université
PARIS-SACLAY

Human Interaction in a large workspace parallel robot platform with a virtual environment

Thèse de doctorat de l'Université Paris-Saclay

École doctorale n° 580, sciences et technologies de l'information et de la communication (STIC)
Spécialité de doctorat: Robotique
Unité de recherche: Université Paris-Saclay, Univ Evry, IBISC, 91020, Evry-Courcouronnes, France.
Référent: Faculté des sciences d'Orsay,

Thèse présentée et soutenue à ....., le .... 202X, par

Taha HOUDA

Composition du jury:

Prénom Nom Professeur des universités, Affiliation	Président/e
Yacine AMIRAT Professeur des universités, Université Paris-Est Créteil	Rapporteur
Faiz BEN AMAR Professeur des universités, Sorbonne Université	Rapporteur
Fethi BEN OUEZDOU Professeur des universités, Université de Versailles Saint-Quentin-en-Yvelines	Examinateur
Adriana TAPUS Professeur des universités, Institut Polytechnique de Paris	Examinatrice
Ali AMOURI Enseignant chercheur, Université Evry Val d'Essonne	Invité
Lotfi BEJI Professeur des universités, Université Evry Val d'Essonne	Codirecteur
Malik MALLEM Professeur des universités, Université Evry Val d'Essonne	Directeur

Thèse de doctorat

NNT: 2020UPASA000

université PARIS-SACLAY

ÉCOLE DOCTORALE
Sciences et technologies de l'information et de la communication (STIC)

Titre: Interaction humaine dans une plateforme robotique parallèle de grand espace de travail avec un environnement virtuel.

Mots clés: Simulateur de mouvement, Robot hybride (série, parallèle), Mécatronique, Modélisation, Identification, Algorithmes de Restitution Inertielle (ARI), PSO (particle swarm optimization), Réalité virtuelle, Personnes à mobilité réduite, Interaction entre l'homme et le robot.

L'objectif de la thèse porte sur la définition, la mise œuvre et l'évaluation d'un algorithme de restitution de mouvement prenant en compte les contraintes de perception du système vestibulaire chez l'humain et les contraintes liées à la physique du simulateur de mouvement utilisé. Ce dernier est constitué par une plateforme robotique série-parallèle à 8 degrés de liberté entièrement conçue dans le laboratoire et destinée principalement à l'assistance aux personnes présentant un handicap moteur. Cette restitution sensorielle nécessite des travaux de recherches pluridisciplinaires en robotique et en réalité virtuelle. Aussi, une formalisation de la modélisation dynamique, basée sur l'état de l'art, a été adaptée et les paramètres dynamiques optimisés et identifiés à la plateforme de mouvement à 8 degrés de libertés. Plusieurs méthodes de génération de mouvement, gérant la redondance de la plateforme, ont été étudiées mises en oeuvre et comparées. La méthode basée sur l'optimisation par essaims particulaires (PSO) plus performante a été retenue. Cet algorithme est par la suite utilisé pour optimiser les paramètres du contrôleur de la plateforme par mode glissant. Le simulateur a été utilisé pour une application de ski en réalité virtuelle reproduisant la station de Combloux en Haute-Savoie dédiée aux personnes handicapées. Les résultats de simulation montrent un très bon suivi des consignes et une bonne réduction des oscillations. Ces travaux seront poursuivis par l'utilisation d'interfaces multi sensorielles de réalité virtuelle d'assistance à l'humain.

Title: Human Interaction in a large workspace parallel robot platform with a virtual environment

Keywords: Motion simulator, Hybrid robot (series, parallel), Mechatronics, Modeling, Identification, Motion Cueing Algorithm (MCA), PSO (particle swarm optimization), Virtual reality, People with reduced mobility, Human-robot interaction.

The thesis objective relates to the definition, the implementation and the evaluation of a Motion Cueing Algorithm taking into account the perceptual constraints of the vestibular system in humans and the constraints related to the movement physics of the used simulator. The latter consists of a series-parallel robotic platform with 8 degrees of freedom, entirely designed in the laboratory and intended primarily to assist people with motor disabilities. This sensory restitution requires multidisciplinary research work in robotics and virtual reality. Moreover, a formalization of dynamic modeling, based on the state of the art, was adapted and the dynamic parameters optimized and identified for the 8 degrees of freedom motion platform. Several methods of trajectory generation, exploitation of the platform redundancy, have been studied, implemented, and compared. The most efficient particle swarm optimization (PSO) method was chosen. This algorithm is then used to optimize the parameters of the platform controller in sliding mode. The simulator was used for a virtual reality ski application reproducing the Combloux resort in Haute-Savoie dedicated to disabled people. The simulation results show a very good trajectory tracking

behavior and a good reduction in terms of oscillations. This work will be continued through the use of multi-sensory human-assisted virtual reality interfaces.

Université Paris-Saclay
Espace Technologique / Immeuble Discovery
Route de l'Orme aux Merisiers RD 128 / 91190 Saint-Aubin, France

Acknowledgement

I wish to express my deepest gratitude to my thesis director, Pr. Malik MALLEM, for the patient guidance, encouragement, and advice he gave me both professionally and personally. I have been extremely lucky to have a thesis director who cared so much about my work, and who responded to my questions and queries so promptly.

I wish to express my sincere appreciation to my supervisor, Pr. Lotfi BEJI, who has the substance of a genius: he convincingly guided and encouraged me to be professional and do the right thing even when the road got tough.

I would like to express my sincere gratitude and my warmest thanks to Pr. Ali AMOURI for his availability, his encouragement, and all the constructive criticisms he has given me. Without his persistent help, the goal of this project would not have been realized.

I would like also to thank all the jury members for taking the trouble to evaluate my thesis. I am very honored. I would like to especially thank Mr. Yacine AMIRAT and Mr. Faiz BEN AMAR who have very willingly accepted to be rapporteurs.

I close with a big thank you to my brothers and parents to whom I dedicate my thesis.

Contents

List of Figures

List of Tables

List of symbols and abbreviations

${}^{R_b}A_{R_p}$ - Rotation matrix from R_b to R_p
O_b - Mass center of the lower platform
O_p - Mass center of the upper platfrom
B_1 - Origin of the lower platform
P_1 - Origin of the upper platform
$q_a = [q_{31}, \ldots, q_{36}]^T$ - (6×1) vector of the motorized joint variables of the platform
$q_t = [q_{10}, q_{20}]^T$ - (2×1) vector of the motorized joint variables of the base
$\dot{q}_t$- vector of the joint base velocities.
$\ddot{q}_t$- vector of the base joint accelerations.
$\dot{q}_k = [\dot{q}_{1k}, \dot{q}_{2k}, \dot{q}_{3k}]^T$- (3×1) vector of the joint velocities of leg_k.
$\ddot{q}_k = [\ddot{q}_{1k}, \ddot{q}_{2k}, \ddot{q}_{3k}]^T$- (3×1) vector of the joint acceleration of leg_k.
a_{3k} - Unit vector along the Z_3k axis
${}^{R_0}L_k$ - Represents the position vector P_1P_i referred to frame R_0
${}^{R_0}\hat{L}_k$ - Denotes the (3×3) skew symmetric matrix associated with the vector ${}^{R_0}L_k$
${}^{R_0}v_u$ - Linear velocity of the origin of the platform P_1
${}^{R_0}w_u$ - Angular velocity of the upper platform
${}^{R_0}\dot{v}_u$ - Linear acceleration of the origin of the platform P_1
${}^{R_0}\dot{w}_u$ - Angular acceleration of the platform
${}^{R_0}x_2 = [{}^0v_u, {}^0w_u]^T$- (6×1) Spatial velocity vector.
${}^{R_0}\dot{x}_2 = [{}^0\dot{v}_u, {}^0\dot{w}_u]^T$- (6×1) Spatial acceleration vector.
${}^{R_0}v_{pk}$ - Linear velocity of point P_k of the platform
${}^{R_0}\dot{v}_{pk}$ - Linear acceleration of point P_k of the platform
$\Gamma = [\Gamma_{31}, \ldots, \Gamma_{36}]^T$ - (6×1) Vector of the prismatic joint forces.
$\Gamma_k = [\Gamma_{1k}, \Gamma_{2k}, \Gamma_{3k}]^T$ - (3×1) Vector of the torques/forces of leg_k.
${}^{R_b}J_{R_p}$ - Jacobian matrix of Stewart platform
${}^{R_b}J_{R_{3k}}$- Jacobian matrix of leg_k
${}^{R_0}\mathbb{F}_{R_p}$ - Total external forces and moments on the platform about the origin P_1
${}^{R_0}g$ - acceleration of gravity
I_3 - (6×1) identity matrix
M_p - (6×1) mass of the platform
${}^{R_0}I_{R_p}$ - (3×3) inertia tensor of the platform with respect to frame R_0
${}^{R_p}I_{R_p}$ - (3×3) inertia tensor of the platform with respect to frame R_p

Chapter 1

General Introduction

1.1 Research context

The idea of a motion simulator is not new today. For a long time, humans have tried to reproduce different real application scenarios. Historically, the motion simulators were designed for military purposes such as flight simulators. These simulators were used to increase the reactivity and the skills of the army in order to avoid the life risks as well as to avoid spending unnecessary money for a real model or real training. After a while, this technology is increasingly used in the automotive industry motivated by the imperative to study human behavior in different driving situations. In recent years, 3D interactions in mixed reality environments and their integration in industry have taken a major boom. The combination of mechanical systems and virtual tools is then a key issue for designing and validating this type of modern simulator. Today, these simulators are more and more used in medical and rehabilitation applications. A mobility simulator proves that it is essential as an effective tool, allowing the training of people suffering from various joint diseases while preserving a clinical safety environment.

A simulator is a physical and/or computer device whose purpose is to reproduce as faithfully as possible the input-output behavior of a (real) reference system. Defining a driving simulator, for example, is like characterizing one's reference system. The reference system for driving simulators, for example, considers that the actions of the driver on the pedals and the steering wheel (inputs) are associated with the psychological perception of changes in the driving environment (output). We cite the flight and driving simulators which have been able to contribute a lot of possibilities and knowledge to the aeronautic and automotive fields. They are particularly useful to study the implementation effect of active safety technologies, driver assistance, and warning devices. Furthermore, they are useful to conduct human factors research such as the driver's reaction to road conditions, fatigue, aging, drugs and alcohol, as well as safety, comfort, and ergonomic studies. As a result, the number of driving simulators, for example, has risen significantly over the last few years. Early simulators have mostly been limited to a hexapod system Reymond, (2000) [1], Grant, (2001) [2], whereas newer driving simulators are increasingly equipped with an additional rail, such as the Renault Driving Simulator in Technocentre Renault, France, (2001) [3], Daimler-Benz Driving Simulator from Daimler AG in Sindelfingen, Germany Giesen, (2011) [4], the National Advanced Driving Simulator (NADS) in IOWA, USA Heydinger, (2002) [5], the Toyota Driving Simulator in Susono, Japan TOYOTA, (2019) [6].

A great improvement on the new simulators is a so-called XY-table, which can perform larger translational movements in the longitudinal and lateral dimensions. The hexapod is mounted on top of this table and therefore the human operator is under both movements. Such simulators have eight degrees of freedom (8 DoF) in total. The idea is to provide a more realistic experience so that the simulator user cannot distinguish the real situation (real ski) from that simulated on simulator Miunske, (2019) [7].
In addition, the other potential use of these devices is in the medical fields due to the large number of people around the world who have lost their mobility, which represents approximately 15 % of the world population WHO, (2019) [8]. A significant percentage of the population, around 10% WHO, (2019) [8], use wheelchairs. It has been proven that this type of device ensures psychological, muscular and social benefits to people with severe disabilities Ku, (2002) [9].

In this perspective, this research work will be focusing on developing a low-cost simulator called XY - 6 DoF used for open applications and it is mainly intended for training schools and university, gaming, and rehabilitation purposes. The design of a hybrid mechatronic platform intented to be used for skiing simulation is a real multidisciplinary challenge. Academic institutions have found in this problematic a true axis of research and applications. The study of motion simulators requires several skills ranging from mechatronics to neuroscience, through robotics and control theory.

1.2 Research objectives and thesis contributions

The ski-motion simulator generally consists of four main parts: virtual reality scene, trajectory generation (MCA- Motion Cueing Algorithm), robotic structure and human user.

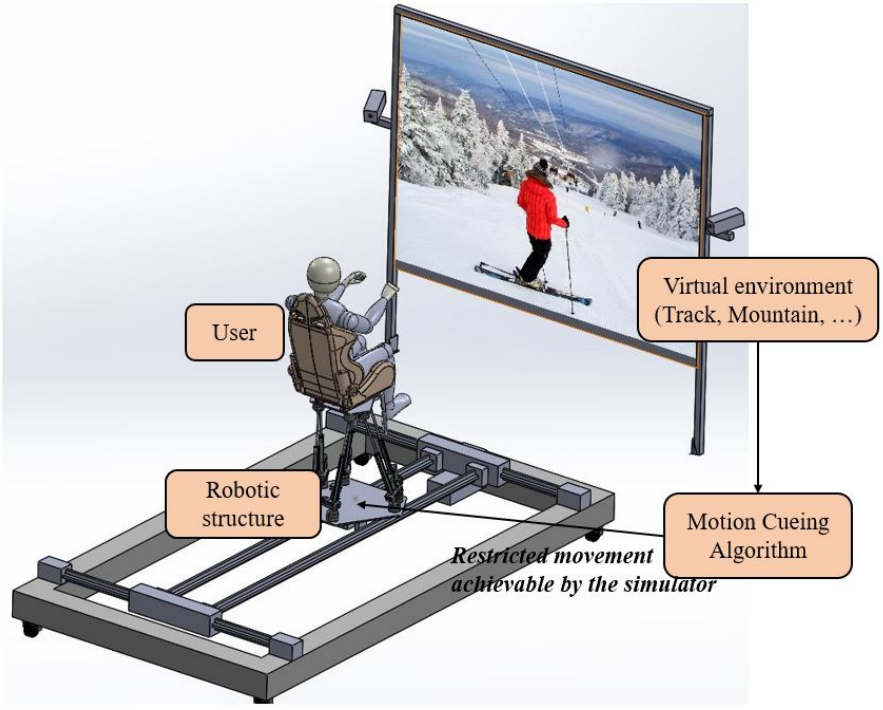


Figure 1.1: General ski simulator componants

Firstly, the virtual reality scene consists of a real virtualized ski scenario. The

VR scene objective is to deceive the human brain through the user's eyes. Secondly, the trajectory generation or MCA which will transform the real trajectories of a real skier into restricted trajectories achievable by the simulator. This transformation must not only satisfy a physical validity (respect for the physical limitations of the simulator movement) but also a perceptual validity (the sensations of movement induced by the simulator must verify the notions of consistency and proximity with those produced in the real case). Thirdly, the user will be attached to the robotic structure. The robotic structure should follow the desired achievable trajectory coming from MCAs.

The thesis objective is to study the production of a ski simulator dedicated to the disabled person. The research objective is to increase the human immersion feeling when using the ski simulator and eliminate what's called "False motion cues" or "simulator sickness". Improvements and contributions on each part constituting the ski simulator are addressed in this thesis from VR to robotic through MCAs. This thesis takes special care of the fact that this simulator should be used for disabled people.

This thesis mainly focuses on three research axes as follows:

1. **Robotic modeling research axis:**

   - **Problematic:** The geometric, kinematic, and dynamic modeling of a robot must be as realistic as possible by studying the robot parts forces/ moments interaction Khalil, (2002) [32], Merlet, (2000) [48]. These models then depend on the dynamic parameters of the system which can be a very large number Khalil, (2010) [40]. The computation and processing time of dynamic models increases with the number of dynamic parameters Long, (2014) [46], which increases what is called the simulator delay. In addition, a friction model should be established to complete the analytical (mathematical) modeling in order to develop the most appropriate control Mata, (2008) [49], Yao, (2018) [60]. Robotic structure can produce a singularity configuration which can be dangerous and even destructive Damien, (2016) [69], Shai, (2009) [85].

     These research subjects are largely discussed in the literature for parallel and series robot architecture, while for hybrid architectures (series and parallel) these questions are topical questions that require additional improvements Ouarda, (2010) [44].

   - **Contribution:** In this thesis, we present an appropriate geometric, kinematic and dynamic modeling of a robot with hybrid architecture (series and parallel). Followed by a validation of the dynamic analytical model accuracy using multi-body modeling software. Then, we propose three methods to optimize the dynamic parameters of the hybrid robot as well as we experimentally identify the optimized inertia and friction parameters [HOUDA1, HOUDA2] . The singularity loci are detailed. we present also the exploitation of the XY-6 DoF (our ski simulator) redundancy to avoid these singularities.

2. **Robotic control research axis:**

   - **Problematic:** Human beings feel movement through their ears, more precisely it is within the vestibular system where the otoliths capture translational acceleration and the semicircular canals capture angular velocity Britannica, (2008) [97], Fernandez, (1976) [98]. However, the

vestibule organ is unable to detect certain movements if the excitation movement is below the perception thresholds. These imperfections are the basis of the trajectory generation algorithms or the motion cueing algorithms (MCAs) used in the driving and flight simulation Shmidt, (1970) [115], Telban, (2005b) [132], Casas, (2018) [127], Miunske, (2019) [7]. The vestibular system will be tricked by the robotic platform. We found two things in the litterature. First, the algorithms are generally presented without great mathematical rigor both in terms of formalization and calculations (cleanliness of the filters used, justification of the choice of parameters, etc.). Second, a lack of transparency in publications in this area: absence of the used filters parameters numerical values, but also of scale factors and saturators. Therefore, these algorithms and their efficiency are still open questions and require further improvement.
In addition, our robotic simulator (XY-6 DoF) has 8 joints, thus it is redundant with respect to the required tasks Lukic, (2020) [144]. Moreover, the achievable simulator trajectory generated by the MCAs must be tracked by the real simulator. It is, therefore, necessary to develop an appropriate control law for trajectory tracking Ma, (2017) [156]. Many methods of redundancy resolution and trajectory tracking control have been developed, but they remain an open question and need to be improved. Furthermore, these methods are specific to each application.

- **Contribution:** In this thesis, we describe in detail how humans perceive movement. We develop a linear model for the vestibular system. We expand and improve existing MCAs to increase the algorithms fidelity and performance as well as to be suitable with sliding application. Our approach is original because it incorporates human perception in the definition of MCAs filters and allow the exploitation of the robot redundancy.
  In addition, we establish a new redundancy exploitation method appropriate to the MCAs and the simulator characteristics. Moreover, we develop a sliding mode trajectory tracking control with optimized parameters. The sliding mode parameters are optimized using the particle swarm optimization method (PSO). Our control approach is optimal to ensure better tracking of the desired trajectories.

3. **Virtual reality scene research axis:**

   - **Problematic:** The existing motion simulators use an oculus helmet or a screen with a 360° panoramic view of the external environment (mountain and snow for ski simulator) to show the virtual scenario Ikeda, (2018) [165], Nozawa, (2019a) [166], Oboe, (2014) [167]. The human immersion effect strongly depends on the resolution, realism and fluidity of the virtual scene. Many VR ski scenarios have been built but still need to be improved in terms of realsim and resolutions Wu, (2019) [168], Hu, (2019) [169]. The main goal is to build the most realistic customized virtual scene in order to deceive the human brain and increase the immersive feeling. In recent years, Handisports have evolved due to the increasing number of humans with poor or disabled mobility. Especially for the handicapped skier Bhambhani, (2012) [170] and for education purposes Caracas, (2019) [171]. For our application, we have to adapt all virtual scenario components so that they are suitable for the disabled person.

- **Contribution:** In this thesis, firstly we virtualized a real ski environment of a real station captured from google maps. This real station is the Combloux resort in Haute-Savoie (the northern part of the French Alps) dedicated to disabled people. Our virtualized environment generates a scene that is more realistic compared to literature. Secondly, we designed a real ski chassis used in real life and the most suitable for disabled person (Dual-ski Nozawa, (2019b) [172]). The virtual scene is customized to meet our specifications. The virtual scene has also been optimized to respect the physical limitation of the robotic platform. The deception of the human brain through their eyes will, therefore, be ensured by the virtual scene [HOUDA3].

1.3 Thesis organization

The thesis plan is organized as follows:

- **In chapter 2**: we offer a general framework description of driving, flight and ski simulation. Moreover, we present a classification of the existing simulator architecture with a historical thread from basic to advanced motion simulator. The attractive areas of simulator application are also presented.

- **In chapter 3**: we develop dynamic models of the IBISC ski simulator (XY-6 DoF). These models will be used for the implementation of our simulator in simulation environment, which means a digital simulation software such as ADAMS, Simmechanics and Simulink. The use of these simulation software will make it possible to test the control algorithms before their implementation on the real simulator. They will also be used for the development of a robust trajectory tracking control. In addition, we conduct a full analytical study of the XY-6 DoF singulatrity locus. Moreover, in order to reduce the modeling processing time and to adjust the control law based, we detail the process of optimization and identification of the entire robotic platform parameters.

- **In chapter 4**: we study how humans perceive movement. We emphasize the vestibular system, for which we are developing linear models. We also synthesize the different approaches of Motion Cueing Algorithms used in the literature. These approaches are extended and improved to give four modified MCAs dedicated for sliding application. These modified algorithms allow exploiting the redundancy of the XY-6 DoF simulator in order to ensure continuity of movement sensation.

- **In Chapter 5**: the trajectory generator or MCAs being calibrated by the methods of the previous chapter, we develop a redundancy resolution method as well as robust trajectory tracking control. The new redundancy resolution method is about to make the simulator non-redundant with respect to the task by finding the appropriate motion distribution of the XY table and the hexapod. The trajectory tracking control consists of using a sliding mode approach with optimized parameters. The sliding mode parameters are optimized using particle swarm optimization (PSO) method to ensure better tracking of the desired trajectories.

- **In chapter 6**: we present in detail the electro-mechanical components and softwares of XY-6 DoF simulator. Since the safety is a primary issue in the field of the Human-Robot interaction, we also present a description of the robot safety environment definition methods. In addition, we describe the construction of a virtual scene that looks like a real ski station in order to increase the human sensation and immersion feeling in a mixed reality environment. The scene is built suitable for the disabled person.
- **In Chapter 7**: we present the general conclusion and perspective.

  At the end we present the bibliography.

Chapter 2

Simulator architectures and main applications

2.1 Introduction

Driving and flight simulators are the most common and widely used simulators. It must be said that this kind of simulators has seduced the industrial and the academic communities due to its many qualities and properties. Sliding simulator for people with poor or reduced mobility is a new field of motion simulator compared to flight and driving simulator Ku, (2002) [9] where we will benefit from research and industry work applied in these type of simulators. The motion simulators (driving simulator as an example) qualities are generally:

1. **Training:** The simulator is a suitable tool for training both beginners and experienced candidates. In the field of aviation, it is sufficient to teach the basic rules to new pilots ("check-list", take-off and landing). In the automobile field, a good behavior does not amount to the good respect of the road rules. Indeed, land vehicles operate in a very constrained and dynamic environment. Urban traffic creates very varied driving situations where sometimes the surprise effect is dominant. Therefore, the goal is to teach drivers the right reflexes and concepts that allow them to avoid accidental driving situations.

2. **Industrial Prototyping:** Driving simulation is becoming more and more an integral part of automotive applications. One reason is the rapid evolution of virtualization technologies and the use of simulation as a tool for competitiveness in vehicle development. This use has matured thanks to the possibility of carrying out representative experiments in the driving conditions studied, and also because of the strengthening of the extreme competition in the development profitability. Thus, a whole discipline was created, called HIL ("Hardware In the Loop"), designed to develop advanced techniques that facilitate the interfacing of vehicle devices on a simulator that interacts with a human operator.

3. **Research:** The design of a computer and mechanical platform dedicated to driving simulation is a real multidisciplinary challenge. Academic institutions have found in this problematic a true axis of research and application. As we will see later in this document, a motion cueing platform requires several skills ranging from mechatronics to neuroscience, through robotics and control theory.

4. **Behavioral study:** The training and awareness of drivers to the various dangers they face requires a good apprehension of human behavior. It is interesting to study the impact of any sensory or decisional dysfunction on driving. As a result, several organizations public and hospitals are interested in driving psychology. Driving simulators provide them with an experimentation tool to recreate specific scenarios, all in a safe environment. Ongoing research, for example, deals with behavior as well as issues related to the deterioration of driving or comfort conditions (drowsiness, drug or narcotic use, disability, etc.).

5. **Requirements for safety and reproducibility of scenarios:** Why simulate a driving situation if it is possible to conduct circuit tests with real vehicles? Real experiments require the use of specialized drivers (pilots or stuntmen), especially to perform dangerous maneuvers. In addition, the results obtained are often biased for safety reasons, as well as to limit the material damage. Similarly, for the psychophysical validation of a given study, it is preferable to cover a large population with different behaviors and experiences, hence the use of simulation that provides a safe and reliable environment for the proper conduct of experiments. Moreover, with the IT platform, the duplicability of the conditions, the environment and the driving scenario is possible, and therefore can draw objective and subjective unbiased conclusions.

6. **Training for critical and rare driving events:** Even experienced drivers have relatively little exposure to critical events, e.g. tire blowouts, animals or pedestrians crossing the road. Simulator training allows learners to practice and to develop greater competence in a wider variety of situations in a shorter period of time than would be possible using traditional driver training. Simulators also facilitate recurrent training in critical situations, e.g. commercial pilots repeat the correct responses to critical events in flight simulators every six months.

Therefore, in this chapter, we describe in details the main components constituting a driving, sliding and rehabilitation simulator. Then, followed by the classification and the evolution of the robotic architectures used over time. In the end, a bibliographic review on driving, sliding and rehabilitation simulator is presented.

2.2 Simulator components

The design of a simulator is a real multidisciplinary challenge, as each simulator is a prototype in itself and there is no standard construction. Its success can only be achieved with collaboration between different skills, from the designer to the end-user. Let’s take an example of skiing application, a skiing simulator can be subdivided into several subsystems that must operate synchronously to create a skiing environment close to the real skiing motion and dynamics (Figure 2.1). It is important that these subsystems be synchronized with a minimum of delay between the virtual scene and the reaction of the simulator Allen, (1999) [10]. Otherwise, the illusion can be altered and lead, among other things, to a difficulty in controlling the simulator.

During the platform design phase, attention should be paid to all subsystems with a central question: "what should we duplicate ? " According to the tests

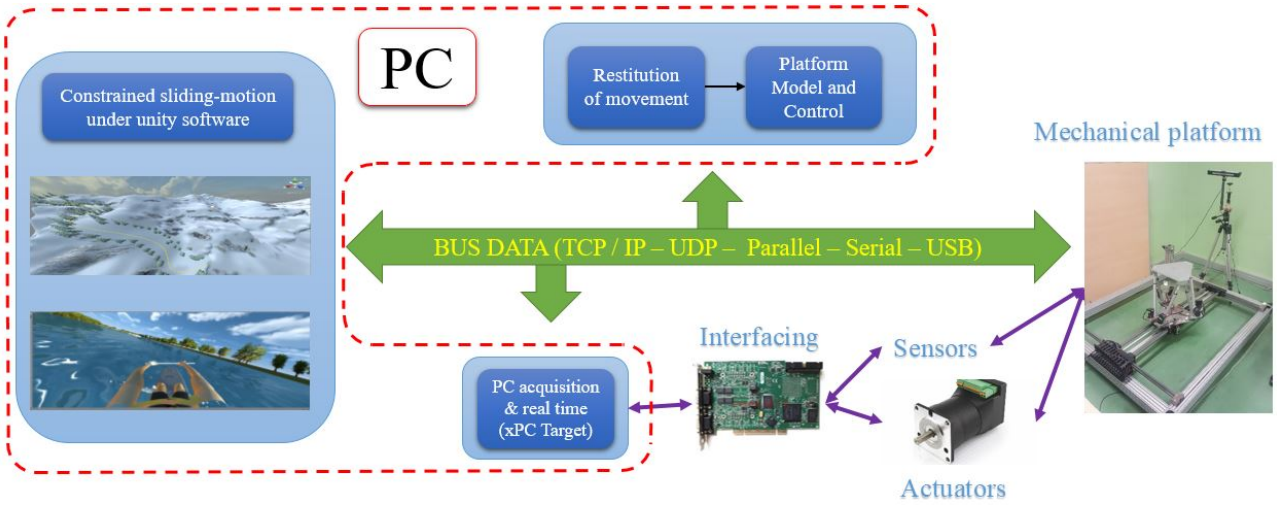


Figure 2.1: Simplified architecture of a simulator

envisaged and the application for which the simulator is intended. For example, to ski with only a right and left movement scenario, a 1 DoF robot is enough. For other situations such as climbing, descending with rotations, a more sophisticated mobile platform may be necessary.
The components of the ski simulators are:

1. **Visual System:** This system consists of a 3D graphic generator and a set of visual reproduction devices. Generally, the images are projected on one or more flat or curved screens, in order to ensure frontal, lateral or even rear vision if necessary. In fact, skiing is a mainly visual task and it is obvious that

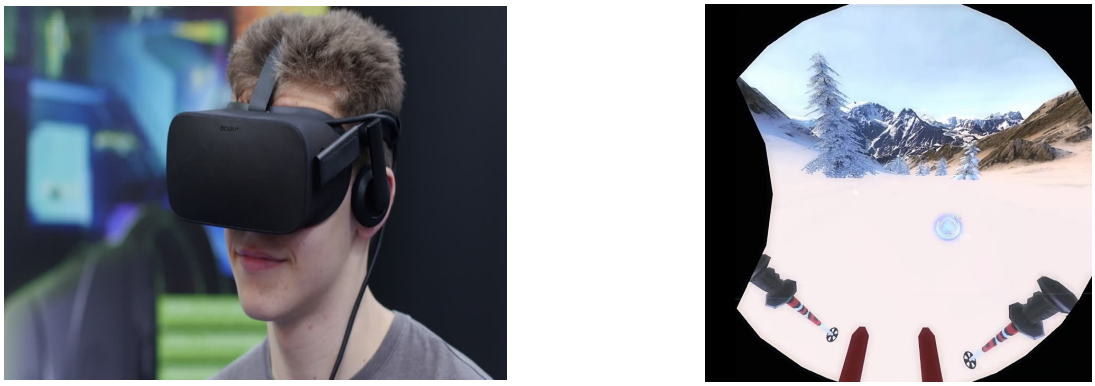


Figure 2.2: Immersive mode: (a) HMD Helmet, (b) Skier scenario view

this information must be extremely neat. Indeed, the quality of an image is measured by its energy properties (brightness, contrast, resolution, color) and its spatial characteristics (fields of vision, depth) and temporal (transport times and refresh rate). As a result, produce a visual scene with great realism depends on the efforts made to satisfy these factors.

In recent decades, another immersion device has been heavily used in virtual reality applications. This is Head Mounted Display. This technique was rarely used in simulators for several reasons, especially because these systems create eye problems Mon-Williams, (1993) [11], fatigue and discomfort due to poor resolution at the beginning (high cost for high resolution headsets). However, because of the quick development of these systems, nowadays, they are available at high resolution and at a reasonable cost. In addition, the virtual system must reproduce the inside of the skiing cabin

and sometimes parts of the driver's body (Figure 2.2). Thus, some brands of such devices include sensors to determine the position and orientation of the driver's head in space, which is a surplus not to be neglected.

2. **Sound system:** Often considered as a secondary priority, the sound system optimizes the realism of the skiing simulation. The main characteristics of this element are the number, the quality and the location of the speakers, and also the diversity of the sounds produced.

3. **Traffic and scenario:** The interactivity of the visual scene is an important element for maximum realism of the driving simulation. The different visual objects (other skiers, trees, etc.) must be endowed with a certain intelligence so that the simulated skiing situations are sufficiently "realistic" and accepted as such by the skier.

4. **Virtual Dynamics of the skier:** The movement of the ski simulator platform is defined from a computer model describing the dynamics of the virtual skier.

5. **Motion cueing :** The system of motion cueing task is to reproduce part of the inertial forces present in a real skiing situation in order to reach a sufficient level of simulation.

   The big challenge is to complete the movements of the platform in the simulator workspace. Geometric and kinematic constraints must be respected by using control algorithms commonly known as the Motion Cueing Algorithm (MCA). These algorithms are based on a simple frequency separation of the various accelerations to be restored by means of two strategies. The first uses the longitudinal movement of the simulator where the transient acceleration is restored by a simple linear motion. The second is to tilt the platform cabin to exploit a component of the gravity vector to reproduce the continuous component of acceleration. A detailed description of the MCA structure and objective is presented in Chapter 4.

6. **Geometric, kinematic and low-level control:** This module is an intermediate step between the generation of setpoint trajectories by the motion cueing algorithms and the platform actuation. According to the platform mechanical architecture, it is necessary to transform the desired trajectories generated in a defined coordinate space to joint coordinate space (actuator space). For example, in the case of parallel platforms, it is necessary to take into account the geometric and the inverse kinematics as well as the couplings between the different axes in the development of a given control strategy.

   The low-level control has for task to ensure the good reference trajectories tracking. However, the primary goal of the skiing simulation is to create a good skiing illusion, thus the developed controller should not be too greedy in terms of computing time to not induce further delay. Simple strategies to compensate for inertial delays have shown their effectiveness. Today, power servo controllers have a high degree of intelligence based on axis-based servo control with multiple nested control loops (position, speed, and torque) for improved tracking and accuracy.

7. **Mechanical platform and actuation system:** In order to achieve the expected performance when drawing up the specifications, it is necessary

to choose the platform architecture (number of DoF) and the corresponding drive system. Advances in industrial computing and power electronics combined have favored the use of electrical actuators at the expense of hydraulic actuators. Indeed, the hydraulic actuators have a great force characteristic, the energy density is almost 100 times greater than that of an ideal electric machine. But this factor alone does not dictate the solution choice, the flexibility of electrical energy use, speed and high speeds offer a price/quality ratio are very in favor of electric actuators. Hydraulic actuators have strong non-linearities marked by hysteresis. This fact is important and can be seen in precision and installation performance.

8. **Measuring devices:** These are the different sensors used for skiing experiments. For example, to measure the concentration, sensors are used at the eyes level. The emotion indicators are calculated from sensors of the respiratory rhythm, the blood pressure, the psychogalvanic reflex.

9. **Communication, acquisition and real-time:** Synchronization, of all the modules described previously, is a key element of a successful development. Delay issues result in a loss of simulator controllability or poor motion record. The computer science offers strategies more and more robust and easy to implement. For the data acquisition from sensors, several solutions are possible such as those based on industrial microcontroller cards, FPGA or DSP (Digital Signal Processor) capable of computing complex mathematical models in "real-time". The communication between computers and sensors is generally done either by analog channels (DAQ, etc.) or via digital fieldbuses with a very advanced task manager (CAN, Interbus, Profibus, etc.). Some simulators benefit from advances in optical technology that offers buses at very high speeds. Finally, between the dynamic model of the virtual skier and the visual rendering system, a simple network communication in TCP / IP or UDP will be enough.

2.3 Classification of simulator architectures

2.3.1 Fixed base platforms

Fixed base platforms consist of an instrumented cabin where the sensation of movement is induced only by the vection caused by visual feedback. These simulators have no mechanical movement, therefore the inertial and dynamic effects are absents. A system of auditory reproduction and haptic modalities (force feedback, vibration, etc.) can be added to create a sufficiently immersive driving environment. Typically, this kind of simulator is used for studies on the physical and mental aspect of driving or aspects related to visual perception such as drivers' behavioral reactions to unexpected events.

2.3.2 Moving base parallel platforms

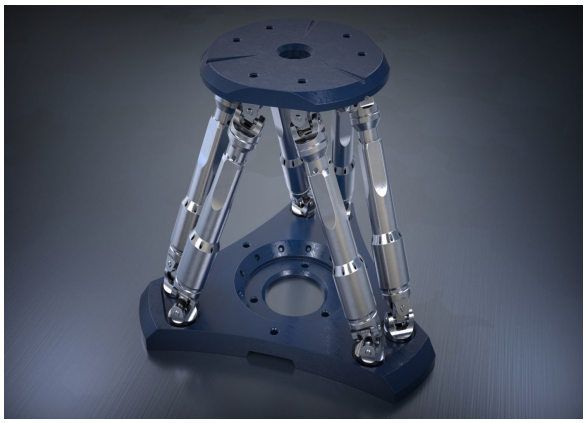
These platforms consist of closed kinematics chain whose terminal body carrying the cabin is supported by several actuators, several actuation type is used such as electric, hydraulic or pneumatic. The most common architecture are the hexapod often referred to by Gough-Stewart platform and the tripod knowns as

(a) Flight Simulator - USA [12]

(b) Mercedes Simulator - Germany [13]

delta robot. Delta robot is more intended to be used as manipulator in industrial application, however, the Gough-Stewart platform is largely intended for simulation applications.

(a) Detla robot [14]



(b) Gough-stewart platform [15]

The parallel platform has several advantages over the serial platform:

1. High accuracy, the movements are less sensitive to measurement errors due to the parallel distribution of the geometry.
2. High payload to weight ratio, possibility of loading very important loads.
3. High speeds operations.

However, some major disadvantages of this type of structure are:

1. Small workspace compared to that offered by the series structure.
2. A complex mechanical design requiring a delicate bonding technology.
3. Singularities issues which requires a specific attention.
4. The Highly nonlinear behavior and dynamic coupling of dynamic parameters make the control implementation more difficult.

Nevertheless, this mechanical architecture has been established as a reference for the mechanisms of motion cueing. It was adopted as a standard of design for flight simulators, driving simulators and often used for sensory and joints rehabilitation. In the following, we present some example of these simulators.

Renault Simulator

Renault, a French manufacturer that has recognized the importance of driving simulation in the vehicle development and prototyping cycle, has several prototypes. One of its simulators, specially designed for ergonomics and human factors studies, is a 6 DoF Gough-Stewart type platform, which offers a working space of $\pm$ 22 cm in displacement and $\pm$ 15 $^\circ$ in rotation, and accelerations and maximum speeds limited to $\pm$ 0.5 g and $\pm$ 0.4 $m.s^{-1}$ in translational motion respectively, and $\pm$ 300 $^\circ.s^{-2}$ and $\pm$ 30 $^\circ.s^{-1}$ in angular motion. In addition, a real Renault Clio cab is mounted on the mobile platform, whose main controls (steering wheel, brake pedal, clutch and accelerator) have a haptic feedback.

Figure 2.5: Renault Simulator - France [1]

The simulation software is based on the SCANeR-II platform (2005) [16], a real-time and multi-tasking application. The distribution between the different modules is done by a UDP communication protocol between the server and the different client processes. The motion cueing algorithms (washout, force feedback) are implemented on the same machine as that of the vehicle dynamic model, through a shared memory. The virtual vehicle model is calculated by the MADA software, derived from the ARHMM model, based on a multi-body approach, making it possible to calculate 250 internal variables in real-time at a frequency of 100 Hz.

VIRTTEX Simulator

The VIRTTEX simulator (VIRtual Test Track EXperiment) is located at the Ford Research Laboratory in Dearborn, USA Grant, (2001) [2]. VIRTTEX is a high-level driving simulator built since 2000. The movement system consists of a large 6 DoF robot, supporting a 1360 kg dome and offering a wide translational displacement of 1.6 m and a response time of 15 ms. The simulator bandwidth is greater than 13 Hz with 80 ms transport time Artz et al. (2001) [17]. VIRTTEX has been developed for three research areas focusing on road safety, vehicle dynamics and driving behavior.

Figure 2.6: VIRTTEX Simulator - USA

2.3.3 Cable driven moving platforms

This cable-driven parallel robot has been built by German-based Max Planck Institute for biological cybernetics (MPI) and fraunhofer IPA. The motion of the cable-driven simulator is controlled by eight unsupported steel cables that are attached to the winches.

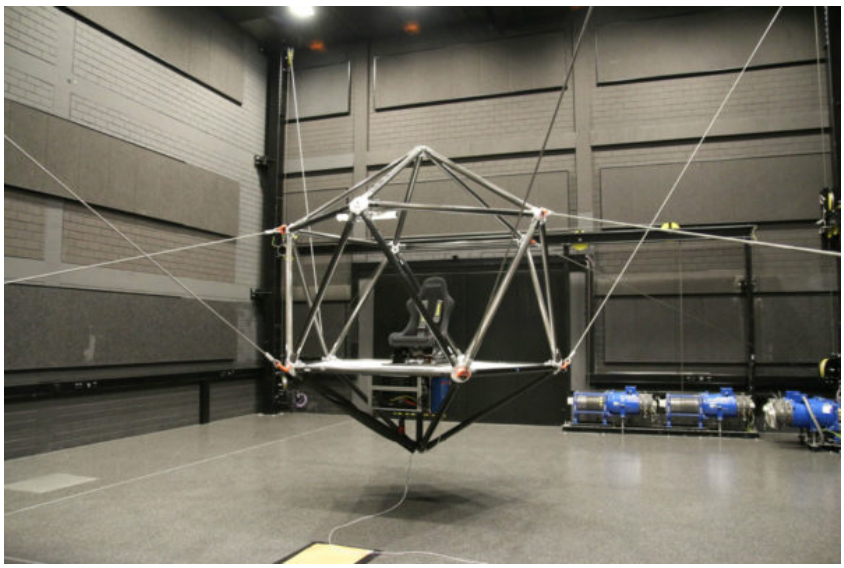


Figure 2.7: Cable robot Simulator - Germany [18]

The cabin can accelerate by 1.5 times gravitational acceleration along programmable paths inside a $5 \times 8 \times 5\ m^3$ workspace through a $348\ KW$ drive power. The simulator can be adapted to different cabins to be used for a different range of scenarios through reattaching the cables under an hour.
The cabin, entirely made of carbon fiber tubes, has a frame that maximizes the cabin's usable volume with a diameter of $260\ cm$ for projection surfaces and cockpit instrumentation, allowing it to be used for high quality projections along with realistic operator interfaces. Furthermore, the $80\ kg$ frame can accelerate at a high speed while withstanding high forces since, in operation, the cables can pull of the outer structures with a load up to 1.5 tons. These features of the cabin starting from its large workspace to its dynamic capabilities make the simulator

suitable for a wide scope of VR applications such as driving and flight simulation as well as investigation of basic human perception processes.

2.3.4 Hybrid platforms

It is a question of combining the advantages of the series and the parallel architectures. Using various technical tricks, the hybrid structures present complex, open and closed kinematic chains. However, the solution more often is to bring a parallel platform on a XY table provide a large workspace with a powerful dynamic. We present:

NADS Simulator

The simulator consists of a dome mounted on a mobile platform 6 DoF which mounted on XY sliders forming a table of $400\ m^2$. It has a vibratory table that can reproduce vibrations of $0.5\ cm$ and up to $20\ Hz$.

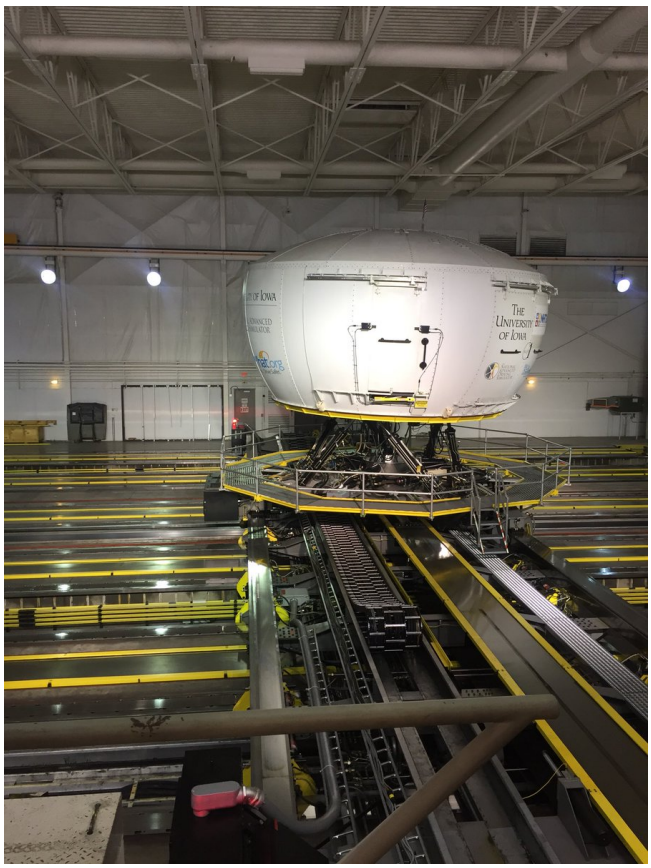


Figure 2.8: NADS Simulator - USA

For the vehicle dynamic model, NADS uses a multibody modeling tool similar to ADAMS called NADSdyna. It consists of three toolbox that define the bodies, articulations and force elements from which it is possible to build different vehicle modules, such as the suspension, the actuator and the gearbox. From these modules, validated separately, the equations of motion are created and then transformed into a set of differential equations that can be solved with appropriate numerical techniques. NADS teams conduct human factors research Chen et al. (2001) [19] such as the driver's reaction to road conditions, fatigue, aging, drugs and alcohol, as well as safety, comfort and ergonomic studies.

Daimler-Chrysler Simulator

The simulator consists of a dome mounted on a 6 DoF mobile platform offering a maximum translational acceleration of $1.2\ g$ and a bandwidth of $3\ Hz$ Käding and Hoffmeyer (1995a) [20]. In addition, the mobile platform can also move laterally

Figure 2.9: Daimler-Chrysler Simulator - Germany

on a rail via a hydraulic cylinder, allowing maximum lateral movement of $3.8\ m$ (Figure 2.9). The maximum lateral acceleration is about $0.7\ g$ with a bandwidth of $5\ Hz$.

The virtual vehicle dynamic model, named CaSimIR, was developed by Daimler-Chrysler. It is divided into two modules: a vehicle configuration file (masses, suspension and motor characteristics) and a module grouping elementary structural models (motor, tire/roadway, suspension and steering). Ca-SimIR allows to simulate a dynamic model from 18 to 37 DoF, and able to include suspensions non-linear properties and steering system flexibilities. Several experiments have been carried out with different driver assistance systems (power steering, active suspension) in which their function, efficiency and acceptance, have been studied.

ULTIMATE Simulator

A hybrid structure driving simulator was built as part of a European project (Eureka) piloted by Renault at the Technical Simulation Center (Figure 2.10). It is a 6 DoF type Gough-Stewart platform with a base that can move on XY rails with a maximum longitudinal and lateral displacement of $6\ m \times 6\ m$ Dagdelen et al. (2004) [21].

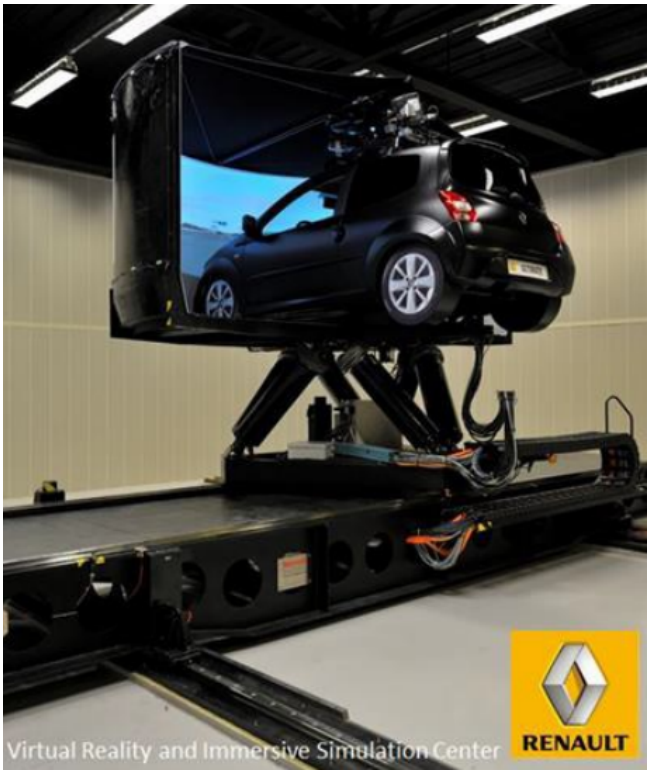


Figure 2.10: Renault Simulator - France

Toyota Simulator

With the same architecture as that of NADS, the Toyota simulator is a hybrid platform composed of two parts. On the one hand, a hexapod supporting a 7.1 m diameter dome that houses a real-world version of a motor vehicle and lined inside a screen with a 360 vue panoramic view of the outdoor environment. On the other, an XY table which allows a longitudinal displacement of 35 m and a lateral displacement of 20 m.

Figure 2.11: Toyota Simulator - Japan

Completed in September 2008, the simulator is mainly aimed at: Driving characteristics analysis. This is to detect unusual behaviors of driving generated by lack of reflex (drowsiness), weakening of sensory faculties (drunkenness), etc.

and The implementation of active safety technologies, driver assistance and warning devices, and Intelligent Transportation Systems (ITS).

Mercedes Simulator

The goal of this simulator was to create the best simulator in the automotive industry. At the end of 2010, Mercedes-Benz inaugurated the world's most advanced mobile-based driving simulator in Sindelfingen at a cost of 220 million dollars.

Figure 2.12: Mercedes Simulator - Sindelfingen, Germany

This simulator consists of the dome attached to the hydraulically-controlled Stewart platform and is sourced from Moog (a US-based designer and manufacturer of motion and fluid control systems and control systems for aerospace applications, defense, industry and medical) mounted on a twelve-meter-long rail for transverse or longitudinal movements. With this driving simulator, highly dynamic driving proceedings, such as changing lane, can be simulated in a realistic manner, enabling in-depth research into driver and vehicle behavior on the road.

The acceleration, braking and driving are recorded more than a thousand times per second and they are sent to the central computer, to which a team monitors the information and communicates with the driver, which calculates the behavior of the driver and sends it in order of movement to the simulator. This system allows the Mercedes Benz to study driver behavior in different driving scenarios, the driver assistance system, as well as the noise and vibration of the car, to ensure a smooth and comfortable ride.

BMW Simulator

The company announced an investment of 100 Million Euros in a new Driving Simulation Center located in Munich at the FIZ Research and Innovation Centre in the north of the city, the new facility is said to be ready by **2020** and will allow researchers to 'bring the road inside' to check out different driving situations up close, in a rather controlled environment.

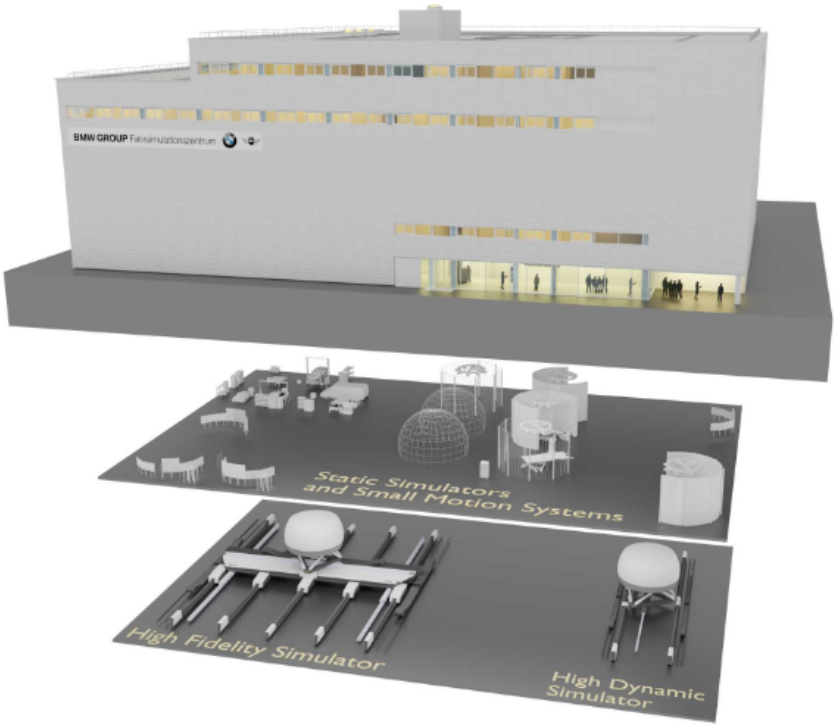


Figure 2.13: BMW new driving simulator center - Munich, Germany

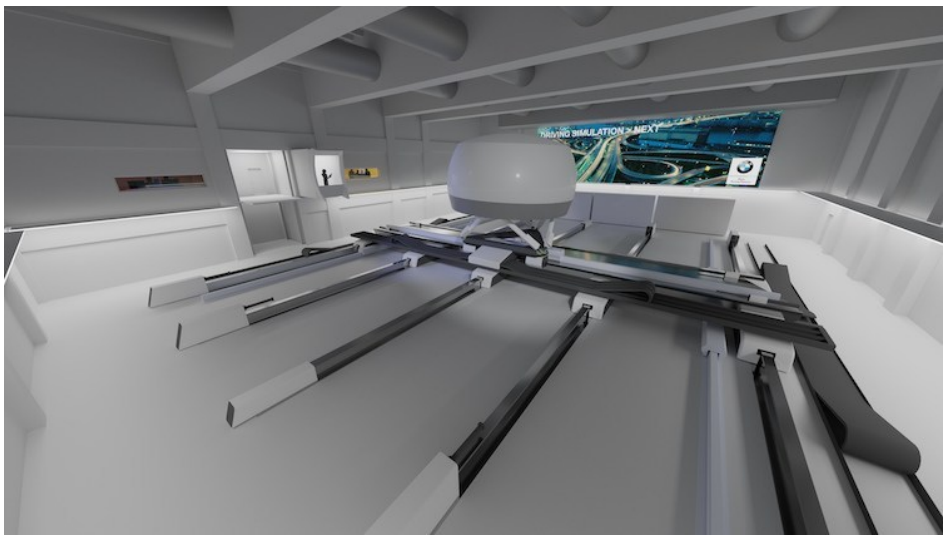


Figure 2.14: BMW High Fidelity Simulator - Munich, Germany

The systems are fitted in a vehicle mock-up attached to a platform inside the dome of the driving simulator. Mounted on an electromechanical hexapod system, the dome can be moved both longitudinally and transversely by an electric drive while also being turned in almost $400\ m^2$ motion area. In order to give the drivers a realistic visual experience of the simulated driving situation, the dome housing the mock-up is equipped with a projection screen. It is used to test new systems and functions by replicating highly dynamic evading proceedings, full braking and hard acceleration. An extremely detailed rendering of real-world driving characteristics is provided by the second unit, the high-fidelity simulator: braking and accelerating while cornering, driving in roundabouts and rapid series of multiple turn-off manoeuvres can be reproduced with high precision.

Low Cost Simulators

The motion simulators use is becoming more and more widespread and adopted by various public and private institutions. However, if the cost of airplane training and passenger safety allow investment in high-cost simulators, there is no reason why it should not be the same to offer a tool for training and psychophysical studies. In addition, the cost and complexity of a simulator does not reflect, in any case, its fidelity to restore all the movements. For this reason, it was interesting to offer targeted solutions based on low-cost mobile platforms intended mainly for training schools and university, hospitals and other users.

In this context, a mini-simulator called XY - 6 DoF has been fully designed and built within the IBISC laboratory of the Evry-Val d'Essonne University - Paris-Saclay University Amouri, (2016) [22], which is presented as a parallel architecture (type Gough-Stewart) mounted on XY table. The platform allows the accelerations reproduction of approximately $0.5g$ thanks to a longitudinal movement of $1m$ and to an optimized transmission system (pinion-wheel) actuated by a brushless electric motor (Fig. 2.15).

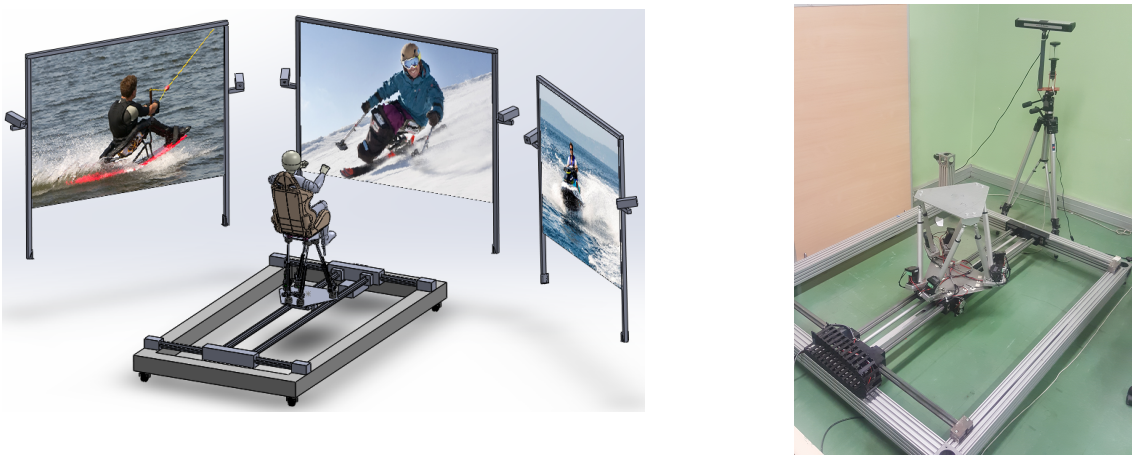


Figure 2.15: Sliding simulator for ski, jet-ski and wake application

2.4 Attractive area of simulators

2.4.1 Industry and training

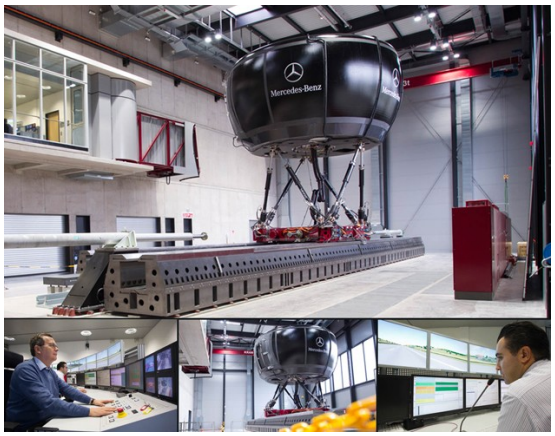
The objectives that give rise to these simulators are mainly military objectives such as flight simulator since World War II and now, such as the F16 and Sukhoi, submarines and tank simulators. These simulators are used to train the army: to increase responsiveness and skills of fighter in the field without spending a lot of money for a real model or a real training.

After a while, this technology is increasingly used in the industry. It is now the automotive industry that takes the torch and leads to a huge improvement, motivated by the imperative need to study human behavior in different situations and at the same time motivated by the improvement of prototypes following several criteria, we cite:

(a) Airbus Simulator

(b) Sukhoi Simulator

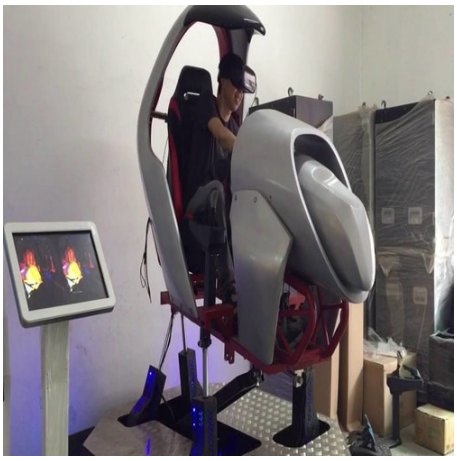
(a) Tank Simulator



(b) Mercedes Simulator

2.4.2 Gaming and commercial

The huge development and the trade growth of the virtual and augmented reality simulator as well as the maturity of the study about the human sensor model. Furthermore, after a certain mature time this technology is transferred to the gaming and commercial purposes. Several types and ideas of gaming simulators have been implemented, we cite:



(a) Car driving Simulator

(b) Motocycle simulator

(a) Flight Simulator

(b) Jetpacks Simulator

(a) Submarine Simulator

(b) Ski Simulator

2.4.3 Medical fields

Seen that the number of handicapped around the world is relevant and the research communities are intended to extract solutions for them, the simulator is currently used for the rehabilitation purposes. These simulators may have the same architecture mentioned above or special configurations.

Motor rehabilitation after a cerebrovascular accident is carried out either by means of specific training tasks or by a training device in the carpet form. In both approaches, the ability to manipulate the environment to reproduce varied walking conditions is limited. It should be noted that a mobility simulator is essential as an effective tool, allowing the training of people affected, on various surfaces while preserving a clinical safety environment. We cite:

Simulator for motor rehabilitation for the ankle

In this framework, a mechanical platform called "Rutgers Mega-Ankle" has been developed, it is presented as a parallel robot type Stewart 6 DoF whose cylinders are actuated by a pneumatic station. The mobility simulator consists of two identical robots attached to the ground (Figure 2.21) next to each other at a distance calculated for a better comfort of rehabilitation Boian et al. (2005) [23].

The feet displacement is simulated by the two platforms coherent movements to reproduce a walk on treadmill. At rest, when both feet touch the virtual floor, the two robots have the role of keeping the subject in a stable position. Then,

Figure 2.21: Rehabilitation Simulator

when one foot gets up, the corresponding platform performs a switching of its control mode: control in position towards control in force, while following the foot movement to compensate the gravity force. In the next step, when the back foot is raised, the front robot returns to the neutral position and so on Boian et al. (2004)) [24].

Wheelchair simulator for the disabled person

The first kind of this simulator was not intended for people with reduced mobility rehabilitation, but rather as a tool for prevention and awareness of having a motor disability. This simulator is like a manual wheelchair interfaced to a virtual navigation environment.

At the same time, engineers in this field began to see the opportunities offered by the virtual reality tools use and simulation for the assessment and training of people with disabilities. The motivations for these projects are different, but strongly related to the limitations of actual clinical rehabilitation. Above all, the user safety is paramount, especially for those who drive a wheelchair for the first time and for people with severe disabilities. Also, this tool makes it possible to establish and test detailed prescriptions of a large number of equipment and subsequently to propose the chair adapted to a given handicap.

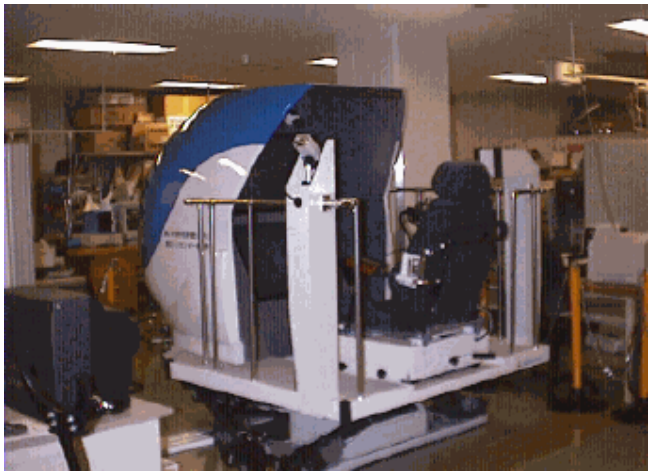


Figure 2.22: Electric wheelchair simulator for the disabled person- Japan

For these reasons, a wheelchair driving simulator was built, to name only one, at the National Rehabilitation Center for Disabled People in Namiki, Japan. The first version was designed by adapting a Stewart 6 DoF platform (Figure 2.22). It simulates motion, electric motor vibration, ground roughness and collisions between the wheelchair and the virtual environment Niniss and Inoue (2006) [25]. The projection of the visual scene is done using two adjacent screens of 29 inches each. The improvements made in the second version mainly concern the visual system, where the two screens have been replaced by a large hemispherical display.

Lokomat Simulator

The virtual reality application in rehabilitation has led to a rapid development of game-like applications. Since the key concept to successful rehabilitation involves the constant task repetition that involves the affected part of the body, such applications motivate patients to practice movements over and over again. Driven gait orthoses provide an excellent tool to use such methods for gait training in a meaningful way.

Figure 2.23: Lokomat Simulator [26]

Many studies have demonstrated the usefulness of repetitive task practice by using robotic-assisted gait training (RAGT) devices, including Lokomat, for the treatment of lower limb paresis. Virtual reality (VR) has proved to be a valuable tool to improve neurorehabilitation training. The aim of our pilot randomized clinical trial was to understand the neurophysiological basis of motor function recovery induced by the association between RAGT (by using Lokomat device) and VR (an animated avatar in a 2D VR) by studying electroencephalographic (EEG) oscillations.

The Lokomat helps patients with a neurological deficit to recover the use of the legs faster in the initial rehabilitation phase. With the neurorehabilitation robot service, they make up to a hundred times more steps than in conventional therapy.

Exoskeleton Simulator

The exoskeleton was only a science fiction for a long time, this technology is mainly used for the purpose of increasing power and rehabilitation.
The presented exoskeleton is able to simulate walking on another planet where the gravity is much greater than that on earth, the user is, therefore, weaker, slower and less coordinated, and everything is adjustable. The system is equipped with a set of earphones, microphones and similar devices for the manufacture of

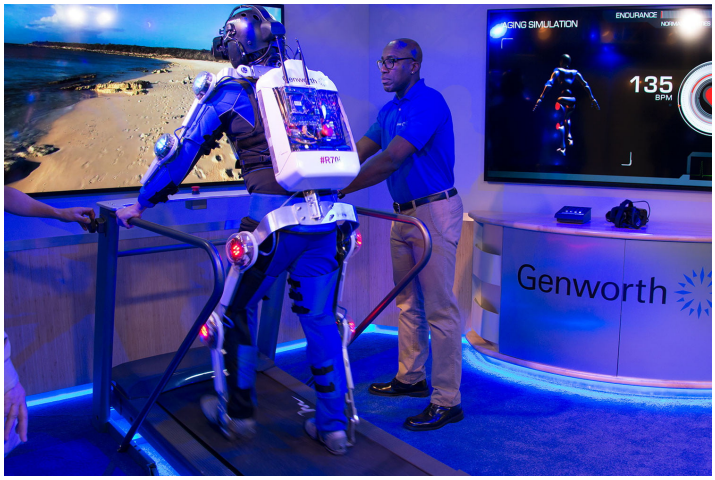


Figure 2.24: Exoskeleton Simulator [27]

augmented reality glasses that transform into a virtual vision. The exoskeleton can be used for rehabilitation reasons for disabled people. However, the lokomat simulator is more used for security reasons.

2.5 Conclusion

In this second chapter, we have presented a state of the art of some existing simulators as well as their application areas. It is obvious that the field of driving and flight simulation is much richer with simulators equipped with the latest technologies. We can therefore see the means and the huge investments, essentially by vehicle manufacturers, in the design of increasingly gigantic platforms. On the other hand, today, there are only a few simulators for ski, jet ski and wake simulations with simplified mechanical architectures, by the opposition, there are some simulators, for rehabilitation purposes, use the hexapod parallel robot (Rutgers Mega-Ankle).

Table Tab. A.1 presents a summary comparative table of the presented simulators. The architecture of the simulators consists of three main types: fixed, mobile stewart and hybrid platform. Tables Tab. A.2 and Tab. A.3 compare the performance of 4 industrial simulators in terms of movement workspace.
The newest and the most advanced architecture is the hybrid ones where the commonly used Gough-Stewart platform is improved by add one or two axis forming a XY-table. The XY table allow the simulator to perform larger translational movements in the longitudinal and lateral dimensions. The hexapod is mounted on top of this table and therefore the human operator is under both movements, which allow to give him a high sensational movement. Such simulators have eight degrees of freedom (8 DoF) in total. From a kinematic point of view, longitudinal movement as many as the transverse movement has been crucial in the ski simulation for creating curves for example. Moreover, the movements about 6 axes are indispensable to be able to educate the human joints in adapted rehabilitation motion. From Table A.1 and Table A.1 (Appendix A), it is clear that the transverse and lateral dynamics is the first concern of motion simulation, and is the major difference of simulators.

In this perspective, we have inspired by the hybrid simulators architecture to build and develop from scratch a low cost simulator called XY - 6 DoF, that respect the same functionality of the presented hybrid simulators with homothetic ratio.

This hybrid platform can be used for open applications, however, it is mainly intended for training schools and university, gaming and rehabilitation purposes. The XY - 6 DoF simulator geometric, kinematic and dynamic modeling will be presented in details in the following chapter.

Chapter 3

XY - 6 DoF Simulator Dynamic Modeling

3.1 Introduction

Current motion simulators most often have a parallel architecture Reymond, (2000) [1], Grant, (2001) [2], which have an inherent advantages with respect to serial architecture in terms of high precision, high payload capacity, high velocity and rigidity Kalani, (2016) [37], Merlet (2000) [48]. However, parallel architecture has limited workspace with respect to their serial counterparts Khalil, (2002) [32]. Researchers and manufacturers have therefore found an interest in developing this type of simulator architecture Käding and Hoffmeyer (1995a) [20], Boian et al. (2004)) [24]. Therefore, the IRA2 team of the IBISC laboratory proposed a new low-cost simulator where it consist of hexapod (parallel architecture) mounted on two series axis (XY table). This architecture has both parallel and series advantages where the hexapod can perform small movements with high precision and the XY table perform the large workspace Amouri, (2016) [22].

Considering that the dynamic performances and the new opportunities introduced by parallel architecture in industrial and research applications, several methods were proposed for their dynamic modeling Khalil, (2004) [33], Lopes, (2009) [36], Kalani, (2016) [37], Merlet (2000) [48]. Among the very few work approached the dynamic modeling of hybrid robots, we can find Freeman, (1988) [41], Sklar, (1988) [42] and Chung (2004) [43]. These methods are difficult to apply for robots having a big number of modules. The methods proposed for the calculation of the inverse and direct dynamic models use recursive Newton–Euler formalism. This technique has been seen as efficient for serial rigid manipulators Khalil, (2004) [33], Ouarda (2010) [44] and for flexible manipulators Briot (2013) [45], Long (2014) [46].

It is well known that the dynamic parameters of parallel architecture play an important role in describing the behavior of a multibody system such as robot manipulators Khalil, (2004) [33]. The precision, performance, stability, and robustness of robots depend mainly on the accuracy of the parameters that describe the dynamic model Khalil and Gautier (2007) [55]. The precise values of dynamic parameters are generally unknown Gautier (1994) [53], Gautier (2008) [54]. Hence, an optimized analytical form of the dynamic model must be developed, and the number of mathematical operations must be reduced. An experimental identifcation of optimized dynamic parameters precise value must be performed Yao (2018) [60], Mirza (2017) [61].

The parallel architecture are known to lead to singular configurations in which the number of degrees of freedom of the mechanism changes instantaneously.

Moreover, the actuator speeds or the actuator forces may become very important which will result in a breakdown of the mechanism. Therefore, it is of primary importance to study the singularities loci in a given workspace to avoid any structural damage and to have better robotic structure control Fichter (1980) [65], Bandyopadhyay (2006) [68], Damien (2016) [69], Jiang (2009) [71], Kotlarski (2012) [72].

In this chapter, first we begin by parametrization of the XY-6 DoF simulator using appropriate notation Khalil, (1986) [31]. Second, we develop the geometric, kinematic and dynamic model of the XY-6 DoF simulator. Third, we verify and validate the analytical dynamic modeling by comparing the behavior with the simulator modeled in dynamic multi-body software. Fourth, we propose several methods to optimize the number of dynamic parametres for the XY-6 DoF, then we identify experimentally the optimized dynamic and friction parameters. Fifth, we present the advantages of the XY-6 DoF workspace over a parallel robot as well as its singularity analysis. At the end, we exploit the XY-6 DoF simulator redundancy to avoid the singularity of the parallel mechanism (hexapod). Note that all electronic, mechanical and technical components of the XY-6 DoF are presented in detail in the Appendix A.

3.2 XY-6 DoF simulator parameterizations

The XY-6 Degree of Freedom (DoF) simulator consists of a hexapod type Gough-Stewart mounted on two prismatic axes forming an XY table. The 6 DoF Gough-Stewart parallel robot is composed of upper and lower platforms connected by six extendable legs. The connection of the legs and the lower and upper platform is ensured by a Universal and a Spherical joints, respectively. The hexapod is carried by the X-axis. The hexapod and the X-axis are carried by the Y-axis. Thus, the X-axis moves only the hexapod and the Y-axis moves the hexapod and the X-axis (Fig. 3.1). Therefore, the XY-6 DoF simulator can be considered as a parallel robot (hexapod) carried on a serial robot (XY table). The length of each leg is actuated using an active prismatic joint. The displacement of the hexapod lower platform in the X and Y axis is ensured by two active prismatic joints.

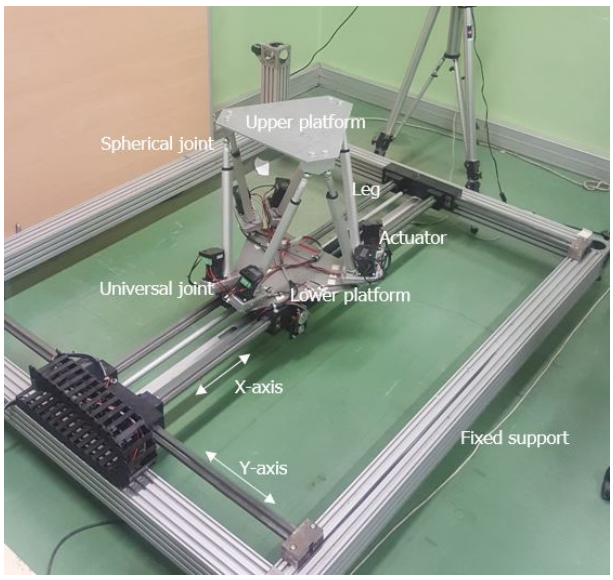


Figure 3.1: Annotation of the XY-6 DoF simulator

In mechanical engineering, the Denavit–Hartenberg parameters are the four parameters associated with a particular convention for attaching reference frames to the links of a spatial kinematic chain, or robot manipulator. Jacques Denavit and Richard Hartenberg introduced this convention in 1955 in order to standardize the coordinate frames for spatial linkages Denavit, (1955) [28], Hartenberg, (1965) [29]. Richard Paul demonstrated its value for the kinematic analysis of robotic systems in 1981 Paul, (1981) [30]. The robot parametrization is a crucial part of reducing complexity for robot modeling, and it is offer the fundamental parameters where all geometric, kinematic and dynamic model will be developed basing on these parameters. A new notations introduced by Khalil and Kleinfinger is described in Khalil, (1986) [31], which is a variation of the Denavit-Hartenberg one. This notation is the most suitable notation to handle complex chains with a tree structure or closed loops. Therefore, the reference frames of the XY- 6 DoF simulator will be defined as recommended in Khalil, (2002) [32].

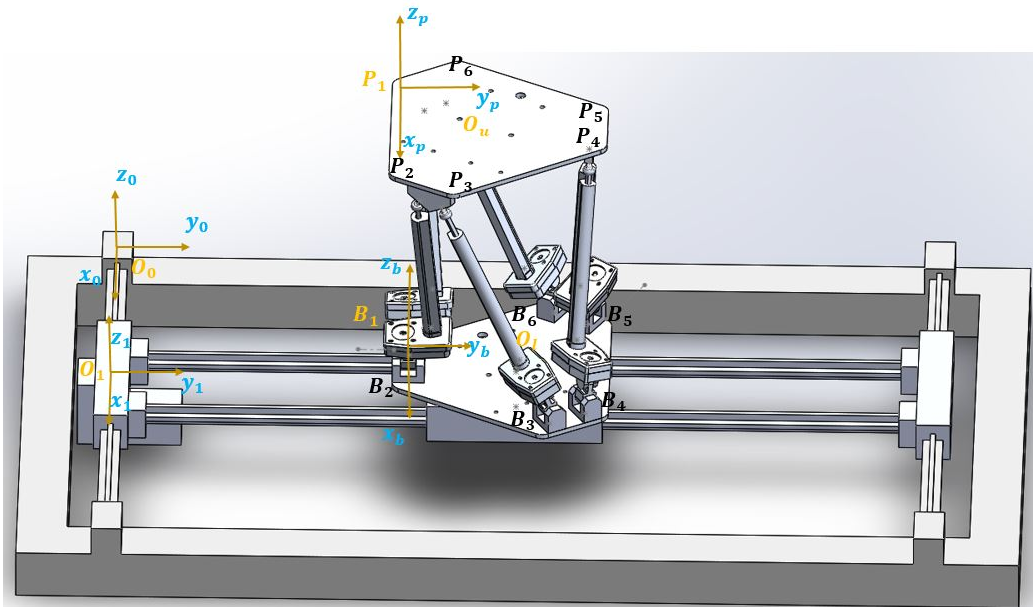


Figure 3.2: 8-DoF mechanical system parametrizations (CAD realization)

Let's define the four reference frames, the earth-fixed frame $R_0(O_0; x_0; y_0; z_0)$, the local frame fixed to the Y-axis $R_1(O_1; x_1; y_1; z_1)$, the local frame attached to the lower platform $R_b(B_1; x_b; y_b; z_b)$ and the local frame attached to the upper platform $R_p(P_1; x_p; y_p; z_p)$. In order to reduce the geometric parameters the hexapod extremities are defined as recommended in Khalil, (2004) [33]. Fig. 3.2 presents the simulator reference frames. The positions of B_i $(i = 1,\ 2,\ ...,\ 6)$ in the lower platform frame R_b are respectively $B_1 = [0,\ 0,\ 0]^T$, $B_2 = [b_1,\ 0,\ 0]^T$, $B_3 = [b_2,\ b_3,\ 0]^T$, $B_4 = [b_4,\ b_5,\ 0]^T$, $B_5 = [-b_6,\ b_5,\ 0]^T$ and $B_6 = [-b_7,\ b_3,\ 0]^T$.
Similarly, the position of P_i $(i = 1,\ 2,\ ...,\ 6)$ in the upper platform frame R_p are respectively $P_1 = [0,\ 0,\ 0]^T$, $P_2 = [p_1,\ 0,\ 0]^T$, $P_3 = [p_2,\ p_3,\ 0]^T$, $P_4 = [p_4,\ p_5,\ 0]^T$, $P_5 = [p_6,\ p_5,\ 0]^T$ and $P_6 = [p_7,\ p_3,\ 0]^T$. The O_0O_1, O_1B_1, b_i and p_i with $(i = 1,\ 2,\ ...,\ 7)$ are measured from the real simulator using measuring tools such as callipers. Fig. 3.3 shows the kinematic chain of the XY-6 DoF simulator.
Let $k = 0$ denotes the XY table and the $k = 1,...,6$ denotes the leg number, let j_k denotes the joint j of the link k. For example, 1_0 is the first joint of the XY table and 2_3 is the second joint of the third leg. q_{jk} denotes the joint j rotational or translational displacement of part k. A frame will be attached to each joint.
These frames are defined by severals geometric parameters given in table 3.1.

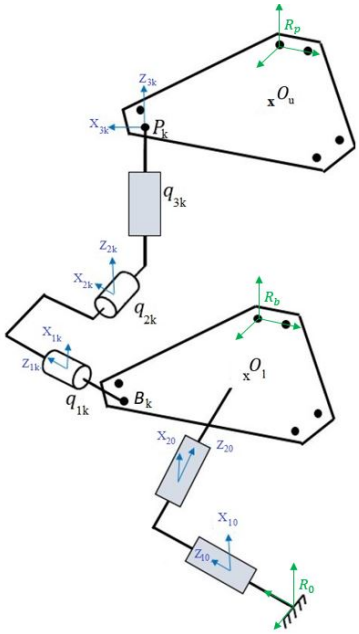


Figure 3.3: Kinematic chain of leg k

j_k	$a(j_k)$	$\mu(j_k)$	$\sigma(j_k)$	$\alpha(j_k)$	$d(j_k)$	$\theta(j_k)$	$r(j_k)$
1_0	0_0	1	1	$+\pi/2$	0	0	q_{10}
2_0	1_0	1	1	$-\pi/2$	0	0	q_{20}
1_k	0_k	0	0	$-\pi/2$	d_{1k}	q_{1k}	0
2_k	1_k	0	0	$\pi/2$	0	q_{2k}	0
3_k	2_k	1	1	$\pi/2$	0	0	q_{3k}

Table 3.1: Khalil and Kleinfinger parameters Khalil, (1986) [31] of the hybrid robot

where $\alpha(j_k), d(j_k), \theta(j_k), r(j_k)$ are used to determine the location of frame R_j with respect to its antecedent $R_{(j-1)_k}$, $a_{(j-1)_k}$ denotes the joint antecedent of j_k, the parameter $\mu(j_k)$ is one for a motorized joint and zero for a passive joint, $\sigma(j_k) = 0$ for a revolute joint, and $\sigma(j_k) = 1$ for a prismatic joint. For more details, we can see Khalil, (1986) [31].

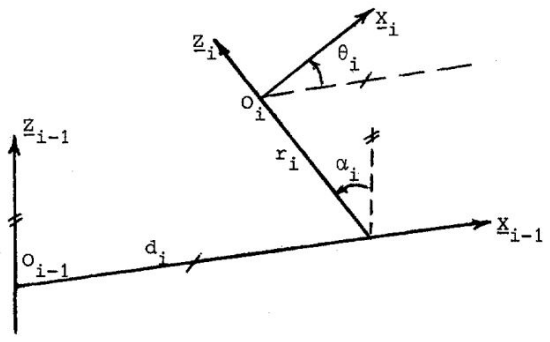


Figure 3.4: notation for a binary link index i from Khalil, (1986) [31]

3.3 Geometric and Kinematic models

The XY-6 DoF configuration can be described by a vector of generalised joint coordinates $q \in \mathbb{R}^8$: $q = [q_{10},\ q_{20},\ q_{3k}]^T$ where $k \in [1,...,6]$. Let us define the task space which is the space of all possible upper platform poses $X \in \mathbb{R}^6$, $X = [x\ y\ z\ \phi\ \theta\ \psi]^T$.

3.3.1 Geometric model

The inverse geometric problem is to find the generalized joint coordinates q correspond to a given desired location of the upper platform X. The inverse geometric can be stated as follows:

$$q = f(X) \tag{3.3.1}$$

The joints space is about $q \in \mathbb{R}^8$ which is bigger than the task space $X \in \mathbb{R}^6$, then for a given task there are endless solutions, thus the XY-6 DoF is a redundant robot. Therefore, to obtain a specific joint configuration q, a redundancy resolution method should be selected. The simplest method is based on the use of the Jacobian matrix pseudo-inverse Whitney, (1969) [34]; this guarantees a least-squares reconstruction of the desired upper velocity. We proposed a new method for redundancy resolution and it is described in details in Chapter 6. For the next work, we will consider that the total desired upper platform trajectory $X(t) \in \mathbb{R}^6$ is composed of XY table $x_1(t) = [x_t(t),\ y_t(t)]^T \in \mathbb{R}^2$ and hexapod $x_2(t) = [x_h(t),\ y_h(t),\ z(t),\ \phi(t),\ \theta(t),\ \psi(t)]^T \in \mathbb{R}^6$ desired trajectories.

For the XY table, Let us define $x_1(t)$ and q_t as $x_1(t) = [x_t(t),\ y_t(t)]^T$ and $q_t = [q_{10},\ q_{20}]^T$, respectively, therefore for a given $x_1(t)$,

$$q_t(t) = x_1(t) \tag{3.3.2}$$

For the hexapod, let us take the leg 2, the active joint can be calculated from the vector relation presented in the following,

$$\overrightarrow{B_2P_2} = \overrightarrow{B_2B_1} + \overrightarrow{B_1O_l} + \overrightarrow{O_lO_u} + \overrightarrow{O_uP_1} + \overrightarrow{P_1P_2} \tag{3.3.3}$$

where O_l, O_u are the lower and upper platform centre of mass. $\overrightarrow{B_2B_1} = [-b_1,\ 0,\ 0]$ and $\overrightarrow{B_1O_l} = [x_{ol},\ y_{ol},\ 0]$ are expressed in R_b where $b_1,\ x_{ol},\ y_{ol}$ are constant and mesured from the real simulator. Similarly, $\overrightarrow{P_1P_2} = [p_1,\ 0,\ 0]$ and $\overrightarrow{O_uP_1} = [-x_{ou},\ -y_{ou},\ 0]$ are expressed in R_p where $p_1,\ x_{ou},\ y_{ou}$ are constant and mesured from the real simulator. $\overrightarrow{O_lO_u} = [x,\ y,\ z]^T$ is the distance between the upper and lower platform center of mass. This vector relation is shown in Fig. 3.5.
To describe the orientation of the upper platform, we use a parametrization by the Euler angles: roll, pitch and yaw (ϕ, θ, ψ). ${}^{R_b}A_{R_p}$ is the rotation matrix between the lower platform R_b and the upper platform R_p frames. Therefore,

$${}^{R_b}(\overrightarrow{P_1P_2}) = {}^{R_b}A_{R_p}\,{}^{R_p}(\overrightarrow{P_1P_2}) \tag{3.3.4}$$

$${}^{R_b}(\overrightarrow{O_uP_1}) = {}^{R_b}A_{R_p}\,{}^{R_p}(\overrightarrow{O_uP_1}) \tag{3.3.5}$$

The vector relation can be, therefore, fully expressed in R_b as:

$$\overrightarrow{B_2P_2} = \overrightarrow{B_2B_1} + \overrightarrow{B_1O_l} + \overrightarrow{O_lO_u} + {}^{R_b}A_{R_p}[{}^{R_p}(\overrightarrow{O_uP_1}) + {}^{R_p}(\overrightarrow{P_1P_2})] \tag{3.3.6}$$

Therefore, for $x_2(t) = [x_h(t),\ y_h(t),\ z(t),\ \phi(t),\ \theta(t),\ \psi(t)]^T$ we can compute the $\overrightarrow{B_2P_2}$

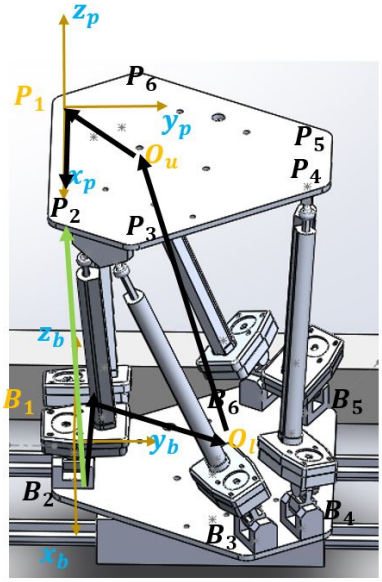


Figure 3.5: vector relation representation

from the previous relation. The leg 2 length q_{32} is the euclidean norm of the B_2P_2 vector.

$$q_{32} = ||\overrightarrow{B_2P_2}|| \tag{3.3.7}$$

This vector relation can be generalized to compute the leg k elongation and can be expressed as follows:

$$\overrightarrow{B_kP_k} = \overrightarrow{B_kB_1} + \overrightarrow{B_1O_l} + \overrightarrow{O_lO_u} + {}^{R_b}A_{R_p}[{}^{R_p}(\overrightarrow{O_uP_1}) + {}^{R_p}(\overrightarrow{P_1P_k})] \tag{3.3.8}$$

and the leg k elongation is equal to:

$$q_{3k} = ||\overrightarrow{B_kP_k}|| \tag{3.3.9}$$

The passive leg k joints rotation which are the universal joints rotation will be required in dynamic modeling, thus the passive joints of leg k will computed in the following. The rotational movement of each P_k, the end point of joint $3k$, is only carried by the universal joints. Thus the rotation matrix between the R_b frame and the R_{3k} frame depend only on the passive joints q_{1k} and q_{2k}. These rotation are along the Y_b and $-X_b$ axis. Therefore, the ${}^{R_b}A_{3k} = R_{-x}(q_{2k}) * A_y(q_{1k})$ with,

$$A_{-x}(q_2) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos(q_{2k}) & sin(q_{2k}) \\ 0 & -sin(q_{2k}) & cos(q_{2k}) \end{bmatrix} and\ A_y(q_{1k}) = \begin{bmatrix} cos(q_{1k}) & 0 & sin(q_{1k}) \\ 0 & 1 & 0 \\ -sin(q_{1k}) & 0 & cos(q_{1k}) \end{bmatrix} \tag{3.3.10}$$

Since the orientation of R_{3k} frame about z is known, which is a_{3k}, the passive joints can be computed by the identifications method.

The direct geometric model defines the upper platform positions X with respect to the reference frame R_b in terms of the joint coordinates q and can be stated as follows:

$$X = g(q) \tag{3.3.11}$$

where, $g = f^{-1}$. For the XY table the displacement in X and Y direction are equal to the sliders (joints) displacement q_{10} and q_{20}, therefore,

$$x_1(t) = q_t(t) \tag{3.3.12}$$

For the hexapod, a numeric calculation using Newton-Raphson method is used to calculate the translational and rotational position of the upper platform. Let us define $x_2(t)$ the translational and rotational displacement of the upper platform and $q_a \in \mathbb{R}^6 = [q_{3k}]^T$ the active hexapod joints where $k = 1,...,6$. This numerical calculation gives an accurate results depending on the tolerance specified. For a certain position x_2 of the upper platform, we will define an error function ξ as Harib, (2003) [35]:

$$\xi(x_2) = q_a(x_2) - q_a^n \tag{3.3.13}$$

where $q_a(x_2)$ is the hexapod active joints calculated from the translational and rotational displacement of the upper platform $x_2(t)$ described above and q_a^n is the hexapod active joints (known). This function $\xi(x_2)$ will vanish when we found the corresponding x_2 which gives $q_a(x_2) = q_a^n$.
A numerical solution using Newton-Raphson method is given by:

$$x_2(i) = x_2(i-1) - \left(\frac{\partial \xi(x_2(i-1))}{\partial x_2}\right)^{-1} \xi(x_2(i-1)) \tag{3.3.14}$$

where $x_2(i)$ is the solution (approximate by the tolerance specified). The relation between upper platform trajectory and the hexapod active trajectories can be expressed as:

$$\frac{d}{dt} q_a = J^{-1} \frac{d}{dt} x_2$$

where J is the jacobian matrix that will be described in the kinematic modeling. By elimination of the ∂t and well grouping the variable we can easily show that

$$\frac{\partial \xi(x_2(i-1))}{\partial x_2} = J(x_2(i-1))$$

Hence, the equation (3.3.14) can be written as:

$$x_2(i) = x_2(i-1) - J(x_2(i-1))(q_a(x_2(i-1)) - q_a^n) \tag{3.3.15}$$

as it is known for the numerical calculation the convergence of the solution is depend on the initial position $x_2(0)$, thus it is (primordial) to choose a initial solution near to the exact solution.

3.3.2 Kinematic model

Inverse kinematic modeling consists of defining the active joint velocities $\dot{q}$ with respect to the translational and rotational speed of the upper platform $\dot{X}$ in the reference R_0.
For the XY table the active joints speed can be expressed as follows:

$$\dot{q}_t = \dot{x}_1(t) \tag{3.3.16}$$

For the hexapod, recall that $q_a = [q_{31}, \ldots, q_{36}]^T$ vector of the hexapod actuated joints (active). For small rotations, the speed of the upper platform is $\dot{x}_2(t)$ which is annotated by $\dot{x}_2 = [v_u \ \omega_u]$ where $\dot{x}_2 \in \mathbb{R}^{(6\times1)}$, $v_u \in \mathbb{R}^{(3\times1)}$ is the speed of the upper platform centre of mass with respect to R_b and $\omega_u \in \mathbb{R}^{(3\times1)}$ is the upper platform rotation speed. The speed of upper platform extremities P_k can be expressed as the summation of the translation speed v_u and the rotation speed w_u cross

product with the ${}^{b}L_k$ where L_k is defined as $L_k = \overrightarrow{P_1 P_k}$. The speed of P_k in R_b is as follows:

$$v_{pk} = v_u + w_u \times L_k \tag{3.3.17}$$

The projection of the $v_{pk} \in \mathbb{R}^{(3\times1)}$ velocity vector on the axis of the leg k gives the elongation speed of the leg k that is expressed in the R_b reference as:

$$\dot{q}_{ak} = v_{pk}^T \ a_{3k} = v_u^T \ a_{3k} + (w_u \times L_k)^T \ a_{3k} \tag{3.3.18}$$

where, $a_{3k} \in \mathbb{R}^{(3\times1)}$ is the leg k unit vector and $\times$ is the cross product. Lets $\hat{L}_k$ denotes the (3×3) skew symmetric matrix associated with the vector L_k . By using the properties of cross product the relation between the leg elongation speed in joint space with respect to the platform cartesian velocities will be:

$$\dot{q}_{ak} = v_{pk}^T \ a_{3k} = v_u^T \ a_{3k} + w_u^T (\hat{L}_k \ a_{3k}) \tag{3.3.19}$$

Denote ${}^{R_b}J_{R_p}$ the Jacobian matrix of hexapod platform which connects the joints speeds with the upper platform speeds, therefore, the actuated joints can be expressed in this relation (known as the inverse kinematic model):

$$\dot{q}_a = {}^{R_b}J_{R_p}^{-1} \ \dot{x}_2 \tag{3.3.20}$$

with,

$${}^{R_b}J_{R_p}^{-1} = \begin{bmatrix} {}^{R_b}a_{31}^T & ({}^{R_b}\hat{L}_1 \, {}^{R_b}a_{31})^T \\ \vdots & \vdots \\ {}^{R_b}a_{36}^T & ({}^{R_b}\hat{L}_6 \, {}^{R_b}a_{36})^T \end{bmatrix} \tag{3.3.21}$$

As mentioned above, the passive leg k joints speed will be required in dynamic modeling. let us introduce, $q_k = [q_{1k}, \ q_{2k}, \ q_{3k}]^T$ with $q_k \in \mathbb{R}^3$ the leg k joint variables, where q_{3k} is the leg k actuated one, $q_{1k}, \ q_{2k}$ are the leg k passive ones. The leg k displacement q_k can be expressed as follows:

$$\dot{q}_k = {}^{R_b}J_{3k}^{-1} \ {}^{R_b}v_{pk} \tag{3.3.22}$$

where J_{3k}^{-1} is the Jacobian matrix of the terminal point of leg k. In order to calculate this jacabian we will present its projection into frame R_{3k} that obtained as:

$${}^{3k}J_{3k} = \begin{bmatrix} 0 & q_{3k} & 0 \\ -q_{3k}\sin q_{2k} & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.3.23}$$

and its inverse is given as:

$${}^{3k}J_{3k}^{-1} = \begin{bmatrix} 0 & -\frac{1}{q_{3k}\sin q_{2k}} & 0 \\ \frac{1}{q_{3k}} & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.3.24}$$

This jacobian is singular when $sin(q_2) = 0$ and/or $q_{3k} = 0$ which physically impossible. Note that: ${}^{R_b}J_{3k}^{-1} = {}^{3k}J_{3k}^{-1} \ {}^{3k}A_b$; ${}^{R_b}J_{3k}^{-T} = {}^{R_b}A_{3k} \, {}^{3k}J_{3k}^{-T}$, where ${}^{R_b}J_{3k}^{-T} = ({}^{R_b}J_{3k}^{-1})^T$.

3.3.3 Second order kinematic model

The acceleration of point P_k is obtained by differentiating the equation (3.3.17) along time that can be expressed as:

$$\dot{v}_{pk} = \dot{v}_u + \dot{w}_u \times L_k + w_u \times (w_u \times L_k) \tag{3.3.25}$$

and then the acceleration of the active joints is calculated by differentiating of equation (3.3.20) along time that can be obtained as:

$$\ddot{q}_a = {}^{R_b}\dot{J}_{R_p}^{-1}\ \dot{x}_2 + {}^{R_b}J_{R_p}^{-1}\ \ddot{x}_2 \tag{3.3.26}$$

The acceleration of passive joints can be obtained by differentiating equation (3.3.22) along time that can be expressed as follows:

$$\ddot{q}_k = {}^{R_b}J_{3k}^{-1}(\dot{v}_{pk} - {}^{R_b}\dot{J}_{3k}\ \dot{q}_k) \tag{3.3.27}$$

Another form of the acceleration matrix can be represented as follows:

$$\ddot{q}_k = {}^{R_b}J_{3k}^{-1}([I_3\ -\hat{L}_k]\ddot{x}_2 + w_u \times (w_u \times L_k - {}^{R_b}\dot{J}_{3k}\ \dot{q}_k)) \tag{3.3.28}$$

3.4 Dynamic models

In this section, we will develop the dynamic modeling of the hexapod type Gough-Stewart alone then followed by the dynamic modeling of the entire simulator architecture. In this section, we develop two dynamic models, the first one is for the hexapod type Gough-Stewart platform and the second described the dynamic modeling of the XY-6 DoF simulator considered as hexapod with moving base.

3.4.1 Dynamic modeling of the Gough-Stewart platform

Inverse dynamic modeling is to defines the forces and/or torques of the active joints based on the position, velocity and acceleration of the upper platform and the body's inertial properties (mass and moment of inertia). Note that the Gough-Stewart platform has been widely studied alone Khalil, (2004) [33], Lopes, (2009) [36], Kalani, (2016) [37]. The Lagrange equations and Newton-Euler equations are the most common used methods to calculate the dynamics models of a given robot. The lagrange equation is given as:

$$\Gamma = \frac{d}{dt}\left[\frac{\partial L}{\partial \dot{q}}\right]^T - \left[\frac{\partial L}{\partial q}\right]^T$$

where Γ is the joint forces or torques, L is the Lagrangian of the robot which defined as the difference between the kinetic energy E and the potential energy U of the system: $L = E - U$ with $E = \frac{1}{2}\dot{q}^T M \dot{q}$.
After developing this equation we obtain another equation with this form:

$$\Gamma = M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q)$$

where $M(q)$ is the inertia matrix of the robot, $C(q,\dot{q})$ is the Coriolis and Centrifuge force/torque and $G(q)$ is the gravity force/torque. we will groupe the $C(q,\dot{q})$ and

$G(q)$ because both are depend on the joint displacements and speeds by $H(q,\dot{q}) = C(q,\dot{q})\dot{q} + G(q)$, therefore,

$$\Gamma = M(q)\ddot{q} + H(q,\dot{q}) \tag{3.4.1}$$

The recursive Newton-Euler algorithms have been shown to be an excellent tool to model a rigid robots rather than the Lagrange equations which need more time to do the same task (bigger time difference for bigger number of degrees of freedom).

The Gough-Stewart platform will be considered as 6 serial robots (6 legs) and one free body which is the upper platform. The dynamic model of the Gough-Stewart parallel platform is generally given as Khalil, (2004) [33] where the dynamic of active joints depends on the upper platform as well as the legs dynamics and it is presented as follows:

$$\Gamma = \underbrace{{}^{R_b}J_{R_p}^{T}\ {}^{R_b}\mathbb{F}_{up}}_{\text{Upper platform dynamic}} + \underbrace{\sum_{k=1}^{6}\left(\frac{\partial \dot{q}_k}{\partial \dot{q}_a}\right)^{T} H_k(q,\dot{q},\ddot{q})}_{\text{Legs dynamic}} \tag{3.4.2}$$

For the upper platform dynamic, the Newton-Euler equations giving the forces and moments of upper platform at the frame R_p origin are given as Khalil, (2007) [38]:

$${}^{R_b}\mathbb{F}_{up} = \underbrace{{}^{R_b}\mathbb{I}_{R_p}\ {}^{R_b}\dot{v}_u}_{\text{Inertial part}} + \underbrace{\begin{bmatrix} {}^{R_b}w_u \times ({}^{R_b}w_u \times {}^{R_b}MS_{R_p}) \\ {}^{R_b}w_u \times ({}^{R_b}I_{R_p}\ {}^{R_b}w_u) \end{bmatrix}}_{\text{Coriolis part}} - \underbrace{\begin{bmatrix} M_p I_3 \\ {}^{R_b}M\hat{S}_{R_p} \end{bmatrix}}_{\text{Gravity part}} {}^{R_b}g \tag{3.4.3}$$

where, the inertia matrix expressed in R_b is as follows: ${}^{R_b}\mathbb{I}_{R_p} = \begin{bmatrix} M_p I_3 & {}^{R_b}M\hat{S}_{R_p} \\ {}^{R_b}M\hat{S}_{R_p} & {}^{R_b}I_{R_p} \end{bmatrix}$;

with, ${}^{R_b}I_{R_p} = {}^{R_b}A_{R_p}\ {}^{R_p}I_{R_p}\ {}^{R_b}A_{R_p}^{T}$ and ${}^{R_p}I_{R_p} = \begin{bmatrix} XX_{R_p} & XY_{R_p} & XZ_{R_p} \\ XY_{R_p} & YY_{R_p} & YZ_{R_p} \\ XZ_{R_p} & YZ_{R_p} & ZZ_{R_p} \end{bmatrix}$;

The dimensions of the upper platform is measured from the real simulator and then sketched in SimMechanics, after we assign the appropriate material to the sketched design. ${}^{R_p}I_{R_p}$ is then the upper platform second moment of inertia expressed in his reference frame R_p and it is computed from the SimMechanics software. The first moment of inertia ${}^{R_b}MS_{R_p}$ can be expressed as ${}^{R_b}MS_p = {}^{R_b}A_{R_p}\ {}^{R_p}MS_{R_p}$ with ${}^{R_p}MS_{R_p} = [MX_{R_p}\ MY_{R_p}\ MZ_{R_p}]^T$ similarly the ${}^{R_p}MS_{R_p}$ is computed from the SimMechanics software.

For the leg dynamic,

$${}^{R_b}\mathbb{F}_{leg} = \sum_{k=1}^{6}\left(\frac{\partial \dot{q}_k}{\partial \dot{q}_a}\right)^{T} H_k(q,\dot{q},\ddot{q}) \tag{3.4.4}$$

where the partial derivative can be expanded as follows:

$$\frac{\partial \dot{q}_k}{\partial \dot{q}_a} = \frac{\partial \dot{q}_k}{\partial v_{pk}}\ \frac{\partial v_{pk}}{\partial \dot{x}_2}\ \frac{\partial \dot{x}_2}{\partial \dot{q}_a} \tag{3.4.5}$$

Therefore, the partial derivative is composed of three jacobian that is obtained as follows Khalil, (2010) [40]:

$$\frac{\partial \dot{q}_k}{\partial \dot{q}_a} = J_{3k}^{-1}\ J_{vk}\ J_{R_p}$$

where, J_{3k} and J_{R_p} are the terminal point of leg k and the kinematic jacobian matrices, respectively, defined above and note that $v_{pk} = J_{vk}\dot{x}_2$ and J_{vk} is depend on the

connection between the leg and the upper platform. The upper platform is connected with the legs through spherical joints where all forces can be transmitted without limitation, therefore, J_{vk} is expressed as follows:

$$J_{vk} = \begin{bmatrix} I_3 \\ {}^{R_b}\hat{L}_k \end{bmatrix} \tag{3.4.6}$$

Denote that $H_k(q, \dot{q}, \ddot{q})$ is the inverse dynamic model of leg_k considering as a serial robot (alone) and different methods can be used to calculate this vector Khalil, (2010) [40]. $H_k(q, \dot{q}, \ddot{q})$ is the inverse dynamic model of leg_k considering as a serial robot (alone). The algorithm consists of two recursive forward and backward calculations. The forward equations Khalil, (2010) [40], from link 1 to link 3, compute the link velocities and accelerations and consequently the dynamic wrench on each link. The backward equations, from link 3 to the lower platform, provide the reaction wrenches on the links and consequently the joint torques. The forward and backward recursive equations are described in the Appendix. For a fixed lower platform, we initialize the calculation by:

$$v_0 = [0\ 0\ 0]^T,\ \omega_0 = [0\ 0\ 0]^T,\ \dot{\omega}_0 = [0\ 0\ 0]^T \text{and}\ \dot{v}_0 = [0\ 0\ -g]^T$$

We can consider the gravity forces automatically in the dynamic equilibrium equations by taking $\dot{v}_0 = [0\ 0\ -g]^T$ (virtually, just to take gravity into consideration on all joints). This algorithm is easy to program numerically or symbolically. The computational cost is linear with the number of degrees of freedom of the robot. To reduce the number of operations of the calculation of this model the optimized inertial parameters can be used instead of the standard inertial parameters detailed in the Optimization and identification chapter 4.
Thus, the forces and moments exerted by the legs are given by:

$${}^{R_b}\mathbb{F}_{leg} = \sum_{k=1}^{6} \left[\begin{bmatrix} I_3 \\ {}^{R_b}\hat{L}_k \end{bmatrix} {}^{R_b}J_{3k}^{-T} H_k(q, \dot{q}, \ddot{q}) \right] \tag{3.4.7}$$

Therefore, the final dynamic model equation can be expressed as follows:

$$\Gamma = {}^{R_b}J_{R_p}^{T} \left({}^{R_b}\mathbb{I}_{R_p}\, {}^{R_b}\dot{v}_u + \begin{bmatrix} {}^{R_b}w_u \times ({}^{R_b}w_u \times {}^{R_b}MS_{R_p}) \\ {}^{R_b}w_u \times ({}^{R_b}I_{R_p}\, {}^{R_b}w_u) \end{bmatrix} - \begin{bmatrix} M_p I_3 \\ {}^{R_b}M\hat{S}_{R_p} \end{bmatrix} {}^{R_b}g + \sum_{k=1}^{6} \left[\begin{bmatrix} I_3 \\ {}^{R_b}\hat{L}_k \end{bmatrix} {}^{R_b}J_{3k}^{-T} H_k(q, \dot{q}, \ddot{q}) \right] \right) \tag{3.4.8}$$

To write the general equation (3.4.8) in compact form as it is written in (3.4.1) an identification of the inertia matrix $M(q)$ and the coriolis and gravity matrix $H(q, \dot{q})$ has been carried out. The inertia matrix $M(q)$ and the coriolis and gravity matrix $H(q, \dot{q})$ can be computed by giving special values of the joint coordinates, velocities and accelertations for both equations (3.4.8) and the equation of lagrange (3.4.1) Khalil, (2002) [32]. It is clear that if:

1. The acceleration $\ddot{q} = 0$ the $H(q, \dot{q}) = \Gamma$
2. $\ddot{q} = n_i,\ \dot{q} = 0,\ g = 0$ and the externales forces and moments are equal to zero the i^{th} column of $M(q)$ is equal to Γ

where, $n_i\ (6 \times 1)$ is the unit vector whose i^{th} element is equal to 1 and all of the other are zeros. By iterating the second condition 6 times and by calculation of the Γ from the equation (3.4.8) respecting the conditions mentionned above we can construct all the Mass $M(q)$ and the coriolis and gravity $H(q, \dot{q})$ matrix. Note that the mass matrix is a symmetric matrix so we can reduce the computational

by 15 item of mass matrix. The compact dynamic equations is then expressed as follows:

$$\Gamma = M(q)\ddot{q} + H(q,\dot{q}) \tag{3.4.9}$$

3.4.2 XY- 6 DoF simulator dynamic modeling

Among the very few work approached the dynamic modeling of hybrid robots, we can find Freeman, (1988) [41], Sklar, (1988) [42] who used the principle of D'Alembert with the equivalent tree structure of the closed structure, and Chung (2004) [43] who calculate at first the local dynamics of each module with respect to the independent joint coordinates and then the dynamics of the hybrid robot is calculated by using the concept of the virtual joints that are attached to the base of each module. These methods are difficult to apply for robots having a big number of modules. The methods proposed for the calculation of the inverse and direct dynamic models use recursive Newton–Euler formalism. This technique has been seen as efficient for serial rigid manipulators Khalil, (2004) [33], Ouarda (2010) [44] and for flexible manipulators Briot (2013) [45], Long (2014) [46].

The structure treated in this section includes a big number of systems such as: cars, mobile robots, mobile manipulators, walking robots, Humanoid robots, eel like robots Khalil, Gallot (2007) [47], snakes like robots, flying robots, spatial vehicle, etc. The difference between all of these systems will be in the calculation of the interaction forces with the environment. In this section, we will present two different ways of expressing the dynamic model of the XY-6 DoF simulator.

A- First method for the XY- 6 DoF dynamic modeling

The XY-6 DoF is considered to be a Stewart platform mounted on a mobile base. In the previous sections the base was fixed thus the acceleration of the base is equal to zero, whereas in the case of a mobile base the acceleration of the base must be determined in both direct and inverse dynamic models. The proposed recursive dynamic models are easy to implement and calculate using numerical calculation. The inverse dynamic model, which is used in general in the control problems, can be used in simulation too when the objective is to study the evolution of the base giving joint positions, velocities and accelerations of the other joints. Denote that the first subsystem is composed of the XY Table (series robot) that carry the second subsystem which is composed of Gough-Stewart Platform (parallel robot).
To study the coupling effect between these two subsystems. A two-additional recursive calculation have been made: one forward and one backward, Firstly a forward recursive calculation starting from the fixed reference to the upper platfrom via the mobile base should be conducted to compute the link velocities and accelerations and consequently the dynamic wrench on each link.
Thus, in this forward recursive equations we calculate:

- the Jacobian matrices J_{3i}^{-1}, J_{vi} and J_p.
- the screw transformation matrix ${}^{j}T_{j-1}$,
- the joint positions, velocities and accelerations of the legs q_i, $\dot{q}_i$,$\ddot{q}_i$ by using the kinematic models of the legs described in Chapter 2.
- the velocity and acceleration vectors ${}^{j}V_j$, ${}^{j}\dot{V}_j$, ${}^{j}\omega_j$ and ${}^{j}\dot{\omega}_j$.

- the total external wrench on the platform F_{R_p}, using the Newton–Euler Equation.

Afterwards a backward recursive equations, from link n to the fixed reference via the mobile base (lower platform), provide the reaction wrenches on the links which allow us to calculate the acceleration of the base platform taking into considerations the effect of the sliders by the first recursive relations and the effect of the Gough-Stewart by the second recursive relations as well as the joints forces/-torques as function of the velocities and external forces/torques.

After having the reactions wrenches, we apply the free body newton equation on the base platform (lower platform), and since the base platform has no rotational movement, the base acceleration of coupled structure can be written as:

$$^{R_b}\dot{v}_b = I_b^{-1}[F_b + M_b g]$$

where, $F_b = F_1 + F_2$, F_b is the external forces applied on the base platform coming from the first subsytem F_1 (series robot) and the second subsytem F_2 (parallel robot).

$$F_1 = M\ ^{R_b}\dot{v}_b \tag{3.4.10}$$

$$F_2 = \sum_{k=1}^{6} {}^{R_b}(\Gamma_1.a_{3k})a_b \tag{3.4.11}$$

where, $^{R_b}\dot{v}_b$ is calculated from the forward recursive calculation, a_b is the unit vector of the base in x and y direction, . is the scalar product and Γ_1 is the hexapod dynamic from the first iteration. Once the base acceleration is computed, the forces/torques of the XY table (series robot) can be expressed as follows:

$$\Gamma_t = M^{R_b}\dot{v}_b \tag{3.4.12}$$

The dynamic of the Gough-Stewart platform can be computed using the relations elaborated in the previous sections, where the $H_k(q, \dot{q}, \ddot{q})$ can be computed in the same way except that the base acceleration is no longer equal zero but equal to $^{R_b}\dot{v}_b$.

$$\Gamma = {}^{R_b}J_{R_p}^T({}^{R_b}\mathbb{F}_{R_p} + {}^{R_b}\mathbb{F}_{leg}) \tag{3.4.13}$$

B- Second method of the XY- 6 DoF dynamic modeling

Recall that the R_0 is the world reference frame with origin O_0, the actuated joints of hybrid robot: $\dot{q}_h = [\dot{q}_x,\ \dot{q}_y,\ \dot{q}_{31},\ \dot{q}_{32},\ \dot{q}_{33},\ \dot{q}_{34},\ \dot{q}_{35},\ \dot{q}_{36}]^T$ and the upper platform speed expressed in R_0: $^{R_0}\dot{X} = [\dot{x},\ \dot{y},\ \dot{z},\ \dot{\phi},\ \dot{\theta},\ \dot{\psi}]^T$ where X is the upper platform position due to the effect of both subsystem. The 8 joints speed can be computed from the following equations:

$$\dot{q}_h = {}^{R_0}J_{p,h}^{-1}\,{}^{R_0}\dot{X} \tag{3.4.14}$$

With,

$$^{R_0}J_{p,h} = \begin{bmatrix} 1 & 0 & \\ 0 & 1 & \\ 0 & 0 & {}^{R_b}J_{R_p}(6\times 6) \\ 0 & 0 & \\ 0 & 0 & \\ 0 & 0 & \end{bmatrix} \tag{3.4.15}$$

The rotation dynamics of the parallel robot and the hybrid robot will not be changed because the base has only a translational movement, the velocities of passive and active joints can be calculated from equations (3.3.22) and (3.4.14). the dynamic modeling of the Gough-Stewart of the hybrid structure is given by the following equation:

$$\Gamma_{s,h} = {}^{R_0}J_p^T(\mathbb{F}_{p,h} + F_e + H_{l,h}) \tag{3.4.16}$$

With,

$${}^{R_0}\mathbb{F}_{p,h} = {}^{R_0}\mathbb{I}_{R_p}\,{}^{R_0}\ddot{X} + \begin{bmatrix} {}^{R_0}w_u \times ({}^{R_0}w_u \times {}^{R_0}MS_{R_p}) \\ {}^{R_0}w_u \times ({}^{R_0}I_{R_p}\,{}^{R_0}w_u) \end{bmatrix} - \begin{bmatrix} M_p I_3 \\ {}^{R_0}M\hat{S}_{R_p} \end{bmatrix} {}^0g \tag{3.4.17}$$

$H_{l,h}$ denotes the resultant of the wrenches applied by the legs on the upper platform taking into account the base acceleration: $\dot{v}_b = [\dot{q}_h(1)\ \dot{q}_h(2)\ 0]^T$.
The dynamic modeling of the XY table is given by the following equations:

$$\begin{aligned} \vec{\Gamma}_x &= [M_{he}\dot{q}_h(1) + P\Gamma_{s,h}]\vec{x}_0 + P\Gamma_{s,h}\vec{z}_0 \\ \vec{\Gamma}_y &= [M_{he+xaxis}\dot{q}_h(2) + P\Gamma_{s,h}]\vec{y}_0 + P\Gamma_{s,h}\vec{z}_0 \end{aligned} \tag{3.4.18}$$

where, M_{he} and $M_{he+xaxix}$ are the mass of the hexapod alone and with the x axis, respectively,

$$P = \begin{bmatrix} \vec{a}_{31}.\vec{x}_0 & . & . & . & . & \vec{a}_{36}.\vec{x}_0 \\ \vec{a}_{31}.\vec{y}_0 & . & . & . & . & \vec{a}_{36}.\vec{y}_0 \\ \vec{a}_{31}.\vec{z}_0 & . & . & . & . & \vec{a}_{36}.\vec{z}_0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{3.4.19}$$

In the same way, as described in (3.4.9), we can generate the compact form of the XY-6 DoF dynamic equation and can be expressed as follows:

$$\Gamma_h = M_h(q)\ddot{q}_h + H(q_h, \dot{q}_h) \tag{3.4.20}$$

where, $\Gamma_h \in \mathbb{R}^8$ is the joints torque vector, $\ddot{q}_h \in \mathbb{R}^8$ is the joints acceleration vector, $H(q_h, \dot{q}_h) \in \mathbb{R}^8$ is the coriolis and gravity vector and $M_h(q) \in \mathbb{R}^{(8\times 8)}$ is the mass matrix.

The analytical dynamic modeling of the mechatronic system expressed in equations (3.4.20) will be validated using a multi-body dynamic software called SimMechanics described in the following.

3.5 Analytical dynamic model validation

The modeling of an hybrid robotic architecture is not an easy task and may occure some imperfection in the dynamic modeling. To verify and validate our dynamic modeling presented in the previous section, we suggest to model the XY-6 DoF in a multi-body dynamic software. Then perform a signals comparison using the analytical dynamic model and the multi-body dynamic software.

3.5.1 Description of multi-body dynamic softwares: SimMechanics

Firstly the system will be modeled using CAD software such as Solidworks.

SolidWorks is a solid modeling computer-aided design (CAD) and computer-aided engineering (CAE) computer program and is published by Dassault Systèmes. SolidWorks use a parametric feature-based approach to create models (parts) and assemblies. This software is very useful in the proof of concept, to demonstrate the feasibility and to save money. A SolidWorks model of the XY-6 DoF platform was developed respecting all the kinematics constraints shown in Fig. 3.2.
SolidWorks is made to design the system in order to be sent to the manufacture and so he has a very large design toolbox and features, but it is not aiming to model the dynamic of a robots and that explain the lack of this toolbox. The dynamic modeling need software that has a large dynamic toolbox to facilitates and master the modeling, therefore another software should be used to model the dynamic of the system. However, the SolidWorks model can be extracted and implemented in several dynamic modeling software such as ADAMS, Gazebo and SimMechanics.
We choose to model the system in SimMechanics because of its advantages comparing to others. SimMechanics is a toolbox of MATLAB that offers a very large dynamic modeling toolbox which facilitates this task and easily connected to MATLAB and Simulink which offer a large control toolbox.

Second the model will be transfered from CAD to Multi-Body dynamic software. The Solidworks (Computer-aided design: CAD) tools allows to model the system geometrically as assemblies which will be translated to the SimMechanics (Simulink) block environment by the intermediate of Physical Modeling XML file which carries all the information of the mass properties, the inertia tensors, the position and orientation of the bodies and the joint types (kinematic constraints).

Third we will describe the multi-Body dynamic software used which is the SimMechanics. The simscape multibody (formerly SimMechanics) provides a multibody simulation environment for 3D mechanical system such as robots. The multibody system can be modeled using a large number of blocks offered in SimMechanics for dynamic modeling representing for example references of bodies, bodies, mass properties, joints, sensors, kinematics and geometrical constraints. SimMechanics formulates and solves the equations of motion for the complete mechanical system using the Newton equations of the torques and forces. After that, an automatically 3D animation can be generated to visualize the system dynamics.
The SimMechanics several family of product permit to modeling any system such as the mechanical, hydraulic, pneumatic and electrical toolbox as well as permit the use of MATLAB which parameterize the multibody system with the Simulink which has a large number of blocks for visualization and design control such as scope, PID block and many control methods. An CAD assembly model can be imported by the intermediate of XML file as described above.
The SimMechanics use the ODE ordinary differential equation solvers offered by Simulink to solve the Newton's equations which relate the cause and effect by integrating either the motion or the forces/torques to solve the well-known modeling (inverse and forward dynamic modeling). The dynamic equations allow to analyze the XY-6 DoF platfrom behavior in forward and inverse kinematic:

1. In forward dynamics: a set of forces/torques are given to the bodies to produce accelerations. The SimMechanics calculat the velocities and positions as function of time by integrating the acceleration and using the initial position and velocities.

2. Inverse dynamics: the motions as function of time is given. The SimMe-

chanics calculat the forces and torque by differentiate the motions and by the application of Newton equation.

3.5.2 XY- 6 DoF Platform SimMechanics Modeling

The platform XY- 6 DoF is modeled under Simmechanics following the detailed procedures described before. Fig. 3.6 shows the 3D animated picture example of the XY-6 DoF platfrom for Top view. Fig. 3.8 shows the simmechanics blocks of the XY-6 DoF platfrom. Dry and viscous friction parameters that will be identified in Chapter 4. These parameters can be added to XY- 6 DoF modeling blocks to have better emulation of the reality.

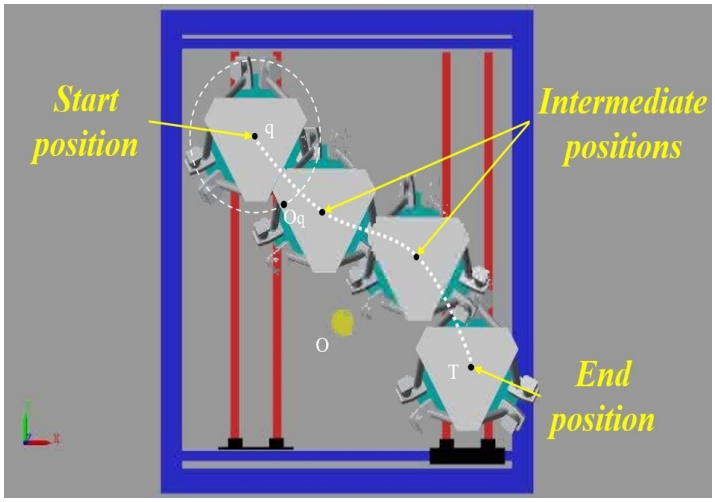


Figure 3.6: 3D animated figure top view

The analytical form of the mechatronic system dynamic model shown in equations (3.4.16) and (3.4.18) depends mainly on two quantities. Let us define $\Gamma = g(G) + f(A)$ where G- gravity quantity and A- acceleration quantity. The validation of the torque response will performed by choose of two input signals. The first trajectory consist of low acceleration movement where the gravity quantity g(G) is dominant and f(A) has minor effect on the torque. The second trajectory consist of high acceleration movement where the f(A) is dominant.
Two trajectory models of the simulator end-effector has the same shape with high and low values and are defined in the following form:

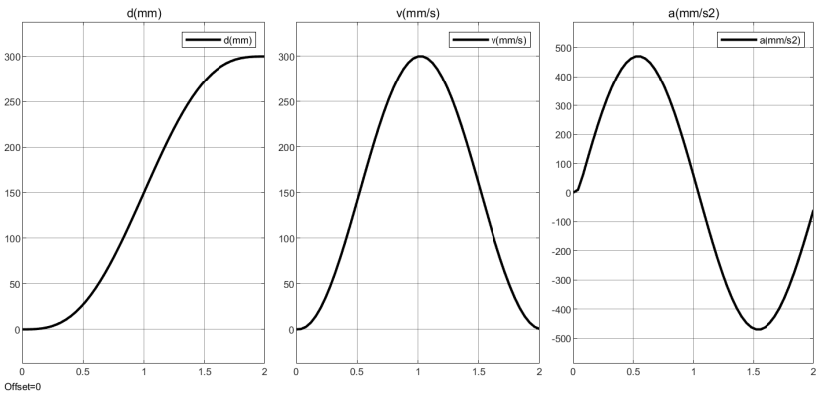


Figure 3.7: Trajectory template for high acceleration

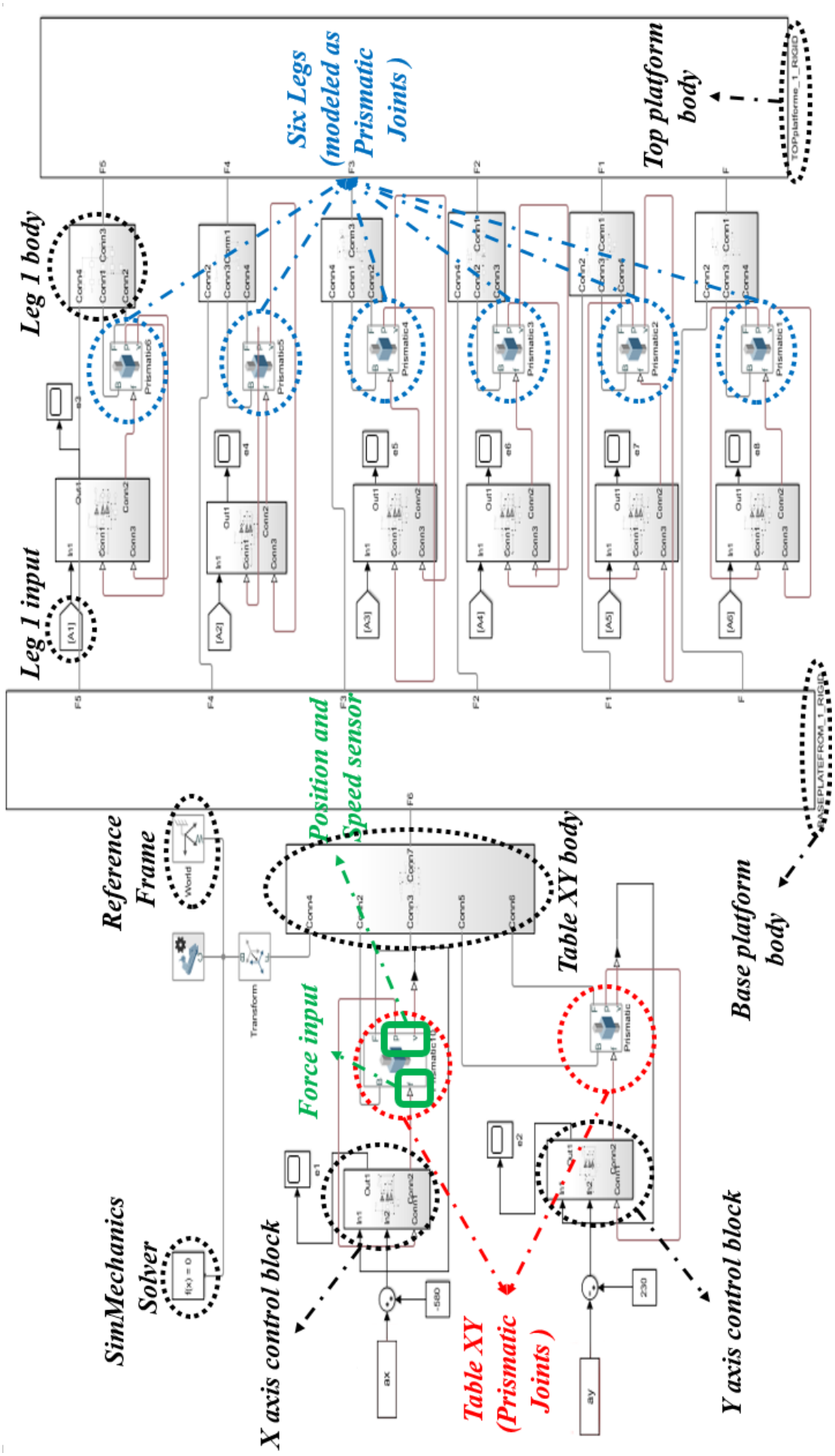


Figure 3.8: XY- 6 DoF Simmechanics Flow blocks

The following template is the upper platform movement, we will search for the required torque that can perform the same required upper platform movement by both analytical and software path. In this section, we will not show the displacement movement, speed and acceleration of each leg and axis and we will only show the comparaison of torque signals. Forces comparaison will be performed for GS movement along Y axis and Table movement along X axis for low and high speed.

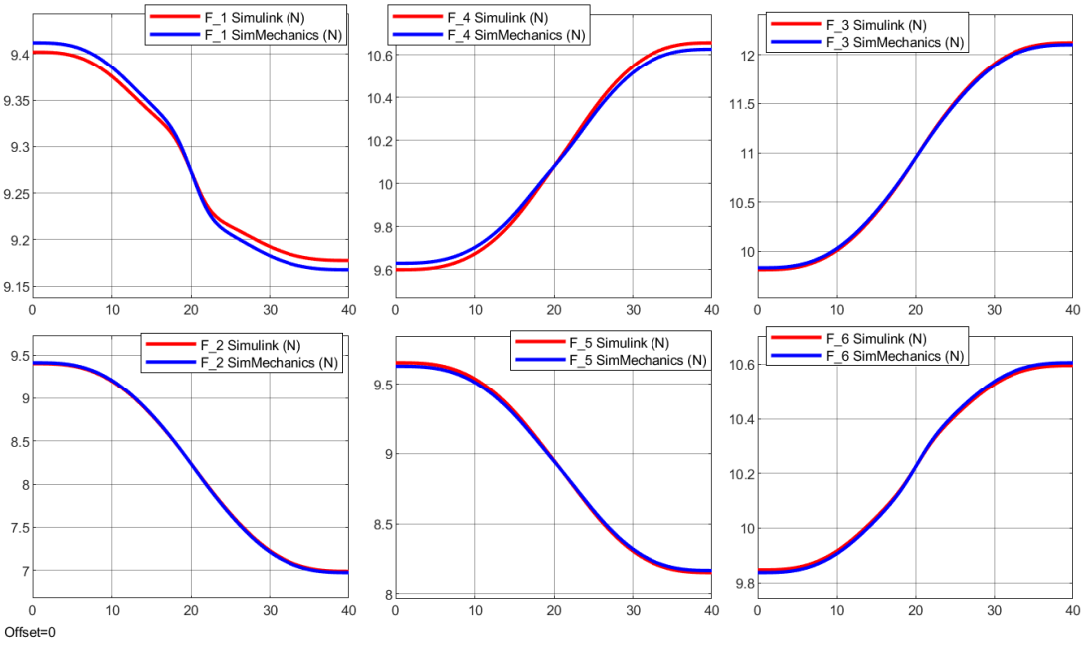


Figure 3.9: Joints torques comparison when gravity quantity is dominant

Fig. 3.9 and Fig. 3.10 show the torque comparaison of the legs joints for low and high acceleration. Fig. 3.9 shows a torque curve in the shape of the upper platform displacement because it is driven by the gravity effect and shows a similar torque signals with errors less then 5%.

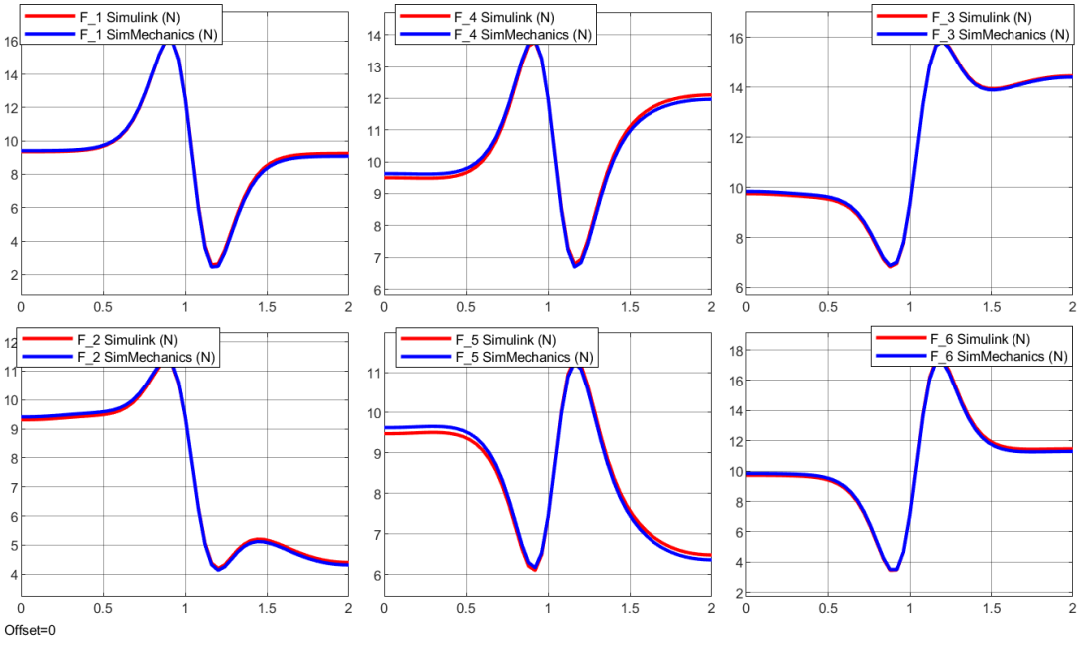


Figure 3.10: Joints torques comparison when acceleration quantity is dominant

Fig. 3.10 shows a torque curve in the shape of the upper platform acceleration

because it is driven by the high acceleration effect and shows a similar torque signals with errors less then 5%. These figures show that the errors between the forces obtained from our proposed dynamic modeling and from the multibody dynamic software model are less than 5% for both high and low acceleration trajectories. For sliding application, the 5% difference is acceptable, therefore, the dynamic modeling of the hybrid mechatronic platform is validated. For more behavior comparaison we can see Appendix.A section dynamic validation.

3.6 Dynamic parameters optimization and identification

It is well known that the dynamic parameters play an important role in describing the behavior of a multibody system such as robot manipulators Lopes, (2009) [36]. Hence, an optimized analytical form of the dynamic model must be developed, and the number of mathematical operations must be reduced. Thus, a control based on an optimized dynamic model can be build and easily implemented in real time. The Gough-Stewart platform has been extensively studied in the literature from modeling and control views due to the high stiffness, payload capacity and high accuracy that occur Kalani, (2016) [37] Merlet (2000) [48]. In addition, the Gough-Stewart platform has retained the attention of industries, and a lot of tests and applications have been developed. One cites, for example, simulators for land vehicles in the automotive industry, flight simulators in the fields of aeronautics and submarine simulators as well as they have been used as manipulators.

The dynamics of a manipulator are a set of nonlinear differential equations, which makes the determination of the inertial parameters challenging. Most advanced control schemes existing in the literature for robots require dynamic models. The precision, performance, stability, and robustness of these schemes depend, to a large extent, on the accuracy of the parameters that describe the dynamic model. Some control schemes aimed at achieving perfect feedback linearization, such as the computed torque technique, assume precise knowledge of the dynamic parameters.

The precise values of dynamic parameters are generally unknown, even for robot manufacturers. In this chapter, we will propose some methods to optimize and reduce the number of dynamic parameters then we will exploit the fact that the energy model is linear in these parameters in order to identify them. The problem will be reduced to the solution of least squares of an overdetermined linear system of equations.

3.6.1 Dynamic parameters optimization

Such a procedure requires a dynamic based control procedure where all system parameters must to be minimized and identified Mata (2008) [49]. With respect to (Denavit-Hartenberg) parametrization and following notations given by Khalil, (2002) [32], the dynamic of the studied system depends on 210 dynamic parameters knowns as standard inertial parameters. In the following, we detail the methods carried to optimize the number of dynamic parameters.

Proposition 1. For a given link, in the link energy, if the column of h_k is equal to zero or a constant number, the corresponding inertial parameter can be eliminated.

Proof. Let us introduce the total energy H_k of the k^{th} link

$$H_k = E_k + U_k = (e_k + u_k)\xi_k^{in} = h_k \xi_k^{in} \tag{3.6.1}$$

where E_k is the kinetic energy of the k^{th} link,

$$\begin{aligned} E_k &= \frac{1}{2}[\omega_k^T J_k \omega_k + M_k V_k^T V_k + 2MS_k^T (V_k \times \omega_k)] && (3.6.2)\\ &= e_k \xi_k^{in} && (3.6.3) \end{aligned}$$

and U_k is the potential energy of the k^{th} link,

$$\begin{aligned} U_k &= -M_k\, {}^0g^T({}^0P_k + {}^0A_k\, {}^kS_k) && (3.6.4)\\ &= u_k \xi_k^{in} && (3.6.5) \end{aligned}$$

where e_k and u_k are functions of the kinematic variables, given as follows:

ξ_k^{in}	XX	XY	XZ	YY	YZ
e_k	$\frac{1}{2}\omega_{1,k}\omega_{1,k}$	$\frac{1}{2}\omega_{1,k}\omega_{2,k}$	$\frac{1}{2}\omega_{1,k}\omega_{3,k}$	$\frac{1}{2}\omega_{2,k}\omega_{2,k}$	$\frac{1}{2}\omega_{2,k}\omega_{3,k}$
u_k	0	0	0	0	0
ξ_k^{in}	ZZ	MX	MY	MZ	M
e_k	$\frac{1}{2}\omega_{3,k}\omega_{3,k}$	$\omega_{3,k}V_{2,k} - \omega_{2,k}V_{3,k}$	$\omega_{1,k}V_{3,k} - \omega_{3,k}V_{1,k}$	$\omega_{2,k}V_{1,k} - \omega_{1,k}V_{2,k}$	$\frac{1}{2}{}^kV_k^T\, {}^kV_k$
u_k	0	$-{}^0g^T\, {}^0s_k$	$-{}^0g^T\, {}^0n_k$	$-{}^0g^T\, {}^0a_k$	$-{}^0g^T\, {}^0P_k$

Table 3.2: e_k and u_k vector elements

where, ${}^k\omega_k = [\omega_{1,k}\ \omega_{2,k}\ \omega_{3,k}]^T$, ${}^kV_k = [V_{1,k}\ V_{2,k}\ V_{3,k}]^T$ and ${}^0R_k$ and ${}^0P_k$ are the matrix and vector appearing in the transformation matrix ${}^0T_k$ with the rotation matrix ${}^0R_k = [{}^0s_k\, {}^0n_k\, {}^0a_k]$, a_k unit vector along Z_k direction, ${}^kJ_k$ - (3×3) inertia tensor of link k with respect to frame F_k. M_k and MS_k are the mass and first moment of inertia of link k respectively.

The link k standard inertial parameters ξ_k and h_k vector are defined by:
$\xi_k^{in} = [XX_k\ XY_k\ XZ_k\ YY_k\ YZ_k\ ZZ_k\ MX_k\ MY_k\ MZ_k\ M_k]^T \in \mathbb{R}^{10\times1}$.
$h_k = [h_{XX_k}\ h_{XY_k}\ h_{XZ_k}\ h_{YY_k}\ h_{YZ_k}\ h_{ZZ_k}\ h_{MX_k}\ h_{MY_k}\ h_{MZ_k}\ h_{M_k}]^T \in \mathbb{R}^{10\times1}$.
where, the vector ξ_k is composed of the standard dynamic parameters of the link k.

1. XX_k , XY_k , XZ_k , YY_k , YZ_k , ZZ_k are the 6 components of the inertia tensor of the body j with respect to the reference R;

2. MX_k , MY_k , MZ_k are the 3 components of the first moment of inertia produced by mass by the coordinates of the center of gravity of the body j in the reference R;

3. M_k is the mass of the link k;

Therefore, k^{th} link energy H_k is as:

$$H_k = \sum_{j=1}^{r_2} h_k^j\ \xi_k^{j,in} \tag{3.6.6}$$

where r_2 is the number of inertial parameters by link, $r_2 = 10$, h_k^j is the j^{th} element of h_k vector corresponding to k^{th} link and $\xi_i^{j,in}$ is the j^{th} element of the inertial vector corresponding to i^{th} link.
Let us take the l^{th} column of link k $h_k^l = const$, we can write H_k as

$$H_k = \sum_{j=1}^{l-1} h_k^j \, \xi_k^{j,in} + \sum_{j=l+1}^{r_2} h_k^j \, \xi_k^{j,in} + const \tag{3.6.7}$$

Since the dynamic effect of such link is seen by the differentiation of the total energy ΔH_k. Therefore, we can set $\xi_k^{l,in} = 0$ without changing the value of ΔH_k. This means that $\xi_k^{l,in}$ has no effect on the dynamic model, then it can be eliminated.

Proposition 2. For one leg, in the link energy, if the h_k column can be expressed as function of the previous link'energy function h_{k-1}, then ξ_k^{in} can be grouped with ξ_{k-1}^{in}, and another grouped inertial parameter $\xi_{R,k-1}^{in}$ appears.

Proof. The robot leg energy H_{leg} is as:

$$H_{leg} \;=\; \sum_{i=1}^{r_1} \sum_{j=1}^{r_2} h_i^j \, \xi_i^{j,in} \tag{3.6.8}$$

where r_1 is the number of link by leg, in Evr@ simulator, $r_1 = 3$, h_i^j is the j^{th} element of the energy vector corresponding to i^{th} link and $\xi_i^{j,in}$ is the j^{th} element of the inertial vector corresponding to i^{th} link, so $h_i \xi_i^{in} = \sum_{j=1}^{r_2} h_i^j \, \xi_i^{j,in}$.
In one leg, let us consider a possible energy linear dependence between l^{th} column of the link k and link $k-1$, which can take the following form $h_k^l = h_{k-1}\alpha_{k-1}$, where $\alpha_{k-1} \in \mathbb{R}^{10\times 1}$. Then,

$$\begin{aligned}
H_{leg} &= \sum_{i=1}^{k-2} \sum_{j=1}^{r_2} h_i^j \, \xi_i^{j,in} + \sum_{i=k+1}^{r_1} \sum_{j=1}^{r_2} h_i^j \, \xi_i^{j,in} + \sum_{j=1}^{r_2} h_{k-1}^j \, \xi_i^{j,in} + \sum_{j=1}^{r_2} h_k^j \, \xi_i^{j,in} \\
&= \sum_{i=1}^{k-2} \sum_{j=1}^{r_2} h_i^j \, \xi_i^{j,in} + \sum_{i=k+1}^{r_1} \sum_{j=1}^{r_2} h_i^j \, \xi_i^{j,in} \\
&\qquad + \sum_{j=1}^{l-1} h_k^j \, \xi_i^{j,in} + \sum_{j=l+1}^{r_2} h_k^j \, \xi_i^{j,in} + h_{k-1}\xi_{k-1}^{in} + h_{k-1}\alpha_{k-1}\xi_k^{l,in} \\
&= \sum_{i=1}^{k-2} \sum_{j=1}^{r_2} h_i^j \, \xi_i^{j,in} + \sum_{i=k+1}^{r_1} \sum_{j=1}^{r_2} h_i^j \, \xi_i^{j,in} \\
&\qquad + \sum_{j=1}^{l-1} h_k^j \, \xi_i^{j,in} + \sum_{j=l+1}^{r_2} h_k^j \, \xi_i^{j,in} + h_{k-1}(\xi_{k-1}^{in} + \alpha_{k-1}\xi_k^{l,in}) \\
&= \sum_{i=1}^{k-2} \sum_{j=1}^{r_2} h_i^j \, \xi_i^{j,in} + \sum_{i=k+1}^{r_1} \sum_{j=1}^{r_2} h_i^j \, \xi_i^{j,in} + \sum_{j=1}^{l-1} h_k^j \, \xi_i^{j,in} + \sum_{j=l+1}^{r_2} h_k^j \, \xi_i^{j,in} + h_{k-1}\xi_{R,k-1}^{in}
\end{aligned} \tag{3.6.9}$$

where a grouped parameter can be identified, $\xi_{R,k-1}^{in} = \xi_{k-1}^{in} + \alpha_{k-1}\xi_k^{l,in}$.

Proposition 3. The 8 DoF optimized dynamics parameters are of number 90 parameters instead of 210 standard inertial parameters

Proof. At First, let us note that all the link velocities are computed using the

recursive Newton-Euler relations such as presented in Khalil, (2002) [32]:

$$\begin{aligned} {}^{k}\omega_k &= {}^{k}R_{k-1}\,{}^{k-1}\omega_{k-1} + \overline{\sigma}_k \dot{q}_k\,{}^{k}a_k = {}^{k}\omega_{k-1} + \sigma_k \dot{q}_k\,{}^{k}a_k \\ {}^{k}V_k &= {}^{k}R_{k-1}({}^{k-1}V_{k-1} + {}^{k-1}\omega_{k-1} \times {}^{k-1}P_k) + \sigma_k \dot{q}_k\,{}^{k}a_k \end{aligned}$$

The columns of h_k which are constant:

1. for sliding joint: $h_{XX},\ h_{XY},\ h_{XZ},\ h_{YY},\ h_{YZ},\ h_{ZZ},\ h_{MX},\ h_{MY},\ h_{MZ}$ (9 parameters eliminated)
2. for the first rotational link of each leg: $h_{XX},\ h_{XY},\ h_{XZ},\ h_{YY},\ h_{YZ}$ (5 parameters eliminated by leg)

Now as a direct application of Propostion 1, a 48 $(9 \times 2 + 5 \times 6)$ standard inertial parameters are eliminated. For example, the first element of h_k vector of the sliding joint is equal to $h_s^1 = \frac{1}{2}\omega_{1,s}\omega_{1,s}$, as the sliding joint has no rotational movement which means $\omega_{1,s} = 0$. Consequently, $h_s^1 = 0$ then the XX_s parameter is eliminated.

For one leg, the h_k columns which have linear dependence with the h_{k-1} previous link, are for the first rotational link: $h_{MZ},\ h_M$, for the second rotational link: $h_{YY}\ h_{MZ},\ h_M$ and for the prismatic link: $h_{XX},\ h_{XY},\ h_{XZ},\ h_{YY},\ h_{YZ},\ h_{ZZ},\ h_M$. After application of Proposition 2, 72 standard inertial parameters are grouped, 12 by leg. Example, for one leg,

$$\omega_2 = \begin{bmatrix} \sin\theta_2\ \dot{q}_1 \\ \cos\theta_2\ \dot{q}_1 \\ \dot{q}_2 \end{bmatrix}, \omega_3 = \begin{bmatrix} \sin\theta_2\ \dot{q}_1 \\ \dot{q}_2 \\ -\cos\theta_2\ \dot{q}_1 \end{bmatrix}; \tag{3.6.10}$$

we can find that $h_3^6 = \frac{1}{2}(\cos\theta_2\ \dot{q}_1)^2$, $h_2^4 = \frac{1}{2}(\cos\theta_2\ \dot{q}_1)^2$ which gives us $h_3^6 = h_2\alpha_2$ where $\alpha_2 = [0\ 0\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 0]^T$. Consequently, The ZZ_3 inertial parameter is grouped with link-2 inertial parameters $\xi_{R,2}^{in} = \xi_2^{in} + \alpha_2\xi_3^{l,in}$ following the equation (3.6.9).

□

The grouping relations of the developed parameters are, therefore :

$$\begin{aligned} ZZ_{1,Rk} &= YY_{2k} + ZZ_{1k} + ZZ_{3k} \\ MY_{1,Rk} &= MY_{1k} - MZ_{2k} \\ XX_{2,Rk} &= XX_{2k} + XX_{3k} - YY_{2k} - ZZ_{3k} \\ XY_{2,Rk} &= XY_{2k} - XZ_{3k} \\ XZ_{2,Rk} &= XY_{3k} + XZ_{2k} \\ YZ_{2,Rk} &= YZ_{2k} - YZ_{3k} \\ ZZ_{2,Rk} &= YY_{3k} + ZZ_{2k} \end{aligned} \tag{3.6.11}$$

Idem, the upper platform grouped parameters are:

$$\begin{aligned} XX_{RP} &= XX_P + \sum_{i=2}^{6}\left[\left({}^{P}L_{2,k}^2 + {}^{P}L_{3,k}^2\right)M_{3k}\right] \\ YY_{RP} &= YY_P + \sum_{i=2}^{6}\left[\left({}^{P}L_{1,k}^2 + {}^{P}L_{3,k}^2\right)M_{3k}\right] \\ ZZ_{RP} &= ZZ_P + \sum_{i=2}^{6}\left[\left({}^{P}L_{1,k}^2 + {}^{P}L_{2,k}^2\right)M_{3k}\right] \\ XY_{RP} &= XY_P - \sum_{i=3}^{5}\left[{}^{P}L_{1,k}^2\,{}^{P}L_{2,k}^2 M_{3k}\right] \end{aligned}$$

$$
\begin{aligned}
XZ_{RP} &= XZ_P - \sum_{i=3}^{5}\left[{}^P L_{1,k}^2\, {}^P L_{3,k}^2 M_{3k}\right] \\
YZ_{RP} &= YZ_P - \sum_{i=3}^{5}\left[{}^P L_{2,k}^2\, {}^P L_{3,k}^2 M_{3k}\right] \\
MX_{RP} &= MX_P + \sum_{i=2}^{6}\left[{}^P L_{1,k}^2 M_{3k}\right];\ MY_{RP} = MY_P + \sum_{i=2}^{6}\left[{}^P L_{2,k}^2 M_{3k}\right] \\
MZ_{RP} &= MZ_P + \sum_{i=2}^{6}\left[{}^P L_{3,k}^2 M_{3k}\right];\ M_{RP} = M_P + \sum_{i=1}^{6} M_{3k}
\end{aligned}
\tag{3.6.12}
$$

Therefore, we present Table. 3.3 that shows the eliminated parameters (dashed boxes) and the grouped parameters.

Joint	XX	XY	XZ	YY	YZ	ZZ	MX	MY	MZ	M
$X-axis$	-	-	-	-	-	-	-	-	-	M_X
$Y-axis$	-	-	-	-	-	-	-	-	-	M_Y
1_k	-	-	-	-	-	ZZ_{1R}	MX_1	MY_{1R}	-	-
2_k	XX_{2R}	XY_{2R}	XZ_{2R}	YZ_{2R}	-	ZZ_{2R}	MX_2	MY_2	-	-
3_k	-	-	-	-	-	-	MX_3	MY_3	MZ_3	-
Upper platform	XX_{RP}	XY_{RP}	XZ_{RP}	YY_{RP}	YZ_{RP}	ZZ_{RP}	MX_{RP}	MY_{RP}	MZ_{RP}	M_{RP}

Table 3.3: Optimized inertial parameters of the XY - 6 DoF simulator

3.6.2 Dynamic parameters identification

There are three main methods used for estimating of the inertial parameters of a robot:

1. Physical experiments: if we could disassemble the robot to isolate each link, the following parameters could be obtained by physical experiment :

   - the mass could be weighed directly;
   - the coordinates of the center-of-mass could be estimated by determining counterbalanced points of the link;
   - the diagonal elements of the inertia tensor could be obtained by pendular motions;

   This method is very tedious and should be realized by the manufacturer before assembling the robot ;

2. Using CAD/CAM models: all robotics CAD/CAM packages provide tools to calculate the inertia parameters from 3D models. This method is prone to errors due to the fact that the geometry of the links is complicated to define precisely, and that certain parts such as bearings, bolts, nuts, and washers are generally neglected;

3. Identification: this approach is based on the analysis of the "input/output" behavior of the robot on some planned motion and on estimating the parameter values by minimizing the difference between a function of the real

robot variables and its mathematical model. This method has been used extensively and was found to be the best in terms of ease of experimentation and precision of the obtained values. In this section, we consider off-line identification methods, for which we collect all the input-output data prior to analysis;

Fig. 3.11 shows the flow block diagram used for the experimental identification.

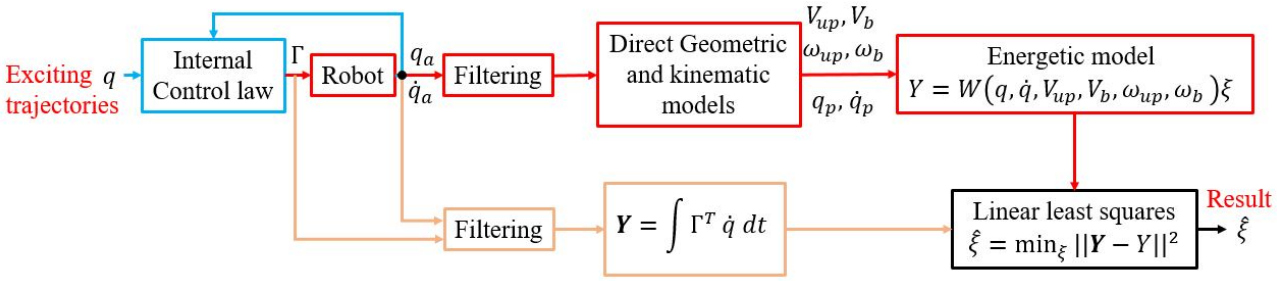


Figure 3.11: Schematic diagram of the energy model

$\Gamma \in \mathbb{R}^8$ is the vector of generalized torques, $q = [q_a, q_p]$, $\dot{q} = [\dot{q}_a, \dot{q}_p]$ where $q_a \in \mathbb{R}^8, q_p \in \mathbb{R}^{12}$ are the active and passive joint positions, (V_b, ω_b) and $(V_{up}, \omega_{up}) \in (\mathbb{R}^3 \times \mathbb{R}^3)$ are the speed vectors of the base and the upper platform respectively, $\xi = [\xi^{in}, \xi^f]$ where $\xi^{in}, \xi^f = [F_v, F_s]^T$ are the inertial and friction parameters respectively, F_v, F_s are the viscous and dry friction coefficients.

Several exciting trajectories q has been applied with the highest velocity to take around 50 000 equations ($> 500\times$number of parameters) to make sure that the trajectories have enough information, all detected variables such as displacements, velocities and torques were filtered in order to eliminate the high frequency which is harmful for the identification procedure.

The parameters identification will be performed based on the energetic model. An energy model based on the optimized dynamic parameters has been applied in the following form, thus, we can write:

$$\begin{aligned} \Delta H &= T^T dq \\ &= T^T \dot{q}\, dt \end{aligned} \tag{3.6.13}$$

where,

$$T = \Gamma - diag(sign(\dot{q}))F_s - diag(\dot{q})F_v \tag{3.6.14}$$

After integration,

$$\int_{t_1}^{t_2} T^T \dot{q}\, dt = H(t_2) - H(t_1), \tag{3.6.15}$$

Hence, the total energy can be expressed linearly as function of the inertial and friction parameters, we can write:

$$\mathbf{Y} = \int_{t_1}^{t_2} \Gamma^T \dot{q}\, dt = \Delta H + \Delta H_{F_v} + \Delta H_{F_s} \tag{3.6.16}$$

$$= \Delta h \xi^{in} + \Delta h_{F_v} F_v + \Delta h_{F_s} F_s \tag{3.6.17}$$

Therefore, the identification energy model form is as follow:

$$\mathbf{Y} = \int_{t_1}^{t_2} \Gamma^T \dot{q}\, dt = [\Delta h\ \Delta h_{F_v}\ \Delta h_{F_s}] \begin{bmatrix} \xi^{in} \\ F_v \\ F_s \end{bmatrix} \tag{3.6.18}$$

$$\mathbf{Y}(\Gamma, \dot{q}) = W(q, \dot{q}) \begin{bmatrix} \xi^{in} \\ F_v \\ F_s \end{bmatrix} + \rho \tag{3.6.19}$$

where W is an $(r \times c)$ observation matrix, or regressor, r is the total number of equations, c is the number of parameters such that $r >> c$, and ρ is the residual error vector.

The most used method for identification is the least squares method. The least-squares estimation method is a well-tested algorithm for robot parameter identification Atkeson (1986) [50] - Khalil and Gautier (2007) [55]. These methods are based upon the Newton-Euler or Lagrangian formulation of dynamics and try to modify the dynamic equation to be expressed as a set of linear equations with respect to the inertial parameters. This formulation allows the least squares technique to estimate the inertial parameters of the payload. There are many different approaches to implementing this method for parameter identification.
In Atkeson (1986) [50], a force sensor was used on the end effector to simplify the parameter identification of the load, whereas Khosla (1986) [51] did not require one in its implementation. In reference Xuecai (1994) [52], the reflected inertias of the actuators and the paths in joint space were used as the measurement data, as opposed to the more usual method of only joint measurements. The method presented in Gautier (1994) [53] utilizes the total least squares field to identify all the inertial and drive gain parameters, which is demonstrated to be more efficient than ordinary least squares techniques. In reference Gautier (2008) [54], a method was proposed that was capable of determining the inertial parameters from using only the torque data of a robot, by using non-linear least squares estimation to match the parameters of a simulation to actual sampled data.

An important sub-problem that this work will focus on is determining an exciting trajectory that produces the best estimation of inertial parameters Armstrong (1978) [56] - Vandanjon (1995) [59]. The identification methods outlined above can work with any trajectory, but produce better estimates when the inertial, coriolis, and gravity terms in the dynamic equation are sufficiently excited. It is shown that the condition number of the excitation matrix directly influences the convergence rate and noise sensitivity of the identification algorithm Armstrong (1978) [56]. It is suggested in Armstrong (1978) [56] that the condition number must be optimized close to unity to produce the best identification results.
In Gautier (1991) [57], this idea is extended in that a trajectory is generated using nonlinear optimization techniques to minimize a cost function involving the condition number. Lucyshyn (1993) [58] proposes a method to generate an exciting trajectory made using mutual product-wise odd functions. The above trajectory generation algorithms are not guaranteed to produce a feasible trajectory for a load-bearing system. The motors may be incapable of tracking the trajectory since it violates the motor torque constraints. The motors may become saturated and unable to move in this situation.
A sequential identification paradigm is presented in Vandanjon (1995) [59], where a set of different trajectories are sampled, with each one exciting different parameters. Each trajectory is also designed to minimize the condition number of the excitation matrix. It is easier to assure feasible trajectories for the manipulator when using this paradigm, and, as such, will be the method implemented in this work. As such, this paradigm will be used when developing exciting trajectories for the identification of the payload.

A least square method has been used to solve the overdetermined system expressed as follows:

$$\hat{\xi} = [W^T W]^{-1} \; W^T \; \mathbf{Y} = W^{+}\mathbf{Y} \tag{3.6.20}$$

where W^{+} is the pseudoinverse matrix of W. The precision of the identified parameter $\hat{\xi}$ is depend on the persistence of the observation matrix Yao (2018) [60], Mirza (2017) [61].
The identification results of the two sliding joints are shown in the Table 3.4. In

Axes	M (Kg)	$F_{v_1}(N.s/mm)$	$F_{s_1}(N)$	$F_{v_2}(N.s/mm)$	$F_{s_2}(N)$
First	33,7	76,38	9,14	72,38	11.23
Second	20,56	37,30	9,60	42,98	10,44

$F_{v_1}, F_{v_2}, F_{s_1}, F_{s_2}$ viscous and dry frictions in the first and second directions respectively.

Table 3.4: Two sliding joints inertial parameters

order to verify the identification results, a torque construction from the model using the Lagrangian formulation and the identified parameters is performed. The Lagrangian is equal to the total kinetic energy differentiated by the total potential energy:

$$L = E - U = e_k \hat{\xi}_k^{in} - u_k \hat{\xi}_k^{in}; \quad \Gamma_{Model} = \Gamma_{\hat{\xi}^{in}} + \Gamma_{\hat{\xi}^{f}} \tag{3.6.21}$$

All legs have similar behavior, thus, we will only present the leg B_4P_4. The torque detected and constructed of leg B_4P_4 (Fig. 3.2) is shown in the following:

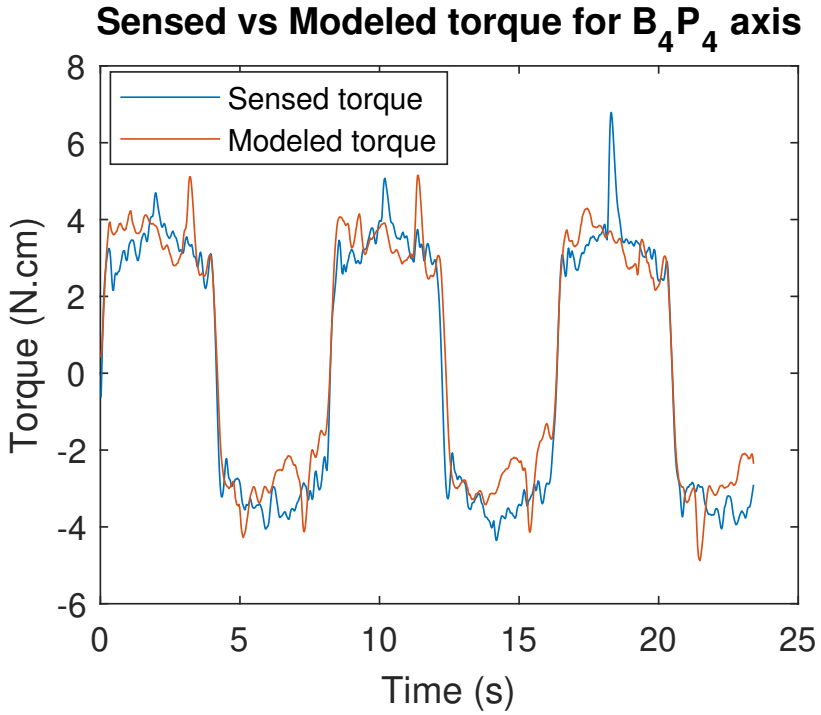


Figure 3.12: Sensed and modeled torque

The corresponding energy signals (sensed energy $\int_{t_1}^{t_2} \Gamma^T \dot{q} \; dt$ and the modeled energy $\Delta h \xi^{in} + \Delta h_{F_v} F_v + \Delta h_{F_s} F_s$) are shown in Fig. 3.13.
The B_4P_4 axis friction was taken in two directions to take into account the dissymmetry of the friction model, we note 1 and 2 for the first and the second direction, respectively: $F_{s_1} = 1{,}39 \; N, F_{v_1} = 56{,}98 \; N.s/mm, F_{s_2} = 1{,}33 \; N, \; F_{v_2} = 58{,}31 \; N.s/mm$, Finally, the grouped mass of upper platform and last 6 links is $M_{R_p} = 6.15 \; Kg$. It should be noted that the presence of peaks is due to the glitch effect of the electronic measurement.

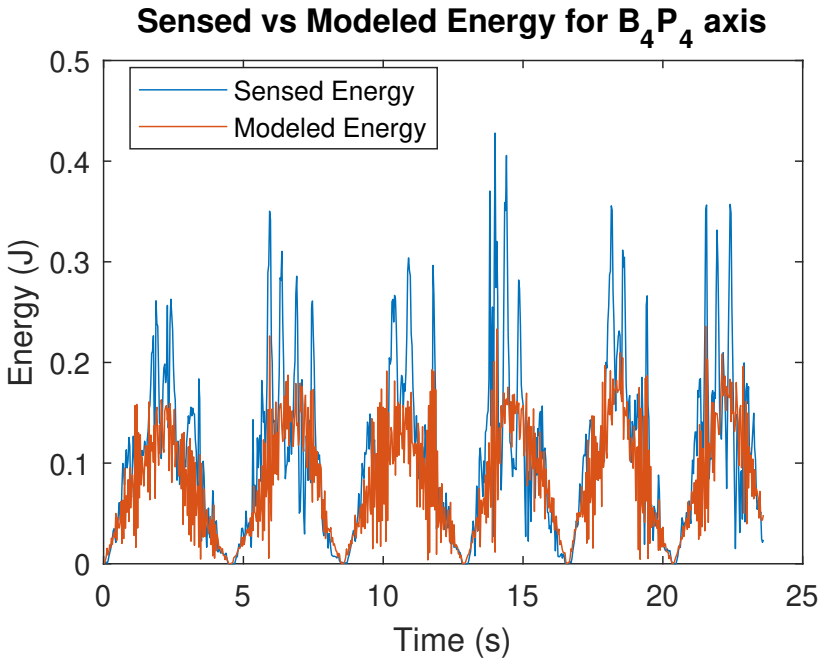


Figure 3.13: Real and modeled energy

The identified frictions of the XY table and the Gough-Stewart platform are presented in forward and backward direction in the following figures. We assume that the friction values change with the movement direction.

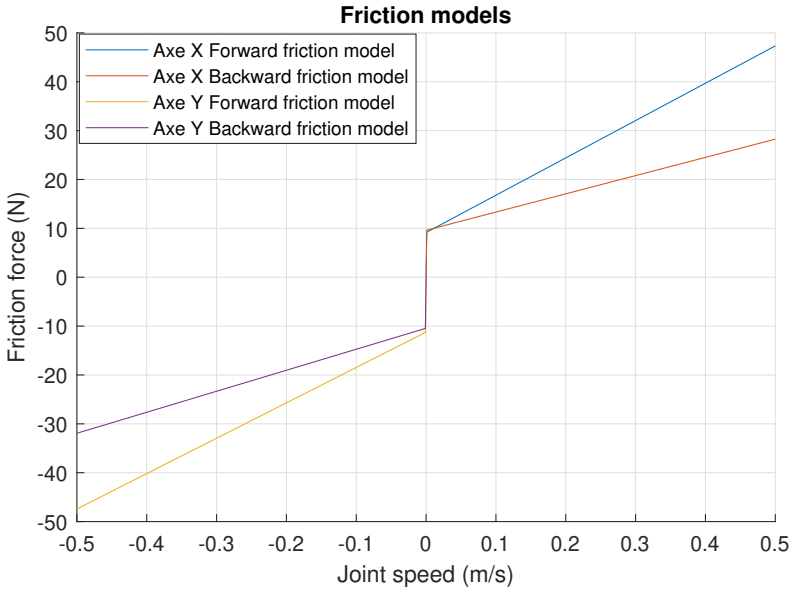


Figure 3.14: Friction models of XY table

It should be noted also that this least-squares estimation is biased because the observation matrix W is random, and because W and ρ are realizations of random and correlated variables Gautier (1986) [62]. Furthermore, the elements of the matrix W are nonlinear functions in q and $\dot{q}$, which leads one to assume some statistical properties of the noise in order to calculate the quality of the estimation process (bias and standard deviation) Raucent (1990) [63]. Consequently, it is important to verify the accuracy of the values obtained using appropriate validation procedures.

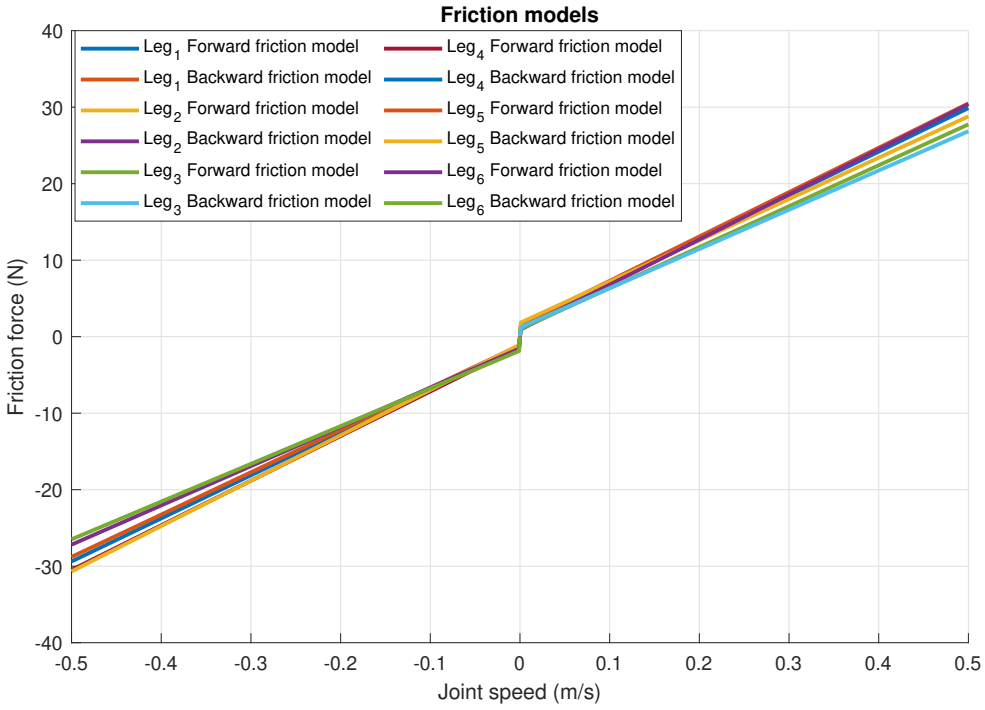


Figure 3.15: Friction models of Gough-Stewart platform

The standard deviations of the estimated values are calculated by assuming that W is deterministic, and ρ is a zero mean additive independent Gaussian noise, with standard deviation σ_p such that Gautier (2001) [64], the variance-covariance matrix C_p is given by:

$$C_p = E(\rho\rho^T) = \sigma_p^2 I_r \tag{3.6.22}$$

where E is the expectation operator, and I_r is the $(r \times r)$ identity matrix. An unbiased estimation of σ_p can be calculated using the following equation:

$$\sigma_p^2 = \frac{||Y - W\hat{\xi}||^2}{(r-c)} \tag{3.6.23}$$

The variance-covariance matrix of the estimation error is given by:

$$C_{\hat{\xi}} = E[(\xi - \hat{\xi})(\xi - \hat{\xi})^T] = W^+ C_p (W^+)^T = \sigma_p^2 (W^T W)^{-1} \tag{3.6.24}$$

The standard deviation on the j^{th} parameter is obtained from the (j,j) element of $C_{\hat{\xi}}$:

$$\sigma_{\hat{\xi}_j} = \sqrt{C_{\hat{\xi}}(j,j)} \tag{3.6.25}$$

The relative standard deviation can be used as a criterion to measure the quality of the identification value for each parameter. It is obtained as:

$$\sigma_{\hat{\xi}_{jr}}\% = 100\frac{\sigma_{\hat{\xi}_j}}{|\hat{\xi}_j|} \tag{3.6.26}$$

For example, if the relative standard deviation of a parameter is greater than ten times the minimum relative standard deviation value, this parameter can be considered as poorly identified.

Parameter	M	F_v	F_d
Slider$_X$ $\sigma_{\hat{\xi}_r}$%	0.77	0.88	0.51
Slider$_Y$ $\sigma_{\hat{\xi}_r}$%	0.79	3.4	1.04

Table 3.5: Identification results of the sliders

The maximum XY table relative standard deviation $\sigma_{\hat{\xi}_r}$ % is lower than 4% which implies that all parameters are well identified.

Parameter	XX_{RP}	XY_{RP}	XZ_{RP}	YY_{RP}	YZ_{RP}	ZZ_{RP}	MX_{RP}	MY_{RP}	MZ_{RP}	M_{RP}
Upper platform $\sigma_{\hat{\xi}_r}$%	1.63	**1.92**	0.74	3.16	1.5	2.06	**0.04**	0.14	0.3	1.15

Table 3.6: Identification results of the upper platform

The maximum upper platform relative standard deviation $\sigma_{\hat{\xi}_r}$% is lower than 2% which implies that all parameters are well identified.

Parameter	Leg$_1$ $\sigma_{\hat{\xi}_r}$%	Leg$_2$ $\boldsymbol{\sigma_{\hat{\xi}_r}}$**%**	Leg$_3$ $\sigma_{\hat{\xi}_r}$%	Leg$_4$ $\sigma_{\hat{\xi}_r}$%	Leg$_5$ $\sigma_{\hat{\xi}_r}$%	Leg$_6$ $\sigma_{\hat{\xi}_r}$%
ZZ_{1R}	0.58	0.64	1.22	1.43	0.84	4.19
MX_1	0.14	0.5	3.35	3.69	1.81	3.04
MY_{1R}	0.26	**0.03**	**0.05**	0.16	**0.12**	0.18
XX_{2R}	**4.18**	**3.93**	0.93	2.01	1.57	3.85
XY_{2R}	1.66	1.15	0.61	3.5	2.07	**4.72**
XZ_{2R}	2.74	2.38	0.88	0.84	**3.17**	3.63
YZ_{2R}	2.04	3.87	0.64	1.42	2.87	1.45
ZZ_{2R}	3.69	0.95	**4.1**	2.01	1.28	0.9
MX_2	1.04	0.2	1.31	0.19	0.41	0.09
MY_2	0.37	0.32	0.34	0.55	0.94	2.05
MX_3	0.17	0.34	0.14	**0.09**	0.56	**0.05**
MY_3	1.55	0.19	0.38	0.2	0.49	0.43
MZ_3	**0.34**	0.32	0.18	**2.31**	0.25	0.31

Table 3.7: Identification results of the 6 legs

The maximum leg relative standard deviation $\sigma_{\hat{\xi}_r}$% is lower than 5% which implies that all parameters are well identified.

3.7 XY - 6 DoF singularity

Parallel mechanisms present remarkable advantages over serial mechanisms in terms of load-carrying capacity, high accuracy, and stiffness. However, they are known to lead to singular configurations in which the number of degrees of freedom of the mechanism changes instantaneously. Singularity is a position where the parallel platform gains or loses degrees of freedom and becomes uncontrollable. Furthermore, the actuator speeds or the actuator forces may become very important which will result in a breakdown of the mechanism. Therefore, it is of primary importance to study the singularities loci in a given workspace to avoid any structural damage and to have better control of the robotic structure Fichter (1980) [65]. This importance can be seen through a huge number of research

works in the field of singular configurations and their neighborhood Laryushkin (2015) [66], Liu (2012) [67], Bandyopadhyay (2006) [68], controller design for avoiding singularity crossing Damien (2016) [69], fastest escape from singular positions Aleshin (2016) [70], by determining of the maximal singularity-free orientation workspace Jiang (2009) [71] and eliminate the singularity using kinematic or actuator redundancy Kotlarski (2012) [72].

The XY - 6 DoF simulator is a parallel mechanism type Gough-Stewart mounted on two series sliders. Thus, in the following sections, we will present the different methods used to find out the singularities for a parallel mechanism.

3.7.1 Parallel mechanisms singularity

Historically, the singularity analysis for a given robotic structure was based exclusively on the analysis of the Jacobian matrix, which is the matrix that connects the joints and the end effector velocities.

The first researchers addressed to singularity issues of parallel mechanism, especially for the Gough-Stewart platform type, was Hunt and Fichter. Hunt Hunt (1978) [73] found that a singularity can occur when all the lines associated with the prismatic actuators intersect a common line. Then, Fichter Fichter (1986) [74] studied the special structure of the general Gough-Stewart platform, and found that another singular configuration can occur when the platform rotates about an axis perpendicular to the base plane by $\pm 90^o$.

Gosselin and Angeles (1990) [75] noticed that since the relationships between joints space and end effector space are implicit, their differentiation leads to two differents Jacobians matrices in this form: $J_q \dot{q} = J_x \dot{x}$ where, $\dot{q}$ is the vector with the active joint rates, $\dot{x}$ is the velocity vector of the end effector, J_x, J_q are the direct and inverse Jacobian matrices, respectively. and according to that, divided the singularities into first and second types by losing or gaining degrees of freedom. With the same definitions, Tsai (1999) [76] terms the first, second, and third kinds of singularities as the inverse, forward, and combined singularities, respectively. The forward kinematic singularities occur when J_x = 0, inverse kinematic singularities occur when J_q= 0 and combined (complex) singularities when both equal zeros.
These classifications have been further refined by Zlatanov, Fenton, and Benhabib (1994) [77], where six types of singular configurations are introduced with detailed physical interpretations. Later on, Zlatanov, Bonev, and Gosselin (2002) [78] discovered another type of singularity, which is called constraint singularities. Ma and Angeles (1992) [79] classify the singularities based on the properties of the Jacobian, into architecture, configuration, and formulation singularities.

It is then easy to identify the locations of singularities within the given workspace and determining whether the singularities can be avoided. In 1998, Kim [80] obtained the analytical expression of the singularity locus equation of the general Gough-Stewart platform with constant orientation using the concept of local structurization method with extra sensors, which is a third-degree polynomial in three-position variables. Later on, Mayer St-Onge and Gosselin (2000) [81] found the same result by expanding the Jacobian matrix of the mechanism with constant orientation using linear decomposition and cofactor expansion. Then Li and Gosselin (2005) [82] shows that the analytical expression for the singularity locus in the six-dimensional Cartesian space can be computed numerically and analytically using the velocity equation of the general Gough-Stewart platform.

Although the various analytical methods have been extensively studied, a general graphical criterion, such as those that are long known for the planar mechanisms, is not known for the 6/6 Gough-Stewart platform. The graphical representation of the Gough-Stewart platform singularities was described for special Gough-Stewart platform constructions or for special singularity cases Hao (1998) [83], Ben-Horin (2009) [84]. If it exists, a graphical criterion could be easily transformed into an algorithm that speeds up the control algorithms so that the control, can be performed in real-time Shai (2009) [85]. The first geometrical or graphical method used to determine the singular configurations of parallel mechanisms and their behavior at these points is based on Grassmann–Cayley algebra. Investigations in the field of line geometry Hao (1998) [83], screw theory and projective geometry Wolf (2003) [86] led to this approach, in which examples of Hunt's and Fichter's and a few selected examples from Merlet's work Merlet (1989, 2000) [87], [48] were analyzed using this method. It describes geometric relations by algebraic methods using a coordinate-free approach.
Merlet (1989, 2000) [87], [48]was one of the first to use this approach especially for 3/6 Gough-Stewart platform namely, the Simplified Symmetric Manipulator, the Minimal Simplified Symmetric Manipulator. For each of these mechanisms, the number of geometric conditions leading to singular configurations is different. The method is general and can be applied to any parallel mechanism. Ben-Horin and Shoham (2006) [88], [89] developed this approach to the general 6/6 Gough-Stewart platform case and to various other special cases. It is worth noting that this method also leads to the Jacobian matrix.

The dual nature of a singularity position as a configuration where the robot not only acquires additional degree of freedom but also possesses inner forces led some authors to investigate kinematics Laryushkin (2015) [66], Gosselin (1990) [75] and other authors to investigate statics Laryushkin (2015) [66], Bandyopadhyay (2006) [68], Hao (1998) [83], in order to determine singular configurations. In the statics analyzes approach Slavutin (2018) [90] has a great result who introduces the three-dimensional (3D) dual Kennedy theorem in statics, and demonstrates its application to characterize the singular configuration of the 6/6 Gough-Stewart Platform which represent another geometrical approach.

Finally, we need to mention an analysis based on the rigidity matrix, which is applicable to bar-and-joint structures. This method is extensively used in rigidity theory. The singularity condition, which was called a pure condition Downing (2002) [91], is the vanishing of the determinant of the rigidity matrix.

From a design point of view, it is desirable to obtain the analytical expression of the singularity locus of a mechanism. Then, with a given set of design parameters, the designers can obtain a graphical representation of the singularity locus of the mechanism. A singularity definition summary can be found in Fig. 3.16.

3.7.2 XY- 6 DoF singularity analysis

The workspace can usually be defined by two workspace types: reachable and dexterous workspace. The reachable workspace is defined as the region within which every point can be reached by the upper platform in at least one orientation. That is, only points where a solution of the inverse displacement problem exists are included. In the Fig. 3.17a and Fig. 3.17b, we present the translation and the rotation reachable workspace of the Gough-Stewart platform used in the XY - 6 DoF.

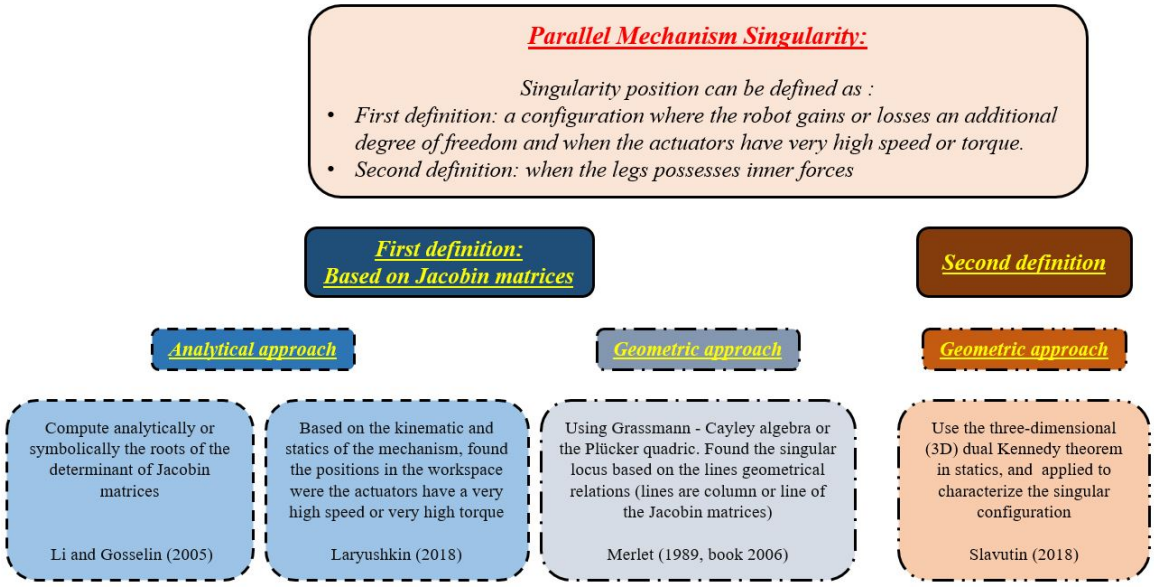
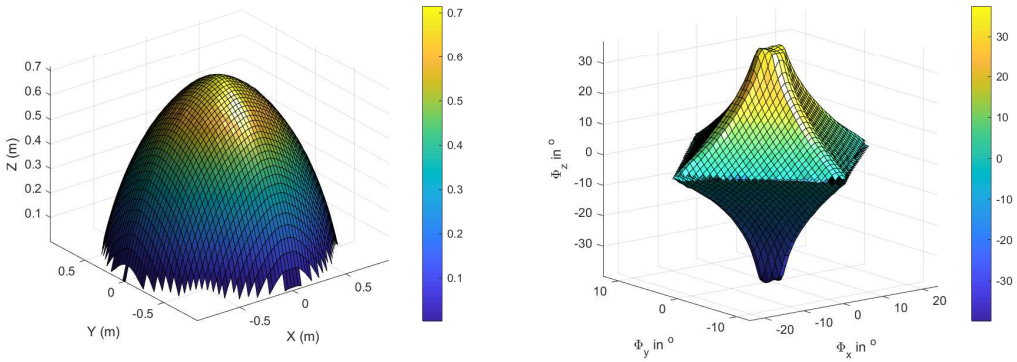


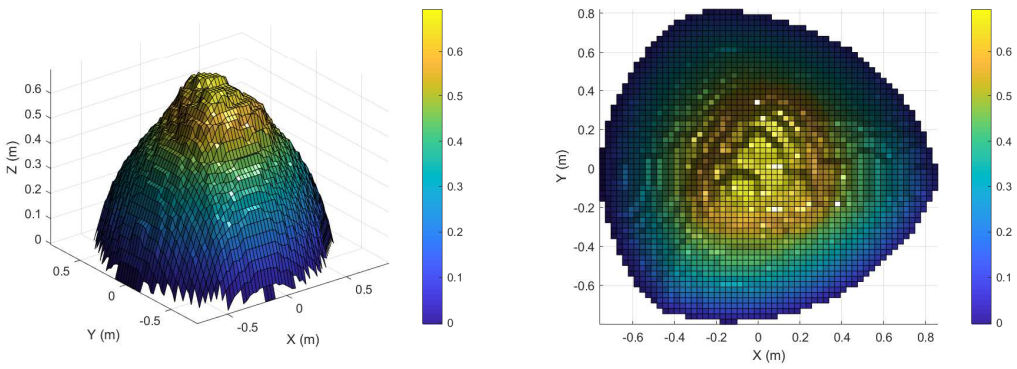
Figure 3.16: Summary of singularity finding methods



(a) Translation workspace *(b) Rotation workspace*

Figure 3.17: Reachable workspace for the Gough-Stewart of the XY-6 DoF

Dexterous workspace is defined as a region of the reachable workspace within which every point can be reached by the upper platform in all possible orientations. In this work, the reachable and dexterous workspaces of the manipulators are obtained by discretizing the region and locating all points within the region that are inside the bounds of the corresponding workspace. The side and top view of the dexterous workspace are presented in the following figure.



(a) Side view *(b) Top view*

Figure 3.18: Dexterous workspace for the Gough-Stewart of the XY-6 DoF

The fact behind the addition of the XY sliding axis is to increase the working space of the parallel robot and to increase the kinematic and dynamic capacity of the robot. Fig. 3.19 presents the dexterous workspace of the XY-6 DoF and shows how the workspace is highly increased. In addition, this use of the XY sliding axis can intervene in the singularity avoidance and this will be present in the following section.

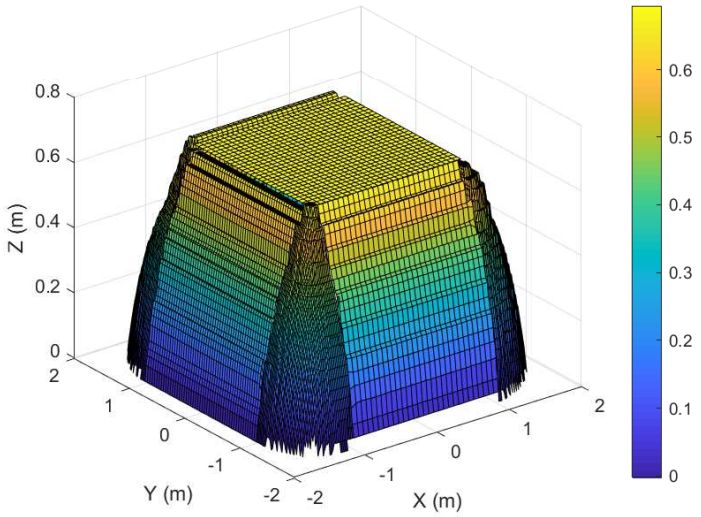


Figure 3.19: Dexterous workspace of the XY-6 DoF

Singularity equations are difficult to obtain for mechanisms with more than three degrees of freedom because the determinant of the Jacobian matrix becomes very complex. For the first time, the singularity equation for the general Gough-Stewart platform was derived in St-Onge (2000) [81]. Actually, this is also a complete singularity equation because the 20 coefficients are functions of the 3 orientation angles. Later, based on the cascaded expansion of the determinant of the Jacobian matrix. A procedure was developed in Li (2005) [82] to obtain the explicit expressions of the 20 coefficients as functions of the orientation angles.

Unfortunately, the singularity equations obtained in St-Onge (2000) [81] and Li (2005) [82] are very complex because they contain too many geometric parameters. Besides, this method lacks flexibility because the considered point P is chosen as the origin O' of the mobile frame. Obviously, for different considered points, différent mobile frames are used.

The singularity equations were also derived in Gregorio (2002) [92] for the general Gough-Stewart platform. To obtain the singularity expression, this work just made the origin O' of the mobile frame coincide with one attachment point (P_i) of the platform. Because of this specific choice, the vector Connecting point O' and point P_i vanishes and the total number of terms of the determinant of the Jacobian matrix reduces from 20 to 10. Otherwise, the number of terms cannot be reduced St-Onge (2000) [81].

Obviously, the singularity equations obtained in Gregorio (2002) [92] is for the special point P_i. Although a singular position of P represents a singular pose of the platform, this special point is not a point of practical interest. In practice, a more interesting point may be the centroid C_p of the platform or any other point chosen according to the application.

To solve this problem, Jiang (2008) [93] proposes to separate the origin O' of the mobile frame from the considered point P and chosen to coincide with

a special point of the platform in order to minimize the number of geometric parameters of the platform. Similarly, the fixed frame should also be chosen to lie at a special point of the base in order to minimize the number of geometric parameters of the base.

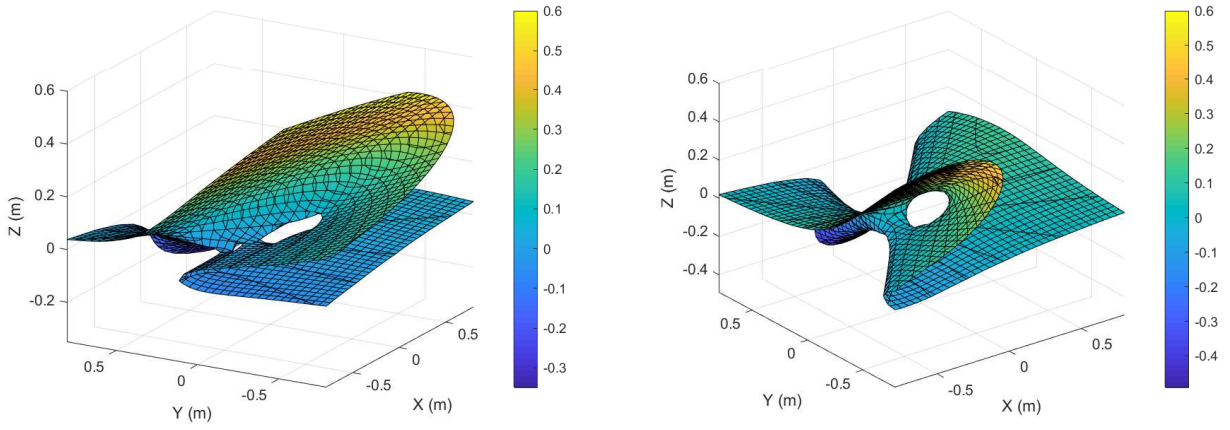
Recall that, the inverse kinematics can be expressed by $J_q\dot{q} = J_x\dot{x}$ and for the parallel robot det(J_x) # 0. By using the condition for the direct singularity for our simulator (J_x= identical matrix, $J_q = {}^{R_b}J_{R_p}^{-1}$), det(${}^{R_b}J_{R_p}^{-1}$) = 0, Jiang (2008) [93] has expressed the singularity equation as follow:

$$\begin{aligned} &f_1x^3 + f_2x^2y + f_3x^2yz + f_4x^2 + f_5y^2x + f_6xyz + f_7xy \\ &+f_8xz^2 + f_9xz + f_{10}x + f_{11}y^3 + f_{12}y^2z + f_{13}y^2 + f_{14}yz^2 \\ &+f_{15}yz + f_{16}y + f_{17}z^3 + f_{18}z^2 + f_{19}z + f_{20} = 0 \end{aligned} \tag{3.7.1}$$

Eq. (3.7.1) is a surface of the third order in three-dimensional space, where the coefficients $f_i,\ i = 1,\ ...,\ 20$ are functions of the geometric parameters and of the orientation angles $\phi_x,\ \phi_y, \phi_z$. Thus, the f_i coefficients can be expressed as follows: $f_i = g(\theta_x, \theta_y, \theta_z)$ and they are explicitly presented in Jiang (2008) [93].

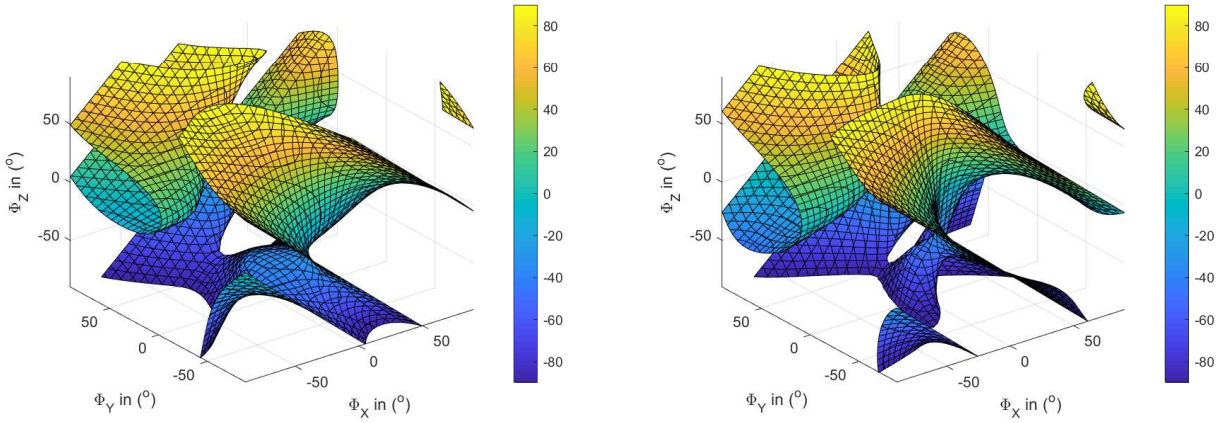
For the Gough-Stewart architecture (Simplified Symmetric Manipulator – Semi-regular hexagons) used in the XY - 6 DoF simulator, and as recommended in Jiang (2008) [93] the reference frame O_{xyz} is attached to the base by selecting the midpoint of line segment B_1B_2 as the origin O and B_1B_2 as the x-axis. The O_{xy} plane coincides with the base plane. The mobile frame $O'_{x'y'z'}$ is attached to the platform by taking the midpoint of line segment P_1P_2 as the origin O' and P_1P_2 as the x' axis. The $O'_{x'y'}$ plane coincides with the platform plane. Hence, the positions of B_i $(i = 1,\ 2,\ ...,\ 6)$ in the fixed frame are respectively $b_1 = [-t_1,\ 0,\ 0]^T$, $b_2 = [t_1,\ 0,\ 0]^T$, $b_3 = [t_2,\ t_3,\ 0]^T$, $b_4 = [t_4,\ t_5,\ 0]^T$, $b_5 = [-t_4,\ t_5,\ 0]^T$ and $b_6 = [-t_2,\ t_3,\ 0]^T$. The number of geometric parameters defining the base is only 5.

Similarly, the position of P_i $(i = 1,\ 2,\ ...,\ 6)$ in the mobile frame are respectively $p'_1 = [-t_6,\ 0,\ 0]^T$, $p'_2 = [t_6,\ 0,\ 0]^T$, $p'_3 = [t_7,\ t_8,\ 0]^T$, $p'_4 = [t_9,\ t_10,\ 0]^T$, $p'_5 = [-t_9,\ t_10,\ 0]^T$ and $p'_6 = [-t_7,\ t_8,\ 0]^T$. The number of geometric parameters defining the mobile platform is also 5. Hence, the total number of geometric parameters for the hexagon used in the XY - 6 DoF is about 10.



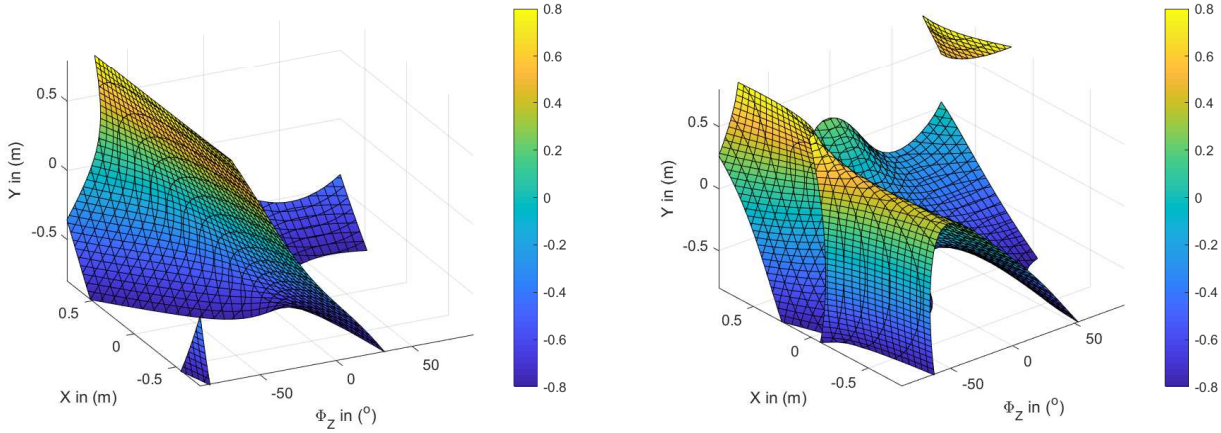
(a) $\phi_x = \phi_z = 25^o,\ \phi_y = 25^o$ (b) $\phi_x = \phi_y = 25^o,\ \phi_z = 0^o$

Figure 3.20: Singularity loci in 3D Cartesian space with constant orientations.



(a) $P_0 = [0.5,\ 0.5,\ 0.5]^T$ *in m* *(b)* $P_0 = [0.1,\ 0.1,\ 0.1]^T$ *in m*

Figure 3.21: Singularity loci in 3D Cartesian space with constant positions.



(a) $\phi_x = \phi_y = 10^o,\ z = 0.2m$ *(b)* $\phi_x = \phi_y = 25^o,\ z = 0.3m$

Figure 3.22: Singularity locus in 3D Cartesian space.

3.7.3 XY- 6 DoF simulator singularity avoidance

The XY-6 DoF hybrid robot will be used as a motion simulator, therefore, there are three recovering movements modes that can be performed: 1. Surge mode: x and ϕ_y, 2. Transverse mode: y and ϕ_x, 3. Vertical mode: z. The vertical mode does not cause any singularity because it is just a movement in the z axis. In what follows, we will study the singularity problem for the surge, transverse and combined modes.

Surge mode

For the surge mode the y, z, ϕ_x, ϕ_z are equal to zero and therefore, the singularity eq. (3.7.1) will be reduced to:

$$\xi_1 x^3 + \xi_2 x^2 + \xi_3 x + \xi_4 = 0 \tag{3.7.2}$$

where, ξ_1, ξ_2, ξ_3 and ξ_4 depend only in ϕ_y.

This equation shows that there is no singularity in the XY-6 DoF in its workspace for the surge mode.

Transverse mode

Similarly, for the transverse mode the x, z, ϕ_y, ϕ_z are equal to zero and therefore, the singularity eq. (3.7.1) will be reduced to:

$$\xi_5 y^3 + \xi_6 y^2 + \xi_7 y + \xi_8 = 0 \tag{3.7.3}$$

where, ξ_5, ξ_6, ξ_7 and ξ_8 depend only in ϕ_x.

Despite the similarity between the surge mode and the transverse mode, the transverse mode presents a certain singularity as shown in the following figure for $x = z = 0$.

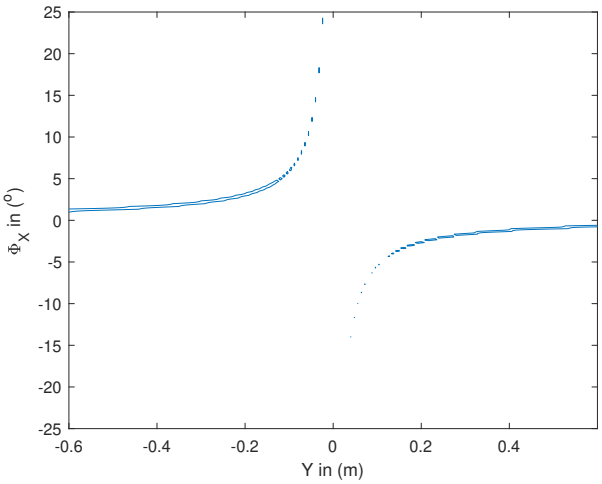


Figure 3.23: Singularity of transverse mode

By increasing the z axis which is constant along the mode the singularity locus follows this equation:

$$\xi_1 y^3 + \xi_2 y^2 + \xi_3 y + \xi_4 yz + \xi_5 y^2 z + \xi_6 yz^2 + \xi_7 z + \xi_8 z^2 + \xi_9 z^3 + \xi_{10} = 0 \tag{3.7.4}$$

where, ξ_i for $i = 1, ..., 10$ depend only in ϕ_x and z is a constant number and it is not a variable. We can found that by increasing the z axis to $0.2m$ all of this singularity can be eliminated, therefore, it suffices to set $z = 0.2\ m$ and perform the transverse mode without problem of singularity.

Combined mode 1: Surge + Transverse mode

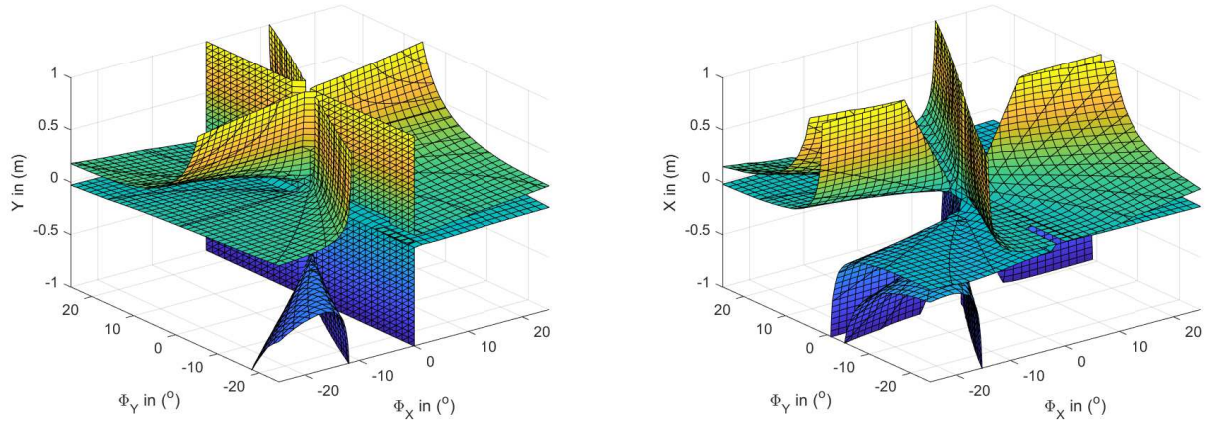
For the surge + transverse mode the z, ϕ_z are equal to zero and therefore, the singularity eq. (3.7.1) will be reduced to:

$$\xi_1 x^3 + \xi_2 x^2 + \xi_3 x + \xi_4 y^3 + \xi_5 y^2 + \xi_6 y + \xi_7 = 0 \tag{3.7.5}$$

where, ξ_1, ξ_2, ξ_3,ξ_4, ξ_5, ξ_6 and ξ_7 depend only in ϕ_x and ϕ_y.

In this mode, we have 4 variables, so we cannot plot the 4 variables together. The next two figures show the singularity locus in 3D Cartesian space for $x = 0$ and $y = 0$, respectively.

It is clear that these figures are not sufficient to define the singularity locus. As presented in the first definition, the XY axis will be used to avoid the singularity in the XY-6 DoF workspace. In a simple way, we can use the x and y of the XY axis



(a) $x = 0\ m$ *(b)* $y = 0\ m$

Figure 3.24: Singularity locus in 3D Cartesian space of Surge and Transverse mode.

to move the upper platform in these directions without any use of the Gough-Stewart platform, and thus the singularity equation will depend only on ϕ_x and ϕ_y and can be written as follows:

$$\xi_1 = 0 \tag{3.7.6}$$

Despite that the singularity equation depends only on rotation, the surge + transverse mode presents some singularity as shown in the following figure for $x = z = 0$.

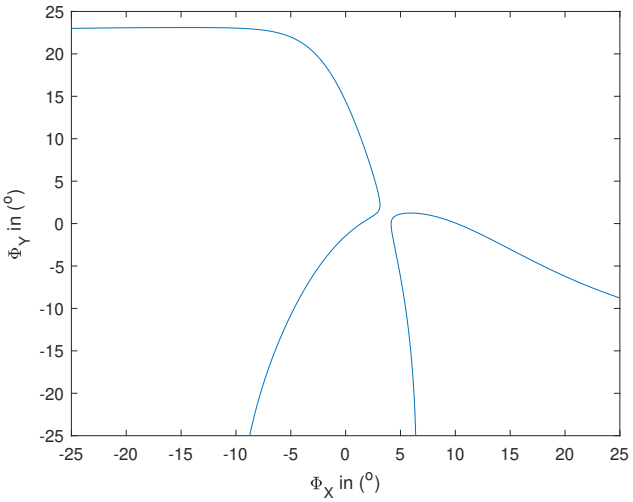


Figure 3.25: Singularity of surge + transverse mode

By increasing the z axis which is constant along this mode the singularity locus follows this equation:

$$\xi_1 z + \xi_2 z^2 + \xi_3 z^3 + \xi_4 = 0 \tag{3.7.7}$$

where, ξ_i for $i = 1, \ldots, 4$ depend only in ϕ_x and ϕ_y and z is a constant number and it is not a variable. We can found that by increasing the z axis to $0.35m$ all of this singularity can be eliminated, therefore, it suffices to set $z = 0.35\ m$ and to move in the x and y axis using only the two sliders. Thus, we can perform the surge + transverse mode without any singularity issue in the defined workspace.

Combined mode 2: Surge + Transverse + Vertical mode

The Surge + Transverse + Vertical mode is the mode that uses the most variables which are x, y, z, ϕ_x,ϕ_y but there is still no need for the ϕ_z, which decreases the DoF used on the parallel platform to 5. By performing all the movements of x and y by the two sliders, the degree of freedom of the parallel robot will be reduced to only 3 variable z, ϕ_x and ϕ_y. The equation for the locus of singularity will be as follows:

$$\xi_1 z + \xi_2 z^2 + \xi_3 z^3 + \xi_4 = 0 \tag{3.7.8}$$

where, ξ_i for $i = 1, ..., 4$ depend only in ϕ_x and ϕ_y. z here is a variable and it is not a constant number. The 3D cartesian plot of the singularity locus is as follow:

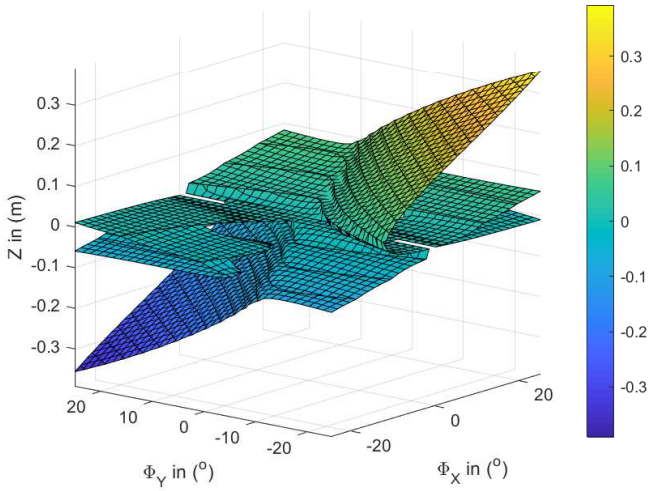


Figure 3.26: Singularity of surge + transverse + vertical mode

We can see that for z more than $0.35\ m$ there is no singularity on the workspace of ϕ_x and ϕ_y. Consequently, it suffices to limit the movement of the axis z from $0.35\ m$ to $0.65\ m$ and to carry out the movements x and y by the two serial sliders to avoid any singularity issue in the defined workspace.

3.8 Conclusion

In this chapter, we have detailed the geometric, kinematic and dynamic modeling of the hexapod type Gough-Stewart platform alone. Then we have developed the appropriate XY- 6 DoF simulator dynamic modeling, which consists of two architectures combination series and parallel using the Newton-Euler recursive relations. A validation of the proposed analytical dynamic modeling through the multi-body dynamic software, SimMechanics, was performed.

In addition, we have presented that the XY-6 DoF simulator depend on 210 standard inertial parameters which are reduced to 90 base inertial parameters only after application of three optimization methods. The given procedure has led to an optimized dynamic/energetic model where the implementation in real time is possible. We experimentally identify the inertia and friction parameters. Obtained torques, experimentally sensed and analytically computed, are comparable. The relative standard deviation shows a very low value for the identified parameters which certify the correct identification.

Furthermore, an analytical singularity study of the XY- 6 DoF simulator hexapod was conducted. We present that the robot redundancy can be exploited to avoid the hexapod singularity mainly for motion simulation application.

At the end of this chapter, we have completely presented the robotic study from modeling to experimental identification [HOUDA1, HOUDA2]. In the next chapter, we will mainly focus on the human sensation problems. Then, how we are going to generate the robot trajectory using the so-called Motion Cuieng Algorithms (MCA) which will be described in detail.

Chapter 4

From Predefined to Desired trajectories based on MCA

4.1 Introduction

The purpose of a motion cueing system, when used in conjunction with a display system, is to stimulate perceptual modalities to the driver. So that he can feel enough indications of movement to pilot the simulator with the same performances as a real one (vehicle, flight or ski) Telban, (2005a) [104]. The reproduction of these indices is therefore the central element of the ski simulation ELLOUMI, (2006) [107]. However, re-establishing real motion to the same is currently impossible due to the inherent limitations of motion simulator mobile platforms. This is why it is necessary to develop techniques which overcome these limitations and make the movement to be reproduced feasible. This is the role of the Motion Cueing Algorithm (MCA), also called " Washout algorithms" Reymond and Kemeny, (2000) [94].

To achieve a compromise between restitution fidelity and respect for the platform physical limits, different platform control strategies have been proposed. We can find classic Shmidt, (1970) [115], adaptive Parrish, (1975) [103], PSO-based Casas, (2018) [127] and optimal Sivan, (1982) [116] MCA approaches. They revolve around three principles, restoring transient movements, tilting the platform for slow movements and returning to a neutral position when the skier's speed is constant. The basic idea is to achieve frequency separation, thanks to dedicated filters, translational accelerations and angular velocities Nehaoua, (2006) [118]. The differences between the different algorithms lie in the way of defining these filters and adjusting the corresponding parameters.

on the one hand, understanding the movement perception mechanism offers key clues for inertial restoring on a ski simulator. In humans, in addition to the eye which receives visual information, the vestibular apparatus is considered to be a set of gravito-inertial sensors responsible for measuring the different quantities of movement Britannica, (2008) [97]. These sensors are divided into two types: linear acceleration sensors, otoliths and angular velocity sensors, semicircular canals . However, the vestibule is unable to detect certain movements if the exciting movement is below perception thresholds Zacharias, (1978) [112]. On the other hand, ambiguities can appear in the motion interpretation, in other words, in the discrimination between a linear translation and a rotation. These imperfections are the basis of the MCAs used in skiing, driving and flight simulation Young, (1969) [111].

We begin this chapter with a physiological and mathematical descriptions of the vestibular system, existing in the literature, responsible for the perception of movement in humans Britannica, (2008) [97], Goldberg, (1984) [117], Young, (1968) [105] and Zacharias, (1978) [112]. Then, the different MCA approaches present in the literature are exposed and applied in detail, classic Shmidt, (1970) [115], adaptive Parrish, (1975) [103], PSO-based Casas, (2018) [127] and optimal Sivan, (1982) [116]. After that, we propose and detail the construction of a four modified MCAs appropriate to the XY-6 DoF simulator. By expanding the definition of MCA from a single use of the hexapod to a shared use between the hexapod and the XY table. Using the XY table will increase the reproduction of MCA indices and improve the restoration of movement in order to boost the human immersive feeling in the virtual environment. Finally, the application of all the MCA approaches from the literature and those proposed on the XY-6 DoF platform is presented and compared.

4.2 Movement perception system in humans

The vestibular system is a set of organs located within the inner ear, and responsible for detecting inertial movements (translational and angular) of the head. It helps maintain balance by intervening in the coordination of reflex movements of the head and neck, trunk and eyes as well as in stabilizing the gaze. Figure 4.1 shows its geometric arrangement relative to the skull. We can distinguish: the otolithic organs (saccule and utricle) indicated by Utr. and Sacculus in the figures Fig. 4.1, Fig. 4.2 and Fig. 4.3 as well as the semicircular canals. These are three channels approximately orthogonal to each other (upper, posterior, lateral). This section is devoted to the study of the vestibular system in the context of the detection of inertial displacements.

4.2.1 Otolithic organs: utricle and saccule

Otolith (Greek: ōto- ear + líthos, a stone) is composed of the utricle and the saccule and they are used to detect linear, horizontal and vertical displacements (respectively). The otolithic organs have fixed positions in the skull. More specifically, the utricle is located on a x_{utr} plane inclined by $20°$ with respect to the horizontal plane and saccules it on a z_{sac} plane inclined by the same angle with respect to the vertical plane Telban, (2000) [95].

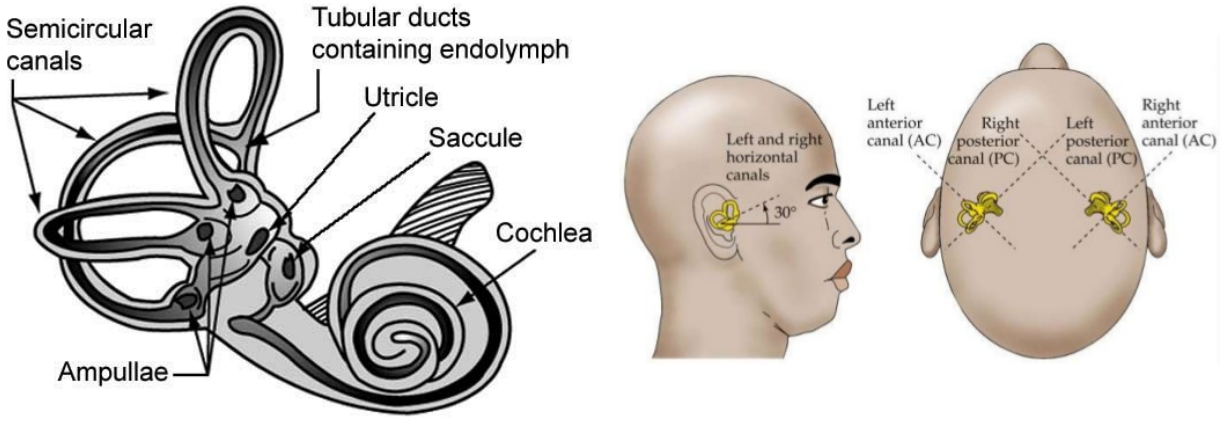


Figure 4.1: The vestibular system (and its position in the skull)

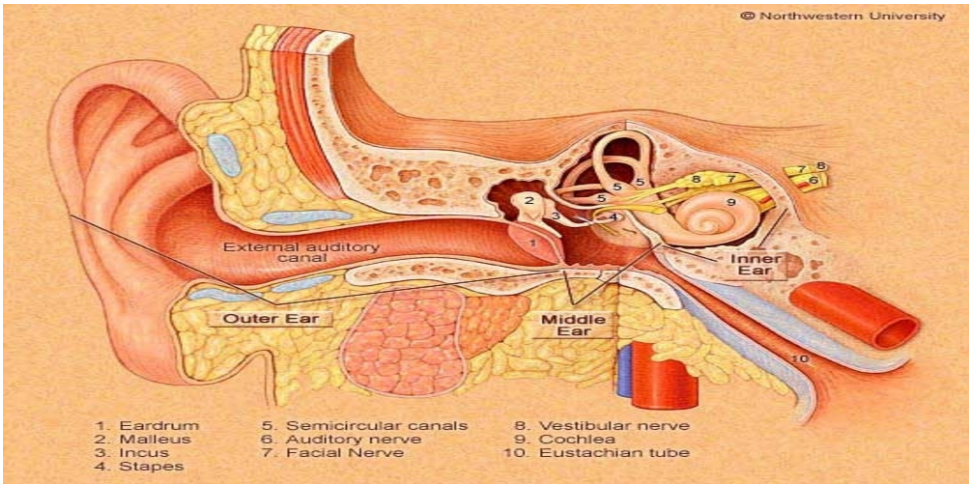


Figure 4.2: Vestibular system location [96]

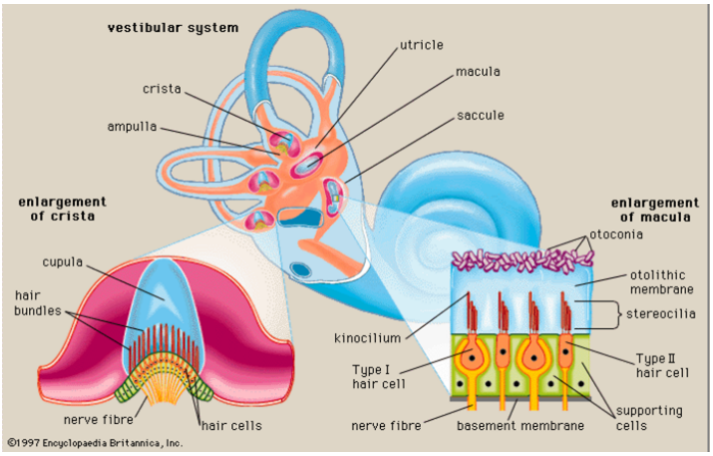


Figure 4.3: Detail of the semicircular canals and of the Britannica otolith macula Britannica, (2008) [97]

The otolithic organs are ovoid cavities filled with a viscous liquid: the endolymph. They possess a two-stage structure (layers):

- The otolithic membrane is composed of an upper layer of otoliths (ear stones which are calcium carbonate concretions bonded by a gelatinous material) and a low density and extremely deformable gelatinous lower layer.
- The macule, the lower layer of sensory cells, is composed of hair cells rooted in a base of support cells attached to the skull (and thus moving with the head) Fernandez, (1976) [98].

According to Berthoz, (1980) [99], the specific stimulus of otolithic organs is the shear force between the two layers of the otolithic membrane. The deformation thus caused of the gelatinous layer will induce the flexion of the hair cells. The operation which transforms the biomechanical signal (bending of the hair cells) into nerve impulses (electrical activity of the hair cells see Fig.4.5) is the transduction mechanism Bohoua, (2004) [100].

When the hair cells tilt towards the kinocil it causes depolarization. The frequency of action potentials (nerve impulses noted N_i and measured in ips: impulse per second) sent to the brain. When the hair cells tilt in the opposite direction, the sensory cells are hyper-polarized and the production of action potential decreases. The variations in polarizations (excitation and inhibition) cause a variation in the frequencies of action potentials of the afferent fibers.

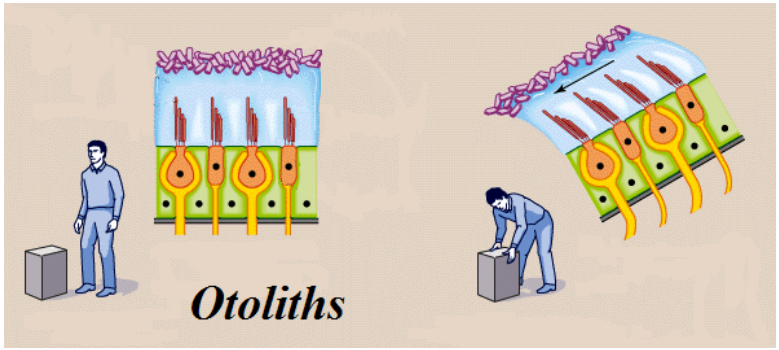


Figure 4.4: Transduction of movement, the resting state vs otolith under a linear movement Britannica, (2008) [97]

The movement of the head and more precisely the combination of linear accelerations and gravity are at the origin of the shear. One of the remarkable properties of otolithic organs is that they do not distinguish between linear acceleration and the component of gravity in the direction of the otolithic organ (caused by a static tilt of the head). This ambiguity will be used in the simulation world to achieve what is called: tilt coordination. In order to fully understand this last point, we will describe in the following the specificity as well as the mechanics of the otolith organs.

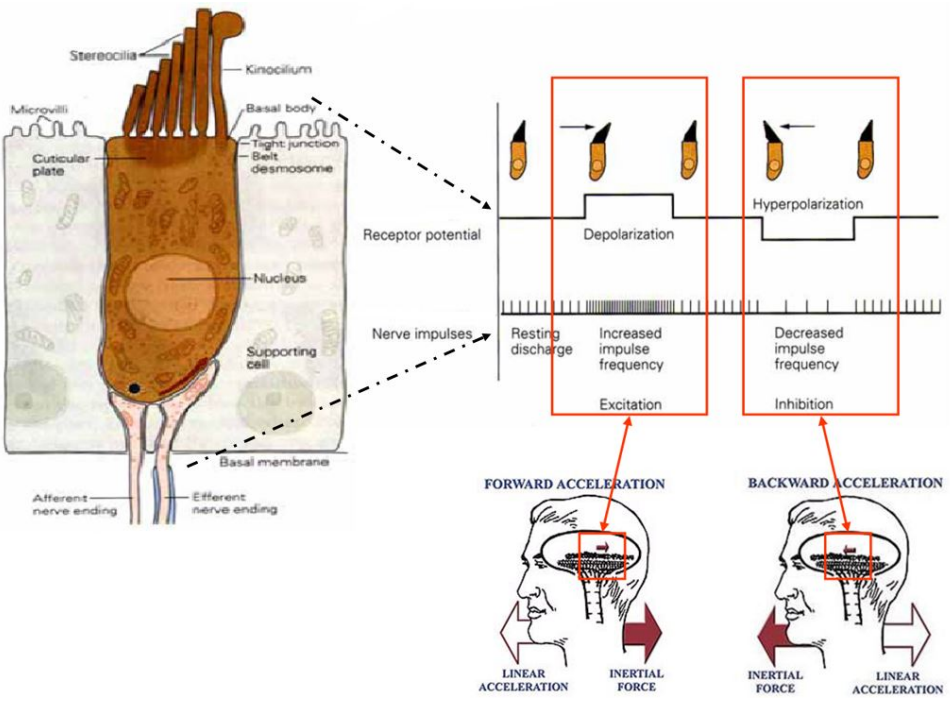


Figure 4.5: Otolithic perception mechanism: from head movement to nerve impulses (in the case of linear acceleration and braking)Redrawn, (1981) [101]

Specificities of otolithic organs

- *Detection limit:* Numerous experiments have demonstrated the presence of a detection threshold for linear accelerations. Let us quote for example, the experiments of linear oscillations reported by Berthoz, (1980) [99]. They consist of moving subjects horizontally and in the dark on a chair with a sinusoidal amplitude force. There are different values for this threshold correponding to the variation of experimental conditions and directions of movement. In addition, this perception threshold seems correlated with the

frequency and amplitude (casilinear dependence) of stimulation. Finally, in order to provide orders of numerical quantities, the perception thresholds of steps of linear acceleration and linear speed can be estimated respectively on average at 5 cm/s^2 and 21 cm/s.

- *The coordination:* (Tilt) coordination Nahon, (1989) [102] is a method inspired by flight simulators which exploits the gravito-inertial ambiguity of otoliths. It consists in performing a very slow rotation (below the canal detection threshold) of the simulation cockpit (and therefore of the skier's head). This rotation is accompanied by the tilting of the virtual reality screen (ski- track scene) in order to guarantee perceptual coherence (see Fig.4.7). Thus the variation in the force of gravity resulting from this rotation will be interpreted as a linear acceleration. This operation can be carried out in two spatial directions: lateral (using the roll angle) and longitudinal (pitch or tilt angle) Parrish, (1975) [103].

The Fig.4.6 schematically shows the head in direct position. We can see the endolymph and the first otolithic layer (formed by the otoliths and the gelatinous layer). Let us define R_0 as a fixed reference attached to the ground with center O_0, R_H as reference frame attached to the head (skull gravity center C_H) and R_{oth} as the reference frame attached to the otoliths center of gravity C_{oth}. Thus, $\overrightarrow{O_0C_H}$ describe the head center displacement and $\overrightarrow{O_0C_{oth}}$ describe the otoliths center displacement. Hence, $\overrightarrow{C_HC_{oth}}$ represents the otoliths displacement relative to the head center.

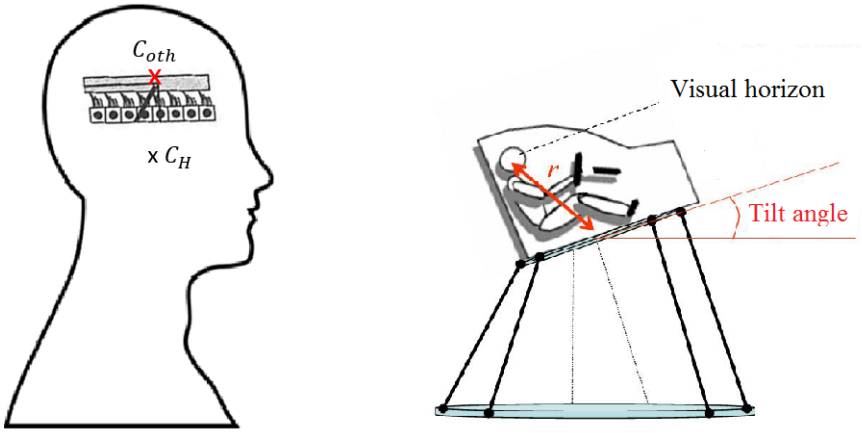


Figure 4.6: Head and otoliths reference frame

To compute the otoliths acceleration in the reference frame R_0, we derive $\overrightarrow{O_0C_{oth}}$ two times with respect to time.

$$\overrightarrow{O_0C_{oth}} = \overrightarrow{O_0C_H} + \overrightarrow{C_HC_{oth}} \tag{4.2.1}$$

$$\frac{d^2}{dt^2}\left(\overrightarrow{O_0C_{oth}}\right) = \frac{d^2}{dt^2}\left(\overrightarrow{O_0C_H}\right) + \frac{d^2}{dt^2}\left(\overrightarrow{C_HC_{oth}}\right) \tag{4.2.2}$$

If the head has only a translational movement, the otolith acceleration is expressed as follows:

$$a_{oth} = a_H + a_{oth/H} \tag{4.2.3}$$

where, a_{oth} is the otoliths translational acceleration in R_0, a_H the head translational acceleration in R_0 and $a_{oth/H}$ is the otoliths translational acceleration with respect to head reference frame R_H.

The Fig.4.7 shows the head in rotation position. In the case of a dynamic rotation of the head, the otoliths acceleration can rewritten as follows:

$$a_{oth} = a_{H,x} + r\ddot{\theta} + a_{oth/H} \tag{4.2.4}$$

where, $a_{H,x}$ is the head acceleration along x-axis, r is the distance between the head and the rotation center and θ is the rotation angle.

The first principle of dynamics provides the following equation (projected along x and normalized by the otoliths mass):

$$a_{oth} = \sum f_{ext} \tag{4.2.5}$$

where $\sum f_{ext}$ is the set of normalized external forces. In R_{oth}, at the level of the otoliths, the head acceleration is considered to be a force: inertial force. The otoliths is then subjected to inertial and gravity forces projected along x axis, therefore,

$$f_x = g_x - a_{H,x} \tag{4.2.6}$$

where, $f_x = a_{oth/H}$ is the otoliths specific force.

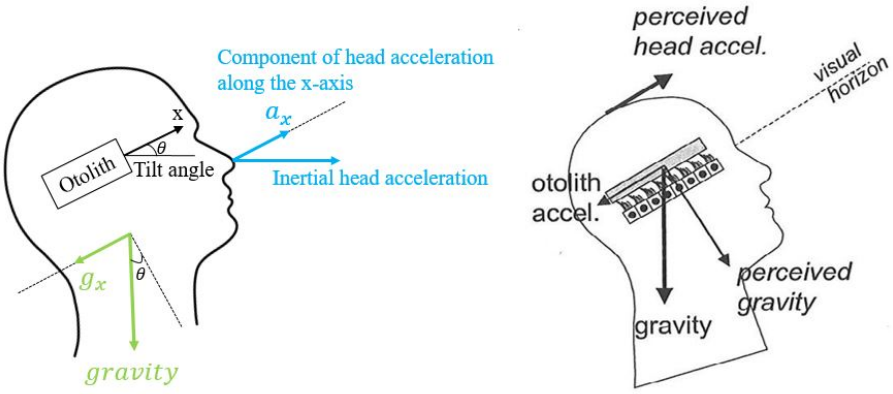


Figure 4.7: Tilt coordination mechanism

Gravity acts equivalent to a linear acceleration of the head. This proves that in a static situation $a_H = 0$, the otoliths signal the presence of an head acceleration equals to $-g_x$. Nevertheless the movement psychological sensation of $-g_x$ will only be reached if the others sensory receptors (eyes, proprioceptive organs, etc.) provide coherent stimulation to the brain (as a generator of sensations). This coherence was concretized within the framework of tilt coordination.

Otoliths transfer function description

The perception of movement is made up of three successive phases. First, the shear movement due to external forces, then the transduction of mechanical movement into a neural electrical signal, and finally the interpretation and transformation of this signal into a Telban and Cardullo sensation Telban, (2005a) [104]. Consequently, developing a perception model consists in mathematically describing these three stages (mechanical, transduction and cerebral). However, we are not going to detail this whole process, what interests us here is the model of the otolith organs which we will use in the next sections.

The dynamics of the otolithic system is assimilated to a mass-spring-shock

absorber system. This dynamic is modeled by the following second order filter:

$$\frac{\hat{v}(s)}{v(s)} = K_{oto}\frac{\tau_1 s}{(1+\tau_1 s)(1+\tau_2 s)} \tag{4.2.7}$$

where $\hat{v}, v$ are respectively the felt linear and stimulating speed. τ_1, τ_2 are two time constants with $\tau_1 >> \tau_2$. Young, (1968) [105] and Meiry, (1965) [106] noted that this model fails to predict the response of otoliths to sustained accelerations. They proposed a revised representation as follows:

$$\frac{\hat{f}(s)}{f(s)} = K_{oto}\frac{\tau_L s+1}{(1+\tau_1 s)(1+\tau_2 s)} \tag{4.2.8}$$

where $\hat{f}, f$ are respectively the specific felt and stimulating force. Recall that the specific force is the difference between the linear acceleration and the vector of gravity $f = a - g$, since at rest, the otolithic organs undergo the vertical acceleration of gravity.

4.2.2 The semicircular canals

The three semicircular canals can be described as rings filled with the endolymph that come together at the surface of the utricle. The plane of the lateral channel makes an angle of approximately $35°$ with the horizontal plane; the anterior and posterior canals make angles of about $40°$ and $50°$ with the sagittal plane. The diameter of the channels is approximately 6 mm and their thickness is 0.24 mm. The semicircular canals are sensitive to the movement of rotation in space (yaw, roll, pitch). In a modeling approach, these angular deviations are conventionally absent.

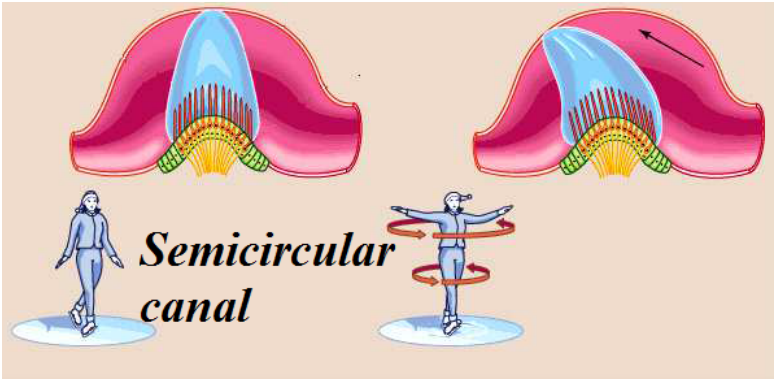


Figure 4.8: Transduction of movement, the resting state vs ampullary crest subjected to a rotational movement Britannica, (2008) [97].

Each canal (Fig.4.8) has a swollen base (bulb) which contains sensory cells and a cup (elastic gelatinous membrane denser than the endolymph). The bulbs also have a receptor area (sensory epithelium) called an ampullary crest similar to the otolith macula. As in the case of otoliths, this zone intervenes in the transformation of the rotational movement of the head into a neural signal. The cup is a gelatinous membrane, attached to the base of the ampullary crest and to the top of the bulb. The ampullary crista or ampullary crista (thick, rounded area) contains the hair cells responsible for neuronal transmission.

When a rotational movement of the head occurs in the plane of one of the channels, (see Fig.4.8 for the horizontal canal) the endolymph undergoes a force

of inertia relative to the labyrinth and deforms the cup. This deformation of the cup will cause the hair cells to flex in the opposite direction to the movement (inertia). Then the fluid continues to rotate in the channel at the same rate as the rotation. However, when the rotation continues at constant speed the cup returns to its equilibrium position (non-linear phenomenon of saturation in neutral position and energy accumulation). Then at the end of the rotation the endolymph-cup system reacts in the opposite direction (energy discharge) to that of the rotation which has ended. Finally the cup returns to its initial position.

Specificities of semicircular canals

- *Detection threshold:* By considering the minimum displacement of the cup, we estimate the detection threshold of the angular acceleration at $0.3°s^{-2}$ for the horizontal channel. For the other channels this threshold is higher at $0.4°s^{-2}$ ELLOUMI, (2006) [107]. The response to a rotation is maximum for a channel when it is located in the plane of rotation. The perception thresholds also depend on the duration of the stimulation.

- *The nystagmus test:* The study of vestibular perception is carried out experimentally, based on psychophysical experiences often coupled with a measure of eye movement Reymond, (2000a) [108]. Indeed, the latter results from a phenomenon known in the world of perception: VOR (Vistibulo-Ocular Reflex) which is a movement of rotation of the eyes in the opposite direction to that of the head even in the absence of visual information. VOR is physiologically associated with the optokinetic reflex (VOR) which produces a rotational movement of the eyes in the direction of movement of the visual scene, so as to minimize retinal slippage and facilitate the recognition of moving objects. These stereotypical movements are in the form of an optokinetic nystagmus (OKN), comprising alternating slow tracking phases and rapid phases of refocusing the eye in its orbit Reymond, (2000a) [108].

Semicircular canals transfer function description

A first approximation of the mechanical model of the semicircular channels consists of a second order low pass filter. This filter reflects the behavior of the cup endolympic shear which is assimilated to an overdamped pendulum system Mayne, (1974) [109]. The transfer function of this filter is given by the following written:

$$\frac{\theta_e(s)}{\alpha(s)} = \frac{\tau_1\tau_2}{(1+\tau_1 s)(1+\tau_2 s)} \tag{4.2.9}$$

where $\alpha(s)$ is the angular acceleration that the head undergoes in the considered channel plane, $\theta_e(s)$ is the angular displacement of the endolymph relative to the head, τ_1 and τ_2 are two time constants with $\tau_1 >> \tau_2$. These two time constants are directly related to the moment of inertia of the endolymph and the elasticity of the cup. The authors in Schmid, (1979) [110] have shown that the deflection of the cup ϕ_c and the angular acceleration θ_e are proportionally related by the relation $\phi_c = -a\theta_e$, where a is a constant. Then, Young, (1969) [111] added an operator of adaptation to the pendulum model to better agree with the experimental results. This refined model is expressed as follows:

$$\frac{\phi_c(s)}{\alpha(s)} = K_{csc}\frac{\tau_a}{1+\tau_a}\frac{1}{(1+\tau_1 s)(1+\tau_2 s)} \tag{4.2.10}$$

where τ_a is the adaptation coefficient and $K_{csc} = a\tau_1\tau_2$. From a transduction point of view, the last model turns out to be consistent in describing the related mechanism. Let us define N_i as the Afferent Response (nerve impulses). The different parameters were measured from experiments carried out on monkeys called "Saïmiri" (squirrel monkey). However, an advance term has been suggested to refine the model of neuronal transduction as follows:

$$\frac{N_i(s)}{\alpha(s)} = K_{csc}\frac{\tau_a}{1+\tau_a}\frac{1+\tau_L s}{(1+\tau_1 s)(1+\tau_2 s)} \tag{4.2.11}$$

Finally, frequently, the semicircular canals seem to have a bandpass dynamic. They have a high sensitivity for angular velocity between $11.45°s^{-1}$ and $572.95°s^{-1}$ Telban, (2005a) [104]. They have the same dynamics for the perception of angular acceleration but with a low frequency bandwidth between $0.26°s^{-2}$ to $79.64°s^{-2}$ Telban, (2005a) [104]. The perception of jerk, derived from acceleration, would have a low-pass dynamic beyond $103.7°s^{-3}$. For these reasons, in our next developments, we will use the following angular velocity model of the semicircular canals:

$$\frac{\hat{\omega}(s)}{\omega(s)} = \frac{K_{scc}\tau_1\tau_a s^2(1+\tau_L s)}{(1+\tau_a s)(1+\tau_1 s)(1+\tau_2 s)} \tag{4.2.12}$$

where $\hat{\omega}$ is the felt angular velocity, ω is the angular velocity of the head.

In summary, we show in the following figure the non-detectable, detectable and high sensitivity detectable movement of the vestibular system, for both movement rotation and translation. The platform's translation and rotation capacity covers the three types of movement, which allows us to perform a highly sensitive movement as well as an undetectable movement to perform the washout.

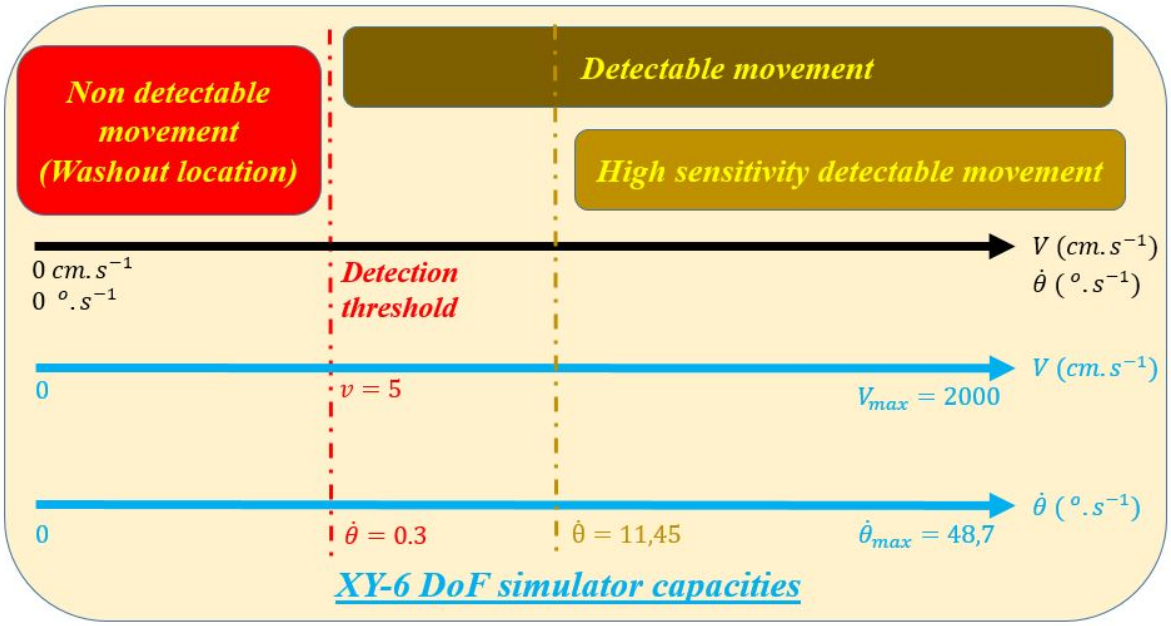


Figure 4.9: XY-6 DoF simulator capacities

4.2.3 Vestibular system state space representation

The longitudinal, transverse and vertical models are decoupled and the development of the algorithms for each direction is similar, thus we present below

the algorithm for the longitudinal mode (pitch/surge mode). Then it is exactly the same for the transverse mode (translation about y-axis and rotation about x-axis) and vertical mode. However, for the vertical mode, there is no rotation movement, it consists only of translation about the z-axis. We begin first, by the elaboration of the otolith state space representation alone, then for the semicircular canals alone and in the end we develop for the vestibular system combining the two models.
The otolith transfer function (4.2.8) can be rewritten as:

$$\frac{\hat{f}_x(s)}{f_x(s)} = \overline{K}_{OTO}\frac{(s+\frac{1}{\tau_L})}{(s+\frac{1}{\tau_1})(s+\frac{1}{\tau_2})} \tag{4.2.13}$$

where, $\overline{K}_{OTO}$, τ_1, τ_2, and τ_L are defined as the computed parameters for the static sensitivity, long time constant, short time constant, and lead operator, respectively. $\overline{K}_{OTO} = K_{oto}(\frac{\tau_L}{\tau_1\tau_2})$ and $f_x(s) = \ddot{x}(s) + (g\frac{1}{s})\dot{\theta}_y$. Let us define the input as $u = [u_1,\ u_2]^T \triangleq [\dot{\theta}_y,\ \ddot{x}]^T$. Therefore, $f_x(s) = u_2(s) + (g\frac{1}{s})u_1(s)$.
Substituting $f_x(s)$ into (4.2.13) and introduce the Laplace inverse transformation leads to:

$$\dddot{\hat{f}}_x + (\frac{1}{\tau_1} + \frac{1}{\tau_2})\ddot{\hat{f}}_x + \frac{1}{\tau_1\tau_2}\dot{\hat{f}}_x = \overline{K}_{OTO}\left(g u_1 + g\frac{1}{\tau_L}\int u_1 dt + \dot{u}_2 + \frac{1}{\tau_L}u_2\right) \tag{4.2.14}$$

Equation (4.2.14) can be rewritten as:

$$\dddot{\hat{f}}_x + \alpha_1\ddot{\hat{f}}_x + \alpha_2\dot{\hat{f}}_x = \alpha_3\dot{u}_1 + \alpha_4 u_1 + \alpha_5\int u_1 dt + \alpha_6\dot{u}_2 + \alpha_7 u_2 \tag{4.2.15}$$

where,

$$\alpha_1 = \frac{1}{\tau_1} + \frac{1}{\tau_2};\ \alpha_2 = \frac{1}{\tau_1\tau_2};\ \alpha_3 = 0;\ \alpha_4 = g\overline{K}_{OTO};\ \alpha_5 = \frac{g\overline{K}_{OTO}}{\tau_L};\ \alpha_6 = \overline{K}_{OTO};\ \alpha_7 = \frac{\overline{K}_{OTO}}{\tau_L}$$

Let us consider the following state vector $x_{OTO} = [x_{oto1},\ x_{oto2},\ x_{oto3}]^T$:

$$x_{oto1} = \hat{f}_x \tag{4.2.16}$$
$$x_{oto2} = \dot{x}_{oto1} - \alpha_3 u_1 - \alpha_6 u_2 \tag{4.2.17}$$
$$x_{oto3} = \alpha_5 u_1 \tag{4.2.18}$$

By taking the derivative state we can express the transfer function (4.2.13) in the state space form as:

$$\begin{aligned}\dot{x}_{OTO} &= A_{OTO}x_{OTO} + B_{OTO}u\\ \hat{y}_{OTO} &= C_{OTO}x_{OTO} + D_{OTO}u\end{aligned} \tag{4.2.19}$$

where x_{OTO}, $\hat{y}_{OTO} = \hat{f}_x$ and u are the otoliths states, felt specific force and the vestibular system input, respectively, and A_{OTO}, B_{OTO}, C_{OTO}, and D_{OTO} are given as:

$$A_{OTO} = \begin{bmatrix} 0 & 1 & 0\\ -\alpha_2 & -\alpha_1 & 1\\ 0 & 0 & 0\end{bmatrix};\ B_{OTO} = \begin{bmatrix} \alpha_3 & \alpha_6\\ \alpha_4 - \alpha_1\alpha_3 & \alpha_7 - \alpha_1\alpha_6\\ \alpha_5 & 0\end{bmatrix}$$

$$C_{OTO} = \begin{bmatrix}1 & 0 & 0\end{bmatrix};\ D_{OTO} = \begin{bmatrix}0 & 0\end{bmatrix}$$

The semicircular canals model is given as Zacharias, (1978) [112]:

$$\frac{\hat{\omega}}{\omega} = \frac{K_{scc}\tau_1\tau_a s^2(1+\tau_L s)}{(1+\tau_a s)(1+\tau_1 s)(1+\tau_2 s)} \tag{4.2.20}$$

where $\hat{\omega}$ is the felt angular velocity pertaining to each of the three axes, and τ_1, τ_2, τ_L, and τ_a are the long time constant, short time constant, lead term, and adaptation operator constant, respectively. The transfer function in 4.2.20 can be rewritten as:

$$\frac{\hat{\omega}}{\omega} = \frac{\xi_4 s^3 + \xi_3 s^2}{s^3 + \xi_2 s^2 + \xi_1 s + \xi_0} \tag{4.2.21}$$

where,

$$\xi_0 = \frac{1}{\tau_a\tau_1\tau_2};\ \xi_1 = \frac{\tau_a+\tau_1+\tau_2}{\tau_a\tau_1\tau_2};\ \xi_2 = \frac{\tau_1\tau_2+\tau_a(\tau_1+\tau_2)}{\tau_a\tau_1\tau_2};\ \xi_3 = \frac{K_s cc}{\tau_2};\ \xi_4 = \frac{K_{scc}\tau_L}{\tau_2};$$

Let us take the semicircular states as $x_{SCC} = [x_{SCC1},\ x_{SCC2},\ x_{SCC3}]^T$

$$x_{SCC1} = \hat{\omega} \tag{4.2.22}$$
$$x_{SCC2} = \dot{x}_{SCC1} - \xi_2 x_{SCC1} - (\xi_3 - \xi_2\xi_4)u_1 \tag{4.2.23}$$
$$x_{SCC3} = \dot{x}_{SCC2} + \xi_1\dot{x}_{SCC1} + \xi_1\xi_4 u_1 \tag{4.2.24}$$

The transfer function in (4.2.21) can be expressed in LTI (Linear Time Invariant) form as:

$$\begin{aligned}\dot{x}_{SCC} &= A_{SCC}x_{SCC} + B_{SCC}u \\ \hat{y}_{SCC} &= C_{SCC}x_{SCC} + D_{SCC}u\end{aligned} \tag{4.2.25}$$

where $\hat{y}_{SCC} = \hat{\omega}_{SCC}$ the felt rotation speed and its canonical observer matrices is:

$$A_{SCC} = \begin{bmatrix} -\xi_2 & 1 & 0 \\ -\xi_1 & 0 & 1 \\ -\xi_0 & 0 & 0 \end{bmatrix};\ B_{SCC} = \begin{bmatrix} \xi_3 - \xi_2\xi_4 & 0 \\ -\xi_1\xi_4 & 0 \\ -\xi_0\xi_4 & 0 \end{bmatrix};\ C_{SCC} = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix};\ D_{SCC} = \begin{bmatrix} \xi_4 & 0 \end{bmatrix}$$

The state space representation of the human vestibular system, which consists of both semicircular canal and otolith organ models, is as follows:

$$\begin{aligned}\dot{x}_V &= A_V x_V + B_V u \\ \hat{y}_V &= C_V x_V + D_V u\end{aligned} \tag{4.2.26}$$

where x_V contains the human vestibular model states, $\hat{y}$ is the felt output representation, and A_V, B_V, C_V, and D_V are:

$$A_V = \begin{bmatrix} A_{SCC} & 0 \\ 0 & A_{OTO} \end{bmatrix};\ B_V = \begin{bmatrix} B_{SCC} \\ B_{OTO} \end{bmatrix};\ C_V = \begin{bmatrix} C_{SCC} & 0 \\ 0 & C_{OTO} \end{bmatrix};\ D_V = \begin{bmatrix} D_{SCC} \\ D_{OTO} \end{bmatrix}$$

The vestibular system state space representation will therefore be used in the filter development for the MCA approaches. Especially, for the optimal approach based on LQR method where a vestibular state space representation is crucial. Furthermore, the vestibular system is used to calculate human specific force and angular velocity senstation. The human movement sensation will then be used to compare MCAs.

4.3 Motion Cueing Algorithm

The restitution of the movement sensations (inertial or vestibular indices) has a primary importance in the ski simulation and more particularly in the simulation of rapid movements such as accidents and road-related disturbances. Unfortunately, the intrinsic limitations of the movement mechanisms of simulators make it impossible to reproduce these indices identically. However, partial restitution is possible. This is the role of Motion Cueing Algorithms (MCA) also known as *"washout algorithms"*.

From our knowledge, no exact solution has been given to this problem. Indeed, the current level of understanding of the perception system does not allow vestibular clues to be reproduced identically. However, several heuristic methods have been developed. We designate them by MCA.

The role of MCA is to transform the real trajectories of a real (vehicle, flight or ski) into restricted trajectories achievable by the simulator. This movement must not only satisfy a physical validity (limitations of the simulator) but also a perceptual validity; sensations of movement induced by the simulator which must verify notions of coherence and proximity with those produced in reality). As it is described in the vestibular section, the vestibular system is sensitive to linear accelerations and angular velocities. These quantities are therefore the most relevant to define a trajectory in the MCA context.

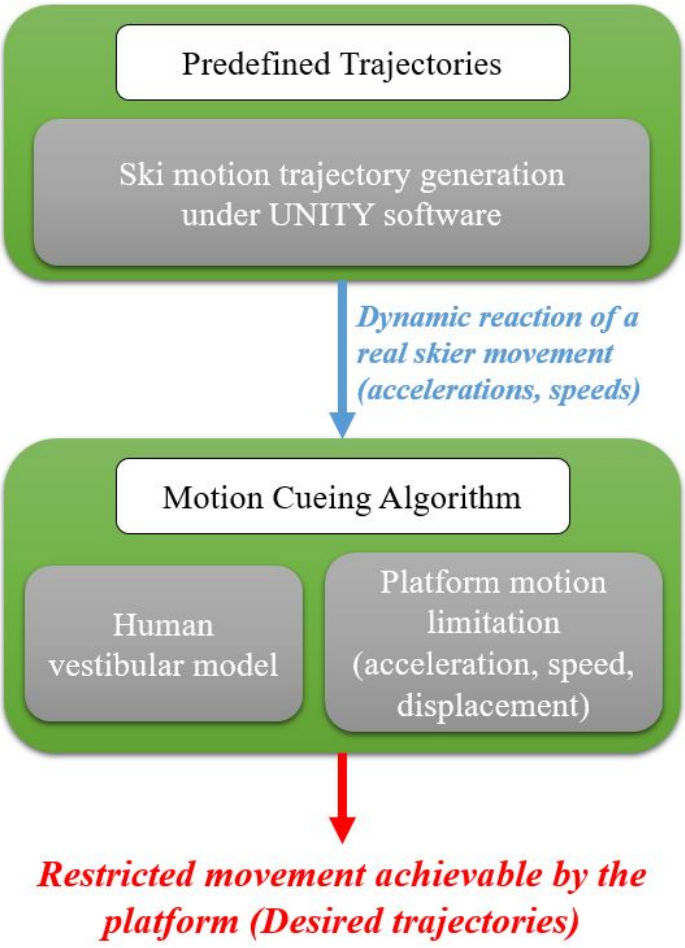


Figure 4.10: From predefined to desired trajectories

A simulator diagram presented in Fig. 4.10 illustrates the generation of a restricted movements achievable by the platform from the predefined trajectories. The two steps are:

1. The model of a real skier allows to calculate the reaction of the real skier following the predefined trajectories. This response is a set of dynamic data: accelerations and speeds characterizing its movement (which can be extracted from *Unity platform* or planned manually).

2. The MCA causally transforms these real accelerations and speeds into accelerations and speeds achievable by the simulator satisfing a perceptual validity.

A great improvement on our new simulator is a so-called XY-table, which can perform larger linear movements in the longitudinal and lateral dimensions. The hexapod is mounted on top of this table and therefore the human operator is under both movements. Such simulators have in total 8 DoF. The idea is to provide a more realistic experience so that the user cannot distinguish the real situation from that simulated Miunske, (2019) [7]. The main advantage of simulators with XY-table is that the movements can now be distributed with the XY-table to help linear or tilting movements of the hexapod. In addition, highly dynamic moves, such as creating curves in the ski application, can be reproduced more realistically. The large workspace offered by the XY-table makes the restoration of acceleration and deceleration in longitudinal movement more adequate and realistic Asadi, (2015) [113], Zhiyong, (2016) [114].

Physically, MCAs are based on the concept of frequency separation of linear accelerations and angular velocities, and take advantage of the imperfections of the vestibular sensors and the central nervous system in the interpretation of certain movements to lure the skier. For frequency separation, high pass and low pass filters are introduced. Most of the proposed approaches differ mainly in the form and the optimization scheme used for the implementation of these filters. In this section, we propose a summary of the different MCAs used for flight or driving simulators. The following paragraph shows the techniques that will be detailed later on:

- The MCA classic approach (see Shmidt, (1970) [115]), the coefficients of the different filters are constant and fixed at the start of the driving simulation.The classical approach has the easiest implementation compared to another approach, however, its performance is not optimal because it is adjusted to the "worst case".

- The MCA adaptive approach (see Parrish, (1975) [103]), allows the parameters of the filters to be adapted by real-time minimization of a cost function reflecting the compromise between restoring accelerations and respecting mechanical constraints. for the adaptive approach, the coefficients of the various filters are adjusted for a specific driving situation instead of a worst-case setting which make it better than classic filter, however, the adaptive approach does not guarantee the respect of the perception thresholds.

- The MCA PSO-based approach (see Casas, (2018) [127]), Particle Swarm Optimisation (PSO) is used as a population based heuristic search method for searching an optimal solution in a complex and large space. PSO is used to tune the MCA classic filter parameters in order to achieve optimum output.

- The MCA optimal approach (see Sivan, (1982) [116]), follows the same philosophy, the filters are optimized off-line or in real-time by minimizing a criterion which directly integrates mechanical and perceptual constraints. However, these filters are optimized for a family of statistical trajectories,

therefore, when the simulation acceleration differs from the one chosen, the optimality is no longer assured.

In the following sections, we detail the predefined trajectory constuction. Then, we develop the MCA algorithms presented above from the literature. After, we propose an improvement for each MCA approach to be adaptable to our redundant robotic simulator and dedicated to disabled people for a sliding application. In the end, we define a performance indicator to show and compare the desired trajectory validity generated by each improved approach.

4.3.1 Predefined trajectory

First, in order to be able to define predefined trajectories, we sketch in Fig.4.11 the skier movement in different directions (whether it is the real skier or the simulation cockpit). We will also use a, v and ω to represent $\ddot{x}$, $\dot{x}$ and $\dot{\theta}$, respectively. Finally we note $g = 9.81 m.s^{-2}$ is the gravity constant.

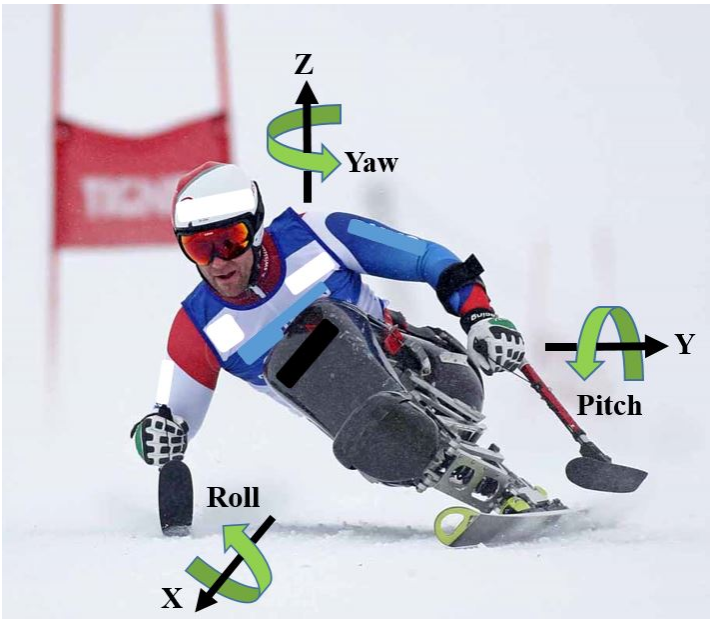


Figure 4.11: The space directions

Below is the list of variables describing the simulation cockpit movement (or the skier) shown in Fig. 4.11.

- Translation: longitudinal "x", lateral "y" and vertical "z".
- Rotation: roll "θ_x", pitch "θ_y" and yaw "θ_z".

In the MCA context, the whole movement is broken down into three modes, which are Telban, (2005a) [104]:

1. Mode 1: Longitudinal mode consist of a translation along $x-$axis and a rotation about y, then it can be represented by two parameters x and θ_y.
2. Mode 2: Lateral or transverse mode consist of a translation along $y-$axis and a rotation about x, then it can be represented by two parameters y and θ_x.
3. Mode 3: Vertical mode consist only by a translation along $z-$axis and then depend only on one parameter z.

For ski application, the longitudinal mode takes place always when the skier is moving down. The transverse mode occurs when the skier moves right and left in the $y-$ direction which is known as ski carving. And the vertical mode appears when the skier has no contact with the ground due to bumps or in ski jumping.

Note that when planning a trajectory, attention should be paid to the linear acceleration and velocity thresholds as well as the angular acceleration threshold. Recall that, $a_{th} = 5cm.s^{-2}$, $v_{th} = 21cm.s^{-1}$ and $\ddot{\theta}_{th} = 0.3°.s^{-2}$. The trajectories will be generated corresponding to the NaO robot and its seat 7 Kg (for security reason). The friction coefficient between snow and ski is about 0.15, the slope (deflection angle) is about $15°$. Recall that, R_0 the fixed reference attached to the ground and R_H is the reference attached to the user head.

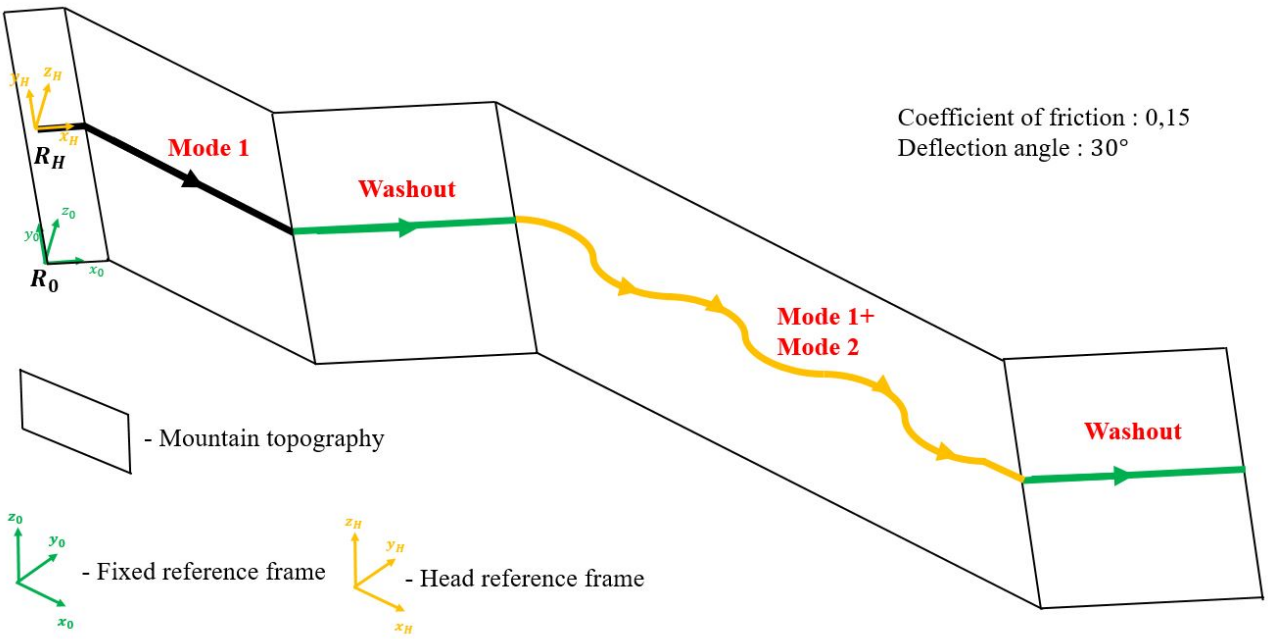


Figure 4.12: Predefined trajectory

Fig. 4.12 presents the different trajectory modes. A movement model made up of several ski-specific submodels has been modeled. The predefined trajectory of the skier consists at first in a descending movement, mode 1. After that, we manage to have a constant slope in order to get the possibility to do the washout. The washout is done to be able to perform another trajectories modes such as mode 1+ mode 2. In the following figures we show the skier specific force, speed and displacement in $x-$direction.

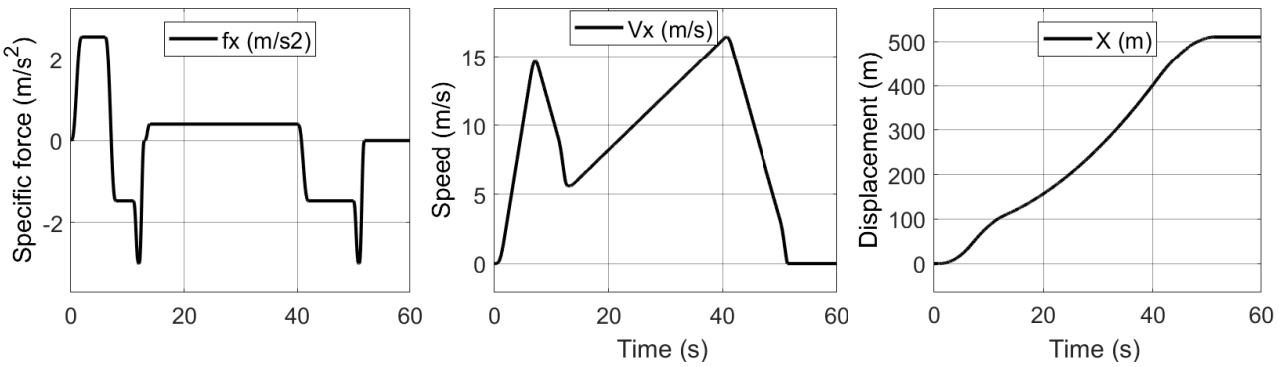


Figure 4.13: Specific force, speed and displacement behaviors in $x-$direction

The skier have a max speed of $16.8\ m/s$ which correspond to $60.5\ km/h$ and move

almost 500 m in the down direction. The felt acceleration is computed from human vestibular system described in section (4.2). The filters parameters are as follows:

Model	Filters Parameters
eq. 4.2.11	$K_{csc} = 3.44, \tau_a = 80, \tau_L = 0.049 \tau_1 = 5.7 \tau_2 = 0.003$ Goldberg, (1984) [117]
eq. 4.2.12	$K_{csc} = 3.44, \tau_a = 80, \tau_L = 0.006 \tau_1 = 5.73 \tau_2 = 0.005$ Zacharias, (1978) [112]
eq. 4.2.8	$K_{oto} = 0.4, \tau_L = 13.2 \tau_1 = 5.33 \tau_2 = 0.66$ Young, (1968) [105]

Table 4.1: Identification of the semicircular canals and otoliths parameters

The following figure shows the predefined specific force trajectories in x and y directions with the felt translational acceleration.

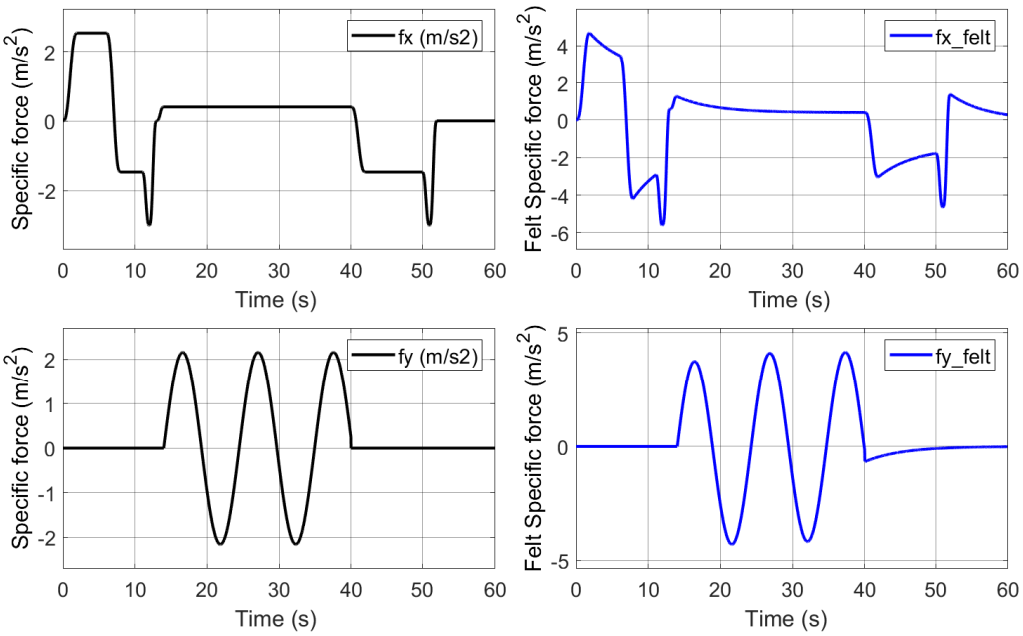


Figure 4.14: Predefined felt specific force trajectory

At the end we present the XY curve motion for Mode 1 + Mode 2.

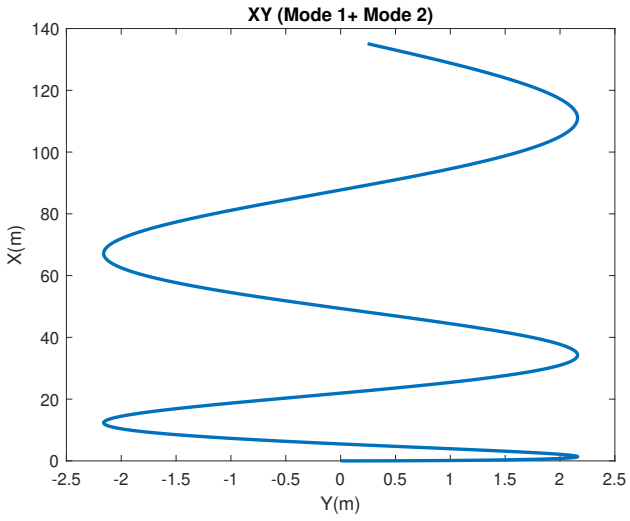


Figure 4.15: XY curve trajectory

4.3.2 Desired trajectory based on classic MCA

The classic MCA is based on the frequency separation algorithm, where the predefined ski acceleration signals are usually separated into two parts. The simulator (Hexapod) can generate short-term high accelerations via translational movements, whereas static accelerations are restored by tilting - also called tilt coordination explained in Section. (4.2).

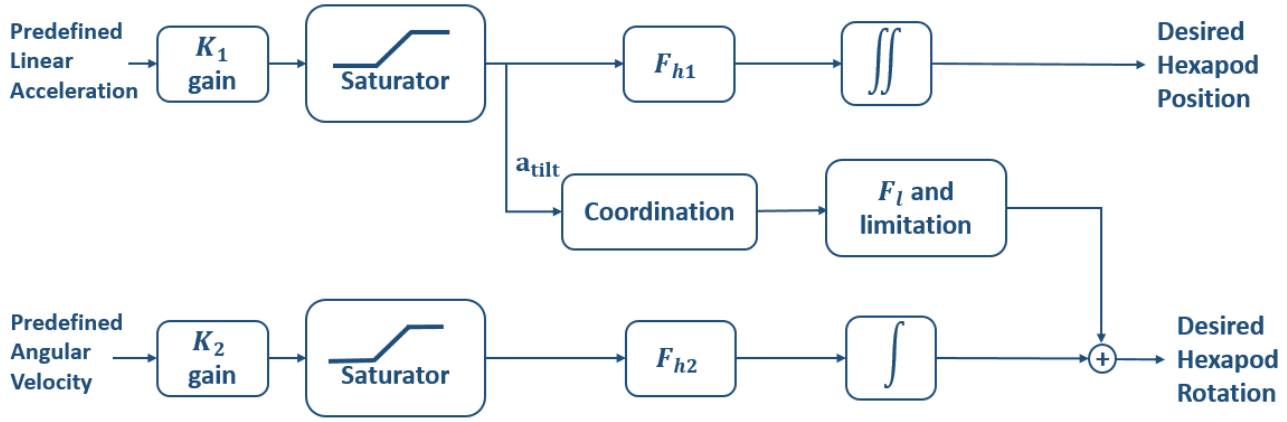


Figure 4.16: Classic algorithm for the longitudinal mode

The structure of this approach in the particular case of the longitudinal mode (surge: x and pitch: θ_y) is as follows Nehaoua, (2006) [118]:

$$x = \left(\frac{1}{s^2} F_{h_1}\right) a^{pd} \tag{4.3.1}$$

$$\theta_y = \left(\frac{1}{s} F_{h_2}\right) \omega^{pd} + \left(\frac{1}{g} F_l\right) a^{pd} \tag{4.3.2}$$

where F_{h_1}, F_{h_2} are high-pass filters while F_l is a low-pass filter and a^{pd}, ω^{pd} are the predefined skier linear acceleration and angular velocity, respectively.
In the Fig. 4.17, we show the restored part of the predefined acceleration by the hexapod position and by the tilt coordination.

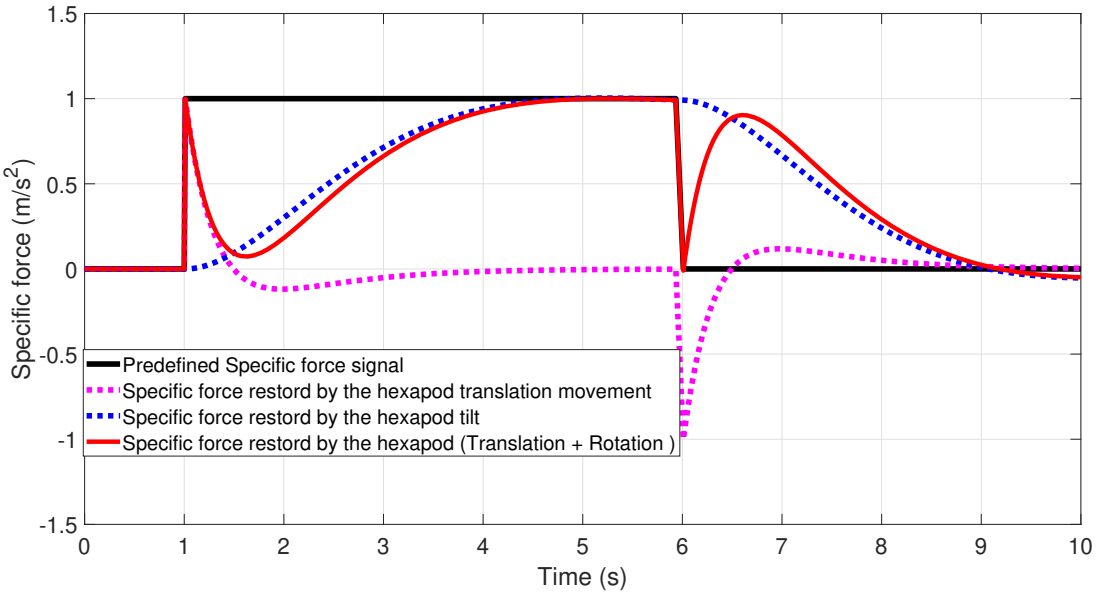


Figure 4.17: Classic washout algorithm applied on the XY- 6 DoF (IBISC)

The adjustment of the parameters as well as the shape of the different filters are of crucial importance, in order to maintain maximum perceptual consistency, but also to reduce the resulting delays. In other words, the frequency response of these filters must be unitary in the bandwidth, while the attenuation must be infinite outside. However, such a filter is not feasible because it is not causal, hence the need to arbitrate various compromises.

The questions that arise are: is the XY platform capable of reproducing all the lack of acceleration? If not, how the XY-table should act to get better acceleration restitution according to our main application?
The XY- table should recover the remaining acceleration after the hexpod recovering. The lack of acceleration, speed and displacement that should be recovered by the XY- table are presented in Fig. 4.18. This figure shows that if all the lack of acceleration is to be recovered, the XY table should have a dimension of 8m. which is higher than our XY- table limits (1.1 m).

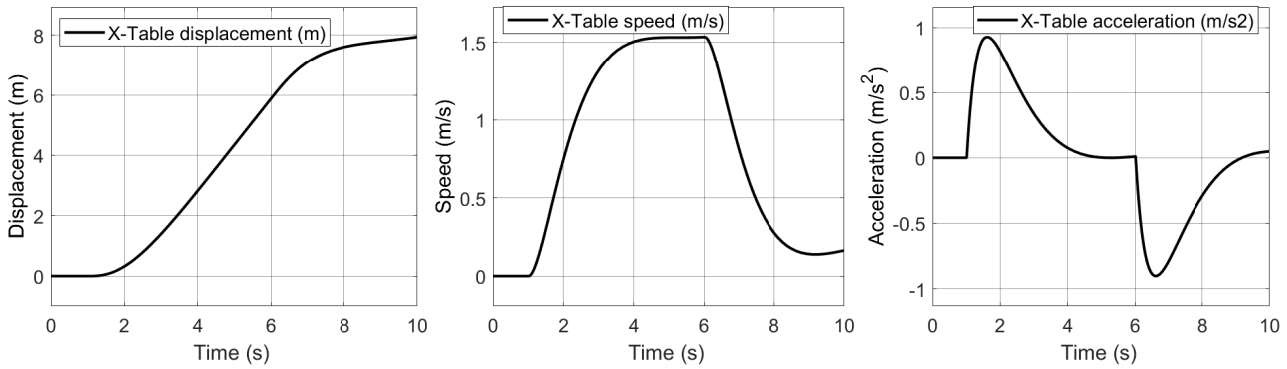


Figure 4.18: The required displacement, speed and acceleration of the XY-table to recover all of acceleration lack

Our XY- table dimension is not sufficient to recover all the acceleration lack. To overcome this limitation, two choices are proposed to use the XY table in order to have the best movement sensations. Recall that, for the hexapod alone the high frequency acceleration is recovered by translational movement and the low frequency acceleration is recovered by the tilt coordination.

1. The XY-table will help the hexapod translational displacement to recover the high frequency part.

2. The XY-table will help the hexapod tilt to recover the low frequency part.

The highest acceleration value among the predefined ski specific force signals illustrated in Fig. 4.13 is less than $3\ ms^{-2}$. Our simulator hexapod is capable of reproducing this maximum value by a worst-case study (square signal). In addition, our application is an alpine ski with a maximum inclination of 15^o which corresponds to the hexapod maximum rotation. Thus, the hexapod can reproduce the low-frequency part of the acceleration signals. Moreover, Fig. 4.17 shows that the lack of acceleration is almost in the medium-frequency range. Therefore, we propose to use the XY-table to recover the medium-frequency since it has a larger workspace than the hexapod. The structure of our modified approach longitudinal mode is presented in Fig. 4.19.
We choose for the hexapod displacement, a third-order high-pass filter (F_{h_i})

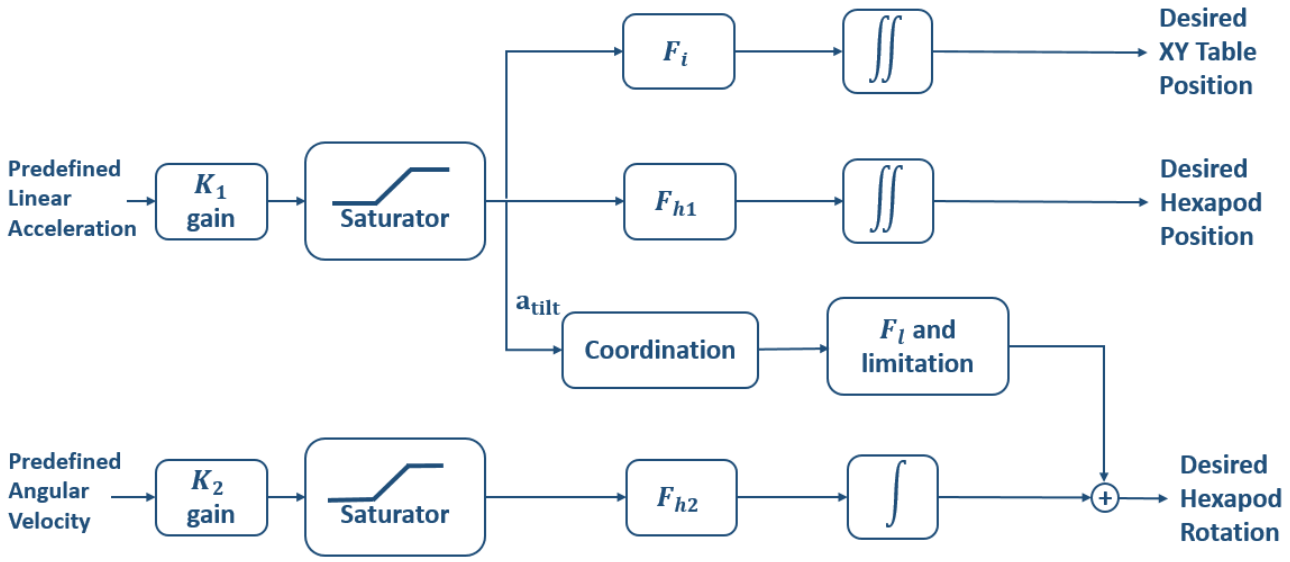


Figure 4.19: Modified classic MCA of our XY- 6 DoF platform (IBISC)

consisting of two parts: filtering (ordinary HPF) and washout. Its transfer function is given by:

$$F_{h_i} = \frac{a^s(s)}{a^{pd}(s)} = \underbrace{K_i \frac{s^2}{s^2 + 2\xi_i \omega_{n,i} s + \omega_{n,i}^2}}_{\text{ordinary HPF}} \underbrace{\frac{s}{s + \omega_{f,i}}}_{\text{Washout}} \tag{4.3.3}$$

where $a^{pd}(s)$, $a^s(s)$ are the acceleration of the real skier and that to be reproduced by the simulator, respectively. K_i, $\omega_{n,i}$, $\omega_{f,i}$ and ξ_i are parameters to be adjusted for the $i^{th} = \{1,2\}$ high-pass filter. We can summarize the effect of the high-pass filter parameters in three points:

- The pulsation ω_n achieves a compromise between the fidelity of the acceleration restitution and the respect of the mechanical constraints in linear displacement.
- The damping coefficient ξ reduces the overshoot of the filtering (the negative part of the filtered acceleration), and thus avoids an inconsistency in the simulation.
- The introduction of the pulsation ω_f is necessary for a return to the neutral position of the mobile platform. Thus, this pulsation achieves a compromise between respecting perceptual constraints and the speed of the washout.

We propose for the hexapod tilt, a second-order low-pass filter F_l to extract the low-frequency part of the predefined accelerations. And it is given below:

$$F_l = K \frac{1}{s^2 + 2\xi\omega_n s + \omega_n^2} \tag{4.3.4}$$

We suggest and choose for the XY-table, a second-order band-pass filter F_i to generate the meduim frequency part of the predefined acceleration. And it is shown as follows:

$$F_i = \frac{H_0 \frac{\omega_0}{Q} s}{s^2 + \frac{\omega_0}{Q} s + \omega_0^2} \tag{4.3.5}$$

where, H_0, Q and ω_0, are the filter gain, selective frequency and the frequency at which the gain of the filter peaks, respectively.

Fig.4.20 shows the acceleration part restored by the X-axis and shows the advantage of the proposed approach to restore the predefined acceleration.

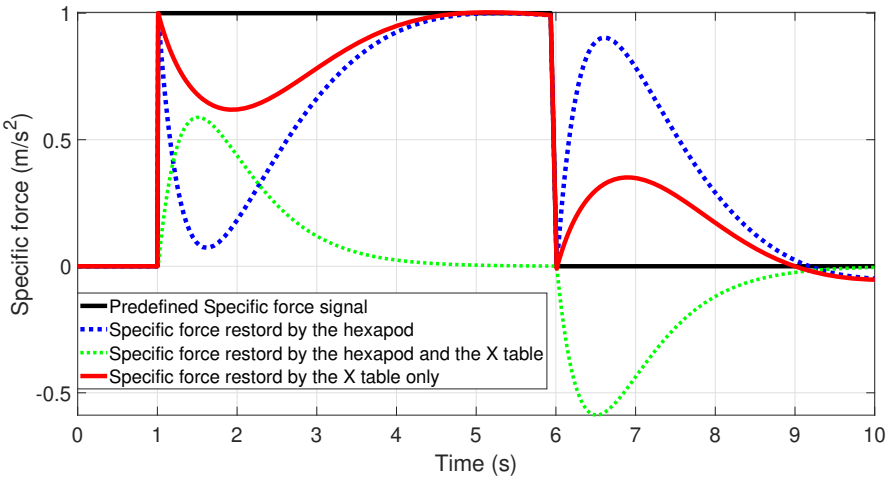


Figure 4.20: Restored specific force for standard and proposed classic MCA approach

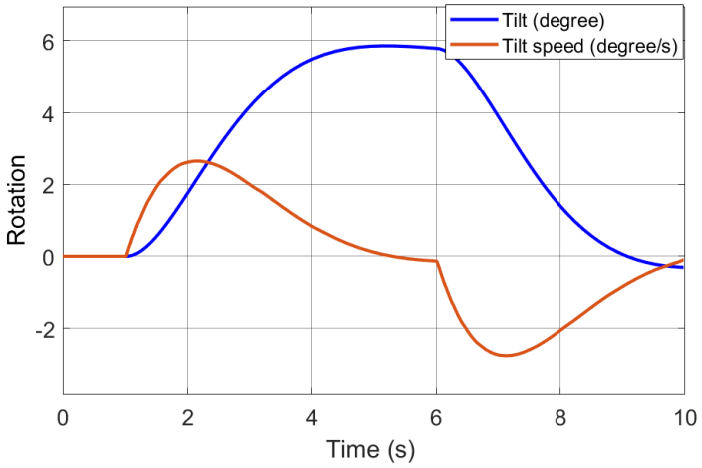


Figure 4.21: Tilt with the corresponding tilt speed

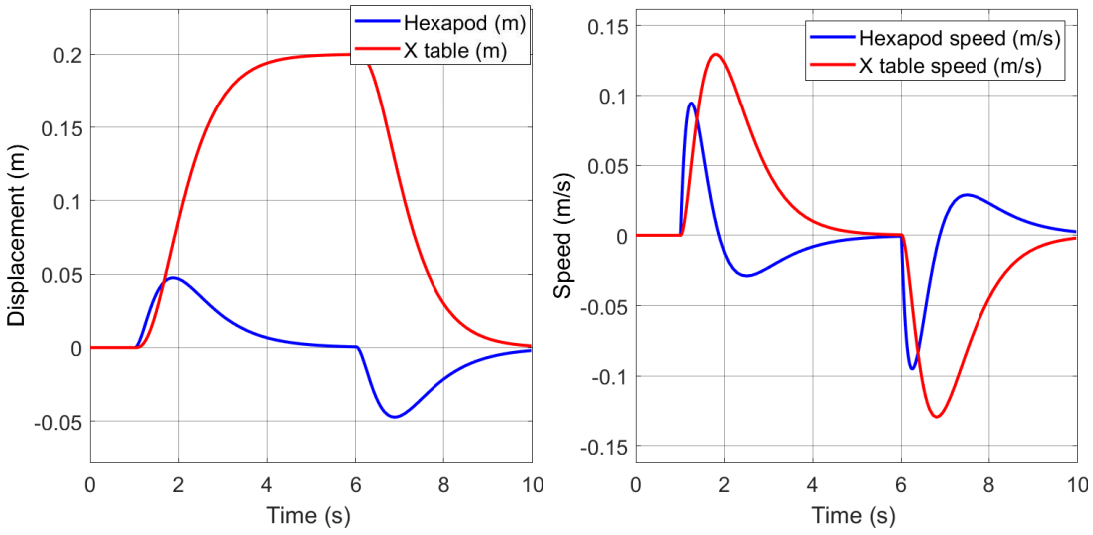


Figure 4.22: Hexapod and X Table displacements and speeds

In Fig.4.21 and Fig.4.22 we show that the translational and rotational displacement for all hexapod and XY- table return to its neutral position after applying the

washout filter. The tilt speed is lower than the angular speed threshold.
We present in the following the restored acceleration and specific force with the use of XY- Table related to the predefined ski trajectory.

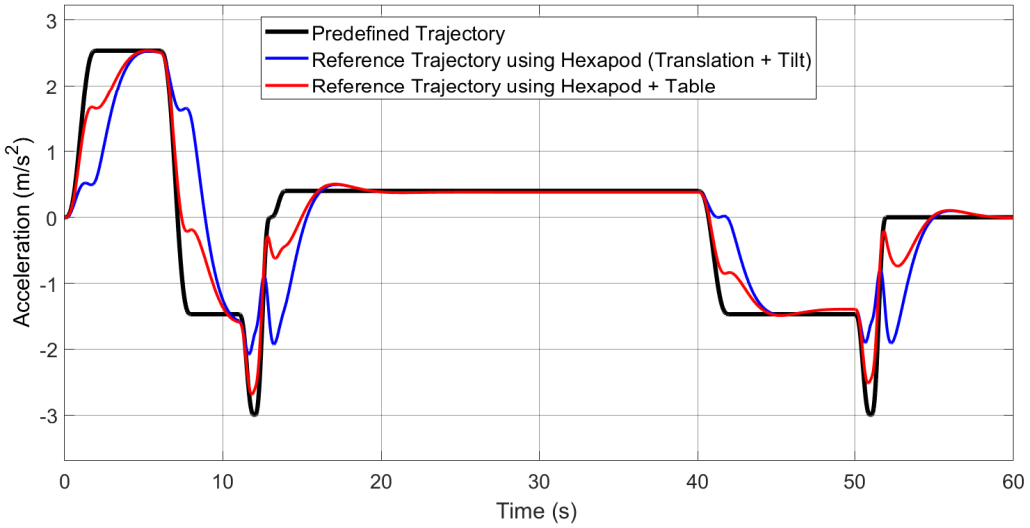


Figure 4.23: Restored acceleration with and without XY Table

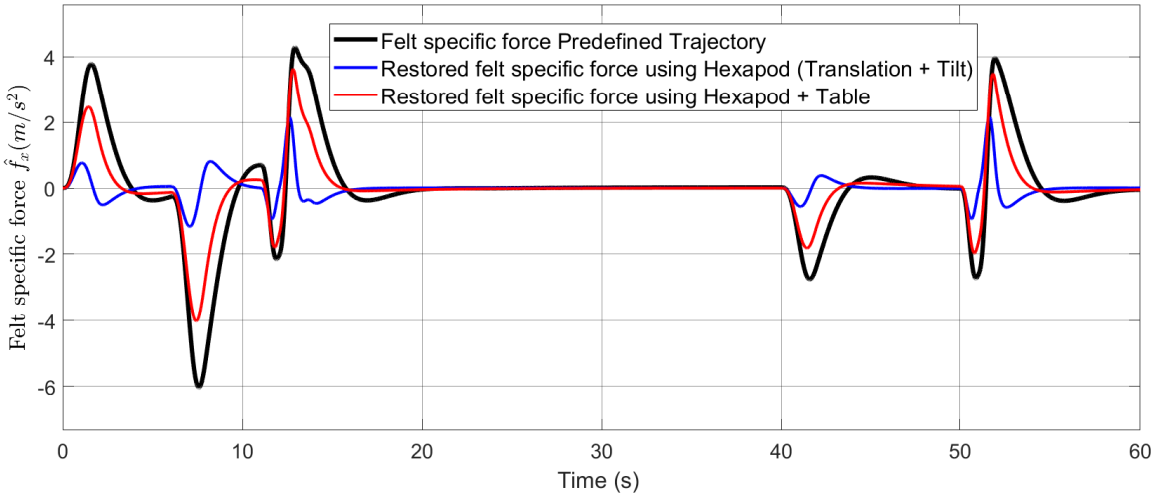


Figure 4.24: Restored specific force with and without XY Table

We present in the following the translation and rotation movement of the hexapod and the XY-table for the proposed classic MCA.

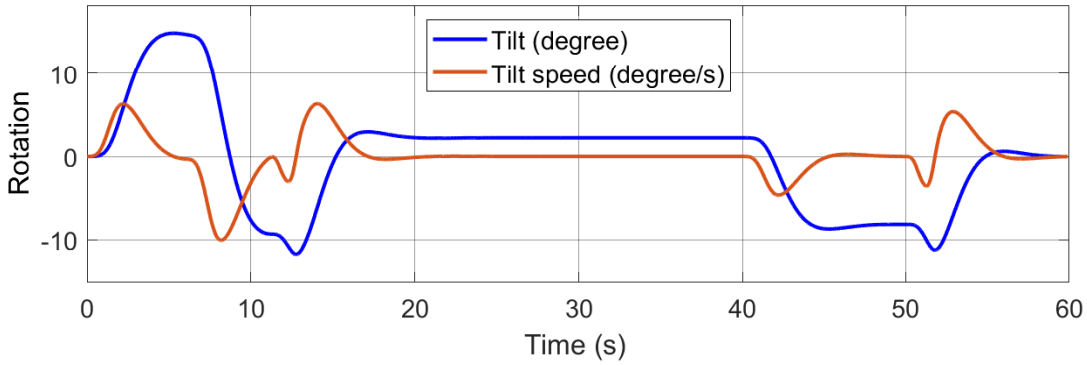


Figure 4.25: Tilt with the corresponding tilt speed

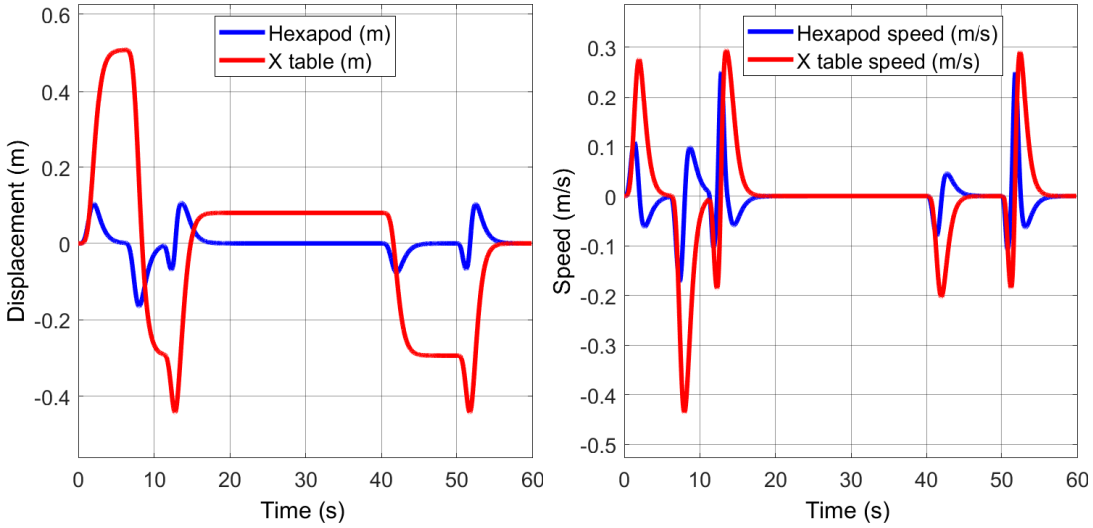


Figure 4.26: Hexapod and X Table displacements and speeds

It is clear that the translation and rotation displacement for both hexapod and table are within the XY-6 DoF workspace limits. The proposed classical MCA approach presents a better restoration of the acceleration compared to the classical MCA approach known in the literature. This improvement will increase the human immersive feeling in the virtual reality scene. However, the proposed classical MCA still needs to be improved due to the high acceleration error.

4.3.3 Desired trajectory based on adaptive MCA

The adaptive MCA was developed for the first time by Parrish, (1975) [103] and implementated on the flight simulator located at the Langley research center. This approach can be interpreted as a classic strategy, the parameters of which are updated in real-time by an optimization method based on adaptive control theory. Many researches have been conducted in this perspective such as Miunske, (2019) [7], Asadi, (2015) [113], Zhiyong, (2016) [114] to improve the human sensation. The goal is to develop non-linear filters to achieve the maximum inertial feedback in a minimum working space. However in its original version, no perception model was integrated into the development of the equations. The filters used to calculate the desired displacement and rotation are expressed by Nehaoua, (2006) [118]:

$$\begin{aligned} a_h &= p_1(a^{pd} - g\theta_h) - d\dot{x} - cx \\ \dot{\theta}_h &= p_2(a^{pd} - g\theta_h) + p_3\dot{\theta}^{pd} \end{aligned} \tag{4.3.6}$$

where a_h, θ_h, $\dot{x}$ and x are the acceleration, rotational speed, linear speed and displacement of the simulator hexapod, respectively. a^{pd}, $\dot{\theta}^{pd}$ are those of the skier. The coefficients d, c are constants and p_i (i = 1..3) are the parameters to be adapted in minimizing this quadratic criterion:

$$J = \underbrace{\frac{1}{2}(a_h - a^{pd})^2 + \frac{1}{2}w_1(\dot{\theta_h} - \dot{\theta}^{pd})^2}_{1st} + \underbrace{\frac{1}{2}w_2x^2 + \frac{1}{2}w_3\dot{x}^2}_{2nd} \tag{4.3.7}$$

where the weights w_i are positive reals. This criterion translates the compromise which exists between on one hand the reproduction of a real skier trajec-

tory (1st) and on the other hand the respect of the constrained robot movement (2nd). Note that the cost of the simulator constrained movement (2nd) reflects the desire to keep the simulation cockpit in a region around the neutral position and speed ($x = 0$ and $\dot{x} = 0$).

We find in Dagdelen, (2004) [119] an interesting interpretation of this setting in the frequency domain. Indeed, if we consider that the factors p_i are constant, we can derive from equations (4.3.6) the filters connecting the simulator accelerations to those of the real vehicle:

$$\frac{a_h(s)}{a^{pd}(s)} = \frac{p_1 s^2}{s^2 + ds + c} \tag{4.3.8}$$

The update of the filters gain is carried out by a gradient descent method to minimize the quadratic criterion:

$$\dot{p}_i = -\gamma_i \frac{\partial J}{\partial p_i} \tag{4.3.9}$$

γ_i is the adaptation step, it plays a main role in the stability of the algorithm.
In this approach, the filter parameters are computed to minimize a quadratic criterion.
We present in Fig. 4.27 the predefined acceleration with the restored one.

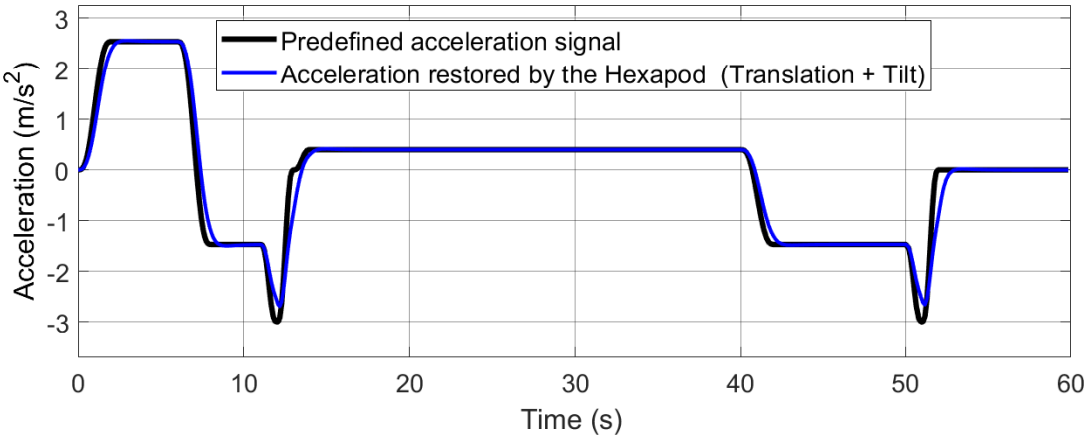


Figure 4.27: The adaptive control introduced in the literature Telban, (2005a) [104]

The desired acceleration, speed and displacement of X-table to recover the acceleration lack is presented in the Fig.4.28.

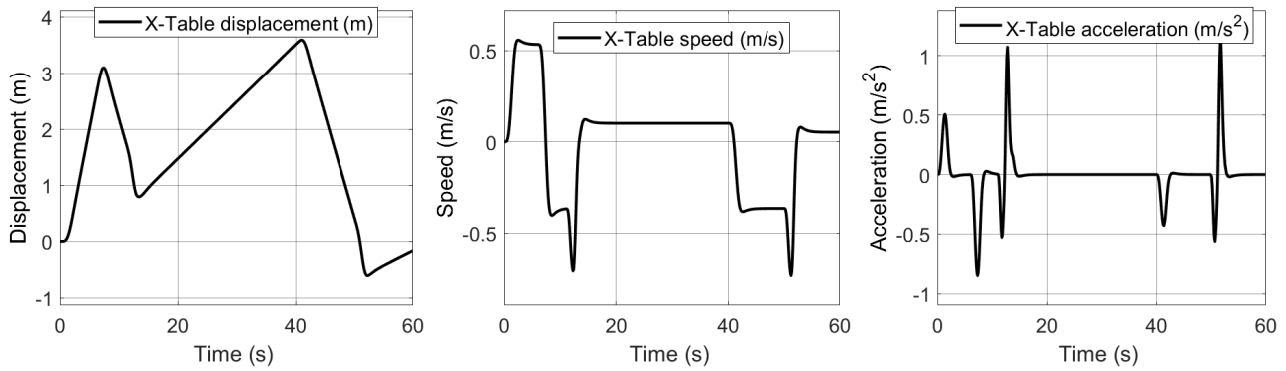


Figure 4.28: Desired acceleration, speed and displacement of X-table to recover the acceleration lack

As we can see, the required displacement for the XY table is around $4m$ which

is lower than those coming form classic MCA $8m$. The $4m$ displacement is still greater than the $1m$ arranged from our platform. Therefore,*For the same reasons mentioned in subsection (4.3.2),* the XY-table are used to recover the intermediate frequency. Thus, we will detail the structure of our new proposed adaptive MCA approach.

The structure of our modified adaptive approach for longitudinal mode is as presented in Fig 4.29.

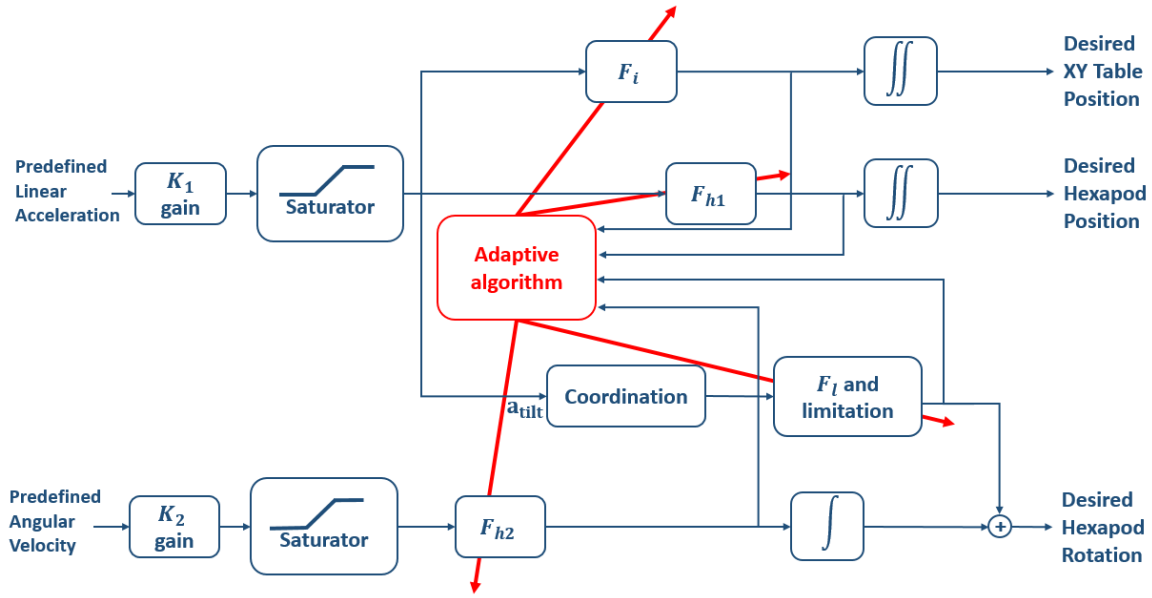


Figure 4.29: Proposed adaptive MCA approach of XY - 6 DoF platform (IBISC)

Let us define x_h, x_t and θ_h the hexapod displacement, the table displacement, and the hexapod tilt, respectively. The filters used to calculate the desired displacement and rotation of the hexapod and the desired displacement along the X-axis are:

$$\begin{aligned} \ddot{x}_h &= p_1\ddot{x}^{pd} - d_x\dot{x} - c_x x \\ \ddot{x}_t &= p_2\ddot{x}^{pd} - d_t\dot{X} - c_t X \\ \dot{\theta}_h &= p_4\ddot{x}^{pd} + p_3\dot{\theta}^{pd} \end{aligned} \tag{4.3.10}$$

where we substitute $\ddot{x}^{pd}$ by $\ddot{x}^{pd} - \ddot{x}_t - g\theta_h$, these equations can be rewritten as:

$$\begin{aligned} a_h(s) &= \frac{p_1 s^2}{s^2 + d_x s + c_x}\left(a^{pd}(s) - a_t(s) - g\frac{\omega_h(s)}{s}\right) \\ a_t(s) &= \frac{p_2 s^2}{p_2 s^2 + (1+d_t)s + c_t}\left(a^{pd}(s) - g\frac{\omega_h(s)}{s}\right) \\ \omega_h(s) &= \frac{p_3}{s + p_4 g}\dot{\theta}^{pd}(s) + \frac{p_4}{s + p_4 g}\left(a^{pd}(s) - a_t(s)\right) \end{aligned} \tag{4.3.11}$$

where, a_h, a_t and ω_h are the hexapod, table linear acceleration, and the hexapod angular velocity, respectively. Let us define $\ddot{q}$ define the hexapod, the table linear acceleration and the hexapod angular velocity. There the q is expressed as follows (and by analogy q^{pd}).

$$q \triangleq \begin{pmatrix} x_h \\ x_t \\ \int\theta_h \end{pmatrix}; \qquad q^{pd} \triangleq \begin{pmatrix} x^{pd} \\ x^{pd} \\ \int\theta^{pd} \end{pmatrix} \tag{4.3.12}$$

We generalize the coordinated adaptive criterion:

$$J = \frac{1}{2}e^T Le + \frac{1}{2}q^T Rq + \frac{1}{2}\dot{q}^T Q\dot{q} \tag{4.3.13}$$

with the weight matrices L, Q and R, they are positive semi-defined symmetric matices (p.s.d.s). We define the tracking error by:

$$e = \ddot{q} - \ddot{q}^r \tag{4.3.14}$$

where $e \in \mathbb{R}^3$, $\ddot{q}^r = \ddot{q}^{pd} - K\ddot{q} - G\dot{q}$, with,

$$G = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & g \end{bmatrix}, \; K = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \end{bmatrix};$$

The tracking error can be developed as:

$$e = (I_d + K)\ddot{q} + G\dot{q} - \ddot{q}^{pd} \tag{4.3.15}$$

where I_d is the identity matrix. Let us define $\overline{K} = I_d + K$.

$$\overline{K} = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 2 & 0 \\ 0 & 1 & 1 \end{bmatrix};$$

Therefore,

$$e = \overline{K}\ddot{q} + G\dot{q} - \ddot{q}^{pd} \tag{4.3.16}$$

The adaptive configuration generalization in (4.3.10) is direct,

$$\begin{aligned} \ddot{q} &= P(\ddot{q}^{pd} - K\ddot{q} - G\dot{q}) - D\dot{q} - Cq \\ \ddot{q} &= \frac{P}{(I_d + PK)}(\ddot{q}^{pd} - G\dot{q}) - D\dot{q} - Cq \end{aligned} \tag{4.3.17}$$

where $P \geq 0$ is the adaptation matrix. D and C are constant matrices p.s.d. The matrix P, therefore, regulates the compromise that the platform must satisfy between maximum tracking and compliance with the kinematic limitations. The resolution of this adaptive problem is carried out by the gradient descent method:

$$\dot{P} = -\gamma\frac{\partial J}{\partial P} \tag{4.3.18}$$

where γ is the adaptation step matrix and J is the adaptive criterion. With,

$$\gamma = \begin{bmatrix} \gamma_1 & 0 \\ \gamma_2 & 0 \\ \gamma_4 & \gamma_3 \end{bmatrix}; \qquad P = \begin{bmatrix} p_1 & 0 \\ p_2 & 0 \\ p_4 & p_3 \end{bmatrix};$$

Therefore the derivative of eq. (4.3.13) can be expressed as,

$$\frac{\partial J}{\partial P} = e^T L\frac{\partial e}{\partial P} + \dot{q}^T Q\frac{\partial \dot{q}}{\partial P} + q^T R\frac{\partial q}{\partial P} \tag{4.3.19}$$

From (4.3.16), we get:

$$\frac{\partial e}{\partial P} = \frac{\partial\left(\overline{K}\ddot{q} + G\dot{q} - \ddot{q}^{pd}\right)}{\partial P} = \overline{K}\frac{\partial \ddot{q}}{\partial P} + G\frac{\partial \dot{q}}{\partial P} \tag{4.3.20}$$

Using (4.3.17), we have:

$$\frac{\partial \ddot{q}}{\partial P} = \frac{\partial}{\partial P}\left(\frac{P}{I_d + PK}\right)[\ddot{q}^{pd} - G\dot{q}] - \left(\frac{PG}{I_d + PK} + D\right)\frac{\partial \dot{q}}{\partial P} - C\frac{\partial q}{\partial P} \tag{4.3.21}$$

We assume that q is regular and smooth trajectory, hence,

$$\frac{\partial \ddot{q}}{\partial P} = \frac{d}{dt}\left(\frac{\partial \dot{q}}{\partial P}\right) = \frac{d^2}{dt^2}\left(\frac{\partial q}{\partial P}\right) \tag{4.3.22}$$

Therefore, $\frac{\partial J}{\partial P}$ can be expressed as function of $\left(P, \frac{d}{dt}\left(\frac{\partial \dot{q}}{\partial P}\right), \frac{\partial q}{\partial P}\right)$ Equations (4.3.20) and (4.3.21) can be rewritten as:

$$\begin{aligned} \frac{\partial e}{\partial P} &= \overline{K}\frac{d^2}{dt^2}\left(\frac{\partial q}{\partial P}\right) + G\frac{d}{dt}\left(\frac{\partial q}{\partial P}\right) \\ \frac{\partial}{\partial P}\left(\frac{P}{I_d + PK}\right)[\ddot{q}^{pd} - G\dot{q}] &= \left(\frac{PG}{I_d + PK} + D\right)\frac{d}{dt}\left(\frac{\partial q}{\partial P}\right) + \frac{d^2}{dt^2}\left(\frac{\partial q}{\partial P}\right) + C\frac{\partial q}{\partial P} \end{aligned} \tag{4.3.23}$$

Adaptive filtering problem can, therefore, be formulated as the integration of a first order differential system for Γ state:

$$\zeta \triangleq [q,\ \dot{q},\ \frac{\partial q}{\partial P},\ \frac{d}{dt}\left(\frac{\partial q}{\partial P}\right),\ P]^T; \quad \dot{\zeta} = \varrho(\zeta) \tag{4.3.24}$$

where ϱ is constructed using equations (4.3.18) and (4.3.23). We present in Fig. 4.30 and Fig. 4.31 the proposed adaptive and adaptive approach acceleration and specific force restoring.

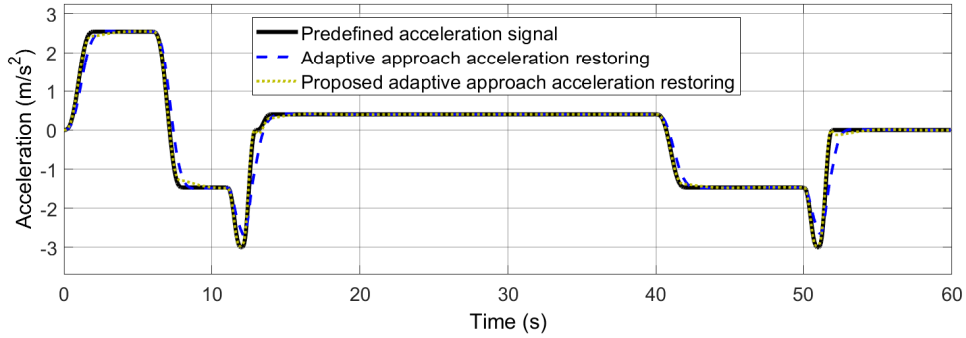


Figure 4.30: Acceleration from the proposed adaptive algorithm vs literature

The specific force errors are presented in Fig. 4.32.

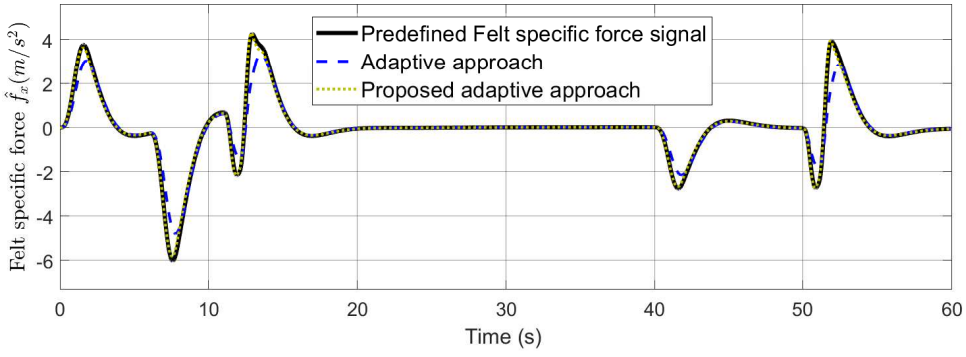


Figure 4.31: Specific force from the proposed adaptive algorithm vs literature

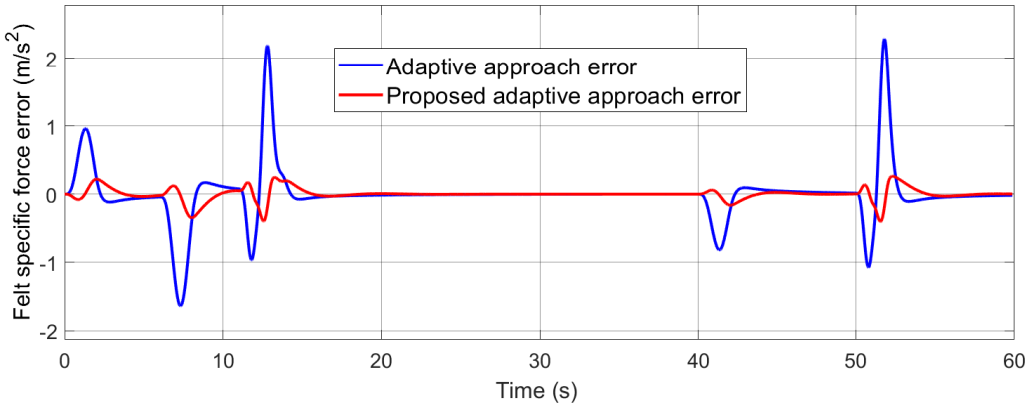


Figure 4.32: Specific force errors from the proposed adaptive algorithm vs literature

The proposed adaptive approach can decrease the specific force error significantly, which shows the efficiency of our proposed method. Fig. 4.33 shows the Hexapod translational acceleration, speed and displacement in x-direction over time (s) for the proposed adaptive approach.

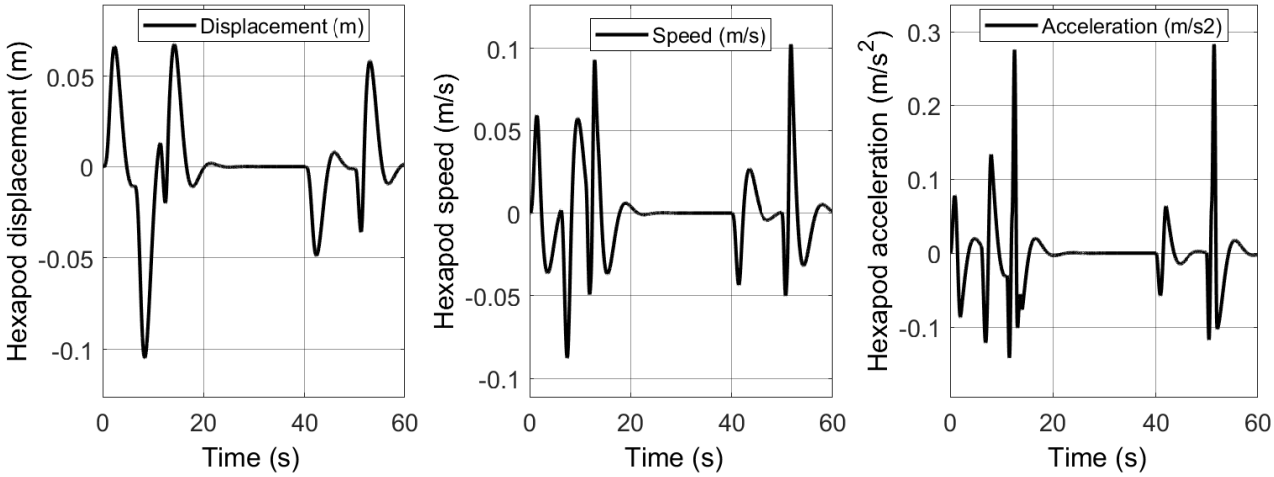


Figure 4.33: The Hexapod translational acceleration, speed and displacement in x-direction over time (s)

Fig. 4.34 shows Tilt angle and speed performed by the hexapod over time for the proposed adaptive approach.

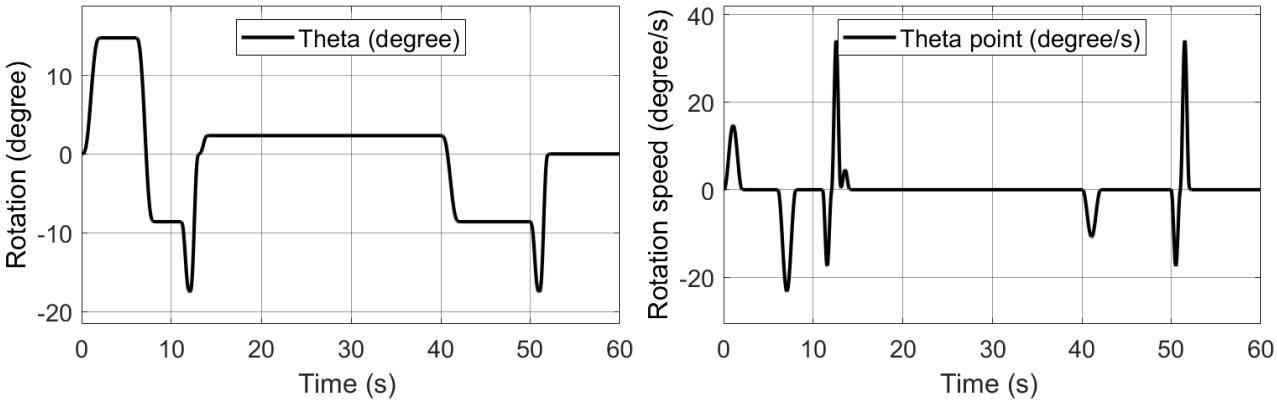


Figure 4.34: Tilt angle and speed over time (s)

Fig. 4.35 shows the acceleration, speed and displacement of the X-Table over time for the proposed adaptive approach.

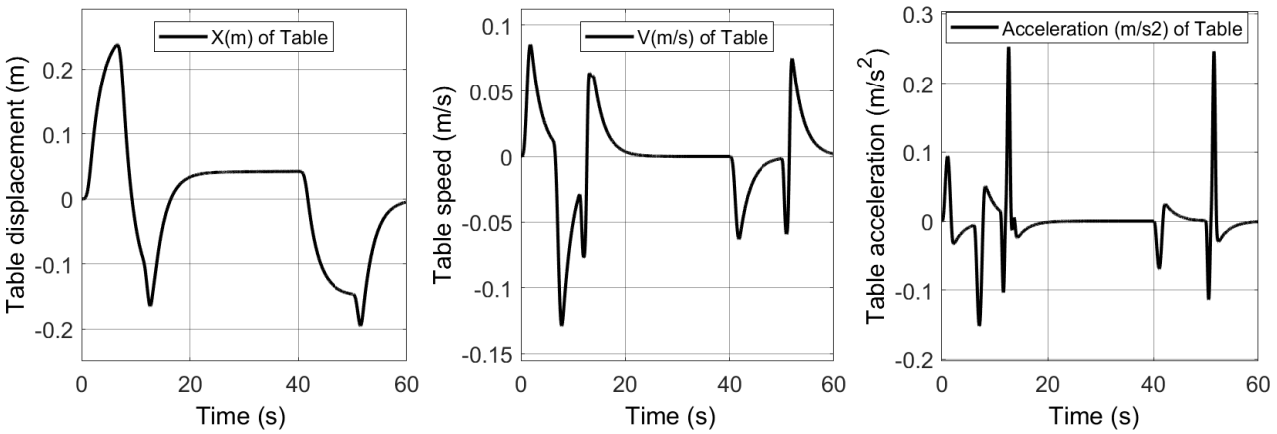


Figure 4.35: Acceleration, speed and displacement of the X-Table over time (s)

4.3.4 Desired trajectory PSO-based MCA optimization

Particle Swarm Optimization (PSO) is a biologically inspired computational search and optimization method which was introduced for the first time in Eberhart, (1995) [120] and Shi, (1998) [121]. It is a relatively recent heuristic search method whose mechanics are inspired from the social behaviors of birds flocking or fish schooling where they have no leaders will get them to the food sources which brings them together to collaborate and share information so that the flocks achieve their best condition simultaneously through communication among members who already have a better situation (see Rini, (2011) [122]).

PSO iteratively improves a candidate solution of an optimization problem, and shares information to all swarm members Hassan, (2005) [123]. PSO has proven to be capable of solving optimization problems with vast and multi-dimensional search spaces Marini, (2015) [124]. PSO has been successfully used for parameter tuning in other algorithms Lin, (2008) [125]. In Asadi, (2017) [126] and Casas, (2018) [127] PSO is also used to adjust washout filters. This shows that the adaptation of this heuristic method to MCA problem is feasible, and attractive.

In the proposed work for the classic approach, the desired trajectory is a result of three filters (high-, band- and low- filters) and the cut-off frequencies and dampings ratio are tuned by the trial and error methods. To overcome this limitation, PSO is used as a population based heuristic search method for searching an optimal solution in complex and large spaces. Morever, PSO presents a faster method for finding the global optimal solution, as compared with previous trial tuning process. A particle k in PSO for the longitudinal mode at iteration j is represented by the following vectors of states,

$$X_k^j = [x_{k,1}^j,\ x_{k,2}^j,\ \cdots,\ x_{k,12}^j] \tag{4.3.25}$$

where the k^{th} particle in the swarm includes $x_{k,1}^j,\ x_{k,2}^j,\ \cdots,\ x_{k,12}^j$ that denote K_l, ξ_l, ω_l, H_0, Q, ω_0, $K_{h,i}, \xi_{h,i}, \omega_{h,i}$ with $i = 1,2$ of the low , band and high pass filter parameters, respectively (the same filter parameters introduced in the section 4.3.2).

The particles of the swarm fly through a hyperspace with two main characteristics: exploration and exploitation capabilities. On the one hand, they have memory of their best locations, known as local best, therefore each particle is stochastically accelerated towards its own local best position. And on the other, they have knowledge of the global best position of the whole population that is known as global best.

Let us introduce, $x_k(t)$ as the particle position in the search space at time step t. The particle location is affected by its velocity and is updated by adding $v_k(t)$ to its existing position. The most general expression of the canonical PSO algorithm for k^{th} particule is as in Eberhart, (1995) [120]:

$$v_k(t+1) = \omega \underbrace{v_k(t)}_{\text{Current motion}} + c_1 \underbrace{r_1(p_{b,k}(t) - x_k(t))}_{\text{particle memory influence}} + c_2 \underbrace{r_2(g_b(t) - x_k(t))}_{\text{swarm influence}} \tag{4.3.26a}$$

$$x_k(t+1) = x_k(t) + v_k(t+1) \tag{4.3.26b}$$

where, $p_{b,k}(t)$, $g_b(t)$ represent the best local and the global extremum, c_1, c_2 represent the acceleration coefficients also known as self confidence, r_1, r_2 represent the random vectors , and ω represents the inertial factor that make a compromise between exploration and exploitation and it's value goes from 1.1 to 0.1 to ensure a better convergence of the PSO-method Yang, (2007) [128]. Note that $x_k(0) \sim U(x_{min}, x_{max})$ where $U(a,b)$ represents the uniform distribution between a and b.

To avoid any non feasible solution, the particule swarm position can be bounded as $x_{lo} \leq x \leq x_{up}$ where x_{lo} and x_{up} indicate the lower and upper limits, respectively. It should be noted that the size of the particle group should not be too large or too small, as there are many possible locations towards the best solution. This is important when the overall value of the particle is uniformly distributed in the search area Casas, (2018) [127]. The PSO algorithm terminates either when a predefined fitness value is obtained by at least one particle, or when the defined maximum number of generations (iterations) is reached.

The MCA objectives is to minimize the human sensation error without violation of the physical limitation. Therefore, PSO is used to minimize the fitness values which is the total human sensation error. PSO should respond to three objectives: reduce the human sensation error, follow the actual skier's sensation signal accurately, and use the workspace more efficiently. By using PSO, the physical limitations of the simulator can be considered in the filter parameter tuning process. Therefore, PSO can offer an optimized balance between the human sensation error and workspace usage, while the physical limitations are not violated.

The total evaluation or fitness function of the PSO-based washout filter can be described as follows:

$$F(X_k^j) = F_{k,se}^j + F_{k,lim}^j \tag{4.3.27}$$

where $F_{k,se}^j$ and $F_{k,lim}^j$ are the fitness functions associated with the total human sensation error and the platform motion displacement, respectively.

Thus, the aim of the PSO method is to decrease simulator sickness and increase motion fidelity. Hence, the PSO fitness function, that should be minimized, is composed mainly by the sensation error between the real and simulator skiers. We will use the Integral of Squared Error performance index to quantify the error and measure the performance. The total human sensation error fitness function includes the human sensation error in the translational and rotational modes can be presented as follows:

$$F_{k,se}^j = \gamma_{e,f} \int \hat{e}_f^2 dt + \gamma_{e,\omega} \int \hat{e}_\omega^2 dt \tag{4.3.28}$$

where $\gamma_{e,f}$ and $\gamma_{e,\omega}$ denote the penalty weights for the sensed specific force and

angular velocity errors. The weights can be selected according to the robotic platform requirements to influence the fitness value positively. The felt specific force error $\hat{e}_f(t)$ and the felt angular velocity error $\hat{e}_\omega(t)$ between the real (predefined) and simulated skiers are defined as follows:

$$\hat{e}_f(t) = \hat{f}_{pd}(t) - \hat{f}_s(t) \tag{4.3.29a}$$

$$\hat{e}_\omega(t) = \hat{\omega}_{pd}(t) - \hat{\omega}_s(t) \tag{4.3.29b}$$

Furthermore, it is requied for the optimized washout filter to constrain the linear and angular simulator motions. This constraint is used in order to avoid any violation of the physical limitation. As such, linear accelerations $\ddot{x}_h$, $\ddot{x}_t$ of hexapod and the XY table, respectively, and angular velocity $\dot{\theta}$ are considered in the designed PSO fitness function. In addition, the linear and angular displacements (x_h, x_t and θ) should be kept in the parallel platform and the XY table limitation ranges. It should be noted that the integral of linear displacements x_h, x_t as well as linear velocity $\dot{x}_h$, $\dot{x}_t$ are considered in the PSO fitness function. It is an essential term as high fidelity motions should be reproduced within the physical limitations. Therefore, the total displacement fitness function can be represented by three limitation functions the XY table, the translation and rotation limitations of the hexapod platform, consequently:

$$F^j_{k,lim} = F^j_{k,lim,T,hexapod} + F^j_{k,lim,R,hexapod} + F^j_{k,lim,XYtable} \tag{4.3.30}$$

where,

$$F^j_{k,lim,XYtable} = \gamma_{x_t}\int x_t^2 dt + \gamma_{return,x_t}\int\left(\int x_t dt\right)^2 dt + \gamma_{\dot{x}_t}\int \dot{x}_t^2 dt + \gamma_{\ddot{x}_t}\int \ddot{x}_t^2 dt \tag{4.3.31a}$$

$$F^j_{k,lim,T,hexapod} = \gamma_{x_h}\int x_h^2 dt + \gamma_{return,x_h}\int\left(\int \gamma_d dt\right)^2 dt + \gamma_{\dot{x}_h}\int \dot{x}_h^2 dt + \gamma_{\ddot{x}_h}\int \ddot{x}_h^2 dt \tag{4.3.31b}$$

$$F^j_{k,lim,R,hexapod} = \gamma_\theta\int \theta^2 dt + \gamma_{\dot{\theta}}\int \dot{\theta} dt \tag{4.3.31c}$$

where $\gamma_{x_h}, \gamma_{return,x_h}, \gamma_{\dot{x}_h}$ and $\gamma_{\ddot{x}_h}$ are the penalty weights of translational displacement, integral of translational displacement, translational velocity and translational acceleration for the hexapod, respectively. γ_θ and $\gamma_{\dot{\theta}}$ are the penalty weights of angular displacement, and angular velocity of the hexapod, respectively. Then $\gamma_{x_t}, \gamma_{return,x_t}, \gamma_{\dot{x}_t}$ and $\gamma_{\ddot{x}_t}$ are the penalty weights of translational displacement, integral of translational displacement, translational velocity, translational acceleration for the XY table, respectively.

In this work, firstly we tune the classical washout filter parameters according to trial and error, as usual. Therefore, the most suitable washout filter parameters found are applied as the initial values in the PSO-based tuning method. In addition, it allows us to have an idea of the lower and upper bounds choices for each parameter. Especially for the threshold rate limiter in the tilt coordination channel. This approach ensures that, in the worst-case scenario, the selected parameters based on PSO are close to the initial parameters. According to the discussion above, the steps of the PSO algorithm are shown below.

Algorithm 1: Steps of the PSO algorithm

Initialize the particle swarm;
n- number of particles; ite- number of iteration; d- dimension of solution space;
x_k^j- particles' position; v_k- particles' speed; ω- inertia factor;
c_1, c_2- self confidence coefficients; r_1, r_2- random vectors;
$l = 0.9$- constant to decrease the inertia factor;
Calculate and initialize;
$p_{b,k}$- local best; g_b- global best;
for $i < ite$ **do**
  Update particles' speed;
  $v_k(t) = \omega\ v_k(t-1) + c_1 r_1(p_{b,k}(t) - x_k(t-1)) + c_2 r_2(g_b(t) - x_k(t-1))$;
  Update particles' position;
  $x_k(t+1) = x_k(t) + v_k(t+1)$;
  Update inertial factor;
  $\omega = l * \omega$;
  Calculate each particle's fitness value;
  $F(X_k^j)$;
  Update personal and global best extremum;
  if $(F(X_k^j) > F(p_{b,k}))$ **then**
    $p_{b,k} = X_k^j$ and $g_b = F(X_k^j)$
  end
end
Result: Optimized particle position x_k

By using PSO, we are able to optimize the washout filter parameters while minimizing the sensation error between real and simulator skiers. The physical limitations of the platform and human threshold in tilt coordination, which are nonlinearities in the system, have also been considered in the PSO-based optimization method.

The proposed PSO-based method is implemented using MATLAB/Simulink software package for tuning the washout filter parameters. The results are presented in the following. As described above, the proposed PSO is applied in this research work to tune the washout filter parameters. Its objective is to improve the filters performance by taking human sensation error and motion constraints into account and by reducing the designer's tuning effort.

The washout algorithm model is first verified. Then, the PSO washout algorithm is simulated and tested, and the PSO based washout results are compared with the common washout algorithm to prove the superiority of the PSO washout algorithm that we propose. Lastly, the performance of the PSO used for resolving the proposed model is simulated and analyzed. For the longitudinal mode, the low-pass filter in the tilt channel and the high-pass filter in the translational channel for the hexapod platform are tuned together. The band-pass filter and the high-pass filter in the rotational channel are tuned separately.

In the experimental study, the PSO parameters used were as follows: swarm size 100; iteration limit 70, and inertia range is [0.1, 1.1]. Fig. 4.36 presents the convergence of the PSO method. As can be seen in Fig. 4.36, the convergence has been reached partially at 28^{th} iteration and totally at 50^{th} iteration. Therefore, satisfactory optimized results are obtained with a minimized sub-fitness values and thus, a minimized total fitness function value.

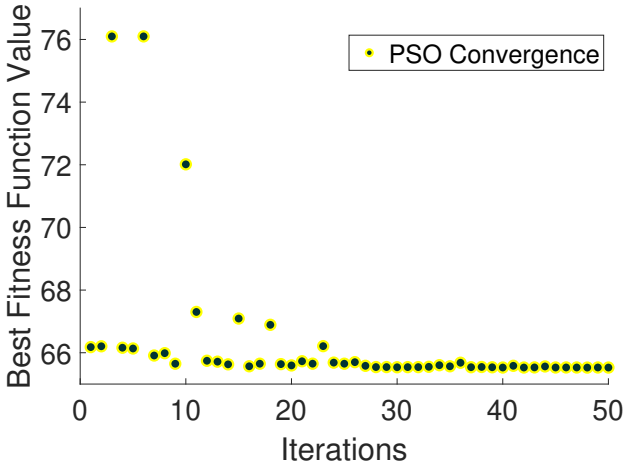


Figure 4.36: Convergence of fitness function according to iterations

For the sake of clarity, the PSO-based approach will first be compared to the classic one, and then, all of these approaches will be compared to each other.

Fig. 4.37 and Fig. 4.38 show accelerations and felt specific forces in the longitudinal mode of the PSO-based and classical washout filters for the hexapod alone and the hexapod with the XY table.

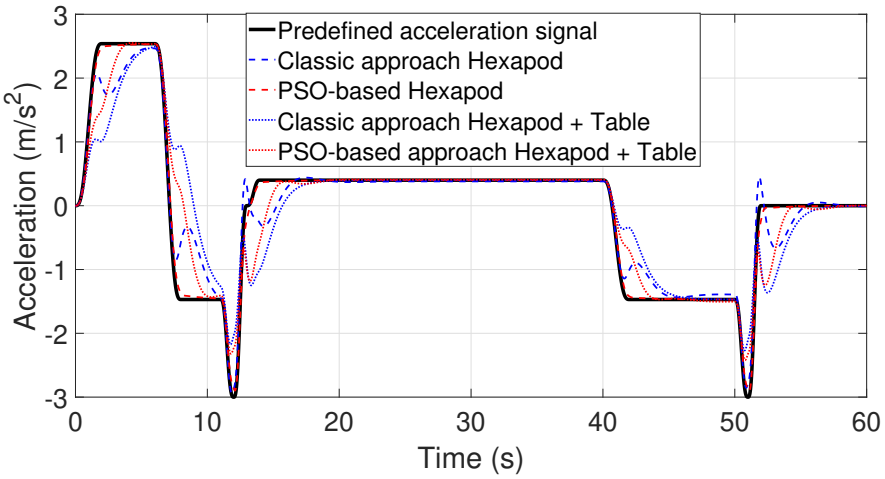


Figure 4.37: Restored acceleration for classic and PSO-based filters

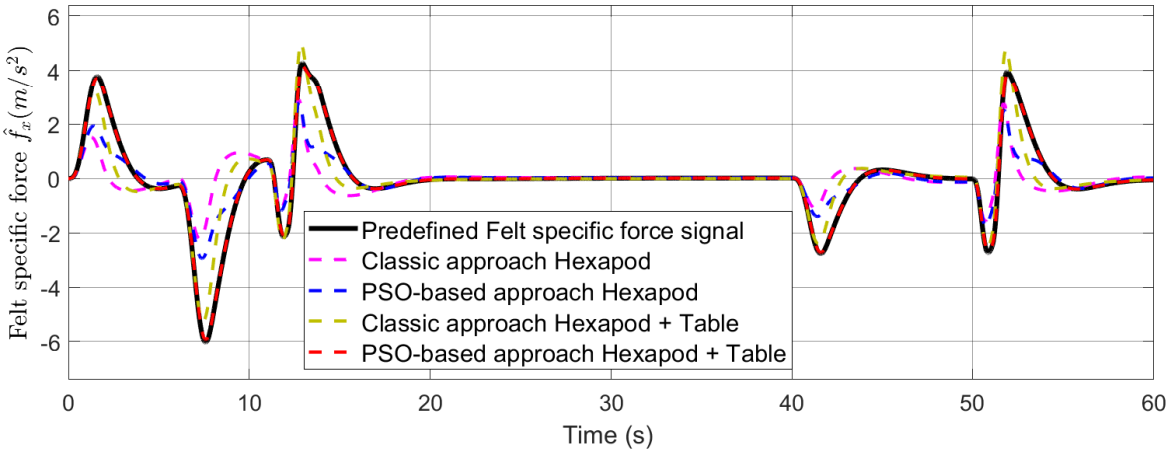


Figure 4.38: Restored specific force for classic and PSO-based filters

According to Fig. 4.38 the felt specific force in the PSO-based washout filter follows the reference signal and its shape more precisely, as compared with that from the classical washout filter for both hexapod and hexapod with XY table. In addition, it can be seen that the felt specific force follows the predefined specific force signal better with the use of the XY table. Fig. 4.39 shows the felt specific force error of the PSO-based and classical washout filters for the hexapod alone and the hexapod with the XY table.

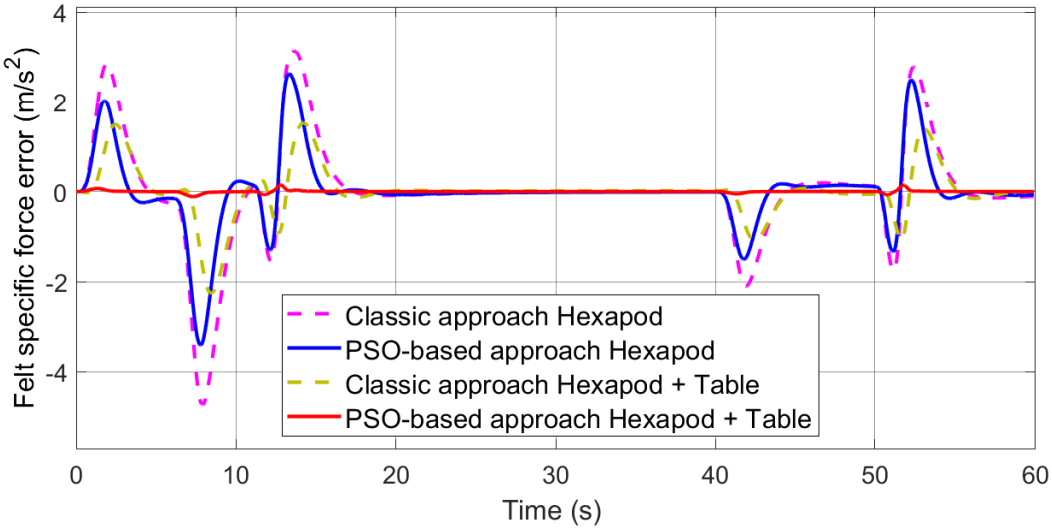


Figure 4.39: Restored felt acceleration errors for classic and PSO-based filters

In Fig. 4.39, it can be clearly observed that some of the sensation error peaks that provide false motion cues are strictly reduced using the PSO-based washout filter or/and by using the XY table. The displacement in the longitudinal direction for the hexapod and the XY table are shown in Fig. 4.40 and Fig. 4.41, respectively.

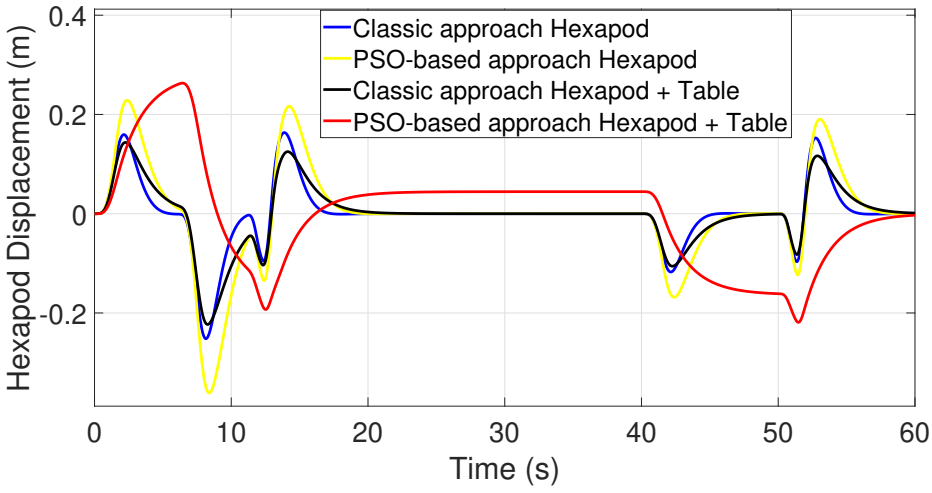


Figure 4.40: Hexapod displacement for classic and PSO-based filters

When using the hexapod alone, the displacement of the hexapod is increased from 0.185 m using the classical washout filter to 0.21 m using the PSO-based washout. Moreover, for the hexapod with the XY-table, the displacement is increased from 0.18 m using the classical washout filter to 0.25 m using the PSO-based washout filter which is the maximum displacement of the hexapod. Similarly, it can be seen from Fig. 4.41 the PSO-based MCA approach use all authorized displacement of the XY- table which is 1m. This shows the effectiveness of

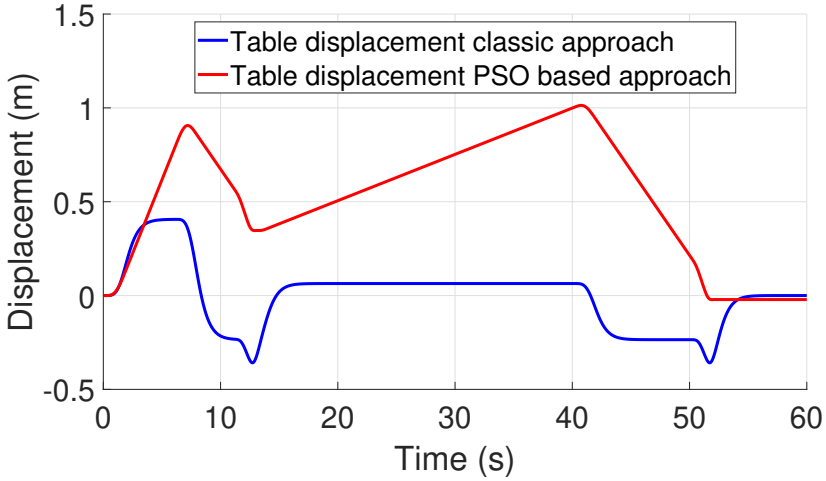


Figure 4.41: Table displacement for classic and PSO-based filters

the proposed PSO based washout filter in using the entire available workspace. Fig. 4.42 indicates the inclination of the platform (tilt angle) in the classical and PSO based washout filters for both with/without using the XY-table.

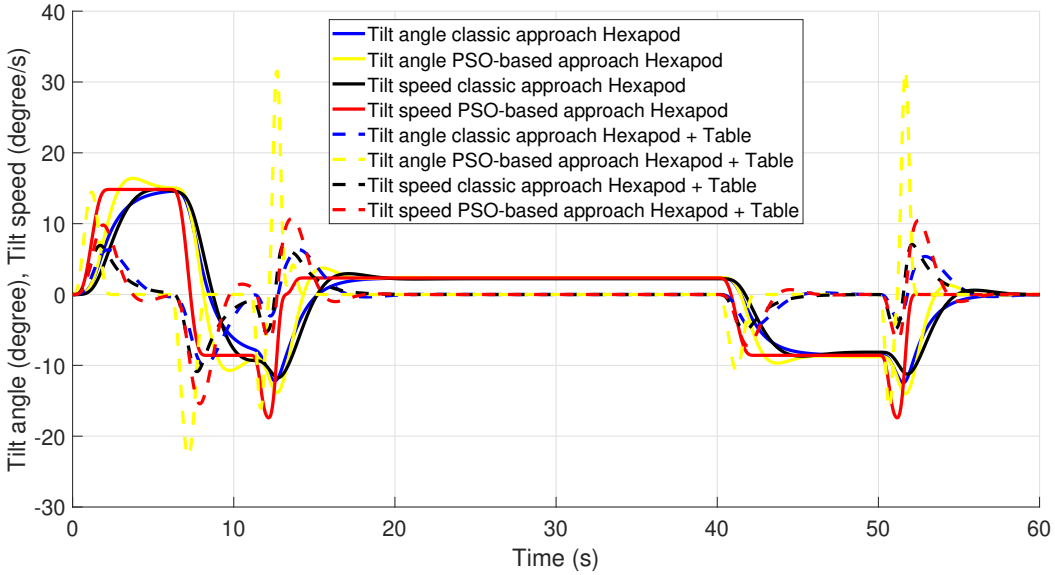


Figure 4.42: Tilt angle and speed for classic and PSO-based filters

The maximum angle remains the same (around 15 degrees) whereas, the angular speed changes faster to meet the goal in following the predefined acceleration signal.

Let's introduce a performance indicator to measure the fidelity of the MCA and to quantify the improvement,

$$PI_{\%} = 100\left(1 - \frac{||e||}{||a^{pd}||}\right) \tag{4.3.32}$$

where $\frac{||e||}{||a^{pd}||}$ measures the generated and predefined acceleration offset and $||.||$ is the $2-$norm. Based on these results, there is a significant improvement in the washout filter performance by using the proposed PSO. The PI increases from 63.25% to 77.57% for the hexapod alone, and from 80.99% to 96.66% when both hexapod and XY table work together. The results show the effectiveness of the pro-

MCA	Classic approach	Classic PSO based approach
Hexapod	63.25 %	82 %
Hexapod with XY table	77.57 %	96.66 %

Table 4.2: PI of MCA approaches for Hexapod with, without XY table

posed PSO-based washout filter in many levels: it reduces the human sensation error, it maximizes the shape following factor and it uses the entire workspace availability. Therefore, this proves the superiority of the proposed PSO-based washout filter. In addition, It can be seen that the proposed band-pass filter used for the XY table can increase the PI of both MCA approaches (classic approach: 17.74%, PSO-based: 19.09%). The idea is that the hexapod and the XY-table can interact, leading to a more realistic simulations of movement and all filter parameteres are tuned using PSO.

4.3.5 Desired trajectory based on optimal MCA approach

This technique was designed in Sivan, (1982) [116], followed by Shalom, (1982) [129] and later implemented by Reid and Nahon, (1985) [130]. Recent work by Cardullo, (1999) [131] and Telban, (2005b) [132] have implemented this type of approach for the NASA Langley Institute simulator as well as for the Simona simulator (University of Delft). Thanks to optimal control, this approach calculates, offline, an inertial rendering filter.
It explicitly integrates linear models of vestibular perception and the platform kinematics. The key idea is to determine, offline, an optimal transfer function $W(s)$ which links the simulator desired input states with those of the real skier.

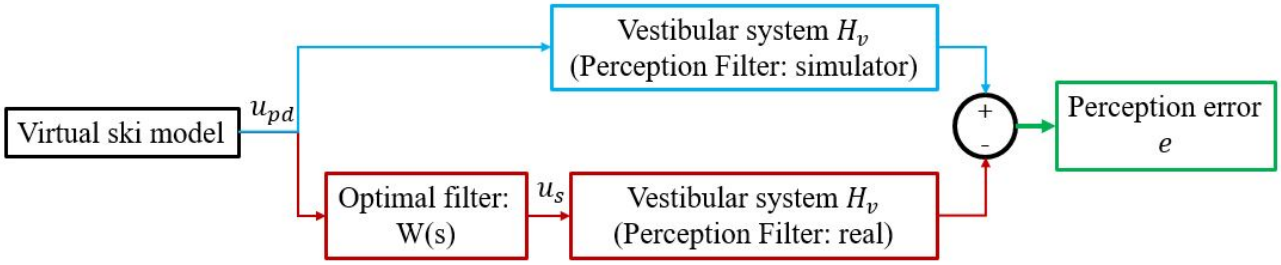


Figure 4.43: Optimal algorithm diagram

According to figure 4.43 this transfer function is calculated, using a relevant technique of optimal control domain, by minimizing the sensitive error e. the sensitive error is defined between, on the one hand, the response of the vestibular system for trajectories of the real skier and, on the other hand, that of the trajectories carried out by the simulator platform, therefore:

$$e = H_v(u_s - u_{pd}) \tag{4.3.33}$$

where H_v is the linear vestibular model which represents the dynamics of otoliths and semicircular canals. u_s, u_{pd} are the input states of the simulator and the skier (predefined) respectively. The state vector $u = [\omega_y, a_x]^T$ is composed of the rotational speed and the translational acceleration.
The vestibular state error is given as $\xi_e = \xi_s - \xi_v$,where ξ_s contains the simulator driver vestibular state and ξ_{pd} contains the skier vestibular state. The skier sen-

sation error, e, is given as:

$$\begin{aligned} \dot{\xi}_e &= A_V \xi_e + B_V u_s - B_V u_{pd} \\ e &= C_V \xi_e + D_V u_s - D_V u_{pd} \end{aligned} \tag{4.3.34}$$

To constrain the motion, the cost function should take the translational velocity, displacement, and integral displacement of the simulator as well as the angular rotation. The integral of the translational displacement is added to return the simulator back to its neutral position. the following additional simulator terms are involved in the state space:

$$\begin{aligned} \xi_d &= \begin{bmatrix} \iiint a_x dt^3 & \iint a_x dt^2 & \int a_x dt & \theta \end{bmatrix}^T \\ \dot{\xi}_d &= A_d \xi_d + B_d u_s \end{aligned} \tag{4.3.35}$$

where they are linked to the simulator input, u_s, through

$$A_d = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} B_d = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 1 \\ 1 & 0 \end{bmatrix}$$

As recommended in Sivan, (1982) [116], filtered white noise can be used as an input to the MCA with the aim to reproduce several skiing situations. This implies that an additional state space model needs to be incorporated to the one created earlier, whose states are the input.
The skier input, u_{pd}, consists of filtered white noise, and can be stated as follows:

$$\begin{aligned} \dot{\xi}_n &= A_n \xi_n + B_n w \\ u_{pd} &= \xi_n \end{aligned} \tag{4.3.36}$$

where, ξ_n contains the filtered white noise states and w indicates white noise. A_n and B_n are given as:

$$A_n = \begin{bmatrix} -\lambda_1 & 0 \\ 0 & -\lambda_2 \end{bmatrix} B_n = \begin{bmatrix} \lambda_1 \\ \lambda_2 \end{bmatrix}$$

where λ_1 and λ_2 are the first-order filter break frequencies for each degree of freedom. The state equations, as in (4.3.34), (4.3.35), and (4.3.36), can be combined to form the system of equations, as follows:

$$\begin{aligned} \dot{\xi} &= A\xi + Bu_s + Nw \\ y &= C\xi + Du_s \end{aligned} \tag{4.3.37}$$

Where $y = \begin{bmatrix} e & \xi_d \end{bmatrix}^T$ is the desired output and $\xi = \begin{bmatrix} \xi_e & \xi_d & \xi_n \end{bmatrix}^T$ contains the combined states. The combined system matrices A, B, C, D, and N are

$$A = \begin{bmatrix} A_V & 0 & -B_V \\ 0 & A_d & 0 \\ 0 & 0 & A_n \end{bmatrix} B = \begin{bmatrix} B_V \\ B_d \\ 0 \end{bmatrix} N = \begin{bmatrix} 0 \\ 0 \\ B_n \end{bmatrix} C = \begin{bmatrix} C_V & 0 & -D_V \\ 0 & I & 0 \end{bmatrix} D = \begin{bmatrix} D_V \\ 0 \end{bmatrix}$$

The optimal MCA provides the design of an optimal scheme and a set of optimal parameters, subject to the assumptions of the workspace motion states and human vestibular model, by solving the Riccati equation.

A cost function J for the optimal MCA based on the LQR over $[t_0\ t_1]$, is specified as

$$J = E\left\{\int_{t_0}^{t_1} (e^T Q e + \xi_d^T R_d \xi_d + u_s^T R u_s) dt\right\} \tag{4.3.38}$$

where $e^T Q e$: the perception error cost (perception validity), $\xi_d^T R_d \xi_d$: the platform displacement cost, $u_s^T R u_s$: another way to take into account the constrained workspace, $[t_0\ t_1]$ is the simulation time, $E\{\}$ is the mathematical mean of a statistical variable (expected value), Q and R_d are positive semi definite matrices, and R is a positive definite matrix. Equation (4.3.38) indicates that three variables are constrained in the cost function. They are the sensation error, e, and extra terms, ξ_d and u_s, which together describe the tanslational and angular motions of the platform in the proposed cost function. Therefore, ξ_d is the state vector of the platform position and velocity, and u_s is the platform longitudinal acceleration. The cost function aims to minimize the human sensation error between the real and simulator skiers and to limit the platform motion. In design procedure of an optimal washout filter based on the LQR theory, several iterations are needed by changing the weighting matrices Q, R, and R_d for generating a suitable LQR-based optimal washout filter. The system equation and cost function can be transformed into the standard optimal control form Kwakernaak, (1972) [133], as follows:

$$\begin{aligned} \dot{\xi} &= A_t \xi + B_t u_t + N w \\ J_t &= \left\{\int_{t_0}^{t_1} (\xi^T R_{t,1} \xi + u_t^T R_2 u_t) dt\right\} \end{aligned} \tag{4.3.39}$$

where,

$$\begin{aligned} A_t &= A - B R_2^{-1} R_{12}^T,\ u_t = u_s + R_2^{-1} R_{12}^T \xi \\ R_{t1} &= R1 - R_{12} R_2^{-1} R_{12}^T,\ R_1 = C^T G C \\ R_{12} &= C^T G D,\ R_2 = R + D^T G D,\ G = diag[Q, R_d] \end{aligned} \tag{4.3.40}$$

The cost function of (4.3.39) is minimized when

$$u_t = -R_2 B_t^T P \xi \tag{4.3.41}$$

where P is the solution of the following algebraic Riccati equation:

$$R_{t1} - P B_t R_2^{-1} B_t^T P + A_t^T P + P A_t = 0 \tag{4.3.42}$$

By substituting (4.3.41) into (4.3.40) and solving u_s

$$u_s = -\left[R_2^{-1}(B_t^T P + R_{12}^T)\right] \xi \tag{4.3.43}$$

And specifying a matrix K, where $u_s = -Kx$, leads to

$$K = R_2^{-1}(B_t^T P + R_{12}^T) \tag{4.3.44}$$

where K can be subdivided, leading to the partition of the ξ state in the following equation:

$$u_s = -\begin{bmatrix} K_1 & K_2 & K_3 \end{bmatrix} \begin{bmatrix} \xi_e \\ \xi_d \\ \xi_n \end{bmatrix} \tag{4.3.45}$$

As $\xi_n = u_{pd}$, the states corresponding to ξ_n are eliminated, that is

$$\begin{bmatrix} \dot{\xi}_e \\ \dot{\xi}_d \end{bmatrix} = \begin{bmatrix} A_V & 0 & -B_V \\ 0 & A_d & 0 \end{bmatrix} \begin{bmatrix} \xi_e \\ \xi_d \\ u_{pd} \end{bmatrix} + \begin{bmatrix} B_V \\ B_d \end{bmatrix} u_s \tag{4.3.46}$$

By substituting the previous two equations and recalculating leads to the following equation:

$$\begin{bmatrix} \dot{\xi}_e \\ \dot{\xi}_d \end{bmatrix} = \begin{bmatrix} A_V - B_V K_1 & -B_V K_2 \\ -B_d K_1 & A_d - B_d K_2 \end{bmatrix} \begin{bmatrix} \xi_e \\ \xi_d \end{bmatrix} + \begin{bmatrix} -B_V(I + K_3) \\ -B_d K_3 \end{bmatrix} u_{pd} \tag{4.3.47}$$

After developing the state space form of (4.3.34) and (4.3.47), the equations in the Laplace domain can be obtained, as follows:

$$u_s(s) = W(s) u_{pd}(s) \tag{4.3.48}$$

$$\begin{aligned} W(s) &= \begin{bmatrix} K_1 & K_2 \end{bmatrix} \begin{bmatrix} sI - A_V + B_V K_1 & B_V K_2 \\ B_d K_1 & sI - A_d + B_d K_2 \end{bmatrix}^{-1} \begin{bmatrix} B_V(I + K_3) \\ B_d K_3 \end{bmatrix} \\ &- K_3 \end{aligned} \tag{4.3.49}$$

where $W(s)$ is a matrix of the optimized transfer function based on the LQR theory that links the simulator motion input, u_s to the skieur motion input, u_{pd}.

Investigation

1. In this method, the structure of the filter is part of the optimization process unlike previous approaches where the structure of the filter is completely predefined (classic approach) or partially defined (adaptive approach).

2. when the predefined trajectory is no longer a simple deterministic signal but a band of signals (filtered white noise), the optimization certainly becomes less precise but in return, we are left with a filter usable in real-time.

3. Note that Sivan, (1982) [116] were the first to integrate perception filters into the optimization process. Their article also offers a review of optimization techniques performed for flight simulation.

4. The equality between the perception models of the vestibular system in the context of the simulation and in the real context represents a strong hypothesis.

5. As in the adaptive case, coordination can be included in the algorithm by means of an adequate choice of the weight on the angular variation in order to keep it below the detection threshold.

6. In terms of implementation, we find in this method the difficulties present in the context of the adaptive approach, namely: the choice of weighting matrices, the choice of perception filters (to guarantee the stability of W (s)), etc.

7. Kinematic constraints are implicitly included thanks to the cost:
Cost= $(\omega_y,\ a_x) R (\omega_y,\ a_x)^T + \xi_d^T R_d \xi_d$. This strategy therefore favors a central operating zone (around ξ_d = 0) and does not allow the exploitation of the movement space in its entirety.

Simulator physical limit

The simulator mechanical limit is important in finding the weight matrices, therefore, we present the following table that shows the XY-6 DoF maximum of displacement, speed and acceleration.

D.o.F	Position	Max velocity	Max acceleration
X, Y	± 0.5 m	1.2 m/s	1.7 m/s^2
X_p, Y_p, Z_p	± 0.1 m	0.8 m/s	1.1 m/s^2
θ_x	± 15 deg	48.7 deg/s	71.6 deg/s^2
θ_y	± 15 deg	48.7 deg/s	65.8 deg/s^2
θ_z	± 15 deg	106 deg/s	154 deg/s^2
Max Γ	37 N.cm		

Table 4.3: XY-6 DoF motion interval

In the following we present the acceleration and the specific force restored by both classic and optimal approach, the predefined trajectory is described and detailed in (section 4.3.1).

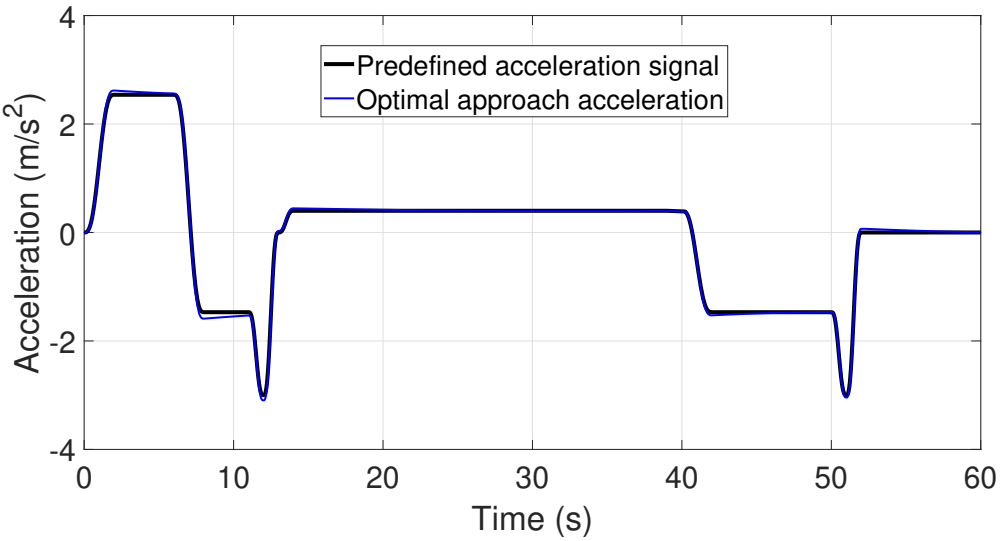


Figure 4.44: Desired acceleration from optimal algorithm

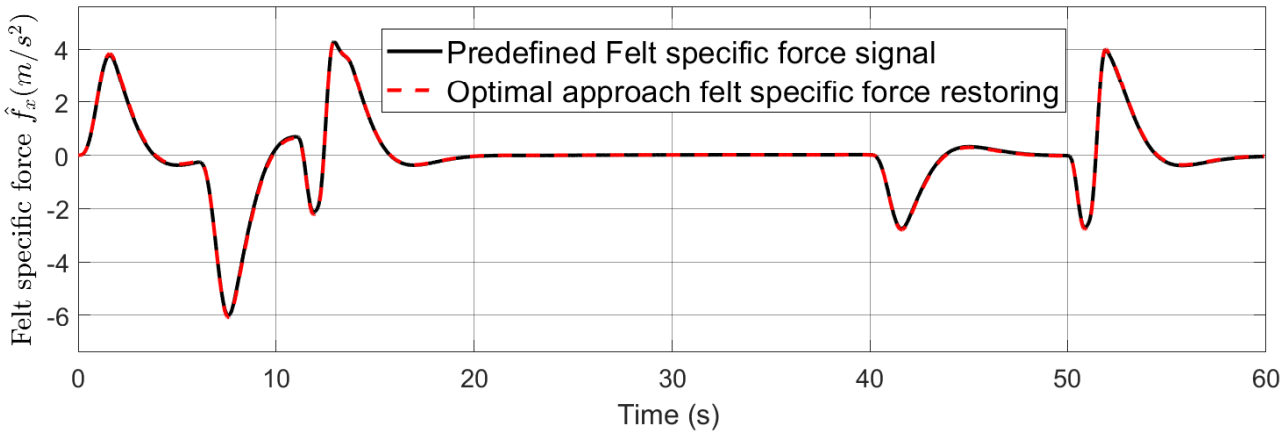


Figure 4.45: Desired specific force from optimal algorithm

The acceleration and specific force generated by optimal filter shows a better

perceptual restoration due to the coherence and proximity with the predefined trajectory. Fig. 4.46 shows respectively the displacement, speed and acceleration of the hybrid robot along the x-direction. Fig. 4.47 shows the tilt angle and tilt speed.

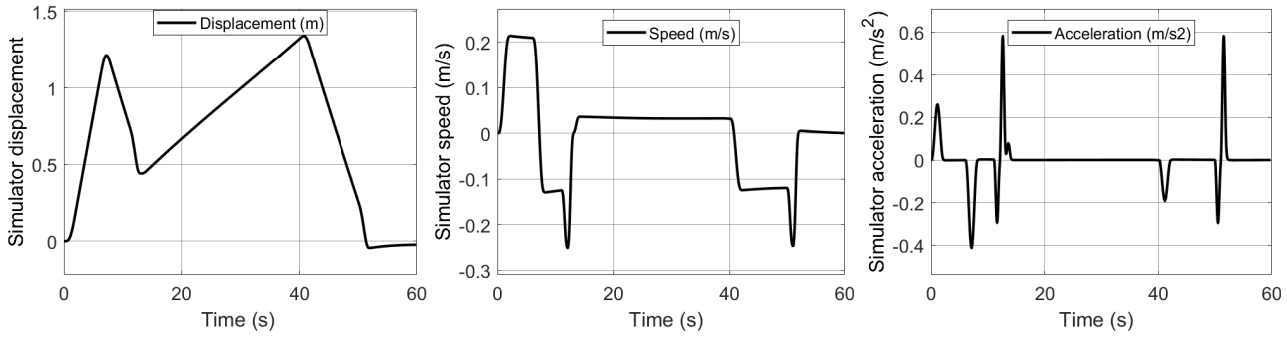


Figure 4.46: Acceleration, speed and displacement of X Table

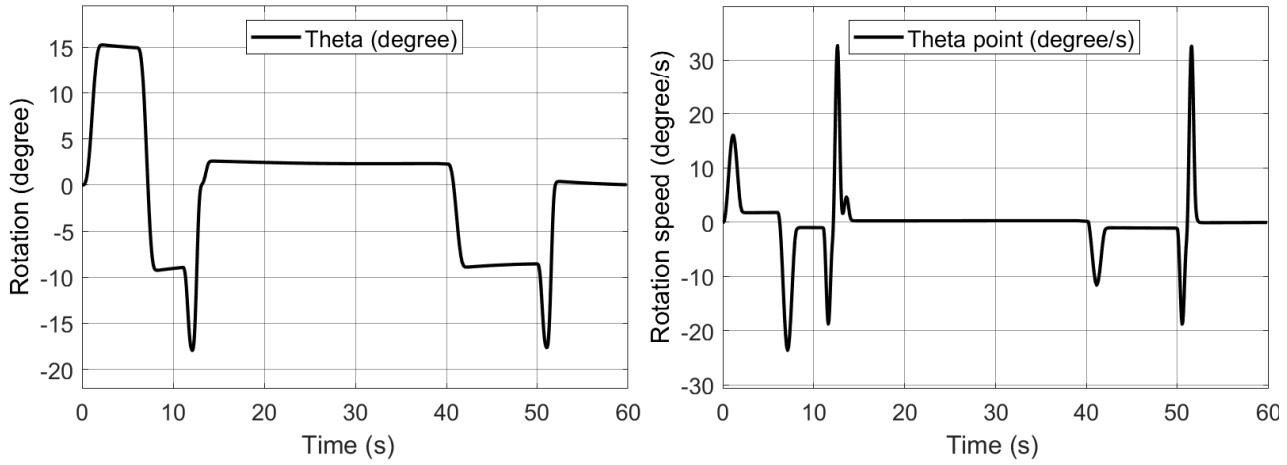


Figure 4.47: Tilt angle and speed

It is clear that the optimal approach can use more efficiently the platform workspace respecting the vestibular threshold, can go faster and can be more precise compared to the classic and adaptive approach.

4.3.6 The retained approach

The developed classic, adaptive, PSO-based and optimal MCAs have been implemented in the XY-6 DoF simulator. Fig. 4.48 and Fig. 4.49 sketche the ski test track predefined signals with all restored accelerations and specific forces. The test track (predefined signal) is recorded once for the proposed classic, adaptive, PSO-based and optimal approach and then compared to each other.
The ski test consists of skier acceleration during the descent and skier deceleration due to friction and stop. The acceleration procedure clearly shows that the strongest positive acceleration of the skier takes place from $2s$ to $6s$. It can be seen that, clearer in classical approaches, the XY-table system builds up the medium-frequency part of the acceleration, while the tilt coordination builds up the amplitude in a delayed low-frequency manner. It can be seen that the predefined ski signal is almost met by the combination of the table and tilt in the proposed adaptive approach. During constant acceleration such as from $8s$ to

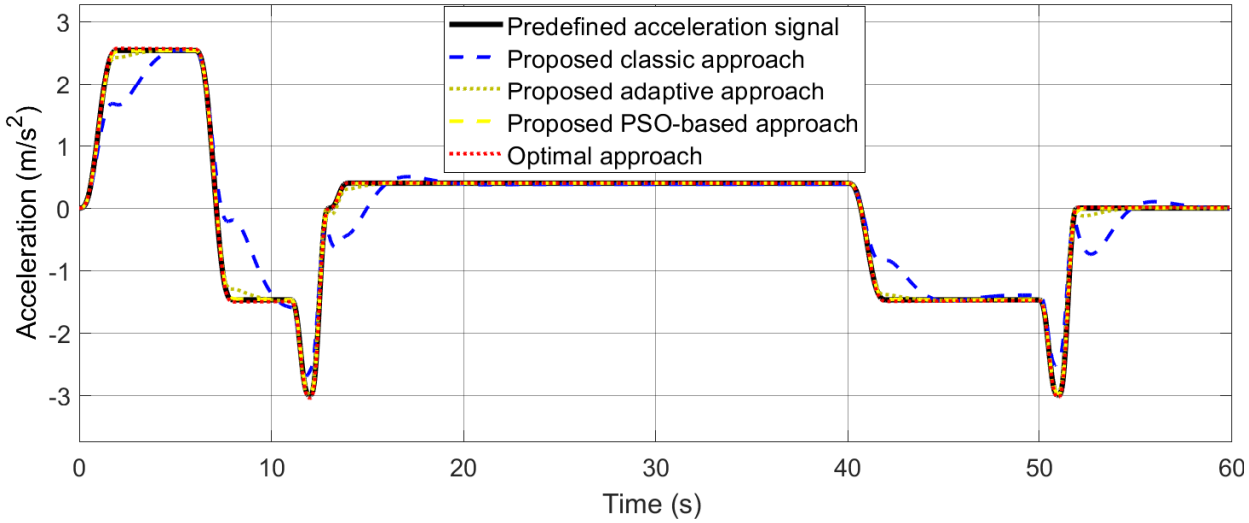


Figure 4.48: Motion Cueing Algorithms approaches comparaison

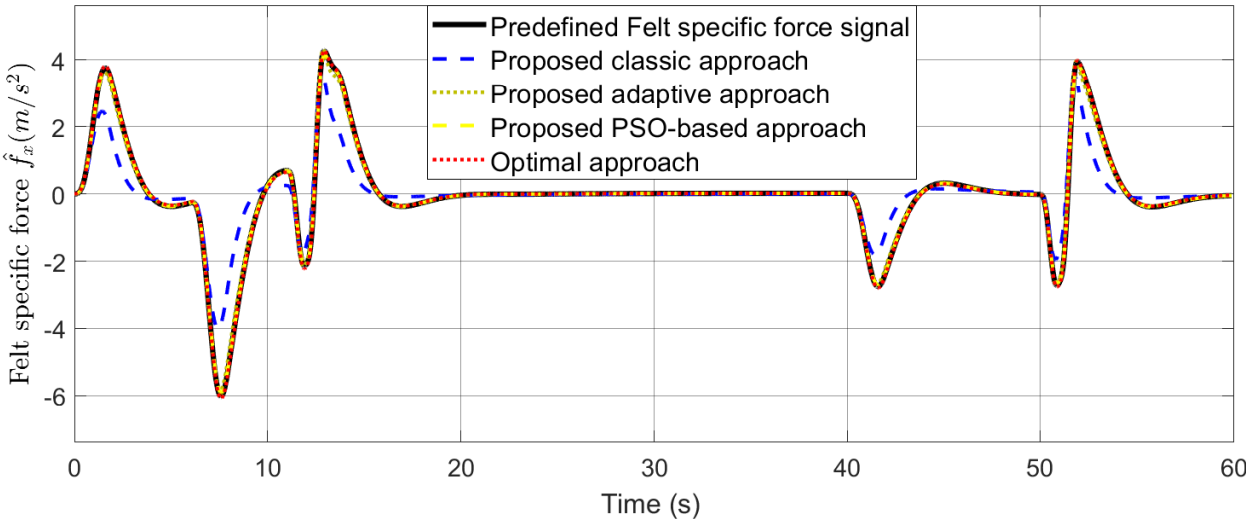


Figure 4.49: Motion Cueing Algorithms approaches comparaison

$12s$ and $16s$ to $42s$, the table system can return to its original position due to the washout effect while the hexapod maintains constant acceleration.
Fig. 4.50 shows the specific force errors for the proposed MCAs.

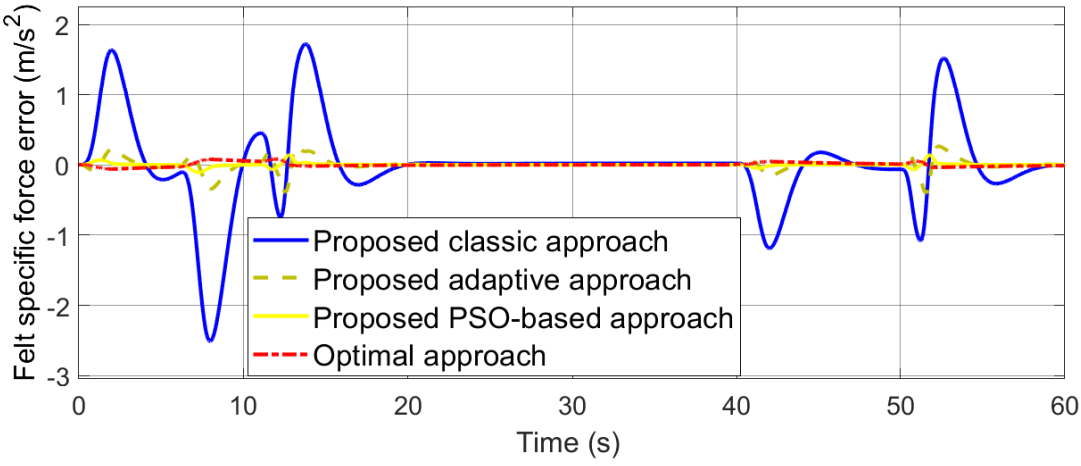


Figure 4.50: Specific force errors of the proposed MCAs

It can be seen that the proposed PSO-based and the optimal approaches have the lowest specific force error compared to others.
The following table shows the performance indicator (PI) for all developed MCA. Subscript "l", "m" denote literature and modified, respectively. The adaptive ap-

MCA	$Classic_l$	$Classic_m$	$Adaptive_l$	$Adaptive_m$	$PSO-based$	$Optimal$
PI	63.25 %	77.57 %	80 %	95%	96.66 %	97%

Table 4.4: Performance Indicator of MCA approaches

proach requires a huge computational load. Therefore, it will be ignored against the optimization approach even if it has a good PI.

4.4 Conclusion

We have proposed modified classical, adaptive and PSO-based motion cueing algorithms that present new approaches for existing motion cueing algorithms. We develop the optimal approach correspond to our platform specification. The idea is that the hexapod and the XY-table interact, providing a more realistic simulation of movement.

Actually, the PI of the proposed adaptive algorithm , PSO-based and of the optimal approach is 95%, 96,66% and 97%, respectively. The displacements of these optimzation approaches respect the vestibular organ threshold and have a better linear acceleration and angular speed then the proposed adaptive approach. Thus, no more false cues are presented to the skier, which is often the case with the standard approach of common motion cueing algorithms.

The PSO-based generates the corresponding desired trajectory for each subsystems; the XY table and the hexapod with PI lower than the optimal approach. However, the optimal approach generates the upper platform (end-effector) desired trajectory with the highest PI. Therefore, the optimal approach need a redundancy resolution to find each subsystem desired trajectory. And it will be detailed in the following chapter.

Chapter 5

XY-6 DoF redundancy resolution and control

5.1 Introduction

During the last years, an increasing number of kinematically redundant robots have been applied in various fields of robotics Kotlarski, (2010) [134], Satzinger, (2014) [135], Tanaka, (2014) [136]. This may be due to a focus on more and more constrained environments where the conventional six degrees of freedom (6-DOF) may prove insufficient for obstacle and singularity avoidance Orekhov, (2019) [137], Cha, (2007) [138], Baillieul, (1986) [139], but also, the real-time control of redundant robots has become less of a problem because of the vast increase in computing performance available Xia, (2004) [140], Zhang, (2020) [141]. Kinematic redundancy occurs when a manipulator has more degrees of freedom than those required to execute a given task. Thus, it is widely recognized that following an end-effector motion trajectory requiring six-degrees-of-freedom, a seven-joint robot arm is usually brought as an example of an inherently redundant manipulator. This implies that a given position and orientation of the end-effector determines an entire solution space rather than a unique solution and a criterion must be defined to select one of them.

To obtain a specific configuration (or joint velocities/torques), a redundancy resolution method should select the "most suitable" element in the solution space. The simplest method is based on the use of the Jacobian matrix pseudo-inverse [**?**]; this guarantees a least-squares reconstruction of the desired end-effector velocity. One drawback of this method is that, despite local minimization of joint velocities, singularity avoidance cannot be guaranteed Baillieul, (1984) [142]. Redundancy can be exploited to meet additional constraints on the solution of the inverse kinematics problem besides the typical end-effector task. A first possibility is to add a term for local optimization of a scalar criterion to the basic pseudo-inverse joint velocity solution; accordingly, a term proportional to the gradient of the criterion is projected onto the null space of the Jacobian matrix so that the end-effector task is not affected Klein, (1983) [143]. Several criteria have been proposed, e.g., to achieve maximum joint range availability Lukic, (2020) [144], obstacle avoidance Zhang, (2020) [141], minimization of joint torques Woolfrey, (2019) [145], maximization of different dexterity measures Chen, (2019) [146].

In chapiter 4, the MCAs were calibrated for the redundant simulator XY-6 DoF. Therefore, in this chapter, we develop a redundancy resolution method as well as robust trajectory tracking control. The new redundancy resolution method is about to make the simulator non-redundant with respect to the task by finding the appropriate motion distribution of the XY table and the hexapod. The trajec-

tory tracking control consists of using a sliding mode approach with optimized parameters. The sliding mode parameters are optimized using particle swarm optimization method to ensure better tracking of the desired trajectories.

5.2 Redundancy resolution for Optimal MCAs

As described in the chapter 4, the proposed modified classic and adaptive approaches as well as the PSO approach don't need a redundancy resolution. The algorithms by its design, based on the frequency seperation, can generate the desired hexapod displacement and rotation trajectories as well as the desired trajectory of the XY table. However, the optimal approach generates the desired diplacement and rotation of the end effector without consideration of the robotic redundancy therefore a redundancy algorithm should be applied to substitued the trajectory of the end effector to hexapod and table desired trajectories. Table 5.1 presents the MCAs that require redundancy resolution.

Modified MCAs	classic	adaptive	PSO-based	Optimal
Redundancy resolution	No need	No need	No need	**Crucial**

Table 5.1: Redundancy resolution of MCAs

There are many methods used to exploit the redundancy architecture, we cite on the one hand the optimization of criteria such as the minimization of joint torques and the obstacles avoidance, and on the other hand the kinematic control of the null space. In the context of motion cueing algorithms there are no elaborate methods. Therefore, we are going to propose a method to exploit the redundancy of the XY-6 DoF simulator suitable for its use. The study will be carried out on the X axis and by analogy will be scaled on the Y axis. Firslty, we propose to use the XY-6 DoF hexapod mainly for tracking and perform the translational task once the XY table is inefficient. That's mean that the XY table will make the largest translational displacement which is coherent with the purpose of adding an XY table to the hexapod.

Let's us define s_1 the first subsystem which is the hexapod, s_2 the second subsystem which is the XY table and finally the s_t the total system. $D_{s_1} = \{D_{x_{s1}},\ D_{v_{s1}},\ D_{A_{s1}}\}$ denotes the s_1 workspaces where $D_{x_{s1}},\ D_{v_{s1}},\ D_{A_{s1}}$ are the displacement, speed and acceleration workspace of s_1. By analogy, we have the same $D_{s_2} = \{D_{x_{s2}},\ D_{v_{s2}},\ D_{A_{s2}}\}$ for s_2 and $D_{s_t} = \{D_{x_{st}},\ D_{v_{st}},\ D_{A_{st}}\}$ for s_t.

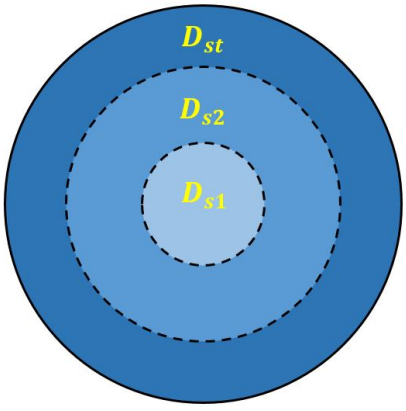


Figure 5.1: Workspace domains

We may define $D_{x_{s1}} = [0\ x_{max,s1}]$, $D_{x_{s2}} = [0\ x_{max,s2}]$ and $D_{x_{st}} = [0\ (x_{max,s1} + x_{max,s2})]$. By analogy, we have the same for D_v and D_A. Note that, $x_{max,s1}$, $v_{max,s1}$, $A_{max,s1}$ are lower than the $x_{max,s2}$, $v_{max,s2}$, $A_{max,s2}$, respectively. Therefore, the entire D_{s1}, D_{s2} relies on D_{s2} and D_{st},respectively. The following figure shows the workspace domain representation.

Let us define the set $M = \{M_x,\ M_v,\ M_a\}$ where the M_x, M_v, M_a denotes the maximum displacement, speed and acceleration of the predefined trajectory, respectively (obviously known for optimal approach). Fig 5.2 presents the displacement, speed and acceleration workspaces with its boundaries. The workspace boundaries are $(x_{si,max},\ v_{si,max},\ A_{si,max})$ where $i = 1,\ 2,\ t$. These boundaries are calculated from the real capacity of the platform motors using the direct kinematic model.

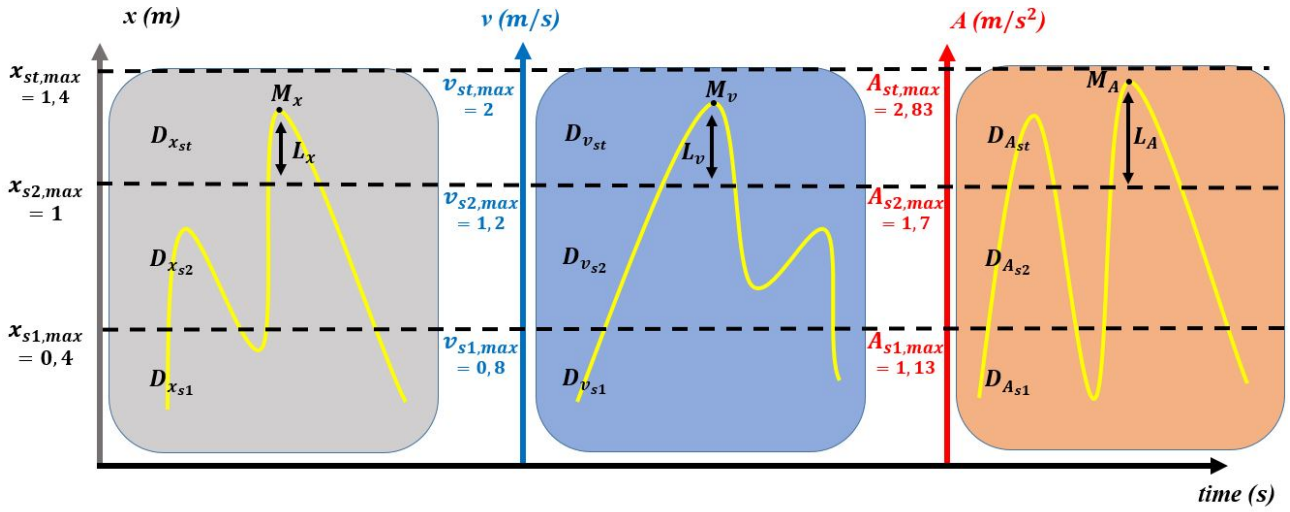


Figure 5.2: Displacement, speed and acceleration workspace domains

As described above the largest displacement will be mainly performed by the XY table. We present in the following the method proposed to exploit the XY-6 DoF redundancy. Firstly, if no one of the M points M_x, M_v, M_A lies on the $D_{x_{st}}$, $D_{v_{st}}$ or $D_{A_{st}}$ therefore the XY table can perform the translational displacement lonely. However, when one of the M points lies on the total system workspace the hexapod should intervene and help the XY table as will be described in the following.

Let us take the case where one or all of M points lie on the total system workspace, thus, $M_x \in D_{x_{st}}$ and/or $M_v \in D_{v_{st}}$ and/or $M_A \in D_{A_{st}}$. L_x, L_v and L_A denote the lack of displacement, speed and acceleration, respectively, where the XY table cannot performed. Therefore,

$$L_x = M_x - x_{s2,max}; \quad L_v = M_v - v_{s2,max}; \quad L_A = M_A - A_{s2,max} \tag{5.2.1}$$

In addition to rotation movement, these translational trajectories lack should be performed by the hexapod. Let us define the dimensionless variables of L_x, L_v and L_A

$$\begin{aligned} L_{x_{dl}} &= \frac{M_x - x_{s2,max}}{M_x} \\ L_{v_{dl}} &= \frac{M_v - v_{s2,max}}{M_v} \\ L_{A_{dl}} &= \frac{M_A - A_{s2,max}}{M_A} \end{aligned} \tag{5.2.2}$$

The total trajectory is designed by $x_{st}(t)$ which equal to the trajectory performed by the XY table $x_{s2}(t)$ and the hexapod $x_{s1}(t)$. The two subsystem should perform the task in the same time conserving the same total desired trajectory, thus the trajectory of each subsystem is composed as:

$$x_{s1}(t) = \alpha x_{st}(t) \tag{5.2.3}$$
$$x_{s2}(t) = (1-\alpha) x_{st}(t) \tag{5.2.4}$$

where α denotes the trajectory part the should be performed by the hexapod, in other word the maximum lack of the XY table trajectory. Thus, α is the maximum of displacement, speed and acceleration lack.

$$\alpha = \mathbf{max}(L_{x_{dl}},\ L_{v_{dl}},\ L_{A_{dl}}) \tag{5.2.5}$$

Since the hexapod model is non-linear thus the displacement, speed and acceleration capabilities are not similar. Therefore, we calculate the maximum displacement, speed and acceleration that can be performed by the hexapod by taking point M reling on the s_t boundaries. Therefore,

$$L_{x_{dl,max}} = 28.5\%, \quad L_{v_{dl,max}} = 40\%, \quad L_{A_{dl,max}} = 40\%, \tag{5.2.6}$$

We can see that even if the hexapod have bigger capacities in speed and acceleration than in displacement but it is still should be limited by the displacement. Therefore α will be limited by the minimum of the maximum lack which is $L_{x_{dl,max}}$.

$$\alpha = \mathbf{min[max(}L_{x_{dl}},\ L_{v_{dl}},\ L_{A_{dl}}),\mathbf{min(}L_{x_{dl,max}},L_{v_{dl,max}},L_{A_{dl,max}}\mathbf{)]} \tag{5.2.7}$$

Thus,

$$\alpha = \mathbf{min[max(}L_{x_{dl}},\ L_{v_{dl}},\ L_{A_{dl}}),L_{x_{dl,max}}\mathbf{]} \tag{5.2.8}$$

where $\alpha \in \mathbb{R}$. These limitation of α bring the system to another speed and acceleration boundaries that is sketched in the following figure.

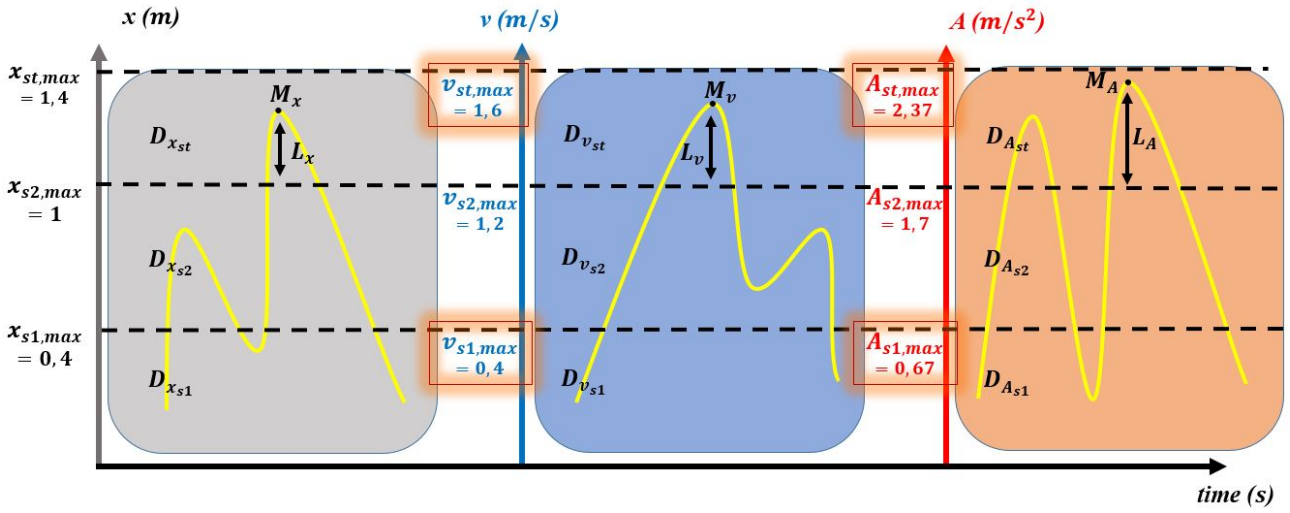


Figure 5.3: New domain definition

Once α is defined the total desired trajectory $x_{st}(t)$ can be divided into two desired trajectories of the XY table $x_{s2}(t)$ and the hexapod $x_{s1}(t)$, respectively.

We summarize the proposed algorithm of redundancy exploitation by:

Algorithm 2: Redundancy exploitation algorithm

Initialisation;
Application of the optimal MCA algorithm;
$M_x = \max(x_{st}(t))$; ($x_{st}(t)$– Total desired displacement);
$M_v = \max(v_{st}(t))$; ($v_{st}(t)$– Total desired speed);
$M_A = \max(A_{st}(t))$; ($A_{st}(t)$– Total desired acceleration);
$x_{s2max} = 1$; (X-Table displacement limit);
$v_{s2max} = 12$; (X-Table speed limit);
$A_{s2max} = 1.7$; (X-Table acceleration limit);
if $((M_x > x_{s2max}) || (M_v > v_{s2max}) || (M_A > A_{s2max}))$ **then**
  $L_{x,dl} = \frac{(M_x - x_{s2max})}{M_x}$ (X-table dimensionless displacement lack);
  $L_{v,dl} = \frac{(M_v - v_{s2max})}{M_v}$ (X-table dimensionless speed lack);
  $L_{A,dl} = \frac{(M_A - A_{s2max})}{M_A}$ (X-table dimensionless acceleration lack);
  $L_m = \max(L_{x,dl}, L_{v,dl}, L_{A,dl})$ (X-table maximum dimensionless lack);
  $\alpha = \min(L_m, 0.285)$;
else
  $\alpha = 0$;
end
Result: $x_{s1}(t) = \alpha\ x_{st}(t)$, $v_{s1}(t) = \alpha\ v_{st}(t)$, $A_{s1}(t) = \alpha\ A_{st}(t)$;
$x_{s2}(t) = (1-\alpha)\ x_{st}(t)$, $v_{s2}(t) = (1-\alpha)\ v_{st}(t)$, $A_{s2}(t) = (1-\alpha)\ A_{st}(t)$;

The upper platform desired trajectories generated using the optimal approach presented in section (6.3.5) are given here.

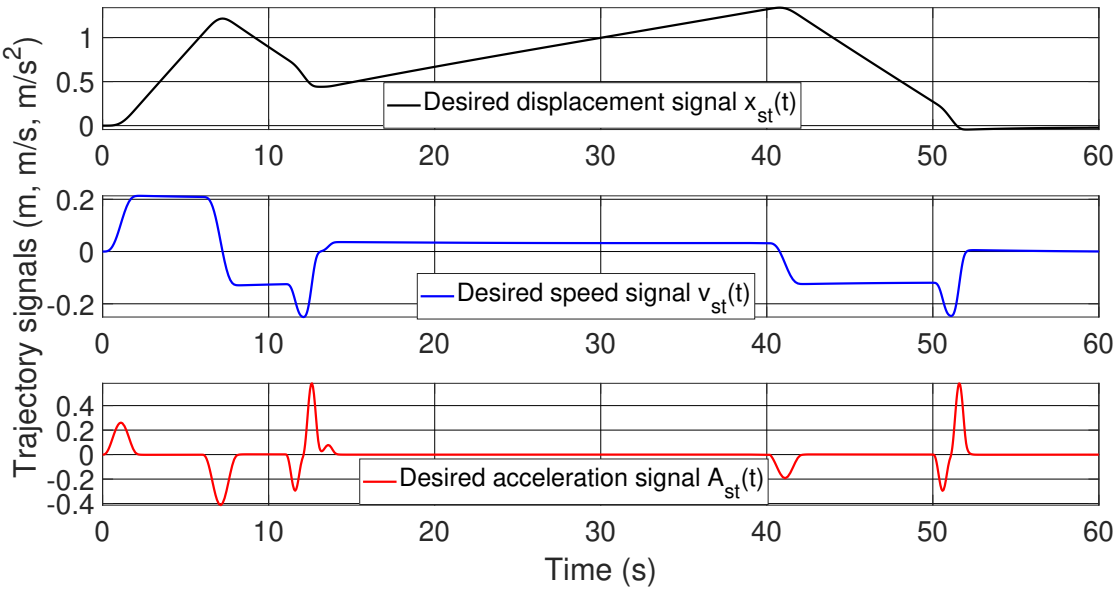


Figure 5.4: Desired total system displacement, speed and acceleration trajectories

The maximum displacement, speed and acceleration are $M : (M_x = 1.276\ m,\ M_v = 0.313\ m/s,\ M_A = 0.798\ m/s^2)$. M_x is bigger than the table displacement workspace which necessite the use of the hexapod. Thus we calculate the amount of the hexapod help $L_x = 0.276\ m$, $L_v = 0\ m/s$, $L_A = 0\ m/s^2$. Consequently, compared to our constraints, for this specific task, the hexapod will carry $\alpha = L_{x_{dl}} = 0.216$ of the total trajectories which correspond to 21.6% and the X-table will carry the 78.4%. The desired total trajectories are distinguished in hexapod and X-table trajectories as presented in the following.

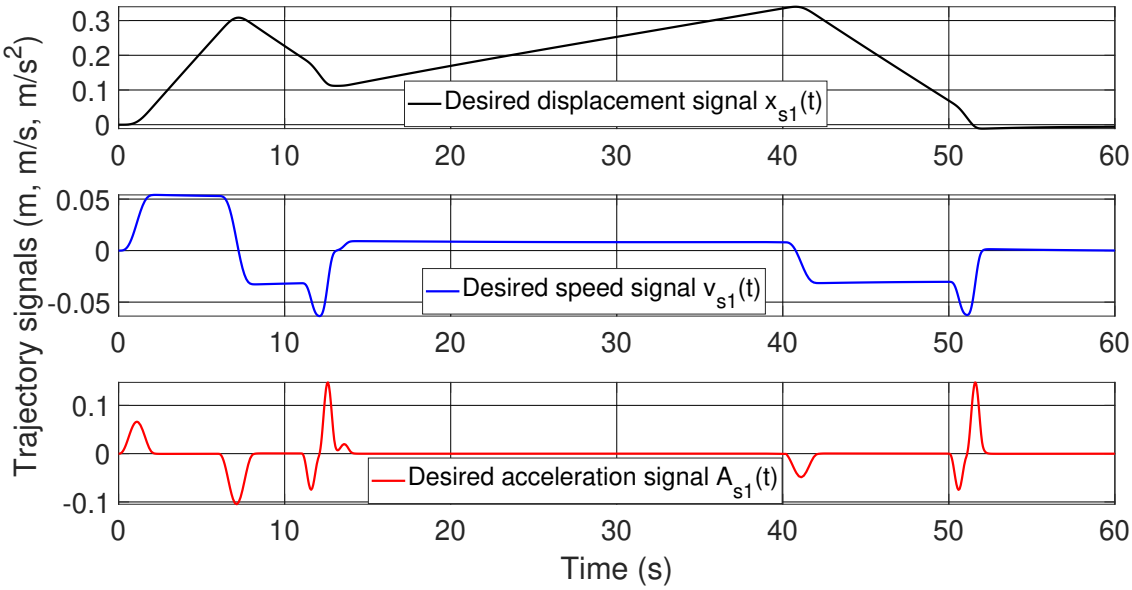


Figure 5.5: Desired Hexapod displacement, speed and acceleration trajectories

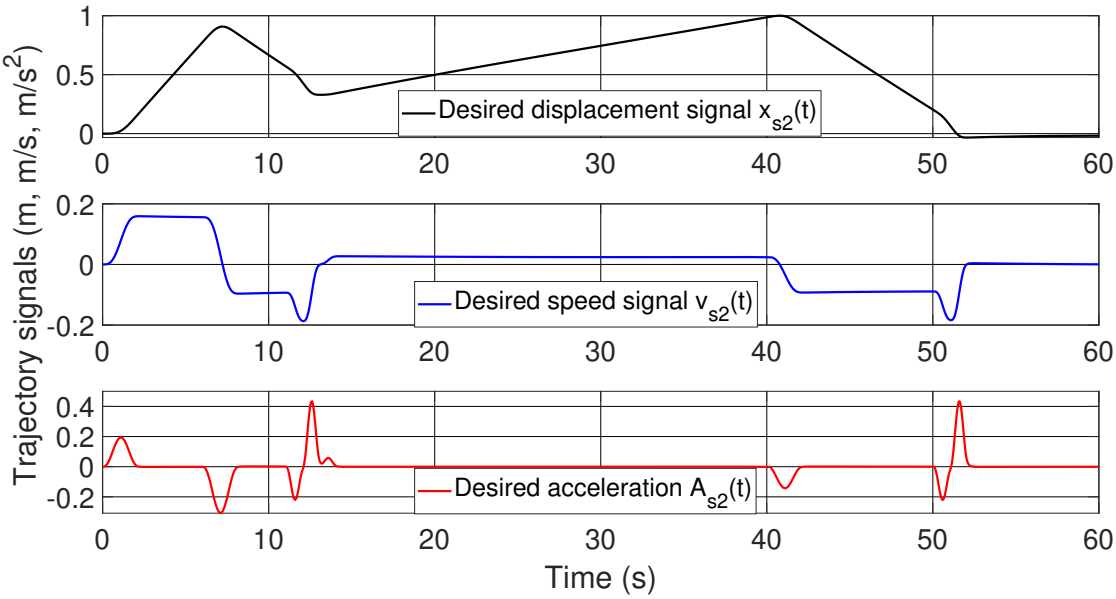


Figure 5.6: Desired X-Table displacement, speed and acceleration trajectories

In summary, we have designed a virtual reality scene using Unity where we extract the predefined trajectory from the real virtualized scene. Different MCAs have been applied to generate the desired trajectory, which is achievable by the XY-6 DoF simulator, from the predefined one. The two best MCAs in terms of highest performance indicator are the PSO and the optimal approaches. From the PSO algorithm we have obtained explicitly the desired trajectories of the XY table and the hexapod. However, using the optimal approach, we may deduce the upper platform behavior (at human vestibular level) that correspond to the total system trajectories. We therefore proposed the redudancy exploitation algorithm to generate the desired trajectories of each subsystem from the total ones. Recall that the human sensation is produced at the vestibular organ level where it is sensitive to translational acceleration and the rotational speed. Therefore, it is essential to define a motion control law to follow the desired trajectories. In the next section, we will develop the motion tracking control for the XY-6 DoF.

5.3 PSO- based sliding mode motion tracking control

Designing robust controllers to control robotic manipulator (coupled dynamic non-linear MIMO systems) is one of the most difficult tasks for many control engineers, especially when there is no complete knowledge of the system available Izadbakhsh, (2018) [147], Khorashadizadeh, (2018) [148], Arteaga-Pérez, (2019) [149]. In particular in the case of a highly non-linear system such as the XY-6 DoF simulator studied here [HOUDA1, HOUDA2]. The XY-6 DoF can suffer from structured and unstructured uncertainties, such as the variation of the payload, friction, external disturbances which require a robust controller to overcome them Chen, (2020) [150].

Sliding mode control presents a very good candidate to solve non-linear system control Van, (2020) [151], Baek, (2020) [152]. The sliding mode control also exhibits good behavior with respect to parameter uncertainties Norsahperi, (2020) [153]. However, the sliding mode suffers from the chattering phenomenon which is due to high frequency switching over discontinuity of the control signal Perruquetti, (2002) [154]. The sliding mode control parameters are not specified in the literature and they are tuned based on trial-error experimental method Ferrara, (2007) [155].

An optimization method to adjust the parameters of the sliding mode in order to have the best reaching rate and to avoid chattering problems is then necessary. As it is described in the "PSO-based" in the MCA approach, the particle swarm optimizer (PSO) presents an excellent optimization method to solve many control parameters in terms of optimality. In this section, we develop the sliding mode control appropriate for the XY-6 DoF, and then adjust the sliding mode parameters using the PSO-based method. At the end, we will present the experimental results.

5.3.1 Sliding mode motion tracking control

The sliding mode control Ma, (2017) [156] possesses strong robustness against the uncertainties since it does not depend completely on the accurate model of plant. In light of this, we synthesize a nonlinear robust sliding mode controller for motion tracking dedicated to the XY-6 DoF robotic platform.

Let us define the trajectory tracking errors $e_p \in \mathbb{R}^8$ of the XY-6 Dof joints as:

$$e_p(t) = q_d(t) - q(t) \tag{5.3.1}$$

where $q_d \in \mathbb{R}^8$ is the desired actuators (joints) trajectories. Let us define the sliding surface as in Slotine, (1991) [157]:

$$s(t) = \dot{e}_p(t) + \Lambda e_p(t) + \lambda \int e_p(t) \tag{5.3.2}$$

where $s \in \mathbb{R}^8$, $\dot{e}_p(t) = \dot{q}_d(t) - \dot{q}(t)$ is the tracking velocity error $\dot{e}_p \in \mathbb{R}^8$ in which $\dot{q}_d \in \mathbb{R}^8$ is the desired speed trajectory, $\Lambda = diag(\Lambda_i)$ with $\Lambda_i > 0$ for $(i = 1, ..., 8)$ is the diagonal positive definite constant matrix that determines the slope of the sliding surface and $\lambda = diag(\lambda_i)$ with $\lambda_i > 0$ for $(i = 1, ..., 8)$ is a diagonal positive definite constant matrix. Reminds that the system dynamics is expressed as:

$$\Gamma = M(q)\ddot{q} + H(q, \dot{q}) \tag{5.3.3}$$

where, $\Gamma \in \mathbb{R}^8$ is the joints torque vector, $\ddot{q} \in \mathbb{R}^8$ is the joints acceleration vector, $H(q,\dot{q}) \in \mathbb{R}^8$ is the coriolis and gravity vector and $M(q) \in \mathbb{R}^{(8\times 8)}$ is the mass matrix.

Making time derivative of (5.3.2), and then substituting (5.3.3) into it yields.

$$\begin{aligned} \dot{s} &= \ddot{e}_p + \Lambda \dot{e}_p + \lambda e_p \\ \dot{s} &= \ddot{q}_d - M^{-1}(\Gamma - H(q,\dot{q})) + \Lambda \dot{e}_p + \lambda e_p \end{aligned} \tag{5.3.4}$$

To improve the dynamic performance of sliding mode reaching motion, an exponential reaching law is selected as:

$$\dot{s} = -Ksgn(s) - \zeta s \tag{5.3.5}$$

where $K = diag(K_i)$ with $K_i > 0$ for $(i = 1,...,8)$, $\zeta = diag(\zeta_i)$ with $\zeta_i > 0$ for $(i = 1,...,8)$ are the control gains and $sgn(.)$ is the sign function.
Combining (5.3.4) and (5.3.5) yields the nonlinear robust sliding mode control law for the trajectory tracking as follows

$$\Gamma = M(\ddot{q}_d + \Lambda \dot{e}_p + \lambda e_p + Ksgn(s) + \zeta s) + H(q,\dot{q}) \tag{5.3.6}$$

To prove that the non-linear robust sliding mode control law shown in (5.3.6) used for motion tracking of the end-effector, the robotic system is asymptotically stable we choose the following Lyapunov candidate function which is positive definite :

$$V = \frac{1}{2} s^T s \tag{5.3.7}$$

Differentiating V with respect to timescale, then substituting (5.3.5) into it yields

$$\begin{aligned} \dot{V} &= s^T \dot{s} \\ &= s^T(-Ksgn(s) - \zeta s) \\ &= -s^T Ksgn(s) - \zeta s^T s \\ &= -K||s|| - \zeta ||s||^2 \leq 0 \end{aligned} \tag{5.3.8}$$

where $||.||$ denotes the vector/matrix Euclidean norm. Since V is a negative semi-definite function, based on the Lyapunov stability theory, the XY-6 DoF robotic system is asymptotically stable.

To further mitigate chatter phenomena, a saturation function which is continuous around the sliding surface $s = 0$ is employed to replace the "$sgn(s)$" in (**??**) so as to ensure s converges to zero smoothly. The saturation function "$sat(s)$" is given by Perruquetti, (2002) [154]:

$$sat(s) = [sat(s_i)],\ i = 1,...,8 \tag{5.3.9}$$

$$sat(s_i) = \begin{cases} 1, & \text{if } s_i > \xi \\ (\frac{1}{\xi})s_i, & \text{if } |s_i| \leq \xi \\ -1, & \text{if } s_i < -\xi \end{cases} \tag{5.3.10}$$

where ξ represents the thickness of the boundary layer. Therefore the sliding mode control law is as follow:

$$\Gamma = M(\ddot{q}_d + \Lambda \dot{e}_p + \lambda e_p + Ksat(s) + \zeta s) + H(q,\dot{q}) \tag{5.3.11}$$

5.3.2 Optimization of the sliding mode control parameters based on PSO

Solving the motion tracking Problem with a sliding mode control, leads to the control laws (5.3.11). The Λ, λ, K and ζ parameters are usually determined through experiments Solea, (2009) [158] and have great impact on the performance of the SMC. ζ and K are the SMC gains. where, the ζ_i parameters force the state x to approach the switching manifolds faster when s is large. And K_i parameters influence the rate at which the switching variable $s(x)$ reaches the switching manifold. The Λ and λ are the slope of the sliding surface.

Unfortunately, there is no direct method exist in the literature to find these parameters because of the nonlinearities and the coupling effect of the robotic systems. Choosing parameters through experiments only depends on experience or repeated debugging. PSO is an efficient method to choose the optimized parameters of the sliding mode controller. The main PSO advantages are: simplicity and efficiency, proven in many other parameters training problems (Mendes, (2002) [159], Kim, (2008) [160], Serbencu, (2010) [161], Zadeh, (2016) [162]).

PSO detailed description was carried out in the section (1.3.4) Desired trajectory based on PSO approach. For this application, PSO algorithm generate, at each step, a number of solutions equal to the number of particles. Let us take the j^{th} particle position and its corresponding velocity as follow: $x_j = [\Lambda_i, \lambda_i, K_i, \zeta_i]$ with $i = 1,2,...,6$ and $v_j = [v_{j,1},\ v_{j,2},...,\ v_{j,32}]$ where $x_j \in \mathbb{R}^(32\times 1)$ and $v_j \in \mathbb{R}^(32\times 1)$. The initialization of a swarm particules is done with a population of random solutions. During the optimisation process that update the swarm particules positions and velocities require importants values. The personal $p_{b,j}(t)$ best position and the global best position $g_b(t)$ and they are updated following these equations:

$$v_k(j+1) = \omega \underbrace{v_j(t)}_{\text{Current motion}} + c_j \underbrace{r_1(p_{b,j}(t) - x_j(t))}_{\text{particle memory influence}} + c_2 \underbrace{r_2(g_b(t) - x_j(t))}_{\text{swarm influence}} \tag{5.3.12a}$$

$$x_j(t+1) = x_j(t) + v_j(t+1) \tag{5.3.12b}$$

Recall that c_1, c_2 represent the acceleration coefficients also known as self confidence, r_1, r_2 represent the random vectors , and ω represents the inertial factor that make a compromis between exploration and exploitation and it's value goes from 1.1 to 0.1 to ensure a better convergence of the PSO-method [128]. Note that $x_k(0) \sim U(x_{min}, x_{max})$ where $U(a,b)$ represents the uniform distribution between a and b. To avoid any non feasible solution, the particule swarm can be borned $x_{lo} \le x \le x_{up}$ where x_{lo} and x_{up} indicate the lower and upper limits, respectively. It should be noted that the size of the particle group should not be too large or too small, as there are many possible locations towards the best solution. This is important when the overall value of the particle is uniformly distributed in the search area [127]. The PSO algorithm terminates either when a predefined fitness value is obtained by at least one particle, or when the defined maximum number of generations (iterations) is reached.

The quality of the solutions is evaluated with the fitness function. The fitness function used in PSO takes into account booth the speed of reaching manifolds and the amplitude of the chattering. The literature offers a variety of functions such as the Integral Absolute Error (IAE), Integral of the Squared Error (ISE) and Integral of the Time weighted Squared Error (ITSE). We propose to define the fitness funcion as the sum of Root Mean Square (RMS) of the accelerations errors. Root mean square error is an old, proven measure of control and quality. The

fitness function, therefore, can be expressed as:

$$F(\ddot{e}_p) = \sqrt{(\sum \ddot{e}_p(i))/N} \tag{5.3.13}$$

where $\ddot{e}_p(i)$ is the acceleration error of the i^{th} sample and N is the sample number.

The PSO optimisation algorithm is working off-line. The results of the algorithm are the sliding mode controller parameters. The parameters found by PSO can be used in real-time implementation of PSO-based SMC controller on the XY-6 DoF. We present in the following figure the structure of the proposed PSO-based sliding mode control for $(j+1)^{th}$ particule.

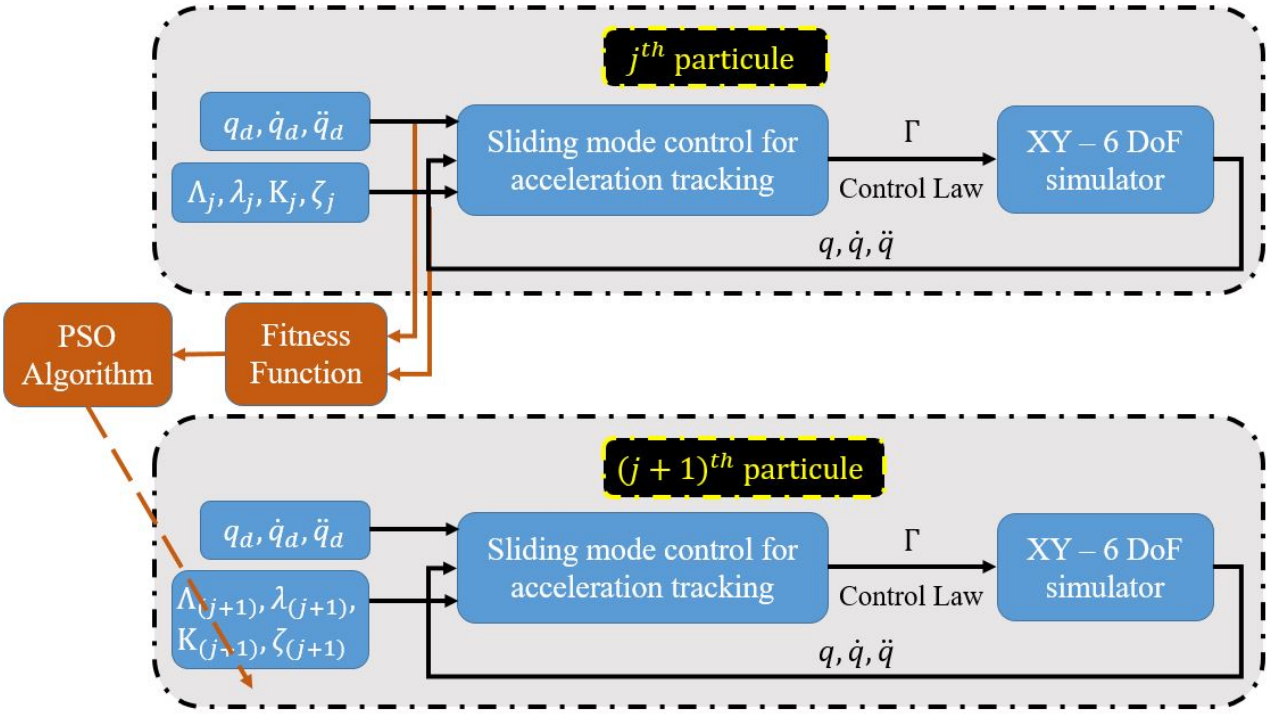


Figure 5.7: Proposed PSO-based SMC optimized schema

The next section will be devoted for the application of the proposed optimzation of the sliding mode control based on the particule swarm optimization.

5.3.3 Simulation results

The PSO-based sliding mode control is applied on the XY-6 DoF simulator. The table desired displacement, speed and acceleration trajectories are defined in Section (5.2) in Fig. 5.7. The hexapod desired translational displacement, speed and acceleration are also defined in Section (5.2) in Fig. 5.5 and the hexapod desired rotation and rotation speed are defined in the optimal approach section.

Therefore, based on geometric and kinematic model we can generate the desired joints (legs) trajectories of the hexapod presented in Fig. 5.8.
In the experimental study, the PSO parameters used were as follows: swarm size 100; iteration limit 50, and inertia range is [0.1, 1.1]. Fig. 5.9 presents the convergence of the PSO method.
As we can be seen in Fig. 5.9, the convergence has been reached partially at 10th iteration and totally at 18th iteration. In the following, we will present the displacement, speed and acceleration errors of the sliding mode control (SMC)

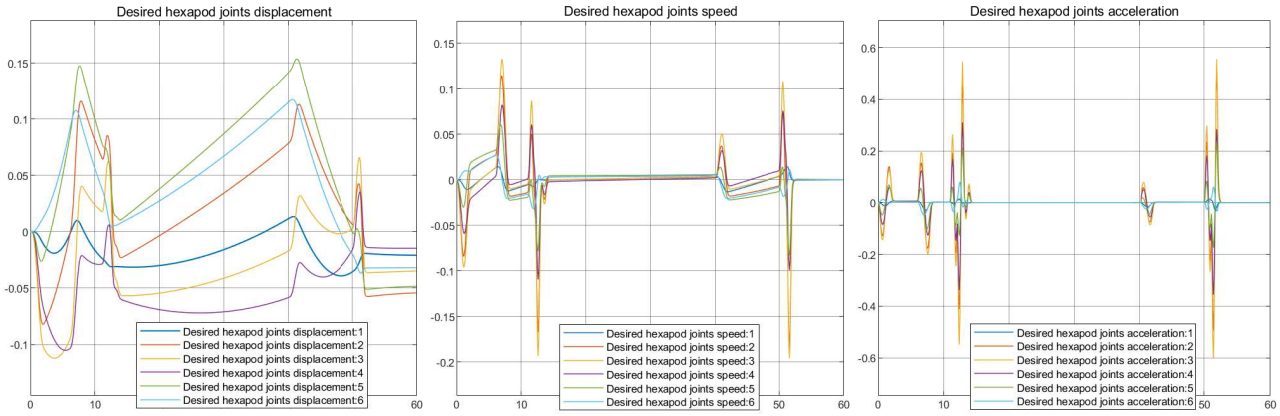


Figure 5.8: Desired hexapod joints

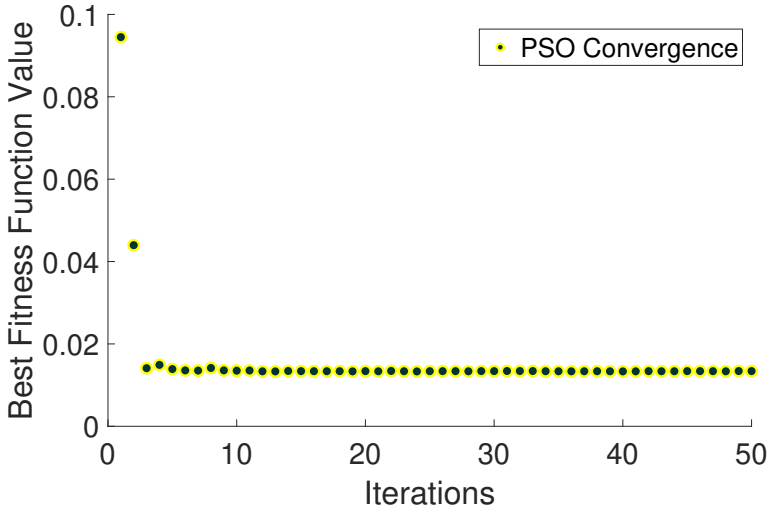


Figure 5.9: Fitness value over iteration

where the parameters are set by the trial and error method and of the corresponding PSO-based SMC tuned parameters. Trajectories and control inputs of the X-table and the hexapod are also presented.

Fig. 5.10 and Fig. 5.11 present the table displacement, speed and acceleration errors for SMC and PSO-based SMC.

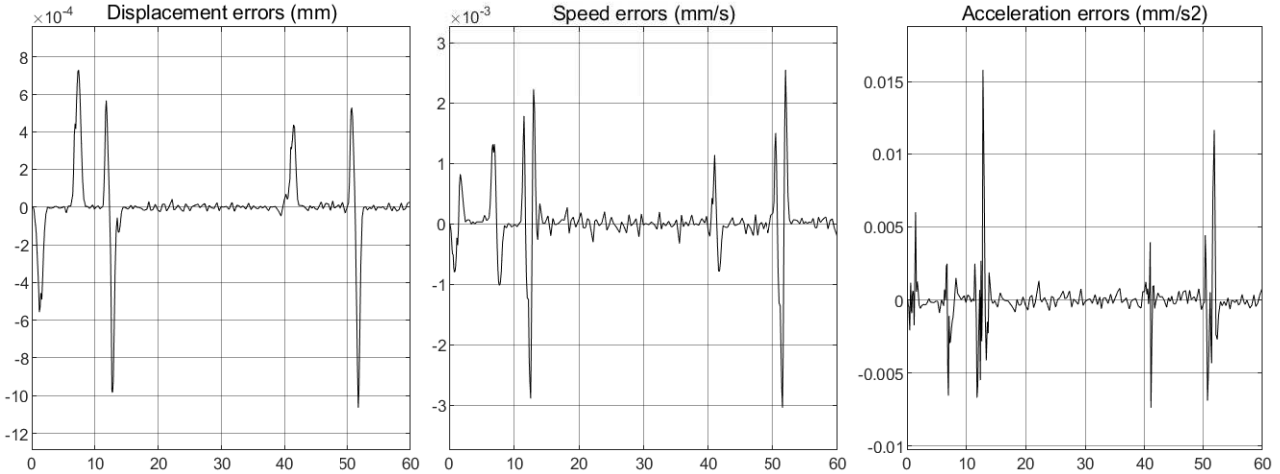


Figure 5.10: Table trajectories errors for SMC

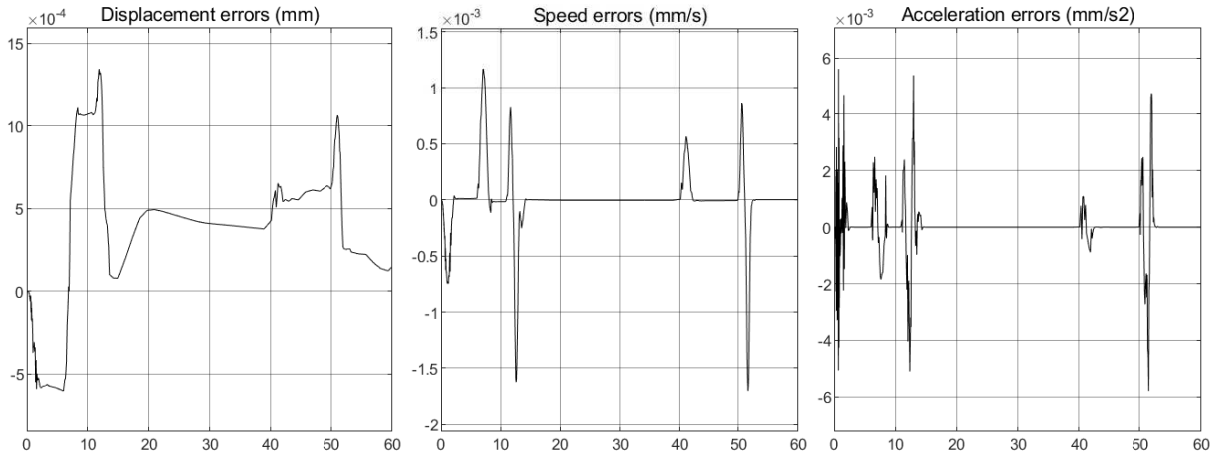


Figure 5.11: Table trajectories errors for PSO-based SMC

The displacement, speed and acceleration of the PSO-based SMC presents no chattering issue. The acceleration error of the SMC based on PSO is lower than that of the SMC due to the definition of the fitness function which decreases the acceleration error. Fig. 5.12 and Fig. 5.13 present the table displacement, speed and acceleration for SMC and PSO-based SMC.

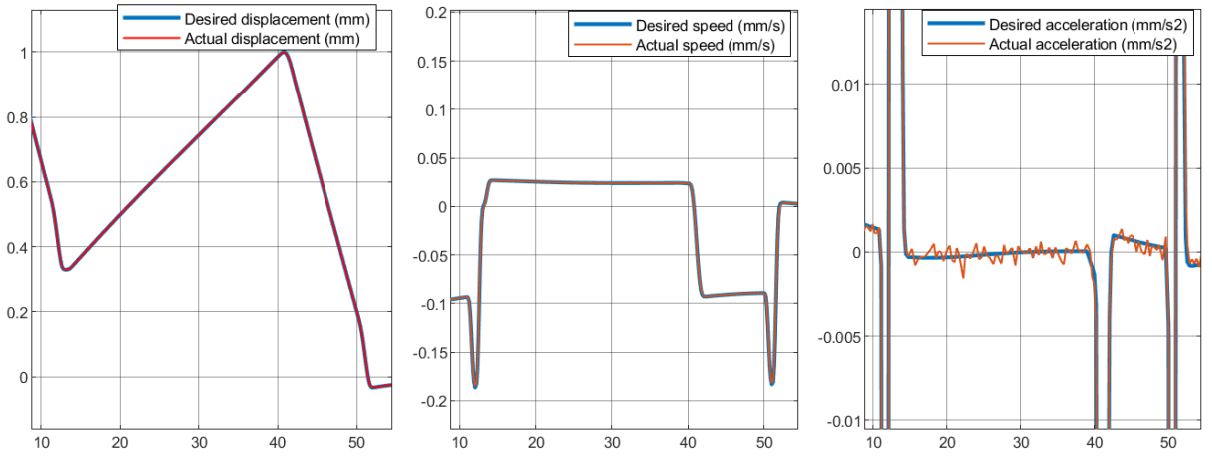


Figure 5.12: Table trajectories for SMC (near view)

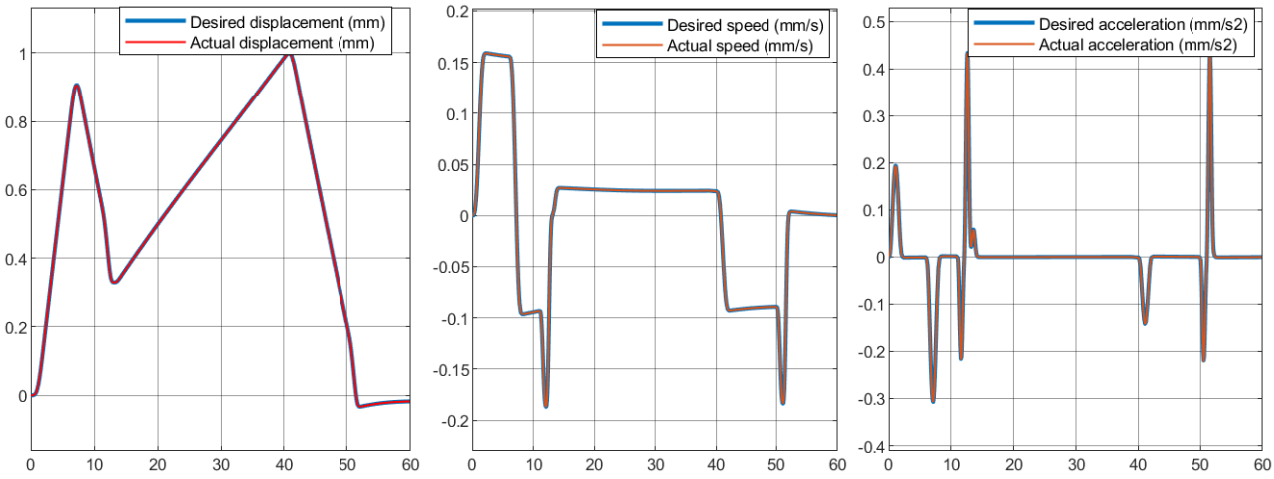


Figure 5.13: Table trajectories for PSO-based SMC

Fig. 5.14 and Fig. 5.15 present the table control input for SMC and PSO-based SMC.

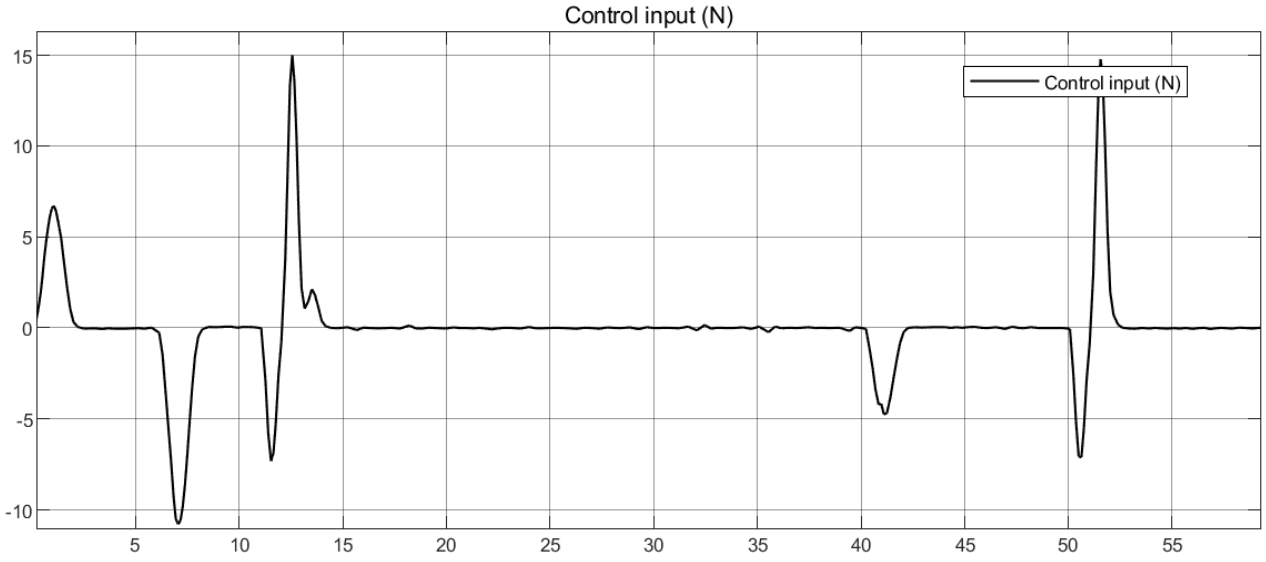


Figure 5.14: Table control input for SMC

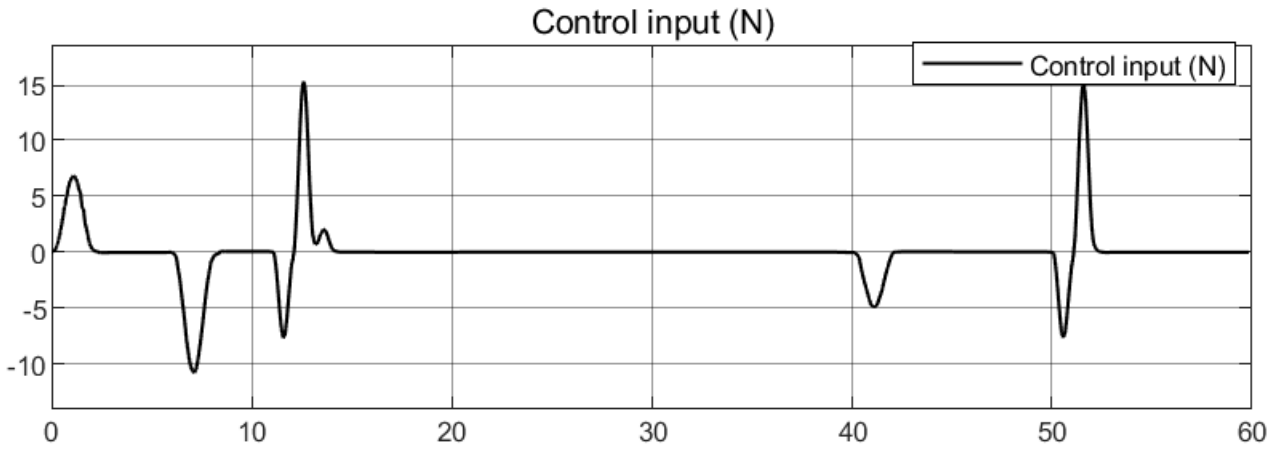


Figure 5.15: Table control input for PSO-based SMC

Fig. 5.16 and Fig. 5.17 present the hexapod displacement, speed and acceleration errors for SMC and PSO-based SMC.

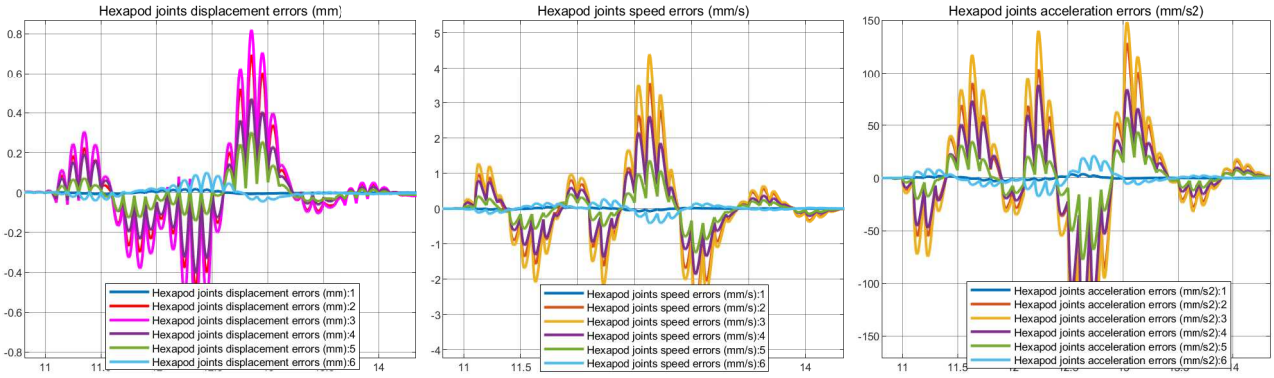


Figure 5.16: Hexapod trajectories errors for SMC (near view)

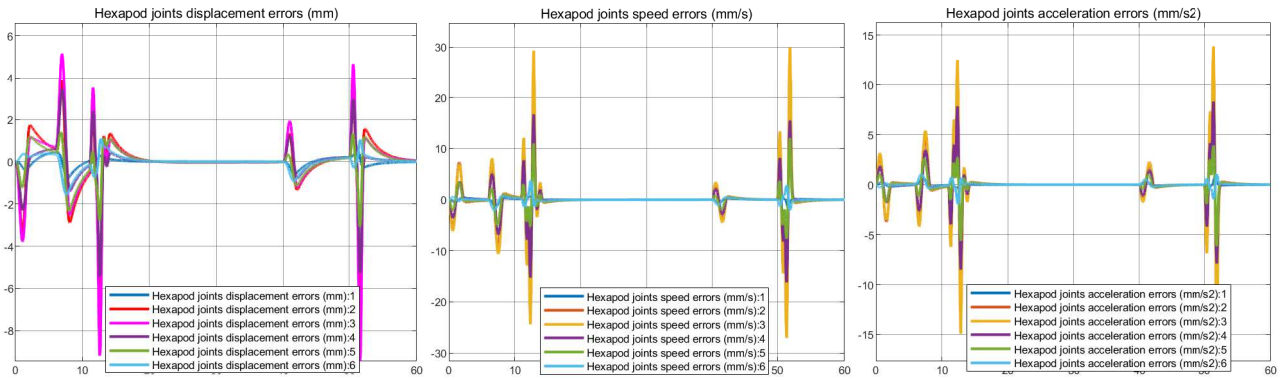


Figure 5.17: Hexapod trajectories errors for PSO-based SMC

Fig. 5.18 and Fig. 5.19 present the joints acceleration for SMC and PSO-based SMC.

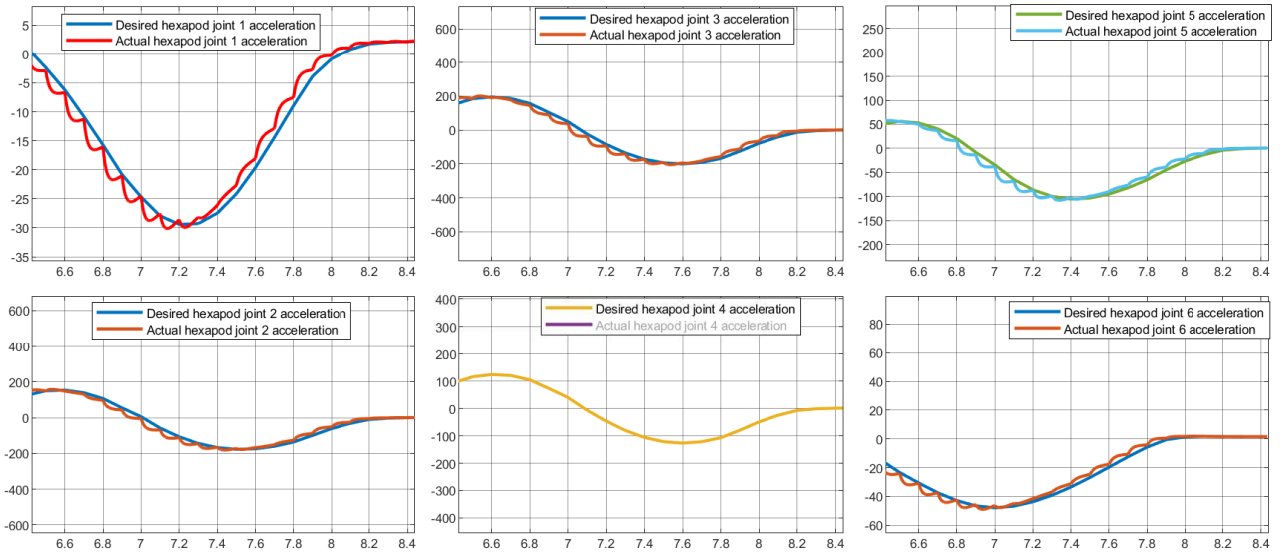


Figure 5.18: Hexapod joints acceleration for SMC (near view)

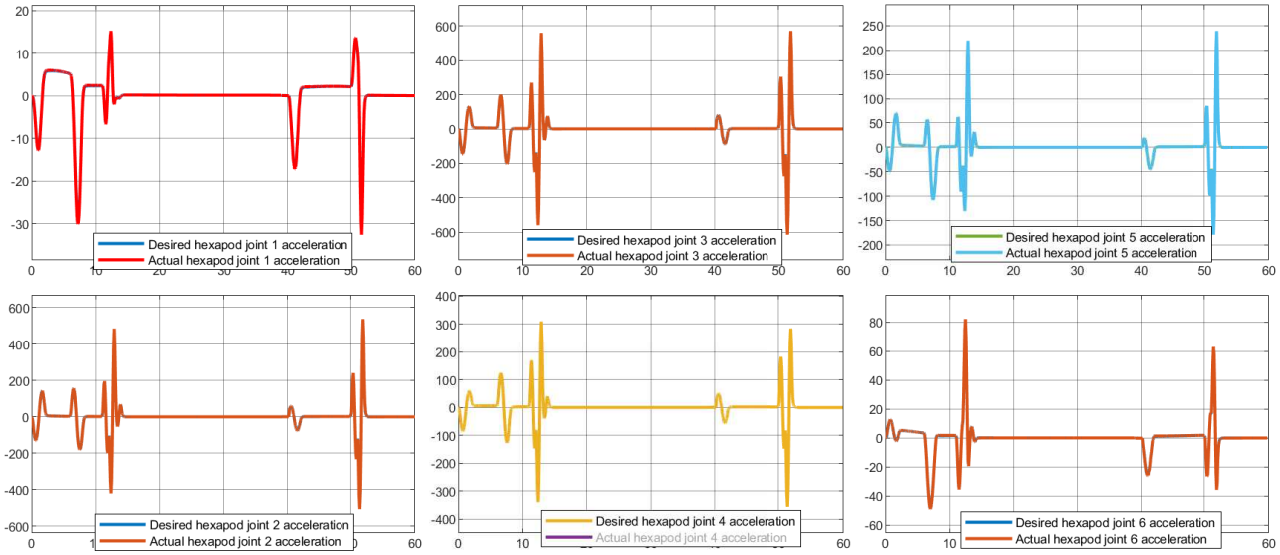


Figure 5.19: Hexapod joints acceleration for PSO-based SMC

Fig. 5.20 and Fig. 5.21 present the hexapod control input for SMC and PSO-based SMC.

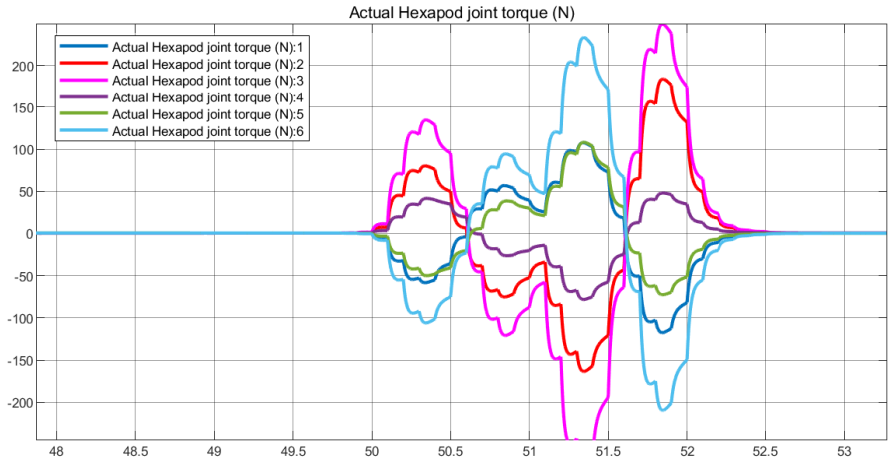


Figure 5.20: Hexapod control input for SMC (near view)

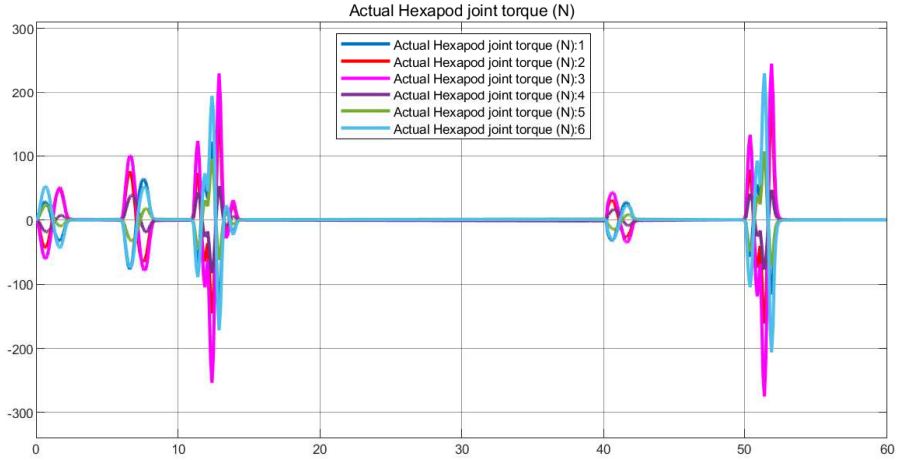


Figure 5.21: Hexapod control input for PSO-based SMC

These figures show that the PSO-based sliding mode control provides a better control specification in terms of the fast response and trajectories tracking, especially for acceleration trajectories. Moreover, the proposed PSO-based SMC shows a good disturbances rejection and a good chattering phenomena avoidance.

5.4 Conclusion

In this chapter, we proposed a new method for redundancy resolution dedicated to the XY-6 DoF platform. Then, this method is applied for the desired trajectories of the hybrid robotic platform coming from the optimal approach. As results, two desired trajectories are generated for each subsystem.

In addition, we proposed a non linear sliding mode controller for the XY-6 DoF. The major problem when using SMC is the chattering phenomena. To eliminate disturbance effect we propose to optimize the SMC parameters using the particule swarm optimization approach. The PSO-based SMC for XY-6 DoF gives a promise results in terms of fast convergence and motion tracking.

Chapter 6

IBISC ski simulator capabilities and Ski VR scenes construction

6.1 IBISC ski simulator experimental capabilities

6.1.1 Introduction

The loss of autonomy and mobility, for a large number of individuals around the world about 15% from the world population in 2019 World Health Organizations (2019), is mainly due to road accident victims, people with disabling diseases like multiple sclerosis and of course elderly people living alone. A significant percentage of the population, around 10%, use wheelchairs.

The parallel manipulator has been widely used in industrial and human joint rehabilitation Chen (2019) applications due to their high dynamic performances. Compared with serial manipulator architecture, parallel robots have some special characteristics in terms of very high accuracy, greater structural rigidity, and high payload capacity. However, it has a common major drawback in terms of limitations of the workspace. The hybrid robots that appear in form of one or two series axis carrying a parallel platform aims to overcome the limited workspace of parallel manipulators and improves the acceleration restitution. In this perspective, we have proposed to design a low cost robotic platform, XY- 6 DoF simulator, that is intended to be used for disabled skiers Amouri, (2016) [22] presented in Fig. 6.1.

Figure 6.1: XY- 6 DoF Simulator CAD

It must provide a realistic ski simulation feeling suitable for people with reduced mobility. However, the tests and works implemented can also used for other sliders applications (water skiing, jet skiing, ...). Several researches have proven that this type of device ensures psychological, muscular and social benefits to people with severe disabilities Abellard (2010), Jeonghun (2002).

Safety is a key issue for a real robotic structure and especially in human-robot interaction. In order to avoid human injury and structural damage, a definition of the safety environment will be detailed. The simulator performance and technical tests will also be carried out to verify the robotic platform operation and capabilities. Recall that, all electronic, mechanical and technical components are presented in detail in appendix A.

6.1.2 Safety environment definition

The actuators should control the robot translational movement along the x-axis (longitudinal displacement), the y-axis (transverse displacement) and z-axis (vertical displacement) as well as the robot rotation along the three axis x (roll), y (pitch) and z (yaw). To do this, we have used the Nanotec brand motors, specifically PD4-CB brushless motors. In total, 8 motors of this type construct the system. They determine the behavior of the platform at all times and must be managed effectively to avoid any accident and to achieve the platform intended use.

Since the safety is a primary issue, before any use of the platform we must solve the localization problem. Therefore, first, it is necessary to launch the process of detecting the physical limits of the platform to define a reference position and thereby to create software security to guarantee structural integrity. We will present in the following the physical limits definition.

The concept is to use the physicals limits of the structure to get back reliable data about the position, then define a software security margin of approximately $5\ cm$. For this reason, we applied a vertical movement to detect the horizontal limit of the platform's edge and then the same process with a horizontal movement to detect the vertical limit, thereby the platform is located in the structure corner. It is necessary to apply a slow velocity to prevent damage of the structure when the platform reaches the edges.

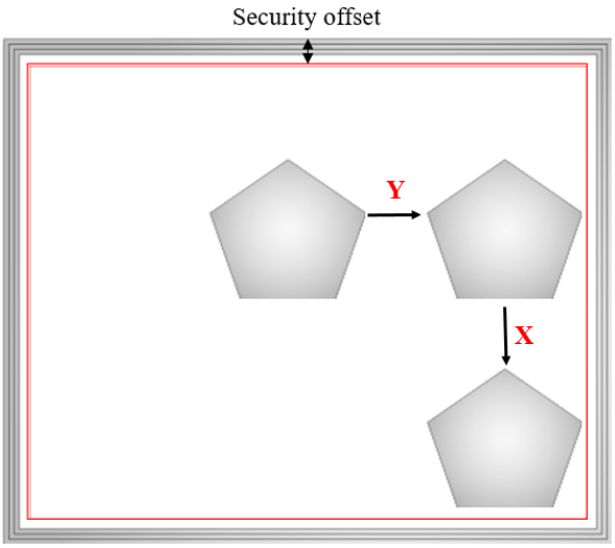


Figure 6.2: Security Offset

The problem is to be able to detect a limit, in order to solve it we compare the distance difference between 2 positions acquired. First, some information about the motor is required to contextualize the movement:

- 1 motor incrementing = 0.18 rotation degree of the motor shaft = $0.0126 \pm \Delta$ mm linear axis displacement.
- 1 motor revolution = 2000 motor incrementing = $25.2 \pm 2000\Delta$ mm linear axis displacement.

where Δ is the measurement uncertainty and depends of displacement velocity, in our case it is around 5 % of the theoretical displacement, i.e $0.63.10^{-3}mm$.
This measurement uncertainty comes from the gear transmission error and the motor characteristics. There is a link between the incremented position p_i from motor encoding position and real position p_r in millimeter as:

$$p_i = \frac{2000\ p_r}{6,5\pi} \tag{6.1.1}$$

For a specified velocity, we compute the desired variation of position between 2 procurations to deduce the impact with a real-time of $0,006$ second which corresponds at position acquisition time. Let us define Δ_x, Δ_y the margin in the x and y direction, respectively.

$$\Delta_x = |x(t_i) - x(t_{i-1})|,\ \Delta_y = |y(t_i) - y(t_{i-1})| \tag{6.1.2}$$

$$V_s = \frac{V \times 25,2}{60} \tag{6.1.3}$$

where, the speed is expressed in mm/s. Finally, we get the equation to compute the threshold value th of the side detection:

$$th = h\ V_s \tag{6.1.4}$$

where, h is a constant calculated from experimental trial and error method. And the resolution of the detection to stop the effective movement after 10 position reports:

$$\Delta_x < th,\ \Delta_y < th \tag{6.1.5}$$

Another way is to detect the fluctuation of the electric power, instead of the position fluctuation detection, which increases if we touch a border because of additional energy requirement due to the constraint of the physical edge. From there the platform is in the corner, we reset the motors encoder position sensor and this point becomes the new reference world map. Now we can design software protection to prevent impact for each movement in the structure, the safety area is defined in Fig. 6.3.

This function is very important especially when the simulator is running at high speeds, unsafe trajectories can lead to system failure or destruction. After the system coordinate center definition and the mechanical stops, the security of the platform is thus established thanks to programming which takes into account all its limitations. All trajectories given to the system are then processed if they are allowed, the system will follow the order, otherwise it will stop and display a security message.

In the practical work, safety is the first important issue, because even if there is a study guaranteeing the safe trajectory, a sudden stop is also necessary to

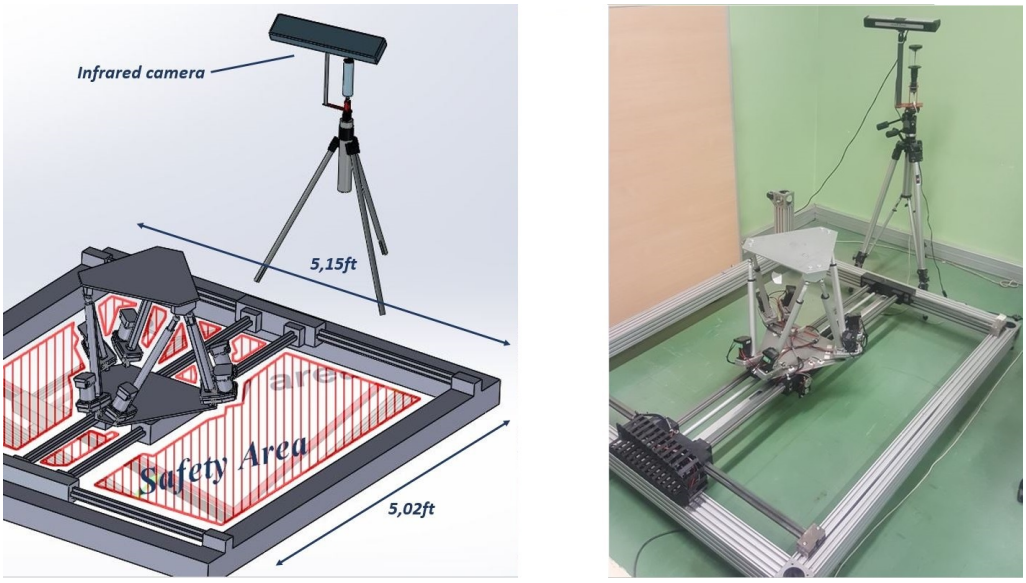


Figure 6.3: Defined safety area of the XY-6 DoF simulator

avoid any malfunction of the system. A very fast stop of the simulator following a predefined deceleration rate is done to reinforce the security of the platform.

A motion calibration using smartTrack should be done. Once the security environment and the security area have been defined, the movements are guaranteed to be safe. However, it is then necessary to check the position by means of external sensor different form the motors sensors. External acquisition through the infrared camera SmarTrack operating frequency in 60 Hz allows recovering an extern acquisition of position data in order to design a trajectory control. This camera can get the current position of the platform in real-time and with a representation of 6 DoF about positions and velocities acquisitions. It needs to grade the camera using the position of the reference point from the homing, so that implies to compute the rotational and translational matrix in order to find the calibration matrix as:

$$R_x = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & \sin\theta \\ 0 & -\sin\theta & \cos\theta \end{pmatrix},\ R_y = \begin{pmatrix} \cos\phi & 0 & \sin\phi \\ 0 & 1 & 0 \\ -\sin\phi & 0 & \cos\phi \end{pmatrix}, R_z = \begin{pmatrix} \cos\psi & \sin\psi & \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{pmatrix}; \tag{6.1.6}$$

$R = R_x R_y R_z$ is the rotation matrix respectively along the X, Y and Z axis.

$$T = \begin{pmatrix} t_x \\ t_y \\ t_z \end{pmatrix};\ C = \begin{pmatrix} f\alpha_u & 0 & u_0 \\ 0 & f\alpha_v & v_0 \\ 0 & 0 & 1 \end{pmatrix}$$

are the translation matrix and the manual calibration matrix respectively, C is obtained from the SmarTrack software, where camera intrinsic parameters are: Focal length f, scale change pixel/mm (α_u, α_v), and coordinate of the impact point (u_0, v_0). Finally, we get the global calibration matrix by $P = C[R|T]$.

6.1.3 Simulator performance and technical tests

After the safety study, A brief description of simulator performance is described below:

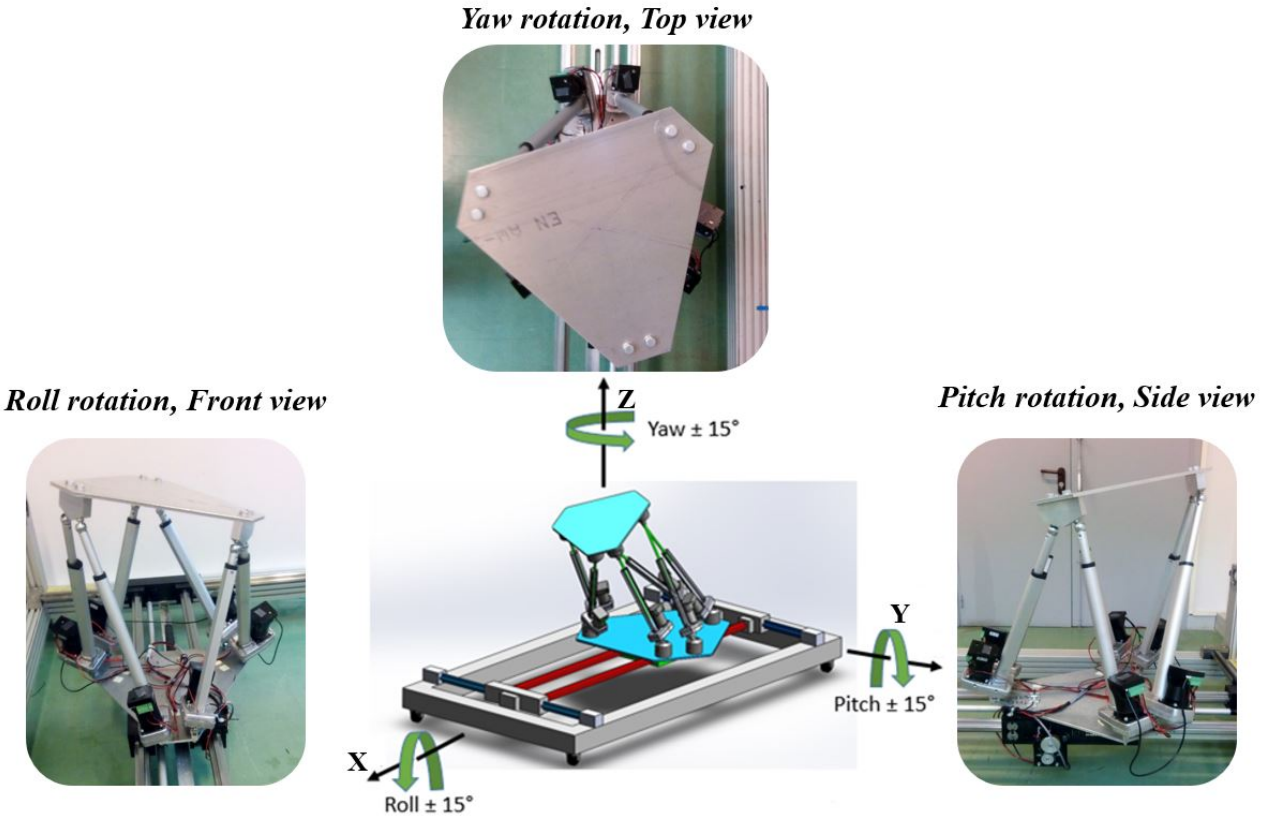


Figure 6.4: Allowed Rotation

D.O.F	Position	Max velocity	Max acceleration
X, Y	± 0.5 m	1.2 m/s	1.7 m/s^2
X_p, Y_p, Z_p	± 0.1 m	0.8 m/s	1.1 m/s^2
θ_x, θ_y	± 15 deg	48.7 deg/s	71.6 deg/s^2
θ_z	± 15 deg	106 deg/s	154 deg/s^2
Max load	50 Kg		

Table 6.1: XY-6 DoF motion range

The sensors information are collected with a sampling time of about 0.0196 s that corresponds to 51 Hz sampling frequency. This sampling frequency is more than enough for the response time of human sensation. In order to test the performance and test the behavior of the system some experiments were made, with different movement and different load.

Figure 6.5: Simulator with loadless

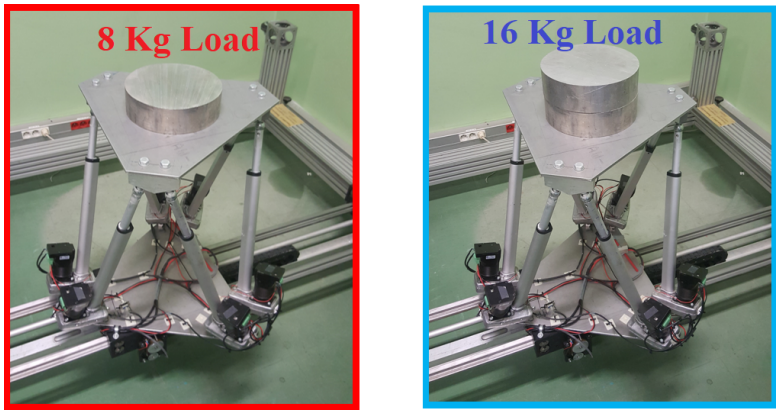


Figure 6.6: Simulator with one and two loads

To validate the mechatronic connection and the modeling implementation as well as to study the reactivity of the simulator, different translation and rotation trajectories have been applied on the platform. The rotations are roll, pitch and yaw rotation and the translation are about the three axis. Then motion trajectories in the form of large eight integrating the two sliders was applied to validate the operationality of the simulator. In this section, we present how the system behave with a simple movement in z direction with differents load. Two position templates have been applied for each experiment to see how our system behave with constant and variable speeds, the first position template is for constant speed. In reality this template consist of three phases, a very small acceleration phase, a constant speed (main) and a very small decelerations phases.

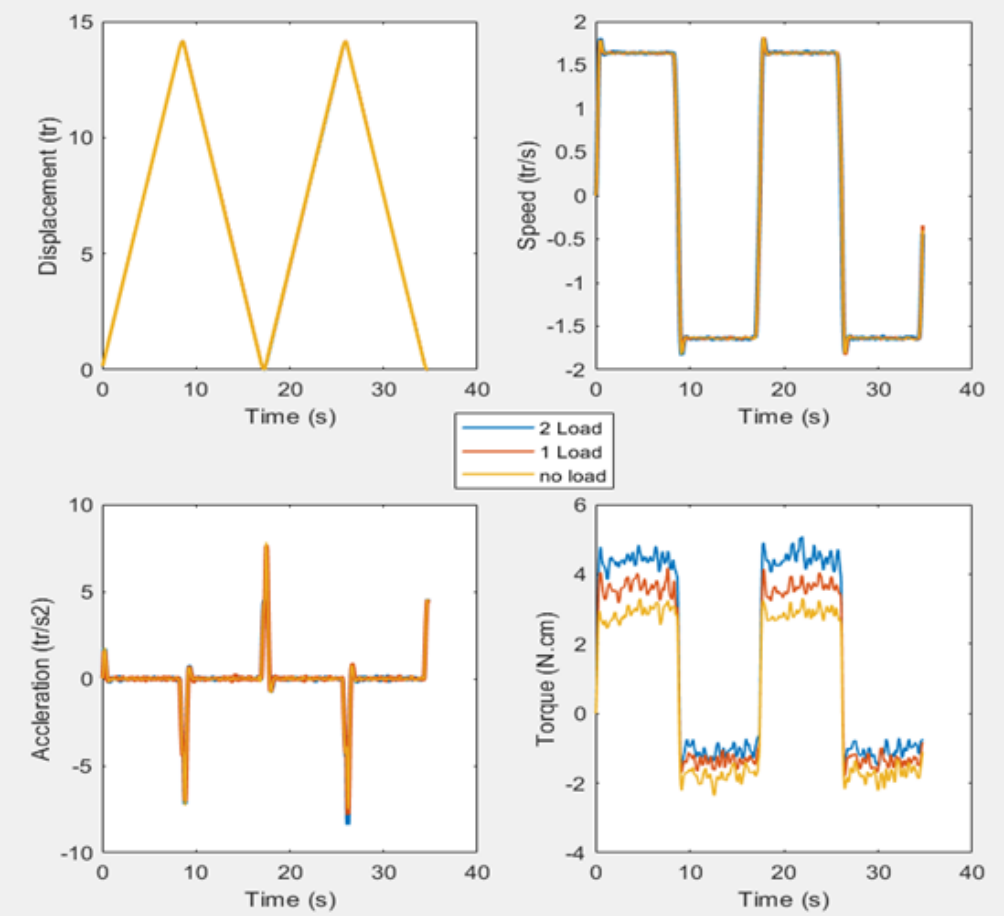


Figure 6.7: First template experiment

The second position template is to divide the motion in only acceleration and deceleration phases without any constant speed.

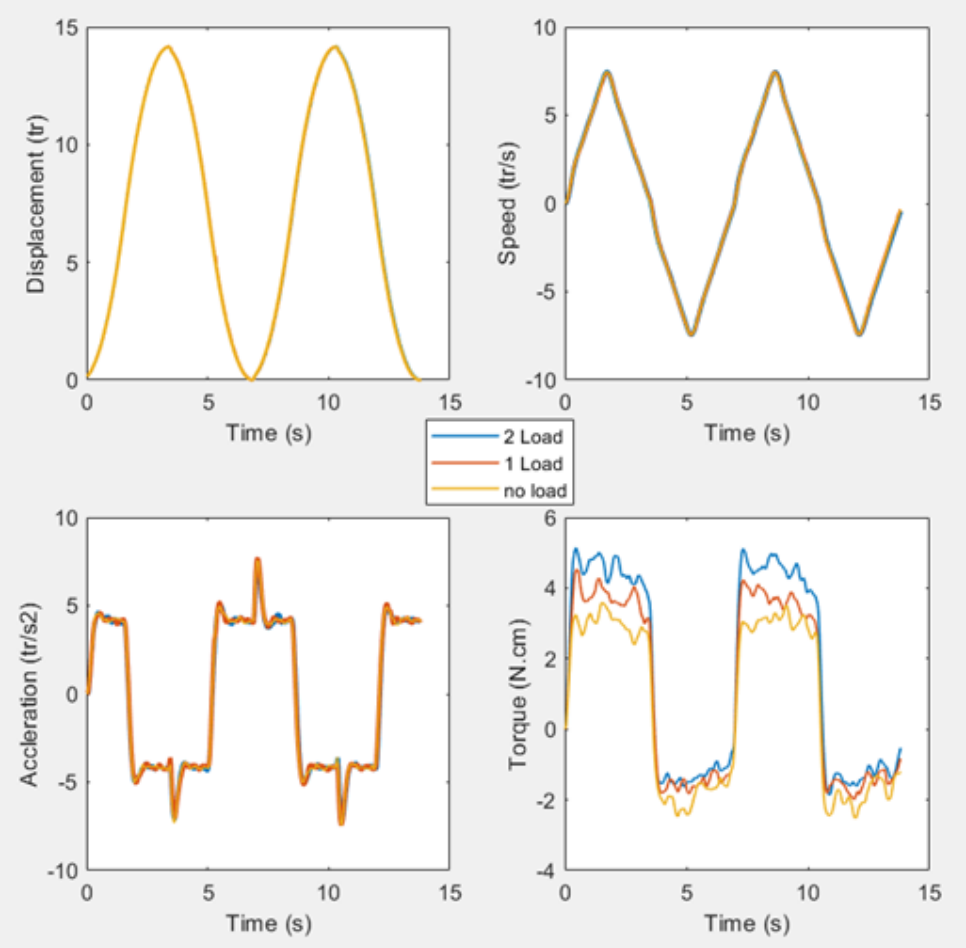


Figure 6.8: Second template experiment

The fact behind the use of these two templates is to well identify the inertial and friction parameters shown in Chapter 3. The first template excites the friction and mass parameters while the second excites the inertial parameters. In addition, the human sensation is within the ear at the vestibular system level that is sensitive to the acceleration, therefore it is essential to study the system behavior with and without accelerations.

These two templates have been applied where a PID controller is applied on both experiments. All of 6 motors of the Gough-Stewart platform show similar behavior, thus we present the sensed data only within the B_4P_4 motor. The load weighs is about $8\ kg$ and is placed on the upper platform center of mass. All velocities, acceleration and torques are filtered by a low pass filter in order to eliminate the harmful and undesired frequency (noise), As these figures show, all the sensed variables of the motors are well tracking the targeted positions, speeds and accelerations despite the addition of load.

The main idea of this section is to present that our study has made it possible to obtain a controllable platform in complete safety, with high DoF and high movement capacities. It has been designed as a system comprising software security ensuring the physical integrity of the structure and the platform preventing any impact when it is in moving state.

The implementation of the tracking module ensures an external feedback of the movement in view of the platform future control to track the trajectory defined in the virtual reality scenario.

6.2 Design of a virtual reality ski scenario under Unity

6.2.1 Introduction and objective statements

The XY-6 DoF robotic platform is intended to be used as motion simulator for disabled skiers [22]. Our project is to create an interactive simulator where the user immersion feeling must be increased thanks to the combination of virtual tools and of robot physical structure. The user has to experience a natural sensation of sliding movement when fully immersed in the mixed virtual reality environment.

Virtual reality (VR) is a simulated experience of the real world and it is used for entertainment (i.e. gaming) and educational purposes (i.e. medical or military training) Grigore, (2003) [163]. In addition, mixed reality (MR) is the merging of real and virtual worlds to produce new environments, where physical and digital objects co-exist and interact in real-time Jung, (2017) [164]. For both, we generate realistic images, sounds and other sensations that simulate a user's physical presence in a virtual environment.

The existing motion simulators use an oculus helmet or a screen with a 360° panoramic view of the external environment (mountain and snow for ski simulator) to show the virtual scenario Ikeda, (2018) [165], Nozawa, (2019a) [166], Oboe, (2014) [167]. The human immersion effect strongly depends on the resolution, realism and fluidity of the virtual scene. Many VR ski scenarios have been built but still need to be improved in terms of realsim and resolutions Wu, (2019) [168] and Hu, (2019) [169]. The main goal is to build the most realistic customized virtual scene in order to deceive the human brain and increase the immersive feeling.

In recent years, Handisports have evolved due to the increasing number of humans with poor or disabled mobility. Especially for the handicapped skier Bhambhani, (2012) [170] and for education purposes Caracas2019, (2019) [171] and Nozawa, (2019b) [172]. For our application, we have to adapt all virtual scenario components so that they are suitable for the disabled person. In order to design a virtual scenario under Unity software meeting the criteria of human disabled sport, it is advisable to move towards references of places and models of equipment adapted to the practice. From these references, it is possible to generate a landform as well as an avatar model.

Therefore, the objectives are mainly focus on building a realistic sliding movement scenarios using Unity software. The Unity features such as create the most realistic environement and the possibility to be coupled with OCULUS is used in order to increase the immersive effect of the human brain. Then special attention must be paid to the ergonomics and fluidity of movement, when planning trajectories in the virtual environment in order to optimize movement by taking into account the human sensory model. And finally carry out the configuration of the trajectory coordinates and manage data communication between the virtual environment and the XY-6 DoF simulator.

6.2.2 Handisport

Handisports Bhambhani, (2012) [170] and Caracas2019, (2019) [171] have evolved with the technology that accompanies them as well as the disabled or handicapped ski. Several ski types can be found such as Nordic skiing, which includes

cross-country skiing and biathlon, downhill skiing and snowboarding. For this project, the scenario revolves around alpine skiing for disabled people.

This category has specific regulations according to the type of disability of a person Jorand, (2013) [173]:

1. Sitting skiing can be practiced alone or with a guide for people with lower limbs disabilities.
2. Standing skiing alone, practiced by disabled person who can practice standing.
3. Standing skiing with a guide for the visually impaired

In addition, it needs to pay attention that not all stations are suitable for the practice of handiskiing. Indeed, infrastructures such as chairlifts and ski-lifts are necessary for the movement in the station of the handiskieur. In addition, trained personnel to meet the needs of the skiers are required to define a station as practicable for the handiski. The responsibility is shared between the ski station and the skier since he must check that his equipment is approved to have access to these stations.

To choose the appropriate real ski frame, several options are available at the hardware level. Indeed, alpine skiing has several types of chassis allowing different experiences Jorand, (2013) [173]:

1. Ski-kart:
   This is a pair of skis mounted on a chassis-frame. The trajectory is created thanks to the cranks on the sides of the device. This device is recommended for people who do not have great physical strength of the upper limbs or a lack of balance.

Figure 6.9: Ski-kart

2. Uni-ski:
   This is a variant of a chassis-frame with only one ski. The change of trajectory is made using stabilizers having a tip in the shape of a ski and the balance of weight of the user during turns.

Figure 6.10: Uni-ski

3. Dual-ski:
   This is a variant of a chassis-frame with two skis. It works in the same way as the uni-ski but allows to accentuate the sensations of slide for the skier.

Figure 6.11: Dual-ski

For our scenario, our choice fell on the dual-ski because it is the one that comes closest to the sensations provided for skier. The ski frame is designed to have the knee joint above the hip joint where the user has better cardiorespiratory responses Hofmann, (2016) [174].

Many VR ski scenarios have been built but still need to be improved in terms of realsim and resolutions Wu, (2019) [168] and Hu, (2019) [169]. Therefore, we are looking to virtualize a real mountain and real track. The track to be created under virtual reality must respect an important criterion which is to be a track where disabled skiers are allowed to practice. Another criterion is the readability of the track. It will make it easier to find the track on software like Google Maps so that it can be transcribed more easily in the scenario. This legibility can be created for example via the vegetation surrounding the track (mainly trees) or the boundaries of the latter (Example: a track between a ditch and a mountain side).

After having studied different tracks, the choice was made on a track of Haute-

Savoie in the station of Combloux. The track is called Tréffléannaise. It starts at Bonjournal (1853 meters above sea level) and at La Giettaz (1200 meters above sea level). It meets the desired criteria since it is reserved for handiskis and is surrounded by trees along its entire length.

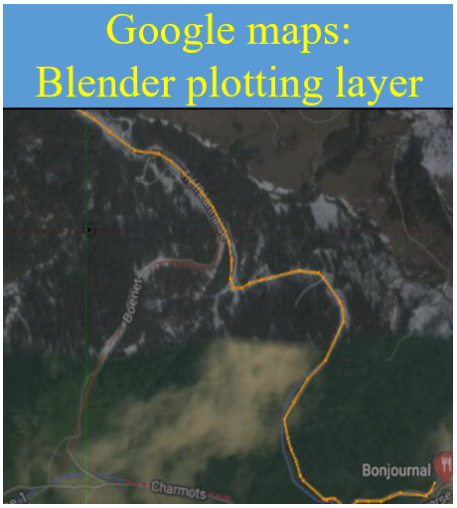


Figure 6.12: Tréffléanaise track definition (top view)

6.2.3 Unity ski scenario building

Unity is a software tool for designing a virtual environment in which we can define our scene, land, track, ski-frame and ski track. In addition, we can define new rules for interaction and trajectory planning [175].

A scene represents a virtual environment on which we can dispose of objects: our terrain (land), our ski model, the brightness, the camera. These objects are made up of components which will first define the position of the object in the scene and then optionally, the visual rendering, the collision box, the associated script, etc., depending on the intended role.

We start by defining our land, it represents the base, the ground, the reliefs and can be painted with textures. It is the foundation where all the elements will be placed on it. To achieve this, our method must be adapted to compliance with certain criteria. We made a choice based on a principle of topology: take the reliefs of a target area in the real world in order to transfer them to our virtual field.

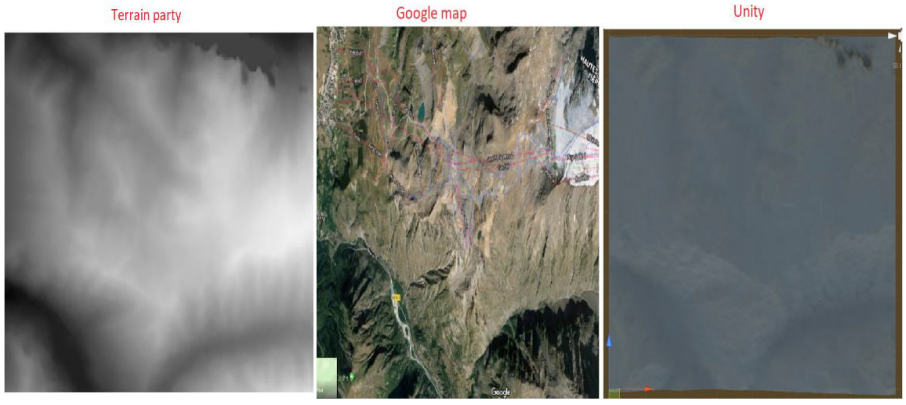


Figure 6.13: Comparison of landforms

It has the advantage of providing a realistic environment at the relief level. However it has its default, change the land in the future can harm its interest, it can be difficult to identify, and we must import 8 kilometers of perimeter at least. We get a result that we can compare with google map presented in Fig. 6.13.
The left part is a black and white image called HeightMap representing the level of relief by variation of the brightness, the white corresponding to a high level and the dark to a low level. A near view of the final unity environment is shown in Fig. 6.14.

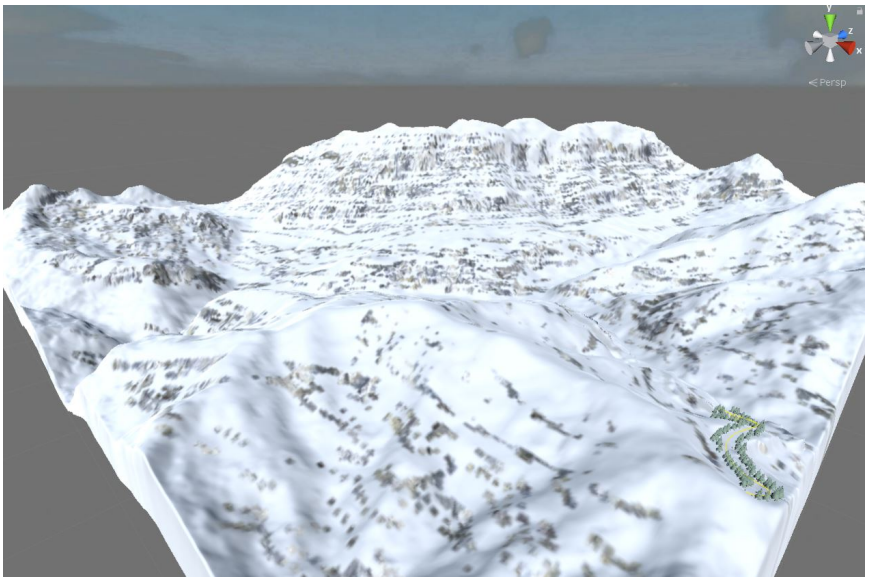


Figure 6.14: Near view

The next step will be to define the plot of our track. For this we took an image via Google Maps in top view. Once this image has been integrated into the Blender modeling software [176] as a layer, we can model a line-by-line route to the track which we can then import into Unity.

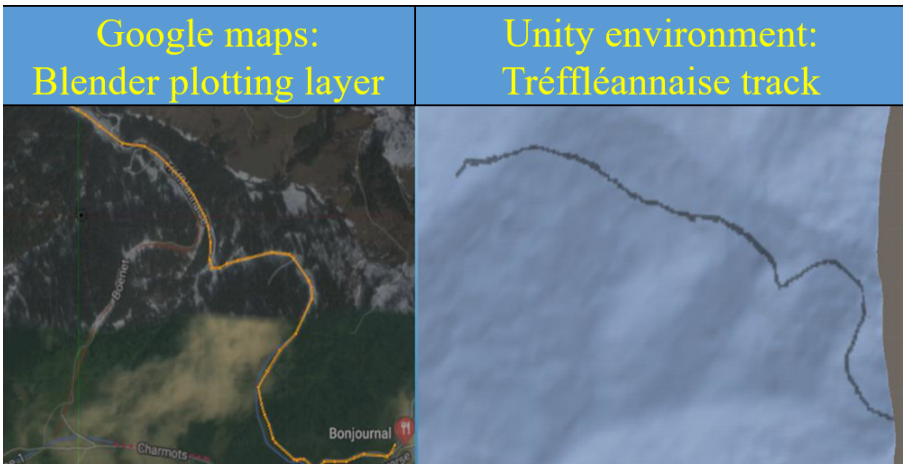


Figure 6.15: Track definition on Unity

The orange line on the left side of Fig. 6.15, represents the 3D modeling curve corresponding to the real track modeled on Blender.

The track is then modeled and imported to Unity software. Therefore, to move on our track, we need an avatar representing the user, equipped with skis, before being integrated with Unity we modeled our own model of ski seat via the Blender software [176] inspired by the model Dual-ski Loisir de Tessier [177].

The design methodology used consists of modeling the different parts by drawing two-dimensional patterns, then developing them on a third dimension of depth, and finally merging them into less complex sets of objects, these sets are delimited. Thus, create coherent mechanical connections between parts. Recall that the ski frame is designed to have the knee joint above the hip joint where the user has better cardiorespiratory responses Hofmann, (2016) [174].

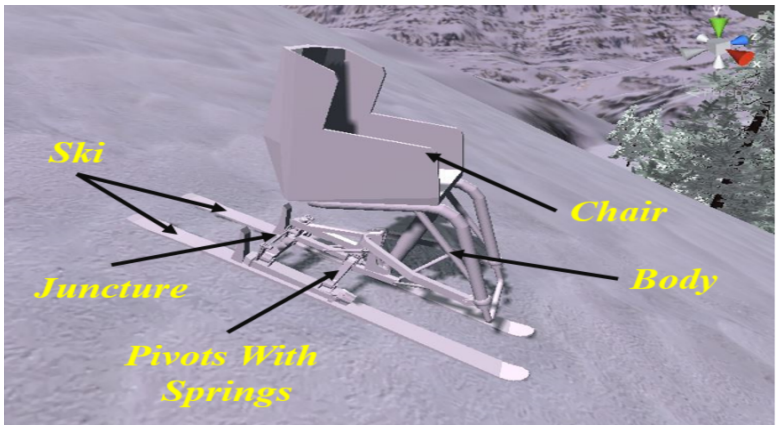


Figure 6.16: CAD ski frame model in UNITY environment

Represented above, named ModeleSkis, it is composed of 4 objects in the form of inheritance: Body, Joins, Pivot, and two skis (right and left) was added on the seat, a cylinder having the role of representing the avatar to which a camera is linked. Following its design, it was configured directly on Unity in order to have a collision with the ground and a sliding effect, pivot links for inclinations, and fixing links.

Once the mountain, the track and the ski frame have been designed and assembled, special attention must be paid to the visual rendering in order to increase the realism of our scenario.
Regarding the VR camera, Unity offers an "Interactive 360 Sample Project" resource in the Learn and Resources tab in the Unity project opening menu. Shaders participate in the visual rendering. They have a major role in the definition of the materials, the brightness that they send back and they can allow to obtain visual effects such as the projection of flake in the airs during the passage of the skis, or the tracks left in the snow by collision detection. A shader unlike a script will compute a graphics processor rather than the CPU. The following figure shows the designed shader.

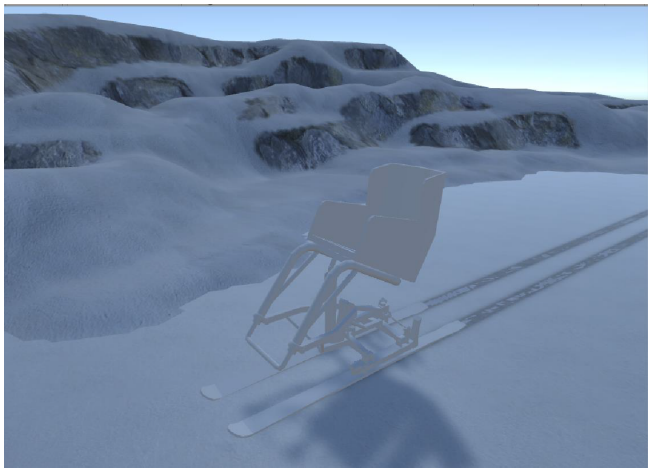


Figure 6.17: Experimenting a shader

In the end, all the entities are assembled together to form a completely new set of customized virtual scenes with high realism. The following figure shows the user view of the final environment.

Figure 6.18: User view of a disabled person in the virtual ski scene

Furthermore, we implemented the oculus rendering to increase visual sensation and it is presented in the following.

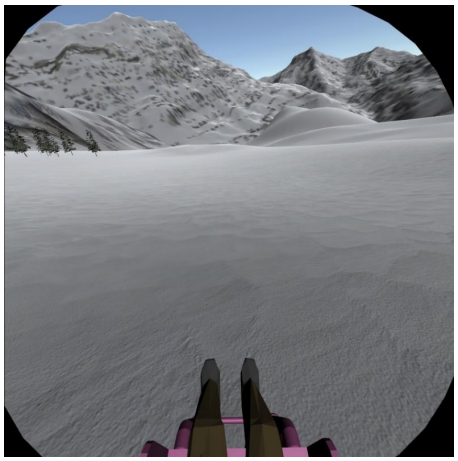


Figure 6.19: Oculus view

A basic motion planning problem is to compute a continuous path that connects a start configuration and a goal configuration while avoiding collision with known obstacles. Many trajectory planning can be used to define the track. We present the following three methods with their advantages and disadvantages.

1. **List of point trajectory:** The method of list of points is a choice on which we went in order to realize the programming of the trajectory tracking by the Dual-ski. It is a very basic method allowing the Dual-ski to navigate from point to point looking in their direction through a script. When the Dual-ski has reached a certain distance from a target point, it moves to the next one and then stops once at the last one.
   The disadvantage of this method is that it does not seem ideal because the ski is fixed in one direction. At the technical level we also encountered two major problems: jerks in the visual rendering of the camera and sudden movements of the latter, when passing from one target point to another.

   Artificial slowdown or acceleration phases can be implemented. For a more natural trajectory, another interesting method that could have been approached is to follow a spline or smooth trajectories.

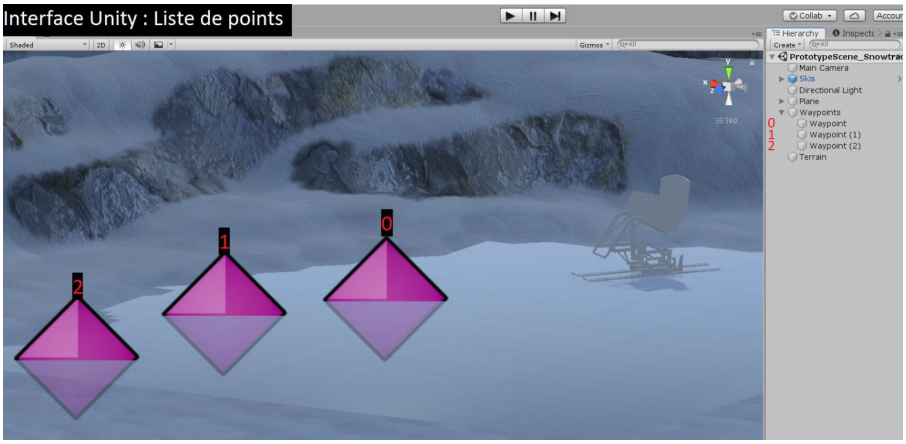


Figure 6.20: List of points and display

2. **Spline trajectory:** To avoid previous method weaknesses, a skiing trajectory was planned and defined on the virtualized mountain using Spline interpolation. We introduce a spline as a piecewise polynomial parametric curve. In general, splines can exist in other spaces as well, for this application however, we restrict them to live in a three-dimensional plane. There are three commonly used ways in the literature to define the polynomial segments of a spline, which are: the polynomial in parameter, the product of matrices, and the weighted sum of control vertice. We will use the last method because its implementation is adapted to your application Sprunk, (2008) [178]. The segments $S(t)$ can be expressed as a weighted sum of control vertices $p_i \in \mathbb{R}^3$ for cubic polynomials $(n = 3)$ there are four of them:

$$S(t) = \sum_{i=0}^{n} p_i B_i(t) \tag{6.2.1}$$

This notation allows the interpretation of the cubic polynomial as a linear combination of the control vertices $p_i \in \mathbb{R}^3$, weighted by the so called basis splines $B_i(t)$, which are cubic polynomials. A particular segment is here characterized by its control vertice whereas the basis splines can be interpreted as defining the family of segments it belongs to. The trajectory planned is shown in yellow in Fig. 6.25.

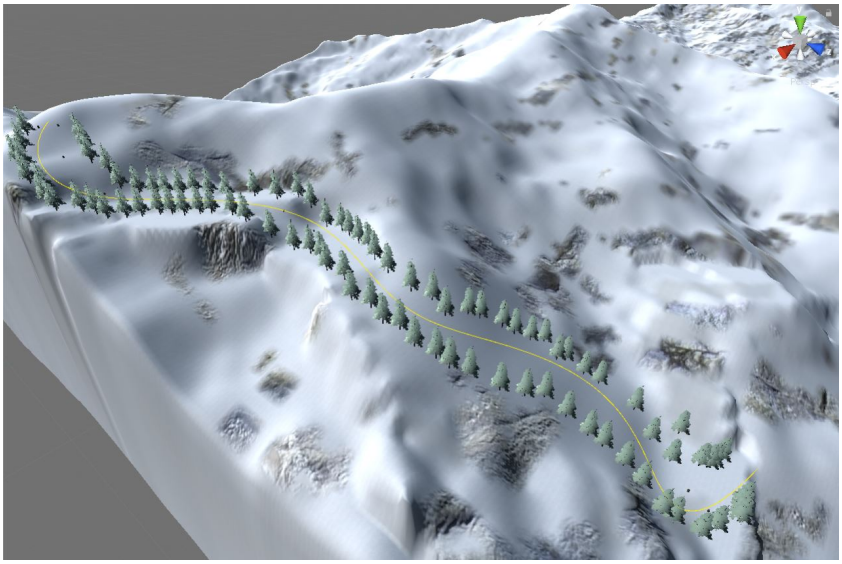


Figure 6.21: Planned trajectory of Tréffléannaise track

All physics of downhill skiing describing forces acting on a skier such as downhill force, parallel downhill force, lateral force, centrifugal force, and kinematic friction force are not taken into account.

3. **Trajectory planning based on the action of gravity:** The first and second solutions were to create crossing points or curve and thus the skier will follow the trajectory. Once the trajectory was created, the skier had a constant speed and a defined degree of leaning when cornering. We realized that the rendering was not fluid enough and a more appropriate method should be developed.

   The new method consists to create physics for the skier, that is to make the skier react according to the speed (taking into account the downhill skiing forces). The skier moves thanks to gravity which is simulated on the Unity software, then we set up speed resistance due to the skis friction on the track but also greater lateral braking during cornering. The skier, when he has a lateral movement speed exceeding a certain threshold, he will brake by a force representative of the friction of the skis in opposition to the track and this deceleration will lead to a gain in frontal speed. This method shows a more realistic / fluid trajectory planning and has no tracking disturbance.

It should be noted that all trajectory outputs are optimized to be within the limits of the platform capacity, taking into account the MCA and the physical limitation of the platform. This optimization is done by trial and error method and by limiting the tilt, speed and acceleration of the skier.

6.2.4 Virtual environment coupling with the XY-6 DoF simulator

The platform must be able to follow a predefined trajectory on the Unity environment. In order to verify that the constraints of the simulator are respected, it is necessary to have data specific to the planned trajectory. This recovery of ground data and model movement is done from the script-related $C-\#$ scripts. The unity skier accelerations and rotation speeds are presented in the following.

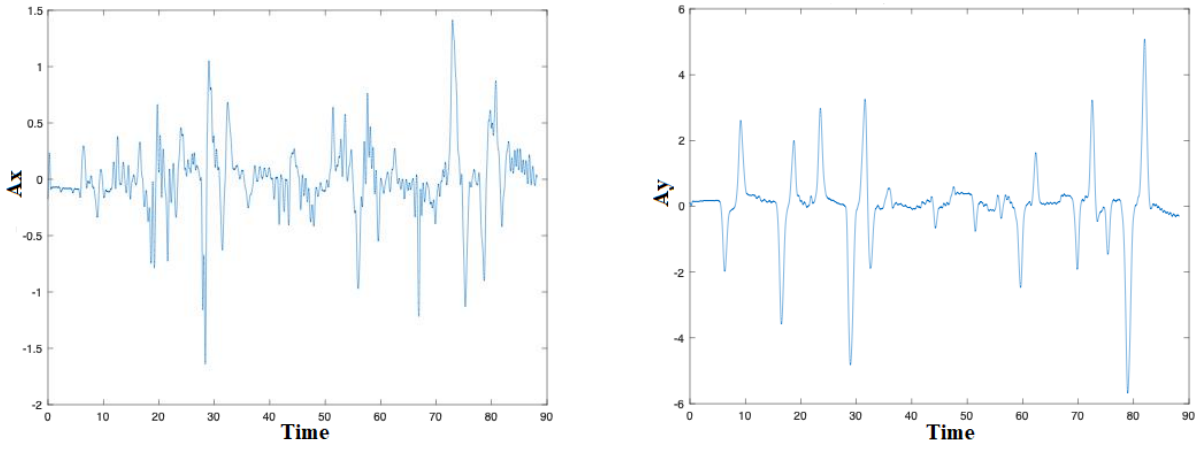


Figure 6.22: Skier accelerations

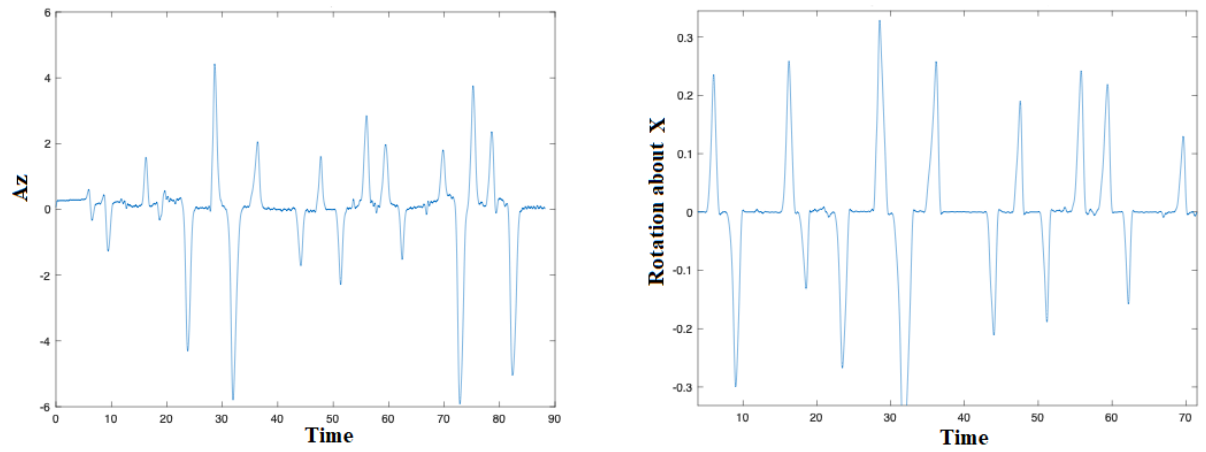


Figure 6.23: Skier accelerations and rotation speed

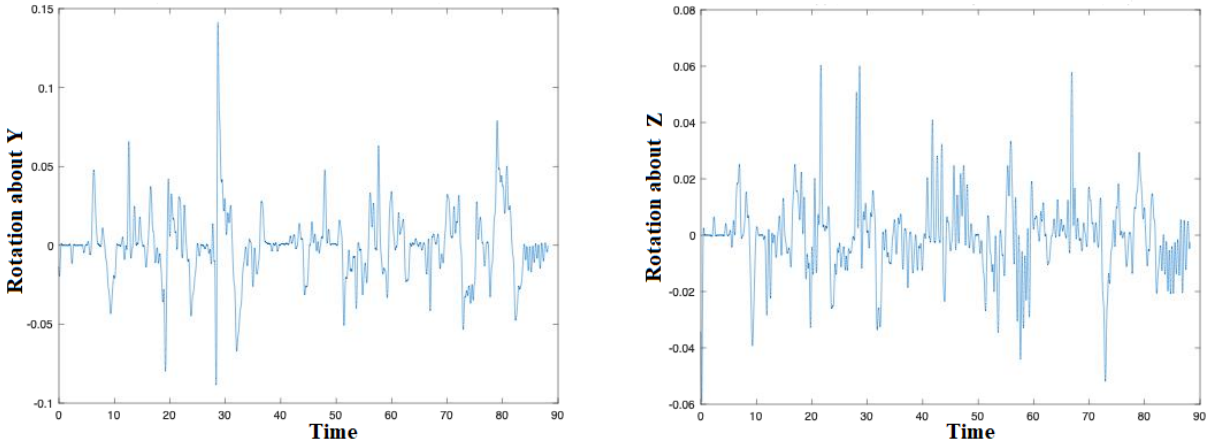


Figure 6.24: Skier rotation speed

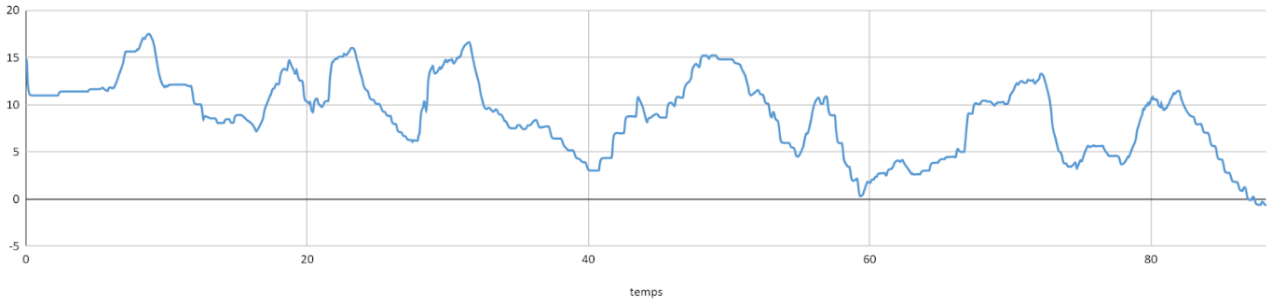


Figure 6.25: Degree of rotation of the skier about z axis

These figures show that the trajectory information is within the platform capacity. For example, the maximum rotation that can be performed by the platform is around 20 degrees and the last figure shows that the skier has no angle greater than 20 degrees.

After the realization and the finalization of the virtual scenarios, the next work will be necessary the sharing of trajectory information with the robotic structure. The main task is to recover the data provided by Unity. Indeed, the definition of a trajectory on Unity corresponds to a set of points defined according to their coordinates in width, length and height $(x,\ y,\ z)$. Matlab must be able to read these

coordinates to transmit them to the platform and make the appropriate movement.
The best method used in order to recover data from Unity is the Implementation of a TCP / IP communication between the Unity project and Matlab. This method allows Matlab and Unity to connect easily in real-time by sharing specific addresses and ports using the **TCP/IP protocol** using dedicated functions.
Unity's data has been tested in practice and gives consistent results. The platform performs the movement observed on the software with little delay.

6.3 Conclusion

The main idea of this section is to present the robotic platform operability. The security environment has been defined to prevent structural damage and be safe for humans. We show that external sensors can be added to increase the robotic platform safety and can also be used as important information for the trajectory tracking controller.

For virtual reality scenarios construction. We have come a long way in modeling the ground using methods that allow us to get closer to the reality of the chosen location using google maps. This method allows a greater immersion of the user by increasing the scenario realism. The creation of a skis model meeting the project criteria which are "Realism, Fluidity, Integration of the Oculus and Optimization" was respected. Our virtualized environment generates a scene that is more realistic compared to literature.

The virtual scene is customized to meet our specifications. The virtual scene has also been optimized to respect the physical limitation of the robotic platform. It should be noted that the material resources are used optimally, the scenario is refined. In addition, the scenario is optimized for the smooth running of the simulation [HOUDA3].

The scenario/platform connection was successfully completed, we were able to recover all information from Unity using TCP/IP connection. The installation of the NI card to greatly increase the communication speed with the motors has been successfully installed.

Chapter 7

Conclusions and Perspectives

The thesis subject concerns the interaction study of a large workspace parallel robot platform with a virtual environment. In other words, realization and characterization of a motorized Gough-Stewart platform mounted on two sliders axis coupled with virtual enviroment. The thesis objective is to improve the driver movement sensations rendering on a skiing simulator. This movement illusion is built, in the driver, from the fusion of information (or sensory excitations) captured by his perceptual system.

This thesis is organized around the following parts. First of all, a bibliographic study is carried out to identify the motion simulation problem in general without focusing on skiing or sliding application. In this part, we learned about the various mechanical architectures previously used as well as the problems encountered. Then, the choice of the architecture of the simulator is justified.

Second, we have supported (chapter 3) our study about the simulation platform by introducing important specificities to parallel robotics: direct and inverse geometric, kinematic, dynamic models and then validated within SimMechanics framework, dynamic parameters optimization and identification, singularities, workspace, Note that the construction of dynamic models in the context of skiing simulation is original work.

Third, the human perceptual systems are, in a real ski frame or in a simulator, visual information (visual scene movement), proprioceptive (relative positions of the body's organs), tactile (detection of vibrations), vestibular (detection of head movement in space), etc. Among these elements, we have deepened the study of the vestibular contribution in order to improve it within the simulation framework. In other words, we proposed to generate with the movement mechanism cockpit simulation trajectories whose perception by the driver is close to that of a real skier.

Fourth while the dynamic model of the simulation robot represents one of this thesis contributions, the vestibular model is a linear model chosen from the models proposed in the literature. The validity of the second model is essential in order to guarantee the coherence and the good foundation of our approach. In this perspective, we have provided (chapter 4) an anatomical description of the vestibular system and a synthesis of its modeling. The classic method of controlling driving simulators is based on a two-stage structure. The first one, Motion Cueing Algorithm (MCA) that makes it possible to project the trajectory of a real skier into a trajectory that can be produced by the simulation robot while preserving the sensations quality (chapter 4). Four MCA approaches have been modified to benefit from our robot redundancy to increase human sensation. the effec-

tiveness of each approach (human feeling) is measured from the performance indicator (chapter 4). The second one, a motion tracking algorithm that makes it possible to track the desired trajectory in the real robot simulator. Our robot has a redundancy capacity, we have developped a method to exploit this redundancy in order to generate the desired trajectory for each subsystem and to increase the human sensation (chapter 5).

Fifth, a very realistic custom virtual reality was built and modeled within the Unity framework adapted to disabled person. By building a real mountain, modeling a real ski frame, and using an oculus vision rendering. Information between the virtual scenario and the real simulator is shared via a TCP/IP connection. We are very confident about the behavior of the proposed algorithms to make human interactions more immersive.

Finally, the first tests were carried out, on the one hand, in an open-loop for the characterization and validation of the entire simulation chain, and on the other hand, in a closed-loop, to assess the proposed architecture validity, in terms of dynamic performance, for the skiing simulation of a hybrid robot. The limitations of the simulation platform are discussed, as well as the corresponding optimizations are proposed in order to improve the present simulator.

The perspectives of this thesis are numerous, namely:

1. The addition of new movement stages, for example longitudinal and transverse, on the platform.

2. The algorithms development for skiing assistance and the implementation of warning systems (loss of grip, excessive speed, cornering, etc.). Cold air devices and haptic force feedback can be added to increase human immersion.

3. The whole structure dynamic model enhancement by adding disabled person models and study the effect of human movement on robot control.

4. Study the impact of kinesthetic feedback on the NAO robot, undergoing the induced movements of the parallel structure. The induced movements are generated by a virtual scenario animated in mixed reality. The NAO robot serving as an immersive and interactive feedback simulator, in order to validate the programs implemented initially.

5. The optimized algorithms developed in the previous phase will be applied and evaluated with the help of human users, for a more in-depth study of the physiological behavior in immersion, within augmented virtual environments. Human factors and ergonomic challenges must be carefully considered.

Perspective: This thesis research work will be continued in studying the interactions between XY-6 DoF simulator and humanoid robotic structure during the virtual scenario following. And by analyzing the impact of the kinesthetic feedback caused on the NAO robot. The NAO robot will serve as a “substitute” for a disabled person before putting the real disabled person on the simulator. This perspective research will be carried out as part of a Ph.D. starting in September 2020.

Personal Bibliography

[HOUDA1] Taha Houda, Ali Amouri, Lotfi Beji, Malik Mallem. Dynamic parameters optimization of a Gough-Stewart Platform mounted on a 2-DOF moving base. 9th ECCOMAS Thematic Conference on Multi-body Dynamics, Jul 2019, Duisburg, Germany. (hal-02299878).
[HOUDA2] Houda T., Amouri A., Beji L., Mallem M. (2020) Dynamic Parameters Optimization and Identification of a Parallel Robot. In: Kecskeméthy A., Geu Flores F. (eds) Multibody Dynamics 2019. ECCOMAS 2019. Computational Methods in Applied Sciences, vol 53. Springer, Cham.
[HOUDA3] Taha Houda, Lotfi Beji, Ali Amouri, Malik Mallem. Dynamic Behavior of an Interactive XY -6 DoF Simulator for People with Reduced Mobility. 4th IEEE Conference on Control Technology and Applications (CCTA 2020), Aug 2020, Montreal, Canada. (hal-02879923).

Bibliography

[1] G. Reymond, A. Kemeny, A. Heidet, et M. Canry. Validation of renault's dynamic simulator for adaptive cruise control experiments. Dans Driving Simulation Conference DSC Europe 2000, pages 181–191, Paris-France, 2000.

[2] P. Grant, B. Artz, J. Greenberg, et L. Cathey. Motion characteristics of the virttex motion system. Dans Proceedings of the 1st Human-Centered Transportation Simulation Conference (HCTSC'01), Iowa, USA, 2001.

[3] G. Reymond and A. Kemeny, Motion cueing in the Renault driving simulator, Vehicle System Dynamics, vol. 34, pp. 249-259, 2000.

[4] N. Giesen, W. Zanker and F. Bracke, The new driving simulator, Fast response and a photorealistic environment, Press Information, Daimler Communications, 70546 Stuttgart/Germany, Mercedes-Benz - A Daimler Brand, 2011.

[5] G. J. Heydinger, M. K. Salaani, W. R. Garrott and P. A. Grygier, Vehicle dynamics modelling for the National Advanced Driving Simulator, Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering, 216(4), pp. 307,318, 2002.

[6] TOYOTA, Driving Simulator - Safety Technology Innovation from a Driver's point of view, retrieved: 06th February 2019. [Online]. Available: http://www.toyota.com.cn/innovation/safety technology/safety measurements/driving simulator.html.

[7] T. Miunske, C. Holzapfel, E. Baumgartner and H. Reuss, "A new Approach for an Adaptive Linear Quadratic Regulated Motion Cueing Algorithm for an 8 DoF Full Motion Driving Simulator," 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 2019, pp. 497-503. doi: 10.1109/ICRA.2019.8794109.

[8] World Health Organization paper in Apr 04, 2019. Available at: https://www.worldbank.org/en/topic/disability.

[9] Ku, Jeonghun , Jang, Dongpyo , Ahn, Heebum , Lee, Jaemin , Kim, Jeong , Lee, Bumseok, Kim, In , Kim, Sun. (2002). The development and clinical trial of a Driving Simulator for the handicapped. Studies in health technology and informatics.

[10] R.W. Allen, T. Rosenthal, B. Aponso, A. Harmsen, et S. Markham. Low cost pc-based techniques for simulation implementation. Driving Simulation Conference DSC Europe 1999, 1999.

[11] M. Mon-Williams, J.P. Wann, et S.K. Rushton. Binocular vision in a virtual world : Visual deficits following the wearing of a head-mounted-display. Ophtalmic and Physiological Optics, 13 :387–391, 1993.

[12] https://www.trusimulation.com/aviation-simulation-services/flight-simulation-training-devices/fixed-base-training-devices.

[13] Klüver, Malte & Herrigel, Carolin & Preuß, Stephanie & Schoener, Hans-Peter & Hecht, Heiko. (2015). Comparing the Incidence of Simulator Sickness in Five Different Driving Simulators.

[14] https://www.acrome.net/delta-robot

[15] https://gallery.autodesk.com/fusion360/projects/124906/gough-stewart-platform

[16] M. Dagdelen. Restitution des stimuli inertiels en simulation de conduite. PhD thesis, Ecole des mines de Paris, 2005.

[17] B. Artz, L. Cathey, P. Grant, D. Houston, et J. Greenberg. The design and construction of the visual subsytem for virttex the driving simulator at the ford research laboratories. Dans Proceedings of Driving Simulation Conference (DSC'01), pages 255–262, Sophia Antipolis, France, 2001.

[18] P. Miermeister et al., "The CableRobot simulator large scale motion platform based on cable robot technology," 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Daejeon, 2016, pp. 3024-3029.doi: 10.1109/IROS.2016.7759468

[19] L.D. Chen, Y. Papelis, G. Waston, et D. Solis. NADS at the University of Iowa : A tool for driving safety research. Dans Proceedings of 1st Human Centered Transprotation Simulation Conference, Iowa, USA, 2001.

[20] W. Käding et F. Hoffmeyer. The advanced daimler-benz driving simulation. SAE Technical Paper Series, (950175), 1995a.

[21] M. Dagdelen, G. Reymond, A. Kemeny, M. Bordier, et N. Maïza. MPC based motion cueing algorithm : Development and application to the ULTIMATE driving simulator. Dans Driving Simulation Conference DSC Europe 2004, pages 221–233, Paris-France, 2004.

[22] Ali Amouri, Fakhr-Eddine Ababsa. Sliding Movement Platform for Mixed Reality Application. 7th IFAC Symposium on Mechatronic Systems, Sep 2016, Loughborough, United Kingdom. 49 (21), pp.662–667, 2016, IFAC-PapersOnLine. <10.1016/j.ifacol.2016.10.676>. <hal-01367402>

[23] R.F. Boian, M. Bouzit, G.C. Burdea, , J. Lewis, et J.E. Deutsch. Dual stewart platform mobility simulator. Dans Proceedings of the 9th IEEE International Conference on Rehabilitation Robotics, pages 550–555, 2005.

[24] R.F. Boian, M. Bouzit, G.C. Burdea, et J.E. Deutsch. Dual stewart platform mobility simulator. Dans Proceedings of the 26th IEEE International Conference of Engineering in Medicine and Biology Society (EMBS'04), pages 4848–4851, 2004.

[25] H. Niniss et T. Inoue. Electric wheelchair simulator for rehabilitation of persons with motor disability. Dans 8th Symposium on Virtual Reality (SVR'06), Belem,Brazil, 2006.

[26] https://www.dreamstime.com/stock-video-footage-person-going-therapy-lokomat-device-training-making-exercises-virtual-reality-environment-video93629303.

[27] https://www.digitaltrends.com/cool-tech/genworth-r70i-exoskeleton/.

[28] Denavit, Jacques; Hartenberg, Richard Scheunemann (1955). "A kinematic notation for lower-pair mechanisms based on matrices". Trans ASME J. Appl. Mech. 23: 215–221.

[29] Hartenberg, Richard Scheunemann; Denavit, Jacques (1965). Kinematic synthesis of linkages. McGraw-Hill series in mechanical engineering. New York: McGraw-Hill. p. 435. Archived from the original on 2013-09-28. Retrieved 2012-01-13.

[30] Paul, Richard (1981). Robot manipulators: mathematics, programming, and control : the computer control of robot manipulators. Cambridge, MA: MIT Press. ISBN 978-0-262-16082-7. Archived from the original on 2017-02-15. Retrieved 2016-09-22.

[31] Khalil, Wisama & Kleinfinger, J.F.. (1986). A new geometric notation for open and closed-loop robots

[32] Khalil W., Dombre E., "Modeling, identification and control of robots", Hermès Penton, London, 2002.
[33] Wisama Khalil, Sylvain Guegan. Inverse and Direct Dynamic Modeling of Gough-Stewart Robots. IEEE Transactions on Robotics and Automation, Institute of Electrical and Electronics Engineers (IEEE), 2004, 20 (4), pp.754-762. <hal-00401751>
[34] D. E . Whitney, “Resolved motion rate control of manipulators and human prostheses,” IEEE Trans. Man, Machine Syst., vol. MMS-10, pp. 47–53, 1969
[35] Harib, Khalifa & Srinivasan, Krishnaswamy. (2003). Kinematic and dynamic analysis of Stewart platform-based machine tool structures. Robotica.
[36] António M. Lopes, Dynamic modeling of a Stewart platform using the generalized momentum approach, Communications in Nonlinear Science and Numerical Simulation, Volume 14, Issue 8, 2009, Pages 3389-3401, ISSN 1007-5704, https://doi.org/10.1016/j.cnsns.2009.01.001.
[37] Kalani, H., Rezaei, A. & Akbarzadeh, A. Improved general solution for the dynamic modeling of Gough–Stewart platform based on principle of virtual work. Nonlinear Dyn 83, 2393–2418 (2016). https://doi.org/10.1007/s11071-015-2489-z
[38] Wisama Khalil, Ouarda Ibrahim. General Solution for the Dynamic Modeling of Parallel Robots. Journal of Intelligent and Robotic Systems, Springer Verlag (Germany), 2007
[39] Wisama Khalil. DYNAMIC MODELING OF ROBOTS USING RECURSIVE NEWTON-EULER TECHNIQUES. ICINCO2010, Jun 2010, Portugal.
[40] Wisama Khalil. "Dynamic Modeling Of Robots Using Recursive Newton-Euler Techniques". ICINCO2010, Jun 2010, Portugal.
[41] R.A. Freeman, D. Tesar, Dynamic modeling of serial and parallel mechanisms/robotic systems, in: Proceedings of the ASME Bienniial Mechanism Conference, Kissimmee, FL, DE-vol. 15(2), 1988, pp. 7–21.
[42] M. Sklar, D. Tesar, Dynamic analysis of hybrid serial manipulator systems containing parallel modules, ASME Journal of Mechanisms Transmissions and Automations 110 (2) (1988) 109–115.
[43] G.B. Chung, B-J. Yi, D.J. Lim, W. Kim, An efficient dynamic modeling methodology for general type of hybrid robotic systems, in: IEEE International Conference on Robotics and Automation, vol. 2, New Orleons 2004, pp. 1795–1802.
[44] Ouarda Ibrahim, Wisama Khalil, Inverse and direct dynamic models of hybrid robots, Mechanism and Machine Theory, Volume 45, Issue 4, 2010.
[45] Briot, Sébastien & Khalil, Wisama. (2013). Recursive Symbolic Calculation of the Dynamic Model of Flexible Parallel Robots. Proceedings - IEEE International Conference on Robotics and Automation.
[46] Philip Long, Wisama Khalil, Philippe Martinet, Dynamic modeling of parallel robots with flexible platforms, Mechanism and Machine Theory, Volume 81, 2014, Pages 21-35, ISSN 0094-114X, https://doi.org/10.1016/j.mechmachtheory.2014.06.009.
[47] W. Khalil, G. Gallot and F. Boyer, "Dynamic Modeling and Simulation of a 3-D Serial Eel-Like Robot," in IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), vol. 37, no. 6, pp. 1259-1268, Nov. 2007, doi: 10.1109/TSMCC.2007.905831.
[48] Merlet J.-P., Parallel robots, Dordrecht, The Netherland, Kluwer, 2000.
[49] Mata, Vicente , Farhat, Nidal , Diaz-Rodriguez, Miguel , Valera, Angel, Page, Alvaro. (2008). Dynamic Parameter Identification for Parallel Manipulators.
[50] C. Atkeson, C. An, and J. Hollerbach. Estimation of inertial parameters of manipulator loads and links. The International Journal of Robotics Research, 1986.

[51] P. Khosla. Estimation of robot dynamics parameters: Theory and application. Institute for Software Research, 1986.

[52] Z. Xuecai and Z. Qixian. An identification method for estimating the combinative dynamic parameters of robots. In IEEE International Conference on Industrial Technology - ICIT, pages 808–812, Guangzhou, China, 1994.

[53] M. Gautier, P.O. Vandanjon, and C. Presse. Identification of inertial and drive gain parameters of robots. In Proc. 33rd Conference on Decision and Control, pages 3764– 3769, Lake Buena Vista, FL, USA, dec 1994.

[54] M. Gautier, A. Janot, and P.O Vandanjon. Didim: A new method for the dynamic identification of robots from only torque data. In IEEE International Conference on Robotics and Automation, pages 2122–2127, Pasadena, CA, USA, 2008.

[55] W. Khalil, M. Gautier, and P. Lemoine. Identification of the payload inertial parameters of industrial manipulators. In IEEE International Conference on Robotics and Automation, pages 4943–4948, Rome, Italy, apr 2007.

[56] B. Armstrong. On finding 'exciting' trajectories for identification experiments involving systems with non-linear dynamics. In Robotics and Automation. Proceedings. 1987 IEEE International Conference, volume 4, pages 1131 – 1139. IEEE, mar 1987.

[57] M. Gautier and W. Khalil. Exciting trajectories for the identification of base inertial parameters of robots. In Proc. 30th Conference on Decision and Control, pages 494– 499, Brighton, England, dec 1991.

[58] R. Lucyshyn and J. Angeles. A symbolic approach to determining exciting trajectories for identification of manipulator dynamic models. In Industrial Electronics, Control, and Instrumentation, 1993. Proceedings of the IECON '93., volume 3, pages 1684 – 1689, Maui, HI, USA, nov 1993. IEEE.

[59] P. Vandanjon, M. Gautier, and Desbats P. Identification of robots inertial parameters by means of spectrum analysis. In IEEE International Conference on Robotics and Automation, 1995.

[60] Yao, Di, Büttner, Kay, and Prokop, Günther. "Identification of Vehicle Inertia Parameters: From Test Bench Design to Movement Trajectory Optimization." Proceedings of the ASME 2018 International Mechanical Engineering Congress and Exposition. Volume 4B: Dynamics, Vibration, and Control. Pittsburgh, Pennsylvania, USA. November 9–15, 2018. V04BT06A014. ASME. https://doi.org/10.1115/IMECE2018-87545.

[61] Mohammed Aquil Mirza, Shuai Li, Long Jin, Simultaneous learning and control of parallel Stewart platforms with unknown parameters, Neurocomputing, Volume 266, 2017, Pages 114-122, ISSN 0925-2312, https://doi.org/10.1016/j.neucom.2017.05.026.

[62] Gautier M., "Identification of robots dynamics", Proc. IFAC Symp. on Theory of Robots, Vienna, Austria, December 1986, p. 351-356.

[63] Raucent B. "Identification des parametres dynamiques des robots manipulateurs", These de Doctorat, University de Louvain, Belgium. 1990.

[64] M. Gautier and W. Khalil, « Chapter 4 : Identification des paramètres des modèles », of book « Analyse et Modélisation des Robots Manipulateurs », under the direction of E. Dombre, Hermes, 2001.

[65] Fichter, E. F., and McDowell, E. D., 1980, "A Novel Design for a Robot Arm,"Computer Technology Conference, New York, Aug. 12–15, pp. 250–256.

[66] Laryushkin, P. A., and Glazunov, V. A., 2015, "On the Estimation of Closeness to Singularity for Parallel Mechanisms Using Generalized Velocities and Reactions," The 14th IFToMM World Congress, Taipei, Taiwan, Oct. 25–30, pp. 286–291.

[67] Liu, X.-J., Wu, C., and Wang, J., 2012, “A New Approach for Singularity Analysis and Closeness Measurement to Singularities of Parallel Manipulators,” ASME J. Mech. Rob., 4(4), p. 041001.

[68] Bandyopadhyay, S., and Ghosal, A., 2006, “Geometric Characterization and Parametric Representation of the Singularity Manifold of a 6–6 Stewart Platform Manipulator,” Mech. Mach. Theory, 41(11), pp. 1377–1400.

[69] Damien Six, Sebastien Briot, Abdelhamid Chriette, Philippe Martinet. A Controller for Avoiding Dynamic Model Degeneracy of Parallel Robots during Type 2 Singularity Crossing. 6th European Conference on Mechanism Science (EuCoMeS2016), Sep 2016, Nantes, France.

[70] Aleshin, A. K., Glazunov, V. A., Rashoyan, G. V., and Shai, O., 2016, “Analysis of Kinematic Screws That Determine the Topology of Singular Zones of Parallel-Structure Robots,” J. Mach. Manuf. Reliab., 45(4), pp. 291–296.

[71] Qimi Jiang, Clément M. Gosselin, Determination of the maximal singularity-free orientation workspace for the Gough–Stewart platform, Mechanism and Machine Theory, Volume 44, Issue 6, 2009, Pages 1281-1293, ISSN 0094-114X, https://doi.org/10.1016/j.mechmachtheory.2008.07.005.

[72] Kotlarski, J., Heimann, B. Ortmaier, T. Influence of kinematic redundancy on the singularity-free workspace of parallel kinematic machines. Front. Mech. Eng. 7, 120–134 (2012). https://doi.org/10.1007/s11465-012-0321-8

[73] Hunt, K. H., 1978, Kinematic Geometry of Mechanisms, Oxford University Press, Cambridge.

[74] Fichter, E. F., 1986, “A Stewart Platform-Based Manipulator: General Theory and Practical Construction,” Int. J. Robot. Res., 5(2), pp. 157–182.

[75] Gosselin, C. M., and Angeles, J., 1990, “Singularity Analysis of Closed-Loop Kinematic Chains,” IEEE Trans. Rob. Autom., 6(3), pp. 281–290.

[76] Tsai, L.-W., 1999, Robot Analysis, Wiley, New York.

[77] Zlatanov, D., Fenton, R. G., and Benhabib, B., 1994, “Singularity Analysis of Mechanisms and Robots via a Velocity-Equation Model of the Instantaneous Kinematics,” in Proceedings of the IEEE International Conference on Robotics and Automation, San Diego, CA, USA, pp. 986–991.

[78] Zlatanov, D., Bonev, I. A., and Gosselin, C. M., 2002, “Constraint Singularities of Parallel Mechanisms,” in Proceedings of the 2002 IEEE International Conference on Robotics and Automation, Washington, D.C., pp. 496–502.

[79] Ma, O., and Angeles, J., 1992, “Architecture Singularities of Parallel Manipulators,” Int. J. Rob. Autom., 7(1), pp. 23–29.

[80] Kim, D., and Chung, W. Y., 1999, “Analytic Singularity Equation and Analysis of Six-DOF Parallel Manipulators Using Local Structurization Method,” IEEE Trans. Rob. Autom., 5(4), pp. 612–622.

[81] Mayer St-Onge, B., and Gosselin, C. M., 2000, “Singularity Analysis and Representation of the General Gough-Stewart Platform,” Int. J. Robot. Res., 19(3), pp. 271–288.

[82] Li, H., Gosselin, C. M., Richard, M. J., and St-Onge, B. M. (April 27, 2005). "Analytic Form of the Six-Dimensional Singularity Locus of the General Gough-Stewart Platform." ASME. J. Mech. Des. January 2006; 128(1): 279–287. https://doi.org/10.1115/1.2118733

[83] Hao, F., and McCarthy, J. M., 1998, “Conditions for Line-Based Singularities in Spatial Platform Manipulators,” J. Rob. Syst., 15(1), pp. 43–55.

[84] Ben-Horin, P., and Shoham, M., 2009, “Application of Grassmann-Cayley Algebra to Geometrical Interpretation of Parallel Robot Singularities,” Int. J. Rob. Res., 28(1), pp. 127–141.

[85] Shai, O., Tehori, I., Bronfeld, A., Slavutin, M., and Ben-Hanan, U., 2009, “Adjustable Tensegrity Robot Based on Assur Graph Principle,” ASME Paper No. IMECE2009-11301.
[86] Wolf, A., and Shoham, M., 2003, “Investigation of Parallel Manipulators Using Linear Complex Approximation,” ASME J. Mech. Des., 125(3), pp. 564–572.
[87] Merlet, J.-P., 1989, “Singular Configurations of Parallel Manipulators and Grassmann Geometry,” Int. J. Rob. Res., 8(5), pp. 45–56.
[88] Ben-Horin, P., and Shoham, M., 2006, “Singularity Condition of Six-Degreeof-Freedom Three-Legged Parallel Robots Based on Grassmann-Cayley Algebra,” IEEE Trans. Rob., 22(4), pp. 577–590.
[89] Ben-Horin, P., and Shoham, M., 2006, “Singularity Analysis of a Class of Parallel Robots Based on Grassmann–Cayley Algebra,” Mech. Mach. Theory, 41(8), pp. 958–970.
[90] Slavutin, M., Sheffer, A., Shai, O., and Reich, Y. (June 1, 2018). "A Complete Geometric Singular Characterization of the 6/6 Stewart Platform." ASME. J. Mechanisms Robotics. August 2018; 10(4): 041011. https://doi.org/10.1115/1.4040133
[91] Downing, D. M., Samuel, A. E., and Hunt, K. H., 2002, “Identification of the Special Configurations of the Octahedral Manipulator Using the Pure Condition,” Int. J. Rob. Res., 21(2), pp. 147–159.
[92] Di Gregorio. R., 2002, "Singularity-Locus Expression of A Class of Parallel Mechanisms", Robotica, Vol.20, No.3, pp.323-328.
[93] Q. Jiang, Singularity-free Workspace Analysis and Geometric Optimization of Parallel Mechanisms, (PhD thesis) Laval University, Quebec, QC, Canada, October 2008.
[94] Gilles Reymond & Andras Kemeny (2000) Motion Cueing in the Renault Driving Simulator, Vehicle System Dynamics, 34:4, 249-259, DOI: 10.1076/vesd.34.4.249.2059
[95] Telban, R. J., Cardullo, F. M., et Guo, L. (2000). Investigation of mathematical models of otolith organs for human centered motion cueing algorithms. In AIAA Modeling and Simulation Technologies Conference, Denver CO.
[96] http://www.dizziness-and-balance.com/disorders/bppv/otoliths.html.
[97] Encyclopedia Britannica. human ear. Ultimate Reference Suite, 2008.
[98] Fernandez, C. et Goldberg, J. M. (1976). Physiology of peripheral neurons innervating otolith organs of the squirrel units, iii : Response dynamics. Journal of Neurophysiology.
[99] Berthoz, A. et Droulez, J. (1980). Tutorial on linear motion perception. Laboratoire de physiology neurosensorielle du CNRS.
[100] Bohoua-Nassé, F.-O. (2004). Compréhension et modélisation du comportement conducteur et de sa perception. Mémoire de Master, Centre de Mathématiques Appliquées, École desMines de Paris, Juillet.
[101] Redrawn from Kandel ER, Schwartz JH: Principles of Neural Science. New York, Elsevier, 1981.
[102] Nahon, M. A. et Reid, L. D. (1989). Simulator motion-drive algorithms : a designer perspective. Journal of guidance, 3(2) :356–362.
[103] Parrish, R. V., Dieudone, J. E., Bowles, R. L., et Martin, D. J. (1975). Coordinated adaptive washout for motion simulators. Journal of aircraft, 12(1) :44–50.
[104] R.J. Telban et F.M. Cardullo. Motion cueing algorithm development : Human-centered linear and nonlinear approaches. Rapport technique, National Aeronautics and Space Administration, Washington DC, 2005a. NASA CR-213747.
[105] L.R. Young et J.L. Meiry. A revised dynamic otolith model. Aerospace Medicine, 39(6) : 606–608, 1968.

[106] J.L. Meiry. The vestibular system and human dynamics space orientation. PhD thesis, Massachusetts Institut of Technology, USA, 1965.

[107] H. ELLOUMI, Commande des plates-formes avancées de simulation de conduite, thèse de doctorat, école mines de paris, 2006.

[108] Reymond, G. (2000a). Contribution des stimuli visuals, vestibulaires et proprioceptifs dans la perception du mouvement du conducteur. Thèse de Doctorat, Collége de France Renault.

[109] R.A. Mayne. A systems concept of the vestibular organs. Handbook of Sensory Physiology, Vestibular System, Ed Springer-Verlag, pages 493–560, 1974.

[110] R. Schmid, A. Buizza, , et D. Zambarbieri. Modelling of the vestibulo-ocular reflex and its use in clinical vestibular analysis. Applied Physiological Mechanics, pages 779–893, 1979.

[111] L.R. Young et C.M. Oman. Model for vestibular adaptation to horizontal rotation. Aerospace Medicine, 40(10) :1076–1080, 1969.

[112] G.M. Zacharias. Motion cue models for pilot-vehicle analysis. Rapport technique, Department of Defense, Alexandria, VA, 1978. AMRL-TR-78-2.

[113] Asadi, Houshyar, Mohamed, Shady, Rah, Del, Nahavandi, Saeid. (2015). Optimisation of nonlinear motion cueing algorithm based on genetic algorithm. Vehicle System Dynamics.

[114] Tang Zhiyong, Ma Hu, Pei Zhongcai and Zhang Jinhui, "Adaptive motion cueing algorithm based on fuzzy tuning for improving human sensation," 2016 IEEE Chinese Guidance, Navigation and Control Conference (CGNCC), Nanjing, 2016, pp. 1200-1205. doi: 10.1109/CGNCC.2016.7828958.

[115] Shmidt, S. F. et Conrad, B. (1970). Motion drive signals for piloted flight simulators. Rapport technique, NASA.

[116] Sivan, R., Ish–Shalom, J., et Huang, J. K. (1982). An optimal control approach to the design of moving flight simulators. IEEE Transactions on SystemsMan and Cybernetics.

[117] J.M. Goldberg et C. Fernandez. The vestibular system. Handbook of Physiology - The nervous system III, pages 916–977, 1984.

[118] L. Nehaoua, H. Arioui, S. Espie and H. Mohellebi, "Motion cueing algorithms for small driving simulator," Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006., Orlando, FL, 2006, pp. 3189-3194. doi: 10.1109/ROBOT.2006.1642187

[119] M. Dagdelen, G. Reymond, A. Kemeny, M. Bordier, et N. Maïza. MPC based motion cueing algorithm : Development and application to the ULTIMATE driving simulator. Dans Driving Simulation Conference DSC Europe 2004, pages 221–233, Paris-France, 2004.

[120] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science, Nagoya, Japan, 1995, pp. 39-43, doi: 10.1109/MHS.1995.494215.

[121] Y. Shi and R. Eberhart, "A modified particle swarm optimizer," 1998 IEEE International Conference on Evolutionary Computation Proceedings. IEEE World Congress on Computational Intelligence (Cat. No.98TH8360), Anchorage, AK, USA, 1998, pp. 69-73, doi: 10.1109/ICEC.1998.699146.

[122] Rini, dian Palupi & Shamsuddin, Siti Mariyam & Yuhaniz, Siti. (2011). Particle Swarm Optimization: Technique, System and Challenges. International Journal of Computer Applications.

[123] Hassan, Rania & Cohanim, Babak & de Weck, Olivier. (2005). A Comparison of Particle Swarm Optimization and the Genetic Algorithm.

[124] Federico Marini, Beata Walczak, Particle swarm optimization (PSO). A tutorial, Chemometrics and Intelligent Laboratory Systems, Volume 149, Part B, 2015, Pages 153-165, ISSN 0169-7439, https://doi.org/10.1016/j.chemolab.2015.08.020.

[125] Shih-Wei Lin, Kuo-Ching Ying, Shih-Chieh Chen, Zne-Jung Lee, Particle swarm optimization for parameter determination and feature selection of support vector machines, Expert Systems with Applications, Volume 35, Issue 4, 2008, Pages 1817-1824, ISSN 0957-4174, https://doi.org/10.1016/j.eswa.2007.08.088.

[126] H. Asadi et al., "A Particle Swarm Optimization-based washout filter for improving simulator motion fidelity," 2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Budapest, 2016, pp. 001963-001968, doi: 10.1109/SMC.2016.7844527.

[127] Sergio Casas, Cristina Portalés, Pedro Morillo, Marcos Fernández, A particle swarm approach for tuning washout algorithms in vehicle simulators, Applied Soft Computing, Volume 68, 2018, Pages 125-135, ISSN 1568-4946, https://doi.org/10.1016/j.asoc.2018.03.044.

[128] Y. Guangyou, "A Modified Particle Swarm Optimizer Algorithm," 2007 8th International Conference on Electronic Measurement and Instruments, Xi'an, 2007, pp. 2-675-2-679, doi: 10.1109/ICEMI.2007.4350772.

[129] Ish–Shalom, J. (1982). The design of optimal control motion for flight simulators. Thèse de Doctorat, MIT.

[130] Reid, L. D. et Nahon, M. A. (1985). Flight simulation motion-base drive algorithms : Part 1 – developing and testing the equations. CN ISSN 0082–5255 296, UTIAS.

[131] Cardullo, F. M., Telban, R. J., et Houck, J. A. (1999). Motion cueing algorithms : A human centered approach. In 5th International Symposium on Aeronautical Sciences.

[132] R.J. Telban et F.M. Cardullo. Motion cueing algorithm development : New motion cueing program implementation and tuning. Rapport technique, National Aeronautics and Space Administration, Washington DC, 2005b. NASA CR-213746.

[133] H. Kwakernaak and R. Sivan, Linear Optimal Control Systems, vol. 1. New York, NY, USA: Wiley, 1972.

[134] J. Kotlarski, T. Do Thanh, B. Heimann and T. Ortmaier, "Optimization strategies for additional actuators of kinematically redundant parallel kinematic machines," 2010 IEEE International Conference on Robotics and Automation, Anchorage, AK, 2010, pp. 656-661, doi: 10.1109/ROBOT.

[135] B. W. Satzinger, J. I. Reid, M. Bajracharya, P. Hebert and K. Byl, "More solutions means more problems: Resolving kinematic redundancy in robot locomotion on complex terrain," 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems, Chicago, IL, 2014, pp. 4861-4867, doi: 10.1109/IROS.2014.6943253.

[136] Tanaka, M., Matsuno, F. Modeling and Control of Head Raising Snake Robots by Using Kinematic Redundancy. J Intell Robot Syst 75, 53–69 (2014).

[137] Orekhov, A. L., and Simaan, N. (July 8, 2019). "Directional Stiffness Modulation of Parallel Robots With Kinematic Redundancy and Variable Stiffness Joints." ASME. J. Mechanisms Robotics. October 2019; 11(5): 051003. https://doi.org/10.1115/1.4043685

[138] S. Cha, T. A. Lasky and S. A. Velinsky, "Singularity Avoidance for the 3-RRR Mechanism Using Kinematic Redundancy," Proceedings 2007 IEEE International Conference on Robotics and Automation, Roma, 2007, pp. 1195-1200, doi: 10.1109/ROBOT.2007.363147.

[139] J. Baillieul, "Avoiding obstacles and resolving kinematic redundancy," Proceedings. 1986 IEEE International Conference on Robotics and Automation, San Francisco, CA, USA, 1986, pp. 1698-1704, doi: 10.1109/ROBOT.1986.1087464.

[140] Y. S. Xia, Gang Feng and Jun Wang, "A primal-dual neural network for online resolving constrained kinematic redundancy in robot motion control," in IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), vol. 35, no. 1, pp. 54-64, Feb. 2005, doi: 10.1109/TSMCB.2004.839913.

[141] H. Zhang, H. Jin, Z. Liu, Y. Liu, Y. Zhu and J. Zhao, "Real-Time Kinematic Control for Redundant Manipulators in a Time-Varying Environment: Multiple-Dynamic Obstacle Avoidance and Fast Tracking of a Moving Object," in IEEE Transactions on Industrial Informatics, vol. 16, no. 1, pp. 28-41, Jan. 2020, doi: 10.1109/TII.2019.2917392

[142] J. Baillieul, J. M. Hollerbach, and R. W. Brockett, “Programming and control of kinematically redundant manipulators,” in Proc. 23rd IEEE Conf. Decision Contr., Las Vegas, NV, Dec. 1984, pp. 768–774

[143] C. A. Klein and C.-H. Huang, “Review of pseudoinverse control for use with kinematically redundant manipulators,” IEEE Trans. Syst., Man, Cybern., vol. SMC-13, pp. 245–250, 1983.

[144] Lukic B., Jovanovic K., Knezevic N., Zlajpah L., Petric T. (2020) Maximizing the End-Effector Cartesian Stiffness Range for Kinematic Redundant Robot with Compliance. In: Zeghloul S., Laribi M., Sandoval Arevalo J. (eds) Advances in Service and Industrial Robotics. RAAD 2020. Mechanisms and Machine Science, vol 84. Springer, Cham.

[145] Woolfrey, J., Lu, W. & Liu, D. A Control Method for Joint Torque Minimization of Redundant Manipulators Handling Large External Forces. J Intell Robot Syst 96, 3–16 (2019).

[146] F. Chen, M. Selvaggio and D. G. Caldwell, "Dexterous Grasping by Manipulability Selection for Mobile Manipulator With Visual Guidance," in IEEE Transactions on Industrial Informatics, vol. 15, no. 2, pp. 1202-1210, Feb. 2019, doi: 10.1109/TII.2018.2879426.

[147] Alireza Izadbakhsh & Saeed Khorashadizadeh (2018) Robust impedance control of robot manipulators using differential equations as universal approximator, International Journal of Control, 91:10, 2170-2186, DOI: 10.1080/00207179.2017.1336669

[148] Saeed Khorashadizadeh, Mahdi Sadeghijaleh, Adaptive fuzzy tracking control of robot manipulators actuated by permanent magnet synchronous motors, Computers & Electrical Engineering, Volume 72, 2018, Pages 100-111, ISSN 0045-7906, https://doi.org/10.1016/j.compeleceng.2018.09.010.

[149] Arteaga, M. A., Pliego Jiménez, J., & Romero Velazquez, J. G. (2019). Experimental Results on the Robust and Adaptive Control of Robot Manipulators Without Velocity Measurements. IEEE Transactions on Control Systems Technology, 1-4. doi: 10.1109/TCST.2019.2945915

[150] D. Chen, S. Li, Q. Wu and X. Luo, "New Disturbance Rejection Constraint for Redundant Robot Manipulators: An Optimization Perspective," in IEEE Transactions on Industrial Informatics, vol. 16, no. 4, pp. 2221-2232, April 2020, doi: 10.1109/TII.2019.2930685.

[151] Mien Van, Xuan Phu Do, Michalis Mavrovouniotis, Self-tuning fuzzy PID-nonsingular fast terminal sliding mode control for robust fault tolerant control of robot manipulators, ISA Transactions, Volume 96, 2020, Pages 60-68, ISSN 0019-0578, https://doi.org/10.1016/j.isatra.2019.06.017.

[152] J. Baek, W. Kwon and C. Kang, "A New Widely and Stably Adaptive Sliding-Mode Control With Nonsingular Terminal Sliding Variable for Robot Manip-

ulators," in IEEE Access, vol. 8, pp. 43443-43454, 2020, doi: 10.1109/ACCESS.2020.2977434.

[153] N.M.H. Norsahperi, K.A. Danapalasingam, An improved optimal integral sliding mode control for uncertain robotic manipulators with reduced tracking error, chattering, and energy consumption, Mechanical Systems and Signal Processing, Volume 142, 2020, 106747, ISSN 0888-3270, https://doi.org/10.1016/j.ymssp.2020.106747.

[154] Perruquetti. W, Barbot. J. P., Sliding Mode Control in Engineering, Marcel Dekker, inc, ISBN 0-8247-0671-4, 2002

[155] Ferrara, A., Magnani, L. Motion Control of Rigid Robot Manipulators via First and Second Order Sliding Modes. J Intell Robot Syst 48, 23–36 (2007).

[156] H. Ma, J. Wu and Z. Xiong, “A Novel Exponential Reaching Law of Discrete-Time Sliding-Mode Control,” IEEE Trans. Ind. Electron., vol. 64, no. 5, pp. 3840-3850, May 2017.

[157] J.J.E. Slotine and W. Li. Applied Nonlinear Control. Englewood Cliffs, NJ: Prentice-Hall, 1991.

[158] R. Solea, D.C. Cernega. Sliding Mode Control for Trajectory Tracking Problem - Performance Evaluation. Artificial Neural Networks – ICANN 2009, Lecture Notes in Computer Science Volume 5769, 2009: 865-874.

[159] R. Mendes, P. Cortez, MR. Rocha, J. Firns. Particle Swarms for Feedforward Networks Trening. International Conference on Neural Networks, Honolulu (Hawaii), USA, 2002: 1895-1889.

[160] T.H. Kim, I. Maruta, T. Sugie. Robust PID controller tuning based on the constrained particle swarm optimization. Automatica 2008, 44: 1104-1110.

[161] Serbencu, Adrian & Serbencu, Adriana & Cernega, D. & Mînzu, Viorel. (2010). Particle swarm optimization for the Sliding Mode controller parameters. 1859-1864.

[162] Hashem Zadeh, Seyed Mohammad & Khorashadizadeh, Saeed & Fateh, Mohammad & Hadadzarif, Mohammad. (2016). Optimal sliding mode control of a robot manipulator under uncertainty using PSO. Nonlinear Dynamics.

[163] Grigore C. Burdea and Philippe Coiffet. 2003. Virtual Reality Technology (2nd. ed.). John Wiley & Sons, Inc., USA.

[164] Timothy Jung and M. Claudia tom Dieck. 2017. Augmented Reality and Virtual Reality: Empowering Human, Place and Business (1st. ed.). Springer Publishing Company, Incorporated.

[165] Atsuki Ikeda, Dong-Hyun Hwang, and Hideki Koike. 2018b. Real-time Visual Feedback for Golf Training Using Virtual Shadow. In Proceedings of the 2018 ACM International Conference on Interactive Surfaces and Spaces (ISS ’18). ACM, New York, NY, USA, 445–448.

[166] T. Nozawa, E. Wu and H. Koike, "VR Ski Coach: Indoor Ski Training System Visualizing Difference from Leading Skier," 2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR), Osaka, Japan, 2019, pp. 1341-1342, doi: 10.1109/VR.2019.8797717.

[167] R. Oboe, R. Antonello and F. Biral, "Development of a water ski simulator for indoor training with proprioceptive and visual feedback," 2014 IEEE 13th International Workshop on Advanced Motion Control (AMC), Yokohama, 2014, pp. 428-433, doi: 10.1109/AMC.2014.6823320.

[168] Wu, Erwin & Perteneder, Florian & Koike, Hideki & Nozawa, Takayuki. (2019). How to VizSki: Visualizing Captured Skier Motion in a VR Ski Training Simulator. 1-9. 10.1145/3359997.3365698.

[169] Hu, Chen-Hui & Lee, Chien-Ying & Liou, Yen-Ting & Sung, Feng-Yu & Lin, Wen-Chieh. (2019). Skiing Simulation Based on Skill-Guided Motion Planning: Skiing Simulation. Computer Graphics Forum. 38. 10.1111/cgf.13606.

[170] Bhambhani Y, Forbes S, Forbes J, Craven B, Matsuura C, Rodgers C. Physiologic responses of competitive Canadian cross-country skiers with disabilities. Clin J Sport Med. 2012;22(1):31-38. doi:10.1097/JSM.0b013e3182432f0c
[171] Caracas, Valentin. (2019). Educating Balance In Alpine Skiing With A Simulation Platform
[172] Nozawa, Takayuki & Wu, Erwin & Perteneder, Florian & Koike, Hideki. (2019). Visualizing expert motion for guidance in a VR ski simulator.
[173] Jorand, Dominique & Suchet, André. (2013). La pratique handiski en station de montagne : implications aménagistes, professionnelles et socio-économiques dans les Alpes du Nord françaises. 10.13140/2.1.3279.3283.
[174] Hofmann K, Ohlsson M, Ho¨ok M, Danvind J, Kersting UG (2016) ¨ The influence of sitting posture on mechanics and metabolic energy requirements during sit-skiing: a case report. Sports Eng 19:213–218
[175] https://unity.com/
[176] https://www.blender.org/
[177] http://www.dualski.com/en/leisure-uniski-dualski/
[178] Sprunk, Christoph. “Planning Motion Trajectories for Mobile Robots Using Splines.” (2008).

Appendix A

Appendix

A.1 Simulator architecture and main applications

Simulator	Country	Type	Architecture	DoF	Movements
Mercedes S	Germany	Automobile	Fixed	0	none
Renault	France - 2000	Automobile	Stewart	6	$x, y, z, \phi, \theta, \psi$
VIRTTEX	USA - 2001	Automobile	Stewart	6	$x, y, z, \phi, \theta, \psi$
Airbus	France	Aerospace	Stewart	6	$x, y, z, \phi, \theta, \psi$
Sukhoi	Russia	Aerospace	Stewart	6	$x, y, z, \phi, \theta, \psi$
Mega-Ankle	USA	Rehabilitation	Stewart	6	$x, y, z, \phi, \theta, \psi$
CableRobot	Germany	Flight,driving	Cable	6	$x, y, z, \phi, \theta, \psi$
Ultimate	France - 2004	Automobile	Hybrid	7	$Y, x, y, z, \phi, \theta, \psi$
Daimler	Germany - **1995**	Automobile	Hybrid	**8**	**X, Y,** $x, y, z, \phi, \theta, \psi$
NADS	USA - **2001**	Automobile	Hybrid	**8**	**X, Y,** $x, y, z, \phi, \theta, \psi$
Toyota	Japan - **2008**	Automobile	Hybrid	**8**	**X, Y,** $x, y, z, \phi, \theta, \psi$
Mercedes	Germany - **2010**	Automobile	Hybrid	**8**	**X, Y,** $x, y, z, \phi, \theta, \psi$
XY-6 DoF	France - **2016**	Open	Hybrid	**8**	**X, Y,** $x, y, z, \phi, \theta, \psi$
BMW	Germany - 2020	Automobile	Hybrid	8	$X, Y, x, y, z, \phi, \theta, \psi$

Table A.1: Summary comparative table of simulators. Displacements: x (longitudinal), y (lateral), z (vertical). Rotations: ϕ (roll), θ (pitch), ψ (yaw)

Tables Tab. A.2 and Tab.A.3 compare the performance of 4 industrial simulators in terms of movement workspace.

Simulator	Longitudinal	Lateral	Vertical	Pitch	Roll	Yaw
VIRTTEX	± 1.6 m 1.2 ms^{-1} >0.6 g	± 1.6 m 1.2 ms^{-1} >0.6 g	± 1 m 1 ms^{-1} 1 g	± 20° >20°s^{-1} >200°s^{-2}	± 20° >20°s^{-1} >200°s^{-2}	± 40° >20°s^{-1} >200°s^{-2}
Daimler Chrysler	+1.38 m, -1.8 m 1 ms^{-1} 1.2 g	± 1.5 m 1 ms^{-1} 1.2 g	± 1.7 m 1 ms^{-1} 1.2 g	± 32° >25°s^{-1} >200°s^{-2}	± 30° >25°s^{-1} >200°s^{-2}	± 47.4° >25°s^{-1} >200°s^{-2}
NADS	± 0.8 m ? ms^{-1} 1 g	± 0.8 m ? ms^{-1} 1 g	± 0.6 m 1.5 ms^{-1} 0.75 g	± 25° 45°s^{-1} >120°s^{-2}	± 25° 45°s^{-1} >120°s^{-2}	± 330° >60°s^{-1} >120°s^{-2}
Ultimate	± 0.28 m 0.7 m° 0.75 g	± 0.26 m 0.7 ms^{-1} 0.75 g	± 0.2 m 0.4 ms^{-1} 0.5 g	± 18° 40°s^{-1} 300°s^{-2}	± 18° 40°s^{-1} 300°s^{-2}	± 23° 60°s^{-1} 600°s^{-2}

Table A.2: Comparative Table of Motion Envelopes of Hexapod Based Simulation Robots

Simulator	Bandwidth	X	Y
VIRTTEX	13 Hz	0	0
Daimler Chrysler	3 Hz	0	± 2.3 m 4 m/s 0.7 g
NADS	3-20 Hz	± 9.75 m 6 m/s 0.62 g	± 9.75 m 6 m/s 0.62 g
Ultimate	.	± 2.5 m 2.5 m/s 0.5 g	± 2.5 m 3 m/s 0.5 g

Table A.3: Comparison table (continued) bandwidths of hexapods and the corresponding rail system

A.2 Modeling appendix

A.2.1 Recursive equations

Forward recursive equations [**?**]:

$$
\begin{aligned}
{}^{j}\omega_j &= {}^{j}A_{j-1}\,{}^{j-1}\omega_{j-1} + \overline{\sigma}_j \dot{q}_j\,{}^{j}a_j = {}^{j}\omega_{j-1} + \sigma_j \dot{q}_j\,{}^{j}a_j \\
{}^{j}\dot{\omega}_j &= {}^{j}A_{j-1}\,{}^{j-1}\dot{\omega}_{j-1} + \overline{\sigma}_j(\ddot{q}_j\,{}^{j}a_j + {}^{j}\dot{\omega}_{j-1} \times \dot{q}_j\,{}^{j}a_j) \\
{}^{j}V_j &= {}^{j}A_{j-1}({}^{j-1}V_{j-1} + {}^{j-1}\omega_{j-1} \times {}^{j-1}P_j) + \sigma_j \dot{q}_j\,{}^{j}a_j \\
{}^{j}\dot{V}_j &= {}^{j}A_{j-1}({}^{j-1}\dot{V}_{j-1} + ({}^{j}\dot{\omega}_j + {}^{j}\omega_j \times {}^{j}\omega_j) \times {}^{j-1}P_j) \\
&\quad + \sigma_j(\ddot{q}_j\,{}^{j}a_j + 2\,{}^{j}\omega_{i-1} \times \dot{q}_j\,{}^{j}a_j) \\
{}^{j}F_j &= M_j\,{}^{j}\dot{V}_j + ({}^{j}\dot{\omega}_j + {}^{j}\omega_j \times {}^{j}\omega_j) \times {}^{j}MS_j \\
{}^{j}N_j &= {}^{j}J_j\,{}^{j}\dot{\omega}_j + {}^{j}\dot{\omega}_j \times ({}^{j}J_j\,{}^{j}\omega_j) + {}^{j}MS_j \times {}^{j}\dot{V}_j
\end{aligned}
\tag{A.2.1}
$$

Backward recursive equations:

$$\begin{aligned}
{}^{j}f_j &= {}^{j}F_j + {}^{j}f_{ej} \\
{}^{j}n_j &= {}^{j}N_j + {}^{j}n_{ej} \\
{}^{j-1}f_{e,j-1} &= {}^{j}f_{e,j} + {}^{j-1}A_j\,{}^{j}f_j \\
{}^{j-1}n_{e,j-1} &= {}^{j}n_{e,j} + {}^{j-1}A_j\,{}^{j}f_j + {}^{j-1}P_j \times {}^{j-1}f_j \\
\Gamma_j &= (\sigma_j\,{}^{j}f_j + \overline{\sigma}_j\,{}^{j}n_j)^T\,{}^{j}a_j + I a_j \ddot{q}_j \\
&\quad + F_{sj}\,sign(\dot{q}_j) + F_{vj}\dot{q}_j
\end{aligned} \tag{A.2.2}$$

A.2.2 e_k and u_k vectors

ξ_k^{in}	XX	XY	XZ	YY	YZ
e_k	$\frac{1}{2}\omega_{1,k}\omega_{1,k}$	$\frac{1}{2}\omega_{1,k}\omega_{2,k}$	$\frac{1}{2}\omega_{1,k}\omega_{3,k}$	$\frac{1}{2}\omega_{2,k}\omega_{2,k}$	$\frac{1}{2}\omega_{2,k}\omega_{3,k}$
u_k	0	0	0	0	0
ξ_k^{in}	ZZ	MX	MY	MZ	M
e_k	$\frac{1}{2}\omega_{3,k}\omega_{3,k}$	$\omega_{3,k}V_{2,k}$ $-$ $\omega_{2,k}V_{3,k}$	$\omega_{1,k}V_{3,k}$ $-$ $\omega_{3,k}V_{1,k}$	$\omega_{2,k}V_{1,k}$ $-$ $\omega_{1,k}V_{2,k}$	$\frac{1}{2}{}^{k}V_k^{Tk}V_k$
u_k	0	$-{}^{0}g^{T0}s_k$	$-{}^{0}g^{T0}n_k$	$-{}^{0}g^{T0}a_k$	$-{}^{0}g^{T0}P_k$

Table A.4: e_k and u_k vector elements

where, ${}^{k}\omega_k = [\omega_{1,k}\ \omega_{2,k}\ \omega_{3,k}]^T$, ${}^{k}V_k = [V_{1,k}\ V_{2,k}\ V_{3,k}]^T$ and ${}^{0}R_k$ and ${}^{0}P_k$ are the matrix and vector appearing in the transformation matrix ${}^{0}T_k$ with the rotation matrix ${}^{0}R_k = [{}^{0}s_k\,{}^{0}n_k\,{}^{0}a_k]$, a_k unit vector along Z_k direction, ${}^{k}J_k$ - (3×3) inertia tensor of link k with respect to frame F_k. M_k and MS_k are the mass and first moment of inertia of link k respectively.

The vector ξ_k is composed of the standard dynamic parameters of the link k. $\xi_k^{in} = [XX_k\ XY_k\ XZ_k\ YY_k\ YZ_k\ ZZ_k\ MX_k\ MY_k\ MZ_k\ M_k]^T \in \mathbb{R}^{10\times1}$

1. XX_k , XY_k , XZ_k , YY_k , YZ_k , ZZ_k are the 6 components of the inertia tensor of the body j C with respect to the reference j R;
2. MX_k , MY_k , MZ_k are the 3 components of the first moment of inertia produced by mass by the coordinates of the center of gravity of the body j C in the reference j R;
3. M_k is the mass of the link k;

A.3 IBISC simulator description

We describe in the following the mechanical transmissions, the electric motors, the tracking camera and the communication tools that the platform is composed of. Since the safety is a primary issue in the field of the Human-Robot interaction, a description of the safety environment definition methods were also detailed in the following.

Firstly we design the robotic platform using CAD software as the parallel robot type Gough-Stewart platform allowing 6 degree of freedom motion in order to

cover all sliding motion. To overcome the Gough-Stewart platfrom workspace limitation we propose to design two sliders that carry the parallel robot. Then, we manufactured the appropriate designed parts. The designed parts on solidworks were fully manufactured and assembled within the IBISC laboratory.

In the following figure, we present the description and the near view of mechanical parts and mechanical transmission systems of the entire XY-6 DoF simulator, we show in the following figure in yellow color the CAD parts as well as the corresponding manufactured parts.

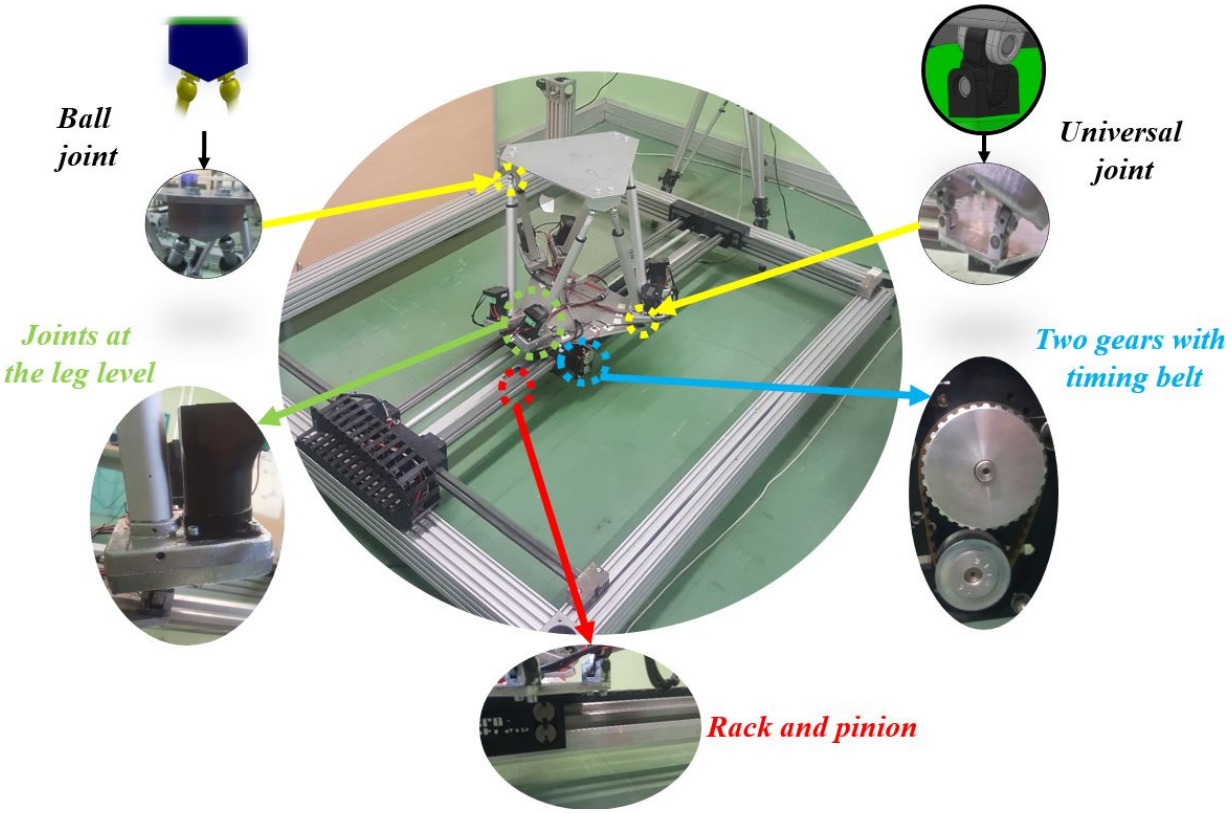


Figure A.1: Real photo of the manufactured platform

The disabled person will be attached to XY-6 DoF simulator, thus, to better recover the acceleration for humans at vestibular system level, the simulator must be able to support a high load. The electric actuator has a high speed with moderate torque, that is why it is important to design a mechanical system that increases the torque and decreases the unnecessary high speed. In other words, to regulate the motor output to the required speed and torque range. The required speed depends on the human vestibular model thresholds that will be described in detail in chapitre 5.

where θ (tr/s), R (mm), X (mm), Γ $(N.cm)$, F (N), $R_1 = 3\ cm$, $R_2 = 6\ cm$ and $R_3 = 0.65\ cm$ are the characteristic parameters of the mechanical transmission of the sliders. Which means that the platform moves $2.042\ cm$ for each motor revolution.

A.3.1 Mechanical transmission of the hexapod legs

Similarly, for the parallel platform, it is necessary to introduce a mechanical system capable of transforming the rotational movement into translational movement with regularization at the required speed and torque.

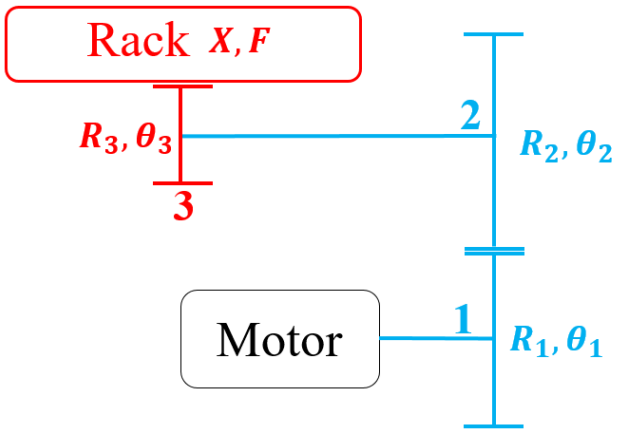


Figure A.2: Sliders Mechanical Transmission Representation

Axis 1	Axis 2	Axis 3	rack
θ_1	$\theta_2 = \frac{\theta_1}{2}$	$\theta_3 = \theta_2$	$X = 2\pi R_3 \theta_3$
$\dot{\theta}_1$	$\dot{\theta}_2 = \frac{\dot{\theta}_1}{2}$	$\dot{\theta}_3 = \dot{\theta}_2$	$\dot{X} = 2\pi R_3 \dot{\theta}_3$
$\ddot{\theta}_1$	$\ddot{\theta}_2 = \frac{\ddot{\theta}_1}{2}$	$\ddot{\theta}_3 = \ddot{\theta}_2$	$\ddot{X} = 2\pi R_3 \ddot{\theta}_3$
Γ_1	$\Gamma_2 = 2\Gamma_1$	$\Gamma_3 = \Gamma_2$	$F = \frac{\Gamma_3}{R_3}$

Table A.5: Sliders Mechanical Transmission

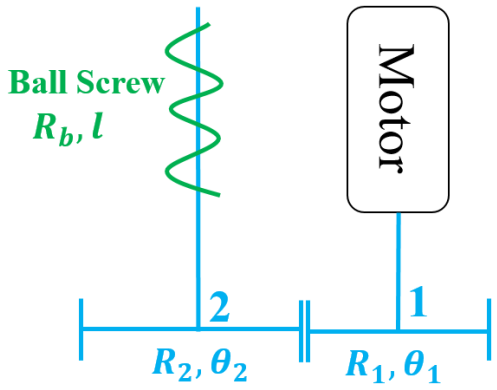


Figure A.3: Leg Mechanical Transmission Representation

Axis 1	Axis 2	Ball screw
θ_1	$\theta_2 = \frac{\theta_1}{2}$	$l = 2\pi R_b \theta_2$
$\dot{\theta}_1$	$\dot{\theta}_2 = \frac{\dot{\theta}_1}{2}$	$\dot{l} = 2\pi R_b \dot{\theta}_2$
$\ddot{\theta}_1$	$\ddot{\theta}_2 = \frac{\ddot{\theta}_1}{2}$	$\ddot{l} = 2\pi R_b \ddot{\theta}_2$
Γ_1	$\Gamma_2 = 2\Gamma_1$	$F = \frac{\Gamma_2}{R_b}$

Table A.6: Leg Mechanical Transmission

with $R_1 = 3\ cm$, $R_2 = 6\ cm$ and $R_b = 0.15\ cm$. Which means that the leg moves $0.4725\ cm$ for each motor revolution.

A.3.2 Nanotec PD4-CB motor

The PD4-CB motor is represented in Fig. A.4.

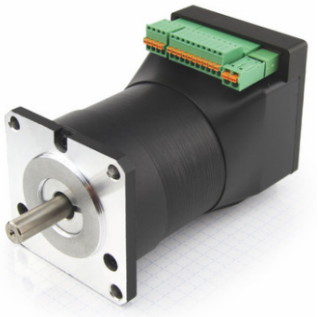


Figure A.4: PD4-CB motor

Table A.7 presents the electric motor characteristics.
It is a brushless motor, that means without brushes. It is powered by DC voltage from 12 V to 24 V with a current of 8 Amperes (maximum current allowed over a period of 1 second: 20 A). The motor offers a complete package: electric

Characteristics	Rated Power	Rated Speed	Peak Current (RMS)	Rated Torque
	135 W	3500 rpm	20 A	37 Ncm
Operating Voltage	Position sensor	Speed sensor	Torque sensor	Weight
12 VDC - 24 VDC	✓	✓	✓	0.9 kg

Table A.7: PD4-CB characteristics

motor with integrated power supply board which deals with signal amplification, connection to CAN protocol, sensor processing and several types of controllers.

Therefore, it is possible to operate the motor according to several modes of operation: position, speed, torque, homing mode, interpolated position, cyclic synchronous position, cyclic synchronous speed, synchronous synchronous torque. This wide use of the motor allows the good functionality of the robotic platform and facilitates handling. Its programming will be via a CANopen bus, transmitting analog values that will be described in the following.

A.3.3 CANopen programming

CANopen is a protocol that occurs at layer 3 of the OSI model (network layer). This protocol is used in fieldbus devices such as the Controller Area Network (CAN) bus and enables real-time control of systems, hence its use in the automotive or medical industry, but also it is very dedicated for robotic applications as ours. It can be added that CANopen is an economical and reliable solution.

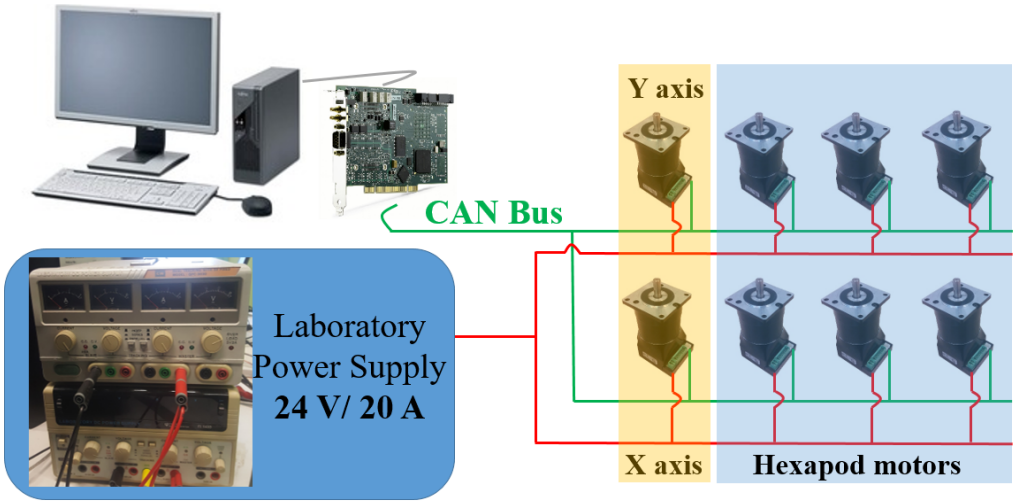


Figure A.5: Computer motors connection

The CANopen network operates according to a master / slave architecture. The master performs actions on one or more slaves:

- Initialization of slaves
- Supervision of slaves
- Communication of slave states

The communication is done by a single cable to which each slave and master are connected. Thus, the 8 actuators of our system are interconnected and controlled by a computer, CANopen integrates different services: NMT, SDO, PDO.

- NMT (Network Management) follows a master / slave structure. It defines the structure of CANopen.
- SDO (Service Data Object) contains all information about the data. Each message can contain a maximum of 8 bytes (64 bits) of data. The SDO configures itself in reception (description of the process of the SDO messages to be received) and in transmission (description of the SDO messages to be sent).
- PDO (Process Data Object) contains only the data to be sent. Each message can contain a maximum of 8 bytes (64 bits) of data. The PDO configures itself in reception (description of the process of the PDO messages to be received) and in transmission (description of the PDO messages to be sent).

A.3.4 Very high speed communication via the NI PCI 8513 Card

To control the robotic platform, we choose the National Instruments PCI-8513 CAN Interface board. This is the software selectable version / FD, 1 port:

Figure A.6: NI PCI-8513 Card

The PCI 8513 is a CAN FD (Controller Area Network / Flexible Data Rate) interface for developing applications with the NI XNET driver. Software-selectable NI XNET CAN interfaces provide the best flexibility for CAN development with integrated transceivers for high-speed / FD, low-speed / fault-tolerant and single-wire ADC as well as any external transceiver . The PCI 8513 is ideally suited to applications that require high-speed, real-time manipulation of hundreds of signals and CAN frames (**1 MBd: 1 megabaud**) , such as the control of our platform for example. It's about being able to send CAN frames to random content through MATLAB, without interruption. All technical specifications can be found in the technical appendices.

A.3.5 Advanced real time tracking

"Motion Capture" is the technique of digitizing the movement of people, animals or robots. In the case of the SmartTrack Motion Capture system, markers are attached to the intersted object and the data is captured by performing a range of movements in front of the SmartTrack. SmartTrack consist of Two tracking cameras and controller where are fully integrated in one housing. SmartTrack

can be pre-calibrated and may be used immediately. This and its compact size make the SmartTrack the ideal mobile, out-of-the-box tracking system.

Characteristics	Frame rate	IR flash	Max. tracking distance
	max. 60fps	850nm	2.5m
	Nominal voltage	Maximal current	Maximal power
	5V DC	4A	20W

Table A.8: SmartTrack characteristics

A Matlab function was built to have access to the observed values gived by the SmartTrack camera (Tracking module) in real time, and therefore a possibility to determine the speed.

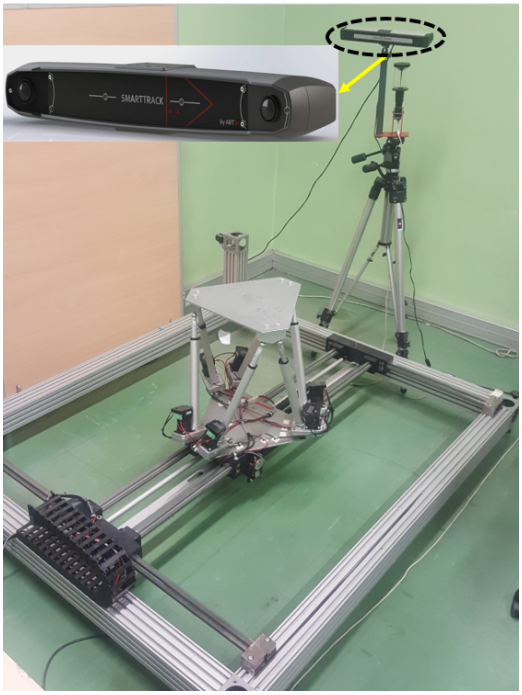


Figure A.7: SMARTTRACK camera

For this system to work at best it was necessary first to perform a physical calibration using tools provided with the SmartTrack camera but also to create a software calibration algorithm to rectify the slight offsets resulting from more or less significant displacements of the structure SmartTrack voluntary or not. This software calibration consists in making successive displacements of one axis at a time in order to check the existing offsets on the recovery of the positions of the axis that has remained motionless. We then apply an algorithm that corrects this difference and apply it to all the values of this axis. We perform this same task on the next axis and are then able to recover the exact values within 0.1mm.

A.4 Dynamic modeling validation

In this section, we present the actuator force behavior for theoratical modeling through simulink with the simMechanics.

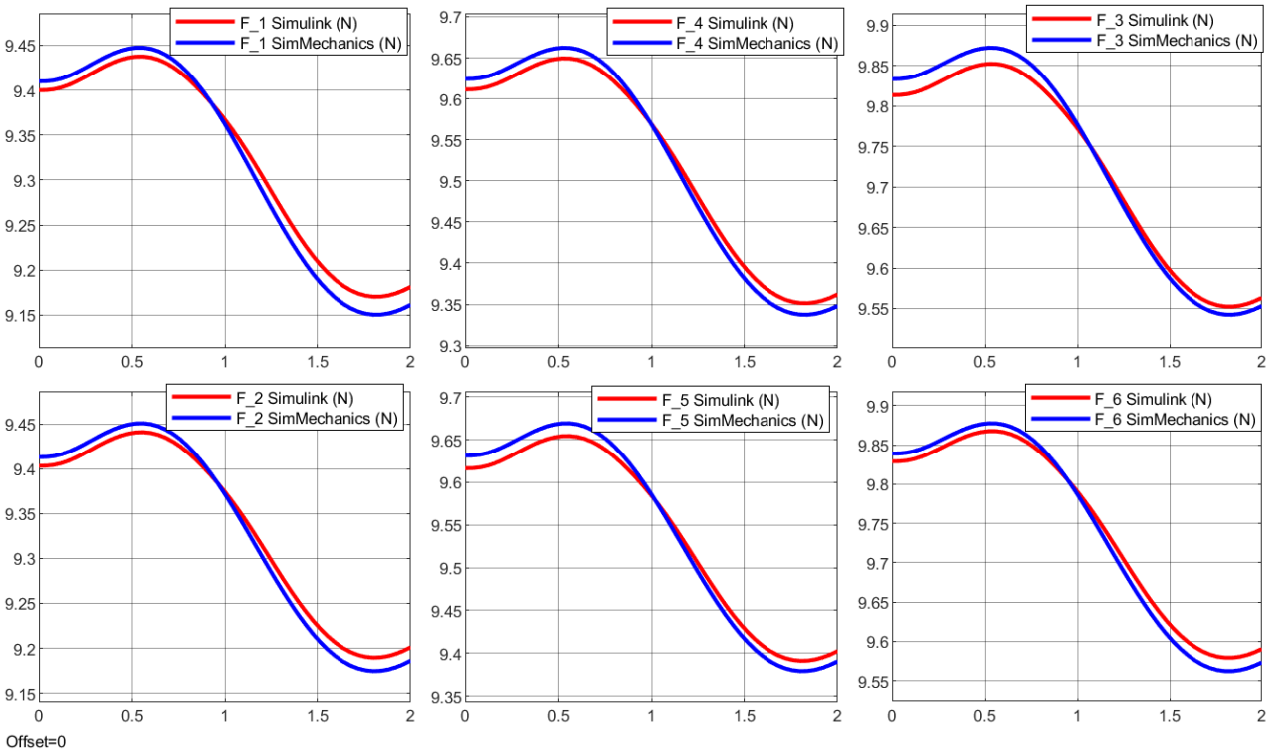


Figure A.8: Forces comparaison for GS movement along Z axis

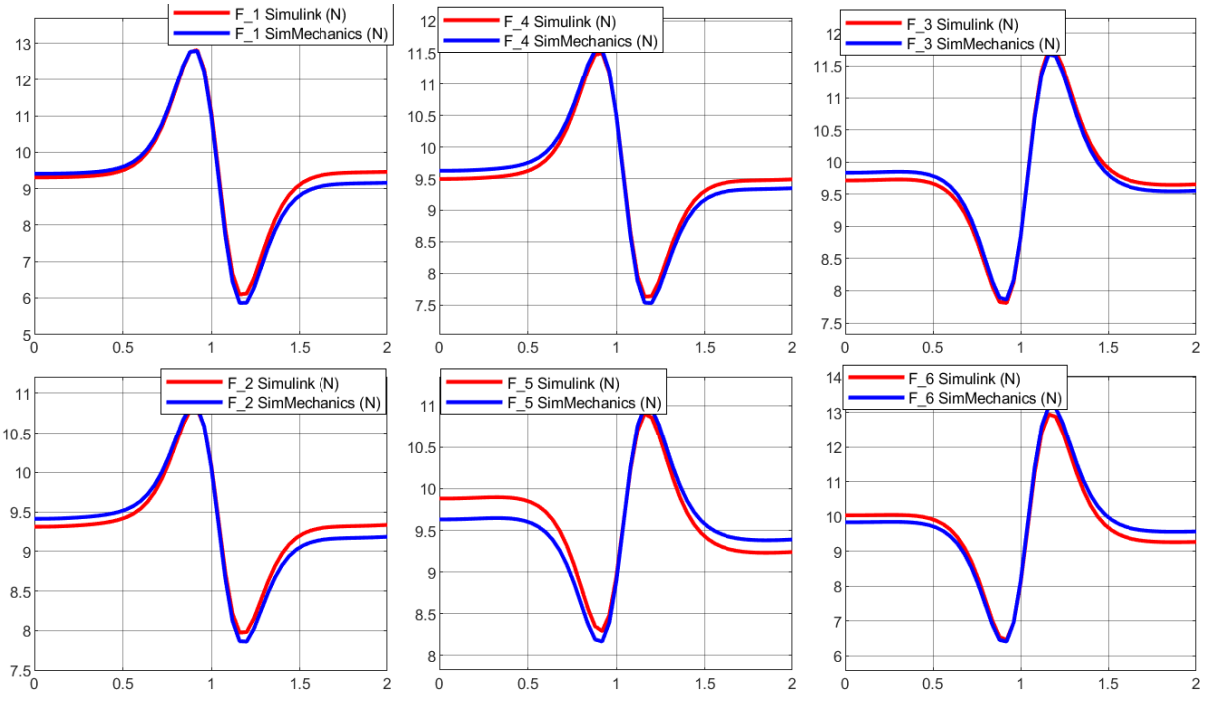


Figure A.9: Forces comparaison for GS movement along Z axis, Table movement along X axis

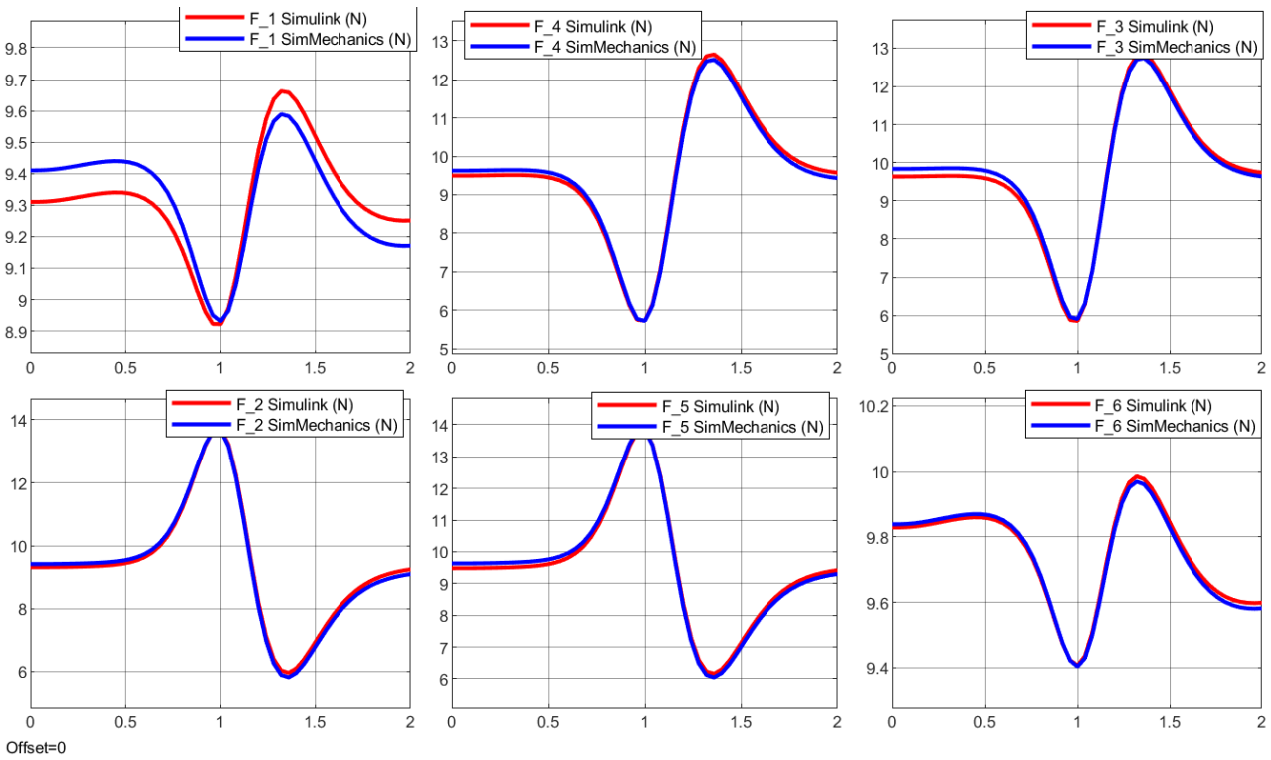


Figure A.10: Forces comparaison for GS movement along Z axis, Table movement along Y axis

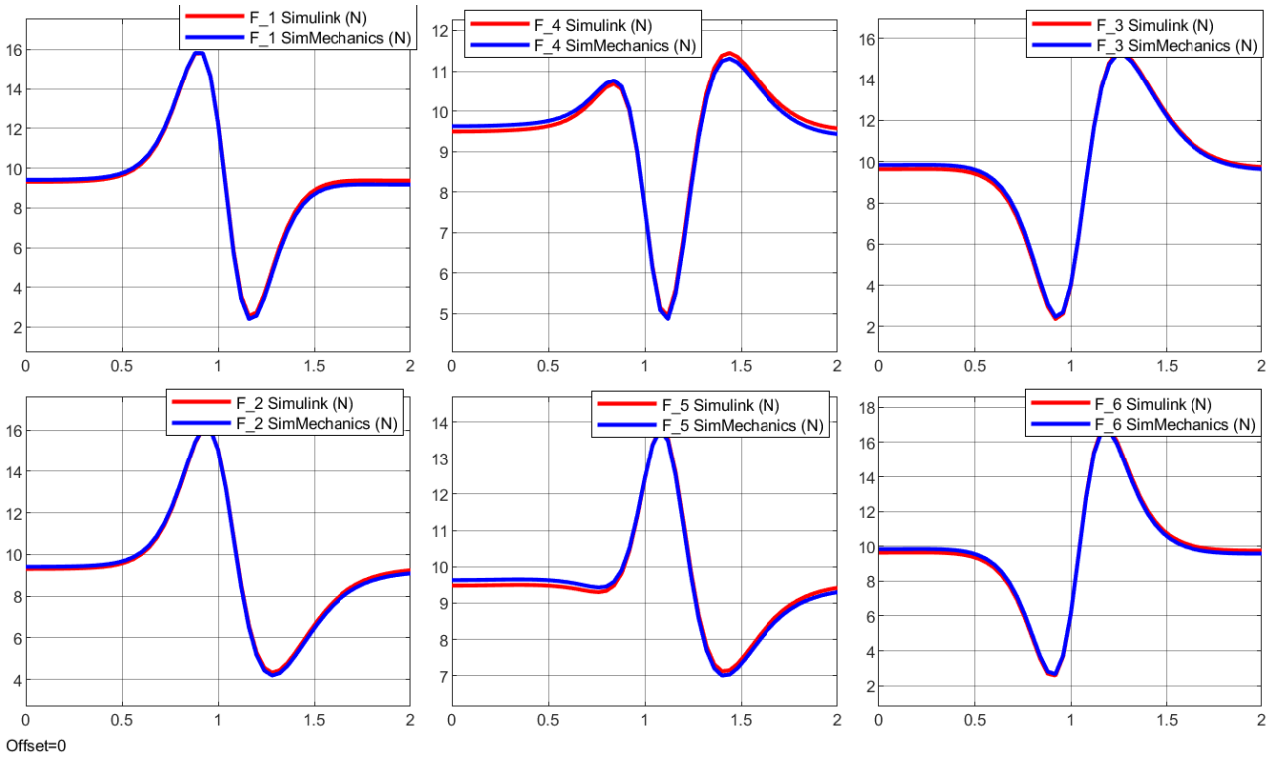


Figure A.11: Forces comparaison for GS movement along Z axis, Table movement along XY axis

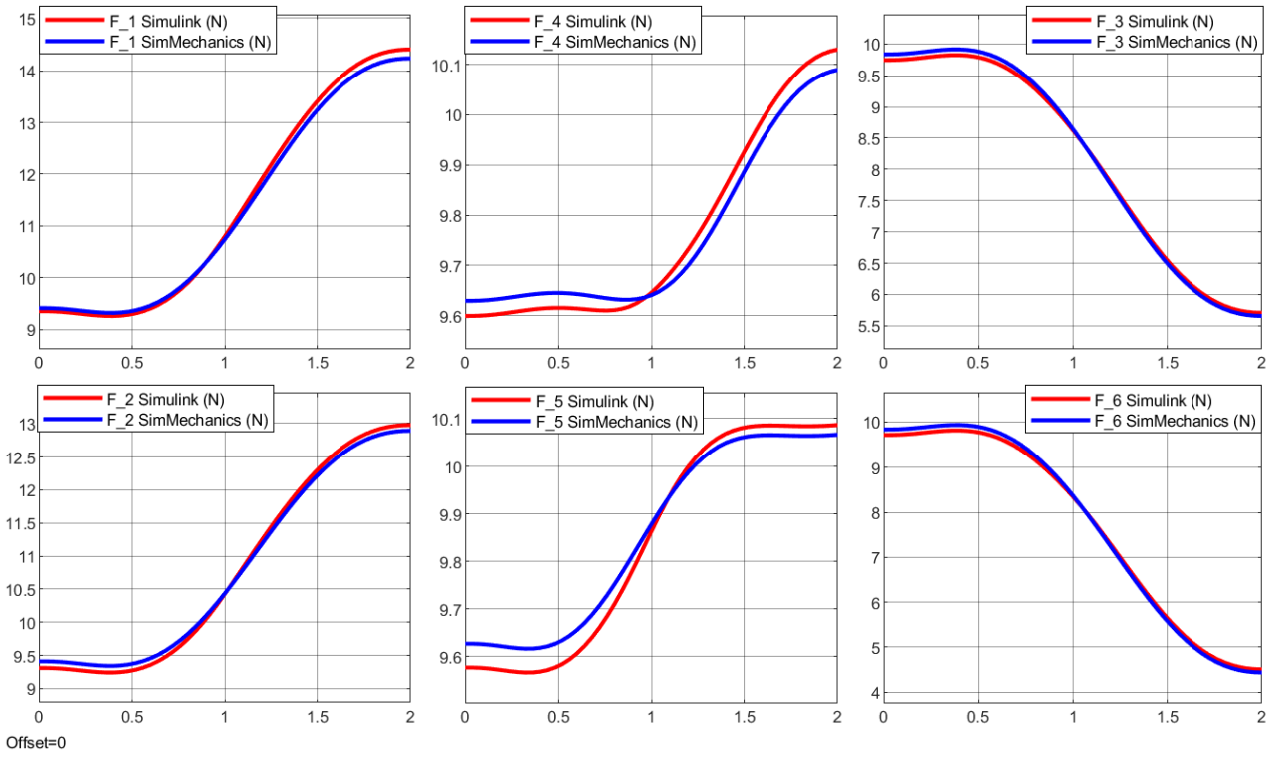


Figure A.12: Forces comparaison for GS movement along X axis

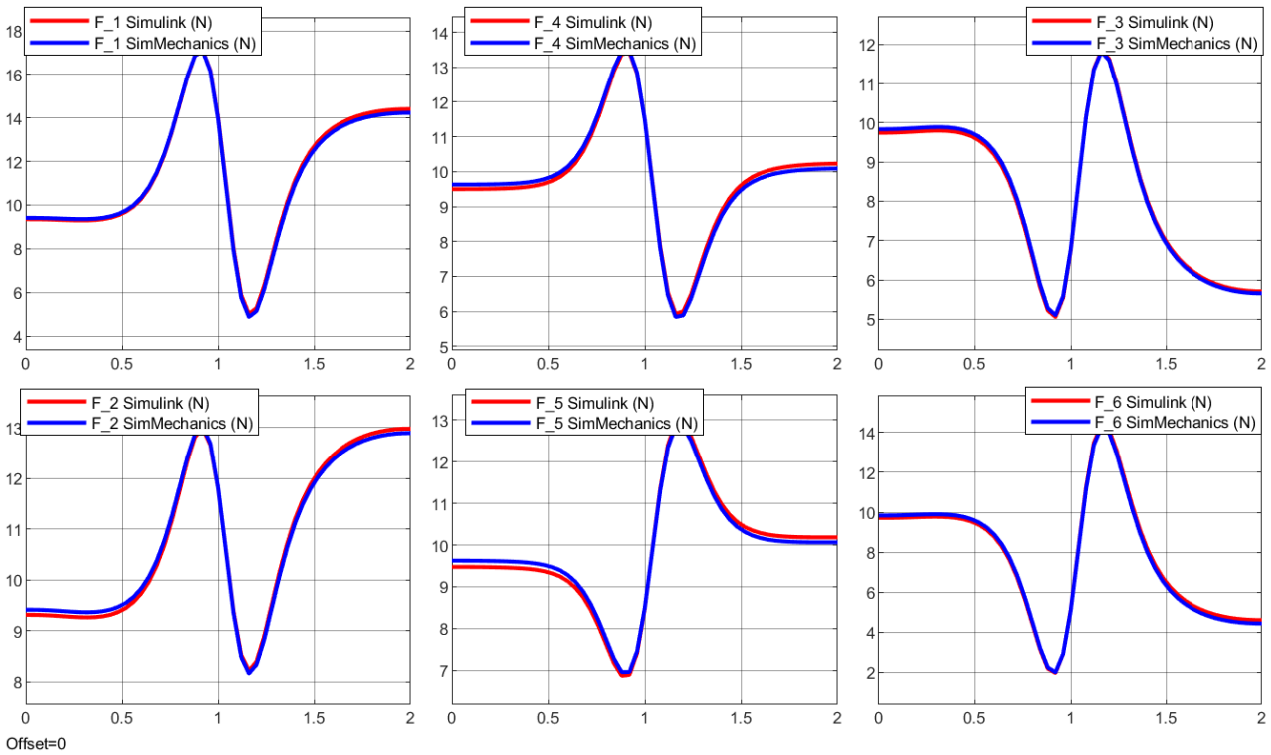


Figure A.13: Forces comparaison for GS movement along X axis, Table movement along X axis

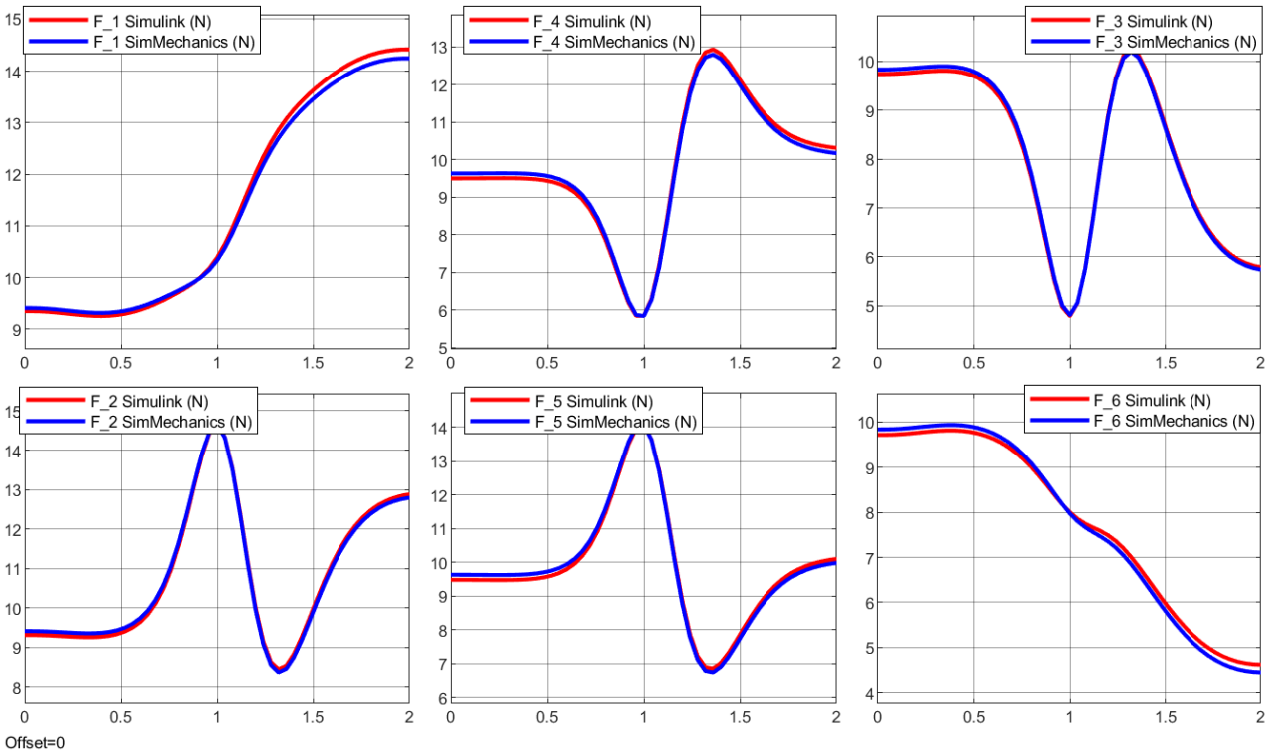


Figure A.14: Forces comparaison for GS movement along X axis, Table movement along Y axis

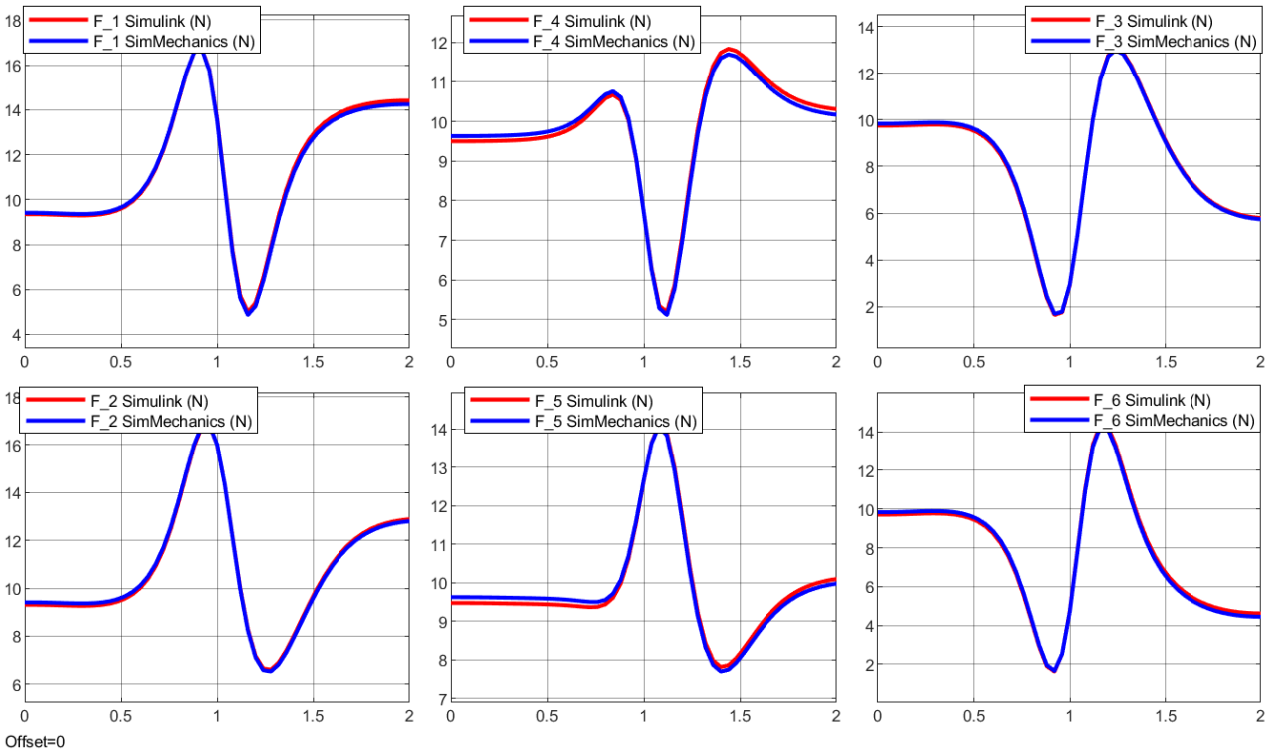


Figure A.15: Forces comparaison for GS movement along X axis, Table movement along XY axis

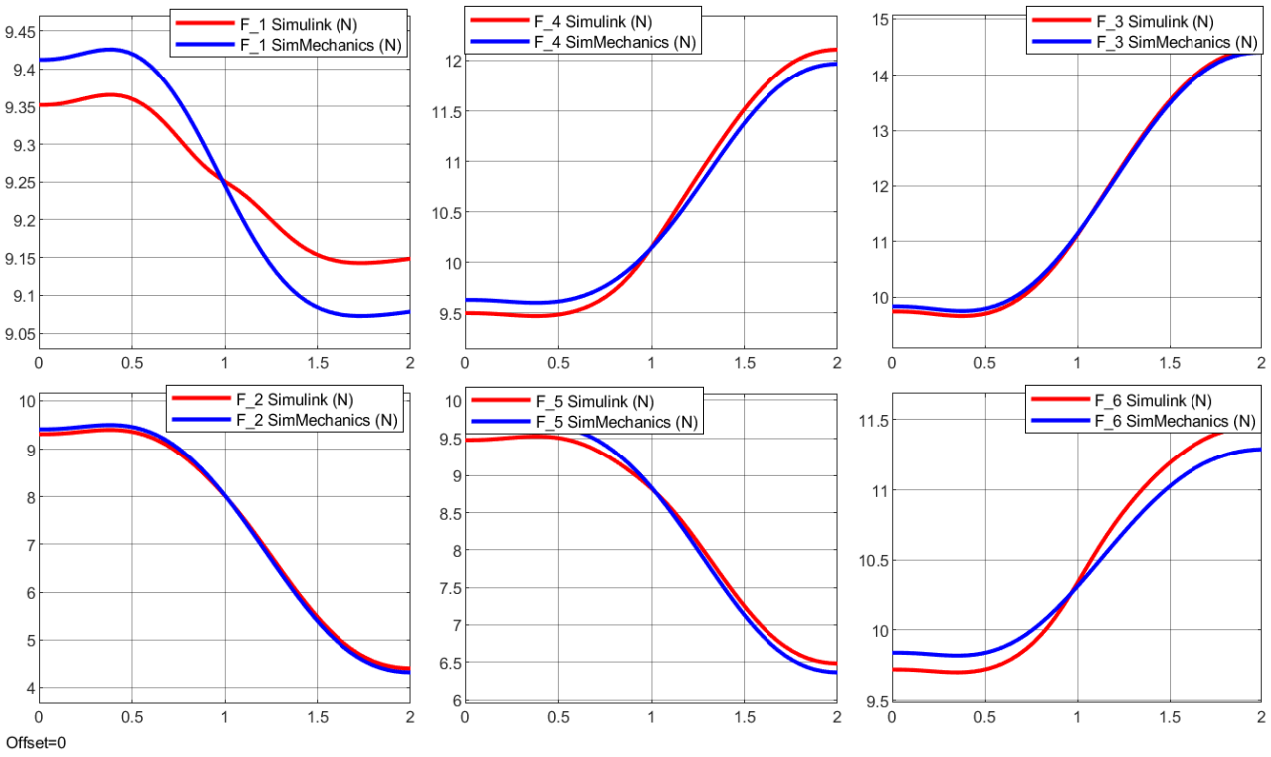


Figure A.16: Forces comparaison for GS movement along Y axis

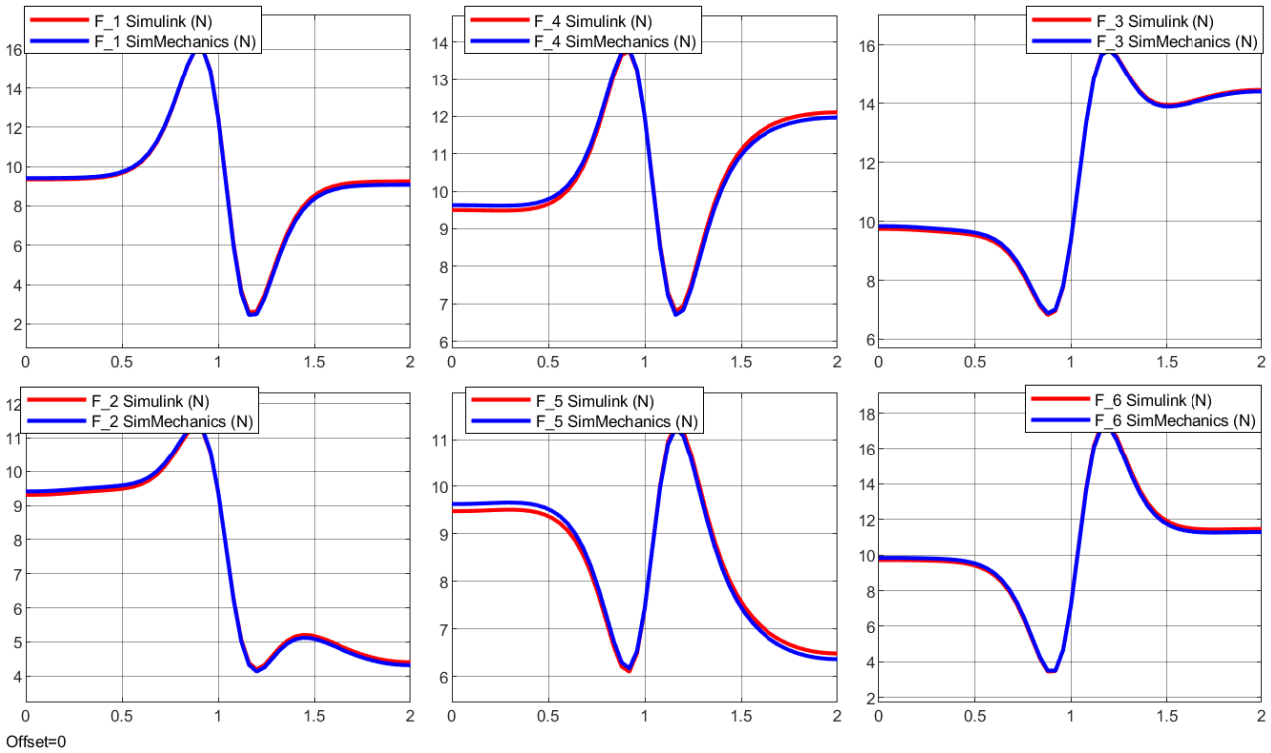


Figure A.17: Forces comparaison for GS movement along Y axis, Table movement along X axis

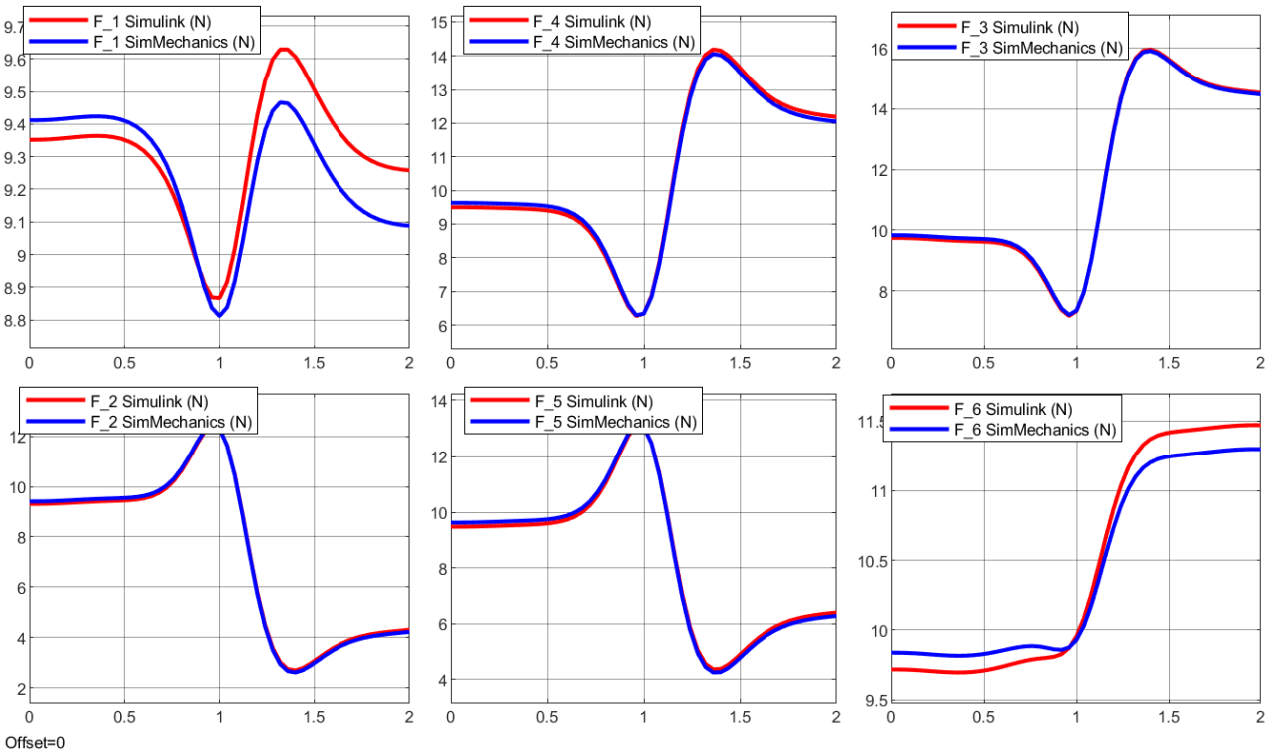


Figure A.18: Forces comparaison for GS movement along Y axis, Table movement along Y axis

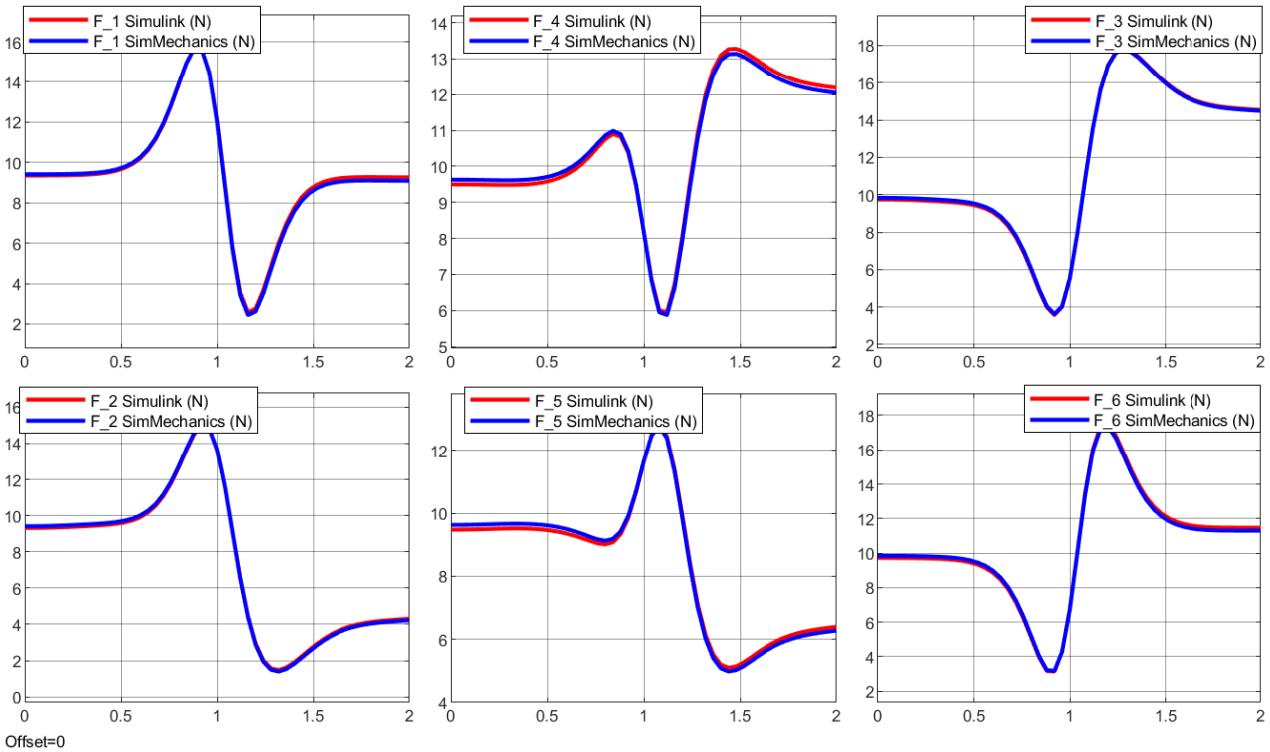


Figure A.19: Forces comparaison for GS movement along Y axis, Table movement along XY axis

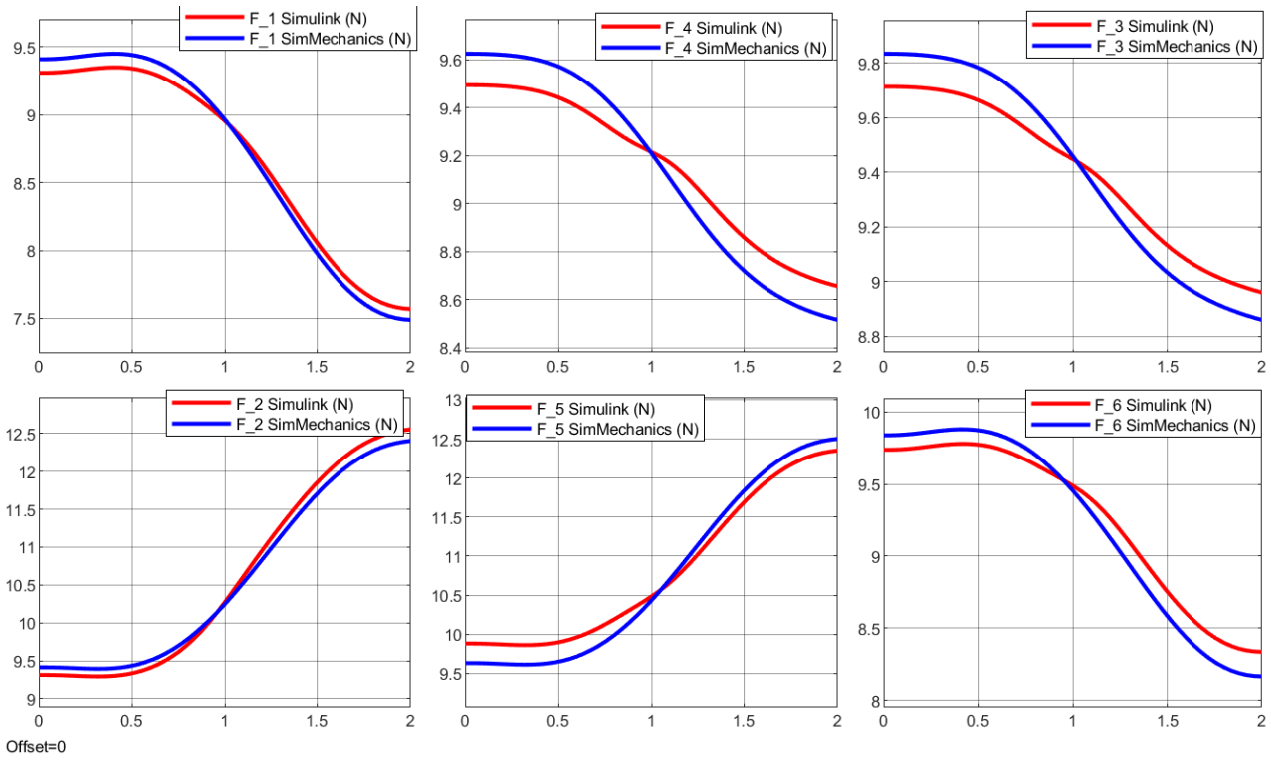


Figure A.20: Forces comparaison for GS rotation movement along X axis

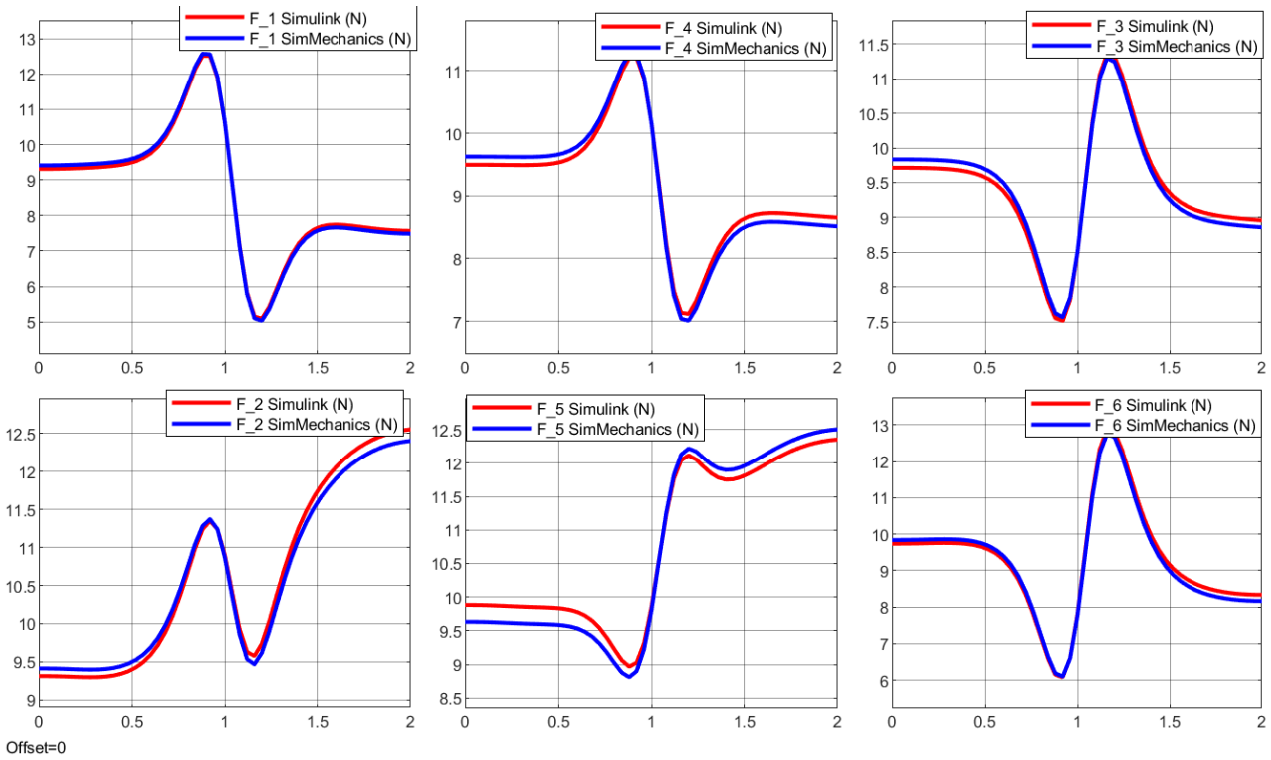


Figure A.21: Forces comparaison for GS rotation movement along X axis, Table movement along X axis

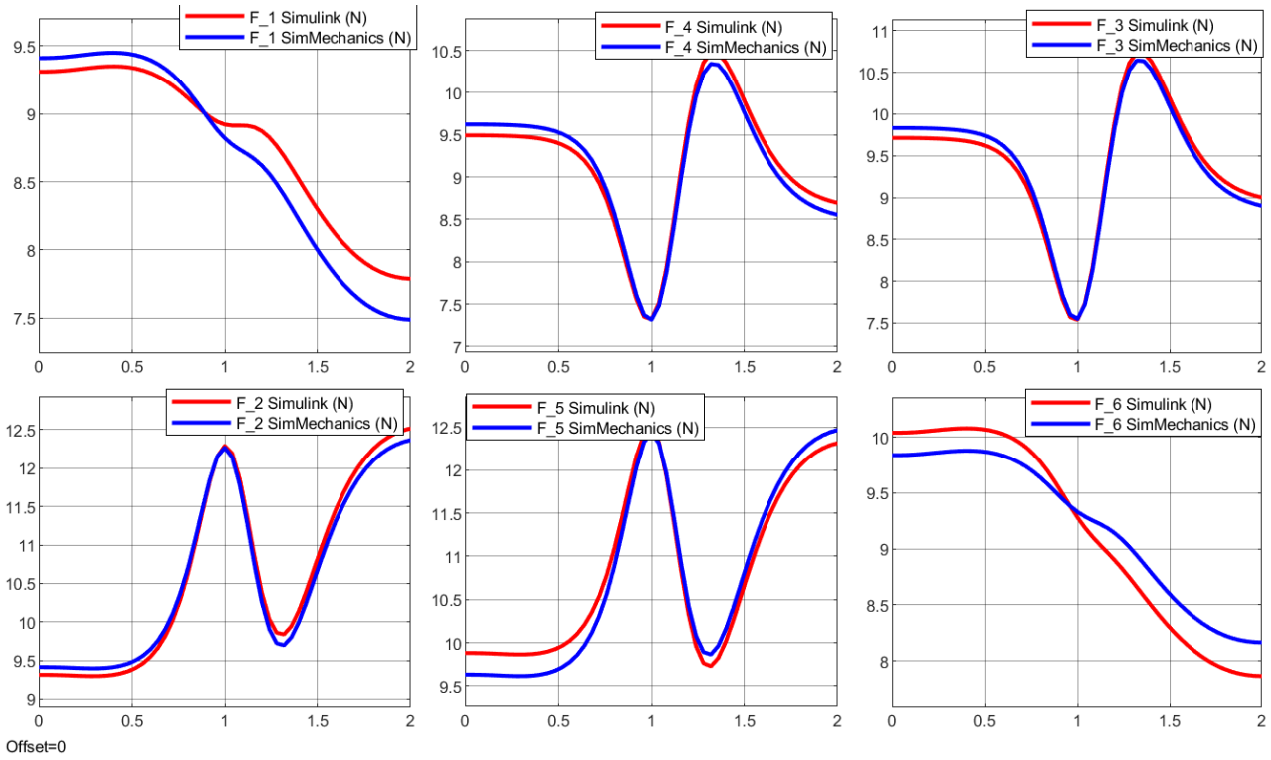


Figure A.22: Forces comparaison for GS rotation movement along X axis, Table movement along Y axis

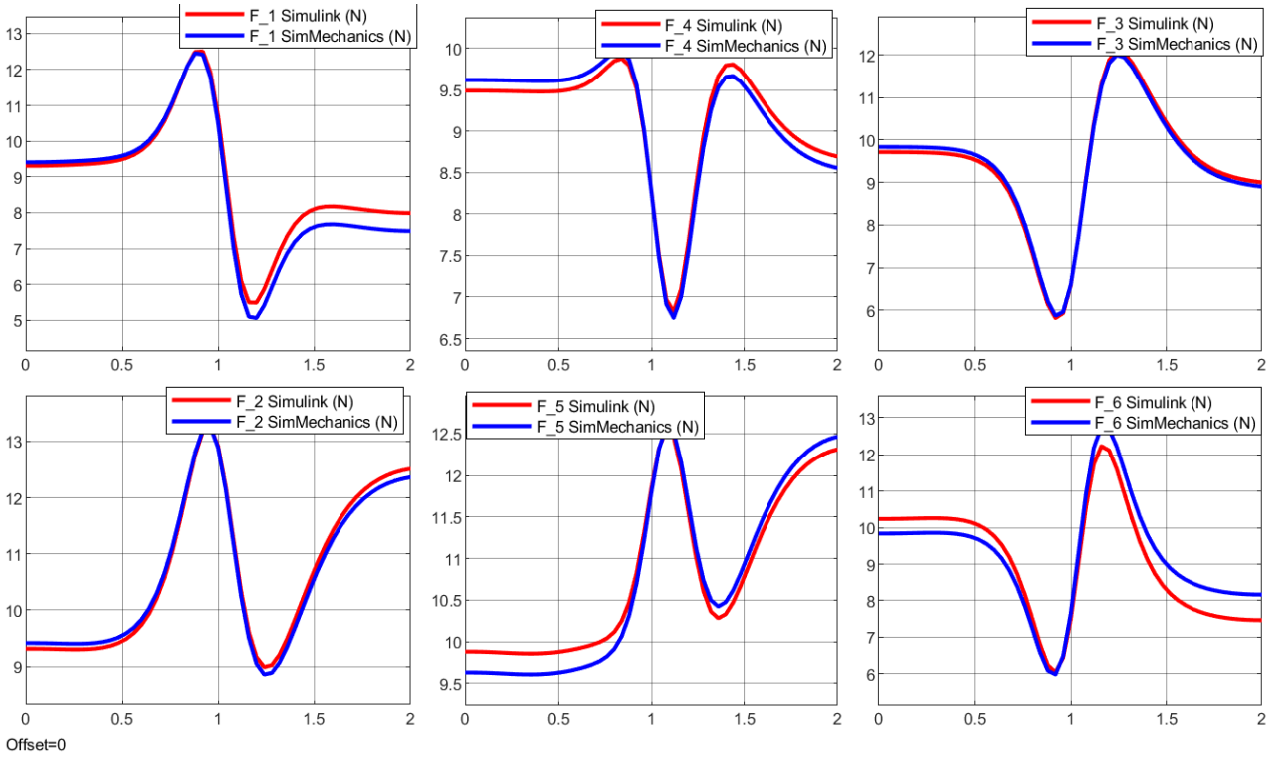


Figure A.23: Forces comparaison for GS rotation movement along X axis, Table movement along XY axis