by the maximum reach of the

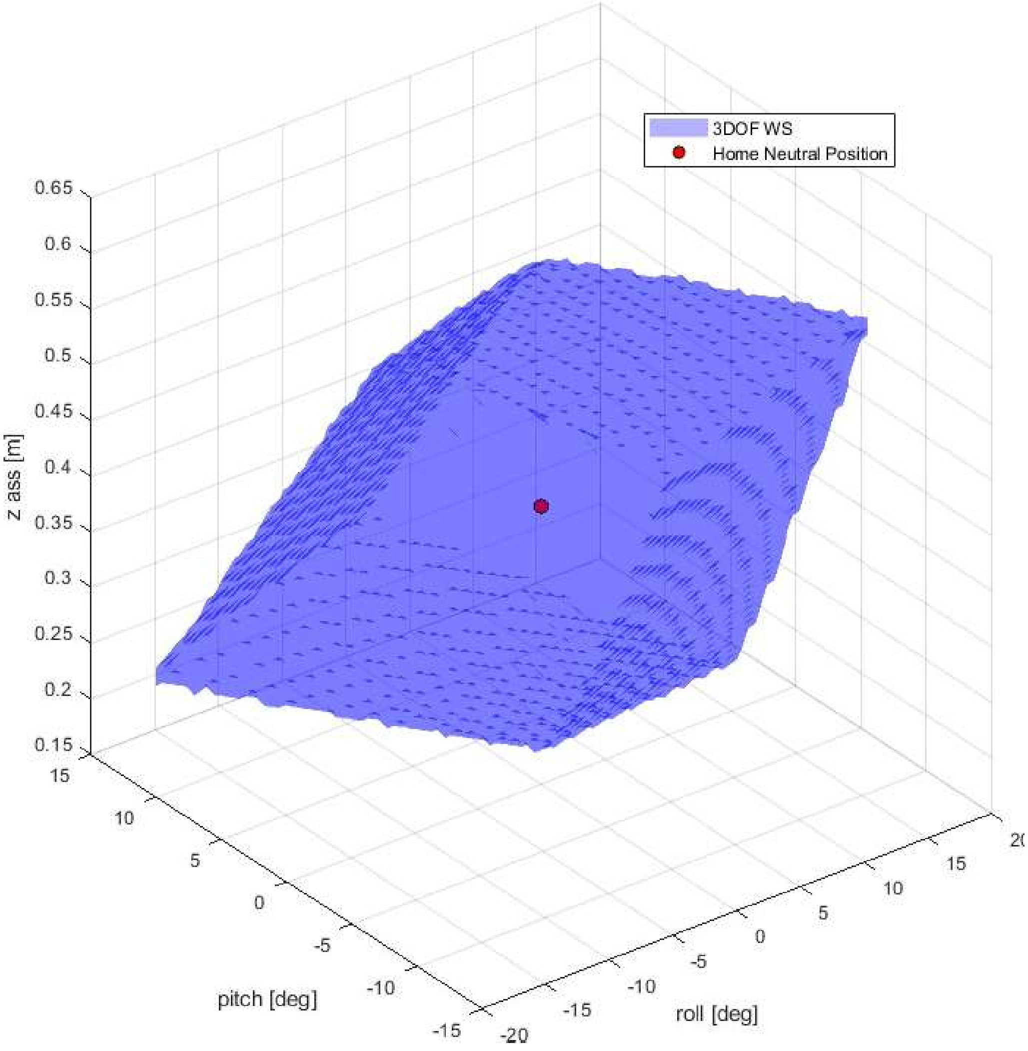
manipulator, as a result of the physical limits on the passive and actuated joints. This tool is used for defining the static performance of a robot. This tool is used for defining the static performance of a robot. One of its major utilities is to allow the definition of defined workspaces based on the movement chosen on the individual DoF. In other words, defining the ability to define the limits for some DoF based on the chosen motion on a single DoF , in this case considered of primary importance.

In the general case, the workspace is six dimensional (three translation x, y and z, and three rotations roll, pitch, yaw) and, for that reason, it is difficult to visualize. Since the robot analyzed is a combination of two devices with 3 DoFs, the workspace is defined as a combination of two three-dimensional spaces, specifically in the three dimensions: [Roll, Pitch, Z] and [X,Y, Yaw]. In order to evaluate the workspace of the analyzed mechanism, a numerical procedure is implemented. In this approach, the mechanism is tested on an equally spaced grid considering the wide ranges of each DoF. In each hypothetical pose, the assessment of all the geometric constraints is performed. If a certain pose fulfills all the requirements, it will be part of the workspace volume. The Workspace is an important tool to compare different geometric configurations since it allows to understand the behavior of the robot and its ability to bear conflict, exploit available joint range and combine DoFs. The workspace depends on the complete parametrization made. In order to better analyze the workspace, it can be divided into subspaces depending on the requirements of the specific application. For example in a PKM, like the upper stage here considered, increasing (or reducing) the heave with respect to the home pose, the capacity of the mechanism to change its roll and pitch orientation changes: in general the ranges of roll and pitch decrease or become asymmetric. It can be useful to consider a subspace of the workspace in which a certain range of heave corresponds to limited ranges of roll and pitch so that no conflicts or joint saturation can be reached.

RPZ PKM

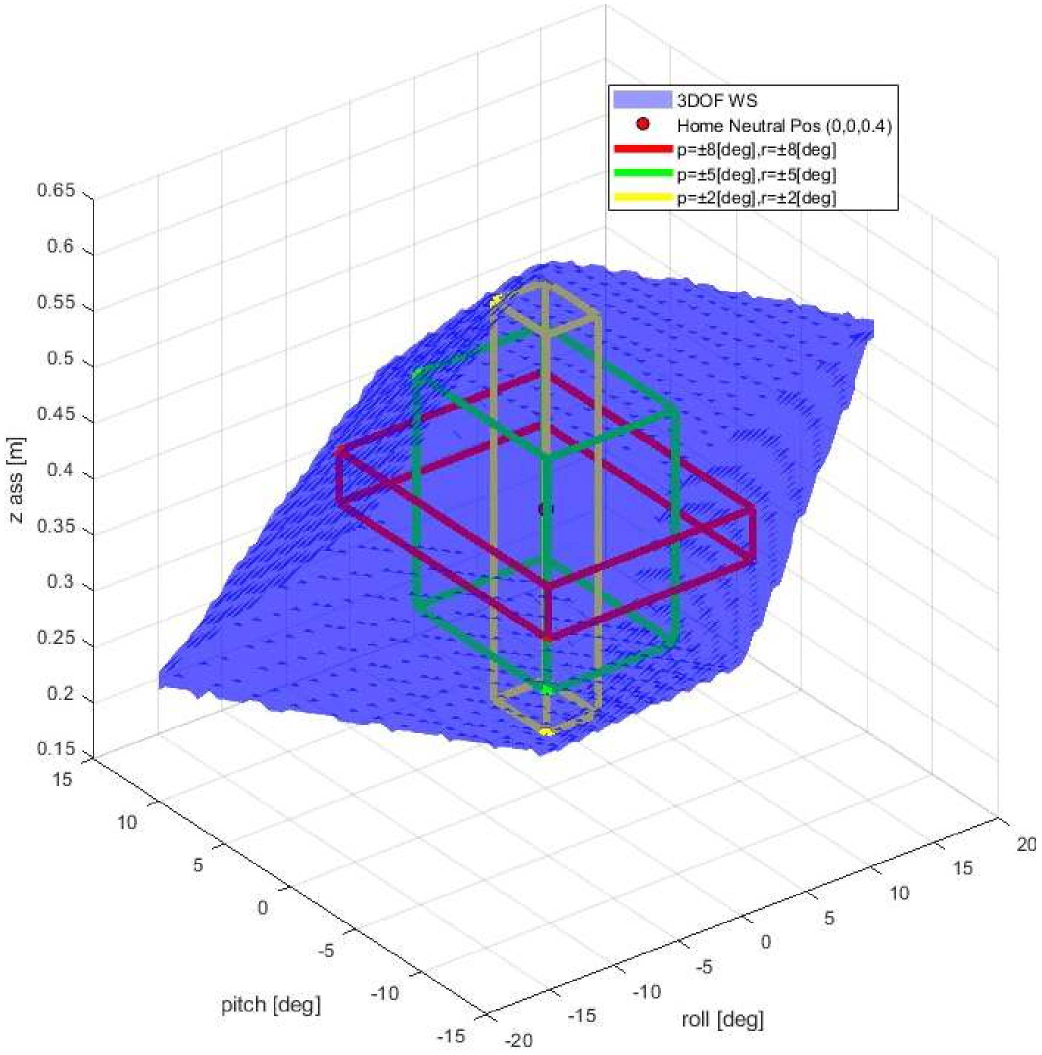
Here the workspace of the upper stage is presented. The absolute maximum workspace of the considered configuration is shown in figure 3.7. 3.7

Figure 3.7: *Roll-Pitch-Z* maximum workspace for the considered platform



Moreover, in the figure are shown some specific volumes of the workspace as described by the table. Precisely, the boxes in figure 3.8 show the maximum admissible displacement on z given a certain range of Roll and Pitch. If it's necessary to guarantee a range of roll and pitch equal to ±8 $[deg]$ with any combination of these two inside the range, the red subspace should be considered which corresponds to a range of heave of ±23 $[mm]$. Taking advantage of the table 3.1 and image 3.8, it can be seen that as the demand for roll and pitch rotations decreases, the allowable movement on z increases.

Figure 3.8: *Roll-Pitch-Z* maximum workspace for the considered platform with maximum z displacement for given roll and pitch rotation.

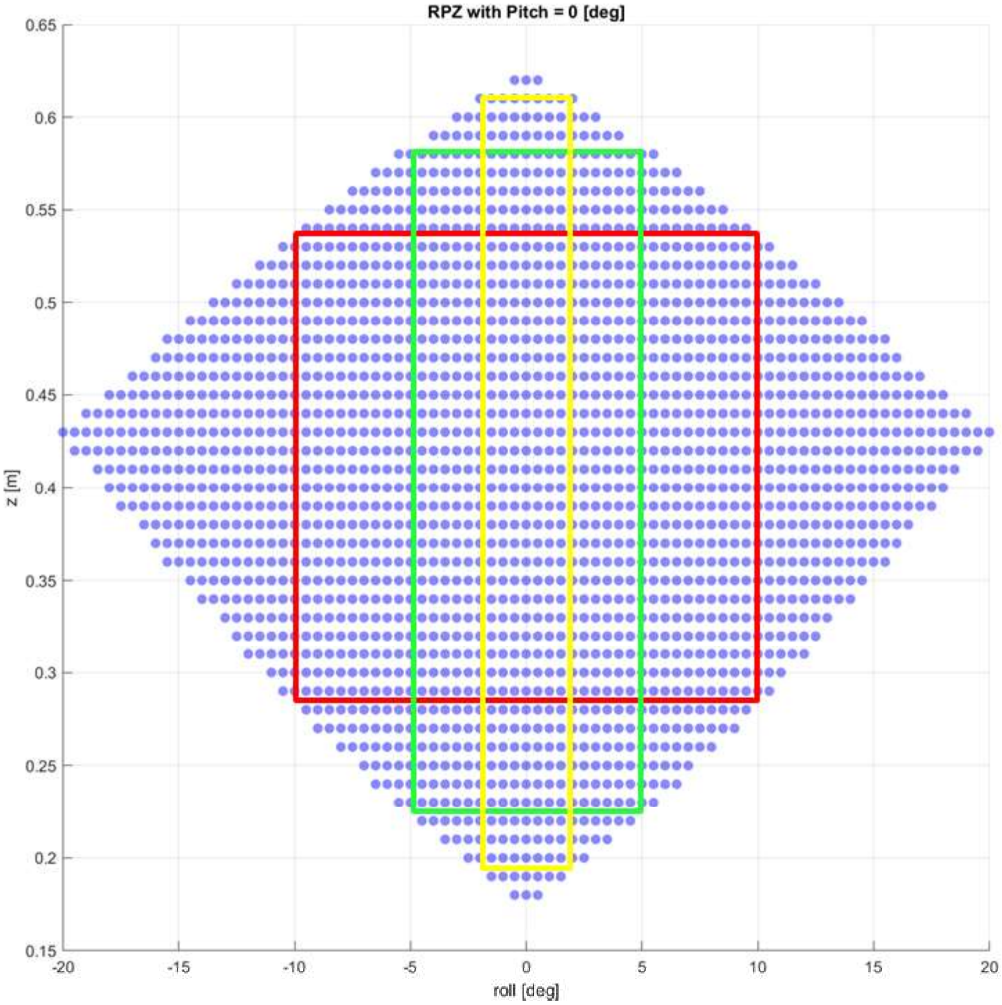


	Z [mm]	Roll [deg]	Pitch [deg]
WS1	±23 (abs mid 403)	±8	±8
WS2	±104 (abs mid 401)	±5	±5
WS3	±178 (abs mid 402)	±2	±2
MAX	±220 (abs mid 400)		

Table 3.1: Maximum Z displacement function of Roll-Pitch ranges

It is also possible to keep one of the degrees of freedom locked and see what effect this has on the behavior of the machine.

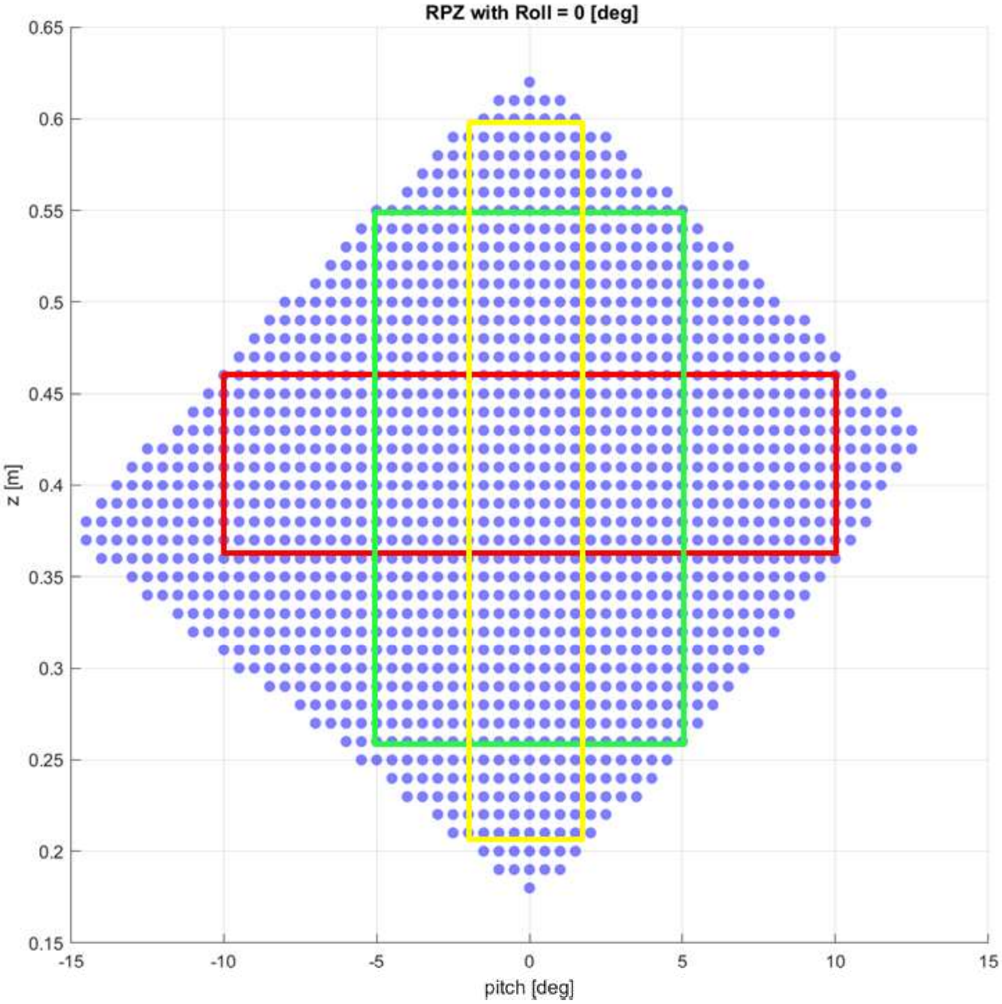
Figure 3.9: *Roll-Z* maximum workspace with $Pitch = 0$ $[deg]$.



prova	**Z [mm]**	**Roll [deg]**	**Pitch [deg]**
RZ1	±125 (abs mid 410)	±10	0
RZ2	±179 (abs mid 404)	±5	0
RZ3	±207 (abs mid 403)	±2	0
MAX	±220 (abs mid 400)		

Table 3.2: Maximum Z displacement function of Roll ranges with Pitch = 0 [deg]

Figure 3.10: *Pitch*-*Z* maximum workspace with $Roll = 0\ [deg]$.



	Z [mm]	Roll [deg]	Pitch [deg]
PZ1	±50 (abs mid 410)	0	±10
PZ2	±145 (abs mid 404)	0	±5
PZ3	±190 (abs mid 403)	0	±2
MAX	±220 (abs mid 400)		

Table 3.3: Maximum Z displacement function of Pitch ranges with Roll = 0 [deg]

From figure 3.9 It visible the symmetry of the workspace with respect to the roll, which cannot be said with respect to pitch (looking at figure 3.10). This was to be expected from the symmetry of the structure of the upper stage mechanism. The two posterior kinematic chains are symmetrical with respect to the x-axis of the global reference system. Since the roll rotation is defined about the x-axis of the topdisk (which coincides with the global reference system). Instead, pitch rotations are defined around the y-axis. The structure with respect to the XZ plane is not symmetrical: on one side there is only a single anterior kinematic chain and on the other side there are two posterior kinematic chains. Furthermore, it is possible to see from figure 3.3 how negative pitch rotations are facilitated by a smaller z than that of home ($z_{home} = 0.4$) instead, the positive pitch rotations are facilitated by a higher z than that of home. Moreover, using table 3.2 and 3.3 it is possible to notice how by keeping one of the two rotational degrees locked, the permissible displacements in the z-axis (table) increase from what both rotational degrees are considered.

Cartesian: x, y, yaw

The Cartesian, lower stage of the overall machine, is characterized by the absence of mutual dependence between the degrees of freedom. Since each line of actuation is responsible for only one DoF here is that one DoF is not constrained by the remaining ones. In this what has been represented with 360 of yaw since, the way the kinematics of the robot is studied, there are no limitations from this point of view.

Figure 3.11: Maximum cartesian workspace

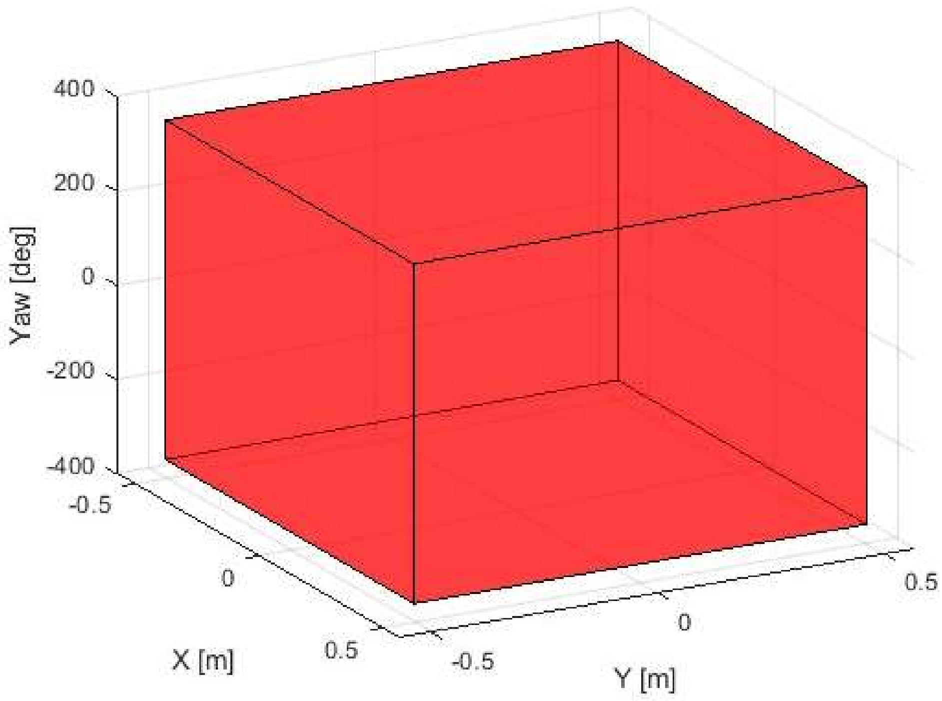


Table 3.4: Cartesian robot's maximum displacement

	X [mm]	**Y [mm]**	**Yaw [deg]**
MAX	±500	±500	±360

3.3.2 Single DOf Performances

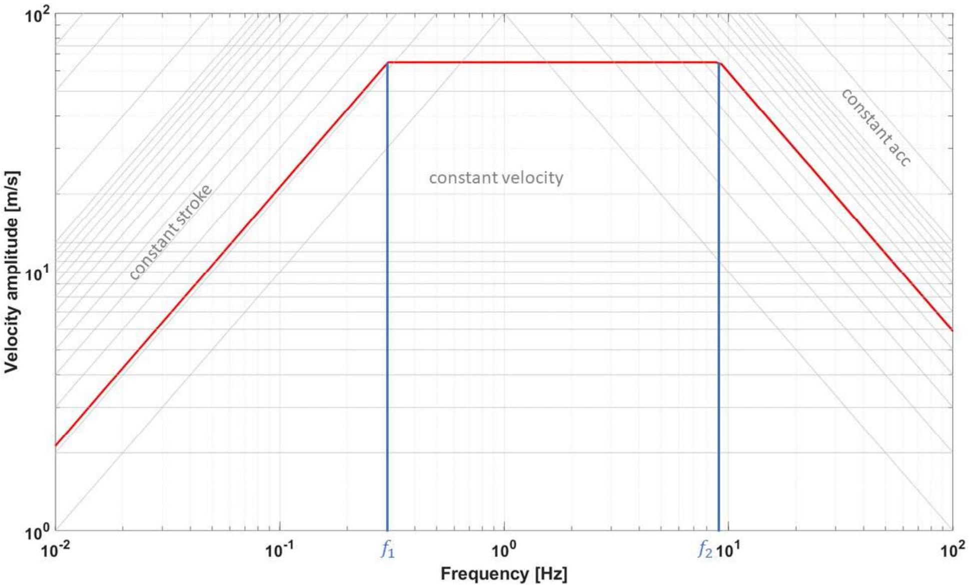
Having defined the kinematic performance of both systems, to define the dynamic performance it is necessary to introduce an additional tool to simplify the problem. The problem is that even to describe the performances of the robot when exploiting one single DoF is not trivial, because of the many related parameters involved. The complexity, when it comes to dynamic performance, increases dramatically. Defining dynamic limits for a multi-degree-of-freedom machine as mentioned in the brvee introduction at the top of this section requires the use of special tools being the ability to perform a movement with a given speed and acceleration dependent on what is required at that time on each individual DoF. The logartmic graphs introduced here allow the problem to be simplified.

Translated with www.DeepL.com/Translator (free version)

Logarithmic graphs

Logarithmic graphs are a valuable tool for characterizing the dynamic performance of a robot. They provide a concise representation of multiple aspects of the robot's behavior in a single graph. To generate these graphs, a sinusoidal motion is applied to the TCP along a single degree of freedom, starting from the home position. The amplitude of the motion is incrementally increased at a constant frequency while monitoring for any violations of the constraints described in section 2.3.9 . The highest admissible amplitude at each frequency is recorded, and the process is repeated for the entire range of frequencies of interest. The resulting limit curve can be visualized using logarithmic scales, with frequency on the abscissa and velocity on the ordinate axis. By exploiting the sinusoidal motion laws and their relationship, the graph exhibits distinct patterns. Lines with positive slopes represent collections of sinusoids at various frequencies with a constant position amplitude (stroke). Instead, lines with negative slopes represent sinusoids with the same acceleration amplitude. The limit curve depicted on the graph represents the locus of points that indicate the maximum attainable sinusoidal velocity amplitude for a specific frequency. In this way, the graph provides a comprehensive visualization of the constraints and their impact on the dynamic performance of the system.

Figure 3.12: Example of logarithmic graph



Looking at figure 3.12, it is possible to define the define three zones:

- $f \leq f_1$: Within this region, the movement is confined to a constant stroke line, indicating that the available space is the primary limiting factor. The points along this line represent kinematic constraints that restrict the motion within a specific range. These constraints are imposed by the physical dimensions and design of the system, determining the maximum achievable stroke for a given set of parameters
- $f_1 < f < f_2$: Within this region, the motion follows a constant velocity line, indicating that either the maximum motor speed or maximum screw speed has been reached. The available space is no longer the limiting factor in this zone, as the focus shifts to the dynamic limits imposed by increased frequency. The higher frequency of the sinusoidal motion puts additional stress on the system, potentially pushing it towards its dynamic limitations.
- $f \geq f_2$: In this region, the resulting curve is determined by the acceleration limits. These limits can be associated with the maximum acceleration of the motor or screw, the maximum torque of the motor or screw, or the continuous operating conditions of the motor and screw.

Typically, these graphs are obtained by characterizing an existing robot through measurements using accelerometers and other sensors. These measurements capture the behavior of the machine under different imposed motions. However, in the present case, these graphs are not based on experimental data but rather derived from a theoretical dynamic analysis of the system. They need to be used in conjunction with the motor-reducer dynamic tool to accurately size the system. By combining the theoretical analysis with the dynamic tool, it becomes possible to determine the appropriate specifications for the motor and reducer that will ensure optimal system performance. Specifically, they are introduced into the statistical process used for sizing, trying to go and limit some situations that are not feasible as iy will be explained in detail in chapter 6.

3.3.3 Kinematic crosstalk workspace

[...]

Chapter 4

Test case: complete system

The previous chapters have covered various steps in the study and design of complex machines such as automotive dynamic simulation platforms. Specifically, chapter 2 and 3 described the fully parameterized kinematic model, and the dynamic model of the two structures that make up the complete system. As mentioned above, the complete system is composed of two separate stages:

- the upper system, the tripod, with its three degrees of freedom Roll α, pitch β anz Heave Z.
- the lower system, the Cartesian, for the reproduction of surge x, sway y, and yaw.

In the following chapters, other key design points will be afforded. In order to have a clearer view of what will be described from here on, it is useful to introduce in more detail the stucture on which one is focused.

Having provided these explanations, it is then useful to turn to the implementation solutions adopted.

First, it is useful to understand broadly how the final structure was defined, and this can be done through the figure 4.1. Also represented in this figure are the peripherals that are not properly part of the design carried out on this machine. These are products readily available on the market. Thus, involving these products in the representation provides a complete view of the simulator.

Figure 4.1: Complete system

So, with this chapter, we aim to provide a clearer view of the structure considered. In particular, more attention will be paid to the most important and particular components of the two stages.

4.1 Upper stage: Tripod

It is now known that the upper stage structure is a parallel kinematics robot. In general, at the design stage, more so for structures of this type, it is useful to have the option of several implementation solutions available. To this end, it is essential to have developed a fully parameterized kinematic model. This greatly reduces the time required to generate the different solutions for comparison. Thus, in the design phase, it is important to have the opportunity to analyze different realization solutions for the considered machine. Before arriving at the current configuration, several implementation possibilities were investigated.

Figure 4.2: Configuration A

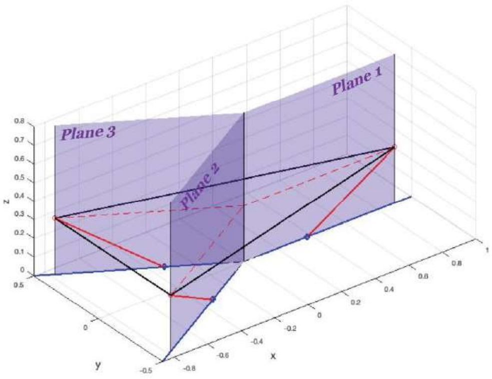


Figure 4.3: Configuration B

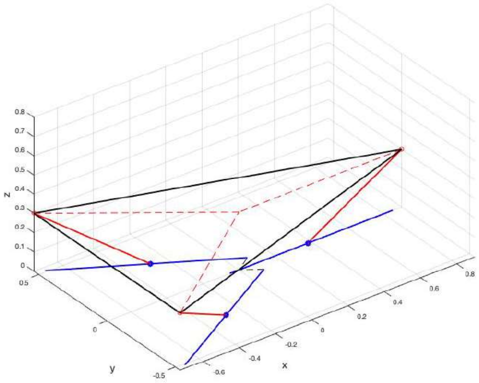


Figure 4.4: Configuration C

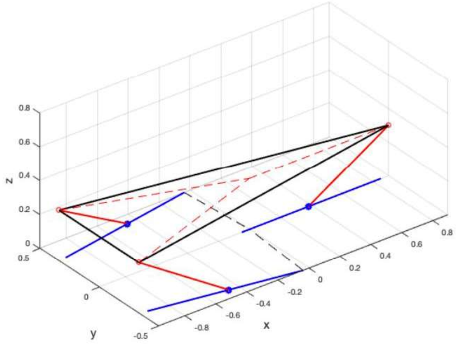
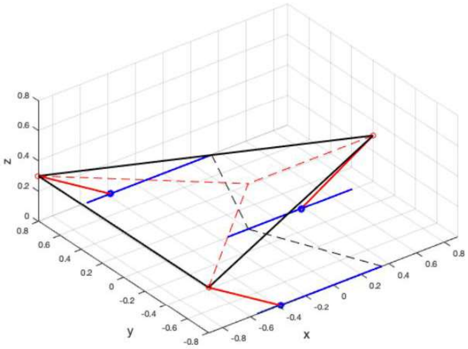


Figure 4.5: Configuration D

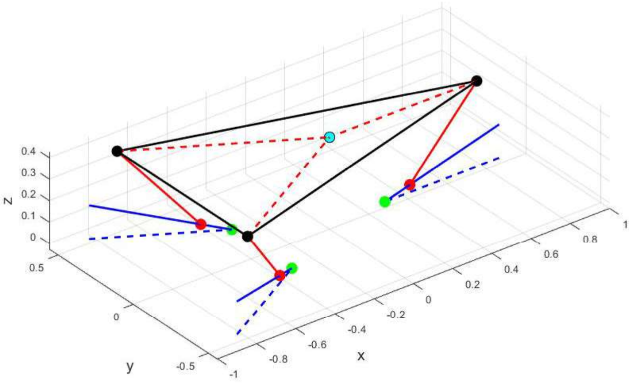


All these configurations are characterized by three actuation planes (the planes where the links are constrained to lay). In the initial configuration A - Fig. 4.2 was considered as a starting point for the study of the other configurations. This setup might encounter an issue with potential interference among distinct linear guides. Altering the geometry in this configuration can prove challenging due to limited freedom in repositioning the slider guides to avoid interference at the origin O. Consequently, the slider's attainable displacement might not ensure a sufficiently extensive workspace.

The configuarion B (Fig. 4.3) is obtained from the previous one by offsetting along the y direction (Actuation R.S.) the second and third actuation planes. The introduction of these offsets solves the problems presented in the previous configuration, introducing the possibility of moving the slider guide also in the negative x direction of the actuation plane allowing an higher stroke for the slider but also a more compact system.

The configuration C is obtained by offsetting the second and third actuation planes starting from the initial configuration. This time the offsets have opposite signs with respect to the previous configuration. In this way the slider guides of the second and third actuation planes diverge with respect to the x-axis (instead of converging as in the previous configurations), as can be seen from Figure 4.4. This configuration could further increase the possibility of changing the position of the slider guides and could generate a more compact system. Lastly, the configuration D 4.5 is obtained by rotating the actuation planes by 180. This is a particular case of the C configuration. The 180-degree rotation of the implementation planes 2-3 implies that the planes are parallel and that the *kinematic crosstalk* in x tends to infinity. So it is not possible to obtain an analytical solution and therefore for this reason this configuration is discarded. In addition to worskspace considerations, the composition of the complete system upper system + lower system should be kept in mind. So of considerable importance is the interface between the two subsystems and the exchange of forces between them. Regarding this, it was noted that configurations B and C, based on the arrangement of actuations 2-3, generated higher loads that then discharged between upper and lower system would compromise the Yaw dynamics. Semlicemente because the forces generated in actuation planes 2 and 3, due to the arrangement of the kinematic chains, acquire an arm relative to the yaw rotation axis. This inconvenience is solved by exploiting configuration A, where the planes intersect in proximity of the yaw rotation axis. To reduce the disadvantages of configuration A in terms of workspace, it was chosen to introduce tilting of the linear guides. So the final solution is as depicted in the stylized figure

Figure 4.6: Final configuration: linear guide rotated around y-axis of the actuation R.S: by an angle of $\mu = 15[deg]$



In order to ensure the commercial viability of the real robot's production, several considerations come into play. The priority is weight reduction, while still meeting the necessary performance criteria. To achieve this, aluminum (specifically ERGAL 7075) was chosen as the primary material, with exceptions made for certain structural components. The second priority revolves around sourcing readily available components from the market, thus minimizing design efforts. Only when it is impossible to find components meeting the required performance standards are new parts designed from the ground up. An early example of this approach concerns the spherical joint realization of the upper stage link. Ball joints are characterized by limited ranges of motion and large clearence. So, a special solution was chosen.

Figure 4.7: View of the universal joint of the upper stage's link

The construction is quite simple. In Figure 4.7, it is possible to see a grey central frame made of aluminum. It provides the first rotational degree of freedom (DOF) using a steel pivot, supported by two roller bearings, and held in place by washers and bolts on both sides. This frame is supported on each side by a red-colored steel pivot, which is also supported by two roller bearings and secured with washers and bolts. This setup provides the second rotational DOF required for the joint. Everything is enclosed within the frame created by the two green aluminum parts and three dark yellow ones. When transitioning from a spherical joint to a universal one, one DOF is lost. To regain it, we allow the two ends of the link to rotate freely along its axis. This degree of freedom is freed through the use of two roller bearings.

Figure 4.8: Upper stage link

The next significant construction solution involves the revolute joints between the base and links. Rather than opting for the common electric cylinders, a custom solution was chosen. This custom solution consists of two recirculating ball bearing guides that are actuated by a ball screw, see figure 4.9. As mentioned earlier some construction choices were also made based on market availability. As for the ball screw, a similar reasoning was made. Due to issues related to the stock of the company with which we collaborated in the development of this machine, we chose to reuse screws already in the stock. In particular, an NKS screw with 0.025 [mm] pitch is used for this stage.

Figure 4.9: Guide assembly

Another important building block is the topdisk, a load-bearing structure for each cockpit that will be installed on it. As mentioned earlier most of the structure is defined as aluminum, and this component is no different. As you can see from the figure 4.10 is not a solid structure, this is to reduce the weight as much as possible but without losing stiffness.

Figure 4.10: Topdisk

Table 4.1: Information of the top disk showed in figure 4.10

Mass	47.7	$[kg]$
I_{xx}	522887106	$[kg \cdot mm^2]$
I_{yy}	15575812	$[kg \cdot mm^2]$
I_{zz}	10678612	$[kg \cdot mm^2]$

The complete upper stage assembly, where the components presented above are contained, is shown in Figure 4.11.

Figure 4.11: Upper stage assembly: 3DoF PKM

Figure 4.12 shows the complete machine but from a side view. From this representation, it is possible to see a detail that is crucial in defining the movements to be imposed on the platform. This will be po detailed in the chapter 5 (section 5.2). So, it can be seen that the position of the seat has an offset from the TCP of the tripod. This offset will have to be included in the transformations necessary to make the signals returned to the pilot as realistic as possible.

Figure 4.12: Complete system lateral view

4.2 Lower stage: Cartesian

The lower stage is responsible 3 for the three movements x, y, and yaw. A special feature of the complete system presented here is modularity. It consists of two assemblies that are designed to coexist but can equally exist as a stand-alone machine. The complete system involves a considerable exchange of forces between upper and lower stages. However, the lower system, as it is studied, allows for a considerable range of its own degrees of freedom. A common choice for systems that include DoFs x, y, and yaw is a single PKM, specifically a 6 DoFs PKM. Typical parallel kinematics robot capable of performing all 6 degrees of freedom in space. This is characterized by limited ranges especially if the design is driven to maximize one. This is also due to the complex kinematics of these types of structures. In fact, often for very forward and state-of-the-art systems in dynamic simulation one solution to overcome these "disadvantages" is to use redundant degrees of freedom. So, one or more degrees of freedom are reproduced by multiple machines. Often, in systems such as these, a frequency-dependent separation of the motions to be replicated is performed.

In the system considered here, the Cartesian, one is able to have a fair range of motion with a small flat footprint. The latter was an important parameter in the design of the structure. A particular solution adopted in this application is the torque motor for implementing yaw rotation. This implies the possibility of unrestricted rotation around the z-axis.

Figure 4.13: Lower stage: cartesian

Specifically in figure 4.14 it is possible to see the torque motor housing used for yaw implementation. This section, in addition to being the housing for the torque motor, includes the interface of the torque motor with the Cartesian (the lower stage) shown as the green plate, and also the interface between the torque motor and the upper stage (blue circular plate). The internal components of this assembly represent an example torque motor.

Figure 4.14: Section of the section devoted to the implementation of the yaw

The uniqueness of this Cartesian robot lies precisely in this atypical component. The other two degrees of freedom implemented surge and sway, are simple mechanisms. As is clearly visible from Figure 4.13, both actuated degrees of freedom are realized with a double linear guide and attached recirculating screw with nut. In addition, each guide is implemented. In this simple way, the torque required from the motor is halved. In assembly, this requires considerable precision; the guides used to actuate the same DoF must be carefully aligned. Misalignment of any kind, even by a few millimeters, would result in staggered errors and a failure to implement that axis accurately.

Chapter 5

Automotive dynamic simulation

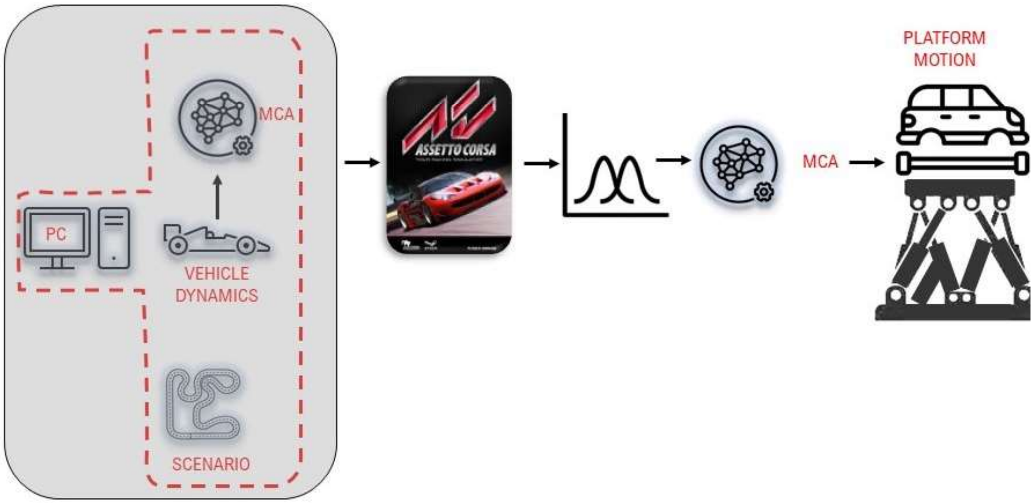
Driving simulation in the first place, as from the name itself, must be simulated. Second, the signals produced by such simulation are used to provide stimuli to the driver in order to generate as realistic a reproduction as possible of what has been simulated. Thus fundamental to the realization of a realistic driving simulation is the environment in which the simulation is produced and the ability of this software to generate accurate related signals. The chosen third-party software dedicated to these tasks is Assetto Corsa. The management of data coming from Assetto Corsa is one of the cornerstones of the work presented in this thesis. As aforementioned in section 1.4, this video game is not only involved in the process as the source of motion cueing algorithm and so as the source of the final platform motion. It is involved also in the statistical method used for the sizing procedure. Thus step by step all the ingredients necessary for a complete understanding of the procedures used are introduced. By the end of this chapter, it will be very clear why.

Precisely, Assetto Corsa (AC) is a commercial video game for sim racing. It is characterized by a realistic driving experience and extensive customization of the experience. It is currently considered one of the best video games in its field. It should not be underestimated as a video game.

5.1 Source signals' origin

The first use of video game, in creating the simulation to be played, is precisely to produce the dynamic behavior of the vehicle thus generating the appropriate signals then transformed into motion input for the platform under consideration, as shown in figure 5.1.

Figure 5.1: Use of the commercial video game Assetto Corsa in the process previously shown in the figure



5.1.1 Shared memory

Assetto Corsa is well known for the ease of experience customization also through a shared memory library ,written in C# language, which allows to access live game data. Characteristics of the C# language are classes, these are simply defined as a data structure that combines data variables and functions into a single unit. Instead, instances of the class are known as objects.
The milestone of this shared memory library is the class "*AssettoCorsa*". and by using this class, it becomes feasible to define event listeners that can be triggered for updates. In this way everyone can access to live game data.
Precisely, from "*AC Shared Memory Documentation"*, three are the events to wait for:

- updating of static information ("*AssettoCorsa.StaticInfoUpdated*")
- updating of graphics information ("*AssettoCorsa.StaticInfoUpdated*")
- updating of physics information ("*AssettoCorsa.StaticInfoUpdated*").

Through these three events, the three datasets are updated with their own timer, which can be customized.

Static information

Assetto Corsa provides some information that does not change from the initialization of the instance "listener" until this is closed. This information are referred as static and they are summarized in the following table:

Table 5.1: static AC available data

C# type	Name	Description
wchar_t	smVersion[15]	Version of the Shared Memory structure
wchar_t	acVersion[15]	Version of Assetto Corsa
int	numberOfSessions	Number of sessions in this instance
int	numCars	Max number of possible cars on track
wchar_t	carModel[33]	Name of the player's car
wchar_t	track[33]	Name of the track
wchar_t	playerName[33]	Name of the player
wchar_t	playerSurname[33]	Nickname of the player
int	sectorCount	Number of track sectors
float	trackSPlineLength	Length of the spline of the selected track
float	aidTireRate	Tire wear rate: 0 (no wear), 1 (normal), 2 (double wear) etc
...	...	...

Physics information

Each member of this packet data is updated at each physics step. Specifically, these information refers to properties of the vehicle that is being used in the game simulation.

Table 5.2: Physics AC available data

C# type	Name	Description
int	packetId	Index of the shared memory's current step
float	gas	Value of gas pedal: 0 to 1 (fully pressed)
float	brake	Value of brake pedal: 0 to 1 (fully pressed)
float	fuel	Liters of fuel in the car
int	gear	Selected gear
int	rpms	Value of rpm
float	steerAngle	Angle of steer
float	speedKmh	Speed in Km/h
float	velocity[3]	Velocity for each axis (world related) [x, y, z]
float	accG[3]	G-force for each axis (local related) [x, y, z]
float	wheelSlip[4]	Spin speed of each tyre
float	wheelAngularSpeed[4]	Angular speed of each tyre
float	accG[3]	G-force for each axis (local related) [x, y, z]
float	drs	If DRS is present and enabled
float	heading	Heading of the car on world coordinates
float	pitch	Pitch of the car on world coordinates
float	roll	Roll of the car on world coordinates
float	cgHeight	Height of Center of Gravity
int	numberOfTyresOut	How many tyres are allowed to stay out of the track to not
float	localAngularVel[3]	Angular velocity of the car [x, y, z]
int	engineBrake	Engine brake setting
float	localVelocity[3]	Vector for local velocity
...	...	...

In the just introduced table 5.2 only some of the members that shared memory makes available have been mentioned. The complete package includes 60 members only 22 are shown in table 5.2.

Graphics information

As for the previous packages, each member of this packet data is updated at each graphics step.

Table 5.3: Graphics AC available data

C# type	Name	Description
int	packetId	Index of the shared memory's current step
AC_STATUS	status	Status of the instance: AC _OFF 0 AC _REPLAY 1 AC_LIVE 2 AC_PAUSE 3
AC_SESSION_TYPE	session	Session type: AC_UNKNOWN -1 AC_PRACTICE 0 AC_QUALIFY 1 AC_RACE 2 AC_HOTLAP 3 AC_TIME_ATTACK 4 AC_DRIFT 5 AC_DRAG 6
wchar_t	currentTime[15]	Current lap time
wchar_t	lastTime[15]	Last lap time
wchar_t	split[15]	Time in sector
int	completedLaps	Number of completed laps by the player
int	iCurrentTime	Current lap time
int	currentSectorIndex	Current sector index
int	numberOfLaps	Number of laps needed to close the session
float	normalizedCarPosition	Car position on the track's spline
float	carCoordinates[3]	Car position on world coordinates [x, y, z]
...	...	...

Again, only a sample of the available members is represented here (table 5.3).

5.1.2 Custom plug-in interface

[...]

5.2 Motion Cueing Algorithm - MCA

The main objective of motion cueing systems in driving simulators is to faithfully recreate the physical motion of the simulated vehicle. By combining this with real-time image generation systems, a proper sound system, thus generating a cohesive and immersive multi-sensory environment, it allows them to perceive and control the motion of their virtual vehicle.
In real applications, the motion platform is characterized by some limitations: due to physical displacement limit and also due to the actuator choice. In this condition, it is not possible to replicate without modification the full vehicle dynamics of the considered vehicle. Thus, the *Motion Cueing Algorithm (MCA)* must be able to generate acceleration signal references that can reproduce the driving sensations experienced during a real-world experience in a virtual environment and at the same time take into account the physical limitations of the platform, and actuators so as to avoid discontinuities in motion that can damage the platform itself:

- it must produce correct references for the dynamic part: desired linear acceleration and angular velocities.
- it must consider the limited workspace available.

Obviously, in the transition from vehicle dynamic model signals to input for the platform the original signals are scaled and/or filtered.

Figure 5.2: Generic use of a motion cueing algorithm

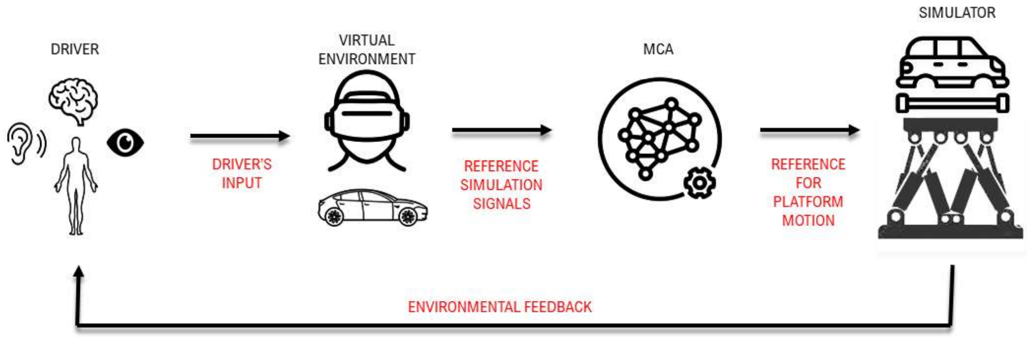


Figure 5.2 represents a generic use of the motion cueing algorithm:

1. the driver responds to the environmental feedback (stimulus) with a given action that could be a steering (sterring wheel motion), braking or accelerating (action on pedal board). Thus, these actions performed by the driver will result in inputs to the virtual simulation environment
2. the virtual simulation environment will produce reference cues for vehicle dynamics
3. the motion cueing algorithm will transform this signals, generated in the virtual environment, in platform motion signals
4. the movement produced by the platform, together with all haptic systems will generate stimuli that will be felt by the pilot who will react accordingly. Thus, starting again from. step 1

.

As aforementioned above there are different MCAs and generally they are divided in classical, optimal, adaptive, and predictive. The first two are commonly based on the succession of different filters. The adaptive and predictive are more complicated to define and also they include the human vestibular system. The vestibular system, situated in the inner ear, plays a vital role in perceiving motion within the human body. In [29], the vestibular model is deeply studied and described and some mathematical models of it are introduced and discussed. These models are those that can be involved in a complex cueing algorithm. Since a classical cueing model was developed in this work the description of the use of vestibular systems models is concluded here. The already introduced reference can be used by the reader to explore the topic further.

5.2.1 Algorithm development

[...]

5.2.2 Cueing parametrization

By referring back to the previous section, it was shown the transformation of reference signals generated within the virtual simulation, such as in a video game, into the corresponding platform reference signals through the implementation of the cueing algorithm. This process involved modifying the original reference signals to accurately reproduce the desired motion cues on the platform or robot. In order to fully characterize the cueing algorithm, several parameters need to be defined by the engineer. These parameters play a crucial role in the transformation process and include scaling factors, cutoff frequencies, and other key variables. They determine the extent of modification applied to the reference signals and help ensure that the platform accurately reproduces the intended motion cues. By carefully selecting and adjusting these parameters, engineers can fine-tune the algorithm to achieve the desired motion fidelity and enhance the immersive experience for users.

Equation ref.	**Parameter**	**Description**
[...]	k_β	scale factor for *side slip* angle velocity
	k_ν	scale factor for *curse* angle velocity
	f_ν^{HP}	high-pass cutoff frequency for *curse* angle velocity
[...]	$k_{\gamma^{c,g}}$	scale factor for yaw velocity in the definition of the *curse angle*
[...]	f_α^{HP}	high-pass cutoff frequency for roll
	f_α^{LP}	low-pass cutoff frequency for roll
	f_β^{LP}	high-pass cutoff frequency for pitch
	f_β^{LP}	low-pass cutoff frequency for pitch
[...]	$f_{\ddot{x}}^{HP}$	high-pass cutoff frequency for longitudinal acceleration
	$f_{\ddot{x}}^{LP}$	low-pass cutoff frequency for longitudinal acceleration
	$f_{\ddot{y}}^{HP}$	high-pass cutoff frequency for lateral acceleration
	$f_{\ddot{y}}^{LP}$	low-pass cutoff frequency for lateral acceleration
	$k_{\ddot{x}_{car}}$	scale factor for filtered longitudinal acceleration
	$k_{\ddot{y}_{car}}$	scale factor for filtered lateral acceleration

Table 5.4: Cueing algorithm parameters that can vary and on which platform signals depend

In table 5.4 are shown those parameters that can be customizable in order to better suit specific track conditions, driver preferences, or vehicle characteristics. These parameter adjustments allow for fine-tuning the algorithm to optimize the motion cues provided to the user.

This digression serves a crucial purpose as it establishes the connection between the signals fromAssetto Corsa and the statistical method that will be presented. Since the signals from AC are utilized in this method, it becomes essential to explain the algorithm for transforming these signals into input for the platform. Additionally, the algorithm and its associated parameters play a significant role as an ingredient within the statistical method, contributing to the overall definition of the actuation system. Therefore, a comprehensive understanding of this algorithm is fundamental in order to fully comprehend and then apply the statistical method effectively.

Chapter 6

Statistical method for sizing

The complexity of the sizing process of a machine depends directly on the complexity of the structure under consideration and its use. Furthermore, the ability to perform the sizing procedure (mechanical and of actuation) depends strictly on the ability and possibility to define the critical and necessary quantities for this process. Having a clear understanding of the tasks that the machine is expected to replicate in its operational environment is crucial. Commonly, the selection of an actuation system requires an in-depth knowledge of the desired motions, above all, related to the application field. In most of the cases these motions and then the critical conditions are known, and predefined, within the machine's workspace. However, Accurately predicting the specific requirements and nature of the task at hand can be challenging. Employing complex systems like parallel kinematics robots for dynamic simulation presents a scenario where the task may not be completely defined. In dynamic simulation, the task is generic handling and not fully defined despite the known application. The complex kinematics of PKMs make the definition of the required dynamic performances and thus the characteristic quantities for the choice of the actuation system.

In section 1.5 some methods developed over the years for defining the implementation system have been introduced through read recalls. These are characterized by knowledge from the task to be imposed and used in the sizing phase. Since no standard procedure to follow is defined for situations where the machine task, platform task, or load cycle is not known a priori, it was chosen to use a procedure already known and involve it in the sizing procedure. So, first, the method for defining the implementation will be introduced. Having provided the necessary information for understanding this method, the statistical method, the focal point of the thesis, will be introduced. The latter is the tool by which the movements to be used in the sizing process are defined.

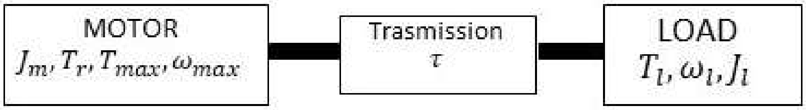
The statistical approach shares its effectiveness with the definition of an analytic dynamic model of the considered robotic platform. As previously mentioned, this process is defined from both a theoretical and practical standpoint. Thus, the presentation

of this tool is correlated with its practical use.
The statistical approach presented is based on Monte Carlo Method.

6.1 Motor-Transmission selection method

Building on what was introduced above, and on what in section 1.5, robotic applications have an actuating system composed of an electric motor and transmission. First, regardless of the application in which it will be used, a motor-transmission-load system can be summarized as in figure 6.1 where the characteristic quantities are also introduced.

Figure 6.1: Example of motor-trasmission-load system



textbfMotor section
Usually, the involved motors in robotic application are brushless motors. Their working range can be divided into two main zones: a continuous working zone and a dynamic zone. The continuous working zone is delimited by the motor's rated torque $T_{m,N}$, while the dynamic zone is delimited by the maximum motor torque $T_{m,max}$. The characteristic motor torques can be obtained from catalogs provided by motor manufacturers. Specifically, the rated torque represents the maximum torque that the motor can continuously supply without overheating. Another important quantity is the root mean square of the motor torque $T_{m,rms}$. This is usually taken into account since most of the industrial applications have cyclical loads and then a characteristic time period. This period tends to be smaller than the thermal constant of the engine [1]. Due to this, the motor cannot follow the fast heat oscillation that occurs in the cycle. Considering the heat dissipation is due to the joule effect, $T_{m,rms}$ can be referred to as the total energy dissipation over a cycle. Therefore, the working limits of a motor must be considered (look at [19]):

- limit on maximum torque

$$\max|T_m(t)| \leq T_{m,max} \tag{6.1}$$

- limit on nominal (rated) torque (thermal check)

$$T_{m,rms} \leq T_{m,nom} \tag{6.2}$$

---

[1] The motor thermal time constant is a parameter that reflects the heat capacity of a motor's windings. It indicates the extent to which a motor can store internal heat.

- limit on maximum speed

$$\max|\omega_m(t)| \leq \omega_{m,max}. \tag{6.3}$$

Trasmission section Looking at the just introduced limits, it is clear that the righ-hand-side (rhs) of Eq. 6.1, 6.2 and 6.3 referes to the motor. Instead, the left-hand-side depends on the specific use, thus on the loads' condition but also on the chosen transmission system and its transmission ratio τ. The transmission is the link between the load side and the motor side. This approach considers an ideal transmission without inertia and with no power loss ($\eta = 1$) (this assumption is frequently considered acceptable).

Load section As shown in figure 6.1 the applied load to the system results as a torque T_l, an external torque, and as an inertia $J_l\dot{\omega}_l$. The load condition depends on the speed and acceleration $(\omega_l, \dot{\omega}_l)$.

Considering together these three sub-systems (motor, transmission and load) some of the introduced inequalities can be rewritten. The maximum admissible motor speed must be compared to the maximum speed reached at the load side. The binding factor between these quantities is τ thus eq. 6.3 becomes:

$$\max|\omega_l(t)| \leq \tau\omega_{m,max}. \tag{6.4}$$

Then, this consition can be defined in term of the transmission ratio τ:

$$\tau \geq \tau_{lim} = \frac{\max|\omega_l(t)|}{\omega_{m,max}} \tag{6.5}$$

For a specific motor, this value τ_{lim} denotes the lowest possible transmission ratio that can be utilized to drive the assigned load effectively.
The power balance that governs the system can be written as:

$$T_m = \tau T_l^* + J_m\dot{\omega}_m = \tau T_l^* + J_m\frac{\dot{\omega}_l}{\tau}, \tag{6.6}$$

where T_l^* represents a generalized term for the resistant torque exerted by the robot (i.e. the load side). The aforementioned root mean square value of the torque can be obtained by

$$T_{m,rms} = \sqrt{\frac{1}{t_a}\int_0 t_a T_m} \tag{6.7}$$

where t_a represents the load's cycle . By squaring the equation 6.8 and introducing equation 6.6 the root mean square value of the torque can be defined

$$T^2_{m,rms} = \int_0^{t_a} \frac{T_m^2}{t_a} = \int_0^{t_a} \frac{1}{t_a}\left(\tau T_l^* + J_m \frac{\dot{\omega}_l}{\tau}\right)^2 \tag{6.8}$$

and then

$$T^2_{m,rms} = \tau^2 {T^*_{l,rms}}^2 + J_m^2 \frac{\dot{\omega}^2_{l,rms}}{\tau^2} + 2J_m (T_l^* \dot{\omega}_l)_{mean}. \tag{6.9}$$

Thus, the thermal check defined using the inequalities 6.2 can be rewritten as:

$$\frac{T^2_{m,nom}}{J_m} \geq \tau^2 \frac{{T^*_{l,rms}}^2}{J_m} + J_m \frac{\dot{\omega}^2_{l,rms}}{\tau^2} + 2(T_l^* \dot{\omega}_l)_{mean}. \tag{6.10}$$

At this point two parameters can be introduced: *load factor* and *accelerating factor*. Precisely the former is defined as:

$$\beta = 2[\dot{\omega}_{l,rms} T^*_{l,rms} + (\dot{\omega}_l T_l^*)_{mean}]. \tag{6.11}$$

instead the latter

$$\alpha = \frac{T^2_{m,nom}}{J_m}. \tag{6.12}$$

These two quantities do not depend on the transmission ratio τ. Moreover, *accelerating factor* α is uniquely defined by the motor and its parameters. From its definition in equation 6.12 it is clear the absence of any load callback. This can be easily defined once the motors catalog is chosen. Otherwise, the *load factor* is defined uniquely by the load conditions. Finally, the equation 6.2 first became equation 6.10 is now

$$\alpha \geq \beta + \left[T^*_{l,rms}\left(\frac{\tau}{\sqrt{J_m}}\right) - \dot{\omega}_{l,rms}\left(\frac{\sqrt{J_m}}{\tau}\right)\right]^2 \tag{6.13}$$

and from this equation, it can be seen that the component in the brackets is always positive, or null. From this, two different conditions are inferred:

- $\alpha < \beta$, which means that no transmission ratio cloud satisfy Eq. 6.13;
- $\alpha > \beta$, which means that there is a range of τ, $\tau_{min} \leq \tau \leq \tau_{max}$ that satisfy Eq. 6.13. The motor could have enough torque if the transmission is chosen properly.

Instead, a particular case occurs when τ has a value such that it drops the term in the brackets to zero ($\alpha = \beta$). This value of τ is called *optimum value*:

$$\tau = \tau_{opt} = \sqrt{\frac{J_m \dot{\omega}_{l,rms}}{T^*_{l,rms}}}. \tag{6.14}$$

In this condition the motor and load inertia assumes the same value then it could be said that "*motor and load are balanced*".
Up to this point, it is clear that the condition sought for the motor choice is $\alpha > \beta$. This condition is necessary but not sufficient. From equation 6.11 and its description it is now clear how the choice of a motor is strictly related to the coupled transmission system (tau). Specifically. the transmission system must be chosen in order to satisfy the inequality presented in Eq. 6.10 From equation 6.13 and eq. 6.5.
So in the procedure considered here, motors that have an accelerating factor such that it is greater than the load factor pass the first filtering step. This condition guarantees the existence of a range of values for tau such that the equation 6.11 is satisfied. The lower and upper bound of this range can be found by solving the equation 6.11, leading to

$$\tau_{min}, \tau_{max} = \frac{J_m}{2T^*_{l,rms}} \left[\sqrt{\alpha - \beta + 4\dot{\omega}_{l,rms} T^*_{l,rms}} \pm \sqrt{\alpha - \beta} \right]. \tag{6.15}$$

Therefore, the final condition is

$$\tau_{max} \geq \tau \geq \tau_{min} \tag{6.16}$$

but it must be remembered the condition described in equation 6.5, thus integrating it inside eq. 6.17 allows to obtain

$$\tau_{max} \geq \tau \geq \max(\tau_{lim}, \tau_{min}). \tag{6.17}$$

As shown until now, the conditions specified in Eq. 6.2 and 6.3 can be straightforwardly formulated as functions of τ, thereby establishing a range of permissible transmission ratios for each viable motor. However, due to the challenge of expressing the constraint imposed on the maximum torque of the motor as a function of tau, this criterion will be assessed after the selection of the motor and its transmission.

6.1.1 Uncertain loading condition

The procedure that is incorporated for choosing the system sizing of the implementation involves the definition of the two previously defined factors: load factor (β) and acceleration factor (α). As is well understood at this point this work focuses on those cases where the motion is not well defined as in the case of parallel kinematics robots for dynamic driving simulation. The generality of the task that will be performed by the robot does not allow to define with certainty what will be the condition of use of the machine. Even more complex is the definition of the condition that must be taken into account in the sizing phase. Wanting to take advantage of an already tried and used procedure such as the one presented above, from work by Giberti et al [30]. There it is that the will to address sizing issue for these particular machines, with unspecified motion, results in the problem of defining the load factor (β). This would eventually lead to the possibility of the choice of a suitable actuation system.

The next section of this chapter will introduce and explain in detail how the motions for defining the beta load factor are defined.

6.2 Sizing tool

The movement of the platform in the considered case is not known in advance. This makes the procedure for sizing peculiar. The field of application can be considered to simplify the overall problem. Due to the involved structure and the application field, a set of parameters that characterized the motion of the platform could be identified and then used to search for the most critical movement and thus useful for sizing. However, since there is no indication of the references to be followed in the combination of these parameters it would appear necessary to investigate all possible combinations. A deterministic procedure could be:

1. Use the robot workspace as starting point. Considering sections of the workspace.
2. Determine the admissible ranges of the parameters within the previously defined workspace sections and then discretize them in order to obtain a finite number
3. Combine them in order to generate different platform time motion histories
4. Solve the inverse kinematics and inverse dynamics to obtain the requires desired quantities.

It is clear that the investigation of the possible combination would require a huge number of calculations.
The statistical approach comes to the rescue here. The following sections explain how.

6.2.1 Platform motion: characterization and prametrization

The characterization of handling presented here is specific to the structure being analyzed. As demonstrated and described in detail earlier, the robot being considered is composed of two sections and has a total of 6 degrees of freedom. Therefore, the characterization concerns them and their definition. Precisely, it is done individually for each degree of freedom. It should be remembered that the procedure is intended to be as general as possible precisely because of the uncertainties about the final handling already described.

The characterization presented below enables the definition of a generic platform movement, which is determined by the parameters associated with each individual degree of freedom. The degrees of freedom, as mentioned earlier, are individually characterized and here they are represented as time-varying sinusoidal functions defined by four parameters.

Therefore, each sinusoid representing a DoF is described by:

$$p_j(t) = \mathbf{A}_{0,j} + \mathbf{A}_j \cdot \sin(2\pi \mathbf{f}_j + \boldsymbol{\phi}_j). \tag{6.18}$$

The four parameters that completely define each sinusoid are the ones in bold: starting pose $\mathbf{A}_{0,j}$, amplitude $\mathbf{A}_j$, frequency $\mathbf{f}_j$ and the phase $\boldsymbol{\phi}_j$. The subscript j denotes the different degrees of freedom: roll α, pitch β, yaw γ, x, y, and z. Referring to the problem addressed, i.e., sizing, one might be prompted to consider that given the choice to synthesize the various DoFs as sinusoids, then the most critical condition and thus the one required by sizing is defined by the highest frequencies ($\mathbf{f}_j$) and amplitudes ($\mathbf{A}_j$). However, this does not take into account the non-linearity of the inverse kinematics, and furthermore, it is not certain that the coupling of high frequencies and amplitude corresponds to a realistic scenario of the robot use considered. It is difficult for structures such as these to define how the implementation (in terms of torques and forces generated/required) will react following a given movement. Moreover, the staring pose of each motion plays a crucial role.

Regardless of its use, the four parameters that completely characterize a degree of freedom must be considered with some limitations:

- *Starting pose* $\mathbf{A}_{0,j}$ which may vary within the limits provided by the workspace. Its limit is the limit related to the kinematics.
- *Amplitude* $\mathbf{A}_j$. As for the starting position, it has as limit, the boundaries defined by the workspace. However, it must be considered combined with the initial position ($\mathbf{A}_{0,j}$) since the actual position reached is given by $\mathbf{A}_{0,j} + \mathbf{A}_j$.

- *Frequency* $\mathbf{f}_j$ which must be between two limit values f_{min} and f_{max}.
- *Phase* $\boldsymbol{\phi}_j$, which can vary between 0 $[deg]$ and 360 $[deg]$ (0 $[rad]$ and 2π $[rad]$).

6.2.2 Monte Carlo Method

As already introduced the statistical method that comes into play is based on the Monte Carlo Method [31]. Monte Carlo methods are an ensemble of algorithms that employ repeated random sampling in order to obtain a certain numerical results. The kernel idea behind these methods is to use randomness to solve problems that may be deterministic in principle, thus the random generation of numbers, variable or parameters that are "indipendent variables" of a problem to be solved. As reported in [32] by Coates F.W.R et al, "*Whatever sophisticated extensions of method we may subsequently discover, Monte Carlo simulation is based upon the generation of nominally uniformly distributed, pseudorandom variates and on the weighting of such variates to establish other required distribution functions*". Thus, the MCM rely on the definition of a probability density function [2] for the involved variables. The scope of the method may vary, and its use may also vary, but they generally follow this procedure:

- Determine the variables that define and characterize a problem
- Define Probability Density Functions (PDFs) for these variables
- Generate the required inputs for the involved problem. Each variable's value is obtained by samply the pre-defined PDF
- Combine and analyze the obtained results.

Applying this generality to the specific case analyzed here, the starting point for the developed method in the presented work is the following:

1. Definition of the overall number M of simulation that must be generated. The domain of the generated sampled must be large enough and reliable.
2. Definition of the Probability Density Functions (PDFs) for the input quantities. In order to reproduce the motion platform and to maintain its uncertainty of it, it is defined as the combination of the 6 DOFs, each one is presented as a sinusoid 6.18. Generally, no preliminary decision must be performed in order to follow the rule of being as generic as possible. If the overall motion, defined by the 6 sinusoids ,lays inside the admissible robot's workspace is not discarded in

---

[2] A probability density function (PDF) is a function which defines the probability of a specific random variable occurring within a statistical distribution for a continuous variable [33].

advance.
Without any information on the platform application, thus the most generic case the PDFs can be defined as follow:

- *Starting pose* $\mathbf{A}_{0,j}$: uniform distribution
- *Amplitude* $\mathbf{A}_j$: uniform distribution
- *Frequency*$\mathbf{f}_j$: uniform distribution
- *Phase* $\boldsymbol{\phi}_j$: uniform distribution.

3. Sampling of the PDF: for each of the motion histories generated, a random value is assigned to each dimension, according to the corresponding PDF and following a defined algorithm. In this case, for a 6 DOFs platform, for a definition of 6 sinusoids 24 parameters are used to completely characterize each motion platform: 6 starting poses, 6 amplitudes, 6 frequencies, and 6 phases. As already mentioned no restrictions are involved for phase and frequency definition. Instead, due to the particular shape of the platform's workspace the amplitude of each translation must be evaluated with respect to the sub-workspace resulting from the previously extracted starting position and amplitude in order not to exit from the boundaries. If this method is used for mechanical dimensioning, where the actuation has therefore already been defined, a further limitation related to the use of the logarithmic graph 3.3.2 can be introduced. On the other hand, for the choice of actuation, it is obviously not possible to use these graphs precisely because of the lack of information on the engines. However, this condition is not sufficient to ensure motion feasibility but allows to increase the ratio between reasonable histories and useless combinations produced by random generation, feasible from a geometric point of view but completely out of dynamic boundaries.

4. Computation of the Inverse Kinematic (IK) and Inverse Dynamic (ID) solutions. Using an analytical dynamic model it is possible to obtain forces, velocities, and accelerations. Thus, all those quantities are required for the sizing procedure. Specifically, in the presented work as explained in section 6.1 the motor-transmission selection occurs through the definition of the *load factor* β (Eq. 6.11) which is then compared with the *acceleration factor* α (6.12) that characterized a motor.

5. Results analysis (it depends on the objective)

Point 3 of the just mentioned list deserves an in-depth study.

PDF Sampling The first note to make about this precise step in the primary procedure is the need to define a hierarchy among the degrees of freedom of the machine being considered. This hierarchy obviously depends mainly on design parameters. Moreover, the need to define this hierarchy is mainly due to a peculiarity of parallel kinematics robots; the workspace is multidimensional, and the value assumed by one of the degrees of freedom has consequences on the limits of the remaining DoFs. Taking the parallel kinematics robot (RPZ) involved in this study as an example, it is intended below to provide an example of how step 3 is carried out. In this example, the PDFs used are considered uniform.
So, the point 3. hides the following steps:

1. Definition of the hierarchy of the available DoFs. In the presented example they will be considered in this order: pitch β, roll α and heave z.

   Pitch β sinusoid's parameter definition

2. Sampling of the pitch frequency f_β from a uniform PDF distribution:
$$f_\beta = PDF_{uni}(f_{min}, f_{max}). \tag{6.19}$$

3. Extraction the limit amplitude at f_β from the logarithmic graph $A^{LOG}_{max,\beta}(f_\beta)$. Precisely, $A^{LOG}_{max,\beta}$ represents the maximum admissibile value of pitch at frequency f_β

4. Definition of the starting pose $A_{0,\beta}$ sampling a uniform pdf distribution :
$$A_{0,\beta} = PDF_{uni}(0, A^{LOG}_{max,\beta}). \tag{6.20}$$

   In this way, the previous limit can be taken into account.

5. Extraction of the limit pitch amplitude from the TOW $A^{TOW}_{max,\beta}$, $A^{TOW}_{min,\beta}$

6. Given the starting pose, the limit amplitude within which to search A_β is the minimum between the following values:
$$\left.\begin{aligned} \beta^{TOW}_{LB} &= |A^{TOW}_{min,\beta} - A_{0,\beta}| \\ \beta^{TOW}_{UB} &= |A^{TOW}_{max,\beta} - A_{0,\beta}| \\ \beta^{LOG}_{UB} &= |A^{TOW}_{max,\beta} - A_{0,\beta}| \end{aligned}\right\} \beta_{UB} = min(\beta^{TOW}_{LB}, \beta^{TOW}_{UB}, \beta^{LOG}_{UB}). \tag{6.21}$$

7. Definition of the pitch sinusoid amplitude

$$A_{\beta} = PDF_{uni}(0, \beta_{UB}). \tag{6.22}$$

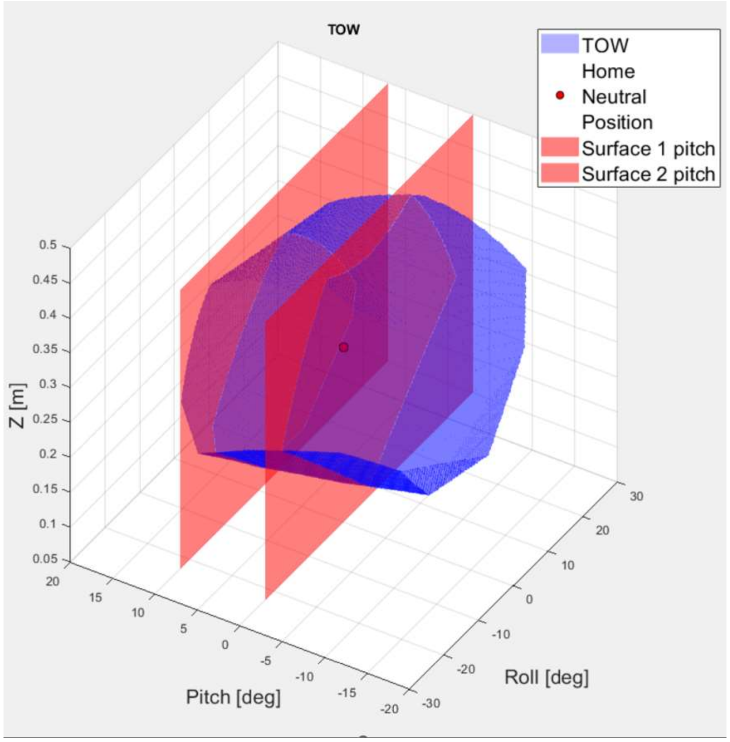
8. Definition of the phase for the sinusoid (Eq. 6.18) :

$$\phi_{\beta} = PDF_{uni}(0, 360) \tag{6.23}$$

Roll α sinusoid's parameter definition

9. Extraction from TOW of the minimum positive and negative limit for Roll. At this point must be considered the mutual connection of the degrees of freedom for a parallel kinematic machine should be considered. So, the choice of one degree of freedom limits the remaining DOFs. The selection of the pitch sine motion, as depicted in Figure 6.2, imposes constraints on the forthcoming choices regarding the yet-to-be-defined degrees of freedom. Specifically, it is possible to extract: α^{β}_{UB}, α^{β}_{LB}, z^{β}_{UB} and z^{β}_{LB}. The subscript discerns between Lower and Upper Bound (LB and UB) instead, the superscript, it this level it is β, indicates from which chosen DoF such a limit comes.

Figure 6.2: PKM TOW where the planes defining the limits of the pitch sinusoid are shown in red



10. Sampling the roll (α) frequency on a uniform PDF distribution:

$$f_\alpha = PDF_{uni}(f_{min}, f_{max}) \tag{6.24}$$

11. Extraction the limit amplitude at f_α from the logarithmic graph $A^{LOG}_{max,\alpha}(f_\alpha)$.

12. Definition of the starting pose $A_{0,\alpha}$ sampling on a uniform PDF distribution considering the limit of point 9. (α^{β}_{UB}, α^{β}_{LB}) and 11. ($A^{LOG}_{max,\alpha}(f_\alpha)$):

$$A_{0,\alpha} = PDF_{uni}(0,\ min(\alpha^{\beta}_{UB},\ \alpha^{\beta}_{LB},\ A^{LOG}_{max,\alpha})\ ) \tag{6.25}$$

13. Extraction of the limit roll amplitude from the TOW: $A^{TOW}_{max,\alpha}$, $A^{TOW}_{min,\alpha}$

14. Given the starting pose, the limit amplitude within which to search A_β is the minimum between the following values:

$$\left.\begin{aligned} \alpha^{TOW}_{LB} &= |A^{TOW}_{min,\alpha} - A_{0,\alpha}| \\ \alpha^{TOW}_{UB} &= |A^{TOW}_{max,\alpha} - A_{0,\alpha}| \\ \alpha^{new}_{UB} &= min(\ |[\alpha^{\beta}_{UB},\ \alpha^{\beta}_{LB},\ A^{LOG}_{max,\alpha}] - A_{0,\alpha}|\ ) \end{aligned}\right\} \alpha_{UB} = min(\alpha^{TOW}_{LB}, \alpha^{TOW}_{UB}, \alpha^{new}_{UB}). \tag{6.26}$$

The superscript *new* manifest the novelty of this constraint where the roll limits due to the pitch motion are introduced.

15. Definition of the roll sinusoid amplitude

$$A_\alpha = PDF_{uni}(0, \alpha_{UB}). \tag{6.27}$$

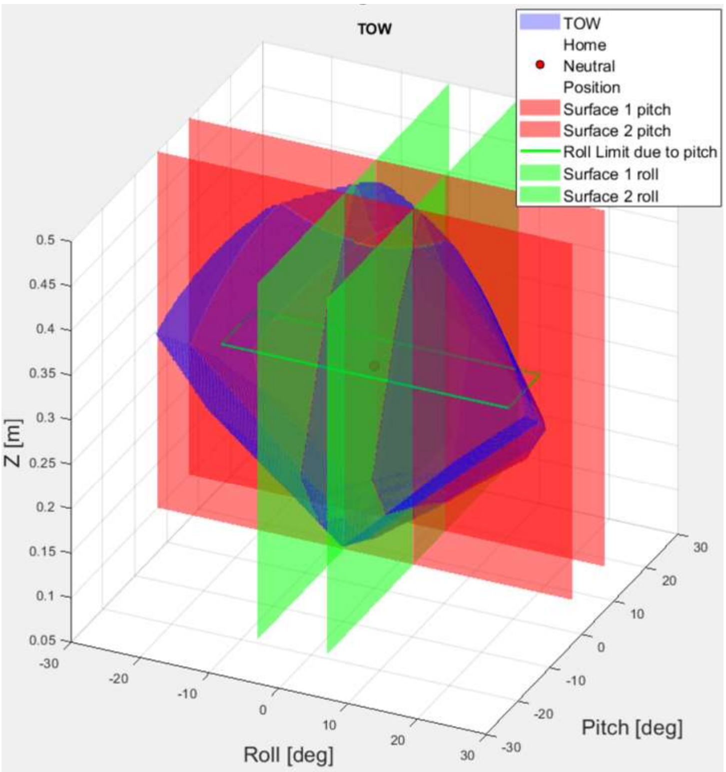
16. Definition of the phase for the roll sinusoid (Eq. 6.18) :

$$\phi_\alpha = PDF_{uni}(0, 360) \tag{6.28}$$

Heave z sinusoid's parameter definition

17. Extraction from TOW of the minimum positive and negative limit for Z. As done in the point 9., the limit on z due to the just defined roll motion needs to be considered (see Fig. 6.3): z^{α}_{UB}, z^{α}_{LB}.

Figure 6.3: PKM TOW where the planes defining the limits of the pitch and roll sinusoid are shown as red and green surfaces.



18. Sampling the z frequency on a uniform PDF distribution:

$$f_z = PDF_{uni}(f_{min}, f_{max}). \tag{6.29}$$

19. Extraction the limit amplitude at f_z from the logarithmic graph $A^{LOG}_{max,z}(f_z)$.

20. Definition of the starting pose $A_{0,z}$ sampling on a uniform PDF distribution considering the limit of point 9. and 17. (z^{α}_{UB}, z^{α}_{LB}, z^{β}_{UB}, z^{β}_{LB}) and 11. ($A^{LOG}_{max,\beta}(f_\beta)$):

$$A_{0,z} = PDF_{uni}(0,\ min(\ z^{\alpha}_{UB},\ z^{\alpha}_{LB},\ z^{\beta}_{UB},\ z^{\beta}_{LB},\ A^{LOG}_{max,z})\ ) \tag{6.30}$$

21. Extraction of the limit z amplitude from the TOW: $A^{TOW}_{max,z}$, $A^{TOW}_{min,z}$

22. Given the starting pose, the limit amplitude within which to search A_z is the

minimum between the following values:

$$\left.\begin{aligned} z_{LB}^{TOW} &= |A_{min,z}^{TOW} - A_{0,z}| \\ z_{UB}^{TOW} &= |A_{max,z}^{TOW} - A_{0,z}| \\ z_{UB}^{new} &= min(\ |[0,\ min(\ z_{UB}^{\alpha},\ z_{LB}^{\alpha},\ z_{UB}^{\beta},\ z_{LB}^{\beta},\ A_{max,z}^{LOG}] - A_{0,z}|\ ) \end{aligned}\right\} z_{UB} \tag{6.31}$$

$$z_{UB} = min(z_{LB}^{TOW}, z_{UB}^{TOW}, z_{UB}^{new})$$

23. Definition of the z sinusoid amplitude

$$A_z = PDF_{uni}(0, z_{UB}). \tag{6.32}$$

24. Definition of the phase for the z sinusoid (Eq. 6.18) :

$$\phi_s = PDF_{uni}(0, 360) \tag{6.33}$$

Figure 6.4: PKM TOW where the planes defining the limits of the pitch, roll and z sinusoid are shown as red, green and yellow surfaces.

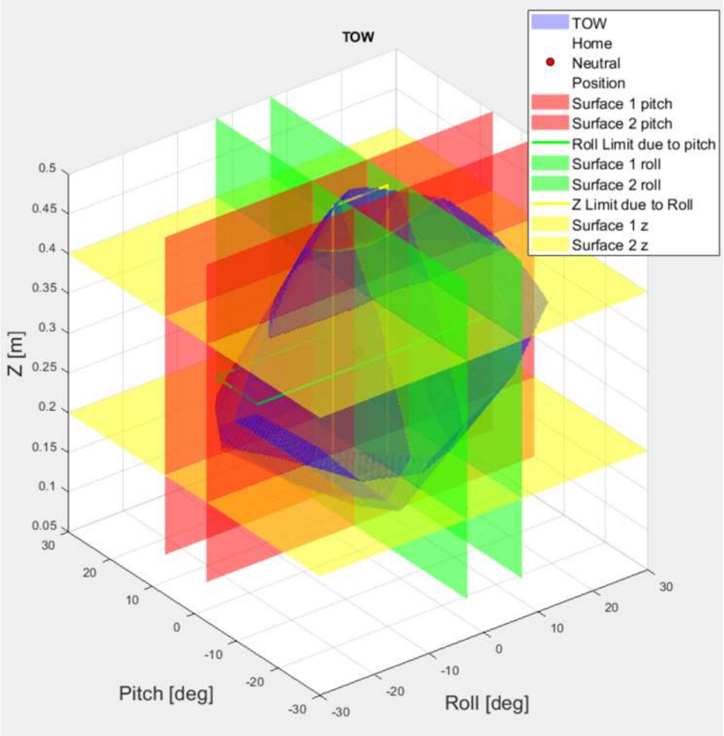


Figure 6.4 shows all the plans that go into limiting the choice of a particular movement following what has just been described. So, now it is clear how much is hidden behind only the step 3. of the procedure 6.2.2.

Load factor definition β Finally, the general procedure used for dimensioning was analyzed and introduced. Given the variety of topics that had to be introduced in order to describe the custom procedure from here on, it is useful to recall what was described in Chapter 1, section 6.1 section.
It was introduced how following the procedure defined in [30], is possible to arrive at the equation 6.13:

$$\alpha \geq \beta + \left[T^{*}_{l,rms} \left(\frac{\tau}{\sqrt{J_m}} \right) - \dot{\omega}_{l,rms} \left(\frac{\sqrt{J_m}}{\tau} \right) \right]^2$$

$$\alpha \geq \beta + \gamma(\tau)$$

$$\gamma(\tau) = \left[T^{*}_{l,rms} \left(\frac{\tau}{\sqrt{J_m}} \right) - \dot{\omega}_{l,rms} \left(\frac{\sqrt{J_m}}{\tau} \right) \right]^2 .$$

These quantities should not be confused with rotations.
Therefore, recalling that procedure the selection of the actuation system is contingent upon the *load factor*, β. To fulfill this requirement, the statistical method based on Monte Carlo simulation (MCM) described earlier is employed. This method allows for the exploration of a vast array of operating conditions, yielding valuable quantities for sizing purposes. Precisely, the *Load Factor* β of each of the generated motion histories is evaluated with the Eq. 6.11, involving the resultant forces 3.42 at the upper stage .

Thus far, It has been presented a general procedure for defining the *Load Factor*, but the specific utilization of data from *Assetto Corsa* has not been elucidated. The following section aims to delve into this aspect in detail, providing a comprehensive analysis and outlining the precise procedure that has been formulated and implemented in this study.

6.3 Procedure's customization

[...]

6.3.1 Customization of logarithmic graphs

[...]

6.3.2 PDF customization

[...]

Chapter 7

Real Case

The last section of chapter 1 introduced the sizing problem in the robotic field by referring to the increase of difficulties considering the field of application as dynamic simulation. To make this scope clear, Chapter 1 introduced dynamic simulation and then driving simulators. Chapter 2 described in detail the kinematic and dynamic model of the considered platform, because as already introduced, in addition to the theoretical description of the statistical method aimed at dimensioning, this was applied to a real process of dimensioning. Chapter 5, on the other hand, was concerned with the description of a critical component for the dynamic simulation: the definition of the signals to be replicated. This need was translated into an opportunity to define a method that can generate simulations more similar to reality. In fact, as shown in Chapter 6, data from running trim are used to modify the Monte Carlo Method. The description of this method and its customization are the main topics of Chapter 6.

At this point, all the difficulties involved have been introduced, and all the expediencies considered have been explained. So in this chapter, the concrete use that is made of the statistical method, described in the previous chapter 6, is here detailed.

Firstly, the dataset generatin will be introduced. Then, the extraction of the necessary quantities for the customized MCM method (ρ and τ described in 6) is addressed. The MCM method in then used for defining the required sizing quantities.

7.1 Dataset generation

As explained in Chapter 6, specifically in the section, data *Assetto Corsa - AC* was used for customization of a generic method (section 6.3). Precisely, they were used for the modification of the PDFs involved in step 3. of the process 6.2.2. The custom plug-in interface described in section 5.1.2 is the one used to extract the data that were then later analyzed.

Precisely the dataset analyzed consists of 131 elements generated involving: 31 different simulations (combination of track, vehicle, and driver) and 5 different sets of cueing parameters.

The total number of combinations generated does not correspond to the number of simulations per number of cueing because a cueing set was defined for only one car model. So this was used only in the case of simulations involving that particular type of car. The vehicles and cueing parameter sets deserve to be mentioned.

7.1.1 Vehicles, Cueings, and Tracks

Vehicles

To generate the dataset for analysis, various types of vehicles were included, each with its specific characteristics and properties. They are:

- *Formula RSS3 V6* (Fig. 7.1a)
- *Ferrari 488 GT3* (Fig. 7.1b)
- *Abarth 500* (Fig. 7.1c)
- *Audi A1 S1* (Fig. 7.1d)
- *Maserati Levante* (Fig. 7.1e)
- *Praga R1* (Fig. 7.1f)

(a) Formula RSS3 V6

(b) Ferrari 488 GT3

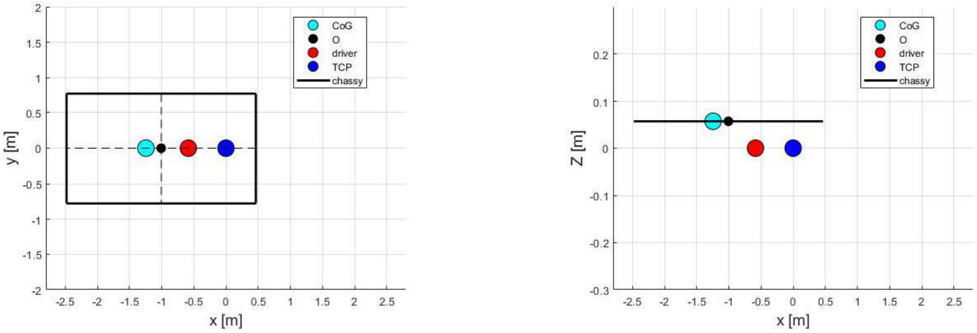
(c) Abarth 500

(d) Audi A1 S1

(e) Maserati Levante

(f) Praga R1

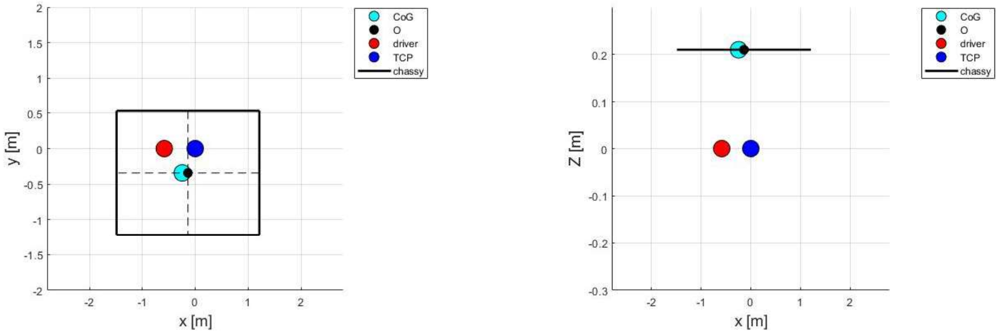
Figure 7.1



(a) XY chassy’s view

(b) XZ chassy’s view

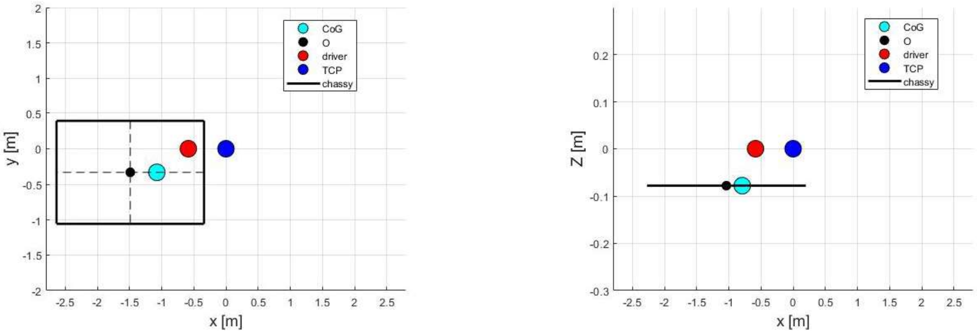
Figure 7.2: Formula RSS3 V6: chassy with representation of CoG, driver position, TCP positions



(a) XY chassy’s view

(b) XZ chassy’s view

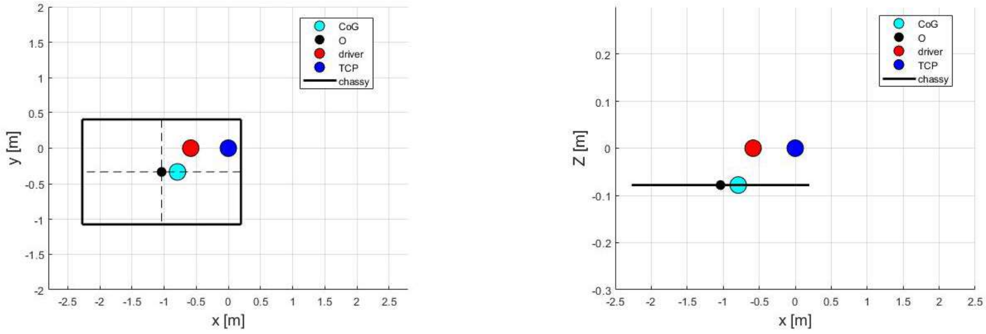
Figure 7.3: Ferrari 488 gt3: chassy with representation of CoG, driver position, TCP positions



(a) XY chassy’s view

(b) XZ chassy’s view

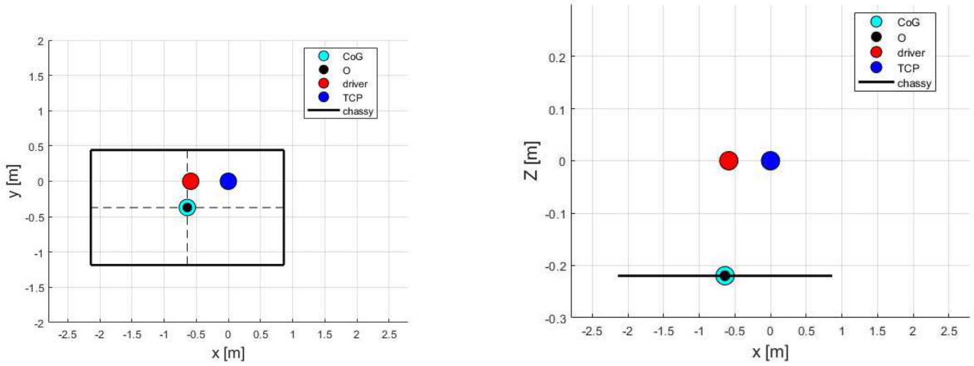
Figure 7.4: Abarth 500: chassy with representation of CoG, driver position, TCP positions



(a) XY chassy's view

(b) XZ chassy's view

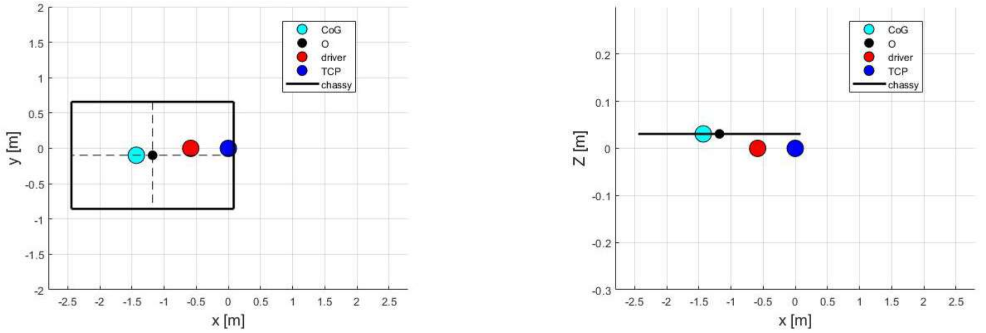
Figure 7.5: Audi a1s1: chassy with representation of CoG, driver position, TCP positions



(a) XY chassy's view

(b) XZ chassy's view

Figure 7.6: Maserati Levante: chassy with representation of CoG, driver position, TCP positions



(a) XY chassy's view

(b) XZ chassy's view

Figure 7.7: Praga r1: chassy with representation of CoG, driver position, TCP positions

Figures from 7.2 to 7.7 show the chassis of the cars used (list 7.1.1). It can be seen that they have different characteristics. Especially, those quantities (as shown in the section 5.2) useful for the transformation of signals from AC from vehicle degrees of freedom to platform degrees of freedom are represented: CoG position (light blue), driver position (red), TCP(blue) and the center of the axes of symmetry of the car ("o" in black).

Cueing parameters

As mentioned earlier, there are 31 simulations but these 31 simulations are sufficient to generate a total of 131 different conditions just through the combination of AC simulation and different sets of cueing parameters. The parameters that are varied and result in different cueing have already been introduced in the table 5.4.

Table 7.1: Cuieng parameters of the five sets involved in platform DOFs generation

Parameter	**Set 1**	**Set 2**	**Set 3**	**Set 4**	**Set 5**
k_{β}	0.7	1	0.5	0	0.7
k_{v}	0.5	0.1	0.3	0	0.5
f_{v}^{HP}	0.05	0.05	0.1	0.05	0.05
$k_{\gamma^{c,g}}$	0.7	1	1	0.7	0
f_{α}^{HP}	0	0	0	0	0
f_{α}^{LP}	10	10	10	10	10
f_{β}^{HP}	0	0	0	0	0
f_{β}^{LP}	10	10	10	10	10
$f_{\ddot{x}}^{HP}$	5	8	8	5	5
$f_{\ddot{x}}^{LP}$	10	15	15	10	10
$f_{\ddot{y}}^{HP}$	5	5	5	5	5
$f_{\ddot{y}}^{LP}$	10	10	10	10	10
$k_{\ddot{x}_{car}}$	0.5	0.6	1	0.5	0.5
$k_{\ddot{y}_{car}}$	0	0	0	0	0

Set 1 is similar to set 4 and 5. It has a scale factor on the yaw of 0.7. As is well understood this leads to having a scaling of the yaw dynamics but without having any loss of information, This value combined with that of k_{β}, k_{v}, $f_{\ddot{x}}^{LP}$ and $k_{\ddot{x}_{car}}$ can be considered as a starting point for cueing for vehicles of the formula family. Set 2 is the most aggressive set among those used. The combination of the factors used leads to an aggressive reproduction of the vehicle dynamics. Indicatively, this can be a reference in case cars are used whose dynamics are not too pushy and therefore it is not necessary to go limiting. In contrast, Set 3 represents a more balanced aversion of Set

2. Consequently, this is a starting point for cueing related to more presented machines such as GTs may be. Finally, the last two cueings go differentiated by the handling of a boundary condition related to yaw dynamics. Specifically, set 4 sets the yaw to zero by acting on the two coefficients k_β, k_v. In contrast, set 5 limits the yaw by going to act on the curse angle with the $k_\gamma^{c,g}$.

7.2 Statistical method customization

In the generic statistical method, described in 6.2.2 (List 6.2.2), the substantive changes, introduced in sec. 6.3 can be summarized in the following two: use of logarithmic graphs defined through data from AC, modification of probability density functions (phase and frequency) from which parameters are sampled that then go on to define the individual DOF time history. Thus, to apply the method introduced in chapter 6 and then customized, it is necessary to define those parameters that allow its use: the Pearson correlation coefficients ρ , the tau τ value from the cross-correlation analysis and the "dummy" logarithmic graphs.

Distribution parameters definition

In order to customize the distribution from which the parameters of the sinusoid 6.18 it is necessary to define the Pearson correlation coefficient ρ and the time lag τ. These two parameters allow you to go and edit the Uniform PDFs. Generating trends to sample from that depend on the mutual correlation between the degrees of freedom. For each combination of two DoFs these two quantities must be generated. Having a dataset of time histories of the platform DOFs (which characterize a movement) it is therefore necessary to take an average behavior of these quantities so as not to force the history generator too much. So the PDFs of the frequencies will be modified according to the intensity of the correlation (see table 7.2 where the correlation values involved in the procedure are shown). As can be seen from the table, there are strong relationships between the platform DoFs. The new distributions will be centered on the frequency value of the DoF already defined with the highest correlation with the DoF under consideration for definition. Instead, the value of τ from the analysis of AC is used as a tool to change the distribution of phase. This is because τ from the cross-correlation study indicates that the two signals under consideration have a peak correlation considering a shift in time equal to tau (of only one of the two). So, since the time histories of DOFs generated by the statistical method are sinusoids 6.18, the information contained in τ can be translated into a phase shift between the sinusoids. The distribution from which the phase will be extracted will be centered on the tau value described above, and the standard deviation instead will depend on the correlation intensity.

	β	z	x	y	γ
α	-0.1916	0.3956	-0.0597	0.3415	-0.4588
β	-	-0.8584	0.0357	-0.118	-0.2094
z	-	-	-0.2194	0.1990	-0.2699
x	-	-	-	0.1447	0.1315
y	-	-	-	-	0.7524

Table 7.2: Correlation values involved in the customized MCM procedure (α= roll, β = pitch, γ = yaw).

The DOF hierarchy for the platform in question is as follows: pitch, roll, z, x, y, yaw. So only the first one, pitch, will continue to be defined by 4 parameters (A_0, A, f and ϕ) from classical uniform distributions. The others starting with roll will then go to consider correlations with those already chosen:

- *Pitch-β* Uniform Distribution for all four parameters ($A_{0,\beta}$, A_β, f_β and ϕ_β)
- *Roll-α*

Starting pose $A_{0,\alpha}$:	Uniform distribution
Amplitude A_α:	Uniform distribution
Frequency f_α:	Custom distribution function of Pearson correlation coeff. between Pitch and Roll $\rho_{\beta,\alpha}$
Phase ϕ_β:	Custom distribution function of $\tau_{\beta,\alpha}$ (as mean value of the distribution) and of $\rho_{\beta,\alpha}$ (for the standard deviation)

- *Z*

Starting pose $A_{0,z}$:	Uniform distribution
Amplitude A_z:	Uniform distribution
Frequency f_z:	Custom distribution function of the most intense Pearson correlation coeff. between $\rho_{\beta,z}$ and $\rho_{\alpha,z}$ (Pitch-z or Roll-z)
Phase ϕ_z:	Custom distribution function of τ ($\tau_{\beta,z}$ or $\tau_{\alpha,z}$) based on which of the two correlations is higher ($\rho_{\beta,z}$ or $\rho_{\alpha,z}$)

- *X*

Starting pose $A_{0,x}$: Uniform distribution
Amplitude A_x: Uniform distribution
Frequency f_z: Custom distribution function of $max(\ |\rho_{\beta,x},\rho_{\alpha,x},\rho_{z,x}|\ )$
Phase ϕ_z: Custom distribution function of a τ ($\tau_{\beta,x}$, $\tau_{\alpha,x}$, $\tau_{z,x}$) based on max. ρ

- *Y*

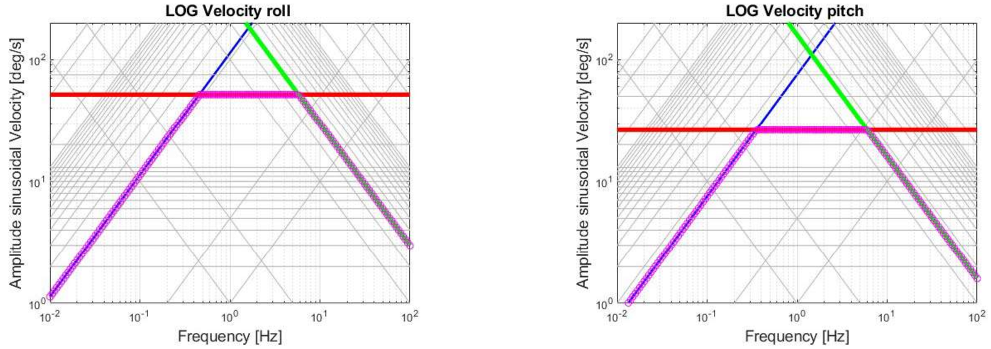
Starting pose $A_{0,y}$: Uniform distribution
Amplitude A_y: Uniform distribution
Frequency f_y: Custom distribution function of $max(\ |\rho_{\beta,y},\rho_{\alpha,y},\rho_{z,y},\rho_{x,y}|\ )$
Phase ϕ_y: Custom distribution function of a τ ($\tau_{\beta,y}$, $\tau_{\alpha,y}$, $\tau_{z,y}$, $\tau_{x,y}$) based on max. ρ

- *Yaw-γ*

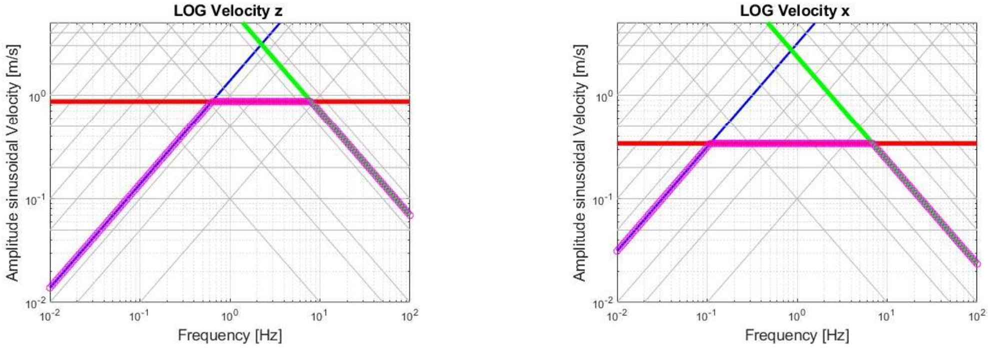
Starting pose $A_{0,\gamma}$: Uniform distribution
Amplitude A_γ: Uniform distribution
Frequency f_γ: Custom distribution function of $max(\ |\rho_{\beta,\gamma},\rho_{\alpha,\gamma},\rho_{z,\gamma},\rho_{x,\gamma},\rho_{y,\gamma}|\ )$
Phase ϕ_γ: Custom distribution function of a τ ($\tau_{\beta,\gamma}$, $\tau_{\alpha,\gamma}$, $\tau_{z,\gamma}$, $\tau_{x,\gamma}$, $\tau_{x,\gamma}$) based on max. ρ

Dummy Logarithmic graphs

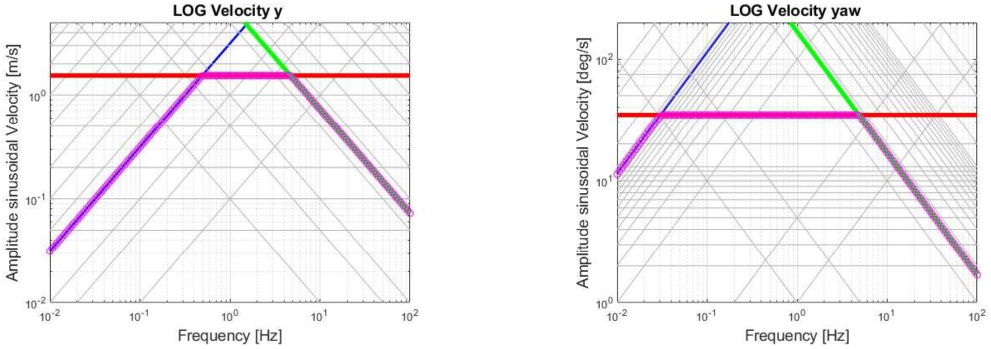
The other ingredient introduced in the modification of point 3. of list 6.2.2. It is the one related to the Logarithmic graph and it's been introduced in section 6.3.1. Following what has been described in that section and using the 131 different platform movements generated with AC simulations and different cueings, it is possible to finally define those introduced as "dummy" logarithmic graphs.



(a) "Dummy" Roll log graph (b) "Dummy" Pitch log graph



(c) "Dummy" Z log graph (d) "Dummy" X log graph



(e) "Dummy" Y log graph (f) "Dummy" Yaw log graph

Figure 7.8: Dummy logarithmic graphs obtained from AC simulations. Precisely, the blue line represents the limit in position, the red line represents the limit in speed, and the green line represents the limit in acceleration. The locus of the points represented as purple dots is what is called the dummy log graph.

Table 7.3: Position, velocity and acceleration limit coming from the analysis of the AC simulation.

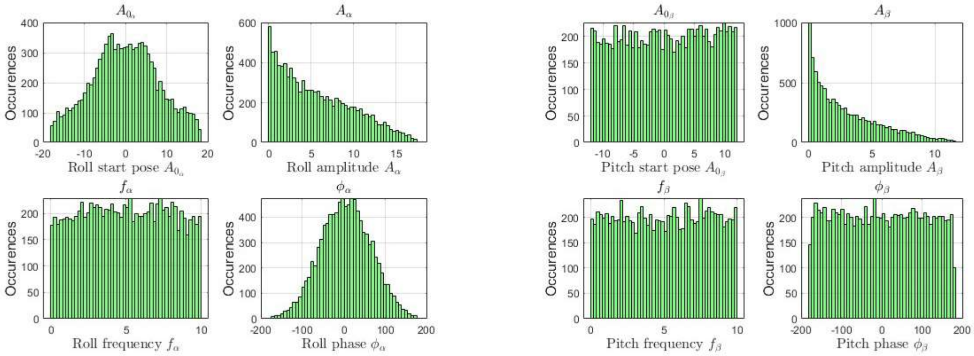
	Position	**Velocity**	**Acceleration**
X	0.5 $[m]$	0.3399 $[m/s]$	14.78 $[m/s^2]$
Y	0.5 $[m]$	1.5422 $[m/s]$	45.78 $[m/s^2]$
Z	0.22 $[m]$	0.87 $[m/s]$	43.58 $[m/s^2]$
Roll α	18 $[deg]$	51.65 $[deg/s]$	1878 $[deg/s^2]$
Pitch β	12 $[deg]$	26.03 $[deg/s]$	1002 $[deg/s^2]$
Yaw γ	180 $[deg]$	34.77 $[deg/s]$	1061 $[deg/s^2]$

The table 7.3 summarizes the limits of the logarithmic graphs shown in figure 7.8. It is possible to see that these limits are always presented as a positive number. This is due to how the single DoF performance is interrogated and since it is done involving sinusoids the positive limit represents the negative one.
At this point all the particularities of the method used have been introduced both from a theoretical point of view, in the previous chapters, and from a practical point of view.

7.3 MCM results

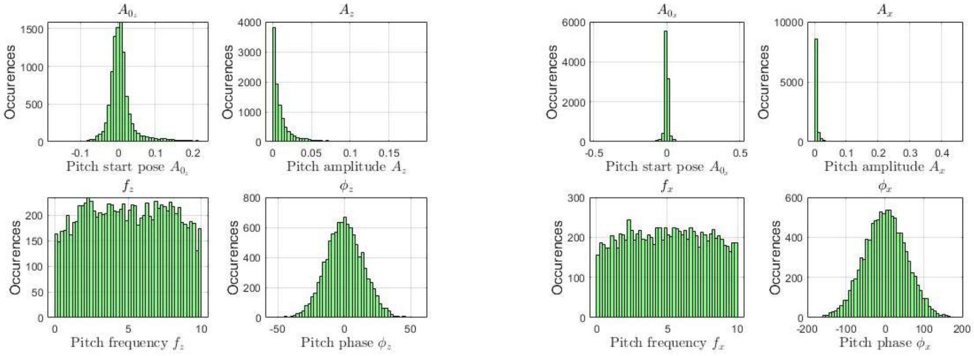
As mentioned from the outset, the goal was to address the problem of sizing in the context of dynamic simulation. It was chosen to do so using a biased statistical method from the use of realistic data.
The biased MCM was used to generate a large number of movements for the considered platform so that those information useful for sizing could be extracted. In particular, as explained in Chapter 6.1, it was used to define the β load factor (Eq. 6.11).
Precisely 10 000 simulations were generated through the introduced method. The story generator is used to define the 24 parameters, 4 for each of the sine waves of the 6 DoFs, that complete a platform movement. Figure 7.9 shows the values sampled from the custom distributions (exploiting ρ and τ). Each subfigure focuses on a single DoFs and shows the histogram for all four of its parameters. As can be seen from one of the subfigures, for example, the roll subfigure 7.9a, the starting position A_0 of each individual degree of freedom also takes on negative values. The randomness of the sign of the starting pose was introduced on all 6 degrees of freedom machines although the main reason is the complex dependence between the DoFs of the upper system, the parallel kinematics robot, which results in different admissible motions depending on the starting point. By doing so, a wider range of conditions can be investigated. In addition, another trend can be noted, which in this case concerns the effective amplitude

A of the sine wave. It can be seen that the histogram of the amplitude, first-row second column of each sub-figure of figure 7.9, is always centered toward low values. This is mainly related to the large number of simulations and the relationship between chosen frequency and the allowable amplitude of the sinusoid. As explained in the point in section 6.2.2 and precisely in the PDF sampling procedure 6.2.2, a value that meets certain constraints is chosen as the amplitude of the sinusoid. These constraints imply that the higher the frequency, the lower the amplitude of the sinusoid that can be realized. So, these parameters define histories on the individual DOFs of the machine that when combined define a complete platform movement and by doing so it is possible to obtain the various load factors, on each singular chain of actuation, corresponding to each movement. The results are presented below, plotted in histograms, with the shape of the load factor Beta for each axis of the simulator considered.



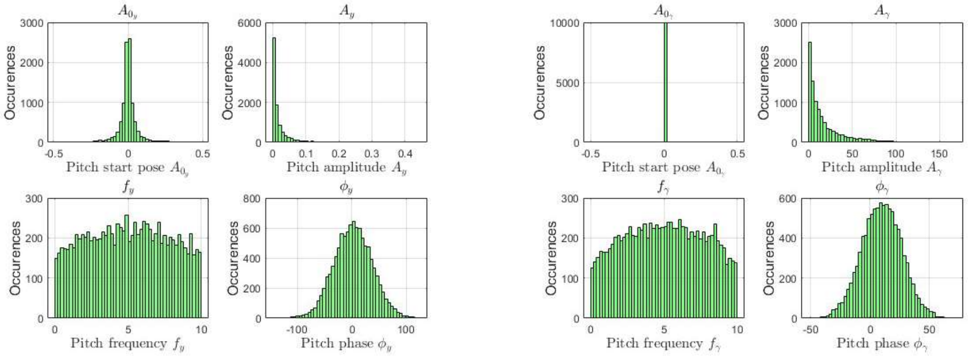
(a) Histogram for samples of the roll's sinusoid parameters

(b) Histogram for samples of the pitch's sinusoid parameters



(c) Histogram for samples of the heave's sinusoid parameters

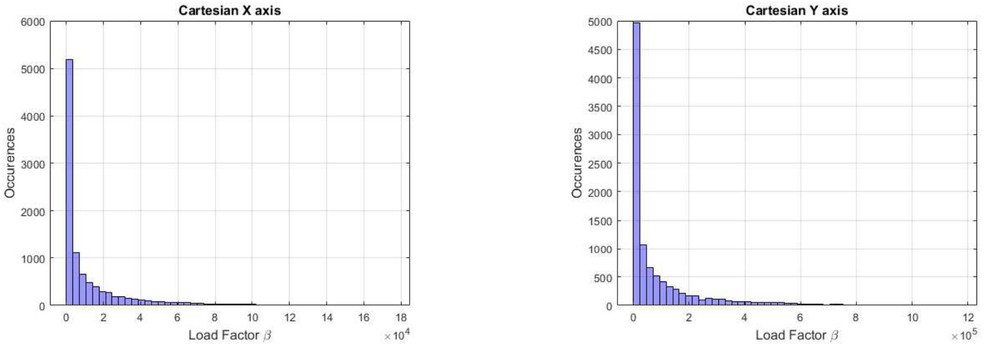
(d) Histogram for samples of the X sinusoid parameters



(e) Histogram for samples of the Y sinusoid parameters

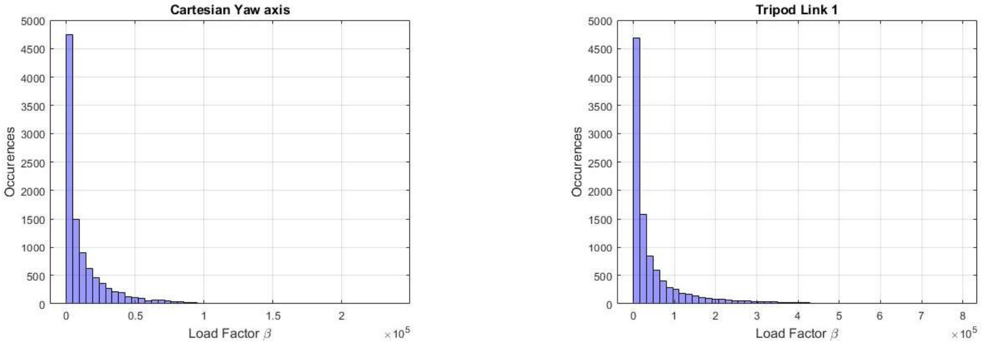
(f) Histogram for samples of the yaw's sinusoid parameters

Figure 7.9: Histograms representing the samples of the 6 DOFs describing the platform movements defined by the story generator



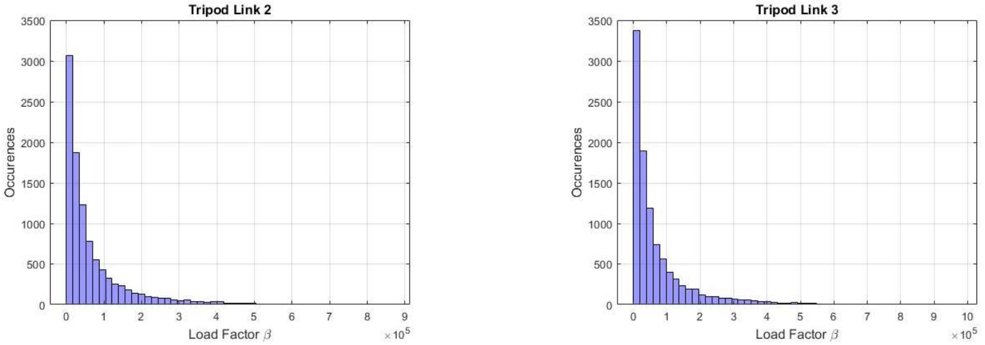
(a) Histogram for load factors on x-axis of the cartesian robot

(b) Histogram for load factors on y-axis of the cartesian robot



(c) Histogram for load factors on yaw-axis of the cartesian robot

(d) Histogram for load factors on 1^{st} kinematic chain of the RPZ PKM

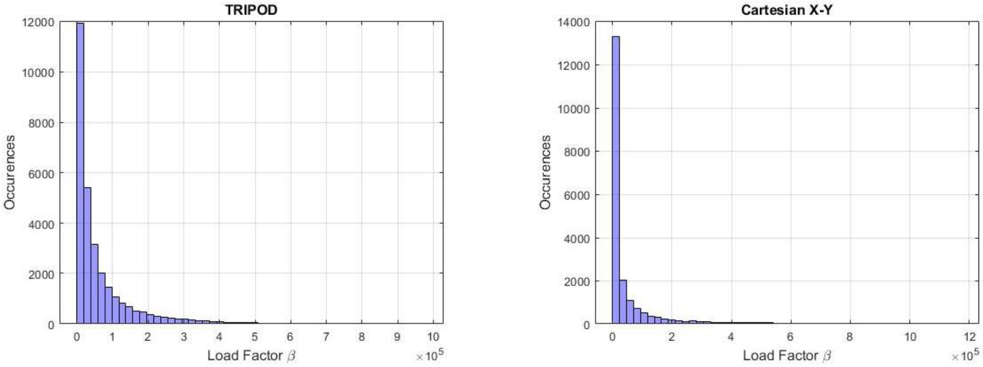


(e) Histogram for load factors on 2^{nd} kinematic chain of the RPZ PKM

(f) Histogram for load factors on 3^{rd} kinematic chain of the RPZ PKM

Figure 7.10: Histograms representing the load factors generated through the story generator on all six kinematic axes of the considered robot.

For real application convenience, It is helpful to define the same motor on the three kinematic chains of the tripod. The same thing is done for the x and y axes of the Cartesian. Wanting to use a torque motor for yaw actuation, is considered individually. So figures 7.10d 7.10e 7.10f can be merged as 7.10a 7.10b. Therefore, the following load factor distributions for the 3 choices to be made can be obtained:



(a) Histogram for the merged load factors of the tripod's kinematic chains

(b) Histogram for merged load factors on X-Y axis of the cartesian robot

Figure 7.11: Merged histograms

To avoid oversizing the actuation system, it's been chosen to use a load factor βcorresponding to 95 percent of the maximum generate load factor with the history generator (β_{95}).

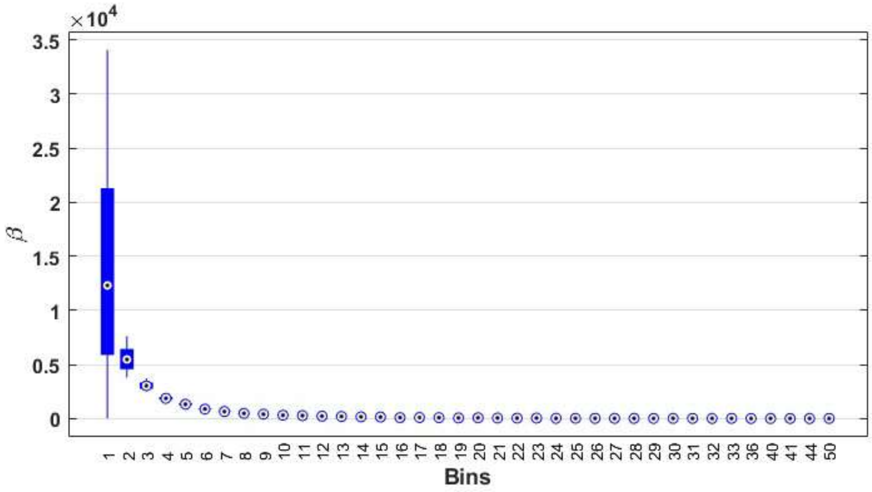
Tripod

Figure 7.12 represents the dispersion of the load factors, for the tripod, enclosed in the 50 bins in Figure 7.11a. As can be seen from Figure 7.12, the variations of the load factors generated in the first zone are very large. On the right side of the graph, it can be seen that the variability of the load factors variation diminishes significantly. This means that choosing to use a value associated with 95 of the coverage of the betas generated, represents a good estimate.

β_{95}	$1.9380e+05$	$[W/s]$

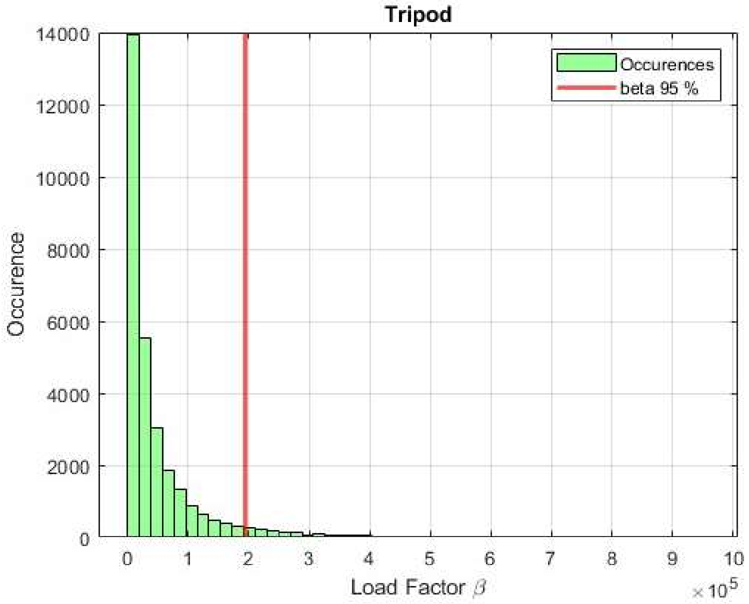
Table 7.4: β_{95} , the value considered to exclude engines that cannot be used based on limited performance

Figure 7.12: β dispersions over the 50 subsets defined by the histogram 7.11a



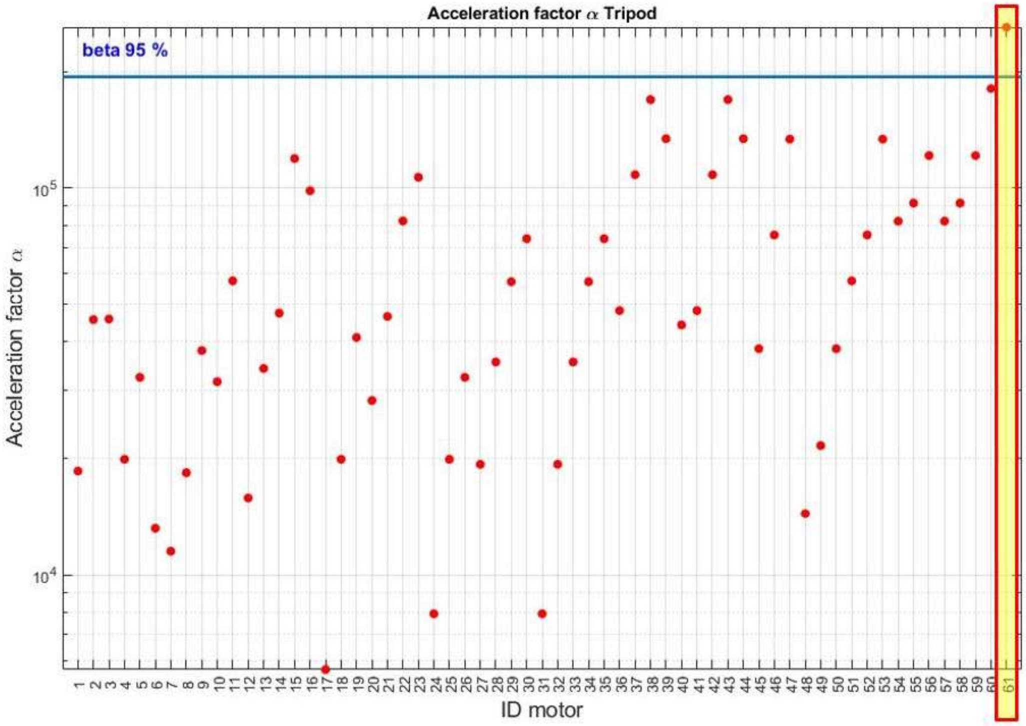
Also, it is possible to note that on the x-axis some references are missing, this shows that in some bins there are no factors demonstrating the above.

Figure 7.13: β_{95} for the tripod



At this point, it is time to recall the procedure described in section 6.1. Where, the equation 6.13 was introduced, which describes the situation that is sought in the choice of an actuation system, namely, that described in the point 6.1. threshold of β_{95}. In this particular instance, a dataset comprising engines obtained through collaboration with the company involved in this research and the introduction of the method and platform is employed.

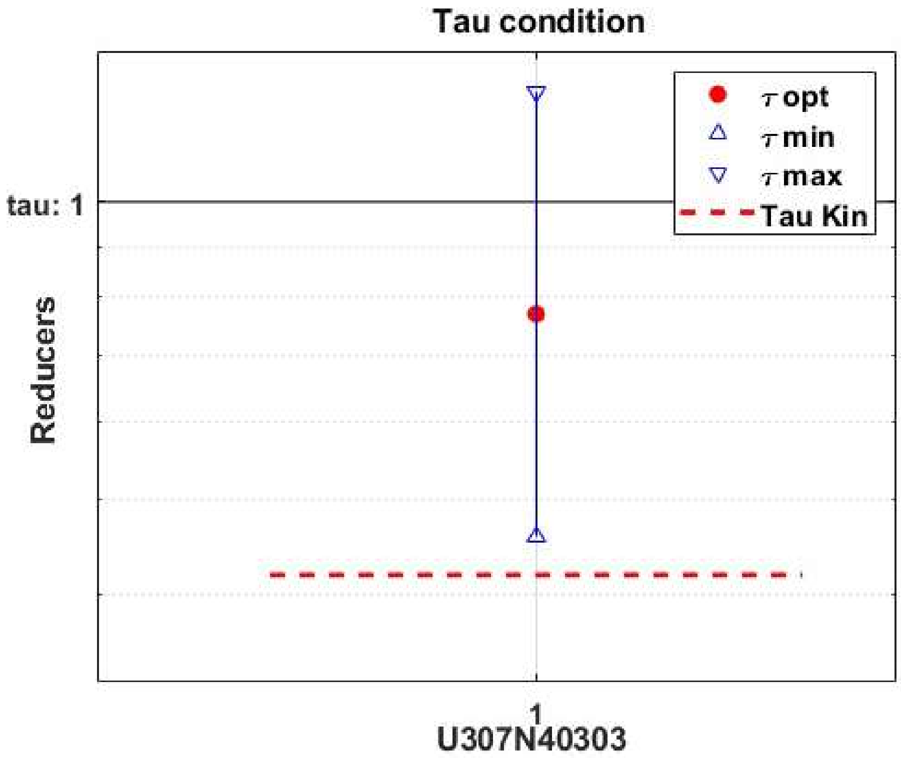
Figure 7.14: β_{95} of the tripod versus the accelerating factor of the involved motors



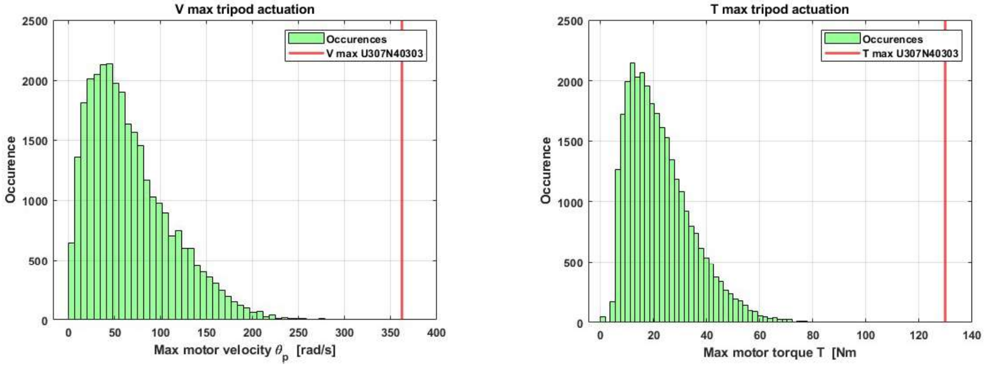
From the comparison shown in figure 7.14, the only one of the engines that satisfy the 6.1 condition is highlighted. Precisely it corresponds to the motor U307N40303 from the PHASE catalog.

The research does not end there, as explained in the procedure already mentioned 6.1 if the engine guarantees an alpha greater than the beta considered (β_{95}), it means that there is a range of gear ratios τ that satisfies equation 6.13. The limits of this range are defined by the tau τ expressed in the eq. 6.15. It is worth recalling the other condition 6.5 which results in the final condition 6.17. Direct coupling might be convenient conscious of the machine being considered and the high speeds that might be required during actual use. Furthermore, a reducer could introduce backlashes thus compromising the accuracy of the system. This led to the choice of using a coupling between motor and linear guides.

Figure 7.15: Admissibile τ range (τ_{min}, τ_{max}) for the chosen motor on the upper stage (tripod).



However, it must be verified that the choice of using direct coupling is feasible. It should therefore be verified that $\tau = 1$ is contained within the allowable range of tau. In figure 7.15, The rhombus symbol in green represents the minimum gear ratio τ_{min}, while the light blue rhombus represents the tau max τ_{max}. With a black horizontal line, the level corresponding to $\tau = 1$ is shown, and the red dot represents the value of tau that resets the tau-dependent component in equation 6.13 to zero. Thus, Figure 7.15 shows how the transmission ratio $\tau = 1$ is among the values that satisfy the equation 6.13.



(a) Histogram for the max velocities [rad/s] generated in each generated history at the actuation of the tripod

(b) Histogram for the max Torques [Nm] generated in each generated history at the actuation of the tripod

Figure 7.16: Charcteristic quantities at tripod actuation after the introduction of the chosen motors

The graphs in Figure 7.16 show how the chosen motor, with direct coupling, is able to meet the maximum speeds and torques obtained from the 10 000 stories generated by the static method. The red lines in graphs 7.16 show the max velocity and torque of the chosen motor.

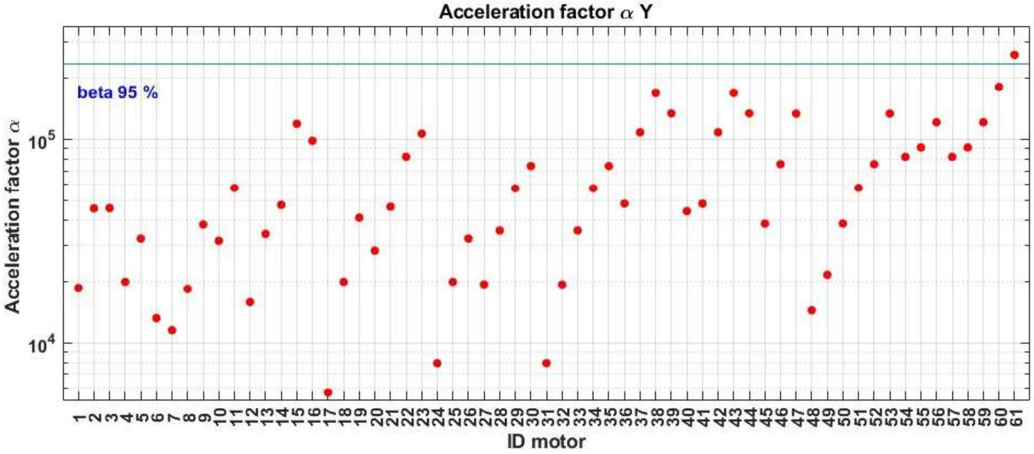
The same procedure should be repeated for the motor choice on the axes of the Cartesian: x, y, and yaw (γ).

Cartesian

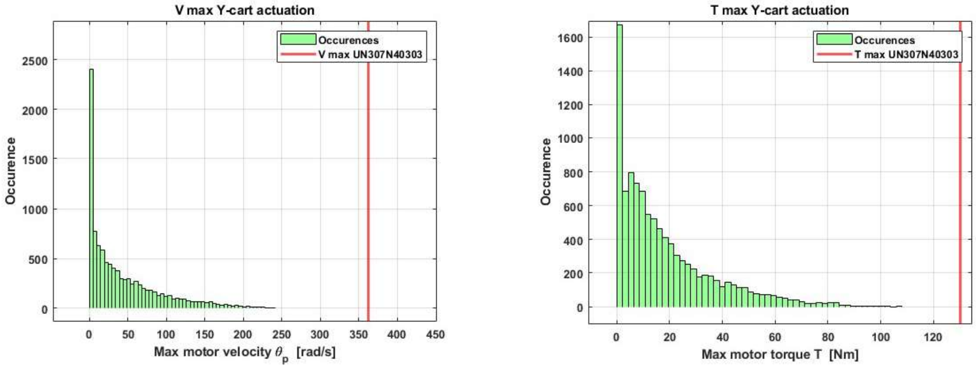
As done for the tripod, a realistic design solution is to mount on both axes (of the Cartesian robot) related to linear displacements in the same implementation. For this reason, only the load factors generated on the y-axis of the Cartesian were considered when choosing the motor. This represents the stage that supports everything else, so having to carry everything else in the machine will certainly have larger inertial loads.

X-Y axes Combining the X-axis and Y-axis loads would have distorted the use of the chosen percentage value because the loads generated on the X-axis have lower averages and would therefore have caused an imprudent or otherwise distorted choice due to the data used.

Figure 7.17: β_{95} of the the Y cartesian axis versus the accelerating factor of the involved motors



As for the tripod, the motor that shows an α greater then β is the one identidfied by the $ID = 61$. It is the U307N40303. Direct coupling is involved ($\tau = 1$).



(a) Histogram for the max velocities [rad/s] generated in each generated history at the actuation of Y-axis

(b) Histogram for the max Torques [Nm] generated in each generated history at the actuation of Y-axis

Figure 7.18: Charcteristic quantities at tripod actuation after the introduction of the chosen motors

As can be seen from the distributions of the maximum speed 7.18a carried to the motor and the maximum torque 7.18b required to the motor from the histories generated with MCM, the PHASE U307N40303 is able to meet the demands to the Y-axis implementation. As aforementioned, for convenience, as is often done in the industrial design process, it has been chosen to consider the same motor on the x-axis as well. It is slightly oversized for this axis since, being less loaded than the y-axis, it

receives smaller loads. But this oversizing is repaid with the convenience of installing two identical motors.
For the yaw actuation, some consideration must be made.

Yaw axes The considered system comprises two distinct subsystems: the upper stage and the lower Cartesian system. The upper stage generates significant inertial loads that need to be sustained and transported by the lower Cartesian system during the execution of dynamic simulations. Additionally, for the yaw motion, there is a specific requirement to employ a torque motor to ensure a wide range of motion without limitations on stroke. It's worth noting that torque motors, while offering high torque capabilities, exhibit high inertia and operate at relatively low speeds.

Figure 7.19 illustrates the load factors associated with the torque motors in question, indicating their ability to meet the alpha constraint. However, Figure 7.20 reveals that in this particular case, the torque motor cannot be employed, as the solution with tau = 1 does not meet the acceptable criteria.

Figure 7.19: β_{95} of the Yaw cartesian axis versus the accelerating factor of the involved motors

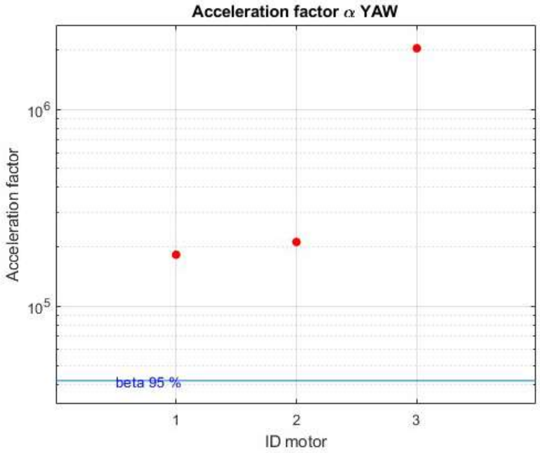
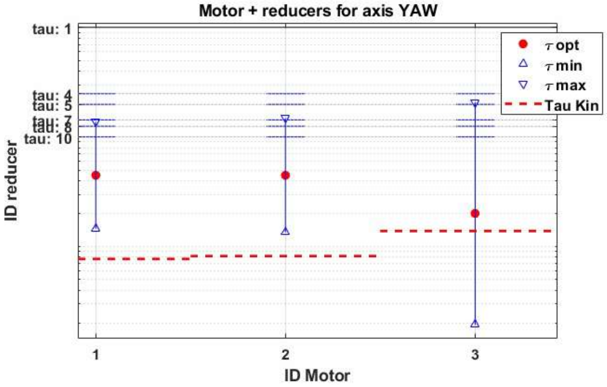
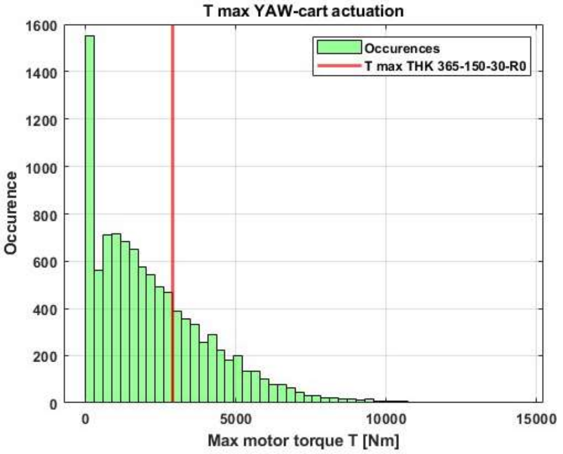


Figure 7.20: Admissibile τ range (τ_{min}, τ_{max}) for the chosen motor for Yaw actuation.



As explained in the previous chapters, verification of the thermal check (6.13) is not sufficient. It represents a condition for the continuous operation of the motor-load. So, if this is verified it must therefore be checked that the motor, chosen also the transmission, is able to compensate for the peak torques.

Figure 7.21: Histogram for the max Torques [Nm] generated in each generated history at the actuation of Yaw-axis



In this case, it is possible to see (figure 7.21)how the motor chosen for yaw axis actuation is able to verify peak torques of only a percentage of those generated. Precisely this torque phase motor is able to cover 70% of the peak torques. it is well understood how the statistical method used to generate the numerous time histories is questioning the possible use of this motor for the considered application. At this point in the analysis of the results, it is necessary to mention that the motors used to verify the tripod axes, the x-y axes of the Cartesian, are components from the supplier's catalog. So they are components that tend to be commercially available and from relatively low prices. In

contrast, the motor being considered as a possible solution for yaw axis actuation is a non-catalog motor. It is built only upon request from the manufacturer and therefore timelines are lengthened but most importantly the price increases considerably. So, we chose to use such a motor as a boundary condition thus proceeding in the design phase by choosing such a motor for yaw axis actuation aware of its peak torque limits. As mentioned above, the modularity of this simulator creates this type of inconvenience by offloading inertial loads of the upper system onto the lower system because of the distribution of degrees of freedom in the two subsystems. If increasing the coverage for the required torques on the yaw axis became a key prerogative, one could consider bringing the yaw axis implementation to TCP (upper stage). This would obviously require redesigning the two stages.
So the motors chosen for the six axes are:

- PHASE U307N40303 for the tripod's actuation and the $X-Y$ cartesian axes.
- PHASE THK 365-150-30-R0 for the yaw actuation.

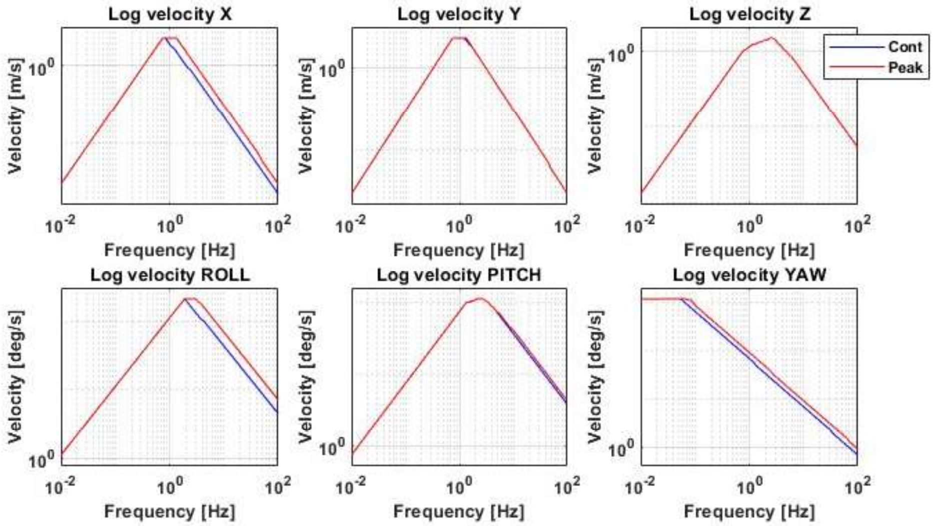
The following table 7.5 summerize the main characteristic of these motors.

Manufacturer	**Code**	**RMS speed [rpm]**	**MAX speed [rpm]**	**RMS Torque [Nm]**	**MAX Torque [Nm]**
PHASE	U307N40303	3000	3466	26	130
PHASE	THK 365-150-30-R0	111	240	661	2881

Table 7.5: Summary of the torque and speed characteristics of the motors chosen for axes actuation.

The transmisisoon of parallel-kinematic robots is often composed of ball screws, and that solution was also chosen in this application. In particular, a 0.04[m] pitch screw was chosen for the Cartesian and a 0.025 [m] pitch screw for the tripod.
Having made the necessary considerations and then defined the motors for each individual implementation, it is possible to characterize the machine with logarithmic graphs for each degree of freedom. The hour machine is fully defined by considering: masses, inertia. torque limits, speed limits of the motor-gearbox arrangements.

Figure 7.22: Logarithmic graphs for each platform DoFs obtained considering the motors in tab 7.5



	Position		Velocity		Acceleration	
	Peak	**Cont.**	**Peak**	**Cont.**	**Peak**	**Cont.**
X	± 0.5 [m]	± 0.5 [m]	2.2431$[\frac{m}{s}]$	2.2361 $[\frac{m}{s}]$	2.01g	1.93g
Y	± 0.5 [m]	± 0.5 [m]	2.2431$[\frac{m}{s}]$	2.2413 $[\frac{m}{s}]$	2.01g	1.44g
Z	± 0.22 [m]	± 0.22 [m]	1.51$[\frac{m}{s}]$	1.51$[\frac{m}{s}]$	3g	3g
Roll	± 18 [deg]	± 18 [deg]	212$[\frac{deg}{s}]$	212$[\frac{deg}{s}]$	4400 $[\frac{deg}{s^2}]$	2827 $[\frac{deg}{s^2}]$
Pitch	± 12 [deg]	± 12 [deg]	112 $[\frac{deg}{s}]$	112$[\frac{deg}{s}]$	2660 $[\frac{deg}{s^2}]$	2370 $[\frac{deg}{s^2}]$
Yaw	± inf [deg]	± inf [deg]	1219 $[\frac{deg}{s}]$	1219 $[\frac{deg}{s}]$	592 $[\frac{deg}{s^2}]$	437 $[\frac{deg}{s^2}]$

Table 7.6: Platform DoFs performances

The graphs shown in Figure 7.22 are obtained by following the procedure described in section 3.3.2. Compared to the dummy ones obtained from the running trim data these actually provide the performance on the single degree of freedom of the machine. Logarithmic graphs, being a useful tool for visualizing the performance of the single DoF of a complex machine such as robots used in the field of dynamic simulation, they can therefore be used to compare different mechanisms (as will be shown later).
Upon comparing the graphical data presented in the summary table of the logarithmic graphs (Table 7.3) and that of Table 7.6, it becomes evident that the velocity constraints are adequately adhered to. However, it is noteworthy that the selected implementations

impose certain limitations, particularly in meeting the acceleration requirements for the y and yaw motions.

Furthermore, it should be noted that the accelerations generated by the actuation system for the y-direction, measured in gravitational units (g), are considered sufficient. This assessment is based on the unique characteristics of the application.

Having fully defined the implementation through the statistical method, it is also possible to reutilise this method for the mechanical sizing of the structure. The considerations made so far about the difficulty in defining the critical condition useful for the choice of motors can be made in parallel for the definition of the critical condition of the stresses generated on the structure for the mechanical sizing of it.

7.4 Mechanical Sizing with custom MCM

In the previous section, it was shown how the specialised statistical method can be involved in defining the implementations of a complex system such as the dynamic platform introduced in this thesis. In particular, for the choice of actuations, a procedure already defined in the literature was used, which involved the definition of the load factor β. Once the implementation of the system has been fully defined, the same procedure (customised MCM) can be used to generate the loads to be used for the mechanical dimensioning of the structure. One main difference is present between the procedure used for defining load factors and that for defining loads for mechanical dimensioning: how the logarithmic graphs involved are defined. For mechanical dimensioning, as the dynamic limits are defined with the implementation, the logarithmic graphs involved are actually those defined by the chosen engines. Thus, through this new use of the introduced method, it is possible to evaluate the characteristic physical quantities for the dimensioning of the structure.

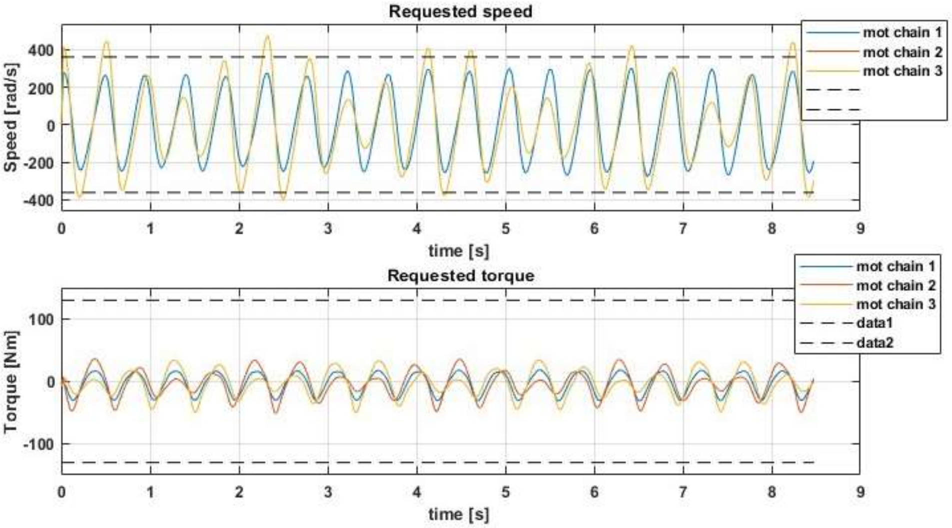
Especially, the forces useful for the mechanical sizing procedure of the tripod are:

- $F_{link,i} (i = 1, 2, 3)$ as directed as each tripod link
- $F^{A}_{L,i}$, lateral force in the actuation plane
- $F^{G,U}_{V,i}$, forces along the z-axis of the global upper reference systems
- $F^{A}_{Q,i}$, aligned with rotated x-axis of the actuation plane. It takes into account the rotation of μ around the y-axis of the actuation R.S.
- $F^{A}_{X,i}$, forces along the x-axis of the actuation plane
- $F^{A}_{Z,i}$, forces along the z-axis of the actuation plane

- $M^A_{Z,i}$, moment around the z-axis of the actuation plane
- $M^A_{V,i}$, moment around the z-axis of the actuation plane belonging to the reference system rotated by μ
- $M^A_{Q,i}$, moment around the x-axis rotated by μ
- $M^A_{X,i}$, moment around the x-axis of Actuation R.S. .

Instead, for the cartesian are the forces on the x-y actuation axis whose distrubutions are shown in figure 7.34 - 7.35 and the system of forces and moments exerted by the upper stage on the lower stage (from figure 7.36 to 7.41)

Figure 7.23: Requested motor torque and speed for not admissible motion



Using the same statistical method, despite the precautions taken when defining the platform movement (utilizing log graphs as limits for velocities and accelerations on each individual degree of freedom), not all of the generated histories can be practically realized using the selected actuation system. Figure 7.23 illustrates the three torques and the three speeds required at the three actuations for one of the 10,000 simulations generated. It can be observed that the required speed exceeds the peak speed capability of the chosen motor. For simplification, these quantities are represented relative to the tripod only. However, they are sufficient to illustrate how certain simulations defined by the statistical method may not be feasible with the chosen actuation system.

The presence of simulations that cannot be realized gives rise from the mechanical dimensioning point of view to the need, as done previously for the choice of motors, to decide whether to dimension the structure considering all simulations or only those that can actually be realized. From figure 7.25 to 7.41 it is possible to see the histograms representing the distributions of the forces considered above. It can be seen that both the number of occurrences and the width of the distribution are lower in the case of the simulations that can actually be carried out with chosen actuation.

Precisely, each characteristic quantity is represented with its distribution for all 10,000 simulations generated in combination with the distribution of only the values of the simulations admissible with the drives (motors and transmission) chosen. Excluded simulations include simulations excluded due to violation of kinematic limits and dynamic limits. A simulation is excluded upon violation of even one kinematic limit and/or dynamic limit for a specific axis. The permissible simulations obtained with the presented method are approximately 62% of the total. However, some considerations must be made in this regard:

- the admissible simulations for the tripod are more than 93% of the totals.
- the admissible simulations for the Cartesian x-y axes are 99%.
- the admissible simulations for the yaw axis are 65%.

The values quoted in this list were obtained by considering limits on specific axes only. Instead, the value of globally admissible simulations (62%) is obtained by comparing all violations.

As is clear from what has just been described, the bottleneck for admissible simulations is the implementation of the yaw axis as described above. Therefore, it is possible to see how the method, barring the limitations due to the implementation of the yaw axis, is able to generate a large number of admissible simulations useful for dimensioning.

Figure 7.24: Distribution of max force directed as tripod's link

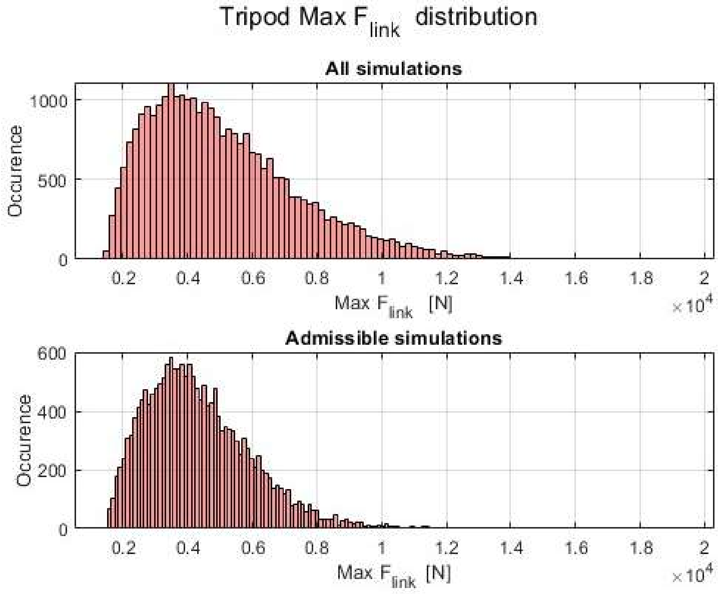


Figure 7.25: Distribution of max lateral force in tripod's actuation planes

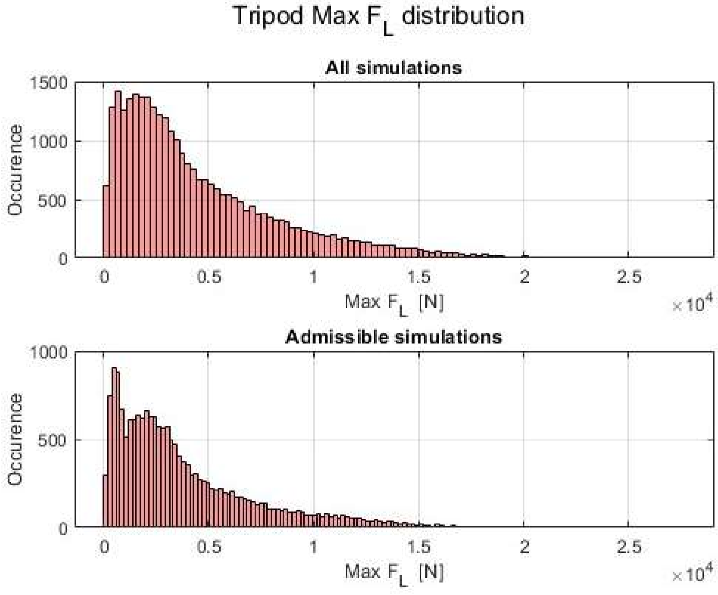


Figure 7.26: Distribution of max force directed as z-axis of Actuation R.S. rotated by μ

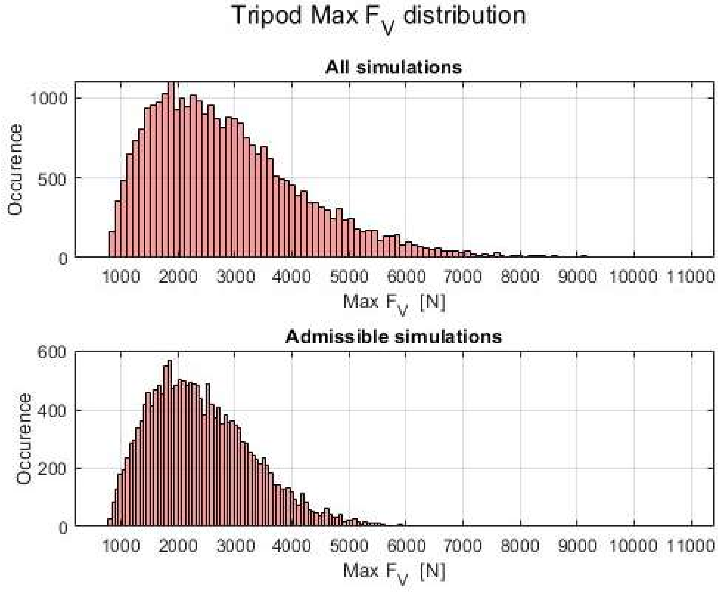


Figure 7.27: Distribution of max force directed as x-axis of Actuation R.S. rotated by μ

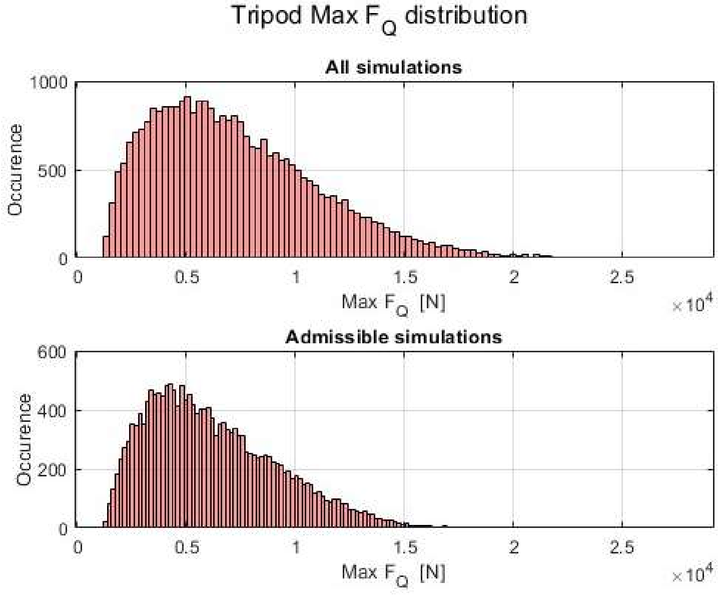


Figure 7.28: Distribution of max force directed as x-axis of Actuation R.S.

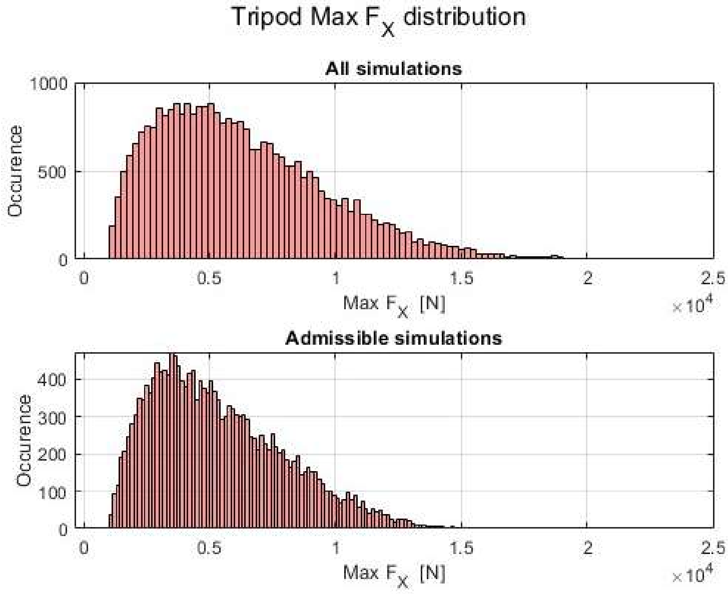


Figure 7.29: Distribution of max force directed as z-axis of Actuation R.S.

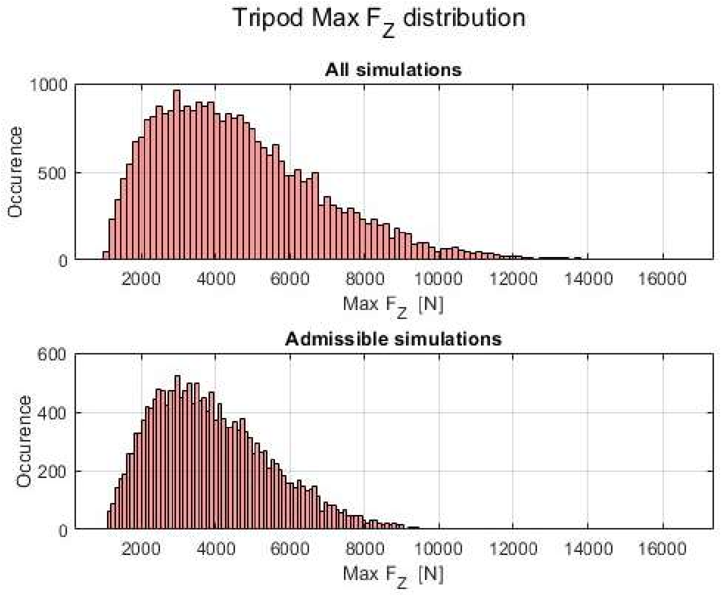


Figure 7.30: Distribution of M_Z around the z-axis of Actuaion R.S.

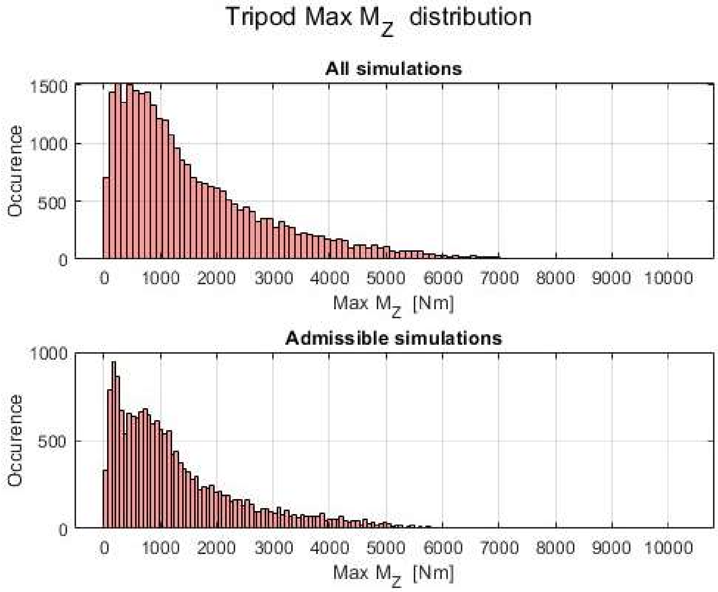


Figure 7.31: Distribution of M_V around the z-axis of Actuaion R.S. rotated by μ

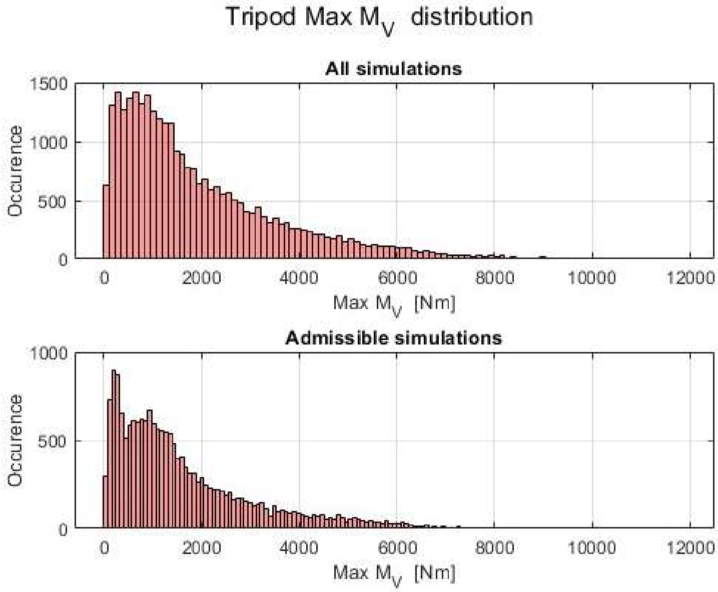


Figure 7.32: Distribution of M_Q around the x-axis of Actuaion R.S. rotated by μ

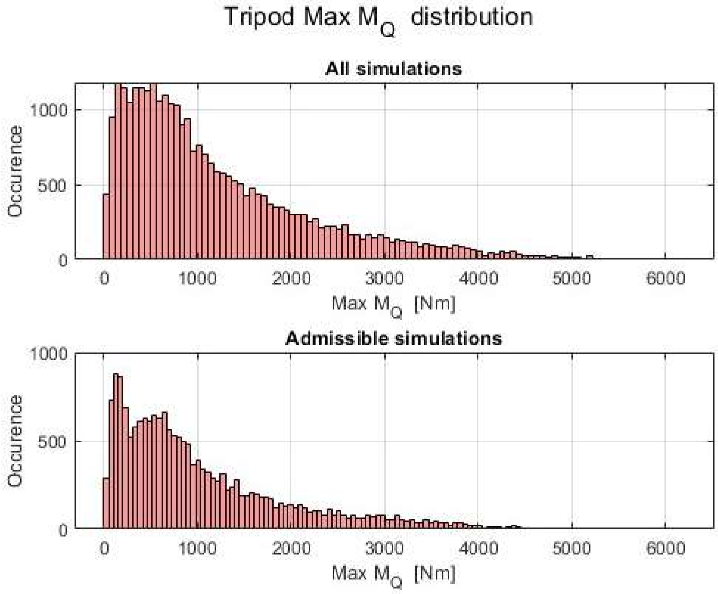


Figure 7.33: Distribution of M_X around the x-axis of Actuaion R.S.

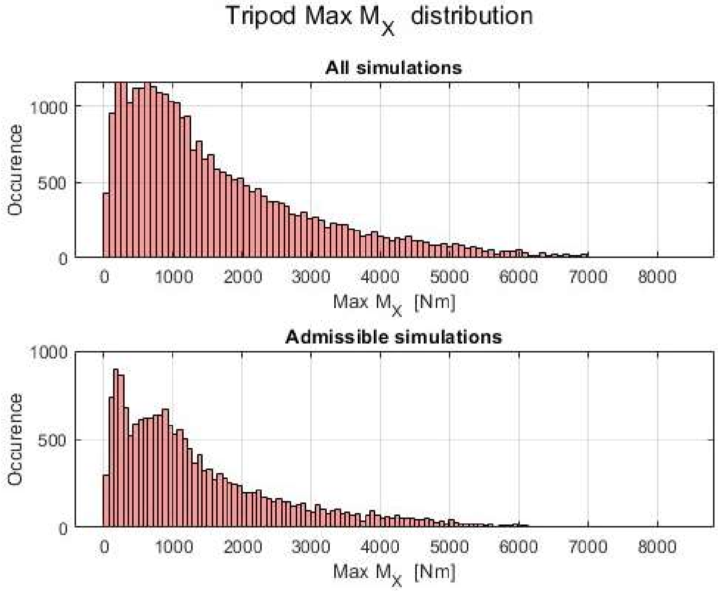


Figure 7.34: Distribution of max force directed as x- actuation axis of the cartesian robot

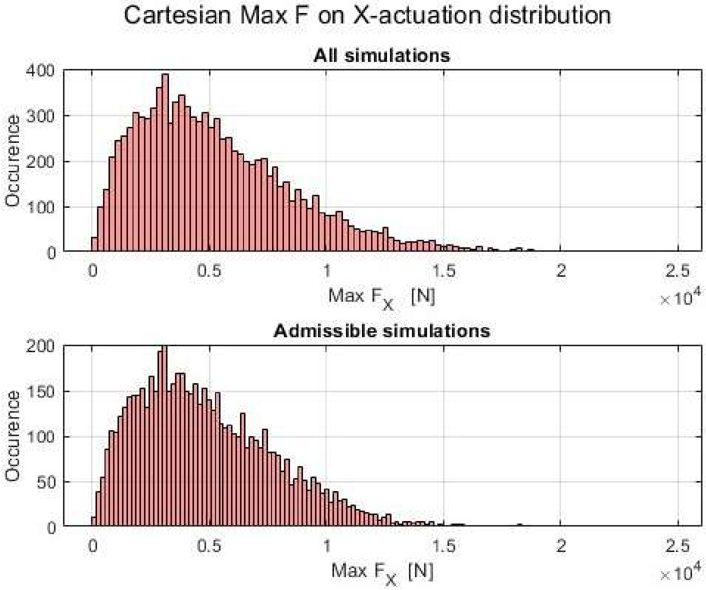


Figure 7.35: Distribution of max force directed as y-actuation axis of the cartesian robot

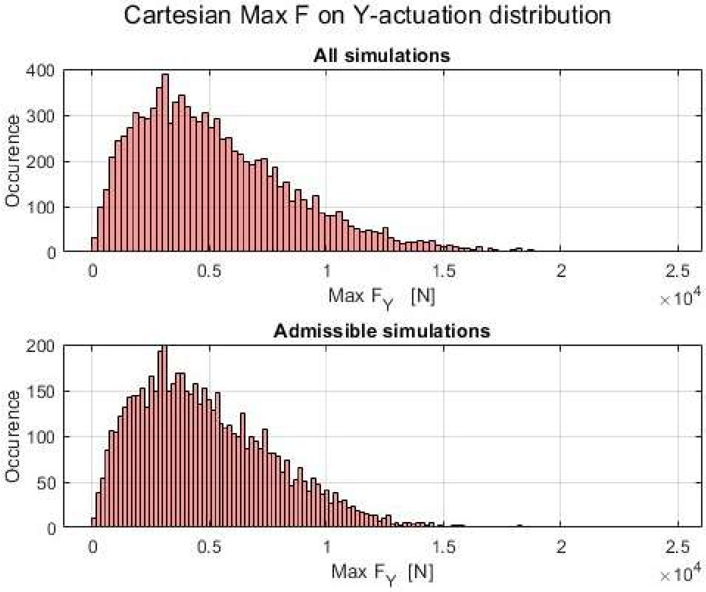


Figure 7.36: Distribution of max force discharged from the upper system to the lower one aligned as the x-axis of the Global Lower R.S.

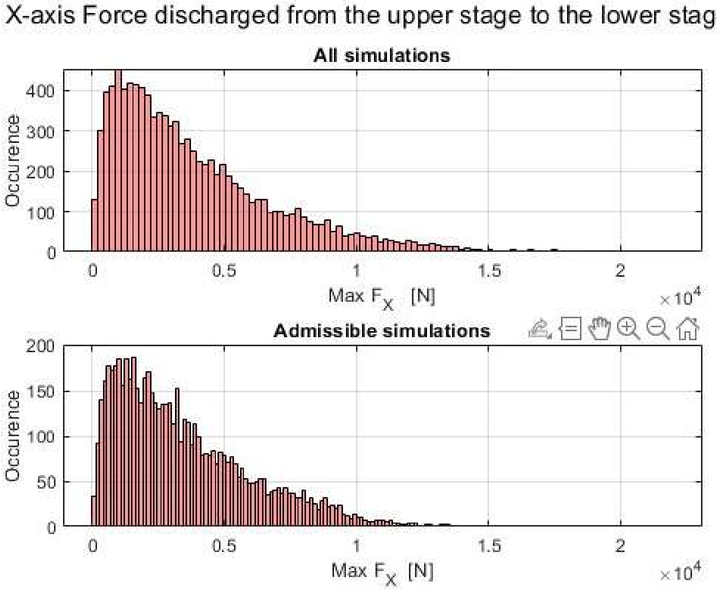


Figure 7.37: Distribution of max force discharged from the upper system to the lower one aligned as the y-axis of the Global Lower R.S.

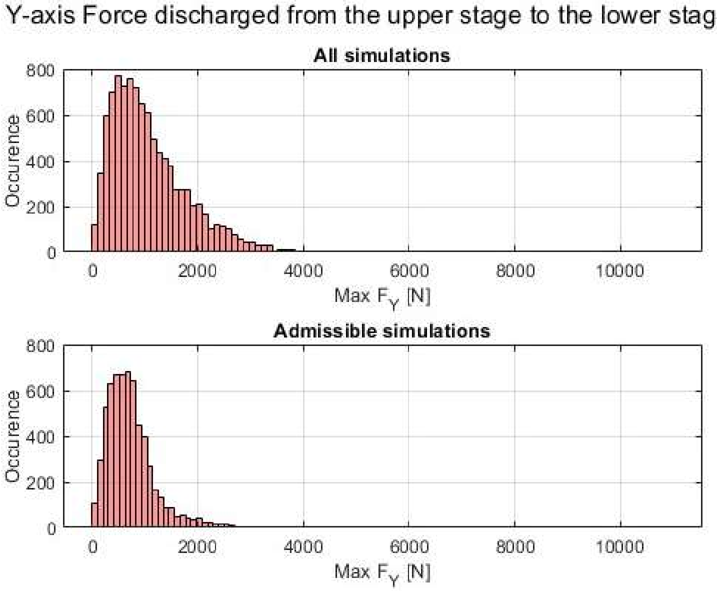


Figure 7.38: Distribution of max force discharged from the upper system to the lower one aligned as the z-axis of the Global Lower R.S.

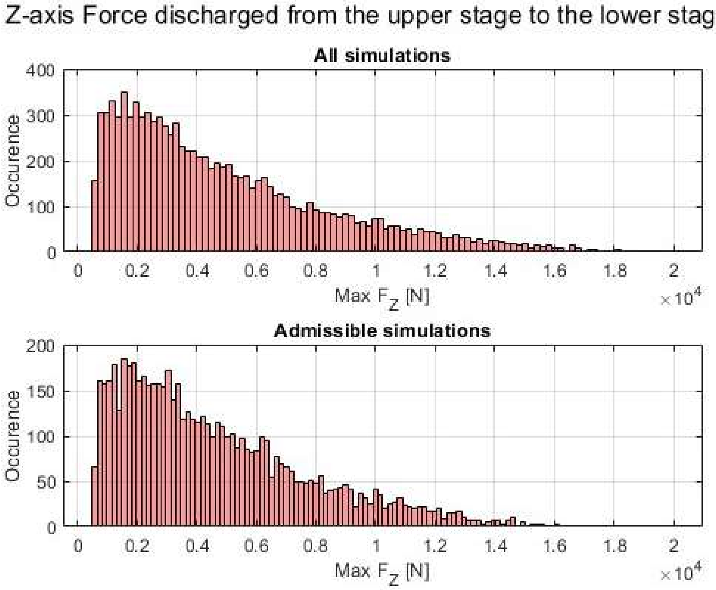


Figure 7.39: Distribution of M_x (around x-axis of the Global Lower R.S.) discharged from the upper system to the lower one

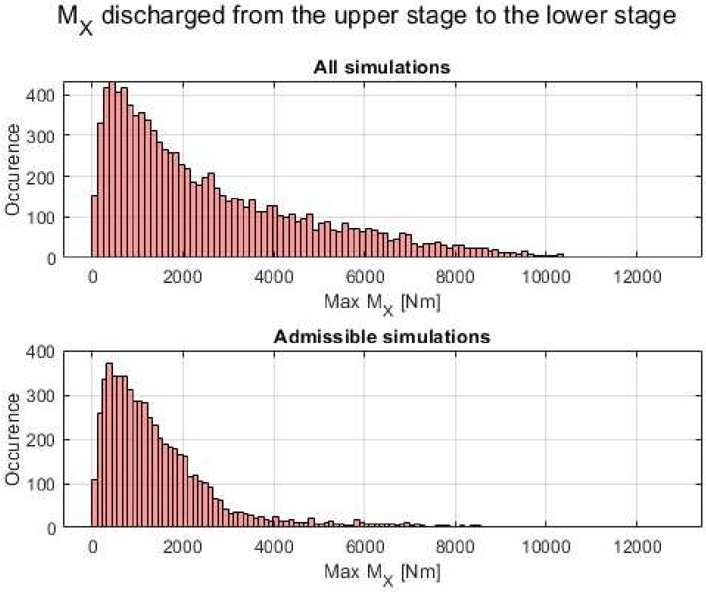


Figure 7.40: Distribution of M_y (around y-axis of the Global Lower R.S.) discharged from the upper system to the lower one

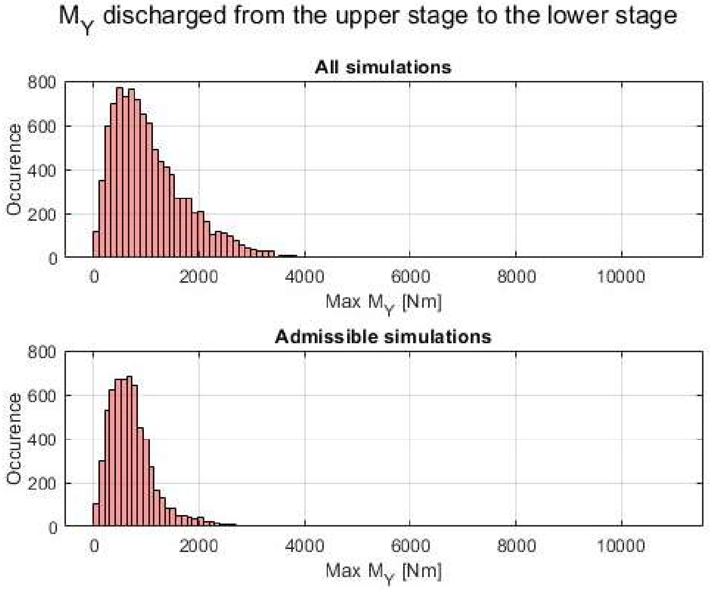
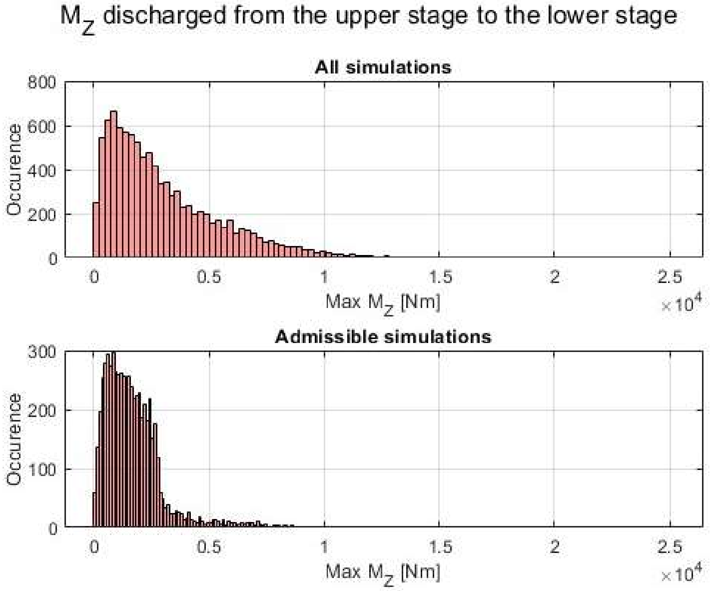


Figure 7.41: Distribution of M_z (around z-axis of the Global Lower R.S.) discharged from the upper system to the lower one

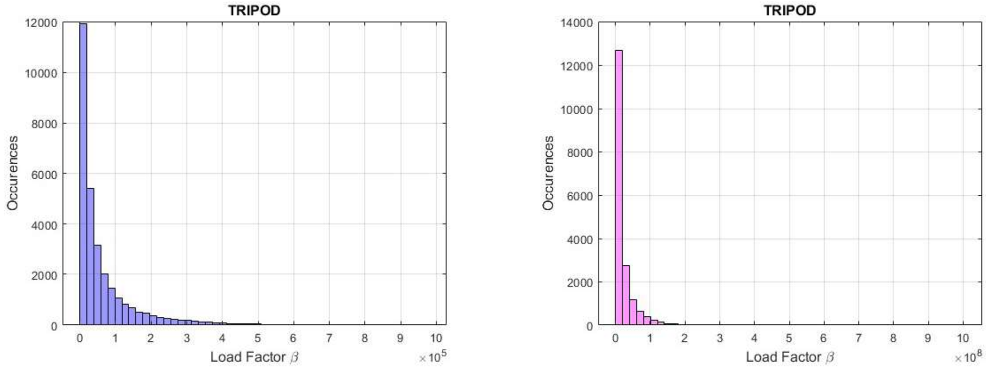


In the case of engine selection, exceeding their limits would lead to saturation and failure to replicate the imposed dynamics. This would lead to problems with the control and realism of the simulation, but would not be a problem for the safety of the pilot and operators. On the contrary, in the case of mechanical dimensioning, the presence of simulations that exceed the allowed limits leads to the failure of the integrity of the structure due to one or more components. This represents a possible dangerous situation. As mentioned above, the presence of simulations that cannot be carried out due to the limits of the chosen motors leads to the question of which simulation to use for dimensioning. Force values from unfeasible simulations could also be used if they had a sufficient safety coefficient to guarantee structural integrity. For some of the forces shown above, the ratio between the maximum permissible value and the maximum generated (absolute) value is less than 2. Generally, a safety factor between 2.5 and 3 is acceptable for applications of this type.

7.5 Monte Carlo Method: customized vs generic

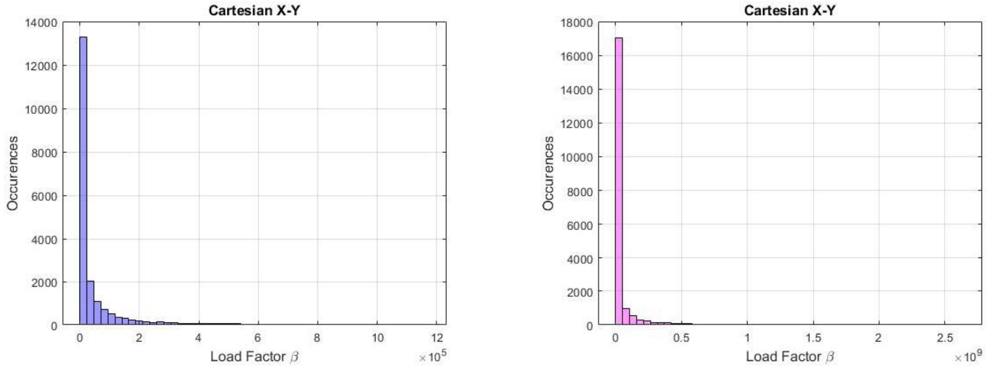
Briefly, this section aims to provide a comparison between the generic Monte Carlo Method and the specialized MCM obtained by introducing all the modifications presented in this thesis. In particular, it is intended to highlight how the adjustments put in place through the study of stroke attitude data analysis are fundamental to obtaining more realistic simulations. To recap briefly, the generic Monte Carlo method is the one introduced in 6.2.2: each sinusoid parameter is sampled by a uniform probability density function. The chosen choices of amplitude and starting pose are influenced only by the constraints provided by the workspace (information on static poses reachable from the platform and not dynamic poses). In addition to generating more realistic stories related to the scope, the changes made are intended to reduce as much as possible the simulations that cannot be realized by the system. The definition of the initial pose and the effective amplitude of the sine wave undergoes a variation to be emphasized in case the introduced changes are not considered. In the case of the generic MCM, the only limitation on the choice of these two parameters is provided by the mutual dependence between the degrees of freedom. So, for the upper stage, the parallel kinematics robot, for which this feature has already been mentioned and also shown through the figure 6.4 this filtering exists, although only related to the kinematic limitations due to the structure. For amplitude and initial pose of the lower stage, the Cartesian, there is no limitation since the degrees of freedom and the related workspace are completely decoupled (a motion on x does not limit, in terms of available travel, a motion on y). This difference, theoretically, should lead to unrealistic motions for the system under consideration. In particular, the absence of a limit on the dynamics of the generated motions should lead to very high load factors compared to those previously defined. this is because regardless of the sampled frequency, the current degree of freedom can have any amplitude (within the limits of the workspace) and thus generate unrealistic (very high) velocities and accelerations.

Instead, the customized MCM 7.2 was defined by exploiting the analysis performed involving a parameterized cueing algorithm and virtual simulations coming from AC. It corresponds to the generic procedure with the modifications introduced in 6.3.
The following comparison takes into account the load factors generated by 10 000 complete movements using the two methods just mentioned: generic MCM and customized MCM.



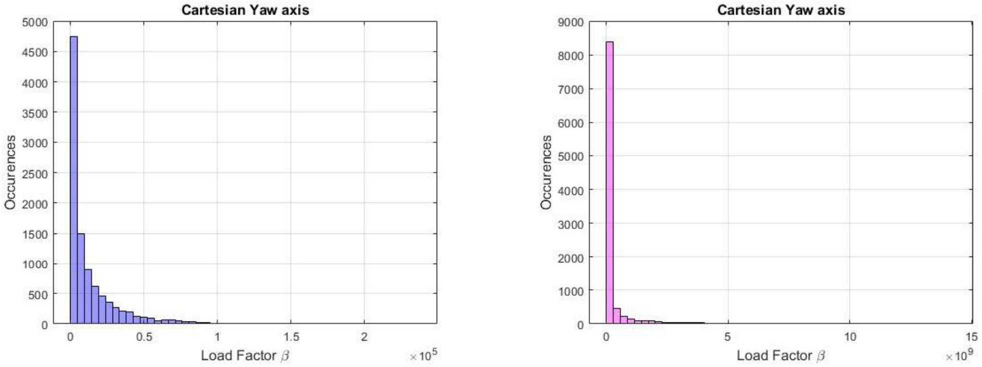
(a) Customized MCM (b) Generic MCM

Figure 7.42: Tripod Load factor (β) histogram: Customized MCM vs Generic MCM



(a) Customized MCM (b) Generic MCM

Figure 7.43: X-Y Load factor (β) histogram: Customized MCM vs Generic MCM



(a) Customized MCM

(b) Generic MCM

Figure 7.44: Yaw Load factor (β) histogram: Customized MCM vs Generic MCM

The trends in the graphs turn out to be very similar except for one change that is, however, fundamental to the evaluation of the two methods: the full scale on the x-axis. As it can be seen we go from a full scale of 10^5 to one of 10^9. This change in full scale obviously indicates an increase in β values for all axes implemented in the generic MCM case. In order to use these results for the definition of the implementation system of the complex platform this would result in the oversizing. The causes of this significant increase in the load factor lie mainly in the differences of the two methods: in particular in not considering any dynamic limits. This leads the generalized method to generate stories in which the frequency and amplitude chosen are incompatible with realistic handling. In contrast, the use of coupling information between the GDLs leads to having motions that are more related to the scope considered.

Chapter 8

Mastering Tools: Unveiling Practical Applications

[...]

8.1 Hexalift

[...]

8.1.1 Requirements

[...]

8.2 Static and Dynamic comparison

[...]

8.2.1 Performance comparison

[...]

8.2.2 Exploring platform variations

[...]

8.2.3 Workspace investigation

[...]

Chapter 9
Conclusions

Beginning with a comprehensive review of driving simulators, the study highlighted the intricate nature of these mechatronic solutions, often referred to as Human-In-the-Loop (HiL) systems. Notably, the inclusion of an operator, typically a pilot in the context of driving simulators, adds complexity by influencing system motion and, consequently, actuation performance. In the automotive domain and similar fields, the structures used for dynamic HiL simulators are frequently based on parallel kinematics robots or hybrid, redundant designs.

Within this context, the study addressed the challenge of defining critical parameters for mechanical and structural sizing. This challenge arises from the uniqueness of each movement due to the presence of a human operator and the specific kinematics of these systems, especially in the automotive sector. To assist in the design and performance evaluation of complex machines like these, various tools were introduced, including workspace analysis, single Degree of Freedom (DoF) logarithmic graphs, and kinematic crosstalk evaluation.

To address the issue of defining characteristic quantities for mechanical and structural sizing, a specialized procedure was introduced. This builds upon an established method for actuation selection, which involves defining two key parameters: the alpha (α) accelerating factor and the beta (β) load factor. The study demonstrated how a statistical method like the Monte Carlo method could be employed to define a wide range of scenarios, avoiding the exclusion of specific movement types. The standard Monte Carlo model was tailored for this application by incorporating information from simulations, particularly those from Assetto Corsa.

The simulations from Assetto Corsa provided insights into correlations between various degrees of freedom. To maintain the generality of the Monte Carlo method, different parameterizations of the cueing algorithm were employed, considering factors that make each simulation unique, such as the car, driver, and track. The cueing algorithm plays a crucial role in transforming vehicle degrees of freedom into platform degrees of freedom, further adding to the uncertainty in defining characteristic handling and sizing parameters.

This specialized procedure was integrated into the design of a compact 6-degree-of-freedom simulator featuring two distinct structures: the Cartesian (x, y, and yaw) and the tripod (RPZ PKM). The thesis delved into the kinematics and dynamics of both solutions, utilizing a fully parameterized kinematic model and an analytical dynamic model. The latter proved essential for efficiently evaluating critical quantities like required torques and generated forces.

In conclusion, the study showcased how this procedure can inform the selection of implementation and mechanical sizing, using the design of the considered compact simulator as an example. Additionally, by introducing another parallel kinematic structure with 6 degrees of freedom, several potential applications of the aforementioned tools (workspace, single DOF logarithmic graph) were presented.

Bibliography

[1] K.-M. Lee and D.K. Shah. Kinematic analysis of a three-degrees-of-freedom in-parallel actuated manipulator. *IEEE Journal on Robotics and Automation*, 4(3):354–360, 1988.

[2] K.-M. Lee and D.K. Shah. Dynamic analysis of a three-degrees-of-freedom in-parallel actuated manipulator. *IEEE Journal on Robotics and Automation*, 4(3):361–367, 1988.

[3] Clement Gosselin and Jorge Angeles. The optimum kinematic design of a planar three-degree-of-freedom parallel manipulator. 1988.

[4] Min Ki Lee and Kun Woo Park. Kinematic and dynamic analysis of a double parallel manipulator for enlarging workspace and avoiding singularities. *IEEE Transactions on Robotics and Automation*, 15(6):1024–1034, 1999.

[5] Meng Li, Tian Huang, Jiangping Mei, Xueman Zhao, Derek G Chetwynd, and S Jack Hu. Dynamic formulation and performance comparison of the 3-dof modules of two reconfigurable pkm—the tricept and the trivariant. 2005.

[6] V.E. Gough. *Contribution to discussion of papers on research in automobile stability, control and tyre performance.* 1956.

[7] Sunjoo Kan Advani. *The kinematic design of flight simulator motion bases*. 1998.

[8] Staffan Nordmark. *Driving simulators, trends and experiences: Paper prepared for RTS'94" Driving Simulation" Conference, January 11-14, 1994, Palais des Congres Porte Maillot, Paris, France*. Statens Väg-och Transportforskningsinstitut., VTI särtryck 204, 1994.

[9] T Fortmüller. The development of driving simulation at ifas. *Soesterberg, The Netherlands*, 2007.

[10] Johannes Drosdol, Wilfried Käding, and Ferdinand Panik. The daimler-benz driving simulator new technologies demand new instruments. *Vehicle System Dynamics*, 15(sup1):44–57, 1986.

[11] S Hahn and E Kalb. The daimler-benz driving simulator set-up and results of first experiments. In *19. ANNUAL SUMMER COMPUTER SIMULATION CONFERENCE, 1987,*, pages 993–998, 1987.

[12] J Greenberg, R Curry, M Blommer, K Kozak, B Artz, L Cathey, and B Kao. The validity of last-second braking and steering judgments in advanced driving simulators. *DSC 2006 Europe*, 2006.

[13] Alexander Huesmann and Josef Nauderer. Applications to driving simulation and their requirements to the tool. In *Motion Simulation Conference, Braunschweig*, 2007.

[14] Nico A Kaptein, Jan Theeuwes, and Richard Van Der Horst. Driving simulator validity: Some considerations. *Transportation research record*, 1550(1):30–36, 1996.

[15] H Jamson. Driving me round the bend–behavioural studies using the new university of leeds driving simulator, aus der reihe" motion simulator conference. *Braunschweig, Germany (September 2007).*

[16] Lloyd D Reid and Meyer A Nahon. Flight simulation motion-base drive algorithms: part 1. developing and testing equations. *UTIAS Report, No. 296*, 1985.

[17] Peter Kopacek. A mechatronics management laboratory. In P. Kopacek, editor, *Improving Stability in Developing Nations through Automation 2006*, IPV–IFAC Proceedings Volume, pages 37–41. Elsevier, Oxford, 2006.

[18] Kenneth A Pasch and WP Seering. On the drive systems for high-performance machines. 1984.

[19] Herman J Van de Straete, Pascal Degezelle, Joris De Schutter, and Ronnie JM Belmans. Servo motor selection criterion for mechatronic applications. *IEEE/ASME Transactions on mechatronics*, 3(1):43–50, 1998.

[20] Herman J Van de Straete, Joris De Schutter, and Ronnie Belmans. An efficient procedure for checking performance limits in servo drive selection and optimization. *IEEE/ASME transactions on mechatronics*, 4(4):378–386, 1999.

[21] Giancarlo Cusimano. Generalization of a method for the selection of drive systems and transmissions under dynamic loads. *Mechanism and machine theory*, 40(5):530–558, 2005.

[22] Giancarlo Cusimano. A procedure for a suitable selection of laws of motion and electric drive systems under inertial loads. *Mechanism and machine theory*, 38(6):519–533, 2003.

[23] Giancarlo Cusimano. Optimization of the choice of the system electric drive-device—transmission for mechatronic applications. *Mechanism and Machine Theory*, 42(1):48–65, 2007.

[24] Jean-Pierre Merlet. Jacobian, manipulability, condition number and accuracy of parallel robots. pages 175–184, 10 2005.

[25] R Cravel. Delta, a fast robot with parallel geometry. *Proc. ISIR, 1988*, 91, 1988.

[26] Xin-Jun Liu and Jinsong Wang. Hana: a novel spatial parallel manipulator with one rotational and two translational degrees of freedom. *Robotica*, 23:257–270, 03 2005.

[27] Woon-Sung Lee, Jung-Ha Kim, and Jun-Hee Cho. A driving simulator as a virtual reality tool. In *Proceedings. 1998 IEEE International Conference on Robotics and Automation (Cat. No. 98CH36146)*, volume 1, pages 71–76. IEEE, 1998.

[28] Kuiper J Bertollini G, Johnston C and Kukula J et al. A driving simulator as a virtual reality tool. *SAE 1994 Transactions: Journal of Passenger Cars-V103-6*, 1994.

[29] Jacob A Houck, Robert J Telban, and Frank M Cardullo. Motion cueing algorithm development: Human-centered linear and nonlinear approaches. Technical report, 2005.

[30] Hermes Giberti, Simone Cinquemani, and Giovanni Legnani. A practical approach to the selection of the motor-reducer unit in electric drive systems. *Mechanics based design of structures and machines*, 39(3):303–319, 2011.

[31] Mark EJ Newman and Gerard T Barkema. *Monte Carlo methods in statistical physics*. Clarendon Press, 1999.

[32] Rodney FW Coates, Gareth J Janacek, and Kenneth V Lever. Monte carlo simulation and random number generation. *IEEE Journal on Selected Areas in Communications*, 6(1):58–66, 1988.

[33] Winston Haynes. *Probability Distributions*, pages 1752–1754. Springer New York, New York, NY, 2013.