



**UNICAMP**

UNIVERSIDADE ESTADUAL DE CAMPINAS

## **ES827 - Projeto de Robótica Industrial**

*Nome*

Matheus Gustavo Alves Sasso  
Pedro Jairo Nogueira Pinheiro Neto  
Breno Vicente de Cerqueira  
Otávio Leite Bastos  
Tibério Duarte Alves Rodrigues Ferreira

*RA*

158257  
156992  
154817  
147567  
139187

June 24, 2019

# Contents

<b>1</b>	<b>Introdução</b>	<b>4</b>
1.1	Justificativa para tarefa proposta . . . . .	5
1.2	Justificativa para seleção dos manipuladores escolhidos . . . . .	5
1.3	Possibilidade do uso deste manipulador em outras tarefas . . . . .	5
<b>2</b>	<b>Cinemática Direta</b>	<b>6</b>
2.1	Número de graus de liberdade . . . . .	6
2.2	Tabela de Denavit-Hartenberg . . . . .	7
2.3	Matrizes Homogêneas . . . . .	11
2.3.1	Relativas(Entre Juntas) . . . . .	12
2.3.2	Em relação ao referencial inercial . . . . .	13
<b>3</b>	<b>Singularidades</b>	<b>16</b>
3.1	Robô RRRRRR . . . . .	16
3.2	Robô PPPR . . . . .	19
<b>4</b>	<b>Incertezas do posicionamento da ferramenta</b>	<b>20</b>
<b>5</b>	<b>Área máxima de trabalho e limitação de movimento das juntas</b>	<b>24</b>
5.1	Dimensionamento dos Robôs . . . . .	25
5.1.1	RRRRRR . . . . .	25
5.1.2	PPPR . . . . .	27
<b>6</b>	<b>Trajetória e Cinemática Inversa</b>	<b>28</b>
6.1	Linear Segments with Parabolic Blends . . . . .	29
6.2	Cinemática Inversa . . . . .	31
6.2.1	Robô cartesiano . . . . .	31
6.2.2	Robô articulado . . . . .	32
6.3	Variáveis das juntas em função do tempo . . . . .	35
6.3.1	PPPR . . . . .	35
6.3.2	RRRRRR . . . . .	36
<b>7</b>	<b>Modelo Dinâmico</b>	<b>38</b>
7.1	Energia cinética . . . . .	38
7.2	RRRRRR . . . . .	38
7.3	PPPR . . . . .	38
7.4	Energia potencial . . . . .	38
7.4.1	RRRRRR . . . . .	39
7.4.2	PPPR . . . . .	39
7.5	Equação de movimento . . . . .	39
7.5.1	RRRRRR . . . . .	39
7.5.2	PPPR . . . . .	40
<b>8</b>	<b>Estratégia de Controle</b>	<b>43</b>
8.1	PPPR . . . . .	43
8.2	RRRRRR . . . . .	46

<b>9</b>	<b>Esforços resultantes nas juntas</b>	<b>47</b>
9.1	PPPR . . . . .	48
9.2	RRRRRR . . . . .	49
<b>10</b>	<b>Vantagens e desvantagens de cada modelo</b>	<b>51</b>
10.1	Velocidade de execução da tarefa . . . . .	51
10.2	Esforços envolvidos . . . . .	51
10.3	Presença de singularidades . . . . .	51
10.4	Incerteza de posicionamento da ferramenta . . . . .	51
10.5	Erros de controle . . . . .	51
10.6	Problemas encontrados . . . . .	52
<b>11</b>	<b>Escolha do manipulador</b>	<b>53</b>
<b>12</b>	<b>Códigos</b>	<b>54</b>
12.1	Singularidades . . . . .	54
12.1.1	PPPR . . . . .	54
12.1.2	RRRRRR . . . . .	56
12.2	Dinâmica . . . . .	58
12.2.1	PPPR . . . . .	58
12.2.2	RRRRRR . . . . .	65
12.3	Trajectoria . . . . .	74
12.3.1	PPPR . . . . .	74
12.3.2	RRRRRR . . . . .	79
12.4	Esforços . . . . .	89
12.4.1	PPPR . . . . .	89
12.4.2	RRRRRR . . . . .	90
<b>13</b>	<b>Referência Bibliográficas</b>	<b>92</b>

## List of Figures

1	A curva em vermelho mostra a trajetória do robô para a pintura do veículo automotivo. O carro na imagem é meramente ilustrativo. . . . .	4
2	Primeiro robô escolhido, RRRRRR . . . . .	6
3	Segundo robô escolhido, PPPR . . . . .	6
4	Dimensões do robô cartesiano. . . . .	7
5	Dimensões do robô cartesiano. . . . .	7
6	Eixos do Robô RRRRRR . . . . .	9
7	Eixos do Robô PPPR . . . . .	11
8	Área de trabalho de um manipulador articulado. . . . .	24
9	Área de trabalho de um robô cartesiano. . . . .	24
10	Distância mínima da base para área de trabalho. . . . .	25
11	Distância mínimo entre robô e área de pintura. . . . .	26
12	Distância máxima entre base do robô e área de pintura. . . . .	26
13	Dimensões do robô cartesiano. . . . .	27
14	Dimensões do robô cartesiano. . . . .	27
15	A curva em vermelho mostra a trajetória do robô para a pintura do veículo automotivo. O carro na imagem é meramente ilustrativo. . . . .	28
16	Representação das juntas da base do robô conforme dado em [1]. . . . .	32
17	Representação das juntas da base do robô mais um offset "d" conforme dado em [1]. . . . .	33
18	Projeções dos links 2 e 3 no plano. . . . .	33
19	Valor das variáveis das juntas com o tempo. . . . .	35
20	Valor das variáveis das juntas 1 a 3 com o tempo. . . . .	36
21	Valor das variáveis das juntas 4 a 6 com o tempo. . . . .	37
22	Resumo da estratégia de controle. . . . .	43
23	Datasheet do atuador tomado como referência. . . . .	43
24	Implementação Simulink. . . . .	44
25	Esforço resultante nas juntas. . . . .	45
26	Erro de controle. . . . .	45
27	Controle Feed-forward . . . . .	46
28	Atuador linear tomado como base. . . . .	47
29	Robo PPPR. . . . .	47
30	Esforços nas juntas 2 e 3 em função do tempo para o robô cartesiano. . . . .	49
31	Esforços nas juntas 1 a 6 em função do tempo para o robô articulado. . . . .	50

## List of Tables

1	Constantes de dimensões utilizadas para o modelo. . . . .	4
2	Tabela de Denavit-Hartenberg para robô RRRRRR . . . . .	8
3	Tabela de Denavit-Hartenberg para robô PPPR . . . . .	10
4	Incertezas relativas ao manipulador articulado. . . . .	20
5	Incertezas relativas ao manipulador cartesiano. . . . .	20
6	Constantes de dimensões utilizadas para o modelo. . . . .	29
7	Constantes de dimensões utilizadas para o modelo. . . . .	29
8	Tempos de movimento e velocidades usadas nos modelos. . . . .	31

# 1 Introdução

A tarefa escolhida foi a pintura da lateral de carros. Nesta tarefa, um robô manipulador deverá borrifar o material de pintura da superfície metálica de veículos automotivos de cima para baixo e da esquerda para a direita indo e voltando, realizando uma trajetória de modo a completar a pintura da superfície lateral do automóvel enquanto ele é movido numa linha de produção durante a sua fabricação.

Esse carro estará passando numa esteira numa linha de produção. Dessa forma, quando o carro chega na etapa de pintura da lateral, a esteira deve parar, o robô faz a pintura e depois o carro continua para as etapas seguintes da sua manufatura.

A trajetória está ilustrada na Figura 15 abaixo.

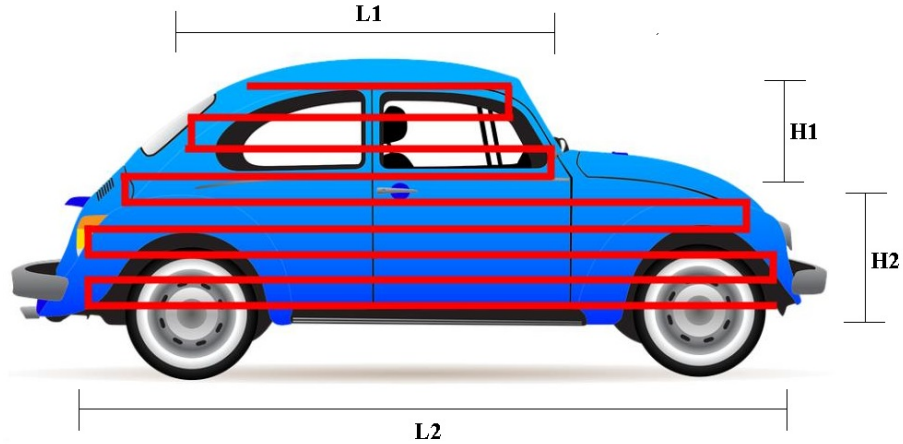


Figure 1: A curva em vermelho mostra a trajetória do robô para a pintura do veículo automotivo. O carro na imagem é meramente ilustrativo.

O modelo de lateral de carroceria utilizado será uma versão simplificada do modelo real para que uma trajetória não muito complexa seja criada. De toda forma, o material de pintura deve ser borrifado à velocidade constante na superfície, logo ele se espalhará e por isso não é preciso ter uma trajetória tão fina.

O modelo de carroceria utilizado pode ser compreendido como dois retângulos, um superior e um inferior. O superior teria  $L_1$  metros de comprimento ao longo do eixo X do sistema de coordenadas adotado e  $H_1$  metros de altura ao longo do eixo y. O mesmo é válido para o retângulo inferior com  $L_1$  metros de comprimento e  $H_2$  metros de altura. Além disso, a tabela 7 mostra os valores escolhidos para as constantes.

Table 1: Constantes de dimensões utilizadas para o modelo.

Constantes	Valores
$L_1$	2.5
$H_1$	0.5
$L_2$	4
$H_2$	1

As posições de referência utilizadas para definir a trajetória e a trajetória em si serão apresentados na seção de cinemática inversa.

## **1.1 Justificativa para tarefa proposta**

Essa tarefa foi escolhida por ser uma tarefa real de manipuladores em sistemas produtivos. De fato o mercado de veículos automotivos é muito grande no mundo inteiro e existem diversas técnicas de pintura. Porém dada a complexidade de todo o processo nós decidimos pegar uma tarefa específica como é o caso da pintura apenas da lateral do veículo para fazer um estudo e análise do desempenho de dois manipuladores.

Neste tipo de tarefa um manipulador pode facilitar e agilizar bastante o processo de montagem. Em sistemas produtivos de veículos automotivos a pintura quase sempre é a parte mais demorada, ou seja, o gargalo produtivo mesmo considerando-se que ela já esteja automatizada, o que mostra uma clara necessidade de processos automáticos para a sua realização.

## **1.2 Justificativa para seleção dos manipuladores escolhidos**

Os manipuladores foram escolhidos de modo a apresentarem princípios totalmente diferentes como é o caso dos robôs RRRRRR (manipulador articulado com punho esférico) e PPPR (manipulador cartesiano com pulho rotacional). Enquanto o primeiro tem uma base suportada por juntas rotacionais, o segundo tem uma base suportada por juntas prismáticas. A ideia é ver o quanto um pode ser mais simples ou complexo do que o outro em termos de modelização, qual deles tem o melhor desempenho e se o fato de as bases terem tipos de movimentos diferentes influencia nesses resultados.

A hipótese inicial é de que o modelo do manipulador RRRRRR será mais complexo e difícil de gerar do que o do robô cartesiano, porém ele terá uma maior liberdade de movimentos e maior flexibilidade de utilização. O quanto isso vai compensar a complexidade do modelo e se essa flexibilidade é mais importante do que simplicidade depende apenas dos requisitos de projetos. A ideia é poder mostrar em quais situações é preferível usar um ou o outro.

## **1.3 Possibilidade do uso deste manipulador em outras tarefas**

O manipulador RRRRRR é bastante genérico e tem uma grande área de atuação assim como o PPPR, logo eles podem ser readaptados para diversas outras tarefas, até mesmo de pintura de outras partes, contudo eles precisariam ser montados de outra forma e ser instalados em outro ponto da linha de montagem ou mesmo em uma parte separada.

## 2 Cinemática Direta

### 2.1 Número de graus de liberdade

Para o robô do tipo RRRRRR mostrado abaixo podemos observar que ele pode ser modelado por 6 juntas rotacionais, cada uma contribuindo com um grau de liberdade rotacional.

Logo este robô possui 6 graus de liberdade.

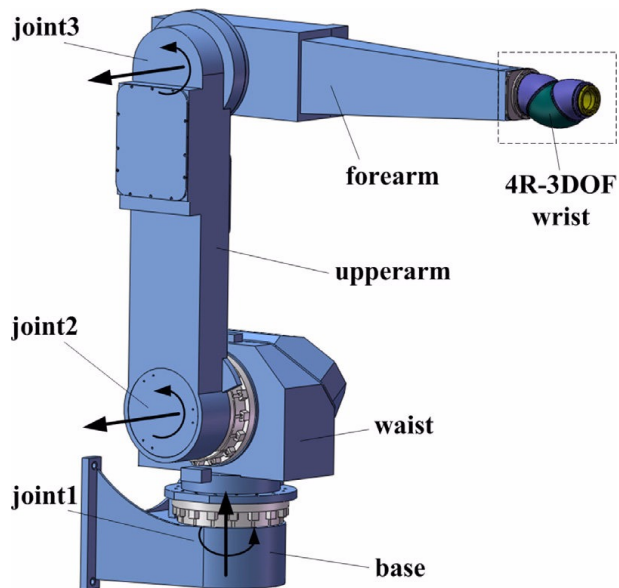


Figure 2: Primeiro robô escolhido, RRRRRR

Analogamente o robô abaixo é composto por 3 juntas prismáticas e uma junta rotacional. Cada junta prismática contribui com um grau de liberdade, assim como a junta rotacional. Logo este robô possui 4 graus de liberdade.



Figure 3: Segundo robô escolhido, PPPR

## 2.2 Tabela de Denavit-Hartenberg

### Dimensões dos robôs

Antes de aplicar o método de Denavit-Hartenberg, é preciso determinar dimensões dos robôs adequadas a nossa aplicação alvo. As dimensões escolhidas estão mostradas abaixo. A discussão do porquê destas dimensões foram feitas na seção *Área máxima de trabalho e limitação de movimento das juntas*.

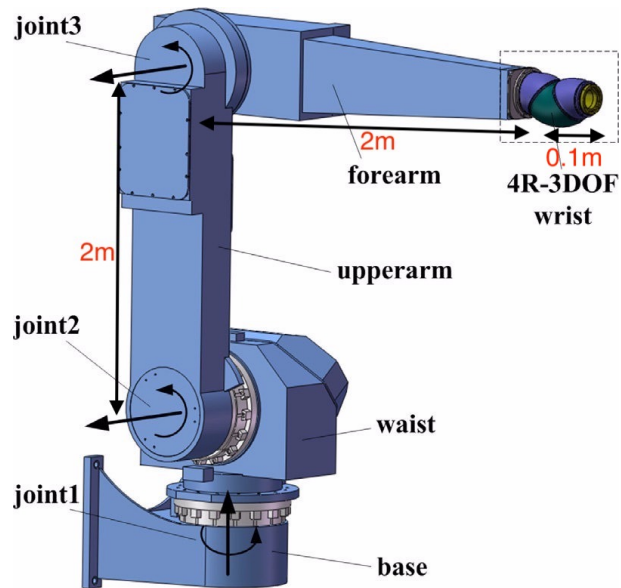


Figure 4: Dimensões do robô cartesiano.



Figure 5: Dimensões do robô cartesiano.

### Tabelas de Denavit-Hartenberg

A tabela de Denavit-Hartenberg é utilizada para se modelar a cinemática direta do manipulador. Para os robôs escolhidos é dada por:

- Robô RRRRRR



Table 2: Tabela de Denavit-Hartenberg para robô RRRRRR

<b>Elo</b>	$\theta$	<b>d</b>	<b>a</b>	$\alpha$
1	$\theta_1^*$	0	0	$-90^\circ$
2	$\theta_2^*$	0	$a_2$	0
3	$\theta_3^*$	0	$a_3$	$-90^\circ$
4	$\theta_4^*$	0	0	$90^\circ$
5	$\theta_5^*$	0	0	$-90^\circ$
6	$\theta_6^*$	$d_6$	0	0

Com os valores de a e de definidos por:

1.  $a_2 = 2\text{m}$
2.  $a_3 = 2\text{m}$
3.  $d_6 = 0.1\text{m}$

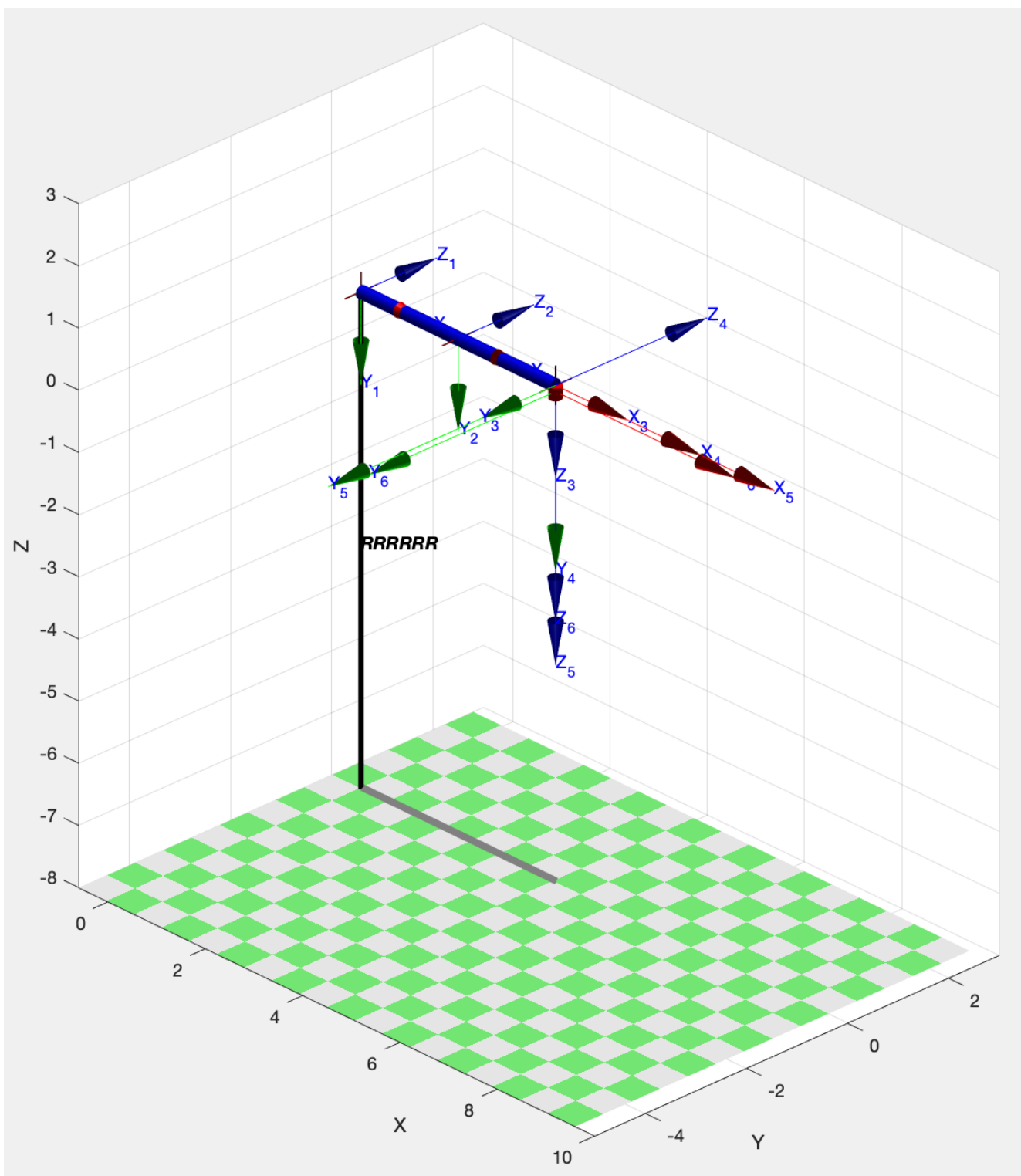


Figure 6: Eixos do Robô RRRRRR

- Robô PPR

Com os valores de  $a$  e  $d$  definidos por:

1.  $d1^* =$  de 0 a 1m
2.  $d2^* =$  de 0 a 5m

Table 3: Tabela de Denavit-Hartenberg para robô PPPR

<b>Elo</b>	$\theta$	<b>d</b>	<b>a</b>	$\alpha$
1	0	$d_1^*$	0	$90^\circ$
2	$90^\circ$	$d_2^*$	0	$-90^\circ$
3	0	$d_3^*$	0	$90^\circ$
4	$\theta_4^*$	0	0	$-90^\circ$

3.  $d_3^* =$  de 0 a 2m

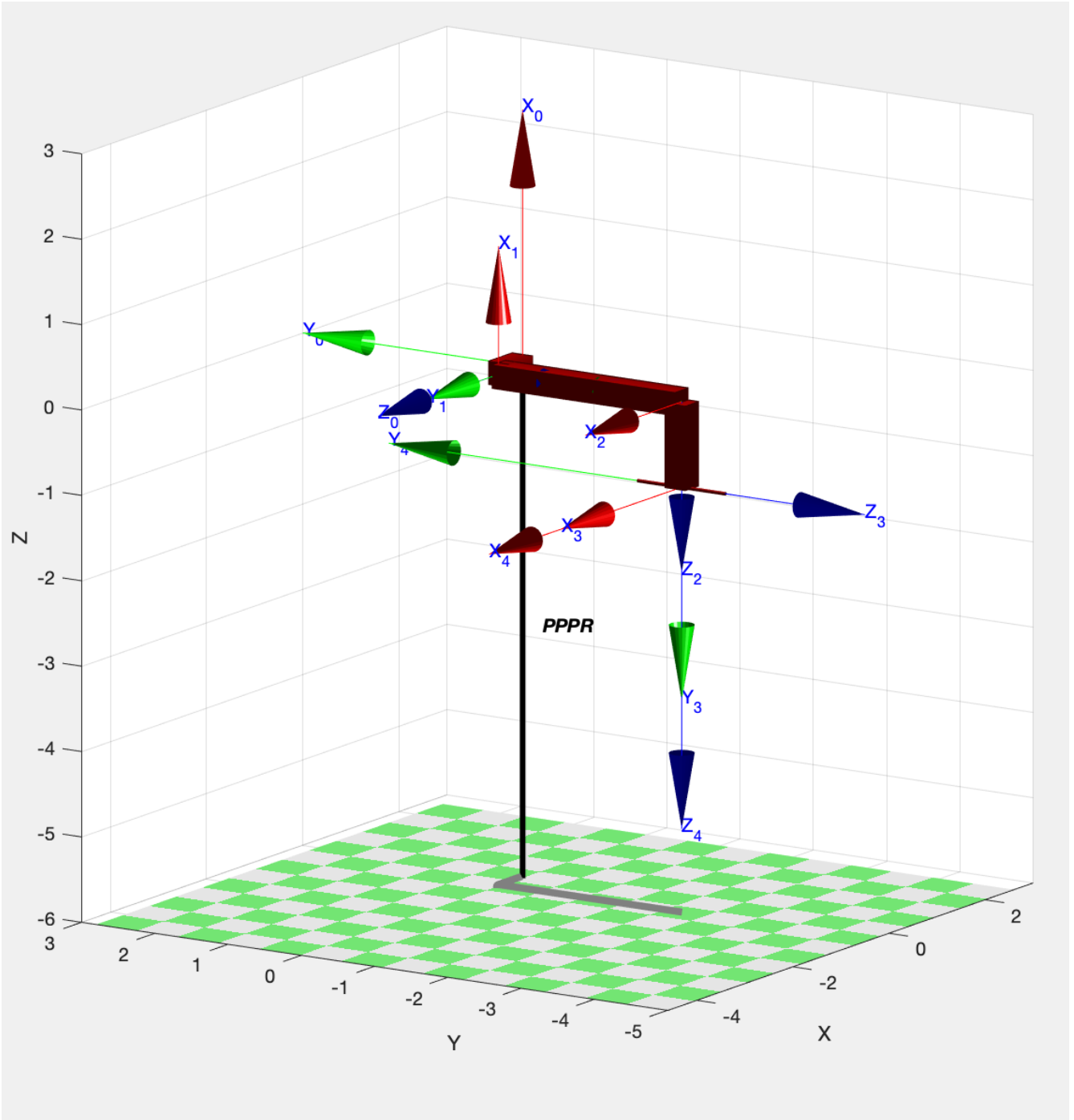


Figure 7: Eixos do Robô PPPR

### 2.3 Matrizes Homogêneas

Para encontrar as Matrizes homogêneas relativas entre as juntas basta aplicar a convenção de Denavit-Hartenberg a partir de cada linha da tabela. A mesma é dada por:

$$A_i = Rot_{z,\theta_i} \cdot Trans_{z,d_i} \cdot Trans_{x,a_i} \cdot Rot_{x,\alpha_i} \quad (1)$$

### 2.3.1 Relativas(Entre Juntas)

- Robô RRRRRR

$$A_1 = H_1^0 = \begin{pmatrix} c_1 & 0 & -s_1 & 0 \\ s_1 & 0 & c_1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$A_2 = H_2^1 = \begin{pmatrix} c_2 & -s_2 & 0 & a_2 c_2 \\ s_2 & c_2 & 0 & a_2 s_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$A_3 = H_3^2 = \begin{pmatrix} c_3 & 0 & -s_3 & a_3 c_3 \\ s_3 & 0 & c_3 & a_3 s_3 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$A_4 = H_4^3 = \begin{pmatrix} c_4 & 0 & s_4 & 0 \\ s_4 & 0 & -c_4 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$A_5 = H_5^4 = \begin{pmatrix} c_5 & 0 & -s_5 & 0 \\ s_5 & 0 & c_5 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$A_6 = H_6^5 = \begin{pmatrix} c_6 & -s_6 & 0 & 0 \\ s_6 & c_6 & 0 & 0 \\ 0 & 0 & 1 & d_6 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

- Robô PPPR

$$A_1 = H_1^0 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & d_1^* \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$A_2 = H_2^1 = \begin{pmatrix} 0 & 0 & -1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & d_2^* \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$A_3 = H_3^2 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & d_3^* \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$A_4 = H_4^3 = \begin{pmatrix} c_4 & 0 & -s_4 & 0 \\ s_4 & 0 & c_4 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

### 2.3.2 Em relação ao referencial inercial

- Robô RRRRRR

$$A_1 = H_1^0 = \begin{pmatrix} c_1 & 0 & -s_1 & 0 \\ s_1 & 0 & c_1 & 0 \\ 0 & -1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$H_2^0 = \begin{pmatrix} c_1 c_2 & -c_1 s_2 & -s_1 & a_2 c_1 c_2 \\ c_2 s_1 & -s_1 s_2 & c_1 & a_2 s_1 c_2 \\ -s_2 & -c_2 & 0 & -a_2 s_2 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$H_3^0 = \begin{pmatrix} c_1 c_2 & -c_1 s_2 & -s_1 & a_2 c_1 c_2 \\ c_2 s_1 & -s_1 s_2 & c_1 & a_2 c_2 s_1 \\ -s_2 & -c_2 & 0 & -a_2 s_2 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$H_4^0 = \begin{pmatrix} s_1 s_4 - c_4(c_1 s_2 s_3 - c_1 c_2 c_3) & -s_{23} c_1 & -c_4 s_1 - s_4(c_1 s_2 s_3 - c_1 c_2 c_3) & c_1(a_3 c_{23} + a_2 c_2) \\ -c_1 s_4 - c_4(s_1 s_2 s_3 - c_2 c_3 s_1) & -s_{23} s_1 & c_1 c_4 - s_4(s_1 s_2 s_3 - c_2 c_3 s_1) & s_1(a_3 c_{23} + a_2 c_2) \\ -s_{23} c_4 & -c_{23} & -s_{23} s_4 & -a_3 s_{23} - a_2 s_2 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$H_5^0 = \begin{pmatrix} r_{11} & r_{12} & r_{13} & r_{14} \\ r_{21} & r_{22} & r_{23} & r_{24} \\ r_{31} & r_{32} & r_{33} & r_{34} \\ r_{41} & r_{42} & r_{43} & r_{44} \end{pmatrix}$$

$$\begin{aligned} r_{11} &= c_5(s_1 s_4 - c_4(c_1 s_2 s_3 - c_1 c_2 c_3)) - s_5(c_1 c_2 s_3 + c_1 c_3 s_2) \\ r_{21} &= -c_5(c_1 s_4 + c_4(s_1 s_2 s_3 - c_2 c_3 s_1)) - s_5(c_2 s_1 s_3 + c_3 s_1 s_2) \\ r_{31} &= -c_{23} s_5 - s_{23} c_4 c_5 \\ r_{41} &= 0 \\ r_{12} &= c_4 s_1 + s_4(c_1 s_2 s_3 - c_1 c_2 c_3) \\ r_{22} &= s_4(s_1 s_2 s_3 - c_2 c_3 s_1) - c_1 c_4 \\ r_{32} &= s_{23} s_4 \\ r_{42} &= 0 \\ r_{13} &= -s_5(s_1 s_4 - c_4(c_1 s_2 s_3 - c_1 c_2 c_3)) - c_5(c_1 c_2 s_3 + c_1 c_3 s_2) \\ r_{23} &= s_5(c_1 s_4 + c_4(s_1 s_2 s_3 - c_2 c_3 s_1)) - c_5(c_2 s_1 s_3 + c_3 s_1 s_2) \\ r_{33} &= s_{23} c_4 s_5 - c_{23} c_5 \\ r_{43} &= 0 \\ r_{14} &= c_1(a_3 c_{23} + a_2 c_2) \\ r_{24} &= s_1(a_3 c_{23} + a_2 c_2) \\ r_{34} &= -a_3 s_{23} - a_2 s_2 \\ r_{44} &= 1 \end{aligned}$$

$$H_6^0 = \begin{pmatrix} r_{11} & r_{12} & r_{13} & r_{14} \\ r_{21} & r_{22} & r_{23} & r_{24} \\ r_{31} & r_{32} & r_{33} & r_{34} \\ r_{41} & r_{42} & r_{43} & r_{44} \end{pmatrix}$$

$$\begin{aligned} r_{11} &= s_6(c_4 s_1 + s_4(c_1 s_2 s_3 - c_1 c_2 c_3)) + c_6(c_5(s_1 s_4 - c_4(c_1 s_2 s_3 - c_1 c_2 c_3)) - s_5(c_1 c_2 s_3 + c_1 c_3 s_2)) \\ r_{21} &= -s_6(c_1 c_4 - s_4(s_1 s_2 s_3 - c_2 c_3 s_1)) - c_6(c_5(c_1 s_4 + c_4(s_1 s_2 s_3 - c_2 c_3 s_1)) + s_5(c_2 s_1 s_3 + c_3 s_1 s_2)) \\ r_{31} &= s_{23} s_4 s_6 - c_6(c_{23} s_5 + s_{23} c_4 c_5) \\ r_{41} &= 0 \end{aligned}$$

$$\begin{aligned}
r12 &= c_6(c_4s_1 + s_4(c_1s_2s_3 - c_1c_2c_3)) - s_6(c_5(s_1s_4 - c_4(c_1s_2s_3 - c_1c_2c_3)) - s_5(c_1c_2s_3 + c_1c_3s_2)) \\
r22 &= s_6(c_5(c_1s_4 + c_4(s_1s_2s_3 - c_2c_3s_1)) + s_5(c_2s_1s_3 + c_3s_1s_2)) - c_6(c_1c_4 - s_4(s_1s_2s_3 - c_2c_3s_1)) \\
r32 &= s_6(c_{23}s_5 + s_{23}c_4c_5) + s_{23}c_6s_4 \\
r42 &= 0 \\
r13 &= -s_5(s_1s_4 - c_4(c_1s_2s_3 - c_1c_2c_3)) - c_5(c_1c_2s_3 + c_1c_3s_2) \\
r23 &= s_5(c_1s_4 + c_4(s_1s_2s_3 - c_2c_3s_1)) - c_5(c_2s_1s_3 + c_3s_1s_2) \\
r33 &= s_{23}c_4s_5 - c_{23}c_5 \\
r43 &= 0 \\
r14 &= a2c_1c_2 - d6(s_5(s_1s_4 - c_4(c_1s_2s_3 - c_1c_2c_3)) + c_5(c_1c_2s_3 + c_1c_3s_2)) + a3c_1c_2c_3 - a3c_1s_2s_3 \\
r24 &= d6(s_5(c_1s_4 + c_4(s_1s_2s_3 - c_2c_3s_1)) - c_5(c_2s_1s_3 + c_3s_1s_2)) + a2c_2s_1 + a3c_2c_3s_1 - a3s_1s_2s_3 \\
r34 &= (d6s_{23}s_{45})/2 - a2s_2 - a3s_{23} - d6c_{23}c_5 - (d6\sin(th4 - th5)s_{23})/2 \\
r44 &= 1
\end{aligned}$$

- Robô PPPR

$$A_1 = H_1^0 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$H_2^0 = \begin{bmatrix} 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & -d_2 \\ 1 & 0 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$H_3^0 = \begin{bmatrix} 0 & -1 & 0 & -d_3 \\ 0 & 0 & -1 & -d_2 \\ 1 & 0 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$H_4^0 = \begin{bmatrix} -s_4 & 0 & -c_4 & -d_3 \\ 0 & 1 & 0 & -d_2 \\ c_4 & 0 & -s_4 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



### 3 Singularidades

É sempre necessário analisar as singularidades de manipuladores robóticos afim de evitar que esses robôs operem nessas zonas, que podem causar perdas abruptas e inesperadas de rigidez e manipulabilidade dos robôs. Pra encontrar as é necessário construir o Jacobiano e observar em quais casos o mesmo diminui os valores do seu range. O número de colunas do Jacobiano é dado pelo número de elos do manipulador. Além disso o mesmo pode ser dividido horizontalmente entre translação e rotação e verticalmente entre estrutura do robô( $J_P$ ) e garra( $J_O$ ). Como podemos observar a seguir:

- **Divisão Horizontal**

$$J = \begin{bmatrix} J_{trans} & J_{rot} \end{bmatrix}$$

- **Divisão Vertical**

$$J = \begin{bmatrix} J_P \\ J_O \end{bmatrix}$$

Desta maneira, criamos 4 regiões  $J_{11}$ ,  $J_{12}$ ,  $J_{21}$  e  $J_{22}$  na forma:

$$J = \begin{bmatrix} J_{11} & J_{12} \\ J_{21} & J_{22} \end{bmatrix}$$

#### 3.1 Robô RRRRRR

O jacobiano para o robô de 6 juntas rotacionais é dado por  $J$  na forma:

$$J = \begin{bmatrix} z_0 \times (o_6 - o_0) & z_1 \times (o_6 - o_1) & z_2 \times (o_6 - o_2) & z_3 \times (o_6 - o_3) & z_4 \times (o_6 - o_4) & z_5 \times (o_6 - o_5) \\ z_0 & z_1 & z_2 & z_3 & z_4 & z_5 \end{bmatrix}$$

Sendo  $z_i$  o vetor de tridimensional correspondente à coluna 3, linhas 1 a 3, e o  $o_i$  o vetor tridimensional correspondente à coluna 4, linhas 1 a 3, da matriz  $H_{i+1}^i$  calculada na seção anterior. Os valores de  $a_2$ ,  $a_3$  e  $d_6$  foram escolhidos como 2, 2 e 0 respectivamente, conforme detalhado na seção 5.

Dividindo a matriz nas quatro regiões  $J_{11}$ ,  $J_{12}$ ,  $J_{21}$  e  $J_{22}$  na forma:

$$J = \begin{bmatrix} J_{11} & J_{12} \\ J_{21} & J_{22} \end{bmatrix}$$

Para encontrar as singularidades do braço robótico, queremos encontrar um conjunto  $\{\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6 | \det(J) = 0\}$ . Para isso,  $\det J_{ij} = 0 \forall (i, j)$  é condição suficiente.

Com o auxílio do *Matlab* é trivial encontrar que  $\det J_{12} = \det J_{21} = 0$ . Resta verificar se  $J_{11} = 0$  e  $J_{22} = 0$ . Vamos, então, nos focalizar sobre o estudo do determinante dos jacobianos  $J_{11}$  e  $J_{22}$ .

Podemos dividir o quadrante  $J_{11}$  da matriz jacobiana na forma:

$$J_{11} = \begin{bmatrix} J_{11}^{11} & J_{12}^{11} & J_{13}^{11} \\ J_{21}^{11} & J_{22}^{11} & J_{23}^{11} \\ J_{31}^{11} & J_{32}^{11} & J_{33}^{11} \end{bmatrix}$$

Sendo cada  $J_{ij}^{11}$  o elemento da matriz  $J_{11}$  na linha  $i$  e coluna  $j$ , tal que:

- $J_{11}^{11} = -2 \sin \theta_1 (\cos(\theta_2 + \theta_3) + \cos \theta_2)$
- $J_{12}^{11} = -\cos \theta_1 (2 \sin(\theta_2 + \theta_3) + 2 \sin \theta_2)$
- $J_{13}^{11} = -2 \sin \theta_2 + \theta_3 \cos \theta_1$
- $J_{21}^{11} = 2 \cos \theta_1 (\cos(\theta_2 + \theta_3) + \cos \theta_2)$
- $J_{22}^{11} = -\sin \theta_1 (2 \sin(\theta_2 + \theta_3) + 2 \sin \theta_2)$
- $J_{23}^{11} = -2 \sin \theta_2 + \theta_3 \sin \theta_1$
- $J_{31}^{11} = 0$
- $J_{32}^{11} = -2 \cos(\theta_2 + \theta_3) - 2 \cos \theta_2$
- $J_{33}^{11} = -2 \cos(\theta_2 + \theta_3)$

Utilizando a função *solve* e *eval* do *Matlab* para avaliar para que valores de  $\theta_1, \theta_2, \theta_3, \theta_4, \theta_5$  e  $\theta_6$  para os quais teríamos  $J_{11} = 0$ , encontram-se 3 possibilidades:

• **Caso 1:**

$$\theta_1 = 0 \quad \theta_2 = 0 \quad \theta_3 = 0 \quad \theta_4 = 0 \quad \theta_5 = 0 \quad \theta_6 = 0 \quad (2)$$

• **Caso 2:**

$$\theta_1 = 0 \quad \theta_2 = 0 \quad \theta_3 = 180^\circ \quad \theta_4 = 0 \quad \theta_5 = 0 \quad \theta_6 = 0 \quad (3)$$

• **Caso 3:**

$$\theta_1 = 0 \quad \theta_2 = 61.3521^\circ \quad \theta_3 = 57.2958^\circ \quad \theta_4 = 0 \quad \theta_5 = 0 \quad \theta_6 = 0 \quad (4)$$

É relevante notar que se trata de singularidades referentes ao movimento de translação e à estrutura do robô.

Podemos agora nos concentrar sobre a matriz  $J_{22}$  que é da forma:

$$J_{22} = \begin{bmatrix} J_{11}^{22} & J_{12}^{22} & J_{13}^{22} \\ J_{21}^{22} & J_{22}^{22} & J_{23}^{22} \\ J_{31}^{22} & J_{32}^{22} & J_{33}^{22} \end{bmatrix}$$

Sendo cada  $J_{ij}^{22}$  o elemento da matriz  $J_{22}$  na linha  $i$  e coluna  $j$ , tal que:

- $J_{11}^{22} = -\sin(\theta_2 + \theta_3) \cos \theta_1$
- $J_{12}^{22} = -\cos \theta_4 \sin \theta_1 - \sin \theta_4 (\cos \theta_1 \sin \theta_2 \sin \theta_3 - \cos \theta_1 \cos \theta_2 \cos \theta_3)$
- $J_{13}^{22} = -\sin \theta_5 (\sin \theta_1 \sin \theta_4 - \cos \theta_4 (\cos \theta_1 \sin \theta_2 \sin \theta_3 - \cos \theta_1 \cos \theta_2 \cos \theta_3)) - \cos \theta_5 (\cos \theta_1 \cos \theta_2 \sin \theta_3 + \cos \theta_1 \cos \theta_3 \sin \theta_2)$
- $J_{21}^{22} = -\sin(\theta_2 + \theta_3) \sin \theta_1$
- $J_{22}^{22} = \cos \theta_1 \cos \theta_4 - \sin \theta_4 (\sin \theta_1 \sin \theta_2 \sin \theta_3 - \cos \theta_2 \cos \theta_3 \sin \theta_1)$
- $J_{23}^{22} = \sin \theta_5 (\cos \theta_1 \sin \theta_4 + \cos \theta_4 (\sin \theta_1 \sin \theta_2 \sin \theta_3 - \cos \theta_2 \cos \theta_3 \sin \theta_1)) - \cos \theta_5 (\cos \theta_2 \sin \theta_1 \sin \theta_3 + \cos \theta_3 \sin \theta_1 \sin \theta_2)$
- $J_{31}^{22} = -\cos(\theta_2 + \theta_3)$
- $J_{32}^{22} = -\sin(\theta_2 + \theta_3) \sin \theta_4$
- $J_{33}^{22} = \sin(\theta_2 + \theta_3) \cos \theta_4 \sin \theta_5 - \cos(\theta_2 + \theta_3) \cos \theta_5$

Utilizando a função *solve* e *eval* do *Matlab* para avaliar para que valores de  $\theta_1, \theta_2, \theta_3, \theta_4, \theta_5$  e  $\theta_6$  para os quais teríamos  $J_{22} = 0$ , encontra-se uma única possibilidade:

$$\theta_1 = 0 \quad \theta_2 = 0 \quad \theta_3 = 0 \quad \theta_4 = 0 \quad \theta_5 = 0 \quad \theta_6 = 0 \quad (5)$$

É relevante notar que se trata de singularidades referentes ao movimento de rotação e à estrutura da garra.

Esse conjunto de ângulos é redundante, pois já está contemplado no caso 1 do estudo feito para o jacobiano  $J_{11}$ .

Quando há singularidades, como neste caso, significa que há pontos na área de trabalho (detalhada na seção 5) que não podem ser alcançados pelo robô.

### 3.2 Robô PPPR

O jacobiano para o robô de 6 juntas rotacionais é dado por  $J$  na forma:

$$J = \begin{bmatrix} z_0 & z_1 & z_2 & z_3 \times (o_4 - o_3) \\ 0 & 0 & 0 & z_3 \end{bmatrix}$$

Sendo  $z_i$  o vetor de tridimensional correspondente à coluna 3, linhas 1 a 3, da matriz  $H_{i+1}^i$  calculada na seção anterior. O valor de  $\theta_4$  foi escolhido como sendo igual a zero, conforme detalhado na seção 5.

Portanto, a matriz jacobiana do robô PPPR será dada por:

$$J = \begin{bmatrix} 0 & 0 & -1 & 0 \\ 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Pode-se buscar por singularidades procurando um conjunto de valores  $\{d_1, d_2, d_3 \text{ e } \theta_4\}$  para os quais  $\det J = 0$ .

É trivial notar que  $\det J = \det J^*$  tal que  $J^*$  é a matriz simplicada dada por:

$$J^* = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix}$$

Portanto, para todo e qualquer valor das variáveis do conjunto  $\{d_1, d_2, d_3 \text{ e } \theta_4\}$ , temos que:

$$\det J = \det J^* = 1 \tag{6}$$

Isso significa que não há casos em que o determinante do jacobiano do robô PPPR (com junta rotacional estática) possa vir a ser nulo. Logo, não há zonas nem pontos de singularidade para este manipulador sob as hipóteses assumidas.

Quando não há singularidades, como neste caso, significa que todo e qualquer ponto do plano cartesiano na área de trabalho (detalhada na seção 5) que podem ser alcançados pelo robô.

## 4 Incertezas do posicionamento da ferramenta

Quando fazemos uma medição de uma grandeza é necessário avaliar a qualidade dessa medição. Sabemos que por melhor que seja o instrumento e sua calibração, haverá sempre uma incerteza associada a ela. Assim, a incerteza descreve um intervalo em torno do valor medido, no qual deve estar contido o valor verdadeiro.

Por exemplo:

- Posição da ferramenta em  $Z = 1 \pm 0.001mm$
- limite superior =  $1.001mm$
- limite inferior =  $0.999mm$

Esse processo de identificação, quantificação e combinação de erro recebe o nome de análise de incerteza.

Neste projeto iremos supor incertezas de posicionamento ligadas aos sensores e à outros fatores de envelhecimento do robô e a partir desse valor global iremos analisar como o movimento e posição do robô afetam essa incerteza através da sensibilidade. Usaremos então o método de Kline e McClintock para calcular a incerteza de posicionamento em cada direção para a ferramenta de cada manipulador dado uma posição no espaço (x,y,z).

Assim as incertezas relacionadas a cada junta para os robôs foram escolhidas como mostrado nas tabelas 4 e 5 a seguir.

Table 4: Incertezas relativas ao manipulador articulado.

Junta	Incerteza de posicionamento
$\theta_1$	$1^\circ$
$\theta_2$	$2^\circ$
$\theta_3$	$1^\circ$
$\theta_4$	$2^\circ$
$\theta_5$	$1^\circ$
$\theta_6$	$2^\circ$

Table 5: Incertezas relativas ao manipulador cartesiano.

Junta	Incerteza de posicionamento
$d_1$	$1mm^\circ$
$d_2$	$1mm^\circ$
$d_3$	$2mm^\circ$
$\theta_4$	$2^\circ$

Os valores das tabelas acima foram escolhidos apenas para ilustrar o procedimento de análise de incerteza e para ter uma noção de como o erro pode se propagar através da análise das sensibilidades pelo método citado anteriormente. Dessa forma iremos definir uma única posição para fazer essa análise, sendo que para os demais pontos o cálculo seria o mesmo apenas mudando os valores das coordenadas. Assim nós escolhemos o ponto  $(L_2, H_2, 0)$  como ponto de análise cujo valor é  $(4, 1, 0)$ . Esse ponto fará parte da trajetória do robô como veremos na seção de trajetória.

O método de Kline e McClintock funciona da seguinte forma. Seja R a matriz de posicionamentos do robô uma função das variáveis de suas juntas tal que:

$$R = f(q_1, q_2, q_3, q_4, \dots, q_n) \quad (7)$$

A sensibilidade é dada por:

$$\Theta_i = \frac{\partial R}{\partial q_i} \quad (8)$$

Então a incerteza para a direção (x, y ou z) de R pode ser calculada como:

$$u_{direção} = \pm [(\frac{\partial R}{\partial q_1} u_{q_1})^2 + (\frac{\partial R}{\partial q_2} u_{q_2})^2 + \dots + (\frac{\partial R}{\partial q_n} u_{q_n})^2]^{\frac{1}{2}} \quad (9)$$

Portanto, dado que já temos as incertezas relativas à cada junta, o próximo passo é o cálculo das sensibilidades. Esse cálculo será feito relativo ao posicionamento do último elo, pois iremos analisar a incerteza na posição da ferramenta do manipulador. Assim teremos 18 derivadas parciais de R relativas a cada direção e cada variável "q" para o manipulador articulado e mais 12 derivadas para o manipulador cartesiano conforme mostrado abaixo.

Manipulador articulado (RRRRRR):

$$\frac{\partial X}{\partial \theta_1} = a_3 s_1 s_2 s_3 - a_2 c_2 s_1 - a_3 c_2 c_3 s_1 - d_6 (s_5 (c_1 s_4 + c_4 (s_1 s_2 s_3 - c_2 c_3 s_1)) - c_5 (c_2 s_1 s_3 + c_3 s_1 s_2)) \quad (10)$$

$$\frac{\partial X}{\partial \theta_2} = d_6 (c_5 (c_1 s_2 s_3 - c_1 c_2 c_3) + c_4 s_5 (c_1 c_2 s_3 + c_1 c_3 s_2)) - a_2 c_1 s_2 - a_3 c_1 c_2 s_3 - a_3 c_1 c_3 s_2 \quad (11)$$

$$\frac{\partial X}{\partial \theta_3} = d_6 (c_5 (c_1 s_2 s_3 - c_1 c_2 c_3) + c_4 s_5 (c_1 c_2 s_3 + c_1 c_3 s_2)) - a_3 c_1 c_2 s_3 - a_3 c_1 c_3 s_2 \quad (12)$$

$$\frac{\partial X}{\partial \theta_4} = -d_6 s_5 (c_4 s_1 + s_4 (c_1 s_2 s_3 - c_1 c_2 c_3)) \quad (13)$$

$$\frac{\partial X}{\partial \theta_5} = -d_6 (c_5 (s_1 s_4 - c_4 (c_1 s_2 s_3 - c_1 c_2 c_3)) - s_5 (c_1 c_2 s_3 + c_1 c_3 s_2)) \quad (14)$$

$$\frac{\partial X}{\partial \theta_6} = 0 \quad (15)$$

$$\frac{\partial Y}{\partial \theta_1} = a_2 c_1 c_2 - d_6 (s_5 (s_1 s_4 - c_4 (c_1 s_2 s_3 - c_1 c_2 c_3)) + c_5 (c_1 c_2 s_3 + c_1 c_3 s_2)) + a_3 c_1 c_2 c_3 - a_3 c_1 s_2 s_3 \quad (16)$$

$$\frac{\partial Y}{\partial \theta_2} = d_6 (c_5 (s_1 s_2 s_3 - c_2 c_3 s_1) + c_4 s_5 (c_2 s_1 s_3 + c_3 s_1 s_2)) - a_2 s_1 s_2 - a_3 c_2 s_1 s_3 - a_3 c_3 s_1 s_2 \quad (17)$$

$$\frac{\partial Y}{\partial \theta_3} = d_6 (c_5 (s_1 s_2 s_3 - c_2 c_3 s_1) + c_4 s_5 (c_2 s_1 s_3 + c_3 s_1 s_2)) - a_3 c_2 s_1 s_3 - a_3 c_3 s_1 s_2 \quad (18)$$

$$\frac{\partial Y}{\partial \theta_4} = d_6 s_5 (c_1 c_4 - s_4 (s_1 s_2 s_3 - c_2 c_3 s_1)) \quad (19)$$

$$\frac{\partial Y}{\partial \theta_5} = d_6 (c_5 (c_1 s_4 + c_4 (s_1 s_2 s_3 - c_2 c_3 s_1)) + s_5 (c_2 s_1 s_3 + c_3 s_1 s_2)) \quad (20)$$

$$\frac{\partial Y}{\partial \theta_6} = 0 \quad (21)$$

$$\frac{\partial Z}{\partial \theta_1} = 0 \quad (22)$$

$$\frac{\partial Z}{\partial \theta_2} = (d_6 c_{23} s_{45})/2 - a_2 c_2 - a_3 c_{23} + d_6 s_{23} c_5 - (d_6 s_{4-5} c_{23})/2 \quad (23)$$

$$\frac{\partial Z}{\partial \theta_3} = d_6 s_{23} c_5 - a_3 c_{23} + d_6 c_{23} c_4 s_5 \quad (24)$$

$$\frac{\partial Z}{\partial \theta_4} = -d_6 s_{23} s_4 s_5 \quad (25)$$

$$\frac{\partial Z}{\partial \theta_5} = d_6 c_{23} s_5 + d_6 s_{23} c_4 c_5 \quad (26)$$

$$\frac{\partial Z}{\partial \theta_6} = 0 \quad (27)$$

A posição é dada por :

$$\theta_1 = 1.3258rad, \theta_2 = 0.7854rad, \theta_3 = 1.5708rad, \theta_4 = 2.3562rad, \theta_5 = 0rad, \theta_6 = -1.1781rad.$$

Manipulador cartesiano (PPPR):

$$\frac{\partial X}{\partial d_1} = 0 \quad (28)$$

$$\frac{\partial X}{\partial d_2} = 0 \quad (29)$$

$$\frac{\partial X}{\partial d_3} = -1 \quad (30)$$

$$\frac{\partial X}{\partial \theta_4} = 0 \quad (31)$$

$$\frac{\partial Y}{\partial d_1} = 0 \quad (32)$$

$$\frac{\partial Y}{\partial d_2} = -1 \quad (33)$$

$$\frac{\partial Y}{\partial d_3} = 0 \quad (34)$$

$$\frac{\partial Y}{\partial \theta_4} = 0 \quad (35)$$

$$\frac{\partial Z}{\partial d_1} = 1 \quad (36)$$

$$\frac{\partial Z}{\partial d_2} = 0 \quad (37)$$

$$\frac{\partial Z}{\partial d_3} = 0 \quad (38)$$

$$\frac{\partial Z}{\partial \theta_4} = 0 \quad (39)$$

A posição é dada por :

$$d_1 = 0m, d_2 = -1m, d_3 = -4m, \theta_4 = 0rad.$$

Assim calculando as incertezas temos que para o manipulador articulado:  $u_x = \pm 0.0241$ ,  $u_y = \pm 0.0961$ ,  $u_z = \pm 0.0260$ .

Manipulador cartesiano:  $u_x = \pm 0.002$ ,  $u_y = \pm 0.001$ ,  $u_z = \pm 0.001$ .

Assim podemos ver que para valores de incerteza de posicionamento dos elos pequeno, a incerteza total nas 3 direções cartesianas fica pequena no manipulador articulado e menor ainda no cartesiano.



## 5 Área máxima de trabalho e limitação de movimento das juntas

Todo manipulador tem um limite espacial de movimentação conhecido como área de trabalho. Nesta seção iremos mostrar esquematicamente quais os formatos das áreas de trabalho dos manipuladores escolhidos e de acordo com as dimensões e parâmetros definidos para este projeto, quais são as limitações de movimentos das juntas.

Nas Figuras 8 e 31 abaixo podemos ver a área de trabalho de um manipulador cartesiano e de um manipulador articulado. Em geral, são as juntas da base que definem a área quase toda a área de trabalho. Como no nosso caso os punhos são esférico e rotacional, essa área não será afetada em termos de limite, porém internamente haverá mais opções de movimentos para se atingir as mesmas posições.

**Workspace of articulated manipulator**

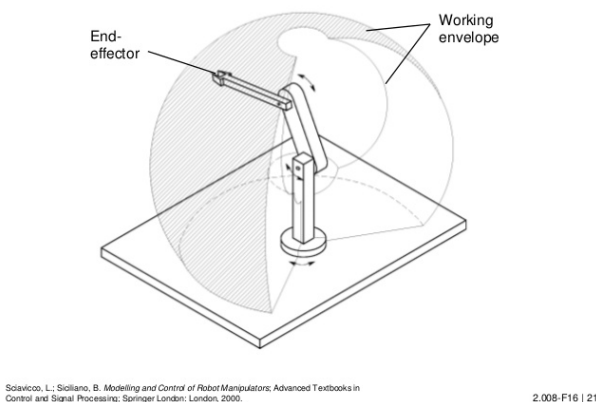


Figure 8: Área de trabalho de um manipulador articulado.

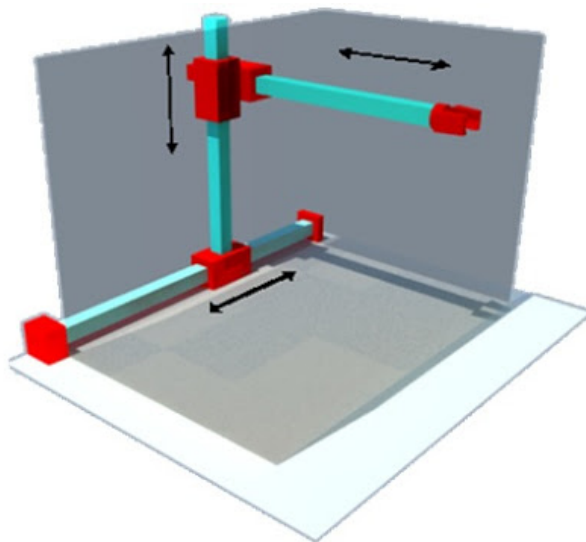


Figure 9: Área de trabalho de um robô cartesiano.

Como podemos ver dependendo de como foi projetado o manipulador ele pode ter limitações, às vezes, não esperadas à priori, como é o caso do manipulador articulado da Figura 8 que não tem o movimento de  $360^\circ$  na base.

No nosso caso, nós escolhemos que este o manipulador articulado terá um movimento de  $270^\circ$  (de  $-135^\circ$  a  $135^\circ$ ) em  $\theta_1$  e não terá outras limitações. Por outro lado o robô cartesiano terá apenas o movimento do

$\theta_4$  do punho limitado em  $180^\circ$  ( de  $-90$  a  $90$  graus).

Além disso, relembramos que as posições de singularidade calculadas anteriormente não são posições válidas para o robo articulado, não pertencendo portanto da área de trabalho.

## 5.1 Dimensionamento dos Robôs

A tarefa exige a pintura de uma área de comprimento 4m e altura 1.5m.

### 5.1.1 RRRRRR

Para o bom funcionamento deste robô decidimos que em seu ponto mínimo de altura o ângulo entre os braços do robô não deveria ser menor que  $30$  graus para evitar pontos em que os braços estejam na vertical, pois isto traria problemas tanto no aspecto de não existir margens de erro, assim como possíveis problemas de singularidade.

Esta distância é análoga à distância em vermelho mostrada abaixo para um robô similar ao tratado neste trabalho:

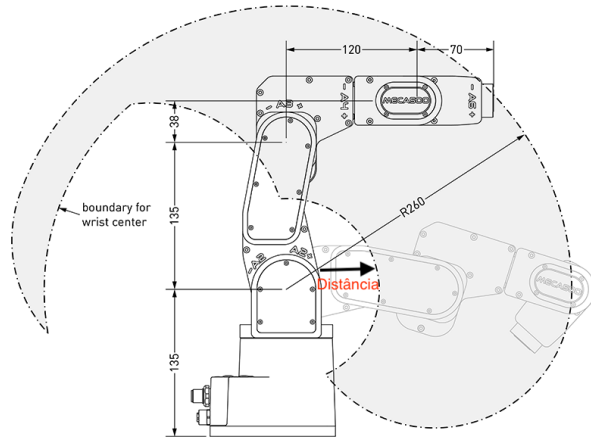


Figure 10: Distância mínima da base para área de trabalho.

O robô usado será posicionado imediatamente a frente de sua área de pintura. Sendo assim, ao pintar a parte mais baixa desta área obtemos a seguinte relação geométrica com a hipótese de braços de 2m de comprimento:

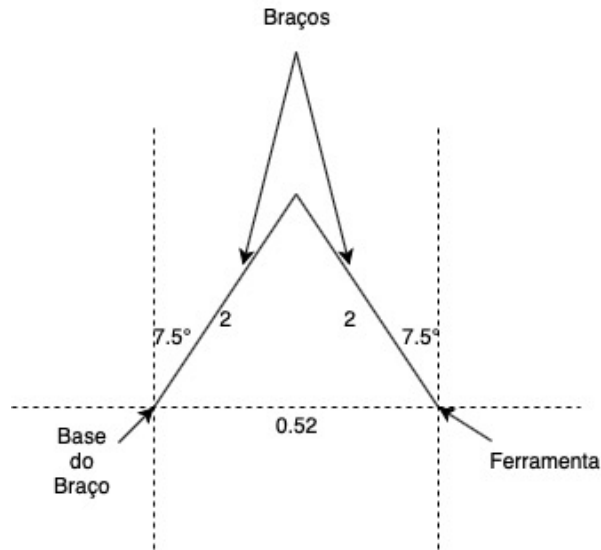


Figure 11: Distância mínimo entre robô e área de pintura.

Logo a área a ser pintada deve estar a uma distância mínima de 0.5m da base do robô, no caso **escolhemos 1.5m** para trabalharmos com os braços mais próximos da posição esticada.

Levando isto em conta, ao pintar as extremidades a distância entre a base e o ponto de pintura será de no máximo 2.91m. como visto abaixo. Desta forma os braços não ficarão totalmente "esticados" e são evitados problemas de singularidade.

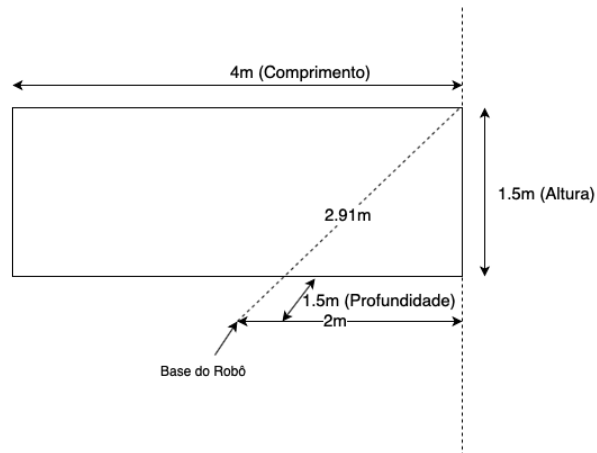


Figure 12: Distância máxima entre base do robô e área de pintura.

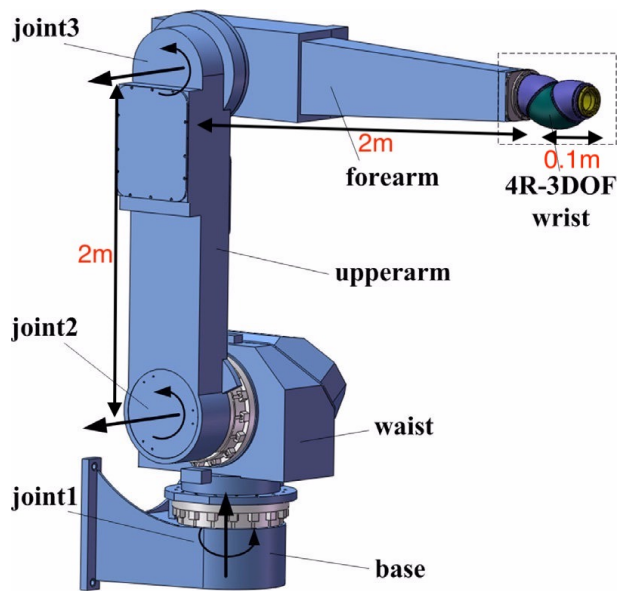


Figure 13: Dimensões do robô cartesiano.

Acima estão as dimensões finais do robô. É considerado que as distâncias entre a junta 1 e junta 2 são desprezíveis. Além disso é considerado que o punho esférico da ponta não apresenta distância entre suas juntas e que a distância entre o punho e o atuador ("pistola" de tinta) é 10cm.

### 5.1.2 PPPR

O robô cartesiano possui dimensões um pouco maiores que as mínimas necessárias para realizar a tarefa de modo a proporcionar certa margem de segurança: 1m a mais de comprimento e 0.5m a mais de altura.



Figure 14: Dimensões do robô cartesiano.

No manipulador cartesiano é considerado que a "pistola" de tinta está na mesma posição da junta rotacional, sem distância adicional.

## 6 Trajetória e Cinemática Inversa

Como discutido durante a introdução, nesta seção serão apresentados os pontos que definem a trajetória escolhida tal qual mostrado na Figure 15 e a cinemática inversa dos manipuladores que permitem transportar a trajetória do domínio cartesiano para o domínio de variáveis generalizadas das juntas.

A tabela a seguir mostra um conjunto de 20 pontos que serão utilizados para definir as quinas da trajetória. Entre eles será definida uma trajetória à partir de um método no qual a velocidade deverá ser constante. Nesta aplicação a velocidade constante é um aspecto chave para garantir uma pintura mais uniforme.

A seguir está a ilustração da trajetória como apresentada na introdução:

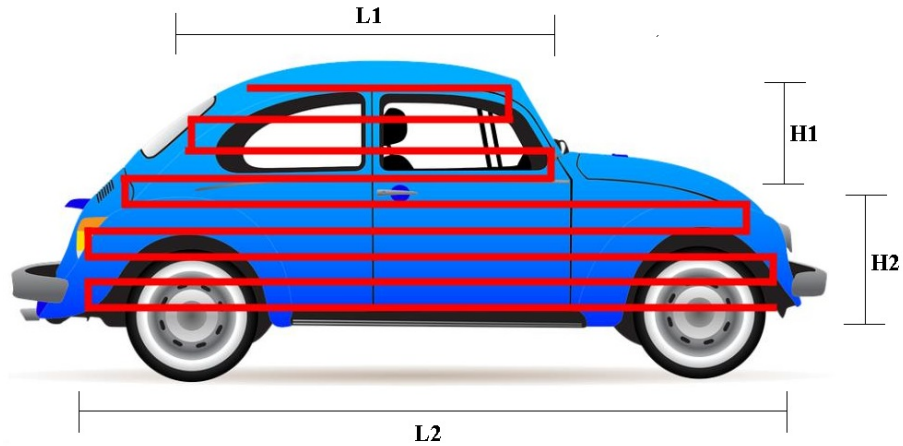


Figure 15: A curva em vermelho mostra a trajetória do robô para a pintura do veículo automotivo. O carro na imagem é meramente ilustrativo.

Abaixo estão os segmentos que aproximam essa trajetória.

Table 6: Constantes de dimensões utilizadas para o modelo.

Constantes	Valores
$(x_0, y_0)$	$(0, 0)$
$(x_1, y_1)$	$(L_2, 0)$
$(x_2, y_2)$	$(L_2, 0.2H_2)$
$(x_3, y_3)$	$(0, 0.2H_2)$
$(x_4, y_4)$	$(0, 0.4H_2)$
$(x_5, y_5)$	$(L_2, 0.4H_2)$
$(x_6, y_6)$	$(L_2, 0.6H_2)$
$(x_7, y_7)$	$(0, 0.6H_2)$
$(x_8, y_8)$	$(0, 0.8H_2)$
$(x_9, y_9)$	$(L_2, 0.8H_2)$
$(x_{10}, y_{10})$	$(L_2, H_2)$
$(x_{11}, y_{11})$	$(0, H_2)$
$(x_{12}, y_{12})$	$(0, H_2 + 0.33H_1)$
$(x_{13}, y_{13})$	$((L_2 + L_1)/2, H_2 + 0.33H_1)$
$(x_{14}, y_{14})$	$((L_2 + L_1)/2 - L_1, H_2 + 0.33H_1)$
$(x_{15}, y_{15})$	$((L_2 + L_1)/2, H_2 + 0.66H_1)$
$(x_{16}, y_{16})$	$((L_2 + L_1)/2 - L_1, H_2 + 0.66H_1)$
$(x_{17}, y_{17})$	$((L_2 + L_1)/2, H_2 + H_1)$
$(x_{18}, y_{18})$	$(0, H_2 + H_1)$
$(x_{19}, y_{19})$	$(0, 0)$

A seguir estão as constantes usadas.

Table 7: Constantes de dimensões utilizadas para o modelo.

Constantes	Valores
$L_1$	2.5
$H_1$	0.5
$L_2$	4
$H_2$	1

Observe que todas as posições na tabela 6 estão definidas em função das dimensões do modelo da carcaça para as quais os valores estão dados na tabela 7. Além disso, note que todas as trajetórias de um ponto a outro estão descritas no plano XY, logo o robô não se locomoverá no plano Z durante a tarefa. Dessa forma, tudo o que é preciso fazer é definir trajetórias para X e Y com velocidade constante entre cada par de pontos consecutivos. Para isso, o método de segmentos lineares com curvas parabólicas a seguir.

## 6.1 Linear Segments with Parabolic Blends

No método de mistura de segmentos lineares com parabólicos, nós tomamos uma coordenada de movimento que pode ser uma coordenada cartesiana (como é o nosso caso) ou uma coordenada generalizada e definimos que ela terá a forma de um polinômio do segundo grau. Assim a velocidade não será constante durante todo o movimento mas apenas após um período  $t_b$  de movimento parabólico e até  $t_f - t_b$  quando outro movimento parabólico levará à parada do movimento do robô na coordenada escolhida.

Esse método nos permite definir uma trajetória entre dois pontos de uma coordenada num dado

intervalo de tempo com velocidade de movimento constante. Assim nós podemos definir:

$$x(t) = a_0 + a_1 t + a_2 t^2 \quad (40)$$

$$\dot{x}(t) = a_1 + 2a_2 t \quad (41)$$

Como o movimento deve começar com velocidade 0 e terminar com velocidade 0, ou seja,  $x_0 =$  posição inicial em  $x$  e  $\dot{x}(0) = 0$ . Logo podemos concluir que:

$$a_0 = x_0 \quad (42)$$

$$a_1 = 0 \quad (43)$$

No tempo  $t_b$  ou "blend time" nós queremos que a velocidade atinja um valor  $V$  constante. Assim temos que:

$$\dot{x}(t_b) = 2a_2 t_b = V \quad (44)$$

O que nos leva a concluir que:

$$a_2 = \frac{V}{2t_b} \quad (45)$$

Assim a trajetória entre dois pontos entre 0 e  $t_b$  é dada pelas seguintes equações.

$$x(t) = x_0 + \frac{V}{2t_b} t^2 = x_0 + \frac{\alpha}{2} t^2 \quad (46)$$

$$\dot{x}(t) = \frac{V}{t_b} t = \alpha t \quad (47)$$

$$\ddot{x} = \frac{V}{t_b} = \alpha \quad (48)$$

Agora se analisarmos a trajetória entre  $t_b$  e  $t_f - t_b$  (parte linear do movimento), veremos que:

$$x(t) = x(t_b) + V(t - t_b) \quad (49)$$

Dada a simetria do movimento nós temos:

$$x\left(\frac{t_f}{2}\right) = \frac{x_0 + x_f}{2} \quad (50)$$

$$\frac{x_0 + x_f}{2} = x(t_b) + V\left(\frac{t_f}{2} - t_b\right) \quad (51)$$

$$x(t_b) = \frac{x_0 + x_f}{2} - V\left(\frac{t_f}{2} - t_b\right) \quad (52)$$

Uma vez que os dois segmentos, o parabólico e o linear, se misturam em  $t_b$  isso implica que:

$$x_0 + \frac{V}{2} t_b = \frac{x_0 + x_f - V t_f}{2} + V t_b \quad (53)$$

Assim resolvendo para o  $t_b$ , temos:

$$t_b = \frac{x_0 - x_f + V t_f}{V} \quad (54)$$

Porém para que tudo isso seja válido é preciso restringir o valor de  $t_b$  como  $0 < t_b \leq \frac{t_f}{2}$ . Essa restrição também pode ser lida como uma restrição ao valor de  $t_f$  ou  $V$  conforme mostrado nas equações abaixo.

$$\frac{x_f - x_0}{V} < t_f \leq \frac{2(x_f - x_0)}{V} \quad (55)$$

$$\frac{x_f - x_0}{t_f} < V \leq \frac{2(x_f - x_0)}{t_f} \quad (56)$$

Assim a posição será dada em três intervalos de tempo, o início do movimento, o movimento constante e o final do movimento, conforme as equações a seguir.

$$x(t) = x_0 + \frac{\alpha}{2}t^2 \quad \text{para} \quad 0 \leq t \leq t_b \quad (57)$$

$$x(t) = \frac{x_f + x_0 - Vt_f}{2} + Vt \quad \text{para} \quad t_b < t \leq t_f - t_b \quad (58)$$

$$x(t) = x_f - \frac{\alpha t_f^2}{2} + \alpha t_f t - \frac{\alpha}{2}t^2 \quad \text{para} \quad t_f - t_b < t \leq t_f \quad (59)$$

$$t_b = \frac{x_0 - x_f + Vt_f}{V} \quad (60)$$

O método acima pode ser usado para calcular a trajetória entre dois pontos  $x_0$  e  $x_f$  quaisquer numa dada coordenada. Logo ele pode ser utilizado tanto para  $x$  quando para  $y$  e à partir dele temos um método de encontrar os pontos da trajetória.

Na tabela abaixo se encontram os valores das constantes utilizadas neste projeto.

Table 8: Tempos de movimento e velocidades usadas nos modelos.

Constantes	Valores
$t_f$ em $x$ para a parte inferior da carroceria	10.05 s
$V_x$	0.4 m/s
$t_f$ em $y$ para a parte inferior da carroceria	0.55 s
$V_y$	0.4 m/s
$t_f$ em $y$ para a parte mediana-superior da carroceria	0.4668 s
$t_f$ em $x$ para a parte mediana-superior da carroceria	8.175 s
$t_f$ em $y$ para a retorno à posição inicial	3.8 s

Esses valores foram escolhidos para manter  $t_b$  e a velocidade  $V = 0.4m/s$  constantes.

## 6.2 Cinemática Inversa

A próxima etapa é a de definir a cinemática inversa para os robôs manipuladores escolhidos. Uma vez definida essa cinemática nós poderemos relacionar a trajetória em coordenadas cartesianas com a trajetória das coordenadas generalizadas formada pelas juntas dos robôs.

### 6.2.1 Robô cartesiano

Começaremos pelas cinemática inversa do braço robótico sem considerar o punho. No caso do robô cartesiano as coordenadas de generalizadas  $d_1, d_2, d_3$  e  $\theta_4$  são dados em função das posições escolhidas para a trajetória no plano cartesiano ( $x, y$  e  $z$ ).



Assim, como temos um punho rotacional no plano zy, temos que:

$$x(t) = d_2(t) \quad (61)$$

$$y(t) = -d_3(t) + d_4 \sin(\theta_4) \quad (62)$$

$$z(t) = d_1(t) + d_4 \cos(\theta_4) \quad (63)$$

Neste trabalho, nós decidimos por motivos de simplificação, escolher o valor de  $d_4$  como 0 e  $\theta_4 = 0$ . Além disso, como não há movimento em z,  $d_1$  será constante. Logo:

$$x(t) = d_2(t) \quad (64)$$

$$y(t) = -d_3(t) \quad (65)$$

$$z(t) = d_1(t) = \text{constante} \quad (66)$$

### 6.2.2 Robô articulado

Para o robô manipulador com punho esférico as equações que regem a cinemática inversa estão descritas abaixo de forma resumida, para mais detalhes verifique [1].

Vamos inicialmente falar dos três primeiros ângulos da base  $\theta_1, \theta_2$  e  $\theta_3$ . O primeiro ângulo é o mais simples.  $\theta_1$  pode ser dado em relação a  $x_c, y_c$  (posições finais do atuador) como:

$$\theta_1 = \text{Atan2}(x_c, y_c) \quad (67)$$

Duas outras distâncias podem ser definidas como  $r^2 = x_c^2 + y_c^2 - d^2$  e  $s = z_c - d_1$ . Onde d é a distância de offset (Figura 19) entre o primeiro braço e o segundo. Ele geralmente é adicionado para evitar singularidade de posição na linha vertical e centrada na base do manipulador. Além disso,  $d_1$  é uma distância vertical como mostrado na Figura 16 entre a primeira e a segunda junta que para nós foi definida como 0.

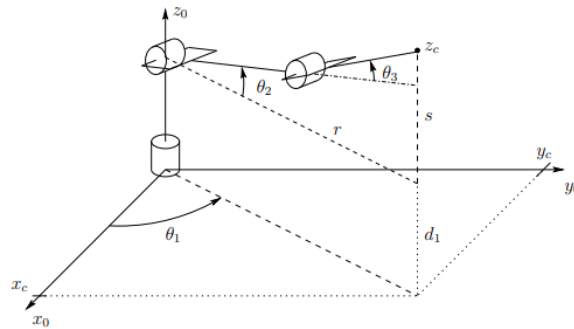


Figure 16: Representação das juntas da base do robô conforme dado em [1].

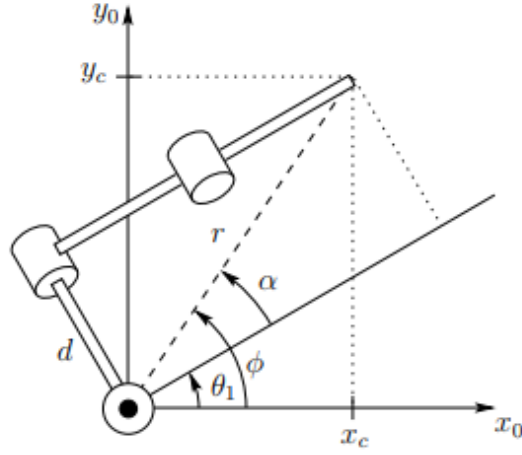


Figure 17: Representação das juntas da base do robô mais um offset "d" conforme dado em [1].

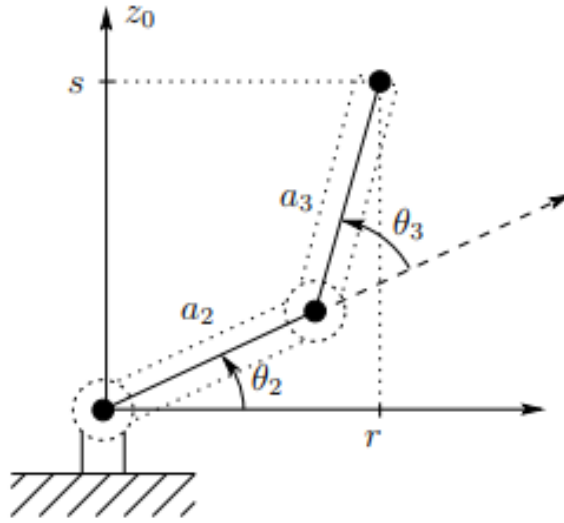


Figure 18: Projeções dos links 2 e 3 no plano.

Como mostrado na Figura 19 e usando lei dos cossenos, temos que o ângulo  $\theta_3$  é dado por.

$$\cos(\theta_3) = \frac{r^2 + s^2 - a_2^2 - a_3^2}{2a_2a_3} \quad (68)$$

$$\cos(\theta_3) = \frac{x_c^2 + y_c^2 - d^2 + (z_c - d_1)^2 - a_2^2 - a_3^2}{2a_2a_3} = D \quad (69)$$

Assim usando o  $\cos(\theta_3)$  definido como D conforme mostrado acima, temos que:

$$\theta_3 = \text{Atan2}(D, \sqrt{1 - D^2}) \quad (70)$$

E finalmente, olhando para a Figura 18 vemos que:

$$\theta_2 = \text{Atan2}(r, s) - \text{Atan2}(a_2 + a_3\cos\theta_3, a_3\sin\theta_3) \quad (71)$$

$$\theta_2 = \text{Atan2}(\sqrt{x_c^2 + y_c^2 - d^2}, z_c - d_1) - \text{Atan2}(a_2 + a_3 c_3, a_3 s_3) \quad (72)$$

Assim terminamos de definir a cinemática inversa para a base do robô articulado.

Em seguida precisamos definir a cinemática inversa do punho esférico desse mesmo robô. Para isso usaremos as matrizes de rotação. Sendo  $R_3^0$  dada por  $A_1 A_2 A_3$  onde as matrizes "A" são as matrizes de rotação relativas de uma junta à outra entre as 3 primeiras juntas. Considerando também  $R_6^3$  a matriz de rotação entre as juntas do punho conforme mostrado abaixo, temos que:

$$R_3^0 = \begin{pmatrix} c_1 c_{23} & -c_1 s_{23} & s_1 \\ s_1 c_{23} & -s_1 s_{23} & -c_1 \\ s_{23} & c_{23} & 0 \end{pmatrix}$$

$$R_6^3 = \begin{pmatrix} c_4 c_5 c_6 - s_4 s_6 & -c_4 c_5 s_6 - s_4 c_6 & c_4 s_5 \\ s_4 c_5 c_6 + c_4 s_6 & -s_4 c_5 s_6 + c_4 c_6 & s_4 s_5 \\ -s_5 c_6 & s_5 s_6 & 0 \end{pmatrix}$$

$$R_6^3 = (R_3^0)^T R \quad (73)$$

Assim, como conhecemos os valores dos 3 primeiros ângulos da cinemática inversa calculada acima e como conhecemos R (matriz de rotação do atuador entre os referenciais inerciais e do atuador) podemos usar os ângulos de Euler para calcular os ângulos das três juntas finais dados da matriz  $R_6^3$ . Assim igualando os três elementos da última coluna da matriz  $R_6^3$ , nós temos :

$$c_4 s_5 = c_1 c_{23} r_{13} + s_1 c_{23} r_{23} + s_{23} r_{33} \quad (74)$$

$$s_4 s_5 = -c_1 s_{23} r_{13} - s_1 c_{23} r_{23} + c_{23} r_{33} \quad (75)$$

$$c_5 = s_1 r_{13} - c_1 r_{23} \quad (76)$$

Assim usando a transformação dos ângulos de Euler achamos que:

$$\theta_4 = \text{Atan2}(c_1 c_{23} r_{13} + s_1 c_{23} r_{23} + s_{23} r_{33}, -c_1 c_{23} r_{13} - s_1 c_{23} r_{23} + c_{23} r_{33}) \quad (77)$$

$$\theta_5 = \text{Atan2}(s_1 r_{13} - c_1 r_{23}, \pm \sqrt{1 - (s_1 r_{13} - c_1 r_{23})^2}) \quad (78)$$

$$\theta_6 = \text{Atan2}(-s_1 r_{11} + c_1 r_{21}, s_1 r_{12} - c_1 r_{22}) \quad (79)$$

Finalmente, reunindo todos os resultados, temos que a cinemática inversa do manipulado articulado é dada pelas seguintes equações.

$$\theta_1 = \text{Atan2}(x_c, y_c) \quad (80)$$

$$\theta_2 = \text{Atan2}(\sqrt{x_c^2 + y_c^2 - d^2}, z_c - d_1) - \text{Atan2}(a_2 + a_3 c_3, a_3 s_3) \quad (81)$$

Aqui escolhemos  $\sqrt{1 - D^2}$  em vez de  $-\sqrt{1 - D^2}$  para termos a configuração do braço para cima.

$$\theta_3 = \text{Atan2}(D, \sqrt{1 - D^2}) \quad (82)$$

$$D = \frac{x_c^2 + y_c^2 - d^2 + (z_c - d_1)^2 - a_2^2 - a_3^2}{2a_2a_3} \quad (83)$$

$$\theta_4 = \text{Atan2}(c_1c_{23}r_{13} + s_1c_{23}r_{23} + s_{23}r_{33}, -c_1c_{23}r_{13} - s_1c_{23}r_{23} + c_{23}r_{33}) \quad (84)$$

$$\theta_5 = \text{Atan2}(s_1r_{13} - c_1r_{23}, \pm\sqrt{1 - (s_1r_{13} - c_1r_{23})^2}) \quad (85)$$

$$\theta_6 = \text{Atan2}(-s_1r_{11} + c_1r_{21}, s_1r_{12} - c_1r_{22}) \quad (86)$$

É preciso lembrar que no nosso caso o valor de  $d_1 = 0$  e  $d = 0$ . Assim à partir dessas relações de cinemática inversa é possível calcular tanto a trajetória dos robôs no plano cartesiano como em termo de variáveis generalizadas de Denavit Hartenberg. Assim terminamos o estudo da cinemática dos manipuladores, dessa forma as próximas seções serão dedicadas ao estudo da dinâmica.

## 6.3 Variáveis das juntas em função do tempo

### 6.3.1 PPPR

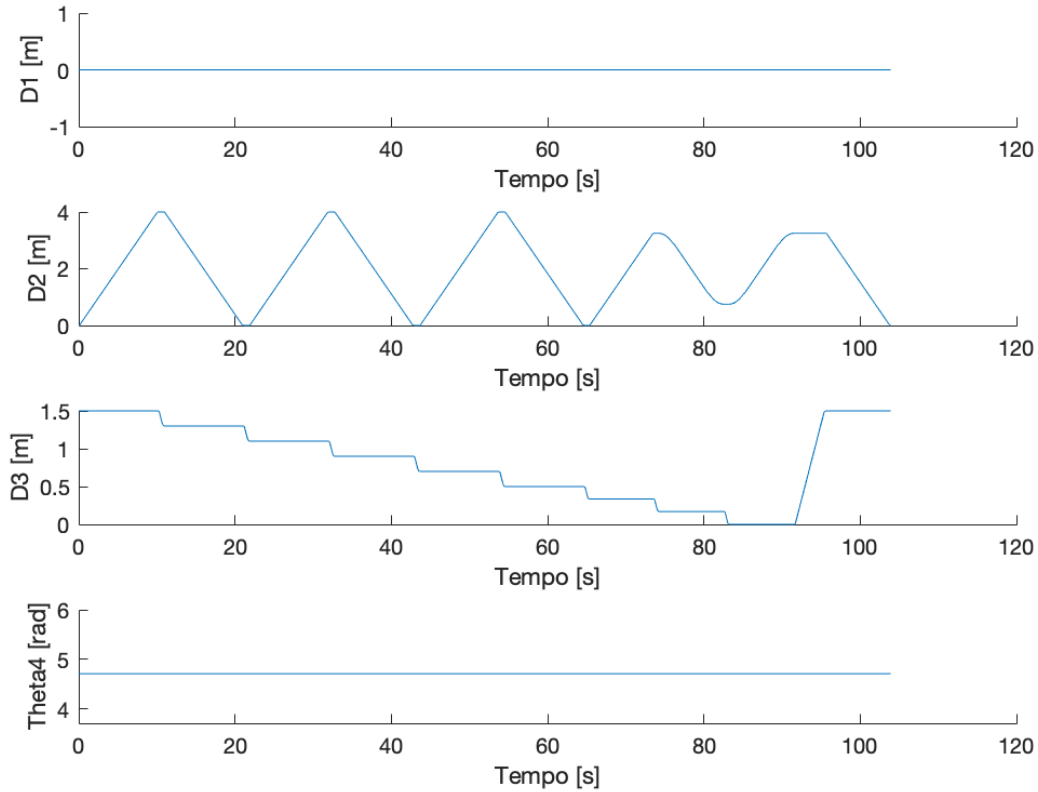


Figure 19: Valor das variáveis das juntas com o tempo.

### 6.3.2 RRRRRR

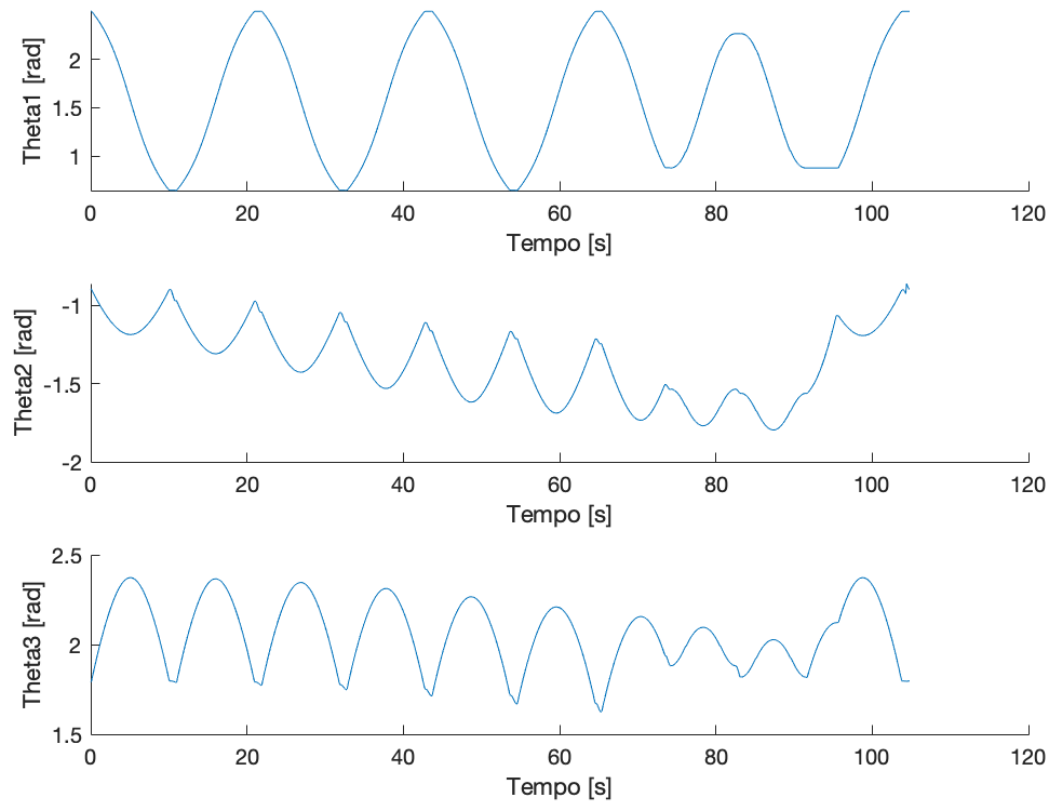


Figure 20: Valor das variáveis das juntas 1 a 3 com o tempo.

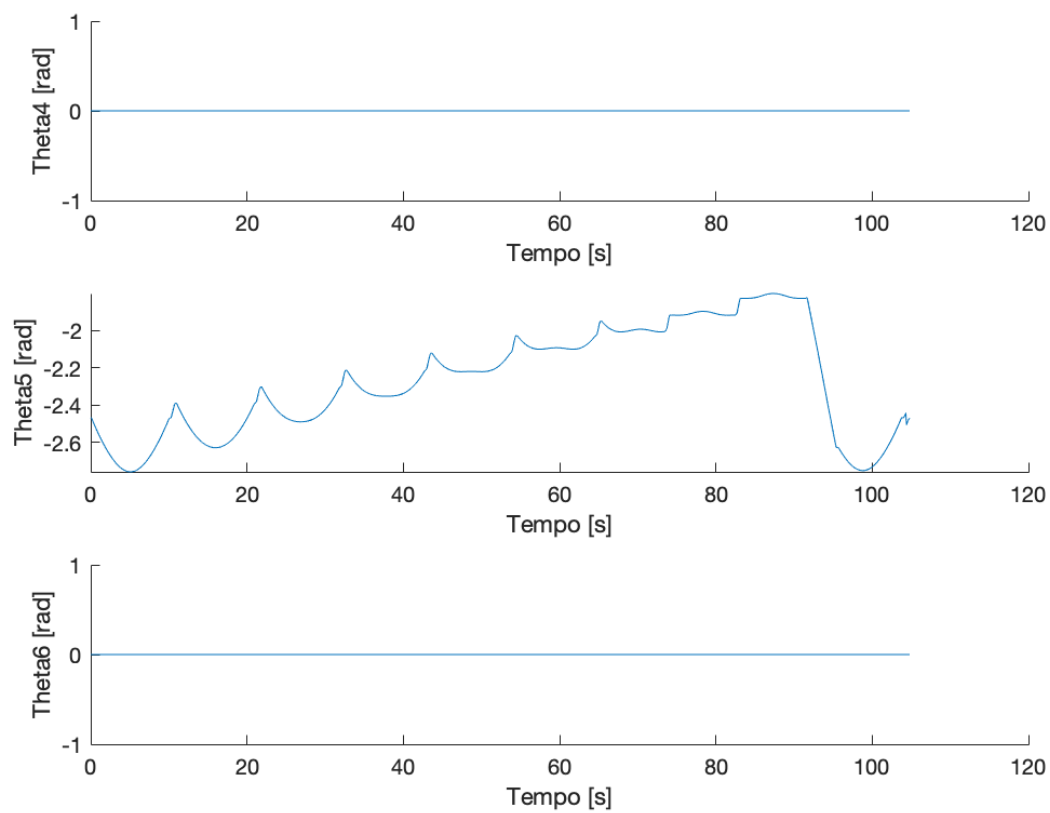


Figure 21: Valor das variáveis das juntas 4 a 6 com o tempo.

## 7 Modelo Dinâmico

A equação de Euler-Lagrange para n graus de liberdade é definida como:

$$\frac{d}{dt} \frac{\delta L}{\delta \dot{q}_i} - \frac{\delta L}{\delta q_i} = \tau \quad (87)$$

Sendo:

$$L = K - P \quad (88)$$

P é a energia potencial

K é a energia cinética

Ao reorganizar os termos da equação, o modelo dinâmico matricial pode ser visto como:

$$D(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) = \tau \quad (89)$$

D é matriz de Inercia. Esta é obtida através da matriz energia cinética por:

$$D = \sum_{i=1}^n (m_i J_{vi}(q)^T J_{vi}(q) + J_{wi}(q)^T R_i(q) I_i R_i(q)^T J_{wi}(q)) \quad (90)$$

C é a matriz de Christoffel, a qual é obtida através de:

$$C = \sum_{i=1}^n \frac{1}{2} \left( \frac{\delta d_{kj}}{\delta q_i} + \frac{\delta d_{ki}}{\delta q_j} - \frac{\delta d_{ij}}{\delta q_k} \right) \quad (91)$$

A matriz g é a matriz de gravidade que vem de P por meio de:

$$g = \sum_{i=1}^n \frac{\delta P}{\delta q_i} \quad (92)$$

### 7.1 Energia cinética

A matriz de energia cinética vem da matriz D por meio de:

$$K = \frac{1}{2} \dot{q}^T D(q) \dot{q} \quad (93)$$

### 7.2 RRRRRR

Visto que K é dependente da matriz D, a qual tem modelagem matemática suficientemente grande, optamos por não colocar no relatório, mas pode ser visto a partir da simulação.

### 7.3 PPPR

$$K = (\dot{d}_1^2 (m_1 + m_2 + m_3 + m_4))/2 + (\dot{d}_3^2 (m_3 + m_4))/2 + (I_{yy4} \dot{\theta}_4^2)/2 + (\dot{d}_2^2 (m_2 + m_3 + m_4))/2$$

### 7.4 Energia potencial

A matriz Pi vem do elemento na posição da linha 3, coluna 4 das matrizes homogêneas em relação ao referencial dividido por dois, pois consideramos todas as barras do mesmo material e uniformes

$$\sum_{i=1}^n P_i \quad (94)$$

#### 7.4.1 RRRRRR

$$P1 = 0$$

$$P2 = -a2gm2s_2$$

$$P3 = -gm3(a3s_{23} + a2s_2)$$

$$P4 = -gm4(a3s_{23} + a2s_2)$$

$$P5 = -gm5(a3s_{23} + a2s_2)$$

$$P6 = -gm6(a3s_{23} + a2s_2 - (d6s_{23}s_{45})/2 + d6c_{23}c_5 + (d6\sin(th4 - th5)s_{23})/2)$$

$$P = -gm3(a3s_{23} + a2s_2) - gm4(a3s_{23} + a2s_2) - gm5(a3s_{23} + a2s_2) - gm6(a3s_{23} + a2s_2 - (d6s_{23}s_{45})/2 + d6c_{23}c_5 + (d6\sin(th4 - th5)s_{23})/2) - a2gm2s_2$$

#### 7.4.2 PPPR

$$P1 = 2 * gm1$$

$$P2 = 2 * gm2$$

$$P3 = (2 - d3)gm3$$

$$P3 = (2 - d3)gm4$$

$$P = d1g(m1 + m2 + m3 + m4)$$

### 7.5 Equação de movimento

Como já mencionado, equação de movimento é:

$$D(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) = \tau \quad (95)$$

Para encontrar para os dois casos as matrizes se tornaram demasiadamente grande que não foi possível visualizar a integridade dos valores no matlab, o que pode ser visto na seção 12 com a simulação dos códigos

#### 7.5.1 RRRRRR

Matriz D

Para o robô RRRRRR temos:

$$D = \begin{pmatrix} d11 & d12 & d13 & d14 & d15 & d16 \\ d21 & d22 & d23 & d24 & d25 & d26 \\ d31 & d32 & d33 & d34 & d35 & d36 \\ d41 & d42 & d43 & d44 & d45 & d46 \\ d51 & d52 & d53 & d54 & d55 & d56 \\ d61 & d62 & d63 & d64 & d65 & d66 \end{pmatrix}$$



## Matriz C

Para valores de i que vão de 1 a 6 temos:

$$C_i = \begin{pmatrix} ci11 & ci12 & ci13 & ci14 & ci15 & ci16 \\ ci21 & ci22 & ci23 & ci24 & ci25 & ci26 \\ ci31 & ci32 & ci33 & ci34 & ci35 & ci36 \\ ci41 & ci42 & ci43 & ci44 & ci45 & ci46 \\ ci51 & ci52 & ci53 & ci54 & ci55 & ci56 \\ ci61 & ci62 & ci63 & ci64 & ci65 & ci66 \end{pmatrix}$$

Sendo:

$$C = C1 \cdot \dot{\theta}1 + C2 \cdot \dot{\theta}2 + C3 \cdot \dot{\theta}3 + C4 \cdot \dot{\theta}4 + C5 \cdot \dot{\theta}5 + C6 \cdot \dot{\theta}6 \quad (96)$$

## Matriz g

A matriz de gravidade g é dada como:

$$g = \begin{pmatrix} g1 \\ g2 \\ g3 \\ g4 \\ g5 \\ g6 \end{pmatrix}$$

$$g1 = 0$$

$$g2 = -gm3(a3c_{23} + a2c_2) - gm4(a3c_{23} + a2c_2) - gm5(a3c_{23} + a2c_2) - gm6(a3c_{23} + a2c_2 - (d6c_{23}s_{45})/2 - d6s_{23}c_5 + (d6\sin(th4 - th5)c_{23})/2) - a2gm2c_2$$

$$g3 = gm6(a3c_{23} - (d6c_{23}s_{45})/2 - d6s_{23}c_5 + (d6\sin(th4 - th5)c_{23})/2) - a3gm3c_{23} - a3gm4c_{23} - a3gm5c_{23}$$

$$g4 = gm6((d6\cos(th4 + th5)s_{23})/2 - (d6\cos(th4 - th5)s_{23})/2)$$

$$g5 = gm6((d6\cos(th4 + th5)s_{23})/2 + d6c_{23}s_5 + (d6\cos(th4 - th5)s_{23})/2)$$

$$g6 = 0$$

### 7.5.2 PPR

Para o robo PPR temos:

## Matriz D

$$D = \begin{pmatrix} d11 & d12 & d13 & d1 \\ d21 & d22 & d23 & d24 \\ d31 & d32 & d33 & d34 \\ d41 & d42 & d43 & d44 \end{pmatrix}$$

$$d11 = m1 + m2 + m3 + m4$$

$$d21 = 0$$

$$d31 = 0$$

$$d41 = 0$$

$$d12 = 0$$

$$d22 = m2 + m3 + m4$$

$$d32 = 0$$

$$d42 = 0$$

$$d13 = 0$$

$$d23 = 0$$

$$d33 = m3 + m4$$

$$d43 = 0$$

$$d14 = 0$$

$$d24 = 0$$

$$d34 = 0$$

$$d44 = I_{zz4} \cos^2 \theta_4 + I_{xx4} \sin^2 \theta_4$$

## Matriz C

Para valores de i que vão de 1 a 4 temos:

$$Ci = \begin{pmatrix} ci11 & ci12 & ci13 & ci1 \\ ci21 & ci22 & ci23 & ci24 \\ ci31 & ci32 & ci33 & ci34 \\ ci41 & ci42 & ci43 & ci44 \end{pmatrix}$$

Sendo:

$$C = C1 \cdot \dot{\theta}1 + C2 \cdot \dot{\theta}2 + C3 \cdot \dot{\theta}3 + C4 \cdot \dot{\theta}4 \quad (97)$$

$$C11 = 0$$

$$C21 = 0$$

$$C_{31} = 0$$

$$C_{41} = 0$$

$$C_{12} = 0$$

$$C_{22} = 0$$

$$C_{32} = 0$$

$$C_{42} = 0$$

$$C_{13} = 0$$

$$C_{23} = 0$$

$$C_{33} = 0$$

$$C_{43} = 0$$

$$C_{14} = 0$$

$$C_{24} = 0$$

$$C_{34} = 0$$

$$C_{44} = \dot{\theta}_4 \sin(2\theta_4) (I_{xx4}/2 - I_{zz4}/2)$$

### **Matriz g**

A matriz de gravidade g é dada como:

$$g = \begin{pmatrix} g1 \\ g2 \\ g3 \\ g4 \end{pmatrix}$$

$$g1 = 0$$

$$g2 = 0$$

$$g3 = -g(m3 + m4)$$

$$g4 = 0$$

## 8 Estratégia de Controle

### 8.1 PPPR

O modelo dinâmico do PPPR é simples o suficiente para podermos levar em conta sua dinâmica no controle do robô, pois ele é um modelo linear.

O sistema usado está resumido abaixo.

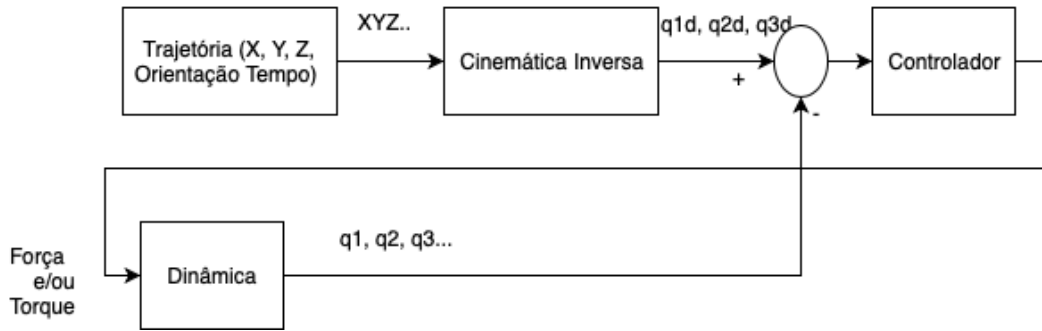


Figure 22: Resumo da estratégia de controle.

A partir das equações da trajetória foram obtidos os pontos desejados em função do tempo. Cada um destes pontos da trajetória foram então convertidos para variáveis das juntas, D1, D2, D3 e Theta4 utilizando as equações de cinemática inversa. Essas variáveis mudam de valor com o tempo de forma a compor a trajetória.

O Controlador recebe tanto as variáveis atuais, quanto as desejadas e então envia Torque/Forças para os motores e atuadores lineares.

Estes Torque/Forças são então convertidos em novas variáveis das juntas pelo bloco da dinâmica.

Neste bloco a velocidade é integrada numericamente gerando mudanças na posição e a aceleração é integrada gerando mudanças na velocidade.

As novas variáveis das juntas então realimentam o controlador.

O output do controlador é limitado à 600N conforme datasheet do atuador de referência:

Specifications	
Travel Range	400 mm, 600 mm, 800 mm, 1000 mm
Resolution	0.4 $\mu$ m (@ 125 microsteps), 2.5 $\mu$ m (servo motor)
Encoder	Optical rotary encoder mounted to read of motor, 4000 CPR with index
Lead Screw Pitch	10 mm
Accuracy	0.0006 mm/mm of travel
Max. Speed	100 mm/sec (stepper motor), 100 mm/sec (servo motor), 100 mm/sec (MDrive motor)
Maximum Load	68 kg (150 lb.)
Stage Weight	DS-400: 6 kg (13 lb.)   DS-600: 7 kg (15.5 lb.)   DS-800: 8.5 kg (18 lb.)   DS-1000: 9.5 kg (21 lb.)
Material	Aluminum Alloy Construction
Finish	Black Anodize
Operating Temperature	0°C to 50°C

Figure 23: Datasheet do atuador tomado como referência.

Apesar dele suportar até 68KgF, ou 667.08N, o valor conservador de 600N foi utilizado como máximo. A implementação do controlador em si está mostrada abaixo:

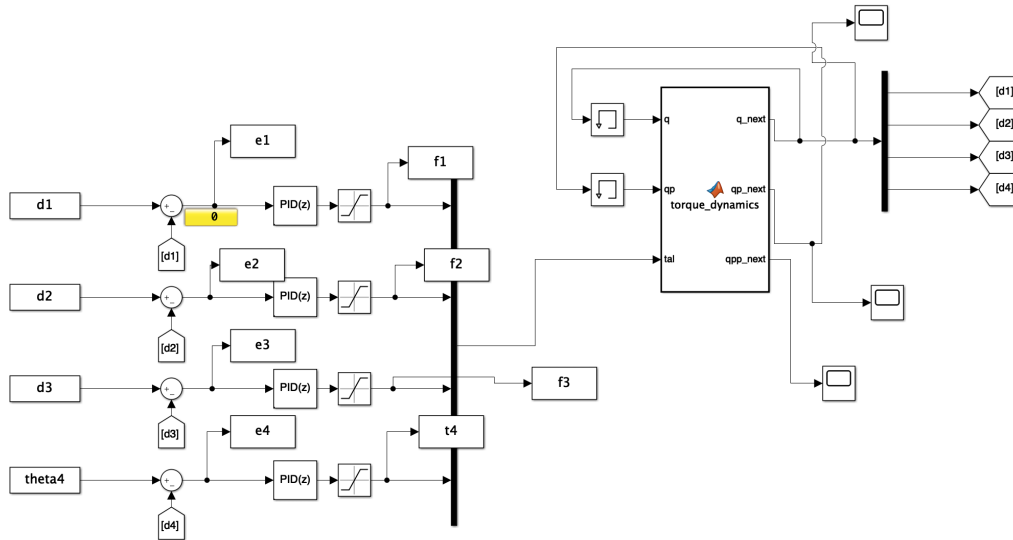


Figure 24: Implementação Simulink.

Os controladores escolhidos foram PD com os seguinte ganhos:

- D1 -  $P = 10$  ;  $D = 100$
- D2 -  $P = 5000$  ;  $D = 3000$
- D3 -  $P = 6000$  ;  $D = 3000$
- Theta4 -  $P = 50$  ;  $D = 10$

Theta4 utiliza o controlador para chegar a seu ângulo constante desde o repouso.

Abaixo são mostrados os esforços de controle em cada junta, assim como os erros de controle.

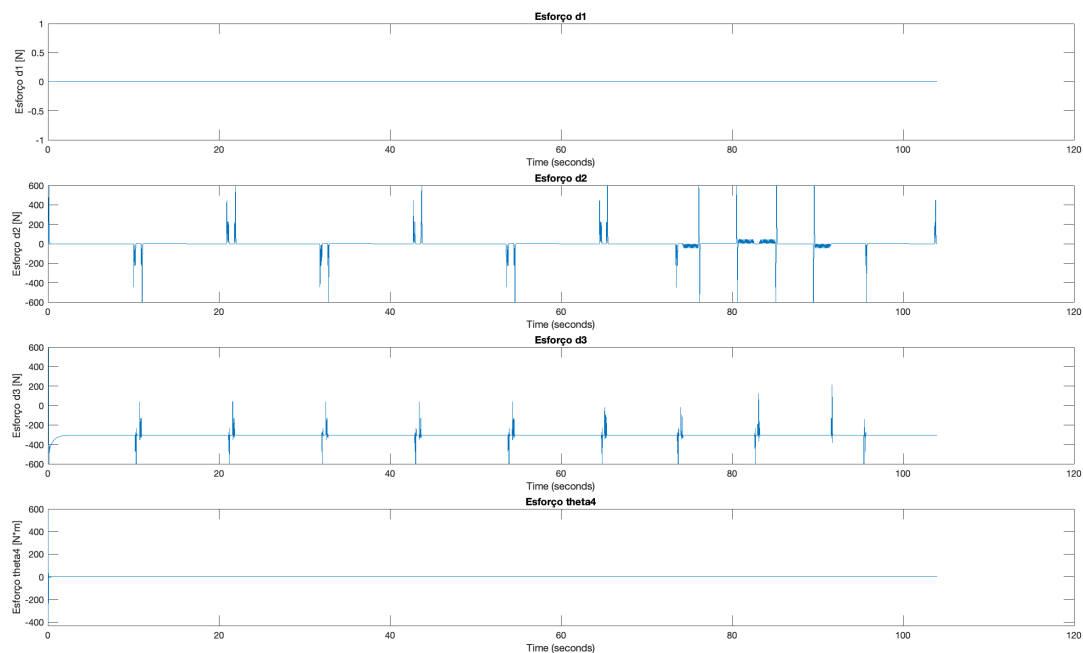


Figure 25: Esforço resultante nas juntas.

Como visto a seguir os erros são bem pequenos e o controle cumpre sua função de maneira muito satisfatória.

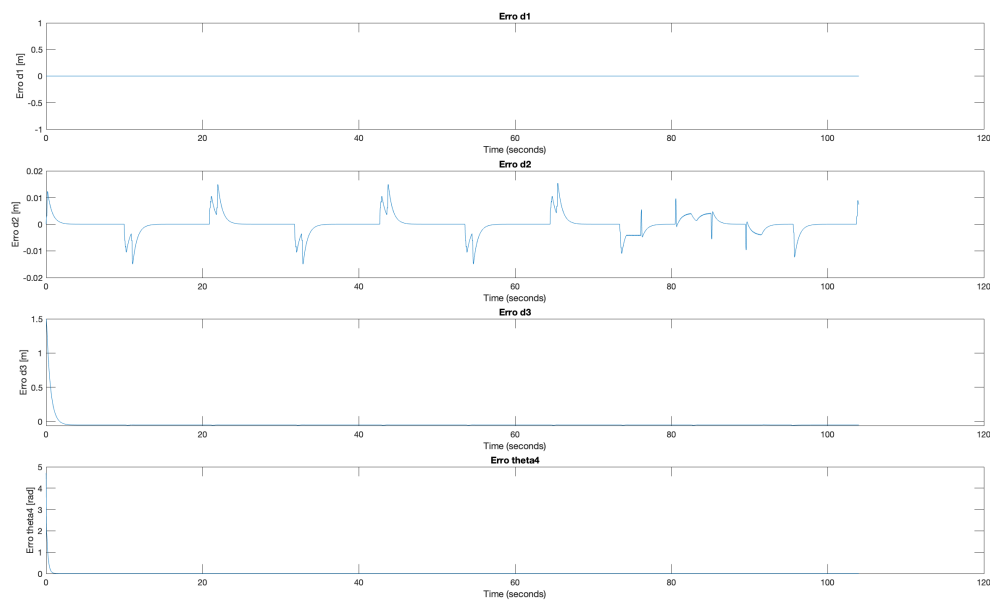


Figure 26: Erro de controle.

## 8.2 RRRRRR

Inicialmente tentamos realizar a estratégia de controle anterior, mas percebemos que não obteríamos resultados adequados uma vez que o modelo dinâmico do manipulador RRRRRR e portanto sua planta são não lineares.

Desta forma optamos pelo controle **Feed-Forward** uma vez que a trajetória já era conhecida.

A configuração deste controle é mostrado a seguir.

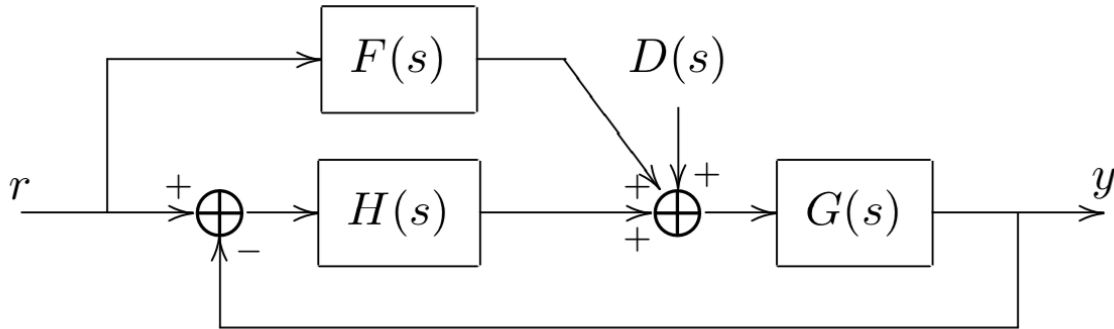


Figure 27: Controle Feed-forward

Escolhemos  $F$  como  $1/G$ , isto corresponde, fisicamente, à calcular o torque necessário para cada junta usando as equações dinâmicas do modelo conforme será feito na seção ***"Esforços resultantes nas juntas"***.

Como a saída de  $F$  já são os torques ideais, a função  $H(s)$  que é o controlador fica responsável por compensar os distúrbios. Porém neste trabalho os distúrbios foram considerados nulos e portanto o  $H$  é unitário. Porém, na prática ele seria um controlador PID.

É importante salientar que este método é trabalhoso e depende muito do conhecimento da planta para o projeto de  $F$ .

## 9 Esforços resultantes nas juntas

Para calcular os esforços nas juntas precisamos estimar a massa do robô.

As partes móveis são compostas basicamente de atuadores lineares. Tomamos como base o modelo a seguir que pode ser visto no link <https://www.newmarksystems.com/linear-positioners/d-slide-linear-slide/> :



Figure 28: Atuador linear tomado como base.



Figure 29: Robo PPR.

Para a primeira junta que corresponde ao movimento de até 1m, atrelamos um atuador de massa 9.5Kg, então estimamos a massa total dela em 15Kg.

Para a segunda junta que corresponde ao movimento de até 5m, atrelamos um atuador de massa 50Kg, então estimamos a massa total dela em 70Kg.

Para a terceira junta que corresponde ao movimento de até 2m, atrelamos um atuador de massa 20Kg, então estimamos a massa total dela em 30Kg.

A inércia rotacional desta junta foi considerada  $0.15Kg * m^2$  apesar de não mudar os resultados, pois esta junta não é usada na trajetória.

Para a quarta junta que corresponde ao movimento de giro, atrelamos uma massa de 1Kg, pois ela será acionada por um servo motor MG959 .



Usando a equação do modelo dinâmico (eq. 55), podemos obter os esforços nas juntas em função das suas posições, velocidades e acelerações.

Para o Robô PPPR os esforços foram obtidos a partir do controle na secção anterior.

Sabemos que:

$$\dot{x}(t) = \alpha_x t \quad e \quad \ddot{x}(t) = \alpha_x \quad para \quad 0 \leq t \leq t_b \quad (98)$$

$$\dot{x}(t) = V_x \quad e \quad \ddot{x}(t) = 0 \quad para \quad t_b < t \leq t_f - t_b \quad (99)$$

$$\dot{x}(t) = \alpha_x(t_f - t) \quad e \quad \ddot{x}(t) = -\alpha_x \quad para \quad t_f - t_b < t \leq t_f \quad (100)$$

e:

$$\dot{y}(t) = \alpha_y t \quad e \quad \ddot{y}(t) = \alpha_y \quad para \quad 0 \leq t \leq t_b \quad (101)$$

$$\dot{y}(t) = V_y \quad e \quad \ddot{y}(t) = 0 \quad para \quad t_b < t \leq t_f - t_b \quad (102)$$

$$\dot{y}(t) = \alpha_y(t_f - t) \quad e \quad \ddot{y}(t) = -\alpha_y \quad para \quad t_f - t_b < t \leq t_f \quad (103)$$

## 9.1 PPPR

Aplicando a equação do modelo dinâmico para as matrizes D, C e g encontradas na secção 7.5.2, temos que:

$$\tau = \begin{pmatrix} \ddot{d}_1(m1 + m2 + m3 + m4) \\ \ddot{d}_2(m2 + m3 + m4) \\ (\ddot{d}_3 - g)(m3 + m4) \\ \sin(2\theta_4)(I_{xx4}/2 - I_{zz4}/2)\dot{\theta}_4^2 + \ddot{\theta}_4(I_{zz4}\cos(\theta_4)^2 + I_{xx4} * \sin(\theta_4)^2) \end{pmatrix} \quad (104)$$

E a partir das trajetórias e da cinemática inversa calculadas na secção 6, derivando q(t) duas vezes podemos achar  $\ddot{q}(t)$ . Não temos movimento em z ( $d_1 = constante$ ) e  $\theta_4 = 0$ , logo  $\dot{q}_1 = \dot{q}_4 = \ddot{q}_1 = \ddot{q}_4 = 0$ .

$$\ddot{q} = \begin{pmatrix} \ddot{d}_1 \\ \ddot{d}_2 \\ \ddot{d}_3 \\ \ddot{\theta}_4 \end{pmatrix} = \begin{pmatrix} 0 \\ \ddot{x}(t) \\ -\ddot{y}(t) \\ 0 \end{pmatrix}$$

Aplicando na equação 104:

$$\tau = \begin{pmatrix} 0 \\ \ddot{x}(t)(m2 + m3 + m4) \\ -(\ddot{y}(t) + g)(m3 + m4) \\ 0 \end{pmatrix}$$

A figura 30 abaixo mostram a variação teórica de  $\tau_2$  e  $\tau_3$  no tempo. Os gráfico dos esforços nas juntas 1 e 4 não serão plotados pois são sempre zero.

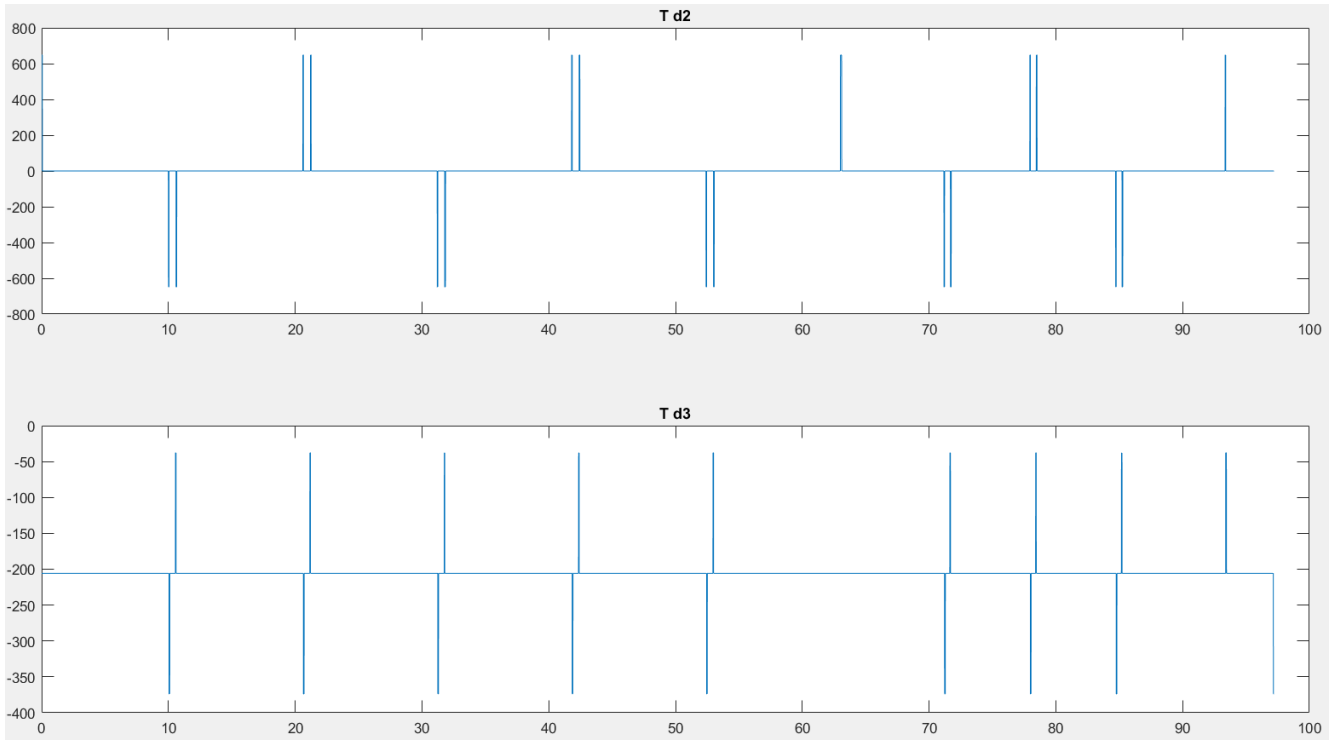


Figure 30: Esforços nas juntas 2 e 3 em função do tempo para o robô cartesiano.

## 9.2 RRRRRR

Para o manipulador articulado RRRRRR, a matriz de esforços é muito grande e não poderia ser exibida de uma forma compreensível nesse relatório.

Logo, somente o gráfico da evolução dos esforços será mostrado ao longo do tempo. O método utilizado é o mesmo feito para o manipulador cartesiano PPPR apresentado anteriormente. A figura 31 mostra o gráfico dos esforços em cada junta em função do tempo. As massas consideradas foram  $m1 = 135,21kg$ ,  $m2 = m3 = 270,42kg$  e  $m4 = m5 = m6 = 1,13kg$ . Todos os corpos foram considerados cilíndricos com 10cm de raio com comprimentos  $l1 = 1m$ ,  $l2 = l3 = 2m$  e  $l4 = l5 = l6 = 0.07m$ ;

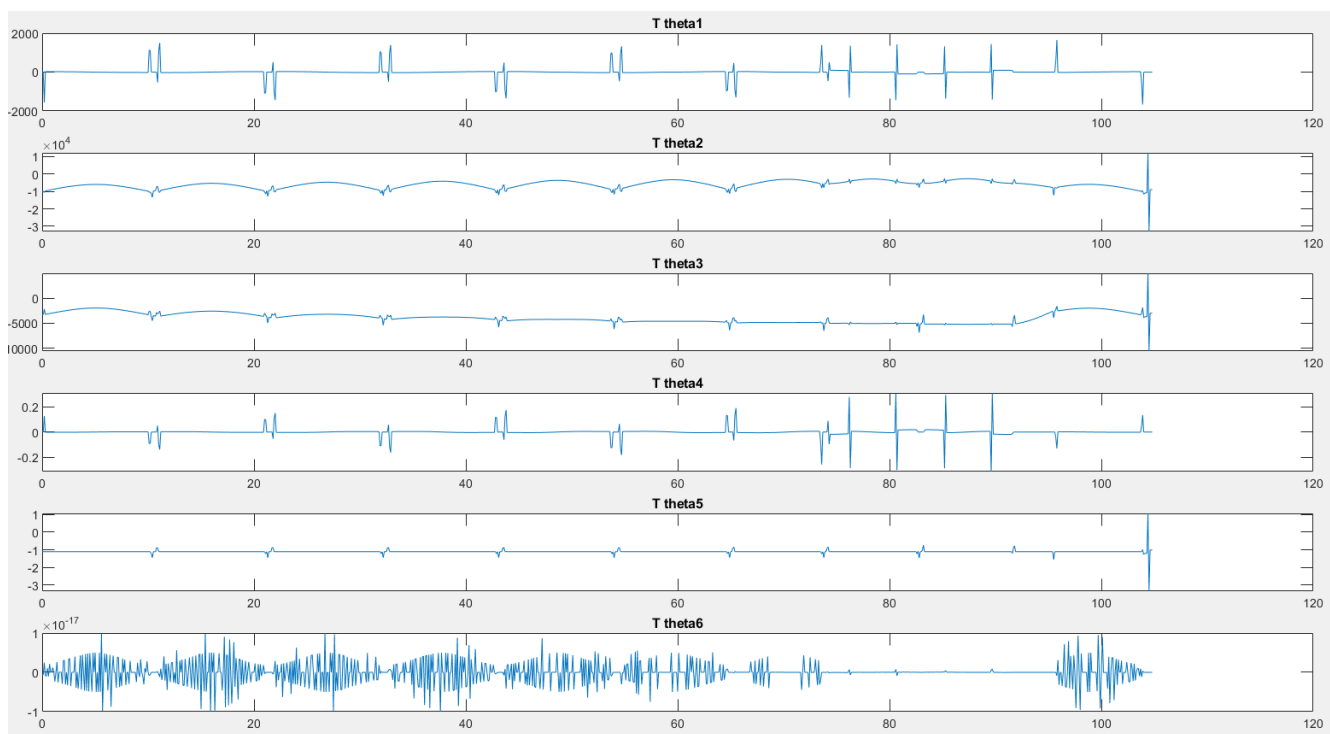


Figure 31: Esforços nas juntas 1 a 6 em função do tempo para o robô articulado.

## 10 Vantagens e desvantagens de cada modelo

### 10.1 Velocidade de execução da tarefa

Como a cinemática inversa do robô PPPR é muito mais simples que a do RRRRRR, computacionalmente se consegue velocidades maiores no PPPR.

Além disso, conforme será visto a seguir o robô articulado na prática sobre esforços em todas as juntas, pois todas precisam se movimentar para chegar na posição e orientação necessária para executar a tarefa e compensar a força gravitacional. Logo a com o robô PPPR se obtém velocidades maiores.

### 10.2 Esforços envolvidos

Para o modelo PPPR apenas duas juntas tem esforços não nulos visto que somente as duas precisam ser acionadas para a realização da tarefa desejada. Já para o modelo RRRRRR, todas as 6 juntas sofrem esforços por menor que sejam. Além disso, os esforços para o manipulador cartesiano são sempre pontuais e alternados para cada tipo de trajetória. Cada junta executa um tipo de trajetória, ao passo que o manipulador articulado tem esforços quase que constantemente sendo modificados independentemente da direção da trajetória.

O modelo RRRRRR se torna muito mais custoso em termos de esforços pois deve lidar com seis esforços diferentes a cada instante, ao passo que o PRRR só lida com um por vez. Isso torna o controle do manipulador articulado bem mais complexo do que aquele necessário para o cartesiano, o que faz com que o manipulador PRRR seja bem mais adequado para a tarefa em questão.

### 10.3 Presença de singularidades

Enquanto o manipulador RRRRRR apresenta três conjuntos de suas variáveis controláveis como pontos de singularidade (inclusive configurações de "braços para cima" e "braços para baixo"), o manipulador PPPR com junta rotacional estática não apresenta singularidades, podendo alcançar qualquer ponto em seu espaço de trabalho sem limitações.

Do ponto de vista de presença de singularidades, o manipulador PPPR se mostra mais confiável se aplicado em tarefas que possa executar. O robô de seis juntas rotacionais, por outro lado, apesar de gerar pontos de singularidade, pode executar tarefas que o PPPR não conseguiria, como por exemplo alcançar com a garra o interior de estruturas.

### 10.4 Incerteza de posicionamento da ferramenta

Pôde-se perceber na seção 4 que as incertezas de posicionamento usando o manipulador RRRRRR foi da ordem de 10 vezes maior do que usando o manipulador PPPR. Isso mostra que apesar de ter limitações de aplicação, o robô PPPR tem posicionamento com menos incertezas, então para tarefas que possa realizar se mostrará uma opção mais robusta e confiável.

### 10.5 Erros de controle

O controlador PID se mostrou uma alternativa confiável e precisa para o manipulador PPPR, enquanto para o manipulador RRRRRR apresentou erros expressivos. Isso se deve ao fato de que o robô PPPR apresenta dinâmica linear, enquanto o robô RRRRRR apresenta alta não linearidade em sua dinâmica.

A literatura apresenta propostas de utilização de PID em contextos não lineares, seja com estratégias de controle diferentes, seja com alternativas de separação do sistema em diversos controles diferentes com ajuste de ganhos diferentes.

Para o caso RRRRRR, uma modelagem de controle por estados com linearização em torno da trajetória de referência poderia ter se mostrado mais vantajosa, mas dado o tempo de projeto, não foi possível chegar a uma estratégia consistente para o caso.

## 10.6 Problemas encontrados

O maior problema encontrado foi conseguir controlar com PID o robô RRRRRR, algo que não foi possível devido a não linearidade dele.

## 11 Escolha do manipulador

Após analisar todas as características dos dois manipuladores para a realização desta tarefa percebe-se que o robô cartesiano é a melhor escolha. Ele é muito mais simples que o robô articulado tanto em termos de cinemática, como de dinâmica e controle.

Além disso, dado que a tarefa é apenas um movimento em um plano cartesiano, a trajetória se torna muito fácil de ser implementada nesse tipo de robô, por isso não se justifica a utilização de um complexo manipulador articulado para esta tarefa.

Caso a tarefa fosse mais complexa, como por exemplo a pintura de partes internas da carroceria como a porta por exemplo, uma pintura mais precisa de partes curvas, ou qualquer tarefa onde a garra devesse alcançar áreas internas de uma estrutura, passando por alguns obstáculos, teria sido inviável o uso de uma alternativa simples como a do manipulador PPPR e teríamos que usar o RRRRRR.

Em algumas aplicações específicas, o movimento seria mais complexo e exigiria maior flexibilidade do robô, então o mais indicado seria o manipulador articulado visto a sua área de trabalho, sua precisão e sua grande liberdade de movimentos.

De toda forma, para a aplicação proposta no contexto deste projeto, pelos motivos explicitados, o uso do manipulador PPPR é certamente a melhor escolha.

## 12 Códigos

### 12.1 Singularidades

#### 12.1.1 PPPR

```
1 %%Singularidades
2
3 J = [J1,J2,J3,J4];
4 % J = J(1:4,1:4);
5
6 % J = J([1,2,5,6],:);
7
8 J11 = J([1,2],[1,2]);
9
10 J12 = J([1,2],[3,4]);
11
12 J21 = J([3,4],[1,2]);
13
14 J22 = J([3,4],[3,4]);
15
16 % simplify(det(J11))
17 if det(J11)~=0
18 % [J11_sing_theta1, J11_sing_theta2, J11_sing_theta3, J11_sing_theta4, ...
19 % J11_sing_theta5, J11_sing_theta6]
20 S11 = solve (simplify(det(J11))==0, d1, d2, d3, theta4);
21
22 SingularitiesJ11_PPPR = [eval(S11.d1), eval(S11.d2), eval(S11.d3), eval(S11.
    theta4)];
23
24 end
25
26 % simplify(det(J12))
27 if det(J12)~=0
28 % [J12_sing_theta1, J12_sing_theta2, J12_sing_theta3, J12_sing_theta4, ...
29 % J12_sing_theta5, J12_sing_theta6]
30 S12 = solve (simplify(det(J12))==0,d1, d2, d3, theta4);
31
32 SingularitiesJ12_PPPR = [eval(S12.d1), eval(S12.d2), eval(S12.d3), eval(S12.
    theta4)];
33
34 end
35
36 % simplify(det(J21))
37 if det(J21)~=0
38 % [J21_sing_theta1, J21_sing_theta2, J21_sing_theta3, J21_sing_theta4, ...
39 % J21_sing_theta5, J21_sing_theta6]
40 S21 = solve (simplify(det(J21))==0, d1, d2, d3, theta4);
41
42 SingularitiesJ21_PPPR = [eval(S21.d1), eval(S21.d2), eval(S21.d3),eval(S21.
```

```

        theta4)];
43  end
44
45  % simplify(det(J22))
46  if det(J22)~=0
47  % [J22_sing_theta1, J22_sing_theta2, J22_sing_theta3, J22_sing_theta4, ...
48  %     J22_sing_theta5, J22_sing_theta6]
49  S22 = solve (simplify(det(J22))==0, d1, d2, d3, theta4);
50
51  SingularitiesJ22_PPPR = [eval(S22.d1), eval(S22.d2), eval(S22.d3),eval(S22.
        theta4)];
52  end
53
54  %%

```



### 12.1.2 RRRRRR

```
1  %% Singularidades
2
3  J = [J1,J2,J3,J4,J5,J6];
4
5  J11 = [J1v, J2v, J3v];
6
7  J12 = [J4v, J5v, J6v];
8
9  J21 = [J1w, J2w, J3w];
10
11 J22 = [J4w, J5w, J6w];
12
13 % simplify(det(J11))
14 if det(J11)~=0
15 % [J11_sing_th, J11_sing_th2, J11_sing_th3, J11_sing_th4, ...
16 %     J11_sing_th5, J11_sing_th6]
17 S11 = solve (simplify(det(J11))==0, ...
18     th, th2, th3, th4, th5, th6);
19
20 SingularitiesJ11_RRRRRR = [eval(S11.th), eval(S11.th2), eval(S11.th3), ...
21     eval(S11.th4), eval(S11.th5), eval(S11.th6)];
22
23 end
24
25 % simplify(det(J12))
26 if det(J12)~=0
27 % [J12_sing_th, J12_sing_th2, J12_sing_th3, J12_sing_th4, ...
28 %     J12_sing_th5, J12_sing_th6]
29 S12 = solve (simplify(det(J12))==0, ...
30     th, th2, th3, th4, th5, th6);
31
32 SingularitiesJ12_RRRRRR = [eval(S12.th), eval(S12.th2), eval(S12.th3), ...
33     eval(S12.th4), eval(S12.th5), eval(S12.th6)];
34
35 end
36
37 % simplify(det(J21))
38 if det(J21)~=0
39 % [J21_sing_th, J21_sing_th2, J21_sing_th3, J21_sing_th4, ...
40 %     J21_sing_th5, J21_sing_th6]
41 S21 = solve (simplify(det(J21))==0, ...
42     th, th2, th3, th4, th5, th6);
43
44 SingularitiesJ21_RRRRRR = [eval(S21.th), eval(S21.th2), eval(S21.th3), ...
45     eval(S21.th4), eval(S21.th5), eval(S21.th6)];
46 end
47
48 % simplify(det(J22))
```

```

49  if det(J22)~=0
50  % [J22_sing_tH, J22_sing_th2, J22_sing_th3, J22_sing_th4, ...
51  %      J22_sing_th5, J22_sing_th6]
52  S22 = solve (simplify(det(J22))==0, ...
53      tH, th2, th3, th4, th5, th6);
54
55  SingularitiesJ22_RRRRRR = [eval(S22.tH), eval(S22.th2), eval(S22.th3), ...
56      eval(S22.th4), eval(S22.th5), eval(S22.th6)];
57  end
58
59  %%

```

## 12.2 Dinâmica

### 12.2.1 PPPR

```
1 clear all;
2
3 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Projeto Rob?tica %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
4
5 %% Dados do Robo
6     syms theta1 d1 a1 alpha1
7     syms theta2 d2 a2 alpha2
8     syms theta3 d3 a3 alpha3
9     syms theta4 d4 a4 alpha4
10
11
12 %% Cinematica Direta
13
14 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Denavit Hatenberg %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
15 %Variaveis
16     %todos os thetas
17 DH =[0 d1 0 pi/2;
18     +pi/2 d2 0 -pi/2;
19     0 d3 0 pi/2;
20     theta4 0 0 -pi/2];
21
22
23 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Encontro dos A %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
24 for i = 1:4
25
26     Rotz = [cos(DH(i,1)) -sin(DH(i,1)) 0 0;
27             sin(DH(i,1)) cos(DH(i,1)) 0 0;
28             0 0 1 0;
29             0 0 0 1];
30
31     Transd =[ 1 0 0 0;
32              0 1 0 0;
33              0 0 1 DH(i,2);
34              0 0 0 1];
35
36     Transa =[ 1 0 0 DH(i,3);
37              0 1 0 0;
38              0 0 1 0;
39              0 0 0 1];
40
41     Rotx = [1 0 0 0;
42             0 cos(DH(i,4)) -sin(DH(i,4)) 0;
43             0 sin(DH(i,4)) cos(DH(i,4)) 0;
44             0 0 0 1];
45
46
```

```

47
48
49     M = (Rotz)*(Transd)*(Transa)*(Rotx);
50
51     if i ==1
52         A_1 = M;
53     elseif i ==2
54         A_2 = M;
55     elseif i ==3
56         A_3 = M;
57     elseif i ==4
58         A_4 = M;
59     end
60
61 end
62
63
64
65 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Encontro da matriz H em rela??p ao referencial
66 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
67 H_01 = A_1;
68 H_02 = A_1*A_2;
69 H_03 = A_1*A_2*A_3;
70 H_04 = A_1*A_2*A_3*A_4;
71
72 H_01 = simplify(H_01);
73 H_02 = simplify(H_02);
74 H_03 = simplify(H_03);
75 H_04 = simplify(H_04);
76
77 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Encontro da matriz R de rota??o %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
78 R1 = H_01(1:3,1:3);
79 R2 = H_02(1:3,1:3);
80 R3 = H_03(1:3,1:3);
81 R4 = H_04(1:3,1:3);
82
83 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Encontro da matriz o e z %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
84 z0 = [0 0 1]';
85 o0 = [0 0 0]';
86 z1 = H_01(1:3,3);
87 o1 = H_01(1:3,4);
88 z2 = H_02(1:3,3);
89 o2 = H_02(1:3,4);
90 z3 = H_03(1:3,3);
91 o3 = H_03(1:3,4);
92 z4 = H_04(1:3,3);
93 o4 = H_04(1:3,4);
94
95 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Encontro da matriz J %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

96
97 %Velocidade Linear
98 J1v = z0;
99 % simplify(J1v);
100 J2v = z1;
101 J2v = simplify(J2v);
102 J3v = z2;
103 J3v= simplify(J3v);
104 J4v = cross(z3,(o4-o3));
105 J4v = simplify(J4v);
106
107
108 % Velocidade Angular
109 zero3 = o0;
110 J1w = zero3;
111 J2w = zero3;
112 J3w = zero3;
113 J4w = z3;
114
115
116 % Matriz completa
117 J1 = [J1v;J1w];
118 J2 = [J2v;J2w];
119 J3 = [J3v;J3w];
120 J4 = [J4v;J4w];
121
122
123
124 J = [J1,J2,J3,J4];
125
126 %% Dinamica
127 %D(q)q'' + C(q,q')q' + g(q) = tal
128
129 % Variaveis
130 syms q1 q2 q3 q4 q5 q6
131 q1 = d1;
132 q2 = d2;
133 q3 = d3;
134 q4 = theta4;
135 q = [q1,q2,q3,q4];
136
137 syms d1_dot d2_dot d3_dot theta4_dot
138
139 q1_dot = d1_dot;
140 q2_dot = d2_dot;
141 q3_dot = d3_dot;
142 q4_dot = theta4_dot;
143
144 q_dot = [q1_dot;q2_dot;q3_dot;q4_dot];
145

```

```

146 %Valores I hipoteticos
147 syms Ixx1 Iyy1 Izz1
148 syms Ixx2 Iyy2 Izz2
149 syms Ixx3 Iyy3 Izz3
150 syms Ixx4 Iyy4 Izz4
151
152
153 %Massa elos
154 syms m1 m2 m3 m4
155
156
157 %Colocando o J no formato do exercicio
158 %
159 J1vd = [ J1v , zero3 , zero3 , zero3 ];
160 J2vd = [ J1v , J2v , zero3 , zero3 ];
161 J3vd = [ J1v , J2v , J3v , zero3 ];
162 J4vd = [ J1v , J2v , J3v , J4v ];
163
164 J1wd = [ J1w , zero3 , zero3 , zero3 ];
165 J2wd = [ J1w , J2w , zero3 , zero3 ];
166 J3wd = [ J1w , J2w , J3w , zero3 ];
167 J4wd = [ J1w , J2w , J3w , J4w ];
168
169 %Colocando o I no formato do exercicio
170 I1 = [ Ixx1 0 0;
171        0 Iyy1 0;
172        0 0 Izz1 ];
173
174 I2 = [ Ixx2 0 0;
175        0 Iyy2 0;
176        0 0 Izz2 ];
177
178 I3 = [ Ixx3 0 0;
179        0 Iyy3 0;
180        0 0 Izz3 ];
181
182 I4 = [ Ixx4 0 0;
183        0 Iyy4 0;
184        0 0 Izz4 ];
185
186
187 %% Calculo da Equa??o Dinamica
188
189 % Calculo da Matriz D
190
191 %D = m[Jv]'[Jv]+[Jw]'[R][I][R]'[Jw]
192
193 D1 = m1*(transpose(J1vd))*J1vd + transpose(J1wd)*(R1*I1*transpose(R1))*
    J1wd;
194 D1 = simplify(D1);

```

```

195 D2 = m2*(transpose(J2vd))*J2vd + transpose(J2wd)*(R2*I2*transpose(R2))*
    J2wd;
196 D2 = simplify(D2);
197 D3 = m3*(transpose(J3vd))*J3vd + transpose(J3wd)*(R3*I3*transpose(R3))*
    J3wd;
198 D3 = simplify(D3);
199 D4 = m4*(transpose(J4vd))*J4vd + transpose(J4wd)*(R4*I4*transpose(R4))*
    J4wd;
200 D4 = simplify(D4);
201
202
203 D = D1+D2+D3+D4;
204 D = simplify(D);
205
206
207 %Calculo da Matriz C
208 %P=diff(f,x);
209 %subs(P,{x,y},{1.5,2});
210
211 C = zeros(4,4,4,'sym');
212 for i=1:4
213     for j=1:4
214         for k=1:4
215
216             %meio = 0.5
217             C(i,j,k) = 0.5*diff(D(k,j),q(i))+ diff(D(k,i),q(j)) - diff(D(i,j),q(
                k));
218
219         end
220     end
221 end
222
223 C1 = [C(1,1,1) C(1,2,1) C(1,3,1) C(1,4,1);
224       C(1,1,2) C(1,2,2) C(1,3,2) C(1,4,2);
225       C(1,1,3) C(1,2,3) C(1,3,3) C(1,4,3);
226       C(1,1,4) C(1,2,4) C(1,3,4) C(1,4,4)];
227 C1 = simplify(C1);
228
229
230 C2 = [C(2,1,1) C(2,2,1) C(2,3,1) C(2,4,1) ;
231       C(2,1,2) C(2,2,2) C(2,3,2) C(2,4,2) ;
232       C(2,1,3) C(2,2,3) C(2,3,3) C(2,4,3) ;
233       C(2,1,4) C(2,2,4) C(2,3,4) C(2,4,4) ] ;
234 C2 = simplify(C2);
235
236
237 C3 = [C(3,1,1) C(3,2,1) C(3,3,1) C(3,4,1) ;
238       C(3,1,2) C(3,2,2) C(3,3,2) C(3,4,2) ;
239       C(3,1,3) C(3,2,3) C(3,3,3) C(3,4,3) ;
240       C(3,1,4) C(3,2,4) C(3,3,4) C(3,4,4)] ;

```

```

241 C3 = simplify(C3);
242
243 C4 =[C(4,1,1) C(4,2,1) C(4,3,1) C(4,4,1);
244       C(4,1,2) C(4,2,2) C(4,3,2) C(4,4,2);
245       C(4,1,3) C(4,2,3) C(4,3,3) C(4,4,3);
246       C(4,1,4) C(4,2,4) C(4,3,4) C(4,4,4) ];
247 C4 = simplify(C4);
248
249
250 C = C1*q1_dot + C2*q2_dot + C3*q3_dot + C4*q4_dot;
251 C = simplify(C);
252
253 %% Calculo da Energia Cinetica E Potencial
254 %%Calculo da Energia Cinetica
255
256 K = 0.5*transpose(q_dot)*D*q_dot;
257 K = simplify(K);
258
259 %Energia Potencia
260 syms g% acelera o da gravidade
261 P1 = H_01(3,4)*m1*g;
262 P2 = H_02(3,4)*m2*g;
263 P3= H_03(3,4)*m3*g;
264 P4= H_04(3,4)*m4*g;
265 P = P1+P2+P3+P4;
266 P = simplify(P);
267
268 %Encontro da Matriz g(q)
269 g1 = diff(P,q(1));
270 g2 = diff(P,q(2));
271 g3 = diff(P,q(3));
272 g4 = diff(P,q(4));
273
274 G = [g1;g2;g3;g4];
275
276 %Tals Simbolicos-> Substituir por valores Reais
277 %syms Tal1 Tal2 Tal3 Tal4
278 %Tal = [Tal1;Tal2;Tal3;Tal4];
279
280 syms d1_dot2 d2_dot2 d3_dot2 theta4_dot2
281
282 q1_dot2 = d1_dot2;
283 q2_dot2 = d2_dot2;
284 q3_dot2 = d3_dot2;
285 q4_dot2 = theta4_dot2;
286
287 q_dot2 = [q1_dot2;q2_dot2;q3_dot2;q4_dot2];
288
289 q = transpose(q);
290

```



```

291
292
293     Tau = D*q_dot2+C*q_dot+G;
294
295
296
297
298
299     %Equa??o Geral para encontrar a acelera??o
300     %q_dot2 = D\(TaI-C*q_dot-g);

```

### 12.2.2 RRRRRR

```

1  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Projeto Rob tica %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2
3  %% Dados do Robo
4
5      syms tH d1 A  alphA
6      syms th2 d2 a2  alpha2
7      syms th3 d3 a3  alpha3
8      syms th4 d4 a4  alpha4
9      syms th5 d5 a5  alpha5
10     syms th6 d6 a6  alpha6
11
12  %% Cinematica Direta
13
14  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Denavit Hatenberg %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
15
16
17  %Variaveis
18      %todos os thetas
19  DH =[tH      0      0      -pi/2;
20        th2      0      a2      0;
21        th3      0      a3      -pi/2;
22        th4      0      0      pi/2;
23        th5      0      0      -pi/2;
24        th6      d6      0      0];
25
26
27  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Encontro dos A %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
28  for i = 1:6
29      Rotz = [cos(DH(i,1)) -sin(DH(i,1)) 0 0;
30              sin(DH(i,1))  cos(DH(i,1)) 0 0;
31              0              0              1 0;
32              0              0              0 1];
33
34      Transd =[ 1 0 0 0;
35                0 1 0 0;
36                0 0 1 DH(i,2);
37                0 0 0 1];
38
39      Transa =[ 1 0 0 DH(i,3);
40                0 1 0 0;
41                0 0 1 0;
42                0 0 0 1];
43
44      Rotx = [1              0              0              0;
45              0              cos(DH(i,4)) -sin(DH(i,4)) 0;
46              0              sin(DH(i,4))  cos(DH(i,4)) 0;
47              0              0              0              1];
48

```

```

49
50
51
52     M = (Rotz)*(Transd)*(Transa)*(Rotx);
53
54         if i ==1
55             A_1 = M;
56         elseif i ==2
57             A_2 = M;
58         elseif i ==3
59             A_3 = M;
60         elseif i ==4
61             A_4 = M;
62         elseif i ==5
63             A_5 = M;
64         elseif i ==6
65             A_6 = M;
66         end
67
68     end
69
70     %%%%%%%%%%%%%%% Encontro da matriz A %%%%%%%%%%%%%%%
71     A_1 = simplify(A_1);
72     A_2 = simplify(A_2);
73     A_3 = simplify(A_3);
74     A_4 = simplify(A_4);
75     A_5 = simplify(A_5);
76     A_6 = simplify(A_6);
77
78     %%%%%%%%%%%%%%% Encontro da matriz H em rela  p ao referencial
79     %%%%%%%%%%%%%%%
80     H_01 = A_1;
81     H_02 = A_1*A_2;
82     H_03 = A_1*A_2*A_3;
83     H_04 = A_1*A_2*A_3*A_4;
84     H_05 = A_1*A_2*A_3*A_4*A_5;
85     H_06 = A_1*A_2*A_3*A_4*A_5*A_6;
86
87     H_01 = simplify(H_01);
88     H_02 = simplify(H_02);
89     H_03 = simplify(H_03);
90     H_04 = simplify(H_04);
91     H_05 = simplify(H_05);
92     H_06 = simplify(H_06);
93
94     %%%%%%%%%%%%%%% Encontro da matriz de rota  o R %%%%%%%%%%%%%%%
95
96     R1 = H_01(1:3,1:3);
97     R2 = H_02(1:3,1:3);
98     R3 = H_03(1:3,1:3);

```

```

98  R4 = H_04(1:3,1:3);
99  R5 = H_05(1:3,1:3);
100 R6 = H_06(1:3,1:3);
101
102
103 %%%%%%%%%%%%%%% Encontro da matriz o e z %%%%%%%%%%%%%%%
104
105 z0 = [0 0 1]';
106 o0 = [0 0 0]';
107 z1 = H_01(1:3,3);
108 o1 = H_01(1:3,4);
109 z2 = H_02(1:3,3);
110 o2 = H_02(1:3,4);
111 z3 = H_03(1:3,3);
112 o3 = H_03(1:3,4);
113 z4 = H_04(1:3,3);
114 o4 = H_04(1:3,4);
115 z5 = H_05(1:3,3);
116 o5 = H_05(1:3,4);
117 z6 = H_06(1:3,3);
118 o6 = H_06(1:3,4);
119
120
121 %%%%%%%%%%%%%%% Encontro da matriz J total %%%%%%%%%%%%%%%
122 %Velocidade Linear
123 J1v = cross(z0,(o6-o0));
124 J1v= simplify(J1v);
125 J2v = cross(z1,(o6-o1));
126 J2v = simplify(J2v);
127 J3v = cross(z2,(o6-o2));
128 J3v= simplify(J3v);
129 J4v = cross(z3,(o6-o3));
130 J4v = simplify(J4v);
131 J5v = cross(z4,(o6-o4));
132 J5v = simplify(J5v);
133 J6v = cross(z5,(o6-o5));
134 J6v = simplify(J6v);
135 % Velocidade Angular
136 J1w = z0;
137 J2w = z1;
138 J2w = simplify(J2w);
139 J3w = z2;
140 J3w = simplify(J3w);
141 J4w = z3;
142 J4w = simplify(J4w);
143 J5w = z4;
144 J5w = simplify(J5w);
145 J6w = z5;
146 J6w = simplify(J6w);
147

```

```

148 % Matriz completa
149 J1 = [ J1v;J1w ];
150 J2 = [ J2v;J2w ];
151 J3 = [ J3v;J3w ];
152 J4 = [ J4v;J4w ];
153 J5 = [ J5v;J5w ];
154 J6 = [ J6v;J6w ];
155
156 J = [ J1 , J2 , J3 , J4 , J5 , J6 ];
157
158
159
160
161 %% Dinamica
162 % $\mathcal{D}(q)q'' + C(q,q')q' + g(q) = \tau$ 
163
164 % Variaveis
165 syms q1 q2 q3 q4 q5 q6
166 q1 = tH;
167 q2 = th2;
168 q3 = th3;
169 q4 = th4;
170 q5 = th5;
171 q6 = th6;
172 q = [ q1 , q2 , q3 , q4 , q5 , q6 ];
173
174 syms tH_dot th2_dot th3_dot th4_dot th5_dot th6_dot
175
176 q1_dot = tH_dot;
177 q2_dot = th2_dot;
178 q3_dot = th3_dot;
179 q4_dot = th4_dot;
180 q5_dot = th5_dot;
181 q6_dot = th6_dot;
182
183 q_dot = [ q1_dot ; q2_dot ; q3_dot ; q4_dot ; q5_dot ; q6_dot ];
184
185 %Encontro da Acelera o
186 syms tH_dot2 th2_dot2 th3_dot2 th4_dot2 th5_dot2 th6_dot2
187
188 q1_dot2 = tH_dot2;
189 q2_dot2 = th2_dot2;
190 q3_dot2 = th3_dot2;
191 q4_dot2 = th4_dot2;
192 q5_dot2 = th5_dot2;
193 q6_dot2 = th6_dot2;
194
195 %Valores I hipoteticos
196 syms Ixx1 Iyy1 Izz1
197 syms Ixx2 Iyy6 Izz2

```

```

198 syms Ixx3 Iyy5 Izz3
199 syms Ixx4 Iyy4 Izz4
200 syms Ixx5 Iyy3 Izz5
201 syms Ixx6 Iyy2 Izz6
202
203 %Massa elos
204 syms m1 m2 m3 m4 m5 m6
205
206 %Colocando o I no formato do exercicio
207 I1 = [ Ixx1  0  0;
208        0  Iyy1  0;
209        0  0  Izz1 ];
210
211 I2 = [ Ixx2  0  0;
212        0  Iyy2  0;
213        0  0  Izz2 ];
214
215 I3 = [ Ixx3  0  0;
216        0  Iyy3  0;
217        0  0  Izz3 ];
218
219 I4 = [ Ixx4  0  0;
220        0  Iyy4  0;
221        0  0  Izz4 ];
222
223 I5 = [ Ixx5  0  0;
224        0  Iyy5  0;
225        0  0  Izz5 ];
226
227 I6 = [ Ixx6  0  0;
228        0  Iyy6  0;
229        0  0  Izz6 ];
230
231
232
233
234
235 %% Calculo da Equa o Dinamica
236
237 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Encontro da matriz J Dinamica %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
238 %Colocando o J no formato do exercicio
239 %%J1
240 J1v = cross(z0,(o1-o0));
241 J1w = z0;
242 J1vd = [J1v,0,0,0,0,0];
243 J1wd = [J1w,o0,o0,o0,o0,o0];%se simplificar da erro
244 %%J2
245 J1v = cross(z0,(o2-o0));
246 J2v = cross(z1,(o2-o1));
247 J1w = z0;

```

```

248 J2w = z1;
249 J2vd = [J1v, J2v, 0, 0, 0, 0];
250 J2wd = [J1w, J2w, 0, 0, 0, 0];
251 J2vd = simplify(J2vd);
252 J2wd = simplify(J2wd);
253 %%J3
254 J1v = cross(z0, (o3-o0));
255 J2v = cross(z1, (o3-o1));
256 J3v = cross(z2, (o3-o2));
257 J1w = z0;
258 J2w = z1;
259 J3w = z2;
260 J3vd = [J1v, J2v, J3v, 0, 0, 0];
261 J3wd = [J1w, J2w, J3w, 0, 0, 0];
262 J3vd = simplify(J3vd);
263 J3wd = simplify(J3wd);
264 %%J4
265 J1v = cross(z0, (o4-o0));
266 J2v = cross(z1, (o4-o1));
267 J3v = cross(z2, (o4-o2));
268 J4v = cross(z3, (o4-o3));
269 J1w = z0;
270 J2w = z1;
271 J3w = z2;
272 J4w = z3;
273 J4vd = [J1v, J2v, J3v, J4v, 0, 0];
274 J4wd = [J1w, J2w, J3w, J4w, 0, 0];
275 J4vd = simplify(J4vd);
276 J4wd = simplify(J4wd);
277 %%J5
278 J1v = cross(z0, (o5-o0));
279 J2v = cross(z1, (o5-o1));
280 J3v = cross(z2, (o5-o2));
281 J4v = cross(z3, (o5-o3));
282 J5v = cross(z4, (o5-o4));
283 J1w = z0;
284 J2w = z1;
285 J3w = z2;
286 J4w = z3;
287 J5w = z4;
288 J5vd = [J1v, J2v, J3v, J4v, J5v, 0];
289 J5wd = [J1w, J2w, J3w, J4w, J5w, 0];
290 J5vd = simplify(J5vd);
291 J5wd = simplify(J5wd);
292 %%J6
293 J1v = cross(z0, (o6-o0));
294 J2v = cross(z1, (o6-o1));
295 J3v = cross(z2, (o6-o2));
296 J4v = cross(z3, (o6-o3));
297 J5v = cross(z4, (o6-o4));

```

```

298 J6v = cross(z5,(o6-o5));
299 J1w = z0;
300 J2w = z1;
301 J3w = z2;
302 J4w = z3;
303 J5w = z4;
304 J6w = z5;
305 J6vd = [J1v,J2v,J3v,J4v,J5v,J6v];
306 J6wd = [J1w,J2w,J3w,J4w,J5w,J6w];
307 J6vd = simplify(J6vd);
308 J6wd = simplify(J6wd);
309
310 % Calculo da Matriz D
311
312 %D = m[Jv]'[Jv]+[Jw]'[R][I][R]'[Jw]
313
314 D1 = m1*(transpose(J1vd))*J1vd + transpose(J1wd)*(R1*I1*transpose(R1))*
    J1wd;
315 D1 = expand(D1);
316 D1 = simplify(D1);
317 D2 = m2*(transpose(J2vd))*J2vd + transpose(J2wd)*(R2*I2*transpose(R2))*
    J2wd;
318 D2 = simplify(D2);
319 D3 = m3*(transpose(J3vd))*J3vd + transpose(J3wd)*(R3*I3*transpose(R3))*
    J3wd;
320 D3 = simplify(D3);
321 D4 = m4*(transpose(J4vd))*J4vd + transpose(J4wd)*(R4*I4*transpose(R4))*
    J4wd;
322 D4 = simplify(D4);
323 D5 = m5*(transpose(J5vd))*J5vd + transpose(J5wd)*(R5*I5*transpose(R5))*
    J5wd;
324 D5 = simplify(D5);
325 D6 = m6*(transpose(J6vd))*J6vd + transpose(J6wd)*(R6*I6*transpose(R6))*
    J6wd;
326 D6 = simplify(D6);
327 D = D1+D2+D3+D4+D5+D6;
328 D = simplify(D);
329 %Calculo da Matriz C
330 %P=diff(f,x);
331 %subs(P,{x,y},{1.5,2});
332
333 C = zeros(6,6,6,'sym');
334 syms meio
335 for i=1:6
336 for j=1:6
337 for k=1:6
338
339 %meio = 0.5
340 C(i,j,k) = 0.5*diff(D(k,j),q(i))+ diff(D(k,i),q(j)) - diff(D(i,j),q(
    k));

```



```

341
342 end
343 end
344 end
345
346 C1 = [C(1,1,1) C(1,2,1) C(1,3,1) C(1,4,1) C(1,5,1) C(1,6,1);
347        C(1,1,2) C(1,2,2) C(1,3,2) C(1,4,2) C(1,5,2) C(1,6,2);
348        C(1,1,3) C(1,2,3) C(1,3,3) C(1,4,3) C(1,5,3) C(1,6,3);
349        C(1,1,4) C(1,2,4) C(1,3,4) C(1,4,4) C(1,5,4) C(1,6,4);
350        C(1,1,5) C(1,2,5) C(1,3,5) C(1,4,5) C(1,5,5) C(1,6,5);
351        C(1,1,6) C(1,2,6) C(1,3,6) C(1,4,6) C(1,5,6) C(1,6,6)];
352 C1 = simplify(C1);
353
354 C2 = [C(2,1,1) C(2,2,1) C(2,3,1) C(2,4,1) C(2,5,1) C(2,6,1);
355        C(2,1,2) C(2,2,2) C(2,3,2) C(2,4,2) C(2,5,2) C(2,6,2);
356        C(2,1,3) C(2,2,3) C(2,3,3) C(2,4,3) C(2,5,3) C(2,6,3);
357        C(2,1,4) C(2,2,4) C(2,3,4) C(2,4,4) C(2,5,4) C(2,6,4);
358        C(2,1,5) C(2,2,5) C(2,3,5) C(2,4,5) C(2,5,5) C(2,6,5);
359        C(2,1,6) C(2,2,6) C(2,3,6) C(2,4,6) C(2,5,6) C(2,6,6)];
360 C2 = simplify(C2);
361
362 C3 = [C(3,1,1) C(3,2,1) C(3,3,1) C(3,4,1) C(3,5,1) C(3,6,1);
363        C(3,1,2) C(3,2,2) C(3,3,2) C(3,4,2) C(3,5,2) C(3,6,2);
364        C(3,1,3) C(3,2,3) C(3,3,3) C(3,4,3) C(3,5,3) C(3,6,3);
365        C(3,1,4) C(3,2,4) C(3,3,4) C(3,4,4) C(3,5,4) C(3,6,4);
366        C(3,1,5) C(3,2,5) C(3,3,5) C(3,4,5) C(3,5,5) C(3,6,5);
367        C(3,1,6) C(3,2,6) C(3,3,6) C(3,4,6) C(3,5,6) C(3,6,6)];
368 C3 = simplify(C3);
369
370 C4 = [C(4,1,1) C(4,2,1) C(4,3,1) C(4,4,1) C(4,5,1) C(4,6,1);
371        C(4,1,2) C(4,2,2) C(4,3,2) C(4,4,2) C(4,5,2) C(4,6,2);
372        C(4,1,3) C(4,2,3) C(4,3,3) C(4,4,3) C(4,5,3) C(4,6,3);
373        C(4,1,4) C(4,2,4) C(4,3,4) C(4,4,4) C(4,5,4) C(4,6,4);
374        C(4,1,5) C(4,2,5) C(4,3,5) C(4,4,5) C(4,5,5) C(4,6,5);
375        C(4,1,6) C(4,2,6) C(4,3,6) C(4,4,6) C(4,5,6) C(4,6,6)];
376 C4 = simplify(C4);
377
378 C5 = [C(5,1,1) C(5,2,1) C(5,3,1) C(5,4,1) C(5,5,1) C(5,6,1);
379        C(5,1,2) C(5,2,2) C(5,3,2) C(5,4,2) C(5,5,2) C(5,6,2);
380        C(5,1,3) C(5,2,3) C(5,3,3) C(5,4,3) C(5,5,3) C(5,6,3);
381        C(5,1,4) C(5,2,4) C(5,3,4) C(5,4,4) C(5,5,4) C(5,6,4);
382        C(5,1,5) C(5,2,5) C(5,3,5) C(5,4,5) C(5,5,5) C(5,6,5);
383        C(5,1,6) C(5,2,6) C(5,3,6) C(5,4,6) C(5,5,6) C(5,6,6)];
384 C5 = simplify(C5);
385
386 C6 = [C(6,1,1) C(6,2,1) C(6,3,1) C(6,4,1) C(6,5,1) C(6,6,1);
387        C(6,1,2) C(6,2,2) C(6,3,2) C(6,4,2) C(6,5,2) C(6,6,2);
388        C(6,1,3) C(6,2,3) C(6,3,3) C(6,4,3) C(6,5,3) C(6,6,3);
389        C(6,1,4) C(6,2,4) C(6,3,4) C(6,4,4) C(6,5,4) C(6,6,4);
390        C(6,1,5) C(6,2,5) C(6,3,5) C(6,4,5) C(6,5,5) C(6,6,5)];

```

```

391         C(6,1,6) C(6,2,6) C(6,3,6) C(6,4,6) C(6,5,6) C(6,6,6)] ;
392 C6 = simplify(C6);
393
394 C = C1*q1_dot + C2*q2_dot + C3*q3_dot + C4*q4_dot + C5*q5_dot + C6*
    q6_dot;
395 C = simplify(C);
396
397 %% Calculo da Energia Cinetica E Potencial
398 %%Calculo da Energia Cinetica
399
400 K = 0.5*transpose(q_dot)*D*q_dot;
401 K = simplify(K);
402
403 %Energia Potencia
404 syms g% accelera o da gravidade
405 P1 = H_01(3,4)*m1*g;
406 P2 = H_02(3,4)*m2*g;
407 P3= H_03(3,4)*m3*g;
408 P4= H_04(3,4)*m4*g;
409 P5= H_05(3,4)*m5*g;
410 P6= H_06(3,4)*m6*g;
411 P = P1+P2+P3+P4+P5+P6;
412 P = simplify(P);
413
414 %Encontro da Matriz g(q)
415 g1 = diff(P,q(1));
416 g2 = diff(P,q(2));
417 g3 = diff(P,q(3));
418 g4 = diff(P,q(4));
419 g5 = diff(P,q(5));
420 g6 = diff(P,q(6));
421
422 G = [g1;g2;g3;g4;g5;g6];
423
424 %Tals Simbolicos-> Substituir por valores Reais
425 syms Tal1 Tal2 Tal3 Tal4 Tal5 Tal6
426 Tal = [Tal1;Tal2;Tal3;Tal4;Tal5;Tal6];
427
428 q_dot2 = [q1_dot2;q2_dot2;q3_dot2;q4_dot2;q5_dot2;q6_dot2];
429
430
431 Tau = D*q_dot2+C*q_dot+G;
432
433
434
435 %Equa o Geral para encontrar a accelera o
436 %q_dot2 = D\(Tal-C*q_dot-g);

```

## 12.3 Trajetória

### 12.3.1 PPPR

```
1  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Projeto Robótica %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2  %% Vetores das variáveis e suas derivadas em função do tempo.
3
4  syms t
5
6  aCst = 8;
7  VCst = 0.4;
8
9  tra = [1,2,1,2,1,2,1,2,1,2,1,1,2,1,2,1,2,1,2];
10 alpha = [aCst, aCst, -aCst, aCst, aCst, aCst, -aCst, aCst, aCst, aCst, -aCst, aCst,
           aCst, -aCst, aCst, aCst, aCst, -aCst, -aCst];
11 x0 = [0,4,4,0,0,4,4,0,0,4,4,0,3.75,3.75,0.75,0.75,3.75,3.75,0];
12 xf = [4,4,0,0,4,4,0,0,4,4,0,3.75,3.75,0.75,0.75,3.75,3.75,0,0];
13 y0 = [0,0,0.2,0.2,0.4,0.4,0.6,0.6,0.8,0.8,1,1,1,1,1.1667,1.1667,
        1.3333,1.3333,1.5,1.5];
14 yf =
        [0,0.2,0.2,0.4,0.4,0.6,0.6,0.8,0.8,1,1,1,1,1.1667,1.1667,1.3333,1.3333,1.5,1.5,0];
15 tf = [10.05,0.55,10.05,0.55,10.05,0.55,10.05,0.55,10.05,0.55,10.05,8.175,
16       0.4668,6.3,0.4668,6.3,0.4668,8.175,3.8];
17 V = [VCst, VCst, -VCst, VCst, VCst, VCst, -VCst, VCst, VCst, VCst, -VCst, VCst, VCst, -
        VCst, VCst, VCst, VCst, -VCst, -VCst];
18 tb = 0.05;
19 z = 0.5;
20
21 for j = 1:length(tra)
22     if tra(j) == 1
23         k0 = x0(j);
24         kf = xf(j);
25     else
26         k0 = y0(j);
27         kf = yf(j);
28     end
29
30     done1 = 0;
31     done2 = 0;
32
33     for m = 0:0.01:tf(j)
34
35         if (j == 1) && (m == 0)
36             k1(t) = k0 + (alpha(j)/2)*(t^2);
37
38             d1 = z;
39             d2 = k1(t);
40             d3 = -y0(j);
41             th4 = 0;
```

```

42
43 elseif m == 0
44     k1(t) = k0 + (alpha(j)/2)*(t^2);
45
46     if tra(j) == 1
47         d1 = [d1 z];
48         d2 = [d2 k1(t)];
49         d3 = [d3 -y0(j)];
50         th4 = [th4 0];
51     else
52         d1 = [d1 z];
53         d2 = [d2 x0(j)];
54         d3 = [d3 -k1(t)];
55         th4 = [th4 0];
56     end
57
58 elseif m > tb && done1 ~= 1
59     k2(t) = (kf+k0-V(j)*tf(j))/2 +V(j)*t;
60     done1 = 1;
61
62     if tra(j) == 1
63         d1 = [d1 z];
64         d2 = [d2 k2(t)];
65         d3 = [d3 -y0(j)];
66         th4 = [th4 0];
67     else
68         d1 = [d1 z];
69         d2 = [d2 x0(j)];
70         d3 = [d3 -k2(t)];
71         th4 = [th4 0];
72     end
73
74 elseif m > (tf(j) - tb) && done2 ~= 1
75     k3(t) = kf-(alpha(j)/2)*(tf(j)^2)+alpha(j)*tf(j)*t-(alpha(j)/2)*(
76         t^2);
77     done2 = 1;
78
79     if tra(j) == 1
80         d1 = [d1 z];
81         d2 = [d2 k3(t)];
82         d3 = [d3 -y0(j)];
83         th4 = [th4 0];
84     else
85         d1 = [d1 z];
86         d2 = [d2 x0(j)];
87         d3 = [d3 -k3(t)];
88         th4 = [th4 0];
89     end
90 else

```

```

91         s = length(d1);
92         d1 = [d1 d1(s)];
93         d2 = [d2 d2(s)];
94         d3 = [d3 d3(s)];
95         th4 = [th4 0];
96     end
97 end
98 end
99
100 d1_dot = diff(d1,t);
101 d2_dot = diff(d2,t);
102 d3_dot = diff(d3,t);
103 th4_dot = diff(th4,t);
104
105 d1_dot2 = diff(d1_dot,t);
106 d2_dot2 = diff(d2_dot,t);
107 d3_dot2 = diff(d3_dot,t);
108 th4_dot2 = diff(th4_dot,t);
109
110 %% Valores das vari veis no tempo
111 temp=1;
112 tempTot=0;
113 tempV = zeros(1,length(d1));
114 d1V = zeros(1,length(d1));
115 d2V = zeros(1,length(d2));
116 d3V = zeros(1,length(d3));
117 th4V = zeros(1,length(th4));
118
119 for j = 1:1:length(tra)
120     for m = 0:0.01:tf(j)
121         tempV(temp) = m+tempTot;
122
123         d1V(temp) = subs(d1(temp),t,m);
124         d2V(temp) = subs(d2(temp),t,m);
125         d3V(temp) = subs(d3(temp),t,m)+1.5;
126         th4V(temp) = subs(th4(temp),t,m);
127
128         temp = temp+1;
129     end
130     tempTot = tempTot + tf(j);
131 end
132
133 subplot(4,1,1);
134 plot(tempV,d1V)
135 title('d1')
136
137 subplot(4,1,2);
138 plot(tempV,d2V)
139 title('d2')
140

```

```

141 subplot(4,1,3);
142 plot(tempV,d3V)
143 title('d3')
144
145 subplot(4,1,4);
146 plot(tempV,th4V)
147 title('theta4')
148
149 %% Valores das velocidades das variaveis no tempo
150 temp=1;
151 tempTot=0;
152 tempV = zeros(1,length(d1_dot));
153 d1_dotV = zeros(1,length(d1_dot));
154 d2_dotV = zeros(1,length(d2_dot));
155 d3_dotV = zeros(1,length(d3_dot));
156 th4_dotV = zeros(1,length(th4_dot));
157
158 for j = 1:length(tra)
159     for m = 0:0.01:tf(j)
160         tempV(temp) = m+tempTot;
161
162         d1_dotV(temp) = subs(d1_dot(temp),t,m);
163         d2_dotV(temp) = subs(d2_dot(temp),t,m);
164         d3_dotV(temp) = subs(d3_dot(temp),t,m);
165         th4_dotV(temp) = subs(th4_dot(temp),t,m);
166
167         temp = temp+1;
168     end
169     tempTot = tempTot + tf(j);
170 end
171
172 subplot(4,1,1);
173 plot(tempV,d1_dotV)
174 title('vel d1')
175
176 subplot(4,1,2);
177 plot(tempV,d2_dotV)
178 title('vel d2')
179
180 subplot(4,1,3);
181 plot(tempV,d3_dotV)
182 title('vel d3')
183
184 subplot(4,1,4);
185 plot(tempV,th4_dotV)
186 title('vel theta4')
187
188
189 %% Valores das acelerações das variaveis no tempo
190 temp=1;

```

```

191 tempTot=0;
192 tempV = zeros(1,length(d1_dot2));
193 d1_dot2V = zeros(1,length(d1_dot2));
194 d2_dot2V = zeros(1,length(d2_dot2));
195 d3_dot2V = zeros(1,length(d3_dot2));
196 th4_dot2V = zeros(1,length(th4_dot2));
197
198 for j = 1:1:length(tra)
199     for m = 0:0.01:tf(j)
200         tempV(temp) = m+tempTot;
201
202         d1_dot2V(temp) = subs(d1_dot2(temp),t,m);
203         d2_dot2V(temp) = subs(d2_dot2(temp),t,m);
204         d3_dot2V(temp) = subs(d3_dot2(temp),t,m);
205         th4_dot2V(temp) = subs(th4_dot2(temp),t,m);
206
207         temp = temp+1;
208     end
209     tempTot = tempTot + tf(j);
210 end
211
212 subplot(4,1,1);
213 plot(tempV,d1_dot2V)
214 title('acc d1')
215
216 subplot(4,1,2);
217 plot(tempV,d2_dot2V)
218 title('acc d2')
219
220 subplot(4,1,3);
221 plot(tempV,d3_dot2V)
222 title('acc d3')
223
224 subplot(4,1,4);
225 plot(tempV,th4_dot2V)
226 title('acc theta4')

```

### 12.3.2 RRRRRR

```

1  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Projeto Rob tica %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2  %% Vetores das vari veis e suas derivadas em fun o do tempo.
3  clear all;
4
5  syms theta1;
6  syms theta2;
7  syms theta3;
8  syms theta4;
9  syms theta5;
10 syms theta6;
11
12 th1EQ = zeros(1,1);
13 th2EQ = zeros(1,1);
14 th3EQ = zeros(1,1);
15 th4EQ = zeros(1,1);
16 th5EQ = zeros(1,1);
17 th6EQ = zeros(1,1);
18
19 syms xc yc zc t;
20
21 aCst = 8;
22 VCst = 0.4;
23
24 tra = [1,2,1,2,1,2,1,2,1,2,1,1,2,1,2,1,2,1,2];
25 alpha = [aCst,aCst,-aCst,aCst,aCst,aCst,-aCst,aCst,aCst,aCst,-aCst,aCst,
26          aCst,-aCst,aCst,aCst,aCst,-aCst,-aCst];
27 x0 = [0,4,4,0,0,4,4,0,0,4,4,0,3.75,3.75,0.75,0.75,3.75,3.75,0];
28 yf =
29     [0,0,0.2,0.2,0.4,0.4,0.6,0.6,0.8,0.8,1,1,1,1.1667,1.1667,1.3333,1.3333,1.5,1.5];
30 yf =
31     [0,0.2,0.2,0.4,0.4,0.6,0.6,0.8,0.8,1,1,1,1.1667,1.1667,1.3333,1.3333,1.5,1.5,0];
32
33 tf = [10.05,0.55,10.05,0.55,10.05,0.55,10.05,0.55,10.05,0.55,
34       10.05,8.175,0.4668,6.3,0.4668,6.3,0.4668,8.175,3.8];
35 V = [VCst,VCst,-VCst,VCst,VCst,VCst,-VCst,VCst,VCst,VCst,-VCst,VCst,VCst,-
36       VCst,VCst,VCst,VCst,-VCst,-VCst];
37 tb = 0.05;
38 z = 0.5;
39 a2 = 2;
40 a3 = 2;
41
42 % Matriz R
43 r11 = sin(theta6)*(cos(theta4)*sin(theta1) + sin(theta4)*(cos(theta1)*sin(
44     theta2)*sin(theta3) - cos(theta1)*cos(theta2)*cos(theta3))) + cos(theta6)
45     *(cos(theta5)*(sin(theta1)*sin(theta4) - cos(theta4)*(cos(theta1)*sin(
46     theta2)*sin(theta3) - cos(theta1)*cos(theta2)*cos(theta3))) - sin(theta5)

```



```

    * (cos(theta1)*cos(theta2)*sin(theta3) + cos(theta1)*cos(theta3)*sin(
theta2)));
40 r12 = cos(theta6)*(cos(theta4)*sin(theta1) + sin(theta4)*(cos(theta1)*sin(
theta2)*sin(theta3) - cos(theta2)*cos(theta2)*cos(theta3))) - sin(theta6)
*(cos(theta5)*(sin(theta1)*sin(theta4) - cos(theta4)*(cos(theta1)*sin(
theta2)*sin(theta3) - cos(theta1)*cos(theta2)*cos(theta3))) - sin(theta5)
*(cos(theta1)*cos(theta2)*sin(theta3) + cos(theta1)*cos(theta3)*sin(
theta2)));
41 r13 = - sin(theta5)*(sin(theta1)*sin(theta4) - cos(theta4)*(cos(theta1)*sin(
theta2)*sin(theta3) - cos(theta1)*cos(theta2)*cos(theta3))) - cos(theta5)
*(cos(theta1)*cos(theta2)*sin(theta3) + cos(theta2)*cos(theta3)*sin(
theta2));
42 r21 = - sin(theta6)*(cos(theta1)*cos(theta4) - sin(theta4)*(sin(theta1)*sin(
theta2)*sin(theta3) - cos(theta2)*cos(theta3)*sin(theta1))) - cos(theta6)
*(cos(theta5)*(cos(theta1)*sin(theta4) + cos(theta4)*(sin(theta1)*sin(
theta2)*sin(theta3) - cos(theta2)*cos(theta3)*sin(theta1))) + sin(theta5)
*(cos(theta2)*sin(theta1)*sin(theta3) + cos(theta3)*sin(theta1)*sin(
theta2)));
43 r22 = sin(theta6)*(cos(theta5)*(cos(theta1)*sin(theta4) + cos(theta4)*(sin(
theta1)*sin(theta2)*sin(theta3) - cos(theta2)*cos(theta3)*sin(theta1))) +
sin(theta5)*(cos(theta2)*sin(theta1)*sin(theta3) + cos(theta3)*sin(
theta1)*sin(theta2))) - cos(theta6)*(cos(theta1)*cos(theta4) - sin(theta4)
*(sin(theta1)*sin(theta2)*sin(theta3) - cos(theta2)*cos(theta3)*sin(
theta1)));
44 r23 = sin(theta5)*(cos(theta1)*sin(theta4) + cos(theta4)*(sin(theta1)*sin(
theta2)*sin(theta3) - cos(theta2)*cos(theta3)*sin(theta1))) - cos(theta5)
*(cos(theta2)*sin(theta1)*sin(theta3) + cos(theta3)*sin(theta1)*sin(
theta2));
45 r31 = sin(theta2 + theta3)*sin(theta4)*sin(theta6) - cos(theta6)*(cos(theta2
+ theta3)*sin(theta5) + sin(theta2 + theta3)*cos(theta4)*cos(theta5));
46 r32 = sin(theta6)*(cos(theta2 + theta3)*sin(theta5) + sin(theta2 + theta3)*
cos(theta4)*cos(theta5)) + sin(theta2 + theta3)*cos(theta6)*sin(theta4);
47 r33 = sin(theta2 + theta3)*cos(theta4)*sin(theta5) - cos(theta2 + theta3)*
cos(theta5);
48
49 q1(xc, yc, zc) = atan2(xc, yc);
50 q2(xc, yc, zc, theta3) = atan2(sqrt(xc^2 + yc^2), zc) - atan2(a2 + a3*cos(
theta3), a3*sin(theta3));
51 D = (xc^2 + yc^2 + zc^2 - a2^2 - a3^2)/(2*a2*a3);
52 q3(xc, yc, zc) = atan2(D, sqrt(1 - D^2));
53 q4(xc, yc, zc, theta1, theta2, theta3) = atan2(cos(theta1)*cos(theta2 +
theta3)*r13 + sin(theta1)*cos(theta2 + theta3)*r23 + sin(theta2 + theta3)
*r33, -cos(theta1)*cos(theta2 + theta3)*r13 - sin(theta1)*cos(theta2 +
theta3)*r23 + cos(theta2 + theta3)*r33);
54 q5(xc, yc, zc, theta1, theta2, theta3) = atan2(sin(theta1)*r13 - cos(theta1)
*r23, sqrt(1 - (sin(theta1)*r13 - cos(theta1)*r23)^2));
55 q6(xc, yc, zc, theta1, theta2, theta3) = atan2(-sin(theta1)*r11 + cos(theta1)
)*r21, sin(theta1)*r12 - cos(theta1)*r22);
56
57 for j = 1:length(tra)

```

```

58     if tra(j) == 1
59         k0 = x0(j);
60         kf = xf(j);
61     else
62         k0 = y0(j);
63         kf = yf(j);
64     end
65
66     for m = 0:(tf(j)/10):tf(j)
67
68         if (j == 1) && (m == 0)
69             k1(t) = k0 + (alpha(j)/2)*(t^2);
70
71             th1 = q1(k1(m), y0(j), z);
72             th3 = q3(k1(m), y0(j), z);
73             th2 = q2(k1(m), y0(j), z, th3);
74             th4(theta4, theta5, theta6) = q4(k1(m), y0(j), z, th1, th2, th3)
75             ;
76             th5(theta4, theta5, theta6) = q5(k1(m), y0(j), z, th1, th2, th3)
77             ;
78             th6(theta4, theta5, theta6) = q6(k1(m), y0(j), z, th1, th2, th3)
79             ;
80
81             eq4 = theta4 == th4;
82             eq5 = theta5 == th5;
83             eq6 = theta6 == th6;
84
85             [solT4, solT5, solT6] = solve([eq4, eq5, eq6], [theta4, theta5,
86                 theta6]);
87
88             th1EQ = th1;
89             th2EQ = th2;
90             th3EQ = th3;
91             th4EQ = solT4;
92             th5EQ = solT5;
93             th6EQ = solT6;
94
95         elseif m >= 0 && m <= tb
96             k1(t) = k0 + (alpha(j)/2)*(t^2);
97
98             if tra(j) == 1
99                 th1 = q1(k1(m), y0(j), z);
100                 th3 = q3(k1(m), y0(j), z);
101                 th2 = q2(k1(m), y0(j), z, th3);
102                 th4(theta4, theta5, theta6) = q4(k1(m), y0(j), z, th1, th2,
103                     th3);
104                 th5(theta4, theta5, theta6) = q5(k1(m), y0(j), z, th1, th2,
105                     th3);
106                 th6(theta4, theta5, theta6) = q6(k1(m), y0(j), z, th1, th2,
107                     th3);

```

```

101
102      eq4 = theta4 == th4;
103      eq5 = theta5 == th5;
104      eq6 = theta6 == th6;
105
106      [solT4, solT5, solT6] = solve([eq4, eq5, eq6], [theta4,
      theta5, theta6]);
107
108      th1EQ = [th1EQ th1];
109      th2EQ = [th2EQ th2];
110      th3EQ = [th3EQ th3];
111      th4EQ = [th4EQ solT4];
112      th5EQ = [th5EQ solT5];
113      th6EQ = [th6EQ solT6];
114  else
115      th1 = q1(x0(j), k1(m), z);
116      th3 = q3(x0(j), k1(m), z);
117      th2 = q2(x0(j), k1(m), z, th3);
118      th4(theta4, theta5, theta6) = q4(x0(j), k1(m), z, th1, th2,
      th3);
119      th5(theta4, theta5, theta6) = q5(x0(j), k1(m), z, th1, th2,
      th3);
120      th6(theta4, theta5, theta6) = q6(x0(j), k1(m), z, th1, th2,
      th3);
121
122      eq4 = theta4 == th4;
123      eq5 = theta5 == th5;
124      eq6 = theta6 == th6;
125
126      [solT4, solT5, solT6] = solve([eq4, eq5, eq6], [theta4,
      theta5, theta6]);
127
128      th1EQ = [th1EQ th1];
129      th2EQ = [th2EQ th2];
130      th3EQ = [th3EQ th3];
131      th4EQ = [th4EQ solT4];
132      th5EQ = [th5EQ solT5];
133      th6EQ = [th6EQ solT6];
134  end
135
136  elseif m > tb && m <= tf(j) - tb
137      k2(t) = (kf+k0-V(j)*tf(j))/2 +V(j)*t;
138
139      if tra(j) == 1
140          th1 = q1(k2(m), y0(j), z);
141          th3 = q3(k2(m), y0(j), z);
142          th2 = q2(k2(m), y0(j), z, th3);
143          th4(theta4, theta5, theta6) = q4(k2(m), y0(j), z, th1, th2,
      th3);
144          th5(theta4, theta5, theta6) = q5(k2(m), y0(j), z, th1, th2,

```

```

145         th3);
146         th6(theta4, theta5, theta6) = q6(k2(m), y0(j), z, th1, th2,
147         th3);
148
149         eq4 = theta4 == th4;
150         eq5 = theta5 == th5;
151         eq6 = theta6 == th6;
152
153         [solT4, solT5, solT6] = solve([eq4, eq5, eq6], [theta4,
154         theta5, theta6]);
155
156         th1EQ = [th1EQ th1];
157         th2EQ = [th2EQ th2];
158         th3EQ = [th3EQ th3];
159         th4EQ = [th4EQ solT4];
160         th5EQ = [th5EQ solT5];
161         th6EQ = [th6EQ solT6];
162
163     else
164         th1 = q1(x0(j), k2(m), z);
165         th3 = q3(x0(j), k2(m), z);
166         th2 = q2(x0(j), k2(m), z, th3);
167         th4(theta4, theta5, theta6) = q4(x0(j), k2(m), z, th1, th2,
168         th3);
169         th5(theta4, theta5, theta6) = q5(x0(j), k2(m), z, th1, th2,
170         th3);
171         th6(theta4, theta5, theta6) = q6(x0(j), k2(m), z, th1, th2,
172         th3);
173
174         eq4 = theta4 == th4;
175         eq5 = theta5 == th5;
176         eq6 = theta6 == th6;
177
178         [solT4, solT5, solT6] = solve([eq4, eq5, eq6], [theta4,
179         theta5, theta6]);
180
181         th1EQ = [th1EQ th1];
182         th2EQ = [th2EQ th2];
183         th3EQ = [th3EQ th3];
184         th4EQ = [th4EQ solT4];
185         th5EQ = [th5EQ solT5];
186         th6EQ = [th6EQ solT6];
187
188     end
189
190     elseif m > (tf(j) - tb)
191         k3(t) = kf - (alpha(j)/2)*(tf(j)^2) + alpha(j)*tf(j)*t - (alpha(j)/2)*(
192         t^2);
193
194         if tra(j) == 1
195             th1 = q1(k3(m), y0(j), z);
196             th3 = q3(k3(m), y0(j), z);

```

```

187     th2 = q2(k3(m), y0(j), z, th3);
188     th4(theta4, theta5, theta6) = q4(k3(m), y0(j), z, th1, th2,
189         th3);
189     th5(theta4, theta5, theta6) = q5(k3(m), y0(j), z, th1, th2,
190         th3);
190     th6(theta4, theta5, theta6) = q6(k3(m), y0(j), z, th1, th2,
191         th3);
191
192     eq4 = theta4 == th4;
193     eq5 = theta5 == th5;
194     eq6 = theta6 == th6;
195
196     [solT4, solT5, solT6] = solve([eq4, eq5, eq6], [theta4,
197         theta5, theta6]);
197
198     th1EQ = [th1EQ th1];
199     th2EQ = [th2EQ th2];
200     th3EQ = [th3EQ th3];
201     th4EQ = [th4EQ solT4];
202     th5EQ = [th5EQ solT5];
203     th6EQ = [th6EQ solT6];
204
205     else
206         th1 = q1(x0(j), k3(m), z);
207         th3 = q3(x0(j), k3(m), z);
208         th2 = q2(x0(j), k3(m), z, th3);
209         th4(theta4, theta5, theta6) = q4(x0(j), k3(m), z, th1, th2,
210             th3);
210         th5(theta4, theta5, theta6) = q5(x0(j), k3(m), z, th1, th2,
211             th3);
211         th6(theta4, theta5, theta6) = q6(x0(j), k3(m), z, th1, th2,
212             th3);
212
213         eq4 = theta4 == th4;
214         eq5 = theta5 == th5;
215         eq6 = theta6 == th6;
216
217         [solT4, solT5, solT6] = solve([eq4, eq5, eq6], [theta4,
218             theta5, theta6]);
218
219         th1EQ = [th1EQ th1];
220         th2EQ = [th2EQ th2];
221         th3EQ = [th3EQ th3];
222         th4EQ = [th4EQ solT4];
223         th5EQ = [th5EQ solT5];
224         th6EQ = [th6EQ solT6];
225
226     end
227 end
228

```

```

229 %% Velocidades e Acelera es
230
231 tempV = zeros(1,length(th1EQ));
232
233 temp=1;
234 tempTot=0;
235
236 for j = 1:length(tra)
237     for m = 0:(tf(j)/10):tf(j)
238         tempV(temp) = m+tempTot;
239         temp = temp+1;
240     end
241     tempTot = tempTot + tf(j);
242 end
243
244 temp=1;
245 for j = 1:length(tra)
246     for m = 0:(tf(j)/10):tf(j)
247         if temp == 2
248             th1EQ_dot = (th1EQ(2)-th1EQ(1))/(tempV(2)-tempV(1));
249             th2EQ_dot = (th2EQ(2)-th2EQ(1))/(tempV(2)-tempV(1));
250             th3EQ_dot = (th3EQ(2)-th3EQ(1))/(tempV(2)-tempV(1));
251             th4EQ_dot = (th4EQ(2)-th4EQ(1))/(tempV(2)-tempV(1));
252             th5EQ_dot = (th5EQ(2)-th5EQ(1))/(tempV(2)-tempV(1));
253             th6EQ_dot = (th6EQ(2)-th6EQ(1))/(tempV(2)-tempV(1));
254         elseif temp > 2
255             th1EQ_dot(temp-1) = (th1EQ(temp)-th1EQ(temp-1))/(tempV(temp)-
                tempV(temp-1));
256             th2EQ_dot(temp-1) = (th2EQ(temp)-th2EQ(temp-1))/(tempV(temp)-
                tempV(temp-1));
257             th3EQ_dot(temp-1) = (th3EQ(temp)-th3EQ(temp-1))/(tempV(temp)-
                tempV(temp-1));
258             th4EQ_dot(temp-1) = (th4EQ(temp)-th4EQ(temp-1))/(tempV(temp)-
                tempV(temp-1));
259             th5EQ_dot(temp-1) = (th5EQ(temp)-th5EQ(temp-1))/(tempV(temp)-
                tempV(temp-1));
260             th6EQ_dot(temp-1) = (th6EQ(temp)-th6EQ(temp-1))/(tempV(temp)-
                tempV(temp-1));
261         end
262         temp = temp + 1;
263     end
264 end
265 th1EQ_dot = [th1EQ_dot(1) th1EQ_dot];
266 th2EQ_dot = [th2EQ_dot(1) th2EQ_dot];
267 th3EQ_dot = [th3EQ_dot(1) th3EQ_dot];
268 th4EQ_dot = [th4EQ_dot(1) th4EQ_dot];
269 th5EQ_dot = [th5EQ_dot(1) th5EQ_dot];
270 th6EQ_dot = [th6EQ_dot(1) th6EQ_dot];
271
272

```

```

273 temp=1;
274 for j = 1:length(tra)
275     for m = 0:(tf(j)/10):tf(j)
276         if temp > length(th1EQ_dot)
277
278             elseif temp == 2
279                 th1EQ_dot2 = (th1EQ_dot(2)-th1EQ_dot(1))/(tempV(2)-tempV(1));
280                 th2EQ_dot2 = (th2EQ_dot(2)-th2EQ_dot(1))/(tempV(2)-tempV(1));
281                 th3EQ_dot2 = (th3EQ_dot(2)-th3EQ_dot(1))/(tempV(2)-tempV(1));
282                 th4EQ_dot2 = (th4EQ_dot(2)-th4EQ_dot(1))/(tempV(2)-tempV(1));
283                 th5EQ_dot2 = (th5EQ_dot(2)-th5EQ_dot(1))/(tempV(2)-tempV(1));
284                 th6EQ_dot2 = (th6EQ_dot(2)-th6EQ_dot(1))/(tempV(2)-tempV(1));
285             elseif temp > 2
286                 th1EQ_dot2(temp-1) = (th1EQ_dot(temp)-th1EQ_dot(temp-1))/(tempV(
                    temp)-tempV(temp-1));
287                 th2EQ_dot2(temp-1) = (th2EQ_dot(temp)-th2EQ_dot(temp-1))/(tempV(
                    temp)-tempV(temp-1));
288                 th3EQ_dot2(temp-1) = (th3EQ_dot(temp)-th3EQ_dot(temp-1))/(tempV(
                    temp)-tempV(temp-1));
289                 th4EQ_dot2(temp-1) = (th4EQ_dot(temp)-th4EQ_dot(temp-1))/(tempV(
                    temp)-tempV(temp-1));
290                 th5EQ_dot2(temp-1) = (th5EQ_dot(temp)-th5EQ_dot(temp-1))/(tempV(
                    temp)-tempV(temp-1));
291                 th6EQ_dot2(temp-1) = (th6EQ_dot(temp)-th6EQ_dot(temp-1))/(tempV(
                    temp)-tempV(temp-1));
292             end
293             temp = temp + 1;
294         end
295     end
296
297     th1EQ_dot2 = [th1EQ_dot2(1) th1EQ_dot2];
298     th2EQ_dot2 = [th2EQ_dot2(1) th2EQ_dot2];
299     th3EQ_dot2 = [th3EQ_dot2(1) th3EQ_dot2];
300     th4EQ_dot2 = [th4EQ_dot2(1) th4EQ_dot2];
301     th5EQ_dot2 = [th5EQ_dot2(1) th5EQ_dot2];
302     th6EQ_dot2 = [th6EQ_dot2(1) th6EQ_dot2];
303
304     %% Plot Position
305     figure
306     subplot(6,1,1);
307     plot(tempV,th1EQ)
308     title('theta1')
309
310     subplot(6,1,2);
311     plot(tempV,th2EQ)
312     title('theta2')
313
314     subplot(6,1,3);
315     plot(tempV,th3EQ)
316     title('theta3')

```

```

317
318 subplot(6,1,4);
319 plot(tempV,th4EQ)
320 title('theta4')
321
322 subplot(6,1,5);
323 plot(tempV,th5EQ)
324 title('theta5')
325
326 subplot(6,1,6);
327 plot(tempV,th6EQ)
328 title('theta6')
329
330 %% Plot Velocidade
331 figure
332 subplot(6,1,1);
333 plot(tempV,th1EQ_dot)
334 title('vel theta1')
335
336 subplot(6,1,2);
337 plot(tempV,th2EQ_dot)
338 title('vel theta2')
339
340 subplot(6,1,3);
341 plot(tempV,th3EQ_dot)
342 title('vel theta3')
343
344 subplot(6,1,4);
345 plot(tempV,th4EQ_dot)
346 title('vel theta4')
347
348 subplot(6,1,5);
349 plot(tempV,th5EQ_dot)
350 title('vel theta5')
351
352 subplot(6,1,6);
353 plot(tempV,th6EQ_dot)
354 title('vel theta6')
355
356 %% Plot Acelera o
357 figure
358 subplot(6,1,1);
359 plot(tempV,th1EQ_dot2)
360 title('acc theta1')
361
362 subplot(6,1,2);
363 plot(tempV,th2EQ_dot2)
364 title('acc theta2')
365
366 subplot(6,1,3);

```



```
367 plot(tempV, th3EQ_dot2)
368 title('acc theta3')
369
370 subplot(6,1,4);
371 plot(tempV, th4EQ_dot2)
372 title('acc theta4')
373
374 subplot(6,1,5);
375 plot(tempV, th5EQ_dot2)
376 title('acc theta5')
377
378 subplot(6,1,6);
379 plot(tempV, th6EQ_dot2)
380 title('acc theta6')
```

## 12.4 Esforços

### 12.4.1 PPPR

```
1 %% Calculo dos Esforços
2 % Rodar Trajetoria e Dinamica Primeiro
3
4 T1 = zeros(1,length(d2_dot2V));
5 T2 = zeros(1,length(d2_dot2V));
6 T3 = zeros(1,length(d2_dot2V));
7 T4 = zeros(1,length(d2_dot2V));
8
9 for m = 1:1:length(d2_dot2V)
10     T2(m)=eval(subs(Tau(2),d2_dot2,d2_dot2V(m)));
11     T3(m)=eval(subs(Tau(3),d3_dot2,d3_dot2V(m)));
12 end
13
14 %% Plot Esforços
15 figure
16 subplot(2,1,1);
17 plot(tempV,T2)
18 title('T d2')
19
20 subplot(2,1,2);
21 plot(tempV,T3)
22 title('T d3')
```

## 12.4.2 RRRRRR

```
1 %% Calculo dos Esforços
2 % Rodar Trajetoria e Dinamica Primeiro
3
4 T1 = zeros(1,length(th1EQ_dot));
5 T2 = zeros(1,length(th1EQ_dot));
6 T3 = zeros(1,length(th1EQ_dot));
7 T4 = zeros(1,length(th1EQ_dot));
8 T5 = zeros(1,length(th1EQ_dot));
9 T6 = zeros(1,length(th1EQ_dot));
10
11 for m = 1:length(th1EQ_dot)
12     T1(m)=eval(subs(Tau(1),[th1,th2,th3,th4,th5,th6,th1_dot,th2_dot,th3_dot,
13         th4_dot,th5_dot,th6_dot,th1_dot2,th2_dot2,th3_dot2,th4_dot2,th5_dot2,
14         th6_dot2],[animation(m,1),animation(m,2),animation(m,3),animation(m,
15         4),animation(m,5),animation(m,6),th1EQ_dot(m),th2EQ_dot(m),th3EQ_dot
16         (m),th4EQ_dot(m),th5EQ_dot(m),th6EQ_dot(m),th1EQ_dot2(m),th2EQ_dot2(m)
17         ),th3EQ_dot2(m),th4EQ_dot2(m),th5EQ_dot2(m),th6EQ_dot2(m)]));
18     T2(m)=eval(subs(Tau(2),[th1,th2,th3,th4,th5,th6,th1_dot,th2_dot,th3_dot,
19         th4_dot,th5_dot,th6_dot,th1_dot2,th2_dot2,th3_dot2,th4_dot2,th5_dot2,
20         th6_dot2],[animation(m,1),animation(m,2),animation(m,3),animation(m,
21         4),animation(m,5),animation(m,6),th1EQ_dot(m),th2EQ_dot(m),th3EQ_dot
22         (m),th4EQ_dot(m),th5EQ_dot(m),th6EQ_dot(m),th1EQ_dot2(m),th2EQ_dot2(m)
23         ),th3EQ_dot2(m),th4EQ_dot2(m),th5EQ_dot2(m),th6EQ_dot2(m)]));
24     T3(m)=eval(subs(Tau(3),[th1,th2,th3,th4,th5,th6,th1_dot,th2_dot,th3_dot,
25         th4_dot,th5_dot,th6_dot,th1_dot2,th2_dot2,th3_dot2,th4_dot2,th5_dot2,
26         th6_dot2],[animation(m,1),animation(m,2),animation(m,3),animation(m,
27         4),animation(m,5),animation(m,6),th1EQ_dot(m),th2EQ_dot(m),th3EQ_dot
28         (m),th4EQ_dot(m),th5EQ_dot(m),th6EQ_dot(m),th1EQ_dot2(m),th2EQ_dot2(m)
29         ),th3EQ_dot2(m),th4EQ_dot2(m),th5EQ_dot2(m),th6EQ_dot2(m)]));
30     T4(m)=eval(subs(Tau(4),[th1,th2,th3,th4,th5,th6,th1_dot,th2_dot,th3_dot,
31         th4_dot,th5_dot,th6_dot,th1_dot2,th2_dot2,th3_dot2,th4_dot2,th5_dot2,
32         th6_dot2],[animation(m,1),animation(m,2),animation(m,3),animation(m,
33         4),animation(m,5),animation(m,6),th1EQ_dot(m),th2EQ_dot(m),th3EQ_dot
34         (m),th4EQ_dot(m),th5EQ_dot(m),th6EQ_dot(m),th1EQ_dot2(m),th2EQ_dot2(m)
35         ),th3EQ_dot2(m),th4EQ_dot2(m),th5EQ_dot2(m),th6EQ_dot2(m)]));
36     T5(m)=eval(subs(Tau(5),[th1,th2,th3,th4,th5,th6,th1_dot,th2_dot,th3_dot,
37         th4_dot,th5_dot,th6_dot,th1_dot2,th2_dot2,th3_dot2,th4_dot2,th5_dot2,
38         th6_dot2],[animation(m,1),animation(m,2),animation(m,3),animation(m,
39         4),animation(m,5),animation(m,6),th1EQ_dot(m),th2EQ_dot(m),th3EQ_dot
40         (m),th4EQ_dot(m),th5EQ_dot(m),th6EQ_dot(m),th1EQ_dot2(m),th2EQ_dot2(m)
41         ),th3EQ_dot2(m),th4EQ_dot2(m),th5EQ_dot2(m),th6EQ_dot2(m)]));
42     T6(m)=eval(subs(Tau(6),[th1,th2,th3,th4,th5,th6,th1_dot,th2_dot,th3_dot,
43         th4_dot,th5_dot,th6_dot,th1_dot2,th2_dot2,th3_dot2,th4_dot2,th5_dot2,
44         th6_dot2],[animation(m,1),animation(m,2),animation(m,3),animation(m,
45         4),animation(m,5),animation(m,6),th1EQ_dot(m),th2EQ_dot(m),th3EQ_dot
46         (m),th4EQ_dot(m),th5EQ_dot(m),th6EQ_dot(m),th1EQ_dot2(m),th2EQ_dot2(m)
47         ),th3EQ_dot2(m),th4EQ_dot2(m),th5EQ_dot2(m),th6EQ_dot2(m)]));
48 end
```

```

19
20 %% Plot Esfor os
21 figure
22 subplot(6,1,1);
23 plot(time,T1)
24 title('T theta1')
25
26 subplot(6,1,2);
27 plot(time,T2)
28 title('T theta2')
29
30 subplot(6,1,3);
31 plot(time,T3)
32 title('T theta3')
33
34 subplot(6,1,4);
35 plot(time,T4)
36 title('T theta4')
37
38 subplot(6,1,5);
39 plot(time,T5)
40 title('T theta5')
41
42 subplot(6,1,6);
43 plot(time,T6)
44 title('T theta6')

```

## 13 Referência Bibliográfica

### References

- [1] Mark W. Spong — Seth Hutchinson — M. Vidyasagar. *Robot Modeling and Control*, volume 1. Wiley, 2006.