Trajectory Control of a Single Link Rigid-Flexible Manipulator

Pedro Miguel Santos Pires

Dissertação para obtenção do Grau de Mestre em Engenharia Mecânica

Júri

Presidente: Professor José Manuel Gutierrez Sá da Costa  
Orientador: Professor José Manuel Gutierrez Sá da Costa  
Co-Orientador: Professor Jorge Manuel Mateus Martins  
Vogal: Professor Miguel Afonso Dias de Ayala Botto

Setembro - 2007
Este trabalho reflecte as ideias dos seus autores que, eventualmente, poderão não coincidir com as do Instituto Superior Técnico.
Abstract

The primary goal of this work is to control the end-point position of a planar single link flexible robot manipulator from point A to point B.

We first introduce the non-minimum phase problem of flexible manipulators. The simulation tools used, which were developed by J. Martins are then presented. An experimental manipulator was also used to test the control methods.

In this report, three control techniques were used. The first technique, introduced by M. Benosman and G. Le Vey, searches for proper output trajectories with polynomial form. Those output trajectories are obtained in order that the effects of the unstable zeros are completely cancelled.

The second technique was recently studied by A. P. Aguiar, J. P. Hespanha, and P. Kokotović called path-following. The path-following with internal model control main objective is to steer a physical object to converge to a geometric path. The secondary objective is to ensure that an object’s motion along the path satisfies a given dynamic specification. Path-following technique ensures that the system is no longer time dependent, this is suppressed by parameterizing the path with an auxiliary variable.

The last studied technique is the evolution of the methodology previously presented. The normal form path following is a control methodology that can satisfy three tasks.

- Geometric task.
- Dynamic task.
- Boundedness of the zero dynamics states.
In the final section of this report we present the experimental results. There is also specified some conclusions and analysis of the developed work, with a future work recommendation.

**Keywords:** Flexible robot manipulators, motion planning, path-following, internal model control, normal form representation, vibration control.
Resumo

O principal objectivo desta tese é controlar a posição do elemento terminal de um robô manipulador planar flexível do ponto A para o ponto B.

Começamos por introduzir o problema de fase não mínima dos manipuladores flexíveis. As ferramentas de simulação utilizadas, desenvolvidas por J. Martins são apresentadas. Apresentamos o manipulador experimental usado para testar os vários métodos de controlo.

Nesta tese, utilizou-se três técnicas de controlo. A primeira técnica, introduzida por M. Benosman e G. Le Vey, procura uma saída polinomial apropriada. Essa trajectória de saída é obtida de modo que os efeitos dos zeros instáveis do sistema sejam cancelados.

A segunda técnica foi recentemente estudada por A. P. Aguiar, J. P. Hespanha, e P. Kokotović chamada path-following. O objectivo principal do path-following com controlo por modelo interno, é levar um objecto físico a convergir para um determinado caminho geométrico. O objectivo secundário assegura que o movimento do objecto ao longo do caminho satisfaça uma determinada especificação dinâmica. No controlo por path-following, o sistema deixa de ser dependente no tempo, devido ao facto de se parametrizar o caminho com uma variável auxiliar.

A última técnica a ser estudada é a evolução da metodologia anteriormente apresentada. O path-following pela representação na forma normal, é uma metodologia de controlo que satisfaz três tarefas:

- Tarefa geométrica.
- Tarefa dinâmica.
- Estabilidade dos estados com a dinâmica dos zeros instáveis.
Na parte final deste relatório apresentamos os resultados experimentais. Sendo também especificado algumas conclusões e análise do trabalho desenvolvido, com recomendações para o trabalho futuro.

**Keywords:** Robô manipulador flexível, planeamento de trajectória, seguimento de caminhos, modelo de controlo interno, representação na forma normal, controlo de vibrações.
I would like to express my sincere thanks to my supervisor Professor José Sá da Costa.

I would like to express my deepest gratitude to Professor Jorge Martins for his support, dedication and orientation on this work. For his permanent availability for analyze and discuss ideas.

I need to refer to my family, for their support for this work.

I wish to acknowledge appreciated support and encouragement from Professor Pedro Aguiar for his help in the path-following analysis.

Least but not last, I wish to thank all my colleagues from Department of Mechanical Engineering, for the friendly work environment they provided, and for the help they gave me in the many different occasions.

I would like to thank all the others that aren’t mentioned and helped me on this thesis.
## Contents

1 Introduction ................................................................. 1
   1.1 Control solutions for non-minimum phase systems ............... 5
       1.1.1 Path-Following .............................................. 5
       1.1.2 Inversion approach ...................................... 7
   1.2 Notation and terminology ......................................... 8
       1.2.1 Motion planning ........................................... 8
       1.2.2 Path-following ........................................... 9
       1.2.3 System properties ....................................... 9
   1.3 Contributions of this thesis ..................................... 10
   1.4 Objectives and outline of this work ............................. 10

2 Simulator and Experimental setup ..................................... 11
   2.1 Dynamic equations .............................................. 11
   2.2 IST planar flexible robot manipulator .......................... 14
   2.3 Model Simulation ............................................... 16

3 End-effector motion planning for a one link non-minimum phase robot 17
   3.1 Stable Inversion method ....................................... 17
       3.1.1 Example of implementation .............................. 19
   3.2 Stable inversion of the IST planar flexible link robot manipulator 20
CONTENTS

3.3 Simulation Results .................................................. 23
3.3.1 Control where the input is the torque .......................... 24
3.3.2 Control where the input of the system is the angle of the joint. 32

4 Path-Following for one link non-minimum phase robot 43
4.1 Path-Following method with internal model control .................. 43
4.1.1 Problem Statement ................................................. 44
4.1.2 Controller Design - Internal model control ...................... 44
4.1.3 Simulation results .................................................. 47
4.2 Normal Form Path Following ...................................... 62
4.3 Parameter estimation ................................................. 65

5 Experimental Results ................................................. 71
5.1 End-effector motion planning ....................................... 71
5.1.1 Identification ..................................................... 72
5.1.2 Control .......................................................... 73

6 Conclusions .......................................................... 77
6.1 Stable Inversion method ............................................. 77
6.2 Path-Following ....................................................... 78
6.3 Future work ........................................................ 78

Bibliography .......................................................... 80

A Special Coordinate basis ............................................. 85

B Recursive Least Squares equations .................................. 91
List of Figures

1.1 Graph depicting the number of manuscripts indexed per year in a PubMed search containing the words robot or robotic ................................................................. 2
1.2 The Robodoc surgical system. ................................................................. 3
2.1 IST planar flexible link robot manipulator ................................................ 12
2.2 Mechanical model of a single flexible link ............................................. 12
2.3 The first two mode shapes ................................................................. 14
3.1 Location of the zeros and poles for transfer function (3.16) .................... 21
3.2 Desired input .................................................................................. 23
3.3 Desired Output ............................................................................. 23
3.4 Simulation tracking error .................................................................. 24
3.5 Simulated displacement of the End-Effector ....................................... 24
3.6 Simulated close-loop torque ............................................................ 25
3.7 Simulated output with an external input torque (Amplitude 0.002 Nm) .... 26
3.8 Simulated close-loop torque with an external input torque (Amplitude 0.002 Nm) ....... 26
3.9 Simulation tracking error for an external input torque (Amplitude 0.002 Nm) ........ 27
3.10 Simulated close-loop torque with an external input torque (Amplitude 0.1 Nm) .... 27
3.11 Simulation output with an external input torque (Amplitude 0.1 Nm) ........ 28
3.12 Simulation tracking error for an external input torque (Amplitude 0.1 Nm) ........ 28
LIST OF FIGURES

3.13 Simulated close-loop torque with noise ........................................ 29
3.14 Simulation output with noise .......................................................... 29
3.15 Simulation tracking error with noise ................................................. 30
3.16 Simulated close-loop torque with miscalculated zeros ......................... 30
3.17 Simulation output with miscalculated zeros ...................................... 31
3.18 Simulation tracking error with miscalculated zeros ............................ 31
3.19 Desired $\theta$ ............................................................................. 33
3.20 PD controller .............................................................................. 34
3.21 Simulated output ........................................................................... 34
3.22 Simulated end-effector deformation ............................................... 34
3.23 Simulated error ........................................................................... 35
3.24 Simulated output with an external input torque (Amplitude 0.002 Nm) . 35
3.25 Simulated close-loop torque with an external input torque (Amplitude 0.002 Nm) .......................................................... 36
3.26 Simulation tracking error for an external input torque (Amplitude 0.002 Nm) .......................................................... 36
3.27 Simulated output with an external input torque (Amplitude 0.1 Nm) .... 37
3.28 Simulated close-loop torque with an external input torque (Amplitude 0.1 Nm) .......................................................... 37
3.29 Simulation tracking error for an external input torque (Amplitude 0.1 Nm) .......................................................... 38
3.30 Simulated close-loop torque with noise ........................................... 38
3.31 Simulation output with noise ............................................................ 39
3.32 Simulation tracking error with noise ................................................. 39
3.33 location of zeros and poles in a closed loop form for miss calculation of transfer function . 40
3.34 Simulated closed-loop torque with miscalculated zeros ..................... 41
3.35 Simulation output with miscalculated zeros ..................................... 41
3.36 Simulation tracking error with miscalculated zeros ........................... 42

4.1 Controller block diagram ............................................................... 45
<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.2</td>
<td>The path in (x, y) coordinates</td>
<td>48</td>
</tr>
<tr>
<td>4.3</td>
<td>First Path - Evolution of the desired output versus the path variable</td>
<td>49</td>
</tr>
<tr>
<td>4.4</td>
<td>First Path - Simulation tracking error</td>
<td>50</td>
</tr>
<tr>
<td>4.5</td>
<td>First Path - Deformation of the end-effector</td>
<td>50</td>
</tr>
<tr>
<td>4.6</td>
<td>First Path - Evolution of the desired path variable versus the real path variable</td>
<td>51</td>
</tr>
<tr>
<td>4.7</td>
<td>Second Path - Simulation velocity profile</td>
<td>51</td>
</tr>
<tr>
<td>4.8</td>
<td>Second Path - Simulation tracking error</td>
<td>52</td>
</tr>
<tr>
<td>4.9</td>
<td>Second Path - Evolution of the desired path variable versus the real path variable</td>
<td>53</td>
</tr>
<tr>
<td>4.10</td>
<td>Second Path - Deformation of the end-effector</td>
<td>53</td>
</tr>
<tr>
<td>4.11</td>
<td>Second Path - Simulation velocity profile</td>
<td>54</td>
</tr>
<tr>
<td>4.12</td>
<td>Second Path - Deformation of the end-effector</td>
<td>54</td>
</tr>
<tr>
<td>4.13</td>
<td>Second Path - Simulation tracking error</td>
<td>54</td>
</tr>
<tr>
<td>4.14</td>
<td>First Path validation - Simulation input torque (White noise with the noise power of 0.1)</td>
<td>55</td>
</tr>
<tr>
<td>4.15</td>
<td>First Path validation - Deformation of the end-effector (White noise with the noise power of 0.1)</td>
<td>56</td>
</tr>
<tr>
<td>4.16</td>
<td>First Path validation - Simulation output (White noise with the noise power of 0.1)</td>
<td>56</td>
</tr>
<tr>
<td>4.17</td>
<td>First Path validation - Simulation tracking error (White noise with the noise power of 0.1)</td>
<td>57</td>
</tr>
<tr>
<td>4.18</td>
<td>Second Path validation - Simulation tracking error (External step input torque with amplitude of 1Nm)</td>
<td>57</td>
</tr>
<tr>
<td>4.19</td>
<td>Second Path validation - Simulation input torque (External step input torque with amplitude of 1Nm)</td>
<td>58</td>
</tr>
<tr>
<td>4.20</td>
<td>Second Path validation - Deformation of the end-effector (External step input torque with amplitude of 1Nm)</td>
<td>58</td>
</tr>
<tr>
<td>4.21</td>
<td>Second Path validation - Simulation Tracking error (White noise with the noise power of 1)</td>
<td>59</td>
</tr>
<tr>
<td>4.22</td>
<td>Second Path validation - Simulation input torque (Path properties: (a_1 = 1, a_2 = 0.1) and (m_s = 1))</td>
<td>59</td>
</tr>
<tr>
<td>4.23</td>
<td>Second Path validation - Deformation of the end-effector (Path properties: (a_1 = 1, a_2 = 0.1) and (m_s = 1))</td>
<td>60</td>
</tr>
<tr>
<td>Figure</td>
<td>Description</td>
<td>Page</td>
</tr>
<tr>
<td>--------</td>
<td>-----------------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>4.24</td>
<td>Second Path validation - Simulation tracking error (Path properties: $a_1 = 1$, $a_2 = 0.1$ and $m_s = 1$)</td>
<td>60</td>
</tr>
<tr>
<td>4.25</td>
<td>Second Path validation - Evolution of the desired path variable versus the real path variable (Path properties: $a_1 = 1$, $a_2 = 0.1$ and $m_s = 1$)</td>
<td>61</td>
</tr>
<tr>
<td>4.26</td>
<td>Second Path validation - Simulation output (Path properties: $a_1 = 1$, $a_2 = 0.1$ and $m_s = 1$)</td>
<td>61</td>
</tr>
<tr>
<td>4.27</td>
<td>Second Path validation - Joint velocity versus error (Path properties: $a_1 = 1$, $a_2 = 0.1$ and $m_s = 1$)</td>
<td>62</td>
</tr>
<tr>
<td>4.28</td>
<td>Output values of flexible link identification</td>
<td>68</td>
</tr>
<tr>
<td>4.29</td>
<td>Flexible link identification validation</td>
<td>70</td>
</tr>
<tr>
<td>4.30</td>
<td>Flexible link identification error</td>
<td>70</td>
</tr>
<tr>
<td>5.1</td>
<td>Identification input</td>
<td>72</td>
</tr>
<tr>
<td>5.2</td>
<td>Identification output</td>
<td>72</td>
</tr>
<tr>
<td>5.3</td>
<td>Location of zeros and poles in a close loop form</td>
<td>73</td>
</tr>
<tr>
<td>5.4</td>
<td>End-effector vibration at low average velocity</td>
<td>74</td>
</tr>
<tr>
<td>5.5</td>
<td>End-effector vibration at high average velocity</td>
<td>74</td>
</tr>
<tr>
<td>5.6</td>
<td>Input torque at low average velocity</td>
<td>75</td>
</tr>
<tr>
<td>5.7</td>
<td>Input torque at high average velocity</td>
<td>75</td>
</tr>
<tr>
<td>A.1</td>
<td>Interpretation of structural decomposition of a SISO system</td>
<td>86</td>
</tr>
</tbody>
</table>
List of Tables

2.1 IST flexible robot manipulator characteristics ........................................ 15
LIST OF TABLES

xiv
Chapter 1

Introduction

Robotics will have in the future an important role in our life (fig.1.1). Robot manipulator technology is constantly improving, by increasing efficiency and becoming more human interactive [1]. The way to achieve the same performance in terms of speed, acceleration and operational load with less weight and increased human interactivity, is with flexible manipulator robots, due to their light weight structures and mechanical flexibility.

But what are flexible manipulators robots, and why do we call them that? Generally, flexible manipulators robots are manipulators that have mechanical flexibility in their links and joints, which results in the vibration or oscillation of the end-effector either during the manipulator’s motion or immediately after it stops. This behavior reduces the position precision, but on the other hand one can achieve higher velocities due to the lower inertia and mass.

Flexibility can appear due to two factors: high accelerations in the motion or long and slender links of the manipulators. Therefore, flexibility is a result of the demand for the manipulator operation. If the manipulator operates at high acceleration, it may become flexible. If the manipulator operates in small spaces, like surgical robots, it may need to be flexible. If the manipulator has to be light, because of transportation demands for example, the use of small quantities of material or light materials results in flexible structures.

To solve the problems resulting of structural flexibility, and have a satisfactory precision in the positioning of the end-effector, industrial robot manipulators are built with a rigid structure. Such rigid structures are
Figure 1.1: Graph depicting the number of manuscripts indexed per year in a PubMed search containing the words robot or robotic

achieved by using heavy and stiff designs, that increase the weight, therefore increasing the consumption of energy during motion.

As interesting as flexible robots may seem for robotic surgery, the existing solutions are rigid. Active (rigid) robotic devices, in which pre-programmed data and computer-generated algorithms function without real-time operator input, were the first robots to be used in live surgical applications.

In 1985 [1], the first surgical application of industrial robotic technology was described when an industrial robotic arm was modified to perform a stereotactic brain biopsy with 0.05 mm accuracy. This served as the prototype for Neuromate (Integrated Surgical Systems, Sacramento, CA, USA) which received Food and Drug Administration (FDA) approval in 1999. Meanwhile, for orthopedics, in 1992, the Robodoc (Integrated Surgical Systems) was introduced for use in hip replacement surgery (fig. 1.2). The Robodoc is a computer-guided mill used to core the femoral head to receive a hip replacement prosthesis. Clinical trials demonstrate greater accuracy comparing the well drilled by the Robodoc to conventional techniques. Whereas the Robodoc has been used in thousands of patients in Europe, it has not yet received FDA approval in the United States because of concerns regarding complication rates. Similar devices have been
designed for use in knee replacement and temporal bone surgery, notably the Acrobot (The Acrobot Company, Ltd., London, UK) and the RX-130 robot (Staubli Unimation Inc., Faverges, France), respectively. Neither device has yet completed clinical testing nor received FDA approval [1].

The dynamics equations of flexible manipulator robots are significantly more complex than those of rigid robots due to the link’s flexibility. Flexibility yields a robot model with infinite degrees of freedom, which must be truncated to a finite number.

There are three principal objectives in the control of the flexible robot arms:

- O1 - Point to point motion of the end-effector.
- O2 - Trajectory tracking in the joint space (tracking of a desired angular trajectory).
- O3 - Trajectory tracking in the operational space (tracking of a desired end-effector trajectory).

The control of the motion of flexible manipulators is one of the areas under great investigation in robotics. In rigid manipulator robots, the control task is less complicated, since one achieves tip position by controlling the joint angles. This is known as dead-reckoning, as this kind of control approach does not contemplate feedback of the position of the end-effector. For flexible robots, there is the need to use a controller designed to damp out the oscillations at the tip that appear during motion. In this project
we will discuss two methods to solve this problem. They are path-following [2], [3], [4], [5] and reference tracking [6], [7], [8], [9].
1.1 Control solutions for non-minimum phase systems

Due to the elastic deflection of a flexible beam, its linearized model possesses one real zero per vibration mode in the right half complex plane. As is well known, unstable zero dynamics results in limitations on the achievable performance and tracking capabilities. Dačic [5] presented three distinct approaches for tracking in the presence of unstable zeros dynamics. He referred to them as: the Internal Model approach, the Flatness approach and the Inversion approach. In this report we will focus on the internal model approach and the inversion approach.

1.1.1 Path-Following

A path-following problem is first considered in robotics literature in [10]. The objective is to control a robot using limited torques to move along a pre-arranged path in minimal time. Hauser and Hindman [11] started a new path-following methodology, where the main objective, is to determine on-line a desirable velocity profile along the path. They introduced a state path-following problem, where the path is specified for each state variable of the system. Encarnação and Pascoal [12] proposed a different kind of path-following, called output path-following, where the path is defined only for the system output. This report follows this line of work, as in Aguiar [4], [2], [3] and Dačic [5]. In these works the path-following problem is defined as a combination of two tasks: geometric and dynamic. Then, it is shown that it is possible to construct feedback laws that can achieve an arbitrarily small $L_2$ norm of the path-following error. This property highlights the difference between path-following and reference tracking, where a fundamental limitation exists in terms of a lower bound on the $L_2$ norm of the tracking error imposed by the unstable zero dynamics.

One of the major problems with non-minimum phase systems is to design control laws in order to drive the output to reach and follow a geometric reference (path). Another problem is to compel the system to follow the geometric reference and satisfy some additional dynamic specifications (as speed, acceleration...). Since the limitations introduced by unstable zeros are structural, they cannot be avoided without changing the system structure or reformulating the tracking problem. One of the reformulation possibilities is to select a new output in which the zero dynamics become stable (change the output from the end-effector to the hub rotation).
1.1 Control solutions for non-minimum phase systems

One approach to the path-following problem is shown in the work by Aguiar et al. [3] and [2], where the authors introduce a new path variable $\gamma$, in order to suppress the time dependence of the motion. In Dačić [5], an intuitive description is given to the problem. In his interpretation of path-following, a fictitious object called the leader, is introduced. The leader moves along the path, and its current position at the time $t$ is $\gamma(t)$. The main objective of the path-following control theory is to steer a physical object, called the follower, to converge to the leader geometric path. The secondary objective is to ensure that the object’s motion along the path satisfies a given dynamic specification. To assure this dynamic specification, it is required to design a timing law for $\gamma$. This is a feedback law for the leader, which ensures that its motion asymptotically satisfies the dynamics specification. We can assume that reference tracking is a special case of path following, where the timing law $\gamma(t)$ is pre-determined.

Internal model approach

The goal of the internal model approach is to design control laws that are capable of driving an object (such as a robot arm, ship or aircraft) to reach and follow a geometric path, and force the object to satisfy some additional dynamic specification. This dynamic specification can be represented as outputs of a neutral stable system, called exosystem. In Aguiar et al. [3] and [2], it was demonstrated that it is possible to secure asymptotic convergence of the tracking error to zero. Since this approach is the one that originates the output regulation problem, we can consider the following system:

$$\dot{x} = Ax + Bu$$

(1.1)

$$\dot{w}(\theta) = S \times w(\gamma)$$

(1.2)

$$e = Cx + Q \times w(\gamma)$$

(1.3)

where $x(t)$ is the state, $u(t)$ is the input, $e$ is the regulated output available for measurement, and $w(\gamma)$ is the state of the exosystem (1.2) assumed to be stable.

Kwakernaak and Sivan [13] presented the demonstration of the following theorem:

**Theorem 1.** The regulator problem is solvable if and only if $(A, B)$ is stabilizable, $(C, A)$ is detectable, the number of inputs is at least as large as the number of outputs ($m \geq q$) and the equations

$$\Pi S = A\Pi + B\Gamma$$

(1.4)

$$0 = C\Pi + D\Gamma - Q$$
have solution for matrices II and Γ.

It can also be shown that the last condition on Theorem 1 is equivalent to requiring that the eigenvalues of the matrix $S$ differ from the transmission zeros of (1.1). But the internal model approach does not avoid the performance limitations due to unstable zero dynamics. In recent studies Aguiar et al. [3], [2] and [4] has presented a solution, where this limitation can be avoided by the combination of the internal model approach and the path-following. Their methodology consists in replacing the variable $t$ in the geometric path $y_d$, by a path variable $\gamma$ and then selecting a timing law to it. In this way, the variable $\gamma$ is treated as an additional control variable. Considering the new variable $\gamma$, it is possible to achieve an $L_2$ norm of the path-following error relatively small.

1.1.2 Inversion approach

The main idea of the inversion approach is to obtain a stable reference in order to acquire a finite smooth control torque, that implies an exact reproduction of the computed trajectory, by using a simple input-output inversion technique applied to our non-minimum phase system. Benosman and Le Vey [6], [7], [8] and [9] presented a methodology to obtain a stable inversion via output planing of a single-input-single-output (SISO) non-minimum phase system.

End-effector motion planning

One technique to reduce all the undesired behavior characteristics (such as vibration of the end-effector) of a flexible robot, is to calculate the end-effector motion in such a way not to disturb the unstable zeros of the system. Much work has been dedicated in the past two decades to the control of the flexible arm models, and much of those works have yielded good results for the first two objectives described in O1 and O2.

Benosman and Le Vey [6] made a profound analysis about stable inversion of a SISO non-minimum phase linear system. Instead of searching for proper initial conditions associated to a given desired output, this approach, consists in a search for a proper output, associated to the desired initial conditions. This output is planned in a way that it is possible to cancel all the effects of the unstable zeros.
1.2 Notation and terminology

After knowing the required output, it is possible to estimate the capable input to obtain the desired results. This methodology addresses the problem of planning a target tip trajectory, leading to a finite smooth control torque that implies an exact reproduction of the computed trajectory, by using a simple input-output inversion technique. Just as other common methodologies, the result of this methodology is presented on an open loop control torque in the time domain.

1.2 Notation and terminology

This report follows a certain general notation and terminology, excepting in some particular examples, where new auxiliary variables appear. Their meaning will then be described.

Time is always denoted by $t$. The particular time events, as final time and initial time are referred as $t_f$ and $t_i$ respectively.

When a given function is $i$-times differentiable, appears $f^{(i)}$ for its $i^{th}$ time derivative, $f^{(i)} \triangleq \frac{d^i f(t)}{dt^i}$.

Identity matrix of dimension $n$ is defined by $I_n$. The $i^{th}$ vector element is defined by $V(i)$. The matrix elements are represented with $A_{ij}$, since they have two entries ($i^{th}$ row and $j^{th}$ column).

1.2.1 Motion planning

In chapter 3.1 we present some polynomial manipulations, where:

- $A_i$ corresponds to the $i^{th}$ index of the polynomial representation of the stable and unstable transient solution.
- $A_{ist}$ corresponds to the $i^{th}$ index of the polynomial representation of the stable transient solution.
- $B_i$ correspond to the $i^{th}$ index of the polynomial representation of the particular solution.

The parameter $\alpha$ represents the system zeros, and $\alpha_{ist}$ correspond to the $i^{th}$ stable zero. The variables $K_p$ and $K_v$ represent the PD controller proportional and derivative gain respectively.
1.2.2 Path-following

Like in the literature, the system state space matrices are commonly represented by A,B,C,D. In path-following methodology the leader is represented by the letter $w$, and the path-variable is represented by $\gamma$.

The variables $K$ and $L$ belong to the path-following controller. $K$ is the gain vector obtained by the LQR method, and $L$ is the leader gain matrix.

$\text{Kron}(X,Y)$ is the Kronecker tensor product of $X$ and $Y$ representation. The result is a large matrix formed by taking all possible products between the elements of $X$ and those of $Y$. For example:

If $X$ is a 2 by 3 matrix and $Y$ is of any dimension, then $\text{Kron}(X,Y)$ is:

$$\begin{bmatrix} X_{(1,1)} \times Y & X_{(1,2)} \times Y & X_{(1,3)} \times Y \\ X_{(2,1)} \times Y & X_{(2,2)} \times Y & X_{(3,1)} \times Y \end{bmatrix}$$

(1.5)

1.2.3 System properties

The physical system properties are defined as:

- $\omega$: frequency.
- $\tau$: input torque.
- $r$: hub radius.
- $I_h$: hub inertia.
- $I_b$: flexible beam inertia.
- $l$: flexible link length.
- $v(x)$: flexible link displacement at distance $x$ from the base of the link.
- $\theta$: Hub rotation angle.
- $\eta$: flexible link modal amplitude.
- $\phi$: modal function.
1.3 Contributions of this thesis

This thesis is a general analysis approach to the flexible manipulator robot control problem. It compares motion-planning and path-following control methodologies for the IST planar flexible manipulator robot. It presents simulation and experimental results of the studied methodologies.

There are two innovative contributions of this thesis. First, in terms of reference planning, we present a modification of the original approach in order to overcome joint friction. We introduce a joint controller modifying the input from torque to joint angle. Second, to the knowledge of the author, this works presents the first attempt to apply a path-following controller to flexible link manipulators.

1.4 Objectives and outline of this work

This thesis has as primary objective, the control of the position of the end-effector of a flexible robot with a significant precision, assuming a path, from point one to point two, driven at variable speeds profile. To this end, this thesis is composed of the following chapters:

Chapter 1: Introduction to the problem at hand.

Chapter 2: Overview of the modeling and experimental setup. The IST planar flexible manipulator robot is presented, its dynamics equations and properties.

Chapter 3: Trajectory-planning control methodology. This methodology allows for computing a path, leading to a feedforward torque, producing an exact reproduction of the path.

Chapter 4: Path-following control methodology, which allows us to control the non-minimum phase system with a closed-loop form controller.

Chapter 5: Experimental results, obtained with end-effector motion planing

Chapter 6: Conclusions, summary of the work and recommendations for future work.
Chapter 2

Simulator and Experimental setup

In this chapter, the mathematical equations and a brief overview of the IST planar flexible robot manipulator will be presented.

2.1 Dynamic equations

In Martins [14] two experimental robotic setups are presented. One for planar experiments, and another for general three dimensional experiments. The work presented here refers to the planar IST flexible manipulator, namely its flexible link, fig. 2.1. This robot has been studied in [15], [16] and [17] where it is widely described.

The flexible link is clamped to a rigid hub with a moment of inertia \( I_H \), radius \( r \) and an input torque \( \tau \). Considering linear displacements, the position of the link located at the distance \( x \) from the frame origin, in the \( OX \) direction, relative to the \( \{O_0, X_0, Y_0, Z_0,\} \) (inertial), reference frame is given by (fig. 2.2):

\[
\vec{p}(x,t) = [x \cos \theta(t) - \nu(x,t) \sin \theta(t)] e_1 + [x \sin \theta(t) - \nu(x,t) \cos \theta(t)] e_2
\]

(2.1)

where \( e_1 \), \( e_2 \) and \( e_3 \) are the unit vectors along the \( X_0 \), \( Y_0 \) and \( Z_0 \) axes respectively. The velocity of the same infinitesimal element is accordingly given by

\[
\dot{\vec{p}}(x,t) = \left[ -\left(\dot{x} \theta + \nu \right) \sin \theta(t) - \nu \dot{\theta} \cos \theta(t) \right] e_1 + \left[ \left(\dot{x} \theta + \nu \right) \cos \theta(t) - \nu \dot{\theta} \sin \theta(t) \right] e_2
\]

(2.2)
The kinetic energy of the system can be deduced as [16]

\[ T = \frac{1}{2} (I_H + I_b) \dot{\theta}^2 + \frac{1}{2} \int r^{r+L} \rho (\dot{\nu}^2 + 2\dot{\nu}\dot{\theta}) \, dx \]  

(2.3)

and the elastic potential energy of the beam as

\[ V = \frac{1}{2} \int r^{r+L} EI \left( \frac{\partial^2 \nu}{\partial x^2} \right)^2 \, dx \]  

(2.4)

Martins et al. [16] also describes how to obtain the discrete ordinary differential equations of the current system as

\[ M\ddot{q} + Kq = T \]  

(2.5)
Simulator and Experimental setup

Where $M$ is the system inertia matrix, $K$ is the system stiffness matrix, $T$ is the vector of external torques and $q$ is the vector of generalized coordinates.

\[
q = [\theta \, \eta_1 \, \eta_2 \ldots \, \eta_k]^T
\]

\[
T = [\tau \, 0 \, 0 \ldots 0]^T
\]

Here, $\eta_i$ is the modal amplitude of the $i$th mode considering the assumed modes discretization procedure, being $k$ the total number of assumed modes. The mass matrix and the stiffness matrix are given by

\[
M = \begin{bmatrix}
I_H + I_b & \rho \int_{-L}^{L} x \phi_1(x) \, dx & \rho \int_{-L}^{L} x \phi_2(x) \, dx & \cdots & \rho \int_{-L}^{L} x \phi_k(x) \, dx \\
\rho \int_{-L}^{L} x \phi_1(x) \, dx & \rho L & 0 & 0 & 0 \\
\rho \int_{-L}^{L} x \phi_2(x) \, dx & 0 & \rho L & 0 & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
\rho \int_{-L}^{L} x \phi_k(x) \, dx & 0 & 0 & 0 & \rho L
\end{bmatrix}
\]

(2.7)

\[
K = \begin{bmatrix}
0 & 0 & 0 & \cdots & 0 \\
0 & \rho L \omega_1^2 & 0 & 0 & 0 \\
0 & 0 & \rho L \omega_2^2 & 0 & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & 0 & 0 & \rho L \omega_k^2
\end{bmatrix}
\]

(2.8)

The inertial displacement at a distance $x$ from the frame origin in the $OX$ direction is given by

\[
y(x, t) = x \, \theta(t) + \nu(x, t)
\]

(2.9)

Observing (2.9) we can verify that the total displacement is a function of the rigid body motion $\theta(t)$ and the elastic deflection (displacement) $\nu(x, t)$, where this latter term is discretized as

\[
\nu(x, t) = \sum_{i=1}^{k} \phi_i(x) \, \eta_i(t)
\]

(2.10)

$\phi_i(x)$ and $\eta_i(t)$ represent the modal functions and modal amplitudes of the $i$th Hermite-cubic mode respectively. In this report we consider two Hermite-cubic modes. The modal functions are:

\[
\phi_1(x) = 3 \left( \frac{x}{L} \right)^2 - 2 \left( \frac{x}{L} \right)^3
\]

\[
\phi_2(x) = \left( \frac{x}{L} \right)^2 - \left( \frac{x}{L} \right)^3
\]

(2.11)

\[
\phi(x) = \begin{bmatrix} \phi_1(x) & \phi_2(x) \end{bmatrix}
\]

where the modes shape are presented in fig. 2.3.
2.2 IST planar flexible robot manipulator

In this project the IST planar flexible robot manipulator (fig. 2.1) is used. It is based on a modular structure that can be transformed into a single-link flexible manipulator, where the second joint of the manipulator is allowed to rotate and the respective link is made of a very flexible spring-steel beam.

The actuation mechanism is a Harmonic Drive RH-14-6002 servo system, current driven by a 12A8 servo amplifier from Advanced Motion Controls. The measurement chain of this prototype is constituted

The state space representation of the flexible robot manipulator can be represented as,

\[
\begin{bmatrix}
\dot{\mathbf{q}} \\
\ddot{\mathbf{q}}
\end{bmatrix} = \begin{bmatrix}
0_{3x3} & I_{3x3} \\
-M^{-1}K & -M^{-1}C
\end{bmatrix} \begin{bmatrix}
\mathbf{q} \\
\dot{\mathbf{q}}
\end{bmatrix} + \begin{bmatrix}
0_{3x3} \\
M^{-1}
\end{bmatrix} \tau
\]

(2.12)

and the transfer function representation can be obtained as

\[
G(s) = \frac{Y(s)}{\tau(s)} = C [sI - A]^{-1} B_1
\]

(2.13)

where

\[
A = \begin{bmatrix}
0_{3x3} & I_{3x3} \\
-M^{-1}K & -M^{-1}C
\end{bmatrix}; \quad B = \begin{bmatrix}
0_{3x3} \\
M^{-1}
\end{bmatrix}; \quad C = \begin{bmatrix}
r + l & \phi_1 & \phi_2 & 0 & 0 & 0
\end{bmatrix}
\]

(2.14)

Throughout this work, the above linear model is used for controller design, whereas a non-linear model (quadratic displacements) is used for simulation.
by an incremental encoder with 2000 pulses per revolution that measure the position of the joint, and three Wheatstone bridges along the flexible arm, at 4.5cm, 18cm and 32cm from joint 2, to measure the bending represented by variables $\nu''(x_1)$, $\nu''(x_2)$ and $\nu''(x_3)$ respectively. For the purposes of this work only the first two were used. It was verified that for the smooth trajectories tested here the first two modes suffice. In table 2.1 the most important properties of the IST planar manipulator are presented.

<table>
<thead>
<tr>
<th>IST Manipulator physical properties</th>
</tr>
</thead>
<tbody>
<tr>
<td>Beam length ($l$)</td>
</tr>
<tr>
<td>Beam width</td>
</tr>
<tr>
<td>Beam height</td>
</tr>
<tr>
<td>Mass density ($\rho$)</td>
</tr>
<tr>
<td>Beam Young modulus ($E$)</td>
</tr>
<tr>
<td>Hub radius ($r$)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Natural clamped-free beam frequencies</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\omega_1$</td>
</tr>
<tr>
<td>$\omega_2$</td>
</tr>
<tr>
<td>$\omega_3$</td>
</tr>
</tbody>
</table>

Table 2.1: IST flexible robot manipulator characteristics

To integrate the various components of the experimental setup, and implement the real time control of the robot, the MATLAB toolbox xPC Target was used. The xPC Target, in its basic configuration, needs two computers: the Host computer, where the control programs are created, and the Target computer that is responsible for the real time computation. The connections to the robot are performed through an I/O Servotogo board, which drives are created as a library for MATLAB.

In the Host computer, the control programs are created in the Simulink environment of MATLAB that are compiled in C code, creating an executable that is sent to the Target computer in which the application runs in real time. In this configuration, the Target computer needs a processor and a RAM memory with enough capacity to run the applications only, since an operating system isn’t necessary. The programs used in this work were created in MATLAB version 7.0.1 which has the Simulink version 6.1 and the version 2.6 of the xPC Target toolbox. The Target computer has a Pentium MMX processor at 233MHz and 28MB of RAM memory.
For model simulation, a Matlab toolbox called *MECANISMO*, has been used. This toolbox was developed in Martins [14], it works seamlessly with other Matlab tools, and has been coded with special attention to its usage in a real time framework. The modules of *MECANISMO*, along with other tools available in Simulink, especially the numerical integrators, allow for simulation in free, constrained and impact motion of flexible manipulators. A quadratic model has been built with *MECANISMO* for model simulation.
Chapter 3

End-effector motion planning for a one link non-minimum phase robot

The objective of this study is to stabilize a single-input single-output non-minimum phase system. In this study it is common to use a $u$ as input, a $y$ as output and an $s$ as the Laplace transform variable.

3.1 Stable Inversion method

Consider $u$ and $y$ related by a given input-output equation

$$P\left(\frac{d}{dt}\right)u(t) = Q\left(\frac{d}{dt}\right)y(t)$$

(3.1)

Here, $P$ and $Q$ are polynomials in a differential operator $d/dt$, with degrees $m$ and $n$ respectively ($m < n$).

In a first analysis, due to the linear nature of equation (3.1), its solution is composed by two terms.

- The transient.
- The Steady-State.

Since the system has a non-minimum phase characteristic, the transient of the system's inverse contains divergent terms, resulting from the unstable zeros. So, in this case, one solution to the problem is to plan the output trajectory in such a way that the undesirable response of the system input is cancelled. To
3.1 Stable Inversion method

cancel these undesired terms, a polynomial in time form for the output trajectory is considered

\[ y_d = \sum_{i=1}^{p} a_i t^{i-1} \tag{3.2} \]

where the degree of the polynomial form \( p \) depends on the number of output initial and final constraints, as well as the number of unstable zeros associated to equation (3.1).

The input in equation (3.1) may be written as

\[ u(t) = u_t(t) + u_p(t) \tag{3.3} \]

where \( u_t \) is the transient solution and \( u_p \) is the particular solution of the inverse system. The transient solution can be represented by:

\[ u_t(t) = \sum_{i=1}^{m} A_i(a_i, t_0, u_{00}, u_{01}, \ldots, u_{0(n-1)}) e^{r_i t} \tag{3.4} \]

where the \( r_i \) are all the roots of the characteristic equation of the inverse problem. The \( A_i \) are linear functions of the \( a_i \) coefficients and all initial conditions. It is shown in [6] that their general expression is

\[ A_i = u_0 + \sum_{j=1}^{p} \frac{a_j}{\text{zero}_j} \tag{3.5} \]

Furthermore, the particular solution can be represented by:

\[ u_p(t) = \sum_{i=1}^{p} B_i(a_i) t^{i-1} \tag{3.6} \]

To cancel the effect of the unstable zeros on the transient solution (3.4) (all the zeros in the right half plane or pure imaginary zeros), the \( A_i \) associated with the unstable zeros must be equal to zero

\[ A_i(a_i, t_0, u_{00}, u_{01}, \ldots, u_{0(n-1)}) = 0 \tag{3.7} \]

With this constraint in the linear system, we can obtain the output coefficients \( a_i \). Adding the final and initial constraints, this leads to the linear system:

\[
\begin{cases}
A_i(a_i, t_0, u_{00}, u_{01}, \ldots, u_{0(n-1)}) = 0 \\
y_d(i)|_{t_0} = \text{initial conditions} \\
y_d(i)|_{t_f} = \text{final conditions}
\end{cases} \tag{3.8}
\]

where \( i \) is the highest order for the specified output derivatives. From equation (3.8) we have the coefficients \( a_i \) and the necessary output to cancel the unstable zeros. So, the result of the remaining \( A_i \) is known.
complete the desired input, \( u(t) \), it is only necessary to obtain the particular input solution (3.6). The \( B_i \) elements are obtained as linear functions of the output coefficients \( a_i \), through substitution of equation (3.4) into the differential equation (3.1).

At this point, all necessary elements to obtain the desired input in an open loop form are:

\[
u_{ol} = \sum_{i=1}^{p} B_i(a_i) e^{t_{i-1}} + \sum_{i=1}^{m} A_{ist}(a_i, t_0, u_0, u_0^{(1)}, ..., u_0^{(n-1)}) e^{r_{ist} t}
\]

(3.9)

where \( r_{ist} \) and \( A_{ist} \), are the stable zeros from the corresponding \( A_i \) terms. In a final approach to the problem, it is recommended to close the loop around the joint angle in order to gain some robustness. The final control law in a closed-loop form is:

\[
u_{cl}(t) = u_{ol}(t) + Ke^\theta
\]

(3.10)

where the error is \( e^\theta(t) = \theta_d(t) - \theta(t) \)

### 3.1.1 Example of implementation

As an introduction to the problem, let us consider a simple example. Consider the following input-output equation:

\[
u(t) - \alpha u(t) = y(t), \tag{3.11}
\]

where \( \alpha \) is the unstable zero. In order to cancel the undesired non-minimum phase characteristics, we calculate the input as in (3.9).

1. To obtain the output (3.2) which cancels all unstable zeros, we write:

\[
y_d(t) = a_1 + a_2 t,
\]

(3.12)

where \( p = 2 \), since we have an unstable zero in \( s = \alpha \), and an initial output value \( y(t = 0) = y_0 \).

2. Substituting (3.12) into (3.11) and solving for \( u \), we obtain:

\[
u(t) = (u_0 + \frac{a_1}{\alpha} + \frac{a_2}{\alpha^2}) e^{\alpha t} - (\frac{a_1}{\alpha} + \frac{a_2}{\alpha^2}) - \frac{a_2}{\alpha} t
\]

(3.13)
3.2 Stable inversion of the IST planar flexible link robot manipulator

where we verify that the first term in the right hand side follows the general form of (3.5). Furthermore, we verify that

\[ B_1 = -\left(\frac{a_1}{a} + \frac{a_2}{a^2}\right) \]
\[ B_2 = -\frac{a_2}{a} \]  

(3.14)

3. Calculate the indices \( a_i \):

\[
\begin{align*}
A_1 &= u_0 + \frac{a_1}{a} + \frac{a_2}{a} \\
\frac{y_0(t)}{y_0} &= \frac{-a_2}{a} \\
\frac{a_2}{a_1} &= \frac{a_2^2}{a_1} \left( -u_0 - \frac{y_0}{a} \right) \\
\end{align*}
\]

(3.15)

Alternatively, we can use (3.5) in step 3 in order to calculate the \( a_i \) and then solve only for the particular solution of the differential equation to obtain the \( B_i \). This is the method implemented in the generalized procedure.

3.2 Stable inversion of the IST planar flexible link robot manipulator

After the previous description and simple example, we can now advance to the case of the one link flexible arm. The joint torque to tip inertial displacement transfer function of the system is

\[
y \tau = \frac{92.5 s^4 + 8.545 \times 10^{-12} s^3 - 4.924 \times 10^6 s^2 - 3.015 \times 10^{-7} s + 1.163 \times 10^{10}}{s^2 (s^4 + 1.066 \times 10^{-14} s^3 + 56780 s^2 + 5.795 \times 10^{-10} s + 1.307 \times 10^8)} \]

(3.16)

For simplification we will call the coefficients of the numerator as a vector \( Num \), and the coefficients of the denominator as a vector \( Den \):

\[
Num = \begin{bmatrix} 92.5 & 8.545 \times 10^{-12} & -4.924 \times 10^6 & -3.015 \times 10^{-7} & 1.163 \times 10^{10} \end{bmatrix}^T
\]
\[
Den = \begin{bmatrix} 1 & 1.066 \times 10^{-14} & 56780 & 5.795 \times 10^{-10} & 1.307 \times 10^8 & 0 & 0 \end{bmatrix}^T
\]

(3.17)

The transfer function zeros are:

\[ Zero_{1,2} = \pm 225.2893 \quad Zero_{3,4} = \pm 49.7712 \]

where two are unstable. The transfer function poles are:

\[ s_{1,2} = 0, \quad s_{3,4} = \pm 49.0463i \quad s_{5,6} = \pm 2.3318 \times 10^2i \]

The pole zero map is shown in fig. 3.1.

The differential equation representing (3.16) is

\[
Num(1) y^{(4)}(t) + Num(2) y^{(3)}(t) + Num(3) y^{(2)}(t) + Num(4) y(t) + Num(5) = \\
= Den(1) y^{(6)}(t) + Den(2) y^{(5)}(t) + Den(3) y^{(4)}(t) + Den(4) y^{(3)}(t) + Den(5) y^{(2)}(t)
\]

(3.18)
End-effector motion planning for a one link non-minimum phase robot

Figure 3.1: Location of the zeros and poles for transfer function (3.16)

associated to the initial conditions:

\[
\tau(0) = \tau^{(1)}(0) = \tau^{(2)}(0) = \tau^{(3)}(0) = 0
\]

\[
y(0) = y^{(1)}(0) = y^{(2)}(0) = y^{(3)}(0) = y^{(4)}(0) = y^{(5)}(0) = 0
\]

As shown previously, the obtained input-output equation can be associated to any initial conditions. For this particular case, they have been assumed to be equal to zero.

The output that will cancel the unstable zero dynamics is now found. From (3.2), with \( p = 12 \) (2 unstable zeros plus 4 initial conditions on \( \tau \) plus 6 initial conditions on \( y \)) we have

\[
y_d = a_1 + a_2 \times t + a_3 \times t^2 + a_4 \times t^3 + a_5 \times t^4 + a_6 \times t^5 + a_7 \times t^6 + a_8 \times t^7 + a_9 \times t^8 + a_{10} \times t^9 + a_{11} \times t^{10} + a_{12} \times t^{11}
\]

(3.19)

\( y_d \) must satisfy 12 constraints:

- The indices \( A_i \) corresponding to the unstable zeros
- Initial and final conditions.

21
3.2 Stable inversion of the IST planar flexible link robot manipulator

\[ A_1 = 7_0 + \frac{a_1}{\text{zero}_1} + \frac{a_2}{\text{zero}_2} + \frac{a_3}{\text{zero}_3} + \frac{a_4}{\text{zero}_4} + \frac{a_5}{\text{zero}_5} + \frac{a_6}{\text{zero}_6} + \frac{a_7}{\text{zero}_7} + \frac{a_8}{\text{zero}_8} + \frac{a_9}{\text{zero}_9} + \frac{a_{10}}{\text{zero}_{10}} + \frac{a_{11}}{\text{zero}_{11}} + \frac{a_{12}}{\text{zero}_{12}} = 0 \]

\[ A_3 = 7_0 + \frac{a_1}{\text{zero}_3} + \frac{a_2}{\text{zero}_2} + \frac{a_3}{\text{zero}_3} + \frac{a_4}{\text{zero}_4} + \frac{a_5}{\text{zero}_5} + \frac{a_6}{\text{zero}_6} + \frac{a_7}{\text{zero}_7} + \frac{a_8}{\text{zero}_8} + \frac{a_9}{\text{zero}_9} + \frac{a_{10}}{\text{zero}_{10}} + \frac{a_{11}}{\text{zero}_{11}} = 0 \]

\[ y'(0) = 0 \quad i \in \{0, 1, 2, 3\} \]

\[ y(t_f) = y_f \]

\[ y'(t_f) = 0 \quad i \in \{1, 2, 3, 4, 5\} \]

Note that these conditions were chosen to force the desired torque to be symmetric. The desired \( a_i \) coefficients are then directly obtained, solving the linear system above. In this way it’s possible to calculate the transient solution of the system (3.4).

The next step is to obtain the particular solution of the system (3.6). The coefficients \( B_i \) are obtained as linear functions of the output coefficients \( a_i \), substituting (3.6) and (3.19) into (3.18) yields.

For the constant values:

\[ 720a_7Den(1) + 120a_6Den(2) + 24a_5Den(3) + 6a_4Den(4) + 3a_3Den(5) = 24Num(1)B_5 + 6Num(2)B_4 + 2Num(3)B_3 + Num(4)B_2 + Num(5)B_1 \]

For the coefficients of \( t \):

\[ 5040a_5Den(1) + 720a_7Den(2) + 120a_6Den(3) + 24a_5Den(4) + 6a_4Den(5) = 120Num(1)B_5 + 24Num(2)B_5 + 6Num(3)B_4 + 2Num(4)B_3 + Num(5)B_2 \]

For the coefficients of \( t^2 \):

\[ 20160a_9Den(1) + 2520a_8Den(2) + 360a_7Den(3) + 60a_6Den(4) + 12a_5Den(5) = 360Num(1)B_7 + 60Num(2)B_6 + 12Num(3)B_5 + 3Num(4)B_4 + Num(5)B_3 \]

For the coefficients of \( t^{12} \):

\[ Num(5)B_{12} = 0 \]

Solving this system of linear equations, we obtain all indices \( B_i \) for the particular solution (3.6). The resulting of desired output and input are shown in figs. 3.2 and 3.3.

As introduced in section 3.1, equation (3.10), in order to bring some robustness to this control, a closed-loop form should be used. Two types of control were chosen.
End-effector motion planning for a one link non-minimum phase robot

1. A partial state feedback, based on the joint position and velocity variables.

\[ T_{cl} = T_{ol} + K_p(\theta_d(t) - \theta(t)) + K_v(\dot{\theta}_d(t) - \dot{\theta}(t)) \]

\[ K_p > 0, \quad K_v > 0 \]

2. An open loop form where the input to the system is the angle of the joint.

### 3.3 Simulation Results

In this section we report some simulation results on the IST planar flexible manipulator shown in fig. 2.2.
3.3 Simulation Results

3.3.1 Control where the input is the torque

The method was applied for $t_f = 2.7s$ and $y_f = -0.35m$ (fig. 3.2 and fig. 3.3). The closed loop control was obtained using $K_p = 4$ and $K_v = 0.03$.

![Plot of simulated Error](image)

Figure 3.4: Simulation tracking error

![End-effector vibration](image)

Figure 3.5: Simulated displacement of the End-Effector

In figs. 3.4 and 3.5 we present the simulation tracking error and the corresponding displacement of the end-effector. After a brief analysis, we obtain the stationary error equal to $2.5 \times 10^{-7}m$, and the vibration on the end effector equal to about $2 \times 10^{-7}m$. Those are very small values compared to those obtained during the path evolution. The minor residual oscillations ($t > 2.5s$) verified in fig. 3.4 is due to the fact that the simulated model is quadratic in deformation [15], as explained in section 2.3.
End-effector motion planning for a one link non-minimum phase robot

![Graph showing simulated close-loop torque](image)

**Figure 3.6: Simulated close-loop torque**

Fig. 3.6 presents the output torque of the close-loop form. Indiscernible in this figure, are the small differences from the desired, due to the addition of the tracking error.

**Validation**

Hereafter the effects of exterior factors are simulated, such as noise, external torque input, and miss calculation of the transfer function zeros.

Introducing an input torque starting at $t = 1s$ and ending at $t = 2.4s$, with an amplitude $0.002Nm$, the following results were obtained. Analyzing fig. 3.7, fig. 3.8 and fig. 3.9 it is visible that the closed-loop control is very robust on eliminating the unstable zeros effects, even when there is an external input torque. Naturally there are some limitations. If the input torque is 10 times larger than the maximum torque ($0.1Nm$), the controller will have some problems to achieve the desired results (figs. 3.10, 3.11 and 3.12).
3.3 Simulation Results

Figure 3.7: Simulated output with an external input torque (Amplitude 0.002Nm)

Figure 3.8: Simulated close-loop torque with an external input torque (Amplitude 0.002Nm)
Figure 3.9: Simulation tracking error for an external input torque (Amplitude 0.002Nm)

Figure 3.10: Simulated close-loop torque with an external input torque (Amplitude 0.1Nm)
3.3 Simulation Results

Figure 3.11: Simulation output with an external input torque (Amplitude 0.1Nm)

Figure 3.12: Simulation tracking error for an external input torque (Amplitude 0.1Nm)
When analyzing the effect of noise in the system, a noise signal with 1% of the maximum standard input value was introduced in the closed-loop input signal and in the output measurement. The results are shown in figs. 3.13, 3.14 and 3.15. As expected, the noise affects the precision of the system, and produces an undesired response of the closed-loop torque.

Figure 3.13: Simulated close-loop torque with noise

Figure 3.14: Simulation output with noise
In order to analyze the miscalculation effect of the transfer function zeros, the following transfer function is used for controller design.

\[
\frac{y}{\tau} = \frac{101.8s^4 + 8.545 \times 10^{-12}s^3 - 4.924 \times 10^6s^2 - 3.317 \times 10^{-7}s + 1.5 \times 10^{10}}{s^2(s^4 + 1.066 \times 10^{-14}s^3 + 56780s^2 + 5.795 \times 10^{-10}s + 1.307 \times 10^8)}
\]  

(3.21)

where the coefficients of the numerator have been changed in 10% resulting in the new transfer function zeros:

\[\text{Zero}_{1,2} = \pm 212.732 \quad \text{Zero}_{3,4} = \pm 57.1573\]
Figure 3.17: Simulation output with miscalculated zeros

Figure 3.18: Simulation tracking error with miscalculated zeros
3.3 Simulation Results

Observing figs. 3.16 to 3.18, we can conclude that this miscalculation of the transfer function zeros is not very relevant, since the calculation of the input torque is obtained in a way that doesn’t excite a reasonable band of frequency. It is important to notice that the error shown in fig. 3.17 tends to a smaller value than the obtained from fig. 3.4 due to the increase of the static gain, originated by the new transfer function.

3.3.2 Control where the input of the system is the angle of the joint.

In this type of controller, the main idea is to make the system robust to external factors such as joint friction. Instead of calculating the transfer function between the torque and the position of the end-effector, the transfer function between the angle of the joint and the position of the end-effector has been calculated.

Considering the following control law

\[ \tau = k_p (\theta_r - \theta) - k_v \dot{\theta} \]  

(3.22)

associated to equation (3.16), we obtain the following transfer function

\[ \frac{y}{\dot{\theta}} = \frac{1330s^4 + 1.398 \times 10^{-9}s^3 - 7.079 \times 10^7s^2 + 7.493 \times 10^{-5}s + 1.672 \times 10^{11}}{s^6 + 334.3s^5 + 7.198 \times 10^7s^4 + 1.456 \times 10^9s^3 + 7.925 \times 10^8s^2 + 6.393 \times 10^7 + 2.907 \times 10^{11}} \]  

(3.23)

Calculating the desired angle in a way that the undesired transient response is eliminated, the following curve is obtained (fig 3.19).

The joint controller is obtained in the following steps.

1. Calculate the second moment of inertia of the rigid system

\[ I = J_{Hub} + m \times \left( \frac{L}{2} \right)^2 \]  

(3.24)

Where \( J_{Hub} \) is the second moment of inertia of the Hub and \( m \) and \( L \) are the mass and the length of the flexible part respectively. For this system \( I = 0.006kg/m^2 \).

2. Now it’s possible to obtain the derivative part of the controller since:

\[ \frac{\dot{\theta}}{\tau} = \frac{1}{0.006 \times s} \]  

(3.25)
which in closed loop becomes

\[ G_{cl} = \frac{K_v}{0.006 \times s + K_v}, \]  

\[ \text{(3.26)} \]

where \( K_v \) is the gain of the derivative part.

3. The desired pole is:

\[ P = -\frac{K_v}{0.006 \times 2} \]

4. The calculation of the proportional part of the controller is obtained by the following transfer function:

\[ G_{clp} = \frac{K_p \times K_v}{0.006 \times s^2 + K_v \times s + K_p \times K_v} \]

\[ \text{(3.27)} \]

5. Solving two equations with two variables:

\[ \begin{cases} 
K_p = -\frac{P}{2} \\
K_v = -0.0055 \times 2 \times P 
\end{cases} \]

\[ \text{(3.28)} \]

where \( P \) is the position of our desired pole. (Note that \( P \) is always negative)

The following results were obtained using \( P = -50 \). Fig. 3.20 shows the representation on a block diagram of the PD controller.

By observing figs. 3.21 to 3.23, we notice that the stationary error has been reduced to an insignificant value. Using this controller the system has become more robust to external factors such as joint friction as shown next, and reduces the residual vibration in the steady state.
3.3 Simulation Results

Figure 3.20: PD controller

Figure 3.21: Simulated output

Figure 3.22: Simulated end-effector deformation
Validation

As in the previous validation, the effects of external factors, such as noise, external torque input, and miss calculation of the transfer function zeros will be analyzed here.

Introducing an input torque starting at $t = 1$ s and ending at $t = 2.4$ s, with an amplitude of $0.002 N/m$, the obtained results are.

Analyzing figs. from 3.24 to 3.26 it’s visible that this control is very robust on eliminating the effect of the unstable zeros, even when there is an external input torque. The main difference between these
3.3 Simulation Results

Figure 3.25: Simulated close-loop torque with an external input torque (Amplitude 0.002Nm)

Figure 3.26: Simulation tracking error for an external input torque (Amplitude 0.002Nm)
controllers is when the undesired torque increases. Setting the perturbation torque to $0.1\,Nm$, the following results are achieved. Comparing figs. 3.27 to 3.29 with figs. 3.10 to 3.12, it’s clear that this controller has become much more robust than the previous one. Also by analyzing fig. 3.28, it’s observable where the external input was applied ($t=1s$ and $t=1.4s$).

![Plot of simulated Y](image)

**Figure 3.27:** Simulated output with an external input torque (Amplitude $0.1\,Nm$)

![Plot of simulated Closed-loop Torque](image)

**Figure 3.28:** Simulated close-loop torque with an external input torque (Amplitude $0.1\,Nm$)

Analyzing now the effect of noise in this controller, we submit the system to the same noise tests as the previous controller. The results in figs. 3.30, 3.31, 3.32 have been obtained. As in the previous controller, it is clear that the effect of the noise isn’t significant to the output value.
3.3 Simulation Results

Figure 3.29: Simulation tracking error for an external input torque (Amplitude 0.1Nm)

Figure 3.30: Simulated close-loop torque with noise
End-effector motion planning for a one link non-minimum phase robot

Figure 3.31: Simulation output with noise

Figure 3.32: Simulation tracking error with noise
3.3 Simulation Results

To analyze the effect of the miscalculation of the transfer function zeros, the following transfer function was introduced in order to obtain the input angle.

\[
y \theta = \frac{1463s^4 + 1.258 \times 10^{-9}s^3 - 6.371 \times 10^7s^2 + 8.242 \times 10^{-5}s + 1.505 \times 10^{11}}{(s^6 + 334.3s^5 + 7.198s^4 + 1.456 \times 10^7s^3 + 7.925 \times 10^8s^2 + 6.396 \times 10^9s + 2.907 \times 10^{11})} (3.29)
\]

where the new transfer function zeros are:

\[
\text{Zero}_{1,2} = \pm 202.5884 \quad \text{Zero}_{3,4} = \pm 50.0614
\]

and the new transfer function poles are:

\[
s_{1,2} = -15.0428 \pm 213.3005i \quad s_{3,4} = -0.5513 \pm 19.6926i
\]

\[
s_5 = -232.7184 \quad s_6 = -70.3933
\]

The location of the zeros and the poles is represented in fig. 3.33:

Figure 3.33: location of zeros and poles in a closed loop form for miss calculation of transfer function

As before the miscalculation of the transfer function zeros for this case does not pose a significant problem.
Figure 3.34: Simulated closed-loop torque with miscalculated zeros

Figure 3.35: Simulation output with miscalculated zeros
Figure 3.36: Simulation tracking error with miscalculated zeros
Chapter 4

Path-Following for one link non-minimum phase robot

The objective of the Path-Following method is to force the non-minimum phase system output to follow a geometric path without a timing law assigned to it. As presented in chapter 1, systems with unstable zero dynamics have limited tracking capabilities. The only way to overcome this performance limitation is to change the input-output structure of the system. This structure can be changed by reformulating the problem as path-following, rather than reference tracking. With this reformulation, it is possible to add a new timing law $\gamma(t)$ that becomes an additional control input. In this chapter we define the path-following problem, and present two methodologies to solve it. The first one is called Internal model control, and it has the goal to achieve asymptotic tracking of reference signals; it is demonstrated by Aguiar et al. [2]. The controller that incorporates an internal model of the exosystem is capable to ensure an asymptotic convergence of the tracking error to zero for every possible reference signal generated by the exosystem. The second methodology uses the path-following controller to achieve three requirements, a geometric task, a dynamic task and boundedness of the zero dynamics states.

4.1 Path-Following method with internal model control

The Internal model approach originated in the output regulation problem.
4.1 Path-Following method with internal model control

4.1.1 Problem Statement

As previously described, the following linear time-invariant system is assumed:

\[
\begin{align*}
\dot{x}(t) &= Ax(t) + Bu(t) \quad x(t_0) = x_0 \\
y(t) &= Cx(t) + Du(t)
\end{align*}
\]  

(4.1)

where \(x(t)\) is the state, \(u(t)\) is the input, and \(y(t)\) is the output. The main objective of this method is to reach and follow a desired geometric path \(y_d(\gamma)\). The geometric path \(y_d(\gamma)\) can be generated by an exosystem of the form:

\[
\begin{align*}
\frac{d}{d\gamma} w(\gamma) &= S \times w(\gamma) \quad w(\gamma_0) = w_0 \\
y_d(\gamma) &= Q \times w(\gamma)
\end{align*}
\]

(4.2)

where \(w \in \mathbb{R}^{2n}\) is the exogenous state and \(S + S^T = 0\). For any timing law \(\gamma(t)\), the path-following error can be defined as:

\[
e(t) = y(t) - y_d(\gamma(t))
\]

(4.3)

The following problems can be associated to the Path-Following methodology:

**Geometric path-following:** For the desired path \(y_d(\gamma)\), it is necessary to design a controller that achieves:

- **Boundedness:** the state \(x(t)\) is uniformly bounded for all \(t \geq t_0\), and every initial condition \((x(t_0), w(\gamma_0)), \gamma_0 = \gamma(t_0)\).
- **Error convergence:** the path-following error \(e(t)\) converges to zero as \(t \to \infty\).
- **Forward motion:** \(\dot{\gamma}(t) > c\) for all \(t \geq t_0\), where \(c\) is a positive constant.

**Speed-assigned path-following:** Given a desired speed \(v_d > 0\), it is required that \(\dot{\gamma} \to v_d\) as \(t \to \infty\).

As demonstrated by Aguiar et al. [2] and Aguiar [4], we can always assume a small \(L_2\)-norm of the path following error,

\[
J = \int_0^\infty \|y(t) - y_d(t)\|^2 dt = \int_0^\infty \|e(t)\|^2 dt < \delta
\]

(4.4)

that verifies a \(\delta\) arbitrarily small in order to consider a perfect tracking problem.

4.1.2 Controller Design - Internal model control

One way to control the non-minimum phase system is represented in fig. 4.1. Aguiar et al. [2] presented...
one solution to achieve a path controller for (4.1), such that the close loop state is bounded.

If \((A, B, C, D)\) is a non-minimum phase system, the pair \((A, B)\) is stabilizable, the pair \((C, A)\) is detectable, the number of inputs is as large as the number of outputs \((m \geq q)\) and the zeros of \((A, B, C, D)\) do not coincide with the eigenvalues of \(S\) (4.2), then for the geometric path-following problem there exist constant matrices \(K\) and \(L\), and a timing law \(\gamma(t)\) such that the feedback law is:

\[
u(t) = Kx(t) + L(\dot{\gamma}_d)w(\gamma(t))
\] (4.5)

To calculate the matrices \(K\) and \(L\), the following internal model approach is considered:

\[
\begin{align*}
\Pi & = AI + BD \\
0 & = CI + DT - Q
\end{align*}
\] (4.6)

As shown before, the Sylvester equations (4.6), are solvable if the system \((A, B, C, D)\) is right-invertible and its zeros do not coincide with the eigenvalues of \(v_dS\). The methodology to solve this equation is described as follows:

- Transform system (4.6) into the following system:

\[
\begin{align*}
NewA & = \begin{bmatrix}
Kron(I_{n_x}, A) - Kron(S', I_{n_u}) & Kron(I_{n_x}, B) \\
Kron(I_{n_x}, C) & Kron(I_{n_x}, D)
\end{bmatrix} \\
NewB & = \begin{bmatrix} 0 \\ Q \end{bmatrix}
\end{align*}
\] (4.7)

where \(n_x\) is the size of the square matrix \(S\) and \(n_a\) is the size of the square matrix \(A\).
4.1 Path-Following method with internal model control

- From equations (4.7) and (4.6) the following formula is obtained

\[
\begin{bmatrix}
\Pi \\
\Gamma
\end{bmatrix} = [NewA]^{-1} [NewB] \tag{4.8}
\]

- Since we now have \( \Pi \) and \( \Gamma \) it is possible to obtain the controller gains \( K \) and \( L \). \( K \) is calculated by a minimum quadratic regulator, that minimizes the quadratic cost function,

\[
J (u) = \int_{0}^{\infty} (x^T U x + u^T R u + 2x^T N u) \, dt \tag{4.9}
\]

and \( L \) is equal to:

\[
L = \Gamma - K \times \Pi \tag{4.10}
\]

Now that the path controller design is complete, an evolution rule to \( \gamma \) has to be created, in a way that:

- \( \lim_{t \to \infty} \gamma = \gamma_d \)
- \( \lim_{t \to \infty} \dot{\gamma} = v_d \)

The \( \gamma \) controller has been developed assuming the error function equal to

\[
e = \gamma - \gamma_d \tag{4.11}
\]

Considering a first cost function as

\[
V_1 = \frac{1}{2} e \tag{4.12}
\]

assuming \( \dot{\gamma}_d = v_d (\gamma_d) \), the derivative of equation (4.12) yields:

\[
\dot{V}_1 = e [\dot{\gamma} - v_d (\gamma_d)] \tag{4.13}
\]

Considering now as control law the following equation

\[
\dot{\gamma} = \frac{dy_d}{d\gamma} - k_1 e \tag{4.14}
\]

with the new error variable \( Z_1 \)

\[
Z_1 = \dot{\gamma} - \frac{dy_d}{d\gamma} + k_1 e \Leftrightarrow \dot{\gamma} = Z_1 + \frac{dy_d}{d\gamma} - k_1 e \tag{4.15}
\]
finally the first cost function result is obtained

\[ \dot{V}_1 = e \left[ Z_1 + \frac{dy}{d\gamma} - k_1 e - \frac{dy}{d\gamma} \right] \Leftrightarrow \dot{V}_1 = -k_1 e^2 + Z_1 \quad (4.16) \]

To ensure that the system evolves to the desired path, another cost function involving the velocity error must be assumed

\[ V_2 = V_1 + \frac{1}{2} Z_1^2 \quad (4.17) \]

obtaining

\[ \dot{V}_2 = -k_1 e^2 + Z_1 \left[ e + \frac{d^2 y_d}{d\gamma^2} \right] \quad (4.18) \]

Using as control law

\[ \frac{d^2 y_d}{d\gamma^2} = \frac{d^2 y_d}{d\gamma^2} + e + k_2 Z_1 \quad (4.19) \]

finally results

\[ \ddot{\gamma} = \frac{d^2 y_d}{d\gamma^2} - e - k_2 \left[ \dot{\gamma} - \frac{d y_d}{d\gamma} \right] \quad (4.20) \]

### 4.1.3 Simulation results

The variable \( y_d \) was used as a desired path output. Two kinds of paths were computed:

The first path is

\[ y_d = \left( 1 - e^{(-w_d \times \gamma)} \times (1 + w_d \times \gamma) \right) y_f \]

\[ \frac{dy}{d\gamma} = \left( w_d e^{(-w_d \times \gamma)} \times (1 + w_d \times \gamma) - w_d e^{(-w_d \times \gamma)} \right) y_f \quad (4.21) \]

\[ \frac{d^2 y_d}{d\gamma^2} = \left( -w_d^2 e^{(-w_d \times \gamma)} \times (1 + w_d \times \gamma) + 2w_d^2 e^{(-w_d \times \gamma)} \right) y_f \]

where the variables \( w_d \) and \( y_f \) set the convergence velocity of the system to the final value \( y_f \).

Using the same path example as in [18], the second path is

\[ y_d = y_f \sin \left( \frac{\pi}{2} \frac{\gamma}{\gamma} \right) ; \gamma \in [0, 1] \]

\[ \begin{align*}
&v_a = \begin{cases} 
\frac{m_s}{\pi} \frac{dx}{d\gamma} \arctan \left( \frac{\gamma - a_1}{a_2} \right) + \frac{m_s}{2 \frac{dx}{d\gamma}} \quad \gamma \in [0, 0.5], \\
\frac{m_s}{\pi} \frac{dx}{d\gamma} \arctan \left( \frac{1 - \gamma - a_1}{a_2} \right) + \frac{m_s}{2 \frac{dx}{d\gamma}} \quad \gamma \in [0.5, 1]
\end{cases} \\
&w_a = \begin{cases} 
\frac{m_s}{\pi} \frac{dx}{d\gamma} \frac{a_3}{a_1^2 + (\gamma - a_1)^2} \quad \gamma \in [0, 0.5], \\
-\frac{m_s}{\pi} \frac{dx}{d\gamma} \frac{a_3}{a_1^2 + (1 - \gamma - a_1)^2} \quad \gamma \in [0.5, 1]
\end{cases}
\end{align*} \quad (4.22) \]
4.1 Path-Following method with internal model control

where the parameter $a_1$ sets the width of the low speed regions at the beginning and the end of the path, while $a_2$ smoothens the square wave, and the $m_s$ variable constrains the maximum speed. As expected, the obtained path in $x,y$ coordinates is the arc represented in fig. 4.2.

![Plot of position](image_url)

Figure 4.2: The path in $x,y$ coordinates

For the results presented in the next section, the following data was used. The system state space representation in continuous mode is:

$$A = \begin{bmatrix} 0 & 0 & 0 & 16040.41 & 9038.62 \\ 0 & 0 & 0 & 1143.68 & 2422.93 \\ 0 & 0 & 0 & -81331.5 & -57923.19 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$ (4.23)

$$B = \begin{bmatrix} 607.9 & -296.4 & -364.7 & 0 & 0 & 0 \end{bmatrix}^T$$

$$C = \begin{bmatrix} 0 & 0 & 0 & 0.575 & 1 & 0 \end{bmatrix}$$

$$D = 0$$

while in this representation, the states are represented in the following way.

$$q = \begin{bmatrix} \dot{\theta} \\ \dot{\eta}_1 \\ \dot{\eta}_2 \\ \theta \\ \eta_1 \\ \eta_2 \end{bmatrix}$$ (4.24)
where the variable $\theta$ represents the joint angle. For the controller calculus we used the Matlab LQR command, which requires the matrix $A$ and $B$ equation (4.23) and the matrix $Q$ and $R$ with the following values

$$Q = 100 \times C^T \times C$$

$$R = 0.001 \times I$$

which result in the following gain state space matrix

$$K = \begin{bmatrix} -10.2518 & -16.8531 & -2.3238 & -181.8310 & -126.2391 & 8.1224 \end{bmatrix}$$

(4.26)

The value of the controller $L$ is calculated for different values of speed assignments between 0m/s and 5m/s. The simulation results for the paths in question, refered in equations (4.21) and (4.22) will be treated in the following section.

First path simulation results

The fist path simulation has the following properties:

- A final $y$ value $y_f = \pi/8$ in a five seconds simulation
- $\omega_d$ equal to 20.

In fig. 4.3 we present the desired $y_d$ versus the $\gamma$ variable.

Figure 4.3: First Path - Evolution of the desired output versus the path variable
4.1 Path-Following method with internal model control

In figs. 4.4 and 4.5 we present the simulation tracking error and the corresponding deformation of the end-effector. The tracking error and the end-effector deformation converge to zero value, as expected.

![Figure 4.4: First Path - Simulation tracking error](image1)

![Figure 4.5: First Path - Deformation of the end-effector](image2)

Fig. 4.6 presents the evolution of the path variable during the simulation period. In this figure, some initial variations are visible due to the integrators initial conditions.

50
Second path simulation results

The second path shows better results in the end-effector deformation domain, because the path is smoother and there was no need to introduce initial conditions in the integrators. This was due to the characteristics of the $v_s$ variable which already has initial values, so the path could evolve naturally. Also, the mean velocity is larger in the first path than in the second path, which could affect the end-effector deformation results.

By changing the variables of equation (4.22), two paths will be produced. First, $a_1 = 0.1$, $a_2 = 0.1$, $m_s = 0.1$ and $y_f = 0.575 \times \pi/2$ will be considered. This means that the end-effector will produce a 90 degree rotation. As it is seen in fig. 4.7, the desired velocity profile has non-zero initial and final conditions.
So there is no need to introduce integrator’s initial conditions.

In fig. 4.8 the simulation tracking error is presented. It is visible that the maximum error appears at the maximum speed, and stabilizes to a very small value near zero. One thing that can be highlighted is the fact that the final value of the error is negative. This means that the system has passed \( y_f \) by a small value, and it’s cause can be due to the acceleration profile, since it has non zero initial and final values.

![Figure 4.8: Second Path - Simulation tracking error](image)

Fig. 4.9 presents the evolution of the path variable during the simulation period. Comparing to fig. 4.6 it is observable that this path is smoother, and the control variable has better results. As in fig. 4.8 the final values of the path variable (fig. 4.9) can be due to the acceleration profile. This kind of property appeared to be no problem, since this controller is very robust to all kind of signals, as it will be shown in the validation section.

Analyzing fig. 4.10, it is observed that the end-effector deformation is very small, and this method, with this path is capable of completely removing it. Other tests were made demonstrating that the deformation does not depend on how long the path is, opposing the results in chapter 3.

For the second path example, the robustness of the controller was studied for different changes of the velocity profile. Considering the path in equation (4.22), with \( a_1 = 0.1, \ a_2 = 1, \ m_s = 0.2 \) and \( y_f = 0.575 \times \pi/2 \), the end-effector will produce a 90 degree rotation.

With these variables, the velocity profile presented in fig. 4.11 has higher maximum velocity in a
shorter period of time. With a normal controller, a bigger end-effector deformation should be expected. Nevertheless, this controller defines the system velocity in order to reduce the error and the end-effector deformations, so, in contrary to what should be expected, the error and end-effector deformations values didn’t change. Comparing fig. 4.12 with 4.10 and fig. 4.13 with 4.8 we observe the end-effector deformation remains similar, except for the undershoot at the beginning of the path. As in the previous example (fig. 4.10), the controller could remove all system deformation. Analyzing the tracking error, all the characteristics remain (stationary error and values).
4.1 Path-Following method with internal model control

Figure 4.11: Second Path - Simulation velocity profile

Figure 4.12: Second Path - Deformation of the end-effector

Figure 4.13: Second Path - Simulation tracking error
Validation

Like before, the effects of external factors, such as external input torque, will be studied.

- Consider the path equation (4.21), with \( w_d = 7 \) and \( y_f = \pi/8 \).

An input torque white noise perturbation with the noise power of 0.1 and sample time of 0.1s has been introduced. The external perturbation torque is 5 times greater than the normal input torque (fig. 4.14). These values can be considered as a reasonable external perturbation.

![Plot of the input torque](image)

Figure 4.14: First Path validation - Simulation input torque (White noise with the noise power of 0.1)

Despite the high input perturbation, the end-effector deformation has reasonable values. Observing fig. 4.16, after \( t = 1s \), one verifies an increase of end-effector perturbation, due to the velocity decrease, despite, by observation of fig. 4.15 the end-effector deformation remains unchanged.

In fig. 4.17 it’s visible, in the first 2 seconds, the controller management between the leader position and the system state boundedness.
4.1 Path-Following method with internal model control

Figure 4.15: First Path validation - Deformation of the end-effector (White noise with the noise power of 0.1)

Figure 4.16: First Path validation - Simulation output (White noise with the noise power of 0.1)
Path-Following for one link non-minimum phase robot

- Considering the path in equation (4.22), with $a_1 = 0.1$, $a_2 = 0.1$, $m_s = 0.1$ and $y_f = 0.575 \times \pi/2$.

A step input torque with the periodicity of 2s and amplitude of 1Nm has been added to the system input.

- Figure 4.17: First Path validation - Simulation tracking error (White noise with the noise power of 0.1)

- Figure 4.18: Second Path validation - Simulation tracking error (External step input torque with amplitude of 1Nm)

In figs. 4.19 to 4.20, the ability of this controller to eliminate external input torques perturbations can be observed. The only negative point is the static error. In fig. 4.18 the controller’s difficulties to eliminate the induced static error are presented.
4.1 Path-Following method with internal model control

Figure 4.19: Second Path validation - Simulation input torque (External step input torque with amplitude of 1Nm)

Figure 4.20: Second Path validation - Deformation of the end-effector (External step input torque with amplitude of 1Nm)
Next are presented a series of tests in order to validate the controller in terms of extreme external input torque, and their implications on the path. The external input torque was changed to white noise with the noise power of 1 and sample time of 0.1s, and the path properties remain as $a_1 = 0.1$, $a_2 = 0.1$, and $m_s = 0.1$.

![Figure 4.21: Second Path validation - Simulation Tracking error (White noise with the noise power of 1)](image)

Since the obtained results were very similar to the results of figs. 4.22 to 4.27, there is only one figure presented (fig. 4.21). The path properties for figs. 4.22 to 4.27 are $a_1 = 1$, $a_2 = 0.1$ and $m_s = 1$. The external input torque remains the same.

![Figure 4.22: Second Path validation - Simulation input torque (Path properties: $a_1 = 1$, $a_2 = 0.1$ and $m_s = 1$)](image)
4.1 Path-Following method with internal model control

It is also visible that in figs. 4.23, 4.24 and 4.26, the deformation of the end-effector and the path error values increase, at the same time as the desired theta limitation occurs.

Figure 4.23: Second Path validation - Deformation of the end-effector (Path properties: $a_1 = 1$, $a_2 = 0.1$ and $m_s = 1$)

Figure 4.24: Second Path validation - Simulation tracking error (Path properties: $a_1 = 1$, $a_2 = 0.1$ and $m_s = 1$)

In fig. 4.25, the $\gamma$ variable signal after $t = 6.5s$ presents a strange behavior. This is caused by limitation of $\gamma$ in the interval $[0, 1]$. Imposing a limitation into the $\gamma$, a control value limitation was preset, increasing the path-variable error. It is also visible that in figs. 4.23, 4.24 and 4.26, the deformation of the end-effector and the path error values increase, at the same time as the desired path variable limitation occurs.
Comparing this simulation (fig. 4.24) with the one which has a better velocity profile (fig. 4.21), one realizes there isn't much difference between them. This is due to the controller's ability to manage between the leader tracking and the boundedness of the system states. In order to achieve a better boundedness of system states, the controller sacrifices the dynamical and geometrical tasks.

![Plot of γ Error](image)

**Figure 4.25:** Second Path validation - Evolution of the desired path variable versus the real path variable (Path properties: \(a_1 = 1, a_2 = 0.1\) and \(m_s = 1\))

![Simulation output](image)

**Figure 4.26:** Second Path validation - Simulation output (Path properties: \(a_1 = 1, a_2 = 0.1\) and \(m_s = 1\))
4.2 Normal Form Path Following

Figure 4.27: Second Path validation - Joint velocity versus error (Path properties: $a_1 = 1$, $a_2 = 0.1$ and $m_s = 1$)

4.2 Normal Form Path Following

Since with the previous controller the end-effector position wasn’t feedback to the controller, a new methodology was introduced. Since the goal problem of this thesis is to control a flexible manipulator robot from point a to point b, and all the bibliography work considers path-following at constant velocity, the adaptation from one problem to another is not straightforward. Nevertheless, the Normal Form path following methodology will be presented here.

Assuming equation (4.1) with uniform relative degree $r$, and the path $y_d$, the goal is to construct feedback laws for $u$ and for $\gamma^{(r)}$, the $r^{th}$ derivative of $\gamma$, such that the closed-loop solution satisfy the following requirements.

R1 Geometric task:

$$\lim_{t \to \infty} \sup \| y(t) - y_d(\gamma(t)) \| = 0$$

R2 Dynamic task:

$$\dot{\gamma}(t) \geq 0 \quad \text{and} \quad \lim_{t \to \infty} \gamma(t) = \infty$$

R3 Boundedness of the zero dynamics states

System (4.1) is refered as the follower, since it is the one that asymptotically converges to the leader
Path-Following for one link non-minimum phase robot

for any admissible leader’s motion. The constraint \( \lim_{t \to \infty} \gamma(t) = \text{constant} \) is introduced to disqualify feedback laws which force the followers output to converge to a point on the path \( y_d \). By imposing \( \dot{\gamma}(t) \geq 0 \), the followers output is restricted to the forward motion along the path \( y_d \). One of the key features of this method is the use of the additional feedback law for \( \gamma^r \).

The starting point for the SISO system controller design is the normal form representation of the follower (Appendix A).

\[
\dot{z} = A_z z + B_z y \tag{4.27}
\]

\[
\dot{\zeta} = A_{r,m} \zeta + B_{r,m} \hat{u}, \quad \zeta \triangleq \begin{bmatrix} \zeta_1 & \cdots & \zeta_n \end{bmatrix}^T \tag{4.28}
\]

\[
y = C_{r,m}, \quad y \triangleq \begin{bmatrix} y_1 & \cdots & y_m \end{bmatrix}^T \tag{4.29}
\]

where subsystem (4.27) represents the zero dynamics driven by the output \( y \) exhibiting the transmission zeros as the eigenvalues of \( A_z \). Subsystem (4.28) represents \( m \) chains of \( r \) integrators relating the input \( \hat{u} \) and the output \( y \).

The input variable \( \hat{u} \) is given by:

\[
\hat{u} = \begin{bmatrix} u & \zeta^T & z^T \end{bmatrix} \begin{bmatrix} 1 & T_u^T \ & B_z^T \end{bmatrix}^T \tag{4.30}
\]

where \( T_u \) is an input transformation vector, and \( B_z \) is a state transformation vector. The methodology behind this representation can be found in [19]. There, it is explained the structural decomposition of MIMO and SISO systems to strictly proper systems (Normal form representation).

The trajectories of system (4.27) - (4.29) are related to the leader’s motion via the error coordinates \( y_e = y - y_d(\gamma) \). Furthermore, we define

\[
e = \begin{bmatrix} e_1 \\ e_2 \\ \vdots \\ e_m \end{bmatrix} \triangleq \begin{bmatrix} \zeta_1 - y_d^1(\gamma) \\ \zeta_2 - y_d^2(\gamma) \\ \vdots \\ \zeta_m - y_d^m(\gamma) \end{bmatrix}, \quad y_{di}(\gamma) \triangleq \begin{bmatrix} y_{di}(\gamma) \\ \dot{y}_{di}(\gamma) \\ \ldots \end{bmatrix}, \quad i = 1, \ldots, m \tag{4.31}
\]

The main distinguishing feature of the path-following problem is that \( \ddot{y}_d(\gamma) = \frac{\partial y_d}{\partial \gamma} \dot{\gamma}, \quad \dddot{y}_d(\gamma) = \frac{\partial^2 y_d}{\partial \gamma^2} \dot{\gamma}^2 + \frac{\partial y_d}{\partial \gamma} \ddot{\gamma}, \ldots \) give raise to \( r \) leader’s states \( \gamma \ldots \gamma^{(r-1)} \), and \( \gamma^{(r)} \) becomes the additional control input \( w \) which controls the leader’s motion in the following system.
\[ \dot{z} = A_z z + B_z (y_e + y_d (\gamma)) \quad (4.32) \]

\[ \dot{e} = A_{r,m} e + B_{r,m} \left( \hat{u} - y_d^{(r)} (\gamma) \right), \quad (4.33) \]

\[ \gamma^{(r)} = w \quad (4.34) \]

Subsystem equation (4.34) is referred as the leader, since it influences the followers zero dynamics equation (4.32) through the bounded function \( y_d \). The ability to stabilize the zero dynamics via a feedback law for \( w \) depends on the geometric properties of the path \( y_d(\gamma) \).

The controller approach is to decouple the task \( R3 \) of keeping \( z(t) \) bounded, from the geometric task \( R1 \). Task \( R3 \) is accomplished by designing a feedback law for \( \gamma^{(r)} = w \), leaving the original control variable \( \hat{u} \) in (4.33) to be used for \( R1 \). Due to boundedness of \( y_d \), the feedback law for \( w \) can only achieve \( R3 \) in a compact set. The path-following variable will be decomposed into \( \gamma(t) = \hat{\gamma}(t) + \tilde{\gamma}(t) \) where \( \hat{\gamma}(t) = \hat{\gamma}_k \), \( t \in [kT, (k + 1)T] \), being \( T \) the sampling period. \( \hat{\gamma}(t) \) is the inter-sample correction to be designed. The resulting sampled-data equation of the zero dynamics is

\[ z_{k+1} = A_z z_k + B_z v_k + d_{ek} + d_{sk} \quad (4.35) \]

To any sampling period \( T \), \( z_k \triangleq z(kt) \), \( A_z = e^{A_z T} \), \( B_z = \int_0^T e^{A_z s} ds B_z \), \( v_k \triangleq y_d(\hat{\gamma}_k) \) and \( d_{ek} \) and \( d_{sk} \) are disturbances due to \( y_e(t) \).

\[ d_{ek} = \int_0^T e^{A_z (T - s)} B_z y_e (s + kT) \, ds \quad (4.36) \]

\[ d_{sk} = \int_0^T e^{A_z (T - s)} B_z [y_d(\gamma(s + kT)) - v_k] \, ds \quad (4.37) \]

The repeatable path and system (4.35) are essential to construct a stabilizing feedback law \( \gamma^{(r)} = w \) for the followers zero dynamics equation (4.32), subject to the dynamic task \( R2 \). In [5], the methodology to obtain the controller is explained in more detail. Like the previous controller, the control actuation
is divided in two parts. The first part ensures boundedness of $z_k$, and the second part of the controller ensures the leader control. The obtained result is:

$$u = -k_c e + y_d^{(r)} \gamma$$  \hspace{1cm} (4.38)

where $k_c \in R^{m \times m}$ is chosen such that $A_{r,m} - B_{r,m}K_c$ is Hurwitz.

### 4.3 Parameter estimation

All the previous path-following control methods require the system to be in a state space representation. Since the common identification Matlab® toolboxes can only identify the system in a state space representation, we can not guarantee that those spaces have any physical meaning. So, a different approach had to be considered, in a way that it was possible to guarantee that the states have some physical meaning, which could be measured by sensors.

The purpose of this system identification is to determine the linear model, so that the state vector be represented like equation (2.6). The IST manipulator model can be written in the following form (equation (4.39)):

$$M\ddot{q} + C\dot{q} + Kq = T$$  \hspace{1cm} (4.39)

Where the $M$, $C$ and $K$ are all a 3 by 3 matrix, $q$ represent the generalized coordinates, and $T$ the input torque vector. Here we have included a damping matrix $C$. Furthermore to contemplate any type of shape functions, the $M$ matrix is assumed with the following structure

$$M = \begin{bmatrix} M_{11} & M_{12} & M_{13} \\ M_{21} & M_{22} & M_{23} \\ M_{31} & M_{32} & M_{33} \end{bmatrix}$$  \hspace{1cm} (4.40)

and the $C$ matrix

$$C = \begin{bmatrix} C_{11} & 0 & 0 \\ 0 & C_{22} & C_{23} \\ 0 & C_{32} & C_{33} \end{bmatrix}$$  \hspace{1cm} (4.41)

and finally the $K$ matrix

$$K = \begin{bmatrix} 0 & 0 & 0 \\ 0 & K_{22} & K_{23} \\ 0 & K_{32} & K_{33} \end{bmatrix}$$  \hspace{1cm} (4.42)
4.3 Parameter estimation

The $q$ vector represents the various generalized coordinates

$$q = [\theta \eta_1 \eta_2]^T \quad (4.43)$$

and the $T$ vector represents the system input

$$T = [\tau 0 0]^T \quad (4.44)$$

Transforming the system equations into state space representation we obtain

$$\begin{bmatrix} \dot{q} \\ \ddot{q} \end{bmatrix} = \begin{bmatrix} 0_{3x3} & I_{3x3} \\ -M^{-1}K & -M^{-1}C \end{bmatrix} \begin{bmatrix} q \\ \dot{q} \end{bmatrix} + \begin{bmatrix} 0_{3x3} \\ M^{-1} \end{bmatrix} T \quad (4.45)$$

Just for simplicity we will make the following representation (4.46).

$$\dot{Z} = \begin{bmatrix} \dot{q} \\ \ddot{q} \end{bmatrix} ; \quad Z = \begin{bmatrix} q \\ \dot{q} \end{bmatrix}$$

$$A = \begin{bmatrix} 0_{3x3} & I_{3x3} \\ -M^{-1}K & -M^{-1}C \end{bmatrix} ; \quad B = \begin{bmatrix} 0_{3x3} \\ M^{-1} \end{bmatrix} \quad (4.46)$$

Now, the system is represented as

$$\dot{Z} = AZ + BT \Leftrightarrow \dot{Z} = \begin{bmatrix} A & B \end{bmatrix} \begin{bmatrix} Z \\ T \end{bmatrix} \quad (4.47)$$

which in detail becomes

$$\begin{bmatrix} \dot{Z}_1 \\ \dot{Z}_2 \\ \dot{Z}_3 \\ \vdots \\ \dot{Z}_6 \end{bmatrix} = \begin{bmatrix} A & B \end{bmatrix}_{6 \times 9} \begin{bmatrix} Z_1 \\ \vdots \\ Z_6 \\ T_1 \\ \vdots \\ T_3 \end{bmatrix} \quad (4.48)$$

For $n$ sample times during trajectory execution, data for $q$, $\dot{q}$, $\ddot{q}$ and $\tau$ must be acquired. Renaming
\[ M^{-1}_{ij} = G_{ij} \], from equation (4.45) and (4.48) the following equation is obtained:

\[
\begin{bmatrix}
\dot{\theta} \\
\eta_1 \\
\dot{\eta}_2 \\
\ddot{\theta} \\
\ddot{\eta}_1 \\
\ddot{\eta}_2
\end{bmatrix} =
\begin{bmatrix}
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & -K_{22}G_{12} - K_{23}G_{13} & -K_{23}G_{12} - K_{33}G_{13} & 0 \\
0 & -K_{22}G_{22} - K_{23}G_{23} & -K_{23}G_{22} - K_{33}G_{23} & 0 \\
0 & -K_{22}G_{32} - K_{23}G_{33} & -K_{23}G_{32} - K_{33}G_{33} & 0
\end{bmatrix}
\begin{bmatrix}
\theta \\
\eta_1 \\
\eta_2 \\
\dot{\theta} \\
\dot{\eta}_1 \\
\dot{\eta}_2
\end{bmatrix}
\]

Simplifying equation (4.49) yields

\[
\begin{bmatrix}
\dot{\theta} \\
\eta_1 \\
\dot{\eta}_2 \\
\ddot{\theta} \\
\ddot{\eta}_1 \\
\ddot{\eta}_2
\end{bmatrix} =
\begin{bmatrix}
0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & H_{12} & H_{13} & J_{11} & J_{12} & J_{13} & -G_{11} \\
0 & H_{22} & H_{23} & J_{21} & J_{22} & J_{23} & -G_{12} \\
0 & H_{32} & H_{33} & J_{31} & J_{32} & J_{33} & -G_{13}
\end{bmatrix}
\begin{bmatrix}
\theta \\
\eta_1 \\
\eta_2 \\
\dot{\theta} \\
\dot{\eta}_1 \\
\dot{\eta}_2
\end{bmatrix}
\]

therefore using a Recursive Least Square (RLS) method (Appendix B), where all the data acquired during the time of trajectory execution was used, we may obtain the following variables.

\[
\begin{bmatrix}
H_{12} & H_{13} & J_{11} & J_{12} & J_{13} & -G_{11} \\
H_{22} & H_{23} & J_{21} & J_{22} & J_{23} & -G_{12} \\
H_{32} & H_{33} & J_{31} & J_{32} & J_{33} & -G_{13}
\end{bmatrix}
\]

To identify the parameters, a torque input step with amplitude of \(-2Nm\) was used. For the data acquisition, the sampling time used was 0.5ms in order to acquire a sufficient amount of data so that the RLS method presents better results. The obtained results of identification output values are presented in fig. 4.28. After using the RLS method, values for the variables in equation (4.51) to a sampling time of 0.0005s were finally achieved. The system state space matrices \((A, B, C, D)\) are:
4.3 Parameter estimation

![Graph showing output values of flexible link identification](image)

Figure 4.28: Output values of flexible link identification

\[
A_{T_0=0.0005} = \begin{bmatrix}
1 & 0 & 0 & 0.0005 & 0 & 0 \\
0 & 1 & 0 & 0 & 0.0005 & 0 \\
0 & 0 & 1 & 0 & 0 & 0.0005 \\
0 & 0.3815 & -0.0180 & 0.4923 & -0.0013 & 0.0064 \\
0 & -4.8354 & -0.2830 & 0.0352 & 0.4568 & -0.3116 \\
0 & -0.3470 & 0.0284 & 0.0026 & -0.0022 & 0.4596
\end{bmatrix}
\]

\[
B_{T_0=0.0005} = \begin{bmatrix}
0 \\
0 \\
0 \\
0.0154 \\
-0.1117 \\
-0.0082
\end{bmatrix}
\]

\[
C_{T_0=0.0005} = \begin{bmatrix}
0.575 & 0.9997 & 0.0161 & 0 & 0 & 0
\end{bmatrix}
\]

But, since all the control is made at the sample time 0.001s a resampling transformation from sample time 0.0005s to 0.001s was required.
The system representation for $T_0 = 0.001s$ is obtained as

$$A = \begin{bmatrix} 1 & 0 & 0 & 0.001 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0.001 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0.001 \\ 0 & 0.7631 & -0.036 & 0.9846 & -0.0027 & 0.0127 \\ 0 & -9.6709 & -0.5661 & 0.0704 & 0.9135 & -0.6232 \\ 0 & -0.6939 & 0.0568 & 0.0053 & -0.0043 & 0.9191 \end{bmatrix}$$ (4.55)

$$B = \begin{bmatrix} 0 \\ 0 \\ 0 \\ -0.0307 \\ -0.2233 \\ -0.0163 \end{bmatrix}$$ (4.56)

$$C = \begin{bmatrix} 0.575 & 0.9997 & 0.0161 & 0 & 0 & 0 \end{bmatrix}$$ (4.57)

$$D = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$ (4.58)

Comparing the identified system response to an input torque step of $-2.3 Nm$ with the real system response (fig. 4.29), the system main characteristics can be observed, such as non-minimum phase and gain. The only problem is due to the permanent error, which is to be taken in small consideration, since the system will not be in free state for a large amount of time.

In fig. 4.30 is presented the quadratic error values, with a average quadratic error of 0.00035862.
4.3 Parameter estimation

Figure 4.29: Flexible link identification validation

Figure 4.30: Flexible link identification error
Chapter 5

Experimental Results

In this chapter we present experimental results for the motion planing experiments

5.1 End-effector motion planning

Due to the importance of friction on the joints, the controller where the control variable is the angle of the joint has been considered. This control presents the best results when an external torque perturbation is applied. In a first analysis, the transfer function from equation (3.16) for simulation was used to calculate the trajectory control. In real experimentation, the way to calculate the gains $K_p$ and $K_v$ were obtained in a different way than in simulation.

1. Set $K_v$ value equal to zero.

2. Introduce a $K_p$ value high enough to make the system very fast on a response to a step and with a slight oscillation.

3. After the previous two steps, change the $K_v$ value in order to eliminate the oscillation.

The values of $K_p$ and $K_v$ were 600 and 8 respectively. In order to achieve better results, the physical system had to be identified, since the location of poles and zeros aren’t the same as in simulation.
5.1 End-effector motion planning

5.1.1 Identification

The identification of the system was obtained using an input step with amplitude of 0.02 rad and a period of 64 s (figs. 5.1 and 5.2).

![Identification input](image1.png)

**Figure 5.1:** Identification input

![Identification output](image2.png)

**Figure 5.2:** Identification output

To identify the system, an ARMAX model with a time sample of 0.001, and a Tustin transformation from discrete to continuous model was used. This model had the following properties. The system transfer function considered had a 6 order denominator, a 5 order numerator, considering a noise transfer function with a 6 order denominator and a 4 order numerator and with one time sample delay.
The transfer function is:

\[
\frac{y}{\theta} = \frac{0.9422s^3 - 1481s^4 - 4.955 \times 10^6 s^3 + 7.761 \times 10^9 s^2 - 9.703 \times 10^{10} s + 9.118 \times 10^{12}}{(s^6 + 1625s^5 + 7.281 \times 10^5 s^4 + 2.544 \times 10^8 s^3 + 3.918 \times 10^{10} s^2 + 1.168 \times 10^{11} s + 1.603 \times 10^{13})}
\]  

(5.1)

Fig. 5.3 presents zeros and poles of the system.

![Pole-Zero Map](image)

Figure 5.3: Location of zeros and poles in a close loop form

### 5.1.2 Control

Now that the system is identified, with recalculation of the trajectory planning and new transfer function, the system appears to be well behaved since there is almost no vibration on the end-effector (fig. 5.4). It’s important to notice that if the average velocity of the trajectory becomes too high, the effect of the non-linearities increases, causing some vibration on the end-effector (fig. 5.5). If the trajectory time is reduced for the same amplitude, the torque must increase (figs. 5.6 and 5.7).
5.1 End-effector motion planning

![Graph 1: Linear system (Experimental results) t=2.5s yf=0.9032m](image1)

Figure 5.4: End-effector vibration at low average velocity

![Graph 2: Non-linear System (Experimental results) t=2.5s yf=−0.9032m](image2)

Figure 5.5: End-effector vibration at high average velocity
Experimental Results

Figure 5.6: Input torque at low average velocity

Figure 5.7: Input torque at high average velocity
5.1 End-effector motion planning
Chapter 6

Conclusions

In this final chapter, we present the conclusions of the developed work.

6.1 Stable Inversion method

Friction has an important role in the real system. Comparing fig. 3.2 with figs. 5.6 and 5.7 we observe that the undershoot presented on fig. 3.2 no longer exists. This behavior is due to existing friction on joints, which acts as a resistive torque. Therefore there is no need to add a negative torque to stop the robot arm. Another characteristic of this trajectory planning is that it is impossible to eliminate the stationary error. This characteristic is due to the fact that the controller is a PD controller and by definition it does not remove any stationary error.

Regarding the control where the input of the system is the angle of the joint, it is evident that this kind of control is more robust and efficient than the control where the input is the torque. This controller was the chosen one to the experimental results due to the enormous friction in the joints. There is also some limitations about the controller since it has been planned around a linear model. When the average velocity of trajectory becomes too high, the system non-linear effects increase (figs. 5.4 to 5.7).
6.2 Path-Following

Path following is still under intensive study. It is a completely different approach from the standard control, which involves some advanced mathematics and control theory. One of the major problems on the implementation of this controller was related with the defined velocity. Since the objective of this thesis is to control a manipulator robot from point A to point B, and most of path-following solutions aimed to path-following at constant speed, there were some implementation difficulties. However, we were able to obtain a path-following controller at continuos state in simulation mode. These results showed that the path-following controller removes dynamical limitations which reference tracking controllers can’t. Comparing the external perturbation curves from figs. 3.7 to 3.12 with the curves from figs. 4.14 to 4.27 it’s relevant the ability that the path-following controller has to remove major external disturbances.

Comparing the various studied path types, it is shown that the controller error depends on the path and it’s limitations. For the first path, there wasn’t any variable limitation so $\gamma$ can evolve to infinity, but there is the necessity to insert integrators initial conditions, which develops undesired initial vibrations. For the second path, $\gamma$ as been limited between $[0, 1]$, this develops actuation limitations and more difficulty to attenuate external disturbances. On the other hand, there’s no need to insert integrator initial conditions, eliminating the initial end-effector vibration.

Since the reference-tracking controller is an open-loop controller, when the reference achieves the final value, the controller becomes passive, and it doesn’t observe the end-effector deflection, resulting on a permanent vibration. The path-following ability to separate the dynamic follower controller from the states boundedness controller, improves the control actuation. Even when the system is reached to the final value, it still removes the external perturbation. It becomes evident that the path-following controller is a much more developed and robust controller than the reference tracking controller.

6.3 Future work

One of the reasons to describe the Normal-Form Path-Following controller in this thesis, is due to the fact that the error from the end-effector position enters directly into the control action. Therefore, it should be expected better results than currently presented. So, for future work it is highly recommended a profound
study and implementation of Normal-form and exo-system path-following controller.

Since in the long run this study aims at the development of a flexible manipulator surgical robot, it remains to be shown the performance of path-following controllers for robotic medical procedures.
6.3 Future work
Bibliography


Appendix A

Special Coordinate basis

Considering a SISO system characterized by:

\[ \Sigma : \dot{x} = Ax + Bu, \quad y = Cx \]  \hspace{1cm} (A.1)

Where \( x, u \) and \( y \) are the state, the input and the output. Assuming that the transfer function of \( \Sigma \) is not identically to zero, the following special coordinate basis (SCB) decomposition for \( \Sigma \) is obtained.

**Theorem 2.** consider the SISO system of A.1. The exist nonsingular state, input and output transformations \( \Gamma_s \in \mathbb{R}^{n \times n}, T_i \in \mathbb{R} \) and \( T_o \in \mathbb{R} \) which decompose the state space of \( \Sigma \) into two subspaces \( x_a \) and \( x_d \) these two subspaces correspond to the finite zero and infinite zero structures of \( \Gamma \), respectively. The new state space, input and output of the decomposed system are described by the following set of equations:

\[ x = \Gamma_s \bar{x}, \quad y = \Gamma_o \bar{y}, \quad u = \Gamma_i \bar{u} \]  \hspace{1cm} (A.2)

\[ \bar{x} = \begin{pmatrix} x_a \\ x_d \end{pmatrix}, \quad x_a \in \mathbb{R}^{n_a}, \quad x_d \in \mathbb{R}^{n_d}, \quad x_d = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_{n_d} \end{pmatrix} \]  \hspace{1cm} (A.3)

and

\[ x_a = A_{aa} x_a + L_{ad} \bar{y}, \]  \hspace{1cm} (A.4)

\[ \dot{x}_1 = x_2, \quad \bar{y} = x_1. \]  \hspace{1cm} (A.5)
\[ \dot{x}_2 = x_3, \quad \dot{x}_3 = x_4, \quad \vdots \] (A.6)

\[ \dot{x}_{n_d-1} = x_{n_d}, \] (A.7)

\[ \dot{x}_{n_d} = E_{d_1}x_{d_1} + E_1x_1 + E_2x_2 + \cdots + E_{n_d}x_{n_d} + \tilde{u}, \] (A.8)

Where \( \lambda(A_{aa}) \) contains all the system invariant zeros and \( n_d \) is the relative degree of \( \Gamma \).

Figure A.1: Interpretation of structural decomposition of a SISO system

[19] represented a step-by-step algorithm to construct the required \( \Gamma_x, \Gamma_i \) and \( \Gamma_o \) that realize the structural decomposition or the special coordinate basis of \( \Sigma \).

1. Determination of the relative degree The relative degree of \( \Sigma \) can be obtained by differentiating the output \( y \). Being \( n_d \) such that:

\[ CB = CAB = \cdots = CA^{n_d-2}B = 0 \] (A.9)

and

\[ \beta = CA^{n_d-1}B \neq 0 \] (A.10)
2. Construction of a preliminary state transformation Let $Z_0$ be an $n_a \times n$ constant matrix, where $n_a$ is equal to $n_a = n - n_d$, such that

\[ Z = \begin{bmatrix} \ Z_0 \\ \ Z_d \end{bmatrix} = \begin{bmatrix} \ Z_0 \\ C \\ C A \\ \vdots \\ C A^{n_d-1} \end{bmatrix} \] (A.11)

is nonsingular. $Z_0$ can be chosen where whose rows form a basis of the null space of $Z_d$. Being,

\[ \bar{x} = \begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \\ x_{n_d} \end{pmatrix} = Zx = \begin{pmatrix} Z_0 \\ C \\ C A \\ \vdots \\ C A^{n_d-1} \end{pmatrix} x \] (A.12)

then

\[
\begin{align*}
\dot{x}_1 &= CA\dot{x} = CAx + CBu = CAx = x_2, \quad y = x_1, \\
\dot{x}_2 &= CA\dot{x} = CA^2 x + CABu = CA^2 x = x_3, \\
& \vdots \\
\dot{x}_{n_d-1} &= x_{n_d}, \\
\dot{x}_{n_d} &= E_{da}x_0 + \sum_{i=1}^{n_d} \xi_i x_i + \beta_0 u
\end{align*}
\] (A.13)

For appropriate $E_{da}$, and $\beta_i, i = 1, 2, \ldots, n_d$, and

\[ \dot{x}_0 = A_{00}x_0 + \sum_{i=1}^{n_d} \alpha_{0,i}x_i + \beta_0 u \] (A.14)

for some appropriate vectors $A_{00}, \alpha_i, i = 1, 2, \ldots, n_d$, and $\beta_0$

3. The elimination of $u$ in the state equation of $x_0$ It follows from A.13 that:

\[ u = \frac{1}{\beta} \left[ \dot{x}_{n_d} - E_{da}x_0 - \sum_{i=1}^{n_d} \xi_i x_i \right] \] (A.15)

which together with 2 imply that

\[
\begin{align*}
\dot{x}_0 &= A_{00}x_0 + \sum_{i=1}^{n_d} \alpha_{0,i}x_i + \frac{\beta_0}{\beta} \left[ \dot{x}_{n_d} - E_{da}x_0 - \sum_{i=1}^{n_d} \xi_i x_i \right] \\
\dot{x}_0 &= A_{00}x_0 + \sum_{i=1}^{n_d} \alpha_{0,i}x_i + \frac{\beta_0}{\beta} \left[ \dot{x}_{n_d} - E_{da}x_0 - \sum_{i=1}^{n_d} \xi_i x_i \right] \\
\dot{x}_0 &= A_{aa}x_0 + \sum_{i=1}^{n_d} \alpha_{0,i}x_i + \beta\dot{x}_{n_d}
\end{align*}
\] (A.16)
4. The elimination of \( x_{n_d} \) in the state equation of \( x_0 \) Defining a new state variable,

\[
\tilde{x}_0 = x_0 - \beta x_{n_d}
\]  (A.18)

Obtaining

\[
\dot{\tilde{x}}_0 = \dot{x}_0 - \beta \dot{x}_{n_d} = A_{aa} x_0 + \sum_{i=1}^{n_d} \bar{\alpha}_0,i x_i + \beta \dot{x}_{n_d} - \beta \ddot{x}_{n_d}
\]

\[
= A_{aa} (\tilde{x}_0 + \beta x_{n_d}) + \sum_{i=1}^{n_d} \bar{\alpha}_0,i x_i
\]

\[
= A_{aa} \tilde{x}_0 + \sum_{i=1}^{n_d} \tilde{\alpha}_0,x_i
\]

for some appropriate constant vectors \( \tilde{\alpha}_{0,i}, i = 1, 2, \ldots, n_d \), also A.13 can be re-written as:

\[
\dot{x}_{n_d} = E_{da} \tilde{x}_0 + \sum_{i=1}^{n_d} \xi_0,i x_i + \beta u
\]  (A.19)

5. The elimination of \( x_2, \ldots, x_{n_d} \) from the state equation of \( \tilde{x}_0 \) If \( n_d = 1 \), there’s no further transformation needed, and can proceed to the following step. Otherwise, let \( s = 0 \), \( \tilde{x}_{0,0} = \tilde{x}_0 \), \( \tilde{\alpha}_{0,0,i} = \tilde{\alpha}_{0,i} \) and \( \tilde{\xi}_{0,i} = \tilde{\xi}_i, i = 1, 2, \ldots, n_d \). It is necessary to carry on the following iterative sub-steps

(a) Eliminate \( x_{n_d} \) It follows from equation A.19:

\[
\dot{\tilde{x}}_{0,s} = A_{aa} \tilde{x}_{0,s} + \sum_{i=1}^{n_d-s} \tilde{\alpha}_{0,i} x_i
\]  (A.20)

\[
\dot{x}_{n_d} = E_{da} \tilde{x}_{0,s} + \sum_{i=1}^{n_d-s} \tilde{\xi}_{s,i} x_i + \beta u
\]

eliminating \( x_{n_d-s} \) from the above expression by defining

\[
\tilde{x}_{0,s+1} = \tilde{x}_{0,s} - (\tilde{\alpha}_{s,0,n_d-s}) x_{n_d-s-1}
\]  (A.21)

which together with \( \dot{x}_{n_d-s-1} = x_{n_d-s} \) imply that

\[
\dot{\tilde{x}}_{0,s+1} = A_{aa} \tilde{x}_{0,s+1} + \sum_{i=1}^{n_d-s-1} \tilde{\alpha}_{s+1,0,i} x_i
\]  (A.22)

For some constant vectors \( \tilde{\alpha}_{s+1,0,i}, i = 1, 2, \ldots, n_d-s-1 \). After having eliminated \( x_{n_d-s} \) in the above expression, follows

\[
\dot{x}_{n_d} = E_{da} \tilde{x}_{0,s+1} + \sum_{i=1}^{n_d} \tilde{\xi}_{s+1,i} x_i + \beta u
\]  (A.23)

for some appropriate constant vectors \( \tilde{\xi}_{s+1,i}, i = 1, 2, \ldots, n_d-s-1 \).
(b) If \( s = n_d - 2 \), finish iteration, otherwise set \( s = s + 1 \) and go back to the previous step.

6. Finishing touch Finally, is obtained:

\[
x_a = \tilde{x}_{0,s+1}, \quad y = \Gamma_o \tilde{y}, \quad u = \Gamma_i \tilde{u} = \frac{1}{\beta} \tilde{u}
\]

and

\[
L_{ad} = \tilde{\alpha}_{s+1,0,1}, \quad E_i = \tilde{\xi}_{s+1,i}, \quad i = 1, 2, \ldots, n_d
\]

where the states are equal to:

\[
\begin{align*}
\dot{x}_a &= A_{aa} x_a + L_{ad} \tilde{y}, \\
\dot{x}_1 &= x_2, \quad \tilde{y} = x_1, \\
\dot{x}_2 &= x_3, \\
&\vdots \\
\dot{x}_{n_d} &= E_{da} x_a + E_{1} x_1 + \cdots + E_{n_d} x_{n_d} + \tilde{u}
\end{align*}
\]
Appendix B

Recursive Least Squares equations

The Least Squares method consist in the determination of the parameters vector that minimize the value of the square error given by [21]

\[ ||v||^2 = ||F - HX||^2 \]  \hspace{1cm} (B.1)

where \( v \) is the vector of the modeling and measurement errors, \( F \) is the vector of inputs and \( H \) is the regression matrix. Expanding this expression:

\[
||v||^2 = (F - HX)^T (F - HX) \\
= (F^T - X^T H^T)(F - HX) \\
= F^T F - F^T HX - X^T H^T F + X^T H^T HX \\
= F^T F - (HX)^T F - X^T H^T F + X^T H^T HX \\
= F^T F - X^T H^T F - X^T H^T F + X^T H^T HX \\
= F^T F - 2X^T H^T F + X^T H^T HX 
\]  \hspace{1cm} (B.2)

By determining the derivative relative to the parameters vector \( X \) and equaling to zero results in

\[
\frac{d||v||^2}{dX} = H^T H X - H^T F = 0 
\]  \hspace{1cm} (B.3)

So the vector \( X \) that minimize the quadratic error is given by

\[
X = (H^T H)^{-1} H^T F 
\]  \hspace{1cm} (B.4)

For the Recursive Least Squares method the estimated parameters vector \( \hat{X} \) for an instant \( t \) is obtained by using the last estimation having a parameters vector that converge to the values of the real parameters.
For time $t - 1$ the parameter vector is

$$\hat{X}(t - 1) = [H^T(t - 1)H(t - 1)]^{-1}H^T(t - 1)F(t - 1) \tag{B.5}$$

At time $t$, further measurements from the process are acquired enable to form the matrix

$$H(t) = \begin{bmatrix} H(t - 1) \\ Y^T(t) \end{bmatrix}, \tag{B.6}$$

and the vector

$$F(t) = \begin{bmatrix} F(t - 1) \\ \tau(t) \end{bmatrix}. \tag{B.7}$$

The estimation at time $t$ is then given by

$$\hat{X}(t) = [H^T(t)H(t)]^{-1}H^T(t)F(t) \tag{B.8}$$

From expression (B.6) we get

$$H^T(t)H(t) = \begin{bmatrix} H^T(t - 1) & Y(t) \end{bmatrix} \begin{bmatrix} H(t - 1) \\ Y^T(t) \end{bmatrix} = H^T(t - 1)H(t - 1) + Y(t)Y^T(t) \tag{B.9}$$

From equations (B.6) and (B.7) we get the term

$$H^T(t)F(t) = \begin{bmatrix} H^T(t - 1) & Y(t) \end{bmatrix} \begin{bmatrix} F(t - 1) \\ \tau(t) \end{bmatrix} = H^T(t - 1)F(t - 1) + Y(t)\tau(t) \tag{B.10}$$

To introduce some shorthand the matrix $P(t - 1)$ and the vector $B(t - 1)$ are defined as

$$\left\{ \begin{align*}
P(t - 1) &= [H^T(t - 1)H(t - 1)]^{-1} \\
B(t - 1) &= H^T(t - 1)F(t - 1)
\end{align*} \right. \tag{B.11}$$

resulting

$$\left\{ \begin{align*}
\hat{X}(t) &= P(t)B(t) \\
\hat{X}(t - 1) &= P(t - 1)B(t - 1)
\end{align*} \right. \tag{B.12}$$

The terms for the actual time, $P(t)$ and $B(t)$ are given by

$$P^{-1}(t) = P^{-1}(t - 1) + Y(t)Y^T(t) \tag{B.13}$$

and

$$B(t) = B(t - 1) + Y(t)\tau(t) \tag{B.14}$$
Equation (B.14) gives a direct update from $B(t-1)$ to $B(t)$, but the update of $P(t-1)$ emerge the problem of inverting the matrix at each sample time that reduces the computational efficiency of the algorithm. The standard way to overcome this problem is by applying the Matrix Inversion Lemma:

$$(A + BCD)^{-1} = A^{-1} - A^{-1}B(C^{-1} + DA^{-1}B)^{-1}DA^{-1} \quad (B.15)$$

Assigning

$$A = P^{-1}(t-1), \quad B = Y(t), \quad C = I, \quad D = Y^T(t), \quad (B.16)$$

gives

$$P(t) = P(t-1) - P(t-1)Y(t)[I + Y^T(t)P(t-1)Y(t)]^{-1}Y^T(t)P(t-1) \quad (B.17)$$

Now defining the error variable as

$$\epsilon(t) = \tau(t) - Y^T(t)\hat{X}(t-1) \quad (B.18)$$

and substituting $\tau(t)$ in equation (B.14) we obtain

$$B(t) = B(t-1) + Y(t)Y^T(t)\hat{X}(t-1) + Y(t)e(t) \quad (B.19)$$

Substituting $B(t-1)$ and $B(t)$ by using equations (B.12)

$$P^{-1}(t)\hat{X}(t) = P^{-1}(t-1)\hat{X}(t-1) + Y(t)Y^T(t)\hat{X}(t-1) + Y(t)e(t) \quad (B.20)$$

substituting $P^{-1}(t-1)$ through equation (B.13) results in

$$\hat{X}(t) = \hat{X}(t-1) + P(t)Y(t)e(t) \quad (B.21)$$

Expanding $e(t)$ and making $K(t) = P(t)Y(t)$ we get the equations for the Recursive Least Squares algorithm:

$$\begin{cases}
\hat{X}(t) = \hat{X}(t-1) + K(t)[\tau(t) - Y^T(t)\hat{X}(t-1)] \\
K(t) = P(t-1)Y(t)[I + Y^T(t)P(t-1)Y(t)]^{-1} \\
P(t) = P(t-1) - K(t)Y^T(t)P(t-1)
\end{cases} \quad (B.22)$$