École Centrale de Nantes **University of Genova**

MASTER ERASMUS MUNDUS

EMARO "EUROPEAN MASTER IN ADVANCED ROBOTICS"

2014/ 2016

Master Thesis Report

Presented by

Rafael Balderas Hill

On 26 / 08 / 2016

Title
Automatic Dynamic Reconfigurability of Parallel Robot by Crossing Type 2 Singularities

Jury

President:	**Olivier KERMORGANT**	**Professor at ECN-IRCCYN**
Evaluators:	**Sébastien BRIOT**	**CNRS researcher at IRCCYN**
	Abdelhamid CHRIETTE	**Professor at ECN-IRCCYN**
	Philippe MARTINET	**Professor at ECN-IRCCYN**
	Damien SIX	**PhD student at ECN-IRCCYN**
	Stéphane CARO	**CNRS researcher at IRCCYN**

Supervisor(s):

BRIOT Sébastien (CNRS-IRCCYN)
SIX Damien (ECN-IRCCYN)
CHRIETTE Abdelhamid (ECN-IRCCYN)
MARTINET Philippe (ECN-IRCCYN)
ZLATANOV Dimiter (UNIGE)

Abstract

In the production robotics field, it is well known that parallel robots present several operational advantages compared to the serial robots, such as higher stiffness, higher payload, small inertia and higher speed and acceleration. However, due to their geometric complexity in the kinematic chains, parallel robots present abundant singularity problems that affect their performance. The most constraining singularities are the Type 2 singularities due to the fact that in such singular configurations the workspace is divided into different aspects corresponding to different assembly modes, and this results in decreasing the reachable workspace size. Moreover, the parallel mechanism gains one or more uncontrollable degrees of freedom, which makes the robot present an underactuated behavior at singularity. In order to overcome the crossing through these type of singularities, an advanced control law based on virtual holonomic constraints is proposed.

The present research work explains and gives the rigorous mathematical derivations for synthesizing a feedback control law based on virtual holonomic constraints that takes into account the analysis of the degeneracy conditions of the dynamic model, the dynamic criterion to cross these singularities and the controlled and uncontrolled variables identified at singularity. This to define a control architecture that deals with the continuous tracking of non-optimal trajectories both in the singularity locus and far from the singularity. The case of study and experimentation is the five-bar mechanism Dual-arm SCARA robot DexTAR (*Dextrous Twin-Arm Robot*).

Chapter 1 presents an introduction and state of the art in Type 2 singularity techniques in the literature and also the dynamic modeling of parallel robots by recalling the geometric model and the first and second order kinematics. Moreover, a validation of the dynamic model is performed via Co-Simulation between ADAMS and MATLAB.

In addition to that, the degeneracy conditions of the Dynamic model of the five-bar mechanism are analyzed. This analysis will be used for the computation of the free dynamics and the modeling of the virtual holonomic constraints. In addition to that, an overview of the crossing Type 2 singularity techniques is presented, this to ensure a

complete analysis on the conceptual and technical context of this problem.

In chapter 2 and chapter 3, a novel approach based on virtual holonomic constraints, for addressing the underactuation of parallel robots at singularity is developed, including an analysis of the computation for the free dynamics, virtual holonomic constraints modeling, dynamic model consistency at singularity and an input-output linearization for the enforcement of the virtual holonomic constraints. Two different solutions for dealing with the underactuation at singularity are proposed, the first one by modeling the virtual holonomic constraints using joint space controlled variables, and a second one by modeling the virtual holonomic constraints using cartesian space controlled variables.

Finally, in chapter 4, a multi-controller architecture is proposed for a continuous trajectory tracking including the parallel mechanism undergoing far from the singularity locus and in the singularity locus, which is the critical position when the robot becomes underactuated. Moreover, in order to define the transition between control laws, a performance-based supervisory block is added in the scheme in order to define the switching between control laws. Then, the results of testing the multi-controller architecture with non-optimal trajectories are presented and discussed. In addition to that, as a complementary validation process, an Inertial Measurement Unit *IMU* is used to validate the Type 2 singularity crossing.

Acknowledgements

Before starting this research work, I would like to express some words to the people that have guided me during my master studies and master thesis.

Firstly, I would like to thank my research supervisors, Dr. Sébastien Briot, Ph.D. student Damien Six, Dr. Abdelhamid Chriette and Dr. Philippe Martinet, for the time they have shared with me, their patience and their advise provided for the development of this research work. Their combined expertise and experience have been extremely appreciated and valuable for the fluent development of this research work. Moreover, I am extremely grateful for the balance they have put between guiding me and encouraging me to be autonomous during the research.

I would like to thank to the EMARO programme for their valuable research skills they have provided me. I have had the opportunity to study my first year in the University of Genova and my second year in the Ecole Centrale de Nantes. In both universities I have found the opportunity to interact within a cooperative environment, where I have been able to exchange technical and cultural ideas.

I would like to express a special thank to the Mexican Council of Science and Technology (CONACyT) for sponsoring my Master Studies, their support was a very important part during these two years of Master Studies.

Finally, I would like to express my tremendous gratitude and love to my family, for their never ending support.

Contents

List of Figures

Nomenclature

n	degrees of freedom
n_d	passive joints present in the legs of the robot
$\mathbf{q}_a$	n dimensional vector of active joints
$\dot{\mathbf{q}}_a$	n dimensional vector of active joint velocities
$\mathbf{q}_d$	n_d dimensional vector of active joints
$\dot{\mathbf{q}}_d$	n_d dimensional vector of active joint velocities
$\mathbf{x}$	vector representing the robot pose
$\dot{\mathbf{x}}$	velocity of the platform
$\ddot{\mathbf{x}}$	acceleration of the platform
$\mathbf{t}$	robot twist with parametrization of the end effector orientations
$\mathbf{A}$	$(n \times n)$ parallel jacobian matrix
$\mathbf{B}$	$(n \times n)$ serial jacobian matrix
i	an integer
$\boldsymbol{\tau}$	the $(n \times 1)$ vector of the robot input efforts
$\boldsymbol{\lambda}_1$, $\boldsymbol{\lambda}_2$	the $(n \times 1)$ and $(n_d \times 1)$ vector of Lagrange multipliers
$\mathbf{J}_{ta}$, $\mathbf{J}_t$, $\mathbf{J}_{td}$	$(n_d \times n)$, $(n_d \times n)$ and $(n_d \times n_d)$ relating $\dot{\mathbf{q}}_d$, $\dot{\mathbf{q}}_a$ and ${}^0\mathbf{t}_p$
L_t	Lagrange of the virtual tree structure
$\boldsymbol{\tau}_{ta}$	$(n \times 1)$ vector of virtual input efforts in active joints related to the Lagrange of virtual tree structure L_t
L_p	Lagrange of the moving platform
$\mathbf{w}_p$	$(n \times 1)$ vector of the wrench of the free platform related to L_p
$f_{off1}, f_{off2}, f_{s11}, f_{s21}$	friction terms for the five bar mechanism
$\boldsymbol{\tau}_{td}$	$(n_d \times 1)$ vector of virtual input efforts in passive joints related to the Lagrange of the virtual tree structure L_t
$\mathbf{J}$	Jacobian matrix relating $\mathbf{t}$ and $\dot{\mathbf{q}}_a$
$\mathbf{J}_d$	Jacobian matrix relating $\dot{\mathbf{q}}_d$ and $\dot{\mathbf{q}}_a$
$\mathbf{q}$	vector of active joints for five bar mechanism
$\dot{\mathbf{q}}$	vector of active joint velocities for five bar mechanism

a	distance between actuated joints of the five bar mechanism
l	link lengths of the five bar mechanism
zz_{11R}, zz_{21R}	inertias of the two proximal links of the five bar mechanism
m	mass of the end effector of the five bar mechanism
$\mathbf{ZZ}$	positive diagonal matrix of inertias
$\mathbf{f}$	$(n \times 1)$ vector that groups active joint friction terms
$\mathbf{M}$	inertia matrix
$\mathbf{H}$	matrix that groups the gravitational, centrifugal and Coriolis terms
$\mathbf{u}$	n dimensional vector of control inputs
$\mathbf{q}^d$	n dimensional vector of desired joint position
$\dot{\mathbf{q}}_d$	n dimensional vector of desired joint velocity
$\ddot{\mathbf{q}}_d$	n dimensional vector of desired joint acceleration
x, y	moving platform pose
$\dot{x}$, $\dot{y}$	moving platform velocities
$\ddot{x}$, $\ddot{y}$	moving platform accelerations
$\mathbf{t}_s$	twist of uncontrollable motion at singularity
h_i	virtual holonomic constraint for both representations (joint and task space)
$\phi_i(y)$	desired evolutions of the controlled variables
$\mathbf{A}(y)$	$(n \times n)$ parallel jacobian matrix in y coordinates
$\mathbf{B}(y)$	$(n \times n)$ serial jacobian matrix in y coordinates
$\mathbf{A}_s$	n dimensional vector of reduced parallel jacobian matrix at singularity
$\boldsymbol{v}_1$	n dimensional vector of virtual constraint controller
K_p	proportional gain
K_d	derivative gain
$\mathbf{h}$	n dimensional vector of virtual holonomic constraints
$\boldsymbol{v}_2$	escalar control input for virtual constraint with cartesian controlled variable
β	end-effector orientaton
$\mathbf{u}_1$	n dimensional vector of control inputs for Computed Torque Control
$\mathbf{u}_2$	$(n \times 1)$ control inputs with virtual holonomic constraint in joint space
$\mathbf{u}_3$	$(n \times 1)$ control inputs with virtual holonomic constraint in cartesian space
σ	scalar supervisory control variable

Context

During the last decades, the robotic research communities on studying dynamic reconfigurability of parallel mechanisms have kept special attention on addressing the singularity problems on such complex mechanical architectures in order to come up with novel solutions for undergoing in the singularity locus of the parallel robots, describing a stable behavior. Moreover, researchers have been attracted for parallel mechanisms due to their operational advantages in processes, such as machine-tools processes, medical applications, pick and place processes, etc.

Nevertheless, parallel robots present abundant singularity problems that influence their performances in terms of reachable workspace, which is reduced due to these problems. Therefore, the workspace size is reduced, and the accessibility performances of the parallel robots are affected.

The literature review shows that some designers have overcome the problems of workspace reachability by implementing more complex mechanisms, but this usually leads to the design of robots with small workspace or parallel robots with low practicability. On the other hand, in the last years it has been shown that it is possible to pass through Type 2 singularities by planning a trajectory respecting a physical criterion developed from the analysis of the degeneracy conditions of the dynamic model. This last approach is promising since it has shown to cross the Type 2 singularities and increase the reachable workspace. However, due to the dynamic model degeneracy at the singularity locus, just by planning a trajectory respecting the criterion is not enough. Thus, a last solution has proposed the implementation of a Multi-Model Controller in order to track the trajectory that respects the physical crtierion, and by swithing between two different dynamic models. Nevertheless, this solution neglects the platform dynamics in the feedback linearization, and this may not succeed in crossing and generate an overconstraint in the system at the singularity.

The results presented in this thesis show that it is possible to develop an advanced feedback control law in order to cross Type 2 singularities. This is done, by modeling virtual holonomic constraints to take into account the underactuation that appears in the singularity locus. This virtual constraint can be modeled based on on the analysis of the free dynamics of the robot at singularity and be enforced through feedback control action. Moreover, it is shown that the virtual holonomic constraint technique is a novel approach for crossing Type 2 singularities in parallel robots.

Contributions

The main contributions of the present research work are listed as follows:

- The modeling of the free dynamics and its equivalence with the dynamic criterion that must be respected in order to cross the Type 2 singularities. This leads to the identification of the controlled and uncontrolled variables in order to develop the virtual holonomic constraint and its controller at the singularity when the robot is underactuated;

- Definition of the virtual holonomic constraint in the two different spaces, by using controlled joint space variables and controlled cartesian space variables;

- Synthesize of a virtual constraint controller at the singularity for the enforcement of the virtual holonomic constraint expressed with joint space and cartesian space variabes. In addition to that, the feedback linearization of the system is computed demonstrating consistency in the dynamic model;

- Development of a Multi-Controller architecture for a continuous tracking of non-optimal trajectories with stable behavior undergoing far from the singularity and in the singularity locus.

Chapter 1

Introduction

This chapter is dedicated to explain the definition and evolution of the parallel mechanisms. First, by giving a brief historical overview, and then by stating the advantages and drawbacks over the serial robots. Afterwards, it is presented a performance analysis that involves changing assembly modes, which implies also to generally talk about the most constraining drawbacks, which are the singularities. In the second section the dynamic modeling of parallel robots will be addressed by computing a general dynamic modeling formulation. This chapter seeks also to analyze the Degeneracy conditions of the Dynamic model due to the parallel jacobian matrix $\mathbf{A}$*. Then an overview of crossing Type 2 singularities techniques will be given by analyzing the technical and conceptual problems when meeting Type 2 singularities. Two main approaches that have been proposed in the literature will be discussed. First, a physical criterion deduced from the Degeneracy conditions that must be respected while crossing Type 2 singularities. And then, a Multi-Model Computed Torque Control for tracking optimal trajectories respecting the criterion of the first approach. Then the dynamics of the five-bar mechanism are modeled and the identified dynamic parametrization is included and a cross-validation of the dynamic model is performed via a Co-simulation ADAMS/MATLAB. Finally a summary of this chapter will be given by pointing out the main details that will be used for further analyses of the present research work.*

1.1. Parallel robots

1.1.1 History of parallel robots

Parallel robots are robots whose base and end-effector are connected by several closed kinematic chains in which not all joints are actuated. In [1], a fully parallel robot is defined as a mechanism that includes as many elementary kinematic chains as the moving platform does admit degrees of freedom. In addition, every elementary kinematic chain possesses only one actuated joint. Moreover, no segment of an elementary kinematic chain can be linked to more than two bodies. Parallel robots can be classified based on the type of motion that their end-effector can produce, for example:

- 2 translations: 2T
- 2 translations and 1 rotation in the plane: 2T1R
- 3 translations: 3T
- 3 translations and 1 rotation (Shönflies motion): 3T1R
- 3 translations and 2 rotations: 3T2R
- 3 translations and 3 rotations: 3T3R
- 2 or 3 rotations (spherical robots): 2R or 3R
- 2 rotations and 1 translation: 2R1T

Some examples are depicted in Figure 1.1. The robot in the left is the Gough-Stewart platform which is fully-parallel robot, whose base and end-effector are connected by identical serial chains with only one actuated joint in each chain. This parallel robot was the first development in the industrial environment by the Dr. Eric Gough in 1962. The robot in the right is a Delta robot classified as 3T1R, which means it can describe Shönflies motions, it was invented by Clavel in 1986.

There are also planar 2T parallel robots with fixed or mobile actuators and with revolute or linear actuated joints. An example of a fixed linear actuated joint PRP-PR architecture is depicted in the Figure 1.2. And also an example of fixed revolute actuated joints architectures: RRR-RR (five-bar mechanism), Figure 1.3.

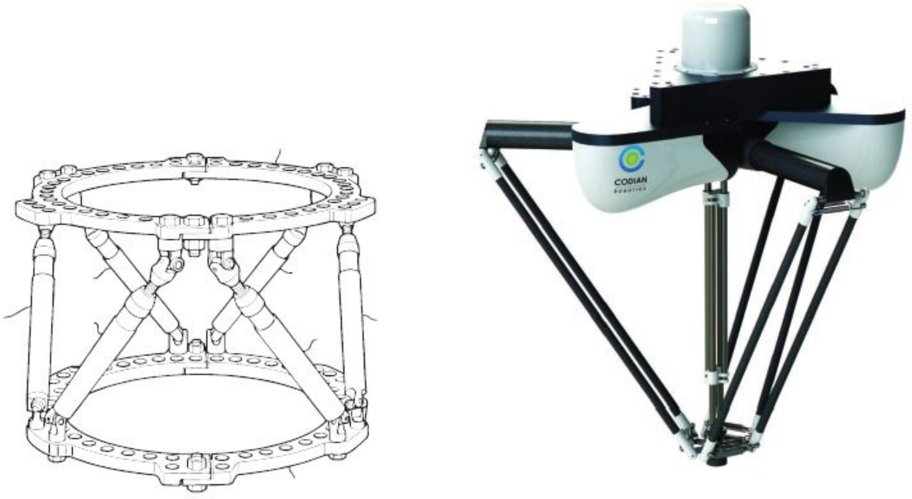

Figure 1.1: Gough-Stewart platform and Robot Delta[2]

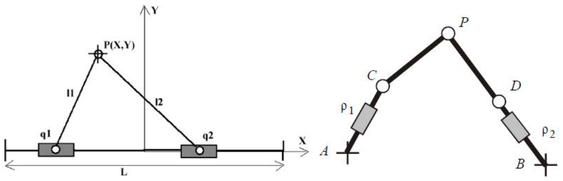

Figure 1.2: PRR-PR mechanism architecture [4]

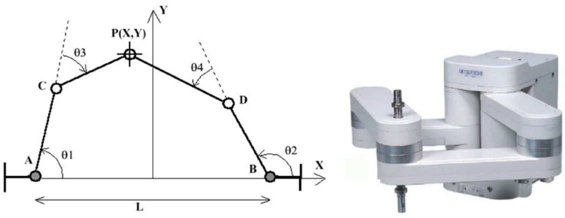

Figure 1.3: RRR-RR (five-bar mechanism) [4]

1.1.2 Advantages and drawbacks

There exist several advantages of parallel robots with respect to the serial robots, the main advantages are listed as follows:

- High stiffness
- High payload
- Small inertia
- High speed and acceleration

The main drawbacks are:

- Small workspace

- Abundant singularity problems

The main drawbacks of the parallel structures are related to the reachable workspace, which is a primary operation condition which must be taken into account for a good performance of the robot. If the robot can have more reachable points in the entire workspace, then the operational time to perform a task can be reduced. However, in order to address the problem of reachability, it is necessary to analyze the second main drawback which is the singularity problems of the complex dynamics of a parallel robot.

1.1.3 Changing assembly mode for parallel robots

Parallel robots have multiple solutions from the direct kinematic model. It means that the parallel robot can perform several configurations, which allow it to admit several positions and orientations in the workspace for a given set of joint values.

Moreover, parallel robots have multiple inverse kinematic solutions. This means, that the robot can admit several joint values to one given configuration of the end effector [1][3]. Based on the previous notion of direct and inverse kinematic solutions, two performance concepts of parallel robots are defined. A posture changing trajectory is equivalent to a trajectory between two inverse kinematic solutions. Similarly, an assembly mode changing trajectory is equivalent to a trajectory between two direct kinematic solutions [2][4]. In figure 1.4, it is possible to observe the two possible direct kinematic solutions for a five-bar mechanism $P(X, Y)$ and $P'(X, Y)$, which depict the assembly modes of the mechanism.

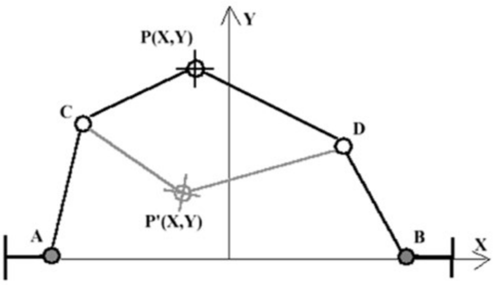

Figure 1.4: Assembly modes for a five-bar mechanism [4]

1.1.4 Singularities of parallel robots

Singularities in parallel robots can be classified into three different types of singularities based on the analysis of the kinematic models, this classification is proposed by Grosselin and Angeles [5]:

Type 1 singularities: Also known as serial singularities occur when the robot loses the ability to move in one given direction. It means that in such configuration there is a direction in which no task space velocity can be performed.

Type 2 singularities: Also known as parallel singularities occur when the parallel mechanism loses its ability to change from assembly mode. It means that in such singular configuration, the robot cannot move from one configuration to another. These singular configurations are located inside the workspace, which makes the workspace be divided into different aspects. From Figure 1.5, it is observed that the parallel mechanism is in a singular configuration, and the robot has an uncontrollable motion along $\mathbf{t}_s$, which is perpendicular to $\overrightarrow{B_1C}$ and $\overrightarrow{B_2C}$.

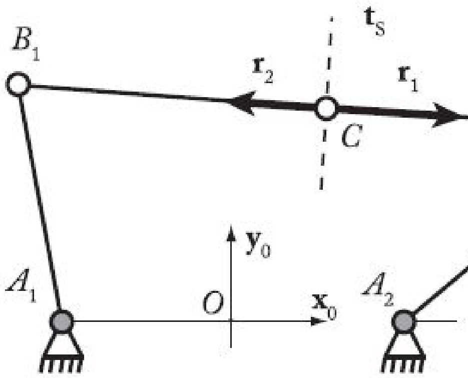

Figure 1.5: Parallel robot in a singular position [6]

Type 3 singularities: These singularities occur when both Type 1 and Type 2 singular postures appear at the same time. In such configurations, the robot loses locally the ability to perform a motion along one direction of the workspace and gains one or more uncontrollable motions along another direction.

1.2. Dynamic modeling of parallel robots

In order to compute the dynamic model, the methodology presented in [7] [8] will be used. Let us assume a parallel robot composed of m links, n degrees of freedom driven by n actuators and n_d passive joints present in the legs of the robot. The position and velocity of the parallel robot can be described by the n dimensional vector of active joint variables $\mathbf{q}_a = [q_1, q_2, ..., q_n]^T$, the n dimensional vector active joint velocities $\dot{\mathbf{q}}_a = [\dot{q}_1, \dot{q}_2, ..., \dot{q}_n]^T$, and two n_d vectors of passive joint variables and passive joint velocities. The cartesian space of the robot can be described by: $\mathbf{x} = [x, y, z, \phi, \psi, \theta]^T$ pose and $\mathbf{t} = [\dot{x}, \dot{y}, \dot{z}, \dot{\phi}, \dot{\psi}, \dot{\theta}]^T$ velocities of the platform. The first input-output kinematic constraint that relates the platform twist ${}^0\mathbf{t}_p$ (expressed in the base frame) with the active joint velocities $\dot{\mathbf{q}}_a$, can be found by the following loop closure equations of the parallel robot:

$$\mathbf{A}{}^0\mathbf{t}_p + \mathbf{B}\dot{\mathbf{q}}_a = \mathbf{0} \tag{1.1}$$

where:

$\mathbf{A}$ and $\mathbf{B}$ are the $(n \times n)$ parallel and serial Jacobian matrices, respectively.

Now, the computation of the passive joint velocities is necessary for obtaining the dynamic model. Let us consider the Figure (1.6), which is composed of m_i joints.

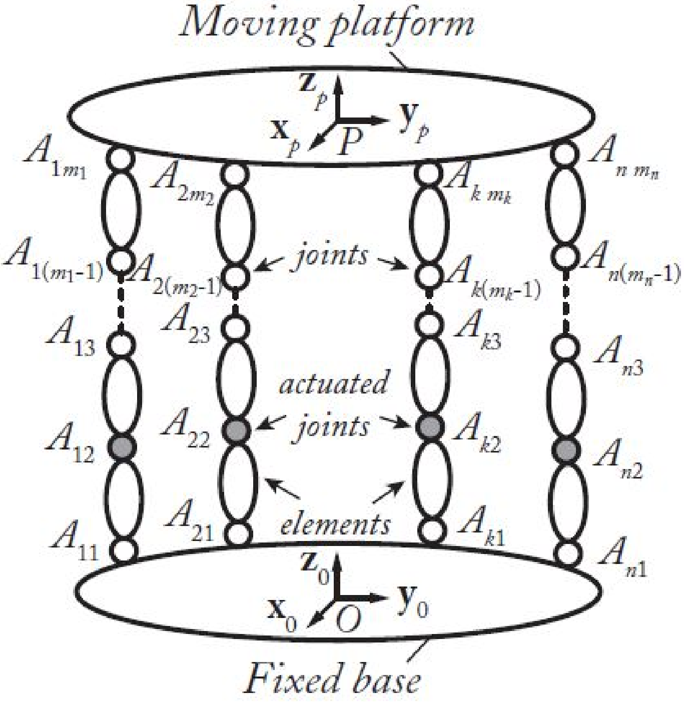

Figure 1.6: Moving platform, actuated joints denoted by color grey

By computing the twist of the platform at point A_{im_i}, it is possible to express the following equation:

$${}^0\mathbf{t}_p^i = {}^0\mathbf{t}_p + \begin{bmatrix} {}^0\boldsymbol{\omega}_p \times {}^0\mathbf{r}_{PA_{im_i}} \\ \mathbf{0} \end{bmatrix} \tag{1.2}$$

And by writing in its matrix form:

$$ {}^{0}\mathbf{t}_{p}^{i} = \begin{bmatrix} \mathbf{1}_3 & {}^{0}\mathbf{r}_{PA_{im_i}} \\ \mathbf{0} & \mathbf{1}_3 \end{bmatrix} {}^{0}\mathbf{t}_{p} = \mathbf{J}_{t_i} {}^{0}\mathbf{t}_{p} \tag{1.3} $$

As the joint located at A_{im_i} also belongs to the leg i, its twist can be expressed as:

$$ {}^{0}\mathbf{t}_{p}^{i} = ^{0} \mathbf{J}_{im_i} \dot{\mathbf{q}}_i = [{}^{0}\$_{im_i}^{i1} \dots {}^{0}\$_{im_i}^{im_i}] \dot{\mathbf{q}}_i \tag{1.4} $$

where $\dot{\mathbf{q}}_i$ represents all the joint velocities of the leg i, ${}^{0}\mathbf{J}_{im_i} = {}^{0}\mathbf{R}_{im_i} {}^{im_i}\mathbf{J}_{im_i}$ is the chain i kinematic Jacobian matrix of dimension $(6 \times m_i)$ and $\$_{im_i}^{i1}$ is the twist representing the displacement of the chain tip A_{im_i} when joint ik is only moving. Then the matrix ${}^{0}\mathbf{J}_{im_i}$ can be expressed as a group of sub-matrices ${}^{0}\mathbf{J}_{a_i}$ corresponding to the active joint velocities $\dot{\mathbf{q}}_{a_i}$, and the sub-matrix ${}^{0}\mathbf{J}_{d_i}$ corresponding to the passive joint velocities $\dot{\mathbf{q}}_{d_i}$. Thus (1.4), becomes:

$$ {}^{0}\mathbf{t}_{p}^{i} = {}^{0}\mathbf{J}_{im_i} \dot{\mathbf{q}}_i = [{}^{0}\mathbf{J}_{ai}, {}^{0}\mathbf{J}_{di}] \begin{bmatrix} \dot{\mathbf{q}}_{a_i} \\ \dot{\mathbf{q}}_{d_i} \end{bmatrix} =^{0} \mathbf{J}_{ai} \dot{\mathbf{q}}_{a_i} +^{0} \mathbf{J}_{di} \dot{\mathbf{q}}_{d_i} \tag{1.5} $$

Then, the second constraint relation can be written as:

$$ \mathbf{J}_{t} {}^{0}\mathbf{t}_{p} - \mathbf{J}_{ta} \dot{\mathbf{q}}_a = \mathbf{J}_{td} \dot{\mathbf{q}}_d \tag{1.6} $$

where: $\mathbf{J}_{ta}$, $\mathbf{J}_{t}$ and $\mathbf{J}_{td}$ are the $(n_d \times n)$, $(n_d \times n)$ and $(n_d \times n_d)$ matrices relating the passive joint velocities $\dot{\mathbf{q}}_d$, active joint velocities $\dot{\mathbf{q}}_a$ and platform twist ${}^{0}\mathbf{t}_p$. Then, by using the Lagrange formalism, the dynamic model of the robot can be written as follows:

$$ \boldsymbol{\tau} = \boldsymbol{\tau}_{ta} - \mathbf{B}^T \boldsymbol{\lambda}_1 - \mathbf{J}_{ta}^T \boldsymbol{\lambda}_2 \tag{1.7} $$

$$ \mathbf{w}_p = \mathbf{A}^T \boldsymbol{\lambda}_1 - \mathbf{J}_{t}^T \boldsymbol{\lambda}_2 \tag{1.8} $$

$$ \boldsymbol{\tau}_{td} = \mathbf{J}_{td}^T \boldsymbol{\lambda}_2 \tag{1.9} $$

where:

- $\boldsymbol{\tau}$ is the $(n \times 1)$ vector of the robot input efforts;
- $\boldsymbol{\lambda}_1$ and $\boldsymbol{\lambda}_2$ are the $(n \times 1)$ and $(n_d \times 1)$ vector of Lagrange multipliers;
- $\boldsymbol{\tau}_{ta}$ is a $(n \times 1)$ vector, which corresponds to the virtual input efforts in the actuated joints of the parallel robot related to the Lagrange L_t of the virtual tree structure.

$$ \boldsymbol{\tau}_{ta} = \frac{\mathrm{d}}{\mathrm{d}t} \left(\frac{\partial L_t}{\partial \dot{\mathbf{q}}_a} \right)^T - \left(\frac{\partial L_t}{\partial \mathbf{q}_a} \right)^T \tag{1.10} $$

- $\mathbf{w}_p$ is the $(n \times 1)$ vector, which corresponds to the wrench of the free platform expressed in the base frame and related with the Lagrange L_p of the moving platform, and it is computed as:

$$\mathbf{w}_p = \frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial L_p}{\partial \dot{\mathbf{x}}}\right)^T - \left(\frac{\partial L_p}{\partial \mathbf{x}}\right)^T \tag{1.11}$$

- $\boldsymbol{\tau}_{td}$ is the $(n_d \times 1)$ vector, which corresponds to the virtual input efforts in the passive joints of the parallel robot related to the Lagrange L_t of the virtual tree structure, and can be computed as:

$$\boldsymbol{\tau}_{td} = \frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial L_t}{\partial \dot{\mathbf{q}}_d}\right)^T - \left(\frac{\partial L_t}{\partial \mathbf{q}_d}\right)^T \tag{1.12}$$

The Lagrange multipliers, which are the wrenches projected towards the platform controlled point through the use of the matrix $\mathbf{J}_t^T$, and can be expressed as:

$$\boldsymbol{\lambda}_2 = \mathbf{J}_{td}^{-T}\boldsymbol{\tau}_{td} \tag{1.13}$$

and:

$$\boldsymbol{\lambda}_1 = \mathbf{A}^{-T}(\mathbf{w}_p + \mathbf{J}_t^T\mathbf{J}_{td}^{-T}\boldsymbol{\tau}_{td}) \tag{1.14}$$

Then, the dynamic model is written as:

$$\boldsymbol{\tau} = \boldsymbol{\tau}_{ta} + \mathbf{J}^T\mathbf{w}_p + \mathbf{J}_d^T\boldsymbol{\tau}_{td} \tag{1.15}$$

where:
$\mathbf{J} = \mathbf{A}^{-1}\mathbf{B}$ is the Jacobian relating the platform twist ${}^0\mathbf{t}_p$ and the active joint velocities $\dot{\mathbf{q}}_a$. And $\mathbf{J}_d$ allows to express the passive joint velocities $\dot{\mathbf{q}}_d$ as a function of the active joint velocities.

1.3. Parallel or Type 2 Singularities

1.3.1 Degeneracy conditions of the Dynamic model

Let us analyze the dynamic model of the five-bar robot of figure 1.7, (Section 1.4 will address the dynamic model of the five-bar mechanism in more detail). The vector of active joints is given by $\mathbf{q} = [q_{11}, q_{21}]^T$ and the vector of active joint velocities $\dot{\mathbf{q}} = [\dot{q}_{11}, \dot{q}_{21}]^T$. The vector of moving platorm pose is given by $\mathbf{x} = [x, y]^T$ and its time derivatives $\dot{\mathbf{x}} = [\dot{x}, \dot{y}]^T$. And finally a is the distance between the actuated joints. All link lengths are identical.

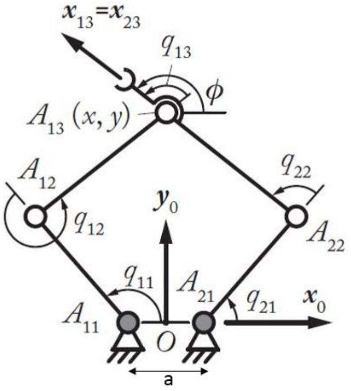

Figure 1.7: five-bar mechanism

The dynamic model of the five-bar mechanism can be written as follows:

$$\boldsymbol{\tau} = \mathbf{Z}\mathbf{Z}\ddot{\mathbf{q}} + \mathbf{B}^T\boldsymbol{\lambda} \tag{1.16}$$

$$\mathbf{w}_p = \mathbf{A}^T\boldsymbol{\lambda} \tag{1.17}$$

$$\boldsymbol{\tau} = \begin{bmatrix} zz_1 & 0 \\ 0 & zz_4 \end{bmatrix} \begin{bmatrix} \ddot{q}_{11} \\ \ddot{q}_{21} \end{bmatrix} + \begin{bmatrix} \ell\sin(q_{12}) & 0 \\ 0 & \ell\sin(q_{22}) \end{bmatrix} \begin{bmatrix} \lambda_{11} \\ \lambda_{21} \end{bmatrix}$$

$$\mathbf{w}_p = m \begin{bmatrix} \ddot{x} \\ \ddot{y} \end{bmatrix} = \begin{bmatrix} \cos(q_{11}+q_{12}) & \cos(q_{21}+q_{22}) \\ \sin(q_{11}+q_{12}) & \sin(q_{21}+q_{22}) \end{bmatrix} \begin{bmatrix} \lambda_{11} \\ \lambda_{21} \end{bmatrix}$$

Parallel or Type 2 singularity is depicted in figure 1.8. In such configuration the parallel robot gains an uncontrollable motion perpendicular to $\overrightarrow{A_{12}A_{13}}$ and $\overrightarrow{A_{22}A_{13}}$ (underactuation in the system).

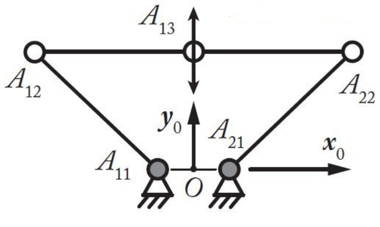

Figure 1.8: Singular position (Type 2 singularity)

The wrench applied on the platform by the legs and external forces take the following form:

$$\mathbf{w}_p = m \begin{bmatrix} \ddot{x} \\ \ddot{y} \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \lambda_{11} \\ \lambda_{21} \end{bmatrix}$$

And it is possible to notice that the kinematic matrix $\mathbf{A}^T$ becomes rank deficient and therefore:

$$\mathbf{w}_p = \mathbf{A}^T\boldsymbol{\lambda} \Longrightarrow \mathbf{A}^T\textbf{not invertible}$$

If the parallel robot is in a Type 2 singular configuration, the kinematic matrix $\mathbf{A}^T$ cannot be inverted, and then the dynamic model degenerates and cannot be solved. Moreover, in the neighborhood of the singularity, the torques $\boldsymbol{\tau}$ increase as their expression is proportional to the inverse of the determinant of $\mathbf{A}^T$, which is close to zero in the singularity locus.

1.3.2 Overview of crossing Type 2 singularity methods

In order to address the problem of crossing Type 2 singularities, several approaches have been proposed in the literature [6][11][12]. In this section two main approaches for non-singular assembly mode changing are discussed. The first one is the develpment of a physical criterion deduced from the degeneracy conditions of the dynamic model [12]. The second one is the computation of a Multi-Model Computed Torque Control for tracking optimal trajectories that respect the physical criterion through the use of optimal trajectory planning [6].

Dynamic criterion for non-singular assembly mode changing

In the singularity locus it has been shown that the determinant of the kinematic matrix $\mathbf{A}^T$ tends to zero, and therefore the input torques may increase to infinite values. The work in [12] proposes the deduction of a physical criterion through the degeneracy conditions of the dynamic model as follows:
Knowing that in Type 2 singularity, the matrix $\mathbf{A}$ is degenerated:

$$\boldsymbol{\lambda} = \mathbf{A}^{-T}\mathbf{w}_p \Longrightarrow \mathbf{A}^T\textbf{not invertible} \tag{1.18}$$

There exists a non-null vector $\mathbf{t}_s$ in the kernel of $\mathbf{A}^T$, such that:

$$\mathbf{t}_s^T\mathbf{A}^T\boldsymbol{\lambda} = \mathbf{t}_s^T\mathbf{w}_p = 0 \tag{1.19}$$

This is the condition of non-degeneracy of the dynamic model and implies that equation 1.19 means that the wrench applied on the platform by the legs and external forces $\mathbf{w}_p$ must be reciprocal to the twist $\mathbf{t}_s^T$ of uncontrollable motion in the singularity locus. In order to satisfy this physical condition, an optimal trajectory that respects the dynamic criterion must be planned.

Multi-Model Computed Torque Control

Planning optimal trajectories may not be enough for crossing Type 2 singularities due to uncertainties between the mathematical model and the real robot. However, it has been shown to be the most promising approach. Thus, the implementation of a computed torque controller for tracking optimal trajectories based on the dynamic model is proposed in [6] by using two dynamic models, one full dynamic model far from the singularity and one dynamic model that does not degenerate in the singularity locus. This allows crossing the singularity without torques discontinuities as long as the trajectory to be tracked respects the dynamic criterion in the singularity locus.

The computed torque control [13] is an advanced controller that computes the input torques that the actuators must apply to the robot, in order to track a given trajectory. It is based on the dynamic model because the input torques are synthesized to feedback linearized the system. The goal of the computed torque control is to minimize the error in the joint or cartesian space. From [14] and [15], it is possible to rewrite the dynamic model (1.16), and express it as follows:

$$\boldsymbol{\tau} = \mathbf{M}\ddot{\mathbf{q}} + \mathbf{H} \tag{1.20}$$

where $\mathbf{M}$ is the inertia matrix and $\mathbf{H}$ is the matrix that groups the gravitational, centrifugal and Coriolis terms. By using an input-output linearization, and knowing that the control input torques appear in the second time derivative, it is possible to replace $\ddot{\mathbf{q}}$, by a control signal $\mathbf{u}$, then the dynamics of the system is linear with respect to the control variable:

$$\boldsymbol{\tau} = \mathbf{M}\mathbf{u} + \mathbf{H} \tag{1.21}$$

Then:

$$\mathbf{u} = \ddot{\mathbf{q}} \tag{1.22}$$

It means that a Proportional Derivative control law can be used to impose the control signal:

$$\mathbf{u} = \ddot{\mathbf{q}}^d + \mathbf{K}_d(\dot{\mathbf{q}}^d - \dot{\mathbf{q}}) + \mathbf{K}_p(\mathbf{q}^d - \mathbf{q}) \Rightarrow \ddot{\mathbf{e}} + \mathbf{K}_d\dot{\mathbf{e}} + \mathbf{K}_p\mathbf{e} = 0 \tag{1.23}$$

where

- $\mathbf{u}$ is a $(n \times 1)$ vector;
- $\mathbf{q}^d$ is a $(n \times 1)$ is the desired joint position;
- $\dot{\mathbf{q}}^d$ is the desired joint velocity;

- $\mathbf{q}$ is the measured joint position;
- $\dot{\mathbf{q}}$ is the measured joint velocity;
- $\mathbf{e}, \dot{\mathbf{e}}, \ddot{\mathbf{e}}$ are the position, speed and acceleration errors in the joint space;
- $\mathbf{K}_p$, $\mathbf{K}_d$ are positive definite matrices of proportional and derivative gains to be tuned.

This controller is based on the dynamic model, and if the model is not accurate, then the tracking error can be noticeable, but the control signal still guarantees the convergence to zero of the tracking error with the desired second order dynamics. Figure 1.9 illustrates the CTC scheme. Finally, the CTC computes the input torques that ensure the second order dynamics on the tracking error is given by the following equation.

$$\boldsymbol{\tau} = \mathbf{M}(\ddot{\mathbf{q}}^d + \mathbf{K}_d(\dot{\mathbf{q}}^d - \dot{\mathbf{q}}) + \mathbf{K}_p(\mathbf{q}^d - \mathbf{q})) + \mathbf{H}(\mathbf{q}, \dot{\mathbf{q}}) \tag{1.24}$$

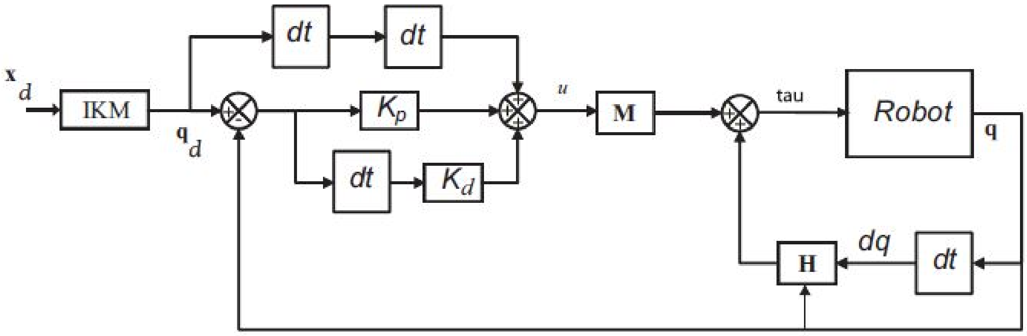

Figure 1.9: Computed Torque Control scheme [6]

It is important to mention that in order to feedback linearized the plant dynamics by means of the computed torque control, it is mandatory to be able to compute the dynamic model 1.20, which means that it must be possible to compute the inertia and Coriolis matrix. However, at singularity this is not possible because of the following computation:

$$\boldsymbol{\tau} = \mathbf{M}\ddot{\mathbf{q}} + \mathbf{H} = \boldsymbol{\tau}_{ta}(\ddot{\mathbf{q}}, \dot{\mathbf{q}}, \mathbf{q}) + \mathbf{J}^T\mathbf{w}_p \tag{1.25}$$

And,

$$\mathbf{J}^T = -\mathbf{B}^T\mathbf{A}^{-T} \tag{1.26}$$

In the singularity locus the kinematic matrix $\mathbf{A}^T$ is not invertible. Thus in [6] also propose a new reduced dynamic model that does not degenerate at the singularity locus by vanishing the wrench applied the on the platform by the legs and external forces, which means using a solution of ($\boldsymbol{\lambda} = \mathbf{0}$) for the lagrange multipliers.

Vanishing $\mathbf{w}_p \implies \mathbf{w}_p = 0$, then $\mathbf{J}^T\mathbf{w}_p = 0$, $\mathbf{J}^T = -\mathbf{B}^T\mathbf{A}^{-T} \implies$ is **vanished**, and then:

$$\boldsymbol{\tau}_B = \boldsymbol{\tau}_{ta}(\ddot{\mathbf{q}}, \dot{\mathbf{q}}, \mathbf{q}) \tag{1.27}$$

$$\boldsymbol{\tau}_B = \mathbf{M}'\ddot{\mathbf{q}} + \mathbf{H}' \tag{1.28}$$

$$\boldsymbol{\tau}_B = \mathbf{M}'(\ddot{\mathbf{e}} + \mathbf{K}_d\dot{\mathbf{e}} + \mathbf{K}_p\mathbf{e}) + \mathbf{H}'(\mathbf{q}, \dot{\mathbf{q}}) \tag{1.29}$$

where $\mathbf{M}'$ is the inertia matrix of the reduced dynamic model and $\mathbf{H}'$ is the matrix that groups the gravitational, centrifugal and Coriolis terms of the reduced dynamic model.

To sum up, the authors in [6] proposed a Multi-Model Computed Torque Control by using one full dynamic model $\boldsymbol{\tau}_A$ far from the singularity and $\boldsymbol{\tau}_B$, which is a reduced dynamic model that does not degenerate in the singularity locus and respecting the dynamic criterion. And the Computed Torque Control is applied to both models for tracking the optimal trajectory. Figue 1.10 shows the control scheme using the two different models. Moreover, for the transition between control laws a sigmoid time dependent function is used.

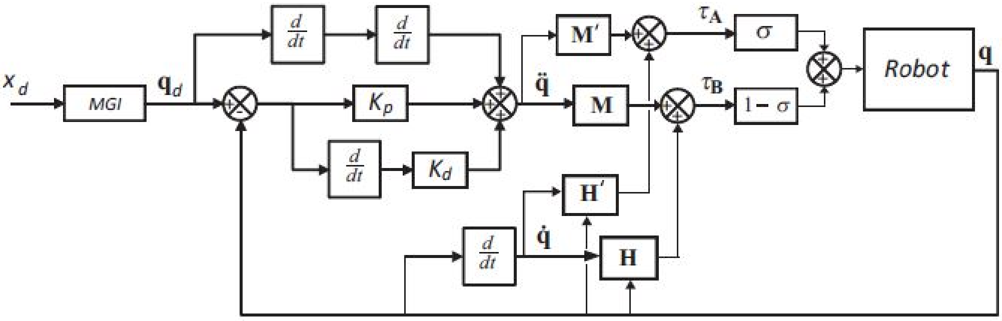

Figure 1.10: Multimodel Computed Torque Control scheme [6]

1.4. Five-bar mechanism: Benchmark in Type 2 singularity crossing

The control approach for crossing Type 2 singularities in the following chapters will be developed taking as the case of study the parallel manipulator, five-bar mechanism shown in figure 1.11. It is composed of legs attached to the base and the end-effector. The five-bar mechanism is a planar parallel robot with two actuated joints at A_{11} and

A_{21}, and three passive joints located at A_{12}, A_{22} and A_{13}. The actuation is provided by n active joints and n_d passive joints. The vector of active joints is given by $\mathbf{q} = [q_{11}, q_{21}]^T$ and the vector of active joint velocities $\dot{\mathbf{q}} = [\dot{q}_{11}, \dot{q}_{21}]^T$. The vector of passive joints is given by $\mathbf{q}_d = [q_{12}, q_{22}, q_{13}]^T$ and the vector of passive joint velocities $\dot{\mathbf{q}}_d = [\dot{q}_{12}, \dot{q}_{22}, \dot{q}_{13}]^T$. The vector of moving platorm pose is given by $\mathbf{x} = [x, y]^T$ and its time derivatives $\dot{\mathbf{x}} = [\dot{x}, \dot{y}]^T$. And finally a is the distance between the actuated joints. All link lengths are identical.

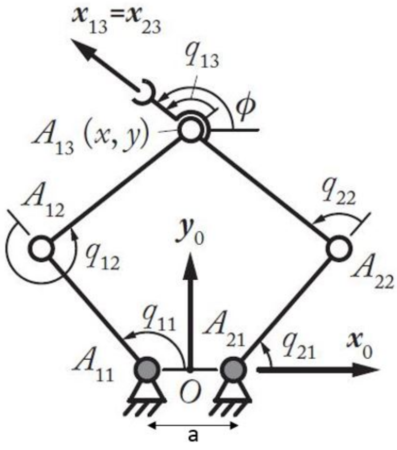

Figure 1.11: five-bar mechanism

1.4.1 Mechanical Architecture of the DexTAR robot

The five-bar mechanism used in the experimentation is the Dual-arm SCARA robot DexTAR (*Dextrous Twin-Arm Robot*) from the company **Mecademic**, it is shown in figure 1.12. All link lengths are identical ($l = 90mm$), and the distance between the actuated joints ($a = 118mm$).

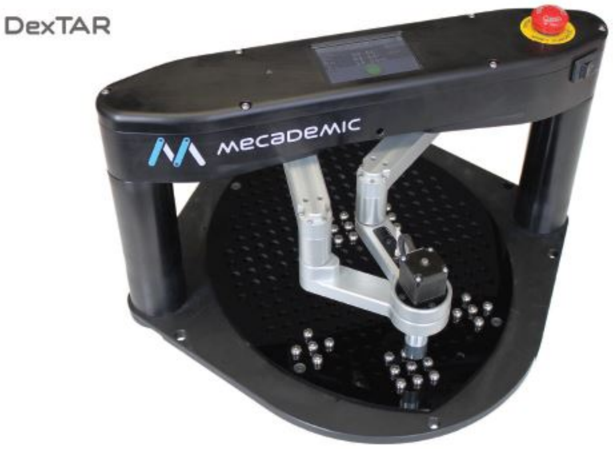

Figure 1.12: Robot DexTAR

The dyamic identified parameters used for the computation of the dynamic model were computed based on [9][10] and they have the following values:

- Inertias of the two proximal links: $zz_{11R} = 0.0133\frac{kg}{m^2}$ and $zz_{21R} = 0.0142\frac{kg}{m^2}$;
- Mass of the end-effector: $m = 0.537kg$;
- Friction terms: $f_{off11} = 0.0855\frac{Nm}{rad/s}$, $f_{off21} = 0\frac{Nm}{rad/s}$, $f_{s11} = 0.345Nm$ and $f_{s21} = 0.430Nm$

1.4.2 Dynamic model

The resulted dynamic model, from the identification of the dynamic parameters and the methodology presented in the section 1.2 of dynamic modeling, has the following form:

$$\boldsymbol{\tau} = \mathbf{ZZ}\ddot{\mathbf{q}} + \mathbf{B}^T\boldsymbol{\lambda} + \mathbf{f} \tag{1.30}$$

$$\mathbf{w}_p = \mathbf{A}^T\boldsymbol{\lambda} = m\ddot{\mathbf{x}} \tag{1.31}$$

$$\boldsymbol{\tau} = \begin{bmatrix} zz_{11R} & 0 \\ 0 & zz_{21R} \end{bmatrix} \begin{bmatrix} \ddot{q}_{11} \\ \ddot{q}_{21} \end{bmatrix} + \begin{bmatrix} \ell\sin(q_{12}) & 0 \\ 0 & \ell\sin(q_{22}) \end{bmatrix} \begin{bmatrix} \lambda_{11} \\ \lambda_{21} \end{bmatrix}$$

$$\mathbf{w}_p = m \begin{bmatrix} \ddot{x} \\ \ddot{y} \end{bmatrix} = \begin{bmatrix} \cos(q_{11}+q_{12}) & \cos(q_{21}+q_{22}) \\ \sin(q_{11}+q_{12}) & \sin(q_{21}+q_{22}) \end{bmatrix} \begin{bmatrix} \lambda_{11} \\ \lambda_{21} \end{bmatrix}$$

where:

The kinematic parallel and serial matrices **A** and **B** are computed from the input-output kinematic constraint relations 1.1 and 1.6. The matrix **ZZ** is positive diagonal resulted from the identified dynamic parameters:

$$\mathbf{A}^T = \begin{bmatrix} \cos(q_{11}+q_{12}) & \cos(q_{21}+q_{22}) \\ \sin(q_{11}+q_{12}) & \sin(q_{21}+q_{22}) \end{bmatrix}$$

$$\mathbf{B} = \begin{bmatrix} \ell \sin(q_{12}) & 0 \\ 0 & \ell \sin(q_{22}) \end{bmatrix}$$

$$\mathbf{ZZ} = \begin{bmatrix} zz_{11R} & 0 \\ 0 & zz_{21R} \end{bmatrix}$$

f group the active joint friction terms:

$$\mathbf{f} = \begin{bmatrix} f_{s11}\text{sign}(\dot{q}_{11}) \\ f_{s21}\text{sign}(\dot{q}_{21}) \end{bmatrix} + \begin{bmatrix} f_{off11} \\ f_{off21} \end{bmatrix} \tag{1.32}$$

1.4.3 Validation via Co-Simulation ADAMS/MATLAB

In order to validate the dynamic model of the robot DexTAR through a Co-Simulation between ADAMS and MATLAB, first the preparation of the computerized model of the robot was performed. Figures 1.13 and 1.14 show the CAD model done in CATIA.

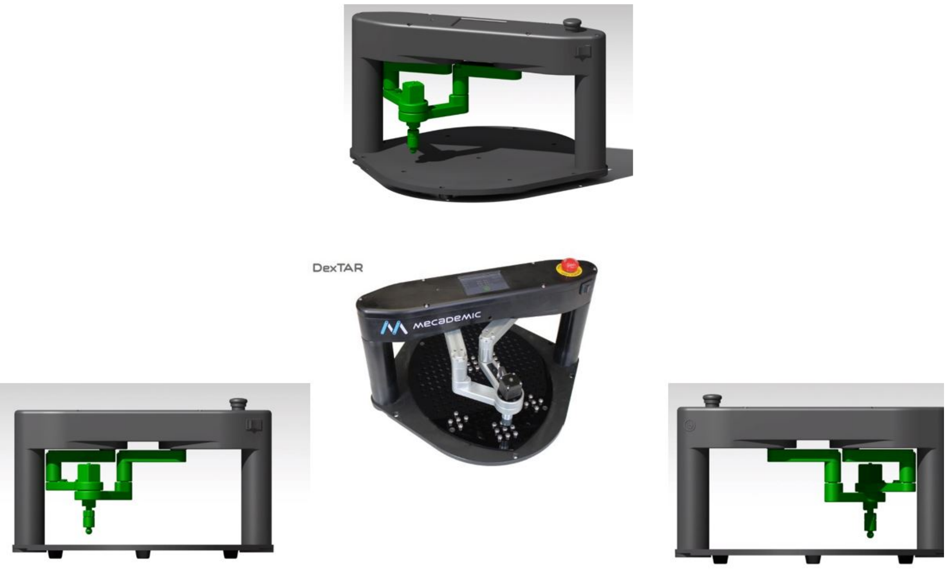

Figure 1.13: Assembly DexTAR

Once the computerized model was performed, the parts and the assembly from CATIA were exported to ADAMS software by following the next steps:

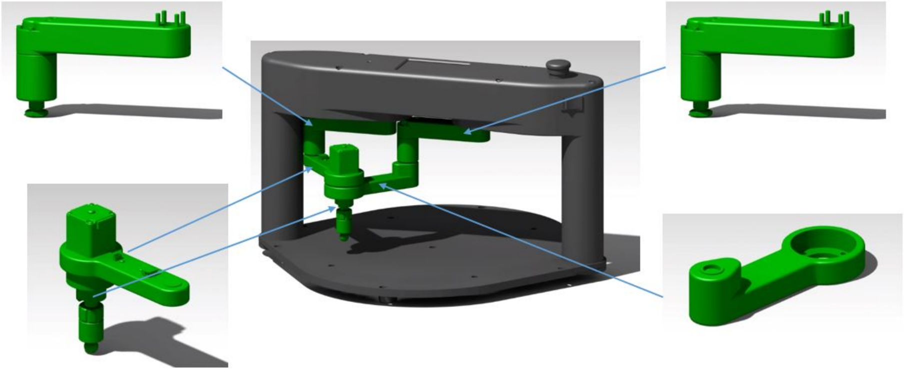

Figure 1.14: Assembly Parts

1. Open the Assembly (.CATProduct file) and regenerate each part (.CATPart) of the Assembly by selecting the option: Generate a CATPart from a Product. Do this process with each part of the Assembly.

2. Save each part generated as a .stl file.

Remark: By regenerating the CATPart from the CATProduct, once the .stl files are imported by ADAMS, ADAMS will respect the Assembly as it was in CATIA. The process is illustrated in the figure 1.15.

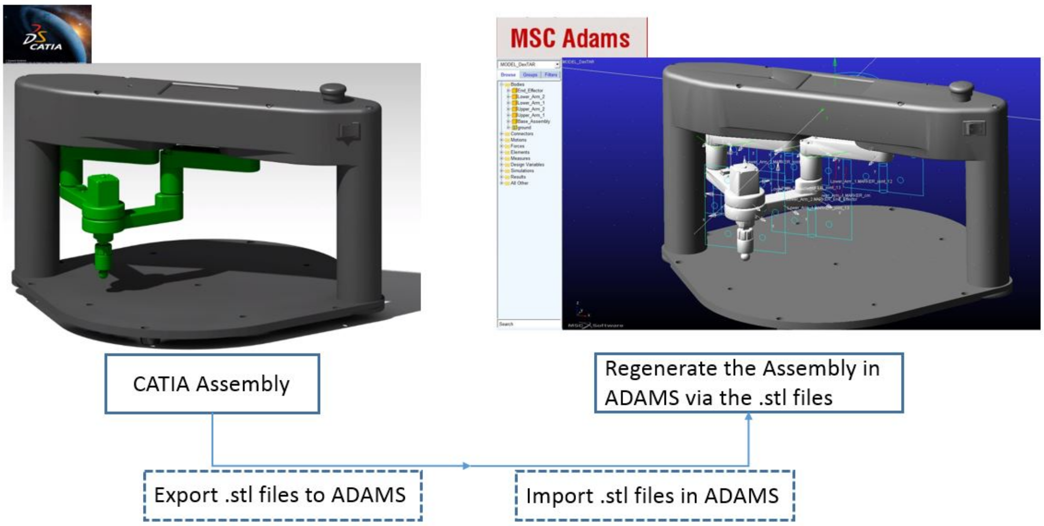
CATIA
MSC Adams
CATIA Assembly
Regenerate the Assembly in ADAMS via the .stl files
Export .stl files to ADAMS
Import .stl files in ADAMS

Figure 1.15: CATIA-ADAMS

Once the process of exporting the CATIA model to ADAMS was done, the ADAMS Mock-up was defined in order to perform the Co-simulation between ADAMS and MATLAB, by exporting the ADAMS plant and use the state variables from the plant of ADAMS into MATLAB, and then compare the dynamic model computed in matlab with the plant from ADAMS. The ADAMS Mock-up is shown in figure 1.16, and it shows all the reference frames of each joint and end-effector. It was also included the dynamic parameters in the Mock-up.

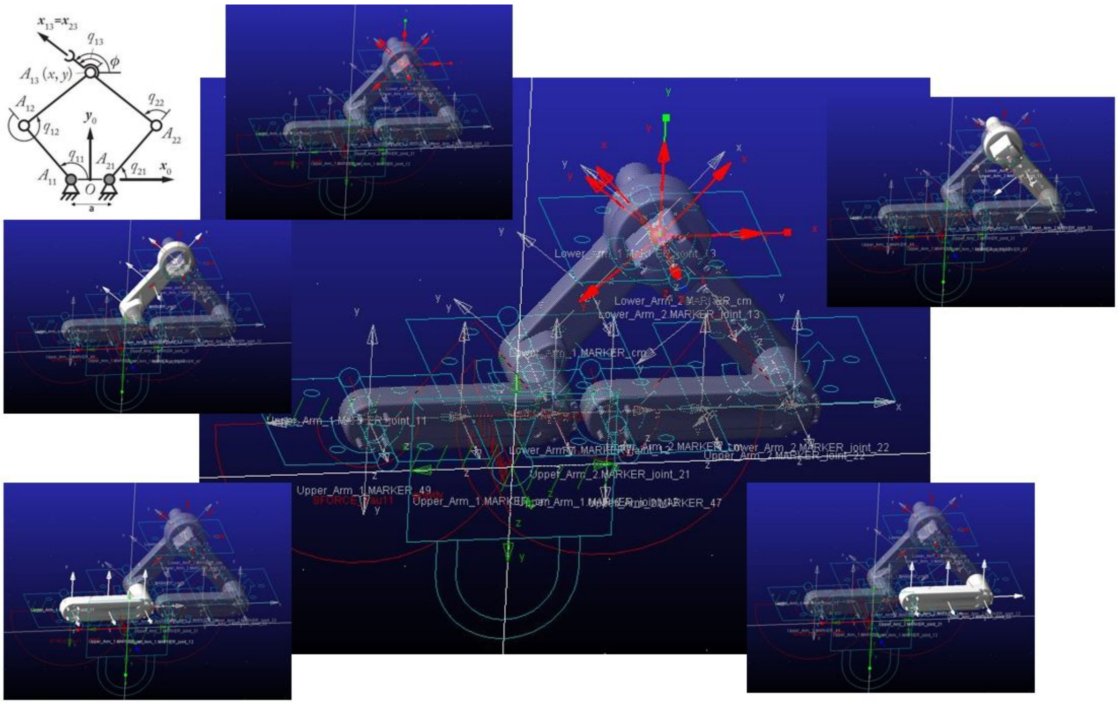

Figure 1.16: ADAMS Mock-up

Finally, having performed the preliminary set up for exporting the ADAMS plant of the DexTAR robot modeled in CATIA to MATLAB. The cross-validation of the dynamic model was performed obtaining the following results:

The trajetory given as an input for comparing the model and the ADAMS plant was a sinusoidal signal of $\frac{\sin(t)}{10}$ for the both active joints q_{11} and q_{21}.

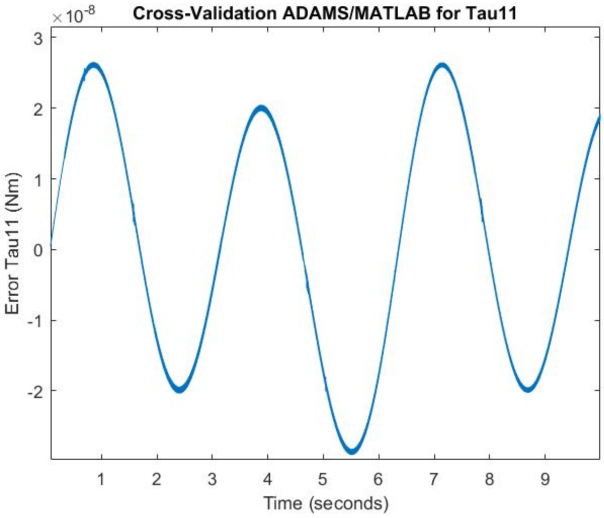

Figure 1.17: Error in the torque of the joint q_{11}

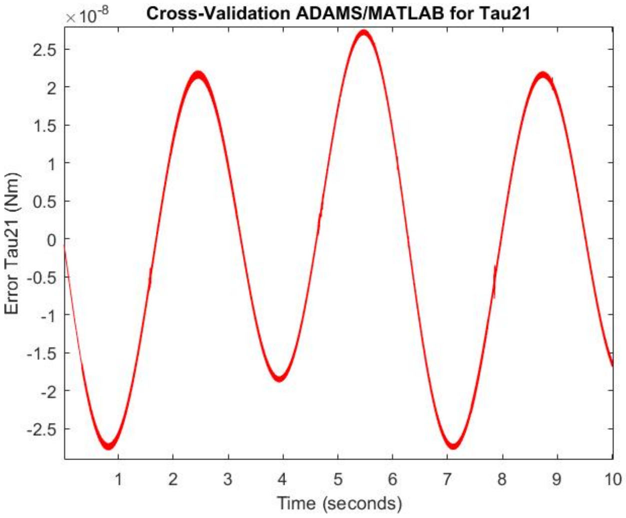

Figure 1.18: Error in the torque of the joint q_{21}

It can be seen from the figures 1.17 and 1.18 that the error of the inverse dynamic model seen from the error in the torques is minimum, having an error of the order 10^{-8}. From the minimum error in the dynamic model it is possible to make sure that the mathematical model of the dynamics of the five-bar mechanism including the identified dynamic parameters, is accurate and consistent.

Moreover, the built model of the five-bar mechanism developed in ADAMS would be useful for testing the results presented in terms of the control law for crossing Type 2 singularities in the following sections.

1.4.4 Summary

Firstly, in this chapter a general context of the present research work was discussed by giving some insights of the use and challenges when studying complex robot architertures, such as parallel robots. The description of the overall objective and contributions of this master thesis were listed by pointing out the main results that will be shown in this manuscript, and moreover the technical and conceptual mathematical problems that will be addressed when talking about singularities were introduced, such as underactuation, dynamic modeling consistency and performance indexes like degeneracy of the dynamic model when approaching singularities. And also the impact of the singularities in the overall performance of the parallel mechanisms.

In this chapter, first in section 1.1 a brief review of the parallel robots have been presented including their definition, and short overview of their inclusion into the industrial development. Parallel manipulators have been attracted for different researchers and industries due to their operational capabilities, such as high stiffness, high payload, small inertia and high speed and acceleration. Nevertheless, it has been listed the main drawback, which is the reachablility of the workspace due to the abundant singularity problems. Then, the assembly mode changing has been discussed by explaining how this operational feature is affected by meeting the Type 2 singularity when changing of configuration. Also an overview of the singularities in parallel robots have been introduced by classifying the three main types of singularities.

Section 1.2 presented the modeling of the dynamics of parallel mechanisms by recalling the dynamics principles of parallel robots going from the analysis of the closed kinematic chain itself to the analysis and understanding of the virtual tree structures and the Lagrangians. Moreover, important formulations, such as the two kinematic constraint relations for computing the parallel and serial jacobian matrices were given.

Section 1.3 analyzed the Degeneracy conditions of the dynamic model, taking as the case of study the five-bar mechanism, and also giving an overview of the crossing Type 2 singularity techniques in the literature. The analysis of the degeneracy conditions of the dynamic model, and the Control technique for tracking far from the singularity presented will be required for the developing of further analyses. Furthermore, the overview of the crossing Type 2 singularity methods will give an insight of how constraining these singularities are when talking about reconfigurability of parallel robots. Thus, the reasons why studying dedicated controllers for dealing with these singularities will be motivated.

Furthermore, in this chapter the conceptual and technical aspects of Type 2 singularities is addressed, including as illustrative example, the five-bar mechanism used for experimentation in the present research work. The degeneracy conditions of the dynamic model were explained and deduced. Then, the most promising Type 2 singularity methods in the literature were discussed, first the dynamic criterion deduced from the degeneracy conditions of the dynamic model and then a Multi-Model Computed Torque Control for tracking the optimal trajectories that respect the dynamic criterion.

In this chapter, in section 1.4., the case of study to be treated in this research was described, the five-bar mechanism robot DexTAR. This was done by giving the main mechanical characteristics, such as description of actuated and non-actuated joints, as well as end-effector description.

In addition to that, the identified dynamic parameters and the resulted dynamic model of the five-bar mechanism was deduced. Finally, in order to cross-validate the mathematical model with ADAMS, the preparation of the CAD model of the DexTAR robot was built, as well as the ADAMS Mock-up for validation of the mathematical model with the ADAMS plant. Accurate results were presented in the validation of the mathematical model by analysing the error in the torques of the actuated joints.

In the following chapters the main contributions of the present research work will be presented. The next chapter will address the modeling of the free dynamics deduced from the degeneracy conditions of the dynamic model and its equivalency with the dynamic criterion, then the design of a virtual holonomic constraint to be enforced via feedback in order to cross Type 2 singularities. This virtual holonomic constraint will be presented by using controlled variables in both spaces, joint and cartesian space.

Chapter 2

Modeling of the Free Dynamics and the Virtual Holonomic Constraints

This chapter will present two of the main contributions of this research work. Firstly, a general definition of virtual holonomic constraints (VHCs) will be discussed by giving illustrative examples and technical aspects to be taken into account. A definition of zero dynamics will be stated by giving a general formulation for a system that by means of a control input looks for enforce the output to zero. Then the derivation of the free dynamics for the five-bar mechanism will be computed based on the degeneracy conditions of the dynamic model at the singularity locus that were derived in chapter 1. This analysis will be used for identifying the controlled and uncontrolled variables at the singularity locus. Then, the development of the VHC in the joint and task space will be computed, and a comparisson between them will be made by pointing out advantages of each other. Finally, the summary of this chapter will be discussed by pointing out further concepts related to the Advanced Controller developed in this work based on virtual holonomic constraints.

2.1. Virtual Holonomic Constraints

Virtual Holonomic Constraints are valuable useful tools that have emerged from analysing complex mechanical architectures that deal with underactuation in its system dynamics. Many mechanical systems are underactuated, they have less number of actuators than the number of degrees of freedom. However, they are able to perform complex motions by intelligently coordinate their actuated joints. This is the case of applications such as bioinspired robots, aerial vehicles, inverted pendulums, etc. Researchers in non-linear control and complex mechanical systems have found this subject of virtually constrained underactuated systems as powerful and challenging control tool for enforcing constraints to perform complex motions that from the nature of the system configuration would be impossible. Among the main contributors on control of underactuated systems via virtual holonomic constraints, there are several authors. For instance humanoid walking in [17], inverted pendulum in [18], aerial systems in [21][22][23]. Moreover, rigorous mathematical formulations have been presented in [18][24][26]. Furthermore, this mathematical dervations are important contributions and tools that are presented as a methdology for orbital stabilization of underactuated systems.

In figure 2.1 from [25], few illustrative examples for understanding the conceptual meaning of virtual constraints are useful. In figure 2.1(a), it is possible to find a geometrical relation among the piston variables and the angle θ in the driven shaft in order to synchronize the rotation of the engine. Then in 2.1(b) for example, two bar robot could be constrained to perform a desired motion in the vertical axis by finding a geometrical relation between q_1 and q_2. The figure 2.1(c) shows the possibility of finding a control strategy to perform coordinated longitudinal and lateral control between the two vehicles. Finally in figure 2.1(4) it would be a typical humanoid example for a walking gait, where the evolution of the links must be coordinated to perform a stable orbital or periodic motion.

Typically the applications where VHCs are needed for addressing problems of underactuation in system dynamics, are related to problems like humanoids walking, for stabilizing periodic motions. Moreover, in such systems the problem is to orbitally stabilize walking gaits. It has been shown that the application of VHCs is typically to orbitally stabilize periodic motions in virtual tree sturctures, such as humanoids. Thus, the concept and enforcement of virtual holonomic constraints is a useful technique to enforce a constraint to deal with the stabilization of walking robots by enforcing limit cycles or periodic orbits.

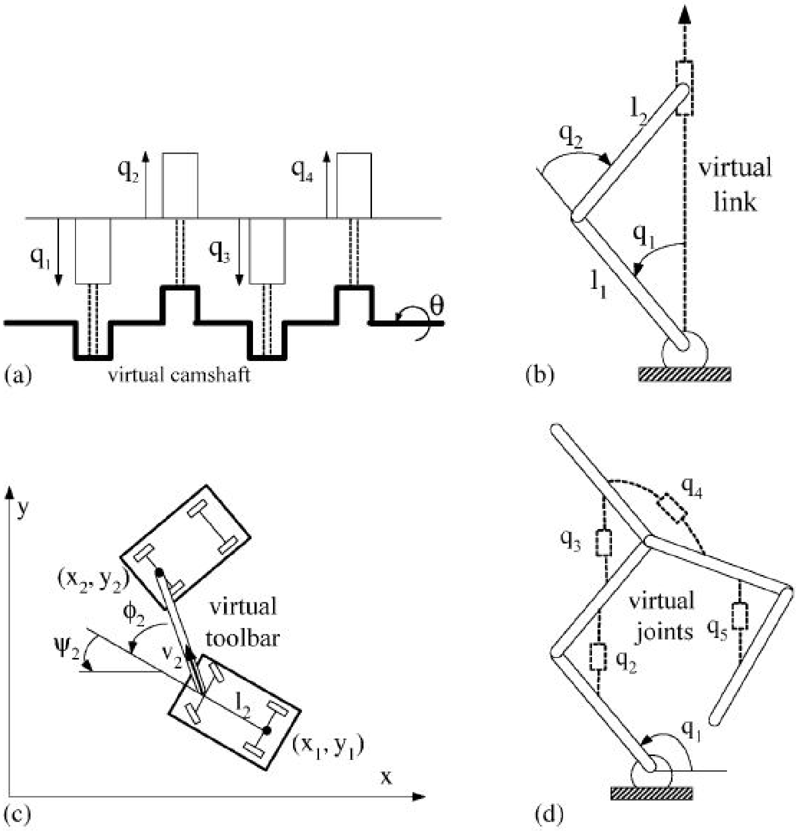

Figure 2.1: Virtual Constraints Examples [25]

When talking about virtual holonomic constraints, it is necessary to define what zero dynamics are in underactuated systems. Normally in a given dynamical system the zero dynamics are represented by the states of the system that in its state space representation the matrix that maps from the control inputs to the control torques/forces, is not full rank, and therefore those states are not controllable. And then the output function of that system is represented by the virtual constraint that is enforced towards zero by means of the states that are controllable directly by the control torques. Thus, basically if the states that are not controllable due to the underactuation of the system, can be included in the output function defined by the virtual constraint, then through control action the tracking error of the virtual constraint can be driven towards zero.

Summing up, applying methods such as the VHCs by Shiriaev (2008) and others [19], have demonstrated that the stabilization is feasible when there is underactuation in the system. Typically this is done as a technique for orbital stabilization of walking gaits, where the mechanisms are virtual tree structures.

Nevertheless, in the present research work the concept and technique of virtual constraints will be used for addressing the problem in closed kinematic chains once the lagrange multipliers have been used to close the virtual tree structures.

In the case of parallel robots the problem is to dynamically reconfigure the robot, which are closed kinematic chains, to change assembly modes. This implies to meet in the process with Type 2 singularities, which brings the system to gain uncontrollable degrees of freedom making the system underactuated at the singularity. It can be then understood that the parallel robots are not underactuated by nature, but by the degeneracy of the dynamic model at the Type 2 singularities.

Virtual Holonomic Constraints are constraints that can be enforced by the application of control inputs that add auxiliary control laws for keeping the constraints invariant [16]. The idea of VHCs is a useful paradigm for the control of underactuated systems. Virtual constraints are geometric relations among the generalized coordinates of the system. The constraint is virtual because it does not arise from a physical connection between the generalized coordinates, but rather from the actions of a feedback controller. Virtual constraints have emerged recently as a valuable tool to solve motion control problems. For an underactuated Euler-Lagrange system, virtual constraints are defined as relations among the system's variables and are enforced by feedback, which leads to reducing the system's degrees of freedom [17][18]. To sum up:

- In the case of a fully actuated system, a given trajectory can be stabilized asymptotically by a linear controller, but if there exists underactuation it might not be the case.

- The VHCs are geometrical relations among the generalized coordinates in order to virtually constrained a system to perform a desired motion even if the system is not fully actuated.

Why VHCs for addressing the problem of Type 2 singularities? It has been shown in the Chapter 1 that when a Type 2 singularity occurs, the five-bar mechanism gains one uncontrollable motion perpendicular to $\overrightarrow{A_{12}A_{13}}$ and $\overrightarrow{A_{22}A_{13}}$ (Fig 1.8). This means that the system is underactuated in the singularity locus, in the sense that it has gained an uncontrollable degree of freedom. Thus, the use of VHCs for solving this problem seems to be a promising solution for dealing with that uncontrollable variable and cross the Type 2 singularity. In further analyses, it will be shown that by studying the degeneracy conditions of the dynamic model, it is possible to identify from the five-bar mechanism dynamics, the uncontrollable variable at singularity. And therefore, a geometrical relation between the available controlled variables and the uncontrollable or passive variable can be defined. This will be the virtual constraint to be enforced in order to deal with the underactuation. However, the operations in parallel robots are typically to perform motions such as pick and place operations where assembly mode changing is required. In addition to that, the region or where the robot is underactuated is not for the entire workspace, but only when the kinematic jacobian matrix **A** is numerically not invertible. This makes the problem of underactuation at singularity even more interesting and challenging, because the five-bar mechanism by nature is not underactuated, but only in the singularity. Then addressing the problem of underactuation, due to Type 2 singularities, by modeling virtual holonomic constraints is a novel approach to use. Finally, the goal will be then to find a VHC such that at singularity it respects the dynamic criterion 1.19 and this criterion can be included in the VHC to be enforced by the controller. And then, there will not be need of planning optimal trajectories, but by the virtual constraint approach enforce the dynamic criterion. Thus the design and implementation of VHCs in underactuated systems, which is the case of parallel robots at Type 2 singularity, can be listed as follows:

- since virtual constraints are not physically present, the advantage of imposing them in complex mechanical architectures, such as parallel robots for undergoing in Type 2 singularities, brings the possibility of dynamically reconfigure the robot through feedback control action rather than physically, this would be the case of planning optimal trajectories for respecting the dynamic criterion;

- virtual constraints might be trivial to define in closed kinematic chains due to the fact that through the input/output kinematic constraint relations it is possible to easily map the joint space with the cartesian space. Thus, geometrical relations can be found;

- advanced control techniques synthesized from imposing virtual constraints open the possibility to develop dedicated dynamic controllers for crossing Type 2 singularities by virtually constrained the robot when becomes underactuated and demonstrate consistency in its dynamics.

2.2. Free Dynamics and its equivalence with the Dynamic Criterion

Let us recall the main goal to be pursued by defining VHC for the five-bar mechanism in the singularity locus. So far, it has been shown that in the literature the most promising approach for crossing singularities is to define optimal trajectories that respect the dynamic criterion presented in chapter 1. Nevertheless, the objective is now to define a novel Advanced Controller that is not restricted to optimal trajectories, but it can be given any non optimal trajectory, and the controller is able to track continuously this trajectory. In order to synthesized this control law, first is necessary to derive the VHC that will be enforced in the singularity locus based on the analysis of the free dynamics that will take into account the dynamic criterion.

2.2.1 Recalling the Dynamic model and its degeneracy conditions

Recalling the five bar mechanism depicted in figure 1.7. The vector of active joints is given by $\mathbf{q} = [q_{11}, q_{21}]^T$ and the vector of active joint velocities $\dot{\mathbf{q}} = [\dot{q}_{11}, \dot{q}_{21}]^T$. The vector of moving platorm pose is given by $\mathbf{x} = [x, y]^T$ and its time derivatives $\dot{\mathbf{x}} = [\dot{x}, \dot{y}]^T$. And finally a is the distance between the actuated joints. All link lengths are identical.

Figure 2.2 represents a particular posture of singular configuration when the end-effector frame is aligned with the reference frame, it means $(x = 0)$. Thus, the goal is to apply the concept of virtual holonomic constraints to address the problem of parallel or Type 2 singularities, first to the particular case when $(x = 0)$ and then extend it to a singular configuration in any other end-effector posture in the workspace as shown in fig 2.3.

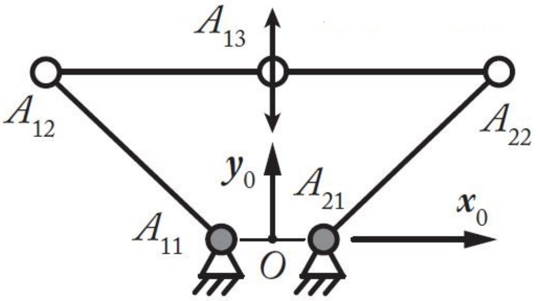

Figure 2.2: Singular Position (Type 2 Singularity)

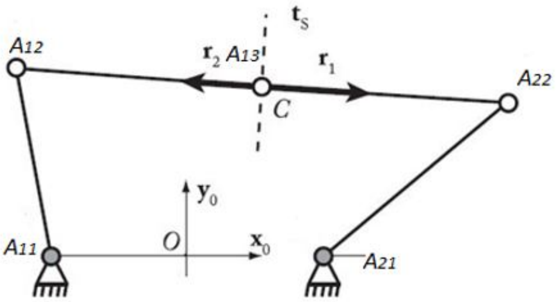

Figure 2.3: Singular Position when end effector frame is not aligned with the reference frame

Then, the first goal is to model a geometrical relation among the generalized coordinates in such a way that the system is virtually constrained to perform a motion in the vertical axis ($x = 0$) around the Type 2 singularity. Once the geometrical relation is defined, a feedback control law is derived for the enforcement of this virtual constraint.

In order to define the input/output relation among the generalized coordinates to compute the virtual holonomic constraint, it is necessary to identify the controlled variables and the uncontrolled variables at singularity. For the specific case when ($x = 0$), depicted in figure 2.2, the wrench applied to the platform has the following form:

$$\mathbf{w}_p = m \begin{bmatrix} \ddot{x} \\ \ddot{y} \end{bmatrix} = \begin{bmatrix} \cos(q_{11}+q_{12}) & \cos(q_{21}+q_{22}) \\ \sin(q_{11}+q_{12}) & \sin(q_{21}+q_{22}) \end{bmatrix} \begin{bmatrix} \lambda_{11} \\ \lambda_{21} \end{bmatrix}$$

$$\mathbf{w}_p = m \begin{bmatrix} \ddot{x} \\ \ddot{y} \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \lambda_{11} \\ \lambda_{21} \end{bmatrix}$$

Then:

$$m\ddot{x} = \lambda_{11} + \lambda_{21} \tag{2.1}$$

$$m\ddot{y} = 0 \tag{2.2}$$

Equation 2.2 represents the evolution of the passive variable, and therefore the identification of the controlled and uncontrolled variables can be done:

- y is uncontrollable at singularity (underactuation).

- x controllable at singularity (task space controlled variable) $\longrightarrow$ VHC with cartesian space controlled variables. This can be prove by expressing the direct dynamic model as follows: Knowing that $\ddot{x}$ has non-zero terms due to the fact that the kinematic jacobian matrix becomes $\mathbf{A}_s = \begin{bmatrix} 1 & 1 \end{bmatrix}$.

Then, the platform dynamics $\ddot{x}$ can be affected by the input torques $\boldsymbol{\tau}$, given that the direct and inverse dynamic model can be computed by considering that $m\ddot{x} \neq 0$. This states that locally even if y is uncontrollable, $x(y)$ is controllable.

$$\ddot{x} = f(\boldsymbol{\tau}) \tag{2.3}$$

In the next chapter, by synthesizing the control law for enforcing the virtual constraints, this proof will be recalled and extended in more detail ,when developing the controller using the cartesian space controlled variable x, by computing the dynamic model in the singularity and showing its consistency.

- $q_{i1}(y)$ $(i = 1, 2)$ controllable at singularity (joint space controlled variables) but compromising the dynamic model computation in the singularity due to the fact that it will be shown that the second order kinematics of the VHC in joint space is an approximation of $\ddot{q}_{i1}(y)$, and not an exact solution $\longrightarrow$ VHC with joint space controlled variables

Finally, it can be shown that for the controller, the VHC to be designed either with joint or cartesian space controlled variables will consider the dynamic criterion since the free dynamics is equivalent with the criterion as follows:

$$m\ddot{y} = \mathbf{t}_s^T \mathbf{w}_p = 0 \longrightarrow \textbf{Dynamic Criterion} \tag{2.4}$$

It is important to highlight that since the case of study in this research work is the five-bar mechanism, the equivalency between the free dynamics and the dynamic criterion shown in the present section is developed for this parallel mechanism. Nevertheless, for a general proof of the equivalency of the free dynamics with the dynamic criterion, refer to Appendix A.

2.3. Virtual Holonomic Constraints using the joint space controlled variables

Based on the previous analysis, the controlled and uncontrolled variables have been identified, and therefore the goal is to find a geometrical relation of the form:

$$h_i = q_{i1} - \phi_i(y) \tag{2.5}$$

where:
q_{i1} → set of controlled variables
$\phi_i(y)$ → the desired evolutions of q_{i1}

From the geometric model of the five bar mechanism is possible to derive the following equations:

$$\left(x + \frac{a}{2} - \ell\text{cos}(q_{11})\right)^2 + \left(y - \ell\text{sin}(q_{11})\right)^2 - \ell^2 = 0 \tag{2.6}$$

$$\left(x - \frac{a}{2} - \ell\text{cos}(q_{21})\right)^2 + \left(y - \ell\text{sin}(q_{21})\right)^2 - \ell^2 = 0 \tag{2.7}$$

Since the goal is to virtually constrained the robot to perform a motion in the vertical axis around the Type 2 singularity, it is simply necessary to impose ($x(y) = 0$) in the geometric model as follows:

$$\left(\frac{a}{2} - \ell\text{cos}(q_{11})\right)^2 + \left(y - \ell\text{sin}(q_{11})\right)^2 - \ell^2 = 0 \tag{2.8}$$

$$\left(-\frac{a}{2} - \ell\text{cos}(q_{21})\right)^2 + \left(y - \ell\text{sin}(q_{21})\right)^2 - \ell^2 = 0 \tag{2.9}$$

In the following chapter, it will be shown that in order to enforce the constraint $x(y) = 0$, the derivation of a feedback control law must be computed.
Finally, from the equations 2.8 and 2.9, it is possible to find the virtual constraints:

$$h_1 = q_{11} - \phi_1(y) = 0 \tag{2.10}$$

$$h_2 = q_{21} - \phi_2(y) = 0 \tag{2.11}$$

which match perfectly with the desired form in equation 2.5

2.4. Virtual Holonomic Constraints using the cartesian space controlled variables

Similarly, the controlled and uncontrolled variables for imposing a motion in the vertical axis, the geometrical relation can be modeled by using cartesian space controlled variables as follows:

$$h = x(y) = 0 \tag{2.12}$$

where:
x → set of controlled variables
y → the desired evolutions of x

From the modeling of the VHC with the cartesian space controlled variable x, it can be seen that computationally is less complex than modeling virtual constraints using controlled variables in joint space. One of the reasons why is less complex to model the virtual constraint in the cartesian space is because the uncontrollable variable is in the cartesian space y, then the left controllable variable x is also found in this domain . On the other hand, by computing the VHC with joint space controlled variables to enforce an uncontrollable variable that belongs to the cartesian space, then the use of the inverse geometric and kinematic models will be needed as shown in section 2.3. Moreover, the virtual constraints in the cartesian space are more simple to express.

It is interesting to note the fact that linking complex mechanical architectures with novel control techniques for stabilizing the system dynamics given any trajectory is a research area which brings many challenges. Then, by analyzing degeneracy conditions in system dynamics in order to deal with singularities in mathematical models is a novel technique that allows to study at low level the uncontrolled and controlled variables. Thus, studying virtual constraints as a novel tool for dynamic reconfigurabilty of parallel robots it is an interesting contribution of the present research work. Furthermore, applying virtual constraints to singularity crossing techniques would bring a new research area for the dynamic reconfigurability in parallel mechanisms in order to deal with the underactuation that Type 2 singularities bring to the robots.

2.5. Summary

In this chapter it has been modeled the evolution of the uncontrolled variable at the singularity for a particular case of singular posture when the end-effector frame is aligned with the reference frame. This evolution of uncontrolled variable describes the free dynamics of the robot at singularity. Then, once the free dynamics have been identified from the wrench applied on the platform, its equivalency with the dynamic criterion was presented, and it will be used in further analyses for the control design.

Finally, the identification of controlled variables in the joint and cartesian space were useful for designing the VHC in the two different spaces. Moreover, the uncontrollable variable was used for knowing how the controlled variables would need to evolve in such a way that the VHC can be enforced around the singularity locus.

In addition to that some observations can be stated about the form it has been found for the VHC in both joint and cartesian space. By imposing the constraint to virtually constrained the system to perform a motion in the vertical axis, the form that the VHC takes by using joint space controlled variables is more complex than the one with cartesian space controlled variables. Nevertheless, both forms of the VHC are equivalent, but it will be seen that there is more consistency for the dynamic model computation in the control design section by using cartesian controlled variables.

Furthermore, if dynamic criterions for dealing with certain physical problems such as underactuation can be included in the enforcement of virtual constraints, then reconfigurability problems could be solved in parallel robots. Moreover, the fact of being able to synthesize inverse and direct geometric and kinematic models brings the opportunity to map from one domain to another giving the possibility to design virtual constraints by using controlled variables in both joint and cartesian spaces, irrelevant of the uncontrollable variable at the singularity locus, as long as dynamic model consistency is proved as will be shown in chapter 3.

The following chapter seeks to synthesized a Virtual Constraint Controller for the enforcement of the VHC in both spaces. Moreover, a rigorous mathematical formulation will be presented in order to demonstrate that the dynamic model is consistent at the singularity. Moreover, the Virtual Constraint Controller will be generalized for the entire workspace, and not just for dealing with the situation when the end-effector is aligned with the reference frame.

Chapter 3

Virtual Constraint Controller

This chapter is dedicated for the development of a control law for the enforcement of the VHC in the singularity locus. This is done by expressing the VHC with joint and task space controlled variables. Then, the Virtual Constraint Controller is derived by using an input-output linearization. Moreover, it is demonstrated that the computation of the inverse and direct dynamic model are consistent when deriving the dynamic based feedback control law. Finally, it gives a general formulation of the control law for extending the VHC to the entire workspace by premultiplying the wrench applied on the platform with a Rotation matrix derived from the computation of the passive joints. This means that the VHC designed for virtually constrained the five-bar mechanism to move in the vertical axis can be extended to cross in other place of the workspace and not when the reference and end-effector frames are vertically aligned. Finally, a summary of the resulted contributions from this chapter are done by emphasizing the main results in terms of the theoretical Virtual Constraint Controller with the VHC modeled in both spaces and showing consistency in the computation of the dynamic model. Furthermore, the synthesized Virtual Constraint Controller will be used for the implementations in the real robot presented in chapter 4.

3.1. Virtual Constraint Controller: Part I (joint space controlled variables)

Defining the VHCs for virtually constrained the five-bar mechanism to move along the vertical axis around the singularity is not enough for dealing with the underactuation at that singular configuration. The second challenge of the formulation of VHCs is to find a control law, such that the geometrical relations (VHCs) are enforced to keep invariant through this control action. In order to do that, firstly in this section we are going to define a Virtual Constraint Controller for the VHC expressed with the joint space controlled variables.

3.1.1 Computation of the Dynamic Model in the Singularity locus

Since the control input appears in the second time derivative of the VHC expressed in the joint space, it is necessary to differentiate twice the virtual constraints:

$$\dot{h}_1 = \dot{q}_{11} - \phi_1^{'}(y)\dot{y} \tag{3.1}$$

$$\dot{h}_2 = \dot{q}_{21} - \phi_2^{'}(y)\dot{y} \tag{3.2}$$

$$\ddot{h}_1 = \ddot{q}_{11} - \phi_1^{''}(y)\dot{y}^2 - \phi_1^{'}(y)\ddot{y} \tag{3.3}$$

$$\ddot{h}_2 = \ddot{q}_{21} - \phi_2^{''}(y)\dot{y}^2 - \phi_2^{'}(y)\ddot{y} \tag{3.4}$$

In addition to that, it is also possible to compute the first and second time derivatives by means of the first and second order kinematic models. Firstly, since $(x = 0)$ is the constraint to be enforced, therefore $(\dot{x} = 0)$ for the first order kinematic model and then $(x = 0)$, $(\dot{x} = 0)$ and $(\ddot{x} = 0)$ for the second order kinematic model.

Then, for computing the first time derivative of the VHC ($\dot{h}_1$ and $\dot{h}_2$) meaning that $(x = 0$ and $\dot{x} = 0)$ for the first order kinematic model:

$$\mathbf{A}\dot{\mathbf{x}} + \mathbf{B}\dot{\mathbf{q}} = 0 \tag{3.5}$$

$$\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} 0 \\ \dot{y} \end{bmatrix} + \begin{bmatrix} B_{11} & 0 \\ 0 & B_{22} \end{bmatrix} \begin{bmatrix} \dot{q}_{11} \\ \dot{q}_{21} \end{bmatrix} = 0$$

where:

$A_{11} = \frac{a}{2} - \ell\cos(\phi_1(y))$

$A_{12} = y - \ell\sin(\phi_1(y))$

$A_{21} = -\frac{a}{2} - \ell\cos(\phi_2(y))$
$A_{22} = y - \ell\sin(\phi_2(y))$
$B_{11} = -\ell(\frac{a}{2} - \ell\cos(\phi_1(y)))\sin(\phi_1(y)) + \ell(y - \ell\sin(\phi_1(y)))\cos(\phi_1(y))$
$B_{22} = -\ell(-\frac{a}{2} - \ell\cos(\phi_2(y)))\sin(\phi_2(y)) + \ell(y - \ell\sin(\phi_2(y)))\cos(\phi_2(y))$

From the equation 3.5 it is possible to define the first time derivative of the virtual constraints in the form of expressions (3.1) and (3.2):

$$\dot{\mathbf{q}} = -\mathbf{B}(y)^{-1}\mathbf{A}(y)\dot{\mathbf{x}}_y \tag{3.6}$$

which is equivalent to:

$$\dot{\mathbf{h}} = \dot{\mathbf{q}} - (-\mathbf{B}(y)^{-1}\mathbf{A}(y)\dot{\mathbf{x}}_y) = 0$$

Then, for computing the second time derivatives of the VHC ($\ddot{h}_1$ and $\ddot{h}_2$) meaning that ($x = 0$, $\dot{x} = 0$ and $\ddot{x} = 0$) for the second order kinematic model:

$$\mathbf{A}\ddot{\mathbf{x}} + \dot{\mathbf{A}}\dot{\mathbf{x}} + \mathbf{B}\ddot{\mathbf{q}} + \dot{\mathbf{B}}\dot{\mathbf{q}} = 0 \tag{3.7}$$

By substituting $\dot{\mathbf{q}}$ from equation 3.6, into equation 3.7 and finally differentiating $\mathbf{A}(y)$ and $\mathbf{B}(y)$, it is possible to compute the second time derivative of the virtual constraint through the second order kinematic model:

$$\ddot{\mathbf{q}} = -\mathbf{B}(y)^{-1}(\mathbf{A}(y)\ddot{\mathbf{x}}_y + \dot{\mathbf{A}}(y,\dot{y})\dot{\mathbf{x}}_y + \dot{\mathbf{B}}(y,\dot{y})\dot{\mathbf{q}}) \tag{3.8}$$

$$\ddot{\mathbf{h}} = \ddot{\mathbf{q}} - (-\mathbf{B}(y)^{-1}(\mathbf{A}(y)\ddot{\mathbf{x}}_y + \dot{\mathbf{A}}(y,\dot{y})\dot{\mathbf{x}}_y + \dot{\mathbf{B}}(y,\dot{y})\dot{\mathbf{q}})) = 0$$

which is equivalent to the relations 3.3 and 3.4, and:
A(y) is the 2×2 parallel jacobian matrix in the y coordinates;
B(y) is the 2×2 serial jacobian matrix in the y coordinates.

In order to compute the dynamic model let us express it in the following form:

$$\boldsymbol{\tau} = \mathbf{ZZ}\ddot{\mathbf{q}} + \mathbf{B}^T\lambda + \mathbf{f} \tag{3.9}$$

$$\mathbf{w}_p = \mathbf{A}^T\boldsymbol{\lambda} = m\ddot{\mathbf{x}} \tag{3.10}$$

Changing it of coordinates:

$$\boldsymbol{\tau} = \mathbf{ZZ}\begin{bmatrix} \phi_1''(y)\dot{y}^2 + \phi_1'(y)\cancelto{0}{\ddot{y}} \\ \phi_2''(y)\dot{y}^2 + \phi_2'(y)\cancelto{0}{\ddot{y}} \end{bmatrix} + \mathbf{B}(y)^T\boldsymbol{\lambda} \tag{3.11}$$

$$\mathbf{w}_p = \mathbf{A}(y)^T\boldsymbol{\lambda} = m\ddot{\mathbf{x}}$$

Then, if the inverse dynamic model is multiplied by $\mathbf{J}(y)^{-T} = \mathbf{A}(y)^T\mathbf{B}(y)^{-T}$, it is obtained:

$$\mathbf{A}(y)^T\mathbf{B}(y)^{-T}\boldsymbol{\tau} = \mathbf{A}(y)^T\mathbf{B}(y)^{-T}\mathbf{Z}\mathbf{Z}\ddot{\mathbf{q}}(y) + \mathbf{A}(y)^T\boldsymbol{\lambda} + \mathbf{A}(y)^T\mathbf{B}(y)^{-T}\mathbf{f} \tag{3.12}$$

which can be rewritten by considering equation 3.13 as follows:

$$\mathbf{A}(y)^T\mathbf{B}(y)^{-T}\boldsymbol{\tau} = \mathbf{A}(y)^T\mathbf{B}(y)^{-T}\mathbf{Z}\mathbf{Z}\ddot{\mathbf{q}}(y) + m\ddot{\mathbf{x}} + \mathbf{A}(y)^T\mathbf{B}(y)^{-T}\mathbf{f} \tag{3.13}$$

Now in order to compute the direct second order kinematic model without degeneracy of the kinematic matrix $\mathbf{A}(y)^T$, only the information from $\ddot{x}$ is taking into account. This is done because of the fact that from the free dynamics it is known that for the controller at singularity $\ddot{y} = 0$. So, basically in the computation of the input torques we can take into account the fact that due to the free dynamics of the system, the second line of equation 3.13 is $m\ddot{y} = 0$, and then the computations are performed as follows:

$$\ddot{\mathbf{x}} = (-\mathbf{B}(y)^{-1}\mathbf{A}_s^T)^+(\ddot{\mathbf{q}}(y) - \mathbf{b}(y)) \tag{3.14}$$

where:
$\mathbf{A}_s = \begin{bmatrix} 1 & 1 \end{bmatrix}$
From: $\mathbf{w}_p = m\begin{bmatrix} \ddot{x} \\ \ddot{y} \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 0 & 0 \end{bmatrix}\begin{bmatrix} \lambda_{11} \\ \lambda_{21} \end{bmatrix}$
Then,
$\mathbf{b}(y) = -\mathbf{B}(y)^{-1}(\dot{\mathbf{A}}(y)\dot{\mathbf{x}} + \dot{\mathbf{B}}(y)\dot{\mathbf{q}}(y))$
And:
$\dot{\mathbf{x}} = (-\mathbf{B}(y)^{-1}\mathbf{A}(y))^+\dot{\mathbf{q}}(y)$

Furthermore, it is important to mention that by using the Pseudo-inverse in the second order kinematic model, the solution found for the second order kinematic model in equation 1.29 is the best approximated solution, but it is not always true. It means that if we look for an exact solution of the second order kinematic model, it would imply that the two lines of the second order kinematics would be equal.

Then, rearranging equation 3.13 and using the approximation of the second order kinematics, the following equation can be formulated:

$$\mathbf{A}_s\mathbf{B}(y)^{-T}\boldsymbol{\tau} = \mathbf{A}_s\mathbf{B}(y)^{-T}\mathbf{Z}\mathbf{Z}\ddot{\mathbf{q}}(y) - m(\mathbf{B}(y)^{-1}\mathbf{A}_s^T)^+(\ddot{\mathbf{q}}(y) - \mathbf{b}(y)) + \mathbf{A}(y)^T\mathbf{B}(y)^{-T}\mathbf{f} \tag{3.15}$$

And finally the inverse dynamic model can be computed as follows:

$$\boldsymbol{\tau} = (\mathbf{A}_s\mathbf{B}(y)^{-T})^+(\mathbf{A}_s\mathbf{B}(y)^{-T}\mathbf{Z}\mathbf{Z}\ddot{\mathbf{q}}(y) - m(\mathbf{B}(y)^{-1}\mathbf{A}_s^T)^+(\ddot{\mathbf{q}}(y) - \mathbf{b}(y)) + \mathbf{A}(y)^T\mathbf{B}(y)^{-T}\mathbf{f}) \tag{3.16}$$

The pseudo-inverse in equation 3.16 ensure the solution for the dynamic model that minimizes the input torques, and therefore any overconstraint in the system is rejected by using this exact solution for the dynamic model.

3.1.2 Input-Output Linearization

Once the VHC in the joint space has been modeled, the free dynamics has been included in the computation of the Dynamic model, and the information from ($\ddot{x}$) has been taken into account by the reduced kinematic Jacobian matrix $\mathbf{A}_s$. Then the inverse dynamic model has been proven to be consistent at the singularity locus.

Therefore, the challenge now is to synthesize a Virtual Constraint Controller for enforcement of the constraint ($h_i = q_{i1} - \phi_i(y)$). In order to be able to derive the control law, it was necessary to demonstrate consistency in the dynamic model. Thus, now it is necessary to recall the second derivative of the VHC, given that in the second differentiation, the control input appears:

$$\ddot{h}_i = \ddot{q}_{i1} - \phi_i^{''}(y)\dot{y}^2 \tag{3.17}$$

Then, by expressing the inverse dynamic model from equation (3.16), and substituting the auxiliary Proportional Derivative Controller 3.18 into this control input equation, it is possible to obtain after having substituted, the input-output linearization 3.19:

The control law for the enforcement of the VHC modeled in the joint space:

$$\boldsymbol{\upsilon}_1 = \ddot{\mathbf{q}} = -\mathbf{K}_p\mathbf{h} - \mathbf{K}_d\dot{\mathbf{h}} = \ddot{\mathbf{h}} \tag{3.18}$$

Control law with auxiliary law for virtual constraint enforcement:

$$\begin{aligned}\boldsymbol{\tau} = (\mathbf{A}_s\mathbf{B}(y)^{-T})^{+}((\mathbf{A}_s\mathbf{B}(y)^{-T}\mathbf{ZZ} - m(\mathbf{B}(y)^{-1}\mathbf{A}_s^T)^{+})\boldsymbol{\upsilon}_1 + \Phi^{''}(y)\dot{y}^2 \\ +m(\mathbf{B}(y)^{-1}\mathbf{A}_s^T)^{+}\mathbf{b}(y) + \mathbf{A}(y)^T\mathbf{B}(y)^{-T}\mathbf{f})\end{aligned} \tag{3.19}$$

where:

$\mathbf{K}_p$ is a positive definite matrix of proportional gains to be tuned;

$\mathbf{K}_d$ is a positive definite matrix of derivative gains to be tuned.

And:

$$\boldsymbol{\Phi(y) = [\phi_1(y), \phi_2(y)]^T}$$

$$\boldsymbol{\Phi^{'}(y) = [\phi_1^{'}(y), \phi_2^{'}(y)]^T}$$

$$\boldsymbol{\Phi^{''}(y) = [\phi_1^{''}(y), \phi_2^{''}(y)]^T}$$

Moreover, from the equation 3.19, it is important to mention that the auxiliary control law $\boldsymbol{\upsilon}_1$ comes from the approximation of the second order kinematics. Furthermore, this control law is only valid locally, in the singularity locus. To sum up the synthesized control law of this section of virtual constraint controller with joint space controlled variables, it is worth it to mention that since the second order kinematics represents an approximation, then we could express the auxiliary control law as $\boldsymbol{\upsilon}_1 = \ddot{\mathbf{q}} = -\mathbf{K}_p\mathbf{h} - \mathbf{K}_d\dot{\mathbf{h}} + \mathbf{e}(\mathbf{h}) = \ddot{\mathbf{h}}$ where $\mathbf{e}(\mathbf{h})$ can be thought as an error of approximation.

3.2. Virtual Constraint Controller: Part II (cartesian space controlled variables)

Similarly for synthesizing the Virtual Constraint Controller for the VHC expressed with cartesian space controlled variables, the challenge is to enforce $(x(y) = 0)$ by defining a feedback control law that takes into account the enforcement of this virtual constraint and also the overconstraint presented in the last subsection.

3.2.1 Computation of the Dynamic model in the Singularity locus

The control input is appearing in the second time derivative, thus it is necessary to differentiate twice the VHC expressed in the cartesian space. In this case the computation is trivial, since the constraint is $(x(y) = 0)$, then:

$$h = x(y) \tag{3.20}$$

$$\dot{h} = \dot{x}(y) \tag{3.21}$$

$$\ddot{h} = \ddot{x}(y) \tag{3.22}$$

In order to compute the dynamic model, let us express it in the following form:

$$\boldsymbol{\tau} = \mathbf{Z}\mathbf{Z}\ddot{\mathbf{q}} + \mathbf{B}^T\boldsymbol{\lambda} + \mathbf{f} \tag{3.23}$$

$$\mathbf{w}_p = \mathbf{A}^T\boldsymbol{\lambda} = m\ddot{\mathbf{x}} \tag{3.24}$$

Then, if the inverse dynamic model is multiplied by $\mathbf{J}^{-T} = \mathbf{A}^T\mathbf{B}^{-T}$, it is obtained:

$$\mathbf{A}^T\mathbf{B}^{-T}\boldsymbol{\tau} = \mathbf{A}^T\mathbf{B}^{-T}\mathbf{Z}\mathbf{Z}\ddot{\mathbf{q}} + \mathbf{A}^T\boldsymbol{\lambda} + \mathbf{A}^T\mathbf{B}^{-T}\mathbf{f} \tag{3.25}$$

which can be rewritten by considering equation 3.24 as follows:

$$\mathbf{A}^T\mathbf{B}^{-T}\boldsymbol{\tau} = \mathbf{A}^T\mathbf{B}^{-T}\mathbf{Z}\mathbf{Z}\ddot{\mathbf{q}} + m\ddot{\mathbf{x}} + \mathbf{A}^T\mathbf{B}^{-T}\mathbf{f} \tag{3.26}$$

From the free dynamics we know that ($\ddot{y} = 0$), and from equation 3.24 it is possible to know that ($m\ddot{x} = \lambda_{11} + \lambda_{21} = \mathbf{r}_1 + \mathbf{r}_2$), which implies that we can express the controlled dynamics from ($\ddot{x}$) by reducing the kinematic Jacobian matrix $\mathbf{A}^T$ as follows:
Free dynamics analysis gives:

$$\mathbf{w}_p = m\begin{bmatrix}\ddot{x}\\ \ddot{y}\end{bmatrix} = \begin{bmatrix}1 & 1\\ 0 & 0\end{bmatrix}\begin{bmatrix}\lambda_{11}\\ \lambda_{21}\end{bmatrix}$$

$$m\ddot{x} = \lambda_{11} + \lambda_{21}$$

$$m\ddot{y} = 0$$

And recalling the two generated colinear forces r_1 and r_2:

$$m\ddot{x} = \lambda_{11} + \lambda_{21} = \mathrm{r}_1 + \mathrm{r}_2$$

We will express:

$$\mathbf{A}_s = \begin{bmatrix}1 & 1\end{bmatrix} \tag{3.27}$$

Then, it is possible to express the dynamic model by using the reduced kinematic Jacobian matrix $\mathbf{A}_s$, which incorporates the information coming from $\ddot{x}$ and considers that $\ddot{y} = 0$ from the free dynamics, but knowing that the VHC ($x(y) = 0$) will be enforced through an auxiliary control action in order to consider the evolution of the controlled variable x according to the uncontrolled variable y. Then if we rearrange the dynamic model by considering this last analysis, thus:

$$\mathbf{A}_s\mathbf{B}^{-T}\boldsymbol{\tau} = \mathbf{A}_s\mathbf{B}^{-T}\mathbf{Z}\mathbf{Z}\ddot{\mathbf{q}} + m\ddot{x} + \mathbf{A}^T\mathbf{B}^{-T}\mathbf{f} \tag{3.28}$$

Then, by computing the inverse second order kinematic model in order to express the dynamic model in the cartesian space:

$$\ddot{\mathbf{q}} = -\mathbf{B}^{-1}(\mathbf{A}_s^T\ddot{x} + \dot{\mathbf{A}}\dot{\mathbf{x}} + \dot{\mathbf{B}}\dot{\mathbf{q}}) \tag{3.29}$$

which is equivalent to:

$$\ddot{\mathbf{q}} = \mathbf{J}_s^{-1}\ddot{x} + \mathbf{b} \tag{3.30}$$

where:
$\mathbf{J}_s^{-1} = -\mathbf{B}^{-1}\mathbf{A}_s^T$

$\mathbf{b} = -\mathbf{B}^{-1}(\dot{\mathbf{A}}\dot{\mathbf{x}} + \dot{\mathbf{B}}\dot{\mathbf{q}})$
And:
$\dot{\mathbf{q}} = -\mathbf{B}^{-1}\mathbf{A}\dot{\mathbf{x}}$
The dynamic model can be expressed:

$$\mathbf{A}_s\mathbf{B}^{-T}\boldsymbol{\tau} = -\mathbf{A}_s\mathbf{B}^{-T}\mathbf{Z}\mathbf{Z}(\mathbf{B}^{-1}\mathbf{A}_s^T\ddot{x} + \mathbf{b}) + m\ddot{x} + \mathbf{A}^T\mathbf{B}^{-T}\mathbf{f} \tag{3.31}$$

$$\mathbf{A}_s\mathbf{B}^{-T}\boldsymbol{\tau} = -\mathbf{A}_s\mathbf{B}^{-T}\mathbf{Z}\mathbf{Z}\mathbf{B}^{-1}\mathbf{A}_s^T\ddot{x} - \mathbf{A}_s\mathbf{B}^{-T}\mathbf{Z}\mathbf{Z}\mathbf{b} + m\ddot{x} + \mathbf{A}^T\mathbf{B}^{-T}\mathbf{f} \tag{3.32}$$

Rearranging the equation 3.32

$$\mathbf{A}_s\mathbf{B}^{-T}\boldsymbol{\tau} = (m - \mathbf{A}_s\mathbf{B}^{-T}\mathbf{Z}\mathbf{Z}\mathbf{B}^{-1}\mathbf{A}_s^T)\ddot{x} - \mathbf{A}_s\mathbf{B}^{-T}\mathbf{Z}\mathbf{Z}\mathbf{b} + \mathbf{A}^T\mathbf{B}^{-T}\mathbf{f}$$

The inverse dynamic model that minimizes the input torques at the singularity locus is expressed by using the Moore-Penrose pseudo-inverse:

$$\boldsymbol{\tau} = (\mathbf{A}_s\mathbf{B}^{-T})^{+}((m - \mathbf{A}_s\mathbf{B}^{-T}\mathbf{Z}\mathbf{Z}\mathbf{B}^{-1}\mathbf{A}_s^T)\ddot{x} - \mathbf{A}_s\mathbf{B}^{-T}\mathbf{Z}\mathbf{Z}\mathbf{b} + \mathbf{A}^T\mathbf{B}^{-T}\mathbf{f}) \tag{3.33}$$

From the computation of the dynamic model at the singularity locus by using virtual constraints in the cartesian space, it can be shown that no problems of approximation were ecountered. Moreover, the exact solution for inverse dynamic model minimizes the input torques.

3.2.2 Input-Output Linearization

Once the VHC in the cartesian space has been modeled, the free dynamics has been included in the computation of the Dynamic model, and the controlled dynamics from ($\ddot{x}$) has been taken into account by the reduced kinematic Jacobian matrix $\mathbf{A}_s$. Then the inverse dynamic model has been proven to be consistent at the singularity locus.

Therefore, the challenge now is to synthesize a Virtual Constraint Controller for enforcement of the constraint ($x(y) = 0$). In order to be able to derive the control law, it was necessary to demonstrate consistency in the dynamic model. Thus, now it is necessary to recall the second derivative of the VHC, given that in the second differentiation, the control input appears:

$$\ddot{h} = \ddot{x}(y)$$

Then, by expressing the direct dynamic model from equation (3.33), it is possible to obtain $\ddot{x}$, then by substituting into the equation 3.22, which is the second differentiation of the VHC, it is possible to obtain:

$$\ddot{h} = \ddot{x} = \frac{(\mathbf{A}_s\mathbf{B}^{-T}\boldsymbol{\tau} + \mathbf{A}_s\mathbf{B}^{-T}\mathbf{Z}\mathbf{Z}\mathbf{b} - \mathbf{A}^T\mathbf{B}^{-T}\mathbf{f})}{(m - \mathbf{A}_s\mathbf{B}^{-T}\mathbf{Z}\mathbf{Z}\mathbf{B}^{-1}\mathbf{A}_s^T)} \tag{3.34}$$

After having substituted, the input-output linearization is computed:

$$\boldsymbol{\tau} = (\mathbf{A}_s\mathbf{B}^{-T})^{+}((m - \mathbf{A}_s\mathbf{B}^{-T}\mathbf{ZZB}^{-1}\mathbf{A}_s^T)\upsilon_2 - \mathbf{A}_s\mathbf{B}^{-T}\mathbf{ZZb} + \mathbf{A}^T\mathbf{B}^{-T}\mathbf{f}) \tag{3.35}$$

where:

$$\upsilon_2 = \ddot{x} = -K_p x - K_d \dot{x} \tag{3.36}$$

And υ_2 is the control law that enforce the VHC.
where:
K_p is the proportinal gain;
K_d is the derivative gain.

As a final note in the derivation of the control input torques with auxiliary control law υ_2 for virtual constraint enforcement, there are two important remarks. The first one is that there are no problems of approximation in the computation of second order kinematics and the second one is that the solution for the inverse dynamic model is an exact solution that minimizes the input torques by using the pseudo-inverse.

3.3. Control Formulation for the Virtual Holonomic Constraint in the entire singularity locus

Let us depict the five-bar mechanism in singular position when the end-effector is not vertically aligned with the reference frame. Recalling the vector of passive joints given by $\mathbf{q}_d = [q_{12}, q_{22}, q_{13}]^T$ and the vector of passive joint velocities $\dot{\mathbf{q}}_d = [\dot{q}_{12}, \dot{q}_{22}, \dot{q}_{13}]^T$. Also the vector of moving platorm pose including the orientation of the end-effector is given by $\mathbf{x}_\phi = [x, y]^T$. From figure 3.1, it is possible to observe that in order to know the orientation of the end-effector with respect to the reference frame, it is simply necessary to know the value of the passive joints and then relate them with the end-effector. This is trivial due to the fact that the passive joints and the end-effector orientation can be measured both in the distal links. Thus, the passive joints can be computed as follows:

$$q_{12} = \text{atan2}(y - l\text{sin}(q_{11}), x + \frac{a}{2} - l\text{cos}(q_{11})) - q_{11} \tag{3.37}$$

$$q_{22} = \text{atan2}(y - l\text{sin}(q_{21}), x - \frac{a}{2} - l\text{cos}(q_{21})) - q_{21} \tag{3.38}$$

$$q_{13} = q_{21} + q_{22} - q_{11} - q_{12} \tag{3.39}$$

$$\beta = q_{11} + q_{12} + q_{13} \rightarrow \text{by using Leg 1} \tag{3.40}$$

$$\beta = q_{21} + q_{22} \rightarrow \text{by using Leg 2} \tag{3.41}$$

Then, in order to compute the inverse dynamic model to synthesized the input-output linearization for the Virtual Constraint Controller for the entire singularity locus for a crossing different than $(x(y) = 0)$, it is necessary to premultiply $\mathbf{w}_p$ by the rotation matrix around z and with angle β. Moreover, the reduced kinematic jacobian vector A_s must be used in order to take into account the information from the controlled dynamics x.

$$\mathbf{R}_z(\beta)\mathbf{W}_p = \mathbf{R}_z(\beta)\mathbf{A}_s^T\lambda \tag{3.42}$$

where:

$$\mathbf{R}_z(\beta) = \begin{bmatrix} \cos(\beta) & -\sin(\beta) \\ \sin(\beta) & \cos(\beta) \end{bmatrix} \tag{3.43}$$

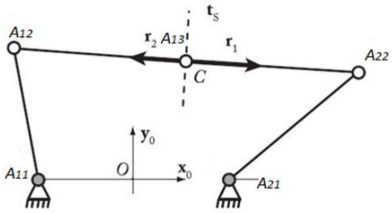

Figure 3.1: Singular Position when end effector frame is not aligned with the reference frame

The goal is to compute the rotation matrix and integrate it into the controlled and uncontrolled variables for developing the virtual constraints in the entire singularity locus. This is done to ensure the perpendicular alignment of the end-effector with respect to the uncontrollable motion wherever the robot is undergoing of the workspace. Thus, basically even if the robot is not vertically align with the reference frame at $(x(y) = 0)$, the premultiplication of the virtual constraints with the rotation matrix allows to always align the end-effector frame perpendicular to the direction of the uncontrollable motion which at $(x = 0)$ would be trivial because the end-effector frame is align with the reference frame, then $\beta = 0$.

In order to prove the aforementioned statement, let us consider the following computations:

At the singularity locus with the virtual constraint $(x = 0)$ and crossing the Type 2 singularity in the center $(\beta = 0)$, the numerical solution for $\mathbf{R}_z(\beta)\mathbf{A}_s^T$ is:

$$\begin{bmatrix} \cos(0) & -\sin(0) \\ \sin(0) & \cos(0) \end{bmatrix} \begin{bmatrix} \cos(0) \\ \cos(0) \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \mathbf{A}_s^T \tag{3.44}$$

Now, if we want to cross the Type 2 singularity with the same virtual constraint ($x = 0$), but at a different value of β, which it would mean that the robot does not cross in the center of the singularity locus, the solution would remain the same equal to $\mathbf{A}_s^T$.

Let us assume that the five-bar mechanism has deviation with respect the vertical alignment with the end-effector at the singular positions $\mathbf{x} = [-0.06745, 0.03989]$ which means $\beta = 0.4373\ rad = 25\ deg$, and indeed $(x \neq 0)$ as shown in figure 3.2.

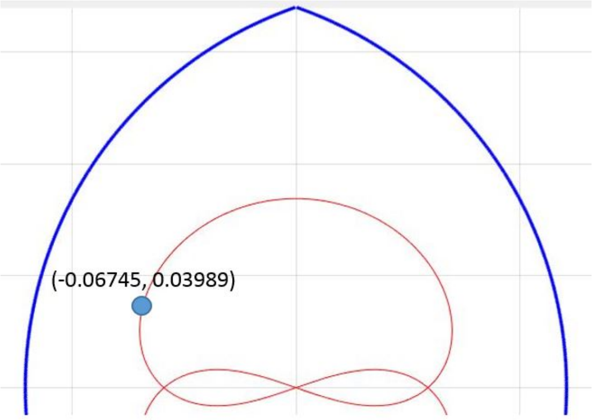

Figure 3.2: End-effector position at $x \neq 0$

Then let us prove that the solution for the first row will be always $\mathbf{A}_s^T = \begin{bmatrix} 1 & 1 \end{bmatrix}$.

$$\begin{aligned} &\begin{bmatrix} \cos(\beta) & -\sin(\beta) \\ \sin(\beta) & \cos(\beta) \end{bmatrix} \begin{bmatrix} \cos(q_{11}+q_{12}) & \cos(q_{21}+q_{22}) \\ \sin(q_{11}+q_{12}) & \sin(q_{21}+q_{22}) \end{bmatrix} \\ = &\begin{bmatrix} \cos(0.4373) & -\sin(0.4373) \\ \sin(0.4373) & \cos(0.4373) \end{bmatrix} \begin{bmatrix} \cos(3.1218-2.6846) & \cos(2.0933-4.7977) \\ \sin(3.1218-2.6846) & \sin(2.0933-4.7977) \end{bmatrix} \\ = &\begin{bmatrix} 0.9059 & -0.4235 \\ 0.4235 & 0.9059 \end{bmatrix} \begin{bmatrix} 0.9059 & 0.9060 \\ 0.4235 & 0.4235 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 0 & 0 \end{bmatrix} \end{aligned} \tag{3.45}$$

And then finally by taking the $\ddot{x}$ information for the computation of the reduced kinematic matrix $\mathbf{A}_s^T$, it is proved that by premultiplying by the rotation matrix, the uncontrollable direction is still perpendicular to the distal links in the singularity and the virtual constraint works and can be enforced in any part of the workspace.

The derived control law for extending the VHC to the entire workspace still follows the same methodology, with the only difference that the terms that group the wrench applied on the platform with information different from zero from $\ddot{\mathbf{x}}$ are premultiplied by the deduced Rotation matrix presented in this analysis.

3.4. Summary

In this chapter, the theoretical part of one of the main contributions of the present research work have been presented. First, the development of a feedback control law for the enforcement of the VHC modeled in the joint space has been presented. Then, the control law based on the dynamic model as well was synthesized for the VHC in the cartesian space. It is important to mention that since the controller developed through the VHC is dynamic-model-based, then consistency of the inverse and direct dynamic model was shown.

As a second remark of the present chapter, it could be seen that computationally, the modeling of the VHC in joint space was more complex. On the other hand, by expressing the VHC in the cartesian space, the control law is more trivial to rigorously prove consistency in the dynamic model.

Finally, there are three very important remarks to say in this chapter:

1. This control law has been developed for the particular virtual holonomic constraint when the end-effector frame and the reference frame are aligned, which means that the constraint is $x = 0$. Nevertheless an extension of the control law was presented by premultiplying the wrench applied on the platform by the Rotation matrix that takes into account the alignment of the end-effector through the information known from the passive joints.

2. The expression of the dynamic model derived from the VHC and then the Virtual Constraint Controller apply only in the singularity locus, which is the situation when the five-bar mechanism is underactuated since it gains an uncontrollable motion. Moreover, it is important to mention that the controller evolves in the cartesian space.

3. In order to control the robot behavior far from the singularity it is necessary to implement a different controller dedicated for the robot far from Type 2 singularities. This strategy will be presented in the next chapter, as well as, the integration of the two different control laws, far from the singularity and in the singularity locus.

Chapter 4

Multi-Controller scheme with Virtual Constraint Controller to cross Type 2 singularities

This chapter presents a dedicated controller for tracking far from the singularity based on the Computed Torque Control technique derived in the chapter 1. Afterwards, the integration of the two different control laws, including the Virtual Constraint Controller for crossing the Type 2 singularity is performed by using a Supervisory Control Architecture. Then, in order to have a smooth and continuous transition between control laws, a performance based supervisory block is designed. Moreover, an Adaptive Gain strategy is added in order to avoid discontinuities in the tracked variables of the Multi-Controller architecture. Finally, the results of Co-simulation are presented and also the experimentation in the real five-bar mechanism with non optimal trajectories. Also a validation of the Control Laws is performed via the extraction of the platform dynamics using an IMU. Furthermore, the resulted contributions of this chapter are emphasized including the Multi-Controller schemes by using VHC modeled in both spaces, the task and joint space. Also after this chapter the final conclusion of this research work is given by also pointing out future works for dedicated advanced controllers for singularity crossing.

4.1. Supervisory Control Architecture

In [20], the Supervisory Control Architectures are defined as schemes for switching in control systems. The reasons for switching in a Supervisory Control Architecture could vary . For instance, the nature of the control problem, sensor or actuator limitations, large modeling uncertainties or a combination of all of them. The types of supervision that have been discussed in the literature are:

- Prerouted supervision
- Performance-based supervision
- Estimator-based supervision

Considering the nature of the control problem presented in this research work, it seems to be a promising idea to implement a Multi-Controller scheme based on the Supervisory Control Architecture. This, in order to have a list of candidate controllers to be activated during the robot trajectory, i.e. controllers for the situation when the robot is undergoing far from Type 2 singularity and another specific controller based on virtual constraints when the robot is underactuated (around Type 2 singularity). (Figure 4.1)

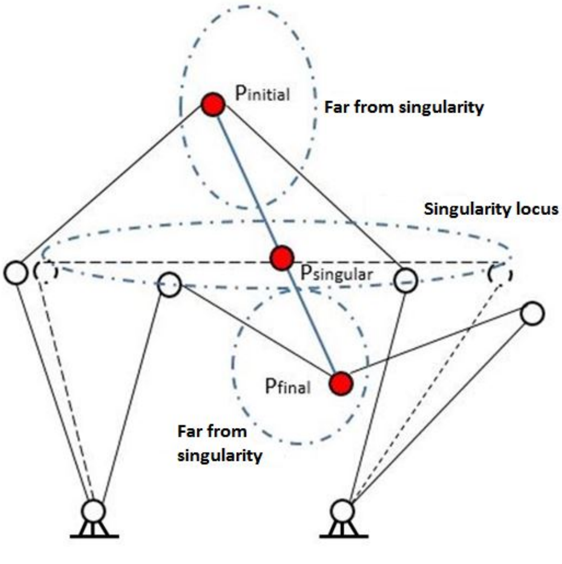

Figure 4.1: Robot trajectory from initial position to final position

Supervisory Control Architectures seems to be a promising technique in order to integrate a set of different candidate controllers in order to track continuously the reference trajectory and avoiding issues in terms of discontinuities in the tracked variables among the different candidate controllers.

One more interesting aspect of the flexibility that these schemes present is the fact that it allows to define the kind of supervision, according to the nature of the application. In the case of integrating a list of candidate controllers for crossing Type 2 singularities, it seems natural to choose a Performance based supervision, due to the fact that it has been shown that in the singularity locus the kinematic Jacobian matrix $\mathbf{A}^T$ is not numerically invertible. Thus, this could be taken as a performance index to be taken into account for desigining the Supervisory block. Moreover, by implementing performance-based supervision, all the decisions are made inside the Control Architecture.

There are four main challenges in order to synthesize a Multi-Controller scheme:

1. Define the controllers for the Multi-Controller Architecture
2. Synthesize the Multi-Controller Architecture
3. Design a performance-based supervisory block for the transition among the controllers
4. Avoid discontinuities of the tracking controlled variables when switching control laws

4.1.1 Integration of Candidate Controllers for non-singular assembly mode changing

In order to address the first challenge of choosing the candidate controllers for the Multi-Controller, it is necessary to recall the Control Laws synthesized in the chapter 1 and chapter 3.

In chapter 1 it has been shown that a Computed Torque Control law in the joint space is ideal for controlling far from the singularity locus with the full dynamic model, irrelevant if the trajectory is optimal or not, the control law is recalled from equation 1.24:

$$\mathbf{u}_1 = \boldsymbol{\tau} = \mathbf{M}(\ddot{\mathbf{q}}^d + \mathbf{K}_d(\dot{\mathbf{q}}^d - \dot{\mathbf{q}}) + \mathbf{K}_p(\mathbf{q}^d - \mathbf{q})) + \mathbf{H}(\mathbf{q}, \dot{\mathbf{q}})$$

Now, for controlling in the singularity locus, the two control laws derived in the chapter 3 by using VHC in both spaces will be used, recalling the control law derived by using VHC modeled with joint space controlled variables from equations (3.19 and 3.18):

$$\begin{aligned}\mathbf{u}_2 = \boldsymbol{\tau} = (\mathbf{A}_s\mathbf{B}(y)^{-T})^{+}((\mathbf{A}_s\mathbf{B}(y)^{-T}\mathbf{Z}\mathbf{Z} - m(\mathbf{B}(y)^{-1}\mathbf{A}_s^T)^{+})\boldsymbol{v}_1 + \Phi^{''}(y)\dot{y}^2 \\ +m(\mathbf{B}(y)^{-1}\mathbf{A}_s^T)^{+}\mathbf{b}(y) + \mathbf{A}(y)^T\mathbf{B}(y)^{-T}\mathbf{f})\end{aligned}$$

where the control law for the enforcement of the VHC modeled in the joint space:

$$\boldsymbol{\upsilon}_1 = \ddot{\mathbf{q}} = -\mathbf{K}_p\mathbf{h} - \mathbf{K}_d\dot{\mathbf{h}} = \ddot{\mathbf{h}}$$

And then, recalling the control law derived by using VHC modeled with cartesian space variable ($h = x(y)$) from (3.35 and 3.36):

$$\mathbf{u}_3 = \boldsymbol{\tau} = (\mathbf{A}_s\mathbf{B}^{-T})^{+}((m - \mathbf{A}_s\mathbf{B}^{-T}\mathbf{ZZB}^{-1}\mathbf{A}_s^T)\upsilon_2 - \mathbf{A}_s\mathbf{B}^{-T}\mathbf{ZZb} + \mathbf{A}^T\mathbf{B}^{-T}\mathbf{f})$$

where:

$$\upsilon_2 = \ddot{x} = -K_p h - K_d\dot{h}$$

Having defined the control laws for tracking the robot trajectory, the previous list of three candidate control laws will be integrated in a supervisory scheme. First by using the Virtual Constrint Controller modeled by using VHC with joint space controlled variables, and then with the one modeled with the cartesian space controlled variable. Thus, the supervisory schemes will be grouped as follows:

1. Candidate Controllers (Figure 4.2):
   Computed Torque Control ($\mathbf{u}_1$): controller in the joint space for undergoing far from Type 2 Singularity (Matrix $\mathbf{A}^T$ is numerically invertible);
   Virtual Constraint Controller ($\mathbf{u}_2$): controller in the task space modeled by using VHC with joint space controlled variables for undergoing around Type 2 Singularity.

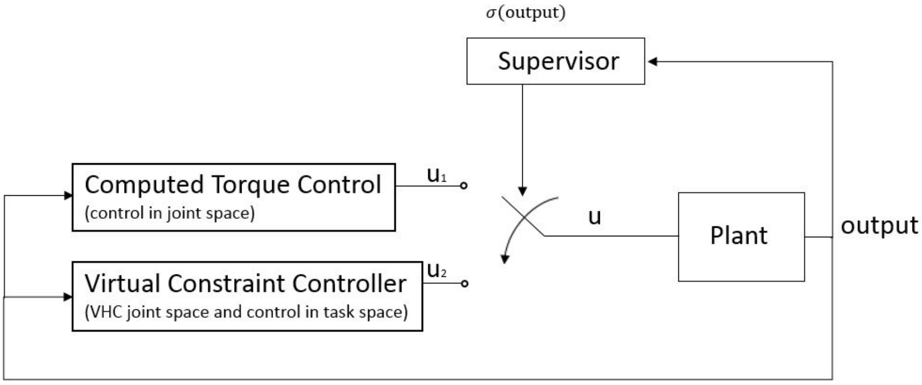

Figure 4.2: Supervisory Control scheme for first list of Candidate Controllers

2. Candidate Controllers (Figure 4.3):
   Computed Torque Control ($\mathbf{u}_1$): controller in the joint space for undergoing far from Type 2 Singularity (Matrix $\mathbf{A}^T$ is numerically invertible);

Virtual Constraint Controller ($\mathbf{u}_3$): controller in the task space modeled by using VHC with cartesian space controlled variables for undergoing around Type 2 Singularity.

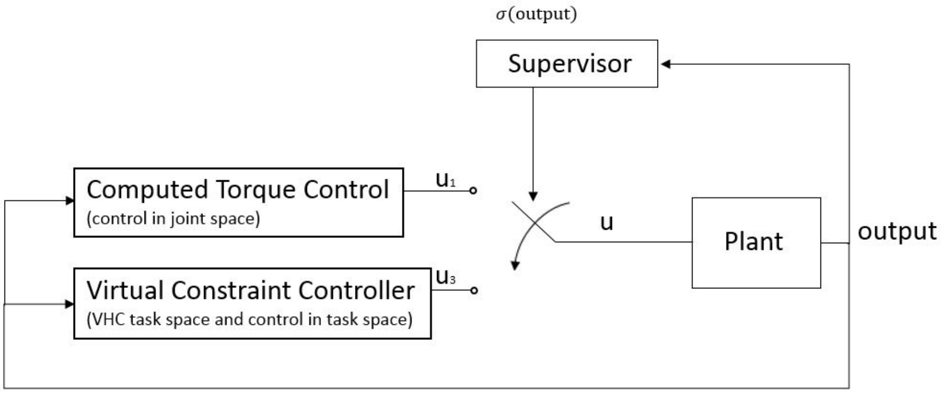

Figure 4.3: Supervisory Control scheme for second list of Candidate Controllers

4.1.2 Performance-based supervisory block

In order to design a performance-based supervisory block for the multi-controller architecture that takes decisions on the control laws transition, it is necessary to define a performance index to be taken into account in order to switch the control laws. The performance index to be modeled in the Supervisory block is the proximity to the singularity locus by computing the condition number of the matrix cond($\mathbf{A}$), which is the kinematic matrix that degenerates in the singularity locus.

Thus, the performance based supervisory block is defined as follows:
σ is equal to 0 far from the singularity and equal to 1 in the singularity locus.

- $\sigma = 1$, if $\left(\frac{1}{\text{cond}(\mathbf{A})}\right) \geq 2\epsilon$
- $\sigma = \left(\frac{1-\text{cond}(\mathbf{A})\epsilon}{\text{cond}(\mathbf{A})\epsilon}\right)$, if $\epsilon < \left(\frac{1}{\text{cond}(\mathbf{A})}\right) < 2\epsilon$
- $\sigma = 0$, if $\left(\frac{1}{\text{cond}(\mathbf{A})}\right) \leq \epsilon$

ϵ is a threshold to be tuned experimentally.

Finally, for the first list of candidate controllers in figure 4.2, the computation of the control laws $\mathbf{u}$ and its transition will be stated by the following equation:

$$\mathbf{u} = (1-\sigma)\mathbf{u}_1 + \sigma\mathbf{u}_2 \tag{4.1}$$

And similarly for the second list of candidate controllers in figure 4.3:

$$\mathbf{u} = (1-\sigma)\mathbf{u}_1 + \sigma\mathbf{u}_3 \tag{4.2}$$

4.1.3 Adaptive Gain Tuning Strategy

One main issue when implementing Multi-Controller architectures is that the variables to be tracked may or may not be in the same space, being in the joint space or in the task space. Thus, when switching control laws a main issue is how to avoid discontinuities in order to have smooth and continuous tracking. This is the case of the Multi-Controller developed in the present research work, the Computed Torque Control has been implemented in the joint space, while the Virtual Constraint Controller, in the task space. In order to deal with the problem of discontinuities when switching control laws, even if the performance-based supervisory block has been implemented, an error-based Adaptive Gain algorithm is performed as follows:

- Initialize Proportional and Derivative Gains K_p, K_d
- Get $\text{Error}_{taskspace} = |\text{Error}_{x_{CTC}} - \text{Error}_{x_{VHC}}| \leqslant \text{error}_{lim}$ and $\text{Error}_{taskspace_2} = |\text{Error}_{y_{CTC}} - \text{Error}_{y_{VHC}}| \leqslant \text{error}_{lim}$
- Get $\text{Error}_{output} = |x_{CTC} - x_{VHC}| \leqslant \text{error}_{lim}$ and $\text{Error}_{output_2} = |y_{CTC} - y_{VHC}| \leqslant \text{error}_{lim}$ when switching from Virtual Constraint Controller to Computed Torque Controller and viceversa

```
– if → $|\text{Error}_{taskspace} - \text{Error}_{output}| \leqslant \text{error}_{lim} \;||\; |\text{Error}_{taskspace_2} - \text{Error}_{output_2}| \leqslant \text{error}_{lim}$
    $\text{K}_{p_{gain}} = \text{K}_p$
    $\text{K}_{d_{gain}} = \text{K}_d$
– else →
    $\text{K}_{p_{gain}} = \text{K}_p\alpha$
    $\text{K}_{d_{gain}} = \text{K}_d\alpha$
```

where α and error_{lim} are thresholds to be set experimentally.

4.2. Results of Simulations

In order to validate the theoretical formulations derived for the control laws far from the singularity and in the singularity locus, presented in chapters 3 and 4, tests are carried out on the five-bar mechanism DexTAR.

Let us define a non-optimal trajectory by using a fifth order polynomial which can fix the position, velocity and acceleration of the five bar mechanism at the trajectory extremities ($P_{initial}$ and P_{final}), (Figure 4.4). This means that in the trajectory there is not information regarding the singular position. Moreover, the controller must be able to bring the end-effector from $P_{initial}$ to P_{final}, irrelevant if the singular position must be met during this assembly mode changing.

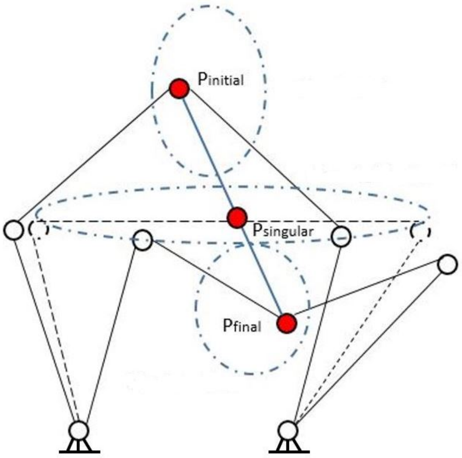

Figure 4.4: Only $P_{initial}$ and P_{final} given for the trajectory planning

The first non-optimal trajectory will be defined by giving initial and final points where $x = 0$, this is to validate the enforcement of the VHC, which means that both multi-controller schemes must be able to track the motion of the five bar mechanism in the vertical axis, which would mean to cross in the middle.
Let us define the first trajectory between two points $P_{initial}(\mathbf{x}_{P_{initial}} = [x_{P_{initial}}, y_{P_{initial}}] = [0, 0.15]m)$ and $P_{final}(\mathbf{x}_{P_{final}} = [x_{P_{final}}, y_{P_{final}}] = [0, 0.02]m)$. The trajectory is a return motion (figure 4.5), it means that first it goes from $P_{initial}$ to P_{final} and return again to $P_{initial}$. Therefore, the robot will cross twice the singularity.

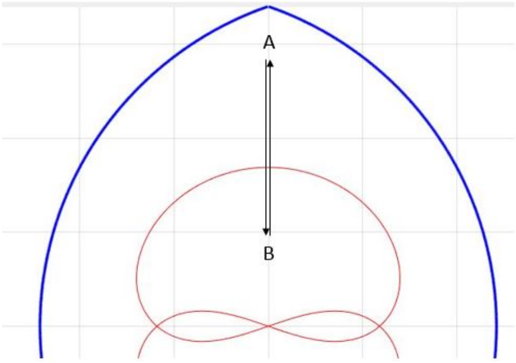

Figure 4.5: Middle crossing trajectory with the sequence: $A \rightarrow B \rightarrow A$

Finally for generating the polynomials, the initial and final conditions are:

$$x(t_0) = x_{P_{initial}}; \dot{x}(t_0) = 0; \ddot{x}(t_0) = 0 \tag{4.3}$$

$$y(t_0) = y_{P_{initial}}; \dot{y}(t_0) = 0; \ddot{y}(t_0) = 0 \tag{4.4}$$

$$x(t_f) = x_{P_{final}}; \dot{x}(t_f) = 0; \ddot{x}(t_f) = 0 \tag{4.5}$$

$$y(t_f) = y_{P_{final}}; \dot{y}(t_f) = 0; \ddot{y}(t_f) = 0 \tag{4.6}$$

It is important to mention that no condition for singularity is included.

Let us first simulate the behavior of the robot when using the Multi-Controller scheme for the first list of candidate controllers, which use the Virtual Constraint Control modeled by using the the VHC with joint space controlled variables.

The figures 4.6 and 4.7 show the results in terms of the input torques and the tracking error of the VHC expressed through the joint space controlled variable $q_{i1} = \phi_i(y)$. It can be seen that for the non-optimal trajectory given as a reference to the Multi-Controller scheme of the first list of candidate controlles, the robot can cross the singularity with finite torques. Moreover, it can be seen that the tracking error of the VHC is almost zero, having an error of order 10^{-4}. Finally, it is important to mention that also the performance-based supervisory block works for switching between the control laws. An important last remark is that, for this set of candidate controllers, the use of Adaptive Gain was necessary due to the fact of uncertainties in the computation of the dynamic model when doing the input-output linearization with VHC in joint space and therefore the model is less accurate with respect to the real robot.

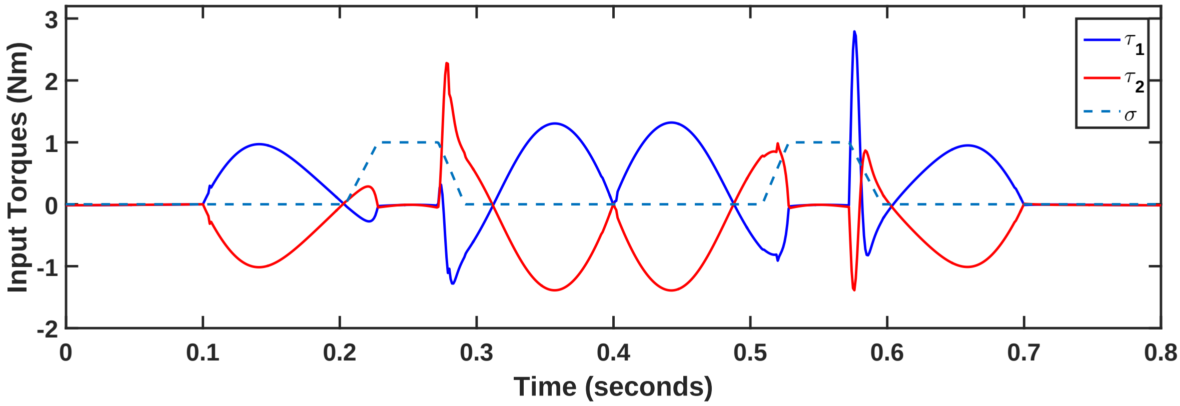

Figure 4.6: Evolution of the Torques with the Supervisory Control scheme for the first list of candidate controllers

Now let us simulate the behavior of the robot when using the Multi-Controller scheme without Adaptive Gain for the second list of candidate controllers, which use the Virtual Constraint Control modeled by using the VHC with the cartesian space controlled variable.

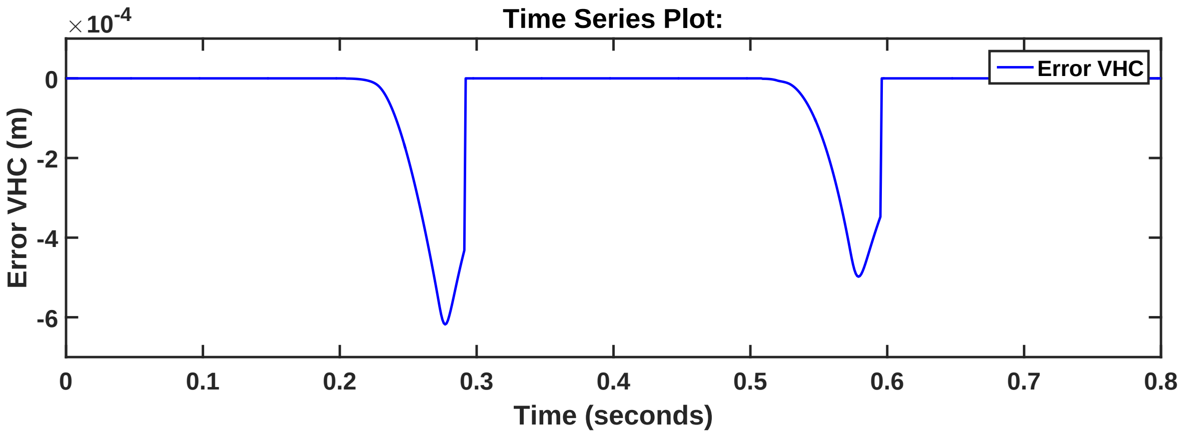

Figure 4.7: Error of the VHC $q_{i1} = \phi_i(y) \rightarrow x(y)$

The figures 4.8 and 4.9 show the results in terms of the input torques and the tracking error of the VHC expressed through the cartesian space controlled variable x. It can be seen that for the non-optimal trajectory given as a reference to the Multi-Controller scheme of the second list of candidate controllers, the robot can cross the singularity with finite torques. Moreover, it can be seen that the tracking error of the VHC is almost zero, having an error of order 10^{-4}. Finally, it is important to mention that also the performance-based supervisory block works for switching between the control laws. An important last remark is that, for this set of candidate controllers, the use of Adaptive Gain was not necessary due to the fact that the consistency in the dynamic model for the input-output linearization with VHC in cartesian space is more rigorous mathematically and therefore the model is more accurate, contrary to the VHC expressed with joint space controlled variables, where the Adaptive Gain was needed.

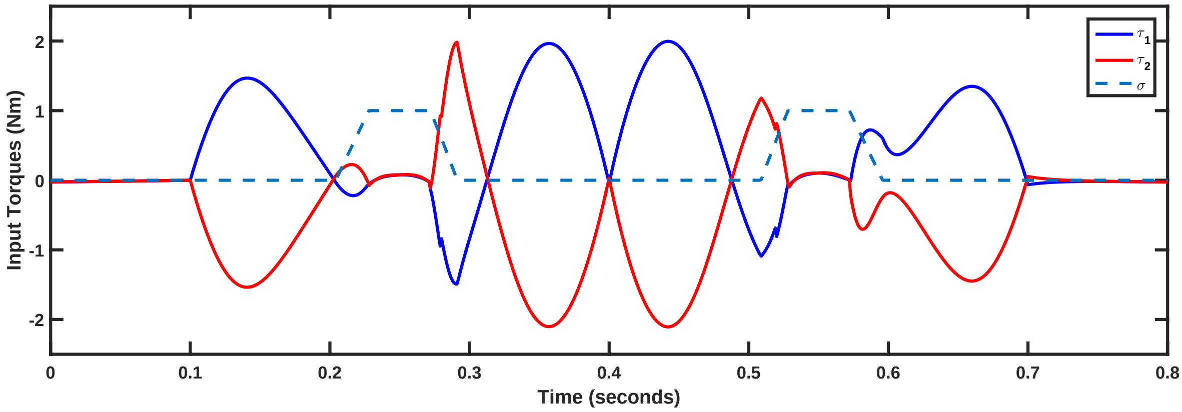

Figure 4.8: Evolution of the Torques with the Supervisory Control scheme for the second list of candidate controllers

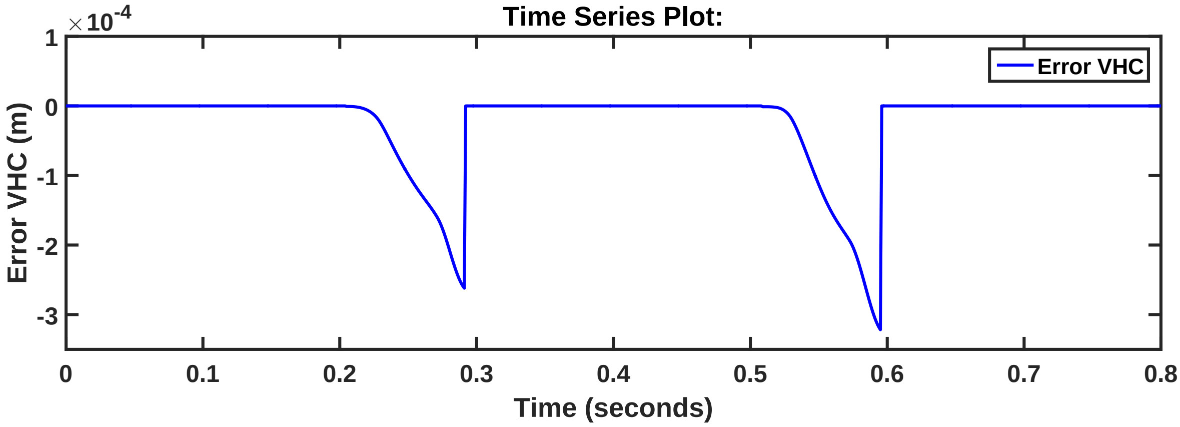

Figure 4.9: Error of the VHC $(x(y))$

Finally, let us define a non-optimal trajectory with multiple crossings to test the derived Virtual Constraint Controller with model of VHC extend it to the entire singularity locus by using the rotation matrix presented in chapter 3. In order to test this Controller, we are going to use the Multi-Controller scheme for the second list of candidate controllers. This is done because the cartesian space controlled variables are computationally less complex to use and more accurate in terms of its dynamic model.

Thus, let us define initial and final points for the multiple crossing non-optimal trajectory as follows: $A = [0, 0.15]$, $B = [0, 0.035]$, $C = [-0.025, 0.025]$, $D = [-0.025, 0.125]$, $E = [0.025, 0.025]$, $F = [0.025, 0.125]$. And the sequence of the trajectory is defined as the figure 4.10 depicts. The results of the non-optimal multiple crossing trajectory are shown in figures 4.11 and 4.12. It can be seen that the robot can cross the singularity with finite torques. Moreover, the tracking error of the VHC is of order 10^{-3}.

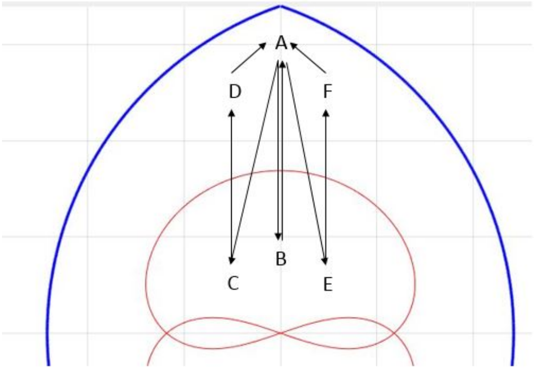

Figure 4.10: Multiple crossing trajectory with the sequence: $A \rightarrow B \rightarrow A \rightarrow B \rightarrow A \rightarrow C \rightarrow D \rightarrow A \rightarrow E \rightarrow F \rightarrow A$

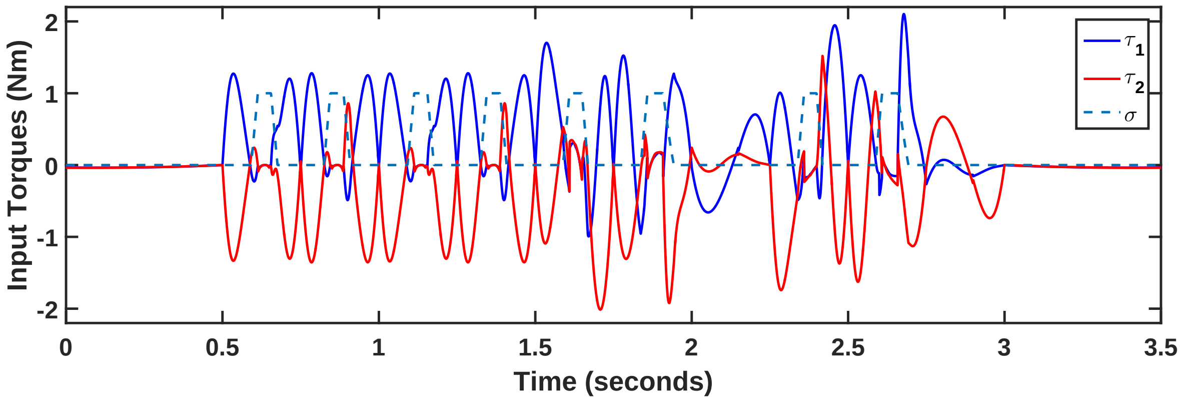

Figure 4.11: Evolution of the Torques with the Supervisory Control scheme for the second list of candidate controllers for a multiple crossing non-optimal trajectory

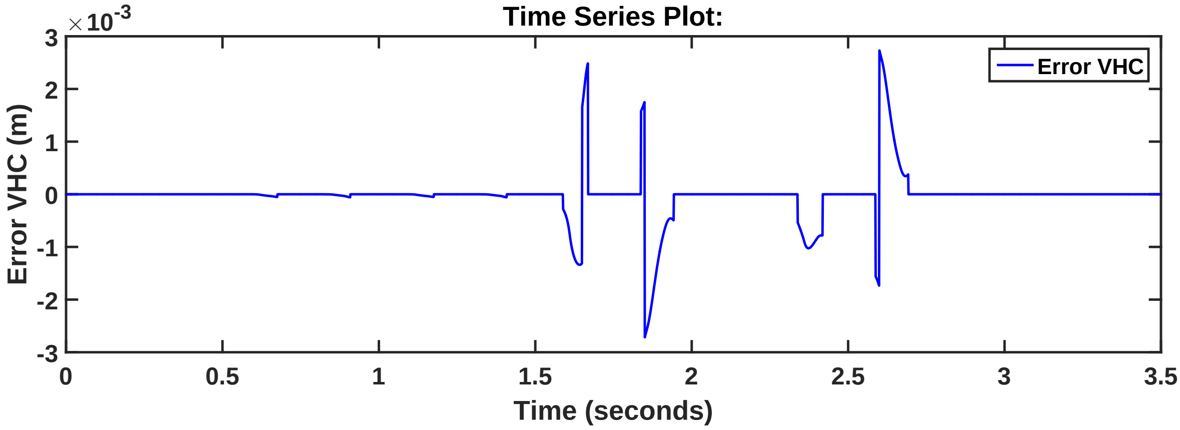

Figure 4.12: Error of the VHC ($x(y)$) multiple crossing trajetory

4.3. Results of experimentation

Now, once the two list of candidate controllers in the Multi-Controller scheme have been satisfactory validated, it has been shown that due to rigorous modeling aspects, the VHC modeled in the cartesian space shows mathematically more consistency in the dynamic model and there is no need of adaptive gain. Whereas in the controller with VHC modeled in the joints space, the adaptive gain is necessary.

Now, let us load the middle crossing fifth oder polynomial trajectory into the real five-bar mechanism DexTAR in order to test the Multi-Controller schemes. The sequence of the trajectory was depicted in figure 4.5.

In figure 4.13 it can be observed that for the middle crossing non-optimal trajectory and the second list of candidate controllers, the robot can cross the singularity with finite torques.

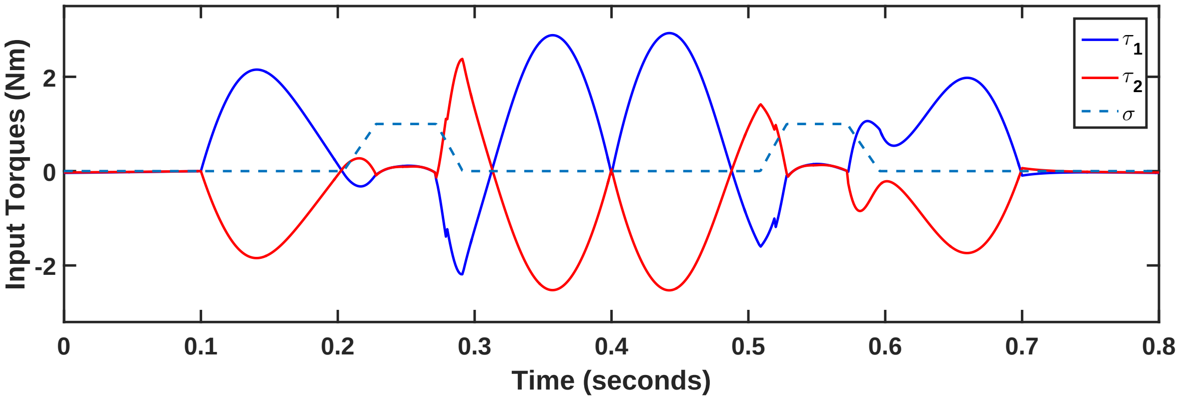

Figure 4.13: Input torque for the first and second actuator of the robot DexTAR crossing Type 2 singularity locus with second Multi-Controller tracking non-optimal trajectory

Now, in order to validate that the trajectory is actually being correctly tracked by the Multi-Controller scheme, the platform dynamics are extracted through an Inertial Measurement Unit shown in figure 4.14. Then, a comparisson between the measurement of the robot acceleration and the desired one is shown in figure 4.15 by depicting the tracking error of the magnitude of the acceleration.

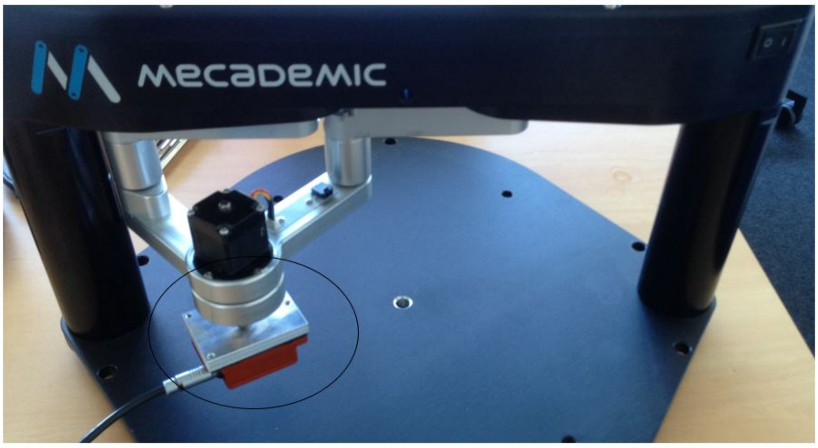

Figure 4.14: *IMU* attached to the end effector of the robot DexTAR

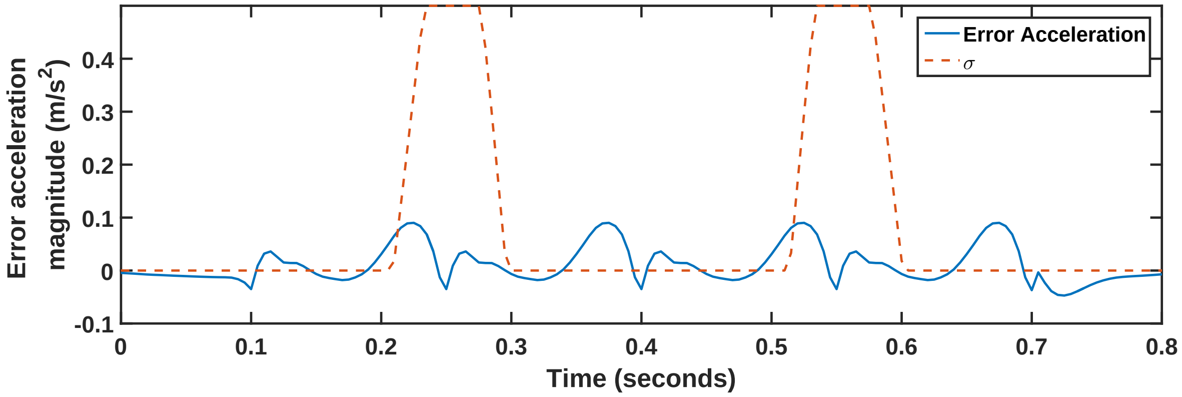

Figure 4.15: Tracking error of the magnitude of the acceleration for the Multi-Controller using VHC with the cartesian space variable $(x(y))$.

Finally, the test of multiple crossings trajectory are carried out in the real robot DexTAR. The trajectory and sequence of the points to be reached in the trajectory were depicted in figure 4.10. In figure 4.16, it can be observed that for the multiple crossings non-optimal trajectory and the second list of candidate controllers, the robot can cross the singularity with finite torques multiple times. Moreover, it is shown that the supervisory block change to virtual constraint controller for the singularity crossings and also the robot is able to cross in the workspace where $(x \neq 0)$. Then, in figure 4.17, the tracking error of the magnitude of the acceleration extracted from the IMU shows convergence towards zero. It is worth to notice that the continuous variation in the acceleration may be due to assembly issues of the external sensor in the end-effector. Nevertheless, the Multi-Controller and specially the Virtual Constraint Controller is able to track the reference acceleration.

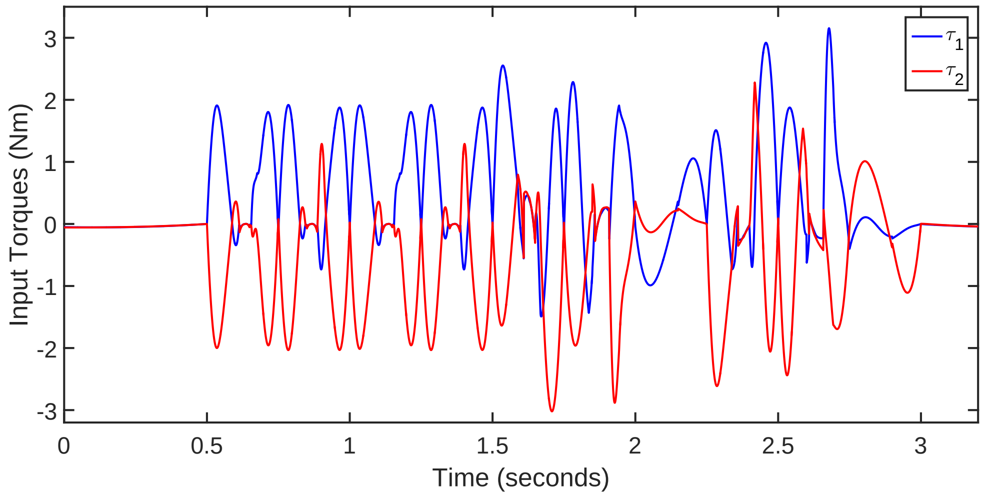

Figure 4.16: Input torque for the first actuator and second actuator of the robot DexTAR crossing Type 2 singularity locus with second Multi-Controller tracking multiple crossings of non-optimal trajectory

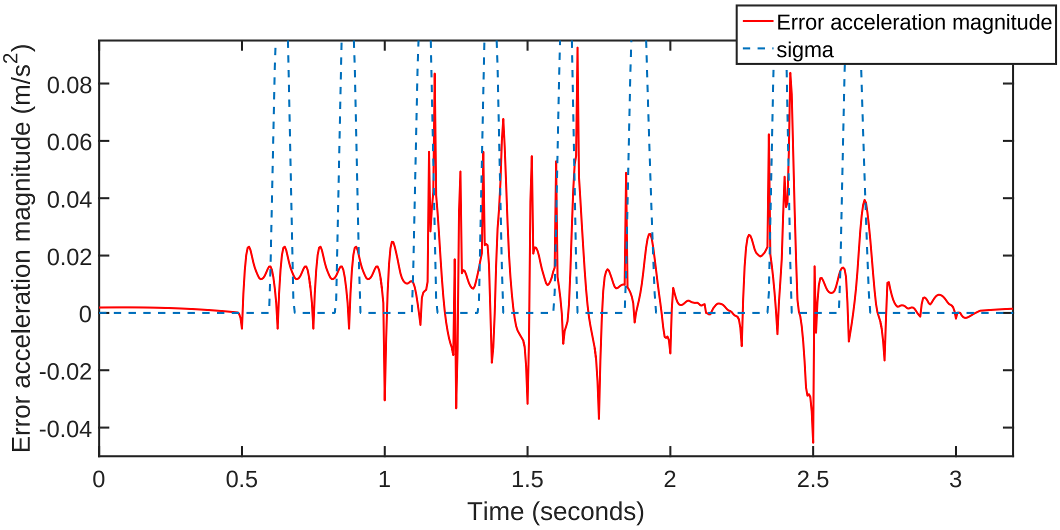

Figure 4.17: Tracking error of the magnitude of the acceleration for the Multi-Controller using VHC with the cartesian space variable $(x(y))$ for multiple crossings in the workspace.

4.4. Summary

In this last chapter the main results of the present research work were presented. The implementation of the theoretical results from chapter 2 and chapter 3 were carried out by simulation and experimentation. The simulation was performed by Co-Simulation between MATLAB and ADAMS. It was seen that in the simulation for non-optimal trajectory for crossing in the center of the singularity, which means test the VHC $(x = 0)$, the results were satisfactory since the torques at singularity locus were finite and moreover the tracking error of the VHC was minimum. Then, it was shown that experimentally the implementation of the Virtual Constraint Controller in the Multi-Controller Architecture worked, which also was cross-validated through an Inertial Measurement Unit.

Finally, for extending the synthesized controller to the entire singularity locus, the use of the Rotation matrix presented in chapter 3 was implemented in the controller. The results showed that the robot in simulation and experimentally in the real environment was able to cross Type 2 singularities with finite torques and tracking non-optimal trajectories, which is the main contribution of the present research work. Moreover, the cross-validation of the Multi-Controllers were performed by the addition of an external sensor in the end-effector of the five-bar mechanism in order to extract the platform dynamics and then make a comparisson of the desired accelerations with the measured ones from the sensor. It was shown that the tracking error of the acceleration was minimum proving that the Virtual Constraint Controller synthesized for the singularity locus was working as the theoretical and simulated results shown. It is important to mention that also in the next section of Conclusion, the main contributions of this research work are recalled and future works on dedicated advanced controllers for singularity crossing are mentioned.

Conclusion

Summary and resulted Contributions of this thesis.

The topic of this thesis was to develop an automatic dynamic controller in order to cross Type 2 singularities. Moreover, the most important requirement for this task was that the crossing would need to be achieved by considering non-optimal trajectories. It means that the dynamic criterion would not be included in the trajectory generation. Thus, given an initial point in one assembly mode and a final point in a different assembly mode, the controller would be able to achieve the trajectory tracking between these two points without having any information regarding the position at which the singularity would be met.

The first chapter of the present research work gave a general description and definition of parallel robots including its advantages, such as high paypload, small inertia, high speeds and accelerations, and also the drawbacks, like abundant singularity problems that reduce the reachable workspace. In addition to that, an insight of what changing assembly mode for parallel robots was given, this included a general explanation of its consequences in terms of the constraining Type 2 singularities when performing this change in assembly modes. Then, also in this chapter, a general conceptual description of the three types of singularities that exist in parallel mechanisms was given.

Also in chapter 1 the modeling of dynamics in parallel robots was addressed, first by explaining a general methodology including the analysis of virtual tree structures and then lagrange multipliers for closing the kinematic chains.

Then, the Parallel or Type 2 Singularities were explained in a technical and conceptual approach. This is done by recalling the Degeneracy conditions of the dynamic model due to the kinematic matrix **A**. Once the conditions of degeneracy of the dynamic model were stated, an overview on the crossing Type 2 singularities methods was discussed. Firstly by recalling the dynamic criterion that needs to be respected by implementing optimal trajectories, and then by discussing the Multi-Model Computed Torque Control proposed in the literature. So far, the first chapter of this research work gave preliminary and very important concepts and technical aspects that were needed for the evolution of the following chapters which present the main contributions of the present research work.

In chapter 2, the case of study of the present research work was a five-bar mechanism (robot DexTAR), thus in this chapter the dynamic model of this parallel mechanism was deduced, including the identification of the dynamic parameters and then performing a validation by doing Co-simulation ADAMS/MATLAB. In addition to that the robot DexTAR was computerized and then exported to ADAMS in order to do cross-validation of the mathematical model and the ADAMS plant. The results were satisfactory in the sense that the error in the dynamic model was minimum. Moreover, the ADAMS Mock-up was useful for further testings in the control development part. Then, two main contributions of the present research were performed, modeling the free dynamics and the VHC. This chapter has presented the theoretical and conceptual analysis for the identification of the controlled and uncontrolled variables at the singularity locus. This study was done by analyzing the underactuation happening in the five-bar mechanism in the singularity locus. From the wrench applied on the platform, the free dynamics or evolution of the uncontrolled variable was stated and moreover its relation or equivalency with the dynamic criterion was given. The VHC, modeled from the analysis of the free dynamics, was presented and implemented in the two spaces, by using joint space controlled variables and cartesian space controlled variables.

Chapter 3 and 4 are the main contributions in terms of the development of the automatic dynamic controller for crossing Type 2 singularities. Chapter 5 synthesized an Input-Output linearization in order to develop the Virtual Constraint Controller needed for enforcing the VHC at singularity. In addition to that, it has shown the rigorous mathematical consistency of the dynamic model in order to compute the inverse dynamic model and the direct dynamic model to feedback linearize the system and generate the control law for enforcing the VHC in both spaces. Finally in chapter 6, two Multi-Controller schemes were synthesized based on performance-based Supervisory Control techniques. Moreover, two list of candidate controllers for a continuous trajectory tracking far from the singularity and in the singularity locus were integrated in the two Multi-Controller schemes. In addition to that, to ensure avoiding discontinuities in the controlled variables, the addition of an Adaptive Gain strategy was added in the Controllers. Finally, the most important results are that the Multi-Controller schemes by using VHC in the joint space and in the cartesian space demonstrated continuous tracking of non-optimal trajectories.

Future works.

The future researh on the development of Advanced Controller techniques for dealing with the underactuation at the Type 2 singularities, it would be interesting to take the Virtual Constraint Controller developed in this research work and extend it to a controller that could work not only in the singularity locus, but also far from the singularity. And then, make a comparisson in terms of performance with respect to the Multi-Controller Architecture presented in this research work. Comparing performance capabilities, such as workspace reachability, stability analyses and rigorous dynamic model consistency.

In terms of the validation of the presented Multi-Controller architectures, an Inertial Measurement Unit has been used. Nevertheless, it would be interesting to implement a visual servoing scheme and validate the singularity crossing by using external camera information. Moreover, an interesting issue to deal with, it would be to develop the driver and interface between the external sensor and the software architecture already developed for the robot DexTAR.

Another line of research related to the Virtual Constraint approach that it would be interesting to study, is the development of a virtual holonomic constraint that it could be valid to enforce even if the robot is not underactuated, which it would be related to have a single controller working for the entire workspace by extending the approach of Multi-Controllers presented in this work.

Finally, it could be really ambitious and interesting to synthesize a dedicated controller to deal with the three type of singularities, which it would make the robot to be more intelligent when approaching to the singularity locus, irrelevant of the singularity which it is approaching. Typically in order to avoid Type 1 singularities in parallel robots, controllers in joint space are synthesized which means the kinematic matrix **B** would not degenerate. However, extending the concept of virtual constraints to the three type of singularities seems to be an interesting challenge. Moreover, supervisory schemes seems to be promising techniques due to the fact that performance indexes coming from the dynamic model can be used for taking decisions on-line when the robot is performing any given task, and these performances can be integrated in supervisory blocks for switching control laws depending on the singularity that the robot would approach.

Bibliography

[1] Merlet, J-P. Les robots parallèles. 2nd ed. rev. et augm. Paris: Hermès, 1997, 304 p. (Collection Robotique). ISBN 2-86601-599-1.

[2] Pagis, G.: Augmentation de la taille de l'espace de travail opérationnel des robots parallèles en traversant les singularités de Type 2. PhD thesis, Nantes, January 2015.

[3] Chablat, D., Wenger Ph.: Working modes and aspects in fully-parallel manipulator. In Proceedings of International Conference on Robotics and Automation, May 1998, Bruxelles, Belgium, pp. 1964-1969.

[4] Chablat, D., Wenger Ph.: On the characterization of the regions of feasible trajectories in the workspace of parallel manipulators. Tenth World Congress On The Theory Of Machines and Mechanisms, June 1999, Oulu, Finland. IFTOMM, pp.1-6.

[5] Gosselin, C. and Angeles, J.: Singularity analysis of closed-loop kinematic chains. IEEE Transactions On Robotics And Automation, Vol. 6, No. 3, pp. 281-290, June 1990.

[6] Pagis, G., Bouton N., Briot S., Martinet P.: Enlarging parallel robot workspace through type-2 singularity crossing. in Control Engineering Practice, Vol. 39, pp. 1-11, June 2015.

[7] Ibrahim, O. and Khalil,W. (2010). Inverse and direct dynamic models of hybrid robots. Mechanism and Machine Theory, 45, pp. 627-640.

[8] Briot, S., Gautier, M. (2012). Global Identification of Drive Gains and Dynamic Parameters of ParallelRobots – Part1: Theory, Proceedings of the 19th CISM-IFToMM Symposium on Robot Design, Dynamics, and Control (ROMANSY 2012), 2012.

[9] M. Gautier, W. Khalil, and P. P. Restrepo, “Identification of the dynamic parameters of a closed loop robot,” in Proceedings of IEEE International Conference on Robotics and Automation, Nagoya, Aichi, Japan, 1995, pp. 3045-3050.

[10] P. Renaud, A. Vivas, N. Andreff, P. Poignet, P. Martinet, F. Pierrot, and O. Company, “Kinematic and dynamic identification of parallel mechanisms,” Control Engineering Practice, vol. 14, no. 9, pp. 1099-1109.

[11] Briot, S., Arakelian, V.: On the dynamic properties of rigid-link flexible-joint parallel manipulators in the presence of Type 2 singularities. ASME Journal of Mechanisms and Robotics, 2(2): 021004, May 2010.

[12] Briot, S., Arakelian, V.: Optimal Force Generation in Parallel Manipulators for Passing through the Singular Positions. International Journal of Robotics Research, 27(8), pp. 967-983, August 2008.

[13] Paccot, F., Andreff, F., Martinet, P.: A review on the dynamic control of parallel kinematic machine: Theory and experiments. International Journal of Robotics Research, March 2009, 28(3), pp. 395-416.

[14] Spong, M., Hutchinson, S., Vidyasagar,M.: Robot modeling and control. New York: Wiley. December 2005, 496 p. ISBN: 978-0-471-64990-8.

[15] Ghorbel, F., Chételat, O., Longchamp, R.: A reduced model for constrained rigid bodies with application to parallel robots. In Proceedings of the IFAC symposium on Robot Control, Capri, Italy, 1994, pp. 57-62.

[16] Maggiore, M., Consolini, L.: Virtual holonomic constraints for Euler-Lagrange systems. IEEE Transactions on Automatic Control, 58(4), pp. 1001-1008, April 2013.

[17] Westervelt, E., Grizzle, J., Chevallereau, C., Choi, J. and Morris B.: Feedback Control of Dynamic Bipedal Robot Locomotion, Series: Automation and Control Engineering. CRC Press, June 2007, 528 p., ISBN 9781420053722.

[18] Canudas-de-Wit, C., Espiau, B., and Urrea, C.: Orbital stabilization of underactuated mechanical systems. 15th World Congress IFAC, Barcelona, Spain, 2002, 21-26 July.

[19] Shiriaev, A., Perram, J., and Canudas-de-Wit, C.: Constructive tool for orbital stabilization of underactuated nonlinear systems: Virtual constraints approach. IEEE Transactions on Automatic Control, 50(8), pp. 1164-1176, August 2005.

[20] Liberzon, D.: Switching in Systems and Control. Birkhäuser Boston, 2003, 233 p., Print ISBN 978-1-4612-6574-0, Online ISBN 978-1-4612-0017-8.

[21] Ortiz Morales, D., La Hera, P. and Ur Réhman S.: Generating Periodic Motions for the Butterfly Robot. 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), November 3-7, 2013. Tokyo, Japan, pp. 2527-2532.

[22] Ahmed, M.S., Hably, A. and Bacha, S.: Kite generator system periodic motion planning via virtual constraints. 39th Annual Conference of the IEEE Industrial Electronics Society (IECON 2013), November 2013, Vienne, Austria. pp.1-6.

[23] Westerberg, S., Mettin, U. and Shiriaev, A.: Motion planning and control of an underactuated 3dof helicopter. In IROS - IEEE/RSJ Intelligent Robots and Systems, Leiden, Netherlands, June 2010, pp. 3759-3764.

[24] Shiriaev, A., Freidovich, L. and Gusev., S.: Transverse linearization for controlled mechanical systems with several passive degrees of freedom. IEEE Transactions on Automatic Control, 55(4), pp. 893-906, April 2010.

[25] Canudas de Wit, C.: On the concept of virtual constraints as a tool for walking robot control and balancing. Anual Reviews in Control, 28: 157-166, 2004.

[26] Isidori, A.: Nonlinear Control Systems. Communications and Control Engineering. Springer-Verlag London, 1995, 549 p., ISBN 978-3-540-19916-8.

Appendix A

Let us consider the wrench applied on the platform by the legs and external forces:

$$\mathbf{w}_p = \mathbf{A}^T \boldsymbol{\lambda} \tag{4.7}$$

At singularity, equation 4.7 is partially affected by the control inputs due to the degeneracy of the kinematic matrix **A**. However, the evolution of the uncontrollable variables or free dynamics can be deduced. Now, let us express the singular value decomposition of the kinematic jacobian matrix **A**, that degenerates in singularity is:

$$\mathbf{A} = \mathbf{U}^T \boldsymbol{\Sigma} \mathbf{V} \tag{4.8}$$

but also can be expressed as:

$$\mathbf{A}^T = \mathbf{V}^T \boldsymbol{\Sigma} \mathbf{U} \tag{4.9}$$

$\boldsymbol{\Sigma}$ is a diagonal matrix of singular values of the matrix **A**. And the matrices **U** and **V** are orthonormal. Thus:

$$\boldsymbol{\Sigma} = \mathrm{diag}(\varepsilon_n) \tag{4.10}$$

$$\mathbf{U}\mathbf{U}^T = \mathbf{U}^T\mathbf{U} = \mathbf{I} \tag{4.11}$$

$$\mathbf{V}\mathbf{V}^T = \mathbf{V}^T\mathbf{V} = \mathbf{I} \tag{4.12}$$

At singularity, the last singular value is null $\varepsilon_n = 0$. If we substitute equation 4.9 into equation 4.7, then:

$$\mathbf{V}^T \boldsymbol{\Sigma} \mathbf{U} = \mathbf{w}_p \tag{4.13}$$

And by rearranging, it is possible to express the following relation:

$$\boldsymbol{\Sigma} \mathbf{U} = \mathbf{V}^T \mathbf{w}_p \tag{4.14}$$

Now, since the last line of the matrix $\mathbf{V}^T$ is in the Kernel of the matrix **A**, the following expressions can be derived:

$$[\mathbf{v}]_1 \mathbf{w}_p = 0 \tag{4.15}$$

where indeed

$$[\mathbf{v}]_1 = \mathbf{t}_v^T \tag{4.16}$$

$$\mathbf{A}\mathbf{t}_v = \mathbf{U}^T \mathbf{\Sigma} \mathbf{V} \mathbf{t}_v \tag{4.17}$$

$$\mathbf{t}_v \in \text{Kernel}(\mathbf{A}) \tag{4.18}$$

Then, the uncontrollable motion:

$$\mathbf{t}_s \in \text{Kernel}(\mathbf{A}) \neq 0 \tag{4.19}$$

And finally since:

$$\mathbf{t}_v = \alpha \mathbf{t}_s \tag{4.20}$$

$$[\mathbf{v}]_1 \mathbf{w}_p = \alpha \mathbf{t}_s^T \mathbf{w}_p = \mathbf{t}_s^T \mathbf{w}_p = 0 \tag{4.21}$$

where α is a scalar and the equality 4.21 gives the general derivation for expressing the equivalence with the dynamic criterion.