

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
DEPARTAMENTO ACADÊMICO DE ELETRÔNICA
CURSO DE ENGENHARIA ELETRÔNICA

BRUNO SCHUAVAB PENHA

**MODELAGEM E SIMULAÇÃO DE UM ROBÔ COM 6 GRAUS DE LIBERDADE
UTILIZANDO MATLAB® E SIMULINK®**

TRABALHO DE CONCLUSÃO DE CURSO

CAMPO MOURÃO
2016

BRUNO SCHUAVAB PENHA

**MODELAGEM E SIMULAÇÃO DE UM ROBÔ COM 6 GRAUS DE LIBERDADE
UTILIZANDO MATLAB® E SIMULINK®**

Trabalho de Conclusão de Curso apresentado na disciplina de Trabalho de Conclusão de Curso - TCC, do curso superior de Engenharia Eletrônica do Departamento Acadêmico de Eletrônica – DAELN da Universidade Tecnológica Federal do Paraná, como requisito parcial para obtenção do título de Engenheiro Eletrônico.

Orientador: Flávio Luiz Rossini

CAMPO MOURÃO
2016

TERMO DE APROVAÇÃO
DO TRABALHO DE CONCLUSÃO DE CURSO INTITULADO
MODELAGEM E SIMULAÇÃO DE UM ROBÔ COM 6 GRAUS DE
LIBERDADE UTILIZANDO MATLAB® E SIMULINK®

por

Bruno Schuavab Penha

Trabalho de Conclusão de Curso apresentado no dia 21 de Novembro de 2016 ao Curso Superior de Engenharia Eletrônica da Universidade Tecnológica Federal do Paraná, Campus Campo Mourão. O Candidato foi arguido pela Banca Examinadora composta pelos professores abaixo assinados. Após deliberação, a Banca Examinadora considerou o trabalho Aprovado.

Prof. Gilson Junior Schiavon
(UTFPR)

Prof. Osmar Tormena Junior
(UTFPR)

Prof. Flávio Luiz Rossini
(UTFPR)
Orientador

RESUMO

Este trabalho trata sobre a modelagem e estudo de um robô manipulador de seis graus de liberdade com juntas rotativas, sendo três para posicionamento e três para orientação. Assim, objetivou-se aplicar e analisar técnicas já existentes para obtenção das cinemáticas direta e inversa, além da descrição dinâmica do mesmo. Para auxiliar nos cálculos e manipulações foi utilizado o *software* Matlab®. A representação virtual do manipulador foi desenvolvida no *software* SolidWorks®, e posteriormente realizou-se a interação entre os dois *softwares* através do *toolbox* SimMechanics™. Os resultados obtidos são mostrados visualmente através de gráficos e simulação, os quais apresentaram uma boa correspondência entre os resultados obtidos e o esperado.

Palavras-chave: Cinemática direta e inversa, Modelagem dinâmica, Robô 6-DOF, Controle não linear de robô manipulador.

ABSTRACT

This project is about the modeling and study of a manipulator robot of six degrees of freedom with rotary joints, where three joints are for positioning and three for guidance. Therefore, the objective was to apply and realize the analysis of existent techniques for the obtainment of the direct and inverse cinematics, and the description of the same dynamics. With the backing of the software Matlab®, was possible to compute the equations and realize the virtual representation of the manipulator, at first built utilizing the software SolidWorks®, and afterwards, was developed the interaction between both of them utilizing the toolbox SimMechanics™. The results obtained were shown in graphics and simulations, obtaining a good correspondence between the obtained results and the expected ones.

Keywords: direct and inverse Kinematic, Dynamic modeling, Robot 6-DOF, Nonlinear control of manipulators.

LISTA DE FIGURAS

Figura 1 - Vetor que relaciona o ponto P ao sistema {0} Fonte: autoria própria	13
Figura 2 - Decomposição dos eixos Fonte: Autoria própria.....	14
Figura 3 - Vetor que relaciona origem de {1} com {0} Fonte: autoria própria	14
Figura 4 - Figura para exemplificar a Jacobiana Fonte: autoria própria	17
Figura 5 - Figura pra exemplificar velocidade angular Fonte: autoria própria	18
Figura 6 - Algoritmo para geração de trajetória Fonte: autoria própria	26
Figura 7 - Designer do manipulador proposto Fonte: autoria própria.....	28
Figura 8 - Sistemas de referência no manipulador Fonte: autoria própria	29
Figura 9 - Sistemas de referência Fonte: autoria própria.....	29
Figura 10 - Fluxograma para encontrar ângulo da primeira junta Fonte: Autoria própria	36
Figura 11 - Representação bidimensional do manipulador Fonte: Autoria própria	37
Figura 12 - Relações trigonométricas no manipulador Fonte: Autoria própria.....	38
Figura 13 - Fluxograma para encontrar o ângulo da terceira junta Fonte: Autoria própria	39
Figura 14 - Interface <i>Software SolidWorks®</i> Fonte: Autoria Própria.....	44
Figura 15 - Utilização <i>toolbox SimMechanics™</i> Fonte: Autoria Própria	44
Figura 16 - Diagrama de simulação Fonte: <i>Software Simulink®</i>	45
Figura 17 - Bloco <i>Bode</i> Fonte: <i>Software Simulink®</i>	45
Figura 18 - Parâmetros do bloco <i>Bode</i> Fonte: <i>Software Simulink®</i>	46
Figura 19 - Bloco <i>Revolute</i> Fonte: <i>software Simulink®</i>	46
Figura 20 - Parâmetros do bloco <i>Revolute</i> Fonte: <i>Software Simulink®</i>	47
Figura 21 - Diagrama do sistema simulado Fonte: <i>Software Simulink®</i>	47
Figura 22 - Subsistema Comandos Fonte: <i>Software Simulink®</i>	48
Figura 23 - Gerador de tempo de trajetória Fonte: <i>Software Simulink®</i>	49
Figura 24 - Parâmetros do bloco <i>Integrator</i> Fonte: <i>Software Simulink®</i>	49
Figura 25 - Diagrama de Controle Fonte: <i>Software Simulink®</i>	50
Figura 26 - Diagrama do atuador Fonte: <i>Software Simulink®</i>	50
Figura 27 - Parâmetros do bloco <i>Joint Actuator</i> Fonte: <i>Software Simulink®</i>	51
Figura 28 - Interface criada Fonte: <i>Software Matlab®</i>	52
Figura 29 - Bloco <i>Machine Environment</i> Fonte: <i>Software Simulink®</i>	53
Figura 30 - Parametros do bloco <i>Machine Environment</i> Fonte: <i>software Simulink®</i>	54
Figura 31 - Diagrama de controle genérico Fonte: Autoria própria	54
Figura 32 - Bloco <i>PID Controller</i> Fonte: <i>Software Simulink®</i>	55
Figura 33 - Parâmetros do bloco <i>PID Controller</i> Fonte: <i>Software Simulink®</i>	56
Figura 34 - Representação tridimensional do movimento do robô Fonte: Autoria própria	58
Figura 35 - Nuvem de pontos de singularidade Fonte: autoria própria	59
Figura 36 - Geração de trajetórias da primeira junta Fonte: Autoria própria	61
Figura 37 - Geração de trajetórias da segunda junta Fonte: Autoria própria	61
Figura 38 - Geração de trajetórias da terceira junta Fonte: Autoria própria.....	62
Figura 39 - Geração de trajetórias da quarta junta Fonte: Autoria própria.....	62
Figura 40 - Geração de trajetórias da quinta junta Fonte: Autoria própria.....	63
Figura 41 - Geração de trajetórias da sexta junta Fonte: Autoria própria	63

Figura 42 - Geração de trajetória no espaço cartesiano Fonte: Autoria própria.....	64
Figura 43 - Fluxograma para simulação da jacobiana Fonte: Autoria própria.....	65
Figura 44 - Geração de trajetória com cinemática direta e jacobiana Fonte: Autoria própria .	66
Figura 45 - Bloco <i>Joint Sensor</i> Fonte: <i>Software Simulink®</i>	67
Figura 46 - Parâmetros do bloco <i>Joint Sensor</i> Fonte: <i>Software Simulink®</i>	68
Figura 47 - Comportamento temporal da junta 1 Fonte: Autoria própria	69
Figura 48 - Comportamento temporal da junta 2 Fonte: Autoria própria	70
Figura 49 - Comportamento temporal da junta 3 Fonte: Autoria própria	71
Figura 50 - Comportamento temporal da junta 4 Fonte: Autoria própria	72
Figura 51 - Comportamento temporal da junta 5 Fonte: Autoria própria	73
Figura 52 - Comportamento temporal da junta 6 Fonte: Autoria própria	74
Figura 53 - Resposta do sistema para entrada em rampa Fonte: Autoria própria	76
Figura 54 - Resposta do sistema para entrada senoidal Fonte: Autoria própria	77

LISTA DE ABREVEATURAS, SIGLAS OU ACRÔNOMOS

SR	Sistema de referência
DOF	Degree of freedom

LISTA DE SÍMBOLOS

s_1	$\sin \theta_1$
s_2	$\sin \theta_2$
s_3	$\sin \theta_3$
s_4	$\sin \theta_4$
s_5	$\sin \theta_5$
s_6	$\sin \theta_6$
c_1	$\cos \theta_1$
c_2	$\cos \theta_2$
c_3	$\cos \theta_3$
c_4	$\cos \theta_4$
c_5	$\cos \theta_5$
c_6	$\cos \theta_6$
{1}	Sistema de referência 1
{2}	Sistema de referência 2
{3}	Sistema de referência 3
{4}	Sistema de referência 4
{5}	Sistema de referência 5
{6}	Sistema de referência 6

SUMÁRIO

1 INTRODUÇÃO.....	9
1.1 OBJETIVOS.....	10
1.1.1 Objetivos Gerais	10
1.1.2 Objetivos Específicos	10
1.2 JUSTIFICATIVA.....	10
2 FUNDAMENTAÇÃO TEÓRICA	12
2.1 DESCRIÇÃO ESPACIAL E TRANSFORMAÇÕES	12
2.1.1 Descrição de uma posição	12
2.1.2 Descrição de uma orientação	13
2.1.3 Descrição dos sistemas de referência	15
2.2 CINEMÁTICA DIRETA	15
2.3 CINEMÁTICA INVERSA	16
2.4 JACOBIANO	17
2.4.1 Velocidade linear e angular	17
2.4.2 Propagação das velocidades de um robô	19
2.4.3 Matriz jacobiana	20
2.5 DINÂMICA	20
2.5.1 Inércia	20
2.5.1.1 Tensor de inércia.....	21
2.5.2 Formulação de Lagrange	22
2.5.3 Newton-Euler	23
2.6 GERAÇÃO DE TRAJETÓRIA	24
2.7 CONTROLE NÃO LINEAR	26
2.7.1 Controle de posição	26
3 METODOLOGIA	28
3.1 MODELO PROPOSTO	28
3.2 SISTEMAS DE REFERÊNCIA.....	29
3.3 CINEMÁTICA DIRETA	30
3.4 CINEMÁTICA INVERSA	32
3.5 GERAÇÃO DE TRAJETÓRIA	40
3.6 MATRIZ JACOBIANA DO MANIPULADOR	42
3.7 SOLIDWORKS.....	43

3.8 MATLAB	45
3.8.1 Simulink	45
3.8.2 Guide	51
3.9 CONFIGURAÇÃO DA SIMULAÇÃO PARA ANÁLISE DINÂMICA	53
3.10 CONTROLE.....	54
4 SIMULAÇÃO.....	57
4.1 PARÂMETROS DE SIMULAÇÃO	57
4.2 CINEMÁTICA DIRETA	57
4.3 CINEMÁTICA INVERSA	59
4.4 GERAÇÃO DE TRAJETÓRIA	60
4.5 SIMULAÇÃO DA JACOBIANA.....	65
4.6 SIMULAÇÃO DINÂMICA.....	67
4.7 CONTROLE.....	75
5 CONCLUSÃO	78
REFERÊNCIAS.....	79
APÊNDICE 01.....	81
APÊNDICE 02.....	82
APÊNDICE 03.....	84
APÊNDICE 04.....	87
APÊNDICE 05.....	89
APÊNDICE 06.....	92

1 INTRODUÇÃO

Nos tempos modernos, a palavra *robot* foi utilizada pela primeira vez em 1921, quando o escritor tcheco Karel Capek (1890-1938) estreou no teatro nacional de Praga sua obra *Rossum's Universal Robot (R.U.R.)*. Esta, por sua vez, origina-se do termo *robota*, palavra sérvia, que em russo significa “trabalho forçado”.

Segundo Ullrich (1987) robô é um equipamento multifuncional e reprogramável, projetado para movimentar materiais, peças, ferramentas ou dispositivos, especializados por meio de movimentos variáveis e programados, para a execução de diversas tarefas.

Ainda, segundo o mesmo autor, os robôs podem ser equipados com sensores para sentir ou perceber calor, pressão, impulsos elétricos e objetos; sendo usualmente associados com sistemas de visão computacional, os quais podem monitorar as suas tarefas. Outra função, é de aprender e se lembrar das tarefas, reagir ao seu ambiente de trabalho, operar outras máquinas e se comunicar quando ocorrem problemas em seu funcionamento.

Os robôs mais utilizados na indústria, são os manipuladores robóticos capazes de serem definidos como um conjunto de corpo rígido interligados em série por juntas rotativas ou prismáticas (ARAÚJO, 2010).

Essas interligações formam uma cadeia cinemática aberta, na qual uma das extremidades do braço robótico, se encontra fixado a uma base, enquanto a oposta possui o órgão terminal onde pode-se mover livremente no espaço (LOPES, 2002). Essas máquinas são utilizadas em várias atividades tais como: Paletização de materiais, deslocamento de cargas, pintura, soldagem, etc.

De acordo com Silva (1999), na indústria, os robôs desempenham as mais variadas tarefas. Estas atividades, na sua maioria exigem muita atenção, precisão, além de outras repetitivas ou até mesmo as entediantes, além de outras atividades como as de alto risco. Contudo, uma das aplicações mais comuns dos robôs, e foi abordada nesse trabalho, é para utilização em linhas de montagem, onde este mostra maior eficiência quando comparado a um operador humano.

Dado a importância do robô na indústria, o presente trabalho visou a modelagem cinemática de um robô manipulador de seis graus de liberdade, sendo todas rotativas, além do

mais, as três primeiras são para posicionamento espacial e as três últimas para orientação da ferramenta. Construíram-se rotinas de programação nos *softwares* Matlab® e Simulink®, as quais constataam e auxiliam na análise dos movimentos, posições e orientações do robô.

1.1 OBJETIVOS

1.1.1 Objetivos Gerais

Projetar, modelar e simular um robô, com juntas rotativas e redundantes.

1.1.2 Objetivos Específicos

- Desenvolver o projeto mecânico do robô no ambiente virtual;
- Obter uma representação equivalente do robô;
- Levantar a equação cinemática direta do mecanismo;
- Construir a cinemática inversa;
- Elaborar o equacionamento dinâmico do robô;
- Estabelecer uma proposta de controle para sistema não linear do robô;
- Construir a simulação nos *softwares* Matlab® e Simulink®;
- Analisar os resultados da simulação.

1.2 JUSTIFICATIVA

O uso de robôs manipuladores tem aumentado de maneira impressionante, sendo cada vez mais comum seu emprego em diversas finalidades:

"Em 2014, as vendas de robôs aumentaram 29 %, para 229.261 unidades, de longe, o maior nível já registrado por um ano, até então. As vendas de robôs industriais para todas as indústrias aumentaram em relação a 2013. Os fornecedores de peças automotivas e indústria elétrica eletrônica foram os principais motores do crescimento" (*INTERNATIONAL FEDERATION OF ROBOTICS*, 2016).

Não se pode deixar de perceber que a automação de processos tem ampliado cada vez mais espaço no mercado, e de acordo com artigo publicado pela Exame (INDÚSTRIA 4.0, 2016), dessa maneira é possível uma produção em rede mais precisa, de baixo custo, a qual permite personalizações em massa com maior velocidade. Para poder competir globalmente, a indústria nacional deve aumentar sua produtividade e diminuir custos, para que assim, consolide-se cada vez mais sua presença no mercado mundial.

A partir das informações relatadas, sobre a "indústria 4.0", fica evidente que a utilização de robôs manipuladores, é uma solução possível para que possa ser reduzida a operação manual nesse processo de fabricação e, conseqüentemente ter ganhos de produtividade.

Como a utilização de novos dispositivos automáticos, é algo essencial, principalmente no que tange o fato, de que quanto menos operadores humanos existirem na fabricação, reduzem-se as chances de falhas, o que torna o processo mais confiável, além de aumentar o seu tempo de vida útil.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 DESCRIÇÃO ESPACIAL E TRANSFORMAÇÕES

A descrição espacial é utilizada com o intuito de especificar algumas características de um determinado objeto, como por exemplo, a posição em que ele se encontra no espaço, ou a rotação em relação a outro corpo e/ou sistema. No universo da robótica essa descrição é algo indispensável, pois possibilita um conhecimento mais profundo do sistema como um todo (MAJA, 2014). A ação de cada parte influenciará na operação final do manipulador.

2.1.1 Descrição de uma posição

Segundo Paul (1981), para se realizar a descrição da posição de um corpo é necessário atribuir a este um Sistema de referência (SR), a qual considera a regra da mão direita. Feito isso, é possível localizar esse sistema em qualquer lugar do espaço, por meio de um vetor de dimensão 3×1 . Neste trabalho utilizou-se a notação 0P , para um vetor o qual indica um determinado ponto em relação a um sistema de referência $\{0\}$, dado pela expressão:

$${}^0P = \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} \quad (2.1)$$

A representação gráfica do vetor 0P , o qual indica a posição do ponto qualquer no espaço, é mostrada na Figura 1.

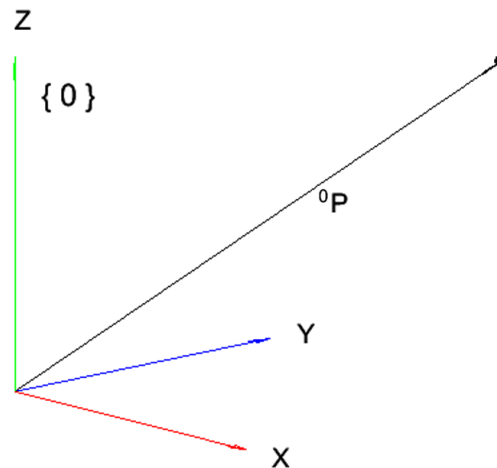


Figura 1 - Vetor que relaciona o ponto P ao sistema {0}
Fonte: autoria própria

2.1.2 Descrição de uma orientação

Outra notação de suma importância é 0P_1 , que descreve a posição da origem do sistema de referência {1}, em relação ao sistema {0}. Este é apresentado pela Equação (2.2) e disposto na Figura 2.

$${}^0P_1 = \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} \quad (2.2)$$

Além da localização do corpo também é necessária sua orientação, ou seja, não só onde, mas como ele está orientado no espaço (CRAIG, 2005).

A rotação pura do sistema {1} em relação ao sistema {0} é expressa por (2.3):

$${}^0R_1 = \begin{bmatrix} \hat{X}_1 \hat{X}_0 & \hat{Y}_1 \hat{X}_0 & \hat{Z}_1 \hat{X}_0 \\ \hat{X}_1 \hat{Y}_0 & \hat{Y}_1 \hat{Y}_0 & \hat{Z}_1 \hat{Y}_0 \\ \hat{X}_1 \hat{Z}_0 & \hat{Y}_1 \hat{Z}_0 & \hat{Z}_1 \hat{Z}_0 \end{bmatrix} \quad (2.3)$$

Onde $\hat{X}_1 \hat{X}_0$ é a projeção do eixo \hat{X}_1 no eixo \hat{X}_0 , funciona de maneira análoga para os outros elementos da matriz. A Figura 2 mostra a rotação do SR {1} em relação a {0}.

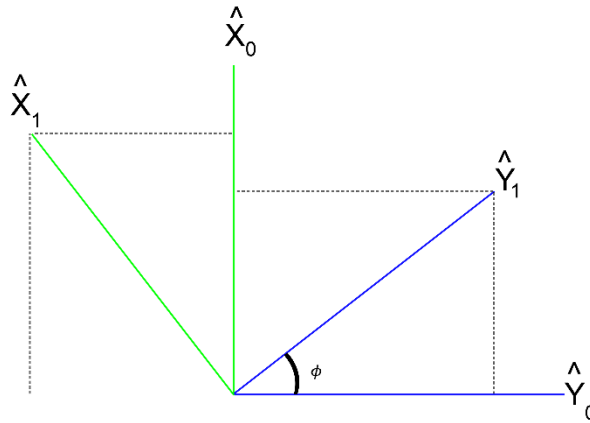


Figura 2 - Decomposição dos eixos
 Fonte: Autoria própria

Para este caso obtém-se a matriz de rotação da equação abaixo:

$${}^0R_1 = \begin{bmatrix} c_\phi & s_\phi & 0 \\ -s_\phi & c_\phi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.4)$$

E de forma mais geral, é mostrado da Figura 3 o SR além de rotacionado também transladado.

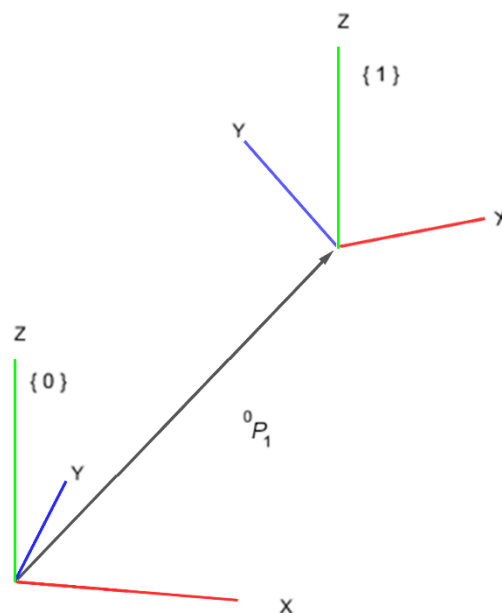


Figura 3 - Vetor que relaciona origem de {1} com {0}
 Fonte: autoria própria

Devido a ortogonalidade e ortonormalidade dos sistemas de referência, existe uma propriedade útil que auxilia nos cálculos, e é mostrada na Equação (2.5) (MAJA, 2014).

$${}^0R_1 = {}^1R_0^{-1} = {}^1R_0^T \quad (2.5)$$

2.1.3 Descrição dos sistemas de referência

A partir da matriz rotacional e do vetor de posição de um sistema, em relação a referência fixa, de acordo com MAJA (2014) é possível descrever, geometricamente, o segundo SR em relação a referência. Para obter tal descrição, por exemplo, de um ponto P em $\{1\}$ em relação a $\{0\}$ utiliza-se a Equação:

$${}^0P = {}^0R_1 \cdot {}^1P + {}^0P_1 \quad (2.6)$$

De maneira mais geral a transformação de um vetor, descrito em relação ao SR $\{1\}$, o qual é descrito em $\{0\}$, é expressa por:

$${}^0P = {}^0T_1 \cdot {}^1P \quad (2.7)$$

Em que a transformação (2.7) pode ser arranjada da seguinte forma:

$$\begin{bmatrix} {}^0P \\ 1 \end{bmatrix} = \begin{bmatrix} {}^0R_1 & {}^0P_1 \\ \mathbf{0} & 1 \end{bmatrix} \cdot \begin{bmatrix} {}^1P \\ 1 \end{bmatrix} \quad (2.8)$$

Após obter-se a transformação é possível fazer a descrição do sistema de referência, por meio da utilização dessa matriz. É importante citar que quando deseja-se realizar uma associação de transformações, tal constituição deve ser da seguinte ordem:

$${}^0T_3 = {}^0T_1 \cdot {}^1T_2 \cdot {}^2T_3 \quad (2.9)$$

2.2 CINEMÁTICA DIRETA

Como já é conhecido da física, a cinemática é a ciência que estuda o movimento sem se preocupar com o que o causa. Tendo já os conceitos de posição e orientação, descritos anteriormente, é possível obter a modelagem cinemática do robô manipulador (SILVA, 1999). A cinemática direta pode ser obtida através da seguinte transformação:

$${}^0T_n = \begin{bmatrix} r_{11} & r_{12} & r_{13} & p_x \\ r_{21} & r_{22} & r_{23} & p_y \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.10)$$

Sendo n o número de graus de liberdade do robô, a transformação é obtida da forma:

$${}^0T_n = {}^0T_1 \cdot {}^1T_2 \cdot {}^2T_3 \dots {}^{n-1}T_n \quad (2.11)$$

É importante lembrar que essa multiplicação, ao contrário do convencional, deve ser feita da direita para a esquerda, ou seja, associa-se, as duas últimas parcelas da direita de (2.11), o que corresponde dizer, opera-se da ferramenta à base do robô.

Após esse processo tem-se a descrição espacial e estática do robô. De acordo com Souza (2011) a cinemática direta permite calcular as variáveis no espaço cartesiano, a partir das variáveis no espaço de juntas.

2.3 CINEMÁTICA INVERSA

Após encontrar a cinemática direta, é possível descrever o posicionamento e pseudo movimento do robô manipulador, ao considerar pequenas variações das juntas. Para tanto, consideraram-se os ângulos de cada junta, numa interpretação mais prática, busca-se obter o conjunto de ângulos das juntas a partir dos pontos cartesianos, e para isso torna-se necessária a cinemática inversa, pois esta descreve os valores dos ângulos em função da posição no espaço.

Para o desenvolvimento da cinemática inversa, há complicações quanto a sua obtenção, isso ocorre devido a não existência de um método geral fechado, o qual é obtido por equações não lineares. Além disso, não há apenas uma solução possível (SILVA, 1999). O mesmo autor fundamentou-se na análise algébrica, a partir da cinemática direta.

2.4 JACOBIANO

"O jacobiano de qualquer robô manipulador é uma matriz que relaciona as velocidades cartesianas lineares e angulares do elemento terminal do robô, com as velocidades angulares de rotação individuais das respectivas juntas" (SILVA,1999).

2.4.1 Velocidade linear e angular

A velocidade linear é dada pela derivada temporal da posição, então tomou-se como exemplo a Figura 4, em que a velocidade do ponto Q em relação ao $\{1\}$ é expresso pela Equação (2.12).

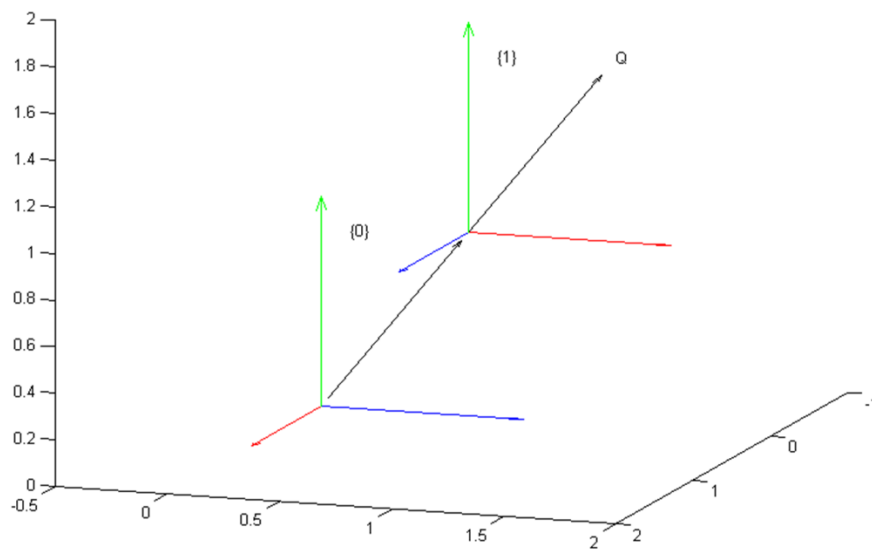


Figura 4 - Figura para exemplificar a Jacobiana
Fonte: autoria própria

$${}^1V_q = \frac{d^1Q}{dt} \quad (2.12)$$

De maneira análoga a velocidade de $\{1\}$ em relação ao $\{0\}$ é dado por:

$${}^0V_1 = \frac{d^0P_1}{dt} \quad (2.13)$$

Similarmente para referir-se a outro ponto de um sistema, para saber a velocidade de um ponto que está descrito no {1} em relação a {0}, torna-se necessário apenas realizar a multiplicação pela matriz rotacional como segue a Equação (2.14) (SPONG, 1989).

$${}^0V_q = {}^0R_1 \frac{d {}^1q}{dt} \quad (2.14)$$

Na Figura 5 mostra-se a velocidade angular ω .

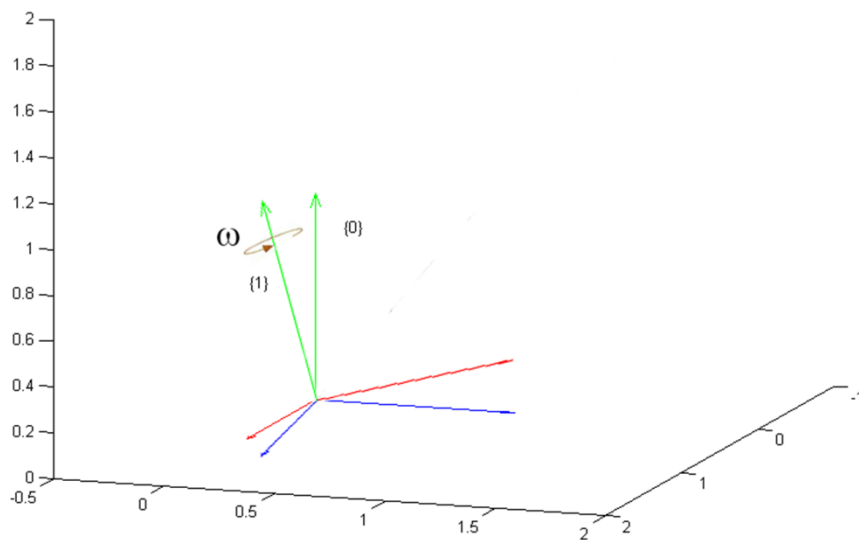


Figura 5 - Figura pra exemplificar velocidade angular
Fonte: autoria própria

Da mesma forma da mecânica escalar, a velocidade angular pode ser obtida através da linear e o raio do trajeto (HIBBELER, 2011), para o caso do manipulador é expressa por:

$${}^1\omega_q = \frac{{}^1V_q}{{}^1P_q} \quad (2.15)$$

2.4.2 Propagação das velocidades de um robô

Um manipulador consiste basicamente em uma série de corpos rígidos unidos entre si, onde cada um destes pode desempenhar um movimento em relação aos outros, descrevendo assim uma velocidade em relação ao seu próprio SR (ARAÚJO, 2010).

A velocidade de um SR depende da velocidade dos sistemas antecessores a ele. A partir disso, observa-se o conceito de propagação de velocidades, para tal descrição, utilizou-se um método iterativo para computar todas as velocidades, lineares e angulares, cujas equações são (2.16) e (2.17) (CRAIG, 2005).

$${}^{i+1}\Omega_{i+1} = {}^{i+1}R_i \, {}^i\Omega_i + \dot{\theta}_{i+1} \, {}^{i+1}\hat{Z}_{i+1} \quad (2.16)$$

$${}^{(i+1)}V_{i+1} = {}^{i+1}R_1 ({}^iV_i + {}^i\Omega_i + {}^iP_{i+1}) \quad (2.17)$$

Algumas observações são indispensáveis:

- Essa formulação pode ser implementada em *software*;
- As iterações iniciam do elo da base até o da ferramenta;
- Para $i = 0$ tem-se as velocidades ${}^0\Omega_0 = 0$ e ${}^0V_0 = 0$;
- \hat{Z} foi adotado como eixo de rotação para todos os SR.

Após encontrado as velocidades ${}^n\Omega_n$ e nV_n , com n sendo o número de graus de liberdade do robô, deve-se obter ${}^0\Omega_n$ e 0V_n , que são, respectivamente, a velocidade angular e linear em relação ao sistema de referência fixo. Para isso é feito a multiplicação pela matriz de rotação (CRAIG, 2005):

$${}^0\Omega_n = {}^0R_n \, {}^n\Omega_n \quad (2.18)$$

$${}^0V_n = {}^0R_n \, {}^nV_n \quad (2.19)$$

2.4.3 Matriz jacobiana

Para encontrar a matriz jacobiana, após obter as velocidades (2.18) e (2.19), é necessário manipular as matrizes de tal maneira a isolar os $\dot{\theta}_1, \dot{\theta}_2, \dots, \dot{\theta}_n$, na Equação (2.20) pode ser vista $J(\theta)$, a qual é chamada jacobiana (SOUZA, 2011).

$$\begin{bmatrix} {}^0V_n \\ {}^0\Omega_n \end{bmatrix} = J(\theta) \begin{bmatrix} \dot{\theta}_1 \\ \vdots \\ \dot{\theta}_n \end{bmatrix} \quad (2.20)$$

2.5 DINÂMICA

A dinâmica é a ciência que estuda o movimento, porém singularmente, atentando-se ao que causa tal movimento (HIBBELER, 2011).

Esta variável é muito importante para a robótica, pois segundo Rosário (2005), um manipulador robótico é um sistema dinâmico com fortes não linearidades. Para esse estudo são analisados posição, velocidade e aceleração das juntas e também a inércias dos elos, o que leva aos próximos tópicos.

2.5.1 Inércia

Segundo Hibbeler (2011), uma força quando aplicada em um corpo pode provocar translação e/ou rotação no mesmo, com o aspecto translacional sendo regido pela Equação (2.21) e o rotacional pela Equação (2.22):

$$F = mA \quad (2.21)$$

$$M = I\alpha \quad (2.22)$$

Para os fins deste trabalho enfatizou-se a Equação (2.22), onde I é chamado de momento de inércia, e é uma medida da resistência do corpo a uma certa aceleração angular (HIBBELER,2011).

2.5.1.1 Tensor de inércia

Para os estudos pertinente a um robô formado por corpos rígidos, torna-se essencial encontrar os tensores de inércia como por exemplo o tensor de um corpo fixado no SR {1} e expresso pela Equação (2.23) (HIBBELER, 2011).

$${}^1I = \begin{bmatrix} {}^1I_{xx} & -{}^1I_{xy} & -{}^1I_{xz} \\ -{}^1I_{yx} & {}^1I_{yy} & -{}^1I_{yz} \\ -{}^1I_{zx} & -{}^1I_{zy} & {}^1I_{zz} \end{bmatrix} \quad (2.23)$$

Sendo os elementos da diagonal principal chamados de momentos de inércia da massa, e o demais são chamados de produto de inércia, e são calculados pelas equações abaixo:

$${}^1I_{xx} = \iiint_V (y^2 + z^2)\rho dV \quad (2.24)$$

$${}^1I_{yy} = \iiint_V (x^2 + z^2)\rho dV \quad (2.25)$$

$${}^1I_{zz} = \iiint_V (x^2 + y^2)\rho dV \quad (2.26)$$

$${}^1I_{xy} = {}^1I_{yx} = \iiint_V xy\rho dV \quad (2.27)$$

$${}^1I_{xz} = {}^1I_{zx} = \iiint_V xz\rho dV \quad (2.28)$$

$${}^1I_{zy} = {}^1I_{yz} = \iiint_V zy\rho dV \quad (2.29)$$

Após definir tais formulações matemáticas é importante considerar alguns conceitos explicados por Hibbeler (2011):

- Se dois eixos do sistema de referência formam um plano de simetria para a distribuição de massa do corpo, os produtos de inércia tendo como índice a coordenada que é normal ao plano de simetria;
- Momentos de inércia sempre são positivos e produtos de inércia podem ter qualquer sinal;
- A soma dos três momentos de inércia é invariante sob mudanças de orientação no sistema de referência;
- Os autovalores de um tensor de inércia são os momentos principais para o corpo e os autovetores associados são os eixos principais.

2.5.2 Formulação de Lagrange

A partir da metodologia apresentada por Spong (1989), no universo do estudo da robótica existem dois métodos principais para cálculo da dinâmica, sendo eles Newton-Euler e a formulação Lagrangiana. O primeiro baseia-se no equilíbrio de forças e o outro na energia do sistema, respectivamente. Na abordagem da formulação Lagrangiana desenvolve-se uma expressão para a energia cinética de cada elo:

$$E = E_v + E_\omega \quad (2.30)$$

Onde E é a energia cinética total do elo, E_v é a energia devido a velocidade linear e E_ω é a energia devido a velocidade angular.

E também se tem a energia potencial dado por:

$$U = -m \cdot {}^0G^T \cdot {}^0P_c + k \quad (2.31)$$

Sendo U a energia potencial do elo, m é a massa do mesmo, 0G é o vetor de gravidade, 0P_c é o vetor que localiza o centro de massa do elo e k é uma constante ajustada arbitrariamente para que o menor valor de U seja zero.

Após calcular as energias de cada elo deve-se somar as energias de todos os elos, para assim ter a energia total do manipulador, da forma:

$$E_t = \sum_{i=1}^n E_i \quad (2.32)$$

$$U_t = \sum_{i=1}^n U_i \quad (2.33)$$

Sendo E_t a energia cinética total, U_t a energia potencial total e n o número total de elos do manipulador.

Encontrado as energias do sistema então é dada a lagrangiana:

$$\mathcal{L}(\Theta, \dot{\Theta}) = E_t(\Theta, \dot{\Theta}) - U_t(\Theta) \quad (2.34)$$

Por fim utiliza-se a Equação (2.35) para encontrar os torques necessários em cada junta, para que o robô desempenhe o movimento.

$$\tau = \frac{d}{dt} \frac{\partial \mathcal{L}}{\partial \dot{\Theta}} - \frac{\partial \mathcal{L}}{\partial \Theta} \quad (2.35)$$

2.5.3 Newton-Euler

Uma alternativa a formulação de lagrange são as equações de Newton e Euler, a seguir encontra-se o algoritmo iterativo de Newton-Euler de acordo com Craig (2005).

Esse método é baseado no equilíbrio das forças atuantes em cada elo do manipulador, e para cada iteração deve-se resolver o conjunto de equações abaixo.

Executa-se esse primeiro conjunto com i variando de 0 até $n - 1$, sendo n o número de juntas do manipulador:

$${}^{i+1}\omega_{i+1} = {}^{i+1}R^i \omega_i + \dot{\theta}_{i+1} {}^{i+1}\hat{Z}_{i+1} \quad (2.36)$$

$${}^{i+1}\dot{\omega}_{i+1} = {}^{i+1}R^i \dot{\omega}_i + {}^{i+1}R^i \omega_i \times \dot{\theta}_{i+1} {}^{i+1}\hat{Z}_{i+1} + \ddot{\theta}_{i+1} {}^{i+1}\hat{Z}_{i+1} \quad (2.37)$$

$${}^{i+1}\dot{v}_{i+1} = {}^{i+1}R^i (\dot{\omega}_i \times {}^i P_{i+1} + \omega_i \times (\omega_i \times {}^i P_{i+1})) + {}^i \dot{v}_i \quad (2.38)$$

$${}^{i+1}\dot{v}_{C_{i+1}} = {}^{i+1}\dot{\omega}_{i+1} \times {}^{(i+1)}P_{C_{i+1}} + {}^{i+1}\omega_{i+1} \times ({}^{i+1}\omega_{i+1} \times {}^{(i+1)}P_{C_{i+1}}) + {}^{i+1}\dot{v}_{i+1} \quad (2.39)$$

$${}^{i+1}F_{i+1} = m_{i+1} {}^{i+1}\dot{v}_{C_{i+1}} \quad (2.40)$$

$${}^{i+1}N_{i+1} = {}^{C_{i+1}}I_{i+1} {}^{i+1}\dot{\omega}_{i+1} + {}^{i+1}\omega_{i+1} \times {}^{C_{i+1}}I_{i+1} {}^{i+1}\omega_{i+1} \quad (2.41)$$

Na sequência calcula-se iterativamente o segundo conjunto variando i de n até 1:

$${}^i f_i = {}^{i+1}R^i {}^{i+1}f_{i+1} + {}^i F_i \quad (2.42)$$

$${}^i n_i = {}^i N_i + {}^{i+1}R^i {}^{i+1}n_{i+1} + {}^i P_{C_i} \times {}^i F_i + {}^i P_{i+1} \times {}^{i+1}R^i {}^{i+1}f_{i+1} \quad (2.43)$$

$$\tau_i = {}^i n_i^T {}^i \hat{Z}_i \quad (2.44)$$

O algoritmo iterativo de Newton-Euler, apresenta uma desvantagem em relação a formulação lagrangeana, as formulas de torque encontradas são relativamente longas, o que torna muitas vezes inviável sua aplicação salvo em um ambiente de simulação computacional.

2.6 GERAÇÃO DE TRAJETÓRIA

A geração de trajetórias baseia-se no modelo geométrico do robô, a qual pode ser implementada tanto no espaço cartesiano quanto no espaço de juntas. No entanto, trabalhar no

espaço de juntas não é muito interessante, salvo alguns casos muito específicos. Assim, será tratado a trajetória apenas no espaço de cartesiano. (ROSÁRIO, 2005).

Toma-se como base que o manipulador tem sua ferramenta em uma determinada posição conhecida, e deseja-se leva-la para outra posição, também conhecida. Para isso torna-se necessário uma combinação de ângulos diferentes da inicial, e também necessita-se de um toque τ para cada junta.

Não se pode negligenciar a necessidade de um movimento suave, para que não haja danos ao robô ou ao meio em que ele se encontra. Assim, define-se uma função que governe esse movimento, tal que, tenha um comportamento mais suave como por exemplo a usual em um polinômio de terceiro grau (ROSÁRIO, 2005).

Após definir o polinômio, ou qualquer outra função que atenda a exigência quanto a suavidade dos movimentos, é possível descrever a variação dos ângulos $\theta_1, \theta_2 \dots \theta_n$ no tempo. Assim então, é descrito um possível algoritmo para geração da trajetória no fluxograma da Figura 5 (ROSÁRIO, 2005).

Vale ressaltar que muitas vezes uma geração de trajetória (Figura 6) simplesmente com os pontos inicial e final, pode resultar em um trajeto impossível de ser percorrido, fazendo com que o manipulador esbarre em algo no seu espaço de trabalho, ou com que ele choque-se em si mesmo. Ao considerar tal problema, torna-se interessante a definição de vários pontos em um determinado trajeto para que o robô possa percorrê-lo (ROSÁRIO, 2005).

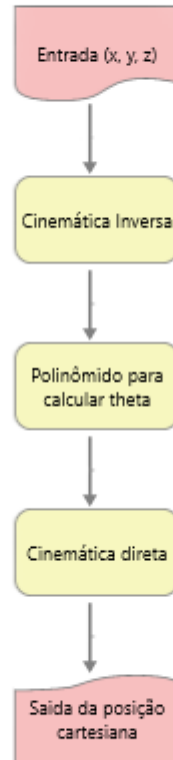


Figura 6 - Algoritmo para geração de trajetória
Fonte: autoria própria

2.7 CONTROLE NÃO LINEAR

2.7.1 Controle de posição

Realizar o controle de um determinado sistema para o qual deseja-se manter a saída deste com o valor desejado (trazer o erro o mais próximo possível de zero) (OGATA, 2010).

Existem inúmeras técnicas para a realização, ou até mesmo apenas auxiliar, no desenvolvimento de um projeto de controle cada uma com suas próprias peculiaridades como por exemplo análise em frequência, Ziegler e Nichols (1942), *root locus*, entre outros.

No entanto, cada uma dessas técnicas citadas anteriormente e outras, baseiam-se em sistemas lineares. Vale ressaltar que um manipulador robótico é um sistema com altas não linearidades o que torna imprescindível a utilização de métodos de controle não lineares (OGATA, 2010).

De acordo com Spong (1989), uma das não linearidades mais estudadas no campo da robótica é o atrito, que pode ser expressado como a soma de duas parcelas. O atrito viscoso mostrado na Equação (2.45) (linear), e o atrito de coulomb que pode ser visualizado na Equação (2.46) (não linear). Para a construção de um modelo mais aproximado é feito a junção das duas na Equação (2.47):

$$\tau_f = F_V \cdot \dot{\theta} \quad (2.45)$$

$$\tau_f = F_S \cdot \text{sgn}(\dot{\theta}) \quad (2.46)$$

$$\tau_f = F_V \cdot \dot{\theta} + F_S \cdot \text{sgn}(\dot{\theta}) \quad (2.47)$$

Existem muitas técnicas de controle não linear. Sua escolha depende muito das especificações do projeto e dos resultados desejados (OGATA, 2010). Portanto a alternativa deste e maiores explicações serão deixadas para o futuro, na seção de desenvolvimento.

3 METODOLOGIA

O trabalho é dividido nos seguintes itens:

- Projetar o robô manipulador, a partir das características das tarefas a serem executadas pelo mesmo;
- Fixar os sistemas de referências;
- Obter a cinemática direta;
- Obter a cinemática inversa;
- Construir a matriz Jacobiana no robô manipulador;
- Realizar análise dinâmica;
- Implementar controlador;
- Simular os movimentos do robô manipulador a partir do *software* MatLab®.

3.1 MODELO PROPOSTO

No presente trabalho foi proposto um robô manipulador com 6 graus de liberdade, sendo todas as juntas rotativas e algumas redundantes, conforme mostrado na Figura 7:

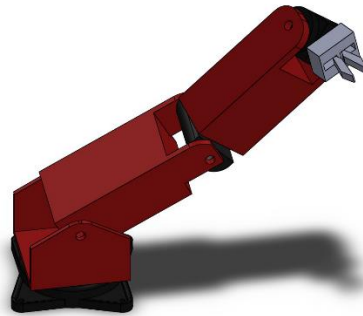


Figura 7 - Designer do manipulador proposto
Fonte: autoria própria

3.2 SISTEMAS DE REFERÊNCIA

Com base no projeto mecânico proposto, usou-se a metodologia para obter a modelagem matemática do manipulador, assim foram fixados sistemas de referências em cada uma das juntas como mostrado na Figura 8.

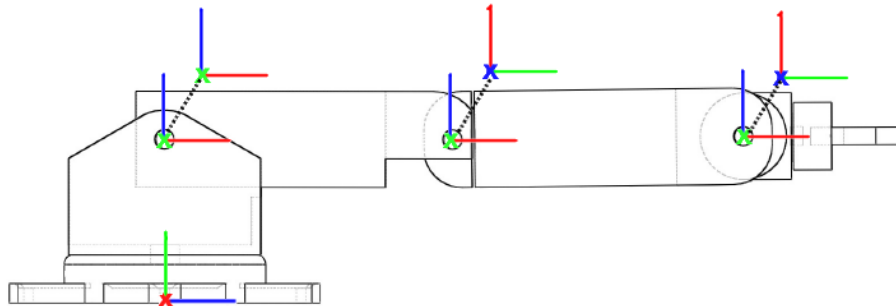


Figura 8 - Sistemas de referência no manipulador
Fonte: autoria própria

Os SRs fora da figura do manipulador possuem as origens coincidentes com outros SRs, com os quais são ligados pelas linhas tracejadas. Para facilitar a visualização, será analisado apenas os sistemas de referências na Figura 9.

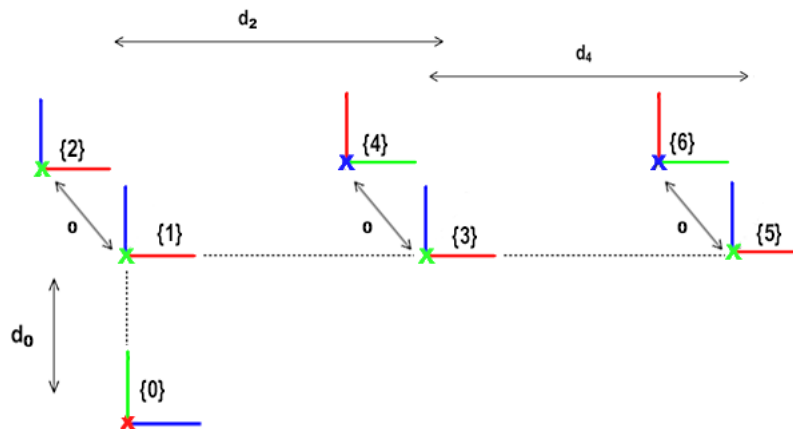


Figura 9 - Sistemas de referência
Fonte: autoria própria

Com os sistemas de referência devidamente fixados no modelo do robô, é possível realizar a modelagem, com o intuito de se obter expressões matemáticas, que descrevem características do movimento de tal robô.

3.3 CINEMÁTICA DIRETA

Tomou-se como base as descrições de posição e orientação dos sistemas de referência para as quais encontraram-se as seguintes transformações:

$${}^0_1T = \begin{bmatrix} c_1 & -s_1 & 0 & 0 \\ s_1 & c_1 & 0 & 0 \\ 0 & 0 & 1 & d_0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.1)$$

$${}^1_2T = \begin{bmatrix} 0 & 0 & 1 & 0 \\ c_2 & -s_2 & 0 & 0 \\ s_2 & c_2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.2)$$

$${}^2_3T = \begin{bmatrix} c_3 & -s_3 & 0 & d_2 \\ s_3 & c_3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.3)$$

$${}^3_4T = \begin{bmatrix} 0 & 0 & 1 & 0 \\ c_4 & -s_4 & 0 & 0 \\ s_4 & c_4 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.4)$$

$${}^4_5T = \begin{bmatrix} s_5 & c_5 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ c_5 & -s_5 & 0 & d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.5)$$

$${}^5_6T = \begin{bmatrix} 0 & 0 & 1 & 0 \\ c_6 & -s_6 & 0 & 0 \\ s_6 & c_6 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.6)$$

A partir de (3.1)-(3.6), constituiu-se a cinemática direta a qual expressa por (3.7)

$${}^0_6T = \begin{bmatrix} r_{11} & r_{12} & r_{13} & p_x \\ r_{21} & r_{22} & r_{23} & p_y \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.7)$$

Sendo:

$$r_{11} = c_1(c_4s_6 + c_5c_6s_4) - s_1(c_2(s_3(s_4s_6 - c_4c_5c_6) - c_3c_6s_5) + s_2(c_3(s_4s_6 - c_4c_5c_6) + c_6s_3s_5)) \quad (3.8)$$

$$r_{12} = c_1(c_4c_6 - c_5s_4s_6) - s_1(c_2(s_3(c_6s_4 + c_4c_5s_6) + c_3s_5s_6) + s_2(c_3(c_6s_4 + c_4c_5s_6) - s_3s_5s_6)) \quad (3.9)$$

$$r_{13} = c_1s_4s_5 - s_1(c_2(c_3c_5 - c_4s_3s_5) - s_2(c_5s_3 + c_3c_4s_5)) \quad (3.10)$$

$$r_{21} = c_1(c_2(s_3(s_4s_6 - c_4c_5c_6) - c_3c_6s_5) + s_2(c_3(s_4s_6 - c_4c_5c_6) + c_6s_3s_5)) + s_1(c_4s_6 + c_5c_6s_4) \quad (3.11)$$

$$r_{22} = c_1(c_2(s_3(c_6s_4 + c_4c_5s_6) + c_3s_5s_6) + s_2(c_3(c_6s_4 + c_4c_5s_6) - s_3s_5s_6)) + s_1(c_4c_6 - c_5s_4s_6) \quad (3.12)$$

$$r_{23} = s_1s_4s_5 - c_1(s_2(c_5s_3 + c_3c_4s_5) - c_2(c_3c_5 - c_4s_3s_5)) \quad (3.13)$$

$$r_{31} = s_2(s_3(s_4s_6 - c_4c_5c_6) - c_3c_6s_5) - c_2(c_3(s_4s_6 - c_4c_5c_6) + c_6s_3s_5) \quad (3.14)$$

$$r_{32} = s_2(s_3(c_6s_4 + c_4c_5s_6) + c_3s_5s_6) - c_2(c_3(c_6s_4 + c_4c_5s_6) - s_3s_5s_6) \quad (3.15)$$

$$r_{33} = s_2(c_3c_5 - c_4s_3s_5) + c_2(c_5s_3 + c_3c_4s_5) \quad (3.16)$$

$$p_x = -s_1(c_2(d_2 + d_4c_3) - d_4s_2s_3) \quad (3.17)$$

$$p_y = c_1(c_2(d_2 + d_4c_3) - d_4s_2s_3) \quad (3.18)$$

$$p_z = d_0 + d_2s_2 + d_4s_{23} \quad (3.19)$$

Com tal descrição pode-se obter a posição e orientação do robô no espaço cartesiano, a partir dos ângulos de cada junta.

3.4 CINEMÁTICA INVERSA

A cinemática inversa descreve o ângulo necessário das juntas, para que o manipulador esteja em uma posição desejada no espaço cartesiano. Para encontrar as expressões de cada ângulo, com uma análise algébrica, utilizaram-se algumas comparações entre as transformações.

Para encontrar a expressão matemática que descreve a posição angular da junta 1 (θ_1), considerou-se na cinemática direta, a transformação 0T_6 e os elementos 1×4 e 2×4 tem-se:

$$p_x = -s_1(c_2d_2 + d_4c_{23}) \quad (3.20)$$

$$p_y = c_1(c_2d_2 + d_4c_{23}) \quad (3.21)$$

Dividiu-se a Equação (3.20) pela (3.21) e realizou-se o arco tangente de ambos os lados para obtenção da Equação (3.22).

$$\theta_1 = \tan^{-1}\left(-\frac{p_x}{p_y}\right) \quad (3.22)$$

Para encontrar a expressão matemática que descreve a posição angular da junta 3 (θ_3) efetuou-se a seguinte comparação:

$${}^0T^{-1}{}^6T = {}^3T \quad (3.23)$$

Usou-se o elemento 2×4 :

$$p_y c_1 - p_x s_1 = d_4 c_{23} + d_2 c_2 \quad (3.24)$$

E também o elemento 3×4 :

$$p_z - d_0 = d_4 s_{23} + d_2 s_2 \quad (3.25)$$

Efetuarão-se os termos ao quadrado e somou-se da forma:

$$\theta_3 = \tan^{-1} \left(\frac{(2d_2 d_4 (1 - k_{13})^{\frac{1}{2}})}{d_0^2 - 2d_0 p_z - d_2^2 - d_4^2 + p_x^2 + p_y^2 + p_z^2} \right) \quad (3.26)$$

Sendo.

$$k_{13} = \frac{(d_0^2 - 2d_0 p_z - d_2^2 - d_4^2 + p_x^2 + p_y^2 + p_z^2)^2}{4d_2^2 d_4^2} \quad (3.27)$$

Encontrou-se a expressão matemática que descreve a posição angular da junta 2 (θ_2) assim utilizou-se a mesma comparação da Equação (3.28), a partir da relação 1×4 :

$$p_z s_{23} - d_0 s_{23} - d_2 c_3 + p_y c_{23} c_1 - p_x c_{23} s_1 = d_4 \quad (3.28)$$

Separou-se s_{23} e c_{23} da seguinte forma:

$$s_{23}(p_z - d_0) + c_{23}(p_y c_1 - p_x s_1) = d_4 + d_2 c_3 \quad (3.29)$$

Arranjou-se:

$$\rho s_\phi = p_y c_1 - p_x s_1 \quad (3.30)$$

$$\rho c_\phi = p_z - d_0 \quad (3.31)$$

Sendo ρ :

$$\rho = \sqrt{(p_y c_1 - p_x s_1)^2 + (p_z - d_0)^2} \quad (3.32)$$

Onde ϕ é dado por:

$$\phi = \tan^{-1} \left(\frac{p_y c_1 - p_x s_1}{p_z - d_0} \right) \quad (3.33)$$

Com isso a Equação (3.29) ficou:

$$s_{23} c_\phi + c_{23} s_\phi = \frac{d_4 + d_2 c_3}{\rho} \quad (3.34)$$

Usou-se a propriedade da soma de ângulos do Seno manipulou-se a Equação (3.34) para a (3.35).

$$s_{23\phi} = \frac{d_4 + d_2 c_3}{\rho} \quad (3.35)$$

Realizou-se o arco tangente chega-se em:

$$\theta_2 = \tan^{-1} \left(\frac{d_4 + d_2 c_3}{\rho} \right) - \theta_3 - \phi \quad (3.36)$$

Para encontrar a expressão matemática que descreve a posição angular da junta 5 (θ_5) fez-se uso da comparação da Equação (3.23) na elemento 1×3 :

$$r_{33} s_{23} + r_{23} c_{23} c_1 - r_{13} c_{23} s_1 = c_5 \quad (3.37)$$

Usou-se a identidade trigonométrica fundamental chega-se em:

$$s_5 = \sqrt{1 - c_5^2} \quad (3.38)$$

Através das propriedades trigonométricas definiu-se a Equação (3.39a).

$$\theta_5 = \tan^{-1} \left(\frac{s_5}{c_5} \right) \quad (3.39a)$$

Substituiu-se as Equações (3.37) e (3.38) na (3.39a).

$$\theta_5 = \tan^{-1} \left(\frac{\sqrt{(1 - (r_{33}s_{23} + r_{23}c_{23}c_1 - r_{13}c_{23}s_1)^2)}}{r_{33}s_{23} + r_{23}c_{23}c_1 - r_{13}c_{23}s_1} \right) \quad (3.39b)$$

Para encontrar a expressão matemática que descreve a posição angular da junta 4 (θ_4) usou-se a comparação da Equação (3.23) o elemento 2×3 :

$$r_{33}c_{23} - r_{23}s_{23}c_1 + r_{13}s_{23}s_1 = c_4s_5 \quad (3.40)$$

E também a Equação (3.41).

$${}^0T^{-1}{}^0T = {}^1T \quad (3.41)$$

Efetuuou-se a comparação 1×3 :

$$r_{13}c_1 + r_{23}s_1 = s_4s_5 \quad (3.42)$$

Dividiu-se a Equação (3.41) pela Equação (3.42):

$$\frac{s_4s_5}{c_4s_5} = \frac{r_{13}c_1 + r_{23}s_1}{r_{33}c_{23} - r_{23}s_{23}c_1 + r_{13}s_{23}s_1} \quad (3.43)$$

Considerou-se $s_5 \neq 0$ e fazendo o arco tangente dos dois lados da Equação (3.43) chegou-se na Equação (3.44).

$$\theta_4 = \tan^{-1} \left(\frac{r_{13}c_1 + r_{23}s_1}{r_{33}c_{23} - r_{23}s_{23}c_1 + r_{13}s_{23}s_1} \right) \quad (3.44)$$

Para encontrar a expressão matemática que descreve a posição angular da junta 6 (θ_6) realizou-se a comparação:

$${}^0T^{-1}{}^0T = {}^5T \quad (3.45)$$

No elemento 3×1 :

$$s_6 = r_{21}(c_4s_1 + c_1c_2s_3s_4 + c_1c_3s_2s_4) - r_{11}(c_2s_1s_3s_4 - c_1c_4 + c_3s_1s_2s_4) - r_{31}c_{23}s_4 \quad (3.46)$$

E o elemento 3×2 :

$$c_6 = r_{22}(c_4s_1 + c_1c_2s_3s_4 + c_1c_3s_2s_4) - r_{12}(c_2s_1s_3s_4 - c_1c_4 + c_3s_1s_2s_4) - r_{32}c_{23}s_4 \quad (3.47)$$

Por fim dividiu-se a Equação (3.47) pela (3.46) chegou-se a θ_6 da seguinte forma:

$$\theta_6 = \tan^{-1}\left(\frac{s_6}{c_6}\right) \quad (3.48)$$

Contudo as funções encontradas não são validas para todos os pontos do espaço de trabalho do manipulador. Isso ocorre devido as não linearidades e pontos de singularidade, isso foi mostrado na seção 4.6.

Apesar de não terem validade para todo o espaço cartesiano, as expressões obtidas podem auxiliar, para encontrar resultados melhores. Para isso foi feita uma combinação de métodos para a cinemática inversa.

Combinou-se a expressão encontrada para θ_1 , com uma análise geométrica, baseada em qual octante a ferramenta se encontra. A lógica utilizada para isso encontra-se na Figura 10.

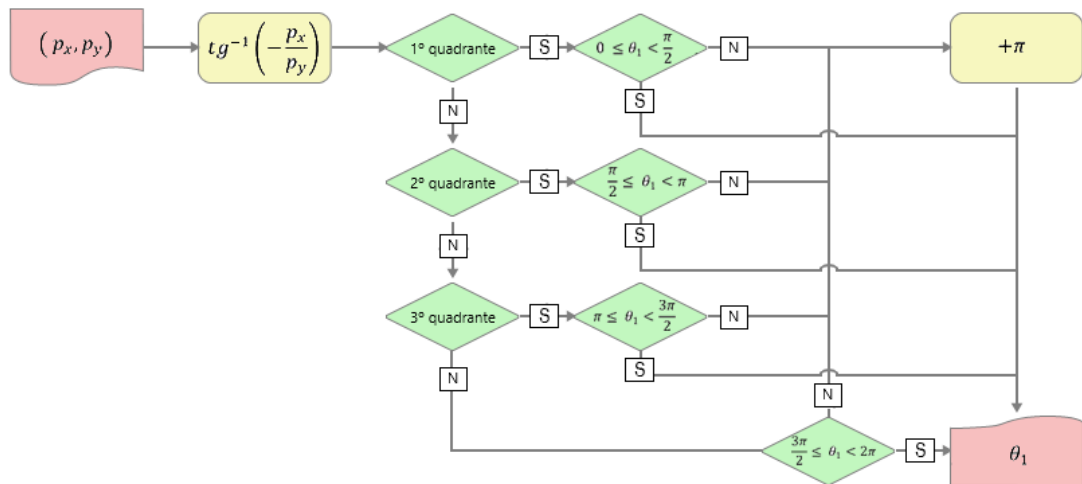


Figura 10 - Fluxograma para encontrar ângulo da primeira junta
Fonte: Autoria própria

O fluxograma da Figura 10 descreve uma análise a partir da Equação (3.22). Por meio das simulações constatou-se que em alguns pontos o ângulo θ_1 apresentou uma variação de π .

Por isso a análise varia por quadrante, e se θ_1 não estiver entre o intervalo esperado é somado π .

A variação que θ_2 demonstrou, não pode ser mapeada para ser corrigida. Então realizou-se uma análise geométrica, obteve-se assim um novo equacionamento para essa junta.

Para isso considerou-se o posicionamento do manipulador sem θ_1 , conforme mostrado na Figura 11.

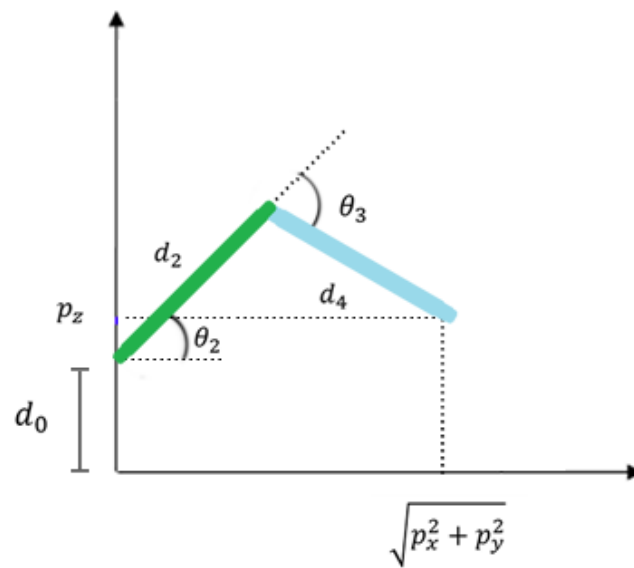


Figura 11 - Representação bidimensional do manipulador
Fonte: Autoria própria

Com isso dividiu-se a Figura 11 em dois triângulos.

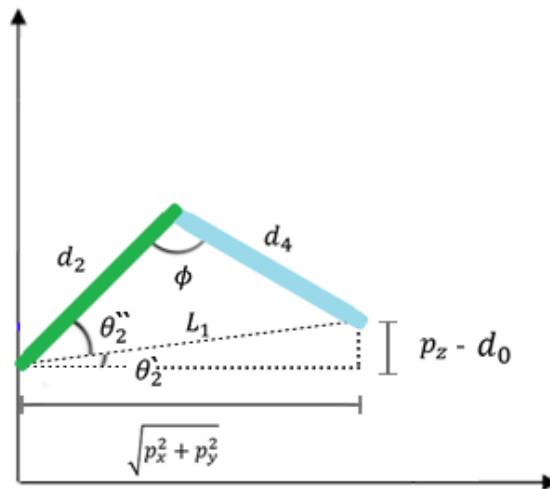


Figura 12 - Relações trigonométricas no manipulador
Fonte: Autoria própria

Usou-se a lei dos senos para encontrar θ_2'' .

$$\frac{\sin(\theta_2'')}{d_4} = \frac{\sin(\phi)}{L_1} \quad (3.49)$$

Isolou-se θ_2'' .

$$\theta_2'' = \sin^{-1}\left(\frac{d_4 \sin(\phi)}{L_1}\right) \quad (3.50)$$

Sendo ϕ :

$$\phi = \pi - \theta_3 \quad (3.51)$$

E L_1 :

$$L_1 = \sqrt{p_x^2 + p_y^2 + (p_z - d_0)^2} \quad (3.52)$$

A partir da relação da tangente encontrou-se θ_2' .

$$\theta_2' = \tan^{-1}\left(\frac{p_z - d_0}{\sqrt{p_x^2 + p_y^2}}\right) \quad (3.53)$$

Por fim determinou-se θ_2 .

$$\theta_2 = \theta_2' + \theta_2'' \quad (3.54a)$$

$$\theta_2 = \sin^{-1} \left(d_4 \frac{\sin(\pi + \theta_3)}{\sqrt{(p_x^2 + p_y^2 + (p_z - d_0)^2)}} \right) + \tan^{-1} \left(\frac{p_z - d_0}{\sqrt{(p_x^2 + p_y^2)}} \right) \quad (3.54b)$$

O termo θ_3 por sua vez, apresentou uma variação no sinal ($\pm\theta_3$). Diferente de θ_1 não foi possível encontrar uma relação direta com a posição onde isso ocorre. Então para resolver esse problema utilizou-se uma lógica, a qual é descrita na Figura 13.

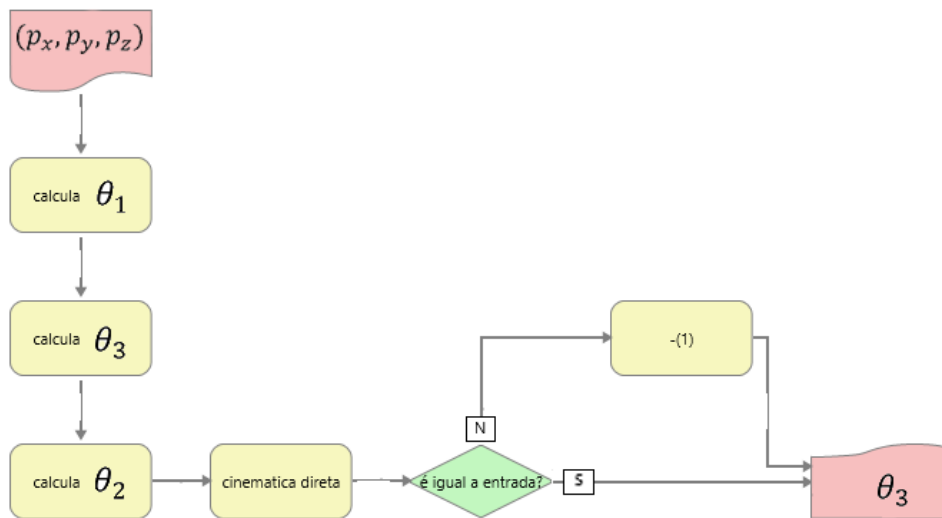


Figura 13 - Fluxograma para encontrar o ângulo da terceira junta
Fonte: Autoria própria

Determinou-se θ_5 , para isso notou-se a orientação dos sistemas de referência. Observou-se a Figura 8, a partir da qual pode-se perceber que independentemente da posição ou orientação do manipulador, o eixo \hat{Z}_6 é colinear com o eixo \hat{X}_5 . Por sua vez, o eixo \hat{X}_5 , inicialmente, é colinear com o eixo \hat{X}_3 . Sendo assim quando houver uma rotação em θ_5 surgirá um ângulo entre \hat{Z}_6 e \hat{X}_3 . Então apenas foi necessário referenciar o efetuador em relação ao SR {3}, isso é feito através da Equação (3.55).

$${}^0R_3^{-1} {}^0R_6 = {}^3R_6 \quad (3.55)$$

Com (3.55) encontrou-se:

$${}^3R_6 = \begin{bmatrix} {}^3r_{11} & {}^3r_{12} & {}^3r_{13} \\ {}^3r_{21} & {}^3r_{22} & {}^3r_{23} \\ {}^3r_{31} & {}^3r_{32} & {}^3r_{33} \end{bmatrix} \quad (3.56)$$

Para obter θ_5 executou-se o arco cosseno do elemento ${}^3r_{13}$.

$$\theta_5 = \cos^{-1}({}^3r_{13}) \quad (3.57)$$

Com a obtenção da cinemática inversa tornou possível, a partir de coordenadas no espaço cartesiano, calcular a posição angular de cada junto do manipulador.

3.5 GERAÇÃO DE TRAJETÓRIA

Para descrever o movimento de forma suave e não ter variações bruscas de torque, para que não haja risco de danificar o mecanismo, a partir de tal restrição, escolheu-se um polinômio de quinto grau para descrever o movimento dos ângulos, conforme mostrado na Equação (3.58a).

$$\theta(t) = a_5t^5 + a_4t^4 + a_3t^3 + a_2t^2 + a_1t + a_0 \quad (3.58a)$$

$$\dot{\theta}(t) = 5a_5t^4 + 4a_4t^3 + 3a_3t^2 + 2a_2t + a_1 \quad (3.58b)$$

$$\ddot{\theta}(t) = 20a_5t^3 + 12a_4t^2 + 6a_3t + a_2 \quad (3.58c)$$

O polinômio em (3.58a) possui seis parâmetros a serem determinados, para encontrá-los definem-se os contornos no problema, com as características iniciais e finais da seguinte forma:

$$\theta(0) = \theta_i \quad (3.59)$$

$$\theta(t_f) = \theta_f \quad (3.60)$$

$$\dot{\theta}(0) = 0 \quad (3.61)$$

$$\dot{\theta}(t_f) = 0 \quad (3.62)$$

$$\ddot{\theta}(0) = 0 \quad (3.63)$$

$$\ddot{\theta}(t_f) = 0 \quad (3.64)$$

Onde θ_i é o ângulo inicial, θ_f é o ângulo final e t_f é o tempo final do trajeto.

Após isso as Equações (3.58) a partir de manipulações chegaram-se nas equações que são mostradas a seguir:

$$a_0 = \theta_i \quad (3.65)$$

$$a_5 t^5 + a_4 t_f^4 + a_3 t_f^3 + a_2 t_f^2 + a_1 t_f + a_0 = \theta_f \quad (3.66)$$

$$a_1 = 0 \quad (3.67)$$

$$5a_5 t^4 + 4a_4 t_f^3 + 3a_3 t_f^2 + 2a_2 t_f + a_1 = 0 \quad (3.68)$$

$$2a_2 = 0 \quad (3.69)$$

$$20a_5 t^3 + 12a_4 t_f^2 + 6a_3 t_f + 2a_2 = 0 \quad (3.70)$$

Como existem seis equações e seis parâmetros a determinar, o sistema é determinado, manipularam-se os parâmetros da seguinte forma:

$$a_0 = \theta_i \quad (3.71)$$

$$a_1 = 0 \quad (3.72)$$

$$a_2 = 0 \quad (3.73)$$

$$a_3 = \frac{10\theta_f}{t_f^3} - \frac{10\theta_i}{t_f^3} \quad (3.74)$$

$$a_4 = \frac{15\theta_i}{t_f^4} - \frac{15\theta_f}{t_f^4} \quad (3.75)$$

$$a_5 = \frac{6\theta_f}{t_f^5} - \frac{6\theta_i}{t_f^5} \quad (3.76)$$

O polinômio para cálculo da geração de trajetória, pode ser definido com o grau que se desejar, com tudo com polinômios de ordem 3 por exemplo, não é possível garantir um torque com partidas e chegadas suaves. Isso pode acarretar falhas na execução de tarefas ou até danos ao manipulador.

3.6 MATRIZ JACOBIANA DO MANIPULADOR

Ao considerar as Equações (2.35) a (2.43) calculou-se a matriz jacobiana do manipulador, a qual é mostrada na Equação (3.77). Realizou-se tais cálculos, em uma rotina no *software* Matlab®, esta encontra-se no Apêndice 3.

$$J(\theta) = \begin{bmatrix} J_{11} & J_{12} & J_{13} & 0 & 0 & 0 \\ J_{21} & J_{22} & J_{23} & 0 & 0 & 0 \\ 0 & J_{32} & J_{33} & 0 & 0 & 0 \\ 0 & c_1 & c_1 & J_{44} & J_{45} & J_{46} \\ 0 & s_1 & s_1 & J_{54} & J_{55} & J_{56} \\ 1 & 0 & 0 & J_{64} & J_{65} & J_{66} \end{bmatrix} \quad (3.77)$$

$$J_{11} = -c_1(d_4c_{23} + d_2c_2) \quad (3.78)$$

$$J_{12} = s_1(d_4s_{23} + d_2s_2) \quad (3.79)$$

$$J_{13} = d_4s_{23}s_1 \quad (3.80)$$

$$J_{21} = -s_1 * (d_4c_{23} + d_2c_2) \quad (3.81)$$

$$J_{22} = -c_1(d_4s_{23} + d_2s_2) \quad (3.82)$$

$$J_{23} = -d_4s_{23}c_1 \quad (3.83)$$

$$J_{32} = d_4 c_{23} + d_2 c_2 \quad (3.84)$$

$$J_{33} = d_4 c_{23} \quad (3.85)$$

$$J_{44} = -c_{23} s_1 \quad (3.86)$$

$$J_{45} = c_1 c_4 - c_2 s_1 s_3 s_4 - c_3 s_1 s_2 s_4 \quad (3.87)$$

$$J_{46} = c_1 s_4 s_5 - c_2 c_3 c_5 s_1 + c_5 s_1 s_2 s_3 + c_2 c_4 s_1 s_3 s_5 + c_3 c_4 s_1 s_2 s_5 \quad (3.88)$$

$$J_{54} = c_{23} c_1 \quad (3.89)$$

$$J_{55} = c_4 s_1 + c_1 c_2 s_3 s_4 + c_1 c_3 s_2 s_4 \quad (3.90)$$

$$J_{56} = s_1 s_4 s_5 + c_1 c_2 c_3 c_5 - c_1 c_5 s_2 s_3 - c_1 c_2 c_4 s_3 s_5 - c_1 c_3 c_4 s_2 s_5 \quad (3.91)$$

$$J_{64} = s_{23} \quad (3.92)$$

$$J_{65} = -c_{23} s_4 \quad (3.93)$$

$$J_{66} = c_2 c_5 s_3 + c_3 c_5 s_2 + c_2 c_3 c_4 s_5 - c_4 s_2 s_3 s_5 \quad (3.94)$$

A matriz jacobiana para este manipulador, resultou em equações relativamente extensas. Com tudo isso pode variar, de acordo com a classe do robô. Quanto mais graus de liberdade e mais juntas redundantes, mais extensas serão as equações.

3.7 SOLIDWORKS

Realizou-se a construção virtual do manipulador, para tal realização escolheu-se o *software* SolidWorks®. E mostrado na Figura 14 a interface inicial do *software* com o desenho do manipulador completo.

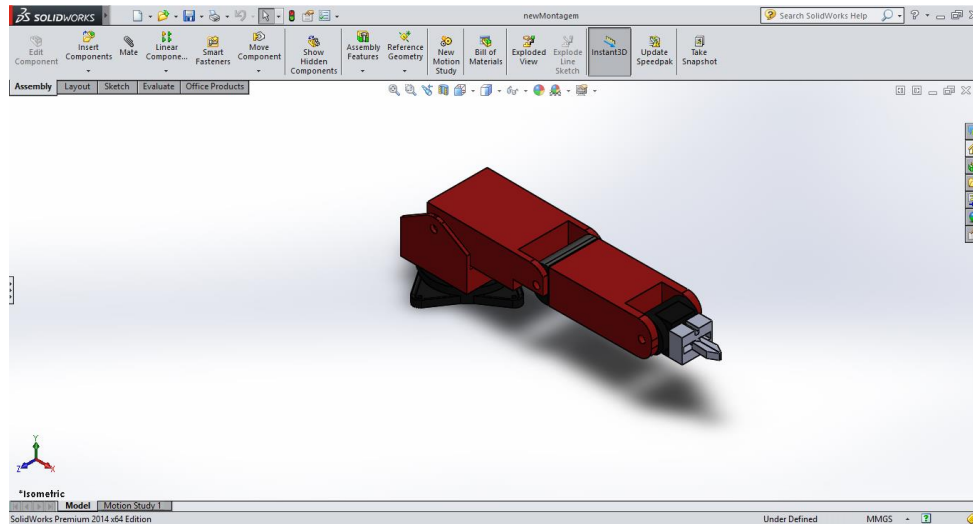


Figura 14 - Interface Software SolidWorks®
Fonte: Autoria Própria

Após concluído o desenho utilizou-se a o toolbox SimMechanics™, que quando instalado, já disponibiliza um menu na interface do SolidWorks®.

Utilizou-se a primeira geração do toolbox como mostra a Figura 15.

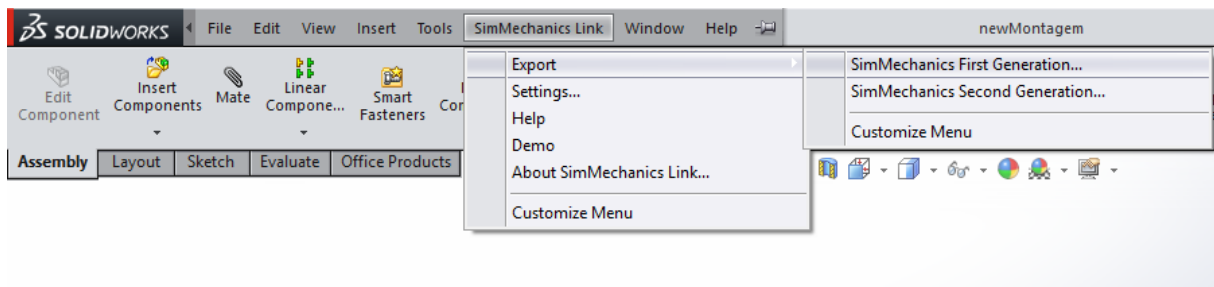


Figura 15 - Utilização toolbox SimMechanics™
Fonte: Autoria Própria

Tal operação gerou um arquivo com extensão xml, o qual foi processado pelo Matlab®, contudo é necessário se atentar a qual diretório salvar o arquivo, este deve ser o mesmo onde estão salvos a montagem do SolidWorks®.

3.8 MATLAB

3.8.1 Simulink

Com a etapa do SolidWorks® concluída, deve-se inserir o comando “mech_import ('nome_do_arquivo.xml')” no Matlab®, o qual criou o mecanismo no ambiente de simulação como mostrado na Figura 16.

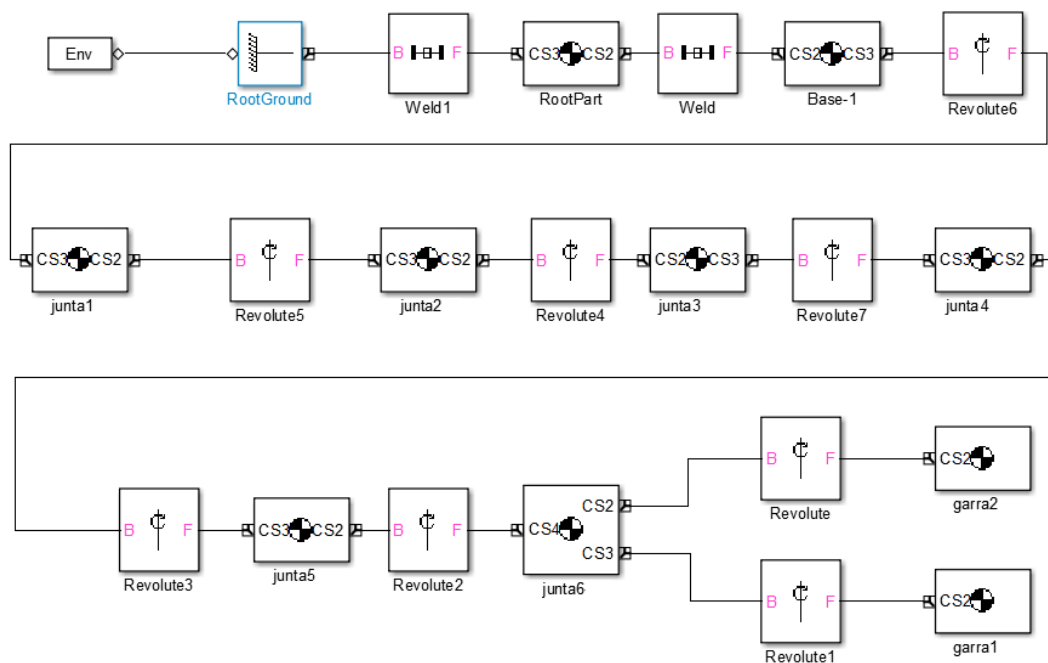


Figura 16 - Diagrama de simulação
 Fonte: Software Simulink®

No esquema acima existem dois blocos que exigem uma atenção maior.

O bloco *Bode*, mostrado na Figura 17, é referente ao corpo da peça.

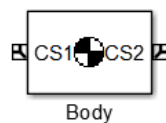


Figura 17 - Bloco Bode
 Fonte: Software Simulink®

Suas configurações, podem ser alterados a massa (*mass*) do corpo, sua inercia (*inertia*) entre outros parâmetros, como a posição e orientação do corpo. A janela de configurações pode ser observada na Figura 18.

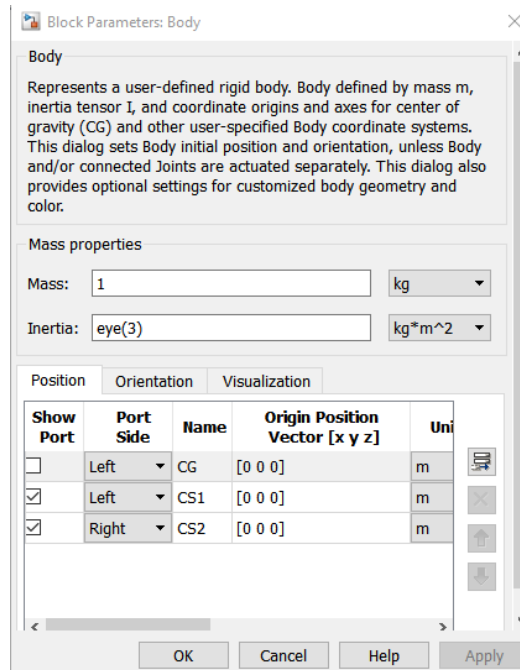


Figura 18 - Parâmetros do bloco *Body*
Fonte: *Software Simulink*®

O segundo bloco a ser comentado, é o *Revolute*, ilustrado na Figura 19.

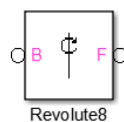


Figura 19 - Bloco *Revolute*
Fonte: *software Simulink*®

Nos parâmetros deste bloco, pode-se selecionar em qual eixo o corpo será rotacionado, e quantos sensores e atuadores existiram, conforme a Figura 20, onde o parâmetro 2 implica que haverá um atuador e um sensor.

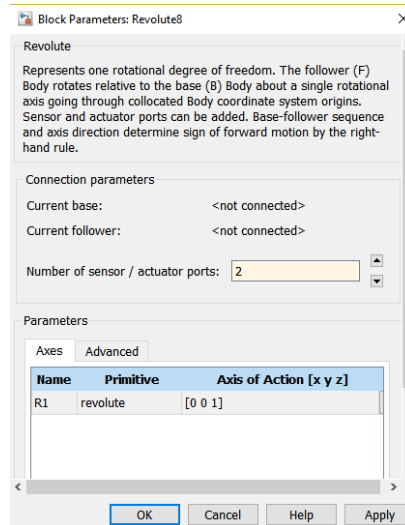


Figura 20 - Parâmetros do bloco *Revolute*
 Fonte: *Software Simulink®*

Após isso, construiu-se os blocos paralelos ao do mecanismo do robô, que de forma sintetizada estão demonstrados na Figura 21.

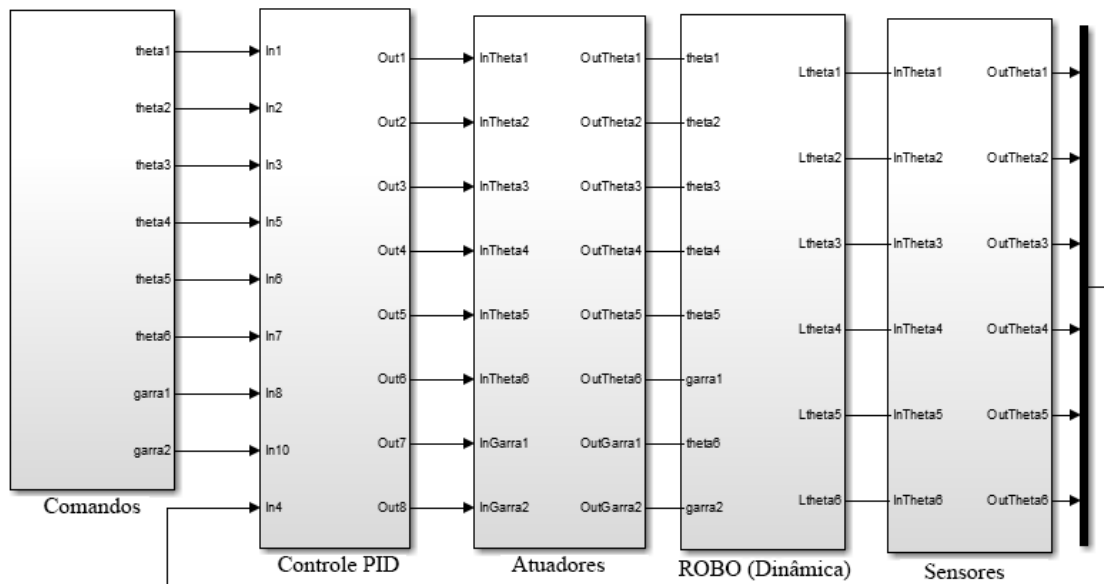


Figura 21 - Diagrama do sistema simulado
 Fonte: *Software Simulink®*

A partir da Figura 21, fragmentou-se o sistema com a finalidade de aprofundamento de cada um dos cinco blocos. Abriu-se o bloco Comandos para implementação das equações (3.71) à (3.76) juntamente com a (3.58a), o que pode ser acompanhado na Figura 22.

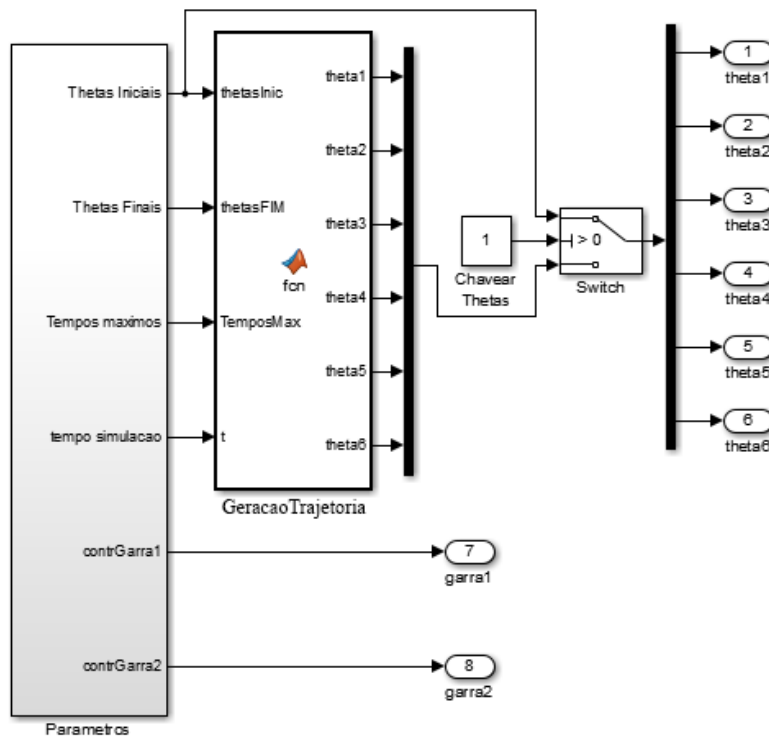


Figura 22 - Subsistema Comandos
Fonte: Software Simulink®

Os parâmetros são constantes, os quais podem ser alterados posteriormente, através de funções do *workspace*, durante a simulação. Enquanto o bloco “GeracaoTrajetoria”, utiliza uma função, esta que está no Apêndice 4. O bloco *switch*, chaveará se for mandado o sistema de ângulos da geração de trajetória ou as condições iniciais. Isso dependerá do que se deseja realizar na simulação, e é alterado a partir da interface gráfica. Contatou-se uma certa dificuldade, para o manipulador realizar vários trajetos em uma única simulação. Dado o fato que, o algoritmo de geração de trajetórias, utiliza o tempo de simulação, e sempre inicia a trajetória com o tempo zero, mas para um segundo movimento será diferente de zero. Para resolver esse problema utilizou-se um bloco de integração para gerar o tempo da trajetória, conforme a Figura 23.

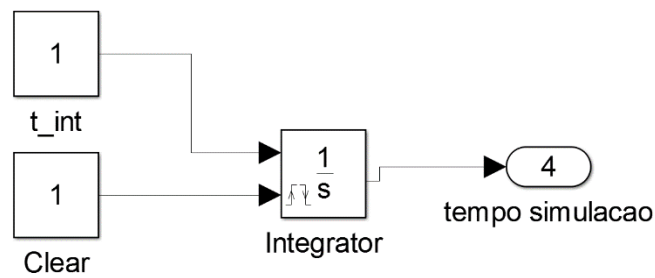


Figura 23 - Gerador de tempo de trajetória
 Fonte: *Software Simulink®*

Dessa maneira, gerou-se uma rampa com coeficiente de inclinação 1, assim quando a constante *Clear* é alterada o integrador é reiniciado, o que traz o valor do tempo novamente a zero, a configuração do bloco é mostrada na Figura 24.

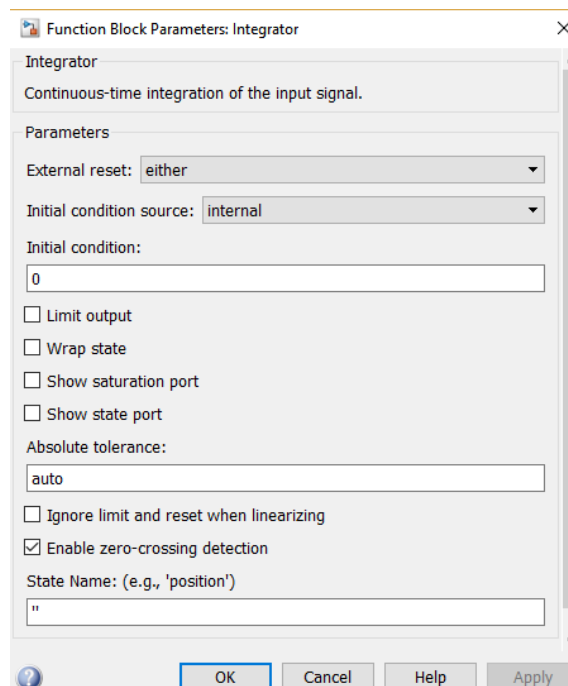


Figura 24 - Parâmetros do bloco Integrator
 Fonte: *Software Simulink®*

Deve-se configurar o campo *External reset* para *either*, assim serão detectadas todas as variações.

Para o controle, utilizou-se um arranjo de blocos, que é repetido para cada junta, como é mostrado na Figura 25, de acordo com a equação (3.95).

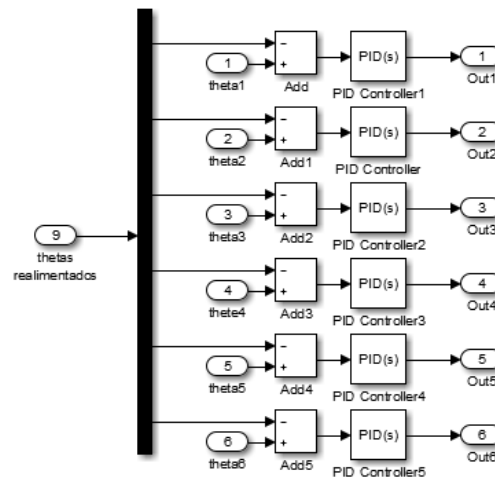


Figura 25 - Diagrama de Controle
 Fonte: *Software Simulink®*

Os atuadores, assim como os controladores, foram replicados para cada junta a partir de um arranjo inicial, este pode ser visto na Figura 26.

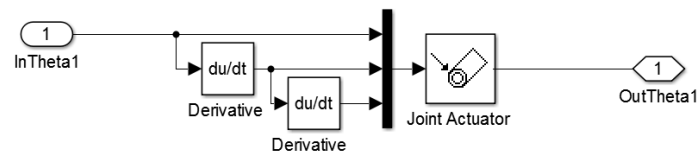


Figura 26 - Diagrama do atuador
 Fonte: *Software Simulink®*

Os atuadores representam as equações (3.58a), (3.58b) e (3.58c).

O bloco do atuador (*Joint Actuator*), foi configurado para atuar com movimento, conforme a Figura 27.

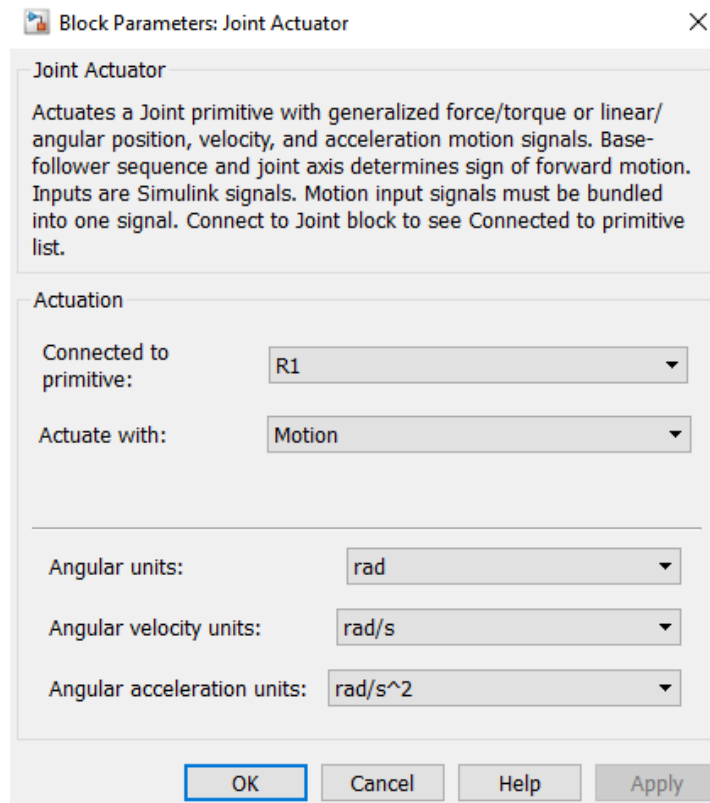


Figura 27 - Parâmetros do bloco Joint Actuator
 Fonte: *Software Simulink®*

Para o presente trabalho, utilizou-se a medida de ângulos em radiano, como ilustrou-se na Figura 27.

A dinâmica do referido robô foi obtida através do código implementado no Matlab® disponível no Apêndice 6. Por apresentar equações demasiadamente longas, não foram explicitadas neste trabalho.

Quanto aos sinais dos sensores de posição angular das juntas, o mesmo mensura e disponibiliza tais informações em caráter ideal, ou seja, não há atraso de transporte, possui ganho unitário e não há comportamento transiente.

3.8.2 Guide

O Matlab® possui uma ferramenta, para construção de interface gráfica, a qual utilizou-se para permitir o controle, a qual facilita a simulação realizada pelo usuário.

A Figura 28 mostra a interface criada.

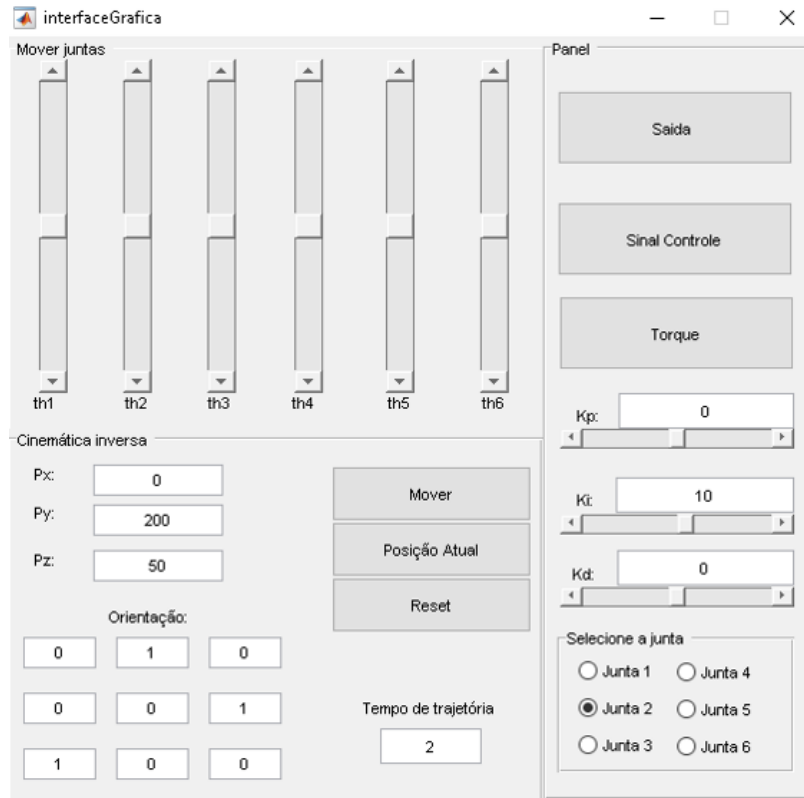


Figura 28 - Interface criada
Fonte: *Software Matlab®*

Esta possui dois módulos de controle, o primeiro por meio dos *Sliders*, que permite uma variação manual dos ângulos das juntas do manipulador, os quais podem ser alterados através dos *slides* th1, th2, th3, th4, th5 e th6 que movimentam respectivamente as juntas de 1 a 6, nos seus respectivos intervalos de movimentos. O segundo módulo faz uso da cinemática inversa e a geração de trajetórias, de acordo com as equações de (3.22) a (3.48). Nesta interface o usuário pode inserir a posição e orientação desejadas, nos campos indicados, e realizar o movimento através do botão Mover. Após um movimento pode-se obter a posição e orientação do manipulador utilizando o botão Posição Atual.

É possível observar os sinais de referência e saída, assim como o sinal de controle e o torque nas juntas através dos seus respectivos botões, para isso, no entanto é necessário selecionar a junta de interesse no campo “Selecione a junta”. Pode-se também alterar os parâmetros do controlador, para um ajuste fino, através dos campos Kp, Ki e Kd.

3.9 CONFIGURAÇÃO DA SIMULAÇÃO PARA ANÁLISE DINÂMICA

A partir da seção 2.5.3 optou-se por utilizar o algoritmo iterativo de Newton-Euler, dado o fato de ser usual um *software* matemático para realizar os cálculos, uma vez que fazer tais cálculos manualmente seria inviável.

Contudo as equações obtidas, foram exageradamente grandes, e não seria viável inseri-las no trabalho. Mas os gráficos da análise dinâmica são mostrados no capítulo de simulação.

Para construção da simulação completa, utilizou-se o toolbox SimMechanics™, do *software* Simulink®, o qual já faz a análise da dinâmica.

A construção desta simulação foi tratada posteriormente, mas para garantir que seja feita uma análise dinâmica são necessárias configurações corretas, conforme relatado a seguir.

O bloco *Machine Environment*, conforme mostrado na Figura 29, é fundamental para a simulação, pois contém as configurações base.

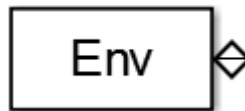


Figura 29 - Bloco *Machine Environment*
 Fonte: *Software Simulink®*

Neste bloco é possível realizar diversas configurações como magnitude e direção do vetor de gravidade, ou também usá-lo como parâmetro externo. Também pode-se selecionar as dimensões da análise, assim como as tolerâncias (passos) da simulação.

É neste bloco que se realiza as configurações referentes a linearização e modelagem do mecanismo. Para este trabalho como já se explicitou a modelagem cinemática optou-se em realizar uma simulação dinâmica, para isso, bastou acessar as propriedades deste bloco, e verificar que o campo *Analysis mode* está selecionado em *Forward dynamics*, conforme a Figura 30.

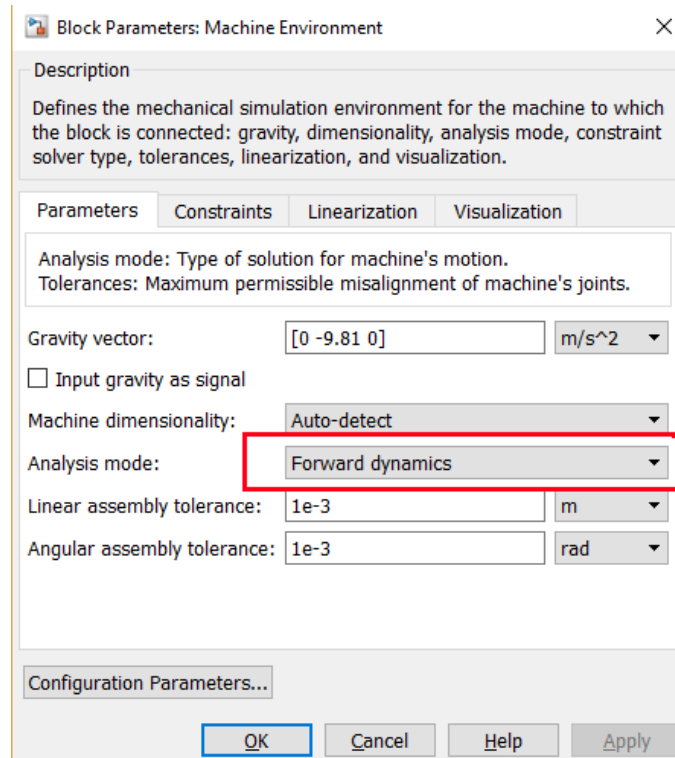


Figura 30 - Parametros do bloco *Machine Environment*
Fonte: software Simulink®

3.10 CONTROLE

Para realização do controle, é necessário um controlador para cada junta do robô. Para este trabalho escolheu-se um controlador PID, pela simplicidade de implementação e pela ampla aplicabilidade a sistemas físicos. A estrutura básica de um sistema com um controlador PID é mostrado na Figura 31.

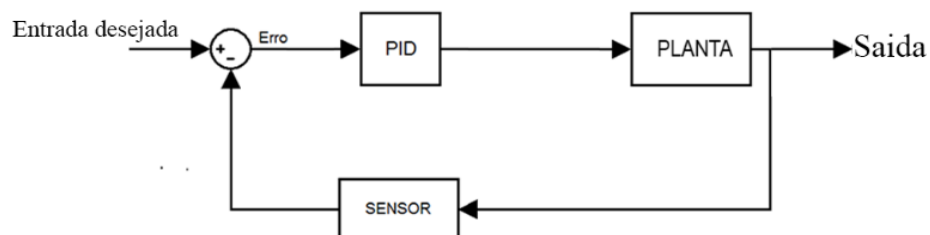


Figura 31 - Diagrama de controle genérico
Fonte: Autoria própria

Este tipo de controlador, apresenta uma ação proporcional + integral + diferencial e possui uma função de transferência genérica, mostrada na Equação (3.95).

$$u(t) = k_p e(t) + k_i \int e(t) + k_d \frac{d e(t)}{dt} \quad (3.95)$$

Onde:

- $u(t)$ é a ação de controle
- $e(t)$ é o erro (diferença entre entrada e saída)
- k_p constante de proporcionalidade
- k_i constante de integração
- k_d constante de diferenciação

Um estudo clássico de controle, não foi possível aplicação do PID a este sistema, pois um manipulador é um sistema não linear e as técnicas clássicas para o projeto, não se aplicam.

Entretanto utilizou-se o bloco do controlador no Simulink®, mostrado na Figura 32, assim tal problema pode ser contornado.



Figura 32 - Bloco PID Controller
Fonte: Software Simulink®

Na Figura 33, ilustra-se a tela de configuração de tal controlador.

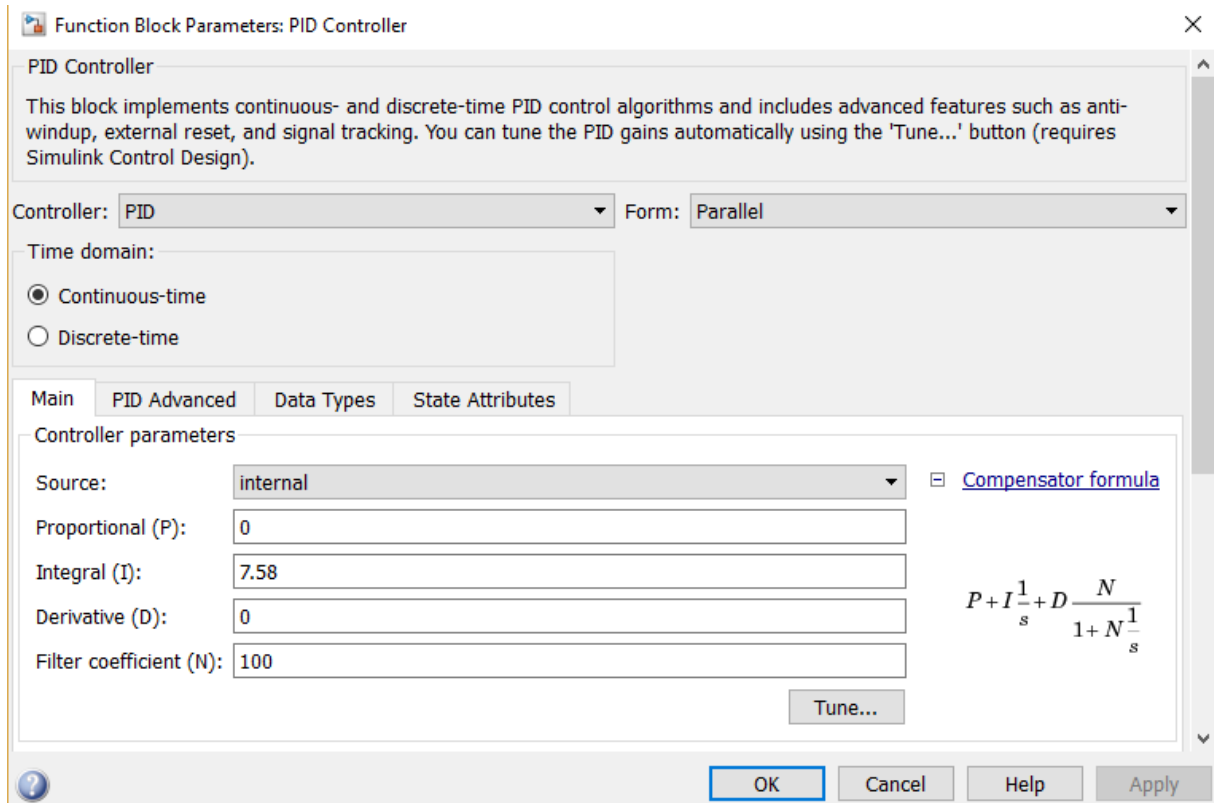


Figura 33 - Parâmetros do bloco *PID Controller*
Fonte: *Software Simulink®*

Sendo N o coeficiente do filtro, P o ganho proporcional, I o ganho integral, e D o derivativo. Realizou-se a sintonia dos controladores, inicialmente, fundamentado na técnica de linearização do sistema através da ferramenta *Looptune*, a qual encontra-se disponível no próprio Simulink®. Tal ferramenta permite o ajuste automático dos ganhos dos controladores PID do sistema completo do robô.

4 SIMULAÇÃO

4.1 PARÂMETROS DE SIMULAÇÃO

Para simulação no Simulink®, os blocos referentes a construção do robô, presentes na Figura 16, foram configurados de acordo com a Tabela 1.

Tabela 1 – Parâmetros de configuração do robô

BLOCO	PARÂMETRO	VALOR
junta1	<i>Mass</i>	0.949652
junta1	<i>Inertia</i>	[0.004965,0,0;0,0.00925,3.886e-05;0,3.886e-05,0.007296]
junta2	<i>Mass</i>	4.7958
junta2	<i>Inertia</i>	[0.0347574,0,0;0,0.017662,-0.0004104;0,-0.0004104,0.0444145]
junta3	<i>Mass</i>	0.599553
junta3	<i>Inertia</i>	[0.000562311,0,0;0,0.00156905,0;0,0,0.00124274]
junta4	<i>Mass</i>	3.50965
junta4	<i>Inertia</i>	[0.0221389,0,0;0,0.0103492,0;0,0,0.0266679]
junta5	<i>Mass</i>	0.757258
junta5	<i>Inertia</i>	[0.000938532,0,0;0,0.00119371,0;0,0,0.00116177]
junta6	<i>Mass</i>	0.220345
junta6	<i>Inertia</i>	[0.000182353,-1.49956e-11,-2.97718e-10;-1.49956e-11,0.000131259,-3.21085e-08;-2.97718e-10,-3.21085e-08,0.000255197]
garra1 e garra2	<i>Mass</i>	0.015923
garra1 e garra2	<i>Inertia</i>	[5.29133e-06,5.39064e-07,0;5.39064e-07,7.69525e-07,0;0,0,5.20135e-06]

4.2 CINEMÁTICA DIRETA

Com as Equações de (3.8) a (3.19), que descrevem a orientação e posição do manipulador, construiu-se uma representação tridimensional, a qual é mostrada na Figura 34:

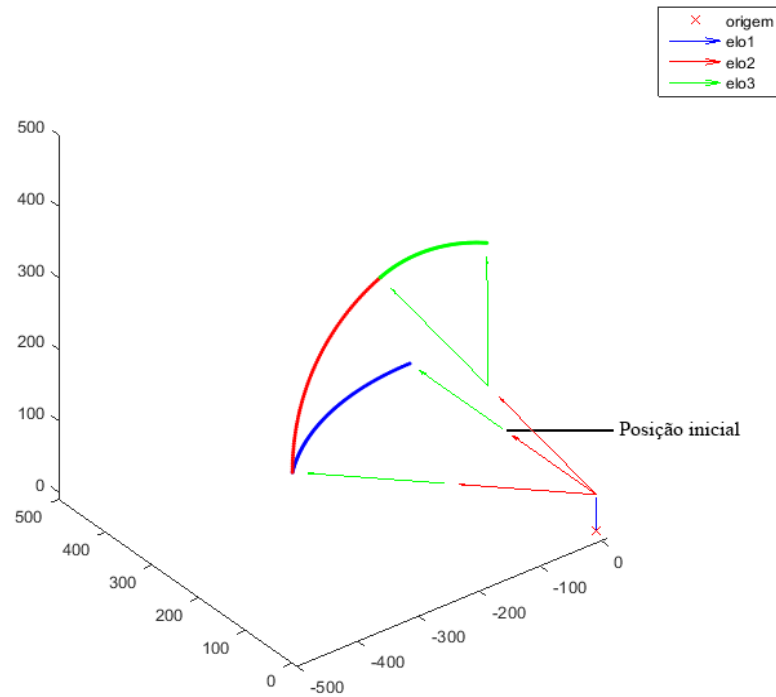


Figura 34 - Representação tridimensional do movimento do robô
Fonte: Autoria própria

Para construção da Figura 34 realizou-se uma simulação no *software* Matlab®, na qual variou-se os ângulos de cada uma das juntas em ordem crescente, começando pela junta 1 e terminando com a junta 6, com uma variação de 45°. No fim da variação de cada junta é plotado a nova posição dos elos do manipulador, realizando desta maneira uma sequência de movimentos. A rotina utilizada para essa simulação encontra-se no Apêndice 2.

Com base nos resultados obtidos na simulação, tornou possível perceber que a cinemática direta está correta, pois a simulação tem uma boa correspondência com o movimento esperado do robô.

As variações dos ângulos θ_4 , θ_5 e θ_6 não são notados nesta representação pois estes influenciam apenas na orientação da ferramenta.

4.3 CINEMÁTICA INVERSA

Devido as altas não linearidades do manipulador existem muitos problemas ao lidar com a cinemática inversa. Determinaram-se as equações de forma fechada, para os ângulos das juntas, contudo houve um grande esforço para isso.

Mesmo com as equações ainda existem considerações a serem feitas como por exemplo o octante onde a ferramenta se localiza, e há também pontos de singularidade, ou seja, pontos em que existe mais de uma combinação de ângulos para atingir aquela posição.

Um exemplo pode ser observado na Figura 35 logo abaixo.

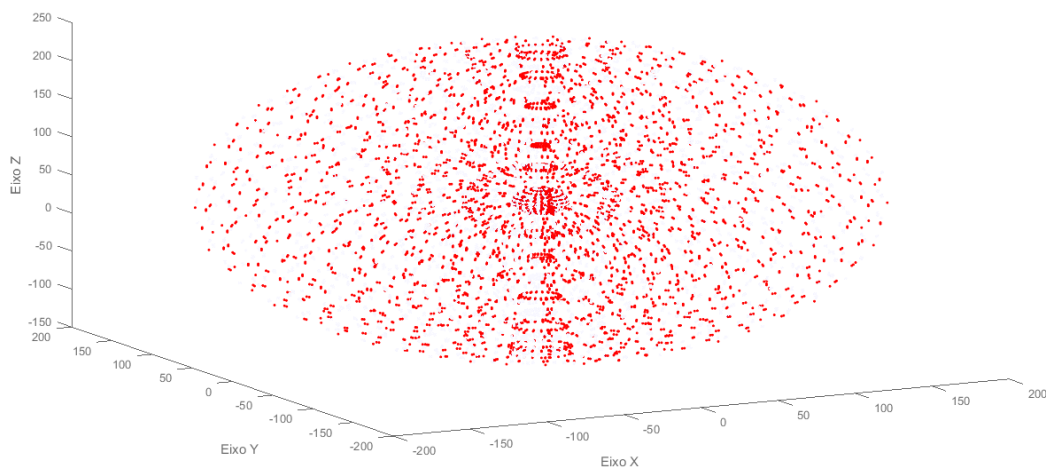


Figura 35 - Nuvem de pontos de singularidade
Fonte: autoria própria

A Figura 35 foi gerada variando os ângulos de cada uma das juntas do manipulador, e mostra o espaço de trabalho do robô, no qual foram marcados os pontos de singularidades. Pontos esses em que as equações obtidas pela cinemática inversa, com análise algébrica, não são válidas.

4.4 GERAÇÃO DE TRAJETÓRIA

Usou-se o algoritmo descrito pelo fluxograma da Figura 6 e o polinômio mostrado na seção 3.6, realizou-se a plotagem dos gráficos os quais descrevem a variação, no tempo, da posição, velocidade e aceleração dos ângulos de cada uma das juntas e logo em sequência ao movimento no espaço cartesiano.

Para isso consideraram-se as condições finais e iniciais descritas na Tabela 2:

Tabela 2 – Parâmetros para simulação da geração de trajetória no espaço de juntas.

JUNTAS	INICIAL	FINAL	TEMPO DE SIMULAÇÃO
θ_1	0	π	10
θ_2	0	$\frac{\pi i}{4}$	12
θ_3	0	$\frac{\pi}{2}$	8
θ_4	0	$\frac{\pi}{8}$	11
θ_5	0	$\frac{\pi}{6}$	20
θ_6	0	$\frac{\pi i}{4}$	15

Com tais condições iniciais e finais e os tempos de simulação gerou-se as Figuras de 36 a 41.

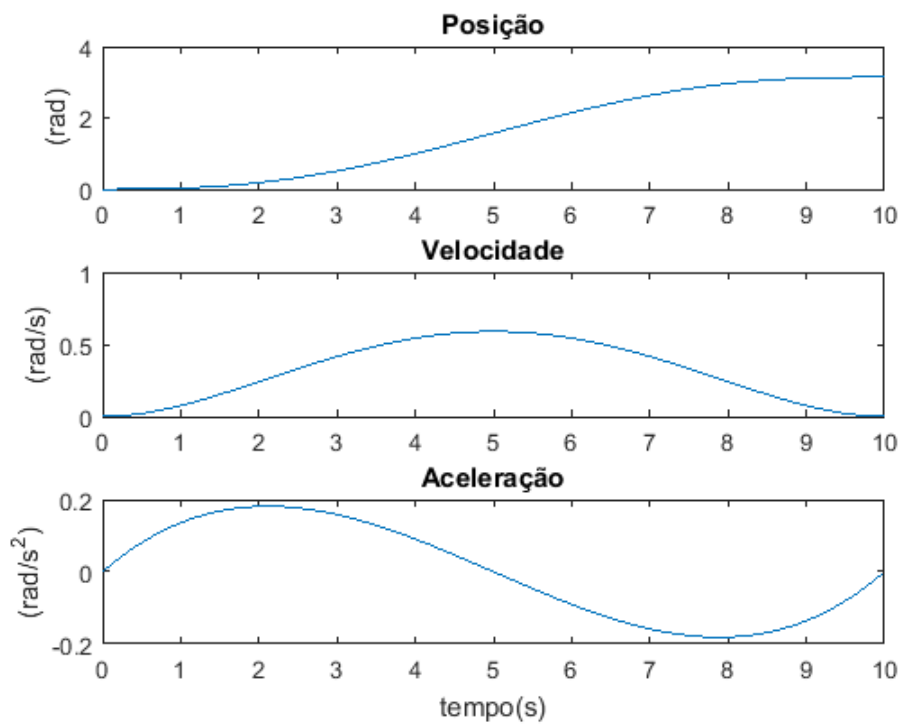


Figura 36 - Geração de trajetórias da primeira junta
Fonte: Autoria própria

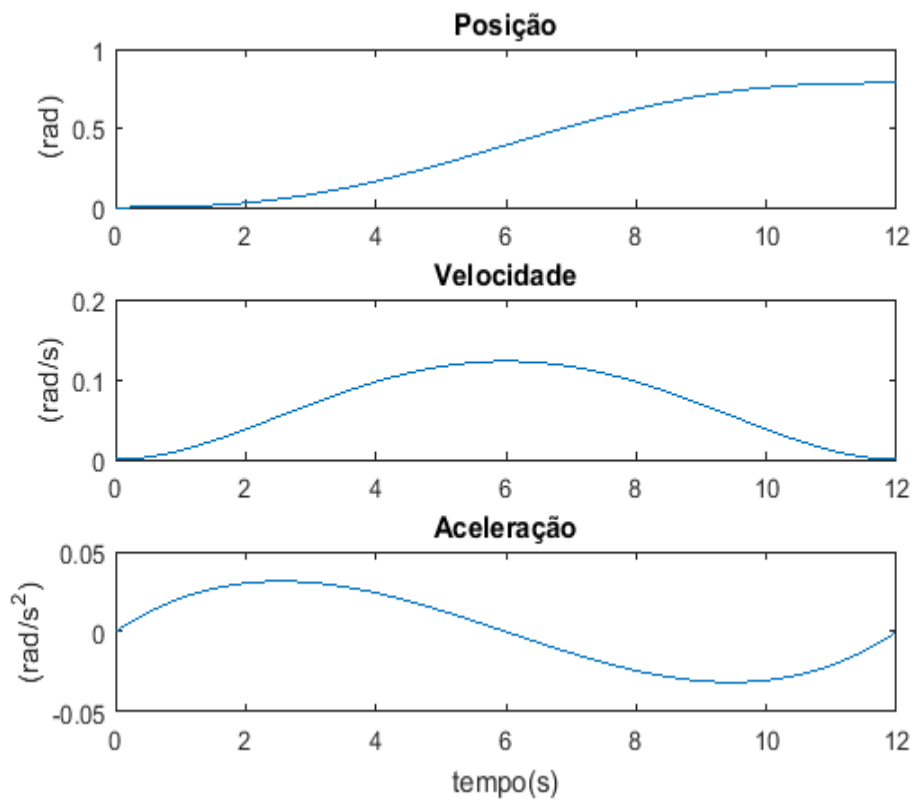


Figura 37 - Geração de trajetórias da segunda junta
Fonte: Autoria própria

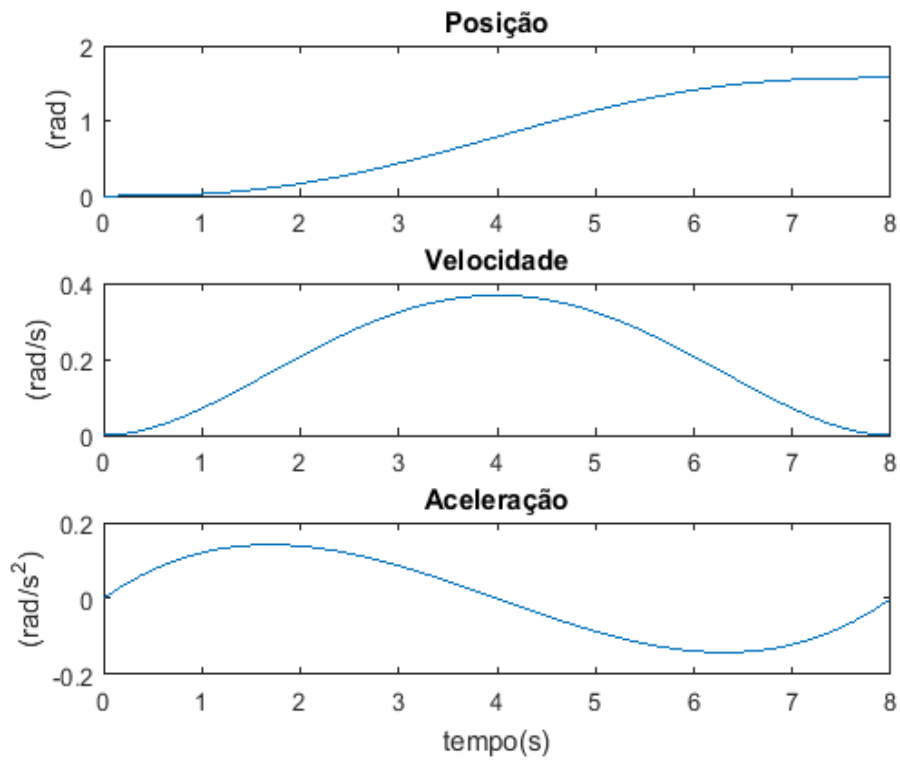


Figura 38 - Geração de trajetórias da terceira junta
Fonte: Autoria própria

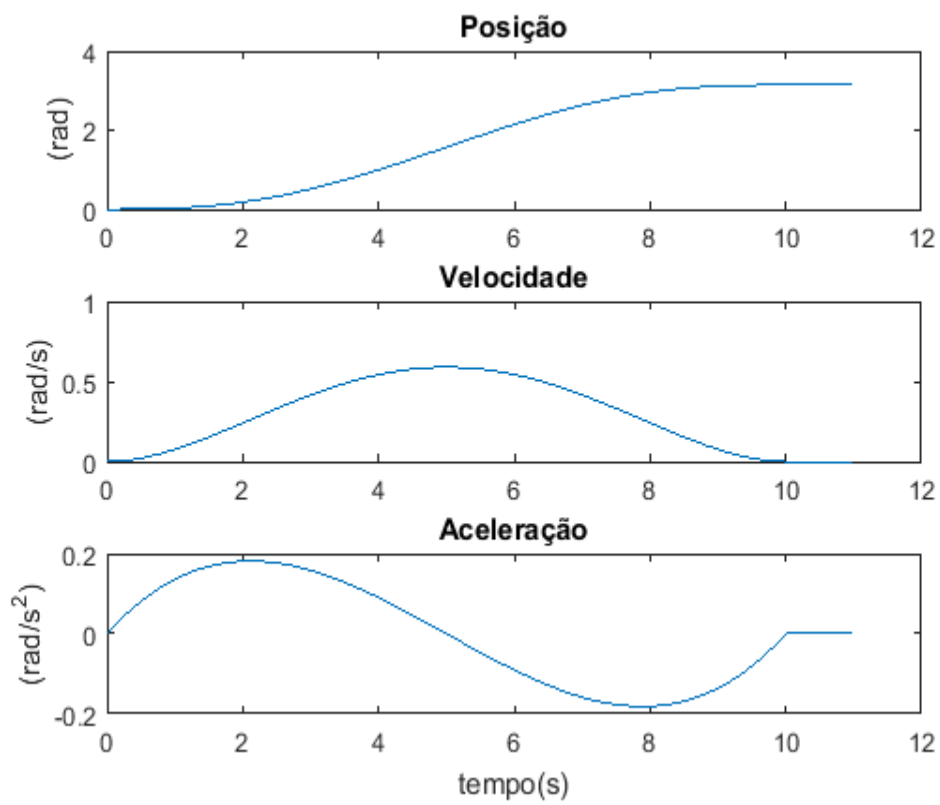


Figura 39 - Geração de trajetórias da quarta junta
Fonte: Autoria própria

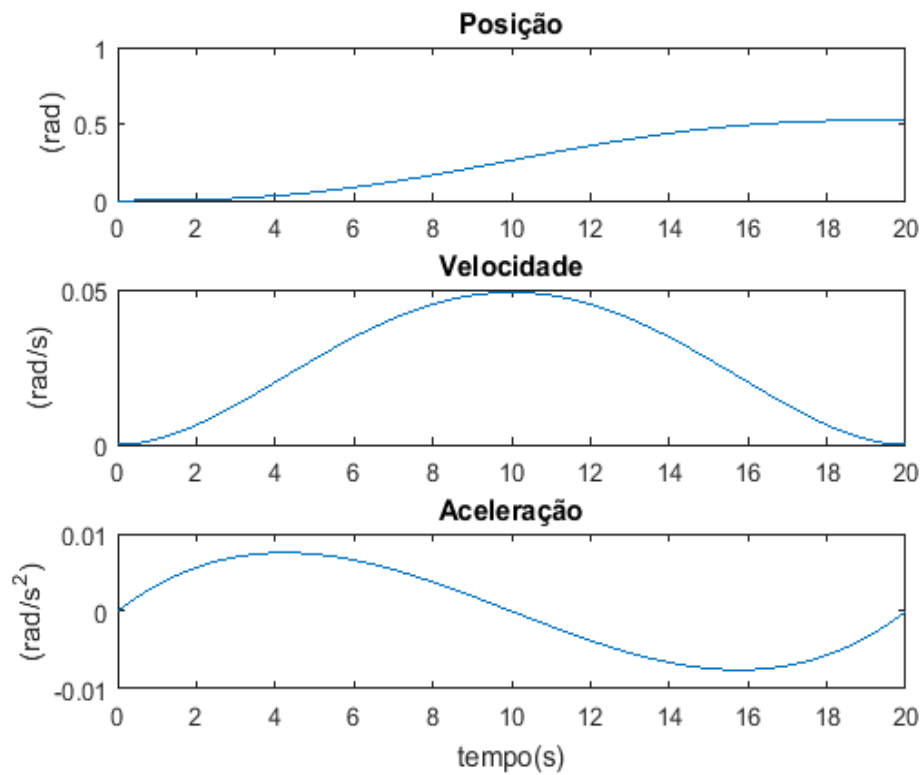


Figura 40 - Geração de trajetórias da quinta junta
Fonte: Autoria própria

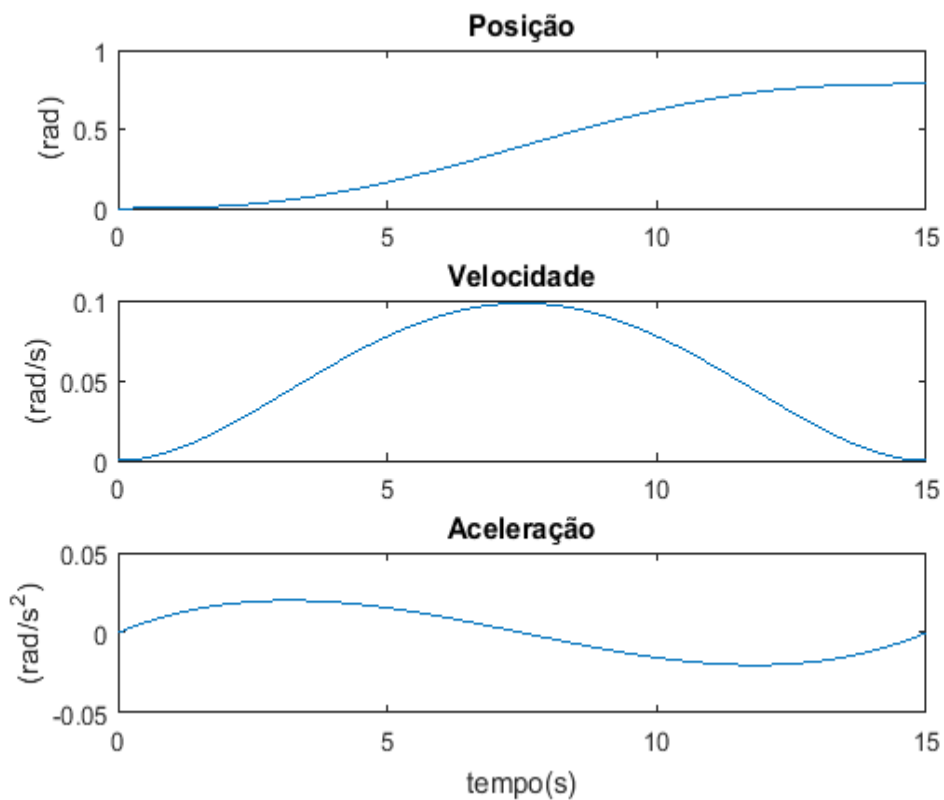


Figura 41 - Geração de trajetórias da sexta junta
Fonte: Autoria própria

Com os gráficos gerados, é possível observar que independente das condições iniciais e finais assim como o tempo de simulação, todas as curvas das trajetórias respeitam os resultados esperado, tendo uma partida suave para o posicionamento assim como iniciando e terminando a velocidade e aceleração em zero.

A partir da variação dos ângulos das juntas também é obtido um movimento no espaço cartesiano o qual é descrito pela Tabela 3 e a Figura 42.

Tabela 3 – Parâmetros para simulação da geração de trajetória no espaço cartesiano.

EIXO	INICIAL	FINAL
X	0	0
Y	400	-141
Z	50	391

Geração de trajetória no espaço cartesiano

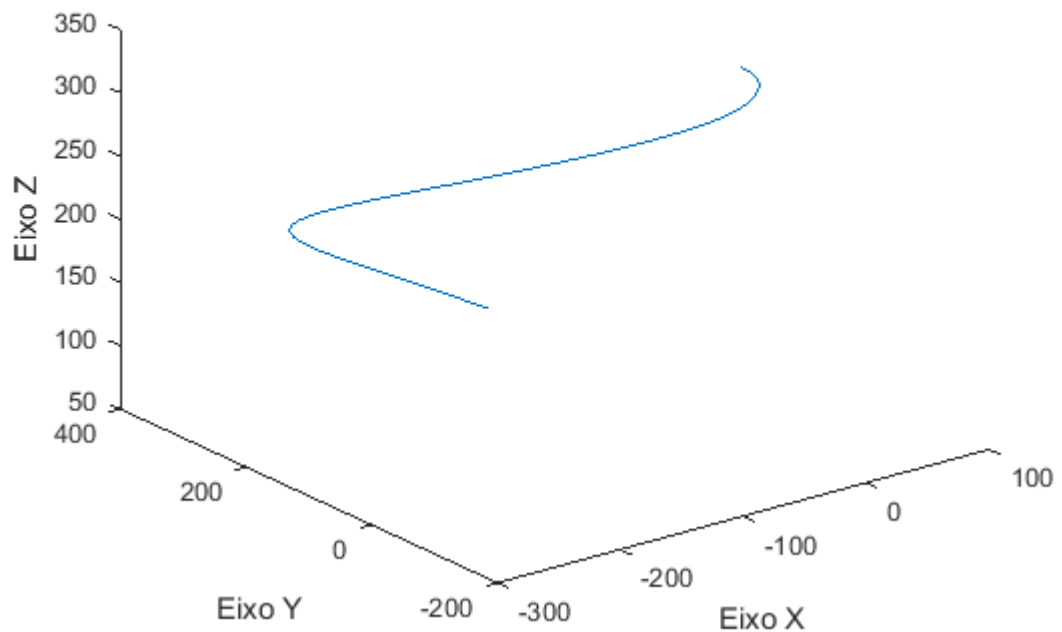


Figura 42 - Geração de trajetória no espaço cartesiano
Fonte: Autoria própria

Ao observar os gráficos da Figura 42, é possível perceber que o movimento ocorreu como projetado. A trajetória descreveu um movimento suave respeitando ao início, fim e tempo de percurso especificados.

4.5 SIMULAÇÃO DA JACOBIANA

Com o intuito de averiguar a matriz jacobiana, realizou-se uma comparação da cinemática direta com as posições cartesianas geradas com a Equação (4.1).

$$\dot{X} = J(\theta) \dot{\Theta} \quad (4.1)$$

Onde \dot{X} é o vetor das velocidades das velocidades lineares e angulares no espaço cartesiano, e $\dot{\Theta}$ o vetor das velocidades das juntas no espaço de juntas do robô. Para realização dessa simulação utilizou-se, no *software* Matlab®, um algoritmo que é descrito pelo fluxograma da Figura 43.

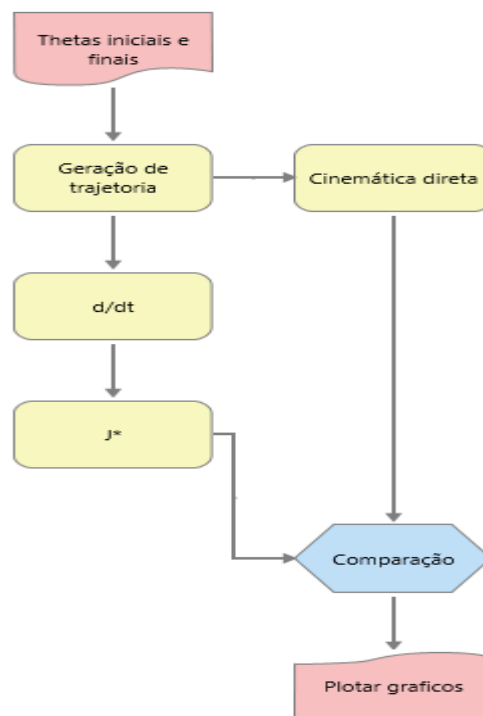


Figura 43 - Fluxograma para simulação da jacobiana
Fonte: Autoria própria

Após a execução dessa rotina, apresentada no Apêndice 5, foi plotado o gráfico da Figura 44.

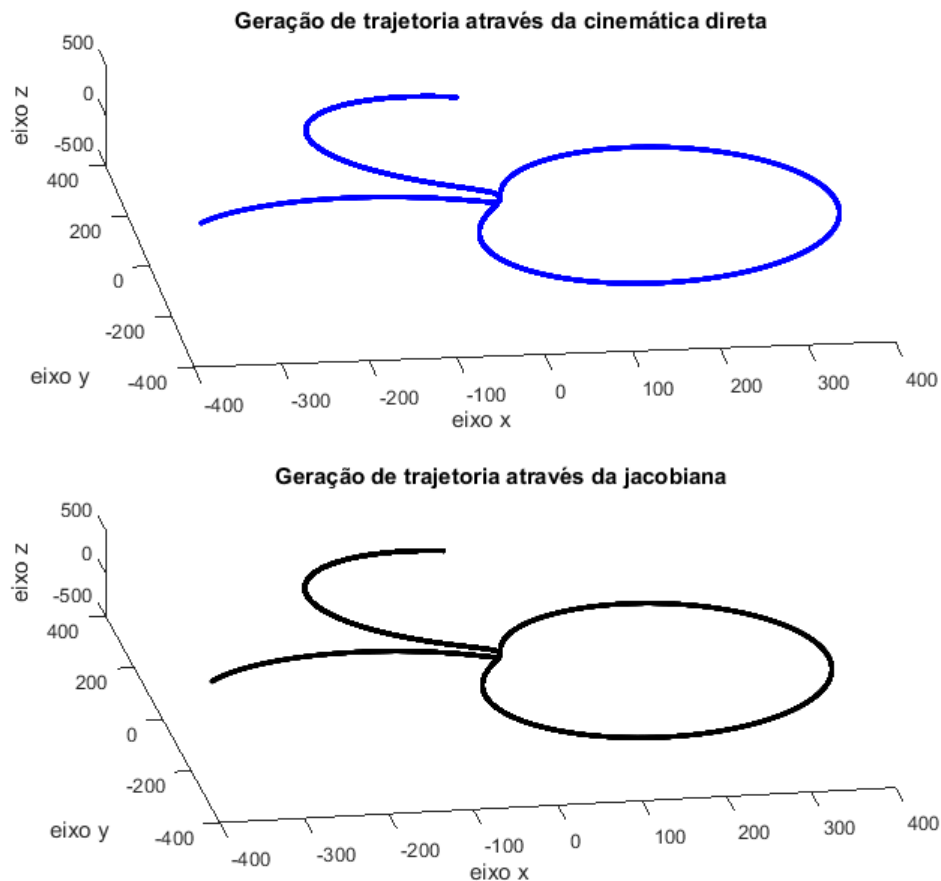


Figura 44 - Geração de trajetória com cinemática direta e jacobiana
Fonte: Autoria própria

A Figura 44, foi construída fazendo uso da cinemática direta e da Equação (4.1), com o intuito de realizar a comparação entre os dois para verificar se a matriz jacobiana foi obtida corretamente.

Como descrito na Equação (4.1), fui multiplicado a matriz jacobiana pelo vetor de velocidades angulares, o que resultou em um vetor de velocidade lineares, mas objetiva-se encontrar a posição linear por esse motivo realizou-se uma integração numérica, mais especificamente o método de integração de Euler, o qual embora não seja o mais sofisticado atende bem as necessidades deste caso.

Por inspeção, nota-se que as duas trajetórias são correspondentes, o que significa que a jacobiana está correta e os resultados seguem conforme o esperado.

4.6 SIMULAÇÃO DINÂMICA

Conforme já relatado na Seção 3.8, para análise dinâmica do sistema utilizou-se o *toolbox* SimMechanics™, o qual auxilia na construção de gráficos de torques e foi utilizado para configuração o sensor de junta, que é mostrado na Figura 45, para torque.

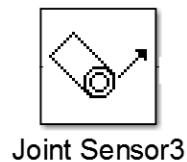


Figura 45 - Bloco *Joint Sensor*
Fonte: *Software Simulink*®

Para isso ajustaram-se os parâmetros do bloco, a partir da opção *computed torque* para que assim a saída do sensor seja o torque aplicado na respectiva junta, a tela de configuração para este bloco é mostrada na Figura 46.

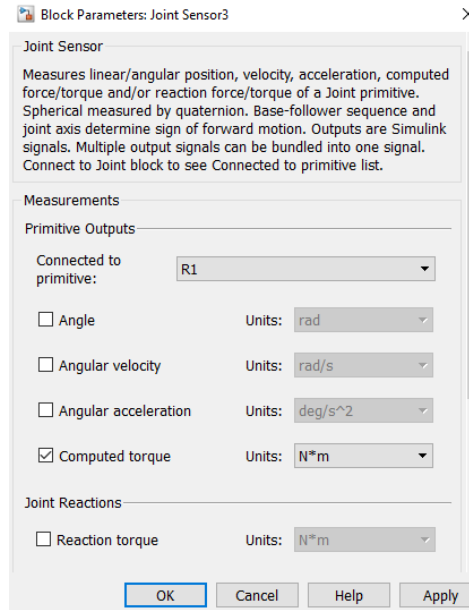


Figura 46 - Parâmetros do bloco *Joint Sensor*
Fonte: *Software Simulink*®

Nesta configuração optou-se em não realizar a leitura angular, mas apenas o torque, e no campo *Connected to primitive* é selecionado R1 pois as juntas são rotacionais. Como mostrou-se na Figura 22, o sistema possui um bloco de geração de trajetórias, por tanto é necessário passar as condições iniciais e finais para simulação, assim como o tempo desta simulação. Com o objetivo de demonstrar que é possível realizar dois movimentos consecutivos foi utilizado um movimento do estado inicial até um estado intermediário e após isso até um estado final.

Para isso foram consideradas as condições mostradas na Tabela 4.

Tabela 4 – Parâmetros para simulação da dinâmica.

JUNTAS	INICIAL	INTERM.	FINAL	TEMPO INTERM.	TEMPO MÁXIMO
θ_1	0	$\frac{\pi}{4}$	π	5	8
θ_2	0	$\frac{\pi}{4}$	$\frac{\pi}{2}$	5	10
θ_3	0	$\frac{\pi}{4}$	$-\frac{\pi}{6}$	5	12
θ_4	0	$\frac{\pi}{4}$	$\frac{\pi}{2}$	5	9
θ_5	0	$\frac{\pi}{4}$	$\frac{\pi}{8}$	5	15
θ_6	0	$\frac{\pi}{4}$	$-\frac{\pi}{2}$	5	14

A partir da parametrização dos blocos, realizou-se a simulação, obtiveram-se os gráficos que descrevem os comportamentos temporais de cada uma das juntas nas Figuras de 47 a 52.

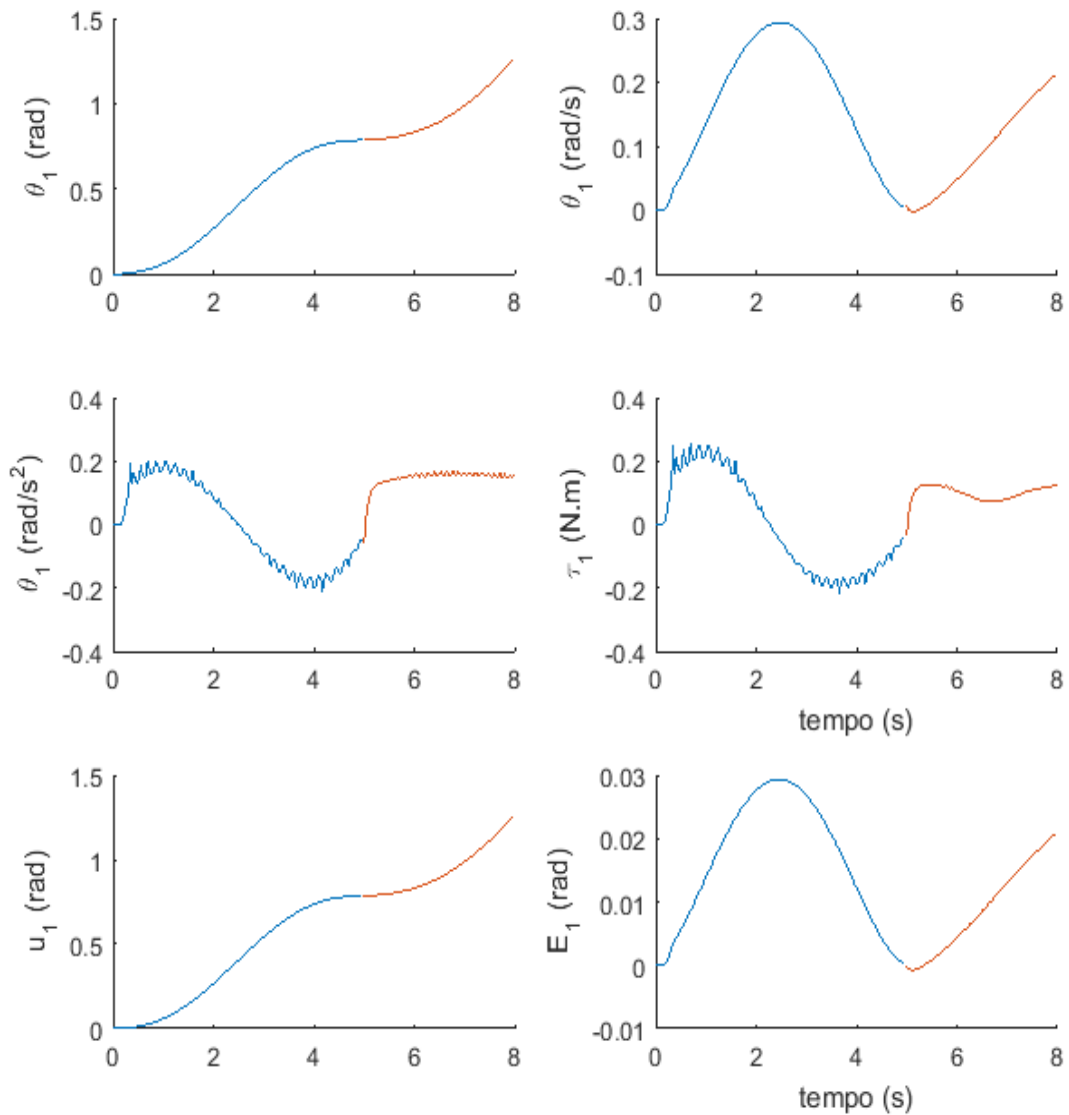


Figura 47 - Comportamento temporal da junta 1
Fonte: Autoria própria

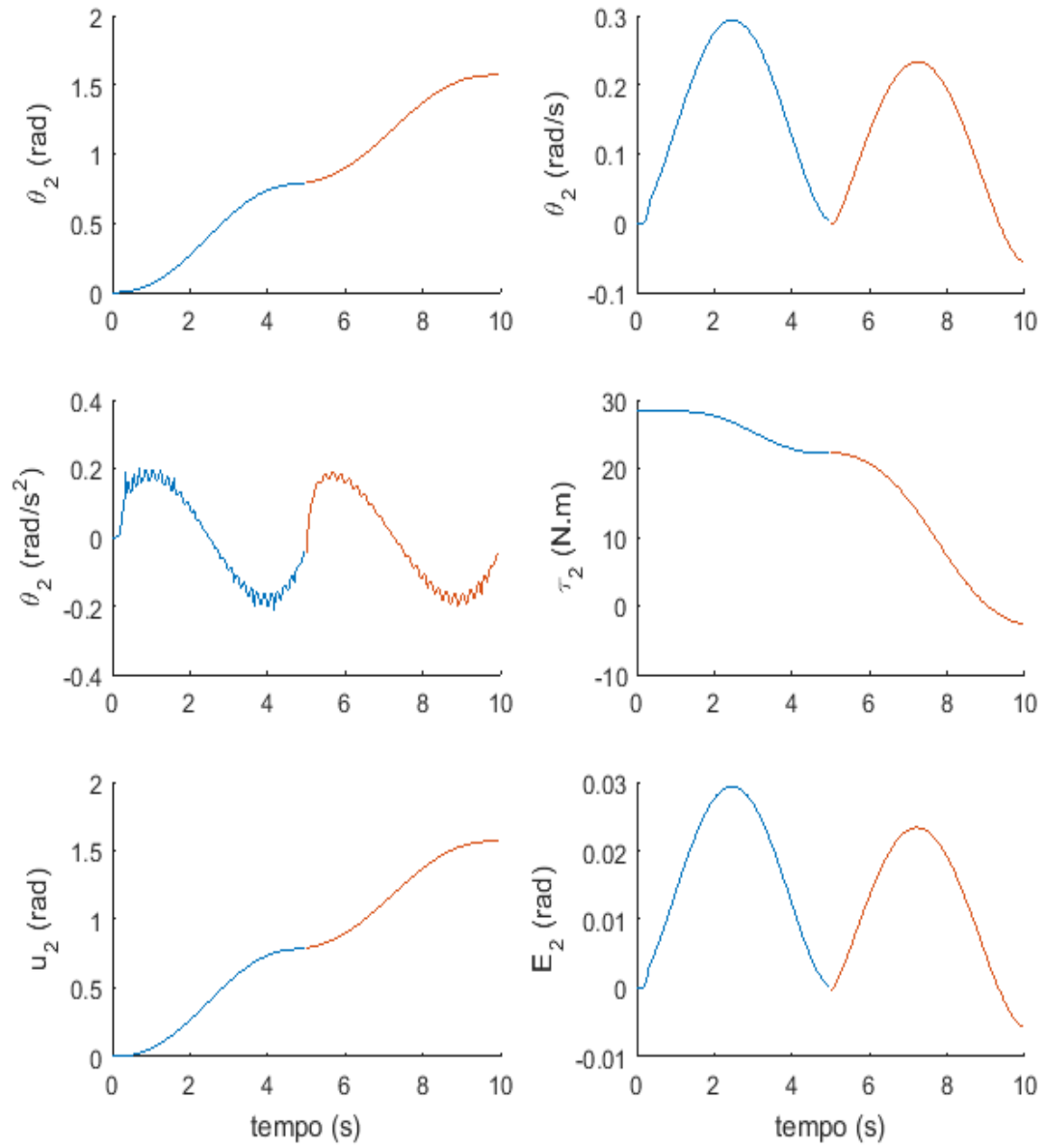


Figura 48 - Comportamento temporal da junta 2
Fonte: Autoria própria

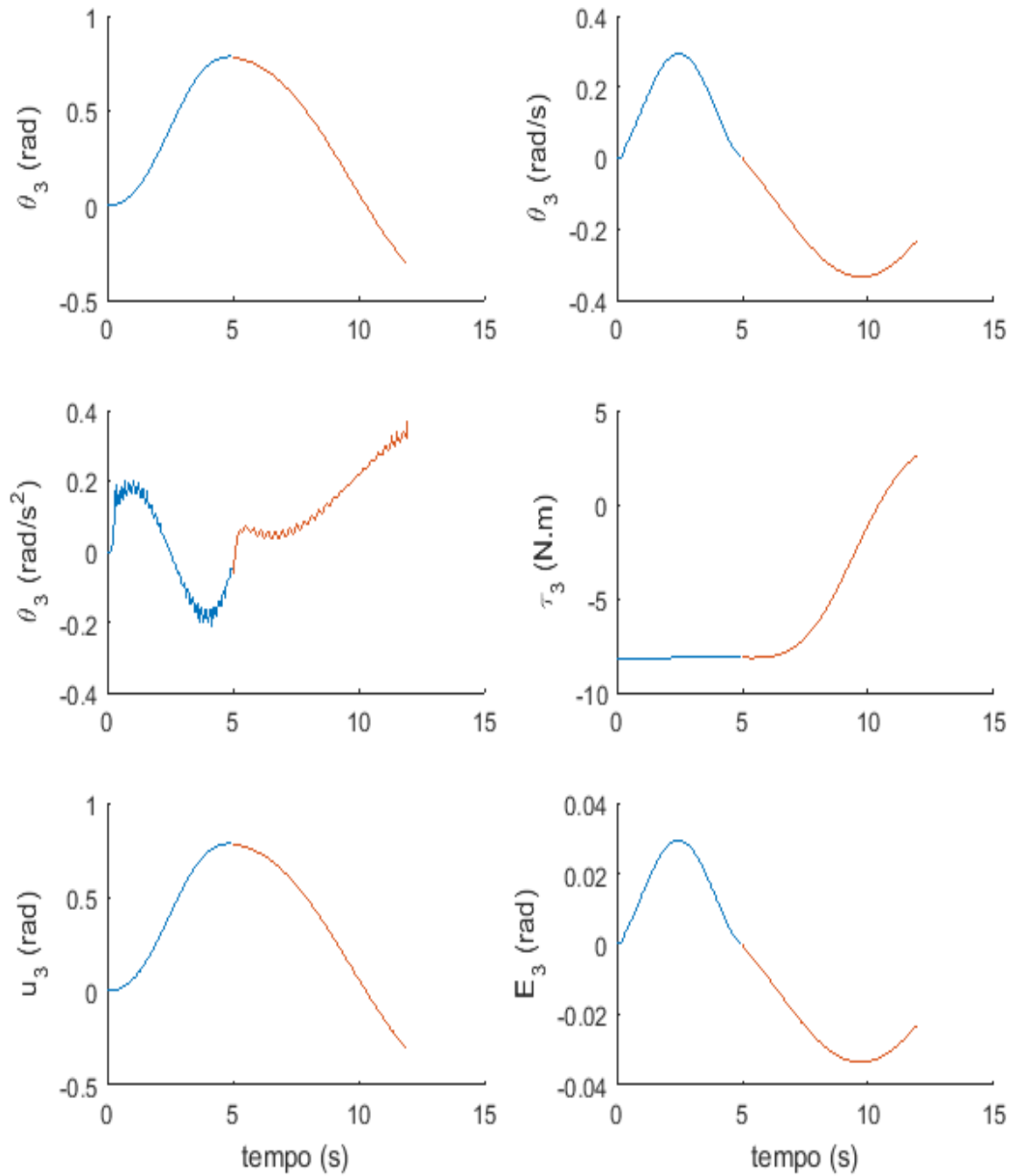


Figura 49 - Comportamento temporal da junta 3
 Fonte: Autoria própria

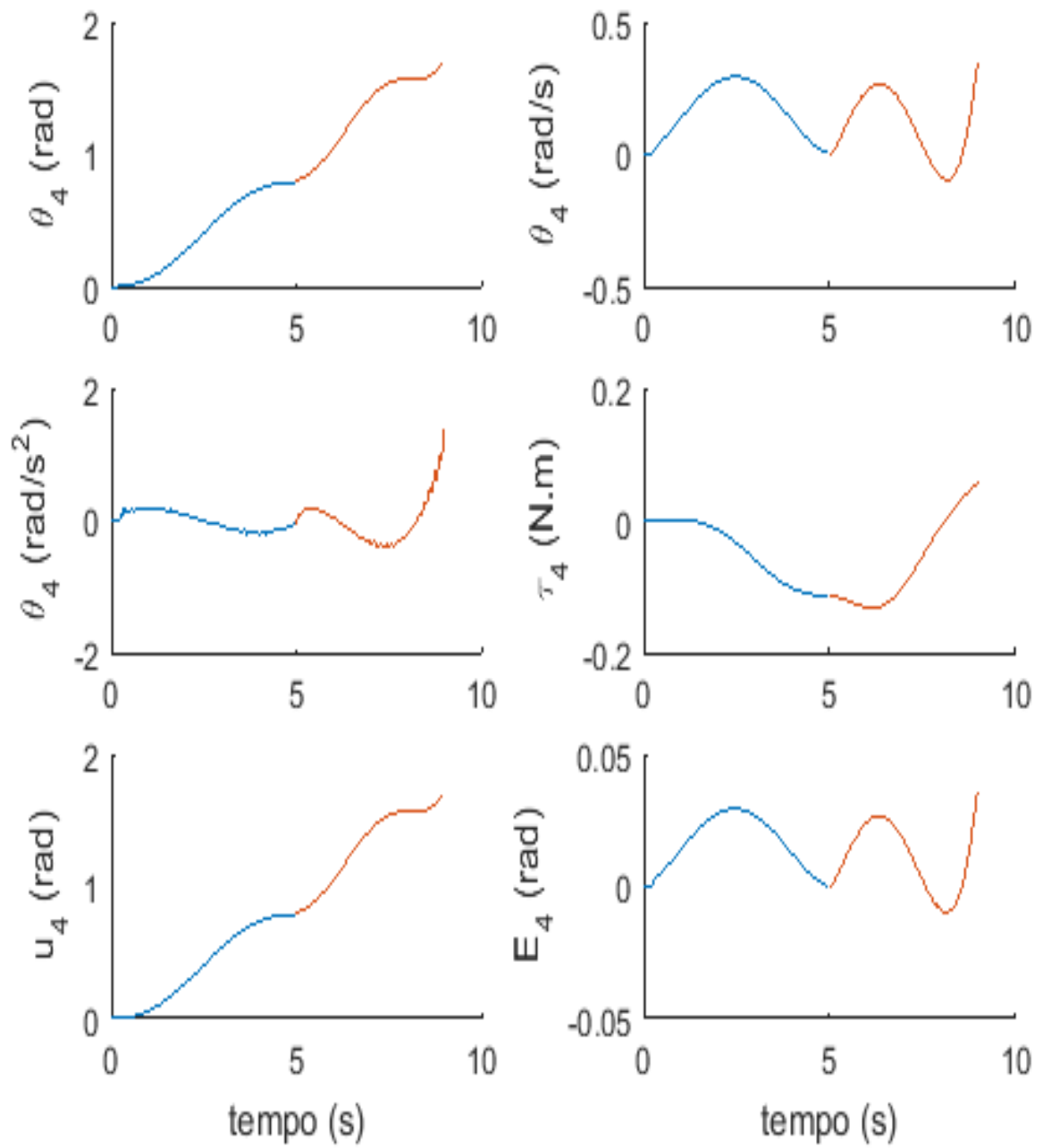


Figura 50 - Comportamento temporal da junta 4
 Fonte: Autoria própria

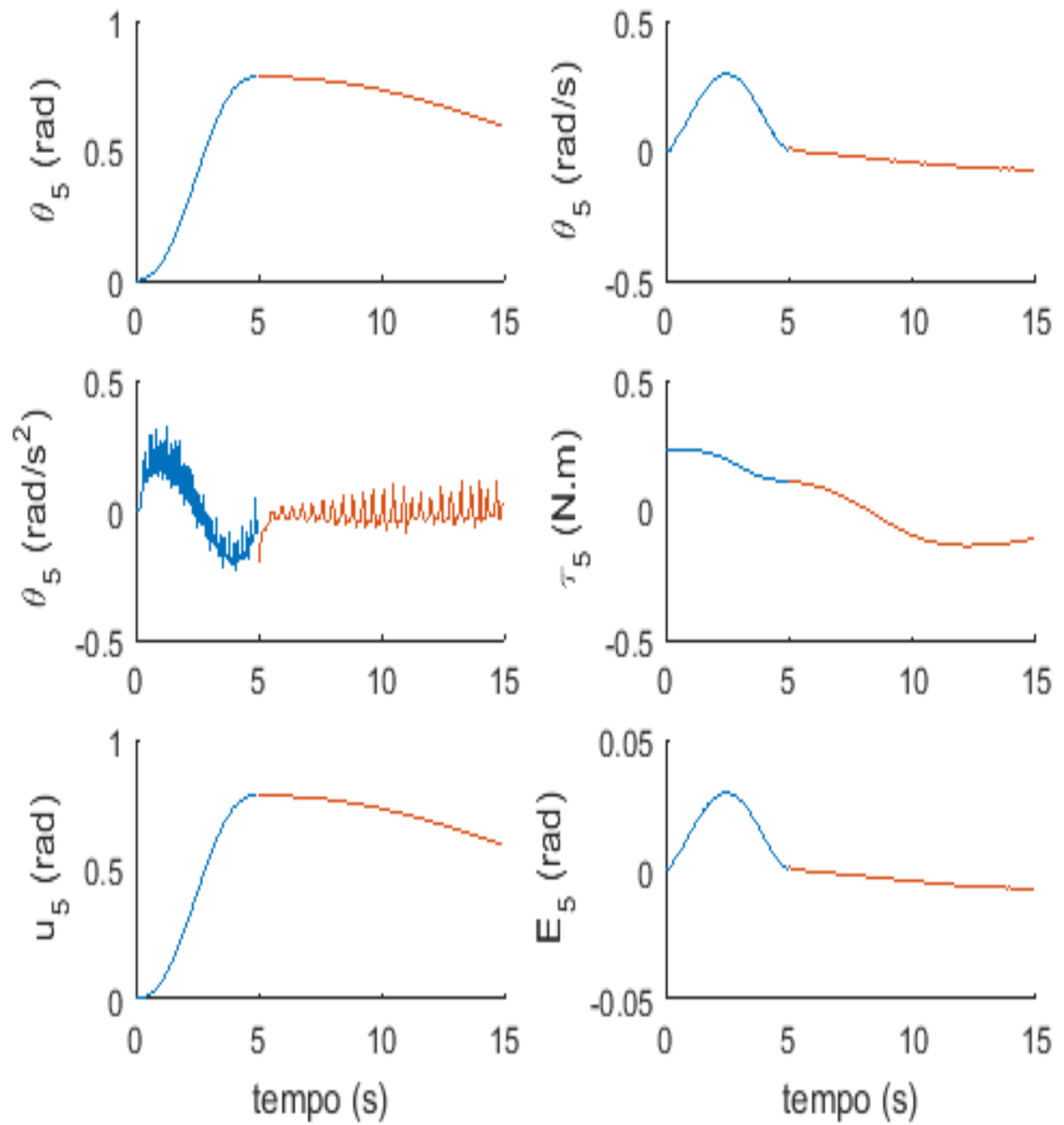


Figura 51 - Comportamento temporal da junta 5
 Fonte: Autoria própria

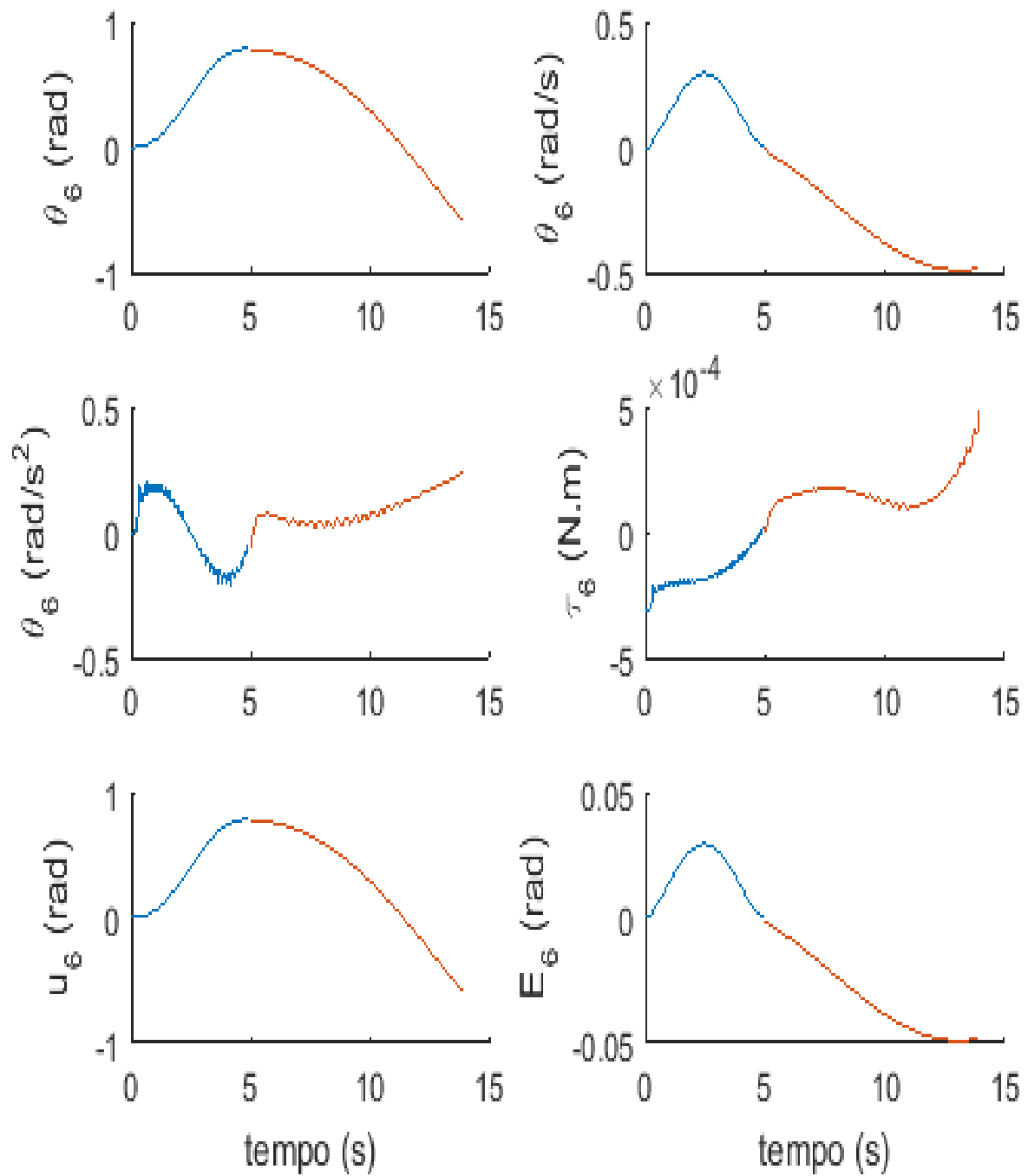


Figura 52 - Comportamento temporal da junta 6

Fonte: Autoria própria

O posicionamento seguiu com precisão a referência, é possível perceber também que o sinal da ação de controle é igual ao sinal de posição, isso se deve ao fato de não haver ruído no sistema, nem interno nem pela ação de cargas externas.

Como esperado a junta 2 apresentou o maior torque, isso ocorre pois tal junta sustenta o manipulador todo, e também entre a junta e a ferramenta existe uma distância maior quando comparado as outras. Enquanto os torques das juntas 4,5 e 6 são os menores, isso acontece porque tais juntas são responsáveis apenas para a orientação do efetuador, o que não demanda de muito torque.

Em algumas juntas foi possível observar uma certa oscilação no gráfico de aceleração, isso ocorre devido aos parâmetros do controlador, a partir de um ajuste fino isso pode ser diminuído, no entanto como é possível observar no gráfico de posição, essas oscilações de aceleração não têm grandes impactos na posição, como a proposta do trabalho é um controle de posicionamento, o controle foi mantido.

Algo que se observou também, foi o gráfico de erro, o qual em todas as juntas possui uma correspondência com o gráfico de velocidade, isso acontece, por que a diferença entre a posição lida pelo sensor e a posição de referência é exatamente a mesma variação das posições entre dois instantes de tempo consecutivos, assim como no caso da ação de controle isso ocorre devido ao fato de não haver ruídos ou distúrbios no sistema.

4.7 CONTROLE

Com base na seção 3.11, implementou-se o controle PID no Simulink®, o qual usou a ferramenta *looptune* e esta gerou os parâmetros do controlador, que estão presentes na Tabela 4.

Tabela 5 – Parâmetros do controlador.

PARÂMETROS	VALOR
k_p	0
k_i	10
k_d	0

A partir desses parâmetros, simulou-se o sistema para duas entradas diferentes, com o intuito de averiguar como o sistema se comporta pra diferentes comandos. A primeira foi um degrau com amplitude π , que tem sua resposta mostrada na Figura 53.

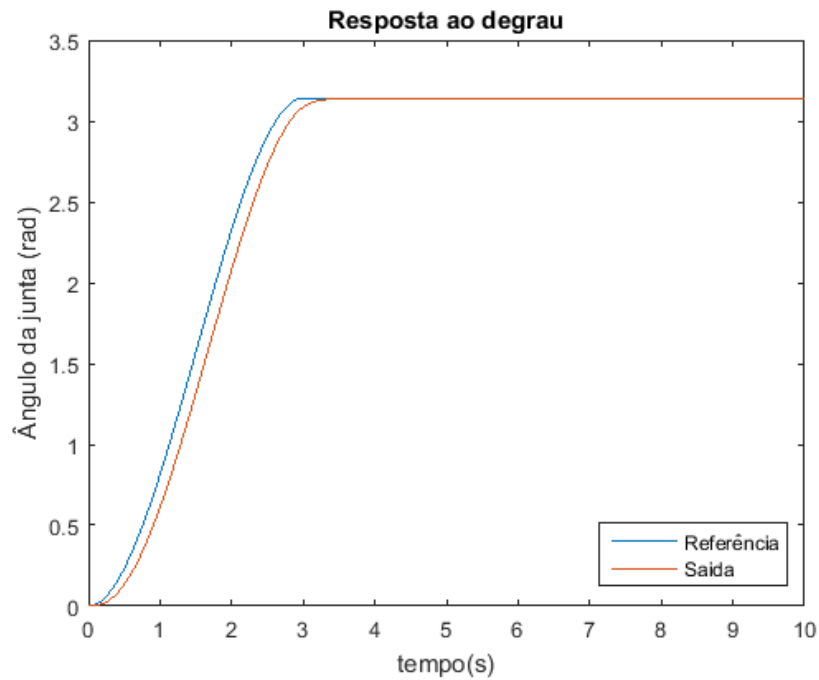


Figura 53 - Resposta do sistema para entrada em rampa
Fonte: Autoria própria

O sistema teve boa resposta para uma entrada em degrau, respeitando os parâmetros utilizados na ferramenta *looptune*, com tempo de aproximadamente 0,5 segundo, não houve ultrapassagem e o erro em regime permanente foi nulo.

Para a segunda uma entrada de teste usou-se uma senoidal, conforme mostrado na Figura 54.

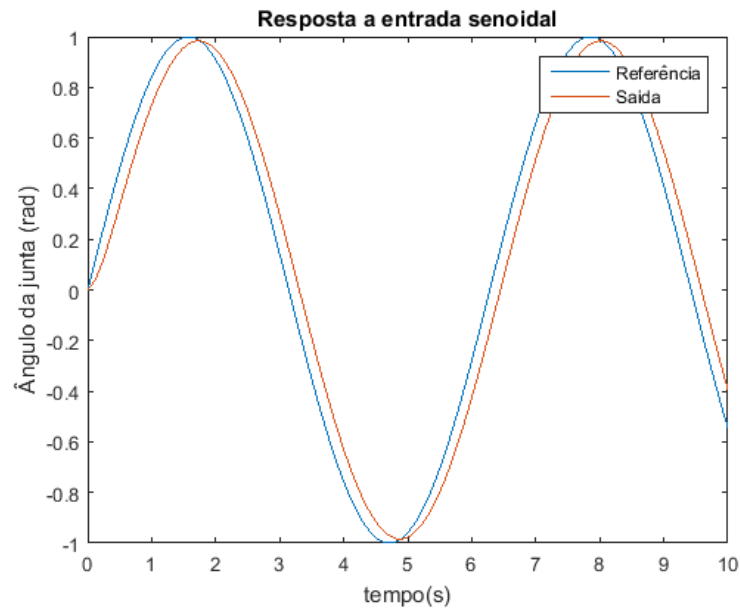


Figura 54 - Resposta do sistema para entrada senoidal
Fonte: Autoria própria

As duas respostas apresentaram tempo de estabelecimento de aproximadamente 0.5 segundos e não tiveram ultrapassagem.

Contudo é possível observar, que para a entrada em degrau o erro em regime permanente é nulo, enquanto a senoidal apresentou um erro de aproximadamente 0.1597 rad.

Isso indica que esse sistema de controle, limita o robô manipulador a trabalhos em linhas de produção estáticas, por exemplo, se o robô precisa desenvolver alguma tarefa em uma esteira, a esteira não pode estar em movimento enquanto o robô estiver desenvolvendo a tarefa, pois o erro nunca será nulo. Pois apenas implementado desta maneira o sistema se torna incapaz de seguir a referência. Contudo existem algumas soluções para este caso, uma delas seria refazer o projeto de controle para tornar o sistema apto a trabalhar com esse tipo de situação, e uma segunda opção, embora não muito didática, mas traz uma facilidade pratica maior, seria no caso de já descontar o atraso do sistema na entrada fazendo assim com que o sistema esteja “adiantado” isso poderia corrigir o problema do atraso.

5 CONCLUSÃO

Ao avaliar um robô manipulador de juntas rotativas e com 6 graus de liberdade, percebeu-se que a programação *off-line* proporciona um grande benefício no momento da aplicação prática. Por meio de uma programação *off-line*, é possível obter informações importantes sobre o manipulador, como por exemplo movimento, forças, posições, entre outros. Uma das maiores importâncias da programação *off-line* encontra-se no fato de desenvolver uma interface para o usuário de maneira a facilitar a programação do manipulador sem necessitar de componentes físicos. Além de minimização do tempo de parada da linha de produção, consequentemente potencializando a produtividade. (CRAIG, 2005).

No presente trabalho foram pesquisados vários conceitos relativos a modelagem de robôs manipuladores. Através do estudo cinemático possibilitou-se realizar uma análise sobre o controle de posição do robô.

Uma opção a cinemática inversa, seria a cinemática inversa diferencial, a qual se baseia em métodos numéricos, através da matriz jacobiana. Por sua vez esse método implica em um custo computacional muito mais elevado. Sendo assim, para se escolher qual método usar, é necessário um compromisso entre custo computacional e tempo gasto na modelagem.

Este trabalho ainda contribuiu no sentido de documentar a aplicação da solução analítica de forma detalhada a um robô. Desta maneira, tem-se uma exemplificação completa de como resolver a cinemática inversa para uma determinada classe de manipuladores. Através dos resultados de simulação, é possível comprovar que desenvolvimento relatado está correto pois as implementações confirmaram a coerência ente a cinemática direta e inversa.

A modelagem dinâmica, embora não ter sido tratada com ênfase neste trabalho, é de suma importância, quando se deseja realizar um controle de forças. Em uma aplicação real torna-se necessário a modelagem dinâmica.

As ferramentas de simulação, os *Softwares* Matlab® e Simulink®, foram de grande importância, os recursos presentes e simplicidade de implementação, proporcionam uma maior facilidade e aprofundamento no estudo. Se fosse necessário a realização dos cálculos manualmente, o tempo necessário para se obter os resultados seria inviável, sem mencionar o risco de erros em etapas intermediárias.

REFERÊNCIAS

ARAÚJO, Luciano de Oliveira. **Controle de robô com câmera móvel utilizando servovisão adaptativa**.2010.289 f. (Dissertação de mestrado) -Departamento de Engenharia Elétrica, Centro Federal de Educação Tecnológica, Rio de Janeiro,2010.

GRAIG, John J. **Introduction to robotics: mechanics and control**. 3rd ed. Upper Saddle River, NJ: Pearson Education; Prentice-Hall, c2005. viii, 400 p.

HIBBELER, R. C. **Dinâmica: mecânica para engenharia**. 12. ed. São Paulo, SP: Pearson Prentice Hall, c2011. xvi, 591 p.

INDÚSTRIA 4.0. Disponível em: <http://exame.abril.com.br/publicidade/siemens/conteudo-patrocinado/o-brasil-esta-pronto-para-a-industria-4-0?utm_source=home&utm_medium=chamada&utm_campaign=nativa-siemens> .Acesso em: 1 jun. 2016

INTERNATIONAL FEDERATION OF ROBOTICS. Disponível em: <<http://www.ifr.org/industrial-robots/statistics/>> . Acesso em: 30 mai. 2016.

LOPES, A. M. **Robótica Industrial**.2002 Dissertação (Mestrado em Automação, Instrumentação e Controle), Universidade do Porto, Porto, 2002.

MAJA, J. Mataric. **Introdução à robótica**. 1. Ed. São Paulo, SP: UNESP, 2014, 368 p.

OGATA, Katsuhiko. **Engenharia de controle moderno**. 5. ed. São Paulo, SP: Pearson Prentice Hall, c2010. x, 809 p.

PAUL, R. P. **Robot Manipulators. Mathematics, Programming and Control**, The MIT Press. 1981.

ROSÁRIO, João Maurício. **Princípios de mecatrônica**. São Paulo, SP: Prentice-Hall, 2005. x, 356 p.

SILVA, Joaquim Norberto Cardoso Pires da. **Realização de Controle de Força em Robôs Manipuladores Industriais**.1999.289 f.Tese (Doutorado em Engenharia Mecânica)- Departamento de Engenharia Mecânica, Faculdade de Ciências e Tecnologia da Universidade de Coimbra,Coimbra,1999.

SOUZA,Samantha Mendonça de. **Controle no espaço das juntas de manipuladores com jacobiano invertido utilizando servovisão**.2011.51 f.Dissertação (Mestrado em Engenharia Elétrica)- Centro Federal de Educação Tecnológica Celso Suckow da Fonseca, Rio de Janeiro, 2011.

SPONG, Mark W.; VIDYASAGAR, M.. **Robot dynamics and control**. New York: J. Wiley, c1989. 336 p.

ULLRICH, Roberto A.. **Robótica: uma introdução; o porquê dos robôs e seu papel no trabalho**. Rio de Janeiro: Campus, 1987. 93 p.

APÊNDICE 01

Cálculo da cinemática direta

```

clear all
clc
syms theta1 theta2 theta3 theta4 theta5 theta6 d0 d2 d4
% matrizes de rotação
R01 = [cos(theta1), -sin(theta1), 0; sin(theta1), cos(theta1), 0; 0, 0, 1];
R12 = [0, 0, 1; cos(theta2), -sin(theta2), 0; sin(theta2), cos(theta2), 0];
R23 = [cos(theta3), -sin(theta3), 0; sin(theta3), cos(theta3), 0; 0, 0, 1];
R34 = [0, 0, 1; cos(theta4), -sin(theta4), 0; sin(theta4), cos(theta4), 0];
R45 = [sin(theta5), cos(theta5), 0; 0, 0, 1; cos(theta5), -sin(theta5), 0];
R56 = [0, 0, 1; cos(theta6), -sin(theta6), 0; sin(theta6), cos(theta6), 0];
% vetores de translação
P01=[0;0;d0];
P12=[0;0;0];
P23=[d2;0;0];
P34=[0;0;0];
P45=[0;0;d4];
P56=[0;0;0];
% matrizes de transformação
T01=[R01 P01;0 0 0 1];
T12=[R12 P12;0 0 0 1];
T23=[R23 P23;0 0 0 1];
T34=[R34 P34;0 0 0 1];
T45=[R45 P45;0 0 0 1];
T56=[R56 P56;0 0 0 1];
% cinemática direta
T06=T01*(T12*(T23*(T34*(T45*T56))));

```

APÊNDICE 02

Gera Representação gráfica da cinemática direta

```

%% dimensoes do robo e condições iniciais
d0=50;
d2=200;
d4=200;
theta1=0;
theta2=0;
theta3=0;
theta4=0;
theta5=0;
theta6=0;
%% fixar grafico
clf;
plot3(0,0,0,'Xr');
hold on;
axis([-500 10 -10 500 -10 500]);

px5 = -sin(theta1)*(cos(theta2)*(d2 + d4*cos(theta3)) -
d4*sin(theta2)*sin(theta3));
py5 = cos(theta1)*(cos(theta2)*(d2 + d4*cos(theta3)) -
d4*sin(theta2)*sin(theta3));
pz5 = d0 + sin(theta2)*(d2 + d4*cos(theta3)) +
d4*cos(theta2)*sin(theta3);
px4= -d2*cos(theta2)*sin(theta1);
py4= d2*cos(theta1)*cos(theta2);
pz4= d0 + d2*sin(theta2);
px3= -d2*cos(theta2)*sin(theta1);
py3= d2*cos(theta1)*cos(theta2);
pz3= d0 + d2*sin(theta2);
px2=0;
py2=0;
pz2=d0;
px1=0;
py1=0;
pz1=d0;
quiver3(0,0,0,px1,py1,pz1,'b');
quiver3(px2,py2,pz2,px3-px2,py3-py2,pz3-pz2,'r');
quiver3(px4,py4,pz4,px5-px4,py5-py4,pz5-pz4,'g');

for theta1=0:0.01:pi/4
    px = -sin(theta1)*(cos(theta2)*(d2 + d4*cos(theta3)) -
d4*sin(theta2)*sin(theta3));
    py = cos(theta1)*(cos(theta2)*(d2 + d4*cos(theta3)) -
d4*sin(theta2)*sin(theta3));
    pz = d0 + sin(theta2)*(d2 + d4*cos(theta3)) +
d4*cos(theta2)*sin(theta3);
    plot3(px,py,pz,'.b');
    pause(0.01);
end

px5 = -sin(theta1)*(cos(theta2)*(d2 + d4*cos(theta3)) -
d4*sin(theta2)*sin(theta3));
py5 = cos(theta1)*(cos(theta2)*(d2 + d4*cos(theta3)) -
d4*sin(theta2)*sin(theta3));
pz5 = d0 + sin(theta2)*(d2 + d4*cos(theta3)) +
d4*cos(theta2)*sin(theta3);
px4= -d2*cos(theta2)*sin(theta1);

```

```

py4= d2*cos(thetal)*cos(theta2);
pz4= d0 + d2*sin(theta2);
px3= -d2*cos(theta2)*sin(thetal);
py3= d2*cos(thetal)*cos(theta2);
pz3= d0 + d2*sin(theta2);
quiver3(0,0,0,px1,py1,pz1,'b');
quiver3(px2,py2,pz2,px3-px2,py3-py2,pz3-pz2,'r');
quiver3(px4,py4,pz4,px5-px4,py5-py4,pz5-pz4,'g');

for theta2=0:0.01:pi/4
    px = -sin(thetal)*(cos(theta2)*(d2 + d4*cos(theta3)) -
d4*sin(theta2)*sin(theta3));
    py = cos(thetal)*(cos(theta2)*(d2 + d4*cos(theta3)) -
d4*sin(theta2)*sin(theta3));
    pz = d0 + sin(theta2)*(d2 + d4*cos(theta3)) +
d4*cos(theta2)*sin(theta3);
    plot3(px,py,pz,'.r');
    pause(0.01);
end

    px5 = -sin(thetal)*(cos(theta2)*(d2 + d4*cos(theta3)) -
d4*sin(theta2)*sin(theta3));
    py5 = cos(thetal)*(cos(theta2)*(d2 + d4*cos(theta3)) -
d4*sin(theta2)*sin(theta3));
    pz5 = d0 + sin(theta2)*(d2 + d4*cos(theta3)) +
d4*cos(theta2)*sin(theta3);
    px4= -d2*cos(theta2)*sin(thetal);
    py4= d2*cos(thetal)*cos(theta2);
    pz4= d0 + d2*sin(theta2);
    px3= -d2*cos(theta2)*sin(thetal);
    py3= d2*cos(thetal)*cos(theta2);
    pz3= d0 + d2*sin(theta2);
    quiver3(0,0,0,px1,py1,pz1,'b');
    quiver3(px2,py2,pz2,px3-px2,py3-py2,pz3-pz2,'r');
    quiver3(px4,py4,pz4,px5-px4,py5-py4,pz5-pz4,'g');

for theta3=0:0.01:pi/4
    px = -sin(thetal)*(cos(theta2)*(d2 + d4*cos(theta3)) -
d4*sin(theta2)*sin(theta3));
    py = cos(thetal)*(cos(theta2)*(d2 + d4*cos(theta3)) -
d4*sin(theta2)*sin(theta3));
    pz = d0 + sin(theta2)*(d2 + d4*cos(theta3)) +
d4*cos(theta2)*sin(theta3);
    plot3(px,py,pz,'.g');
    pause(0.01);
end

    px5 = -sin(thetal)*(cos(theta2)*(d2 + d4*cos(theta3)) -
d4*sin(theta2)*sin(theta3));
    py5 = cos(thetal)*(cos(theta2)*(d2 + d4*cos(theta3)) -
d4*sin(theta2)*sin(theta3));
    pz5 = d0 + sin(theta2)*(d2 + d4*cos(theta3)) +
d4*cos(theta2)*sin(theta3);
    px4= -d2*cos(theta2)*sin(thetal);
    py4= d2*cos(thetal)*cos(theta2);
    pz4= d0 + d2*sin(theta2);
    px3= -d2*cos(theta2)*sin(thetal);
    py3= d2*cos(thetal)*cos(theta2);
    pz3= d0 + d2*sin(theta2);
    quiver3(0,0,0,px1,py1,pz1,'b');
    quiver3(px2,py2,pz2,px3-px2,py3-py2,pz3-pz2,'r');
    quiver3(px4,py4,pz4,px5-px4,py5-py4,pz5-pz4,'g');

```

APÊNDICE 03

Rotina para obtenção da matriz jacobiana

```

%% VARIÁVEIS
syms theta1 theta2 theta3 theta4 theta5 theta6 d_theta1 d_theta2
d_theta3 d_theta4 d_theta5 d_theta6 d0 d2 d4
z=[0;0;1];
% matrizes de rotação
R01 = [cos(theta1), -sin(theta1), 0; sin(theta1), cos(theta1), 0; 0, 0, 1];
R12 = [0, 0, 1; cos(theta2), -sin(theta2), 0; sin(theta2), cos(theta2), 0];
R23 = [cos(theta3), -sin(theta3), 0; sin(theta3), cos(theta3), 0; 0, 0, 1];
R34 = [0, 0, 1; cos(theta4), -sin(theta4), 0; sin(theta4), cos(theta4), 0];
R45 = [sin(theta5), cos(theta5), 0; 0, 0, 1; cos(theta5), -sin(theta5), 0];
R56 = [0, 0, 1; cos(theta6), -sin(theta6), 0; sin(theta6), cos(theta6), 0];
R06 = R01*(R12*(R23*(R34*(R45*R56))));
% vetores de translação
P01=[0;0;d0];
P12=[0;0;0];
P23=[d2;0;0];
P34=[0;0;0];
P45=[0;0;d4];
P56=[0;0;0];
% INVERSAS DA ROTACAO
R10 = R01^-1 ;
R21 = R12^-1 ;
R32 = R23^-1 ;
R43 = R34^-1 ;
R54 = R45^-1 ;
R65 = R56^-1 ;
%% VELOCIDADES ANGULARES
W00 = [0;0;0];
W11 = R10*W00 + d_theta1*z;
W22 = R21*W11 + d_theta2*z;
W33 = R32*W22 + d_theta3*z;
W44 = R43*W33 + d_theta4*z;
W55 = R54*W44 + d_theta5*z;
W66 = R65*W55 + d_theta6*z;
%% VELOCIDADES LINEARES
V00 = [0;0;0];
V11 = R10*(V00+cross(W00,P01));
V22 = R21*(V11+cross(W11,P12));
V33 = R32*(V22+cross(W22,P23));
V44 = R43*(V33+cross(W33,P34));
V55 = R54*(V44+cross(W44,P45));
V66 = R65*(V55+cross(W55,P56));
% jacobiana
V06 = R06*V66;
W06 = R06*W66;
V06 = simplify(V06);
W06 = simplify(W06);
w1=W06(1); w2=W06(2); w3=W06(3); v1=V06(1); v2=V06(2); v3=V06(3);

JV11 = subs(v1,d_theta1,1);
JV12 = subs(v1,d_theta2,1);
JV13 = subs(v1,d_theta3,1);
JV14 = subs(v1,d_theta4,1);
JV15 = subs(v1,d_theta5,1);
JV16 = subs(v1,d_theta6,1);
JV21 = subs(v2,d_theta1,1);

```

```

JV22 = subs (v2,d_theta2,1);
JV23 = subs (v2,d_theta3,1);
JV24 = subs (v2,d_theta4,1);
JV25 = subs (v2,d_theta5,1);
JV26 = subs (v2,d_theta6,1);
JV31 = subs (v3,d_theta1,1);
JV32 = subs (v3,d_theta2,1);
JV33 = subs (v3,d_theta3,1);
JV34 = subs (v3,d_theta4,1);
JV35 = subs (v3,d_theta5,1);
JV36 = subs (v3,d_theta6,1);

```

```

JW11 = subs (w1,d_theta1,1);
JW12 = subs (w1,d_theta2,1);
JW13 = subs (w1,d_theta3,1);
JW14 = subs (w1,d_theta4,1);
JW15 = subs (w1,d_theta5,1);
JW16 = subs (w1,d_theta6,1);
JW21 = subs (w2,d_theta1,1);
JW22 = subs (w2,d_theta2,1);
JW23 = subs (w2,d_theta3,1);
JW24 = subs (w2,d_theta4,1);
JW25 = subs (w2,d_theta5,1);
JW26 = subs (w2,d_theta6,1);
JW31 = subs (w3,d_theta1,1);
JW32 = subs (w3,d_theta2,1);
JW33 = subs (w3,d_theta3,1);
JW34 = subs (w3,d_theta4,1);
JW35 = subs (w3,d_theta5,1);
JW36 = subs (w3,d_theta6,1);

```

```

for i=1:6
JV11 = subs (JV11,d_thetas(i),0);
JV12 = subs (JV12,d_thetas(i),0);
JV13 = subs (JV13,d_thetas(i),0);
JV14 = subs (JV14,d_thetas(i),0);
JV15 = subs (JV15,d_thetas(i),0);
JV16 = subs (JV16,d_thetas(i),0);
JV21 = subs (JV21,d_thetas(i),0);
JV22 = subs (JV22,d_thetas(i),0);
JV23 = subs (JV23,d_thetas(i),0);
JV24 = subs (JV24,d_thetas(i),0);
JV25 = subs (JV25,d_thetas(i),0);
JV26 = subs (JV26,d_thetas(i),0);
JV31 = subs (JV31,d_thetas(i),0);
JV32 = subs (JV32,d_thetas(i),0);
JV33 = subs (JV33,d_thetas(i),0);
JV34 = subs (JV34,d_thetas(i),0);
JV35 = subs (JV35,d_thetas(i),0);
JV36 = subs (JV36,d_thetas(i),0);

JW11 = subs (JW11,d_thetas(i),0);
JW12 = subs (JW12,d_thetas(i),0);
JW13 = subs (JW13,d_thetas(i),0);
JW14 = subs (JW14,d_thetas(i),0);
JW15 = subs (JW15,d_thetas(i),0);
JW16 = subs (JW16,d_thetas(i),0);
JW21 = subs (JW21,d_thetas(i),0);
JW22 = subs (JW22,d_thetas(i),0);
JW23 = subs (JW23,d_thetas(i),0);
JW24 = subs (JW24,d_thetas(i),0);

```



```

JW25 = subs (JW25,d_thetas(i),0);
JW26 = subs (JW26,d_thetas(i),0);
JW31 = subs (JW31,d_thetas(i),0);
JW32 = subs (JW32,d_thetas(i),0);
JW33 = subs (JW33,d_thetas(i),0);
JW34 = subs (JW34,d_thetas(i),0);
JW35 = subs (JW35,d_thetas(i),0);
JW36 = subs (JW36,d_thetas(i),0);
end
JV11 = simplify (JV11);
JV12 = simplify (JV12);
JV13 = simplify (JV13);
JV14 = simplify (JV14);
JV15 = simplify (JV15);
JV16 = simplify (JV16);

JV21 = simplify (JV21);
JV22 = simplify (JV22);
JV23 = simplify (JV23);
JV24 = simplify (JV24);
JV25 = simplify (JV25);
JV26 = simplify (JV26);

JV31 = simplify (JV31);
JV32 = simplify (JV32);
JV33 = simplify (JV33);
JV34 = simplify (JV34);
JV35 = simplify (JV35);
JV36 = simplify (JV36);

JW11 = simplify (JW11);
JW12 = simplify (JW12);
JW13 = simplify (JW13);
JW14 = simplify (JW14);
JW15 = simplify (JW15);
JW16 = simplify (JW16);

JW21 = simplify (JW21);
JW22 = simplify (JW22);
JW23 = simplify (JW23);
JW24 = simplify (JW24);
JW25 = simplify (JW25);
JW26 = simplify (JW26);

JW31 = simplify (JW31);
JW32 = simplify (JW32);
JW33 = simplify (JW33);
JW34 = simplify (JW34);
JW35 = simplify (JW35);
JW36 = simplify (JW36);

JV =
[JV11, JV12, JV13, JV14, JV15, JV16; JV21, JV22, JV23, JV24, JV25, JV26; JV31, JV32, JV33
, JV34, JV35, JV36];
JW =
[JW11, JW12, JW13, JW14, JW15, JW16; JW21, JW22, JW23, JW24, JW25, JW26; JW31, JW32, JW33
, JW34, JW35, JW36];
J = [JV; JW]

```

APÊNDICE 04

Função para geração de trajetória no Simulink®

```
function [theta1,theta2,theta3,theta4,theta5,theta6] =
fcn(thetasInic,thetasFIM,TemposMax,t)
%#eml

thi1 = thetasInic(1);
thi2 = thetasInic(2);
thi3 = thetasInic(3);
thi4 = thetasInic(4);
thi5 = thetasInic(5);
thi6 = thetasInic(6);

thf1 = thetasFIM(1);
thf2 = thetasFIM(2);
thf3 = thetasFIM(3);
thf4 = thetasFIM(4);
thf5 = thetasFIM(5);
thf6 = thetasFIM(6);

tmax1 = TemposMax(1);
tmax2 = TemposMax(2);
tmax3 = TemposMax(3);
tmax4 = TemposMax(4);
tmax5 = TemposMax(5);
tmax6 = TemposMax(6);

a10=thi1;
a11=0;
a12=0;
a13 =10*(thf1-thi1)/(tmax1.^3);
a14 =-15*(thf1-thi1)/(tmax1.^4);
a15 =6*(thf1-thi1)/(tmax1.^5);

a20=thi2;
a21=0;
a22=0;
a23 =10*(thf2-thi2)/(tmax2.^3);
a24 =-15*(thf2-thi2)/(tmax2.^4);
a25 =6*(thf2-thi2)/(tmax2.^5);

a30=thi3;
a31=0;
a32=0;
a33 =10*(thf3-thi3)/(tmax3.^3);
a34 =-15*(thf3-thi3)/(tmax3.^4);
a35 =6*(thf3-thi3)/(tmax3.^5);

a40=thi4;
a41=0;
a42=0;
a43 =10*(thf4-thi4)/(tmax4.^3);
a44 =-15*(thf4-thi4)/(tmax4.^4);
a45 =6*(thf4-thi4)/(tmax4.^5);

a50=thi5;
```

```

a51=0;
a52=0;
a53 =10*(thf5-thi5)/(tmax5.^3);
a54 =-15*(thf5-thi5)/(tmax5.^4);
a55 =6*(thf5-thi5)/(tmax5.^5);

a60=thi6;
a61=0;
a62=0;
a63 =10*(thf6-thi6)/(tmax6.^3);
a64 =-15*(thf6-thi6)/(tmax6.^4);
a65 =6*(thf6-thi6)/(tmax6.^5);

if t>tmax1
    t1 = tmax1;
else
    t1 = t;
end;

if t>tmax2
    t2 = tmax2;
else
    t2 = t;
end;

if t>tmax3
    t3 = tmax3;
else
    t3 = t;
end;

if t>tmax4
    t4 = tmax4;
else
    t4 = t;
end;

if t>tmax5
    t5 = tmax5;
else
    t5 = t;
end;

if t>tmax6
    t6 = tmax6;
else
    t6 = t;
end;

theta1 = a15*t^5+a14*t^4+a13*t^3+a12*t^2+a11*t+a10;
theta2 = a25*t^5+a24*t^4+a23*t^3+a22*t^2+a21*t+a20;
theta3 = a35*t^5+a34*t^4+a33*t^3+a32*t^2+a31*t+a30;
theta4 = a45*t^5+a44*t^4+a43*t^3+a42*t^2+a41*t+a40;
theta5 = a55*t^5+a54*t^4+a53*t^3+a52*t^2+a51*t+a50;
theta6 = a65*t^5+a64*t^4+a63*t^3+a62*t^2+a61*t+a60;

```

APÊNDICE 05

Simula a jacobiana

```

clear all;
close all;
clc;

% dimensoes/parametros
d0=50;
d2=200;
d4=200;

% condições iniciais
theta1i=0;
theta2i=0;
theta3i=0;
theta4i=0;
theta5i=0;
theta6i=0;
x=[0;0;0;0;0;0];

% condições finais
theta1f=pi/2;
theta2f=3*pi/4;
theta3f=-3*pi/4;
theta4f=0;
theta5f=0;
theta6f=0;

% parametros de simulação
t_max=20;
delta=0.001;
n_max=t_max./delta+1;

% polimonios theta
a01=theta1i;
a11=0;
a21=3*(theta1f-theta1i)/t_max.^2;
a31 =2*(theta1f-theta1i)/t_max.^3;

a02=theta2i;
a12=0;
a22=3*(theta2f-theta2i)/t_max.^2;
a32 =2*(theta2f-theta2i)/t_max.^3;

a03=theta3i;
a13=0;
a23=3*(theta3f-theta3i)/t_max.^2;
a33 =2*(theta3f-theta3i)/t_max.^3;

a04=theta4i;
a14=0;
a24=3*(theta4f-theta4i)/t_max.^2;
a34 =2*(theta4f-theta4i)/t_max.^3;

a05=theta5i;
a15=0;

```

```

a25=3*(theta5f-theta5i)/t_max.^2;
a35 =2*(theta5f-theta5i)/t_max.^3;

a06=theta6i;
a16=0;
a26=3*(theta6f-theta6i)/t_max.^2;
a36 =2*(theta6f-theta6i)/t_max.^3;

%% posição velocidade e aceleração de theta
cont = 0;
figure(1);
hold on
for i=1:n_max;
    t_int(i) = (i-1).*delta;

    theta1 = a01 + a11*t_int(i) + a21*t_int(i)^2 + a31*t_int(i)^3;
    d_theta1 = a11 + 2*a21*t_int(i) + 3*a31*t_int(i)^2;
    dd_theta1 = 2*a21 + 6*a31*t_int(i);

    theta2 = a02 + a12*t_int(i) + a22*t_int(i)^2 + a32*t_int(i)^3;
    d_theta2 = a12 + 2*a22*t_int(i) + 3*a32*t_int(i)^2;
    dd_theta2 = 2*a22 + 6*a32*t_int(i);

    theta3 = a03 + a13*t_int(i) + a23*t_int(i)^2 + a33*t_int(i)^3;
    d_theta3 = a13 + 2*a23*t_int(i) + 3*a33*t_int(i)^2;
    dd_theta3 = 2*a23 + 6*a33*t_int(i);

    theta4 = a04 + a14*t_int(i) + a24*t_int(i)^2 + a34*t_int(i)^3;
    d_theta4 = a14 + 2*a24*t_int(i) + 3*a34*t_int(i)^2;
    dd_theta4 = 2*a24 + 6*a34*t_int(i);

    theta5 = a05 + a15*t_int(i) + a25*t_int(i)^2 + a35*t_int(i)^3;
    d_theta5 = a15 + 2*a25*t_int(i) + 3*a35*t_int(i)^2;
    dd_theta5 = 2*a25 + 6*a35*t_int(i);

    theta6 = a06 + a16*t_int(i) + a26*t_int(i)^2 + a36*t_int(i)^3;
    d_theta6 = a16 + 2*a26*t_int(i) + 3*a36*t_int(i)^2;
    dd_theta6 = 2*a26 + 6*a36*t_int(i);

    d_theta=[d_theta1;d_theta2;d_theta3;d_theta4;d_theta5;d_theta6];
    dd_theta=[dd_theta1;dd_theta2;dd_theta3;dd_theta4;dd_theta5;dd_theta6];

%% jacobiana
J = [ -cos(theta1)*(d4*cos(theta2 + theta3) + d2*cos(theta2)),
    sin(theta1)*(d4*sin(theta2 + theta3) + d2*sin(theta2)), d4*sin(theta2 +
theta3)*sin(theta1),0,0,0; -sin(theta1)*(d4*cos(theta2 + theta3) +
d2*cos(theta2)), -cos(theta1)*(d4*sin(theta2 + theta3) + d2*sin(theta2)), -
d4*sin(theta2 + theta3)*cos(theta1), 0, 0, 0; 0, d4*cos(theta2 + theta3) +
d2*cos(theta2), d4*cos(theta2 + theta3), 0, 0, 0; 0, 0, 0, -cos(theta2 +
theta3)*sin(theta1), cos(theta1)*cos(theta4) -
cos(theta2)*sin(theta1)*sin(theta3)*sin(theta4) -
cos(theta3)*sin(theta1)*sin(theta2)*sin(theta4),cos(theta1)*sin(theta4)*sin
(theta5) - cos(theta2)*cos(theta3)*cos(theta5)*sin(theta1) +
cos(theta5)*sin(theta1)*sin(theta2)*sin(theta3) +
cos(theta2)*cos(theta4)*sin(theta1)*sin(theta3)*sin(theta5) +
cos(theta3)*cos(theta4)*sin(theta1)*sin(theta2)*sin(theta5); 0, 0, 0,
cos(theta2 + theta3)*cos(theta1), cos(theta4)*sin(theta1) +
cos(theta1)*cos(theta2)*sin(theta3)*sin(theta4) +
cos(theta1)*cos(theta3)*sin(theta2)*sin(theta4),

```

```

sin(theta1)*sin(theta4)*sin(theta5) +
cos(theta1)*cos(theta2)*cos(theta3)*cos(theta5) -
cos(theta1)*cos(theta5)*sin(theta2)*sin(theta3) -
cos(theta1)*cos(theta2)*cos(theta4)*sin(theta3)*sin(theta5) -
cos(theta1)*cos(theta3)*cos(theta4)*sin(theta2)*sin(theta5);
0, 0, 0, sin(theta2 + theta3), -cos(theta2 + theta3)*sin(theta4),
cos(theta2)*cos(theta5)*sin(theta3) + cos(theta3)*cos(theta5)*sin(theta2) +
cos(theta2)*cos(theta3)*cos(theta4)*sin(theta5) -
cos(theta4)*sin(theta2)*sin(theta3)*sin(theta5)];
d_x = J*d_theta;
x = x + d_x*delta;

% matrizes de rotaçao
R01 = [cos(theta1), -sin(theta1), 0; sin(theta1), cos(theta1), 0; 0, 0, 1];
R12 = [0, 0, 1; cos(theta2), -sin(theta2), 0; sin(theta2), cos(theta2), 0];
R23 = [cos(theta3), -sin(theta3), 0; sin(theta3), cos(theta3), 0; 0, 0, 1];
R34 = [0, 0, 1; cos(theta4), -sin(theta4), 0; sin(theta4), cos(theta4), 0];
R45 = [sin(theta5), cos(theta5), 0; 0, 0, 1; cos(theta5), -sin(theta5), 0];
R56 = [0, 0, 1; cos(theta6), -sin(theta6), 0; sin(theta6), cos(theta6), 0];
% vetores de translaçao
P01=[0;0;d0];
P12=[0;0;0];
P23=[d2;0;0];
P34=[0;0;0];
P45=[0;0;d4];
P56=[0;0;0];
% matrizes de transformaçao
T01=[R01 P01;0 0 0 1];
T12=[R12 P12;0 0 0 1];
T23=[R23 P23;0 0 0 1];
T34=[R34 P34;0 0 0 1];
T45=[R45 P45;0 0 0 1];
T56=[R56 P56;0 0 0 1];
% cinematica direta
T06=T01*(T12*(T23*(T34*(T45*T56))));
subplot(2,1,1),hold on;plot3(T06(1,4),T06(2,4),T06(3,4),'.b');
subplot(2,1,2),hold on;plot3(x(1),x(2)+400,x(3)+50,'.k');
end;
subplot(2,1,1),title('Geraçao de trajetoria através da cinematica direta');
xlabel('eixo x');ylabel('eixo y');zlabel('eixo z');
subplot(2,1,2),title('Geraçao de trajetoria através da jacobiana');
xlabel('eixo x');ylabel('eixo y');zlabel('eixo z');

```

APÊNDICE 06

Gera equações dinâmicas

```

clear all;
clc;
%% variaveis
syms theta1 theta2 theta3 theta4 theta5 theta6 d_theta1 d_theta2 d_theta3
d_theta4 d_theta5 d_theta6 dd_theta1 dd_theta2 dd_theta3 dd_theta4
dd_theta5 dd_theta6
syms x y z

d0=50;d2=200;d4=200;

m1 = 0.949652;
m2 = 4.7958;
m3 = 0.599553;
m4 = 3.50965;
m5 = 0.757258;
m6 = 0.220345;

Z = [0 0 1]';
W00 = [0 0 0]';
Wp00 = [0 0 0]';
f77 = [0 0 0]';
n77 = [0 0 0]';

Vp00 = [0 0 0]';

PC11 = [0 0 d0/2]';
PC22 = [0 0 0]';
PC33 = [d2/2 0 0]';
PC44 = [0 0 0]';
PC55 = [d4/2 0 0]';
PC66 = [0 0 0]';

%% tensores de inercia
Ic11 = [0.00496573,0,0;0,0.0092531,3.88693e-05;0,3.88693e-05,0.00729659];%
[xx1,0,0;0,yy1,0;0,0,zz1];
Ic22 = [0.0347574,0,0;0,0.017662,-0.0004104;0,-0.0004104,0.0444145];%
[xx2,0,0;0,yy2,0;0,0,zz2];
Ic33 =
[0.000562311,0,0;0,0.00156905,0;0,0,0.00124274];%[xx3,0,0;0,yy3,0;0,0,zz3];
Ic44 =
[0.0221389,0,0;0,0.0103492,0;0,0,0.0266679];%[xx4,0,0;0,yy4,0;0,0,zz4];
Ic55 =
[0.000938532,0,0;0,0.00119371,0;0,0,0.00116177];%[xx5,0,0;0,yy5,0;0,0,zz5];
Ic66 = [0.000182353,-1.49956e-11,-2.97718e-10;-1.49956e-11,0.000131259,-
3.21085e-08;-2.97718e-10,-3.21085e-08,0.000255197];

% matrizes de rotação
R01 = [cos(theta1),-sin(theta1),0;sin(theta1),cos(theta1),0;0,0,1];
R12 = [0,0,1;cos(theta2),-sin(theta2),0;sin(theta2),cos(theta2),0];
R23 = [cos(theta3),-sin(theta3),0;sin(theta3),cos(theta3),0;0,0,1];
R34 = [0,0,1;cos(theta4),-sin(theta4),0;sin(theta4),cos(theta4),0];
R45 = [sin(theta5),cos(theta5),0;0,0,1;cos(theta5),-sin(theta5),0];
R56 = [0,0,1;cos(theta6),-sin(theta6),0;sin(theta6),cos(theta6),0];
R67 = [1 0 0;0 1 0;0 0 1];

```

```

% vetores de translação
P01=[0;0;d0];
P12=[0;0;0];
P23=[d2;0;0];
P34=[0;0;0];
P45=[0;0;d4];
P56=[0;0;0];
P67=[0;0;0];
% matrizes de transformação
R10 = R01^-1;
R21 = R12^-1;
R32 = R23^-1;
R43 = R34^-1;
R54 = R45^-1;
R65 = R56^-1;

%% 1ª Iteração
W11 = R10*W00+d_theta1*Z;
Wp11 = R10*Wp00+cross(R10*W00,d_theta1*Z)+dd_theta1*Z;
Vp11 = R10*(cross(Wp00,P01)+cross(W00,cross(W00,P01))+Vp00);
VCp11 = cross(Wp11,PC11)+cross(W11,cross(W11,PC11))+Vp11;
F11 = m1*VCp11;
N11 = Ic11*Wp11+cross(W11,(Ic11*W11));

%% 2ª Iteração
W22 = R21*W11+d_theta2*Z;
Wp22 = R21*Wp11+cross(R21*W11,d_theta2*Z)+dd_theta2*Z;
Vp22 = R21*(cross(Wp11,P12)+cross(W11,cross(W11,P12))+Vp11);
VCp22 = cross(Wp22,PC22)+cross(W22,cross(W22,PC22))+Vp22;
F22 = m2*VCp22;
N22 = Ic22*Wp22+cross(W22,(Ic22*W22));

%% 3ª Iteração
W33 = R32*W22+d_theta3*Z;
Wp33 = R32*Wp22+cross(R32*W22,d_theta3*Z)+dd_theta3*Z;
Vp33 = R32*(cross(Wp22,P23)+cross(W22,cross(W22,P23))+Vp22);
VCp33 = cross(Wp33,PC33)+cross(W33,cross(W33,PC33))+Vp33;
F33 = m3*VCp33;
N33 = Ic33*Wp33+cross(W33,(Ic33*W33));

%% 4ª Iteração
W44 = R43*W33+d_theta4*Z;
Wp44 = R43*Wp33+cross(R43*W33,d_theta4*Z)+dd_theta4*Z;
Vp44 = R43*(cross(Wp33,P34)+cross(W33,cross(W33,P34))+Vp33);
VCp44 = cross(Wp44,PC44)+cross(W44,cross(W44,PC44))+Vp44;
F44 = m4*VCp44;
N44 = Ic44*Wp44+cross(W44,(Ic44*W44));

%% 5ª Iteração
W55 = R54*W44+d_theta5*Z;
Wp55 = R54*Wp44+cross(R54*W44,d_theta5*Z)+dd_theta5*Z;
Vp55 = R54*(cross(Wp44,P45)+cross(W44,cross(W44,P45))+Vp44);
VCp55 = cross(Wp55,PC55)+cross(W55,cross(W55,PC55))+Vp55;
F55 = m5*VCp55;
N55 = Ic55*Wp55+cross(W55,(Ic55*W55));

%% 6ª Iteração
W66 = R65*W55+d_theta6*Z;
Wp66 = R65*Wp55+cross(R65*W55,d_theta6*Z)+dd_theta6*Z;
Vp66 = R65*(cross(Wp55,P56)+cross(W55,cross(W55,P56))+Vp55);

```



```

VCp66 = cross(Wp66,PC66)+cross(W66,cross(W66,PC66))+Vp66;
F66 = m6*VCp66;
N66 = Ic66*Wp66+cross(W66,(Ic66*W66));

```

```

%% 1ª Iteração inversa

```

```

f66 = R67*f77+F66;
n66 = N66+R67*n77+cross(PC66,F66)+cross(P67,(R67*f77));
t6=n66'*Z;

```

```

%% 2ª Iteração inversa

```

```

f55 = R56*f66+F55;
n55 = N55+R56*n66+cross(PC55,F55)+cross(P56,(R56*f66));
t5=n55'*Z;

```

```

%% 3ª Iteração inversa

```

```

f44 = R45*f55+F44;
n44 = N44+R45*n55+cross(PC44,F44)+cross(P45,(R45*f55));
t4=n44'*Z;

```

```

%% 4ª Iteração inversa

```

```

f33 = R34*f44+F33;
n33 = N33+R34*n44+cross(PC33,F33)+cross(P34,(R34*f44));
t3=n33'*Z;

```

```

%% 5ª Iteração inversa

```

```

f22 = R23*f33+F22;
n22 = N22+R23*n33+cross(PC22,F22)+cross(P23,(R23*f33));
t2=n22'*Z;

```

```

%% 6ª Iteração inversa

```

```

f11 = R12*f22+F11;
n11 = N11+R12*n22+cross(PC11,F11)+cross(P12,(R12*f22));
t1=n11'*Z;

```

```

%% simplificações

```

```

t1=simplify(t1);t2=simplify(t2);t3=simplify(t3);t4=simplify(t4);t5=simplify
(t5);t6=simplify(t6);

```