

**MINISTÉRIO DA DEFESA
EXÉRCITO BRASILEIRO
DEPARTAMENTO DE CIÊNCIA E TECNOLOGIA
INSTITUTO MILITAR DE ENGENHARIA
(Real Academia de Artilharia, Fortificação e Desenho/1792)
SEÇÃO DE ENGENHARIA MECÂNICA E DE MATERIAIS**

**Amanda de Oliveira Barros
Bruno Monteiro Rocha Lima
Caio Bittencourt Cardoso Felix
Thiago Augusto Souza da Piedade**

**Resolução Cinemática do
Robô Industrial MOTOMAN MH5F**

**Brasil
05 de outubro de 2017**

INSTITUTO MILITAR DE ENGENHARIA

**Resolução Cinemática do
Robô Industrial MOTOMAN MH5F**

Projeto Final de Curso de Graduação em
Engenharia Mecânica do Instituto Militar de
Engenharia.

Orientador: Luiz Paulo Gomes Ribeiro

Brasil
05 de outubro de 2017

INSTITUTO MILITAR DE ENGENHARIA
Praça General Tibúrcio, 80 - Praia Vermelha
Rio de Janeiro-RJ CEP 22290-270

Este exemplar é de propriedade do Instituto Militar de Engenharia, que poderá incluí-lo em base de dados, armazenar em computador, microfilmar ou adotar qualquer forma de arquivamento.

É permitida a menção, reprodução parcial ou integral e a transmissão entre bibliotecas deste trabalho, sem modificação de seu texto, em qualquer meio que esteja ou venha a ser fixado, para pesquisa acadêmica, comentários e citações, desde que sem finalidade comercial e que seja feita a referência bibliográfica completa.

Os conceitos expressos neste trabalho são de responsabilidade do(s) autor(es) e do(s) orientador(es).

620.1 Barros, Amanda de Oliveira
B277r Resolução Cinemática do Robô Industrial Motoman
MH5F / Amanda de Oliveira Barros, Bruno Monteiro
Rocha Lima, Caio Bittencourt Cardoso Felix, Thiago
Augusto Souza da Piedade, orientados por Luiz Paulo
Gomes Ribeiro - Rio de Janeiro: Instituto Militar de
Engenharia, 2017.

103p. : il.

PROFIC (Projeto de Fim de Curso) - Instituto Militar
de Engenharia - Rio de Janeiro, 2017.

1. Curso de Engenharia Mecânica - Projeto de Fim
de Curso. 2. Robótica Industrial. 3. Cinemática. I.
Barros, Amanda de Oliveira. II. Felix, Caio Bittencourt
Cardoso. III. Lima, Bruno Monteiro Rocha. IV. Piedade,
Thiago Augusto Souza da. V. Ribeiro, Luiz Paulo Gomes.
VI. Resolução Cinemática do Robô Motoman MH5F. VII.
Instituto Militar de Engenharia.

INSTITUTO MILITAR DE ENGENHARIA

**Amanda de Oliveira Barros
Bruno Monteiro Rocha Lima
Caio Bittencourt Cardoso Felix
Thiago Augusto Souza da Piedade**

Resolução Cinemática do Robô Industrial MOTOMAN MH5F


Projeto Final de Curso de Graduação em Engenharia Mecânica do Instituto Militar de Engenharia.

Orientador: Luiz Paulo Gomes Ribeiro


Aprovado em 05 de outubro de 2017 pela seguinte Banca Examinadora:



Luiz Paulo Gomes Ribeiro - Dr.Eng.
Orientador



Maurício Gruzman - D.Sc.



Elias Dias Rossi Lopes - M. C.

Brasil, 05 de outubro de 2017

Agradecimentos

A Deus.

Ao Prof. Luiz Ribeiro pela dedicação ininterrupta e pelas horas investidas no sucesso deste trabalho.

Às famílias, alicerce de nossas vidas. Por acreditar nos sonhos compartilhados e por suportar os momentos de distanciamento durante este curso de graduação. Vocês foram parte fundamental desta etapa de nossas vidas.

Resumo

A robótica industrial apresenta alto potencial de aplicação dual, auferindo vantagens em muitas atividades ligadas tanto aos setores civis quanto aos associados à Defesa. A pesquisa, o desenvolvimento, a busca por inovações e a capacitação de pessoal nesta área do conhecimento é considerada estratégica nos países mais desenvolvidos que utilizam robôs manipuladores em grande número, principalmente, por possibilitar a substituição do emprego de seres humanos em atividades repetitivas, insalubres e/ou perigosas. Além disso, permite um aumento considerável da produtividade total dos processos passíveis de serem automatizados. O objetivo principal deste trabalho é obter a resolução cinemática do robô industrial MOTOMAN MH5F, utilizando os fundamentos da Teoria dos Helicoides. O trabalho apresenta revisão bibliográfica de robótica industrial e dos fundamentos de programação *on-line*, e de modelagem cinemática de robôs, focando a abordagem que utiliza a Teoria dos Helicoides, além de um estudo de caso aplicado, utilizando o Robô Industrial MOTOMAN MH5F disponível no Laboratório de Mecatrônica do IME. Como resultado, foi realizada a implementação de uma ferramenta computacional de Interface Homem Máquina simples e robusta, o CARPA.

Palavras-chave: Robótica industrial. Motoman. Teoria dos Helicoides. Cinemática. Programação de robôs

Abstract

Industrial robotics presents a high potential of dual application, earning advantages in many activities linked to the civil and defense sectors. The research, development and chase for innovation and the training of personnel in this subject is considered strategic in develop countries that utilize robot manipulators in large scale, specially for replacing the human labor in repetitive, unhealthy and or dangerous circumstances. Furthermore, it allows a great increase in total productivity of the processes in which automation is possible. The main goal of this work is to obtain the kinematic solution of the industrial robot MOTOMAN MH5F, using Screw Theory fundamentals. The work presents a literature review of industrial robotics and the fundamentals of on-line programming, kinematic model of robots, focusing in the Screw Theory approach, in addition to an applied case study using the industrial robot MOTOMAN MH5F available at IME's Mechatronics Laboratory. As a result, a computational tool, designed to be an Man Machine interface, is implemented to facilitate the programming of the industrial robot studied.

Keywords: Industrial robotics. Motoman. Screw Theory. Kinematic. Robot programming.

Sumário

1	INTRODUÇÃO	16
1.1	Motivação	16
1.2	Formulação do problema	19
1.3	Objetivos do trabalho	19
1.4	Organização do trabalho	19
2	ROBÓTICA INDUSTRIAL	20
2.1	Robótica	20
2.1.1	Breve Histórico	20
2.1.2	O que é Robótica?	21
2.2	A Robótica Industrial	22
2.2.1	Estrutura mecânica de robôs industriais	22
2.3	Programação de Robôs Industriais	26
2.3.1	Programação <i>On-line</i>	27
2.3.2	Programação <i>Off-line</i>	28
2.4	O robô Yaskawa Motoman MH5F	29
2.4.1	O Controlador FS-100	31
3	CINEMÁTICA DE MANIPULADORES SERIAIS	33
3.1	Introdução	33
3.2	Localização de corpo rígido no espaço	34
3.2.1	Matrizes de rotação	35
3.2.2	Ângulos de Euler	37
3.2.3	Transformação homogênea	38
3.3	Teoria dos Helicoides aplicada à robótica	39
3.3.1	Rotação em torno de um eixo de Helicoide	41
3.3.2	Deslocamento Helicoidal	42
3.3.3	Método dos deslocamentos helicoidais sucessivos	44
3.4	Cinemática direta do robô Motoman MH5F	46
3.4.1	Orientação do Efetuador Final	49
3.5	Cinemática inversa	50
3.5.1	Cinemática diferencial	50
3.5.2	Jacobiano baseado em Helicoides	51
3.5.3	Jacobiano do robô Motoman MH5F	51
3.5.4	Algoritmo de cinemática inversa	56

4	RESULTADOS	58
4.1	Simulação de trajetória	58
4.1.1	Trajétória linear (MOVL)	58
4.1.2	Trajétória circular (MOVC)	64
4.2	Software de simulação de robôs seriais	68
4.3	Modelagem do ambiente virtual de simulação	72
5	CONCLUSÕES E PERSPECTIVAS	75
5.1	Recapitulação sintetizada	75
5.2	Conclusões	76
5.3	Perspectivas	76
	 APÊNDICES	 77
	APÊNDICE A – PROCEDIMENTOS DE SEGURANÇA	78
	APÊNDICE B – PROCEDIMENTO DE PROGRAMAÇÃO ON-LINE DO SISTEMA MOTOMAN (FS-100)	80
B.0.1	Operações nos sistemas de coordenadas do robô	82
B.0.2	Ensinando Tarefas	86
	APÊNDICE C – COMANDOS DA LINGUAGEM INFORM III	95
	REFERÊNCIAS	101

Lista de ilustrações

Figura 1 – Vendas anuais de robôs industriais (IFR - 2016)	18
Figura 2 – Componentes de um sistema robótico (adaptado de Siciliano et al. (2010))	21
Figura 3 – Tipos de juntas elementares (adaptado de Jazar (2010))	22
Figura 4 – Manipulador cartesiano e seu espaço de trabalho (SICILIANO et al., 2010)	23
Figura 5 – Manipulador cilíndrico e seu espaço de trabalho (SICILIANO et al., 2010)	24
Figura 6 – Manipulador esférico e seu espaço de trabalho (SICILIANO et al., 2010)	24
Figura 7 – Manipulador SCARA e seu espaço de trabalho (SICILIANO et al., 2010)	25
Figura 8 – Manipulador antropomórfico e seu espaço de trabalho (SICILIANO et al., 2010)	25
Figura 9 – Robô Yaskawa Motoman MH5F e suas juntas de rotação	26
Figura 10 – Algoritmos de controle de movimento para determinação do caminho real adotado quando o programa é repetido no modo automático (NOF, 1999)	28
Figura 11 – Componentes do sistema robótico Motoman MH5F	30
Figura 12 – Controlador FS-100 (adaptado de YASKAWA (2014))	31
Figura 13 – Composição dos comandos da INFORM III (adaptado de YASKAWA (2014))	32
Figura 14 – Posição e orientação de um corpo rígido (SICILIANO et al., 2010) . . .	34
Figura 15 – Rotação do sistema $O-xyz$ de um ângulo α em torno de z (SICILIANO et al., 2010)	35
Figura 16 – Representação de n sucessivas rotações	37
Figura 17 – Representação da localização do ponto p	38
Figura 18 – Representação da rotação do ponto P (TSAI, 1999)	41
Figura 19 – Projeção de $S_P P_2^r$ sobre $S_P P_1 P$ (TSAI, 1999)	41
Figura 20 – Representação de movimento de rotação e translação em torno de um único eixo	43
Figura 21 – Sucessão de n juntas	44
Figura 22 – Algoritmo para a solução da Cinemática Direta baseada em Helicoides	46
Figura 23 – Representação dos helicoides no robô Yaskawa Motoman MH5F	47
Figura 24 – Orientação de referência do efetuador final	49
Figura 25 – Representação da Cadeia Virtual de Assur para o Motoman MH5F . .	53
Figura 26 – Grafo das juntas do manipulador Motoman MH5F	53
Figura 27 – Algoritmo para o cálculo do Jacobiano baseado em Helicoides para o manipuladores com 6 DOF	55

Figura 28 – Algoritmo para a para a solução da cinemática inversa de manipuladores com 6 DOF	57
Figura 29 – Variação de q ao longo do tempo	59
Figura 30 – Variação de \dot{q} ao longo do tempo	59
Figura 31 – Erro de posição ao longo do tempo	60
Figura 32 – Erro de orientação ao longo do tempo	60
Figura 33 – Variação dos ângulos das juntas ao longo do tempo	62
Figura 34 – Variação da velocidade angular das juntas ao longo do tempo	62
Figura 35 – Trajetória triangular obtida	63
Figura 36 – Erro de orientação na trajetória em função do tempo	63
Figura 37 – Erro de posição na trajetória em função do tempo	64
Figura 38 – Variação dos ângulos das juntas ao longo do tempo	65
Figura 39 – Variação da velocidade angular das juntas ao longo do tempo	66
Figura 40 – Trajetória circular obtida	66
Figura 41 – Erro de orientação na trajetória em função do tempo	67
Figura 42 – Erro de posição na trajetória em função do tempo	67
Figura 43 – Tela inicial do Carpa com o Robô Yaskawa MOTOMAN MH5F	68
Figura 44 – Juntas e helicóide relativo à junta selecionada	69
Figura 45 – Ponto ensinado em destaque	70
Figura 46 – Funcionamento da janela de comandos	71
Figura 47 – Pontos que determinam a trajetória triangular obtidos no CARPA	71
Figura 48 – Pontos que determinam a trajetória circular obtidos no CARPA	72
Figura 49 – Vista com estação do centro de usinagem	73
Figura 50 – Vista com estação do centro de soldagem	73
Figura 51 – Vista com estação do metrologia computadorizada	74
Figura 52 – Desarmando o botão de parada de emergência	79
Figura 53 – Visão geral do <i>Programming Pendant</i> , (adaptado de YASKAWA (2014))	81
Figura 54 – Sistema de coordenadas das juntas (YASKAWA (2014))	83
Figura 55 – Sistema de coordenadas cartesianas e a descrição do movimento (adaptado de YASKAWA (2014))	83
Figura 56 – Sistema de coordenadas cilíndricas e a descrição do movimento (adaptado de YASKAWA (2014))	84
Figura 57 – Sistema de coordenadas da ferramenta (adaptado de YASKAWA (2014))	84
Figura 58 – Sistema de coordenadas do usuário (adaptado de YASKAWA (2014))	85
Figura 59 – Operação por Controle de Ponto (adaptado de YASKAWA (2014))	85
Figura 60 – Janelas de Registro de Trabalho (adaptado de YASKAWA (2014))	87
Figura 61 – Alteração para janela de ensino de tarefas (adaptado de YASKAWA (2014))	87

Figura 62 – Apresentação da janela “JOB CONTENT” (adaptado de YASKAWA (2014))	88
Figura 63 – Esboço da trajetória linear desenvolvida pelo manipulador em trabalho de soldagem (adaptado de YASKAWA (2014))	89
Figura 64 – Exemplo simples de conjunto de tarefas e seus respectivos comandos (adaptado de YASKAWA (2014))	90
Figura 65 – Exemplo de trabalho com suas etapas (adaptado de YASKAWA (2014))	91
Figura 66 – Passos gerais para programação do trabalho proposto (adaptado de YASKAWA (2014))	91
Figura 67 – Conjunto de etapas e a instrução respectiva na janela “JOB CONTENT” (adaptado de YASKAWA (2014))	92
Figura 68 – Alteração de velocidade para “MÉDIO” em operação de ensino (adaptado de YASKAWA (2014))	93
Figura 69 – Passos para validação de trabalho (adaptado de YASKAWA (2014)) . .	94

Lista de tabelas

Tabela 1 – Vetores s e s_0 de cada junta para o Robô Yaskawa Motoman MH5F . . .	47
Tabela 2 – Tipo de interpolação para arco circular individual	89
Tabela 3 – Tipo de interpolação para arcos circulares contínuos	90
Tabela 4 – Instruções de Movimento (adaptado de YASKAWA (2014))	95
Tabela 5 – Instruções de <i>Inputs</i> e <i>Outputs</i> (adaptado de YASKAWA (2014)) . . .	96
Tabela 6 – Instruções de controle (adaptado de YASKAWA (2014))	97
Tabela 7 – Instruções de Deslocamento (adaptado de YASKAWA (2014))	98
Tabela 8 – Itens adicionais	98
Tabela 9 – Itens adicionais (continuação)	99
Tabela 10 – Itens adicionais (continuação)	100

Lista de abreviaturas e siglas

ARM	Algoritmo Advanced Robot Motion
CAD	<i>Computer Aided Design</i>
CAM	<i>Computer Aided Manufacturing</i>
CLP	Controlador Lógico Programável
CVA	Cadeias Virtuais de Assur
DOF	<i>Degree of Freedom</i> , graus de liberdade
EF	Efetuator Final
IDE	<i>Integrated Development Environment</i>
IDR-Lab	<i>Industrial and Defense Robotics Laboratory</i>
IFR	<i>International Federation of Robotics</i>
I/O's	<i>Inputs/Outputs</i>
OLP	<i>Off-line</i> Programming
PD&I	Pesquisa, Desenvolvimento e Inovação
PIB	Produto Interno Bruto
R.U.R	<i>Rossum's Universal Robots</i>
SCARA	<i>Selective Compliance Articulated Robot Arm</i>
TCP	<i>Tool Center Point</i> , ponto de referência para o sistema de coordenadas do efetuator final instalado no flange
USB	<i>Universal Serial Bus</i>

Lista de símbolos

$B_{l,e}$	Matriz de circuitos para Método de Davies
h	Passo - razão entre o deslocamento ao longo do eixo de Helicoide pela rotação em torno dele
J_H	Jacobiano baseado em helicoides
J	Matriz do Jacobiano
M	Matriz transformação dos vetores de dimenssão 4 para dimensão 3
O_i	Origem do i-ésimo referencial
PP	<i>Programming Pendant</i>
P	Junta Prismática
p^k	ponto p escrito no referencial k
\dot{q}	Taxa de variação temporal das posições das juntas
R_n^k	Matriz de rotação do referencial $O-x_k y_k z_k$ para o Referencial $O-x_k y_k z_k$
R	Matriz de rotação
$R_i(\alpha)$	Matriz de rotação em torno de do eixo i de um ângulo α
\$	Representação de Helicoide
$\hat{\$}_i$	Helicoide unitário para a i-ésima junta
s	Vetor que representa a direção do eixo (instantâneo) do helicoide
s_0	Vetor que representa a posição de um ponto qualquer do eixo (instantâneo) do helicoide em relação ao sistema de referência
T_B^A	Matriz de transformação homogênea do referencial B para A
s_{0iRef}	Vetor que representa a posição de um ponto qualquer do eixo do helicoide da junta i em relação ao sistema de referência na posição e referência
s_{iRef}	Vetor representativo do eixo do helicoide da junta i na posição de referência
v_0	Velocidade linear do ponto de referência no efetuador final

ω_n Velocidade angular do efetuador final

\dot{x} Vetor de velocidades generalizadas

1 Introdução

Ao longo dos anos a robótica vem sendo considerada cada vez mais estratégica no que diz respeito ao desenvolvimento econômico. Com o advento da tecnologia e redução de custos, os robôs têm ganhado espaço não só nos setores que envolvem engenharia, mas na medicina, no entretenimento e na vida cotidiana das pessoas. Por isso, a Pesquisa, Desenvolvimento e Inovação (PD&I) em robótica está em alta. Essa pesquisa trata das ferramentas matemáticas necessárias e como fazer para utilizá-las de modo a programar o robô de forma eficiente, rápida e simples, mais especificamente as resoluções das cinemáticas direta e inversa do robô Motoman MH5F utilizando a Teoria dos Helicoides.

Nesse capítulo é apresentado um breve histórico do desenvolvimento da robótica, mostrando as razões que justificam a sua importância comercial. Também são analisados indicadores associados aos países que mais se aprimoraram na tecnologia de automação, de modo a explicitar a motivação do trabalho que é apresentada para contribuir, mais ainda, na pesquisa desse ramo em nível nacional.

1.1 Motivação

A indústria vem se desenvolvendo ao longo do tempo com o objetivo de otimizar as linhas de produção, melhorando o produto, aumentando a produtividade e, como consequência, maximizando os lucros. A utilização de máquinas nos ambientes fabris foi iniciada com a criação inglesa da máquina a vapor, na Primeira Revolução Industrial, no século XVIII. Desde então, os processos de fabricação vem sendo aprimorados. Um grande marco para o processo de automatização da indústria foi o sistema de produção em série implementado por Henry Ford no início do século XX. A partir daí, conceitos como mecanização do trabalho, produção em massa e padronização do maquinário e do equipamento são aplicados na indústria.

A automação industrial vem sendo aprimorada ano a ano com o desenvolvimento de novas tecnologias. A partir da metade do século XX, o surgimento de computadores digitais e componentes miniaturizados permitiram a construção de robôs controlados computacionalmente e reprogramáveis, os robôs industriais, elementos essenciais nos sistemas de automação (SICILIANO; KHATIB, 2016). Os robôs industriais, desde então, vêm sendo aplicados em uma vasta gama de processos, como nas indústrias automobilística, aeroespacial, alimentícia, eletrônica, metal-mecânica, química, dentre outras. De acordo com o relatório da *International Federation of Robotics* (IFR, 2016), em 2015 a indústria

automotiva lidera a instalação anual de robôs no mundo, com um total de aproximadamente 97.500 robôs instalados no ano de 2015, seguida da indústria eletro-eletrônica, com em torno de 64.600 unidades, e da indústria metal-mecânica, com 29.450 robôs fornecidos.

Inicialmente voltado para a substituição do trabalho humano em atividades fisicamente fatigantes, trabalhosas e perigosas, o robô industrial vem sendo aperfeiçoado em função de adventos tecnológicos, passando a ser compactos, adaptáveis e capazes de trabalhar lado a lado com humanos (PEÑA-LÓPEZ et al., 2016). Em sua obra “*Automation, Production Systems, and Computer-Integrated Manufacturing*”, Groover (2007) cita quatro razões que justificam a importância comercial e tecnológica da robótica industrial:

- robôs podem substituir humanos em trabalhos extenuantes e em ambientes desconfortáveis;
- um robô pode atuar de forma cíclica com uma consistência e repetitividade em níveis inatingíveis para o homem;
- robôs são reprogramáveis, ou seja, ele pode ser modificado através da seleção de diferentes ferramentas (efetadores finais) e programado de acordo com a tarefa exigida; e,
- robôs são controlados por computadores que, por sua vez, podem ser conectados a outros sistemas computacionais para compor os *Computer-Integrated Manufacturing* (CIM), i.e., sistemas que aplicam tecnologias computacionais e de comunicações para controlar atividades que englobam desde o projeto, até o planejamento e o controle da manufatura.

Como benefícios do crescente grau de automação em uma tecnologia de processos, é comum citar a economia de custos de mão-de-obra direta e a redução da variabilidade da operação (SLACK; CHAMBERS; JOHNSTON, 2009), sendo que a automação é usualmente justificada pelo primeiro. Com isso, os investimentos em tecnologias de processos, ou seja, em máquinas, equipamentos e dispositivos que ajudam a agregar valor à produção e a atingir seus objetivos estratégicos, são fundamentais para o desenvolvimento e aprimoramento da robustez dos processos de fabricação, auferindo vantagem competitiva aos produtos e, por consequência, ao setor industrial.

Como exemplo, a China é um modelo de economia que apresenta alto grau de investimento no setor industrial, mais especificamente na instalação de robôs industriais. Segundo dados divulgados pela IFR (2016), em 2015 a China foi o país que mais investiu em instalações de robôs industriais em todo o mundo, adquirindo 68.556 do total de

253.748 novas unidades, ou seja, 27% do total, o que supera o total de vendas de toda a Europa (50.073 unidades). Concomitantemente é importante citar que, no ano de 2015, segundo dados divulgados pela *Trading Economics*, o país detinha um PIB correspondente à cerca de 15% da economia mundial, apresentando crescimento médio acima de 9% ao ano no último decênio. Assim sendo, é possível notar uma relação direta entre a robotização das indústrias com a participação dos setores no PIB pois o robô industrial é o principal sistema mecatrônico utilizado na automação dos processos de fabricação, o que motiva a PD&I na área. A Fig. 1 mostra a quantidade de robôs adquiridos em 2014 e 2015 por Brasil, República da Coreia, Japão, América do Norte, Europa e China.

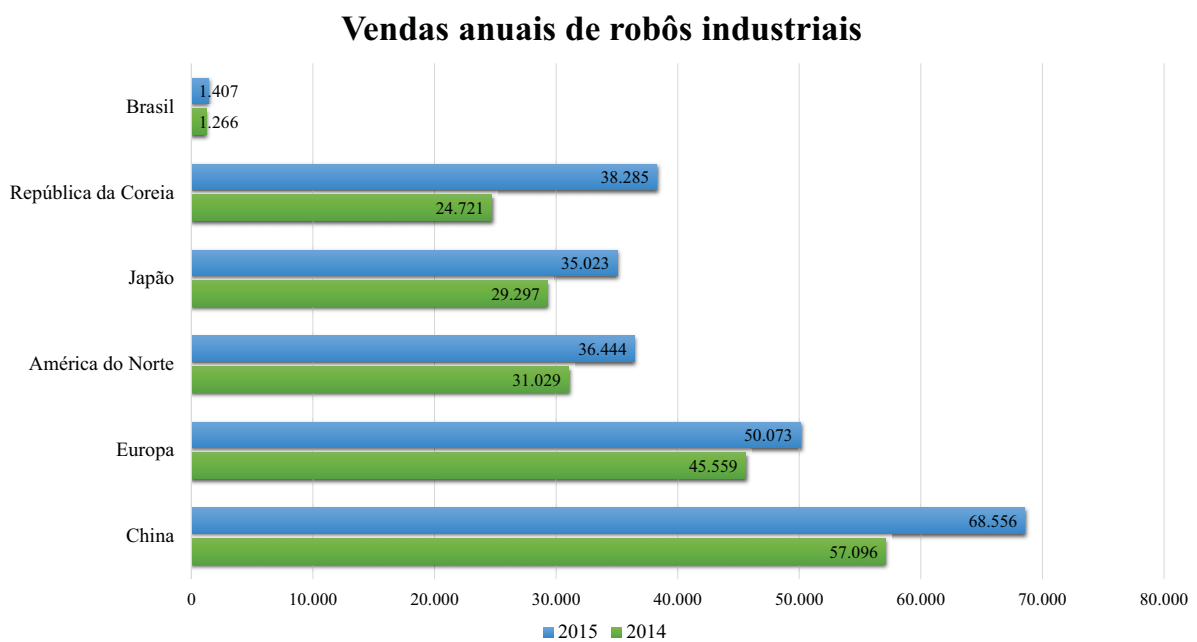


Figura 1 – Vendas anuais de robôs industriais (IFR - 2016)

Apesar da automação baseada na robótica industrial ser uma solução para produtividade e flexibilidade, a programação do robô ainda apresenta complexidade, demandando tempo, pessoal especializado e dinheiro. Pan et al. (2012) cita um caso em que a programação *on-line* de uma linha de soldagem a arco na fabricação do chassi de um veículo dura mais de oito meses, 360 vezes o tempo necessário à de execução do processo da soldagem. Uma solução é a utilização de softwares de programação *off-line*.

Em contrapartida, a utilização de *softwares* de programação *off-line* para produções de pequenos volumes demanda novos investimentos e pessoal especializado, o que pode tornar o investimento economicamente inviável nestes casos, além de ser necessária a criação de procedimentos de calibração entre o modelo real e o virtual. Sendo assim, o estudo da

resolução cinemática de robôs industriais torna parte importante para a elaboração de *softwares* de programação *off-line* e simuladores economicamente viáveis.

1.2 Formulação do problema

Em consonância com os argumentos apresentados na seção antecedente, que motivam a pesquisa na área em questão, o propósito deste trabalho é a implementação da **resolução cinemática do Robô Industrial Motoman MH5F utilizando a Teoria dos Helicoides, com a finalidade de aplicação em algoritmos de simulação e PD&I em programação *off-line*.**

1.3 Objetivos do trabalho

Os objetivos do trabalho são:

- conhecer as funcionalidades e a linguagem de programação do robô Motoman MH5F e seu controlador FS100;
- elaborar as resoluções das cinemáticas direta e inversa do robô Motoman MH5F; e,
- desenvolver um protótipo do software de programação *off-line* que ensine e grave um arquivo, no formato adequado, contendo os pontos destinados à execução de tarefas pelo robô.

1.4 Organização do trabalho

O capítulo 2 detalha sumariamente, desde o histórico e definição da robótica, até os diversos tipos de robôs e os tipos de programação possíveis, bem como alguns detalhes do manipulador em foco e de seu controlador. O capítulo 3 é destinado especificamente à modelagem da cinemática de manipuladores seriais com base na Teoria dos helicoides e a sua aplicação ao Motoman MH5F. O capítulo 4 apresenta os resultados obtidos através das simulações das trajetórias linear e circular, bem como o *software* de simulação do manipulador Motoman MH5F, além do ambiente 3D destinado à criação do ambiente virtual do laboratório.

Finalmente, no capítulo 5 são apresentadas as conclusões e as perspectivas para trabalhos futuros.

2 Robótica Industrial

Este Projeto de Fim de Curso tem seu escopo voltado à Robótica Industrial, no entanto para que haja entendimento do assunto, neste capítulo são abordados conceitos introdutórios atinentes à Robótica, à Robótica Industrial, à Programação de Robôs Industriais e ao robô industrial Yakasawa Motoman MH5F especificamente, pois é o equipamento utilizado no estudo de caso.

2.1 Robótica

Nos últimos anos tem sido comum na indústria cinematográfica o lançamento de filmes de ficção científica, contendo robôs nas mais variadas funções. Seja em tarefas domésticas, industriais ou até mesmo com concepções mais avançadas de poder decisório, os robôs têm ocupado um espaço importante nas telas de cinema. Tal destaque se dá por conta do poder que o cinema tem de incutir ideias na mente das pessoas através do entretenimento, levando-as a refletir sobre como seriam as suas vidas dividindo espaço ou tarefas com robôs.

2.1.1 Breve Histórico

O termo robô (*robot*) foi introduzido em 1921 pelo dramaturgo checo Karel Capek em sua peça satírica R.U.R (*Rossum's Universal Robots*) em alusão ao termo *robot*, que significa trabalho escravo nas línguas eslavas. A peça retratava os robôs como máquinas que se assemelhavam a pessoas, mas trabalhavam incansavelmente. Nela, os robôs se voltam contra seus criadores e tentam aniquilar a raça humana (MURRAY et al., 1994). De acordo com Siciliano et al. (2010), a imagem do robô como um artefato mecânico iniciou na década de 1940 quando o russo Isaac Asimov, conhecido escritor de ficção científica, concebeu o robô como um artefato de aparência humana desprovido de sentimentos. Seu comportamento era determinado por um “cérebro” eletrônico programado por um humano.

Os primeiros trabalhos que levaram aos robôs de hoje começaram após a Segunda Guerra Mundial no desenvolvimento de manipuladores mecânicos controlados remotamente, desenvolvidos nos laboratórios de *Argonne* e *Oak Ridge National* para o manuseio de material radioativo. Esses primeiros mecanismos eram do tipo líder-seguidor, constituído por um manipulador mestre guiado pelo usuário através de uma série de movimentos que eram então duplicados pela unidade seguidora (MURRAY et al., 1994).

2.1.2 O que é Robótica?

De acordo com Siciliano et al. (2010), o termo **robótica** foi introduzido por Asimov, como sendo a ciência voltada ao estudo de robôs que fossem baseados em *três leis fundamentais*:

1. Um robô não pode ferir um ser humano, ou por inação, permitir que um ser humano sofra algum mal;
2. Um robô deve obedecer às ordens dadas por seres humanos, exceto quando essas ordens entrarem em conflito com a primeira lei;
3. Um robô deve proteger sua própria existência, desde que tal proteção não entre em conflito com a 1^a ou a 2^a lei.

Tais leis estabeleceram regras de comportamento ao serem consideradas como especificações para o projeto de robô.

De acordo com Aurélio (2010), “robótica é o ramo do conhecimento, comum à engenharia e à informática, que trata da criação e da programação de robôs”.

Segundo Siciliano et al. (2010) a robótica é comumente definida como a ciência que estuda a conexão inteligente entre percepção e ação. Partindo desta definição, pode-se caracterizar um sistema robótico como sendo um sistema complexo, funcionalmente conectado por vários subsistemas, como observa-se na Fig. 2:

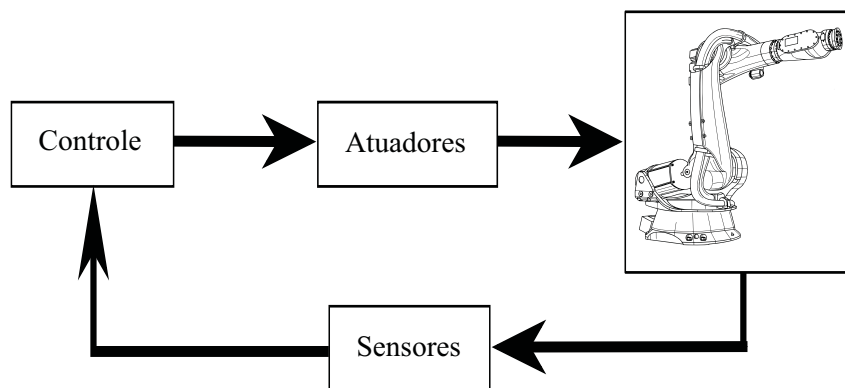


Figura 2 – Componentes de um sistema robótico (adaptado de Siciliano et al. (2010))

Segundo Siciliano et al. (2010), a robótica pode ser classificada de acordo com o ambiente de atuação do robô, conforme a seguir:

- *Robótica Industrial*: apresenta aplicações industriais, em locais que possuem *ambiente estruturado*, ou seja, com características geométricas e físicas conhecidas (baixa necessidade de autonomia).
- *Robótica Avançada*: normalmente se refere ao campo de atuação dos robôs dotados de maior autonomia, operando em ambientes pouco estruturados ou mesmo *não estruturados*, isto é, com características físicas e geométricas que podem não ser conhecidas.

Este trabalho, especificamente, tem seu escopo voltado à Robótica Industrial, tema abordado resumidamente na Sec. 2.2.

2.2 A Robótica Industrial

2.2.1 Estrutura mecânica de robôs industriais

Os robôs industriais, também chamados de manipuladores, possuem uma estrutura bem definida, constituindo-se por uma base e uma sequência de elos ou *links*, todos conectados por juntas. Além disso, eles possuem um punho que proporciona destreza aos movimentos, ou seja, capacidade de modificar a posição e a orientação de um objeto manipulado de uma configuração de referência para outra, arbitrariamente escolhida dentro das possíveis configurações do punho (MA; DOLLAR, 2011), e um efetuador final ou ferramenta, selecionado de acordo com a tarefa a ser realizada.

Os manipuladores são diferenciados de acordo com os tipos de juntas que conectam os elos entre si. Elas são classificadas como juntas prismáticas (P), que executam movimentos lineares, e juntas de revolução (R), que são aquelas que executam deslocamentos angulares. Dependendo das dimensões dos elos e da sequência de posicionamento das juntas entre os mesmos, é modificada a área de trabalho, i.e., a região em que é possível o acesso do robô. A Fig. 3 apresenta os tipos de juntas elementares.

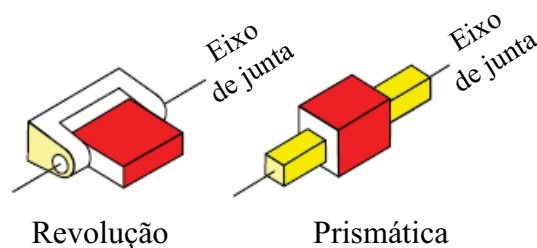


Figura 3 – Tipos de juntas elementares (adaptado de Jazar (2010))

A literatura classifica os robôs manipuladores, de acordo com a sequência e os tipos de junta e as possíveis áreas de trabalho, em *cartesiano*, *cilíndrico*, *esférico*, *Selective Compliance Articulated Robot Arm (SCARA)* e *antropomórfico*.

- **Robôs Cartesianos** possuem em suas estruturas somente juntas prismáticas (PPP), todas instaladas perpendicularmente entre si. Desta forma, este tipo de manipulador possui pouca destreza, executando apenas movimentos retilíneos no espaço (SICILIANO et al., 2010). Tais juntas prismáticas, tipicamente atuadores elétricos e ocasionalmente pneumáticos, garantem movimentos precisos em todo o espaço de trabalho. Como pode ser visto na Fig. 4, o espaço de trabalho é restrito a um paralelepípedo retangular. Devido à boa acurácia e a capacidade de suportar cargas pesadas, o robô cartesiano é empregado de forma significativa na manipulação e montagem que utilizam materiais pesados e com grandes dimensões.

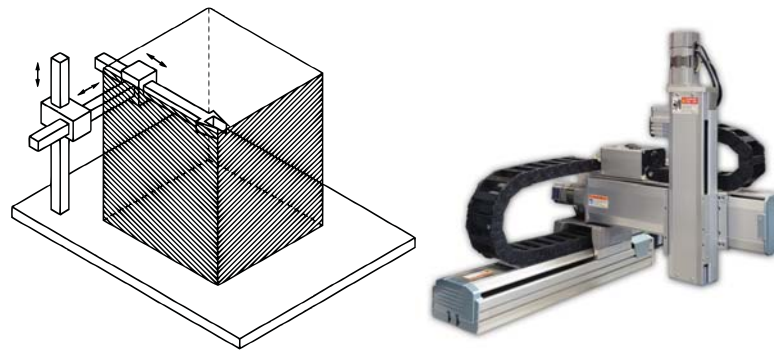


Figura 4 – Manipulador cartesiano e seu espaço de trabalho (SICILIANO et al., 2010)

- **Robôs Cilíndricos** são manipuladores cujo espaço de trabalho constitui parte de um cilindro oco. Diferente do robô cartesiano, neste tipo de robô a primeira junta prismática é substituída por uma junta de rotação com o eixo paralelo ao eixo da junta seguinte, uma junta prismática, que por sua vez é perpendicular à última junta prismática (RPP), como pode ser observado na Fig. 5. São utilizados em manipulação de objetos, inclusive com grandes dimensões (SICILIANO et al., 2010).

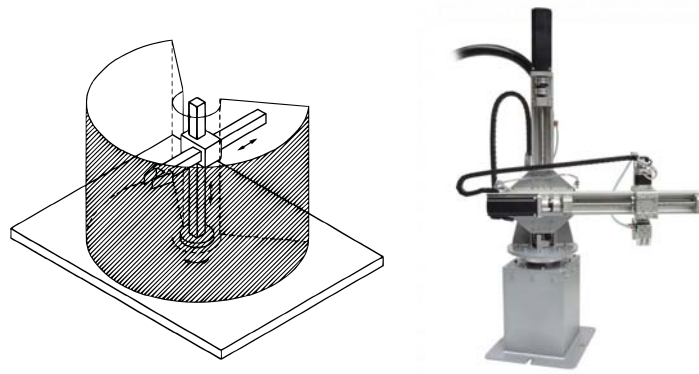


Figura 5 – Manipulador cilíndrico e seu espaço de trabalho (SICILIANO et al., 2010)

- **Robôs Esféricos** são constituídos, na sequência, de duas juntas de rotação perpendiculares e uma prismática (RRP), permitindo que o volume de trabalho seja parte de uma esfera oca, como está evidenciado na Fig. 6. Coordenadas esféricas podem ser utilizadas para descrever as posições de pontos da ferramenta. Os robôs esféricos têm menos rigidez que os dois tipos apresentados anteriormente e são utilizados, geralmente, motores elétricos (SICILIANO et al., 2010).

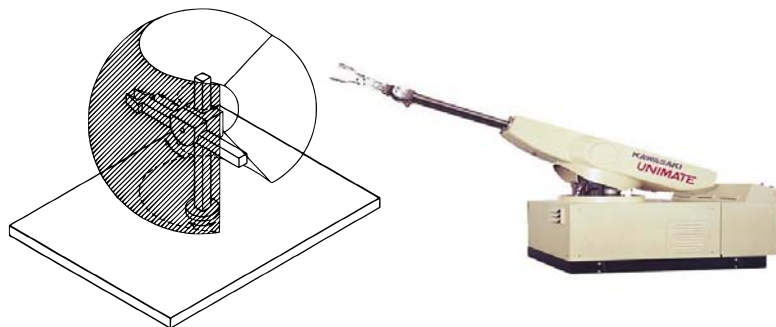


Figura 6 – Manipulador esférico e seu espaço de trabalho (SICILIANO et al., 2010)

- **Robôs SCARA** possuem duas juntas paralelas de rotação, seguidas por uma junta prismática que permite o deslocamento normal ao plano do efetuador final (RRP). Existem alguns manipuladores deste tipo com três juntas de rotação seguidas de uma junta prismática (RRRP) (CRAIG, 2005). Os robôs SCARA são muito empregados em tarefas de montagens verticais e, em geral, seus atuadores são motores elétricos (SICILIANO et al., 2010). A Fig. 7 destaca o volume de trabalho característico dos manipuladores SCARA (*Selective Compliance Assembly Robot Arm*).

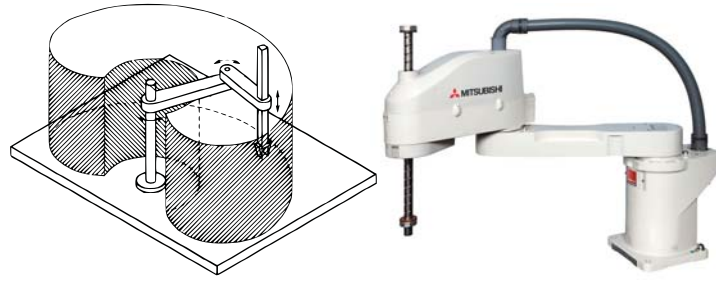


Figura 7 – Manipulador SCARA e seu espaço de trabalho (SICILIANO et al., 2010)

- **Robôs Antropomórficos**, também chamados de manipuladores articulados, são constituídos por três juntas de rotação (RRR), fazendo dele o melhor manipulador quanto à destreza (SICILIANO et al., 2010). O espaço de trabalho característico é, aproximadamente, uma esfera, como é mostrado na Fig. 8. Ele requer menores estruturas mecânicas que os outros tipos de robôs, o que reduz a interferência no volume de trabalho e os custos em aplicações com limitações de espaço de trabalho (CRAIG, 2005), fazendo dele o robô industrial mais utilizado no mundo.

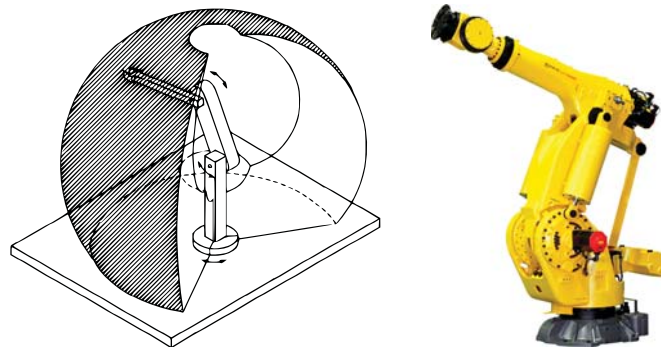


Figura 8 – Manipulador antropomórfico e seu espaço de trabalho (SICILIANO et al., 2010)

Aos diferentes tipos de estruturas apresentadas são conectados os punhos, que dão mais graus de liberdade aos manipuladores. A configuração das estruturas, ou seja, a combinação entre ângulos das juntas de rotação e deslocamentos das juntas prismáticas determinam a posição do ponto de conexão com o punho. A orientação do efetuador final é determinada pela configuração do punho que é formado por juntas de rotação. Os punhos possuem no máximo três graus de liberdade (DOF) sendo que os esféricos, i.e., aqueles cujos eixos de rotação das juntas se interceptam em um único ponto, proporcionam maior destreza aos manipuladores (SICILIANO et al., 2010). Como pode ser observado na Fig. 54, o robô Motoman MH5F é um manipulador antropomórfico e com punho esférico, já que os três eixos de rotação do punho se intersectam no ponto P.

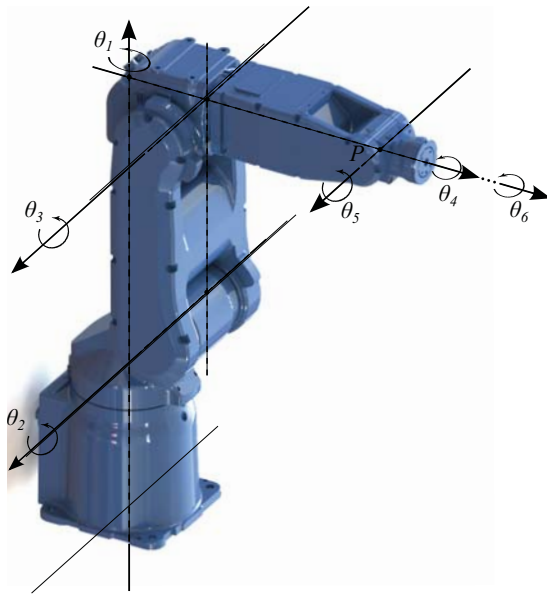


Figura 9 – Robô Yaskawa Motoman MH5F e suas juntas de rotação

O efetuador-final (EF) é adaptado aos robôs industriais de acordo com as tarefas específicas a serem realizadas. Eles podem ser ferramentas que executam certa atividade ou instrumentos para a manipulação de materiais. As garras são produzidas de acordo com as dimensões e as características dos materiais a serem manipulados das tarefas a serem executadas. Outros exemplos de efetuadores-finais são pistolas de pintura, brocas e equipamentos de soldagem.

2.3 Programação de Robôs Industriais

O robô industrial é, basicamente, um mecanismo automático projetado para mover peças e ferramentas por meio de uma sequência de movimentos ou operações desejadas. Como o robô prossegue de um ciclo de tarefas para outro, a sequência de operações do robô pode variar para permitir que o robô execute outras tarefas baseadas em mudanças nas condições externas. Além disso, o mesmo tipo de robô, ou o mesmo robô pode ser utilizado para executar tarefas completamente distintas, envolvendo conjuntos de movimentos ou operações diferentes se a espaço de trabalho é modificada ou se deseja variadas tarefas apenas. Nesse sentido, o programa de controle do robô deve ser capaz de adequar-se a diferentes funções, e deve também ser flexível dentro de uma determinada função para permitir a sequência de operações dinâmicas. A flexibilidade do robô é portanto, até certo ponto, determinada pelos tipos de movimentos e operações que podem ser programadas na unidade de controle e pela facilidade com que o programa pode ser inserido ou modificado (NOF, 1999).

As formas de programação de tarefas, ou seja, as programações *on-line*, são abordadas

a seguir. e *off-line*

2.3.1 Programação *On-line*

A programação *on-line* ou *Teach Programming* trata-se de uma maneira de inserir no controlador do robô, dados necessários ao cumprimento de uma tarefa. Nesta modalidade, o robô é manualmente guiado por meio de uma sequência de movimentos pretendidos pelo operador, que está observando o robô, seus movimentos, assim como outros equipamentos na área de trabalho. Neste processo, o caminho é ensinado ao robô e nele estão envolvidos o ensinamento, a edição e a repetição do caminho desejado. (NOF, 1999).

A abordagem adotada na programação *on-line* depende do algoritmo de controle usado para mover o robô ao longo do caminho pretendido. De acordo com Nof (1999) existem 3 (três) algoritmos mais básicos e utilizados no ensino de caminhos ao robô, são eles:

- *Controle ponto-a-ponto*: robôs com este tipo de controle movem-se de uma posição a outra sem que haja consideração do caminho tomado pelo manipulador. Geralmente cada eixo funciona no seu máximo ou em uma taxa limitada até alcançar determinada posição. Embora todos os eixos comecem seu movimento simultaneamente, eles não necessariamente completarão seus movimentos juntos.
- *Caminho contínuo*: envolve a repetição de pontos próximos que foram gravados a medida que o robô era guiado ao longo da trajetória desejada. A posição de cada eixo é gravada pela unidade de controle em uma base de tempo constante pelo escaneamento dos *encoders* dos eixos durante o movimento do robô.
- *Caminho controlado*: envolve o controle de coordenadas de todos os movimentos das juntas para atingir a trajetória pretendida entre dois pontos programados. Neste método, cada eixo move-se suave e proporcionalmente para resultar um movimento controlado e predito.

Na programação *on-line* existem duas abordagens básicas para guiar o robô por uma trajetória desejada, são elas, de acordo com Nof (1999):

- a) *Programação por Teach-Pendant*: utiliza uma unidade de controle portátil chamada *Teach-Pendant*, que possui uma quantidade de botões ou interruptores utilizados para direcionar o controlador no posicionamento do robô. Normalmente está associado a robôs com movimento ponto-a-ponto ou de caminho controlado.

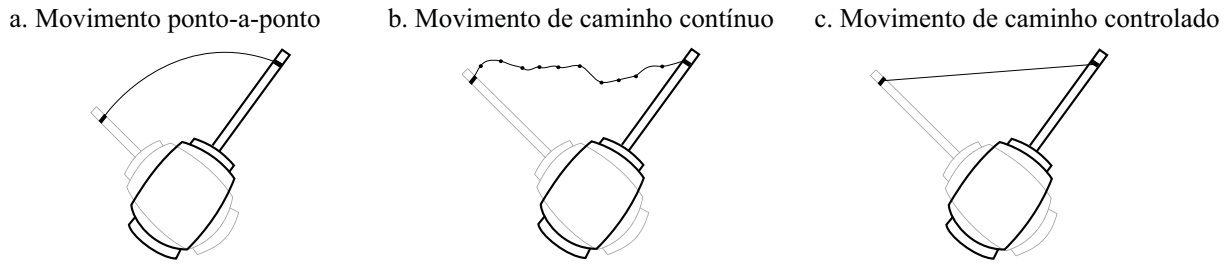


Figura 10 – Algoritmos de controle de movimento para determinação do caminho real adotado quando o programa é repetido no modo automático (NOF, 1999)

- b) *Programação lead-through*: o operador segura um braço do robô e guia o robô ao longo de uma tarefa ou movimento. É frequentemente usada em operações de pintura ou arco de solda. Comumente aplicada em robôs com controle de caminho contínuo, pode também ser utilizada em programas ponto-a-ponto.

2.3.2 Programação *Off-line*

A programação *off-line* pode ser considerada como um processo pelo qual a programação do robô é desenvolvida, parcial ou completamente, sem requerer o uso do próprio manipulador. Isso inclui geralmente dados de coordenadas de pontos, dados de funções e ciclo lógico. O desenvolvimento de tecnologias na área de robótica, tanto em *hardware* quanto em *software*, tem tornado as técnicas de programação *off-line* mais factíveis. Tais desenvolvimentos incluem maior sofisticação em controladores de robô, aumento de precisão posicional e o advento dos sensores (NOF, 1999).

Segundo Pan et al. (2012), dentre inúmeros benefícios, é possível destacar três vantagens da programação *off-line* com relação a *on-line*:

1. o processo de programação não exige a utilização do robô, minimizando o tempo fora de serviço;
2. os programas gerados *off-line* são mais flexíveis que o método de ensino ponto-a-ponto. Mudanças podem ser inseridas facilmente substituindo somente as partes necessárias do programa ou inserindo rotinas; e,
3. geralmente, simulações são incorporadas aos *softwares* de programação *off-line*, possibilitando a pré-visualização dos movimentos do robôs, o que reduz a possibilidade de erros.

Além destas, Nof (1999) enumera mais algumas vantagens, das quais são possíveis destacar:

1. um único programa pode ser usado para vários manipuladores sem a necessidade de se conhecer as particularidades do controlador de cada robô;
2. a remoção do programador de um ambiente potencialmente crítico, reduzindo o risco de acidentes por conta de um comportamento inesperado do robô; e,
3. integração com sistemas CAD/CAM, o que habilita tais sistemas a serem usados em conjunto com outras funcionalidades necessárias ao processo de fabricação tais como planejamento e controle.

Embora seja bastante vantajoso utilizar a programação *off-line*, ela apresenta algumas desvantagens. Segundo Boberg (2015), as seguintes desvantagens podem ser enumeradas:

1. a programação *off-line* requer uma certa qualificação. Com isso, cria-se a necessidade de treinamento de pessoal, que apresenta um custo nem sempre viável dependendo da quantidade a ser treinada;
2. como os dados armazenados são usados a partir de uma base de dados capaz de simular o manipulador desejado, sempre há uma diferença entre os dados e os reais componentes da área de trabalho do robô;
3. a comunicação no momento de transferir aplicações para a memória do controlador do robô é um possível problema, especialmente em sistemas multi-robô; e,
4. cada fabricante possui sua própria linguagem de programação, tornando desgastante para os programadores compreenderem variadas linguagens.

2.4 O robô Yaskawa Motoman MH5F

O robô Yaskawa Motoman MH5F é um manipulador industrial antropomórfico, com seis graus de liberdade, ou seja, a cadeia cinemática aberta é composta por seis juntas, todas de rotação, movimentadas através de servomotores elétricos. Sua carga útil, isto é, a carga máxima de manipulação, é de 5 *kg* e a sua massa total é de, aproximadamente, 27 *kg*. O conjunto é composto pela estrutura mecânica, o *Programming Pendant* (PP) e o controlador FS-100, como pode ser visto na Fig. 11. O controlador é o responsável por armazenar toda a programação e o processamento de informações que são transmitidas

aos motores, produzindo os movimentos desejados das juntas. O PP é a interface homem-máquina e tem uma série de funções que permitem que o operador ensine pontos, edite coordenadas, crie programas (JOB), altere as configurações do sistema robótico, etc.

A função do PP é realizar operações de ensino e edição de pontos e tarefas, ou seja, é por intermédio dele que o manipulador é operado.



Figura 11 – Componentes do sistema robótico Motoman MH5F

A garra ou ferramenta ou efetuador final, acoplável ao flange do robô, é personada de acordo com as necessidades dos processos de produção onde ele pode atuar. Uma informação importante é que as dimensões das ferramentas acopladas devem ser registradas no software de controle do robô para que façam parte da cinemática elaborada pelo controlador.

Além das atividades realizadas somente com o efetuador final, o sistema robótico pode ser composto de outros instrumentos e ferramentas a fim de auxiliar a execução de determinada tarefa ou até mesmo aumentar o controle da segurança do ambiente de trabalho do manipulador. Estes instrumentos facilitadores podem ser prensas pneumáticas, leitores óticos, esteiras e outros sensores que visam a otimização dos processos automatizados. Quanto ao aprimoramento da segurança de trabalho, podem ser instalados botões de parada de emergência adicionais, portais óticos para isolar o espaço de trabalho do robô, sensores de colisão e outros sensores que ajudam a resguardar a integridade do pessoal e do material. Para que sejam interfaceados com o controlador, eles são tratados como

inputs ou *outputs*, ou seja, enviam ou recebem informações do controlador.

A seguir são analisadas as informações necessárias ao entendimento e à operação do sistema robótico, abordando os detalhes sobre o controlador FS-100, o PP e os procedimentos de programação *on-line*.

2.4.1 O Controlador FS-100

O FS-100 é o controlador do sistema robótico alvo deste trabalho. Como características físicas, indicadas na Fig. 12, ele é instalado em um gabinete com $470 \times 200 \times 420 \text{ mm}$, possui três saídas e duas entradas de ar para arrefecimento, um interruptor automático que protege o circuito contra sobrecargas e curto-circuito, um painel indicador, além de conectores destinados aos cabos de conexão dos eixos e de *inputs/outputs* (I/O's). Com o sistema operacional Windows CE, ele é capaz de gerenciar um total de 28 I/O's, realizar a integração com sistemas de automação como o Controlador Lógico Programável (CLP) e controlar até dois manipuladores com até 8 eixos cada um.

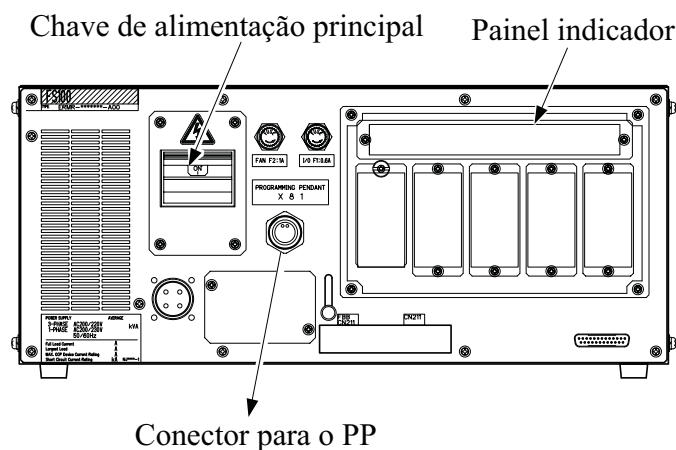


Figura 12 – Controlador FS-100 (adaptado de YASKAWA (2014))

O controlador possui alguns algoritmos e funções que otimizam a sua utilização como a função de áreas de interferência, o modo de economia de energia e o algoritmo *Advanced Robot Motion* (ARM).

- As **áreas de interferências**, parte do sistema a prova de colisão, são regiões definidas pelo usuário onde o robô não deve interferir com a finalidade de evitar colisões com outros manipuladores ou com dispositivos periféricos. Elas são classificadas em **Áreas de Interferência Cúbica**, que são regiões paralelepipedais, ou **Áreas de Interferência Axial**, regiões radiais. Quando o manipulador detecta que irá invadir estas áreas, ele desacelera até parar.

- O **modo de economia de energia** é uma função que permite a interrupção do fornecimento de energia ao robô, durante a execução de tarefas, quando o mesmo não é utilizado por um determinado período de tempo (mínimo de 3 minutos).
- O **algoritmo ARM** calcula dinamicamente os torques e os carregamentos nos eixos com o objetivo de aprimorar a precisão dos percursos do robô e o controle das vibrações, para assim otimizar os movimentos e a velocidade (CONNOLLY, 2009). Para que ele seja executado com máxima precisão, é necessário informar ao sistema dados como peso e centro de gravidade do efetuador final e a posição de instalação do robô, que pode variar de 0° a 180° .

Além de serem utilizados no sistema a prova de colisão e no algoritmo ARM, os dados referentes às ferramentas são importantes para a definição e calibração das posições de HOME (principal e secundária ou "*checkpoint*") e do sistema de coordenadas do efetuador final. Para facilitar a configuração de diferentes ferramentas, o controlador permite a criação de um banco de dados com até dezesseis ferramentas, numeradas de 0 a 15, com informações como as coordenadas do *Tool Center Point* (TCP), i.e., ponto de referência para o sistema de coordenadas do efetuador final instalado no flange, "pose" ou orientação, peso, posição do centro de gravidade e momento de inércia no centro de gravidade.

Com o PP é possível modificar parâmetros, editar as configurações do sistema, criar e editar trabalhos e movimentar o manipulador para a execução de programação *on-line*. O PP, os sistemas de coordenadas do manipulador, bem como o procedimento de programação *on-line* são descritos no Apêndice B.

A linguagem de programação empregada no controlador é a INFORM III que é composta basicamente de instrução e itens adicionais, que se subdividem em *tag* e valor numérico, como indicado na Fig. 13. No Apêndice C encontra-se disponível os comandos da linguagem INFORM III, úteis a programação de tarefas no sistema robótico em questão.

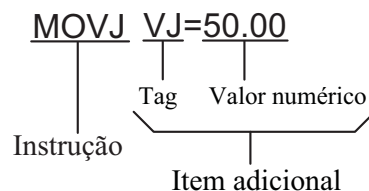


Figura 13 – Composição dos comandos da INFORM III (adaptado de YASKAWA (2014))

3 Cinemática de Manipuladores Seriais

Este capítulo apresenta os conceitos de cinemática de manipuladores seriais, introduzindo conceitos de localização de corpo rígido no espaço e Teoria dos Helicoides, culminando com a resolução cinemática do robô Yaskawa Motoman MH5F.

3.1 Introdução

A modelagem dos robôs industriais é a base para a resolução da cadeia cinemática formada pelos multicorpos unidos por juntas que compõem os robôs. A análise cinemática consiste na descrição matemática da relação que a variação das posições de cada junta da cadeia cinemática irá modificar a posição e a orientação do efetuador-final (SICILIANO et al., 2010). Utilizando transformações lineares, é possível determinar a posição do efetuador-final com relação a um referencial fixo através da posição de cada junta. Este problema é conhecido como *cinemática direta*. Além disso, existe ainda o problema da *cinemática inversa*, o qual as posições de cada junta que compõem a cadeia cinemática são determinadas em função da posição e orientação desejadas para o efetuador-final.

Além das cinemáticas direta e inversa, a *cinemática diferencial* descreve a relação analítica entre as velocidades das juntas e a velocidade generalizada do efetuador final. Essa relação de velocidades lineares e angulares é descrita pela *matriz do Jacobiano*, uma das mais importantes grandezas em análise e controle de movimentos de robôs (SPONG; HUTCHINSON; VIDYASAGAR, 2006).

A representação da posição e orientação de um manipulador não é única. O método mais popular na cinemática de robôs é baseado na notação de Denavit-Hatemberg para mecanismos espaciais e transformação homogênea de pontos. A matriz 4×4 , de transformação homogênea, é utilizada para representar as transformações espaciais de vetores. Na robótica, é usada para descrever um sistema de coordenadas em relação a outro. O método matricial é o mais popular para descrever o movimento de robôs. Alternativamente à metodologia de Denavit-Hatemberg, há a Teoria dos Helicoides, a *álgebra Lie* e *álgebra Epsilon*. A transformação de um corpo rígido ou de um sistema de coordenadas pode ser expressa por meio de um **deslocamento helicoidal** com uma translação ao longo de um eixo ou uma rotação em torno dele (JAZAR, 2010).

3.2 Localização de corpo rígido no espaço

Um corpo rígido é completamente descrito no espaço pela sua localização, ou seja, posição e orientação com relação a um referencial (SICILIANO et al., 2010). A Fig. 14 mostra os vetores unitários \mathbf{x} , \mathbf{y} , \mathbf{z} dos eixos do referencial ortonormal $O - xyz$.

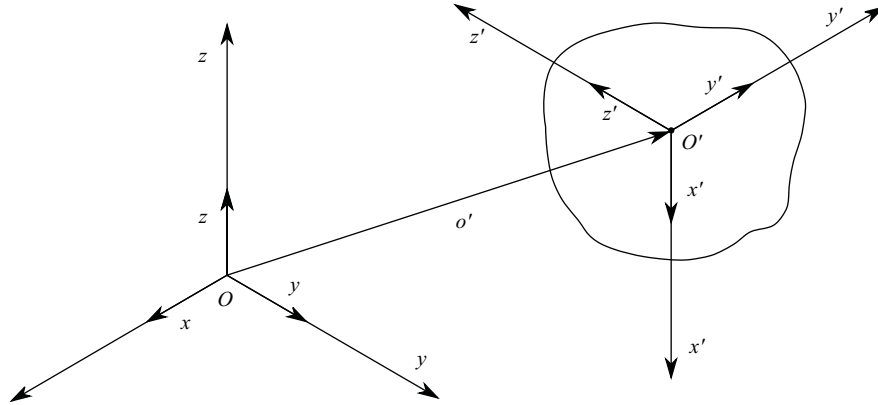


Figura 14 – Posição e orientação de um corpo rígido (SICILIANO et al., 2010)

A posição de O' com relação ao referencial $O - xyz$ é determinada pelo vetor \mathbf{o}' , como se segue:

$$\mathbf{o}' = o'_x \mathbf{x} + o'_y \mathbf{y} + o'_z \mathbf{z}$$

onde o'_x , o'_y , o'_z são as componentes do vetor $\mathbf{o}' \in \mathfrak{R}^3$.

Com o objetivo de descrever a orientação do corpo rígido, considere o referencial $O' - x'y'z'$ com origem em O' , sendo \mathbf{x}' , \mathbf{y}' , \mathbf{z}' os vetores unitários do sistema. Baseada nas componentes de cada vetor unitário, denominadas cossenos diretores dos eixos de $O' - x'y'z'$ relativos à $O - xyz$, a orientação do corpo é determinada através dos vetores expressos em relação ao referencial $O - xyz$:

$$\mathbf{x}' = x'_x \mathbf{x} + x'_y \mathbf{y} + x'_z \mathbf{z} \quad (3.1)$$

$$\mathbf{y}' = y'_x \mathbf{x} + y'_y \mathbf{y} + y'_z \mathbf{z} \quad (3.2)$$

$$\mathbf{z}' = z'_x \mathbf{x} + z'_y \mathbf{y} + z'_z \mathbf{z} \quad (3.3)$$

3.2.1 Matrizes de rotação

Com base nas expressões que determinam a orientação de um corpo rígido, as Eq. 3.1, 3.2 e 3.3 podem ser reescritas na forma matricial, passando a receber a denominação de **matriz de rotação**:

$$R = \begin{bmatrix} \mathbf{x}' & \mathbf{y}' & \mathbf{z}' \end{bmatrix} = \begin{bmatrix} x'_x & y'_x & z'_x \\ x'_y & y'_y & z'_y \\ x'_z & y'_z & z'_z \end{bmatrix} = \begin{bmatrix} \mathbf{x}'^T \mathbf{x} & \mathbf{y}'^T \mathbf{x} & \mathbf{z}'^T \mathbf{x} \\ \mathbf{x}'^T \mathbf{y} & \mathbf{y}'^T \mathbf{y} & \mathbf{z}'^T \mathbf{y} \\ \mathbf{x}'^T \mathbf{z} & \mathbf{y}'^T \mathbf{z} & \mathbf{z}'^T \mathbf{z} \end{bmatrix} \quad (3.4)$$

Os vetores coluna de R são mutuamente ortogonais desde que representem vetores unitários de um sistema ortonormal (SICILIANO et al., 2010). Isso faz com que a matriz seja ortogonal, valendo as seguintes condições:

$$\begin{aligned} \mathbf{x}'^T \mathbf{y}' &= 0, & \mathbf{y}'^T \mathbf{z}' &= 0, & \mathbf{z}'^T \mathbf{x}' &= 0 \\ \mathbf{x}'^T \mathbf{x}' &= 1, & \mathbf{y}'^T \mathbf{y}' &= 1, & \mathbf{z}'^T \mathbf{z}' &= 1 \\ \det(R) &= 1 \\ R^{-1} &= R^T \end{aligned}$$

Com o objetivo de determinar os elementos da matriz de rotação R , considere o sistema $O - xyz$ e o sistema $O' - x'y'z'$ rotacionado de um ângulo α em torno de z , como mostrado na Fig. 15. A rotação é considerada positiva haja vista que ocorre no sentido anti-horário.

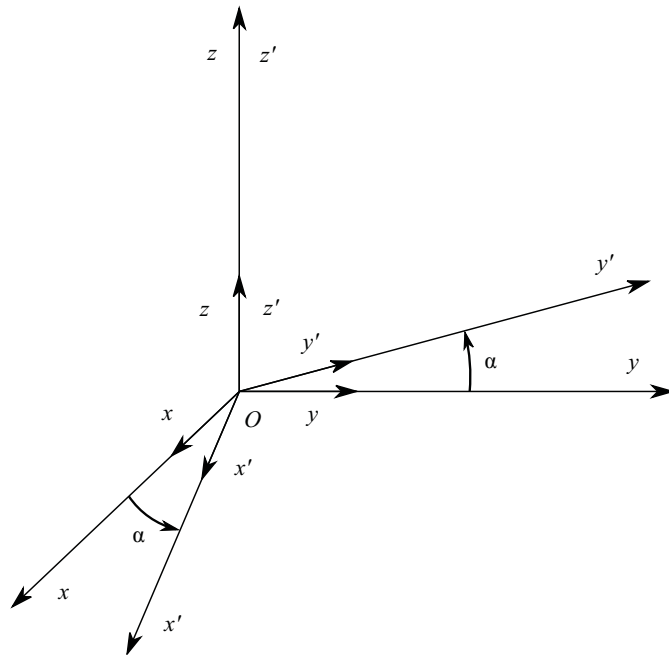


Figura 15 – Rotação do sistema $O - xyz$ de um ângulo α em torno de z (SICILIANO et al., 2010)

As novas componentes dos vetores unitários \mathbf{x}' , \mathbf{y}' , \mathbf{z}' com relação ao referencial $O - xyz$ são:

$$\mathbf{x}' = \begin{bmatrix} \cos\alpha \\ \text{sen}\alpha \\ 0 \end{bmatrix} \quad \mathbf{y}' = \begin{bmatrix} -\text{sen}\alpha \\ \cos\alpha \\ 0 \end{bmatrix} \quad \mathbf{z}' = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

Desta forma, a matriz de rotação de $O' - x'y'z'$ com relação à $O - xyz$ resulta em:

$$R_z(\alpha) = \begin{bmatrix} \cos\alpha & -\text{sen}\alpha & 0 \\ \text{sen}\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.5)$$

Analogamente, as matrizes de rotação de um ângulo β , em torno de y , e γ , em torno de x , representadas, respectivamente, pelar Eq. (3.6) e (3.7):

$$R_y(\beta) = \begin{bmatrix} \cos\beta & 0 & \text{sen}\beta \\ 0 & 1 & 0 \\ -\text{sen}\beta & 0 & \cos\beta \end{bmatrix} \quad (3.6)$$

$$R_x(\gamma) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\gamma & -\text{sen}\gamma \\ 0 & \text{sen}\gamma & \cos\gamma \end{bmatrix} \quad (3.7)$$

As matrizes de rotação representam a transformação linear das coordenadas de um vetor em $O' - x'y'z'$ para as coordenadas do mesmo vetor em $O - xyz$. Modifica-se a denominação dos referenciais $O - xyz$ em $O - x_0y_0z_0$ e $O' - x'y'z'$ em $O - x_1y_1z_1$. A matriz de rotação R passa a ser designada por R_1^0 , valendo a seguinte expressão:

$$\left(R_1^0\right)^T = \left(R_1^0\right)^{-1} = R_0^1$$

onde R_0^1 é a rotação do sistema $O - x_1y_1z_1$ para $O - x_0y_0z_0$.

Considere o referencial absoluto $O - x_0y_0z_0$ seguido de n sucessivas rotações, originando os referenciais $O - x_iy_iz_i$, para $i = 1, 2, \dots, n$, como é mostrado na Fig. 16.

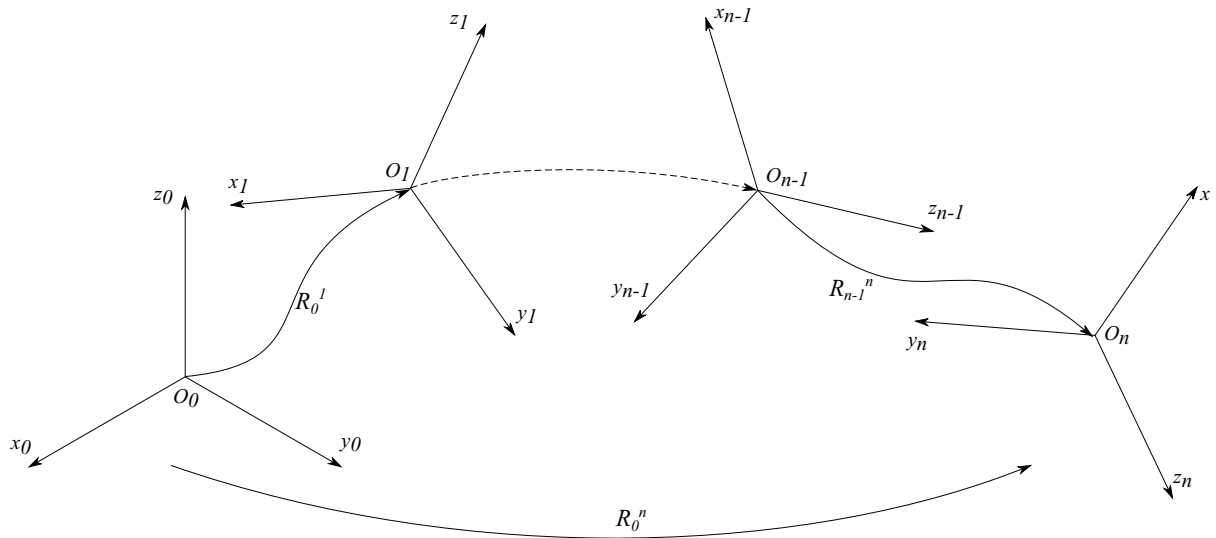


Figura 16 – Representação de n sucessivas rotações

A obtenção da matriz de transformação resultante, que rotaciona do referencial $O - x_0y_0z_0$ para $O - x_ny_nz_n$, pode ser obtida pela composição das rotações sucessivas:

$$R_0^n = R_0^1 R_1^2 \cdots R_{n-2}^{n-1} R_{n-1}^n \quad (3.8)$$

3.2.2 Ângulos de Euler

As matrizes de rotação são caracterizadas pela descrição da orientação de forma redundante (SICILIANO et al., 2010), já que são necessários somente três parâmetros, no caso espacial, para a caracterização completa da orientação e a matriz é composta por nove elementos.

A representação mínima pode ser caracterizada pelo conjunto de três ângulos, denominados **ângulos de Euler** ($\phi = [\varphi \ \vartheta \ \psi]^T$). Em outras palavras, uma única matriz de rotação $R(\phi)$ é obtida pela composição das três rotações sucessivas. No caso do conjunto de rotações sucessivas em ZYX , a matriz resultante é obtida da seguinte forma:

1. Rotacionar o sistema de referência de um ângulo ψ em torno do eixo x (*yaw*);
2. Rotacionar o sistema de referência de um ângulo ϑ em torno do eixo y (*pitch*); e,
3. Rotacionar o sistema de referência de um ângulo φ em torno do eixo z (*roll*);

Desta forma, matriz resultante de *Roll-Pitch-Yaw* será:

$$R(\phi) = R_z(\varphi) R_y(\vartheta) R_x(\psi) = \begin{bmatrix} c\varphi c\vartheta & c\varphi s\vartheta s\psi - s\varphi c\psi & c\varphi s\vartheta c\psi - s\varphi s\psi \\ s\varphi c\vartheta & s\varphi s\vartheta s\psi - c\varphi c\psi & s\varphi s\vartheta c\psi - c\varphi s\psi \\ -s\vartheta & c\vartheta s\psi & c\vartheta c\psi \end{bmatrix} \quad (3.9)$$

A solução inversa, ou seja, os ângulos φ, ϑ, ψ , pode ser obtida através dos elementos da matriz resultante. Para ϑ no intervalo $(-\pi/2, \pi/2)$:

$$\begin{aligned} \varphi &= \text{Atan2}(r_{21}, r_{11}) \\ \vartheta &= \text{Atan2}(-r_{31}, \sqrt{r_{32}^2 + r_{33}^2}) \\ \psi &= \text{Atan2}(r_{32}, r_{33}) \end{aligned} \quad (3.10)$$

Para ϑ no intervalo $(\pi/2, 3\pi/2)$:

$$\begin{aligned} \varphi &= \text{Atan2}(-r_{21}, -r_{11}) \\ \vartheta &= \text{Atan2}\left(-r_{31}, -\sqrt{r_{32}^2 + r_{33}^2}\right) \\ \psi &= \text{Atan2}(-r_{32}, -r_{33}) \end{aligned} \quad (3.11)$$

A função $\text{Atan2}(x, y)$ é denominada arco tangente com dois argumentos e representa a expressão $\text{atg}\left(\frac{x}{y}\right)$.

3.2.3 Transformação homogênea

Para representar completamente um corpo rígido no espaço é necessário descrever a sua posição e a orientação. A Fig. 17 mostra que o ponto P, pertencente a um corpo rígido, pode ser representado no referencial absoluto A como $p^A = \overline{OP}$ e no referencial B como $p^B = \overline{QP}$.

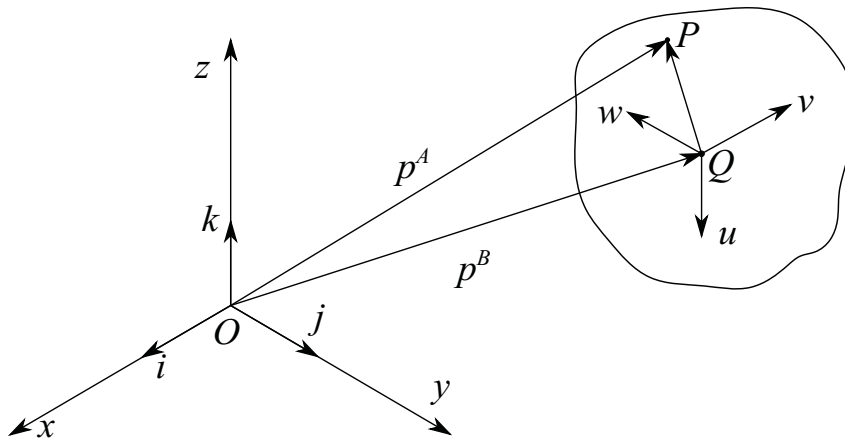


Figura 17 – Representação da localização do ponto p

Sabe-se que $\overline{OQ} = q^A$ e que a matriz R_B^A é a matriz de rotação do sistema B para A . Com isso, é possível obter a expressão que relaciona p^A e p^B :

$$\overline{OP} = \overline{OQ} + \overline{QP} \quad (3.12)$$

$$p^A = R_B^A p^B + q^A \quad (3.13)$$

A Eq. 3.13 representa o Teorema de Chasles. Com o objetivo de obter uma representação compacta, considere a representação homogênea do vetor p^B :

$$p^B = \begin{bmatrix} p_x^B \\ p_y^B \\ p_z^B \\ 1 \end{bmatrix} \quad (3.14)$$

Adotando similar representação para o vetor p^A , a Eq. 3.13 pode ser reescrita da seguinte forma:

$$\begin{bmatrix} p_x^A \\ p_y^A \\ p_z^A \\ 1 \end{bmatrix} = T_B^A \begin{bmatrix} p_x^B \\ p_y^B \\ p_z^B \\ 1 \end{bmatrix} \quad (3.15)$$

onde $[T_B^A]_{(4 \times 4)}$ é a matriz de transformação homogênea dada por:

$$T_B^A = \begin{bmatrix} [R_B^A]_{(3 \times 3)} & [q^A]_{(3 \times 1)} \\ [0]_{(1 \times 3)} & 1 \end{bmatrix} \quad (3.16)$$

Diferentemente do caso da matriz de rotação, a matriz de transformação homogênea não possui as propriedades da ortogonalidade, ou seja, $(T_B^A)^{-1} \neq (T_B^A)^T$. Analogamente ao caso das sucessivas rotações, é possível aplicar sucessivas transformação para o caso de n referenciais:

$$T_0^n = T_0^1 T_1^2 \cdots T_{n-2}^{n-1} T_{n-1}^n \quad (3.17)$$

3.3 Teoria dos Helicoides aplicada à robótica

A teoria dos helicoides foi criada por Robert Stawell Ball, astrônomo irlandês, em 1876, em seu livro “*The theory of screws: a study in the dynamics of a rigid body*”, e tem sido aperfeiçoada ao longo dos anos com o objetivo de modelar mecanismos complexos, de múltiplos corpos e juntas.

Segundo Tsai (1999), um helicóide é um ente geométrico definido por :

1. uma reta no espaço (vetor unitário S), que representa a direção do eixo do helicóide;
2. um vetor ligando a origem do sistemas de coordenadas cartesianas do referencial absoluto a um ponto pertencente a reta que define o eixo do helicóide (vetor S_0);
3. um ângulo θ caracterizando o movimento rotacional em torno do eixo do helicóide; e,
4. um comprimento d representando a translação sobre o eixo do helicóide.

Um helicóide é representado pelo símbolo $\$$ e pode ser escrito como:

$$\$ = \dot{q} \hat{\$} \quad (3.18)$$

onde: $\dot{q} = \dot{\theta}$, que representa a amplitude do movimento rotacional, no caso de juntas de rotação, e $\dot{q} = \dot{d}$, que representa a amplitude do deslocamento linear, para o caso de juntas prismáticas e $\hat{\$}$ representa o helicóide unitário na seguinte forma:

$$\hat{\$} = \begin{bmatrix} s \\ s_0 \times s + hs \end{bmatrix} = \begin{bmatrix} s_x \\ s_y \\ s_z \\ (s_{0y}s_z - s_{0z}s_y) + hs_x \\ (s_{0z}s_x - s_{0x}s_z) + hs_y \\ (s_{0x}s_y - s_{0y}s_x) + hs_z \end{bmatrix} \quad (3.19)$$

onde s é o eixo da direção da rotação de um ponto, s_0 é o vetor posição instantâneo de s em relação ao referencial absoluto e o passo (h), a razão entre o deslocamento ao longo do eixo de Helicóide pela rotação em torno do mesmo ($h = d/\theta$, para deslocamento finito, ou $h = \dot{d}/\dot{\theta}$, para deslocamento infinitesimal).

No caso de juntas prismáticas, a amplitude de rotação em torno do eixo de Helicóide é nula ($\theta = 0$), o que implica um passo infinito. Portanto, o Helicóide unitário resume-se em:

$$\hat{\$} = \begin{bmatrix} 0 \\ s \end{bmatrix} \quad (3.20)$$

Para as juntas de rotação, caso em que o deslocamento ao longo do eixo de Helicóide é nulo ($d = 0$), resultando em um passo nulo, o Helicóide unitário se resume a:

$$\hat{\$} = \begin{bmatrix} s \\ s_0 \times s \end{bmatrix} \quad (3.21)$$

3.3.1 Rotação em torno de um eixo de Helicoide

A Fig. 18 mostra a rotação do ponto P pertencente a um corpo B .

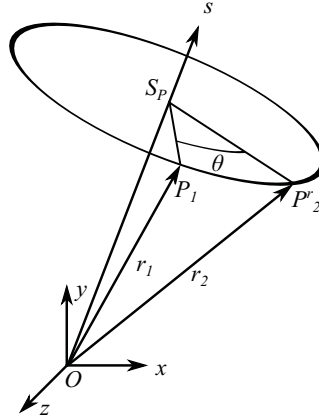


Figura 18 – Representação da rotação do ponto P (TSAI, 1999)

A rotação de um ângulo θ em torno de um eixo do helicoide s com origem no referencial fixo A modifica a posição inicial P_1 para P_2^r . O ponto S_P pertence ao plano que contém os pontos P_1 e P_2^r portanto, é possível obter as seguintes expressões vetoriais:

$$S_P P_1 = P_1 - (P_1 \cdot s) s \quad (3.22)$$

$$S_P P_2^r = P_2^r - (P_2^r \cdot s) s \quad (3.23)$$

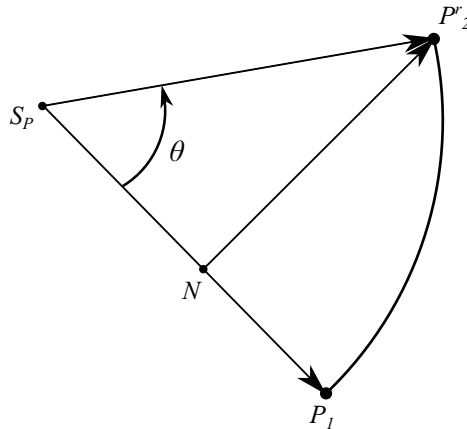


Figura 19 – Projeção de $S_P P_2^r$ sobre $S_P P_1$ P (TSAI, 1999)

Considere o plano formado pelos pontos S_P , P_1 e P_2^r , mostrado na Fig. 19. Seja N a projeção de $S_P P_2^r$ sobre $S_P P_1$. Como $|S_P P_2^r| = |S_P P_1|$ e $s \times S_P P_1 = s \times S_P P_2^r$, tem-se que:

$$S_P N = S_P P_1 \cos \theta \quad (3.24)$$

$$NP_2^r = (s \times S_P P_1) \operatorname{sen} \theta \quad (3.25)$$

O vetor P_2^r pode ser expresso pela seguinte soma vetorial:

$$S_P P_2^r = S_P N + NP_2^r \quad (3.26)$$

Substituindo as Eq. 3.24 e 3.25 na Eq. 3.26, a seguinte expressão pode ser obtida:

$$P_2^r - (P_2^r \cdot S) s = S_P P_1 \cos \theta + (s \times S_P P_1) \operatorname{sen} \theta \quad (3.27)$$

Utilizando a Eq. 3.22 e como $P_1 \cdot s = P_2^r \cdot s$, tem-se que:

$$P_2^r = P_1 \cos \theta + (1 - \cos \theta) (P_1 \cdot s) s + (s \times P_1) \operatorname{sen} \theta \quad (3.28)$$

A Eq. 3.28, também conhecida como fórmula de Rodrigues, pode ser expressa sob a forma do teorema de Euler. Fazendo $P_1 = p_B$ e $P_2^r = p_A$, a expressão escrita sob a forma matricial é apresentada da seguinte forma:

$$p_A = R_A^B p_B \quad (3.29)$$

onde a matriz de rotação R_A^B é dada por:

$$R_B^A = \begin{bmatrix} c\theta + s_x^2(1 - c\theta) & s_y s_z(1 - c\theta) - s_z s\theta & s_z s_x(1 - c\theta) + s_y s\theta \\ s_x s_y(1 - c\theta) + s_z s\theta & c\theta + s_y^2(1 - c\theta) & s_z s_y(1 - c\theta) - s_x s\theta \\ s_x s_z(1 - c\theta) - s_y s\theta & s_y s_z(1 - c\theta) + s_x s\theta & c\theta + s_z^2(1 - c\theta) \end{bmatrix} \quad (3.30)$$

3.3.2 Deslocamento Helicoidal

O teorema de Chasles diz que o deslocamento de um corpo rígido no espaço é obtido através de uma traslação ao longo de uma reta ou eixo de Helicoide, seguido de uma rotação em torno da mesma. O movimento descrito é conhecido como **deslocamento helicoidal**. A Fig. 20 mostra um movimento helicoidal completo do ponto P , ou seja, a rotação de um ângulo θ e a traslação t ao longo do eixo do helicoide.

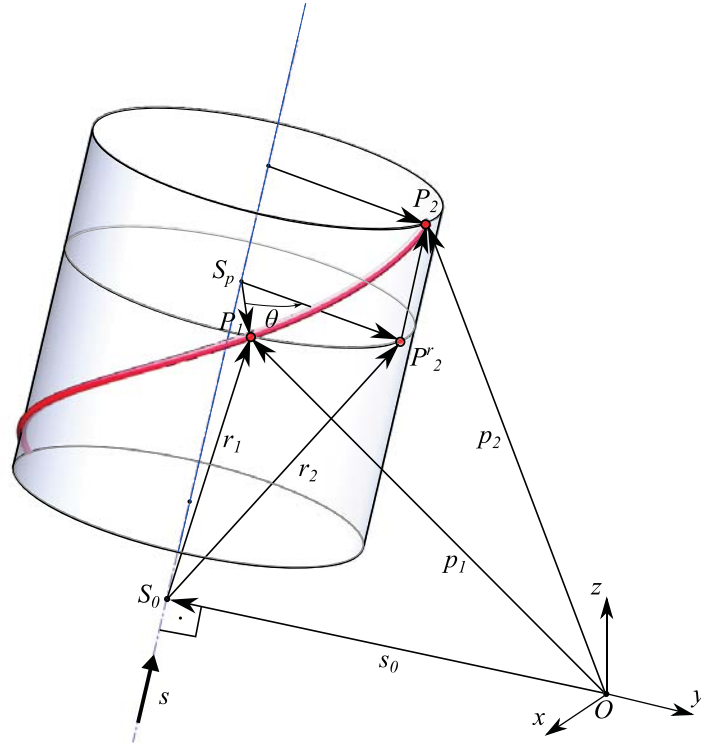


Figura 20 – Representação de movimento de rotação e translação em torno de um único eixo

Do diagrama podem ser obtidas as seguintes somas vetoriais:

$$P_1 = s_0 + s_0 P_1 \quad (3.31)$$

$$P_2 = s_0 P_2^r + s_0 + t s \quad (3.32)$$

Fazendo $s_0 P_1 = r_1$ e $s_0 P_2^r = r_2$, é possível substituir as Eq. 3.31 e 3.32 na fórmula de Rodrigues, obtendo-se as seguintes expressões:

$$s_0 P_2^r = s_0 P_1 \cos \theta + (1 - \cos \theta) (s_0 P_1 \cdot s) s + (s \times s_0 P_1) \operatorname{sen} \theta \quad (3.33)$$

$$P_2 = s_0 + t s + (P_1 - s_0) \cos \theta + (1 - \cos \theta) \left((P_1 - s_0)^T \cdot s \right) s + (s \times (P_1 - s_0)) \operatorname{sen} \theta \quad (3.34)$$

Colocando na forma do teorema de Chasles e fazendo $P_1 = p^B$ e $P_2 = p^A$, tem-se que:

$$p^A = R_B^A p^B + q^A \quad (3.35)$$

$$\begin{bmatrix} p_x^A \\ p_y^A \\ p_z^A \end{bmatrix} = R_B^A \begin{bmatrix} p_x^B \\ p_y^B \\ p_z^B \end{bmatrix} + \begin{bmatrix} q_x^A \\ q_y^A \\ q_z^A \end{bmatrix} \quad (3.36)$$

onde os elementos do vetor q^A são dados por:

$$\begin{aligned} q_x^A &= ts_x - s_{0x}(a_{11} - 1) - s_{0y}a_{12} - s_{0z}a_{13} \\ q_y^A &= ts_y - s_{0y}(a_{22} - 1) - s_{0x}a_{21} - s_{0z}a_{23} \\ q_z^A &= ts_z - s_{0z}(a_{33} - 1) - s_{0x}a_{31} - s_{0y}a_{32} \end{aligned} \quad (3.37)$$

sendo a_{ij} os termos da matriz R_B^A .

As Eq. 3.35 e 3.37 podem ser reescritas da seguinte forma:

$$q^A = ts + s_0 - R_B^A s_0 = ts + [I - R_B^A] s_0 \quad (3.38)$$

$$p^A = R_B^A p^B + ts + [I - R_B^A] s_0 \quad (3.39)$$

Por fim, a matriz de transformação homogênea é reescrita como se segue:

$$T_B^A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & q_x^A \\ a_{21} & a_{22} & a_{23} & q_y^A \\ a_{31} & a_{32} & a_{33} & q_z^A \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} R_B^A & ts + [I - R_B^A] s_0 \\ 0 & 0 \end{bmatrix} \quad (3.40)$$

3.3.3 Método dos deslocamentos helicoidais sucessivos

Quando um mecanismo tem n elos sucessivos, como mostrado na Fig. 21, é necessário levar em conta o efeito de cada corpo e junta no referencial resultante.

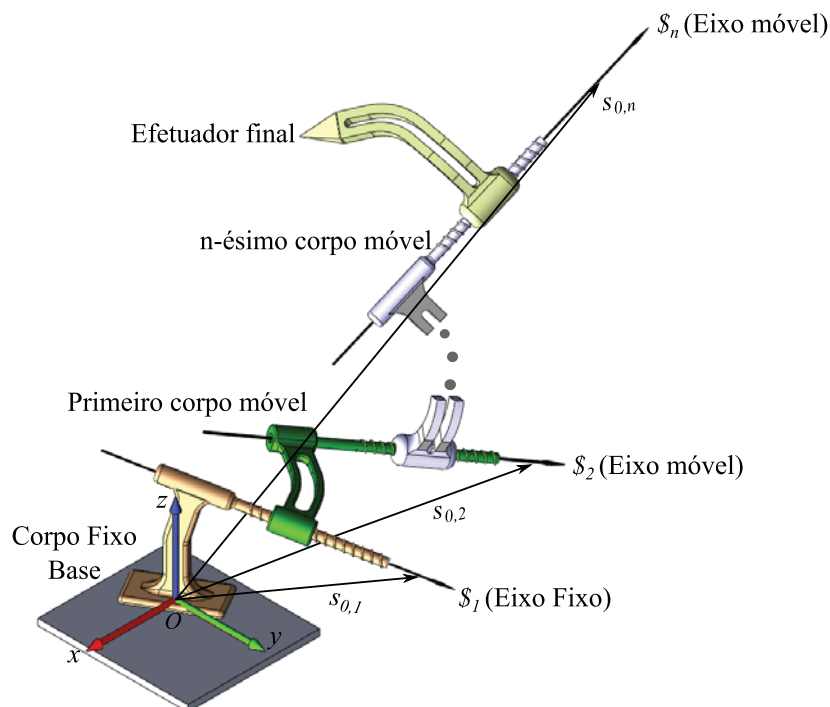


Figura 21 – Sucessão de n juntas

De acordo com o teorema de Chasles, considerando um helicóide $\$i$ qualquer pertencente à cadeia cinemática, ele realiza o deslocamento de (d_i, θ_i) sobre o eixo (S_i, S_{0i}) . Em outras palavras, o Helicóide $\$i (d_i, \theta_i, S_i, S_{0i})$ tem a sua matriz de transformação homogênea representada por:

$$A_i = \begin{bmatrix} R_{i(3 \times 3)}^{i-1} & d_i S_{\$i(1 \times 3)} + (I_{3 \times 3} - R_{i(3 \times 3)}^{i-1})_{(3 \times 3)} S_{0_{si}(1 \times 3)} \\ 0 & 1 \end{bmatrix} \quad (3.41)$$

O efeito combinado de cada junta sobre cada corpo é representado matematicamente pela multiplicação das matrizes de transformação homogênea relativa a cada helicóide $\$i$. O método dos deslocamentos dos helicóides sucessivos é utilizado na determinação da cinemática direta de manipuladores seriais com mais de um elo. Segundo Tsai (1999), costuma-se numerar os helicóides em ordem crescente, partindo da base e chegando ao referencial da ferramenta. Sendo assim, a matriz de transformação homogênea resultante é obtida pelo produtório das matrizes associadas aos deslocamentos helicoidais sucessivos respeitando a ordem em que ocorrem:

$$A_r = A_1 A_2 \dots A_{n-2} A_{n-1} A_n \quad (3.42)$$

A posição desejada, ou posição alvo (TSAI, 1999), caracterizada pelo vetor \mathbf{p} , é obtida mediante multiplicação da matriz de transformação homogênea resultante A_r pelo vetor posição inicial \mathbf{p}_0 , que é usualmente a posição de *home* do manipulador, a referência onde todas as variáveis de junta são atribuídas valor zero. Todo o procedimento para a obtenção da relação descrita serve para resolver o problema da **Cinemática Direta** e o algoritmo para chegar a tal relação, baseado na Teoria dos Helicóides aplicado a um manipulador serial de 6 DOF, encontra-se representado na Fig. 22.

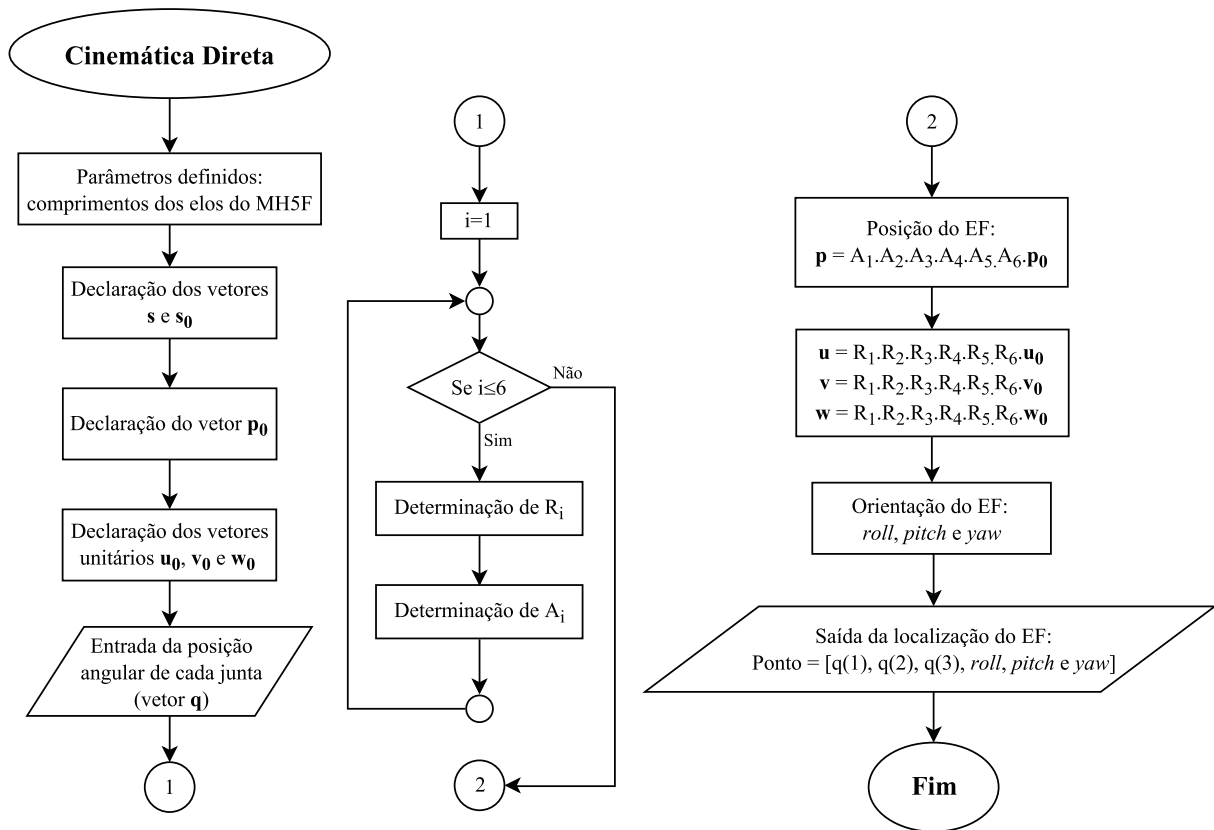


Figura 22 – Algoritmo para a solução da Cinemática Direta baseada em Helicoides

3.4 Cinemática direta do robô Motoman MH5F

Através da cinemática direta é possível determinar a posição de um ponto desejado com base nas coordenadas dos corpos rígidos que estão entre o referencial de base e o ponto em questão. Para isso é necessário saber a localização e a orientação de todos os corpos rígidos em relação ao referencial fixo, ou seja, as direções dos eixos de Helicoide s_i , além da localização s_{0i} dos mesmos.

De modo prático, é interessante montar uma tabela com os vetores s e s_0 de cada junta, em função do referencial base. A partir daí é possível calcular a matriz de transformação homogênea associada a cada junta (Eq. 3.40). No caso do robô Yaskawa Motoman MH5F, a Fig. 23 serve de auxílio para o preenchimento da Tab. 1. Os comprimentos de cada elo são: $l_1 = 88 \text{ mm}$, $l_2 = 330 \text{ mm}$, $l_3 = 310 \text{ mm}$, $l_4 = 40 \text{ mm}$, $l_5 = 305 \text{ mm}$, $l_6 = 80 \text{ mm}$ e $l_{ef} = 112 \text{ mm}$.

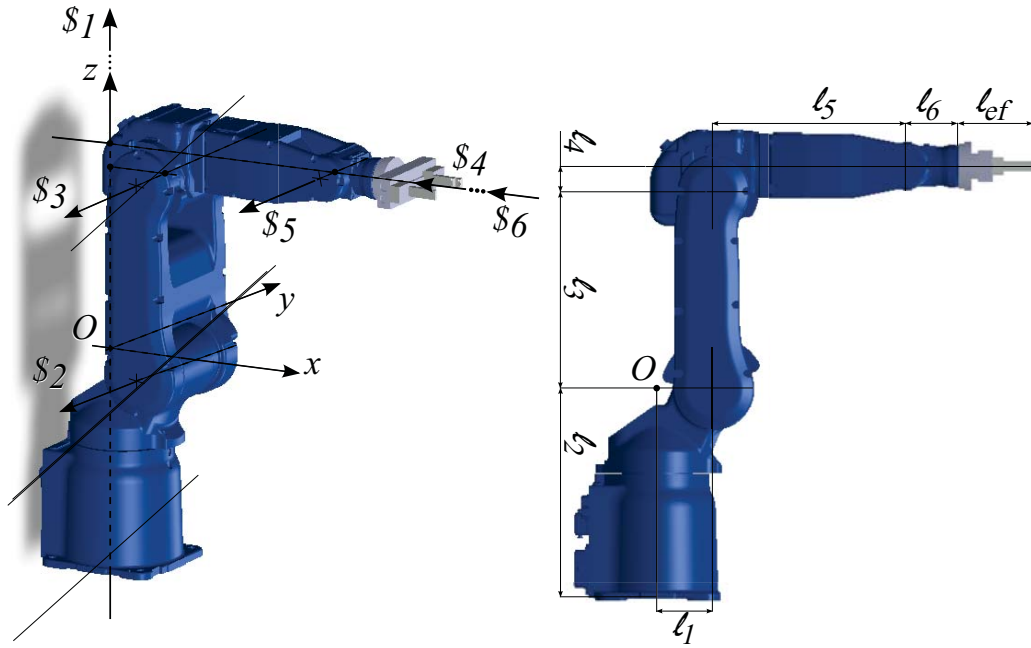


Figura 23 – Representação dos helicoides no robô Yaskawa Motoman MH5F

Tabela 1 – Vetores s e s_0 de cada junta para o Robô Yaskawa Motoman MH5F

JUNTA	s_i^T	s_{0i}^T
1	$[0, 0, 1]$	$[0, 0, 0]$
2	$[0, -1, 0]$	$[\ell_1, 0, 0] = [88, 0, 0]$
3	$[0, -1, 0]$	$[\ell_1, 0, \ell_3] = [88, 0, 310]$
4	$[-1, 0, 0]$	$[\ell_1, 0, \ell_3 + \ell_4] = [88, 0, 350]$
5	$[0, -1, 0]$	$[\ell_1 + \ell_5, 0, \ell_3 + \ell_4] = [393, 0, 350]$
6	$[-1, 0, 0]$	$[\ell_1 + \ell_5, 0, \ell_3 + \ell_4] = [393, 0, 350]$

Substituindo os valores indicados na Tab. 1, na Eq. 3.40, as matrizes de transformação homogênea associadas à variação θ_i de cada junta encontram-se representadas nas Eq. 3.43 e 3.48.

$$A_1 = \begin{bmatrix} \cos\theta_1 & -\text{sen}\theta_1 & 0 & 0 \\ \text{sen}\theta_1 & \cos\theta_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.43)$$

$$A_2 = \begin{bmatrix} \cos\theta_2 & 0 & \text{sen}\theta_2 & -\ell_2 \text{sen}\theta_2 - \ell_1(\cos\theta_2 - 1) \\ 0 & 1 & 0 & 0 \\ -\text{sen}\theta_2 & 0 & \cos\theta_2 & \ell_1 \text{sen}\theta_2 - \ell_2(\cos\theta_2 - 1) \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.44)$$

$$A_3 = \begin{bmatrix} \cos\theta_3 & 0 & \text{sen}\theta_3 & -(\ell_2 + \ell_3) \text{sen}\theta_3 - \ell_1(\cos\theta_3 - 1) \\ 0 & 1 & 0 & 0 \\ -\text{sen}\theta_3 & 0 & \cos\theta_3 & \ell_1 \text{sen}\theta_3 - (\ell_2 + \ell_3)(\cos\theta_3 - 1) \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.45)$$

$$A_4 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta_4 & -\text{sen}\theta_4 & (\ell_2 + \ell_3 + \ell_4) \text{sen}\theta_4 \\ 0 & \text{sen}\theta_4 & \cos\theta_4 & -(\ell_2 + \ell_3 + \ell_4)(\cos\theta_4 - 1) \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.46)$$

$$A_5 = \begin{bmatrix} \cos\theta_5 & 0 & \text{sen}\theta_5 & -(\ell_2 + \ell_3 + \ell_4) \text{sen}\theta_5 - (\ell_1 + \ell_5)(\cos\theta_5 - 1) \\ 0 & 1 & 0 & 0 \\ -\text{sen}\theta_5 & 0 & \cos\theta_5 & (\ell_1 + \ell_5) \text{sen}\theta_5 - (\ell_2 + \ell_3 + \ell_4)(\cos\theta_5 - 1) \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.47)$$

$$A_6 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta_6 & -\text{sen}\theta_6 & (\ell_2 + \ell_3 + \ell_4) \text{sen}\theta_6 \\ 0 & \text{sen}\theta_6 & \cos\theta_6 & -(\ell_2 + \ell_3 + \ell_4)(\cos\theta_6 - 1) \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.48)$$

A matriz de transformação homogênea A_r do manipulador é obtida ao aplicar a Eq. 3.42, ou seja, $A_r = A_1 A_2 A_3 A_4 A_5 A_6$. Para determinar a posição da ferramenta, ou seja, o vetor \mathbf{p} , basta multiplicar a matriz A_r pelo vetor posição inicial \mathbf{p}_0 :

$$\mathbf{p}_0 = \begin{bmatrix} \ell_1 + \ell_5 + \ell_6 + \ell_{ef} \\ 0 \\ \ell_2 + \ell_3 + \ell_4 \end{bmatrix}$$

$$\begin{bmatrix} p_x \\ p_y \\ p_z \\ 1 \end{bmatrix} = [A_r] \begin{bmatrix} p_{0x} \\ p_{0y} \\ p_{0z} \\ 1 \end{bmatrix} \quad (3.49)$$

3.4.1 Orientação do Efetuador Final

A orientação de referência do efetuador final do robô Motoman MH5F é indicada na Fig. 24.

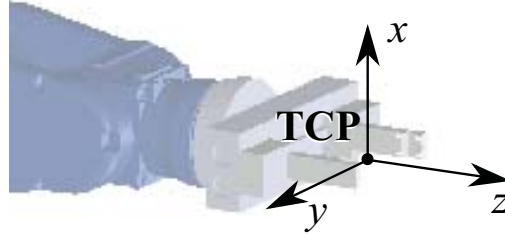


Figura 24 – Orientação de referência do efetuador final

Na forma vetorial, a orientação de referência pode ser representada da seguinte maneira:

$$\begin{aligned}\mathbf{u}_0 &= [0, 0, 1]^T \\ \mathbf{v}_0 &= [0, -1, 0]^T \\ \mathbf{w}_0 &= [1, 0, 0]^T\end{aligned}\quad (3.50)$$

Para obter a orientação final, ou orientação alvo (TSAI, 1999), são aplicadas as rotações sucessivas, conforme as expressões abaixo:

$$\begin{aligned}\mathbf{u} &= R_1^0 R_2^1 R_3^2 R_4^3 R_5^4 R_6^5 \mathbf{u}_0 \\ \mathbf{v} &= R_1^0 R_2^1 R_3^2 R_4^3 R_5^4 R_6^5 \mathbf{v}_0 \\ \mathbf{w} &= R_1^0 R_2^1 R_3^2 R_4^3 R_5^4 R_6^5 \mathbf{w}_0\end{aligned}\quad (3.51)$$

A partir daí, forma-se a matriz resultante $R_6^0 = [\mathbf{u} \ \mathbf{v} \ \mathbf{w}]$, semelhante à Eq. 3.9, bastando aplicar as Eq. 3.10 ou 3.11 para obter os ângulos de *roll*, *pitch* e *yaw*. Com isso, conclui-se o algoritmo da cinemática direta, indicado por $k(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6)$, resultando no vetor compostos por posição e orientação do efetuador-final, como se segue:

$$k(q) = \begin{bmatrix} \psi \\ \vartheta \\ \varphi \\ p_x \\ p_y \\ p_z \end{bmatrix}\quad (3.52)$$

onde:

$$q = \left[\theta_1 \quad \theta_2 \quad \theta_3 \quad \theta_4 \quad \theta_5 \quad \theta_6 \right]^T \quad (3.53)$$

3.5 Cinemática inversa

A cinemática inversa consiste na obtenção das variáveis de junta, ou seja, os ângulos aplicados à cada uma delas, por meio da posição e orientação do efetuador final. Em outras palavras, é o problema inverso da cinemática direta. Segundo Siciliano et al. (2010), o problema de cinemática inversa é mais complexo pois:

- as equações a resolver são na maioria das vezes não-lineares e nem sempre é possível encontrar uma solução fechada;
- múltiplas ou infinitas soluções podem existir; e,
- dependendo da estrutura cinemática, pode não haver soluções admissíveis.

Para o caso de um robô com 6 graus de liberdade, Siciliano et al. (2010) aponta, em regra, mais de 16 soluções admissíveis em casos onde não existem restrições para as juntas.

A solução da cinemática inversa pode ser fechada ou numérica. Por ser mais rápida, a primeira delas é a mais desejável, exigindo um esforço computacional menor. A desvantagem é que ela depende da estrutura mecânica do robô, isto é, ela é vinculada a um grande número de parâmetros relacionados à características geométricas e dos mecanismos envolvidos (SICILIANO; KHATIB, 2016).

A técnica de solução numérica, um dos objetivos deste trabalho, não é dependente do robô, sendo possível aplicá-la à diversas estruturas cinemáticas (SICILIANO; KHATIB, 2016). Um dos métodos utilizados na cinemática inversa é a solução numérica através da equação da cinemática diferencial, empregando a matriz do Jacobiano.

3.5.1 Cinemática diferencial

A cinemática diferencial relaciona as velocidades das juntas com a velocidade do efetuador, relação esta representada por meio da matriz do Jacobiano (J). A relação entre velocidade linear, $\dot{\mathbf{x}}$, e as taxas de variação dos ângulos das juntas de rotação ou dos deslocamentos das juntas prismáticas, é dada por:

$$\dot{\mathbf{x}} = J\dot{\mathbf{q}} \quad (3.54)$$

onde $\dot{\mathbf{x}}$ é o vetor de velocidades generalizadas, definido por:

$$\dot{\mathbf{x}} = \begin{bmatrix} \omega_n \\ v_0 \end{bmatrix} \quad (3.55)$$

Autores como Siciliano et al. (2010) utilizam a metodologia de Denavit-Hartenberg para a obtenção do Jacobiano convencional. No caso deste trabalho, será utilizado o Jacobiano baseado em Helicoides. A vantagem da metodologia adotada é o fácil entendimento do significado físico dos termos que envolvem a cinemática diferencial. No Jacobiano, o eixo do helicóide instantâneo representa a velocidade angular, enquanto o momento $s_0 \times s$ representa a velocidade linear. A velocidade angular do efetuador, ω_n , corresponde à soma das velocidades angulares de cada junta e a velocidade linear do ponto de referência no efetuador final, v_0 , é instantaneamente coincidente com o referencial absoluto (TSAI, 1999).

3.5.2 Jacobiano baseado em Helicoides

Denomina-se Jacobiano baseado em helicoides, J_H a matriz representada por:

$$J_H = \begin{bmatrix} \hat{\$}_1 & \hat{\$}_2 & \dots & \hat{\$}_{n-1} & \hat{\$}_n \end{bmatrix} \quad (3.56)$$

Sabe-se ainda que:

$$\dot{\mathbf{x}} = \begin{bmatrix} w_x \\ w_y \\ w_z \\ v_x \\ v_y \\ v_z \end{bmatrix} = \begin{bmatrix} \hat{\$}_1 & \hat{\$}_2 & \dots & \hat{\$}_{n-1} & \hat{\$}_n \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \vdots \\ \dot{q}_{n-1} \\ \dot{q}_n \end{bmatrix} \quad (3.57)$$

onde $\dot{\mathbf{x}}$ é no vetor que define completamente a cinemática de velocidade de um corpo rígido (TSAI, 1999).

3.5.3 Jacobiano do robô Motoman MH5F

Para obter o Jacobiano do robô Motoman MH5F, é conveniente utilizar as matrizes de transformação homogênea obtidas na Sec. 3.4. As coordenadas homogêneas são dadas por:

$$s_{iRef} = \begin{bmatrix} s_i \\ 0 \end{bmatrix}, \quad s_{0iRef} = \begin{bmatrix} s_{0i} \\ 1 \end{bmatrix} \quad (3.58)$$

para $i = 1, 2, \dots, 6$.

Primeiramente, é necessário transformar todos vetores s_i e s_{0i} para o referencial inercial. Utilizando os vetores constantes da Tab. 1, os novos vetores de direção e posição dos eixos de Helicoide serão:

$$\begin{aligned}
s_1 &= M s_{1Ref} & , & \quad s_{01} = M s_{01Ref} \\
s_2 &= M A_1 s_{2Ref} & , & \quad s_{02} = M A_1 s_{02Ref} \\
s_3 &= M A_1 A_2 s_{3Ref} & , & \quad s_{03} = M A_1 A_2 s_{03Ref} \\
s_4 &= M A_1 A_2 A_3 s_{4Ref} & , & \quad s_{04} = M A_1 A_2 A_3 s_{04Ref} \\
s_5 &= M A_1 A_2 A_3 A_4 s_{5Ref} & , & \quad s_{05} = M A_1 A_2 A_3 A_4 s_{05Ref} \\
s_6 &= M A_1 A_2 A_3 A_4 A_5 s_{6Ref} & , & \quad s_{06} = M A_1 A_2 A_3 A_4 A_5 s_{06Ref}
\end{aligned} \tag{3.59}$$

A matriz M serve apenas para transformar os vetores que anteriormente se encontravam em coordenadas homogêneas para vetores de três elementos. Ela é dada por:

$$M = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{3.60}$$

Uma informação importante é que os valores de s_i e s_{0i} são modificados de acordo com as variáveis de junta, já que são obtidos através dos deslocamentos sucessivos. Os vetores s_{iRef} e s_{0iRef} não dependem das variáveis de junta, apenas da posição inicial do manipulador.

De acordo com Rocha, Tonetto e Dias (2011), o uso da Teoria dos Helicoides permite a aplicação do **Método de Davies**, utilizando Cadeias Virtuais de Assur (CVA) para a obtenção do Jacobiano, o que não é possível com a notação de Denavit–Hartenberg. O método de Davies consiste em aplicar o conceito das Cadeias Virtuais de Assur para fechar uma cadeia cinemática, inicialmente aberta, e utilizar tal representação por meio de grafos (DAVIES, apud Ribeiro (2010)). Estes facilitam a visualização dos procedimentos e simplificam a modelagem da cinemática diferencial inversa para a solução da Eq. 3.54 para $\dot{\mathbf{q}}$. A sua aplicação é uma adaptação da Lei da Circulação de Kirchhoff para a diferença de potencial e, para tal estabelece a equação de restrição para a cadeia cinemática fechada, constituindo relações entre as velocidades das juntas componentes. Assim, é possível escrever as magnitudes das velocidades de um determinado número de juntas, conhecidas como juntas secundárias, em função das magnitudes das demais juntas, chamadas juntas primárias. A Fig. 25 representa Cadeia Virtual de Assur (PPPS - Prismática, Prismática, Prismática, Esférica) acoplada ao robô MOTOMAN MH5F.

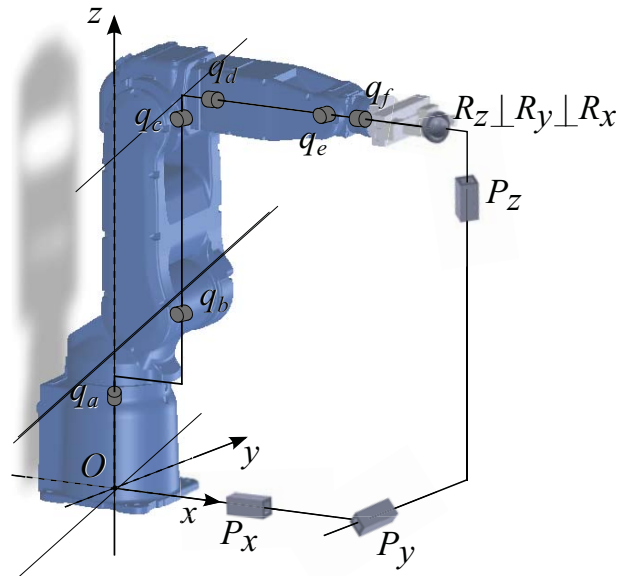


Figura 25 – Representação da Cadeia Virtual de Assur para o Motoman MH5F

A Fig. 26 contém o grafo das juntas associado. Em tal representação, os vértices são numerados conforme os corpos que representam, bem como as arestas são identificadas com as letras associadas às respectivas juntas.

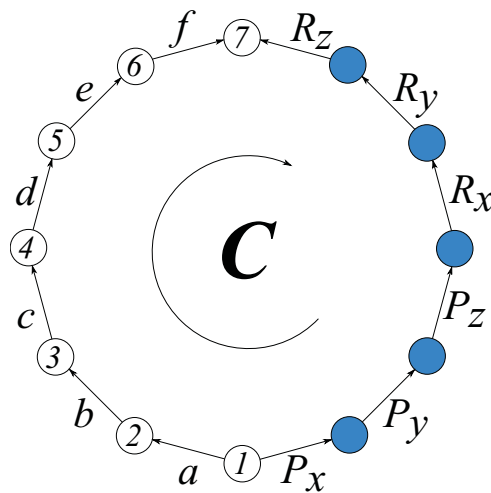


Figura 26 – Grafo das juntas do manipulador Motoman MH5F

Analogamente à Lei da Circulação de Kirchhoff para um circuito elétrico, a Lei da Circulação de Kirchhoff-Davies para uma cadeia cinemática fechada estabelece que a soma das velocidades dos pares cinemáticos ao longo da cadeia fechada é zero.

De modo a sistematizar o método, principalmente quando existe mais de um circuito, é conveniente construir a matriz dos circuitos $B_{l,e}$, partindo do grafo que representa os circuitos existentes na cadeia cinemática. Nesta matriz, l é o número de circuitos e e é o

número de arestas. Cada elemento $b_{i,j}$ de $B_{l,e}$ poderá assumir os seguintes valores:

$$b_{i,j} = \begin{cases} 0 & \text{se o circuito } i \text{ não incide sobre aresta } j; \\ +1 & \text{se o sentido do circuito } i \text{ e da aresta } j \text{ forem os mesmos; e} \\ -1 & \text{se o sentido do circuito } i \text{ e da aresta } j \text{ forem opostos.} \end{cases}$$

Para o exemplo em foco $l = 1$, pois existe somente um circuito e $e = 12$, pois há doze juntas. A matriz dos circuitos assume a seguinte forma:

$$B_{1,12} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & -1 & -1 \end{bmatrix} \quad (3.61)$$

Com a matriz B determinada e assumindo as hipóteses da Lei Circulação de Kirchhoff-Davies, tem-se:

$$\$_a + \$_b + \$_c + \$_d + \$_e + \$_f - \$_{R_z} - \$_{R_y} - \$_{R_x} - \$_{P_z} - \$_{P_y} - \$_{P_x} = 0 \quad (3.62)$$

Representando em termos de helicoides unitários e adotando \dot{q}_i a magnitude da velocidade angular do i -ésimo par cinemático rotativo tem-se:

$$\begin{aligned} & \hat{\$_a}\dot{q}_a + \hat{\$_b}\dot{q}_b + \hat{\$_c}\dot{q}_c + \hat{\$_d}\dot{q}_d + \hat{\$_e}\dot{q}_e + \hat{\$_f}\dot{q}_f - \\ & - \hat{\$_{R_z}}\dot{q}_{R_z} - \hat{\$_{R_y}}\dot{q}_{R_y} - \hat{\$_{R_x}}\dot{q}_{R_x} - \hat{\$_{P_z}}\dot{q}_{P_z} - \hat{\$_{P_y}}\dot{q}_{P_y} - \hat{\$_{P_x}}\dot{q}_{P_x} = 0 \end{aligned} \quad (3.63)$$

Reescrevendo a Eq. 3.63 na forma matricial chega-se à equação de restrição:

$$\begin{bmatrix} N_s & -N_p \end{bmatrix} \begin{bmatrix} \dot{q}_s \\ \dot{q}_p \end{bmatrix} = 0 \quad (3.64)$$

onde:

$$N_s = \begin{bmatrix} \hat{\$_a} & \hat{\$_b} & \hat{\$_c} & \hat{\$_d} & \hat{\$_e} & \hat{\$_f} \end{bmatrix} \quad (3.65)$$

$$N_p = \begin{bmatrix} \hat{\$_{R_z}} & \hat{\$_{R_y}} & \hat{\$_{R_x}} & \hat{\$_{P_z}} & \hat{\$_{P_y}} & \hat{\$_{P_x}} \end{bmatrix} = \begin{bmatrix} I_{3 \times 3} & 0_{3 \times 3} \\ \Omega_{3 \times 3} & I_{3 \times 3} \end{bmatrix} \quad (3.66)$$

$$\dot{q}_s = \begin{bmatrix} \dot{q}_a^T & \dot{q}_b^T & \dot{q}_c^T & \dot{q}_d^T & \dot{q}_e^T & \dot{q}_f^T \end{bmatrix}^T \quad (3.67)$$

$$\dot{q}_p = \begin{bmatrix} \dot{q}_{R_z}^T & \dot{q}_{R_y}^T & \dot{q}_{R_x}^T & \dot{q}_{P_z}^T & \dot{q}_{P_y}^T & \dot{q}_{P_x}^T \end{bmatrix}^T \quad (3.68)$$

e

$$\Omega_{3 \times 3} = \begin{bmatrix} 0 & -p_z & p_y \\ p_z & 0 & -p_x \\ -p_y & p_x & 0 \end{bmatrix} \quad (3.69)$$

Desta maneira, o Método da Davies permite a obtenção das velocidades das juntas secundárias tendo como entrada as velocidades das juntas primárias, conforme se segue:

$$N_s \dot{q}_s = N_p \dot{q}_p \quad (3.70)$$

Alternativamente, o valor das velocidades primárias pode ser escrito como:

$$\dot{q}_p = N_p^{-1} N_s \dot{q}_s \quad (3.71)$$

Percebe-se portanto que o Jacobiano assume a seguinte composição:

$$J = N_p^{-1} N_s \quad (3.72)$$

Dado que o sistema não é redundante, o Jacobiano pode ser invertido, então:

$$\dot{q}_s = [N_p^{-1} N_s]^{-1} \dot{q}_p \quad (3.73)$$

$$J(q) = N_p^{-1} N_s \quad (3.74)$$

$$J^{-1}(q) = [N_p^{-1} N_s]^{-1} \quad (3.75)$$

Ilustrativamente, o diagrama da Fig. 27 mostra o algoritmo para o cálculo do Jacobiano baseado em Helicoides de manipuladores com 6 graus de liberdade.

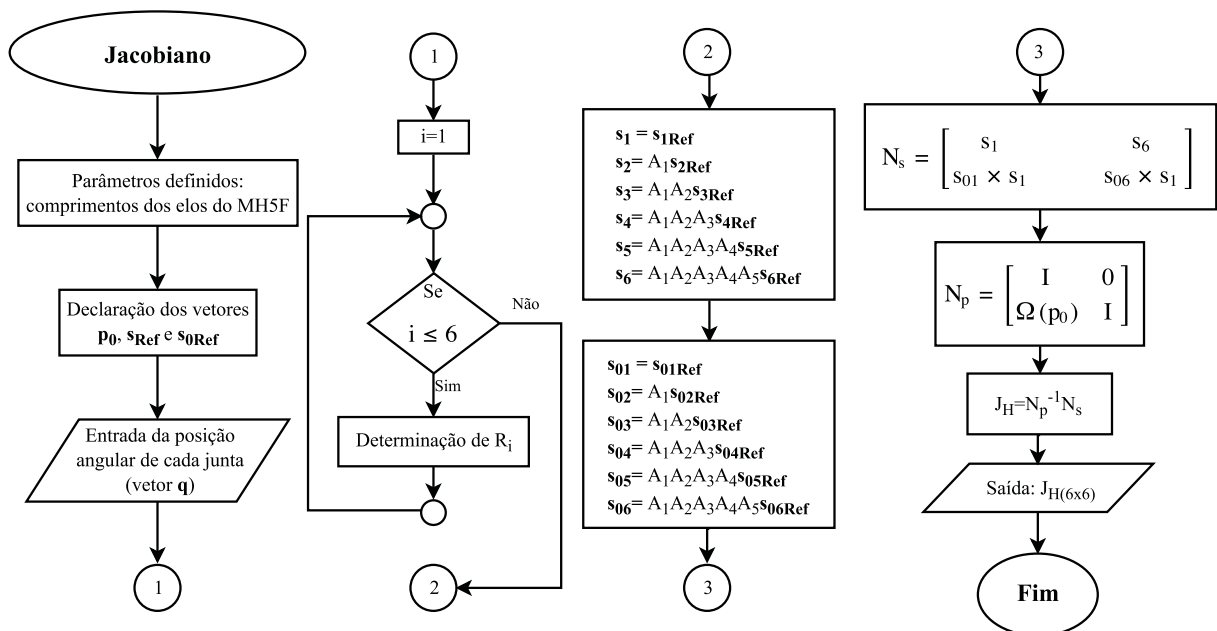


Figura 27 – Algoritmo para o cálculo do Jacobiano baseado em Helicoides para o manipuladores com 6 DOF

3.5.4 Algoritmo de cinemática inversa

O objetivo da cinemática inversa é solucionar a equação diferencial ordinária, dada pela Eq. 3.54, e obter como resposta o vetor de variáveis de junta, q .

A estratégia, descrita em Siciliano et al. (2010), consiste em: para uma dada trajetória a ser executada pelo efetuador final, conhecidas as condições iniciais do manipulador, obter a trajetória definida pelos vetores q e \dot{q} , no espaço das juntas, que faz com que o efetuador execute a trajetória previamente estabelecida. Para isso, considera-se a utilização do método numérico de Euler, considerado simples e adequado neste caso.

Considerando a postura inicial $q(0)$, as posições das juntas podem ser obtidas através da integração das velocidades com relação ao tempo:

$$q(t) = \int_0^t \dot{q}(\zeta) d\zeta + q(0) \quad (3.76)$$

Pelo método de Euler, tem-se que:

$$q(t_{k+1}) = q(t_k) + \dot{q}(t_k) \Delta t \quad (3.77)$$

onde o intervalo de tempo Δt representa o intervalo de integração do método e $t_{k+1} = t_k + \Delta t$.

Como as variáveis de junta são obtidas através de um método de Euler, existe um erro de integração inerente à sua utilização. A diferença entre a posição desejada x_d , determinada pela trajetória definida, e a posição do efetuador final x_e , obtida pela cinemática direta, define o erro, ou seja:

$$e = x_d - x_e = x_d - k(q_{t_k}) \quad (3.78)$$

$$\dot{e} = \dot{x}_d - \dot{x}_e \quad (3.79)$$

De acordo com a cinemática diferencial, a Eq. 3.79 pode ser reescrita como:

$$\dot{e} = \dot{x}_d - J(q)\dot{q} \quad (3.80)$$

Para garantir a convergência do método, deseja-se que o erro se apresente da seguinte forma:

$$\dot{e} + Ke = 0 \quad (3.81)$$

Considerando que a matriz do Jacobiano seja quadrada e não singular, ou seja, que o número de graus de liberdade n seja igual à dimensão do espaço necessário à execução da

trajetória, ao substituir a Eq. 3.81 na Eq. 3.80, isolando \dot{q} , tem-se que:

$$\dot{q} = J_H^{-1}(q)(\dot{x}_d + Ke) \quad (3.82)$$

Aplicando o método de integração descrito ao caso do problema aqui tratado, bem como o Método de Davis para o cálculo do Jacobiano baseado em Helicoides, para a obtenção das posições de junta, tem-se:

$$q_{s(t_{k+1})} = q_{s(t_k)} + [N_p^{-1}(q_{p(t_k)})N_s(q_{p(t_k)})]^{-1} \dot{q}_{p(t_k)} \Delta t \quad (3.83)$$

Para ilustrar o método aplicado na solução da cinemática inversa, a Fig. 28 mostra o algoritmo, na forma de fluxograma, para o caso de um manipulador com 6 DOF.

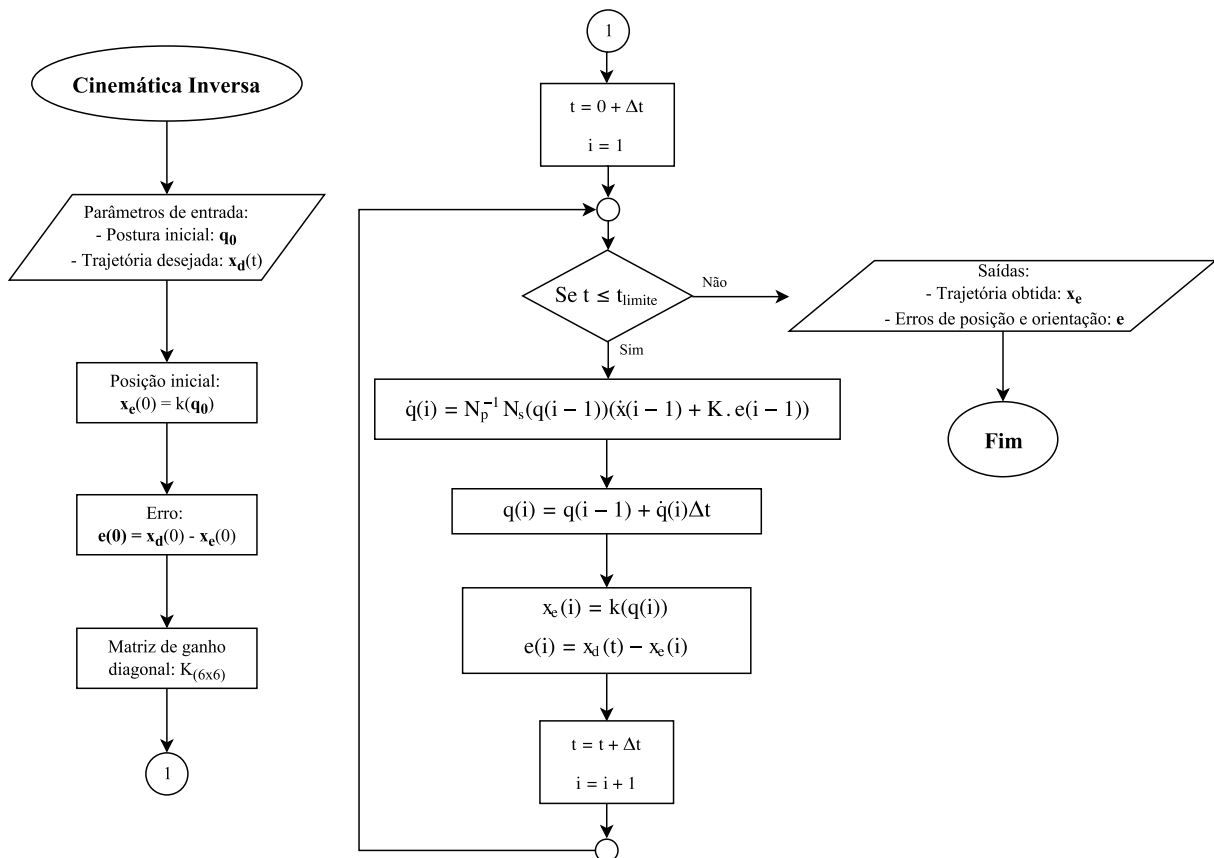


Figura 28 – Algoritmo para a para a solução da cinemática inversa de manipuladores com 6 DOF

4 Resultados

No presente capítulo são apresentados os resultados obtidos com a simulação de trajetória linear, bem como a modelagem do ambiente do Centro Integrado de Manufatura do IME e o protótipo do software de simulação *off-line*.

4.1 Simulação de trajetória

O objetivo dos algoritmos apresentados no capítulo anterior é a simulação de trajetórias a serem executadas pelo manipulador, ou seja, dado um percurso no espaço, o algoritmo deve gerar os valores dos ângulos de junta em função do tempo. Em termos práticos, não se define uma trajetória através de equações paramétricas. Na programação *on-line*, ao posicionar o robô em um determinado ponto, determina-se os ângulos de cada junta e, aplicando-se o algoritmo de cinemática direta, as seis coordenadas que definem a localização de um ponto são obtidas. Desta forma, são determinados dois pontos, entre os quais o robô deverá seguir uma trajetória linear ou circular com uma velocidade estabelecida pelo operador. A seguir, são apresentadas duas simulações envolvendo trajetórias linear e circular.

4.1.1 Trajetória linear (MOVL)

Aplicando os algoritmos a um caso prático, a cinemática inversa para o robô Motoman MH5F é solucionada para uma trajetória linear descrita pelas equações paramétricas que compõem o vetor $x_d(t)$, como se segue:

$$x_d(t) = k(q_0) + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 50t \\ 0 \\ 50t \end{bmatrix} \quad (4.1)$$

Como pode ser observado na Eq. 4.1, a trajetória é linear, com velocidade de 50 mm/s , desenvolvida no plano XZ . Partindo da localização $k(q_0)$, ou seja, da posição resultante da cinemática direta para $q_0 = [0^\circ, 45^\circ, -45^\circ, 0^\circ, 0^\circ, 0^\circ]^T$, o manipulador executa o movimento linear durante $4,5 \text{ s}$. Adotando um passo de integração de $0,001 \text{ s}$ ($\Delta t = 0,001 \text{ s}$), o seguinte resultado é obtido:

- a) Os valores de q e \dot{q} ao longo do tempo podem ser observados nas Fig. 29 e 30, respectivamente. Os resultados gráficos indicam que o algoritmo está de acordo com a proposta do movimento contínuo, ou seja, o manipulador respeita a trajetória imposta, realizando movimentos contínuos em cada junta com o objetivo de manter o efetuador na trajetória determinada.

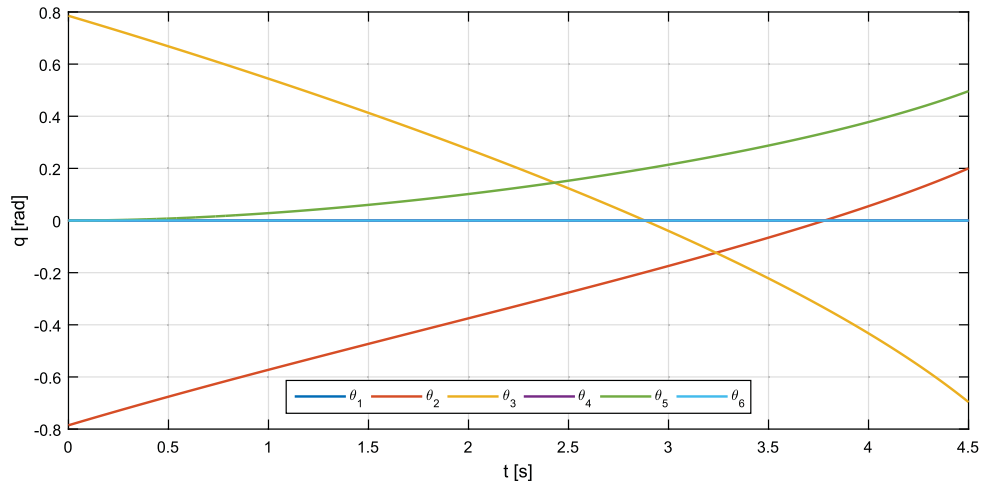


Figura 29 – Variação de q ao longo do tempo

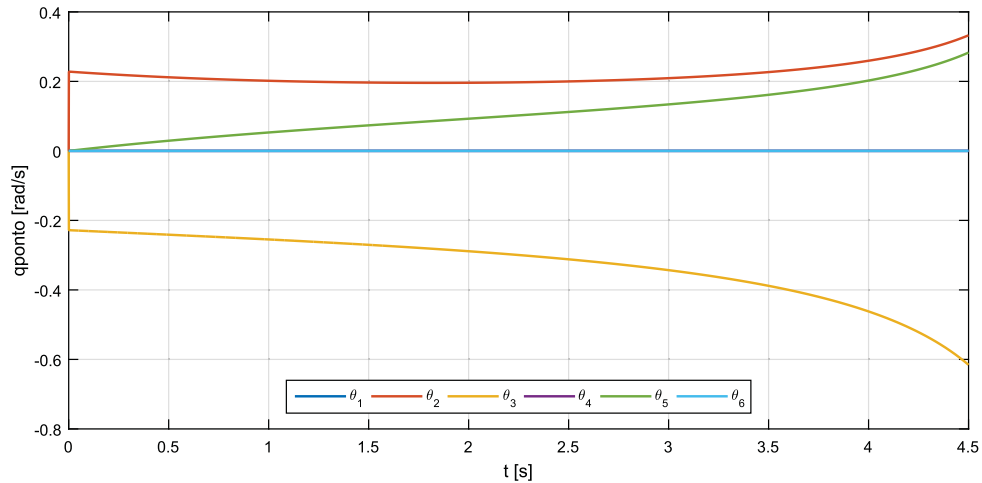


Figura 30 – Variação de \dot{q} ao longo do tempo

- b) Com relação ao erro, as Fig. 31 e 32, mostram que os erros de posição e orientação estão na ordem de 10^{-3} e 10^{-16} , respectivamente, tendo como origem o método de solução numérica utilizado. Com relação ao erro de posição, ele varia entre valores inferiores à repetibilidade manipulador de 0,02 mm, evidenciando a eficiência do método.

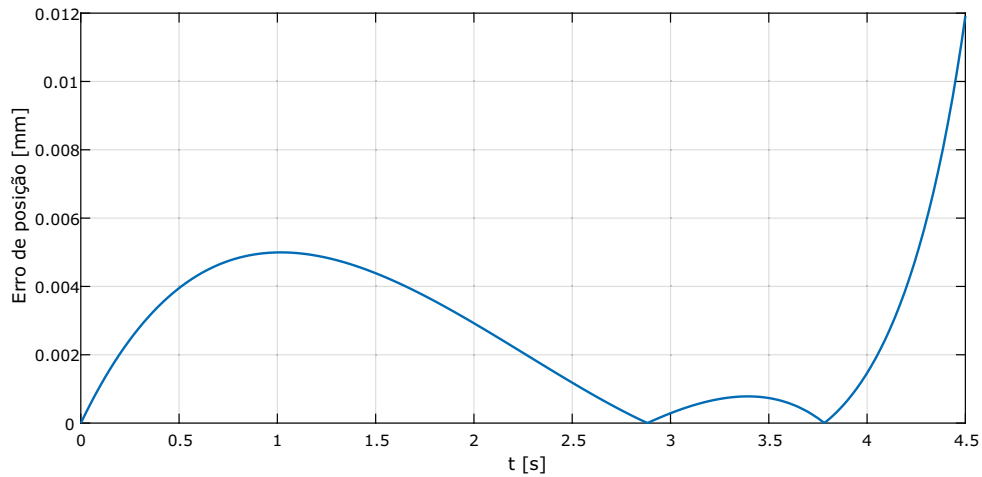


Figura 31 – Erro de posição ao longo do tempo

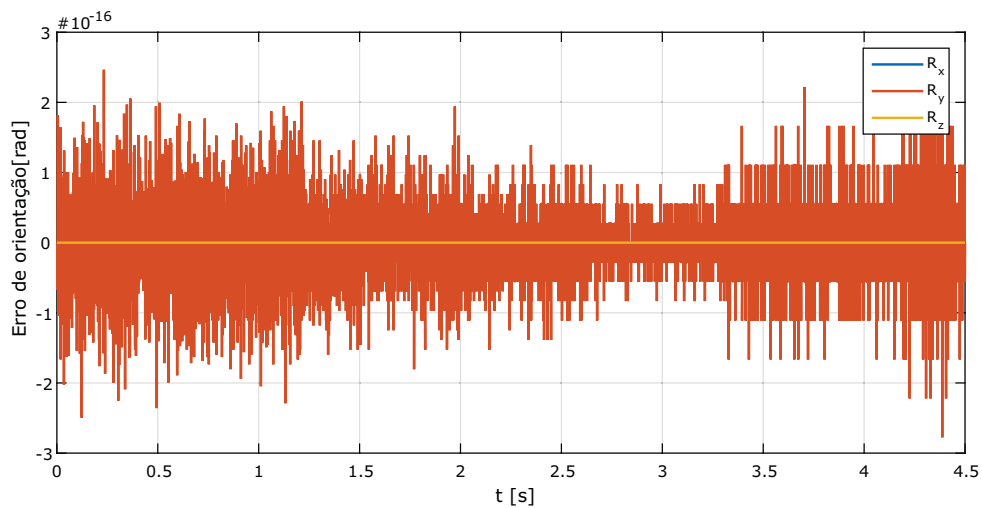


Figura 32 – Erro de orientação ao longo do tempo

Contextualizando para o caso do controlador FS-100, o comando **MOVL** possui dois parâmetros de localização e um de velocidade. A partir da leitura do comando propriamente dito, o manipulador executa a trajetória linear entre o ponto atual e o ponto declarado, com a velocidade determinada, como pode ser observado abaixo:

- 1 : *HOME*
- 2 : *MOVL P1101 V = 80*
- 3 : *MOVL P1102 V = 80*
- 4 : *MOVL P1103 V = 80*
- 5 : *MOVL P1101 V = 80*

A seqüência de comandos executa os seguintes movimentos:

1. Partindo da posição de HOME (linha 1), o manipulador executa a trajetória linear (MOVL) até o ponto P1101, com velocidade de 80 mm/s (linha 2)
2. Partindo do ponto P1101 (linha 2), o manipulador executa a trajetória linear (MOVL) até o ponto P1102, com velocidade de 80 mm/s (linha 3)
3. Partindo do ponto P1102 (linha 3), o manipulador executa a trajetória linear (MOVL) até o ponto P1103, com velocidade de 80 mm/s (linha 4)
4. Partindo do ponto P1103 (linha 4), o manipulador executa a trajetória linear (MOVL) até o ponto P1101, com velocidade de 80 mm/s (linha 5)

Sendo assim, aplicando o algoritmo estudado no presente trabalho, é possível simular o comando exemplificado. Adotando três posturas diferentes, indicadas abaixo pelos vetores q_1 , q_2 e q_3 , aplica-se o algoritmo da cinemática direta para obter os pontos que compõem a trajetória:

- Vetores que determinam as posturas:

$$\begin{aligned} q_1 &= (0.8726^\circ \quad 43.7843^\circ \quad -44.6477^\circ \quad 1.7733^\circ \quad 0.8640^\circ \quad 1.7739^\circ) \\ q_2 &= (0.2722^\circ \quad -35.0977^\circ \quad 28.6634^\circ \quad 5.5858^\circ \quad 6.4647^\circ \quad -5.5502^\circ) \\ q_3 &= (0.2750^\circ \quad -49.8347^\circ \quad 6.3776^\circ \quad 0.9127^\circ \quad 43.4594^\circ \quad -0.6635^\circ) \end{aligned}$$

- Pontos obtidos:

$$\begin{aligned} P1101 &= (371,020 \quad 5,741 \quad 259,204 \quad -3,5470 \quad -0,0002 \quad 0,8994) \\ P1102 &= (765,772 \quad 5,742 \quad 259,202 \quad -0,0002 \quad 0,0001 \quad 0,9001) \\ P1103 &= (765,776 \quad 5,779 \quad 19,197 \quad 0,0010 \quad 0,0013 \quad 0,9028) \end{aligned}$$

Aplicando-se o algoritmo de cinemática inversa, os valores dos ângulos das juntas ao longo do tempo são obtidos (Fig. 33), além da velocidade angular ao longo do tempo (Fig. 34). A partir da análise destes gráficos é possível notar a mudança repentina de direção, aproximadamente nos instantes 5 s e 8 s . Seria impossível implementar os movimentos gerados pela simulação em um sistema real pois, devido às mudanças de direção sem redução de velocidade, acarretaria em esforços excessivos na estrutura do manipulador e nos componentes, tais como os motores das juntas. Este trabalho não se propõe à análise da dinâmica do manipulador, restringindo-se à análise cinemática.

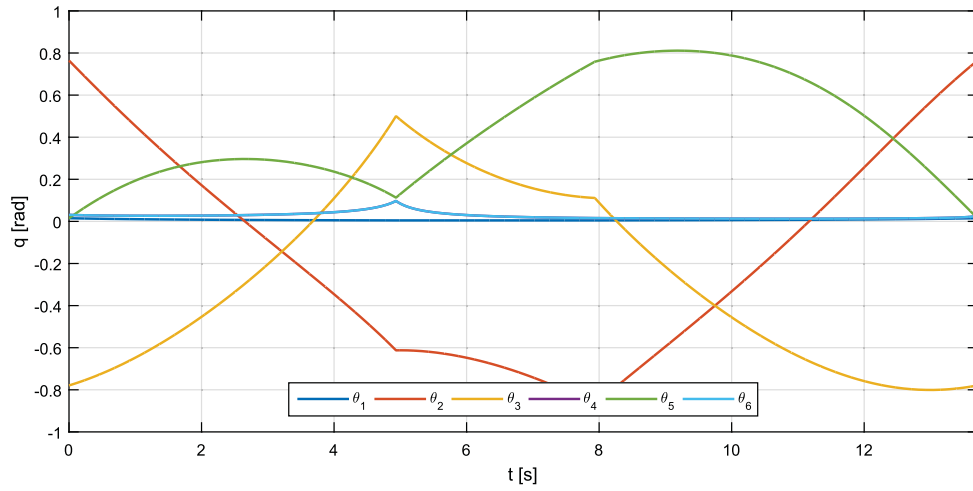


Figura 33 – Variação dos ângulos das juntas ao longo do tempo

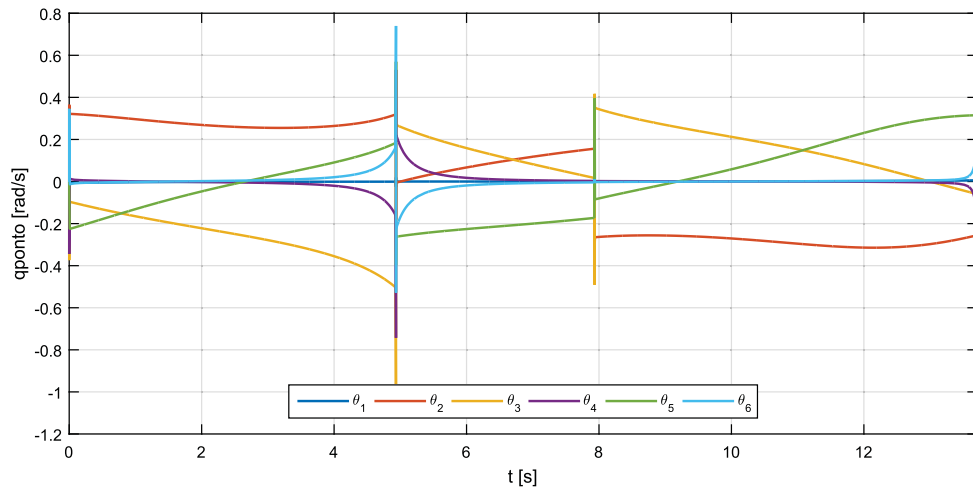


Figura 34 – Variação da velocidade angular das juntas ao longo do tempo

A Fig. 35 indica a trajetória linear entre os pontos indicados, gerada através da simulação, além do espaço de trabalho com o formato aproximado de uma esfera.

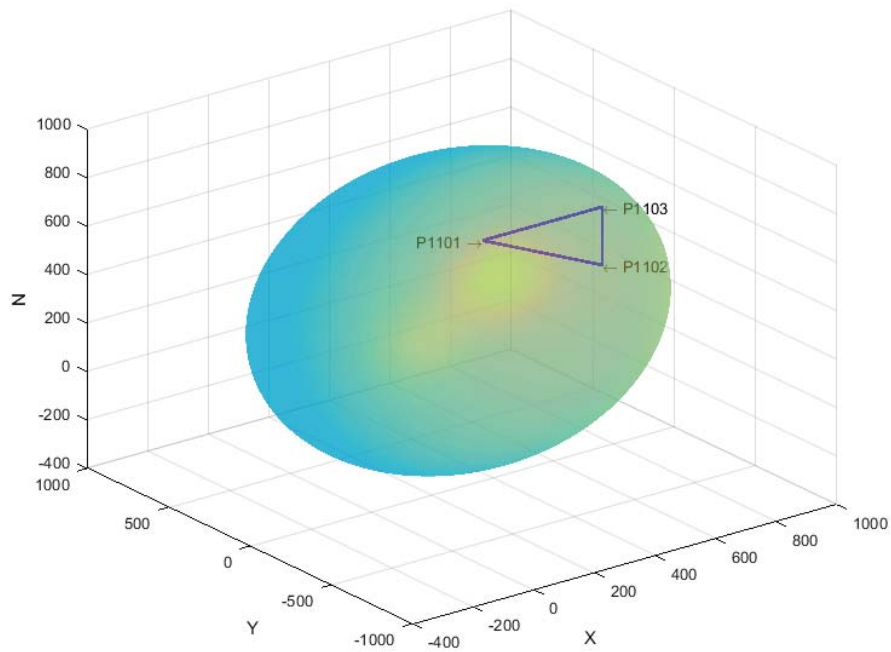


Figura 35 – Trajetória triangular obtida

Com a velocidade de 80 mm/s , o tempo de execução da trajetória foi de $13,7092 \text{ s}$. Quanto ao erro, os valores máximos obtidos foram:

- Orientação (Fig 36): $4,316 \cdot 10^{-6} \text{ rad.}$ (Fig 36)
- Posição (Fig 37): $0,0461 \text{ mm}$.

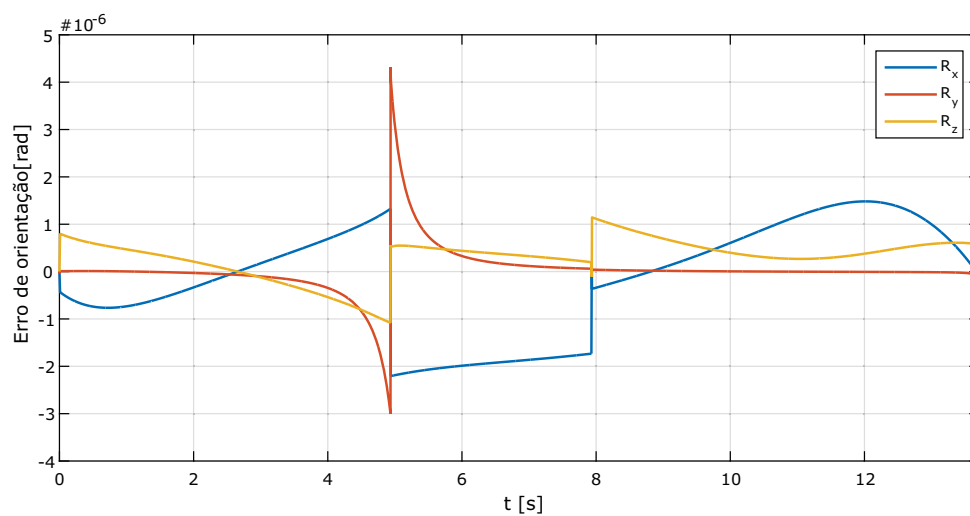


Figura 36 – Erro de orientação na trajetória em função do tempo

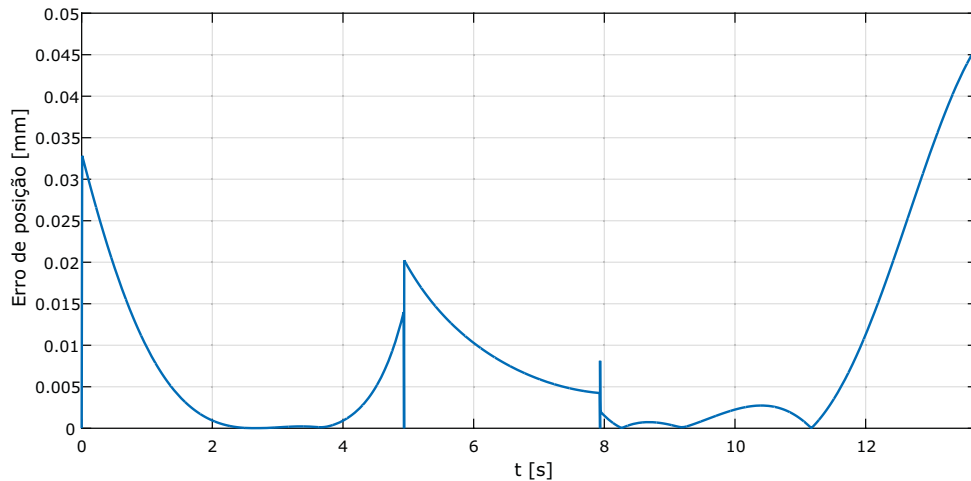


Figura 37 – Erro de posição na trajetória em função do tempo

4.1.2 Trajetória circular (MOVC)

O comando MOVC do controlador FS-100 possui três parâmetros de localização e um de velocidade. A partir dele, o manipulador executa a trajetória circular entre o primeiro e o último ponto declarados, passando pelo segundo, com a velocidade determinada, como pode ser observado abaixo:

```

1 : HOME
2 : MOVL P1104 V = 80
3 : MOVC P1104 V = 80
4 : MOVC P1105 V = 80
5 : MOVC P1106 V = 80

```

A sequência de comandos executa os seguintes movimentos:

1. Partindo da posição de HOME (linha 1), o manipulador executa a trajetória linear (MOVL) até o ponto P1104, com velocidade de 80 mm/s (linha 2)
2. Partindo do ponto P1104 (linha 3), o manipulador executa a trajetória circular (MOVC) até o ponto P1106, com velocidade de 80 mm/s (linha 5), passando pelo ponto P1105 (linha 4)

A fim de exemplificar a trajetória circular, a simulação do movimento pode ser executada aplicando os algoritmos estudados. Utilizando os vetores q_4 , q_5 e q_6 indicados abaixo, aplica-se o algoritmo da cinemática direta para obter os pontos que compõem a trajetória desejada:

- Vetores que determinam as posturas:

$$q_4 = (0,0000^\circ \quad 55,0000^\circ \quad -55,0000^\circ \quad 0,0000^\circ \quad 0,0000^\circ \quad 0,0000^\circ)$$

$$q_5 = (0,0000^\circ \quad -17,1455^\circ \quad 29,4940^\circ \quad 0,0000^\circ \quad 4,3485^\circ \quad 0,0000^\circ)$$

$$q_6 = (0,0000^\circ \quad -20,4984^\circ \quad -49,3731^\circ \quad 0,0000^\circ \quad 69,8625^\circ \quad 0,0000^\circ)$$

- Pontos obtidos:

$$P1104 = (331,063 \quad 0,000 \quad 219,809 \quad 0,0000 \quad 0,0000 \quad 0,0000)$$

$$P1105 = (672,477 \quad 0,000 \quad 359,234 \quad 0,0000 \quad 0,0000 \quad 0,0000)$$

$$P1106 = (531,069 \quad 0,000 \quad 17,804 \quad 0,0000 \quad 0,0000 \quad 0,0000)$$

Resolvendo a cinemática inversa com os algoritmos detalhados nas Fig. 27 e 28, os ângulos das juntas ao longo do tempo são obtidos (Fig. 38), bem como as velocidades angulares das juntas ao longo do tempo (Fig. 39).

Aplicando-se o algoritmo de cinemática inversa, os valores dos ângulos das juntas ao longo do tempo são obtidos (Fig. 38), além da velocidade angular ao longo do tempo (Fig. 39).

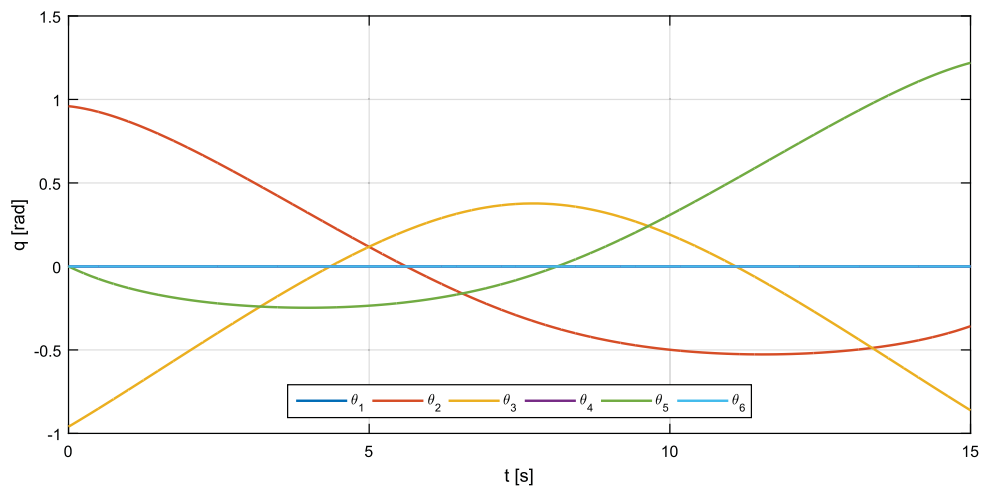


Figura 38 – Variação dos ângulos das juntas ao longo do tempo

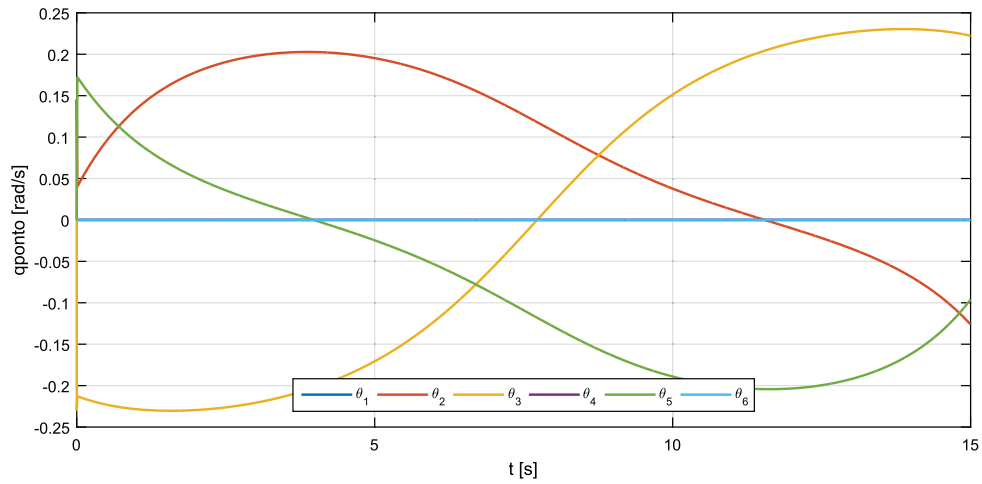


Figura 39 – Variação da velocidade angular das juntas ao longo do tempo

A Fig. 40 indica a trajetória linear entre os pontos indicados, gerada através da simulação, além do espaço de trabalho com o formato aproximado de uma esfera.

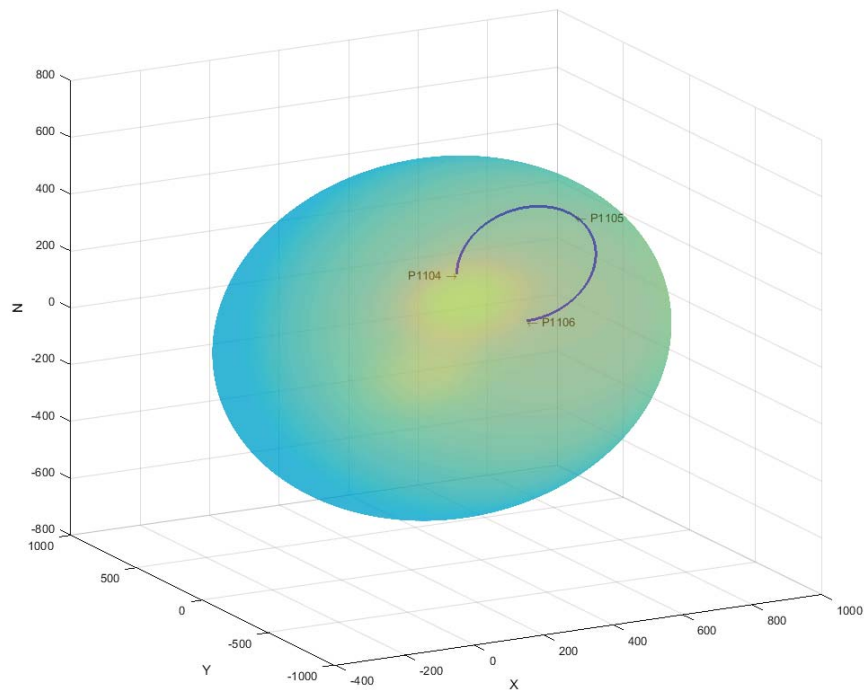


Figura 40 – Trajetória circular obtida

O tempo de execução da trajetória foi de 15 s. Quanto ao erro, os valores máximos obtidos foram:

- Orientação, em módulo (Fig 41): $7,216 * 10^{-16} \text{ rad}$, atribuído ao método de solução numérica utilizado.
- Posição (Fig 42): $0,0379 \text{ mm}$.

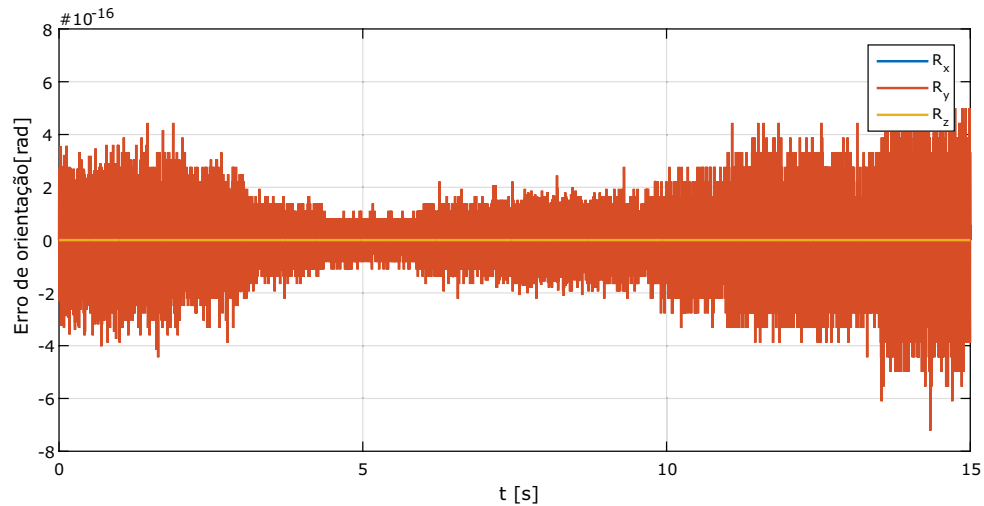


Figura 41 – Erro de orientação na trajetória em função do tempo

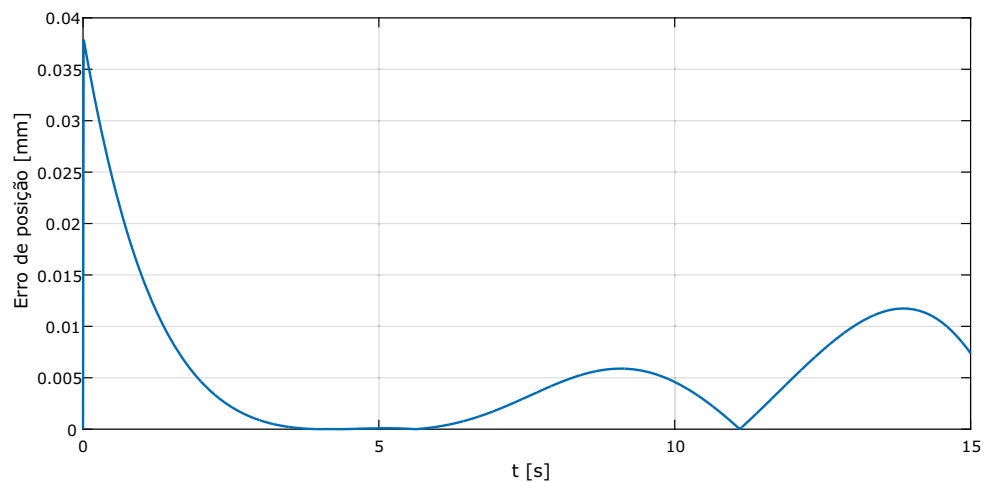


Figura 42 – Erro de posição na trajetória em função do tempo

Para melhor interpretação dos resultados obtidos até aqui, é interessante que haja uma visualização dos movimento do manipulador. Com a finalidade de proporcionar tal potencialidade os resultados foram aplicados a um *software* desenvolvido por alunos de pós graduação do Instituto Militar de Engenharia, que será abordado na seção a seguir.

4.2 Software de simulação de robôs seriais

O *Computer-Aided Serial Robots Programming* (CARPA) é uma interface computacional amigável e intuitiva desenvolvido na versão acadêmica de ambiente desenvolvimento integrado (IDE) de softwares (RAD Studio Berlin) em Delphi por alunos de mestrado, com auxílio de alunos de graduação, no *Industrial and Defense Robotics Laboratory (IDR Lab)*. Tal interface permite a inclusão de cenários, robôs e outros componentes dos ambientes modelados em CAD 3D.

Como já mencionado no Cap.3.1, o tempo de programação *on-line* dentro de uma linha de montagem obriga a intermitência da produção. Por outro lado, a programação *off-line* permite a ininterruptão de processos produtivos enquanto o programador realiza sua atividade remotamente, reduzindo a parada da produção somente ao tempo destinado à calibração e validação da tarefa programada. O CARPA tem seu algoritmo organizado de maneira modular, podendo ser utilizado para qualquer robô industrial com cadeia cinemática aberta, além de ter a potencialidade de receber novas funcionalidades. Nesse sentido, o programa já possui um módulo voltado ao robô AMATROL PEGASUS, existente no laboratório, e, neste trabalho, recebeu um módulo que contempla algumas das funcionalidades do robô Yaskawa MOTOMAN MH5F. A Fig. 43 mostra a tela inicial do referido programa com o manipulador objeto de estudo.

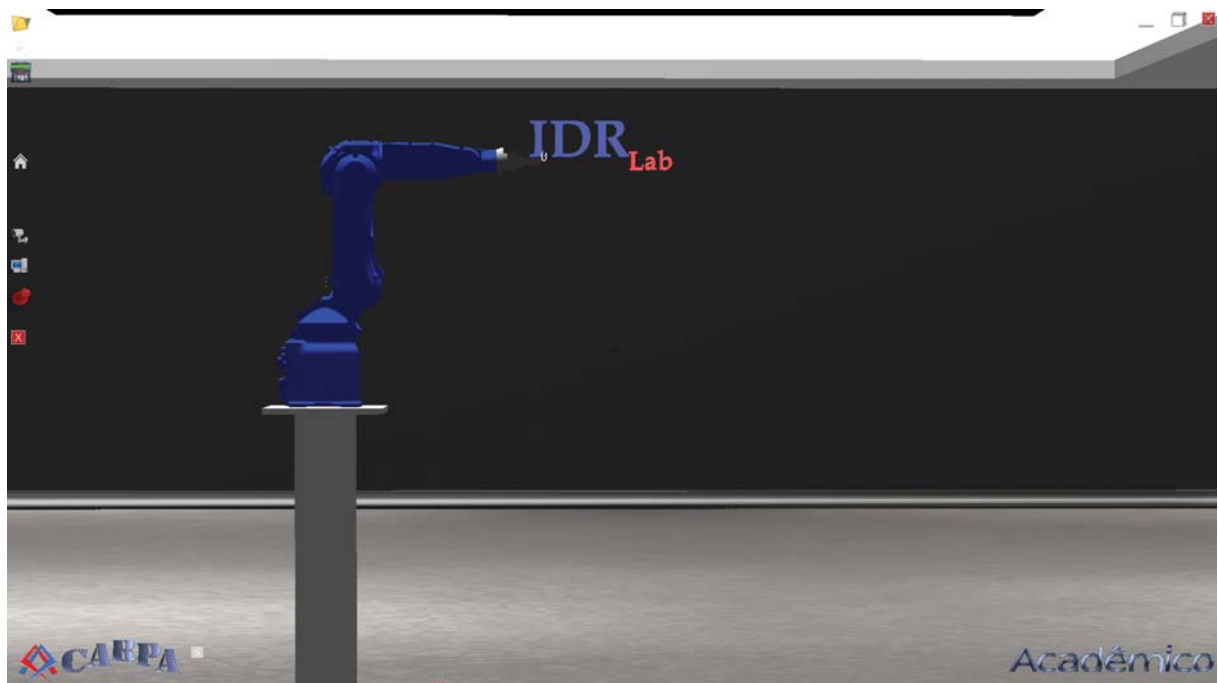


Figura 43 – Tela inicial do Carpa com o Robô Yaskawa MOTOMAN MH5F

O passo inicial para a programação de uma tarefa é o ensino de pontos, que no

CARPA são obtidos pela cinemática direta com base na Teoria dos Helicoides, apresentada na Sec. 3.4. A Fig. 44 mostra a janela de visualização da localização e da postura do manipulador. Ao selecionar os botões “*Teach Pendant*” e “Lupa”, no canto superior esquerdo, tem-se a imagem contida na Fig. 44, na qual as juntas S(*Swing*), L(*Lower arm*), U(*Upper arm*), R(*Wrist Roll*), B(*Wrist Bend*) e T(*Tool*) estão destacadas na árvore de seleção dos componentes, bem como a orientação de cada helicóide. Ao selecionar uma das juntas, pode-se alterar o valor de seu ângulo usando as setas amarelas ou simplesmente digitando seu valor, em graus, no campo ao lado esquerdo das mesmas. Uma vez que o usuário esteja satisfeito com a localização do efetuador final, basta usar o botão “TCH” para gravar esse ponto. Os pontos ensinados são mostrados no campo correspondente assinalado na Fig. 45.

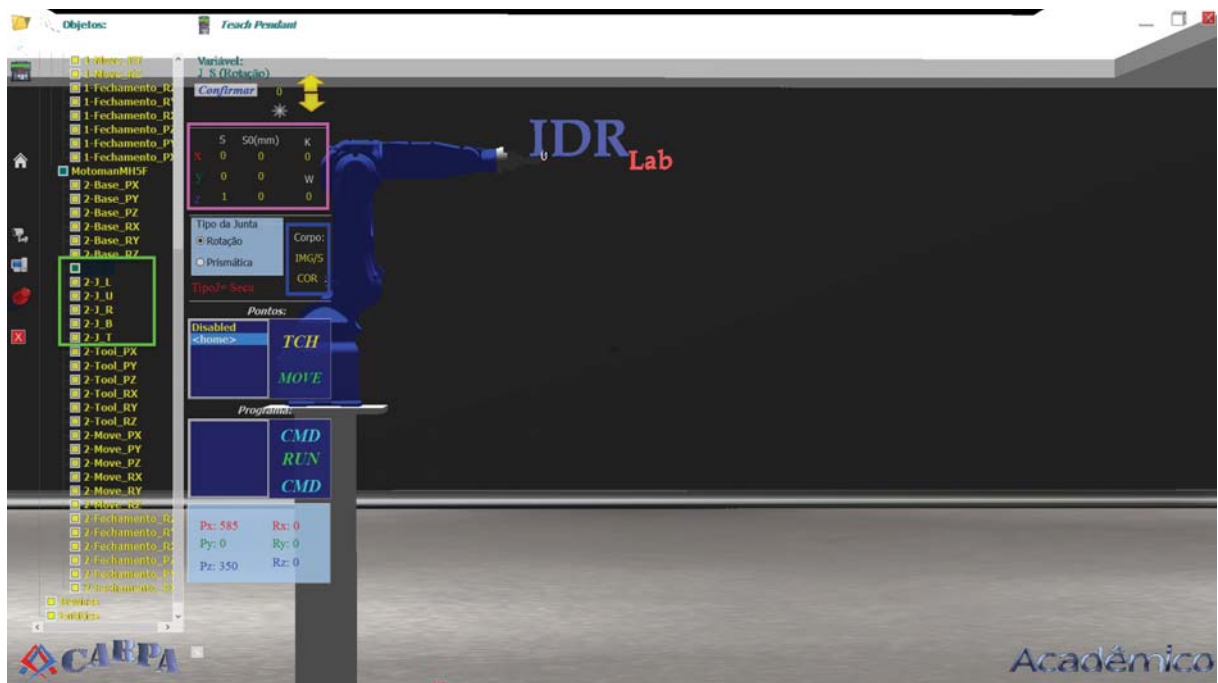


Figura 44 – Juntas e helicóide relativo à junta selecionada



Figura 45 – Ponto ensinado em destaque

Uma vez que os pontos necessários à realização da tarefa foram ensinados, o programador pode iniciar sua construção. Ao clicar no botão “CMD” abre-se uma janela para a formulação da tarefa. Essa janela recebe os pontos ensinados e permite chamar 4 comandos, conforme apresentados no apêndice C: MOVL, MOVJ, DOUT e Timer. O comando MOVL correspondente à trajetória linear, exige como parâmetros o ponto ao qual se deseja chegar e a velocidade variando de 0,1 a 1500mm/s. O comando MOVJ realiza o movimento da garra de forma otimizada, no qual a trajetória não pode ser determinada pelo usuário. Neste caso, a velocidade é dada em taxa percentual de 0,1 a 100% da velocidade máxima de operação nessa interpolação. O controle de *Outputs* permite selecionar o número da saída a ser utilizada, sendo o movimento da garra controlado pelo Output 1, conforme a numeração do laboratório. A garra tem os movimentos de fechar e abrir como OFF e ON respectivamente.

O robô tem um tempo de resposta muito acelerado. Desta forma, quando um ponto que necessite agarrar ou soltar algum objeto é alcançado, há a necessidade de incluir um pequeno atraso na execução (1 s), caso contrário ocorrerá o movimento da garra concomitantemente com o da cadeia cinemática em direção ao próximo ponto da sequência da tarefa. O comando *Timer* proporciona a inclusão de atrasos nas tarefas, sendo necessariamente declarado em segundos.

Por meio do caminho Arquivo→Salvar, o usuário pode exportar o programa criado em

formato *.JBI*, o arquivo lido pelo controlador do manipulador. De posse do arquivo, por intermédio das entradas USB (*Universal Serial Bus*) ou de cartões de memória (*compact flash*) localizadas no PP, é possível executar o programa no robô com a finalidade de validar a tarefa e, após esta etapa, executá-la. A Fig. 46 ilustra o módulo de programação, incorporado ao CARPA, voltado ao manipulador em estudo.

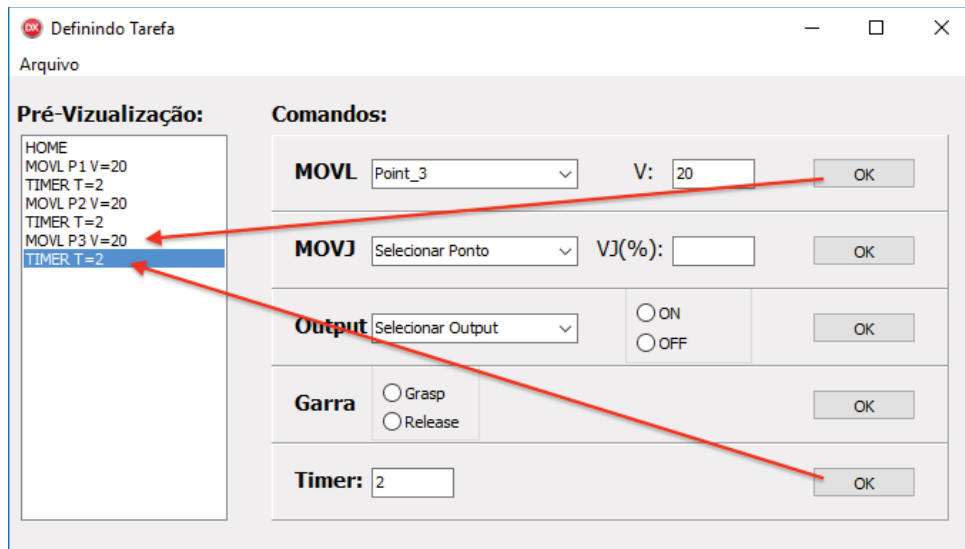


Figura 46 – Funcionamento da janela de comandos

Com tais funcionalidades incorporadas ao programa principal, foram ensinados alguns pontos ao manipulador, já no ambiente virtual. Cabe ressaltar que os mesmos pontos utilizados na simulação foram ensinados por meio da edição dos ângulos de cada junta e as localizações do efetuator final, equivalente às obtidas na Sec. 4.1.2, são determinadas pelo CARPA, como mostram as Fig. 47 e 48. Destaca-se também a visualização das trajetórias geradas pelos pontos que a determinam.



Figura 47 – Pontos que determinam a trajetória triangular obtidos no CARPA



Figura 48 – Pontos que determinam a trajetória circular obtidos no CARPA

4.3 Modelagem do ambiente virtual de simulação

O escopo do trabalho está inserido em um contexto maior de colaboração entre alunos de graduação e pós Graduação do Instituto Militar de Engenharia, seguindo a linha de pesquisa de um *software*, já em desenvolvimento, de análise cinemática de robôs industriais, o CARPA, que tem sua interface voltada à simulação em ambiente de realidade virtual.

Para a montagem do ambiente de realidade virtual do referido software, há a necessidade de modelagem dos objetos em algum *software* CAD. Em consequência, para realização de simulações futuras no CARPA, este grupo de trabalho contribuiu com algumas modelagens do Centro Integrado de Manufatura (Laboratório de Mecatrônica) em *SolidWorks*.

As Fig. 49, 50 e 51 apresentam as principais contribuições do grupo de trabalho referente à modelagem em CAD 3D para a montagem do futuro cenário no ambiente do CARPA, que são as estações com o centro de usinagem, o centro de soldagem e a metrologia computadorizada.

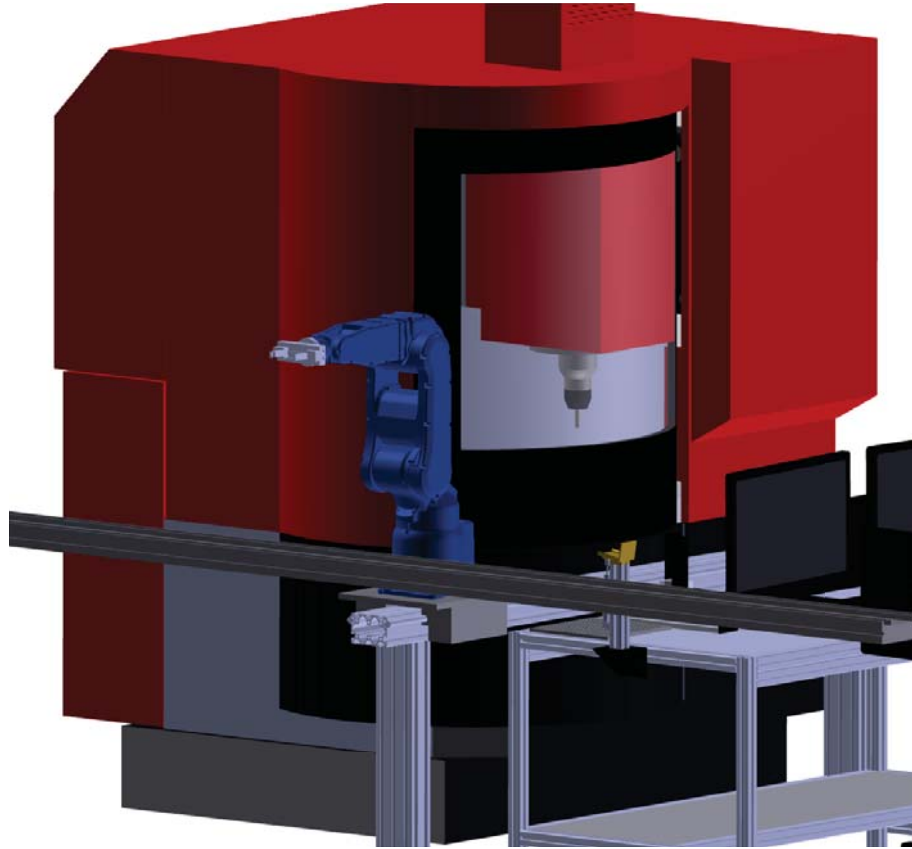


Figura 49 – Vista com estação do centro de usinagem

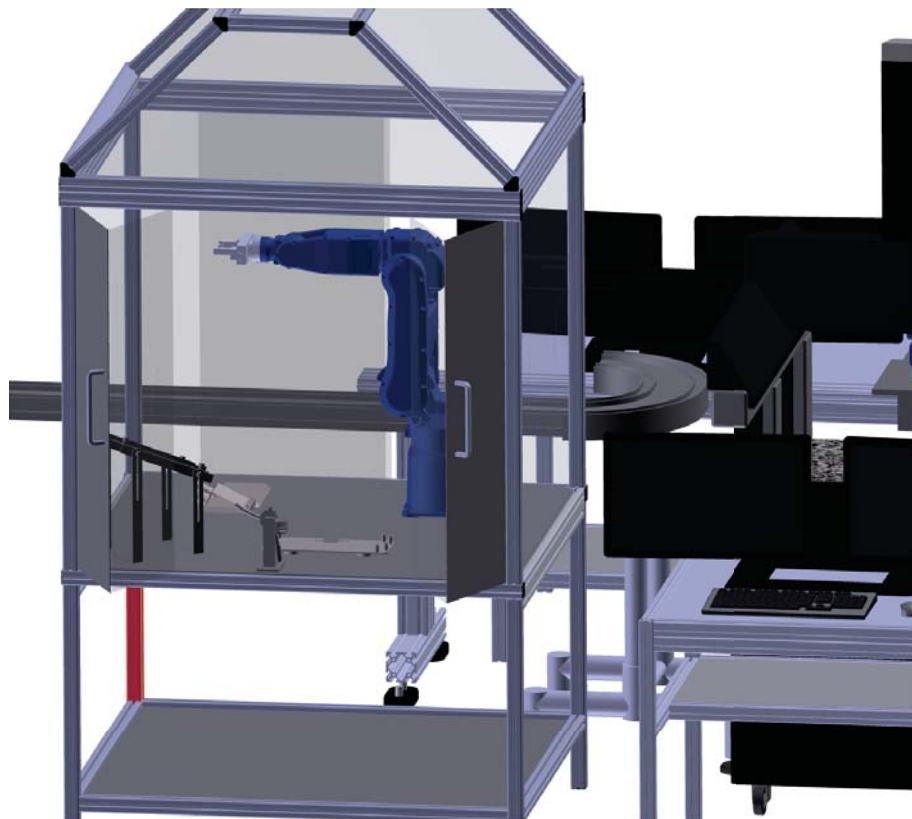


Figura 50 – Vista com estação do centro de soldagem

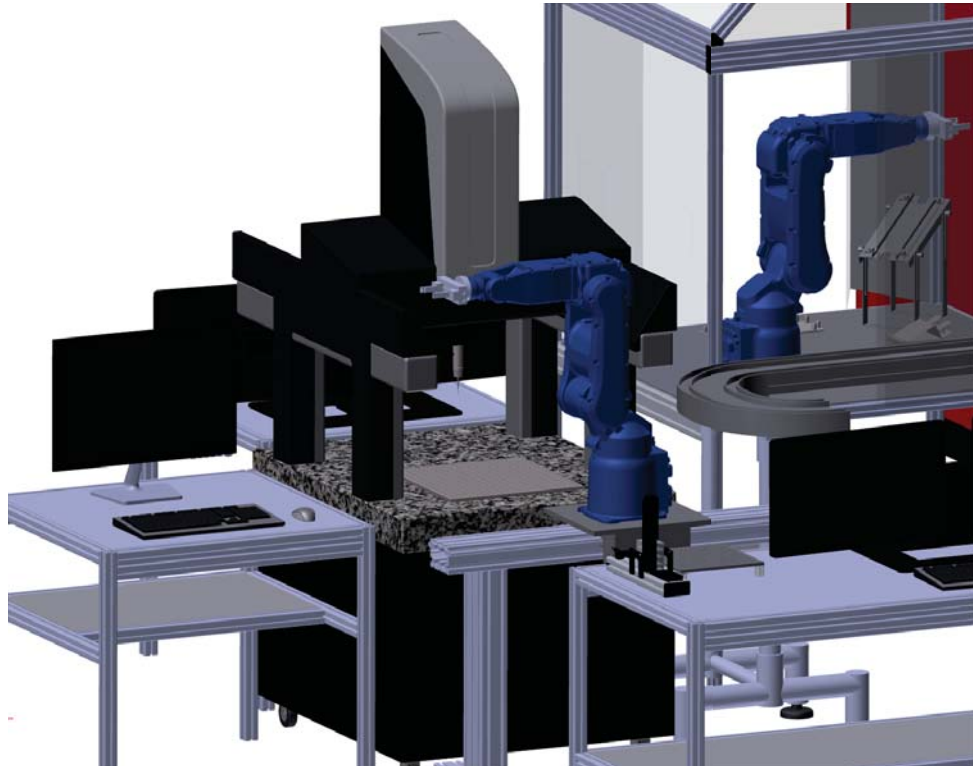


Figura 51 – Vista com estação do metrologia computadorizada

5 Conclusões e Perspectivas

Nesse capítulo serão apresentadas as conclusões do Projeto Final de Curso. A Sec. 5.1 traz um resumo dos capítulos anteriores. A Sec. 5.2 trata da conclusão dos resultados obtidos e as contribuições científicas dos autores. Finalmente, na Sec. 5.3 são expostas perspectivas para trabalhos futuros.

5.1 Recapitulação sintetizada

No Cap. 3.1 é apresentado um breve histórico do desenvolvimento da robótica, ressaltando a motivação para tal desenvolvimento. Também mostra a proposta do presente trabalho que consiste na resolução cinemática do robô industrial Motoman MH5F utilizando a Teoria dos Helicóides.

O Cap. 2 consiste em explicar a robótica industrial, ramo em que o presente trabalho tem seu escopo voltado. Em suas seções mostra um histórico e qual a definição da robótica, a diferença entre a robótica industrial e avançada e os tipos de manipuladores geralmente utilizados no contexto industrial. Além disso, as vantagens e desvantagens da programação *On-line* e *Off-line* e uma síntese do robô Motoman MH5F.

No Cap. 3 é abordada uma revisão bibliográfica sobre localização de corpo rígido no espaço e Teoria dos Helicoides. Na sequência, são apresentados métodos de resolução da cinemática direta e inversa já aplicadas ao manipulador Yaskawa MOTOMAN MH5F.

No Cap. 4 são realizadas simulações para a execução de trajetórias linear e circular realizadas pelo robô em seu volume de trabalho, por intermédio dos comandos MOVJ e MOVL, respectivamente no controlador FS-100. Com base nos estudos de algumas potencialidades do robô, foi implementado um protótipo de módulo voltado à simulação e programação remota do manipulador em estudo e incorporado ao *software*, interface gráfica de análise de robôs seriais desenvolvida no *IDR Lab*. As funcionalidades que foram integradas ao CARPA desse protótipo foram: a implementação das trajetórias otimizada (MOVJ) e linear (MOVL), a programação de tarefas (*JOB*) e a exportação do arquivo com o mesmo protocolo que o criado pelo controlador em formato tipo *.JBI*.

5.2 Conclusões

A proposta do trabalho e os objetivos foram alcançados e são importantes para o complemento de trabalhos futuros no IDR Lab. Baseado nas realizações apresentadas ao longo do texto, foram possíveis obter algumas conclusões

A Teoria dos Helicoides facilita a modelagem cinemática de manipuladores seriais pois utiliza apenas um referencial fixo e outro móvel, diferentemente de outras técnicas de modelagem mais populares como a de Denavit-Hartenberg. Além disso, a solução da cinemática diferencial através do Método de Davies, utilizando cadeias virtuais, é uma abordagem de fácil entendimento e aplicável à manipuladores com cadeia cinemática aberta ou fechada.

O CARPA é genérico e tem um ambiente amigável de modelagem e simulação de robôs, dispensando o conhecimento técnico por parte do usuário de teorias cinemáticas. Assim, ele facilita a modelagem, simulação e análise de cenários contendo manipuladores industriais de forma simples e intuitiva, permitindo obter a resolução cinemática do robô modelado e reduzindo o tempo necessário da programação *On-line*.

Com o presente trabalho integrar ao CARPA, que já possuía uma programação assistida e a obtenção do código de maneira automatizada para o robô Pegasus, funcionalidades análogas para o Motoman MH5F, restando ainda para trabalhos futuros a implementação do simulador das funcionalidades do PP.

5.3 Perspectivas

A partir das conclusões apresentadas, é possível indicar algumas possibilidades de aprofundamento em temas vinculados a este trabalho. O CARPA ainda apresenta oportunidades de melhoria. O módulo do robô Yaskawa MOTOMAN MH5F a ele incorporado trata-se apenas de um protótipo e requer que sejam implementadas outras funcionalidades contidas no *Programming Pendant* (PP) do manipulador. Estudos na área de implementação de tais funcionalidades, além da montagem de cenários com precisão aceitável relativa ao posicionamento dos componentes e as respectivas características geométricas são oportunidades de contribuição para novos trabalhos. Desta forma o ponto ensinado no ambiente virtual será equivalente ao do cenário real, permitindo a programação de tarefas com tempo ainda menor de uso do robô, visto que haverá menos discrepância na validação e calibração.

Apêndices

APÊNDICE A – Procedimentos de segurança

Antes de operar o manipulador é necessário observar as seguintes medidas de segurança:

1. remover todos os obstáculos da área de trabalho do robô;
2. procurar por sinais de avaria no robô;
3. retirar peças de vestuário soltas como gravatas, lenços e mangas;
4. prender cabelos longos;
5. remover o PP da área de trabalho do robô;
6. localizar o botão de parada de emergência;
7. verificar o botão de parada de emergência: o botão Servo ON no PP deve estar aceso enquanto a alimentação do servomotor estiver ligada. Para certificação do funcionamento dos botões de parada de emergência tanto do FS100 quanto do PP é necessário proceder conforme se segue:
 - pressionar o botão de parada de emergência no controlador ou no PP;
 - confirmar se a alimentação do servomotor está DESLIGADA (o LED no botão [SERVO ON] do PP permanece aceso enquanto a alimentação do servomotor está ligada e apaga quando cessa a alimentação dos motores); e,
 - depois de confirmar o passo anterior, deve-se pressionar a tecla [SERVO ON READY] no PP e, assim a alimentação do servomotor é ligada (a alimentação do servomotor pode ser ligada enquanto o botão [SERVO ON], estiver piscando).
8. para o ensino de tarefas, selecionar o modo teach, colocando a chave seletora do modo de operação na posição [TEACH], impedindo a operação por meio do botão [START] ou de uma entrada externa;
9. assegurar-se de que as pessoas na área estão usando óculos de segurança; e,
10. assegurar-se de que as pessoas estão fora da área de trabalho do robô.

O botão de parada de emergência está localizado no canto superior direito do PP. Ele deve ser acionado sempre que houver risco eminente de dano, seja a alguém, à materiais dispostos ou instalados na área de trabalho ou ao próprio manipulador, sendo o principal meio de se evitarem colisões. Para acionar a parada de emergência basta pressionar o botão de parada de emergência e para desativar a parada, basta girar para a direita para que ele seja liberado, como é mostrado na Fig. 52.



Figura 52 – Desarmando o botão de parada de emergência

APÊNDICE B – Procedimento de Programação On-Line do Sistema Motoman (FS-100)

O controlador FS-100 possui três modos de operação distintos: modo ensino, modo execução e modo remoto.

- No **modo *Teach*** é possível realizar a preparação, o ensino e a modificação de tarefas e a mudança de parâmetros.
- No **modo *Play*** as tarefas ensinadas são executadas e podem ser realizadas alterações nas configurações.
- No **modo *Remote***, as tarefas como inicialização e mudança de ciclo de operação podem ser comandadas por sinais de *inputs* externos (comandados por computadores externos).

A seleção dos modos de operação bem como o gerenciamento das configurações e todo o procedimento de programação e execução de tarefas são realizados através do *Programming Pendant*(PP), única interface homem-máquina do sistema robótico apresentado neste trabalho. O PP é equipado com teclas e botões usados para movimentar o manipulador na programação de operações e na edição de tarefas. A Fig. 53 ilustra uma visão geral do PP.

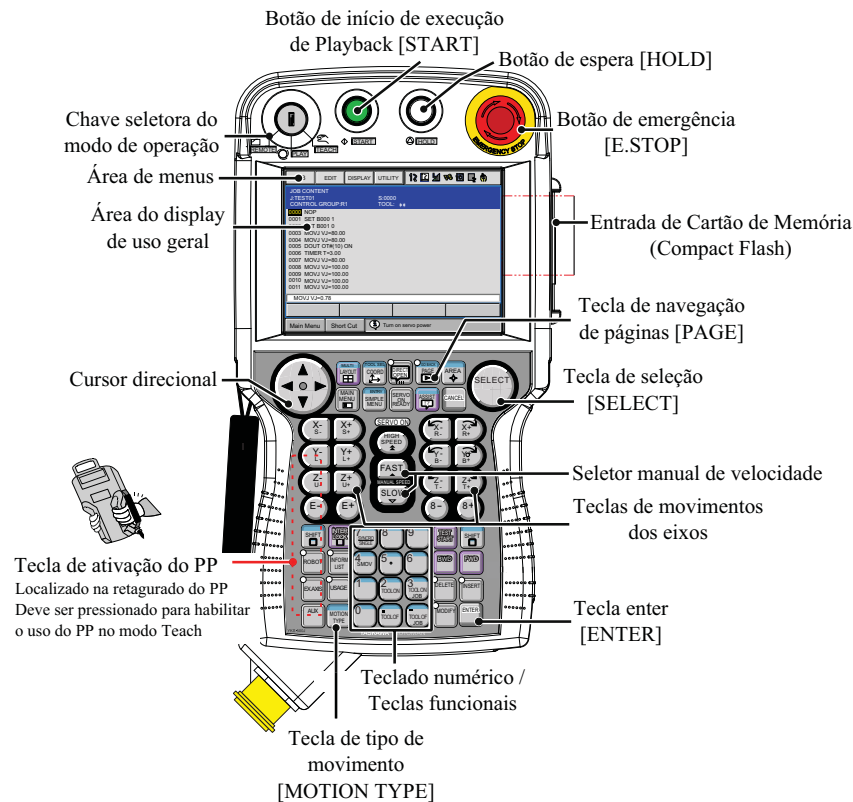


Figura 53 – Visão geral do *Programming Pendant*, (adaptado de YASKAWA (2014))

Para a manutenção da integridade do sistema robótico, o controlador possui um sistema de segurança que permite a realização de operações e a modificação de configurações de acordo com o nível de treinamento daquele que irá operar o robô. Os chamados **modos de segurança** são perfis que permitem determinada execução de tarefas de acordo com o grau de instrução do usuário. São eles:

- **Modo Operação:** permite a execução de operações básicas com o robô (iniciar, parar, etc.). Ele é indicado para aqueles que operam os robôs nas linhas de produção;
- **Modo Edição:** este modo permite o operador ensinar, modificar trabalhos e as configurações dos robôs. Para acessar o modo edição, é solicitada uma senha que contém entre 4 (quatro) e 9 (nove) caracteres; e,
- **Modo Gerenciamento:** permite que usuários autorizados realizem a manutenção e a instalação do sistema robótico, bem como a modificação de parâmetros e de identificação de usuário e suas respectivas permissões. Para acessar o modo gerenciamento, também é solicitada uma senha de acesso.

Como primeiro passo para a programação *on-line*, é necessário compreender a operação do robô nos diversos sistemas de coordenadas, ou seja, a descrição da operação e dos mo-

vimentos resultantes das operações nos seis diferentes sistemas de coordenadas disponíveis: coordenadas das juntas, cartesiana, cilíndrica, da ferramenta e do usuário.

B.0.1 Operações nos sistemas de coordenadas do robô

Para a operação do manipulador nos sistemas de coordenadas é necessário selecionar o modo ensino no PP. O controlador subdivide o sistema robótico em conjuntos de eixos, denominados **Grupos de Controle**. Além do grupo **robô**, conjunto dos seis eixos de rotação do manipulador, existem os eixos externos, podendo ser classificados como **base**, que movimenta paralelamente o robô (movimento de translação), ou **estação**, que executa a rotação desse.

Os quatro primeiros passos para a execução de movimentos em diferentes sistemas de coordenadas são:

1. selecionar o Grupo de Controle desejado: teclas [SHIFT] + [ROBOT], para selecionar o grupo robô, ou teclas [SHIFT] + [EX. AXIS], para selecionar os grupos externos, base ou estação;
2. selecionar o sistema de coordenadas desejado através da tecla [COORD];
3. selecionar a velocidade desejada entre as opções "INCH", "SLW", "MED" e "FST", através das teclas [FAST] ou [SLOW]; e,
4. ativar os servos motores, pressionando a tecla [SERVO ON READY].

Durante a fase de ensino, a velocidade do TCP do robô é limitada em 250 *mm/s*. Com exceção da velocidade "INCH", para utilizar uma velocidade superior ao limite imposto, é necessário pressionar a tecla [HIGH SPEED].

Para executar movimentos nos diferentes eixos do manipulador, algumas teclas específicas do PP são utilizadas conforme as descrições abaixo:

- **Sistema de coordenadas das juntas:** como indicado na Fig. 54, cada eixo ou junta do robô se move independentemente. Nele é possível mover até dois eixos simultaneamente.

Sistema de Coordenadas das Juntas

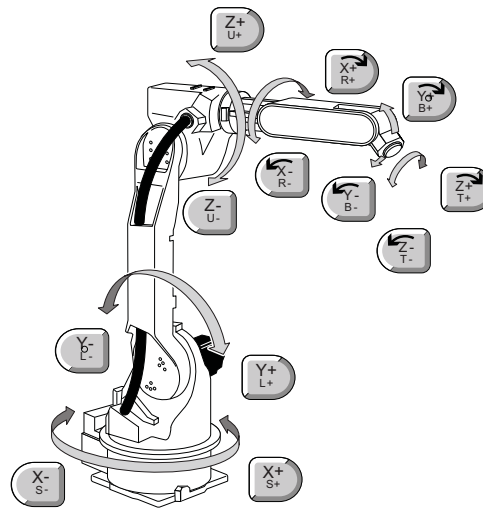


Figura 54 – Sistema de coordenadas das juntas (YASKAWA (2014))

- **Sistema de coordenadas cartesianas:** o efetuador descreve movimentos paralelos aos eixos cartesianos posicionados na base, em destaque na Fig. 55. Ele permite o movimento em até dois eixos, formando um movimento composto.

Sistema de Coordenadas Cartesianas

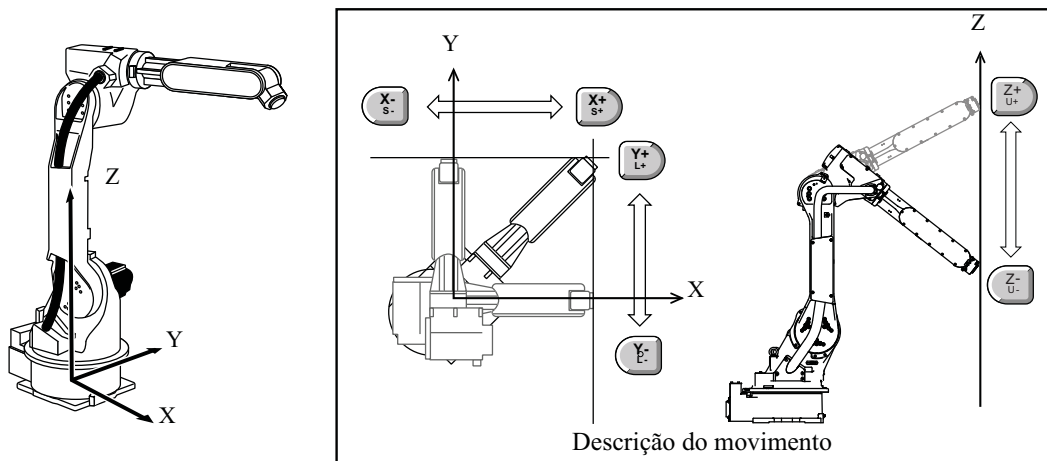


Figura 55 – Sistema de coordenadas cartesianas e a descrição do movimento (adaptado de YASKAWA (2014))

- **Sistema de coordenada das cilíndricas:** o efetuador descreve movimentos radiais e helicoidais em torno do eixo z do sistema posicionado na base, como é evidenciado na Fig. 56.

Sistema de Coordenadas Cilíndricas

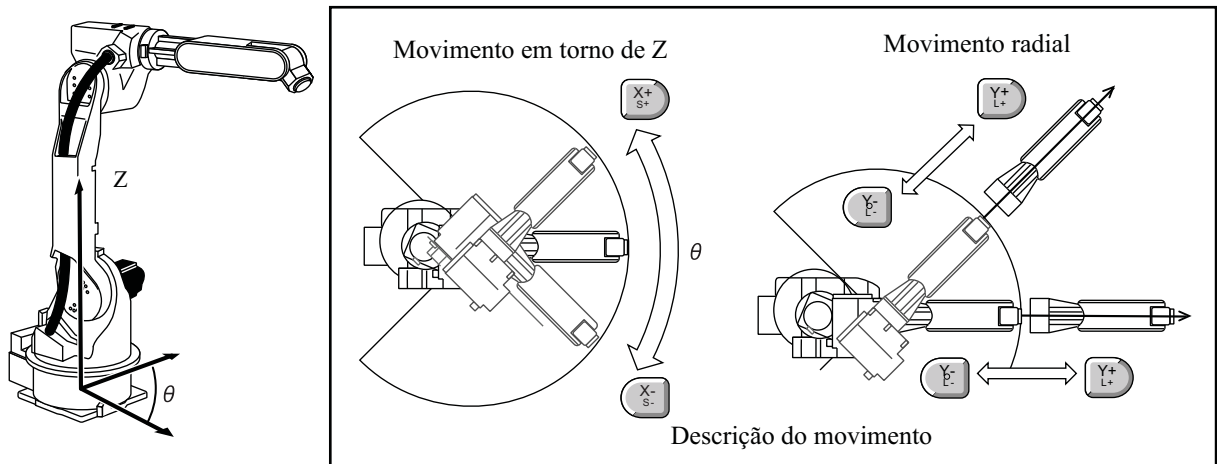


Figura 56 – Sistema de coordenadas cilíndricas e a descrição do movimento (adaptado de YASKAWA (2014))

- **Sistema de coordenadas da ferramenta:** é o sistema definido de acordo com a ferramenta instalada na flange do manipulador, posicionado no TCP. Como pode ser visto na Fig. 57, ele varia de acordo com os movimentos do punho.

Sistema de Coordenadas da Ferramenta

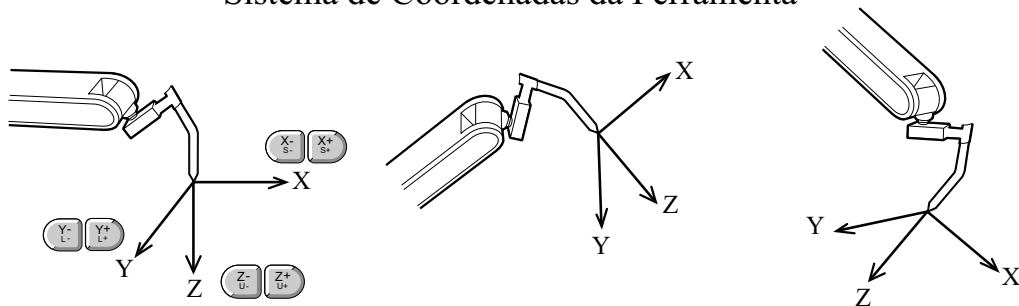


Figura 57 – Sistema de coordenadas da ferramenta (adaptado de YASKAWA (2014))

- **Sistema de coordenadas do usuário:** o efetuator final se move paralelamente a qualquer sistema de coordenadas definido pelo usuário, como mostra a Fig. 58. A utilização de sistemas personalizados pelo usuário facilita a programação em situações como a manipulação do robô através de diferentes acessórios como paletes e esteiras. São disponíveis 24 opções para configurações de diferentes sistemas de coordenadas do usuário.

Sistema de Coordenadas do Usuário

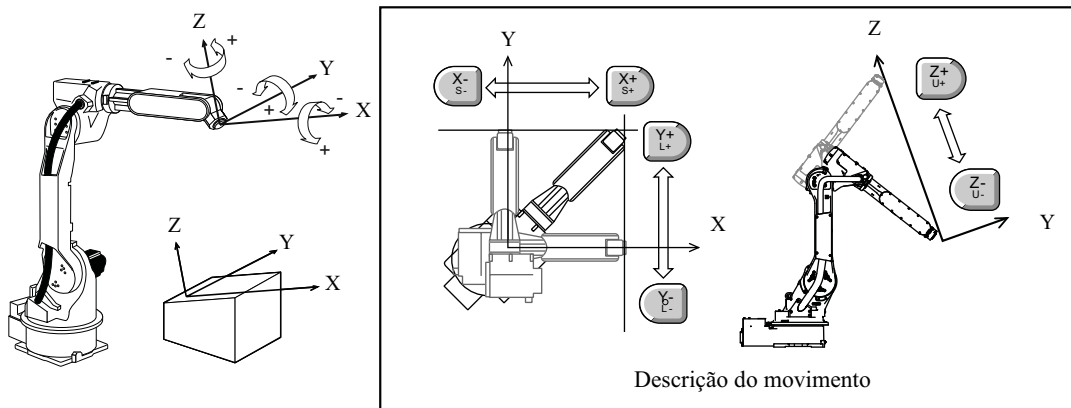


Figura 58 – Sistema de coordenadas do usuário (adaptado de YASKAWA (2014))

Para selecionar o sistema de coordenadas do usuário desejado, basta pressionar a tecla [TOOL SEL COORD], escolhendo a opção "User", e em seguida pressionar simultaneamente as teclas [TOOL SEL COORD] + [SHIFT]. A janela de seleção aparecerá com as opções disponíveis.

Além dos movimentos permitidos nos diferentes sistemas de coordenadas, o robô possui a **Operação por Controle de Ponto** (*Control Point Operation*), que possibilita a mudança apenas da orientação do punho do robô, mantendo a localização do TCP estática, ou seja, determinada a posição do TCP, o usuário pode modificar apenas os ângulos de *roll*, *pitch* e *yaw* do punho. Esse tipo de operação pode ser executado em todos os sistemas de coordenadas, com exceção do sistema de coordenadas das juntas, e permite a composição do movimento em dois eixos simultaneamente. A execução se dá conforme mostrado na Fig. 59.

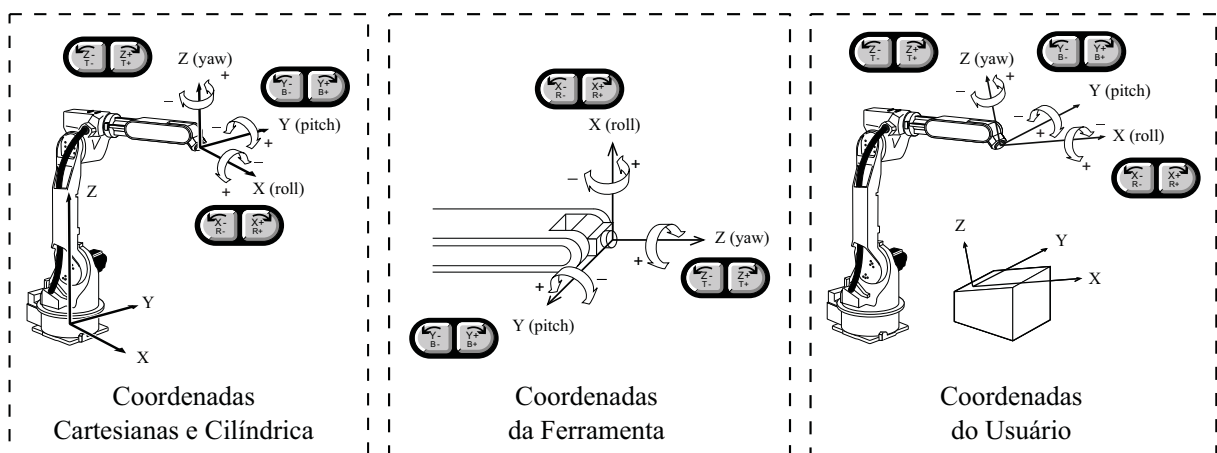


Figura 59 – Operação por Controle de Ponto (adaptado de YASKAWA (2014))

B.0.2 Ensinando Tarefas

Para o ensino de tarefas, se fazem necessárias medidas preliminares de **segurança** e de **inicialização**. Como medida de segurança é importante a verificação do botão [E.STOP] e selecionar a função [TEACH] na chave seletora do modo de operação. Como inicialização, é necessário que seja registrado JOB. Um JOB trata-se de um programa de trabalho que descreve as tarefas que o manipulador executará. Seu registro é realizado conforme se segue:

- **Verificando a Parada de Emergência:** o botão Servo ON no PP deve estar aceso enquanto a alimentação do servomotor estiver ligada. Para certificação do funcionamento dos botões de parada de emergência tanto do FS-100 quanto do PP é necessário adotar os seguintes procedimentos:
 1. Pressionar o botão de parada de emergência no controlador ou no PP.
 2. Confirmar se a alimentação do servomotor está DESLIGADA (o LED no botão [SERVO ON] do PP permanece aceso enquanto a alimentação do servomotor está ligada e apaga quando cessa a alimentação do servomotor).
 3. Depois de confirmar o passo anterior, deve-se pressionar a tecla [SERVO ON READY] no PP e, assim, a alimentação do servomotor é ligada (a alimentação do servomotor pode ser ligada enquanto o botão [SERVO ON] estiver piscando)
- **Selecionando o modo *teach*:** também por segurança é fundamental que sempre se coloque o seletor do modo de operação na posição [TEACH] antes de realizar o ensino de uma tarefa. Assim, o modo de operação fica limitado ao modo *teach* e as máquinas não podem ser operadas por meio do botão [START] ou de uma entrada externa.
- **Registrando um trabalho:** no menu principal (MAIN MENU) do *display* deve ser selecionada a função JOB. No submenu que surge em seguida, selecionando-se o menu CREATE NEW JOB, conforme Fig. 60a., abre a janela “NEW JOB CREATE”, conforme consta na Fig. 60b.

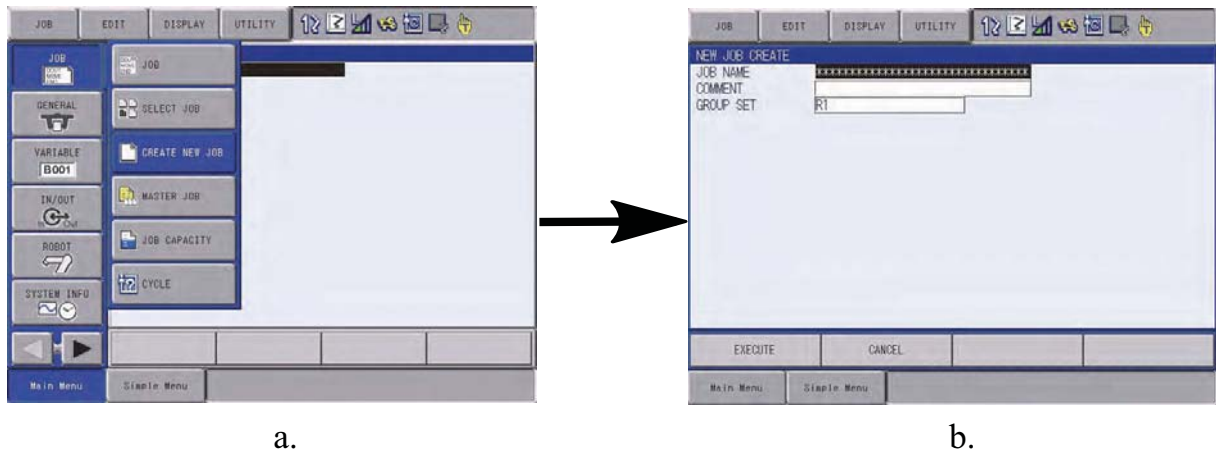


Figura 60 – Janelas de Registro de Trabalho (adaptado de YASKAWA (2014))

Existe a possibilidade de registrar comentários (COMMENT) e grupos de controle (GROUP SET), sendo os últimos necessários para utilização de eixos externos (base ou estação), ou múltiplos sistemas de robô.

Para dar início ao ensino de tarefas propriamente dito é preciso entrar na janela de *teach*. Para isso, basta apertar a tecla [ENTER] ou selecionar “EXECUTE” no *display*, conforme Fig. 61a. para ser direcionado à janela seguinte de ensino de tarefas, que consta na Fig. 61b.

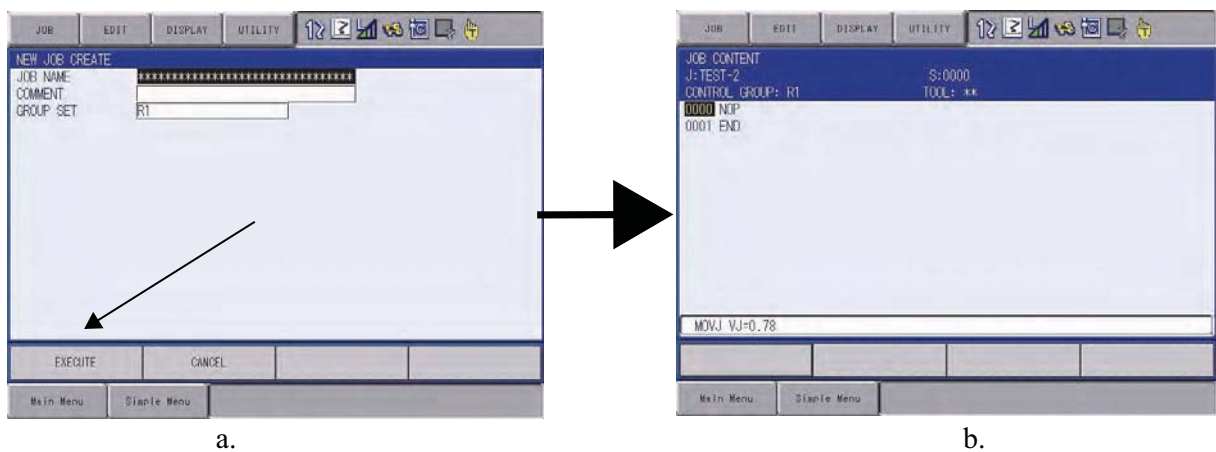


Figura 61 – Alteração para janela de ensino de tarefas (adaptado de YASKAWA (2014))

A atividade de ensino é conduzida na janela “JOB CONTENT”, que tem sua organização conforme apresentado na Fig. 62.

JOB CONTENT	
JOB-A	STEP NO: 0003
CONTROL GROUP: R1	TOOL: 00
0000	NOP
0001	MOVJ VJ=50.00
0002	MOVJ VJ=50.00
0003	MOVL V=1122
0004	TIMER T=5.00
0005	DOUT OT#(1) ON
0006	MOVL V=1122
0007	MOVJ VJ=50.00
0008	END

Diagram annotations:

- a. Número de linha / etapa (points to line numbers 0000-0008)
- b. Cursor (points to line 0003)
- c. Instruções, itens adicionais, comentários, etc. (points to the instruction text area)

Figura 62 – Apresentação da janela “JOB CONTENT” (adaptado de YASKAWA (2014))

Para ensinar tarefas ao manipulador é interessante conhecer a composição dos elementos desta tela, bem como das linhas de comando. Assim, a janela JOB CONTENT, na Fig. 62, é composta por:

- Números de Linha:** Aparecem automaticamente e sua atualização também é automática a medida que as linhas são inseridas ou apagadas;
- Cursor:** Serve para o controle do manipulador. Nas operações FWD, BWD e teste, o movimento do robô inicia-se a partir da linha na qual o cursor se encontra; e,
- Instruções, Itens Adicionais, Comentários, etc:** A velocidade e o tempo são definidos em função do tipo de instrução.

Para reproduzir o movimento do manipulador, a instrução deve estar escrita em um trabalho. Tal instrução é chamada de instrução de movimento e as informações como posição de destino, método de interpolação, velocidade de reprodução, etc. são registradas em uma instrução de movimento. Tais instruções começam com "MOV", na linguagem INFORM III utilizada pelo controlador FS100. **O tipo de interpolação** define a trajetória do manipulador. As principais interpolações são **articular**, **linear** e **circular**, explicadas abaixo:

- **Interpolação Articular** - Instrução: MOVJ

É utilizada quando o robô não precisa desenvolver uma trajetória específica para chegar até a posição da etapa seguinte. **Por segurança, para ensino da primeira etapa utiliza-se a interpolação articular.** Em tal tipo de interpolação,

a velocidade é indicada em **percentual da velocidade máxima**. A definição “0: Speed Omit” determina a velocidade como sendo a última definida na programação.

- **Interpolação Linear** - Instrução: MOVL

O robô movimenta-se segundo uma trajetória linear. É utilizada por exemplo para trabalho de soldagem, como ilustra a Fig. 63.

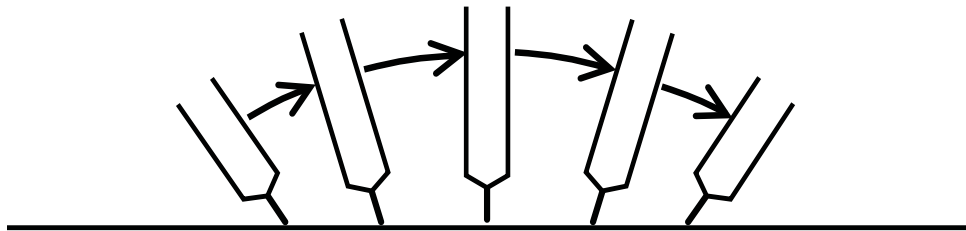


Figura 63 – Esboço da trajetória linear desenvolvida pelo manipulador em trabalho de soldagem (adaptado de YASKAWA (2014))

- **Interpolação Circular** - Instrução: MOVC

O manipulador segue uma trajetória de acordo com um arco que passa por três pontos. Tal interpolação pode ser utilizada de duas formas: por arco circular individual e por arcos circulares contínuos.

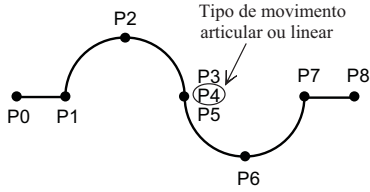
1. *Arco circular individual*: havendo necessidade de apenas um único movimento circular, ensina-se a interpolação circular para três pontos, P1 a P3. Se antes da interpolação circular for ensinada a interpolação linear ou articular, posição P0, o robô move-se em linha reta de P0 a P1, conforme exemplo da Tab. 2:

Tabela 2 – Tipo de interpolação para arco circular individual

	Ponto	Tipo de Interpolação	Instrução
	P0	Articular ou Linear	MOVJ MOVL
	P1 P2 P3	Circular	MOVC
	P4	Articular ou Linear	MOVJ MOVL

2. *Arco circular contínuo*: caso seja necessário executar mais de um movimento circular com curvaturas diferentes, os movimentos devem ser separados um do outro por uma etapa de interpolação articular ou linear. Esta etapa deve ser inserida no mesmo ponto entre as etapas. A etapa no ponto final do movimento deve coincidir com o ponto de partida do movimento circular seguinte, como exemplificado na Tab. 3.

Tabela 3 – Tipo de interpolação para arcos circulares contínuos

	Ponto	Tipo de Interpolação	Instrução
	P0	Articular ou Linear	MOVJ MOVL
	P1 P2 P3	Circular	MOVC
	P4	Articular ou Linear	MOVJ MOVL
	P5 P6 P7	Circular	MOVC
	P8	Articular ou Linear	MOVJ MOVL

A mudança de tipo de interpolação é alterada pressionando [MOTION TYPE] na linha de entrada. A velocidade é alterada movendo o cursor para a velocidade da reprodução e definida pressionando as teclas [SHIFT] + [CURSOR] aumentando ou reduzindo a velocidade.

A Fig. 64 ilustra um exemplo de programação de tarefa, bem como a execução por parte do robô. Nela é possível observar, no *display*, a janela “JOB CONTENT”.

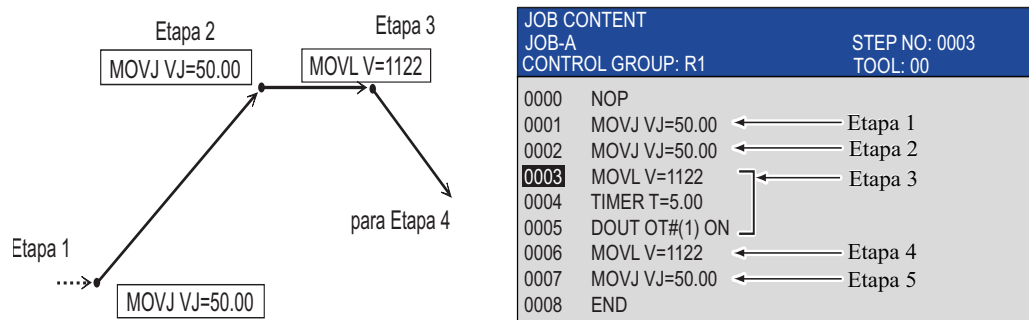


Figura 64 – Exemplo simples de conjunto de tarefas e seus respectivos comandos (adaptado de YASKAWA (2014))

Por outro lado, se faz importante estudar um JOB, ainda simples, mas de maneira mais detalhada para melhor compreensão do manuseio do *Programming Pendant*. Assim, a Fig. 65 apresenta um exemplo mais próximo da realidade de um processo de soldagem.

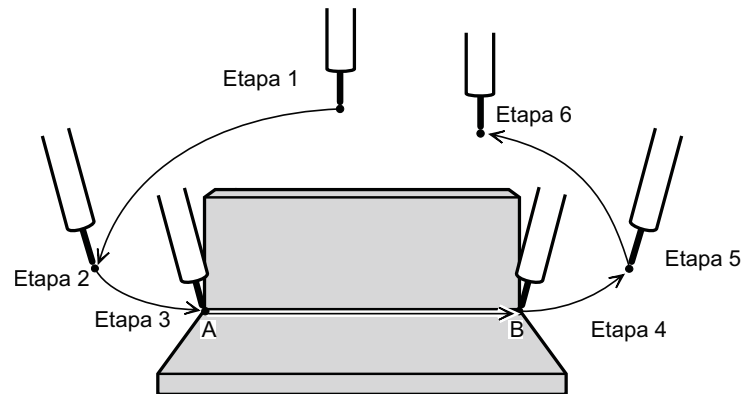


Figura 65 – Exemplo de trabalho com suas etapas (adaptado de YASKAWA (2014))

Nota-se, na Fig. 65, que o trabalho tem 6 etapas, que deverão ser programadas por intermédio do PP. Nesse sentido, a Fig. 66 apresenta os principais passos gerais para a programação (dependendo da etapa poderá haver alguns passos adicionais).

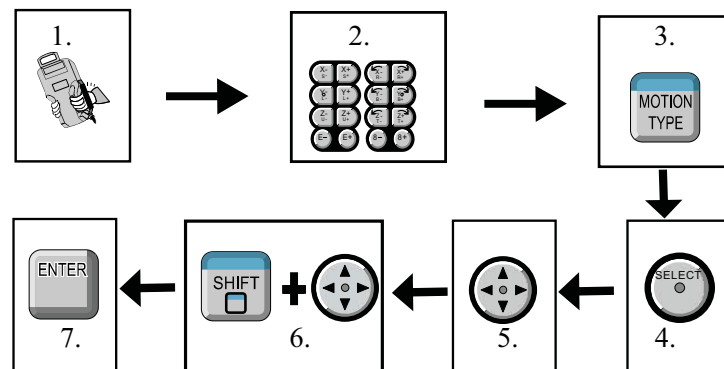


Figura 66 – Passos gerais para programação do trabalho proposto (adaptado de YASKAWA (2014))

Os passos ilustrados na Fig. 66, em linhas gerais consistem em:

1. segurar o [ENABLE SWITCH] para ligar o servomotor;
2. mover o manipulador para a posição desejada usando as teclas de movimento de eixo;
3. determinar o tipo de interpolação a ser desenvolvida usando a tecla [MOTION TYPE], que será indicado na linha de entrada;
4. mover o cursor para a penúltima linha do programa (lembrando que a última tem a instrução: “END”) e pressionar [SELECT] e assim, a linha de entrada aparecerá;
5. na linha de entrada mover o cursor para a direita;

6. selecionar a velocidade pressionando simultaneamente [SHIFT] e o cursor para cima ou para baixo, aumentando ou diminuindo respectivamente, a velocidade até a desejada; e,
7. pressionar [ENTER] para registrar a instrução da Etapa.

Dividindo etapa a etapa, será necessário seguir passos como os listados, para programar cada uma. Assim, chega-se às linhas de código no display do PP, constantes na Fig. 67.

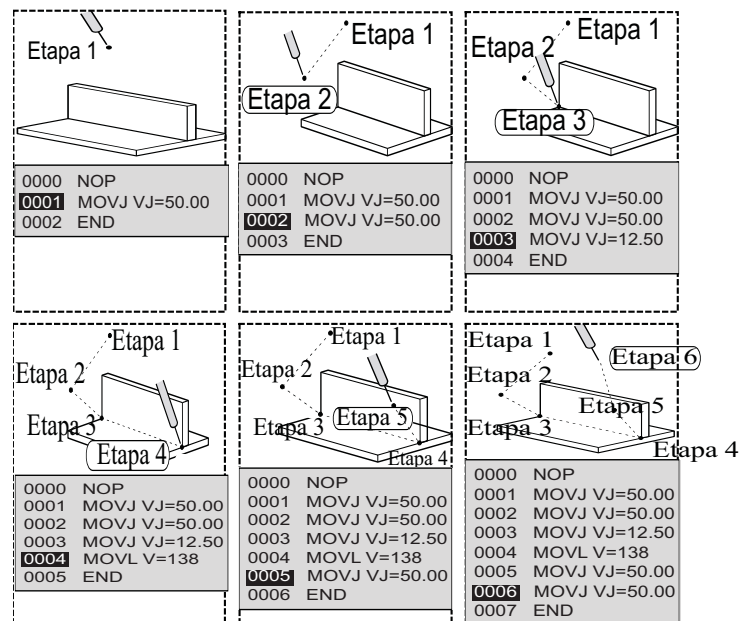


Figura 67 – Conjunto de etapas e a instrução respectiva na janela “JOB CONTENT” (adaptado de YASKAWA (2014))

Para isso, a seguir consta o procedimento mais pormenorizado de cada etapa:

- **Etapa 1:** inicialmente, é importante movimentar o robô para uma posição segura na célula de trabalho, e, no caso do exemplo, deverão ser seguidos os sete passos da Fig. 66;
- **Etapa 2:** posteriormente é preciso definir uma posição próxima da posição de início da tarefa principal. Considerando que o usuário mantém o [ENABLE SWITCH] apertado, dever-se-á seguir os passos 2 e 7 (no caso, devido ao fato de ser programado o mesmo tipo de interpolação e velocidade da Etapa 1);
- **Etapa 3:** neste caso, é recomendável a realização do movimento com a velocidade selecionada em “MÉDIO”, utilizando a tecla [FAST] ou [SLOW], conforme a Fig. 68. Posteriormente, deve-se conduzir a ferramenta do robô até o ponto de início da

tarefa seguindo o passo 2. Com o cursor na linha de número 2, deve-se pressionar [SELECT] e seguir os passos de 3 a 7 com atenção à velocidade, pois como este ponto é um ponto de contato, a velocidade precisa ser menor. Assim, escolhe-se a velocidade de 12,50, ou seja, 12,50% da capacidade do manipulador.

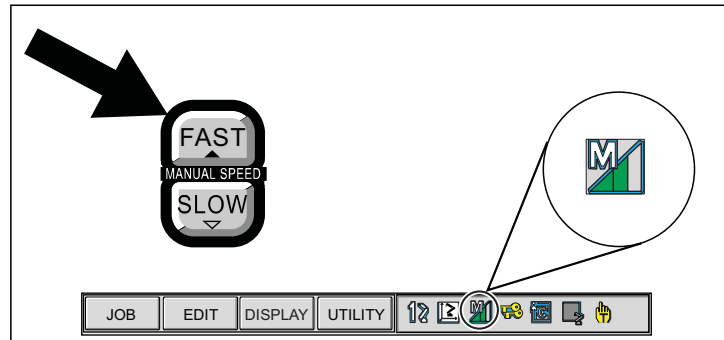


Figura 68 – Alteração de velocidade para “MÉDIO” em operação de ensino (adaptado de YASKAWA (2014))

- **Etapa 4:** em seguida, a próxima posição é a posição final da tarefa, e de acordo com a sua geometria, a ferramenta do robô deve deslocar-se em linha reta da posição da Etapa 3 para a posição da Etapa 4. Para isso, deve-se seguir os passos de 2 a 7 com o cuidado de utilizar a tecla [MOTION TYPE] para escolher MOVL no passo 3, ou seja, a interpolação linear e, por consequência, a velocidade não será VJ e sim V. Optando pelo valor de 138, por exemplo, a ferramenta terá a velocidade de 138 cm/min ;
- **Etapa 5:** após terminar a tarefa principal, precisa-se mover o manipulador para uma posição livre de colisões com a peça de trabalho ou fixações. Para isto, seguem-se os sete passos listados, de forma que no passo 3 seja escolhida a interpolação articular novamente (MOVJ), e a velocidade (VJ) de 50,00%; e,
- **Etapa 6:** na sequência, move-se o manipulador para uma posição próxima à posição inicial, usando as teclas de eixo e registrando conforme a Etapa 2 a mesma interpolação e velocidade que a etapa anterior.

Após cumprir o registro do trabalho, este deve ser validado. Para isso, verifica-se cada etapa separadamente para certificar a não existência de problemas. Assim, precisa-se seguir o passo a passo descrito abaixo e ilustrado na a Fig. 69.

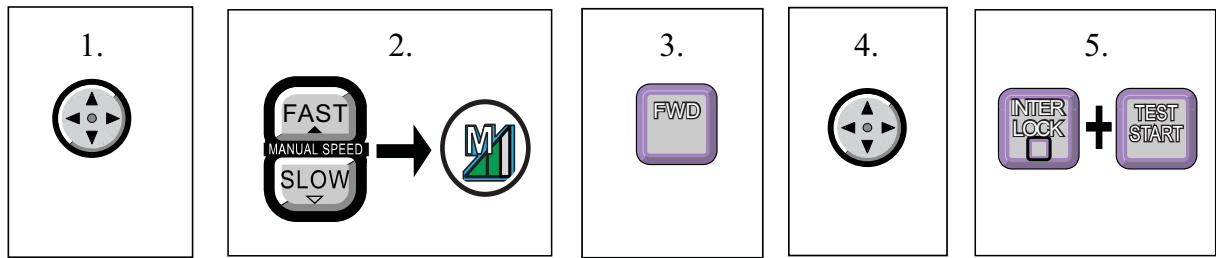


Figura 69 – Passos para validação de trabalho (adaptado de YASKAWA (2014))

1. na janela “JOB CONTENT”, mover o cursor para a Etapa 1 (linha 0001);
2. alterar a velocidade para “MÉDIO” pressionando [FAST] ou [SLOW];
3. pressionar [FWD] para confirmar cada etapa executada pelo manipulador. Cada vez que [FWD] é pressionada o manipulador realiza uma etapa;
4. ao terminar a validação deve-se mover o cursor ao início do trabalho; e,
5. rodar todas as Etapas continuamente. Mantendo pressionada a tecla [INTERLOCK] e pressionando [START] o robô executa todas as etapas continuamente e para quando um ciclo se encerra.

APÊNDICE C – Comandos da linguagem INFORM III

Tabela 4 – Instruções de Movimento (adaptado de YASKAWA (2014))

Instrução	Informações úteis	
MOVJ	Função	Mover para um ponto ensinado com interpolação articular
	Itens adicionais	Dados de posição; VJ; PL; NWAIT; UNTIL; ACC, DCC
	Exemplo	MOVJ VJ=50.00 PL=2 NWAIT UNTIL IN#(16)=ON
MOVL	Função	Mover para um ponto ensinado com interpolação linear
	Itens adicionais	Dados de posição; V, VR, VE; PL; CR; NWAIT; UNTIL; ACC, DCC
	Exemplo	MOVL V=138 PL=0 NWAIT UNTIL IN#(16)=ON
MOVC	Função	Mover para um ponto ensinado com interpolação circular
	Itens adicionais	Dados de posição; V, VR, VE; PL; CR; NWAIT; UNTIL; ACC, DCC
	Exemplo	MOVC V=138 PL=0 NWAIT
IMOV	Função	Mover o incremento especificado da posição atual com interpolação linear
	Itens adicionais	P, BP, EX; V, VR, VE; PL; NWAIT; BF, RF, TF, UF#; UNTIL; ACC, DCC
	Exemplo	IMOV P000 V=138 PL=1 RF
REFP	Função	Definir um ponto de referência
	Itens adicionais	Dados de posição, Número do ponto de referência
	Exemplo	REFP 1
SPEED	Função	Definir a velocidade de reprodução
	Itens adicionais	VJ, V, VR, VE
	Exemplo	SPEED VJ=56.00

Tabela 5 – Instruções de *Inputs* e *Outputs* (adaptado de YASKAWA (2014))

Instrução		Informações úteis
DOUT	Função	Ativar ou desativar os sinais de <i>output</i>
	Itens adicionais	OT#, OGH, OG#
	Exemplo	DOUT OT#(12) ON
PULSE	Função	Emitir um sinal de impulso como sinal de saída externa por um tempo especificado
	Itens adicionais	OT#, OGH#, OG#; T
	Exemplo	PULSE OT# (10) T=0.60
DIN	Função	Definir variáveis para denominar <i>inputs</i> e <i>outputs</i>
	Itens adicionais	OT#, OGH#, OG#; IN#, IGH#, IG#
	Exemplo	DIN B016 IN#(16) DIN B002 IG#(2)
WAIT	Função	Esperar até o estado do sinal de entrada externa corresponder ao estado especificado
	Itens adicionais	OT#, OGH# , OG#; IN#, IGH#, IG#; B; T; "estado"(ON, OFF)
	Exemplo	WAIT IN#(12)=ON T=10.00 WAIT IN#(12)=B002
AOUT	Função	Transmitir a tensão especificada para uma porta de saída analógica
	Itens adicionais	AO#; "tensão de saída"(-14 V a 14 V)
	Exemplo	AOUT AO#(2) 12.7
ARATION	Função	Iniciar a saída analógica correspondente à velocidade
	Itens adicionais	AO#; BV; V; OFV
	Exemplo	ARATION AO#(1) BV=10.00 V=200.0 OFV=2.00
ARATIOF	Função	Terminar a saída analógica correspondente à velocidade
	Itens adicionais	AO#
	Exemplo	ARATIOF AO#(1)

Tabela 6 – Instruções de controle (adaptado de YASKAWA (2014))

Instrução		Informações úteis
JUMP	Função	Saltar para o rótulo ou trabalho especificado
	Itens adicionais	*, JOB, IG#, B, I, D; UF#; IF
	Exemplo	JUMP JOB:TEST1 IF IN#(14)=OFF
* (rótulo)	Função	Indicar o destino de um salto
	Itens adicionais	“linha de destino do salto”
	Exemplo	*432
CALL	Função	Chamar um trabalho especificado
	Itens adicionais	JOB, IG#, B, I, D; UF#; IF
	Exemplo	CALL JOB:TEST1 IF IN# (24)=ON
RET	Função	Retornar ao trabalho original do comando de chamada
	Itens adicionais	IF
	Exemplo	RET IF IN#(12)=OFF
END	Função	Declarar o fim do trabalho
	Itens adicionais	–
	Exemplo	END
NOP	Função	Nenhuma operação
	Itens adicionais	–
	Exemplo	NOP
TIMER	Função	Parada por tempo especificado
	Itens adicionais	T
	Exemplo	TIMER T=12.5
IF	Função	Avaliar a condição especificada (=, <>, <=, >=, <, >) e chegar a uma conclusão correspondente
	Itens adicionais	<Item 1>, <Item 2>
	Exemplo	JUMP *12 IF IN#(12)=OFF
UNTIL	Função	Executar uma ação até que o estado de um sinal especificado seja alcançado
	Itens adicionais	IN#; “estado” (ON, OFF)
	Exemplo	MOVL V=300 UNTIL IN#(10)=ON
PAUSE	Função	Definir uma pausa
	Itens adicionais	IF
	Exemplo	PAUSE IF IN#(12)=OFF
, (comentário)	Função	Marcar o início de um comentário (uma linha)
	Itens adicionais	“comentário”
	Exemplo	,Desenha um quadrado de 100 mm
CWAIT	Função	Esperar pela execução da instrução da próxima linha. Usado juntamente com a instrução NWAIT
	Itens adicionais	–
	Exemplo	MOVL V=100 NWAIT DOUT OT#(1) ON CWAIT DOUT OT#(1) OFF MOVL V=100

Tabela 7 – Instruções de Deslocamento (adaptado de YASKAWA (2014))

Instrução		Informações úteis
SFTON	Função	Iniciar um deslocamento
	Itens adicionais	P, BP, EX, BF, RT, TF, UF#
	Exemplo	SFTON P001 UF#(1)
SFTOF	Função	Parar um deslocamento
	Itens adicionais	–
	Exemplo	SFTOF
MSHIFT	Função	Obter o valor do deslocamento dentre dois pontos (Ponto 2 e 3) e armazená-lo (MSHIFT <Ponto 1> <Coordenada> <Ponto 2> <Ponto 3>)
	Itens adicionais	PX, BP, EX, BF, RT, TF, UF#, MTF
	Exemplo	MSHIFT PX000 RF PX001 PX002

Tabela 8 – Itens adicionais

Tag	Descrição	Variável
NWAIT	Executa as instruções excluindo a instrução de movimentação a partir desse passo, ao mesmo tempo em que o manipulador está realizando esse passo.	–
ENWAIT	Executa as instruções excluindo as instruções de movimentação a partir desse passo, antes de atingir o próximo passo.	–
JOB:“nome”	Especifica um trabalho	–
T=“tempo(s)”	Especifica após quanto tempo a próxima instrução será executada	0–65536

Tabela 9 – Itens adicionais (continuação)

Tag	Descrição	Variável
V="valor"	Especifica a velocidade da ferramenta (TCP)	0,1–1500 mm/s
VJ="valor"	Especifica a velocidade de junta	0.01–100 %
VR="valor"	Especifica a velocidade angular do TCP	0.1–180°/s
VE="valor"	Especifica a velocidade do eixo externo	0.01–100%
AO#("output")	Especifica o número da porta analógica que terá como saída a voltagem correspondente à velocidade	1–40
IN#("input")	Especifica o número do sinal de entrada	1–1024
IGH#("grupo")	Especifica o grupo de entradas (grupos com 4 IN#)	1–256
IG#("grupo")	Especifica o grupo de entradas (grupos com 2 IGH#)	1–128
OT#("output")	Especifica o número do sinal de saída	1–1024
OGH#("grupo")	Especifica o grupo de saídas (grupos com 4 OT#)	1–256
OG#("grupo")	Especifica o grupo de saídas (grupos com 2 OGH#)	1–128
SIN#("input")	Sinal de entrada específico do sistema	1–1280
SOUT#("output")	Sinal de saída específico	1–1600
BV="valor"	Especifica a voltagem de saída para uma velocidade a ser definida (usado na instrução ARATION)	-10–10 V
OFV="valor"	Especifica a voltagem de saída para uma velocidade nula (ver instrução ARATION)	-10–10 V
BF	Deslocamento baseado no sistema de coordenadas da base (grupo de controle externo)	–
RF	Deslocamento baseado no sistema de coordenadas do robô	–
TF	Deslocamento baseado no sistema de coordenadas da ferramenta	–
UF#("user coord")	Deslocamento baseado em um determinado sistema de coordenadas definido pelo usuário	01–16
MTF	Deslocamento baseado no sistema de coordenadas da ferramenta mestre	–

Tabela 10 – Itens adicionais (continuação)

Tag	Descrição	Variável
P“variável” LP“variável”	Especifica a posição (ensinada com relação ao grupo de controle do robô) armazenada na variável	000–127
EX“variável” LEX“variável”	Especifica a posição (ensinada com relação ao grupo de controle externo da estação ou esteira) armazenada na variável	000–127
BP“variável” LBP“variável”	Especifica a posição (ensinada com relação ao grupo de controle externo da base) armazenada na variável	000–127
B“variável” LB“variável”	Variável tipo byte	–
I“variável” LI“variável”	Variável tipo inteiro	–
D“variável” LD“variável”	Variável tipo double precision	–
R“variável” LR“variável”	Variável tipo real	–
S“variável” LS“variável”	Variável tipo carácter	–
ACC=(“valor”)	Especifica a razão de ajuste da aceleração do movimento	20–100 %
DEC=(“valor”)	Especifica a razão de ajuste da desaceleração do movimento	20–100 %
CR	Especifica o raio de curvatura da interpolação circular	1–6553.5 mm
PL	Especifica o nível de posição (nível de aproximação ao passar por um ponto)	0–8
FPT	Especifica o ponto final do arco (ponto onde a curvatura do arco é modificada)	–

Referências

- AURÉLIO, D. Disponível em: <http://www.dicionariodoaurelio.com>. Acesso em, v. 6, 2010. Citado na página 21.
- BOBERG, A. *Virtual lead-through robot programming: Programming virtual robot by demonstration*. 2015. Citado na página 29.
- CONNOLLY, C. Motoman markets co-operative and humanoid industrial robots. *Industrial Robot: An International Journal*, Emerald Group Publishing Limited, v. 36, n. 5, p. 417–420, 2009. Citado na página 32.
- CRAIG, J. J. *Introduction to robotics: mechanics and control*. [S.l.]: Pearson Prentice Hall Upper Saddle River, 2005. v. 3. Citado 2 vezes nas páginas 24 e 25.
- GROOVER, M. P. *Automation, production systems, and computer-integrated manufacturing*. [S.l.]: Prentice Hall Press, 2007. Citado na página 17.
- JAZAR, R. N. *Theory of applied robotics: kinematics, dynamics, and control*. [S.l.]: Springer Science & Business Media, 2010. Citado 3 vezes nas páginas 9, 22 e 33.
- MA, R. R.; DOLLAR, A. M. On dexterity and dexterous manipulation. In: IEEE. *Advanced Robotics (ICAR), 2011 15th International Conference on*. [S.l.], 2011. p. 1–7. Citado na página 22.
- MURRAY, R. M. et al. *A mathematical introduction to robotic manipulation*. [S.l.]: CRC press, 1994. Citado na página 20.
- NOF, S. Y. *Handbook of industrial robotics*. [S.l.]: John Wiley & Sons, 1999. v. 1. Citado 5 vezes nas páginas 9, 26, 27, 28 e 29.
- PAN, Z. et al. Recent progress on programming methods for industrial robots. *Robotics and Computer-Integrated Manufacturing*, Elsevier, v. 28, n. 2, p. 87–94, 2012. Citado 2 vezes nas páginas 18 e 28.
- PEÑA-LÓPEZ, I. et al. World development report 2016: Digital dividends. The World Bank, 2016. Citado na página 17.
- RIBEIRO, L. P. G. *Modelagem cinemática de sistemas robóticos cooperativos: proposta de um Jacobiano de Cooperação*. [S.l.: s.n.], 2010. Citado na página 52.
- ROCHA, C.; TONETTO, C.; DIAS, A. *A comparison between the Denavit–Hartenberg and the screw-based methods used in kinematic modeling of robot manipulators*. [S.l.]: Elsevier, 2011. v. 27. 723–728 p. Citado na página 52.
- SICILIANO, B.; KHATIB, O. *Springer handbook of robotics*. [S.l.]: Springer, 2016. Citado 2 vezes nas páginas 16 e 50.
- SICILIANO, B. et al. *Robotics: modelling, planning and control*. [S.l.]: Springer Science & Business Media, 2010. Citado 13 vezes nas páginas 9, 20, 21, 23, 24, 25, 33, 34, 35, 37, 50, 51 e 56.

SLACK, N.; CHAMBERS, S.; JOHNSTON, R. *Administração da produção*. [S.l.]: Atlas, 2009. Citado na página 17.

SPONG, M. W.; HUTCHINSON, S.; VIDYASAGAR, M. *Robot modeling and control*. [S.l.]: Wiley New York, 2006. v. 3. Citado na página 33.

TSAI, L.-W. *Robot analysis: the mechanics of serial and parallel manipulators*. [S.l.]: John Wiley & Sons, 1999. Citado 6 vezes nas páginas 9, 40, 41, 45, 49 e 51.

YASKAWA. Fs - 100 instructions. 2014. Citado 22 vezes nas páginas 9, 10, 11, 12, 31, 32, 81, 83, 84, 85, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97 e 98.