



Universidade Federal
do Rio de Janeiro

Escola Politécnica

INTERFACE E CONTROLE DE UM MANIPULADOR ROBÓTICO 4-DOF COM ROS E QT

Luís Gustavo Oliveira Silva

Projeto de Graduação apresentado ao Curso de Engenharia de Controle e Automação, da Escola Politécnica, Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Engenheiro de Controle e Automação.

Orientadores: Fernando Cesar Lizarralde
Nome do Segundo Orientador
Sobrenome

Rio de Janeiro
Janeiro de 2017

INTERFACE E CONTROLE DE UM MANIPULADOR ROBÓTICO 4-DOF
COM ROS E QT

Luís Gustavo Oliveira Silva

PROJETO DE GRADUAÇÃO SUBMETIDO AO CORPO DOCENTE DO
CURSO DE ENGENHARIA DE CONTROLE E AUTOMAÇÃO DA ESCOLA
POLITÉCNICA DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO
PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU
DE ENGENHEIRO DE CONTROLE E AUTOMAÇÃO.

Examinado por:

Prof. Nome do Primeiro Examinador Sobrenome, D.Sc.

Prof. Nome do Segundo Examinador Sobrenome, Ph.D.

Prof. Nome do Terceiro Examinador Sobrenome, D.Sc.

Prof. Nome do Quarto Examinador Sobrenome, Ph.D.

Prof. Nome do Quinto Examinador Sobrenome, Ph.D.

RIO DE JANEIRO, RJ – BRASIL
JANEIRO DE 2017

Oliveira Silva, Luís Gustavo

Interface e Controle de um Manipulador Robótico 4-DOF com ROS e Qt/Luís Gustavo Oliveira Silva. – Rio de Janeiro: UFRJ/Escola Politécnica, 2017.

XI, 17 p.: il.; 29, 7cm.

Orientadores: Fernando Cesar Lizarralde

Nome do Segundo Orientador

Sobrenome

Projeto de graduação – UFRJ/Escola Politécnica/Curso de Engenharia de Controle e Automação, 2017.

Referências Bibliográficas: p. 16 – 16.

1. Primeira palavra-chave. 2. Segunda palavra-chave. 3. Terceira palavra-chave. I. , Fernando Cesar Lizarralde *et al.* II. Universidade Federal do Rio de Janeiro, Escola Politécnica, Curso de Engenharia de Controle e Automação. III. Título.

*A alguém cujo valor é digno
desta dedicatória.*

Agradecimentos

Gostaria de agradecer a todos.

Resumo do Projeto de Graduação apresentado à POLI/UFRJ como parte dos requisitos necessários para a obtenção do grau de Engenheiro de Controle e Automação.

INTERFACE E CONTROLE DE UM MANIPULADOR ROBÓTICO 4-DOF COM ROS E QT

Luís Gustavo Oliveira Silva

Janeiro/2017

Orientadores: Fernando Cesar Lizarralde

Nome do Segundo Orientador Sobrenome

Curso: Engenharia de Controle e Automação

Apresenta-se, nesta tese, ...

Abstract of Undergraduate Project presented to POLI/UFRJ as a partial fulfillment of the requirements for the degree of Control and Automation Engineer.

THESIS TITLE

Luís Gustavo Oliveira Silva

January/2017

Advisors: Fernando Cesar Lizarralde

Nome do Segundo Orienta dor Sobrenome

Course: Control and Automation Engineering

In this work, we present ...

Sumário

Lista de Figuras	x
Lista de Tabelas	xi
1 Introdução	1
1.1 Doris...	1
1.2 Tetis.. . . .	1
2 Revisão Bibliográfica	2
2.1 Cinemática Direta	2
2.1.1 Convenção Denavit-Hartenberg	3
2.2 Cinemática Diferencial	4
2.2.1 Jacobiano Geométrico	4
2.2.2 Jacobiano Analítico	4
2.3 Controle Cinemático	4
2.4 Servo Visão	6
2.4.1 Servo Visão Baseada em Posição	6
3 Tetis	7
3.1 Parâmetros Denavit–Hartenberg	7
3.2 Cinemática Direta	7
4 Implementação	9
4.1 Ferramentas	9
4.2 ROS	9
4.3 Conceitos	9
4.4 Modos de Controle	10
4.4.1 Velocidade no Espaço Operacional	10
4.4.2 Posição no Espaço das Juntas	10
4.4.3 Posição no Espaço Operacional	10
4.4.4 Rastreamento de Trajetória	10
4.4.5 Servo Visão	10

4.4.6	Força	10
4.4.7	Master-Slave (Omni)	11
5	Resultados e Discussões	12
5.1	Respostas das Juntas	12
5.2	Rastreamento de Trajetória	13
6	Conclusões	15
	Referências Bibliográficas	16
A	Apêndice	17
A.1	ViSP - Visual Servoing Platform	17

Lista de Figuras

2.1	Diagrama de blocos incluindo a malha de controle interna das juntas.	5
2.2	Diagrama de blocos: Controle Cinemático Proporcional com Feed-Forward	5
3.1	Manipulador	7
5.1	Resposta das juntas	12
5.2	Resposta das juntas	13
5.3	Trajectoria	13
5.4	Rastreamento de trajetória	14
5.5	Destaque para o erro e_1 e e_3	14

Lista de Tabelas

3.1	Parametros Denavit–Hartenberg	7
-----	---	---

Capítulo 1

Introdução

1.1 Doris...

1.2 Tetis..

Capítulo 2

Revisão Bibliográfica

Neste capítulo serão abordados os conceitos necessários para modelagem e controle de manipuladores robóticos.

2.1 Cinemática Direta

Um manipulador robótico é composto de uma série de corpos rígidos denominados *elos* conectados através de *juntas*. Juntas podem ser:

- Revolução
- Prismática

Essa estrutura é chamada de cadeia cinemática. Um extremo da cadeia é fixado a base e o outro ao efetuador. Nesse texto serão abordadas apenas cadeias cinemáticas abertas, ou seja, aquelas em que existe apenas uma sequência de elos conectando os dois extremos da cadeia. Cada junta acrescenta um grau de liberdade (DOF), ao qual está associado a uma variável de junta. No caso de uma junta de revolução um ângulo e no caso de uma junta prismática um deslocamento. O objetivo da cinemática direta é calcular a posição e orientação do efetuador em função das variáveis das juntas.

Uma cadeia cinemática aberta é constituída por $n + 1$ elos numerados de 0 a n , onde o Elo 0 é fixado a base por convenção. O método utilizado consiste em definir um sistema de coordenadas associado a cada elo e calcular a transformação homogênea entre elos consecutivos. Em seguida a transformação do n -ésimo sistema de coordenadas pode ser obtida de forma recursiva como

$$\mathbf{T}_{0n}(\mathbf{q}) = \mathbf{T}_{01}(q_1)\mathbf{T}_{12}(q_2) \dots \mathbf{T}_{n-1,n}(q_n) \quad (2.1)$$

onde $\mathbf{T}_{i-1,i}(q_i)$ denota a transformação homogênea do sistema de coordenadas sólídário ao elo $i - 1$ àquele sólídário ao elo i .

Logo a transformação homogênea que da efetuador final com respeito a base é dada por

$$\mathbf{T}_{be}(\mathbf{q}) = \mathbf{T}_{b0}\mathbf{T}_{0n}(\mathbf{q})\mathbf{T}_{ne} \quad (2.2)$$

2.1.1 Convenção Denavit-Hartenberg

Para calcular a cinemática direta para uma manipulador de cadeia cinemática aberta de acordo com a equação (2.1) um método sistemático foi definido para obter a relação entre a posição e orientação de dois elos consecutivos. A convenção Denavit-Hartenberg especifica um conjunto de regras sobre como definir os sistemas de coordenadas de cada elo.

Seja o Eixo i o eixo da junta que conecta o elo $i - 1$, ao elo i , então:

- Escolher o eixo z_i ao longo do eixo da junta $i + 1$.
- Colocar a origem O_i na interseção do eixo z_i com a normal comum entre os eixos z_{i-1} e z_i
- Escolher x_i ao longo da normal comum aos eixos z_{i-1} e z_i , com direção da junta i para a junta $i + 1$.
- O eixo $y_i = z_i \times x_i$ é escolhido de forma a completar o sistema de coordenadas.

Essa convenção resulta em uma definição não única do sistema de coordenadas nos seguintes casos:

- Para o sistema de coordenadas 0, somente a direção do eixo z_0 é especificada, portanto a escolha de O_0 e x_0 é arbitrária.
- Para o sistema de coordenadas n , como não existe junta $n + 1$, z_n não está definido, mas x_n deve ser normal ao eixo z_{n-1} . Tipicamente escolhe-se z_n alinhado com z_{n-1} .
- Quando dois eixos consecutivos são paralelos, a normal comum entre eles não é definida de forma única. Tipicamente escolhe-se O_i na junta $i + 1$
- Quando dois eixos consecutivos se interceptam, direção de x_i é normal e o sentido é arbitrário. Escolhe-se O_i na inteseccção.
- Quando a junta i é prismática a direção de z_{i-1} é arbitrária.

2.2 Cinemática Diferencial

2.2.1 Jacobiano Geométrico

2.2.2 Jacobiano Analítico

Quando a posição e orientação do efetuador são dadas em função de um número mínimos de parametros no espaço operacional é possível computar o Jacobiano pela diferenciação das equações da cinemática direta em função das variáveis das juntas. Para isso utiliza-se a técnica analítica.

Seja \mathbf{p}_e a posição do sistema de coordenadas do efetuador representada no sistema de coordenadas da base. O vetor $\dot{\mathbf{p}}_e$ é portanto a velocidade de translação, ou linear.

$$\dot{\mathbf{p}}_e = \frac{\partial \mathbf{p}_e}{\partial \mathbf{q}} \dot{\mathbf{q}} = \mathbf{J}_P(\mathbf{q}) \dot{\mathbf{q}} \quad (2.3)$$

Para a velocidade angular, pode ser considerada uma representação mínima da orientação em função de três variáveis ϕ_e . A derivada no tempo $\dot{\phi}_e$ não é igual a velocidade angular, no entanto, conhecida a função $\phi_e(\mathbf{q})$:

$$\dot{\phi}_e = \frac{\partial \phi_e}{\partial \mathbf{q}} \dot{\mathbf{q}} = \mathbf{J}_\phi(\mathbf{q}) \dot{\mathbf{q}} \quad (2.4)$$

Sob essas premissas a cinemática diferencial pode ser obtida como:

$$\dot{\mathbf{x}}_e = \begin{bmatrix} \dot{\mathbf{p}}_e \\ \dot{\phi}_e \end{bmatrix} = \begin{bmatrix} \mathbf{J}_P(\mathbf{q}) \\ \mathbf{J}_\phi(\mathbf{q}) \end{bmatrix} \dot{\mathbf{q}} = \mathbf{J}_A(\mathbf{q}) \dot{\mathbf{q}} \quad (2.5)$$

2.3 Controle Cinemático

A estratégia de controle cinemático pode ser aplicada quando considera-se que a dinâmica do manipulador pode ser desprezada. Essa hipótese se sustenta quando as seguintes premissas são válidas:

- Elevados fatores de redução nas juntas
- Baixas velocidades na realização das tarefas
- Existe uma malha de controle de velocidade de alto desempenho em cada junta

A maioria dos manipuladores possui uma malha de controle de velocidade em nível de juntas como na figura 2.1. Logo, para um controle de alto ganho temos que:

$$u \approx \dot{\theta}$$

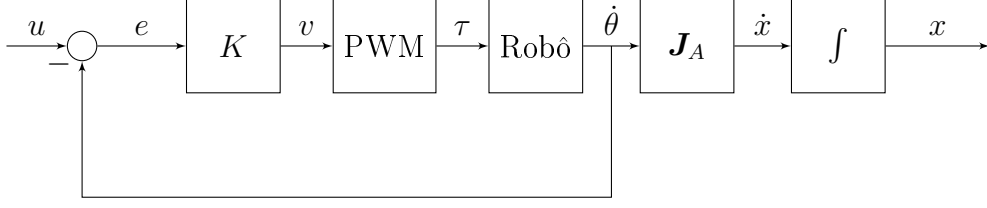


Figura 2.1: Diagrama de blocos incluindo a malha de controle interna das juntas.

Portanto é possível implementar o controle cinemático segundo o diagrama 2.2

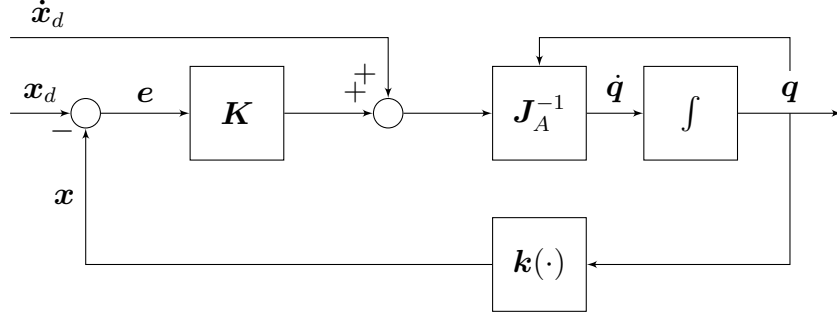


Figura 2.2: Diagrama de blocos: Controle Cinemático Proporcional com Feed-Forward

Se x é uma representação da posição e orientação e x_d o valor desejado nessa representação, seja o erro no espaço operacional:

$$e = x_d - x \quad (2.6)$$

Derivando em relação ao tempo

$$\dot{e} = \dot{x}_d - \dot{x} \quad (2.7)$$

podemos escrever a partir da equação 2.5:

$$\dot{e} = \dot{x}_d - J_a(q)\dot{q} \quad (2.8)$$

Sendo $x_d(t)$ uma trajetória desejada, deseja-se que x atinja $x_d(t)$ em $t \rightarrow \infty$. A entrada de controle para o sistema é um valor de $u = \dot{q}$, logo, assumindo que $J_A(q)$ é quadrada e não singular, a escolha da lei de controle

$$u = J_A^{-1}(q)\bar{u} \quad (2.9)$$

leva ao sistema linear:

$$\dot{e} = \dot{x}_d - \bar{u} \quad (2.10)$$

Se for escolhido $\bar{\mathbf{u}}$:

$$\bar{\mathbf{u}} = \dot{\mathbf{x}}_d + \mathbf{K}(\mathbf{x}_d - \mathbf{x}) \quad (2.11)$$

obtem-se a seguinte dinâmica para o erro

$$\dot{\mathbf{e}} + \mathbf{K}\mathbf{e} = 0 \quad (2.12)$$

2.4 Servo Visão

A tarefa proposta na Servo Visão é controlar a posição e orientação do efetuador do manipulador, em relação a um alvo, usando características visuais extraídas de uma imagem. A câmera pode ser carregada pelo manipulador (montada no efetuador) ou colocada em um ponto fixo, observando tanto o efetuador como o alvo.

2.4.1 Servo Visão Baseada em Posição

Em um sistema de servo visão baseado em posição a posição e orientação do alvo com respeito a câmera \mathbf{T}_{CT} é estimada. O problema de estimação da posição e orientação é discutido no apêndice ?. Especifica-se uma posição desejada relativa ao sistema de coordenadas do alvo \mathbf{T}_{C^*T} e deseja-se determinar o movimento necessário para mover a câmera para a posição desejada, que chamamos de \mathbf{T}_δ .

$$\mathbf{T}_{CT} = \mathbf{T}_\Delta \mathbf{T}_{C^*T} \quad (2.13)$$

$$\mathbf{T}_\Delta = \mathbf{T}_{CT} \mathbf{T}_{C^*T}^{-1} \quad (2.14)$$

Capítulo 3

Tetis

3.1 Parâmetros Denavit–Hartenberg

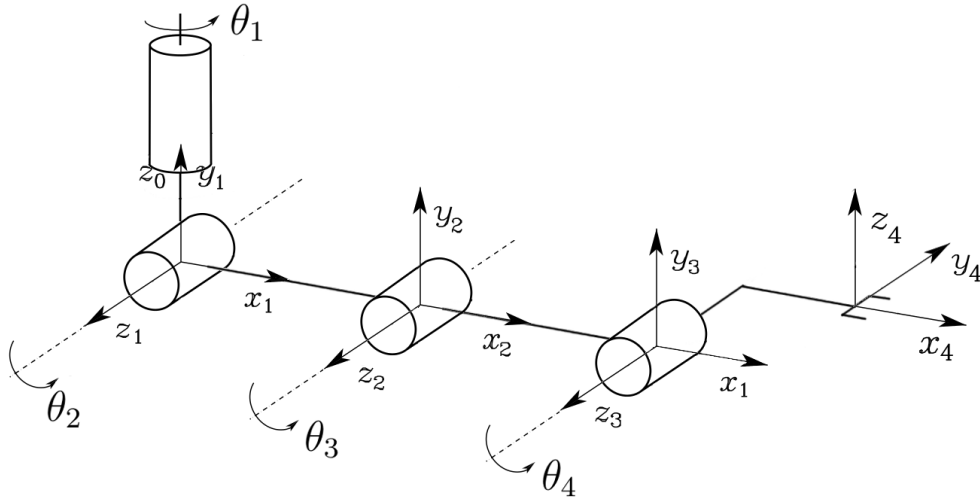


Figura 3.1: Manipulador

Tabela 3.1: Parametros Denavit–Hartenberg

Elo	a_i	α_i	d_i	θ_i
1	0	$\pi/2$	0	θ_1
2	E_3	0	0	θ_2
3	E_4	0	0	θ_3
4	E_5	$-\pi/2$	$-M_5$	θ_4

3.2 Cinemática Direta

$$\mathbf{R}_{i-1,i} = \mathbf{R}_z(\theta_i)\mathbf{R}_x(\alpha_i) \quad (3.1)$$

$$\vec{\mathbf{p}}_{i-1,i} = d_i\vec{\mathbf{z}}_{i-1} + a_i\vec{\mathbf{x}}_i \quad (3.2)$$

$$(\vec{p}_{i-1,i})_{i-1} = d_i(\vec{z}_{i-1})_{i-1} + a_i(\vec{x}_i)_{i-1} \quad (3.3)$$

$$(\vec{p}_{i-1,i})_{i-1} = d_i(\vec{z}_{i-1})_{i-1} + a_i \mathbf{R}_{i-1,i}(\vec{x}_i)_i \quad (3.4)$$

$$\mathbf{T}_{01} = \begin{bmatrix} c_1 & 0 & s_1 & 0 \\ s_1 & 0 & -c_1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{T}_{12} = \begin{bmatrix} c_2 & -s_2 & 0 & E_3 c_2 \\ s_2 & c_2 & 0 & E_3 s_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{T}_{23} = \begin{bmatrix} c_3 & -s_3 & 0 & E_4 c_3 \\ s_3 & c_3 & 0 & E_4 s_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{T}_{34} = \begin{bmatrix} c_4 & 0 & -s_4 & E_5 c_4 \\ s_4 & 0 & c_4 & E_5 s_4 \\ 0 & -1 & 0 & -M_5 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{T}_{04} = \mathbf{T}_{01} \mathbf{T}_{12} \mathbf{T}_{23} \mathbf{T}_{34} = \begin{bmatrix} c_1 c_{234} & -s_1 & -c_1 s_{234} & -M_5 s_1 + E_4 c_{23} c_1 + E_3 c_1 c_2 + E_5 c_{234} c_1 \\ s_1 c_{234} & -c_1 & -s_1 s_{234} & M_5 c_1 + E_4 c_{23} s_1 + E_3 c_2 s_1 + E_5 c_{234} s_1 \\ s_{234} & 0 & c_{234} & E_4 s_{23} + E_3 s_2 + E_5 s_{234} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Capítulo 4

Implementação

A implementação dos algoritmos de controle detalhados nos capítulos anteriores foi feita como extensão ao software RobotGUI desenvolvido pela equipe do LEAD-GSCAR, idealizado por Alex F. Neves Msc.

4.1 Ferramentas

Os seguintes softwares e *frameworks* foram utilizados:

- Linux (Ubuntu) como Sistema Operacional
- C++ como linguagem de programação
- ROS como *framework* principal utilizado o RobotGUI, fornecendo comunicação entre nós através de mensagens e serviços. Será descrito mais detalhadamente na próxima seção.
- Qt como *framework* para elaboração da interface gráfica.

4.2 ROS

4.3 Conceitos

Primeiramente define-se como Computador Base aquele que será utilizado pelo operador para controlar e visualizar dados do robô. Define-se Computador Embarcado, ou do Robô aquele que está no robô conectado a todos os equipamentos, sensores e atuadores. O software executa módulos diferentes no robô e na base.

A arquitetura do RobotGUI baseia-se nos seguintes conceitos principais:

- **Components:** Lidam com a comunicação e processam dados no Computador Base. São essencialmente *Nodelets* de ROS, podendo utilizar funcionalidades

como comunicação através de mensagens e serviços, utilizar parametros e bibliotecas de ROS. Permitem maior modularidade pois componentes podem ser inicializados independentemente do *RobotGUI*.

- **Tools:** São elementos gráficos da interface para o usuário. São utilizadas para interagir com o robô e visualizar informação. São criadas como *plugins* para o ROS *pluginlib* e independentes de qualquer biblioteca do ROS.
- **Interação entre Components e Tools:** Tools e Components podem se conectar, quando isso ocorre eles interagem entre si a nível de "ponteiro para objeto". Essa conexão permite que o desenvolvimento da interface através do *Qt* seja quase que independente do desenvolvimento do código que lida com hardware, lógica e comunicação.
- **RobotGUI**

4.4 Modos de Controle

4.4.1 Velocidade no Espaço Operacional

Base

Efetuator

4.4.2 Posição no Espaço das Juntas

4.4.3 Posição no Espaço Operacional

4.4.4 Rastreamento de Trajetória

4.4.5 Servo Visão

4.4.6 Força

Float

Approach

Considera-se o problema de controle de força na direção de approach para o manipulador robótico 4-DOF em questão. A realimentação de força é feita com o uso do sensor descrito em ???. O objetivo de controle é rastrear uma entrada na forma de um degrau de força, a ser aplicada em uma placa de poliestireno montada em um suporte fixado na vertical como mostra a figura ??.

É possível modelar o ambiente (força de contato), ou seja, a placa de poliestireno como uma mola linear, através da *Lei de Hook*: [?]]

$$f = -k_s(x - x_s) \quad (4.1)$$

onde x é o ponto de contato com a superfície e x_s

Híbrido

4.4.7 Master-Slave (Omni)

Capítulo 5

Resultados e Discussões

5.1 Respostas das Juntas

Para verificar se de fato são válidas as premissas assumidas na seção 2.3 para aplicação de uma estratégia de controle cinemático foi levantada para cada uma das juntas a resposta a uma onda quadrada tal que:

$$u = A \operatorname{sgn}(\sin(2\pi t/f))$$

onde o período $T = 1/f = 1s$ e a amplitude $A = 0.5rad/s$

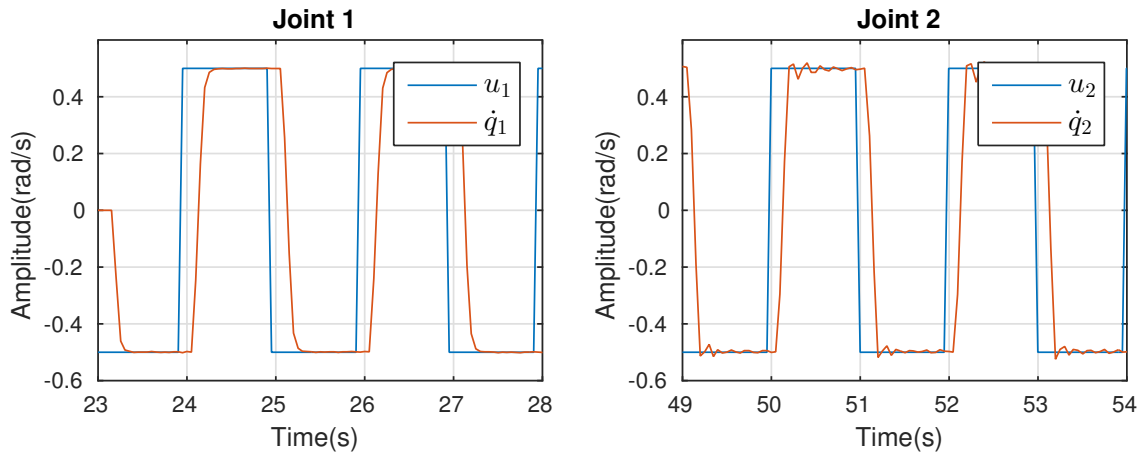


Figura 5.1: Resposta das juntas

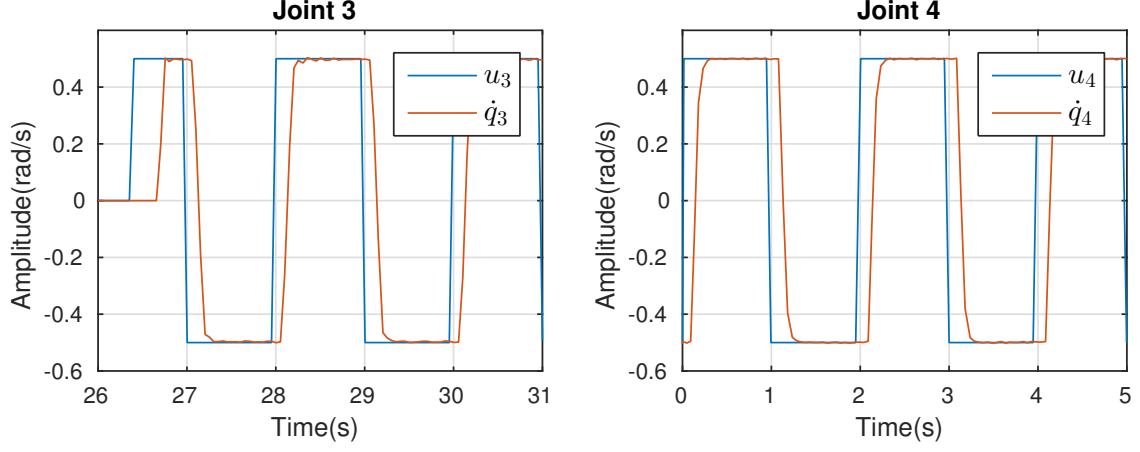


Figura 5.2: Resposta das juntas

5.2 Rastreamento de Trajetória

A trajetória a ser rastreada é dada pelas equações:

$$\mathbf{x}_d = \begin{bmatrix} 75 \sin(\omega_n t) + \sin(4\omega_n t) + 500 \\ 57 \\ 75 \cos(\omega_n t) + \cos(4\omega_n t) - 67 \\ \omega_n \sin(\omega_n t) \end{bmatrix} \quad \dot{\mathbf{x}}_d = \begin{bmatrix} 75\omega_n \cos(t\omega_n) + 300\omega_n \cos(4t\omega_n) \\ 0 \\ -75\omega_n \sin(t\omega_n) - 300\omega_n \sin(4t\omega_n) \\ \omega_n^2 \cos(t\omega_n) \end{bmatrix} \quad (5.1)$$

Onde $\omega_n = \pi/10$

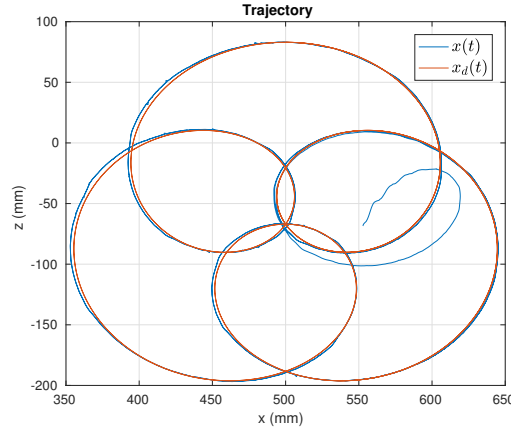
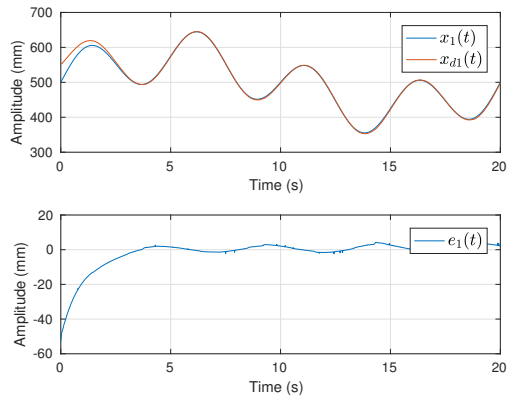
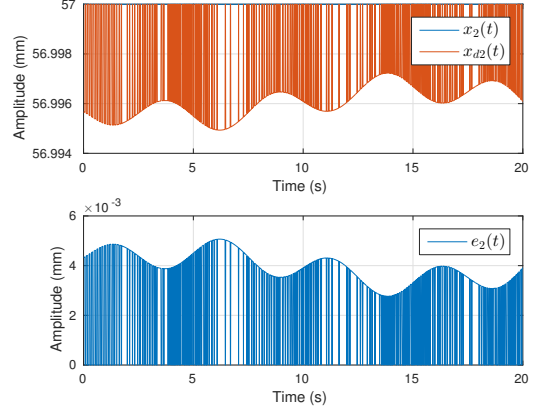


Figura 5.3: Trajetória

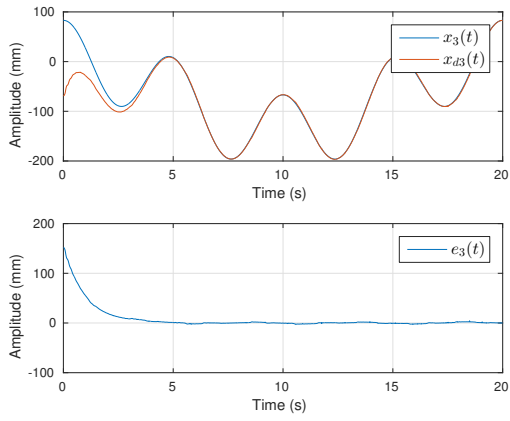
Podem-se observar diversos picos no gráfico do erro na figura 5.5 que se devem ao fato de, por não se tratar de um sistema em tempo-real, o período de controle pode sofrer variação caso aconteça alguma interrupção no sistema operacional. Assim, o comando não seria atualizado, resultando em um erro maior na próxima iteração.



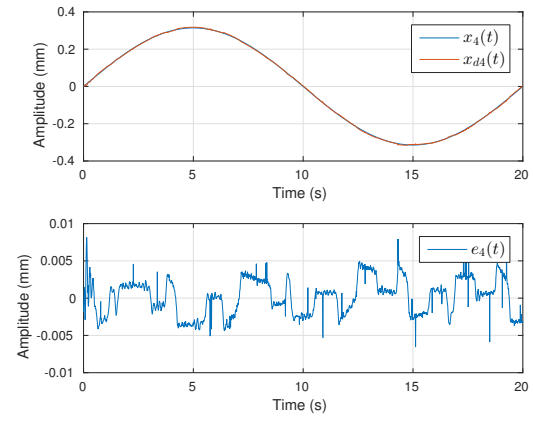
(a) x_1 , x_{d1} e e_1



(b) x_2 , x_{d2} e e_2

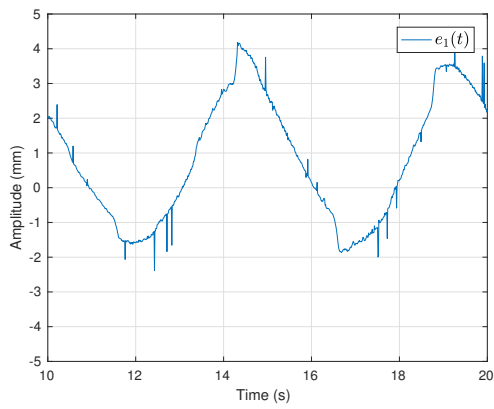


(c) x_3 , x_{d3} e e_3

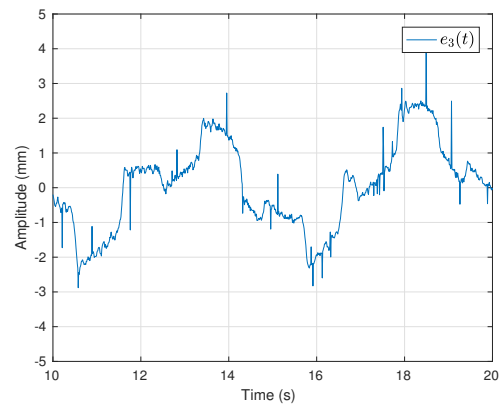


(d) x_4 , x_{d4} e e_4

Figura 5.4: Rastreamento de trajetória



(a) e_1



(b) e_3

Figura 5.5: Destaque para o erro e_1 e e_3

Capítulo 6

Conclusões

Referências Bibliográficas

Apêndice A

Apêndice

A.1 ViSP - Visual Servoing Platform

Para a implementação do modo de controle por Servo Visão, foi utilizada a biblioteca ViSP. A ViSP contém um módulo de visão computacional que permite computar a *pose* de um objeto a ser reconhecido por meio de um padrão pre-determinado. É utilizado um esquema robusto e são fornecidos mecanismos para calibração da câmera.