

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
PROGRAMA DE PÓS-GRADUAÇÃO
DEPARTAMENTO ACADÊMICO DE ELETRÔNICA
ESPECIALIZAÇÃO EM AUTOMAÇÃO INDUSTRIAL

ANA RÚBIA STEFFEN

**UTILIZAÇÃO DE UMA *TOOLBOX* COMO FERRAMENTA DE
SIMULAÇÃO DE UM MANIPULADOR ROBÓTICO.**

MONOGRAFIA - ESPECIALIZAÇÃO

CURITIBA
2011

ANA RÚBIA STEFFEN

**UTILIZAÇÃO DE UMA *TOOLBOX* COMO FERRAMENTA DE
SIMULAÇÃO DE UM MANIPULADOR ROBÓTICO.**

Monografia de conclusão do curso de Pós Graduação em Automação Industrial da Universidade Tecnológica Federal do Paraná apresentada como requisito parcial para obtenção do grau de Especialista em Automação Industrial.

Prof. Dr. Carlos Raimundo Erig Lima

**CURITIBA
2011**



Ministério da Educação
Universidade Tecnológica Federal do Paraná
Diretoria do *Campus* Curitiba
Gerência de Pesquisa e Pós-Graduação
Departamento Acadêmico de Eletrônica

TERMO DE APROVAÇÃO

UTILIZAÇÃO DE UMA *TOOLBOX* COMO FERRAMENTA DE SIMULAÇÃO DE UM MANIPULADOR ROBÓTICO.

Área de conhecimento: Automação Industrial

por

Ana Rúbia Steffen

A presente monografia, requisito parcial para obtenção do título de ESPECIALISTA EM AUTOMAÇÃO INDUSTRIAL, foi avaliada pela banca examinadora, constituída pelos docentes abaixo discriminados, que considerou o trabalho **Aprovado**.

Prof. Dr. Jean Marcelo Simão

Prof. Msc. Alceu Schneider

Prof. Dr. Carlos Raimundo Erig Lima
Orientador

Curitiba, 26 de Março de 2011.

Visto da coordenação

Prof. Dr. Jean Marcelo Simão

A Folha de Aprovação assinada encontra-se na Coordenação do Curso.

AGRADECIMENTOS

A realização deste trabalho somente foi possível devido à motivação que consegui gerar pelo incentivo de Deus, minha vida, meu sorriso, meu melhor amigo, minha fortaleza na alegria e na tristeza.

Meus mais fortes agradecimentos vão para os meus pais: Janice e Erci, bem como minha irmã Renata responsáveis pela minha educação e pela minha formação. Agradeço ao meu orientador, Prof. Dr. Carlos Raimundo Erig Lima, ao Prof. Dr. Jean Marcelo Simão, bem como aos: Prof. Ms. Guilherme Alceu Schneider, Prof. Dra. Simone Mussolini e Prof. Dra. Faimara do Rocio Strauhs que foram extremamente gentis e atenciosos comigo, além de sempre se colocarem à disposição para me auxiliar, desde a escolha do tema até a conclusão do trabalho.

Agradeço a todos os amigos e colegas da Pós Graduação que contribuem para meu aprendizado, além de umas boas gargalhadas e umas calorias a mais (Aline). Tenho certeza que sairemos deste curso muito mais amigos que apenas colegas de profissão.

“Só na imaginação dos homens é que cada verdade encontra uma existência efetiva e inegável. A imaginação, não a invenção é a suprema mestra da arte, tanto quanto da vida” (Robbins, 1993 apud).

RESUMO

STEFFEN, Ana Rúbia. Estudo da **Utilização de uma *toolbox* como ferramenta de simulação de um manipulador robótico**. 2011. 100 p. Monografia (Especialização em Automação Industrial) - Programa de Pós-Graduação em Automação Industrial, CEFET, Curitiba.

Este trabalho tem como objetivo realizar um breve estudo sobre o quão pertinente seria ampliar a utilização de um aplicativo computacional, amplamente difundido no âmbito acadêmico, no industrial. O *software* em questão é o MATLAB e a *toolbox* utilizada como ferramenta de simulação é um pacote de funções, especialmente criado para programação e controle de manipuladores robóticos. Por se tratar de um programa que dispõe uma versão especial (e gratuita) para estudantes, é possível criar novos aplicativos e torná-los públicos. Para efetuar este estudo, foi realizada uma simulação do robô de Stanford. Como resultado, constatou-se que o aplicativo robótico é pertinente em casos onde seja necessária análise dos parâmetros de controle das juntas do robô, (como por exemplo, torque, aceleração e velocidade), desde que não comprometa a produção da empresa. É uma ferramenta muito valiosa, porém sua utilização na indústria depende da tarefa desempenhada pelo manipulador dentro do processo. É um aplicativo que demanda tempo para simular e analisar, além de necessitar uma mão-de-obra bastante dedicada. Dependendo de quem for operar o robô e utilizar o aplicativo podem ocorrer demora no processo e causar problemas à empresa, especialmente com relação à prazo de entrega.

Palavras-Chaves: Robótica. *Toolbox*. Manipuladores Robóticos. MATLAB.

ABSTRACT

STEFFEN, Ana Rúbia. **Utilização de uma *toolbox* como ferramenta de simulação de um manipulador robótico**. 2011. 100 p. Monografia (Especialização em Automação Industrial) - Programa de Pós-Graduação em Automação Industrial, CEFET, Curitiba.

This paper aims to conduct a brief study about how relevant is to expand the use of a computer application, commonly used in academia would be relevant in an industrial environment. The software here described is MATLAB and the toolbox used as a simulation tool is a package of functions, specially created for programming and control of robotic manipulators. Because it is a program that offers a special (and free) version for students, it is possible to create new applications and make them public. To carry out this study, it was performed a simulation of the Stanford robot. As a result, it was found that the robotic application is relevant in cases where it is needed analysis of the control parameters of the joints of the robot, (such as torque, acceleration and speed), since it does not compromise the company's production. It is a very valuable tool, but its use in industry depends on the task performed by the handler within the process. It is an application that requires time to simulate and analyze, and require a skilled workforce dedicated enough. Depending on who is operating the robot and use the application may experience delays in the process and cause problems especially in relation to the company will delivery time.

Key Words: Robotic. Toolbox. Robotic Manipulators. MATLAB.

LISTA DE EQUAÇÕES

Equação 1 – Matriz homogênea T.....	32
Equação 2 – Matriz homogênea resultante.....	33
Equação 3 – Funções: posição, velocidade e aceleração das articulações.....	37
Equação 4 – Equação do movimento de Lagrange.....	39
Equação 5 – PID.....	61

LISTA DE FIGURAS

Figura 1 – Robô UNIMATE.....	9
Figura 2 – Robô PUMA.....	10
Figura 3 – Robô SCARA.....	10
Figura 4 – Anatomia de um robô.....	17
Figura 5 – Sequência de elos e juntas de um braço robótico.....	18
Figura 6 – Tipos de juntas utilizadas em robôs.....	19
Figura 7 – Garra de dois dedos.....	20
Figura 8 – Garra para objetos cilíndricos.....	20
Figura 9 – Garras articuladas.....	21
Figura 10 – Métodos de acionamento.....	22
Figura 11 – Controle de movimentos de um robô.....	25
Figura 12 – Modelo de <i>teach-pendant</i> da UNIMATION.....	26
Figura 13 – Relação não-unívoca entre as cinemáticas direta e inversa.....	31
Figura 14 – Notação de Denavit-Hartenberg.....	32
Figura 15 – Sistema de controle em malha fechada.....	36
Figura 16 – Diagrama de blocos para o controle do torque.....	42
Figura 17 – Braço de Stanford.....	44
Figura 18 – Referenciais geométricos do robô Stanford.....	45
Figura 19 – Braço de Stanford gerado pelo MATLAB em posição inicial.....	47
Figura 20 – Resultado da cinemática direta para as coordenadas dadas por q	49
Figura 21 – Simulação da cinemática para outro vetor de coordenadas angulares.....	49
Figura 22 – Simulação da cinemática inversa.....	50
Figura 23 – Exemplo de trajetória percorrida por um manipulador.....	50
Figura 24 – Modelo de um braço simplificado por apenas 2 juntas rotativas.....	56
Figura 25 – Pêndulo duplo utilizado para equacionar um robô.....	57
Figura 26 – Sistema em malha aberta da junta 3.....	57
Figura 27 – Dinâmica da junta 3.....	59
Figura 28 – Diagrama de blocos de um robô com um compensador PID.....	61
Figura 29 – Diagrama de blocos da junta 3 em malha fechada com o PID.....	63

LISTA DE GRÁFICOS

Gráfico 1 – Deslocamentos das juntas.....	53
Gráfico 2 – Velocidade das juntas.....	53
Gráfico 3 – Aceleração das juntas.....	53
Gráfico 4 – Torque em função do tempo.....	55
Gráfico 5 – Torque influenciado pelas forças centrípeta e de coriolis.....	55
Gráfico 6 – Simulação dinâmica da junta 3.....	57
Gráfico 7 – Deslocamento e velocidade da junta 3 respectivamente.....	59
Gráfico 8 – Aceleração da junta 3.....	59
Gráfico 9 – Dinâmica da junta 3 sem e com o PID, respectivamente.....	63

LISTA DE TABELAS

Tabela 1 – Principais aplicações robóticas nas indústrias nacionais.....	23
Tabela 2 – Exemplos de algumas linguagens de programação para robôs.....	28
Tabela 3 – Valores dos parâmetros de simulação e controle para o braço de Scheinman.....	46
Tabela 4 – Regra de Ziegler-Nichols.....	62

SUMÁRIO

AGRADECIMENTOS	4
RESUMO	6
ABSTRACT	7
SUMÁRIO	12
1 INTRODUÇÃO	8
1.1 TEMA	8
1.1.1 Delimitação do tema	11
1.2 PROBLEMA E PREMISSAS	11
1.3 OBJETIVOS	13
1.3.1 Objetivo geral	13
1.3.2 Objetivo específico	13
1.4 JUSTIFICATIVA	13
1.5 PROCEDIMENTOS METODOLÓGICOS	14
1.6 ESTRUTURA DO TRABALHO	14
2 ROBÓTICA INDUSTRIAL	16
2.1 PRINCIPAIS COMPONENTES DOS ROBÔS	17
2.1.1 Estrutura Mecânica	17
2.1.2 Sensores	21
2.1.3 Acionamentos	21
2.2 ROBÓTICA NO BRASIL	22
2.3 PRINCIPAIS DESAFIOS EM ROBÓTICA	23
2.4 CONSIDERAÇÕES FINAIS DO CAPÍTULO	24
3 PROGRAMAÇÃO E CONTROLE DE ROBÔS	25
3.1 PROGRAMAÇÃO ONLINE	26
3.1.1 Programação por aprendizagem	27
3.1.2 Programação por condução	28
3.2 PROGRAMAÇÃO OFFLINE	28
3.3 A CINEMÁTICA DOS ROBÔS	30
3.3.1 Cinemática Direta	31
3.3.2 Cinemática Inversa	33
3.3.3 Sistemas de Controle de Robôs	34
3.4 DINÂMICA DOS MANIPULADORES ROBÓTICOS	37
3.5 CONTROLE DA TRAJETÓRIA DO ROBÔ	38
3.6 CONSIDERAÇÕES FINAIS DO CAPÍTULO	39
4 A FERRAMENTA DE ROBÓTICA DO MATLAB E UM CASO DE ESTUDO: CONTROLE E SIMULAÇÃO DE UM ROBÔ REAL	41
4.1 A BIBLIOTECA DE ROBÓTICA DO MATLAB	41
4.2 CASO DE ESTUDO: CONTROLE E SIMULAÇÃO DE UM ROBÔ REAL	43
4.3 MODELAGEM DO ROBÔ DE SCHEINMAN	46
4.3.1 Simulação - Cinemática Direta	47

4.3.2 Simulação - Cinemática Inversa	50
4.3.3 A Matriz Jacobiana	51
4.3.4 Simulação da Trajetória	53
4.3.5 Simulação do Modelo Dinâmico.....	54
4.4 SIMULAÇÃO DINÂMICA EM SIMUNLINK	56
4.5 IMPLEMENTAÇÃO DO SISTEMA DE CONTROLE DE UMA JUNTA	58
4.5.1 Implementação de um compensador a partir da dinâmica da junta.....	59
4.5.2 Controlador PID	60
4.5.3 Sintonia do controlador.	62
4.6 CONSIDERAÇÕES FINAIS DO CAPÍTULO	64
5 CONSIDERAÇÕES FINAIS.....	66
5.1 SUGESTÕES DE TRABALHOS FUTUROS	67
REFERÊNCIAS.....	68
ANEXOS	72

1 INTRODUÇÃO

1.1 TEMA

A maioria das pessoas, ao ouvir a palavra robô, relaciona-os diretamente ao andróide ou humanóide dos filmes de ficção científica, sendo sua história repleta de fantasias e mitos ingenuamente criados pela indústria cinematográfica, como os robôs que se tornam parte da família; ou ainda, escravizam as pessoas e dominam a terra. A realidade, porém, é totalmente diferente (STONE; WESLEY, 2005).

A finalidade desses elementos na indústria, por exemplo, é melhorar a capacidade de produção das empresas pela automatização das tarefas realizadas pelas pessoas. Neste sentido a *Robot Institute of América* (Instituto de Robótica da América) designa um robô como "um manipulador multifuncional reprogramável, projetado para mover materiais, peças, ferramentas ou dispositivos especializados através de programas para o desempenho de uma variedade de tarefas" (STONE; WESLEY, 2005).

Algumas tarefas realizadas na indústria implicam velocidade e precisão em sua execução, bem como repetição de movimento. Outras podem apresentar insalubridade. Os robôs são introduzidos na indústria para executar esse tipo de tarefa e preservar a integridade humana (DE SOUZA; DINIZ; BAPTISTA, 2006).

Com o término da segunda guerra mundial, em 1945, os Estados Unidos experimentaram um forte aquecimento econômico impulsionado pelo crescimento da indústria bélica (especialmente na Guerra Fria – 1945 a 1991) e por um rápido avanço em sua tecnologia. Em 1961 a GM (*General Motors*) criou o robô industrial UNIMATE (conforme pode ser visto na figura 1) e instalou-o em sua linha de montagem, dando origem a robótica. A empresa o criou para executar a tarefa de descarga de peças em altas temperaturas de uma máquina de fundição (SICILIANO; KHATIB, 2008).

Três anos depois foram criados e utilizados os primeiros robôs hidráulicos com 5 ou 6 graus de liberdade em uma fábrica na Noruega. Este robô foi utilizado para pintar peças tendo se destacado como o primeiro a utilizar o conceito de coordenadas polares e movimento com trajeto contínuo. Na década seguinte, um

agricultor britânico modificou um robô pulverizador para aplicá-lo em soldagem do tipo arco voltaico (SICILIANO; KHATIB, 2008).



Figura 1-Robô UNIMATE.
Fonte: Clube Robótica, (2010).

Em meados da década de 70, as fábricas japonesas buscavam aperfeiçoar os seus processos de manufatura, unindo corte de custos e melhoria da eficiência. O primeiro robô desenvolvido naquele país foi um *Unimate* em 1972 pela Kawasaki. O sucesso foi tal que a empresa comprou o projeto da *Unimation* e efetuou uma série de melhorias em suas funcionalidades, sendo criado um robô arco-de-solda para fabricar seus chassis de motocicletas (SANTOS, 2004).

Ainda nesta década a *Unimation* cria as duas primeiras linguagens de programação para estes manipuladores robóticos: WAVE em 1973 e AL em 1974, com ambas dando lugar a linguagem VAL alguns anos mais tarde. A partir desses algoritmos foi possível realizar tarefas de atuadores (elétricos, pneumáticos e sonoros), bem como mover um braço mecânico, abrir e fechar uma garra deslocando-se em certos tipos de superfícies em determinadas trajetórias (SANTOS, 2004).

Em 1974 a empresa criadora do primeiro robô UNIMATE desenvolveu um segundo manipulador chamado PUMA - *Programmable Universal Machine for Assembly* (máquina universal programável para montagem), sendo colocado no mercado em 1978. O mesmo é apresentado na figura 2, sendo um dos mais utilizados na indústria ainda nos dias de hoje (FORTUNECITY, 2010).



Figura 2-Robô PUMA.
Fonte: Theatronics, (2010).

No ano de 1979 foi construído pela Universidade de Yamanashi, no Japão, um dos robôs com maior aplicação industrial: *Selective Compliance Assembly Robot Arm* - SCARA (Braço Robótico para Montagem com Flexibilidade Seletiva). Como observado na figura 3, este se destaca por ser um robô compacto, com grande precisão e repetibilidade, embora com um alcance limitado. Estas características o tornam próprios para trabalhos em montagem mecânica ou eletrônica que exigem alta precisão e velocidade (CARRARA, 2009).

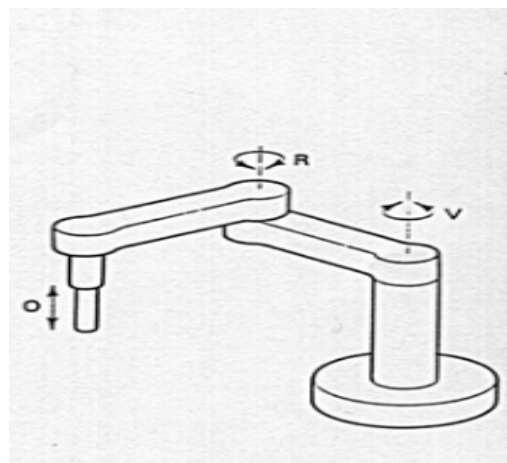


Figura 3-Robô SCARA.
Fonte: CARRARA, (2009).

Atualmente a maioria dos robôs industriais trabalha em linhas de montagem de automóveis, pois são mais precisos e podem fazer este trabalho de forma mais eficaz que os homens, já que sua programação (e, portanto os movimentos

efetuados) permanece a igual, mesmo após consecutivos turnos de trabalho. Outra importante aplicação é na indústria da informática, especialmente em trabalhos minuciosos, como na montagem de *chips* de computador (HARRIS, 2007).

O presente documento pretende abordar os tipos de robôs mais utilizados na indústria e seus respectivos princípios de funcionamentos e aplicações. Subsequentemente será apresentado à ferramenta especialmente construída para simulação e controle de manipuladores robóticos do MATLAB para, enfim, efetuar um caso de estudo. O objetivo é verificar se a ferramenta desenvolvida este tipo de simulação poderia ser estendida às empresas, evitando riscos e custos em relação a utilização de robôs reais.

1.1.1 Delimitação do tema

Por mais vantajoso que possa parecer uma linha de produção formada em sua maioria por robôs, algumas variáveis são fundamentais para as empresas optarem ou não por esta mão-de-obra. Quando um fabricante lança um novo manipulador no mercado, basicamente este será operado por um *software* restrito ao tipo de robô utilizado pelo cliente. Como é possível ocorrer falhas em programação, sendo assim necessário parar o programa para analisar e corrigir as falhas, isto significa produção interrompida. Produção interrompida significa não entrada de lucro ou ainda prejuízo, dependendo de quanto tempo a empresa deixou de produzir.

Ao adotar um *software* “popularizado”, a troca de informações entre os usuários auxilia a otimização de seu uso, pois ao tornar público o código-fonte maior número de programadores têm acesso proporcionando a criação de novas funcionalidades. Como consequência pode ocorrer a diminuição do tempo não-produtivo apenas por diminuir o tempo em que o programa necessita reparo.

1.2 PROBLEMA E PREMISSAS

A estrutura de uma linha de produção está segmentada no fato de que a maioria das tarefas é repetitiva e assim perde-se um considerável tempo para programá-las e reprogramá-las. Logo surgia aí um problema: *como replicar certa*

tarefa para outra máquina e assim aperfeiçoar tempo e recursos empregados na tarefa? Quem tomou consciência desta situação inicialmente foi Sir. Charles Devol, responsável por desenvolver uma forma de registrar uma seqüência de movimentos á serem seguidas pelos robôs (reprogramabilidade) (GROOVER, 1987).

Um dos pontos chaves deste setor é programar o robô para uma determinada tarefa (seqüência específica de comandos) e aplicá-los a linha de montagem. Entretanto está sujeito a erros e como consequência disto pode ocorrer desde simples falhas nos produtos a grandes prejuízos por um ‘efeito dominó’ oriundo da propagação destes erros. Logo para evitar situações como esta e consequentemente prejuízos às fábricas, estas investem quantias substanciais em *softwares* de simulação. Em um programa de simulação é possível criar ambientes virtuais para testar a seqüência de comandos programados para os robôs, para somente depois aplicá-los no robô real.

Uma grande parte das universidades de engenharia de todo o mundo utilizam como *software* de simulação e análise de controle de sistemas o MATLAB. Dentro deste programa existem os chamados *toolboxes* (ferramentas) que são funções diferentes das básicas, originalmente presentes no programa. Estas funções estão disponíveis em bibliotecas externas, podendo ser constantemente atualizadas à medida que forem desenvolvidas novas aplicações. Neste software há um *toolbox* especialmente destinado á simulação de manipuladores robóticos, chamada *Robotics Toolbox for Matlab* (ferramentas robóticas para o Matlab).

Esta ferramenta fornece funções em que é abordada uma série de questões relativas aos robôs, como por exemplo, cinemática, dinâmica e geração de trajetória. Esta ferramenta é extremamente útil para simulação e análise de resultados a partir de experimentos virtuais, baseados em modelos de manipuladores reais.

Muitas indústrias têm implantado o programa em sua linha de produção (MATHWORKS, 2011). Nesta abordagem, vem a questão condutora deste trabalho: **é pertinente utilizar esta ferramenta, tão difundida no âmbito acadêmico, no setor industrial?**

Acredita-se que a partir da análise da literatura técnica existente e um caso de estudo de simulação de um manipulador será possível obter uma direção para este questionamento.

1.3 OBJETIVOS

1.3.1 Objetivo geral

Realizar um breve estudo sobre os princípios de funcionamento e controle dos manipuladores robóticos e após isto, apresentar a *toolbox* de robótica para o mencionado *software*.

1.3.2 Objetivo específico

- Identificar um manipulador robótico: princípios de funcionamento, principais componentes e tipos de controle;
- Analisar o *software* mencionado neste documento e a ferramenta descrita;
- Efetuar um caso de estudo, simulando a dinâmica de um robô industrial no MATLAB;
- Apresentar as considerações finais e idéias de trabalhos futuros.

1.4 JUSTIFICATIVA

Dentre as principais atribuições de um engenheiro, destaca-se a busca por reduzir e impedir possíveis falhas em processos produtivos, bem como a busca por novas tecnologias. Porém, tudo isto tem um preço e quanto menos onerosas forem estas soluções para as empresas, mais recursos poderão para ser investidos em outros setores dela, como qualificação de mão-de-obra ou investimento em novas máquinas. Se for possível otimizar a utilização de um *software*, ou seja, se for possível ter acesso a diferentes funcionalidades para um mesmo programa (gastos com licença e anuidade seriam reduzidos), maior será a economia da empresa.

Obter êxito neste trabalho significará um primeiro passo no sentido de gerar economia para as indústrias de manufatura, já que um único *software* funcionará como ferramenta de análise e simulação, sem restringir sua utilização a um determinado manipulador apenas.

1.5 PROCEDIMENTOS METODOLÓGICOS

Esta monografia será uma pesquisa científica aplicada, explicativa, descritiva e bibliográfica.

O documento será composto de três fases: (i) pesquisa baseada em referenciais teóricos; (ii) análise e simulação de um caso de estudo e (iii) consideração a cerca dos resultados.

1.6 ESTRUTURA DO TRABALHO

Este trabalho compõe-se de 5 (cinco) partes, com 5 (cinco) capítulos, sendo;

- Parte 1 – Capítulo introdutório.
- Parte 2 – Fundamentos teóricos: Capítulos 2, 3 e 4.
- Parte 3 – Caso de estudo: Capítulo 4.
- Parte 4 – Análises e conclusões: Capítulo 5.
- Parte 5 – Referências.

O Capítulo 1, introdutório deste trabalho, estabelece seu tema central, definindo o problema e premissas, o objetivo proposto, a justificativa de execução do mesmo e a metodologia de pesquisa a ser adotada.

O Capítulo 2 inicia os fundamentos teóricos com os conceitos gerais da robótica como ciência, como é composto um manipulador robótico, os tipos de controle que podem ser efetuados e uma descrição quanto aos principais componentes do manipulador.

O Capítulo 3 aborda algumas operações matemáticas que realizam o controle de trajetória dos robôs, as cinemáticas: direta e indireta; bem como apresenta as principais características da programação: *online* e *offline*.

O Capítulo 4 apresenta o *software* Matlab e a *toolbox* utilizado na simulação de robôs, e o caso de estudo de simulação e análise de um robô.

As considerações finais do trabalho são apresentadas no Capítulo 5, seguido das referências bibliográficas e anexos utilizadas para a fundamentação teórica do mesmo.

2 ROBÓTICA INDUSTRIAL

A robótica deixou de ser apenas ficcional e foi elevada ao nível de ciência quando, em 1950 o bioquímico, escritor e divulgador científico Isaak Judah Azimov definiu as três leis da robótica:

1. Um robô não pode maltratar um ser humano, ou pela sua passividade, deixar que este seja maltratado.
2. Deve obedecer às ordens dadas por um ser humano exceto quando entrarem em conflito com a primeira lei.
3. Um robô deve proteger a sua própria existência desde que esta proteção não entre em conflito com a primeira e segunda lei (LEITÃO, 2004).

Atualmente a robótica é um ramo da ciência que engloba as tecnologias: mecânica, eletrônica e computacional, envolvendo: projeto, construção, programação e controle de robôs. Além disto, participam em menor ou maior escala a microeletrônica, inteligência artificial e engenharia de manufatura (LEITÃO, 2004).

Dentre as principais atividades desempenhadas pelos robôs na indústria, destacam-se: a soldagem por arco elétrico, manipulação de peças carga e descarga de máquinas e pintura – especialmente na indústria automotiva (LEITÃO, 2004).

2.1 MANIPULADOR ROBÓTICO

Um manipulador robótico é um dispositivo mecânico fixo a uma base, controlado por um algoritmo computacional que permite através de sensores e atuadores interagir com o meio. Isto ocorre por um sistema composto por *software* e *hardware*, onde este sistema processa os sinais de entrada e converte-os em uma ação programada por meio de um algoritmo (DE PIERI, 2002).

- Sensores: captam informações do meio e são utilizados como componentes do sistema de realimentação do robô. Estes sinais são utilizados para detectar e corrigir os desvios em sistemas de controle.

- Controle digital: através das informações obtidas pelo sensor, eleger as ações a serem executadas para que o erro (diferença entre o que foi medido e o desejado) seja nulo.
- Manipulador robótico (robô): acionamento (atuação) mecânico da ação de controle. É o manuseio de peças ou ferramentas para atingir a saída desejada, conforme o que foi programado.

2.1 PRINCIPAIS COMPONENTES DOS ROBÔS

A anatomia dos manipuladores robóticos é formada basicamente por: estrutura mecânica (antebraço, pulso, junta, braço e base), atuadores (elétricos, hidráulicos ou pneumáticos), sistema de controle, ferramentas (projetadas pelo fabricante conforme as necessidades, servindo para manipular objetos) e *teach pendant* (painel de acionamento de controle) (FERREIRA, 2005).

2.1.1 Estrutura Mecânica

Basicamente a estrutura mecânica de um manipulador robótico é formada por: antebraço, pulso, junta, braço e base conforme apresentado na figura 4 (ROMANO, 2009).

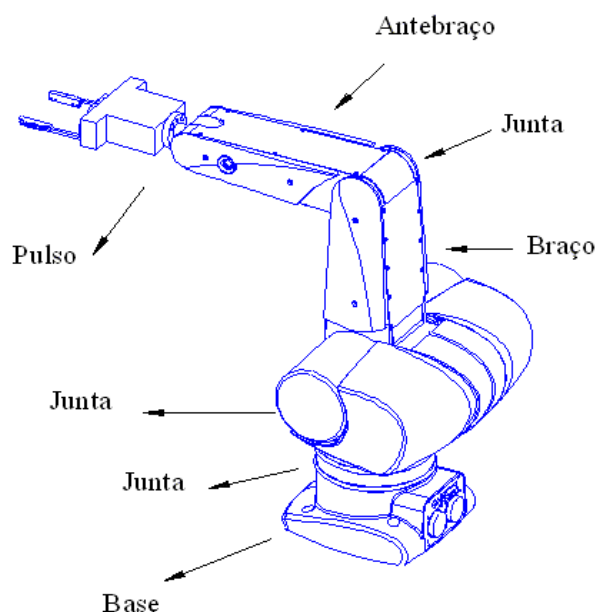


Figura 4: Anatomia de um robô
Fonte: CARRARA, 2009.

2.1.1.1 Base

A base oferece suporte ao manipulador e pode ser móvel ou fixa, dependendo das necessidades do usuário. A maioria dos robôs utilizados na indústria (cerca de 90% em todo o mundo), são robôs de base fixa, chamados manipuladores. Eles operam dentro de um espaço limitado e não podem movimentar-se (DE PIERI, 2002).

2.1.1.2 Braço

Um braço robótico é formado apenas por braço e pulso (também chamado de punho). Conforme ilustrado na figura 5, o braço é constituído por elos que são unidos por juntas de movimento relativo, onde estarão acoplados os acionadores. Através das informações obtidas pelos sensores e instruídas pelo sistema de controle, o movimento será efetuado (CARRARA, 2006).

O pulso é constituído por juntas próximas entre si, que permitem a orientação do órgão terminal (ferramenta, que varia de acordo com a funcionalidade do robô) nas posições correspondentes à tarefa a ser realizada (CARRARA, 2006).

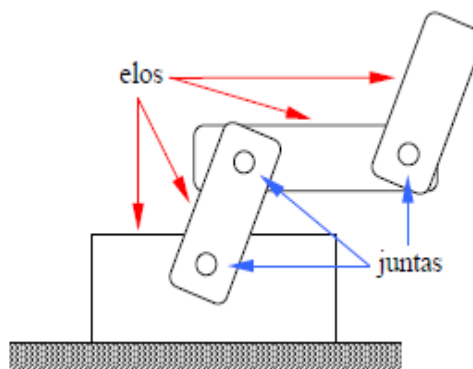


Figura 5: Seqüência de elos e juntas em um braço robótico.
Fonte: CARRARA, 2006.

2.1.1.3 Juntas

As juntas têm a função de orientar o órgão terminal nas posições correspondentes a tarefa que será realizada (CARRARA, 2006).

Conforme a figura 6 há seis tipos de juntas:

- Prismática ou linear (cujo movimento é em linha reta a partir de duas hastes deslizantes entre si);
- Rotacional (efetua um giro rotacional em torno de um eixo de rotação 'imaginário');
- Esférica (combinação de três juntas rotacionais, cujo movimento realizado é em torno de três eixos de rotação);
- Cilíndrica (junção de uma prismática e outra rotacional);
- Planar (duas juntas prismáticas, efetuando movimento em duas direções); parafuso (constituída de um parafuso onde uma porca realiza um movimento prismático em torno de um eixo central).

As mais utilizadas no mercado (especialmente automotivo), devido a sua funcionalidade são as juntas rotativas e prismáticas (CARRARA, 2006).

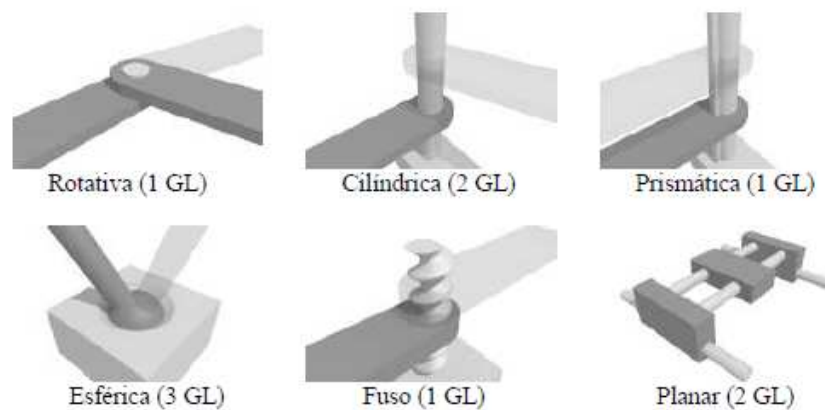


Figura 6: Tipos de juntas utilizadas em robôs.
Fonte: CARRARA, 2006.

2.1.1.4 Órgão terminal

É a ferramenta conectada ao pulso, responsável por efetuar a manipulação dos objetos conforme a função de cada robô. Também é conhecida como garra (CARRARA, 2006).

Este deve ser projetado de modo que possa abrir e fechar os dedos da garra, exercendo força suficiente para que objetos (e.g. peças), quando agarrados, permaneçam seguros. A energia para atuação destes dedos pode ser: mecânica, elétrica, pneumática ou hidráulica (ROMANO, 2009).

Outrossim, as garras podem ser de três tipos, dependendo de sua aplicação: de dois dedos, para objetos cilíndricos e garra articulada (ROMANO, 2009).

A garra de dois dedos é o modelo mais simples, efetuando apenas movimentos paralelos ou rotacionais. Este modelo, apresentado na figura 7, possui uma limitação na abertura dos dedos, sendo pouco versátil na manipulação dos objetos. Logo, a dimensão dos objetos não pode exceder a abertura da garra (ROMANO, 2009).

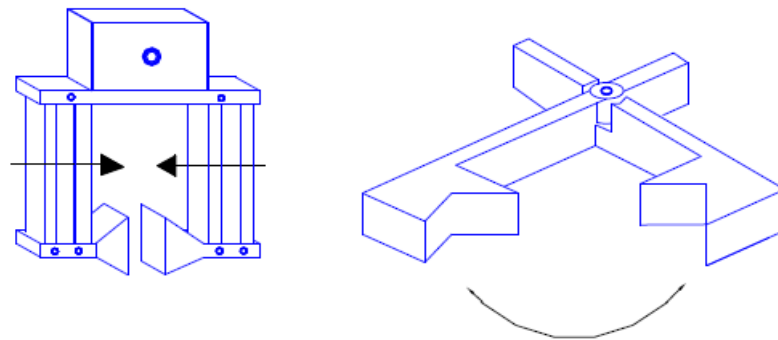


Figura 7: Garra de dois dedos.
Fonte: CARRARA, 2006.

A garra de objetos cilíndricos é formada por dois dedos com semicírculos (como observado na figura 8), permitindo assim segurar objetos cilíndricos de diferentes diâmetros (ROMANO, 2009).

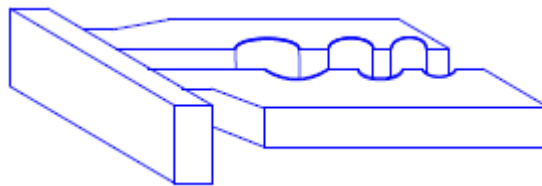


Figura 8: Garra para objetos cilíndricos.
Fonte: CARRARA, 2006.

Por sua vez, a garra articulada é a que mais se assemelha à mão humana, como ilustra a figura 9 e proporciona maior versatilidade para manipular objetos de tamanhos diferentes (ROMANO, 2009).

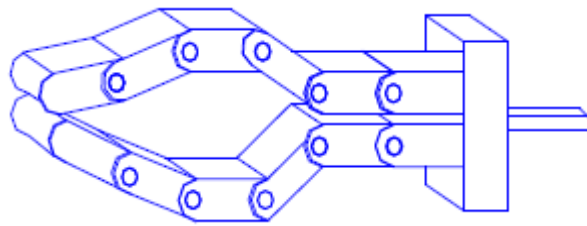


Figura 9: Garra articulada.
Fonte: CARRARA, 2006.

2.1.2 Sensores

São dispositivos cuja finalidade é obter informações acerca do ambiente em que estão inseridos para o sistema de controle com realimentação. Eles permitem que o controle dos robôs interaja com o ambiente de forma “inteligente” e “adaptativa” (ROSÁRIO, 2005).

Dentre as funções dos sensores presentes nos robôs, destacam-se:

- Detecção da posição e orientação das diversas juntas;
- Garantir a qualidade do processo produtivo;
- Detecção do erro nas formas e dimensões das peças produzidas;
- Identificar obstáculos imprevistos durante o movimento do robô;
- Determinar e analisar defeitos (DE PIERI, 2002).

2.1.3 Acionamentos

O acionamento dos manipuladores é a parte responsável pelo movimento das articulações e da dinâmica dos robôs. Os acionadores podem ser classificados em função da energia utilizada:

- Hidráulicos (ou pneumáticos): apresentam torques elevados com massa e volume reduzidos, maior robustez e custo elevado. A principal aplicação é para braços mecânicos que manipulem grandes cargas.
- Elétricos: apresentam sensibilidade, precisão e confiabilidade. São os mais utilizados em robôs presentes em linhas de produção (CARRARA, 2006).

2.1.3.1 Métodos de acionamentos

Acionadores elétricos são, em geral, mais pesados que os hidráulicos e pneumáticos. Devido a deflexões causadas por excesso de peso, bem como restrições de espaço, nem sempre é possível posicionar os acionadores elétricos próximos às juntas. Em virtude disto, os acionadores podem ser acoplados de forma direta ou indireta conforme a figura 10 (SANTOS, 2009).

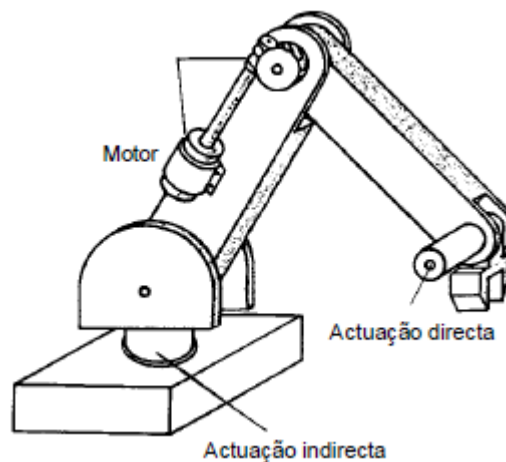


Figura 10: Métodos de acionamento.
Fonte: SANTOS, 2009.

Acionamento direto é aquele em que o acionador é adaptado diretamente à junta, propiciando melhor precisão e rendimento de potência em relação ao indireto. Acionamento indireto é aquele em que o atuador fica afastado da junta movida por ele. Neste acionamento é necessário utilizar algum tipo de transmissão de potência, como polias, correntes, rodas dentadas, engrenagens, parafusos e correias, ou seja, o acionador adaptado deve estar longe da junta pretendida do manipulador (SANTOS, 2009).

2.2 ROBÓTICA NO BRASIL

Conforme um estudo realizado pela comissão econômica das Nações Unidas, no ano de 2001 existiam 756 mil robôs em funcionamento nas indústrias de todo o mundo, sendo 360 mil no Japão e 99 mil na Alemanha. Deste total 1800

pertencia às indústrias latino-americanas, sendo 1200 apenas no Brasil (UOL TECNOLOGIA, 2007).

As maiores usuárias são as indústrias automobilísticas, especialmente as montadoras e indústrias de autopeças (CARRARA, 2006).

Conforme dados relatados pela empresa *ABB Robotics* (líder deste mercado no Brasil com 33% das vendas anuais), as principais aplicações na indústria nacional são:

Tabela 1: Principais aplicações robóticas nas indústrias nacionais.

Aplicação Industrial	Percentual
Soldagem por pontos	33 %
Movimentação / paletização	25%
Soldagem por arco	18 %
Pintura	10 %
Outros (montagem, acabamento, corte por jato-d'água, oxi-corte)	14 %

Fonte: ABB, 2011.

2.3 PRINCIPAIS DESAFIOS EM ROBÓTICA

“Construir um robô envolve uma série de problemas, como erros de posição e orientação, ruídos nas leituras dos sensores, problemas com a carga das baterias, erros de programação entre outros” (DE PIERI, 2002, p.12).

Dentre os principais desafios encontrados no projeto de robôs, destacam-se:

- a) Navegação: dificuldade em criar o ambiente, de forma virtual, em que o robô será inserido para simulação, o mais fiel possível e livre de erros, para evitar surpresas ‘indesejáveis’ em sua utilização, maximizando seu uso.
- b) Aprendizado: na medida em que são adicionadas novas funções ao robô, mais complexa torna-se a sua programação, bem como a calibração de seus sensores e a coordenação dos seus movimentos.
- c) Cooperação: construção de robôs pequenos e baratos que cooperam uns com os outros, como se fossem parte de um ‘formigueiro’ aumentando a produtividade (DE PIERI, 2002).

2.4 CONSIDERAÇÕES FINAIS DO CAPÍTULO

Este capítulo apresentou os conceitos fundamentais relacionados a robótica e aos manipuladores robóticos.

Seu princípio de funcionamento, principais componentes de um robô, componentes mecânicos, breve contextualização deste mercado na indústria brasileira e os principais desafios da robótica.

O próximo capítulo irá relatar os tipos de programação de um robô e como é realizado seu controle.

3 PROGRAMAÇÃO E CONTROLE DE ROBÔS

O robô industrial nasce da união de uma estrutura mecânica articulada e um sistema eletrônico de controle que se integra a um computador. Isto permite a programação e o controle dos movimentos. Programar um robô significa criar um algoritmo com os procedimentos a ser tomados pelo manipulador quando operado pelo controlador.

Conforme ilustrado pela figura 11, cada servomotor é controlado por um *servodrive* (servoconversor) que recebe o monitoramento de posição dos *encoders* acoplados a cada eixo. Desta forma, o sistema de controle sabe a localização exata dos eixos. O computador recebe estes dados de posição e, obedecendo ao programa do usuário, executa as operações de movimento de cada eixo, para atingir a posição programada.

Controlar um robô significa controlar o movimento e a trajetória (direção e orientação) das juntas. O programa é inserido pela unidade de programação, também chamada de *teach pendant*, embora também possa ser editado no teclado do computador. Isto varia de acordo com o modelo e o fabricante do robô (MECATRÔNICA ATUAL, 2010).



Figura 11: Controle de movimentos de um robô
Fonte: MECATRÔNICA ATUAL, 2010.

Devido ao grande número de controladores existentes e a crescente quantidade de sistemas de programação de alto nível lançados pelas empresas,

fez-se necessário criar uma interface entre elas. De acordo com a norma ISO TR 10562: *Manipulating industrial robots – Intermediate code for robots (ICR)* (Manipulando robôs industriais – Código de interface para robôs) um código intermediário entre os programas robóticos orientados ao usuário e controladores de robôs industriais devem ser utilizados como interface entre estes (HARRIS, 2009).

A programação de robôs pode ser realizada de duas maneiras: *online* e *off-line*. A *programação off-line* é feita utilizando linguagens de programação criadas especialmente para robôs tais como VAL, WAVE entre outros. Estes programas são depurados com uso de simuladores. A *programação on-line* utiliza métodos de programação por ensino ou condução (HARRIS, 2009).

3.1 PROGRAMAÇÃO ONLINE

Esta é a chamada programação por ensino: o programador conduz o robô por uma seqüência de posições desejadas, via *teach-pendant* ou por um dispositivo chamado mestre-escravo. Um modelo de *teach-pendant* é apresentado na figura 12. Para isso, são três passos fundamentais: identificação dos pontos da trajetória a ser percorridos, edição e repetição do trajeto ensinado (KURFESS, 2005).



Figura 12: Modelo de *teach-pendant* de um robô Puma fabricado pela Unimation.
Fonte: JET-SESIS, 2010.

Para evitar movimentos desnecessários, oriundos de repetição, ou ainda para algumas outras funcionalidades que podem tornar-se necessárias àquela aplicação, o robô é dependente de algum algoritmo de controle, usado para a

movimentação entre os pontos da trajetória. Os mais utilizados nesta programação são: movimento ponto a ponto, contínuo e controle de trajetória (KURFESS, 2005).

No algoritmo de controle ponto a ponto cada eixo do robô se movimenta de um ponto a outro, sem importar a trajetória percorrida (KURFESS, 2005).

Por sua vez o algoritmo de movimento contínuo move o robô por pequenos incrementos entre si, armazenados em uma trajetória previamente percorrida (KURFESS, 2005).

Por fim, o movimento controlado da trajetória envolve o controle coordenado de todas as juntas para percorrer uma trajetória desejada ao longo de dois pontos. Neste método os eixos movem-se suave e proporcionalmente para gerar o trajeto de controle especificado. Sua principal vantagem é a simplicidade na programação e na sua aprendizagem. Como desvantagem, destaca-se a necessidade de parar o robô – e consequentemente a produção – em eventuais falhas de programação (HARRIS, 2009).

Em programação *on-line* duas abordagens básicas são aplicadas para ensinar o robô a percorrer uma trajetória específica: programação por aprendizagem e linguagens textuais (KURFESS, 2005).

3.1.1 Programação por aprendizagem

Neste método, o operador move fisicamente o efetuator final pelo trajeto desejado. Para trajetórias contínuas, os sensores dos braços enviam informação sobre a posição de cada junta para o controlador do robô, à medida que o braço vai se movimentando. Para trajetórias ponto a ponto, a informação de posição da junta é enviada apenas nos pontos da trajetória onde o operador a posiciona. Em qualquer caso, uma vez gravados na memória do computador os pontos podem ser executados a qualquer momento. É utilizado em tarefas que necessitem uma trajetória contínua, como por exemplo, solda a arco, ou pintura de pulverização (HARRIS, 2009).

A programação por aprendizagem utiliza um *joystick* (controle onde o movimento é semelhante a um leme), um teclado comum ou um teclado portátil chamado de *teach box* para guiar o robô ao longo de uma trajetória planejada (HARRIS, 2009).

3.1.2 Programação por condução

Este sistema é semelhante à programação de computadores: o programador digita o programa utilizado em um terminal na linguagem fornecida pelo fabricante do manipulador. A desvantagem deste método é devido ao fato desses equipamentos possuírem grandes diferenças entre suas configurações e, aliado ao fato serem aplicados a tarefas distintas, cada fabricante desenvolveu sua própria linguagem (vide tabela 2), dificultando o trabalho do operador (MECATRÔNICA ATUAL, 2010).

Tabela 2: Exemplos de algumas linguagens de programação para robôs

Nome	Significado	Empresa	Data
AL	Assembly Language	Stanford University	1974 - 1983
AML	A Manufacturing Language	IBM	1972 - 1982
AUTOPASS	AUTOmated Parts ASsembly System	IBM	1972
IRL	Intuitive Robot Language	Microbo	1978 - 1980
LM	Language de Manipulation	University of Grenoble	1979 - 1983
LPR	Langage de Programmation de Robot	Renault - ACMA	1978 - 1983
MAL	Multipurpose Assembly Language	Milan Polytechnic	1979
MCL	Machine Control Language	McDonnell Douglas	1979 - 1982
PLAW	Programming Language of Arc Welding	Komatsu	1980 - 1982
RAPT	Robot APT	Edinburgh University	1978
ROBEX	ROBot Exapt	Universidade de Aachen	1978
ROL	RObot language	Gixi	1979 - 1983
SERF	Sankyo Easy Robot Formula	Sankyo	1977 - 1979
SIGLA	SIGma Language	Olivetti	1974 - 1976
VAL	Victor's Assembly Language	Stanford University - Unimation	1974 - 1978
VAL II	Victor's Assembly Language II	Unimation - Westinghouse	1979 - 1983

Fonte: MECATRÔNICA ATUAL, 2010.

3.2 PROGRAMAÇÃO OFFLINE

Este método possibilita a programação do manipulador, sem ser necessário a sua utilização durante a tarefa. Neste tipo de programação é importante reprodução fiel do modelo teórico do robô e do ambiente modelo para simular o seu comportamento real. Isto se torna possível graças a controladores cada vez mais sofisticados, ao aumento da precisão no posicionamento e a adoção de sensores mais avançados (ROMANO, 2009).

Para isto, é necessária uma sequência de tarefas como descrito a seguir:

1. Calibração do modelo do robô e da célula.

2. Geração e compilação do código fonte do programa para o manipulador.
3. Descarregar o programa no robô (ROMANO, 2009).

Dentre as principais vantagens deste método, destacam-se:

- Flexibilidade e redução do tempo ocioso: o robô pode manter-se na linha de produção, enquanto a próxima tarefa estiver sendo programada.
- Ambientes potencialmente perigosos: redução do tempo de permanência do operador nestes locais, reduzindo assim o risco de acidentes por comportamento anormal de equipamento.
- Sistema simplificado de Programação: a programação *offline* permite programar uma grande variedade de robôs sem a necessidade de conhecer as peculiaridades de cada controlador.
- Integração com sistemas *Computer Aided Design / Computer Aided Manufacturing*: CAD/CAM (Desenho assistido por computador / Fabricação assistida por computador): realiza interface com banco de dados de peças, centralizando a programação de robôs com estes sistemas, permitindo o acesso a outras funcionalidades, como por exemplo, planejamento e controle.
- Depuração de Programas: sistemas de programação *offline* com CAD/CAM integrados podem produzir um modelo da planta (robô + célula de trabalho), que podem ser utilizados para detectar colisões dentro do espaço de trabalho e a possibilidade de executar determinados movimentos, evitando assim danos aos equipamentos (ROMANO, 2009).

Como principais desvantagens, destacam-se: dificuldade em desenvolver um sistema de programação generalizado que seja independente do robô e de suas aplicações; incompatibilidade entre alguns robôs e sistemas de programação (é necessária a definição de padrões para as interfaces); programas gerados em *offline* devem levar em conta os erros e imprecisões que existem entre o modelo idealizado e o mundo real (ROMANO, 2009).

3.3 A CINEMÁTICA DOS ROBÔS

A cinemática preocupa-se em estudar os movimentos dos robôs, isto é, a posição e a velocidade de seus ligamentos, sendo uma descrição analítica do movimento espacial do manipulador como uma função do tempo. A posição é a posição propriamente dita no espaço tridimensional, enquanto que a velocidade pode ser angular e linear (DE PIERI, 2002). Esta representação depende da posição das juntas e dos elos, onde a base do robô é o ponto de referência (ROMANO, 2009).

A posição do órgão terminal do manipulador depende dos valores dos deslocamentos angulares das juntas rotativas e deslocamentos lineares das juntas prismáticas, para cada instante de tempo. Se for possível conhecer a posição de cada junta, pode-se saber a posição do órgão terminal e, inversamente, caso se conheça a posição da extremidade do robô pode-se calcular qual deve ser a configuração das juntas para atingir tal posição. Cabe salientar que, para o sistema de controle dos braços mecânicos somente as posições das juntas são relevantes (KURFESS, 2005).

Há duas questões a serem resolvidas com relação à cinemática de manipuladores: a *cinemática direta*, cuja função é determinar qual a posição e orientação do manipulador para uma dada posição das articulações, em relação a um sistema de coordenadas de referência; e a *cinemática inversa*, cujo objetivo é resolver o caminho inverso, ou seja, dada uma posição e orientação do extremo do robô, deseja-se descobrir o valor das coordenadas das articulações (DE PIERI, 2002).

Solucionar o problema da cinemática direta é equivalente a encontrar o mapeamento do espaço das juntas para o espaço da configuração da ferramenta, enquanto solucionando o problema da cinemática inversa é equivalente a encontrar o mapeamento do espaço da configuração da ferramenta de volta para o espaço das juntas. A relação entre as duas formas do problema da cinemática pode ser visualizada no esquema da figura 13, a seguir (SANTOS, 2004).

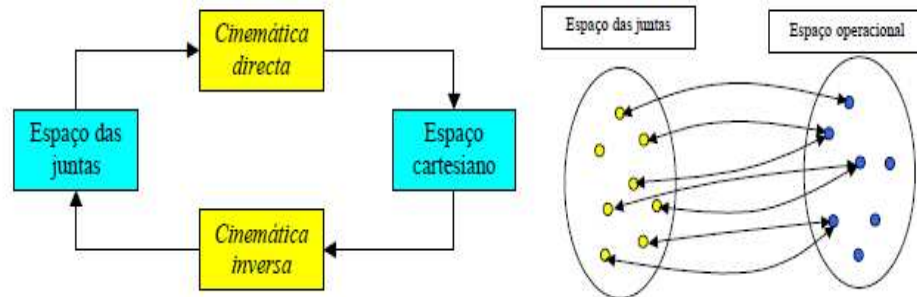


Figura 13: Relação não biunívoca entre as cinemáticas direta e inversa.
Fonte: SANTOS, 2004.

3.3.1 Cinemática Direta

Na cinemática direta, a partir da posição angular de cada uma das juntas que é composto o manipulador, tem-se um vetor composto pelas variáveis das juntas. O objetivo é determinar a posição e a orientação da ferramenta em relação à base, tomada como referência. Conforme já citado neste documento, um manipulador robótico é composto basicamente por uma série de corpos rígidos interligados por articulações. Para programar os movimentos destes dispositivos faz-se necessário estabelecer um sistema de referência situado na base do robô e descrever a localização de cada uma das ligações (juntas) em relação a esta referência. O resultado da cinemática direta é uma equação específica de braço para cada tipo de robô, conforme a sua respectiva configuração. Para isto é necessário conhecer previamente os parâmetros cinemáticos envolvidos em cada um de seus elementos (SANTOS, 2004).

Para posicionar os sistemas de coordenadas nos ligamentos do robô de forma sistemática e obter as equações específicas dos braços, utiliza-se a notação de *Denavit-Hartenberg*. Esta tem por objetivo obter a posição e orientação do manipulador em relação à referência (base). *Denavit e Hartenberg* propuseram um algoritmo para atribuir um sistema de coordenadas ortonormal, sendo um para cada elo em uma cadeia cinemática aberta de elos. Logo, a descrição dos parâmetros de orientação e posição do manipulador torna-se um problema de orientação e descrição de um sistema de coordenadas. Para isto, constrói-se uma matriz 4x4 chamada *matriz de transformação homogênea* (SPONG; VIDYASAGAR, 1989).

Esta matriz, conforme indicado pela equação 2, é formada por uma matriz de rotação R , do tipo 3x3 (responsável por descrever a orientação relativa entre os

dois sistemas de coordenadas, definidos como A e B), e um vetor coluna 3x1, cuja função é indicar a origem do sistema de coordenadas B em relação ao sistema de coordenadas A. Da mesma forma, a segunda e terceira coluna representam as posições dos vetores de B em relação á coordenada A. A última linha da matriz será sempre [0 0 0 1] (SCHILLING, 1990).

$${}^A_B T = \begin{bmatrix} {}^A_B R_{3 \times 3} & {}^A O_{B3 \times 1} \\ 0 & 1 \end{bmatrix} \quad [1]$$

A representação DH de um elo depende de quatro parâmetros, responsável por descrever o comportamento cinemático de uma junta prismática revoluta como ilustrado na figura 14 (SANTOS, 2004).

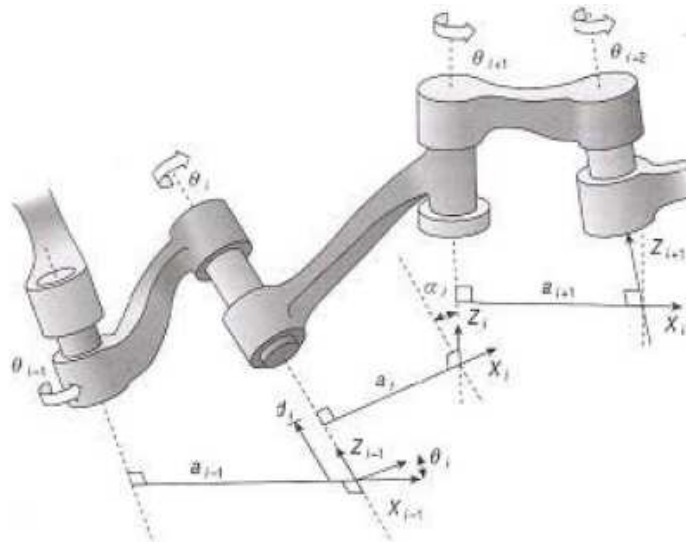


Figura 14: Notação de Denavit Hartenberg.
Fonte: SANTOS, 2004.

De acordo com a figura 14 os parâmetros da notação DH são:

1. θ_i : ângulo de junta obtido entre os eixos X_{i-1} e X_i no eixo Z_{i-1} .
2. d_i : distância entre a origem do $(i-1)$ -ésimo sistema de coordenadas até a intersecção do eixo Z_{i-1} com o X_i , ao longo de Z_{i-1} .
3. a_i : distância entre a intersecção dos eixos Z_{i-1} e X_i até a origem do i -ésimo sistema de referência ao longo do eixo X_i .
4. α_i : ângulo entre os eixos Z_{i-1} e Z_i , medidos no eixo X_i (SCHILLING, 1990).

De posse destes parâmetros, á partir dos sistemas de coordenadas definidos pela notação DH, obtém-se a matriz de transformação homogênea apresentada na equação 3:

$${}^{i-1}A_i = \begin{bmatrix} \cos \theta_i & -\cos \alpha_i \cdot \sin \theta_i & \sin \alpha_i \cdot \sin \theta_i & a_i \cdot \cos \theta_i \\ \sin \theta_i & \cos \alpha_i \cdot \sin \theta_i & -\sin \alpha_i \cdot \sin \theta_i & a_i \cdot \sin \theta_i \\ 0 & \sin \alpha_i & \cos \alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad [2]$$

3.3.2 Cinemática Inversa

A cinemática inversa tem por objetivo encontrar os valores das coordenadas articulares dos manipuladores robóticos para que as suas extremidades assumam um determinado posicionamento. O problema da cinemática inversa consiste, de maneira geral, encontrar as soluções de um sistema de equações com várias variáveis. Todavia três questões devem ser levadas em conta: (SCHILLING, 1990).

- A existência de um conjunto de soluções: neste caso a solução é escolher uma das equações.
- A falta de qualquer solução quando o robô estiver posicionado fora do seu espaço de trabalho: deve-se definir um espaço de trabalho como o conjunto das possíveis posições cartesianas que o robô pode assumir.
- A limitação de soluções em virtude de haver poucos graus de liberdade no robô ou em sua extremidade: em decorrência desta situação, os manipuladores não podem atingir uma posição ou orientação qualquer, pois o espaço de trabalho é um conjunto vazio. A solução é projetar a posição/orientação desejada sobre o espaço de trabalho do robô (implicando na perda de um ou mais graus de liberdade) e calcular a solução conforme o posicionamento desejado (SCHILLING, 1990).

Para resolver o problema da cinemática inversa há três métodos principais (DE PIERI, 2002).

1. Métodos geométricos: é utilizado em casos onde o robô possui poucos graus de liberdade. Basta equacionar as relações geométricas entre as coordenadas articulares e as dimensões de cada um de seus elementos (DE PIERI, 2002).

2. Matriz de transformação homogênea: é o método mais complexo e menos utilizado. Á partir das equações do modelo cinemático direto obtém-se, por manipulação destas, as relações inversas (DE PIERI, 2002)
3. Desacoplamento cinemático: esta solução trata a questão original em dois problemas separados: posição e orientação. Á partir da posição e orientação final conhecidas, são calculados os valores das primeiras variáveis articuladas que atingem este ponto. As outras variáveis são obtidas através dos dados de orientação obtidos na etapa anterior (DE PIERI, 2002).

3.3.3 Sistemas de Controle de Robôs

Um robô pode ser entendido como um sistema de controle em malha fechada com realimentação. O sistema de controle é o responsável pela seqüência lógica do programa de operação, fornecendo os valores teóricos requeridos para cada passo do programa. Através da medição da posição real durante o movimento, ele efetua a diferença entre a esta e a teórica. De acordo com a diferença obtida, a realimentação determinará que o atuador seja acionado para a posição desejada, de modo que a diferença se torne nula em regime permanente (FERREIRA, 2005).

Este controle consiste em girar os seus motores, associados a cada uma das juntas da estrutura do robô de forma a fazer com que o seu órgão terminal atinja a posição e orientação desejada. Logo, este tipo de controle deve responder às mudanças as posições das juntas. Cada ponto do órgão terminal pode ter uma orientação diferente. Para efetuar este controle, faz-se necessário considerar a massa manuseada pelo manipulador, bem como atrito e aceleração envolvidos entre os componentes mecânicos (FERREIRA, 2005).

O controlador é normalmente um microprocessador ou um computador, responsável pelas seguintes funções:

- Realimentação: de acordo com o valor obtido da diferença entre a variável medida e a de referência, a realimentação determina o acionamento do atuador em uma determinada posição para que a diferença entre as duas variáveis torne-se nula, eliminando o erro.

- Cinemática: determina o estado real da garra, a partir do estado real dos atuadores (posição e velocidade). Da mesma forma, dado o estado real da mão, determina-se o estado desejado de cada atuador.
- Dinâmica: utiliza as informações das cargas nos braços (inércia, atrito, gravidade e aceleração) para ajustar o controle de realimentação.
- Sensores: de acordo com as especificações das tarefas a serem realizadas, eles determinam os comandos para os movimentos dos robôs (ROSÁRIO, 2005).

Entretanto, controlar um manipulador robótico significa especificar uma tarefa e determinar quais serão as ações necessárias à conclusão desta. Para tanto, devem ser considerados os seguintes itens: tipo e aplicação do manipulador, trajetória desejada a ser percorrida, atuadores e sensores utilizados nas medições, transmissão entre as juntas e os atuadores e a presença de perturbação (BIANCHI, 2010).

Há duas formas de controlar um manipulador: em malha aberta (manipuladores clássicos) e em malha fechada (também chamados servo-controladores): sistemas ponto-a-ponto e de trajetória controlada (BIANCHI, 2010).

1. Controle ponto-a-ponto: o robô se desloca de um ponto a outro dentro de seu volume de espaço de trabalho e tem como variáveis controladas, distância e trajetória percorrida. É aplicado em algumas operações de soldagem.
2. Controle de trajetória contínua: neste a garra do manipulador segue uma trajetória previamente especificada, permitindo o controle de movimento e velocidade de cada eixo simultaneamente. É aplicado em pintura e soldadura.
3. Trajetória controlada: permite realizar movimentos em trajetórias com geometrias diferentes (por exemplo: linhas, círculos etc.), sendo apenas necessário memorizar os pontos de início e fim de trajetória.
4. Controle inteligente: permite o robô ajustar as trajetórias conforme o ambiente em que ele se encontra. Isto é feito através das informações recebidas pelos sensores (FERREIRA, 2005).

Em malha aberta o sinal de controle não é utilizado a partir de uma medição do sinal de saída: não há sensores nem tampouco retroação. Neste tipo de projeto, o robô pára ao atingir um determinado ponto pré-fixado. É utilizado em robôs de

seqüência limitada, como por exemplo, os *pickandplace*. Dentre as suas principais vantagens, destacam-se a simplicidade e o baixo custo de projeto, bem como fácil operação e manuseio, além de alta repetibilidade de operação. Como desvantagem, destaca-se a não-transmissão de informação, já que este controle não dispõe de sensores (BIANCHI, 2010).

Em malha fechada são utilizadas as informações correspondentes à saída de controle para determinar qual sinal deve ser aplicado a planta em um determinado instante (a entrada controlada depende apenas deste sinal). Seu sistema mede as posições das juntas e as compara com a posição desejada. Caso haja erro (posição atual diferente da programada) é efetuada uma *realimentação* da saída para a entrada para levá-las a posição determinada, além de tornar o sistema robusto o suficiente a ponto de rejeitar as perturbações externas conforme ilustrado pela figura 15 (BIANCHI, 2010).

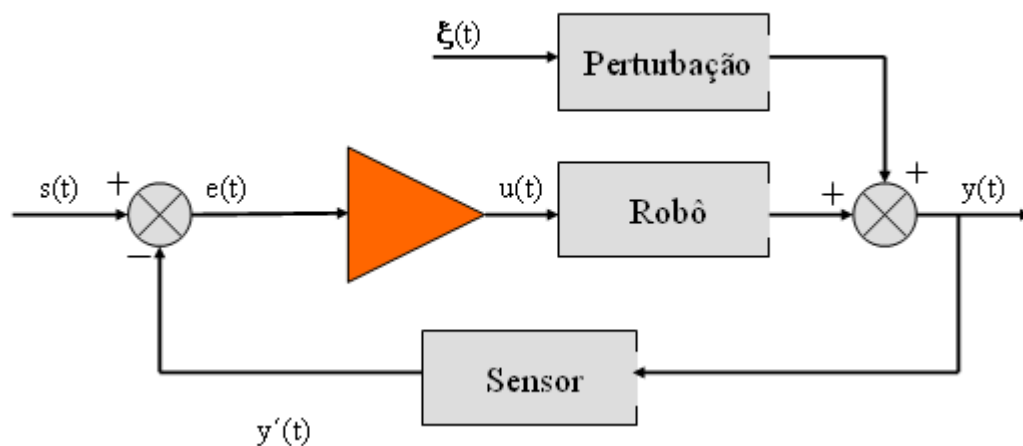


Figura 15: Sistema de controle em malha fechada.
Fonte: BIANCHI, 2010.

Um sinal de saída, dado por $y(t)$ é comparado ao de referência $s(t)$ – *set point* – e a diferença obtida – chamado de *signal de erro*: $e(t)$ – é utilizada para determinar a ação de controle a ser empregada: $u(t)$. Logo o controlador é projetado para eliminar a diferença entre eles, além de eventuais perturbações $\xi(t)$. Como vantagens, destacam-se acurácia, maior torque e velocidade, possibilidade de realizar tarefas mais complexas e de automação flexível. A principal desvantagem está no custo elevado de implementação, já que se faz necessário à presença de profissionais dedicados devido à maior complexidade de programação (BIANCHI, 2010).

Em sistemas robóticos, o objetivo de controle é solucionar o problema de controle de posição e orientação da garra. A arquitetura de controle destes dispositivos tem uma malha específica para cada uma das juntas (graus de liberdade), baseado em um controle do tipo realimentação em cascata da posição e da velocidade (GIMENO, 1988).

3.4 DINÂMICA DOS MANIPULADORES ROBÓTICOS

Visando determinar os esforços necessários á realização do movimento de um robô é necessário o conhecimento de seu modelo dinâmico. Para acelerar um robô, a partir do repouso com o efetuador operando a uma velocidade constante e, finalmente cessando o seu movimento até parar, um complexo conjunto de esforços devem ser aplicados aos efetuadores por meio das articulações. As forças e os torques necessários dependem de certos parâmetros espaciais e temporais da trajetória do manipulador e o cálculo dos esforços dos atuadores para controlar um manipulador, de modo que ele siga fielmente uma trajetória pré-determinada é feito a partir das equações da dinâmica. Ela é necessária para o projeto, simulação e análise de movimento dos dispositivos (LATRE, 1988).

A dinâmica dos manipuladores lida com as equações diferenciais do movimento, de modo que os robôs se movam de acordo com os torques aplicados pelos atuadores ou por forças externas (CORKE, 2008).

A modelagem dinâmica dos manipuladores apresenta termos não-lineares devido aos efeitos gravitacionais, de coriolis, força e torque centrífugos. Essas não-linearidades geram dificuldades na identificação de seus parâmetros, bem como erros de regime e ciclos limites (LATRE, 1988).

O modelo dinâmico das n juntas de um manipulador robótico é dado por:

$$M(q) \ddot{q} + C(q, \dot{q}) \dot{q} + F(\dot{q}) + G(q) = Q \quad [3]$$

Onde:

q = vetor com as coordenadas das juntas que descrevem a posição do robô;

\dot{q} = vetor de velocidade das juntas;

q'' = vetor com a aceleração das juntas;

M = matriz $n \times n$ da massa do manipulador;

C = descreve os efeitos da força centrípeta e o torque de Coriolis;

F = descreve os atritos: viscoso e de Coulomb;

G = aceleração da gravidade;

Q = vetor que descreve as forças associadas às coordenadas dadas por q .

3.5 CONTROLE DA TRAJETÓRIA DO ROBÔ

Para que um manipulador robótico se movimente de forma suave e coordenada a partir de uma posição inicial até uma posição final, deve-se fazer com que cada uma das articulações que compõem o robô se movimente conforme uma função contínua. Cada uma das articulações deve iniciar e terminar o movimento no mesmo instante, de forma que ele seja totalmente coordenado (CARRARA, 2006).

O planejamento da trajetória não envolve apenas as posições: inicial e final, mas também as posições intermediárias que devem ser percorridas pelo efetuador, conforme demonstrado pela figura 16 (CARRARA, 2006).

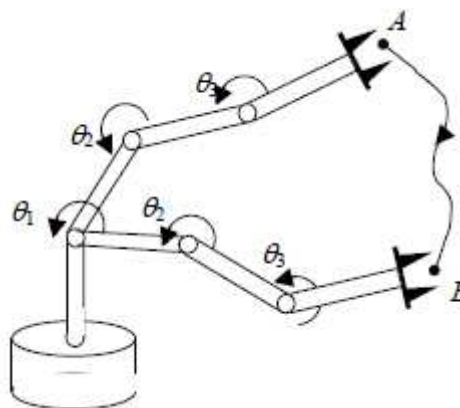


Figura 16: Exemplo de trajetória percorrida por um manipulador.
Fonte: ROMANO, 2009.

Para que o efetuador siga uma determinada trajetória, o movimento deve ser convertido para um movimento equivalente ao das articulações. A obtenção deste movimento é mediante a utilização do método de interpolação com polinômio cúbico para o cálculo da trajetória, que especificam a posição e a velocidade nos extremos de cada trecho. Este planejamento no espaço das juntas implica em

determinar a evolução ao longo do tempo de forma tal que sejam verificadas as condições cinemáticas das juntas de posição, velocidade e aceleração, do início ao final da trajetória. Será a quantidade de pontos intermediários entre o início e fim da trajetória que determinará o quão suave será o movimento do manipulador (CARRARA, 2006).

Supondo uma posição: inicial e final, com ambas as velocidades iguais à zero. Partindo da equação diferencial de terceira ordem, com condições iniciais nulas, obtêm-se, respectivamente, as funções da posição, velocidade e aceleração das articulações.

$$\begin{aligned}\Theta(t) &= a_3 t^3 + a_2 t^2 + a_1 t + a_0 \\ \frac{d\theta}{dt} &= 3a_3 t^2 + 2a_2 t + a_1 \\ \frac{d^2\theta}{dt^2} &= 6a_3 t + 2a_2\end{aligned}\quad [4]$$

3.6 CONSIDERAÇÕES FINAIS DO CAPÍTULO

Este capítulo apresentou os conceitos relacionados à programação e ao controle de manipuladores robóticos.

Controlar um robô significa controlar o movimento e a trajetória (orientação e direção das juntas).

Este movimento pode ser classificado de duas formas: cinemática direta e cinemática indireta. Conforme o tipo de cinemática serão necessárias diferentes variáveis para o seu controle.

Além disso, foram apresentados os dois tipos de programação: *online* e *offline*.

Na programação *online* o programador conduz o robô por uma seqüência de posições desejadas. Isto é feito através da “programação por aprendizagem” e por meio de linguagens textuais.

A programação *offline* pode utilizar dos mesmos recursos da *online* (por aprendizagem e linguagens textuais). Porém o programa é inserido sem que o robô esteja operando.

O próximo capítulo descreverá a ferramenta de simulação e controle robótica do MATLAB e realizará um caso de estudo de um robô real.

4 A FERRAMENTA DE ROBÓTICA DO MATLAB E UM CASO DE ESTUDO: CONTROLE E SIMULAÇÃO DE UM ROBÔ REAL

MATLAB (abreviação de *Matrix Laboratory* – laboratório de matrizes) é um programa de computador desenvolvido na década de 80 por Cleve Moler, no Departamento de Ciência da Computação da Universidade de Novo México, Estados Unidos. Foi projetado inicialmente para efetuar cálculos científicos de engenharia, otimizando o tempo dos usuários. Seu conceito inicial foi baseado em operações matriciais (daí a origem de seu nome), porém com o passar dos anos foi se tornando um sistema flexível, através da implantação de novas funções que tornam possível a resolução de qualquer problema técnico (CHAPMAN, 2003).

O *software* implementa uma linguagem de programação: MATLAB, com funções predefinidas e que admite ferramentas adicionais com as mais diversas finalidades. A unidade fundamental de dados neste programa é a matriz: uma coleção de dados organizados em linhas e colunas e que podem ser utilizados mediante a utilização de um nome específico. Uma variável em MATLAB é uma região utilizada pela memória para o conteúdo de uma matriz específica (CHAPMAN, 2003).

4.1 A BIBLIOTECA DE ROBÓTICA DO MATLAB

Conforme mencionado no início deste capítulo, o *software* tem a sua versão básica, composta de um pacote de funções matemáticas necessárias à resolução de qualquer problema técnico. Para a solução de questões mais complexas, como processamento de sinais em tempo real ou simulação e controle de robôs, é necessário à obtenção de uma ferramenta específica a esta função (CHAPMAN, 2003).

No caso deste trabalho a ferramenta utilizada para controle e simulação de robô é a *Robotic Toolbox for MATLAB* (Ferramenta Robótica para MATLAB) e foi

desenvolvida pelo Phd. australiano em engenharia mecânica, da Universidade de Queensland, Peter Corcke. Ela encontra-se disponível gratuitamente no endereço:

http://petercorcke.com/Robotics_Toolbox.html.

Esta ferramenta possui um conjunto de funções que permite operar e simular os robôs obtidos a partir do Matlab. Entre as principais atribuições deste aplicativo, destacam-se a cinemática, dinâmica e geração de trajetória de um robô. Além disto, ela possui funções necessárias à manipulação de dados, (como por exemplo, transformação homogênea), necessários à representação dos manipuladores em três dimensões. Além disso, possui os modelos de dois manipuladores prontos: PUMA 560 e *Stanford Arm* (CORKE, 2008).

Para o controle e simulação da posição e dos esforços realizados pelas juntas dos robôs é necessário o ambiente de simulação e edição de diagramas de blocos do Matlab: *Simunlink*¹.

O *Simunlink* é utilizado para modelagem, simulação e análise de sistemas dinâmicos, suportando sistemas lineares e não-lineares em tempo contínuo e discreto. A partir de um sistema criado em diagramas de blocos o usuário pode simular o modelo e verificar os resultados enquanto a simulação ainda estiver sendo executada. Este artifício pode ser utilizado em qualquer ponto do programa, em qualquer ambiente de programação, bem como efetuar a mudança dos parâmetros do sistema e verificar o que acontece imediatamente (CHAPMAN, 2003).

Um exemplo deste tipo de diagrama pode ser observado na figura 17. Este diagrama também foi desenvolvido pelo criador da ferramenta e distribuído gratuitamente, junto com a *toolbox* para simular e verificar graficamente a variação do torque de um manipulador. O diagrama foi construído baseado no robô PUMA 560.

Neste caso o usuário fornece como parâmetros de entrada um vetor de posição. Com as derivadas: primeira e segunda, respectivamente, obtêm-se a variação da velocidade e aceleração. Com o bloco chamado *Robot* observa-se a variação do torque das juntas do manipulador.

¹ O software MATLAB e seu simulador: *Simunlink* podem ser obtidos em uma versão sem custos, especialmente para estudantes no endereço: <http://www.mathworks.com/>.

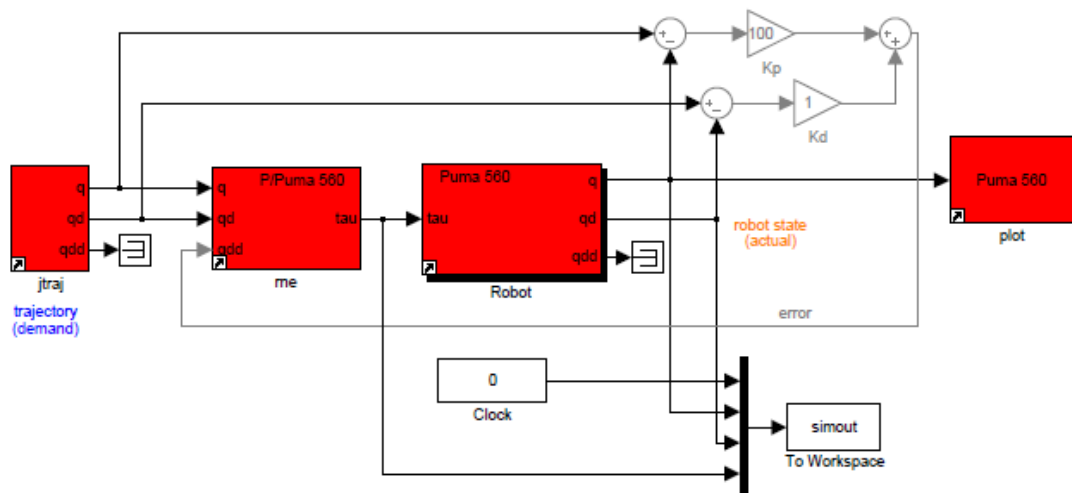


Figura 17: Diagrama de blocos para o controle de torque.
Fonte: CORKE, 2008.

4.2 CASO DE ESTUDO: CONTROLE E SIMULAÇÃO DE UM ROBÔ REAL

Com base na descrição das variáveis necessárias para realizar o controle de um manipulador robótico, será realizado um caso de estudo de um robô real: o “Stanford Arm” (braço de Stanford), desenvolvido pelo engenheiro mecânico norte-americano Vitor Scheinman. O robô é mostrado na figura 18.

Em 1969, foi desenvolvido no SAIL – *Stanford Artificial Intelligence Laboratory* (Laboratório de Inteligência Artificial de Stanford), na Universidade de Stanford, Estados Unidos, o primeiro robô elétrico e controlado por um computador: o braço de Stanford. Este robô se caracterizou por ser um sistema elétrico com seis eixos articulados e projetados para seguir uma trajetória qualquer, pré-definida por um operador, utilizando a anatomia de um braço humano. Sua importância reside no fato de ele ser, até hoje, fundamental para o projeto e construção de braços robóticos (STANFORD.EDU, 2010).

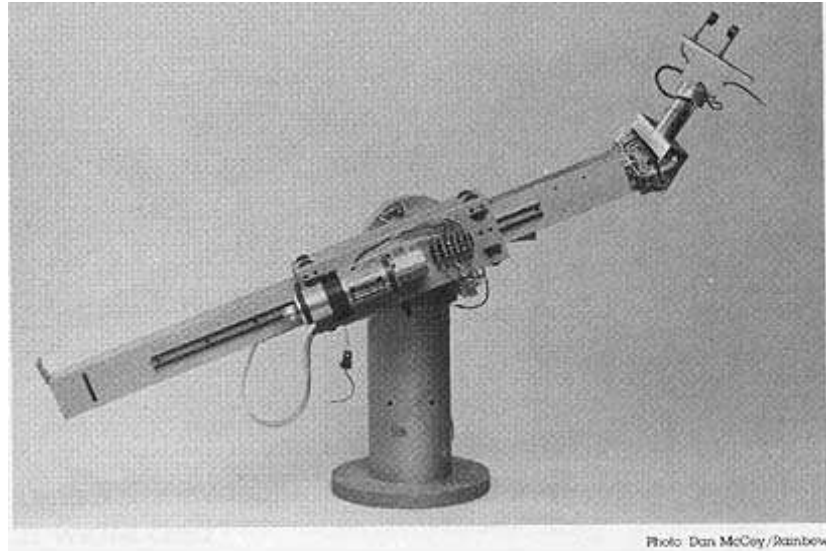


Figura 18: Braço de Stanford.
Fonte: PIMENTA, 2009.

Como mencionado anteriormente, a *toolbox* de robótica do MATLAB fornece uma série de funções necessárias á análise e controle de manipuladores robóticos, tais como a cinemática, dinâmica e geração de trajetória. Para isto, os manipuladores devem ser descritos em notação matricial, de acordo com as 'ligações' entre as juntas dos robôs (CORKE, 2008).

Estabelecendo os referenciais de acordo com a figura 19, conforme a notação de DH tem-se que:

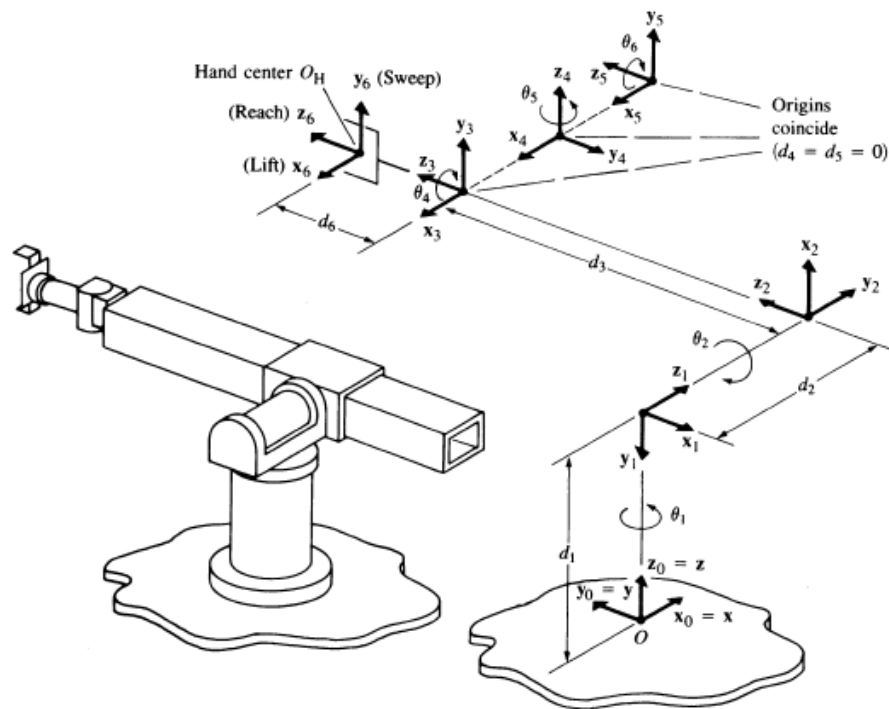


Figura 19: Referenciais geométricos do robô Stanford.
Fonte: LOPES, 2001.

Para efetuar o controle dos manipuladores, são necessários os seguintes parâmetros:

α, a, θ, d – parâmetros de DH;

σ – define se a junta é rotacional (0) ou prismática (1);

m – massa do ligamento das juntas;

r_x, r_y e r_z – coordenadas do centro de massa do ligamento;

$I_{xx}, I_{yy}, I_{zz}, I_{xy}, I_{xz}, I_{yz}$ – matriz de inércia;

Jm – momento de inércia do motor de rotação da base do robô;

Bm – atrito viscoso do motor de rotação da base do robô;

Tc – atrito seco;

G – Relação de redução da transmissão: velocidade do comum / velocidade da junta.

Os parâmetros do robô de Scheinman (Stanford), para cada uma das seis juntas do braço são:

Tabela 3: Valores reais dos parâmetros das juntas do braço de Scheinman.

	Junta 1	Junta 2	Junta 3	Junta 4	Junta 5	Junta 6
α [rad]	$-\pi/2$	$\pi/2$	$-\pi/2$	$-\pi/2$	$\pi/2$	0
a [m]	0	0	0	0	0	0
θ [rad]	0	0	0	0	0	0
d [m]	0,412	0,154	0	0	0	0,263
σ (R/P)	0	0	1	0	0	0
m [kg]	9,29	5,01	4,25	1,08	0,63	0,51
r_x [m]	0	0	0	0	0	0
r_y [m]	0,0175	-1,054	0	0,092	0	0
r_z [m]	-0,1105	0	-6,447	-0,054	0,566	1,554
I_{xx} [kg.m ²]	0,276	0,108	2,51	0,002	0,003	0,013
I_{yy} [kg.m ²]	0,255	0,018	2,51	0,001	0,003	0,013
I_{zz} [kg.m ²]	0,071	0,1	0,006	0,001	0,0004	0,0003
I_{xy} [kg.m ²]	0	0	0	0	0	0
I_{yz} [kg.m ²]	0	0	0	0	0	0
I_{xz} [kg.m ²]	0	0	0	0	0	0
J_m [kg.mm ²]	0,953	2,193	0,782	0,106	0,097	0,02
B_m [Nm/rd/s]	-	-	-	-	-	-
T_c - [Nm]	-	-	-	-	-	-
T_c - [Nm]	-	-	-	-	-	-
G	1	1	1	1	1	1

Fonte: BARBUTO, 2009.

4.3 MODELAGEM DO ROBÔ DE SCHEINMAN

Para efetuar a modelagem do manipulador de Scheinman (ou Stanford), algumas considerações devem ser feitas:

- A base do robô, que é o apoio que sustenta o corpo do manipulador, é fixa.
- As juntas não sofrem deflexões.
- Os torques aplicados aos efetadores podem ser comandados e controlados.
- O motor a ser modelado é o de corrente contínua, com tensão aplicada na armadura e utilizando a força contra-eletromotriz como realimentação (SCHILLING, 1997).

Com os valores variáveis de DH descritos na tabela 2, o MATLAB (a partir da execução de uma rotina²) gera um manipulador geométrico, correspondente ao braço do robô, conforme ilustrado na figura 20.

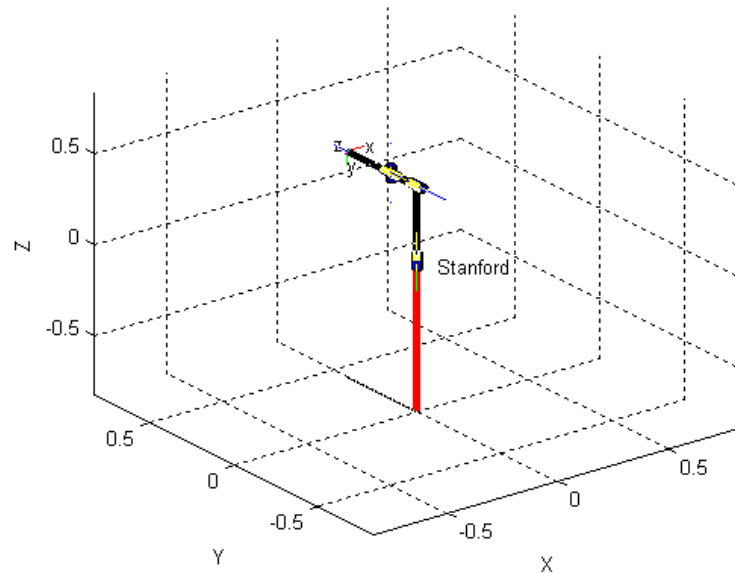


Figura 20: Braço de Stanford gerado pelo MATLAB em sua posição inicial, (coordenadas angulares iguais à zero).

4.3.1 Simulação - Cinemática Direta

Para resolver o problema da cinemática direta o MATLAB cria, (com base na notação de DH), uma matriz T chamada matriz de transformação homogênea, onde cada linha da matriz relaciona um elo do manipulador e suas colunas são representadas conforme as seguintes funções: (CORKE, 2008).

Coluna 1: α_i = ângulo de rotação da junta.

Coluna 2: A_i = deslocamento da junta.

Coluna 3: θ_i = ângulo de rotação da junta.

Coluna 4: D_i = comprimento da junta.

Coluna 5: σ_i = tipo da junta (0 se for prismática ou 1 de revolução).

A partir dos valores do manipulador fornecidos na tabela 3 e seus parâmetros de DH, sua matriz de transformação homogênea, (com base no último elo do manipulador em sua posição inicial), descrita pelo vetor q é dada por:

² As rotinas utilizadas para a simulação neste documento encontram-se no final deste documento em um arquivo .m, em anexo.

$$q = [0 \ 0 \ 0 \ 0 \ 0 \ 0]$$

$$T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0.4170 \\ 0 & -1 & 0 & 0.4120 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

As primeiras três colunas da matriz correspondem à rotação, enquanto que a última coluna da matriz corresponde à translação do manipulador em torno das coordenadas fornecidas pelo segundo argumento da função *fkine* (CORKE, 2008).

A relevância deste conceito está no fato de que a partir de um ponto qualquer dado por suas coordenadas angulares (sempre em *radianos*), o operador pode descobrir a posição do elo final do manipulador. Ou seja, a movimentação do manipulador é feita de forma gradativa e minuciosa, movimentos são executados através do incremento do valor exato do ângulo relativo à uma determinada junta (CORKE, 2008).

Modificando o conjunto de coordenadas angulares dado por q , obtém-se uma nova matriz de transformação homogênea T e uma nova configuração geométrica para o manipulador, cuja simulação é apresentada na figura 21. Após isto foi efetuada uma nova simulação, a partir de um novo vetor q , conforme visto na figura 22.

$$q = [0 \ -\pi/4 \ -\pi/4 \ 0 \ \pi/8 \ 0]$$

$$T = \begin{bmatrix} 0.6533 & 0.7071 & 0.2706 & 0.6265 \\ -0.3827 & 0.0000 & 0.9239 & 0.3970 \\ 0.6533 & -0.7071 & 0.2706 & -0.0722 \\ 0.0000 & 0.0000 & 0.0000 & 1.0000 \end{bmatrix}$$

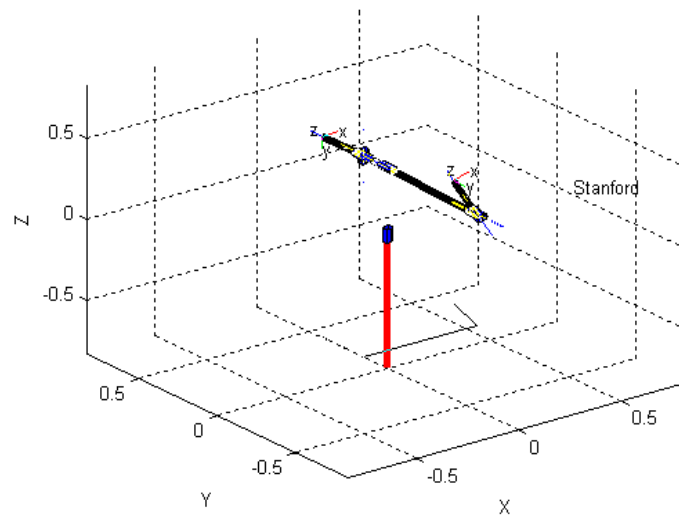


Figura 21: Resultado da cinemática direta para as coordenadas dadas por q .

$$q = [0 \quad 0 \quad 0 \quad \pi/2 \quad 3\pi/4 \quad 2\pi/8]$$

$$T = \begin{bmatrix} -0.7071 & -0.3536 & 0.6124 & 0.1611 \\ 0.0000 & 0.8660 & 0.5000 & 0.2855 \\ -0.7071 & 0.3536 & -0.6124 & 0.2509 \\ 0.0000 & 0.0000 & 0.0000 & 1.0000 \end{bmatrix}$$

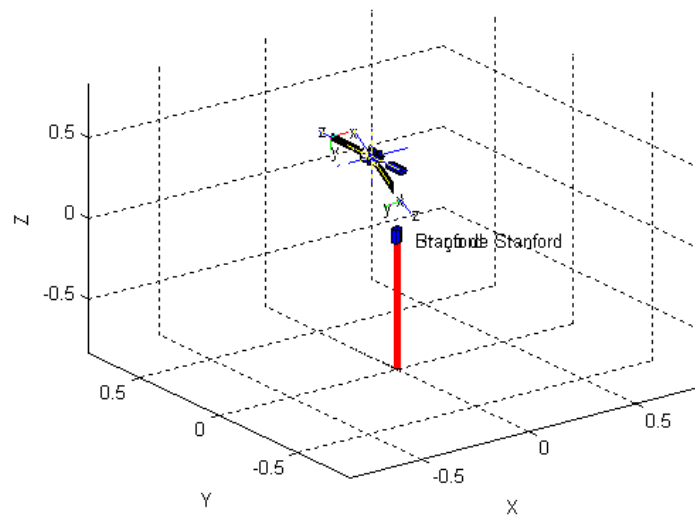


Figura 22: Simulação da cinemática direta para outro vetor de coordenadas angulares.

4.3.2 Simulação - Cinemática Inversa

Para uma posição e orientação do efetuador final, determinam-se quais as posições das juntas que satisfazem à posição do mesmo. Trata-se de uma questão mais complexa que a cinemática direta, já que o algoritmo de solução é diferente para cada robô por apresentar diferença nos graus de liberdade (CORKE, 2008).

Supondo um vetor de posições angulares: $q = [0 \ \pi/12 \ \pi/8 \ 0 \ \pi/6 \ 0]$.

Neste caso, sua matriz de transformação homogênea será:

$$T = \begin{bmatrix} 0.8365 & -0.2588 & 0.4830 & 0.2287 \\ -0.5000 & 0.0000 & 0.8660 & 0.3818 \\ -0.2241 & -0.9659 & -0.1294 & 0.7573 \\ 0.0000 & 0.0000 & 0.0000 & 1.0000 \end{bmatrix}$$

Aplicando a função *ikine*, o vetor encontrado será, em radianos, $q_i = [0.0000 \ 0.2618 \ 0.3927 \ 0.0000 \ 0.5236 \ 0.0000]$, exatamente igual a q . A configuração geométrica será conforme introduzido na figura 23 a seguir:

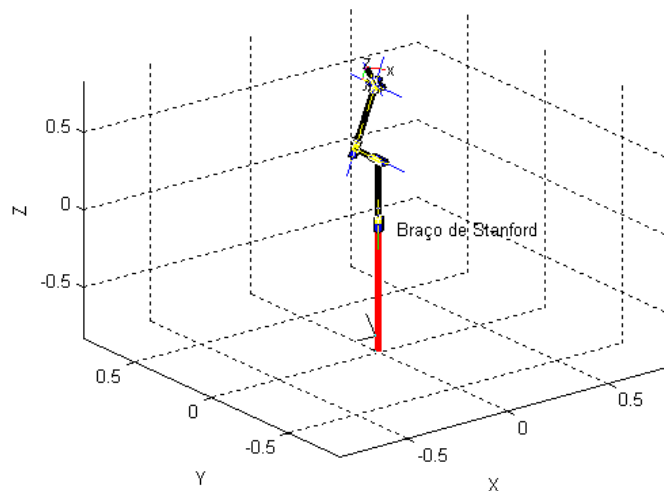


Figura 23: Simulação da cinemática inversa.

Esta mesma solução pode ser obtida através da solução de complexas equações diferenciais. Além de ter uma solução para cada robô (em decorrência de diferentes robôs terem diferentes graus de liberdade), também compromete a escolha de valor inicial, bem como o tempo gasto pelo projetista em buscar a solução ideal, levando em conta todas estas questões. E como último dificultador,

um mesmo manipulador pode ter várias posições que resultem na mesma matriz de transformação homogênea para a última junta. Para a questão da cinemática inversa, nem mesmo a escolha do zero como ponto de partida do robô é uma boa escolha, pois ele limita o controle sobre a solução particular da equação diferencial. Uma solução não será possível se a matriz de transformação homogênea descrever um ponto fora do alcance do manipulador. Esta *toolbox* oferece, em termos da cinemática inversa, apenas uma descrição generalizada do robô, em termos de seus parâmetros cinemáticos (CORKE, 2008).

4.3.3 A Matriz Jacobiana

Visando eliminar as indeterminações causadas pela cinemática inversa utiliza-se o conceito da matriz jacobiana, que serve para relacionar as velocidades das coordenadas articulares às posições e orientações do efetuador no robô, controlando assim a cinemática dos mecanismos. Em outras palavras ela calcula as velocidades necessárias nas juntas (no plano cartesiano) para que o efetuador alcance uma determinada velocidade (SCHILLING, 1997).

Para uma configuração inicial, a partir de pequenos deslocamentos associados às articulações calcula-se esta matriz a partir das relações cinemáticas da arquitetura do manipulador. Para este as coordenadas de seus efetuadores são descritas pelo vetor de posição q (ROMANO, 2009).

A partir das velocidades de cada articulação, com o cálculo da matriz jacobiana inversa obtém-se a velocidade de cada uma das articulações (ROMANO, 2009).

Para $q = [0 \quad -\pi/4 \quad -\pi/4 \quad 0 \quad \pi/8 \quad 0]$; as matrizes de transformação homogênea – chamada de J_1 para diferenciar – e jacobiana J (em função da posição final do efetuador) são, respectivamente dadas por:

$$T = \begin{bmatrix} 1.0000 & 0.0000 & 0.0000 & 0.0000 \\ 0.0000 & 0.0000 & 1.0000 & 0.4170 \\ 0.0000 & -1.0000 & 0.0000 & 0.4120 \\ 0.0000 & 0.0000 & 0.0000 & 1.0000 \end{bmatrix}$$

$$J_1 = \begin{bmatrix} -0.3970 & -0.4842 & -0.7071 & 0.0712 & 0.1718 & 0.0000 \\ 0.6265 & 0.0000 & 0.0000 & 0.0000 & -0.1006 & 0.0000 \\ 0.0000 & -0.6265 & 0.7071 & -0.0712 & 0.1718 & 0.0000 \\ 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.7071 & 0.2706 \\ 0.0000 & 1.0000 & 0.0000 & 1.0000 & 0.0000 & 0.9239 \\ 1.0000 & 0.0000 & 0.0000 & 0.0000 & -0.7071 & 0.2706 \end{bmatrix}$$

$$J = \begin{bmatrix} -0.4991 & -0.7256 & 0.0000 & 0.0000 & 0.2630 & 0.0000 \\ -0.2807 & 0.1006 & -1.0000 & 0.1006 & 0.0000 & 0.0000 \\ 0.4714 & -0.3006 & 0.0000 & 0.0000 & 0.0000 & 0.0000 \\ 0.6533 & -0.3827 & 0.0000 & -0.3827 & 0.0000 & 0.0000 \\ -0.7071 & 0.0000 & 0.0000 & 0.0000 & 1.0000 & 0.0000 \\ 0.2706 & 0.9239 & 0.0000 & 0.9239 & 0.0000 & 1.0000 \end{bmatrix}$$

Se a singularidade do manipulador (fornecida pelo determinante da matriz jacobiana J) for igual à zero, é impossível controlar a velocidade da garra do robô (neste caso, a matriz não admite inversão) (SCHILLING, 1997).

A determinante da matriz J para o vetor q descrito é dada por d = -0.1669. Como esta é diferente de zero, é possível controlar a garra do efetuador.

Supondo que para uma determinada tarefa, um robô necessite alcançar uma velocidade de 0,25 m/s em um movimento de translação. De acordo com a última equação, os efetuadores das juntas deverão ter uma velocidade, cada um, de:

$$q = [-0.1723 \quad -0.2702 \quad 0.0188 \quad -0.0239 \quad -0.1218 \quad 0.3183]^3$$

Quando a determinante da matriz jacobiana é nula (logo ela não é inversível e consequentemente a velocidade do manipulador não pode ser controlada), esta servirá apenas para informar o quão bem posicionado o manipulador está para efetuar determinados movimentos. Esta é a função *maniply* (manipulabilidade), e baseia-se nos parâmetros cinemáticos. Como ela apenas informa o quão 'bem posicionado' o manipulador está o argumento de entrada é uma posição qualquer fornecida pelo operador, e seu retorno determinará o quão longe o efetuador está da posição desejada (CORKE, 2008).

³ O sinal do vetor q representa apenas o sentido de translação da junta.

4.3.4 Simulação da Trajetória

A *toolbox* do MATLAB simula, de acordo com as matrizes de transformação homogêneas, a trajetória percorrida, a variação de velocidade e aceleração, descritas por um manipulador no plano cartesiano, á partir de um ângulo zero de acordo com o intervalo de tempo fornecido por um vetor t (CORKE, 2008).

Foi arbitrado um tempo de deslocamento de 4s, e um período de amostragem de 70ms e calculou-se a trajetória no espaço cartesiano mediante a função *jtraj*, onde a matriz retornada pela função é uma matriz cujas linhas representam um período de amostragem, e a coluna o ângulo da articulação referente ao período anterior (CORKE, 2008).

O resultado desta simulação pode ser observado nos gráficos: 1, 2 e 3, cujo resultado indica que apenas as juntas: 2 e 3 estão movimentando-se. Isto pode ser comprovado pelo fato de todas as linhas referentes às demais juntas serem nulas.

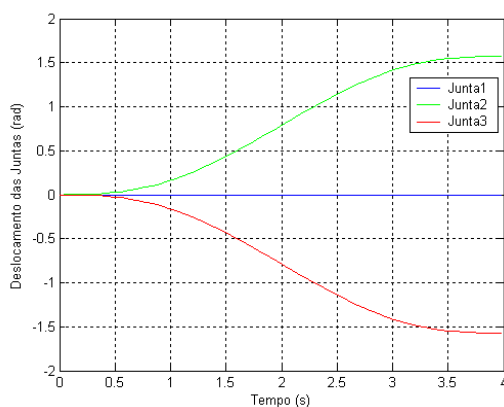


Gráfico 1: Deslocamento das Juntas.

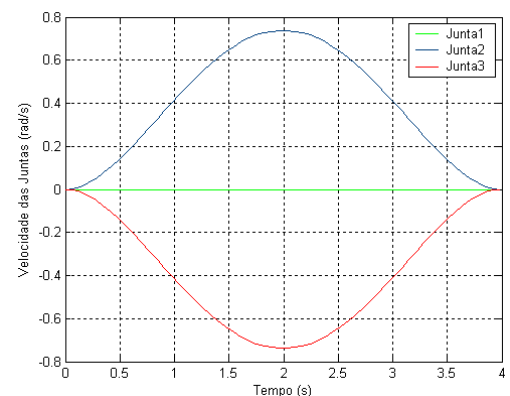


Gráfico 2: Velocidade das Juntas.

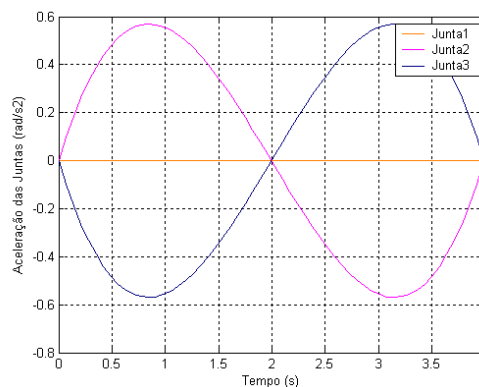


Gráfico 3: Aceleração das juntas.

Como mencionado, apenas as juntas 2 e 3 estão em movimento, e portanto, apenas estas são analisadas. Isto decorre do fato de que apenas estas duas juntas respondem pelo posicionamento de uma ferramenta qualquer acoplada ao efetuador do manipulador. Elas são as juntas mais importantes no sentido de controle e seus acoplamentos são os mais robustos, bem como são as juntas com o maior número de não-linearidades. A junta 1 é considerada como a junta relativa à base de referência do manipulador. Como esta base é fixa, não há trajetória percorrida bem como velocidade e aceleração.

As juntas 4, 5 e 6 não causam qualquer movimento de translação ao efetuador final – característica de manipuladores com punhos esféricos. Logo não há variação de trajetória, velocidade e aceleração. Seus valores iniciais para α ($-\pi/2$, $\pi/2$ e 0, respectivamente) servem apenas para orientação do efetuador.

Como observado nos gráficos, a direção das curvas tomadas pelas juntas 2 e 3 são sempre opostas. Isto acontece porque a rotação destas juntas tem que ser obrigatoriamente em sentidos contrários para que o braço do manipulador gire, pois os ângulos de juntas são obtidos sempre entre os eixos x_{i-1} e x_i . A posição do punho, responsável por esta translação depende, portanto, apenas das três primeiras juntas.

Outro ponto observado é que quando as juntas 2 e 3 atingem respectivamente as velocidades: máxima (e mínima), elas desaceleram (e aceleram) no momento em que atinge a metade da trajetória a ser percorrida pelo efetuador ($t=4s$).

4.3.5 Simulação do Modelo Dinâmico

O modelo dinâmico é apresentado pela *toolbox* em uma matriz *dyn*, como argumento de entrada para a função de modelagem dinâmica. As linhas da matriz representam os elos do manipulador, e as colunas os parâmetros relacionados às variáveis do modelo dinâmico (CORKE, 2008).

A função que calcula quais serão os torques necessários às juntas para atingir uma determinada posição, velocidade e aceleração é *rne*, e foi baseada na formulação de *Newton-Euler* para o cálculo da dinâmica inversa. O torque

necessário para cada junta, em cada ponto da trajetória cartesiana é dado pela matriz τ , cujas colunas correspondem a cada uma das juntas do robô, e as linhas ao torque fornecido para cada articulação com o passar do tempo. Como a aceleração da gravidade influencia no torque para algumas das juntas (2, 3, 4 e 5), este é fornecido pela matriz τ_{ug} (CORKE, 2008).

O torque em função do tempo, para cada uma das juntas influenciadas pela aceleração da gravidade pode ser observado no gráfico 5.

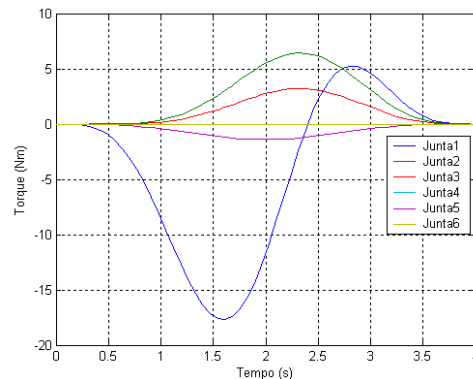
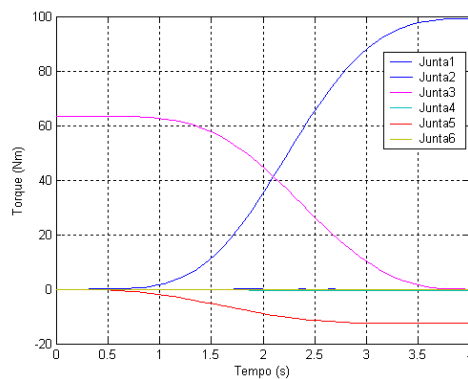


Gráfico 4: Torque em função do tempo **Gráfico 5: Torque em função do tempo influenciado pelas forças: centrípeta e de Coriolis.**

Os perfis de torque dos gráficos 4 e 5 já eram esperados.

No gráfico 4 é notório observar a anti-simetria das duas juntas pertencentes ao mesmo eixo. Isto é necessário, para o caso em que haja uma carga (ferramenta) mais pesada em uma das juntas, para que seja mantido o equilíbrio no centro de massa do elo.

No gráfico 5 observa-se que todas as juntas levam um período de tempo igual a aproximadamente 0.5s (0.3s para a junta 2) para vencer estas forças: centrípeta e de coriolis. Observa-se ainda que a junta 2 precisa superar uma força ainda maior que as outras juntas (possivelmente sua carga é mais pesada, logo a resistência ao movimento é maior). Maior torque significa maior velocidade ou maior carga.

4.4 SIMULAÇÃO DINÂMICA EM SIMUNLINK

Antes de projetar um sistema de controle de um manipulador robótico em *Simunlink* é necessário fazer algumas considerações.

Conforme mencionado, a junta 1 é a de referência para o robô e como ele não é móvel não há necessidade de simulá-la. As juntas 2 e 3 são responsáveis pela direção e posicionamento da ferramenta presente no efetuador. As juntas: 4, 5 e 6 são responsáveis apenas pela orientação de uma ferramenta qualquer presente no terminal do efetuador.

Pela complexidade, maior robustez, inúmeras não-linearidades presentes optou-se por analisar, dentro deste caso de estudo, a junta 3. A análise dinâmica e o controle das demais juntas pode ser feito de forma análoga.

O manipulador de Stanford com seis graus de liberdade pode ser simplificado em um modelo de duas juntas rotativas conforme ilustra a figura 24.

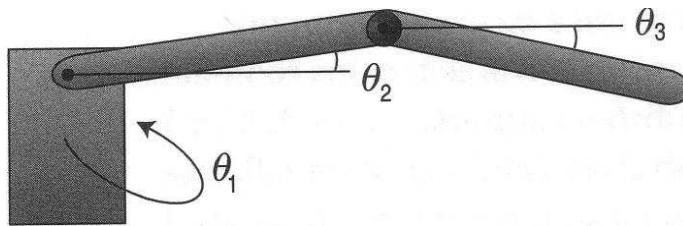


Figura 24: Modelo de um braço simplificado por apenas duas juntas rotativas.
Fonte: ROSÁRIO, 2005.

A partir da parametrização dos termos da equação do movimento $M(q) \ddot{q} + C(q, \dot{q}) \dot{q} + F(\dot{q}) + G(q) = Q$, e substituindo nesta equação a função do torque causado pelo motor nas juntas, obtém-se as funções de transferência de cada uma delas de forma análoga, baseado no estudo dinâmico do modelo geométrico de um pêndulo duplo, como ilustrado na figura 25.

$$m_3 = 4.25 \text{ kg e } l_3 = 0.154;$$

$$\theta_3 = 45^\circ$$

$$T_{m3} = m_3 l_3^2 \ddot{\theta}_3 + m_3 g l_3 \sin \theta_3$$

$$T_{m3} = 0.10 \ddot{\theta}_3 + 4.58$$

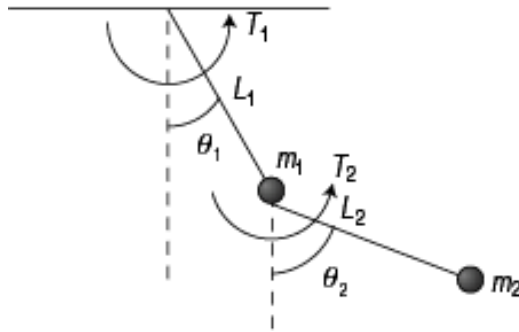


Figura 25: Pêndulo duplo utilizado para equacionar um robô.
Fonte: ROSÁRIO, 2005.

Para realizar a primeira simulação dinâmica, consideram-se a junta inicialmente desacoplada das demais e seu diagrama de blocos em malha aberta, conforme indicado na figura 26.

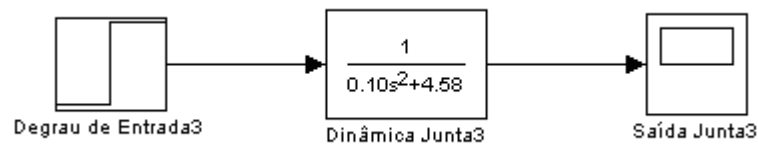


Figura 26: Sistema em Malha Aberta da junta 3.

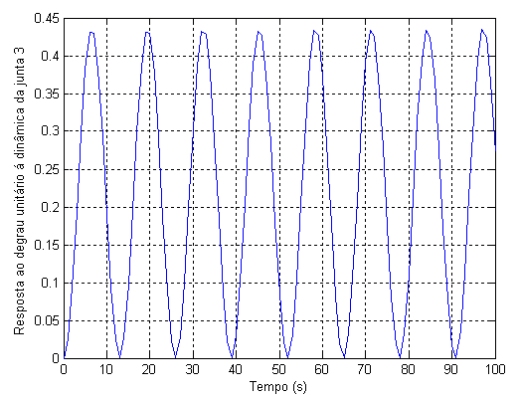


Gráfico 6: Simulação dinâmica da junta 3.

Observando a curva de saída apresentada no gráfico 6 para a simulação da junta em malha aberta percebe-se que o sistema tem uma dinâmica oscilatória e que não alcança a saída definida pelo sinal de referência nem tampouco estabilidade. Logo é necessário aplicar um controlador para que o torque causado pelo motor estabilize, permitindo ao efetuador final alcançar a posição desejada.

4.5 IMPLEMENTAÇÃO DO SISTEMA DE CONTROLE DE UMA JUNTA

Controlar um sistema implica em regular o fluxo de energia conforme a necessidade de um processo. Isto significa torná-lo estável em torno de uma referência desejada, dentro de uma margem de erro considerada aceitável para os padrões do processo em questão. O controlador em malha fechada utiliza o sinal de saída para alterar a ação do sistema de modo que este atinja a especificação de projeto (KATSUHIKO, 2003).

A figura 27 apresenta o diagrama de controle em malha fechada completo para a junta robótica 3.

O sistema de controle é composto pelo: motor cc, redução e carga. As juntas do manipulador são movimentadas pelo motor cc, cuja direção de movimento é determinada pela polaridade da tensão de operação e cada motor tem um controle de realimentação em malha fechada que controla esta direção. Optou-se pela abordagem de controle da tensão aplicada na armadura porque pode utilizar a força contra-eletromotriz como realimentação (SMITH, 1999).

O redutor permite desprezar os efeitos: elástico e de amortecimento da carga quando a relação de redução é alta ($n > 10$). Assim ele reduz a rotação do motor, aumentando o torque na saída. Todavia é necessário considerar o efeito da carga sobre o torque do motor (momento de inércia e coeficiente de atrito viscoso) (ROSÁRIO, 2005).

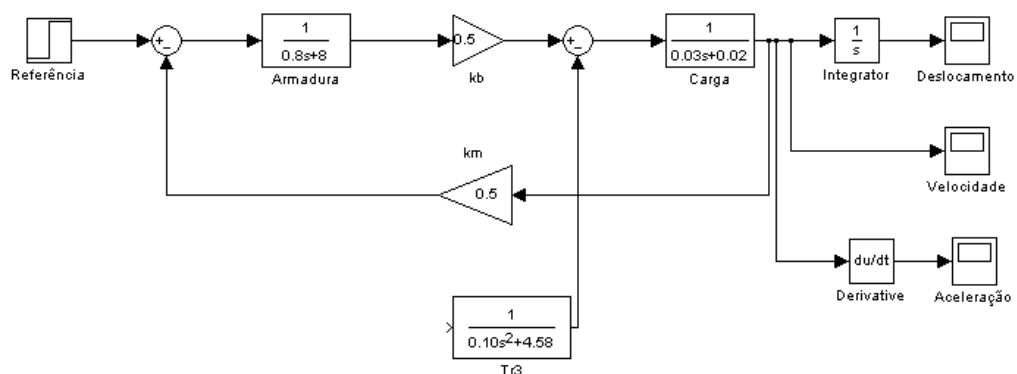


Figura 27: Dinâmica da junta 3

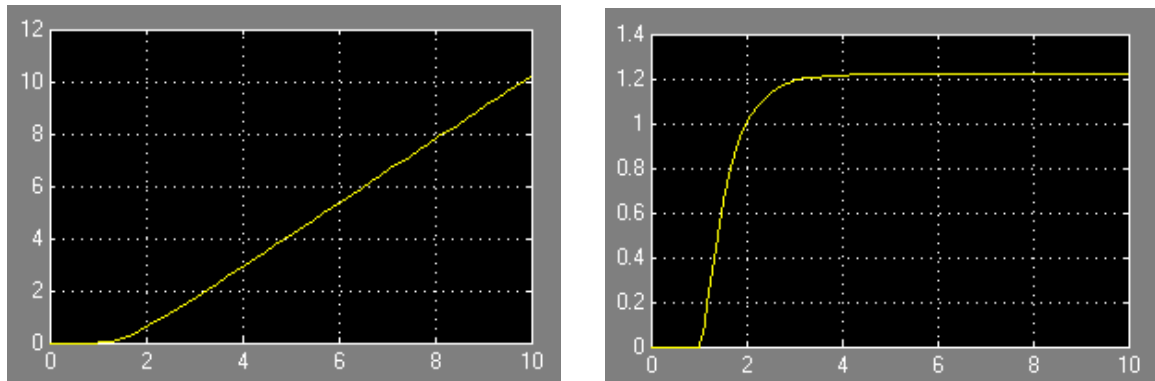


Gráfico 7: Deslocamento e velocidade da junta 3 respectivamente.

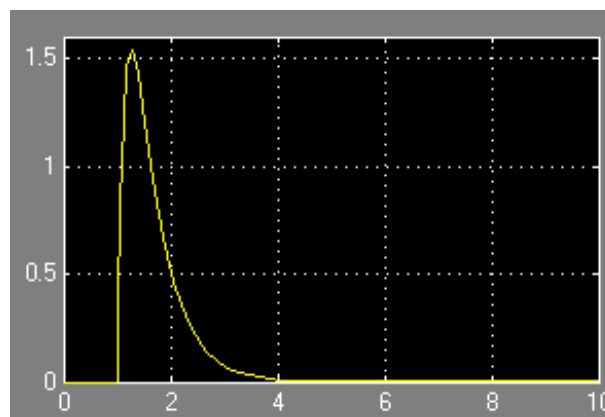


Gráfico 8: Aceleração da junta 3.

4.5.1 Implementação de um compensador a partir da dinâmica da junta

Conforme indicado pelo gráfico 6 a junta 3 não alcançou a saída desejada, bem como estabilidade. Um controlador do tipo “*controle das juntas independentes*” em malha fechada será proposto com base nas configurações inerciais do manipulador.

Cada uma das juntas é controlada como sendo um sistema do tipo *Single Input – Single Output: SISO* (uma entrada – uma saída). A principal vantagem deste tipo de abordagem é um menor esforço computacional e maior confiabilidade do sistema, já que um eventual erro em um ponto do sistema não afetará o desempenho de todo o manipulador, apenas daquela junta em particular. Como desvantagem, destaca-se uma possível deterioração da estratégia de controle realizada em virtude da ferramenta presente no efetuador final. Dependendo da

carga desta ferramenta, o desempenho do controlador pode ser afetado (SPONG, VIDYASAGAR, 1989).

A função do controlador em uma malha de controle é comparar o sinal de saída à um de entrada e corrigir o valor desta diferença. O mais utilizado em robótica industrial é o *PID (proporcional, integral e derivativo)*, cuja vantagem é um desempenho satisfatório quando o sistema controlado é conhecido, linear, invariante no tempo e seus parâmetros corretamente ajustados. Ele é capaz de eliminar o erro de estado estacionário mantendo o ganho proporcional baixo, bem como reduzir eventual perturbação (ROSÁRIO, 2005).

Em um controle de juntas independentes, qualquer efeito de acoplamento causado pelo movimento de outra junta é considerado perturbação (SPONG, VIDYASAGAR, 1989).

A principal desvantagem do PID em controle de juntas robóticas é que este controlador necessita ajuste bastante refinado aos seus parâmetros, além do fato de ser extremamente sensível à dinâmica do sistema. Para que o compensador seja eficiente, é necessário o modelo exato da cinemática inversa para que seja possível a conversão da trajetória especificada pelo operador em uma posição de referência para a junta (ROSÁRIO, 2005).

4.5.2 Controlador PID

A figura 28 ilustra o controle PID de um manipulador robótico com n graus de liberdade. O compensador PID resulta da soma dos componentes: proporcional, integral e derivativo, sendo que cada um destes afetará o sistema de maneiras diferentes. O controle proposto trata apenas da dinâmica da junta em malha fechada, desconsiderando o motor que movimenta o robô.

O controle proporcional reduz o tempo de subida, caracterizado pelo tempo que a junta leva para alcançar posição e orientação de referência. A parcela do controle integrador reduz o erro estacionário e o derivativo é o responsável pela estabilidade em regime permanente (NATIONAL INSTRUMENTS, 2010).

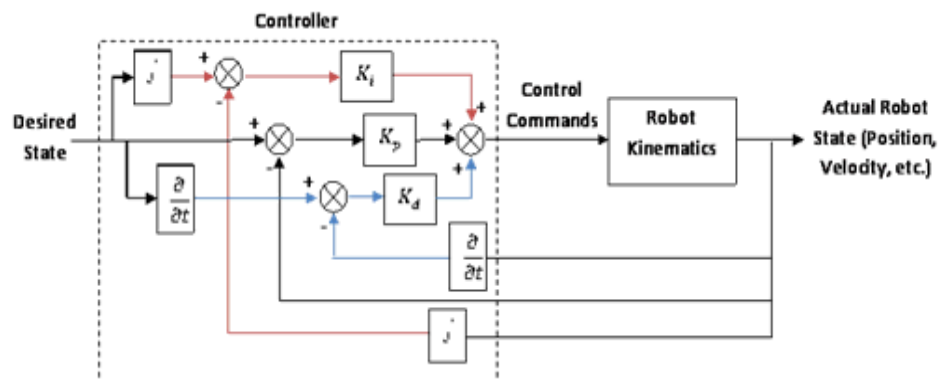


Figura 28: Diagrama de Blocos de um robô com um compensador PID.
Fonte: NATIONAL INSTRUMENTS, 2010.

O controlador do tipo PID apresenta a seguinte relação:

$$U(s) = K_p E(s) + \frac{K_i}{s} E(s) + K_d s E(s) \quad [5]$$

Os parâmetros de controle são: K_p (constante de proporcionalidade associada ao termo proporcional), K_i (constante de proporcionalidade associada ao termo integral e K_d (constante de proporcionalidade associada ao termo derivativo). Estas variáveis devem ser escolhidas em função do desempenho desejado final do sistema a ser controlado. À escolha destes parâmetros chama-se sintonia do PID (ERIG,2009).

Ziegler e Nichols propuseram regras para a sintonia de controladores através do ajuste das constantes de proporcionalidade: K_i , K_p e K_d , baseadas na resposta do sistema quando submetido a um degrau de referência, ou no valor de K_p que torna o sistema estável quando apenas uma ação proporcional é aplicada. Deve-se ressaltar que as regras de ZN fornecem apenas uma estimativa dos parâmetros de controle, servindo como um ponto de partida à sintonia fina, sendo necessárias, portanto, várias simulações até encontrar o modelo mais adequado ao sistema. (OGATA, 2003).

4.5.3 Sintonia do controlador.

Para a sintonia de um controlador proposta por ZN, são apresentados dois métodos. Neste documento utilizou-se o segundo método, pois o primeiro só pode ser aplicado quando a simulação da dinâmica da junta obtida for uma curva em S (logo, dinâmica não-oscilatória). Como foi observado na resposta dinâmica da junta 3 (gráfico 7), ela apresenta dinâmica oscilatória, sendo aplicável apenas o segundo método.

O primeiro passo é admitir apenas a componente proporcional do compensador e simular, a partir de $K_p=0$ o sistema para um degrau unitário. Este valor deve ser incrementado até que a saída obtida exiba uma oscilação sustentada.

Como o sistema em malha aberta já era de natureza oscilatória, foi possível encontrar o chamado P_{cr} (período correspondente ao ganho crítico) de 1.4s, para K_{cr} (ganho crítico) de 0.1. Com base nestes parâmetros é feita sintonia do controlador e os demais parâmetros são calculados conforme a tabela 4.

Tabela 4: Regra de sintonia de Ziegler-Nichols.

Tipo de controlador	K_p	T_i	T_d
P	$0,5 K_{cr}$	∞	0
PI	$0,45 K_{cr}$	$(1/1,2) P_{cr}$	0
PID	$0,6 K_{cr}$	$0,5 P_{cr}$	$0,125 P_{cr}$

FONTE: OGATA, 2003.

De acordo com essa tabela, os valores iniciais para o ajuste do compensador são:

$$K_p=0.57$$

$$T_i=0.5; K_i=2.105$$

$$K_d=T_d=0.175$$

De posse destes valores, substitui-se o controle inicial dado apenas pela componente proporcional pelo controle PID, conforme indicado no diagrama de blocos da figura 29.

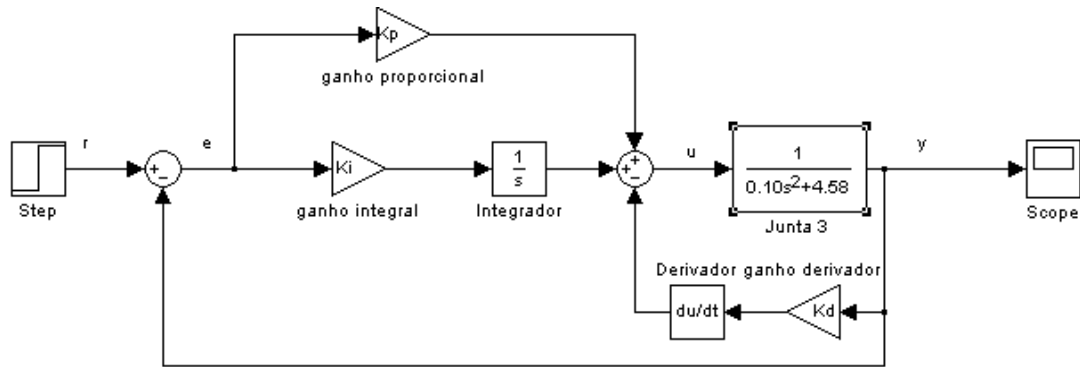


Figura 29: Diagrama de Blocos da junta 3 em malha fechada com o PID.

A resposta do sistema á estes parâmetros não apresentou uma dinâmica satisfatória: o sistema estabilizou bem abaixo da saída unitária e continuou oscilando.

Após algumas simulações, alterando os parâmetros do controlador, o PID que apresentou resposta satisfatória foi:

$$K_p=5.8$$

$$K_i=10$$

$$K_d=5$$

Simulando a resposta dinâmica para a junta robótica 3, sem o PID e com o compensador as curvas demonstradas no gráfico 10:

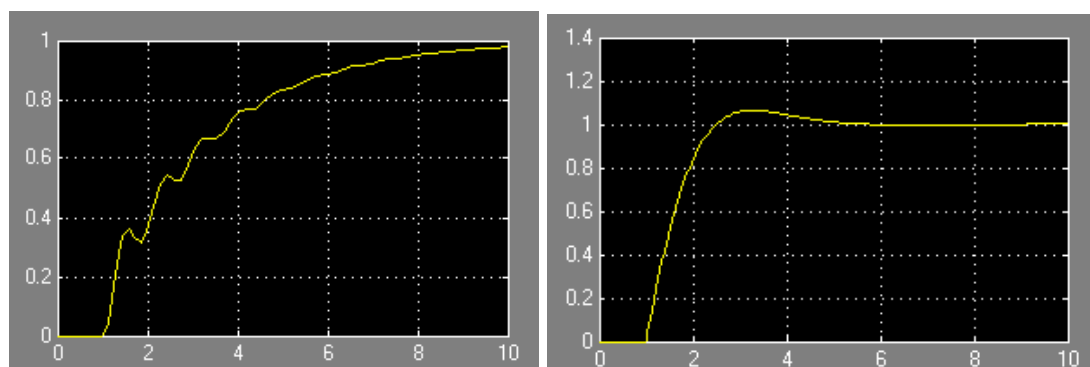


Gráfico 9: Dinâmica da junta 3, sem e com o controlador PID, respectivamente.

Ao comparar os dois gráficos, percebe-se que o controlador PID sintonizado obtém desempenho quase satisfatório, deixando a desejar quanto á demora em acomodar e estabilizar o sistema, bem como uma pequena oscilação na subida. Após a adição do controlador o sistema atingiu a estabilidade, perdeu a

característica oscilatória e atingiu a referência desejada com um mínimo sobressinal.

O aumento do ganho proporcional deixou o sistema mais rápido e diminuiu o erro em regime estacionário, mas como consequência, ocasionou pequeno sobressinal aparentemente aceitável. Este tipo de análise e tomada de decisão implica nas especificações do projeto de controle. Logo a presença ou não de sobressinal depende da tarefa do manipulador e se isto acarreta algum tipo de transtorno para a atividade / produção ou não.

Com a componente integral, o sistema teve apenas o nulo em regime permanente.

O aumento da constante derivativa foi responsável por tornar o sistema estável eliminando a oscilação do sistema durante o tempo de subida da curva.

Esta sintonia foi feita visando especialmente eliminar a característica oscilatória do sistema e sua instabilidade. É pertinente salientar que na prática o que definirá a alteração destes parâmetros será a especificação de controle de acordo com a tarefa que cada manipulador deverá executar.

4.6 CONSIDERAÇÕES FINAIS DO CAPÍTULO

Este capítulo apresentou a ferramenta de programação, simulação e controle robótica especialmente criada para o MATLAB.

Foi enfatizado que, de acordo com as matrizes de transformação homogêneas resultantes, a partir de um vetor de tempo fornecido pelo usuário este pode simular e controlar a trajetória percorrida, a velocidade e a aceleração das juntas de robô.

Este capítulo apresentou principalmente o caso de estudo de um robô real: o braço de Stanford. Este se caracteriza por ter um sistema elétrico com seis eixos articulados e projetados para seguir uma trajetória qualquer, pré-definida por um operador, utilizando a anatomia de um braço humano.

Para realizar a simulação dinâmica analisou-se apenas a junta 3. Esta juntamente com a 2, são as responsáveis pela direção e posicionamento da ferramenta presente no efetuador.

As outras juntas: 4, 5 e 6 são responsáveis apenas pela direção da ferramenta no terminal do efetuator. Logo basta analisar uma junta e as demais serão de forma análoga.

Para efeito desta análise, obteve-se a resposta do sistema da junta para um degrau de entrada em malha aberta. Como a resposta era um sistema oscilatório e instável foi proposto um sistema de controle em malha fechada.

5 CONSIDERAÇÕES FINAIS

Dentro dos objetivos inicialmente propostos consideram-se os resultados alcançados neste estudo extremamente satisfatórios.

De início foi apresentada uma introdução acerca dos manipuladores robóticos, seus principais conceitos e quais são as leis que regem esta ciência.

Após isto foi descrito seu princípio de funcionamento, como é composto um manipulador robótico, bem como um estudo da cinemática: direta e inversa e da dinâmica dos efetadores. A partir do estudo das cinemáticas foram apresentados quais os tipos de controle que podem ser realizados sobre o robô.

De posse de toda a fundamentação teórica necessária foi simulada a variação dos parâmetros do robô de Stanford no aplicativo de robótica do MATLAB, além de uma breve análise dos resultados obtidos. Uma das maiores vantagens ao utilizar um simulador robótico é devido à programação *offline*, onde os programas são previamente construídos para depois ser carregados no robô, permitindo assim que todos os testes e análises sejam executados antes, em um ambiente virtual. Isto é extremamente vantajoso em casos onde há riscos para a vida do operador e do robô, como por exemplo, robôs que realizam medição de temperatura em vulcões, robôs de solda submarina, entre outros.

Verificou-se que este *software*, largamente utilizado no ambiente acadêmico é extremamente valioso no que tange à análise do comportamento cinemático e dinâmico. Entretanto não seria recomendável utilizá-lo para simulação durante o processo produtivo de uma indústria, pois se trata de um *software* bastante dedicado e necessita de operadores ambientados à ferramenta, mas principalmente que estes saibam o que estão simulando e qual a implicação matemática.

Logo se o objetivo for simular apenas o movimento em si, outros programas aparentam melhor utilidade, já que muitos deles já podem ser operados por *joystick*, ficando mais fácil e rápido para o operador.

Se o objetivo da simulação for o controle do manipulador, esta *toolbox* é perfeitamente aplicável no ambiente industrial, desde que os operadores sejam profundos conhecedores da ferramenta, e que haja tempo suficiente para estas simulações, não atrapalhando assim o andamento da produção.

Caso contrário seria mais apropriado utilizar outro simulador onde o manuseio é direto (*joystick*, teclado entre outros) e cuja interface gráfica apresente o robô exatamente como ele é, e não apenas um manipulador geométrico.

5.1 SUGESTÕES DE TRABALHOS FUTUROS

A economia brasileira nunca esteve tão fortemente aquecida. Dados da Associação Brasileira da Indústria de Máquinas e Equipamentos: Abimaq indicam que o faturamento do setor (máquinas industriais e manipuladores robóticos) no país subiu 14,4% no primeiro semestre de 2010 em comparação com o mesmo período do ano passado. A Associação Brasileira dos Importadores de Máquinas e Equipamentos Industriais: Abimei prevê crescimento de 50% nas vendas em 2010, chegando aos US\$ 2,2 bilhões. (JORNAL DO COMÉRCIO, 2010).

Quanto maior o crescimento industrial, maior a necessidade de ferramentas mais rápidas e eficientes ao processo produtivo. O mercado de trabalho está sedento de ferramentas e profissionais com este perfil. A sugestão deixada aqui seria aprimorar ainda mais a ferramenta discutida neste trabalho, visando unir o alto nível e a facilidade na utilização dos outros *softwares* (facilidade de manipulação dos robôs devido ao uso de periféricos como *joysticks*) com a parte de controle tão precisamente realizada no MATLAB.

REFERÊNCIAS

ABB ROBOTICS BRAZIL. Disponível em:

<<http://www.abb.com.br/abblibrary/DownloadCenter/?showresultstab=true&categoryid=9AAC129474>> ACESSO EM 09 DE MAR DE 2011, ÀS 21h17min.

BECKER, Marcelo. SEM 0317 - Aula 08: **Dinâmica de manipuladores robóticos.** Brasil: Universidade de São Paulo, 2010.

BIANCHI, Reinaldo. **Notas de Aula: Robótica.** Brasil: Universidade Federal Itajubá, 2010.

CARLOS E. FERREIRA, João. Capítulo 5: **Robótica industrial.** Brasil: Universidade Federal de Santa Catarina, 2005.

CÉSAR CHARÃO DOS SANTOS, Mauro. **Avaliação do uso de realidade virtual na robótica.** Brasil: Pontifícia Universidade Católica do Rio Grande do Sul, 2006.

CARRARA, Valdemir. **Apostila de robótica.** Brasil: Universidade de Braz Cubas, 2006.

CLOSE, C. M., FREDERICK, D.K. **Modeling and Analysis of Dynamic Systems.** Estados Unidos: Houghton Mifflin Company, 1989.

FERREIRA ROMANO, Vítor. **Robótica Industrial: Aplicação na Indústria de Manufatura e de Processos.** Brasil: Editora Edgar Blucher, 2009.

GIMENO LATRE, Luis. **Modelagem e controle de posição de robôs.** Brasil: Revista da SBA: Sociedade Brasileira de Automática, Volume 2, 1988.

HARRIS, Tom. **How Stuff Works.** Estados Unidos: 2009.

I. CORKE, Peter. **Robotics toolbox for MATLAB.** Release 8. Dezembro, 2008.

J. CHAPMAN, Stephen. **Programação em MATLAB para engenheiros.** Austrália: Editora Thomson, 2003.

JORNAL DO COMÉRCIO. Disponível em:
<http://jcrs.uol.com.br/site/noticia.php?codn=36129&codp=36117&codni=3>
 ACESSO EM 22 DE NOV DE 2010, ÀS 8h10min.

LIMA, Carlos Raimundo Erig. **Fundamentos de Controle.** Especialização em Automação Industrial, Universidade Tecnológica do Paraná, Curitiba: 2009.

MATLAB E SIMUNLINK. Disponível em:
<http://www.mathworks.com>, ACESSO EM 09 DE MAR DE 2011, ÀS 21h34min.
http://www.mathworks.com/solutions/?s_cid=global_nav, ACESSO EM 05 DE JUN DE 2011, ÀS 16h20min.

M. F. SANTOS, Vitor. **Robótica industrial.** Portugal: Editora Universidade de Aveiro, 2004.

MAURÍCIO ROSÁRIO, João. **Princípios de Mecatrônica.** Brasil: Editora Prentice Hall, 2005.

MECATRÔNICA ATUAL. Brasil, Disponível em:
<http://www.mecatronicaatual.com.br/secoes/leitura/404#self>, ACESSO EM 08 de ago. de 2010, às 20h36min.

MENDES LOPES, Antônio. **Robótica Industrial.** Portugal. FEUP – Faculdade de Engenharia da Universidade de Porto. 2001 / 2002.

NATIONAL INSTRUMENTS. National Instruments Products. Disponível em:
digital.ni.com/worldwide/brazil.nsf/web/all/9B327060A95A1EDD8625754500738DB8
 8>. ACESSO EM 08 DE NOV DE 2010, ÀS 03h20min.

OGATA, Katsuhiko. **Engenharia de Controle Moderno.** Brasil: Editora Prentice Hall, Quarta Edição, 2003.

R. KURFESS, Thomas. **Robotics and automation handbook.** Estados Unidos: CRC Press, 2005.

ROBERTO DE PIERI, Edson. **Curso de robótica móvel.** Brasil, Programa de pós - graduação em engenharia elétrica, 2002.

ROBBINS, Anthony. **Desperte seu gigante interior**. Brasil, 2010.

ROSSATO BARBUTO, Daniel. **Desenvolvimento de um sistema aberto para ensino de robôs manipuladores**. Brasil: Escola Politécnica da Universidade de São Paulo, 2009.

SANDIN E., Paul **Robot mechanisms and mechanical devices**. Estados Unidos: Editora McGraw-Hill, 2003.

SCHILLING, Robert Jr. **Fundamentals of Robotics Analysis and Control**. Estados Unidos. Englewood Cliffs Prentice Hall, 1990.

SEDRA, Smith. **Microeletrônica Básica**. Quarta Edição. 1999.

SESISC Disponível em:

http://jet.sesisc.org.br/conteudo/rob_u2_1_OSrobosIndustriais.php#self,
ACESSO EM 08 DE AGO DE 2010, ÀS 20h45min.

SICILIANO, Bruno, KHATIB, Oussama. **Handbook of robotics**, Estados Unidos, Editora Springer, 2008.

SPONG, M. W. and VIDYASAGAR, M. **Robot, Dynamics and Control**, John Wiley and Sons., 1989.

STANFORD. Disponível em:

<http://ai.stanford.edu/> ACESSO EM 10 DE MAR DE 2011, ÀS 22h17min.

TAVARES PIMENTA, Thiago. **Controle de manipuladores robóticos**. Brasil: Pontifícia Universidade Católica do Rio de Janeiro, 2009.

UOL <http://tecnologia.uol.com.br/ultnot/2007/10/01/ult4213u146.jhtm> ACESSO EM 05 DE JUN DE 2011, ÀS 20h18min.

VENTURA BAYAN HENRIQUES, Renato; Capítulo 6: **Programação e simulação de robôs**. Brasil.

WIKIPÉDIA <http://pt.wikilíngua.com/es/Rob%C3%B4 industrial> > ACESSO EM 20 DE SET DE 2010, ÀS 18h10min.

WORDPRESS, Disponível em:

<<http://errorstream.wordpress.com/2008/03/17/roboparty-microsoft-robotics-studio/>>, Acesso em 28 de jun. de 2010, às 09h00min.

ANEXOS

Anexo 1 – Algoritmo Resumido de Denavit-Hartenberg para cinemática direta.

D1: Obter o sistema de coordenadas da base: Estabelecer um sistema ortonormal de coordenadas (X_0, Y_0, Z_0) na base de suporte com o eixo Z_0 colocado ao longo do eixo de movimento da junta 1 apontando para o ombro do braço do robô. Os eixos X_0 e Y_0 podem ser convenientemente estabelecidos e são normais ao eixo Z_0 .

D2: Inicializar e iterar: Para cada $i, i = 1, \dots, N - 1$, efetuar passos D3 até D6.

D3: Estabelecer o eixo das juntas: Alinhar Z_i com o eixo de movimento (rotação ou translação) da junta $i + 1$. Para robôs com configurações de braço esquerdo-direito, os eixos Z_1 e Z_2 são apontados sempre para o ombro e o tronco do braço do robô.

D4: Estabelecer a origem do i -ésimo sistema de coordenadas: Situar a origem do i -ésimo sistema de coordenadas na interseção dos eixos Z_i e Z_{i-1} ou na interseção da normal comum entre os eixos Z_i e Z_{i-1} e o eixo Z_{i-1} .

D5: Estabelecer o eixo X_i : Estabelecer $X_i = \pm(Z_{i-1} \times Z_i) / \|Z_{i-1} \times Z_i\|$ ou ao longo da normal comum entre os eixos Z_i e Z_{i-1} quando eles forem paralelos.

D6: Estabelecer o eixo Y_i : Determina-se $Y_i = \pm(Z_i \times X_i) / \|Z_i \times X_i\|$ para completar o sistema de coordenadas (estender os eixos Z_i e X_i se necessário, para os passos D9 a D12).

D7: Estabelecer a direção do sistema de coordenadas: Normalmente a n -ésima junta é uma junta rotativa. Estabelecer Z_n ao longo da direção do eixo Z_{n-1} apontando para fora do robô. Estabelecer X_n de modo que ele seja normal tanto ao eixo Z_{n-1} como ao Z_n . Determinar y_n para completar o sistema de coordenadas.

D8: Encontrar os parâmetros das juntas e links: Para cada $i, i = 1, \dots, n$, efetuar passos D9 ao D12.

D9: Encontrar d_i : d_i é a distância da origem do $(i - 1)$ -ésimo sistema de coordenadas até a interseção entre o eixo Z_{i-1} e o eixo X_i ao longo do eixo Z_{i-1} . Ela é a variável de junta quando a junta i é prismática.

D10: Encontrar a_i : a_i é a distância da interseção entre o eixo Z_{i-1} e o eixo X_i até a origem do i -ésimo sistema de coordenadas ao longo do eixo X_i .

D11: Encontrar θ_i : θ_i é o ângulo de rotação entre os eixos X_{i-1} e X_i sobre o eixo Z_{i-1} . Esta é a variável de junta quando a junta é rotacional.

D12: Encontrar α_i : α_i é o ângulo de rotação entre os eixos Z_{i-1} e Z_i no eixo X_i .

Fonte: ROSÁRIO, 2005.

Anexo 2 – Rotinas desenvolvidas em MATLAB

```

% alpha A      theta  D      sigma  m      rx      ry      rz      lxx      lyy      lzz      lxy
      lyz      lxz      Jm      G
L{1}=link([-pi/2 0      0      0.412  0      9.29  0 .0175  -0.1105  0.276  0.255  0.071  0
           0      0      0.953  1]);
L{2}=link([ pi/2 0      0      0.154  0      5.01  0 -1.054  0      0.108      0.018
           0.100  0      0      0      2.193  1]);
L{3}=link([-pi/2 0      0      0      1      4.25  0  0      -6.447      2.51  2.51
           0.006  0      0      0      0.782  1]);
L{4}=link([-pi/2 0      0      0      0      1.08  0 0.092 -0.054  0.002  0.001  0.001  0
           0      0      0.106  1]);
L{5}=link([pi/2 0 0      0      0      0.63  0 0      0.566  0.003  0.003  0.0004  0      0
           0      0.097  1]);
L{6}=link([ 0  0 0 0.263  0      0.51  0 0      1.554  0.013  0.013  0.0003  0      0      0
           0.020  1]);
stan=robot(L, 'Stanford');
plot(stan, [0 0 0 0 0 0]);
stan.name = 'Braço de Stanford';
T=fkine(stan, [0 0 0 0 0 0]);

%Para demonstrar a cinematica direta, rot. e trans. a partir de certas
%coordenadas
q = [0 0 0 pi/2 3*pi/4 2*pi/8];
q = [0 -pi/4 -pi/4 0 pi/8 0];
qz= [0 0 0 0 0 0];
qz = [0 0 0 0 0 0];
qr = [0 pi/2 -pi/2 0 0 0];
qstretch = [0 0 -pi/2 0 0 0];
T=fkine (stan, qz);
plot(stan, q);
qi=ikine(stan, T);
J1 = jacob0 (stan,q);
J2 = jacob0 (stan,qz);
J = jacobn (stan,q);
d=det(J);
vel=[0.25 0 0 0 0 0]';
qvel=inv(J)*vel;
a=rank( jacobn(stan, qr) );

% % %Controle/Planejamento da trajetoria
t = [0: .070: 4]';
q = jtraj(qz, qr, t);
plot (t, q(:,1:3)); % plota as trajetorias
grid on;
Ttg = fkine(stan, q);
[q,qd,qdd] = jtraj(qz, qr, t);
plot(t, qd(:,1:3)); %plota as velocidades das juntas
grid on;
plot(t,qdd(:,1:3)); %plota as acelerações
grid on;

% %Dinamica
tau = rne(stan, q, qd, qdd);
taug=gravload(stan, q); %torque sofrendo a influencia de G

```

```

plot(t, taug(:,1:6));
grid on;
c = rne(stan, q, qd, zeros(size(q)), [0;0;0]);
plot(t, c(:,1:6));
grid on;
a=accel (stan, qz, zeros(1,6), zeros(1,6));
tic;
[t q qd]=fdyn(stan, 0, 2);
x=[t q qd];
plot(stan,x(:,1:6));

%Simulação dinamica da junta 3 desacoplada em malha aberta

%Junta 3
num=[1];
den=[0.10 0 4.58];
t=0:1:100;
c=step(num,den,t);
plot(t,c);
grid on;

```