

UNIVERSIDADE FEDERAL DE SANTA CATARINA

**PROGRAMA DE PÓS-GRADUAÇÃO
EM ENGENHARIA ELÉTRICA**

**CONTROLE DE IMPEDÂNCIA EM ROBÔS
MANIPULADORES.**

Dissertação submetida à
Universidade Federal de Santa Catarina
Como parte dos requisitos para a
Obtenção do grau de Mestre em Engenharia Elétrica

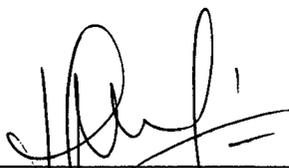
CESAR AUGUSTO MOTEJUNAS

Florianópolis, novembro de 2002

CONTROLE DE IMPEDÂNCIA EM ROBÔS MANIPULADORES

Cesar Augusto Motejunas

‘Esta dissertação foi julgada adequada para obtenção do Título de Mestre em Engenharia Elétrica, Área de Concentração em Automação e Sistemas, e aprovada em sua forma final pelo programa de Pós-graduação em Engenharia Elétrica da Universidade Federal de Santa Catarina.’



Prof. Edson Roberto de Pieri, Dr.

Orientador



Prof. Edson Roberto de Pieri, Dr.

Coordenador do Programa de Pós-Graduação em Engenharia Elétrica

Banca Examinadora:



Prof. Edson Roberto de Pieri, Dr.

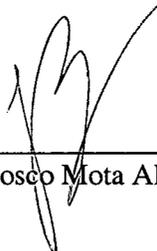
Presidente



Prof. Raul Guenter, Dr.



Prof. Ubirajara Franco Moreno, Dr.



Prof. João Bosco Mota Alves, Dr.

Aos meus pais, sem os quais não cresceria,
À minha esposa, sem a qual não viveria
E ao meu filho, sem o qual não me completaria..

AGRADECIMENTOS

Quero agradecer a minha esposa Magna, amiga e solidária nas intermináveis horas de estudo, que contribuiu muito, com sua compreensão e carinho, para que este trabalho pudesse ser concluído.

Aos meus pais, Paulo e Beatriz, por todas as horas dedicadas a minha educação e saúde. E mesmo estando distantes, em nenhum momento deixaram faltar uma palavra de apoio, um conselho, uma força, tão necessários, quando estamos diante de uma tarefa difícil.

Agradeço ainda aos professores da UFSC que de uma forma toda especial, conseguem transmitir os conhecimentos adquiridos em vários anos de estudo e dedicação. Em especial, agradeço ao meu orientador professor Edson de Pieri, por me orientar durante esta etapa de minha vida.

Gostaria ainda de agradecer ao meu filho Victor, que mesmo pequeno, consegue a cada dia me mostrar que devemos nos ater a pequenos detalhes, que as novidades da vida podem estar tão próximas, e que não devemos nos abater por nada. Temos uma energia muito grande, e devemos aproveitá-la ao máximo a cada dia.

Agradecimentos especiais para o colega e doutorando Fernando Passold, pelo apoio e paciência, durante sábados e feriados, no auxílio do desbravamento da linguagem Xoberon.

E a todos, que de forma direta ou indireta contribuíram para o aperfeiçoamento deste trabalho, com sugestões e críticas construtivas,

Meu muito obrigado!

Resumo da Dissertação apresentada à UFSC como parte dos requisitos necessários para a obtenção do grau de Mestre em Engenharia Elétrica.

CONTROLE DE IMPEDÂNCIA EM ROBÔS MANIPULADORES

Cesar Augusto Motejunas

Novembro / 2002

Orientador: Edson Roberto de Pieri, Dr.

Área de Concentração: Automação e Sistemas.

Palavras-chave: Robôs manipuladores, controle de força, controle de impedância.

Número de páginas:

Este trabalho tem como objetivo estudar o uso do controle de impedância no controle de um robô manipulador rígido.

No trabalho é abordada a aplicação do controle de impedância implementado a partir de um controle de posição, através da inclusão de uma realimentação de esforço/torque, de maneira a tornar a impedância apresentada pelo manipulador exclusivamente dependente do controle.

O desempenho do controle de impedância na forma desenvolvida é verificado em simulações numéricas para o caso do controle de um manipulador SCARA, considerando a sua estrutura completa. São simuladas movimentações no espaço livre e em contato com o meio, para a situação real e com variação paramétrica.

O controle de impedância é implementado no manipulador SCARA Inter do laboratório de Robótica da Universidade Federal de Santa Catarina, utilizando a linguagem de programação Xoberon. Seu desempenho é comparado com os resultados obtidos nas simulações.

Abstract of Dissertation presented to UFSC as a partial fulfillment of the requirements for the degree of Master in Electrical Engineering.

IMPEDANCE CONTROL OF ROBOTIC MANIPULATORS

Cesar Augusto Motejunas

November/2002

Advisor: Edson Roberto de Pieri, Dr.

Area of Concentration: Automation and Systems

Keywords: Robot manipulators, force control, impedance control, control systems

Number of Pages: .

The objective of this work is the evaluation of the employment of Impedance Control in the control of rigid robotic manipulator.

The text discusses the application of the impedance control implemented from a position control, through the inclusion of a strain/torque feedback, just in a way to make the impedance of the manipulator to depend only on the control.

The performance of the impedance control is verified in numerical simulations, in the case of control of a SCARA robot, taking into account its complete structure. Motions in the free space and in contact with environment are both simulated, for the real case and also with parametric variation.

The impedance control is implemented on the SCARA robot. This robot belongs to the Robotics Lab, at the Federal University of Santa Catarina. The impedance controller performance is compared with simulation results.

SUMÁRIO

Resumo	v
Abstract	vi
Listas	viii
1 Introdução	1
1.1 Histórico da Robótica.....	1
1.2 Classificação e Definição dos Robôs Industriais	5
1.3 Robôs manipuladores na Indústria	7
1.4 Controle de Robôs Manipuladores.....	9
1.5 Trabalhos Atuais em Controle de Impedância.....	12
1.6 Objetivos da Dissertação.....	15
1.7 Justificativa	15
1.8 Estrutura da Dissertação.....	16
2 Algoritmos Convencionais de Controle de Força.....	18
2.1 Caracterização do Problema de Controle de Força	18
2.2 Utilização da Realimentação de Força.....	21
2.2.1 Controle de Flexibilidade.....	21
2.2.2 Analogia Mecânica-Elétrica	22
2.2.3 Impedância e Admitância Mecânicas	23
2.3 Controle de Impedância	24
2.3.1 Impedância de uma Junta com Controle de Posição.....	24
2.3.2 Controle de Impedância de uma Junta Isolada	27
2.3.3 Controle de Impedância em um Robô com várias Juntas.....	33
2.4 Controle de Rigidez.....	35
2.4.1 Controle de Rigidez Aplicado a uma Junta	35
2.4.2 Controle de Rigidez Aplicado a um Robô Manipulador	39
2.5 Controle de Amortecimento.....	42
2.6 Estudo do Controlador de Impedância Implementado a Partir do Controle com Compensação pela Dinâmica Inversa	44
3 Simulações do Controle de Impedância para um Robô SCARA.....	49
3.1 Modelagem do Robô SCARA.....	49
3.2 Implementação do Controle de Impedância.....	50

3.3 Resultados Obtidos nas Simulações.....	53
3.3.1 Movimentação sem Restrição e sem Variação Paramétrica	54
3.3.2 Movimentação com Restrição Partindo em Contato com o Meio e sem Variação Paramétrica	56
3.3.3 Movimentação sem Restrição com Variação Paramétrica	60
3.3.4 Movimentação com Restrição Partindo em Contato com o Meio e com Variação Paramétrica	63
3.4 Implementação do Controle de Impedância em um Robô SCARA.....	67
3.5 Implementação do Controle de Impedância Para o Seguimento de Trajetória com Obstáculo.....	68
3.6 Resultados e Discussões.....	71
3.6.1 Simulação de Movimentação em Contato com o Meio e Presença de Obstáculo.....	75
3.7 Observações	77
4 Conclusões e Perspectivas	79
Apêndice A	81
A.1 Modelagem Cinemática e Dinâmica do Robô Scara	81
A.2 Parâmetros do Robô SCARA	81
A.3 Modelagem Cinemática	82
A.3.1 Cinemática Direta	83
A.3.2 Cinemática Inversa	85
A.4 Modelagem Dinâmica	86
Apêndice B – Programas Desenvolvidos Para o Matlab e Simulink.....	88
B.1 Diagramas para o Ambiente Simulink.....	88
B.2 Funções Desenvolvidas para o Matlab.....	92
Apêndice C – Programas Desenvolvidos Para o XOBIRON	109
5 Referências Bibliográficas	118

LISTA DE FIGURAS

Figura 1.1 – Peça a ser polida e ferramenta de polimento fixada na extremidade de um robô puma	13
Figura 1.2 – Trajetória definida a partir do ponto inicial, final e passo.....	13
Figura 1.3 – Esquema de controle proposto para controle de posição de uma junta de acionamento hidráulico	14
Figura 2.1 – Equivalência Mecânica Elétrica	19
Figura 2.2 – Modelo mecânico e elétrico do conjunto atuador – junta.....	21
Figura 2.3 – Esquema elétrico equivalente do conjunto junta – atuador	22
Figura 2.4 – Diagrama de blocos de uma junta com controle de posição/velocidade	22
Figura 2.5 – Esquema elétrico equivalente da junta sobre o controle de posição/ Velocidade.....	22
Figura 2.6 – Modelo mecânico e elétrico do ambiente.....	23
Figura 2.7 – Modelo elétrico equivalente do ambiente.....	23
Figura 2.8 – Esquema elétrico equivalente da junta em contato com o meio.....	23
Figura 2.9 – Esquema elétrico equivalente da junta com controle de flexibilidade em contato com o meio	24
Figura 2.10 – Junta com controle de flexibilidade em contato com o meio	25
Figura 2.11 – Diagrama equivalente da junta com controle de flexibilidade em contato com o meio	25
Figura 2.12 – Diagrama de blocos da junta com controlador utilizando realimentação de aceleração, velocidade e posição.....	27
Figura 2.13 – Diagrama de blocos da junta com controle de flexibilidade modificado	28
Figura 2.14 – Diagrama de blocos da junta com controle de flexibilidade sem medida de aceleração	28
figura 2.15 – Controle de impedância em um robô com várias juntas	31
figura 2.16 – Esquema de controle de rigidez aplicado em uma junta em contato com o meio	33
figura 2.17 – Diagrama em blocos do controle de rigidez no espaço cartesiano.....	37
figura 2.18 – Esquema de controle de rigidez no espaço cartesiano segundo [WHITNEY,85]	38
figura 2.19 – Diagrama em blocos do controle de amortecimento aplicado a uma junta...	39

figura 3.1 – Representação das juntas de um robô do tipo SCARA.....	47
figura 3.2 – Diagrama de blocos representando a dinâmica de um manipulador sobre controle de impedância	48
figura 3.3 –Diagrama de blocos do simulador de controle de impedância aplicado em um robô do tipo SCARA.....	49
Figura 3.4 – Valores iniciais e finais nas direções X e Y	49
Figura 3.5 – Posição do efetuador final em relação a direção X em função do tempo na movimentação sem contato com o meio.....	51
Figura 3.6 – Posição do efetuador final em relação a direção Y em função do tempo na movimentação sem contato com o meio.....	52
Figura 3.7 – Posição do efetuador final em relação as direções X e Y em movimento sem contato com o meio.....	52
Figura 3.8 – Erro nas direções X e Y, para o robô em movimento em contato com o meio	53
Figura 3.9 – Gráfico ampliado da posição desejada em relação a posição atingida e o meio	54
Figura 3.10 – Posição do efetuador final em relação a direção X em função do tempo na movimentação em contato com o meio.....	55
Figura 3.11 – Posição do efetuador final em relação a direção Y em função do tempo na movimentação em contato com o meio.....	55
Figura 3.12 – Força de contato na direção x em função do tempo	56
Figura 3.13 – Força de contato na direção y em função do tempo	56
Figura 3.14 – Tabela contendo os desvios impostos na fase de simulação	57
Figura 3.15 – Posição do efetuador final em relação a direção X em função do tempo na movimentação sem contato com o meio e com variação paramétrica.....	58
Figura 3.16 – Posição do efetuador final em relação a direção Y em função do tempo na movimentação sem contato com o meio e com variação paramétrica.....	58
Figura 3.17 – Posição do efetuador final em relação as direções X e Y em movimento sem contato com o meio e com variação paramétrica.	59
Figura 3.18 – Erro nas direções X e Y em movimento sem contato com o meio e com variação paramétrica	59
Figura 3.19 – Erro nas direções X e Y, para o robô em movimento em contato com o meio e variação paramétrica	60

Figura 3.20 – Posição do efetuador final em relação as direções X e Y em movimento em contato com o meio e variação paramétrica.....	61
Figura 3.21 – Força de contato na direção X em função do tempo para a movimentação do robô em contato com o meio e variação paramétrica	62
Figura 3.22 - Força de contato na direção Y em função do tempo para a movimentação do robô em contato com o meio e variação paramétrica	62
Figura 3.23 – Posição do efetuador final em relação a direção X em função do tempo na movimentação em contato com o meio.....	62
Figura 3.24 – Posição do efetuador final em relação a direção Y em função do tempo na movimentação em contato com o meio.....	63
Figura 3.25 – Foto do robô INTER.....	64
Figura 3.26 – Foto da peça contendo obstáculo semi-elíptico.....	65
Figura 3.27 – Foto do apalpador horizontal.....	66
Figura 3.28 – Gráfico da posição do efetuador final no plano XY.....	69
Figura 3.29 – Gráfico da força na direção X em função do tempo.....	69
Figura 3.30 – Gráfico da força na direção Y em função do tempo.....	69
Figura 3.31 - Gráfico do erro de posição nas direções X e Y durante o experimento.....	71
Figura 3.32 – Gráfico dos erros de posição nas direções X e Y durante a simulação de movimentação em contato com o meio, apresentada no capítulo 3.3.....	71
Figura 3.33 – Gráfico dos erros de posição nas direções X e Y em função do tempo para a simulação de movimento em contato com o meio irregular e variação paramétrica..	73
Figura 3.34 – Gráfico da força na direção X em função do tempo para movimentação em contato com o meio irregular e variação paramétrica	73
Figura 3.35 – Gráfico da força na direção Y em função do tempo para movimentação em contato com o meio irregular e variação paramétrica	74
Figura A.1 – Representação esquemática de um robô do tipo SCARA	82
Figura B.1 – Sistema de controle de impedância aplicado a um robô SCARA.....	88
Figura B.2 – Diagrama esquemático do bloco “gerador de trajetória no espaço operacional”	89
Figura B.3 – Diagrama esquemático do bloco “cálculo do Jacobiano e sua derivada”	89
Figura B.4 – Diagrama esquemático do bloco “simulação do sensor de força”	90
Figura B.5 – Diagrama esquemático do bloco “cálculo da linearização $y \rightarrow u$ ”	90
Figura B.6 – Diagrama esquemático do bloco de “simulação do robô SCARA”	91

Figura B.7 – Menu de seleção principal	105
Figura B.8 – Sub menu de seleção dos parâmetros de simulação	105
Figura B.9 – Sub menu para escolha da posição inicial nas direções X, Y e Z.....	106
Figura B.10 – Sub menu para escolha da posição final nas direções X, Y e Z	106
Figura B.11 – Mensagem alertando para valores fora da faixa de trabalho	106
Figura B.12 – Sub menu para escolha dos ganhos da matriz Md.....	107
Figura B.13 – Alerta de tempo final menor do que tempo inicial	107
Figura B.14 – Alerta de tempo de aceleração maior do que tempo de simulação.....	108

1 INTRODUÇÃO

1.1 HISTÓRICO DA ROBÓTICA

A palavra robô foi utilizada pela primeira vez no ano de 1921, na peça teatral “Rossum’s Universal Robot”, do escritor tcheco Karel Capek (1890 – 1938).

A origem da palavra é eslava, e se refere ao trabalho realizado de maneira forçada. Na peça, os robôs eram máquinas andróides criadas a partir de uma fórmula obtida por um brilhante cientista, chamado Rossum. Estes robôs serviam a seus chefes humanos, desempenhando todos os trabalhos físicos, até que se rebelaram, destruindo toda a vida humana, com exceção do seu criador, com a frustrada esperança de que lhes ensinasse a reproduzir.

Ao longo da história, o homem tem demonstrado um fascínio por máquinas e dispositivos capazes de imitar as funções e movimentos dos seres vivos. Os gregos denominavam estas máquinas de *automatos*.

Desta palavra, deriva a atual autômato, com a definição utilizada em robótica:

“Máquina que imita a forma e os movimentos de um ser animado”.

Os mecanismos animados são utilizados desde o tempo dos gregos para peças teatrais e brinquedos. Sendo que com o passar do tempo mudou-se o foco e os tipos de aplicação.

O robô como máquina tem um desenvolvimento independente da palavra robot. Apesar dos primeiros autômatos serem mais antigos, e mais semelhantes ao homem, considera-se como antecessor dos robôs atuais o sistema de telemanipulação. Em 1948, R.C. Goertz do Argonne National Laboratory, desenvolveu o primeiro telemanipulador, com o objetivo de manipular elementos radioativos, sem risco ao operador.

Este telemanipulador era composto por um sistema mestre-escravo onde o operador movimentava alavancas do sistema mestre, em uma sala isolada, mas que

permitia a visão do sistema escravo a ser manipulado e, através de cabos e acionamentos, o sistema escravo era movimentado. Todo o controle do movimento era executado pelo operador.

Em 1954, Goertz utilizou tecnologia eletrônica e de servocontroles para substituir os sistemas mecânicos por outros, agora eletronicamente acoplados. Porém, o sistema continuou exigindo a presença constante de um operador, o qual era responsável por todos os movimentos do sistema escravo.

Outro exemplo de telemanipulador, desenvolvido em 1958 por Ralph Mosher, engenheiro da General Electric, consistia de 2 braços mecânicos teleoperados mediante a utilização de um sistema mestre denominado de exoesqueleto.

Esta tecnologia foi muito utilizada pela indústria submarina e nuclear durante os anos 60 e pela indústria espacial, durante os anos 70.

A evolução dos telemanipuladores não apresentou grandes mudanças durante duas décadas, tendo ultimamente voltado a ser objeto de estudo em aplicações na área médica, podendo auxiliar em cirurgias comandadas remotamente, cirurgias do tipo não invasiva, utilizando sistemas de transmissão de imagem e comando, porém com todas as ações sendo controladas pelo médico.

A substituição do operador por um computador foi o que marcou a transição entre telemanipuladores e robôs.

Em 1954, o britânico C.W. Kenward patenteou o primeiro dispositivo robótico de que se tem notícia. Após, tivemos o desenvolvimento das bases da robótica moderna pelo engenheiro americano C. Devol. Foi dele a concepção de um dispositivo de transferência de comandos programados. Porém, a utilização desta tecnologia em escala industrial só se iniciaria por volta do ano de 1956, em conjunto com o engenheiro Joseph F. Engelberger, quando fundaram a Consolidated Controls Corporation, que mais tarde passou a se chamar Unimation, palavra originada da concatenação de outras duas palavras: Universal e Automation.

A primeira máquina Unimate foi instalada em uma fábrica da General Motors, nos Estados Unidos, com a função de auxiliar na produção de peças em uma linha de fundição por injeção.

Outras empresas passaram a desenvolver máquinas similares, como por exemplo a máquina chamada de Versatran da AMF (American Machine and Foundry), e com uma aplicação crescente na indústria, impulsionada por interesses comerciais, acabaram incorporando o termo robô, emprestado da ficção científica de Isaac Asimov.

Em 1968, através de um acordo entre J.F. Engelberger, da Unimation, e a Kawasaki, passou-se a fabricar robôs similares aos Unimate também no Japão.

Um outro ponto importante foi a criação da Associação de Robótica Industrial do Japão, JIRA, em 1972, impulsionada pela crescente utilização dos robôs nas indústrias japonesas, inclusive ultrapassando os Estados Unidos em número de máquinas instaladas.

Em 1974 ocorreu a criação do Instituto de Robótica da América, que em 1984 teve seu nome alterado para Associação das Indústrias Robóticas.

No continente europeu a Robótica teve um desenvolvimento tardio, com o primeiro robô sendo desenvolvido pela ASEA, em 1973, contando com um acionamento totalmente elétrico, e em 1980, foi criada a Federação Internacional de Robótica, com sede em Estocolmo, Suécia.

Os primeiros robôs apresentavam basicamente configurações esféricas e antropomórficas, melhores adaptadas para operações de manipulações, sendo que essas configurações descendiam dos sistemas de telemanipulação, e buscavam fornecer uma movimentação das juntas semelhante aos movimentos de um braço humano.

Em 1982, nos laboratórios da Universidade Yamanashi, no Japão, o professor Makino desenvolveu um robô de concepção nova, denominado SCARA (Selective Compliance Assembly Robot Arm) buscando desta maneira um dispositivo com um

número reduzido de graus de liberdade, um custo limitado e uma configuração voltada à montagem, encaixe e transporte de peças, auxiliando na produção de manufaturados.

A partir do desenvolvimento do robô SCARA, intensificaram-se as pesquisas de robôs voltados exclusivamente para o uso industrial, visando o desenvolvimento de máquinas que, ao invés de agirem como extensões do ser humano, pudessem substituir o ser humano na tarefa [NOSTRAND, 90].

A evolução dos robôs para uso industrial vem ocorrendo de uma maneira vertiginosa. Nos últimos trinta anos, as pesquisas e desenvolvimentos sobre robótica industrial têm permitido que os robôs sejam utilizados em diversas áreas do setor produtivo. Independente do porte da fábrica, os robôs podem ser utilizados em substituição ao homem em tarefas árduas, perigosas ou repetitivas, adaptando-se às variações e ritmos de produção, possibilitando um aumento da produtividade e a redução de riscos à saúde humana, melhorando a qualidade dos produtos.

Principais vantagens da aplicação de robôs no ambiente industrial [KNASEL,86]:

- redução do custo da mão de obra;
- aumento da precisão nos movimentos de carga e peças;
- maior garantia de repetibilidade em tarefas de montagem;
- melhoria da qualidade de vida, com o robô executando tarefas em ambientes perigosos, no lugar de operadores humanos.

Além dos robôs para ambiente industrial, existem diversos ramos de pesquisa e desenvolvimento na área de robótica, entre eles [BARRIENTOS,97]:

Robótica espacial: voltada ao desenvolvimento de braços mecânicos para manutenção e locomoção de satélites em órbita, lançamento de satélites, robôs móveis para exploração do solo da Lua e de Marte, entre outros desenvolvimentos e pesquisas em andamento [WAN,95], [WAN,96];

Robótica submersa e subterrânea: auxiliando na colocação de cabos de comunicação submarinos, manutenção em bases de extração de petróleo. Estes robôs são semelhantes a pequenos submarinos, contando com braços manipuladores, podendo

ter uma navegação independente, variando com a tarefa programada, ou ainda ser telemanipulado. Na área da robótica subterrânea, temos como exemplo, máquinas de escavação teleguiadas ou autônomas e robôs móveis para transporte de carga em ambientes insalubres [YOERGER,92], [SMALLWOOD,99];

Robótica para área militar: contando com diversas pesquisas, temos os robôs móveis para reconhecimento de área, patrulhamento, desarmamento de bombas e minas e transporte de cargas;

Robótica para área médica: neste campo existe uma grande interação entre telemanipulação e robótica, muitas vezes confundindo-se os termos. Temos como exemplo, próteses inteligentes que podem ser classificadas como telemanipuladores quando são controladas a partir de impulsos elétricos gerados pelo próprio paciente, cirurgias executadas por um robô pré programado, com capacidade de adaptação a situações não programadas, cirurgias telemanipuladas, utilizando-se canais de comunicação de imagem e controle, permitindo o controle remoto do manipulador pelo médico [MEGGIOLARO, 97];

Robótica voltada ao lazer: por que não dizer, uma volta às origens, pois os primeiros autômatos eram utilizados para fins lúdicos. Neste campo temos como grande representante o cão robô AIBO, da Sony, na verdade um robô móvel que se locomove com pernas. Existem ainda o desenvolvimento de pequenos robôs-insetos, na tentativa de se imitar a natureza, com exemplares de robôs formigas, libélulas, entre outros. Podem ser utilizados em diversas áreas, como por exemplo, espionagem, monitoramento, entre outros [KRAEUTER, 95].

1.2 CLASSIFICAÇÃO E DEFINIÇÃO DOS ROBÔS INDUSTRIAIS

Definir um robô industrial é uma tarefa difícil, a começar pela diferença conceitual existente entre o mercado japonês e o mercado euro-americano [BARRIENTOS, 97].

Para os japoneses, um robô industrial é todo e qualquer dispositivo mecânico dotado de articulações móveis destinado à manipulação de objetos, sendo que o mercado ocidental é mais restritivo, exigindo uma maior complexidade do sistema, principalmente no que diz respeito ao controle.

Outro ponto é a dificuldade de se chegar a uma definição formal do que seja exatamente um robô industrial, ou ainda, quais seriam os requisitos mínimos necessários para que um dispositivo eletro-mecânico receba a denominação de robô industrial.

Neste sentido, a definição mais aceita no mercado ocidental é a formalizada pela Associação de Indústrias Robóticas, onde se tem:

“ Um robô industrial é um manipulador multifuncional reprogramável, capaz de mover materiais, peças, ferramentas ou dispositivos especiais, seguindo trajetórias variáveis, programadas para realizar diversas tarefas.”

Segundo a Organização Internacional de Padronização (ISO), um robô industrial é classificado como:

“ Manipulador multifuncional reprogramável, com vários graus de liberdade, capaz de manipular materiais, peças, ferramentas ou dispositivos especiais seguindo trajetórias variáveis programadas para realizar tarefas diversas”.

A diferença marcante entre as duas definições apresentadas é a necessidade, por parte da definição da ISO, do robô apresentar vários graus de liberdade. Em comum às duas definições é o fato aceito de que o robô industrial é normalmente composto de um braço mecânico com capacidade de manipulação e que incorpora um controle mais ou menos complexo.

Os robôs industriais podem ainda ser classificados por tipo ou por geração, sendo que de novo, cada órgão apresenta uma classificação distinta.

Classificação por tipos definida pela Associação Francesa de Robótica Industrial [TORRALVA, 92]:

Tipo A: Manipulador com controle manual ou telecomando

Tipo B: Manipulador automático com ciclos pré-ajustados; regulação mediante fins de curso ou sensores; controle por CLP; acionamento pneumático, elétrico ou hidráulico.

Tipo C: Robô programável com trajetória contínua ponto a ponto. É necessário prévio conhecimento da área de trabalho.

Tipo D: Robô capaz de adquirir dados da área de trabalho, readaptando a tarefa em função do meio.

Classificação por geração dos robôs industriais [BARRIENTOS, 97]:

1ª Geração: Repete a tarefa programada sequencialmente. Não considera as possíveis alterações na área de trabalho;

2ª Geração: Adquire informação limitada do meio e atua em consequência. Pode localizar, classificar e detectar esforços e adaptar seus movimentos em consequência.

3ª Geração: Sua programação se realiza mediante o emprego de linguagem natural. Possui capacidade para o planejamento automático de tarefas.

1.3 ROBÔS MANIPULADORES NA INDÚSTRIA

A indústria hoje é um dos maiores utilizadores de robôs manipuladores. A maior aplicação ocorre em tarefas repetitivas, de produção continuada, e que normalmente representam riscos à saúde humana, como por exemplo, na solda a ponto, ou pintura, e em movimentação de cargas. Estas aplicações, amplamente disseminadas, são fundamentadas principalmente em sistemas de controle de posição, nas quais o robô manipulador é programado para a repetição de uma tarefa. As diferenças ficam por conta do tipo de programação, geometria do manipulador e a lógica aplicada ao controle. Normalmente são robôs comerciais, de larga produção e utilização pelas indústrias [TARRALVA,92].

Quanto à programação, ela pode ser feita de maneira *off-line*, ou seja, não há a necessidade do robô para o desenvolvimento da programação, sendo esta transferida para o robô quando este se encontra fora do modo de operação, ou ainda de maneira *on-line*, quando a programação da tarefa é feita com o robô sendo utilizado pelo programador. Neste tipo de programação o robô não pode ser usado, por exemplo, na linha de montagem enquanto estiver sendo programado, tendo de estar parado durante todo o período de programação [MAPLES,96].

A trajetória a ser seguida pelo manipulador pode ser definida através de pontos discretos, onde não se programa a trajetória, velocidade e aceleração quando da movimentação entre dois pontos, importando apenas que o robô manipulador, tendo partido de um dado ponto, atinja um determinado ponto definido. Podemos ter ainda a trajetória definida a partir de equações de deslocamento, onde se pode determinar a trajetória a ser seguida entre dois pontos e a velocidade que será desenvolvida pelo robô durante este percurso. Estes dois métodos estão em um estágio avançado de desenvolvimento, apresentando pequenas evoluções, baseadas principalmente no desenvolvimento de novos processadores, com maior capacidade de cálculo, permitindo um controle de posição e trajetória de maneira mais precisa pois, quanto maior for a velocidade de processamento, maior será a quantidade de variáveis que podem ser analisadas a cada período de amostragem cada vez menores.

Um campo que vem sendo bastante estudado, e que começa a ser utilizado em ambiente industrial, é a aplicação de robôs manipuladores em tarefas onde há uma interação com o meio, ou seja, não são tarefas de simples manipulação de cargas ou objetos, mas tarefas mais complexas, como lixar, polir, cortar entre outras [GORINEVSKY et al.,1997].

Durante estas tarefas, é importante que se faça o monitoramento e o controle das forças de interação entre o robô e o meio. Para isto, utilizam-se diversas técnicas de controle para robôs manipuladores, que diferem das técnicas convencionais de controle de posição, pois necessitam dos valores das forças de contato e de uma maior capacidade de processamento por parte do controlador, sendo fatores limitantes na aplicação industrial. No estágio atual das pesquisas estas aplicações ainda são raras em ambientes industriais. Algumas aplicações já foram desenvolvidas segundo os

conceitos de controle de força e entre elas podemos citar robôs para polimento e esmerilhamento de peças, transporte de cargas perigosas por ambiente desconhecidos, montagem de peças com possibilidade de reconhecer peças defeituosas ou mesmo montagens diferentes efetuadas por um mesmo sistema.

Outro ponto limitante na aplicação desta tecnologia de controle é de ordem econômica pois, como ainda são poucos utilizados, os robôs existentes para estas aplicações são construídos em número reduzido e, devido a isso, têm custos proibitivos.

Utilizando-se a classificação por gerações dos robôs, pode-se perceber que a utilização industrial encontra-se na transição entre a primeira e a segunda geração. Porém, em alguns estudos isolados, já utilizam-se robôs de terceira geração, principalmente em robótica móvel em ambientes industriais.

1.4 CONTROLE DE ROBÔS MANIPULADORES

Um sistema de controle para um robô manipulador normalmente é dividido em duas partes distintas: a de **planejamento de trajetória** e a **etapa de controle** [ZENG,97]:

- planejamento de trajetória:

A posição e/ou força do efetuador final é definida para cada instante de tempo, fornecendo valores de referência a serem utilizados pelo controle.

Esta etapa é executada toda vez em que ocorre uma alteração da trajetória desejada a ser seguida pelo robô manipulador.

- controle:

É feito a comparação entre os valores desejados, previamente definidos na etapa de planejamento de trajetória e os atuais de posição e/ou força, e baseado na lei de controle em uso, são gerados os sinais de controle, que serão enviados para os atuadores do robô manipulador.

As aplicações dos robôs manipuladores podem ser divididas em duas categorias: o **controle de posição** e o **controle de força**.

No **controle de posição** o objetivo é controlar a trajetória do efetuador final do robô manipulador. Considera-se que não haverá contato entre o manipulador e o meio. Neste tipo de aplicação, o controle pode ser desenvolvido para trajetórias ponto a ponto ou para seguimento contínuo de trajetória [SCIAVICCO e SICILIANO, 1996].

A escolha entre os dois tipos deve levar em consideração, principalmente, a existência ou não de obstáculos no espaço de trabalho do robô manipulador pois, pelo método da trajetória ponto a ponto, deseja-se que o manipulador efetue o movimento a partir de um ponto inicial para um ponto final, em um determinado tempo. Para que isto ocorra, é calculada uma trajetória que atenda os requisitos de tempo e que seja fisicamente possível de ser atingida, ou seja, com velocidades e acelerações possíveis de serem atingidas pelo robô manipulador, porém não há uma determinação do percurso que será percorrido entre os dois pontos.

No método de controle por seguimento de trajetória, o caminho a ser percorrido é descrito por mais de dois pontos. Pode-se assim, planejar a trajetória de maneira mais precisa, desviando de obstáculos pré determinados e conhecidos. A trajetória calculada fará com que o manipulador percorra os pontos do caminho, nos instantes de tempo especificados. Um dos métodos de determinação de trajetória ponto a ponto é a interpolação polinomial cúbica em posição com velocidades calculadas nos pontos intermediários [SCIAVICCO e SICILIANO, 1996].

Nas tarefas em que existe o contato entre o manipulador e o meio, passamos a ter a necessidade de **controlar a força** de interação entre o manipulador e o meio, para evitar que ocorram danos tanto ao robô como para o meio.

O controle de força envolve a integração de tarefas como modelagem do ambiente, posição, velocidade e realimentação de força e o ajuste dos torques aplicados nas juntas do robô. A realimentação de vários sinais medidos como saídas do robô (posição, força e velocidade) e a escolha dos sinais de comando de entrada para o robô,

resultam em diferentes métodos para controle de força. Em [ZENG e HEMANI, 1997], os controles são apresentados, divididos, em duas categorias principais:

Algoritmos convencionais de controle de força:

Nesta categoria estão incluídos algoritmos baseados na aplicação de relações entre posição e força aplicada, representados pelo **controle de rigidez** [SALISBURY e CRAIG, 1982], [BORREL, 1979] e [WHITNEY, 1987] ou entre velocidade e força aplicada, representados pelo **controle de impedância** [HOGAN, 1980], [HOGAN, 1984], [HOGAN, 1985] e [SCIAVICCO e SICILIANO, 1996] e pelo **controle por admitância** [WHITNEY, 1977], ou ainda a aplicação de realimentações diretas de força, com aplicações em **controle híbrido de posição/força** [MASON, 1981] e [RAIBERT e CRAIG, 1981], **controle híbrido por impedância** [ANDERSON e SPOONG, 1988] e **controle explícito de força** [WHITNEY, 1977], [VOLPE e KHOSLA, 1992].

Algoritmos de controle avançado de força

Nesta categoria, encontram-se algoritmos baseados em estruturas de controle adaptativo e de controle robusto [Zeng and Hemani, 1997].

Dentro da classe de controle adaptativo, temos os controles convencionais acrescidos de rotinas que possibilitam a adequação dos ganhos das malhas de controle durante a execução dos mesmos, permitindo a adaptação do controle a variações de aplicação [Narendra, 1997], [Craig, 1984].

No controle robusto, temos novamente os controles convencionais acrescidos de rotinas que tem como objetivo permitir o alcance da dinâmica desejada e preservar uma estabilidade robusta na presença de limitadas incertezas do modelo do meio e do robô [Cai and Song, 1994], [Kwan, 1998], [Slotine, 1984].

1.5 TRABALHOS ATUAIS EM CONTROLE DE IMPEDÂNCIA

Devido ao tempo utilizado para o desenvolvimento desta dissertação de mestrado ter sido relativamente longo, em torno de três anos, optou-se por fazer um novo levantamento bibliográfico com os recentes trabalhos de aplicação do controle de impedância no controle de robôs manipuladores.

No laboratório de Biomecânica e reabilitação humana [HOGAN,2002], do Instituto de Tecnologia de Massachusets, MIT, coordenado por Neville Hogan, pesquisase a utilização do controle de impedância para a aplicação no controle de próteses e ainda para o levantamento de características de um meio desconhecido, com o manipulador exercendo uma força de contato com este meio, e através do comportamento do robô, fazer a estimativa dos parâmetros do meio.

Os pesquisadores da Universidade do Tennessee [HAMEL,2001] estão desenvolvendo uma metodologia para incorporar um sistema de supervisão baseado em sensores e modelos assistidos por computador, a um sistema de teleoperação mestre escravo, comandado inicialmente por uma pessoa.

Neste sistema, os comandos de movimento gerados pela pessoa são processados pelo sistema de supervisão, que através de um controle de posição baseado em impedância, limitam ou corrigem os sinais de controle que irão gerar a movimentação dos manipuladores. A partir de um mapeamento prévio do ambiente e de características dos materiais que irão ser manipulados, o sistema de supervisão irá limitar os movimentos do manipulador, tornando a operação de movimentação de cargas perigosas mas segura. Esta pesquisa encontra-se em fase final de desenvolvimento e é apresentada em [HAMEL,2001].

Bruno Siciliano segue no estudo e implementação do controle de impedância no espaço de juntas. Atualmente o controle é somente utilizado no espaço operacional, exigindo a utilização da cinemática direta e reversa, o que implica no cálculo do Jacobiano do robô, causando uma sobrecarga computacional. [SICILIANO,1999]

Em recente trabalho, Fusaomi Nagata e Keigo Watanabe apresentam os resultados obtidos no controle de impedância aplicado em um robô do tipo Puma, em uma tarefa de polimento de superfície. O controle de impedância foi escolhido pois a tarefa a ser desempenhada pelo robô consiste no polimento de uma superfície inicialmente irregular.

O controle desenvolvido apresenta como característica a utilização de duas malhas de controle. A primeira consiste de um controle de impedância baseado em controle de força, ou seja, ocorre uma compensação da força desejada de contato em função da variação de velocidade, e uma segunda malha composta de um compensador de posição em função do erro de posição em relação a posição desejada para o instante em análise.

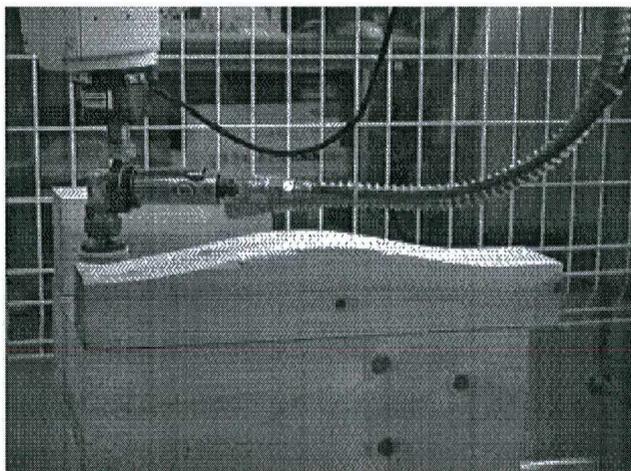


図-7 回転式研磨工具を用いた研磨実験風景

Figura 1.1: peça a ser polida e ferramenta de polimento fixada na extremidade de um robô puma

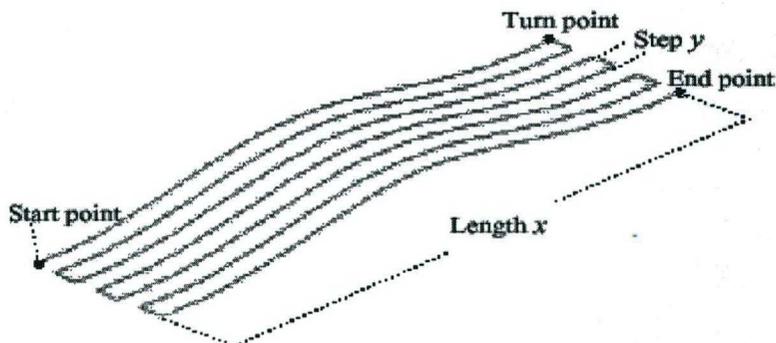


図-8 実験で使用したジグザグパターン

Figura 1.2: trajetória definida a partir do ponto inicial, final e passo.

No experimento apresentado em [NAGATA & WATANABE, 2001], nas direções X e Y, ocorre basicamente um controle de posição, e na direção Z, a partir da diferença entre a força de contato aplicada e a desejada é feita a compensação da altura da ferramenta de polimento, evitando-se a reprogramação de uma máquina CNC a cada pequena mudança na peça a ser polida. A partir da definição do ponto inicial, ponto final e o passo, conforme mostrado na figura 1.2, é feita a geração da trajetória a ser seguida pelo robô e a força de contato é uma variável externa.

Outra linha de pesquisa, apresentada em [PAPADOPOULOS, BILODEAU, 2000], consiste em aplicar o controle de impedância no controle de um robô de juntas de acionamento hidráulico. Para compensar as não linearidades do acionamento hidráulico, é utilizado uma combinação de uma malha de realimentação e uma compensação em avanço.

A trajetória desejada é alterada por um controle de impedância, e servirá de entrada de referência de um modelo de ordem reduzida do atuador hidráulico da junta, de modo a reduzir os efeitos das dinâmicas não lineares do atuador. Valores de posição, velocidade e pressão são realimentados a fim de compensar as dinâmicas não modeladas do sistema.

Desta maneira, obtêm-se um controle de impedância em nível das juntas, porém é necessário especial atenção na escolha dos ganhos de realimentação e do controle de impedância, para permitir uma transição suave entre movimentos sem contato com o meio e movimentos restritos ao meio. Valores de ganho impróprios podem resultar em oscilação.

O esquema de controle utilizado é mostrado na figura 1.3

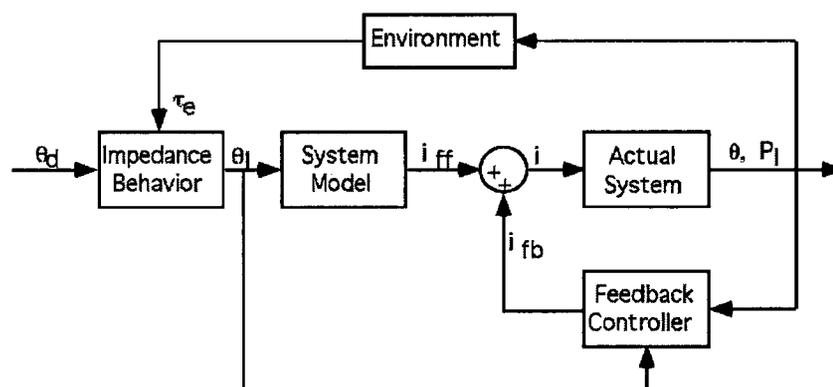


Figura 1.3: esquema de controle proposto para controle de posição de uma junta de acionamento hidráulico

1.6 OBJETIVOS DA DISSERTAÇÃO

O presente trabalho visa o estudo do controle de impedância aplicado ao controle de força em um robô manipulador e objetiva implementar experimentalmente esta técnica de controle de força aplicada a um robô do tipo Scara.

Inicialmente, realiza-se um estudo teórico sobre técnicas conhecidas de controles convencionais de força. Em seguida, é desenvolvido um simulador para o controle de impedância de segunda ordem, utilizando o ambiente de simulação **Simulink**, onde serão feitas várias simulações de movimentos livres e restritos.

Obtidos os dados da simulação, será feita a implementação do algoritmo de controle de impedância utilizando a linguagem de programação orientada a objetos em tempo real *Xoberon* [REISER, 1991], [MOSSENBOCK, 1993], [VESTLI, 1997]. Este algoritmo será utilizado para realizar os testes do controlador em um robô de porte industrial, de configuração SCARA.

Teremos ainda como ponto importante do trabalho, a contribuição para o Laboratório de Robótica da UFSC, através do intercâmbio entre os pesquisadores envolvidos com o robô SCARA, durante a elaboração desta dissertação, além de aumentar as possibilidades de utilização do robô.

1.7 JUSTIFICATIVA

Com a crescente necessidade de utilização de robôs manipuladores na indústria, desempenhando funções que exigem o contato entre o manipulador e o meio, faz-se necessária a pesquisa constante de novas técnicas de controle, capazes de atuar nestas situações.

Visando estas novas aplicações dos robôs manipuladores, o grupo de robótica da UFSC, formado por pesquisadores do Departamento de Engenharia Mecânica, e do Departamento de Automação e Sistemas (DAS) da Universidade Federal de Santa

Catarina (UFSC), concentra esforços no desenvolvimento da área de controle de força aplicado a robôs industriais. Uma demonstração deste esforço foi a aquisição de um robô de porte industrial, de configuração SCARA, denominado Inter, desenvolvido e construído pelo Instituto de Robótica da Universidade Técnica Federal de Zurique (ETH), Suíça. O robô foi equipado com um sensor de força e momentos, acoplado entre o punho e a ferramenta de trabalho, de grande importância para a implementação de algoritmos de controle de força.

O algoritmo de controle de impedância [HOGAN, 85], [SPONG e VIDYASAGAR,89], [SCIAVICCO e SICILIANO, 96], [SURDILOVIC e KIRCHHOF, 96], [CACAVALE et al, 99] aplicado no controle de robôs manipuladores, mostra-se versátil ao permitir a sua utilização em controles de seguimento de trajetória, com e sem contato do manipulador com o meio, e ainda no controle de movimentação em locais onde possam existir obstáculos, gerenciando a dinâmica do erro em função da força de contato entre o robô manipulador e o meio.

A existência de alguns trabalhos de controle de força desenvolvidos nesta universidade por alunos de mestrado e doutorado, serviram de suporte técnico para o desenvolvimento deste projeto. Dentre eles pode-se citar os trabalhos de [BIER, 2000], [AMARAL, 00] , [VARGAS, 00], [SILVEIRA, 98], [MENDES, 99], [WEIMANN, 99] e [BATTISTELA, 99].

1.8 ESTRUTURA DA DISSERTAÇÃO

No Capítulo 2, será feita uma introdução sobre os controles convencionais de força: controle de rigidez, controle de admitância e controle de impedância.

Em seguida, apresenta-se o estudo de um controle de impedância de segunda ordem, baseado na dinâmica inversa do manipulador, e o diagrama de simulação desenvolvido no Simulink para o controle.

No capítulo 3, são expostos os resultados obtidos durante as simulações de duas

situações de movimentação do robô: movimentação livre e com restrição. Faz-se ainda a simulação da presença de variação paramétrica. São determinados ainda os ganhos que serão utilizados durante as simulações e que serão utilizados como valores iniciais para a experimentação prática. Tem-se a descrição da experimentação prática, com a implementação do algoritmo de controle para a linguagem Xoberon, baseado no controle anteriormente desenvolvido para as simulações e a apresentação dos resultados obtidos no experimento que comprovam o bom funcionamento do controlador.

Enfocam-se ainda as conclusões sobre o trabalho, destacando-se os resultados obtidos, as contribuições, possibilidades de uso para o controle de impedância, vantagens, desvantagens e conclusões obtidas com este trabalho.

No Capítulo 4, são apresentadas novas perspectivas de trabalho e linhas de pesquisas em desenvolvimento.

2 ALGORITMOS CONVENCIONAIS DE CONTROLE DE FORÇA

A característica principal do controle de força de robôs manipuladores é poder determinar as forças de interação entre o manipulador e o meio e usar, de maneira eficiente, os sinais de realimentação buscando manter o movimento e a força de contato desejados. As variáveis básicas utilizadas no controle de força convencionais são posição, velocidade, aceleração e força. As diferenças existentes entre os diferentes algoritmos de controle de força são as aplicações distintas destas variáveis básicas e as relações entre as mesmas.

2.1 CARACTERIZAÇÃO DO PROBLEMA DE CONTROLE DE FORÇA

As tarefas de contato são definidas como todas as operações efetuadas por um robô manipulador, nas quais o robô aplica uma força sobre um objeto ou o meio. Pode-se considerar que as tarefas de contato possuem três etapas: movimentação no espaço livre, momento de transição e aplicação da força.

O processo a ser controlado envolve as três partes e cada uma delas apresenta características particulares.

Quando em movimentação no espaço livre, o robô estará sob a influência do controle implementado para fazer o posicionamento de seu efetuador final, podendo ser considerado como um sistema isolado.

No momento da transição, passa a existir o contato entre o manipulado e o meio, com uma alteração abrupta da rigidez do meio, pois durante a movimentação no espaço livre, a rigidez do meio é considerada nula ou desprezível, quando em contato, a rigidez do meio passa a ter valores, da ordem de 10^4 a 10^6 Nm. Isto faz com que a dinâmica do processo a ser controlado seja muito diferente no espaço livre e no contato.

Durante a aplicação de uma força, o robô estará em contato efetivo com um objeto ou superfície, e considerando o robô e o objeto de contato como sendo rígidos, pequenos movimentos poderão provocar grandes esforços de interação, dependendo da

direção e sentido dos mesmos. Para evitar estes grandes esforços, que podem ser danosos ao manipulador e ao objeto ou meio, é conveniente nesta etapa da tarefa controlar a força de interação ao invés do controle da posição. A partir desta consideração, o robô e o objeto passam a ser considerados como dois sistemas acoplados e, para fins de controle, o robô não pode ser considerado como um sistema isolado ou seja, a força de interação não pode ser desprezada.

O controle da força de interação é fundamental nas tarefas de contato, mas estas tarefas também requerem o controle do movimento no espaço livre prévio ao contato. Utilizando-se um controle da força de contato somente, enquanto em movimentação no espaço livre, a variável a ser controlada terá valor nulo, enquanto não for estabelecido o contato entre o manipulador e o objeto a ser manipulado. Este fato nos obriga a utilizar controle de posição durante a movimentação sem contato. É necessário considerar ainda que o momento exato da colisão é de difícil previsão, devido a incertezas existentes no posicionamento do objeto e desvios de trajetória do robô. No momento em que ocorre a colisão, as forças de contato precisam ser monitoradas ou até controladas e desta forma, o problema será sempre de controle de posição e força.

Uma abordagem deste problema é a técnica denominada **Controle Híbrido**, em que o espaço da tarefa é dividido em dois sub-espacos, um controlado em posição e o outro em força. O controle híbrido porém não resolve o problema do momento da transição, pois nos graus de liberdade a serem controlados em força, o robô deve iniciar a tarefa em contato com o meio, pois não havendo contato na direção a ser controlada a força, o sistema de controle irá deslocar o manipulador nesta direção, seguindo uma rampa de aceleração, até que ocorra o contato, podendo ocorrer danos ao sistema e ao meio.

Em algumas situações, a tarefa a ser desenvolvida pelo robô manipulador pode ser considerada como tendo como objetivo a aplicação de uma força, a partir do conhecimento prévio do meio externo, permitindo desta maneira calcular quais serão os movimentos que ocorrerão em função do controle de força. Como exemplo, considere uma tarefa de posicionamento e inserção de uma peça em uma base de montagem. Neste caso, tem-se apenas uma referência de esforço a ser atingido. Ao inserir um peça

em seu encaixe, será necessário um determinado esforço para o encaixe, e portanto, um controle de força que permita a realização da tarefa.

Em muitos casos porém, dado que o meio não é conhecido com suficiente precisão, aplica-se uma força e monitora-se o movimento realizado, modificando-se a força aplicada em função do movimento. Nestas situações, utiliza-se uma referência de posição e um erro de posição, que pode ser utilizado para calcular a força de contato. Este valor de força estimado pode ser utilizado para modificar a referência de posição nos casos em que a força ultrapasse limites determinados de segurança.

Nas tarefas em que a posição é controlada e a força é monitorada, ao controle convencional de posição é acrescida uma malha externa de controle, que terá como entrada a medida de força e como saída, um valor de correção para a posição. Desta maneira, enquanto não ocorrer contato, o robô estará sob a influência única do controle de posição, o qual deve ser projetado para atender as necessidades de utilização do robô manipulador. Uma vez em contato com o meio, os valores medidos de força passarão a influenciar no valor de referência de posição a ser seguido.

As modificações na trajetória de posição devido a realimentação de força causam um desvio entre a posição desejada para o efetuador final do manipulador e a posição real atingida pelo mesmo. O erro na posição e a força que o robô manipulador estará exercendo sobre o meio é relacionada através da impedância do manipulador, considerando-se o meio rígido. Esta impedância será função do manipulador e dos parâmetros do controlador de posição e da malha de realimentação de força. O controle de impedância, que é o objetivo de estudo deste trabalho, coloca o problema como sendo o de projetar uma estrutura de controle, utilizando medidas de posição, velocidade e força, de tal maneira a permitir a determinação da impedância que o manipulador irá apresentar em contato com o meio.

Determinada a impedância do manipulador, e calculando-se a diferença de posição do manipulador, é possível determinar as forças de contato aplicadas. Pode-se assim garantir a relação entre a força aplicada e a posição atingida.

2.2 UTILIZAÇÃO DA REALIMENTAÇÃO DE FORÇA

Basicamente a realimentação da medida da força pode ser utilizada para alterar dois tipos de referência de entrada: referências de força ou referências de variáveis de movimento ou seja, posição, velocidade ou aceleração. As características de cada uma destas abordagens serão discutidas brevemente a seguir.

Neste trabalho serão abordadas técnicas de controle que utilizam a realimentação da medida de força para alteração de referências de movimento. A realimentação de força é implementada em uma malha externa a um sistema de controle de posição previamente desenvolvido. Desta maneira, enquanto não existirem forças de interação entre robô e o meio, o controle será somente de posição, porém a partir do momento em que ocorra contato entre o manipulador e o meio e que o valor das forças medidas deixem de ser nulos, a realimentação através da malha externa de controle irá causar uma alteração nos valores de referência para a posição, fazendo com que a posição a ser atingida pela garra não seja a originalmente planejada, mas sim a máxima posição possível de ser alcançada sem que sejam ultrapassados valores limites para a força de contato.

Tal situação pode ser interpretada como a imposição de uma flexibilidade ao robô através da ação de controle. A esta estratégia de controle, denomina-se Controle Ativo de Flexibilidade e o controle de impedância é uma das formas de implementar esta ação.

2.2.1 CONTROLE DE FLEXIBILIDADE

A flexibilidade pode ser definida como a capacidade que o robô, juntamente com o controlador, possui de modificar a posição do efetuador final quando da ocorrência de uma força de contato. Esta estratégia é bastante útil quando o robô deve lidar com objetos rígidos. A flexibilidade do efetuador final do robô pode ser alcançada pela modificação da estrutura física do robô, com a adoção de mecanismos que permitam uma flexibilização das juntas ou dos elos, quando é chamada de flexibilidade passiva, ou ainda através de uma ação de controle que, a partir de uma monitoração da força de

contato, altera a posição do efetuator final, recebendo assim o nome de flexibilidade ativa.

O controle de flexibilidade ativa irá dotar o manipulador de uma certa capacidade para modificar a posição do efetuator final quando este receber a aplicação de uma força. A flexibilidade desejada será implementada a partir da estrutura de controle. A vantagem deste método é a possibilidade de alteração da flexibilidade implementada através da alteração de ganhos na malha de controle.

2.2.2 ANALOGIA MECÂNICA-ELÉTRICA

Antes de ser iniciada a análise do controle de flexibilidade ativa, e baseado na equivalência das equações diferenciais, é possível fazer uma analogia entre os parâmetros e variáveis envolvidos na descrição de um sistema mecânico com os de um sistema elétrico. Partindo desta comparação, é possível aproveitar o conhecimento relacionado com a resolução e controle de sistemas elétricos na abordagem do problema da interação física robô-ambiente.

O movimento efetuado pode ser relacionado com o fluxo de cargas e a força aplicada pelo robô como sendo um potencial elétrico. A equação que descreve a relação entre a força e a posição em um sistema mecânico de segunda ordem pode ser então comparada com a de um sistema elétrico equivalente.

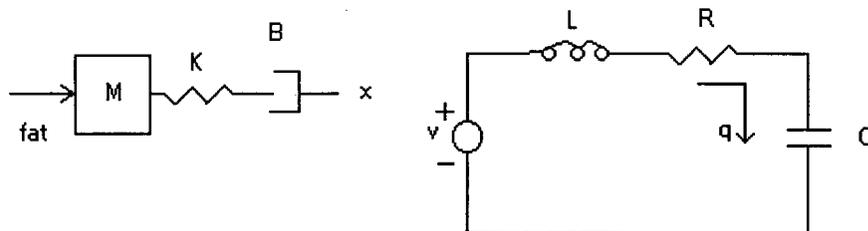


Figura 2.1: equivalência mecânica elétrica

$$fat = M \frac{d^2 x}{dt^2} + B \frac{dx}{dt} + Kx \equiv v = L \frac{d^2 q}{dt^2} + R \frac{dq}{dt} + Cq \quad (2.1)$$

	<i>Sistema mecânico</i>		<i>Sistema elétrico</i>
<i>Fat</i>	<i>Força resultante</i>	<i>v</i>	<i>Tensão</i>
<i>M</i>	<i>Massa(ou inércia)</i>	<i>L</i>	<i>Indutância</i>
<i>B</i>	<i>Atrito</i>	<i>R</i>	<i>Resistência</i>
<i>K</i>	<i>Rigidez</i>	<i>C</i>	<i>Capacitância</i>
<i>X</i>	<i>Posição</i>	<i>q</i>	<i>Carga elétrica</i>

A analogia entre os sistemas mecânicos e elétricos será utilizada para a obtenção do modelo do sistema físico equivalente do robô com controlador em interação com o meio, nos casos de controle a serem estudados a seguir.

2.2.3 IMPEDÂNCIA E ADMITÂNCIA MECÂNICAS

O comportamento de um sistema elétrico pode ser expresso de uma maneira geral a partir dos conceitos de impedância e admitância.

Considerando-se a analogia apresentada anteriormente, pode-se analisar o problema de interação do robô com o meio em termos de impedância e admitância mecânicas. De maneira similar como são definidas na eletricidade, os sistemas físicos podem ser divididos em dois tipos: impedâncias, que a partir de um movimento como entrada, produzem uma força como saída; e admitâncias, que a partir de uma entrada de força, apresentam um movimento como saída.

A partir desta análise e com o conhecimento de que na interação dinâmica entre dois sistemas elétricos ou mecânicos, o que permite a melhor transferência de potência é a dualidade no sentido admitância/impedância, o ideal é que o manipulador sob a ação de controle apresente um comportamento de uma impedância mecânica. Este fato está baseado no princípio da dualidade [ANDERSON, 87], onde o manipulador deve ser controlado para responder como o dual do ambiente.

Quando o ambiente com o qual interage o robô puder ser descrito como uma admitância, o robô deverá se comportar como uma impedância. Se a força originada

pelo robô não é adequada, pode-se alterar a posição de referência em malha fechada. A força realizada dependerá do ambiente, da mesma maneira que em um circuito equivalente de Norton a tensão de saída depende da carga aplicada, e a posição real do robô apresentará um certo desvio, devido a flexibilidade imposta ao mesmo, que no circuito elétrico equivalente está representado pela impedância em paralelo equivalente de Norton. Desta maneira, a posição real do robô, equivalerá a corrente elétrica de saída do circuito equivalente em função da força aplicada no ambiente, o que no circuito equivale a tensão de saída.

2.3 CONTROLE DE IMPEDÂNCIA

Será discutido nesta seção a implementação do Controle de Impedância a partir do modelo de uma junta sobre a ação de um controle de posicionamento. Será apresentado ainda um estudo sobre o controle de impedância para o robô com várias juntas.

2.3.1 IMPEDÂNCIA DE UMA JUNTA COM CONTROLE DE POSIÇÃO

O objetivo desta seção é estudar a implementação de um controlador que permita ao manipulador se comportar como uma impedância mecânica e ainda através da alteração dos parâmetros do controlador, alterar a dinâmica do sistema mecânico.

Um modelo do conjunto atuador-junta é mostrado na figura 2.2:

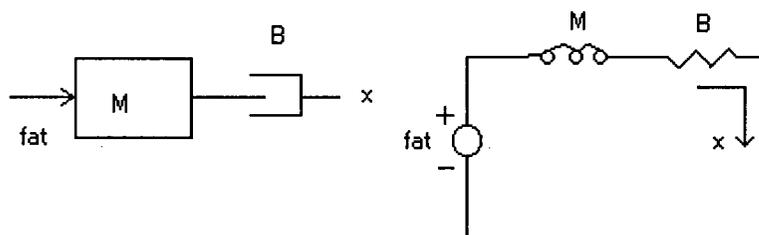


Figura 2.2 – Modelo mecânico e elétrico do conjunto atuador - junta

Note que o modelo é para um sistema de um grau de liberdade, atuando no espaço livre e sem considerar perturbações. M é a massa e B é o atrito equivalente do conjunto atuador-junta, f_{at} é a força gerada pelo atuador e x é o deslocamento da junta.

A equação que representa este sistema mecânico possui as características similares a uma admitância, que será denominada Y_r .

$$\frac{x}{f_{at}} = Y_r = \frac{1}{Ms^2 + Bs} \quad (2.2)$$

O sistema elétrico equivalente é dado pela figura 2.3:

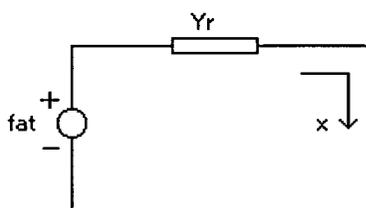


Figura 2.3 – Esquema elétrico equivalente do conjunto junta - atuador

Para um robô com várias juntas, em lugar de Y_r , teremos o modelo dinâmico, fixando a relação entre as variáveis do movimento e a força resultante aplicada pelos atuadores. Analisando um esquema de controle de posição possível de ser aplicado ao sistema, onde uma malha interna faz a realimentação de velocidade e é utilizado um ganho proporcional de posição, teremos o diagrama apresentado na figura 2.4:

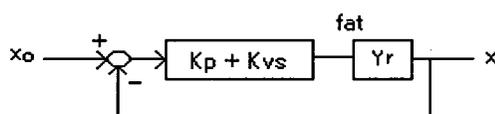


Figura 2.4 – Diagrama de blocos de uma junta com controle de posição/velocidade

E o esquema elétrico equivalente para este controle, quando em atuação no espaço livre e adotando $K_p + K_v s = Z_c$, resultará em:

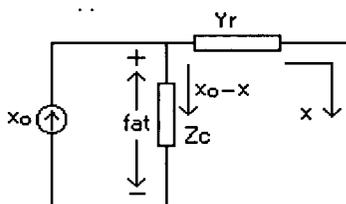


Figura 2.5 – Esquema elétrico equivalente da junta sobre o controle de posição/velocidade

Fazendo uma análise a partir das situações mostradas na figura 2.3 e 2.5, nota-se que a aplicação do controle de posição com realimentação de velocidade promove a substituição da fonte de força presente no esquema 2.3, por uma fonte de movimento, com a implantação de uma impedância Z_c em paralelo.

Analisando agora o ambiente, este pode ser representado por um sistema composto de massa-mola-amortecedor, como na figura 2.6, onde temos que f_e é a força aplicada no ambiente, M_e é a massa do ambiente, e x_e é a posição inicial do ambiente. Fazendo a análise considerando-se que o ambiente é fixo, teremos $x_e = 0$ e o modelo do ambiente resultará em uma admitância equivalente Y_e . (figura 2.7)

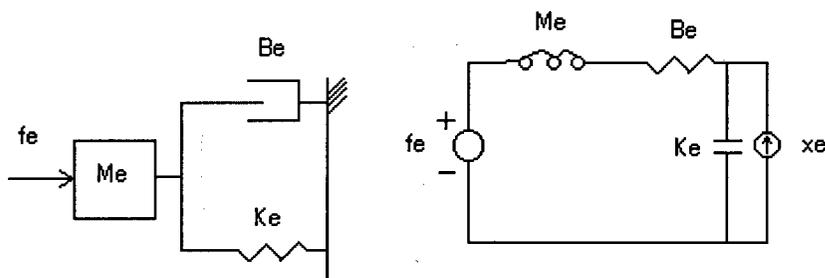


Figura 2.6 – Modelo mecânico e elétrico do ambiente

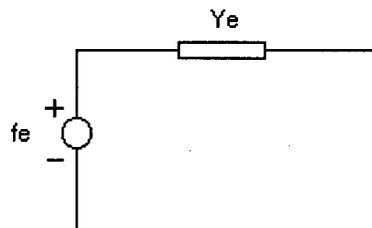


Figura 2.7 – Modelo elétrico equivalente do ambiente

Na ocorrência de contato entre o manipulador e o meio, o conjunto sistema mecânico controlado + meio pode ser esquematizado segundo [GOLDEMBERG,88]:

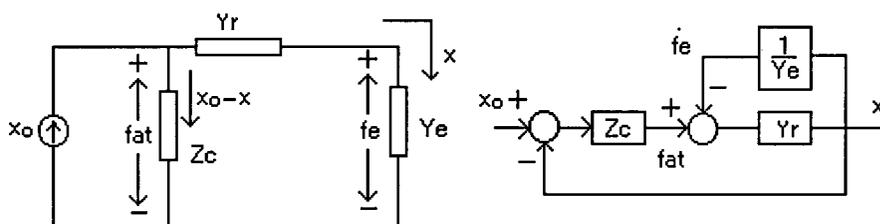


Figura 2.8 – Esquema elétrico equivalente da junta em contato com o meio

Fazendo-se a análise da força aplicada pela junta ao meio e a força de reação do meio sobre a junta, teremos:

$$-f_e = Z_c(x - x_0) + \frac{x}{Y_r} \quad (2.3)$$

a qual mostra que a força depende não só do controlador de posição Z_c , mas também da dinâmica da junta Y_r .

Para tornar o sistema independente da dinâmica da junta, é necessário eliminar o termo x/Y_r , para estabelecer o valor da impedância da junta através da escolha de Z_c , sem depender dos parâmetros ou não linearidades de Y_r . Isto pode ser obtido utilizando-se uma estrutura de controle que faça a compensação da dinâmica do manipulador.

Analisando a equação 2.3, nota-se que mantendo a relação entre f_e e $x_0 - x$ constante e conhecida, ou seja, a impedância do conjunto junta-controlador constante e independente das mudanças de posição da junta, a força f_e poderá ser limitada, monitorada ou controlada através de x_0 , já que na equação 2.3 se o termo da direita for conhecido e constante, com o x sendo medido, f_e fica em função de x_0 .

2.3.2 CONTROLE DE IMPEDÂNCIA DE UMA JUNTA ISOLADA

Com o intuito de eliminar a dependência da impedância da junta com os parâmetros da junta, [GOLDEMBERG,88] apresenta o seguinte esquema de controle da junta quando o contato com o meio já foi estabelecido:

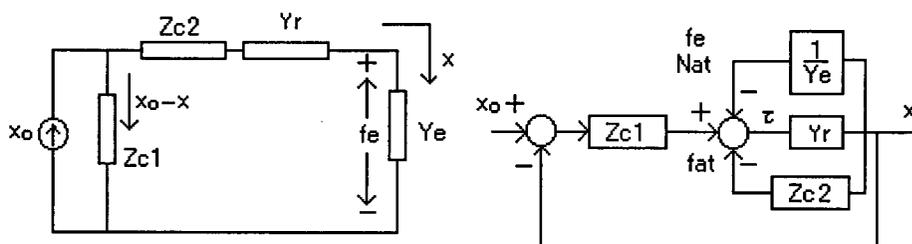


Figura 2.9 – Esquema elétrico equivalente da junta com controle de flexibilidade em contato com o meio

A força f_e será dada por:

$$-f_e = Z_{c1}(x - x_0) + x \left(Z_{c2} + \frac{1}{Y_r} \right) \quad (2.4)$$

Considerando o conhecimento perfeito do modelo da junta e adotando $Z_{c1} = Z_d$ e $Z_{c2} = -1/Y_r$, obtém-se:

$$-f_e = Z_d(x - x_0) \quad \text{ou} \quad \frac{f_e}{x_0 - x} = Z_d \quad (2.5)$$

Esta estrutura de controle da maneira como se encontra só poderia ser utilizada na situação em que o manipulador já se encontra em contato com o meio, pois na ausência de contato o sistema se tornaria instável. Para permitir a utilização desta estrutura de controle na situação de movimentação sem restrição, é necessário fazer a alteração conforme indicado na figura 2.10:

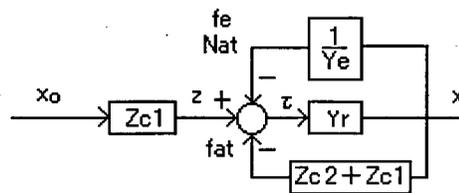


Figura 2.10 – Junta com controle de flexibilidade em contato com o meio

Mantendo-se a escolha anterior para os ganhos Z_{c1} e Z_{c2} , o diagrama de controle equivalente é exibido na figura 2.11.

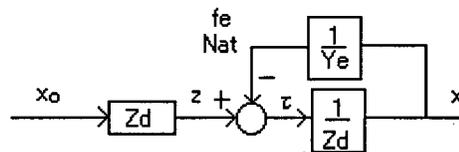


Figura 2.11 – Diagrama equivalente da junta com controle de flexibilidade em contato com o meio

Quando do funcionamento da junta controlada conforme mostrado na figura 2.11 e considerando a impedância desejada Z_d para o sistema junta-controlador como um sistema de segunda ordem dado por:

$$Z_d = M_d s^2 + B_d s + K_d \quad (2.6)$$

e com a junta movimentando-se no espaço livre, segundo o modelo:

$$\tau = M_d \ddot{x} + B_d \dot{x} + K_d x \quad (2.7)$$

Aplicando-se o torque de controle como mostrado na figura 2.11, tem-se:

$$\tau = z = M_d \ddot{x}_0 + B_d \dot{x}_0 + K_d x_0 \quad (2.8)$$

resultando em:

$$M_d (\ddot{x}_0 - \ddot{x}) + B_d (\dot{x}_0 - \dot{x}) + K_d (x_0 - x) = 0 \quad (2.9)$$

tendo-se no espaço livre um resultado idêntico ao obtido com a metodologia geral do torque calculado onde, para fazer a linearização do sistema, os sinais de x e \dot{x} , são substituídos através da ação de controle, pelo erro de posição e velocidade, formando assim uma equação diferencial para o erro assintoticamente estável [CRAIG, 86].

O inconveniente do esquema de controle proposto é a necessidade da utilização da medida de aceleração \ddot{x} , o que pode tornar o controle inviável ou de difícil implementação, devido a complexidade na medição de aceleração.

Uma proposta para contornar o problema da necessidade da medida da aceleração é a utilização da medida da força em substituição. Para possibilitar a análise desta possível implementação, é adotado inicialmente $Y_r = -\frac{1}{Z_{c2}}$ e $-Z_{c2} = Ms^2 + Bs$ e Z_d mantido como um sistema de segunda ordem, como adotado anteriormente em

(2.6). Fazendo a substituição dos novos valores no diagrama de blocos apresentado na figura 2.10, tem-se um novo diagrama:

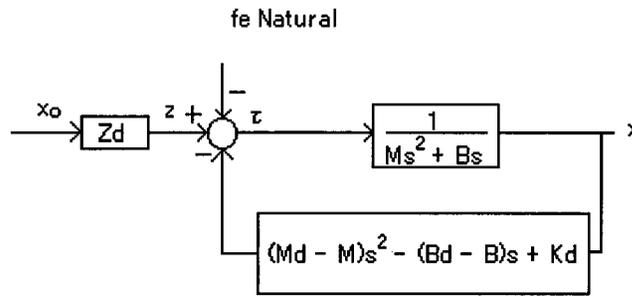


Figura 2.12 – Diagrama de blocos da junta com controlador utilizando realimentação de aceleração, velocidade e posição.

E analisando o diagrama em blocos da figura 2.12, tem-se que:

$$\frac{x}{z} = \frac{1}{M_d s^2 + B_d s + K_d} \quad (2.10)$$

$$\frac{\tau}{z} = \frac{M s^2 + B s}{M_d s^2 + B_d s + K_d} \quad (2.11)$$

$$\frac{\tau}{fe} = \frac{M s^2 + B s}{M_d s^2 + B_d s + K_d} \quad (2.12)$$

Para comprovar que a substituição da utilização do sinal de aceleração pela medição da força é possível, o esquema de controle que utilize só a medida de posição, velocidade e força da junta deverá apresentar as mesmas funções de transferência apresentadas anteriormente (2.10), (2.11), (2.12).

Analisando o diagrama da figura 2.12, pode-se interpretar que a realimentação de posição introduz um termo de rigidez desejada na junta, de valor K_d , a realimentação de velocidade modifica o amortecimento na junta do valor B para o valor B_d e finalmente a realimentação de aceleração altera o valor da massa do valor M para o valor M_d . É possível considerar um sistema que através da realimentação da posição e velocidade, introduza o valor de rigidez desejado K_d e altere o valor de amortecimento

para o valor desejado B_d e que através de um ganho direto na malha de controle altere o valor relativo a massa, sem a necessidade de utilização do valor da aceleração. Este esquema de controle é mostrado na figura 2.13.

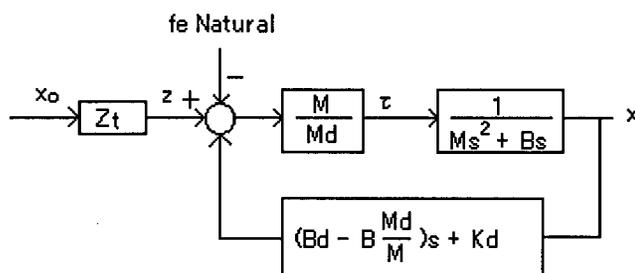


Figura 2.13 – Diagrama de blocos da junta com controle de flexibilidade modificado

Porém, o controle como mostrado na figura 2.13 não é possível de ser implementado, pois a realimentação natural de torque atua diretamente sobre o torque da junta, sem que exista possibilidade de ser colocado o ganho $\frac{M}{M_d}$ como indicado.

Para eliminar este problema, o ganho $\frac{M}{M_d}$ é transferido para fora da malha de controle, resultando no diagrama mostrado na figura 2.14.

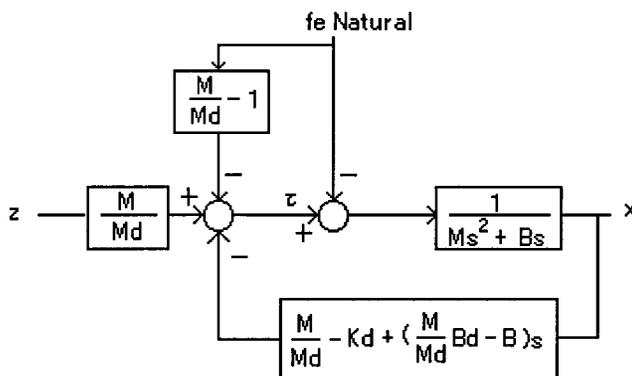


Figura 2.14 – Diagrama de blocos da junta com controle de flexibilidade sem medida de aceleração

Desta maneira, a realimentação unitária de f_e é a realimentação natural, atuando diretamente sobre a junta. A componente de f_e realimentada através do ganho $\frac{M}{M_d} - 1$ é

uma malha de realimentação de força. Chega-se então àquilo que foi discutido no início do capítulo, que o controle de impedância faria uso de uma malha interna de controle de posição e de uma malha externa de realimentação de força.

Na figura 2.13, a variável Z_d foi substituída por Z_t , para permitir a análise do sistema para três situações distintas:

1) Se Z_t for igual a K_d , o esforço f_e , analisado a partir do diagrama exposto na figura 2.14, resultará em:

$$-f_e = M_d \ddot{x} + B_d \dot{x} + K_d (x - x_0) \quad (2.13)$$

conforme apresentado em [HOGAN, 87].

2) Se for adotado Z_t como sendo $B_d s + K_d$, o esforço f_e resulta em:

$$-f_e = M_d \ddot{x} + B_d (\dot{x} - \dot{x}_0) + K_d (x - x_0) \quad (2.14)$$

conforme apresentado em [HOGAN, 85].

3) Como apresentado em [LAWRENCE, 88], para Z_t igual a $M_d s^2 + B_d s + K_d$, o esforço f_e resulta em:

$$-f_e = M_d (\ddot{x} - \ddot{x}_0) + B_d (\dot{x} - \dot{x}_0) + K_d (x - x_0) \quad (2.15)$$

Percebe-se que o principal ponto de diferença entre as três apresentações, é a necessidade ou não da medição e utilização do sinal de velocidade e de aceleração. Para o ambiente de simulação a ser apresentado no final deste capítulo, e nos experimentos práticos, apresentados no capítulo 3, utiliza-se a segunda abordagem, a qual utiliza o sinal de posição e velocidade.

2.3.3 CONTROLE DE IMPEDÂNCIA EM UM ROBÔ COM VÁRIAS JUNTAS

Analisando agora um robô manipulador, composto de duas ou mais juntas, e considerando que a tarefa a ser cumprida pelo robô normalmente é descrita no espaço cartesiano, mais especificamente no sistema de coordenadas do centro do efetuador final, é conveniente que a implementação do controlador seja feita no espaço cartesiano.

A obtenção da equação do controlador que implementa um controle de impedância no espaço cartesiano, conforme descrito em [HOGAN, 87] é baseado na análise da existência de um atuador fictício no extremo da garra, o qual exerce uma força F_{act} . Os torque necessários nas juntas para provocar na garra a mesma força F_{act} é dado por:

$$T_{act} = J^T F_{act} \quad (2.16)$$

Considerando F_e como sendo um vetor de forças externas, tem-se:

$$M(q)\ddot{q} + V(q, \dot{q})\dot{q} + G(q) = J^T (F_{act} - F_e) \quad (2.17)$$

explicitando-se o termo de aceleração das juntas \ddot{q} , obtêm-se:

$$\ddot{q} = M^{-1} [J^T (F_{act} - F_e) - V(q, \dot{q})\dot{q} - G(q)] \quad (2.18)$$

e tendo as relações entre posição, velocidade e aceleração no espaço das juntas e no sistema de coordenadas do centro do efetuador final, descritas no Apêndice A:

$$\ddot{x} = J\ddot{q} + \dot{J}\dot{q} \quad (2.19)$$

$$x = L(q) \quad (2.20)$$

$$\dot{x} = J\dot{q} \quad (2.21)$$

Substituindo-se 2.17, 2.18, 2.20 e 2.21 em 2.19, obtêm-se:

$$\ddot{x} = JM^{-1}J^T (F_{act} - F_e) - JM^{-1}(V(q, \dot{q})\dot{q} - G(q)) \quad (2.22)$$

deseja-se obter:

$$M_d(\ddot{x} - \ddot{x}_0) + B_d(\dot{x} - \dot{x}_0) + K_d(x - x_0) = -F_e \quad (2.23)$$

com M_d , B_d e K_d definidos no espaço cartesiano da tarefa e definindo:

$$W = JM^{-1}J^T \quad (2.24)$$

e explicitando-se T_{act} , obtém-se:

$$T_{act} = J^T W^{-1} M_d^{-1} [M_d \ddot{x}_0 + B_d \dot{x}_0 + K_d x_0] - J^T W^{-1} \dot{J} \dot{q} - J^T W^{-1} M_d^{-1} [B_d J \dot{q} + K_d L(q)] + V(q, \dot{q}) \dot{q} + G(q) - J^T [W^{-1} M_d^{-1} - I] F_e \quad (2.25)$$

que resultará no diagrama de blocos mostrado a seguir:

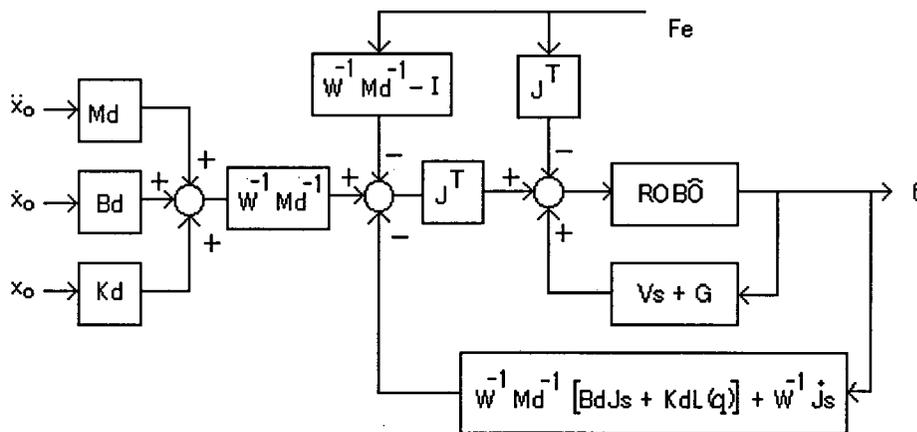


Figura 2.15 – Controle de impedância em um robô com várias juntas

O controlador apresentado, utilizando uma realimentação de força permite, em malha fechada, calcular um vetor de forças/torques aplicado que depende da situação das juntas e da situação comandada, através dos parâmetros ajustados para o controlador pelo usuário. A escolha dos parâmetros irá determinar a relação dinâmica entre a força aplicada e a posição, tendo-se assim um controle de impedância no espaço cartesiano.

2.4 CONTROLE DE RIGIDEZ

O controle de rigidez foi desenvolvido por J. K. Salisbury e apresentado em [SALISBURY, 80]. O controlador implementado tem como característica principal a de fazer com que o efetuador final do robô manipulador se comporte como um conjunto de molas, cada uma agindo em um grau de liberdade. A partir da escolha dos ganhos do controlador, é possível alterar a rigidez demonstrada pelo manipulador quando em contato com o meio.

2.4.1 CONTROLE DE RIGIDEZ APLICADO A UMA JUNTA

Inicialmente será feita uma análise de apenas uma junta isolada, agindo sobre o controle de rigidez, e em seguida será discutido possíveis implementações para aplicação em um robô de várias juntas. Partindo do modelo da junta, dado por:

$$\tau - f_e = (J_{eq} \cdot s^2 + B_{eq} \cdot s) q_1 \quad (2.26)$$

onde q_1 é a posição da junta, J_{eq} é a inércia do conjunto atuador-junta, B_{eq} é o atrito, τ é o torque gerado pelo atuador e f_e a força que é aplicada pela junta ao ambiente.

Segundo [WHITNEY,85], o controle de rigidez utiliza a seguinte lei de controle:

$$\tau = K_p \left(q_0 - q_1 - \frac{f_e}{K_{f1}} \right) - K_v \dot{q}_1 \quad (2.27)$$

onde q_0 é a posição de referência, K_p é o ganho de posição, K_v é o ganho de velocidade, f_e é a medida da força aplicada e $\frac{1}{K_{f1}}$ é o ganho de força. A idéia básica associada à equação de controle mostrada é a de se modificar a posição de referência baseando-se na medida da força.

Aplicando a lei de controle a junta anteriormente modelada, obtém-se o sistema mostrado na figura 2.16:

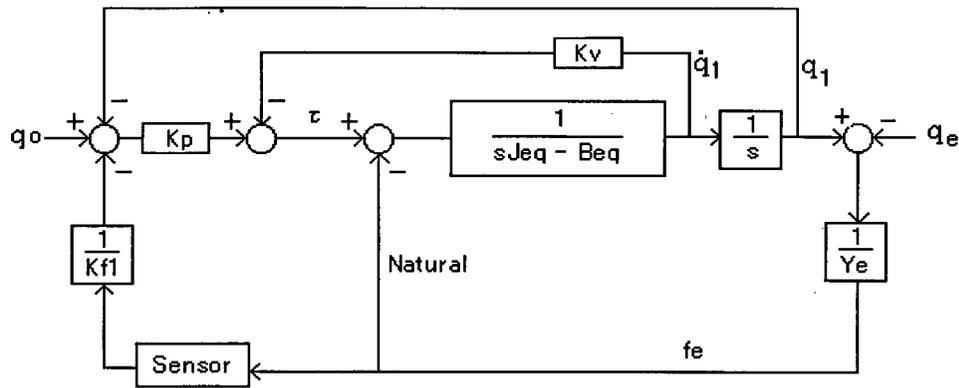


Figura 2.16 – Esquema de controle de rigidez aplicado em uma junta em contato com o meio

No diagrama, $\frac{1}{Y_e}$ e q_e representam o inverso da admitância e a posição do objeto ou meio respectivamente. K_p e K_{f1} possuem unidades de rigidez, ou seja o ganho para a relação força/velocidade e K_v é a unidade de amortecimento, agindo sobre a relação força/velocidade. Supondo que a junta encontra-se em contato com o meio, a equação do sistema resultará:

$$q [J_{eq} \cdot s^2 + (B_{eq} + K_v) s + K_p] = q_0 K_p - f_e \left(\frac{K_p}{K_{f1}} + 1 \right) \quad (2.28)$$

Para uma entrada q_0 constante e igual a \bar{q}_0 , a posição final da junta será:

$$q_1(\infty) = \bar{q}_0 - f_e(\infty) \left(\frac{K_p + K_{f1}}{K_p \cdot K_{f1}} \right) \quad (2.29)$$

Sendo $f_e(\infty)$ a força de contato em regime. Isolando o termo $f_e(\infty)$, tem-se:

$$f_e(\infty) = (\bar{q}_0 - q_1(\infty)) \left(\frac{K_p \cdot K_{f1}}{K_p + K_{f1}} \right) \quad (2.30)$$

e definindo a rigidez apresentada pela junta igual a K , sendo:

$$K = \frac{K_p \cdot K_{f1}}{K_p + K_{f1}} \quad (2.31)$$

A força de contato pode ser reescrita:

$$f_e(\infty) = K(\bar{q}_0 - q_1(\infty)) \quad (2.32)$$

Com o sistema em regime e supondo o meio homogêneo, linear e possível de ser descrito pela lei de Hooke, a força de contato, analisada a partir da reação do meio, será dada por:

$$f_e(\infty) = K_e(\bar{q}_0 - \bar{q}_e) \quad (2.33)$$

onde K_e é a rigidez apresentada pelo meio e \bar{q}_e é a posição final do meio.

A determinação da posição da junta em regime pode então ser definida a partir da igualdade entre as duas funções de f_e :

$$q_1(\infty) = \frac{\bar{q}_0 K + \bar{q}_e K_e}{K + K_e} \quad (2.34)$$

Analisando a força aplicada quando em regime, chega-se a:

$$f_e(\infty) = \left(\frac{K \cdot K_e}{K + K_e} \right) (\bar{q}_0 - \bar{q}_e) \quad (2.35)$$

Pode-se notar que as equações referentes a posição final da junta e a força de contato entre a junta e o meio, quando em regime, são similares às equações que definem a posição e a força de contato final entre duas molas com constantes de rigidez K e K_e . Tem-se assim, o conjunto junta-controlador comportando-se como uma mola de rigidez K e que no contato com um objeto de rigidez K_e , mudará a posição final atingida

para um certo valor $\bar{q}_1(\infty)$. O erro de posição será a diferença entre a posição atingida $\bar{q}_1(\infty)$ e a posição desejada q_1 .

Para analisar a influência dos ganhos K_p , K_v e K_f na alteração da impedância apresentada pela junta, quando sob a influência do controle, será utilizada a equação 2.5:

$$Z_1 = \frac{f_e}{q_v - q_1} \quad (2.36)$$

onde q_v é a posição final que seria atingida pela junta na condição de não existência do obstáculo. A posição q_v pode ser calculada através de:

$$q_v = q_0 \frac{1}{\frac{J_{eq}}{K_p} s^2 + \frac{B_{eq} + K_v}{K_p} s + 1} \quad (2.37)$$

substituindo em (2.37), obtêm-se:

$$Z_1 = \frac{J_{eq} K_{f1}}{K_p + K_{f1}} s^2 + \frac{(B_{eq} + K_v) K_{f1}}{K_p + K_{f1}} s + K \quad (2.38)$$

desta forma, chega-se a análise que o ganho K corresponde exatamente ao termo de rigidez da impedância implementada. Os termos relativos à frequência natural e amortecimento resultam em:

$$M_d = \frac{J_{eq} K_{f1}}{K_p + K_{f1}} \quad (2.39)$$

$$B_d = \frac{(B_{eq} + K_v) K_{f1}}{K_p + K_{f1}} \quad (2.40)$$

A frequência natural e o amortecimento não dependem de K_{f1} , pois este ganho influencia da mesma forma os três coeficientes da impedância, resultando em:

$$\omega_n = \sqrt{\frac{K_p}{J_{eq}}} \quad (2.41)$$

$$\xi = \frac{B_{eq} + K_v}{2\sqrt{K_p J_{eq}}} \quad (2.42)$$

A partir da escolha de K_p e K_v é possível fazer o ajuste do transitório do sistema, quando em movimentação no espaço livre. A partir da escolha de K_{fl} e K_p , é possível fazer a determinação da rigidez apresentada pela junta, sendo que o máximo valor de rigidez obtido a partir do ajuste de K_{fl} será K_p .

2.4.2 CONTROLE DE RIGIDEZ APLICADO A UM ROBÔ MANIPULADOR

Ao implementar o controle de rigidez para um robô composto de várias juntas acopladas, é necessário considerar que a rigidez desejada em cada grau de liberdade só poderá ser definida no espaço cartesiano, baseado no sistema do centro de flexibilidade [ASADA,88]. Encontram-se na literatura duas soluções para esta limitação: transformação da rigidez no espaço cartesiano para o espaço das juntas ou transformação das variáveis de posição ou força para o espaço das juntas.

A primeira forma a ser apresentada é baseada no trabalho de [BOISSONANT,88] na qual não é utilizada a medição da força para aplicação na lei de controle. É definida uma matriz diagonal que possui os valores de rigidez desejados para cada grau de liberdade no centro de flexibilidade. Para obter os torques ou forças que deverão ser fornecidos pelos atuadores de cada junta, inicialmente é feito uso da relação $F_a = K_d \Delta x$, determinando o vetor de forças e torques em função do deslocamento desejado e da rigidez imposta ao controlador. Em seguida, conhecendo a relação entre as coordenadas generalizadas das juntas, q , e as variações de posição do sistema de coordenadas alocado no centro de flexibilidade, Δx , representado pelo Jacobiano J :

$$\Delta x = J \Delta q \quad (2.43)$$

Chega-se a seguinte relação:

$$T = J^T F_a \quad (2.44)$$

e o vetor de torques T a ser fornecido pelos atuadores será expresso por:

$$T = J^T K_d J \Delta q = J^T K_d J (q_0 - q) \quad (2.45)$$

O esquema deste controlador é mostrado na figura 2.17.

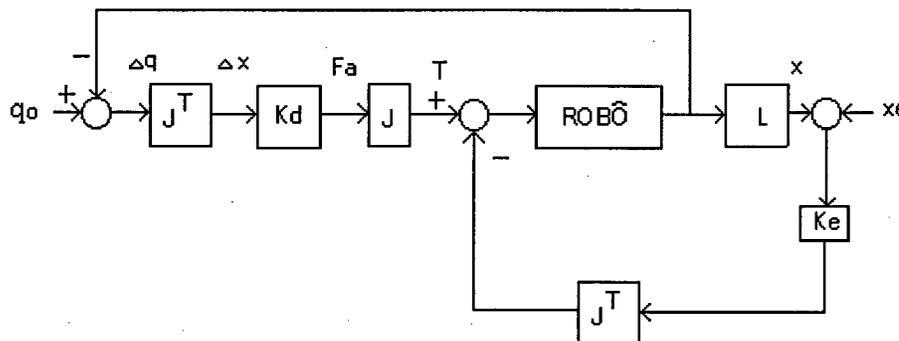


Figura 2.17: diagrama em blocos do controle de rigidez no espaço cartesiano

A rigidez implementada em nível de junta para obter a rigidez K_d em cada grau de liberdade referenciado ao centro de flexibilidade pode ser determinada a partir de:

$$K_q = J^T K_d J \quad (2.46)$$

A segunda forma de controle de rigidez para manipuladores com várias juntas é apresentada por [WHITNEY,85]. Ao contrário da apresentada anteriormente, é definida uma matriz de flexibilidade em relação a posição e força e utilizada na realimentação da força. Como a força é medida no espaço cartesiano, não é necessária a utilização de transformação para o espaço de juntas dos valores medidos. Este controle equivale ao controle de rigidez para uma junta apresentado no início deste capítulo, só que agora aplicado a um robô de várias juntas acopladas.

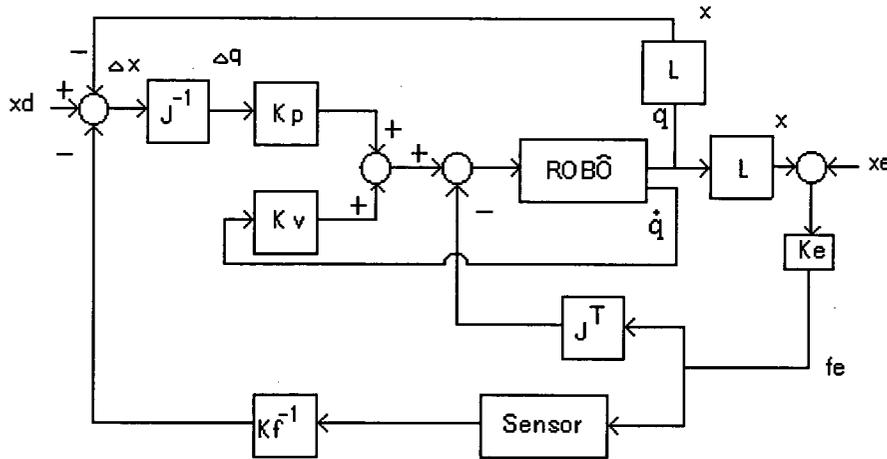


Figura 2.18: Esquema de controle de rigidez no espaço cartesiano segundo [WHITNEY,85]

Fazendo análise similar à desenvolvida no caso de uma junta isolada, a rigidez apresentada pelo robô pode ser calculada a partir das matrizes K_p e K_f^{-1} . Baseado em (2.35), temos:

$$K_p J^{-1} (x_d - x - K_f^{-1} f_e) - K_v \dot{q} - J^T f_e = M(q) \ddot{q} + V(q, \dot{q}) \dot{q} \quad (2.47)$$

Será desconsiderado o termo relativo à gravidade, permitindo assim chegar em uma expressão similar ao caso de uma junta. Considerando x_d e x_e constante, em regime obtém-se:

$$\begin{aligned} K_p J^{-1} (x_d - x(\infty) - K_f^{-1} f_e(\infty)) &= J^T f_e(\infty) \\ K_p J^{-1} (x_d - x(\infty)) &= J^T f_e(\infty) + K_p J^{-1} K_f^{-1} f_e(\infty) \\ J^{-T} K_p J^{-1} (x_d - x(\infty)) &= f_e(\infty) + J^{-T} K_p J^{-1} K_f^{-1} f_e(\infty) \end{aligned} \quad (2.48)$$

A matriz de ganhos de posição no espaço das juntas representado por:

$$K_{pc} = J^{-T} K_p J^{-1} \quad (2.49)$$

é a transformação do ganho representado no espaço operacional para o espaço das juntas, através da transformação inversa da equação (2.46).

Substituindo (2.49) em (2.48) e pré multiplicando o resultado por K_{pc}^{-1} , tem-se:

$$(x_d - x(\infty)) = (K_{pc}^{-1} \cdot K_f^{-1}) f_e(\infty) \quad (2.50)$$

e isolando a força na situação de regime:

$$f_e = (K_{pc}^{-1} + k_f^{-1})^{-1} (x_d - x) \quad (2.51)$$

A matriz K_f^{-1} pode ser escolhida como sendo diagonal mas K_{pc} , de modo geral, não é diagonal. Tem-se assim que os ganhos de posição K_p afetarão de forma dependente da configuração a rigidez que o robô apresenta na garra.

2.5 CONTROLE DE AMORTECIMENTO

O controle de amortecimento é uma forma de controle ativo de flexibilidade que foi desenvolvido por D. E. WHITNEY e apresentada em [WHITNEY, 77]. Está baseado na utilização de um sinal de velocidade de referência, o qual é modificado em função da força aplicada. A diferença conceitual com o controle de rigidez é que ao se modificar somente a velocidade em função da força ao invés da posição de referência, estará sendo modificado o amortecimento do sistema.

Uma aplicação do controle de amortecimento em uma junta é mostrado na figura 2.19:

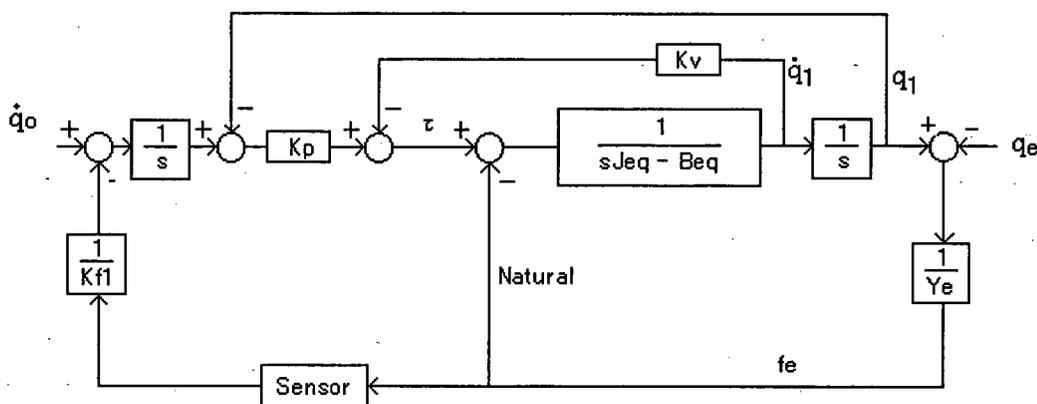


Figura 2.19: diagrama em blocos do controle de amortecimento aplicado a uma junta

A equação do sistema junta-controlador, em contato com o meio resulta em:

$$q[J_{eq} \cdot s^2 + (B_{eq} + K_v)s + K_p] = q_0 K_p - f_e \left(\frac{K_p}{K_{f2} \cdot s} + 1 \right) \quad (2.52)$$

Na situação em que não existe contato entre a junta e o meio, a equação passará a ser:

$$q_v = q_0 \frac{1}{\frac{J_{eq}}{K_p} s^2 + \frac{B_{eq} + K_v}{K_p} s + 1} \quad (2.53)$$

A impedância pode ser definida de maneira similar a (2.36),

$$Z = \frac{f_e}{q_v - q_0} = \frac{J_{eq} \cdot s^2 + (B_{eq} + K_v)s + K_p}{\frac{K_p}{K_{f2} \cdot s} + 1} \quad (2.54)$$

e definindo $B_p = \frac{K_p}{s}$, obtém-se:

$$Z = \frac{J_{eq} K_{f2}}{B_p + K_{f2}} s^2 + \frac{(B_{eq} + K_v) K_{f2}}{B_p + K_{f2}} s + \frac{K_p K_{f2}}{B_p + K_{f2}} \quad (2.55)$$

Um exemplo de aplicação deste controle para um sistema de várias juntas acopladas é descrito em [WHITNEY,85]. Quando sobre a ação do controle de amortecimento, o robô manipulador irá apresentar uma ação do seu efetuator final semelhante a um conjunto de amortecedores quando em contato com o meio. Nas direções em que ocorrerem contato, a velocidade tenderá a ser nula, enquanto que nas direções que o movimento não tiver restrições, a velocidade tenderá a se manter constante e de valor igual ao de referência.

2.6 ESTUDO DO CONTROLADOR DE IMPEDÂNCIA IMPLEMENTADO A PARTIR DO CONTROLE COM COMPENSAÇÃO PELA DINÂMICA INVERSA.

O manipulador é formado por um conjunto de estruturas físicas, sensores e atuadores, combinados com um conjunto de algoritmos de controle. Não é possível dizer que os dois sistemas podem ser isolados, ou seja, deve-se considerar o manipulador como sendo o conjunto de hardware e software, com características dinâmicas sendo influenciadas por ambas as partes. Desta maneira é possível descrever o sistema controlado completo através de um sistema físico equivalente.

O controle de impedância é baseado no conceito de que o controlador deve regular o comportamento dinâmico entre o movimento do manipulador e a força exercida sobre o ambiente, ao contrário de considerar os problemas de controle de movimento e força separadamente. Baseado neste conceito, o projetista especifica o comportamento dinâmico da interação entre o manipulador e o ambiente através da definição de uma impedância dinâmica generalizada para o manipulador.

A relação entre a velocidade desenvolvida pelo manipulador e a força aplicada ao meio será referenciada através da impedância mecânica imposta ao manipulador. Desta maneira, será atribuído um comportamento de segunda ordem ao manipulador, através da seleção dos ganhos das matrizes de inércia, amortecimento e rigidez aparente.

O controle de impedância exibe uma característica interessante: uma vez que o relacionamento dinâmico é atribuído, ele é capaz de tratar convenientemente bem uma classe de ambientes, controlando somente a variável de interação. Não produz uma ação de controle baseada em modelos detalhados do ambiente, mas numa regra de interação, o que resulta numa operação mais segura e autônoma do robô. Entretanto, não é capaz de especificar e controlar ambas as variáveis, força e posição.

Considerando o modelo do robô rígido sob a ação de uma força externa h :

$$M(q)\ddot{q} + V(q, \dot{q})\dot{q} + G(q) = u - J^T h \quad (2.56)$$

Faremos uma análise utilizando a técnica de controle pela dinâmica inversa, por ser uma estrutura de controle que fornece bons resultados de estabilidade e erro.

Iniciaremos a análise considerando a seguinte lei de controle:

$$u = M(q)y + n(q, \dot{q}) \quad (2.57)$$

onde $n(q, \dot{q}) = V(q, \dot{q})\dot{q} + G(q)$.

Este controle é baseado na idéia de se determinar o vetor de controle u , como sendo uma função dos estados do sistema, com a capacidade de realizar um relação entre entrada e saída linear. Ou seja, é desejado realizar não apenas uma linearização local, mas sim global do sistema dinâmico, obtido a partir de uma realimentação de estado não linear. [SCIAVICCO e SICILIANO, 1996].

Na presença de forças do efetuador final, o manipulador controlado será descrito por:

$$M(q)\ddot{q} + V(q, \dot{q})\dot{q} + G(q) = M(q)y + V(q, \dot{q})\dot{q} + G(q) - J^T h \quad (2.58)$$

pré multiplicado por $M^{-1}(q)$ resulta em:

$$\ddot{q} = y - M^{-1}(q)J^T h \quad (2.59)$$

Note que se a força de contato h , for nula, teremos $\ddot{q} = y$, resultado obtido através da técnica do controle inverso aplicado ao controle de posição. Porém, se a força de contato h não for nula, a presença dela causará um acoplamento do sistema.

Continuando a análise, será utilizada uma entrada de controle que permita a modelagem do sistema, caracterizando-o como um sistema de segunda ordem, quando da ocorrência de força de contato ($h \neq 0$):

$$y = J_A^{-1}(q)M_d^{-1}[M_d\ddot{x}_d + B_d\dot{\tilde{x}} + K_d\tilde{x} - M_d J_A(q, \dot{q})\dot{q}] \quad (2.60)$$

Aplicando-se (2.59) e (2.60) na equação A.5, da aceleração do efetuador final no espaço operacional, representado por \ddot{x} , resulta em:

$$\begin{aligned} \ddot{x} &= J_A(q)\ddot{q} + \dot{J}_A(q, \dot{q})\dot{q} \\ \ddot{x} &= J_A(q)[J_A^{-1}(q)M_d^{-1}(M_d\ddot{x}_d + B_d\dot{\tilde{x}} + K_d\tilde{x} - M_d J_A(q, \dot{q})\dot{q}) - M^{-1}(q)J^T(q)h] + \dot{J}_A(q, \dot{q})\dot{q} \\ \ddot{x} &= M_d^{-1}(M_d\ddot{x}_d + B_d\dot{\tilde{x}} + K_d\tilde{x} - M_d J_A(q, \dot{q})\dot{q}) - J_A(q)M^{-1}(q)J^T(q)h + \dot{J}_A(q, \dot{q})\dot{q} \\ M_d\ddot{\tilde{x}} + B_d\dot{\tilde{x}} + K_d\tilde{x} &= M_d J_A(q)M^{-1}(q)J_A^T(q)T_A^T h \\ M_d\ddot{\tilde{x}} + B_d\dot{\tilde{x}} + K_d\tilde{x} &= M_d M_A^{-1}(q)h_A \end{aligned} \quad (2.61)$$

onde $M_A = J_A^{-T}(q)M J_A^{-1}$ e $h_A = T_A^T h$

Tem-se assim a relação entre uma força (impedância), e a dinâmica do erro do sistema. M_d , B_d , e K_d são matrizes que caracterizam completamente o sistema. A escolha das matrizes é feita a partir da escolha da dinâmica desejada para o sistema. Faz-se a alocação dos pólos com a determinação da frequência natural de oscilação e o termo de amortecimento, determinando desta maneira qual será o valor da impedância equivalente apresentada pelo robô manipulador quando em contato com o meio.

Analisando a equação dinâmica do sistema em malha fechada (2,61), nota-se que o sistema é acoplado, pois a matriz $M_A^{-1}(q)$ é uma matriz cheia. Para fazer o desacoplamento, não é possível utilizar a mesma lei de controle do sistema sem atuação de força externa, oriundo do controle pela dinâmica inversa.

Faz-se necessária a escolha de uma nova lei de controle, para que ocorra o desacoplamento do sistema em malha fechada.

A lei de controle proposta para permitir o desacoplamento do sistema é dada por:

$$u = M(q)y + V(q, \dot{q})\dot{q} + G(q) + J^T(q)h \quad (2.62)$$

Note que será necessário medir a força que atua no efetuador final.

Substituindo esta lei de controle no modelo do robô rígido, tem-se:

$$M(q)\ddot{q} + V(q, \dot{q})\dot{q} + G(q) = M(q)y + V(q, \dot{q})\dot{q} + G(q) + J^T(q)h - J^T(q)h \quad (2.63)$$

obtendo como resultado da aplicação da lei de controle, a equivalência: $y = \ddot{q}$, desta maneira, faz-se o desacoplamento do sistema, através da compensação da força de interação.

Após determinada a lei de controle que desacopla o sistema, faz-se a aplicação do controle, como se segue:

$$y = J_A^{-1}(q)M_d^{-1}(M_d\ddot{x}_d + B_d\dot{\tilde{x}} + K_d\tilde{x} - M_d\dot{J}_A(q, \dot{q})\dot{q} - h_A) \quad (2.64)$$

Sob esta ação de controle, a dinâmica do sistema em malha fechada resultará em:

$$M_d\ddot{\tilde{x}} + B_d\dot{\tilde{x}} + K_d\tilde{x} = h_A \quad (2.65)$$

Do mesmo modo, o comportamento do erro do manipulador para uma desejada força de contato h_A , pode ser determinado através da escolha da dinâmica do erro, com a escolha apropriada para as matrizes M_d , B_d , e K_d .

Analisando o comportamento do sistema, quando em equilíbrio, ou seja, $\ddot{\tilde{x}} = \dot{\tilde{x}} = 0$, teremos:

$$K_d\tilde{x}_\infty = h_A \rightarrow \tilde{x}_\infty = K_d^{-1}h_A \quad (2.66)$$

Percebe-se que no equilíbrio, o sistema se comportará como uma mola, similar ao resultado obtido através do controle de rigidez. A grande vantagem do controle de impedância é a de permitir a escolha, através de M_d , B_d , e K_d , da dinâmica do erro.

Uma característica do controle de impedância demonstrado é a utilização de sinais de referência para posição, velocidade e aceleração no espaço operacional. Porém o controle irá atuar no espaço de juntas, utilizando para a transformação entre os espaços, a matriz Jacobiana do robô SCARA, apresentada no apêndice A.

Os valores de posição e velocidade são transformados do espaço de juntas para o espaço operacional através da aplicação da matriz Jacobiana e de sua derivada, são comparados com os valores de referência, e na aplicação da lei de controle, utiliza-se a inversa do Jacobiano para transformar os sinais de controle para o espaço de juntas, os quais serão aplicados para o acionamento das juntas do robô.

3 SIMULAÇÕES DO CONTROLE DE IMPEDÂNCIA PARA UM ROBÔ SCARA.

Antes de iniciar a implementação do controle de impedância para o robô SCARA do Laboratório de Robótica, desenvolveu-se um programa de simulação para verificar a eficiência da estrutura de controle, e para permitir um melhor conhecimento do controle, não correndo assim o risco de danificar o robô, com manobras não previstas.

Foram feitas várias simulações de duas situações distintas de funcionamento do sistema sob a ação do controle de impedância:

- Movimentação sem restrição;
- Movimentação com restrição partindo em contato com o meio;

Em seguida, repetiu-se as simulações para a situação de presença de variação paramétrica.

Outra utilização a que se destina este simulador é o uso didático. Para isto foram desenvolvidas rotinas de escolhas de parâmetros, exibição de resultados, etc., através de menus de seleção, para o ambiente do Simulink.

3.1 MODELAGEM DO ROBÔ SCARA

Inicialmente fez-se a modelagem da cinemática e da dinâmica do robô SCARA, conforme apresentado no apêndice A, utilizando estes dados para o desenvolvimento do simulador, para que os resultados teóricos pudessem se aproximar de uma maneira mais precisa dos resultados práticos.

O robô a ser utilizado nos experimentos práticos, e simulado durante este capítulo, é um robô manipulador do tipo SCARA, fabricado pelo Instituto de Robótica da Universidade Técnica de Zurique(ETH), na Suíça. Instalado no Laboratório de Robótica na Universidade Federal de Santa Catarina (UFSC).

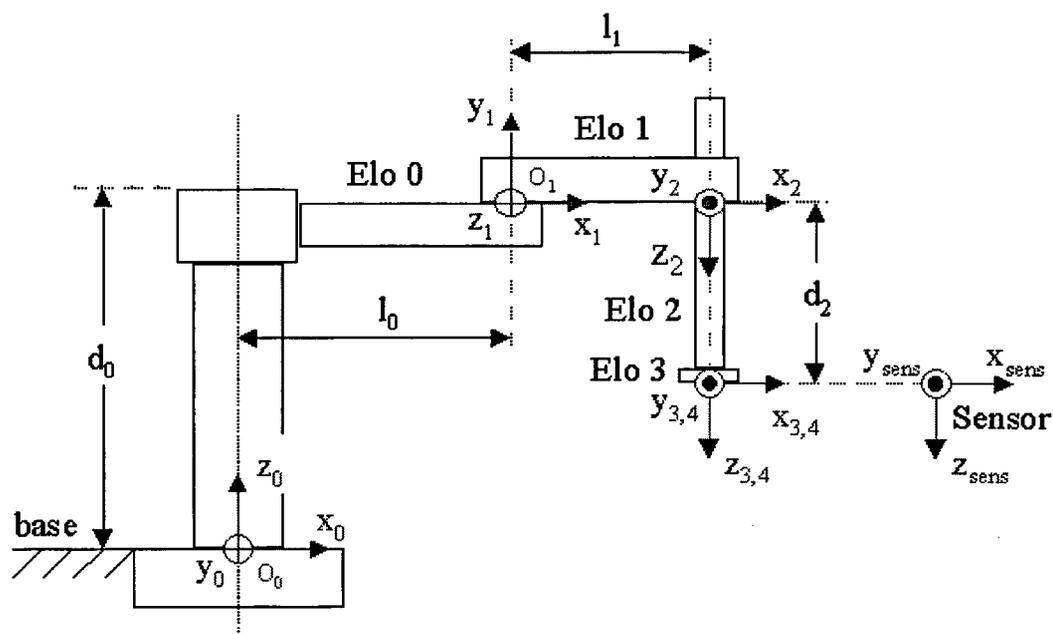


Figura 3.1: Representação das juntas de um robô do tipo SCARA

As duas primeiras juntas do robô SCARA são de rotação, girando em torno dos eixos verticais, paralelos a base do robô, trabalham em um plano horizontal (elo 0 e 1, figura 3.1). A terceira junta é de translação, realizando movimentos perpendiculares ao plano formado pelas duas primeiras juntas (elo 2, figura 3.1). A quarta junta executa movimentos de rotação, em torno do eixo de translação da terceira junta (elo 3, figura 3.1).

O robô SCARA a ser utilizado possui então quatro graus de liberdade, porém nas simulações serão utilizadas somente as duas primeiras juntas.

3.2 IMPLEMENTAÇÃO DO CONTROLE DE IMPEDÂNCIA

Tendo as características do robô previamente determinadas (Apêndice A), e a estrutura do controle de impedância a ser utilizado já definidas (Capítulo 2.6), desenvolveu-se o simulador para ambiente Simulink.

Vários trabalhos [HOGAN, 1985], [SCIAVICO e SICILIANO, 1996], trazem o diagrama de blocos que representa o manipulador em contato com um meio elástico sob o controle de impedância, conforme mostrado na figura 3.2, representando apenas a

dinâmica de um manipulador em contato com o meio, sob a ação do controle de impedância.

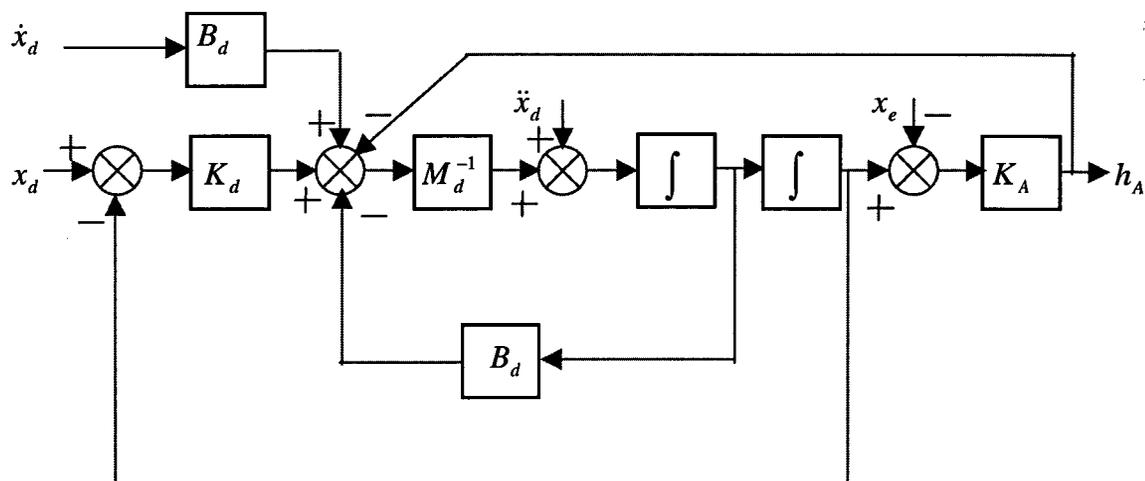


Figura 3.2: Diagrama de blocos representando a dinâmica de um manipulador sobre controle de impedância

A partir da estrutura de controle determinada no capítulo 2.6, desenvolveu-se o simulador do controle de impedância, conforme a figura 3.3.

O programa foi desenvolvido de forma modular, de modo a permitir o seu aproveitamento no desenvolvimento de outros simuladores, sendo preciso para isto, apenas o desenvolvimento de blocos ainda não existentes, ou ainda, a reorganização dos blocos já existentes. Os blocos desenvolvidos são expostos no Apêndice 2, com a listagem, dados de entrada e saída e características de utilização.

Os valores desejados de posição, velocidade e aceleração são dados no espaço operacional, referenciados ao sistema de coordenadas da base. Assim como a posição do meio também é dada no espaço operacional e referenciada na mesma base.

Um problema inerente aos controladores baseados no espaço operacional é a ocorrência de pontos de singularidade, o que não ocorre no controle no espaço de juntas. Para evitar este problema, pode-se criar a trajetória no espaço de juntas, e em seguida fazer a transformação de espaço, ou ainda utilizar algoritmos de verificação de proximidade de ponto de singularidade.

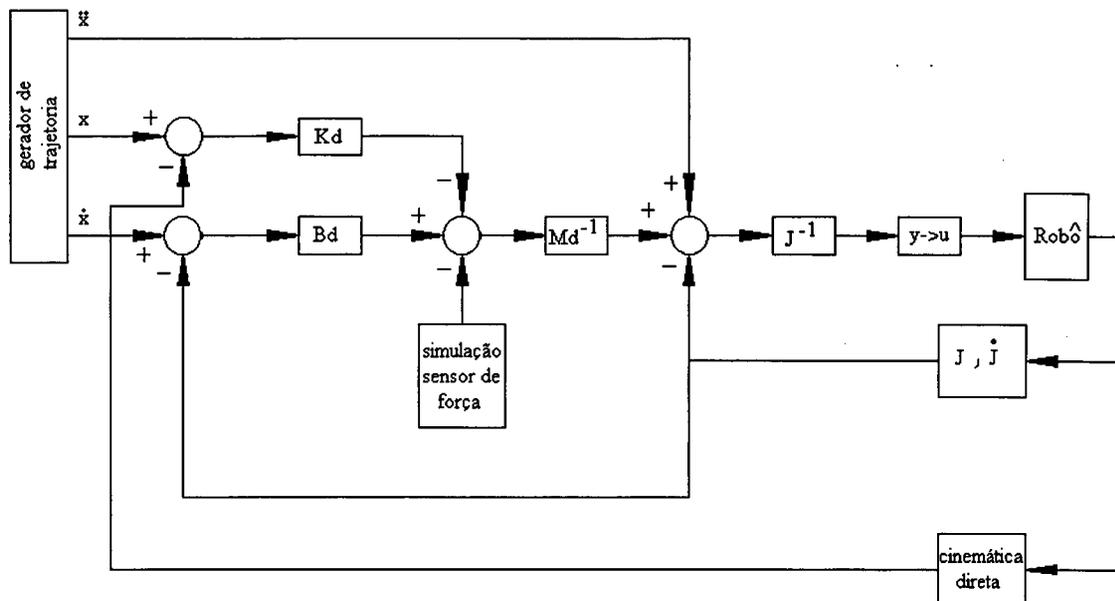
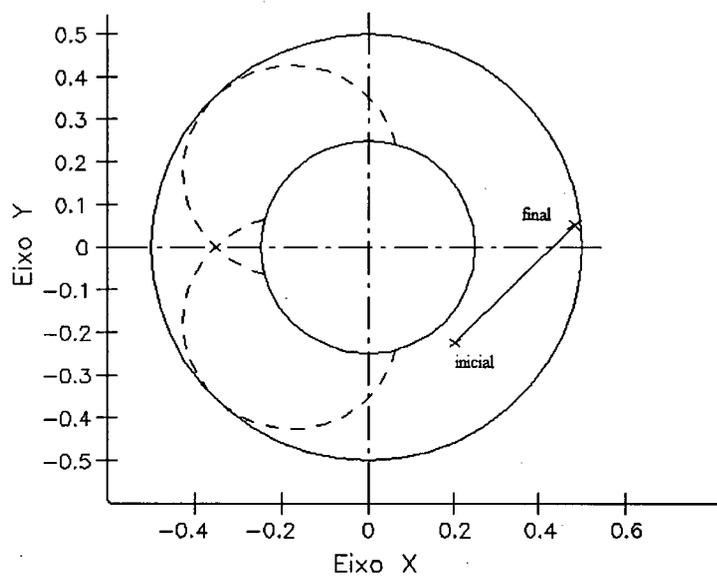


Figura 3.3: Diagrama de blocos do simulador de controle de impedância aplicado em um robô do tipo SCARA

Uma alternativa para se evitar os pontos de singularidade é testar a trajetória desejada, verificando se a mesma contém algum ponto de singularidade.



	X	Y
Posição inicial	0.2042	-0.221
Posição final	0.484	0.0498

Figura 3.4: Valores iniciais e finais nas direções X e Y.

Analisando o espaço da tarefa do robô SCARA instalado no laboratório de Automação, e com a barra fixada (figura 3.26), a qual apresentará resistência ao efetuador final do robô durante sua movimentação no seguimento da trajetória desejada, determinou-se as posições iniciais e finais correspondentes aos extremos da barra, descritos na tabela da figura 3.4. Para isto, foi utilizado a interface do Xoberon que retorna a posição atual do efetuador final.

Utilizando estes valores também para as simulações, obteve-se duas vantagens:

- Melhor comparação de resultados de simulação e de experimento;
- Antever problemas de singularidade, que para este caso, não ocorrem, devido a escolha adequada da posição de fixação da barra.

No seguimento da trajetória escolhida não ocorrerão pontos de singularidade, pois tratando-se de um robô do tipo SCARA, deve-se evitar somente a singularidade do cotovelo, ou seja, quando ocorre o alinhamento da primeira com a segunda junta, no movimento seguinte, a junta 1 em conjunto com a junta 2 podem atingir a posição desejada com duas combinações diferentes de ângulo.

Determinou-se através da escolha do local de fixação da barra, uma trajetória na qual não ocorre o alinhamento entre as juntas 1 e 2.

A geração da trajetória a ser seguida leva em consideração a posição inicial e final desejada, assim como o tempo que deve ser usado para descrever esta trajetória. A aceleração e a velocidade são calculadas para que seja efetuado o movimento desejado no tempo estipulado. A aceleração terá valor constante, a velocidade será uma rampa e a posição seguirá uma parábola.

3.3 RESULTADOS OBTIDOS NAS SIMULAÇÕES

São apresentados a seguir os resultados obtidos nas simulações de movimentação com e sem restrição e com e sem variação paramétrica.

3.3.1 MOVIMENTAÇÃO SEM RESTRIÇÃO E SEM VARIAÇÃO PARAMÉTRICA

Nesta primeira seção de simulação, buscou-se adquirir um maior conhecimento do comportamento do sistema sob a ação do controlador e verificar o bom funcionamento do programa de simulação. Não foram consideradas nesta fase de simulação, a existência de perturbações e variações paramétricas.

Os ganhos do controlador foram definidos a partir da análise do sistema sem contato com o meio:

$$M_d \ddot{x} + B_d \dot{x} + K_d x = K_d x_d \quad (3.1)$$

A partir dos valores desejados para a frequência natural ω , e para a constante de amortecimento ζ , determinaram-se os ganhos M_d , K_d e B_d [SCIAVICCO E SCICILIANO,96]:

$$\omega = 5 \text{ rad/s}$$

$$\zeta = 0.5$$

$$M_d = 100$$

$$K_d = 500$$

$$B_d = 2500$$

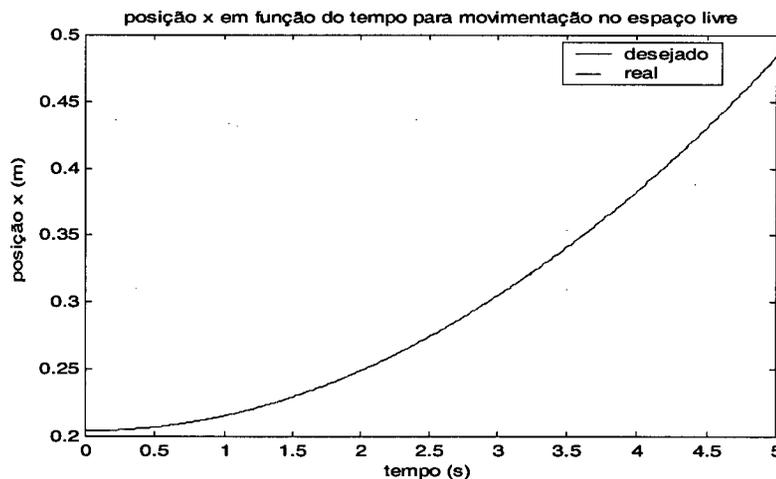


Figura 3.5: posição do efetuador final em relação a direção X em função do tempo na movimentação sem contato com o meio

A partir dos pontos iniciais e finais programados, foi gerado a trajetória nas direções x e y, de tal maneira que entre os pontos iniciais e finais a trajetória a ser

seguida fosse uma reta. As oscilações decorrentes da escolha de $\zeta = 0.5$ apresentaram valor muito reduzido, sendo imperceptíveis no gráfico sem ampliação, fato que pode ser relacionado principalmente com o fato de o ponto inicial da trajetória desejada ser coincidente com o ponto inicial do robô, produzindo assim um erro inicial nulo.

Caso a posição inicial do robô fosse diferente da posição inicial desejada, ou ainda se a trajetória desejada possui-se um degrau, teríamos a presença de uma oscilação com maior amplitude.

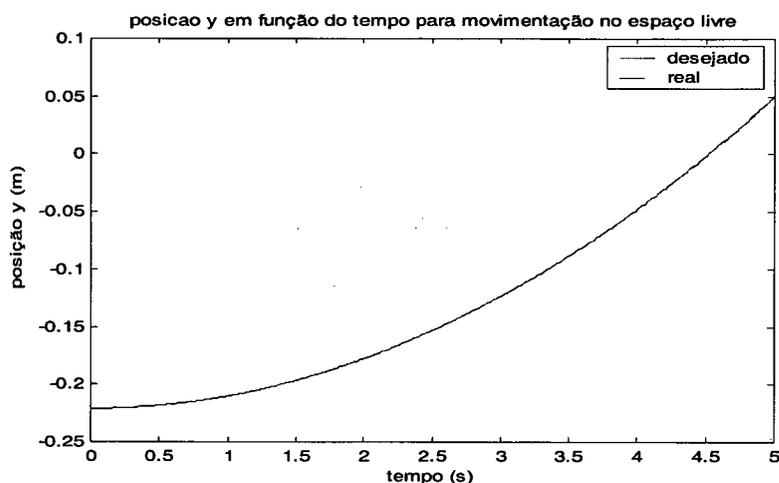


Figura 3.6: posição do efetuador final em relação a direção Y em função do tempo na movimentação sem contato com o meio

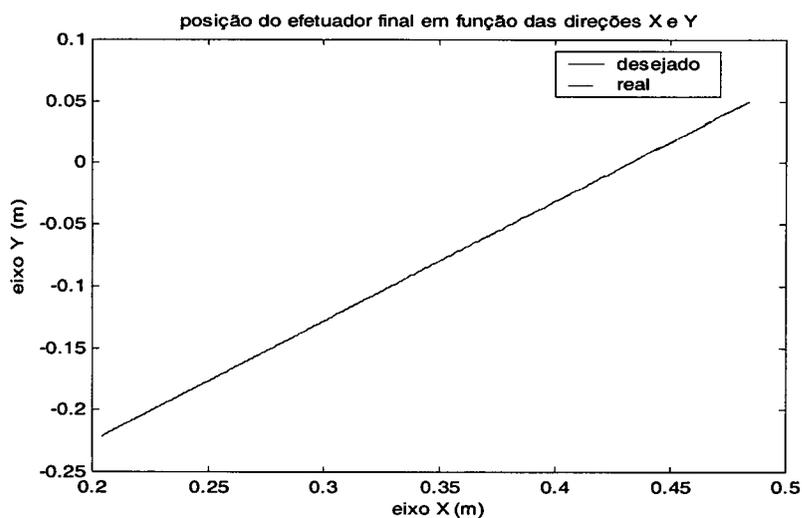


Figura 3.7: posição do efetuador final em relação as direções X e Y em movimento sem contato com o meio.

Como observado, as trajetórias definidas pelo módulo gerador de trajetória descrito no apêndice B.2, a partir dos pontos iniciais e finais, apresentaram na direção X e Y o formato de uma parábola, e o controle, na situação de ausência de contato com o meio, levou o robô manipulador ao seguimento da trajetória definida, com um erro de posição de ordem de 10^{-5} (m). Quando não há contato com o meio, a estrutura de controle torna-se equivalente ao controle pela dinâmica inversa, estrutura utilizada na malha de controle de posição. Sendo que a malha externa, de realimentação de força não apresenta influência sobre o controle, a partir do momento que o sinal de força lido é nulo.

3.3.2 MOVIMENTAÇÃO COM RESTRIÇÃO PARTINDO EM CONTATO COM O MEIO, SEM VARIAÇÃO PARAMÉTRICA

Este experimento tem como objetivo comprovar o funcionamento do controle de impedância atuando em um manipulador em contato com o meio. Na simulação, determinou-se que a posição desejada nas direções X e Y encontra-se um milímetro para dentro do meio na direção X e dois milímetros na direção Y, e que na direção Z não ocorre contato com o meio. E ainda que o meio apresenta uma resistência nas direções X e Y de $10^4 N/m$.

Utilizando os mesmos pontos iniciais e finais para a trajetória desejada, obteve-se os seguintes resultados:

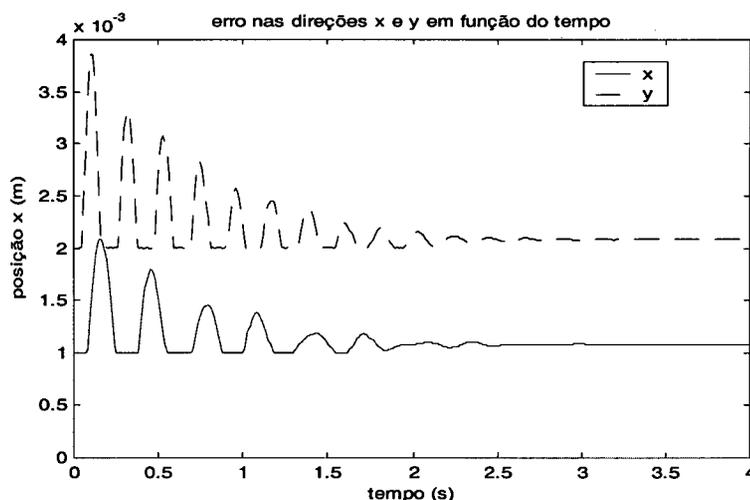


Figura 3.8: Erro nas direções X e Y, para o robô em movimento em contato com o meio

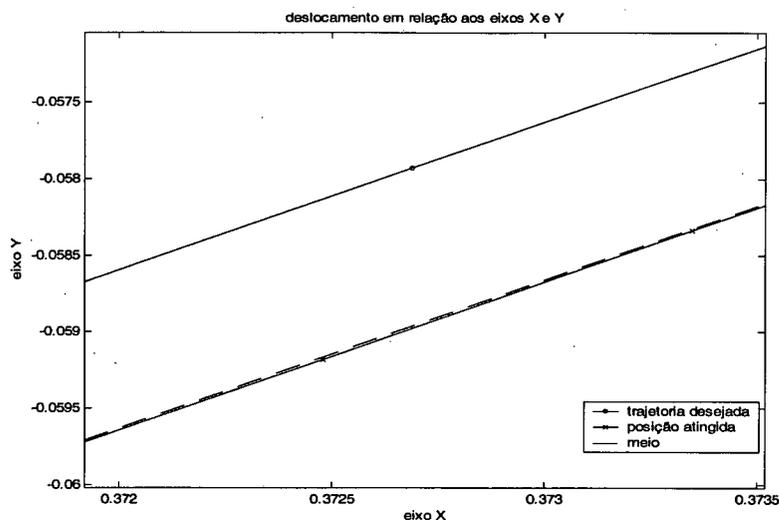


Figura 3.9: gráfico ampliado da posição desejada em relação a posição atingida e o meio

Nota-se que tanto na direção x quanto na direção y , onde o meio apresenta uma dureza imposta através de K_A igual e os ganhos do controlador de impedância também possuem valores iguais, o período de amortecimento foi equivalente nas duas direções, havendo uma pequena diferença quanto a frequência de oscilação, devido a posição desejada ser diferente nas direções x e y . Como nas duas direções os ganhos foram ajustados com igual valor, tem-se assim a comprovação da definição da dinâmica do erro em função da força de contato, com o decaimento independentemente da posição do meio.

No gráfico do deslocamento, apresentado na figura 3.9 de maneira ampliada para permitir uma melhor observação, pode-se ver que a trajetória foi seguida a contento, com um seguimento da superfície de contato. Como a trajetória estava localizada dentro da superfície, o resultado obtido foi o seguimento da trajetória limitada a superfície.

Analisando os gráficos de força nas direções X e Y (figuras 3.10 e 3.11), tem-se novamente que a única diferença entre as respostas obtidas é devido a diferença na posição desejada em relação ao meio. No primeiro caso, na direção X , por ter uma menor diferença entre a posição desejada e o meio, imposta com valor de 1 milímetro, o manipulador iniciando o movimento em contato com a superfície, gera uma força de contato com um valor inicial menor em relação a força obtida na direção Y .

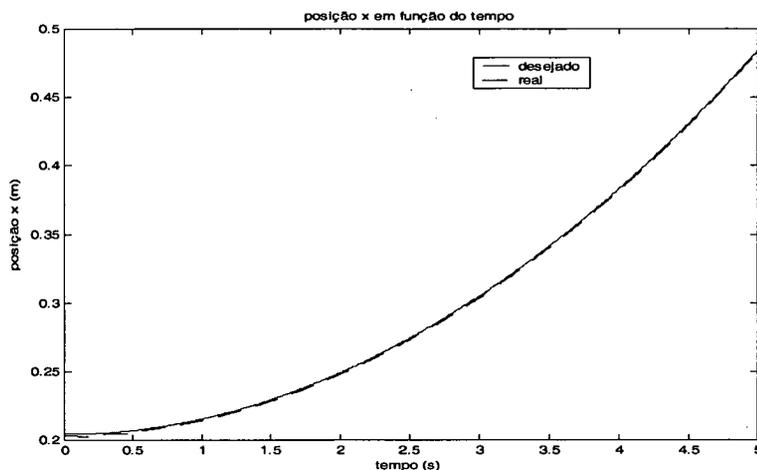


Figura 3.10: posição do efetuator final em relação a direção X em função do tempo na movimentação em contato com o meio

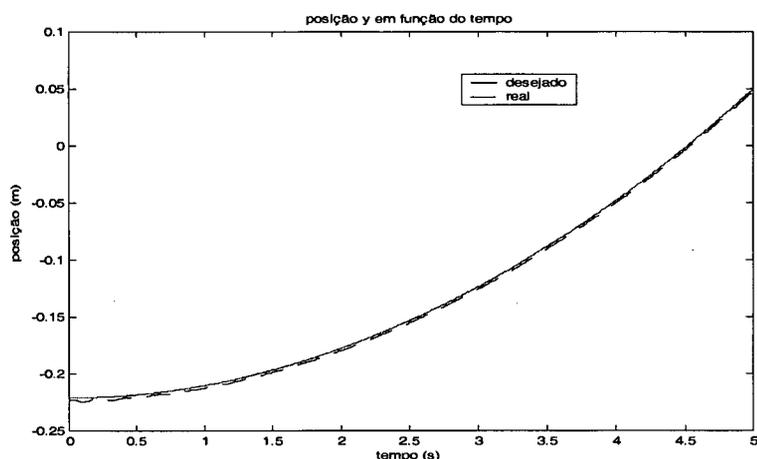


Figura 3.11: posição do efetuator final em relação a direção Y em função do tempo na movimentação em contato com o meio

O erro de posição (figura 3.8) apresentado na direção y é o dobro em relação ao mesmo valor apresentado na direção x, fato devido a escolha da posição desejada para y ser de 2 milímetros para dentro do meio e para x de 1 milímetro para dentro do meio, e com o meio apresentando a mesma rigidez nas duas direções. Pelo mesmo motivo, a força de contato na direção y (figura 3.13), apresentou como valor inicial uma amplitude igual ao dobro da apresentada na direção x (figura 3.12). Apresentando esta relação quase que continuamente, durante a período de amortecimento do erro, que ocorre durante dois segundos de simulação.

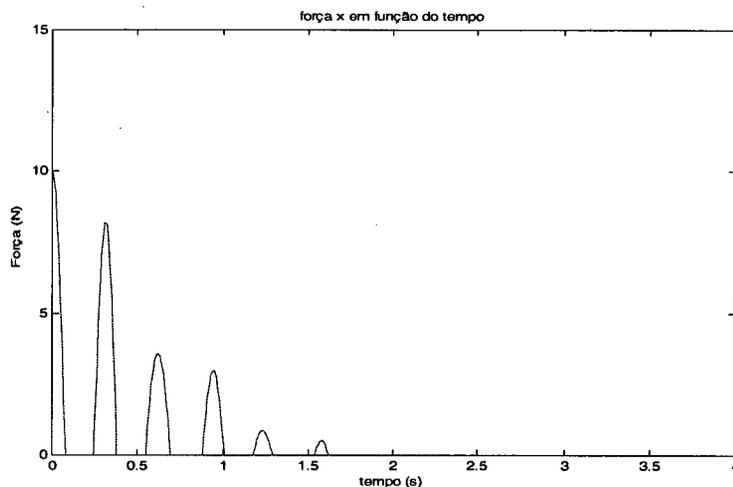


Figura 3.12: força de contato na direção x em função do tempo

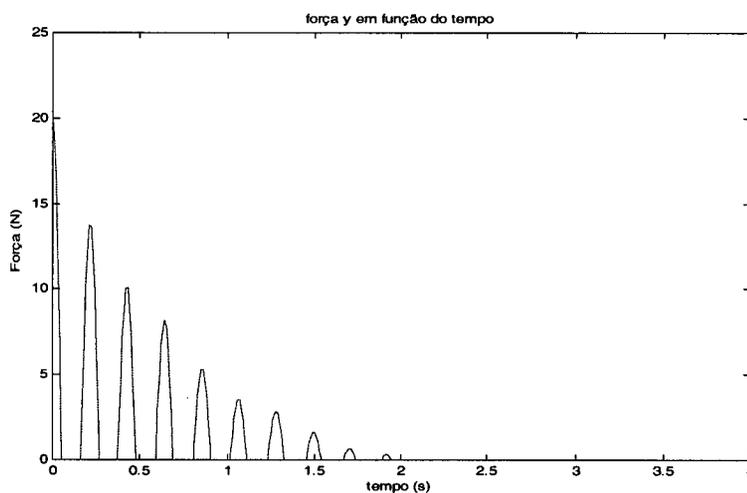


Figura 3.13: força de contato na direção y em função do tempo

Os valores de força nas direções X e Y apresentados inicialmente, não foram novamente alcançados durante a simulação, sendo que este valor foi imposto, devido a posição inicial e a rigidez imposta ao meio, por se tratar de uma simulação. Tem-se assim demonstrado uma característica importante do controle de impedância, que é o seguimento de trajetórias, e quando em contato com o meio, ocorre a regulação da força de contato, através da alteração da referência de posição. Através da escolha apropriada dos ganhos do controlador, pode-se definir a velocidade de amortecimento da força, e ainda a frequência de oscilação, podendo-se assim, utilizar o controle para diversos fins, como por exemplo, o transporte de cargas perigosas ou a manipulação de equipamentos em campos com possíveis obstáculos.

3.3.3 MOVIMENTAÇÃO SEM RESTRIÇÃO COM VARIAÇÃO PARAMÉTRICA

Após verificado o funcionamento do controle de impedância em duas situações distintas, têm-se como objetivo verificar o comportamento do sistema de controle na ocorrência de variações paramétricas.

A estrutura do controle de impedância implementado está baseado em uma malha interna de controle de posição, constituído pelo controle da dinâmica inversa, e uma malha externa, que faz a alteração da referência de posição baseado na força de contato. Neste experimento, será feito a alteração de parâmetros de compensação dinâmica na estrutura de controle pela dinâmica inversa, com o intuito de simular uma variação entre os parâmetros reais do robô e os que são utilizados para a linearização na geração do sinal de controle.

Utilizou-se fatores de multiplicação variando entre 0,9 e 1,1 para alguns elementos das matrizes que representam a inércia do manipulador e a matriz das forças de Coriolis e Centrífuga, conforme mostrado na tabela da figura 3.14. Este local para inserção da perturbação foi escolhido devido ao tipo de controle utilizado e por que durante as simulações anteriores, as matrizes que representam a dinâmica do manipulador foram utilizadas tanto para a realimentação linearizante como também para a simulação do robô.

Desta maneira, diferencia-se a dinâmica utilizada para representar o robô e os valores utilizados para a realimentação. Através da variação paramétrica, pode-se obter dados mais próximos de um experimento prático, pois dificilmente consegue-se uma modelagem que reflita a dinâmica do robô para a utilização em um experimento prático.

Matriz de Inércia M		Matriz de Forças de Coriolis e Centrífuga	
Valores Alterados	Valores originais	Valores Alterados	Valores originais
$0.92 \times M(1,1)$	$M(1,1)$	$1.10 \times C(1,1)$	$C(1,1)$
$0.94 \times M(1,2)$	$M(1,2)$	$0.91 \times C(2,1)$	$C(2,1)$

Figura 3.14 – Valores dos desvios na fase de simulação

Como pode-se ver a partir dos valores na figura 3.14, foram atribuídos pesos a dois componentes de cada uma das matrizes.

Repetiu-se o experimento de movimentação sem contato com o meio, apresentado em 3.3.1, obtendo-se os seguintes resultados:

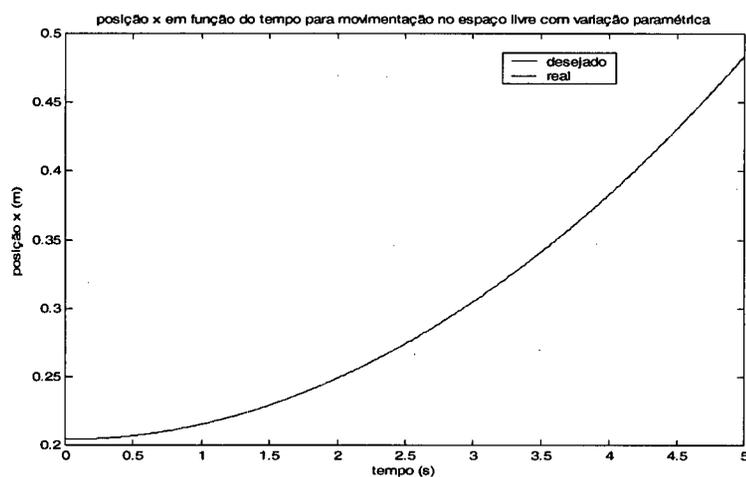


Figura 3.15: posição do efetuador final em relação a direção X em função do tempo na movimentação sem contato com o meio e com variação paramétrica

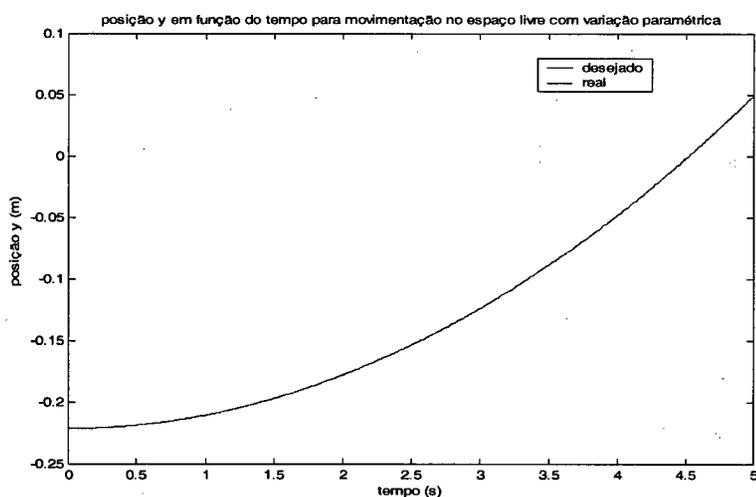


Figura 3.16: posição do efetuador final em relação a direção Y em função do tempo na movimentação sem contato com o meio e com variação paramétrica

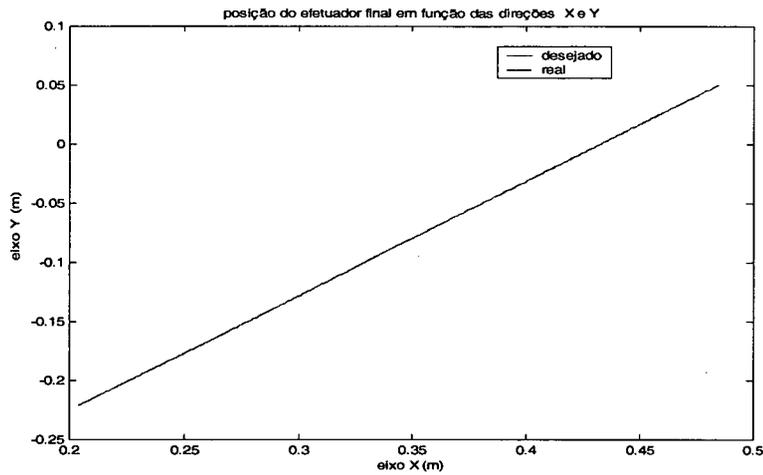


Figura 3.17: posição do efetuador final em relação as direções X e Y em movimento sem contato com o meio e com variação paramétrica

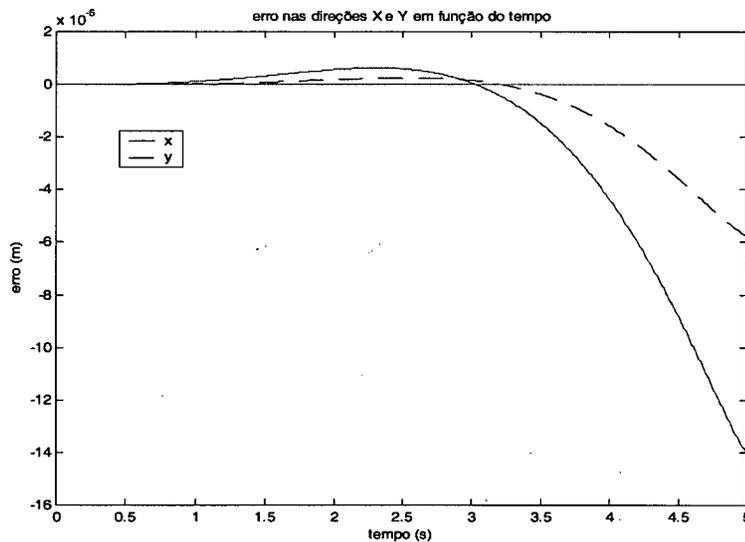


Figura 3.18: erro nas direções X e Y em movimento sem contato com o meio e com variação paramétrica.

Através da verificação do gráfico da figura 3.18, que apresenta o erro de posicionamento do efetuador final em relação aos eixos X e Y, na presença de variação paramétrica, nota-se que o sistema apresenta um erro crescente durante toda a movimentação do manipulador. Fato que demonstra a necessidade de um bom conhecimento da dinâmica do manipulador, para permitir a compensação dos termos não lineares pelo controle. Para uma variação paramétrica maior, ou ainda para ganhos mais elevados, pode-se ter a instabilidade do sistema.

3.3.4 MOVIMENTAÇÃO COM RESTRIÇÃO PARTINDO EM CONTATO COM O MEIO E VARIAÇÃO PARAMÉTRICA

Após verificado o funcionamento do controle em uma situação de variação paramétrica em uma movimentação livre, repetiu-se o experimento, porém simulando o contato do efetuador final com o meio. Para isto, determinou-se que a posição desejada encontra-se um milímetro para dentro do meio na direção X e dois milímetros na direção Y, e ainda que o meio apresenta uma resistência nas direções X e Y de $10^4 N/m$. Utilizou-se os mesmos parâmetros do experimento anterior para permitir uma melhor comparação entre os resultados obtidos.

Manteve-se os ganhos utilizados nos experimentos anteriores e ainda a mesma alteração dos parâmetros, conforme descrito na seção 3.3.3, obtendo-se os seguintes resultados:

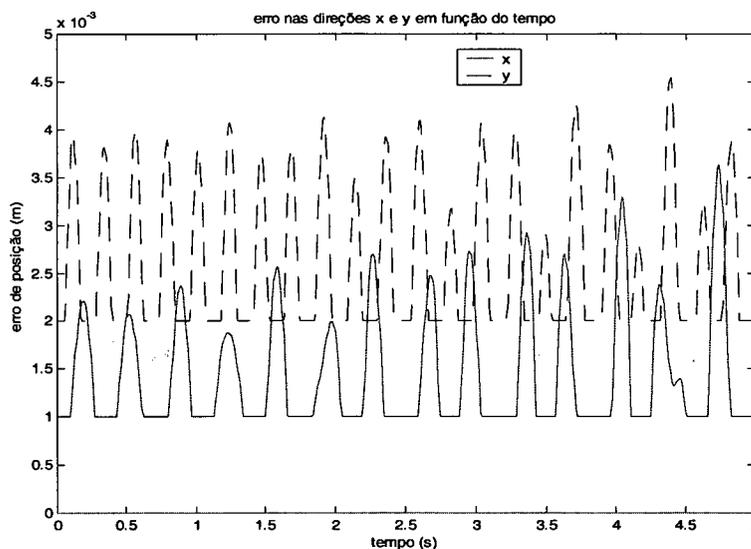


Figura 3.19: Erro nas direções X e Y, para o robô em movimento em contato com o meio e variação paramétrica

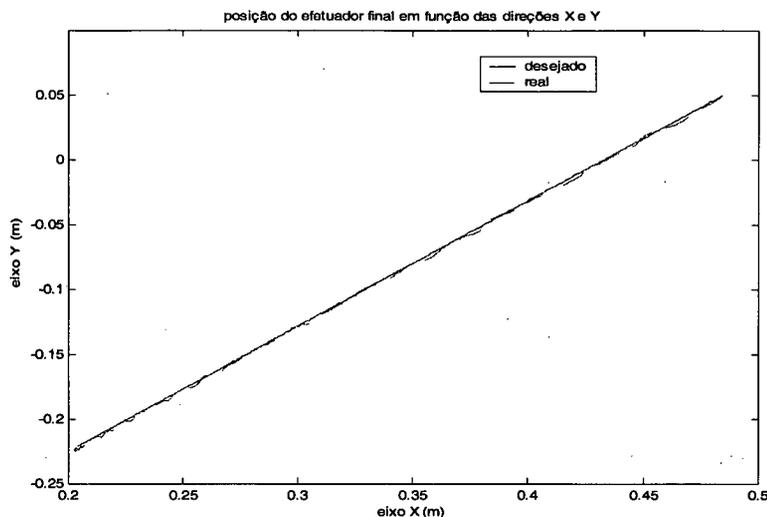


Figura 3.20: posição do efetuador final em relação as direções X e Y em movimento em contato com o meio e variação paramétrica.

Nota-se que de maneira similar ao experimento anterior, o robô manipulador sobre a ação do controle de impedância, mesmo sobre a influência de variação paramétrica imposta, apresenta um seguimento da trajetória limitada ao meio, porém o erro não apresentou um amortecimento similar ao experimento sem variação paramétrica, fato devido a não compensação de toda a dinâmica apresentada pelo robô. Este fato demonstra a importância de uma modelagem do robô o mais próxima possível dos valores reais, para assim permitir a utilização do controle baseado na compensação da dinâmica inversa do robô manipulador.

Analisando os gráficos de força, nas direções X e Y, pode-se notar que similar ao erro de posicionamento, o decaimento da força não ocorreu de maneira igual ao do experimento anterior, ocorrendo uma oscilação no contorno de decaimento. Novamente, este fato é devido a presença de não linearidades do robô manipulador que não foram devidamente anuladas através da ação de controle.

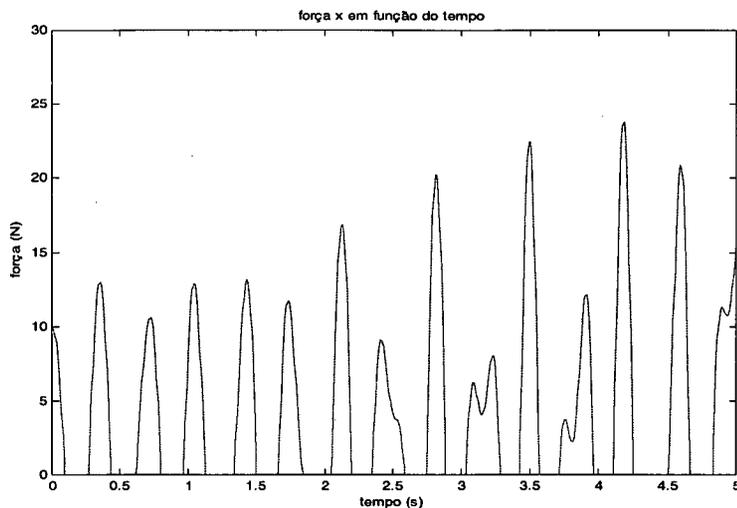


Figura 3.21: força de contato na direção X em função do tempo para a movimentação do robô em contato com o meio e variação paramétrica

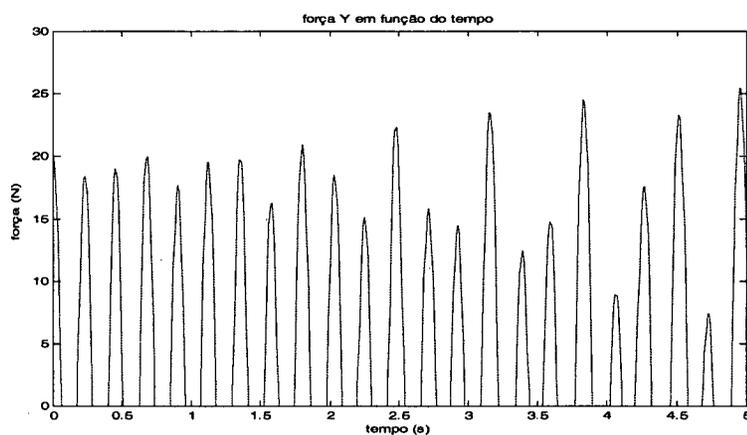


Figura 3.22: força de contato na direção Y em função do tempo para a movimentação do robô em contato com o meio e variação paramétrica

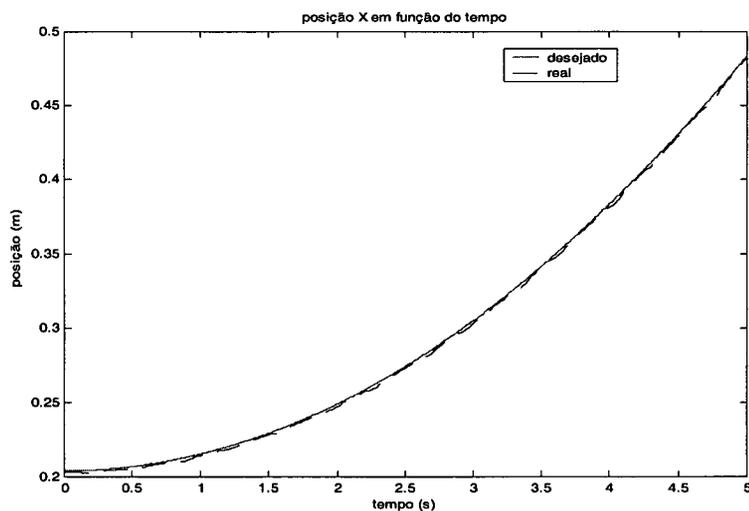


Figura 3.23: posição do efetuador final em relação a direção X em função do tempo na movimentação em contato com o meio

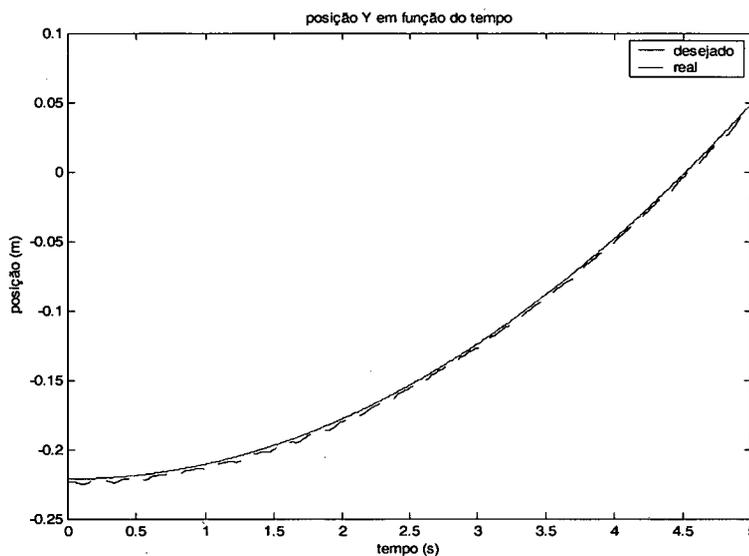


Figura 3.24: posição do efetuator final em relação a direção Y em função do tempo na movimentação em contato com o meio

Analisando os resultados obtidos na simulação com variação paramétrica, pode-se observar que o controle apresentou um comportamento regular, com o seguimento da trajetória imposta ao manipulador, porém com um erro não decrescente como ocorreu na simulação sem variação paramétrica.

Pode-se assim concluir que desvios na modelagem do robô, que para o caso simulado encontram-se na faixa de mais ou menos 10%, de incertezas nos parâmetros, podem apresentar uma influência no funcionamento do robô manipulador controlado. Os sinais de força, nas direções x e y, apresentaram amplitudes similares, porém com variação na taxa de amortecimento.

Porém, variações muito grandes de parâmetros poderiam levar o sistema para situações de instabilidade, principalmente em movimentos em contato com o meio, quando tem-se as duas malhas de controle, a interna de controle de posição e a externa de compensação de posição baseado na força, agindo em conjunto. A imprecisão da compensação dinâmica se traduz em uma perturbação na entrada.

3.4 IMPLEMENTAÇÃO DO CONTROLE DE IMPEDÂNCIA EM UM ROBÔ SCARA

Após feitas as simulações e obtido resultados comprovando o funcionamento do controle de impedância aplicado em um robô do tipo SCARA, passou-se ao desenvolvimento da programação em linguagem Xoberon, para a implementação do controle de impedância para o robô Inter do tipo SCARA (figura 3.25), já descrito anteriormente.

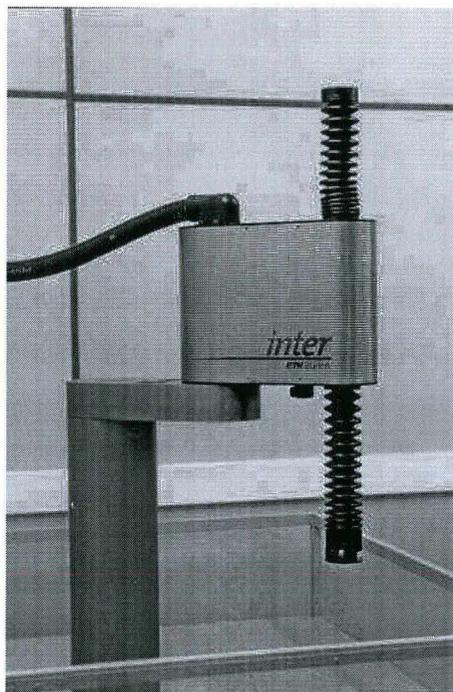


Figura 3.25: Foto do robô INTER

A principal característica do robô Inter é a sua arquitetura aberta, que permite mudanças em seus algoritmos de programação, a implementação de novos algoritmos de controle e a definição de novas trajetórias de referência (GOLIN et al, 1998). A linguagem de programação utilizada é a *Xoberon* (REISER, 1991), (MOSSENBOCK, 1993), (VESTLI, 1997).

A linguagem Xoberon foi desenvolvida no Instituto de Robótica da Universidade Técnica de Zurique – Suíça e é uma extensão da linguagem Oberon, que por sua vez, deriva de outras duas linguagens, o Pascal e Modula-2.

A programação do robô é dividida em blocos de programas, chamados de módulos. Ainda como característica da linguagem, tem-se a programação orientada a objetos, que permite uma grande versatilidade de programação, reutilização de módulos existentes, e ainda como grande vantagem, permite o instanciamento dos objetos on-line, ou seja, é possível alterar o comportamento de um controle, e assim o comportamento do robô, durante o seu funcionamento, sem a necessidade de parar o robô.

A linguagem Xoberon, por operar em tempo real, permite o controle sobre o tempo de execução dos módulos, permitindo assim verificar se as tarefas estão sendo cumpridas dentro de prazos pré determinados. Caso ocorra alguma falha, será exibida mensagem de erro, e o sistema irá ser suspenso automaticamente, evitando assim que possam ocorrer danos aos acionamentos, sensores e ao meio.

3.5 IMPLEMENTAÇÃO DO CONTROLE DE IMPEDÂNCIA PARA O SEGUIMENTO DE TRAJETÓRIA COM OBSTÁCULO

Nesta seção apresenta-se a implementação do controle de impedância para seguimento de uma trajetória contendo um obstáculo de perfil ovalado, conforme mostrado na figura 3.26:

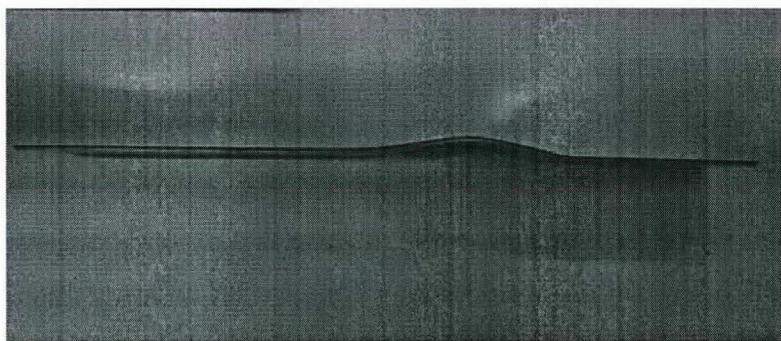


Figura 3.26: Foto da peça contendo obstáculo semi-elíptico

A escolha do formato da superfície a ser percorrida teve como objetivo demonstrar, em um único movimento, o comportamento do manipulador sob a ação do controle de impedância em duas situações distintas: Seguimento de uma reta como obstáculo fixo e variação da superfície de contato. Como a superfície possui uma semi

elipse, de formato não modelado, o controlador será testado em uma situação próxima da real, ou seja, deseja-se seguir uma superfície, porém esta possui relevo irregular de geometria não modelada.

Para ocorrer um melhor deslizamento do efetuator final por sobre a peça a ser seguida, utilizou-se um apalpador horizontal, desenvolvido e utilizado por (BIER, 2000).



Figura 3.27: Foto do apalpador horizontal

O desenvolvimento deste tipo de controle é útil para aplicações onde existe um seguimento de trajetória por sobre uma superfície, porém a superfície pode apresentar irregularidades, conhecidas ou não durante a fase de planejamento da trajetória, e estas alterações de relevo não devem impedir o desenvolvimento da trajetória, ou ainda que se danifique a superfície ou o manipulador. Como exemplos desta situação, podemos citar as tarefas de lixar uma superfície, polimento e solda.

Devido ao fato do apalpador utilizado possuir simetria axial, não foi aplicado controle sobre a quarta junta do robô. Tem-se então, o controle agindo sobre as três primeiras juntas do robô. Com o posicionamento das duas primeiras juntas, tem-se o controle de posição nas direções X e Y no espaço operacional, e a terceira junta é responsável pelo posicionamento da ferramenta na direção Z no espaço operacional. O controle para a terceira junta foi implementado, porém durante os experimentos, não foi utilizado plenamente, pois a altura em Z foi mantida constante durante o percurso desejado.

Para a implementação do controle de seguimento de contornos, não foram criados módulos novos, mas somente adaptados os módulos já existentes e em funcionamento no robô SCARA e que estão descritos no anexo C.

No módulo StateCtrl.mod, inicialmente foi alterado o procedimento algo, onde foram alterados os comandos da lei de controle, trocando-se o controle PD original, pelo

controle de impedância. Neste mesmo módulo, foram acrescentadas chamadas de rotinas acrescentadas no módulo ScaraCartKin.mod, onde foram implementados o cálculo da derivada do Jacobiano, chamada de Jad e a inversa do Jacobiano, chamada de InvJa. Estes cálculos foram acrescentados devido a necessidade de uso na estrutura do controle de impedância.

Optou-se por alterar o StateCtrl, ao invés de desenvolver um novo grupo de módulos, após ter-se verificado durante as simulações, o bom comportamento do controle de impedância na ausência de contato, ou seja, quando utilizado em controle de posição, sem restrições de movimentação. Portanto, o controle de impedância pode ser utilizado para o controle de posição sem trazer com isso problemas ao funcionamento do robô, como foi demonstrado no experimento de seguimento de uma trajetória sem contato com o ambiente, descrito no capítulo 3.

Outro módulo utilizado foi o GolPontosBier.mod. Este programa é responsável pela leitura dos valores desejados de posição, velocidade e aceleração, a partir de um arquivo gerado no Matlab. No caso, utilizou-se o mesmo arquivo criado quando da fase de simulações. Esta escolha teve como principal motivo evitar a ocorrência de pontos de singularidade, pois o controle utiliza a inversa do Jacobiano, e nas proximidades de um ponto de singularidade, o determinante do Jacobiano tende a zero. Outra função implementada no módulo GolPontosBier.mod é a amostragem dos valores de posição, velocidade, aceleração, força e controle em tempo real, sendo possível depois utilizar estes valores para a montagem de gráficos no Matlab.

Fez-se um levantamento dos nomes de variáveis já utilizados na representação da posição, velocidade, aceleração e força, para valores desejados e medidos, e em que base estavam sendo referenciadas, para assim evitar erros de transformação de base, e utilização de valores não compatíveis ou representados em bases diferentes das requeridas.

Após testado o funcionamento do controle implementado no Xoberon com a movimentação do robô sem obstáculos, fixou-se a ferramenta de teste na bancada do robô, utilizando para isto duas morsas.

Para posicionar o robô no ponto inicial da trajetória, utilizou-se o módulo Bahnplaner.mod, em seguida foi transferido o arquivo de pontos da trajetória para a memória do gabinete de controle do robô. Selecionou-se o tipo de saída desejada, ou seja, escolheu-se que informações seriam amostradas e gravadas na memória. Escolheu-se uma taxa de amostragem de 10%, equivalente a um período de 10ms.

3.6 RESULTADOS E DISCUSSÕES

Apresenta-se agora os resultados de experimento prático realizado com um robô de configuração SCARA, no qual foi implementado o algoritmo de controle de impedância descrito no capítulo 2. Realizou-se o experimento no robô SCARA, do Laboratório de Robótica da UFSC.

Após concluído a programação do controle de impedância, feito as verificações de funcionamento das rotinas alteradas e das interfaces de leitura e escrita de dados entre o computador e o painel de controle do robô SCARA, partiu-se para o ajuste dos ganhos do controlador. Como referência, utilizou-se os mesmos valores calculados para as simulações.

O tempo de experimento de cinco segundos, utilizado durante as simulações foi mantido, e a trajetória desejada, a ser seguida pelo manipulador, foi importada do programa gerador de trajetórias, utilizado durante as simulações descritas no capítulo 3, permitindo uma melhor comparação com os resultados obtidos nas simulações.

Com os ganhos ajustados, posicionou-se o robô no início da barra, de maneira a já existir uma pequena força de contato entre o manipulador e o meio.

Fez-se a inicialização dos objetos, que devem ser instanciados com o robô já ativado, e inicializou-se o controlador.

Sobre a ação do controle de impedância, o robô iniciou o movimento, seguindo a trajetória desejada, conforme mostra a figura 3.28.

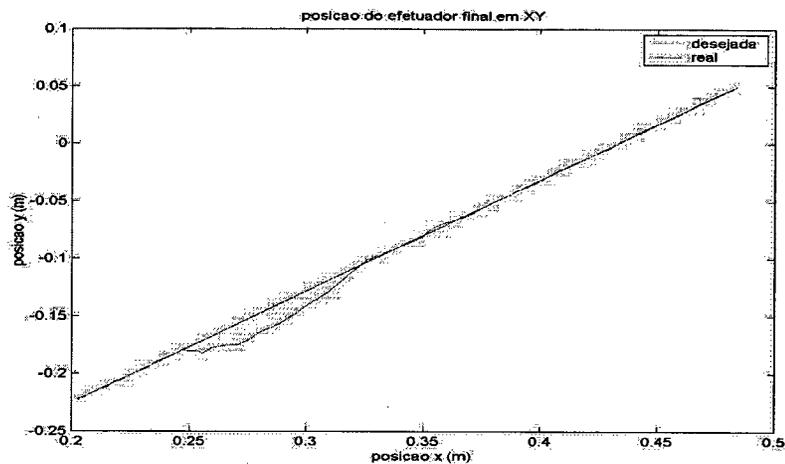


Figura 3.28: Gráfico da posição do efetuador final no plano XY

Pode-se ver ainda, segundo as figuras 3.29 e 3.30, que a força de contato nas direções X e Y apresentaram oscilações, sendo que uma parcela desta oscilação é devido aos ruídos oriundos do sensor de força e dinâmicas não modeladas do robô.

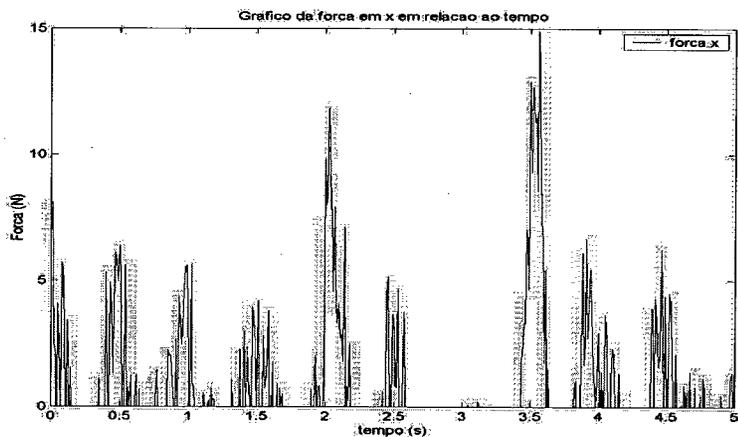


Figura 3.29: Gráfico da força na direção X em função do tempo

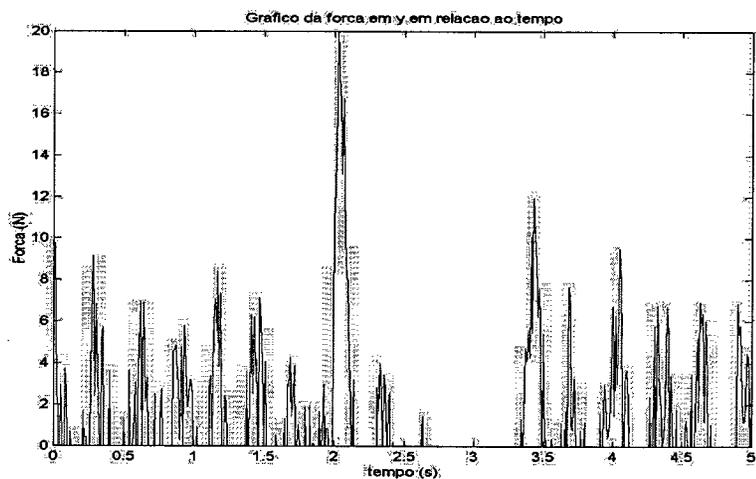


Figura 3.30: Gráfico da força na direção Y em função do tempo

No instante em que a ferramenta atinge o início da semi-elipse, a força de contato tem um aumento momentâneo, apresentando picos de valores. O esperado, do controle de impedância, seria que a força de contato fosse diminuindo durante o percurso, considerando a modelagem da dinâmica do erro eficiente. Porém como o percurso possui um obstáculo de formato semi-elíptico, o erro entre a posição desejada e a posição real terá uma variação crescente, durante o avanço pela primeira metade da semi-elipse, o que necessitaria de uma nova determinação dos ganhos do controle. Ao passar pelo ponto mais distante da trajetória desejada, ocorre uma pequena perda de contato com o meio, porém esta situação é controlada de maneira satisfatória, com o manipulador voltando a entrar em contato com a barra.

Ao retomar o contato com a barra, tem-se novamente um pico de força, porém de menor valor do que no início do obstáculo, e a diminuição gradativa do erro entre a posição desejada e a posição do meio, na segunda metade da semi-elipse, contribui para um decaimento dos valores da força de contato. Porém no trecho final da semi-elipse, ocorre novamente a perda de contato entre o manipulador e o obstáculo, como pode ser analisado nos gráficos das figuras 3.29 e 3.30, representado pela ausência de força próximo aos 3 segundos de experimento, em ambas as direções. A ferramenta percorreu uma trajetória aproximadamente paralela ao final da semi-elipse, chocando-se com a parte reta do obstáculo, aproximadamente no instante 3,3 segundos, representado por dois picos de força, nas direções X e Y. Nota-se que a força apresentada neste instante na direção X teve um valor de pico superior ao obtido no instante 2 segundos, quando do início da semi-elipse. Na direção Y, o fato inverteu-se, com o pico de força apresentando um valor inferior ao obtido no início da semi-elipse, fato explicado devido ao posicionamento da semi-elipse em relação a trajetória desejada, como pode ser observado na figura 3.4.

No início da semi-elipse, a diferença entre a posição desejada e a posição efetiva do efetuador final, imposta pelo obstáculo, apresenta uma variação brusca na direção Y e mais suave na direção X. Já no final da semi-elipse, a situação inverte-se, como pode ser observado na figura 3.26.

O gráfico do erro em X e Y, da mesma maneira que o gráfico da força de contato, reflete a boa atuação do controle de impedância durante o seguimento da trajetória desejada, limitada pela barra.

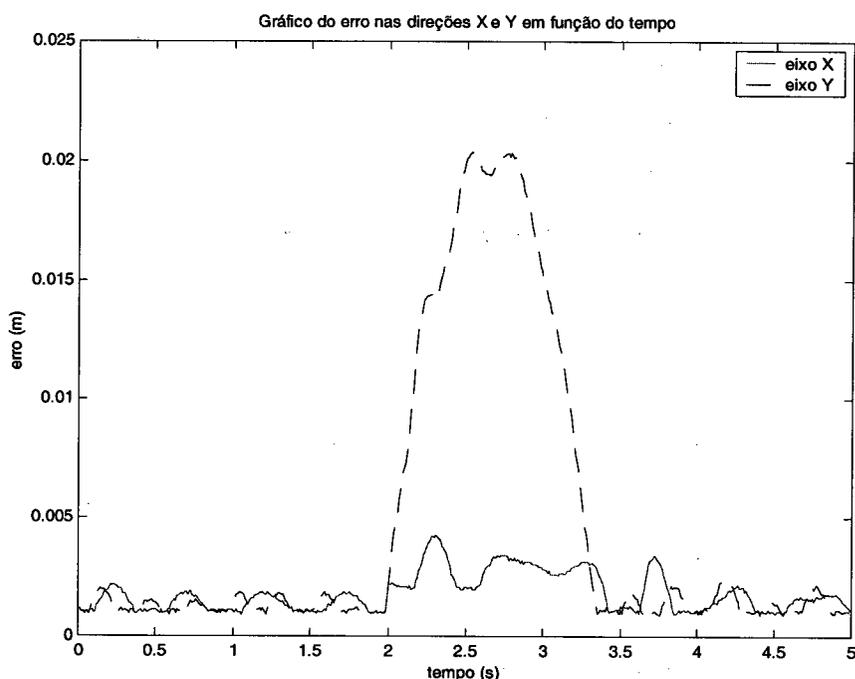


Figura 3.31: Gráfico do erro de posição nas direções X e Y durante o experimento

Para uma melhor comparação, repete-se o gráfico apresentado na seção 3.3.2, que traz o erro apresentado nas direções X e Y, quando simulado a movimentação em contato com o meio, sem variação paramétrica.

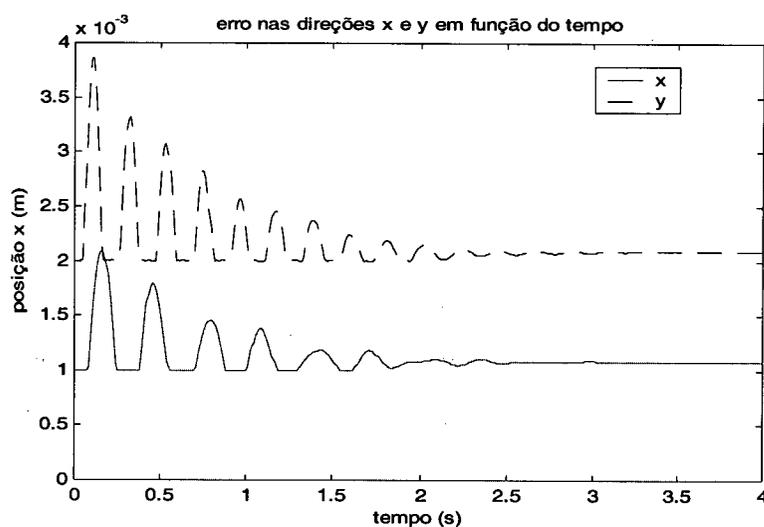


Figura 3.32: Gráfico dos erros de posição nas direções X e Y durante a simulação de movimentação em contato com o meio, apresentada na seção 3.3.

Estes resultados foram obtidos utilizando-se os mesmos ganhos calculados para as simulações. Nota-se porém, que o período de oscilação e o valor máximo atingido de força e de erro de posição, diferem dos obtidos durante as simulações. Fato devido a diferença entre a rigidez apresentada pelo meio e os valores utilizados na simulação, além de pequenas variações paramétricas.

Comparando-se o erro apresentado pelo manipulador durante o experimento (figura.3.31), com o erro de posição da simulação de movimentação em contato com o meio (figura.3.32), no intervalo de 0 a 2 segundos, pode-se perceber que a modelagem do sistema utilizado na fase de simulação encontra-se com valores próximos dos reais, pois há pouca diferença entre os valores máximos de erro e o tempo de amortecimento. No intervalo entre 2 a aproximadamente 3,3 segundos, tem-se uma situação não simulada anteriormente, onde a posição do meio varia de forma irregular. Novamente, no intervalo entre 3,3 e 5 segundos, o manipulador volta a apresentar um erro de posição próximo aos obtidos na fase de simulação.

3.6.1 SIMULAÇÃO DE MOVIMENTAÇÃO EM CONTATO COM O MEIO E PRESENÇA DE OBSTÁCULO

Para uma melhor análise dos resultados obtidos no experimento, fez-se uma nova simulação, com a inclusão de variações paramétricas nas matrizes de compensação dinâmica, igual ao utilizado no capítulo 3.3.2 e utilizou-se para a simulação do meio, uma equação de reta deslocada de um milímetro nas direções X e Y, em relação a trajetória desejada, acrescida de uma variação de superfície de formato irregular, entre os instantes 2 e 3 segundos, buscando assim, uma maior proximidade com a ferramenta utilizada no experimento prático.

Pode-se notar no gráfico da figura 3.33 que o erro apresentado nas direções X e Y apresentaram uma grande oscilação, porém em ambas as direções, ocorre um decaimento nos valores de pico.

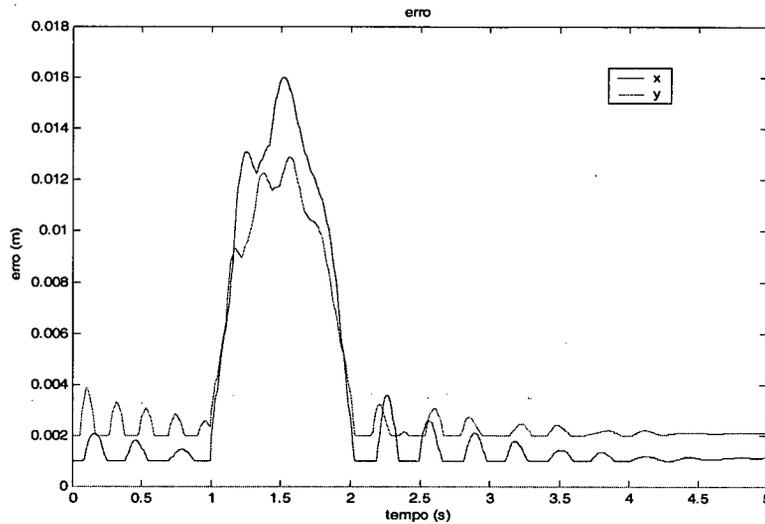


Figura 3.33: Gráfico dos erros de posição nas direções X e Y em função do tempo para a simulação de movimento em contato com o meio irregular e variação paramétrica

Verifica-se ainda, a partir da comparação entre os gráficos das figuras 3.29 e 3.34, que a força na direção X, nas duas situações apresentou um pico no ponto da trajetória relativo ao início da variação do meio e no final do obstáculo, quando o robô perde o contato por alguns instantes. O mesmo ocorre na direção Y, podendo ser observado nos gráficos das figuras 3.30 e 3.35.

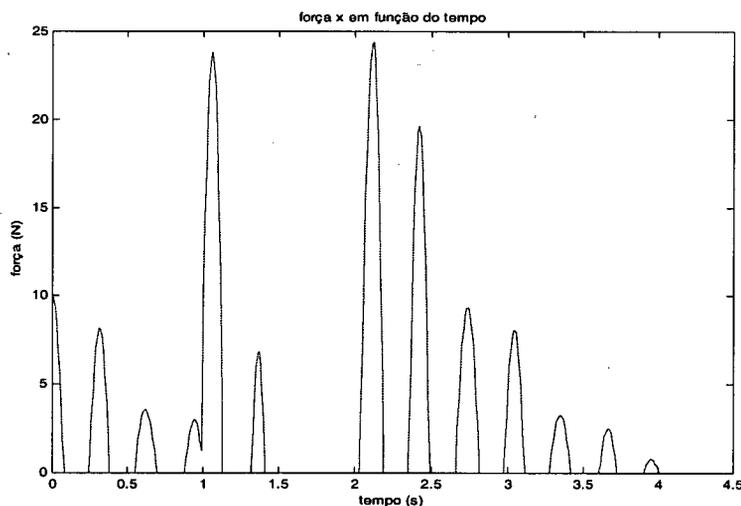


Figura 3.34: Gráfico da força na direção X em função do tempo para movimentação em contato com o meio irregular e variação paramétrica

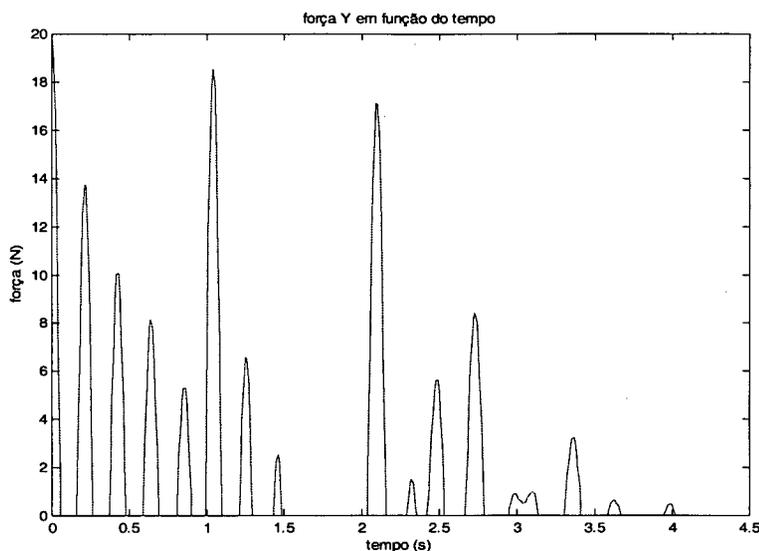


Figura 3.35: Gráfico da força na direção Y em função do tempo para movimentação em contato com o meio irregular e variação paramétrica

A variação entre os valores de pico alcançados durante o experimento e a simulação é devida em parte a variação da rigidez do meio, que para a simulação foi imposta como sendo 10^4N.m e durante o experimento prático possui valor desconhecido. Outro ponto importante a ressaltar é que a ferramenta foi afixada à base através da utilização de duas morsas, o que não permitia um ajuste perfeito. Tem-se ainda a influência das variações paramétricas do robô manipulador, tanto pela imposição no sistema de simulações, como as variações reais devidas ao robô.

3.7 OBSERVAÇÕES

Sabendo-se que a força resultante do contato entre o robô e o meio é diretamente proporcional ao erro entre a posição desejada, localizada dentro do meio, e a superfície limitadora do meio, é importante determinar o ponto inicial de maneira a não exercer uma força muito grande de início, pois durante o posicionamento do robô para o início do experimento, o controle de impedância não se encontra ativo.

Pode-se constatar, a partir dos resultados obtidos, que o controle de impedância, a partir da possibilidade de se modelar a dinâmica do erro causado pelo contato do robô manipulador com o meio, permite um controle de alta versatilidade, podendo ser

aplicado em situações onde ocorrem ou não contato entre o manipulador e o meio, e o seguimento de trajetórias em contato com o meio ou não, em locais onde existam obstáculos não definidos.

Também pode-se ressaltar o bom funcionamento do algoritmo de controle implementado no robô de configuração SCARA. Validando o modelo utilizado para as simulações.

4 CONCLUSÕES E PERSPECTIVAS

Neste trabalho apresentou-se um estudo sobre leis de controle de força aplicadas a robôs manipuladores, enfatizando-se os modelos que utilizam uma realimentação de força para a compensação de posição ou velocidade em função da força de contato entre o robô manipulador e o meio.

A técnica de controle de impedância implementada permite que o efetuador final do robô siga uma trajetória pré determinada e na ocorrência de contato entre o efetuador final e o meio, a dinâmica do erro em função da força de contato pode ser estabelecida através da escolha dos ganhos do controlador.

O robô SCARA, usado para as implementações práticas, é dotado de um sistema de programação, desenvolvido a partir da linguagem Xoberon.

O algoritmo implementado foi escrito com estrutura de programação similar a outros algoritmos já existentes, facilitando o seu entendimento. Da mesma maneira, a chamada para as rotinas desenvolvidas foram acrescentadas a módulos de programas já existentes, permitindo assim a utilização tanto para o controle de impedância, como para outras estruturas de controle já desenvolvidas ou que venham a ser desenvolvidas.

O controlador foi implementado para o robô com dois graus de liberdade, atuando nas direções X e Y do robô, sendo que durante o experimento a terceira e a quarta juntas do robô permaneceram travadas, através do comando de torque nulo.

Analisando o desempenho do controlador de impedância para o seguimento de trajetória em contato com o meio de forma irregular, pode-se afirmar que os resultados da implementação do algoritmo no robô demonstraram o bom desempenho do controle utilizado.

Para iniciar a trajetória, posicionou-se o robô, antes do experimento, no ponto inicial da trajetória a ser seguida. Experimentos com o robô iniciando em um ponto distinto ao ponto inicial da trajetória desejada, causaram altos valores de impacto entre o efetuador final e a superfície de contato pois, ao iniciar o movimento o erro de posição

apresentado era grande e com a inexistência de força de contato, o controle procurava corrigir a trajetória em poucos ciclos de processamento. A partir da escolha dos ganhos do controlador foi possível atenuar este efeito, porém causando um aumento no erro de posição e ainda uma alteração na dinâmica do sistema quando em contato com o meio.

Este controlador, da forma como foi desenvolvido, não é adequado para a execução de tarefas próximas às regiões de singularidade. Nestas regiões, o determinante do Jacobiano assume valores relativamente pequenos, causando grandes velocidades de juntas, podendo levar o sistema à instabilidade. Uma solução para este problema é o desenvolvimento de um módulo de supervisão de proximidade de pontos de singularidade, com a função de exercer um controle paralelo sobre o robô, de tal maneira a compensar os efeitos da proximidade de um ponto de singularidade, ou ainda fazer a correção da trajetória para o desvio de tal região. A solução adotada para evitar a proximidade de pontos de singularidade foi a determinação prévia da trajetória que seria seguida, a partir da escolha de posição de fixação da barra utilizada como obstáculo ao movimento do robô, e a verificação da não existência de pontos de singularidade.

Como perspectivas para trabalhos futuros têm-se o desenvolvimento do controle de impedância robusto e o controle de impedância adaptativo, baseados na estrutura desenvolvida, acrescidos de malhas de controle de ganho e compensação de variações paramétricas.

APÊNDICE A

A.1 MODELAGEM CINEMÁTICA E DINÂMICA DO ROBÔ SCARA

A modelagem de um sistema mecânico estabelece as relações de cinemática e dinâmica através de expressões matemáticas, levando em consideração parâmetros construtivos como massa, comprimento, inércia, graus de liberdade, limites construtivos e outros. Estas relações são úteis para descrever o comportamento do sistema e para o projeto de técnicas de controle. Neste apêndice é apresentado o modelo cinemático e dinâmico do robô manipulador SCARA, que é o objeto de estudo desta dissertação.

A.2 PARÂMETROS DO ROBÔ SCARA

A tabela A.1 traz as características construtivas do robô manipulador SCARA (Selective Compliant Articulated Robot for Assembly), de quatro juntas instalado no Laboratório de Robótica, adquirido através de uma parceria dos departamentos de Engenharia Mecânica (EMC) e de Automação e Sistemas (DAS) da Universidade Federal de Santa Catarina. Este robô foi construído sob encomenda no Instituto de Robótica da Universidade Técnica de Zurique (ETH), na Suíça.

Estes valores foram utilizados na modelagem, simulação e projeto de técnicas de controle aplicadas ao manipulador.

elo	$l_i (m)$	$l_{ci} (m)$	$m_i (kg)$	$I_i (kgm^2)$
0	0,25	0,118	11,4	0,23
1	0,25	0,116	19,5	0,16
2	0,00	0,000	2,0	0,10
3	0,00	0,000	1,5	0,10

Tabela A.1: Valores paramétricos do robô Inter

$l_i (m)$ é o comprimento do elo respectivo a junta I

$l_{ci} (m)$ é a distância da junta $I-1$ ao centro de massa do elo I

$m_i (kg)$ é a massa do elo I

$I_i (kgm^2)$ é o momento de inércia relativo ao centro de massa de cada elo.

A maioria destes valores foram obtidos de catálogos. Alguns, no entanto, foram determinados a partir de experimentos ou foram estimados. Os valores que possuem maior incerteza são os valores de massa e de inércia.

A figura A.1 apresenta um desenho esquemático do robô SCARA, bem como a alocação do sistema de coordenadas adotado para a realização da modelagem cinemática e dinâmica.

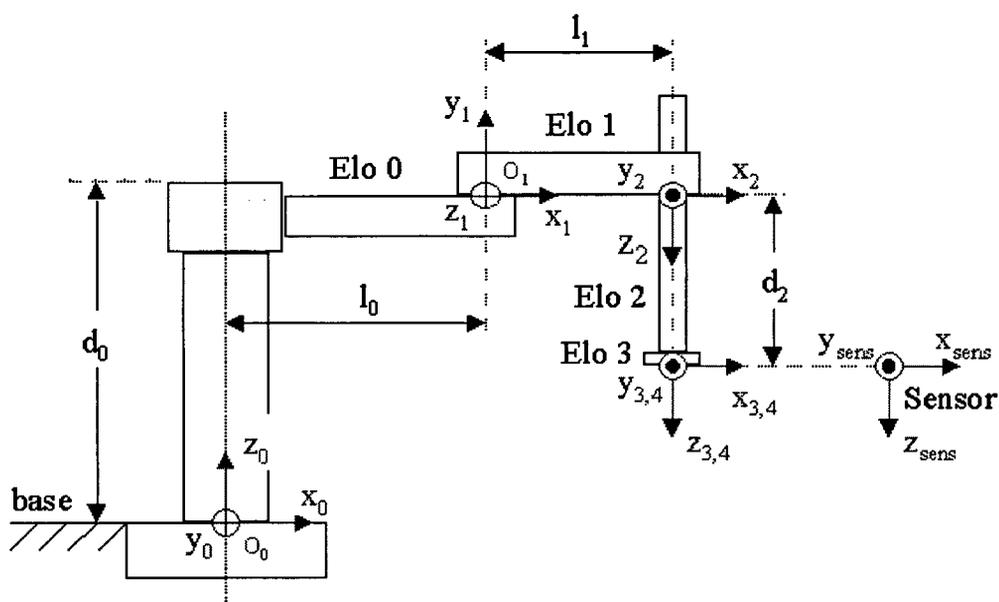


Figura A.1: Representação esquemática de um robô do tipo SCARA

A 3. MODELAGEM CINEMÁTICA

A análise cinemática de robôs preocupa-se em relacionar a posição e a orientação do efetuador final do manipulador em um sistema de coordenadas fixo, normalmente o da base, descrevendo sua movimentação no tempo, sem levar em consideração as forças que produzem este movimento. (ASADA e SLOTINE, 1986), (SCIAVICO e SICILIANO, 1996).

A 3.1. CINEMÁTICA DIRETA

A cinemática direta relaciona as variáveis de posição, velocidade e aceleração nas juntas do robô com as variáveis correspondentes de posição, velocidade e aceleração do efetuador final.

Para o robô SCARA de quatro graus de liberdade, e com referência ao sistema de coordenadas apresentado na figura A.1, podemos escrever os parâmetros de *Denavit-Hartenberg*, como apresentado na tabela A.2.

	α	$a(m)$	$d(m)$	$\theta(rad)$
0	0	l_0	d_0	θ_0
1	π	l_1	0	θ_1
2	0	0	d_3	0
3	0	0	0	θ_3

Tabela A.2: Parâmetros de *Denavit-Hartenberg* para um robô do tipo SCARA

Utilizando os parâmetros acima, a matriz de transformação para o robô manipulador SCARA resultará em (equação A.1):

$$A_4^0 = \begin{bmatrix} c_0c_3 & s_0c_3 - c_0s_3 & 0 & l_0c_0 + l_1c_0 \\ s_0c_3 - c_0s_3 & -s_0s_3 - c_0c_3 & 0 & l_0s_0 + l_1s_0 \\ 0 & 0 & -1 & d_0 - d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (A.1)$$

sendo $c_0 = \cos(\theta_0 + \theta_1)$, $s_3 = \sin(\theta_3)$ etc. A matriz A_4^0 representa a transformação do sistema de coordenadas O_4 para O_0 , ou seja, permite obter a representação da posição e a orientação do efetuador final em relação ao sistema de coordenadas cartesianas fixo a base do robô.

Podemos ainda definir para o robô SCARA, as seguintes relações de cinemática direta:

Posição

$$\begin{cases} x_0 = l_0 c_0 + l_1 c_1 \\ y_0 = l_0 s_0 + l_1 s_1 \\ z_0 = d_0 - d_2 \\ \phi_0 = \theta_0 + \theta_1 - \theta_3 \end{cases} \quad (\text{A.2})$$

Velocidade

$$\dot{x} = J_A(q)\dot{q} \quad (\text{A.3})$$

Sendo $\dot{x} = [\dot{x}_0 \quad \dot{y}_0 \quad \dot{z}_0 \quad \dot{\phi}_0]^T$ o vetor de velocidades no efetuador final, $\dot{q} = [\dot{\theta}_0 \quad \dot{\theta}_1 \quad \dot{d}_2 \quad \dot{\theta}_3]^T$ o vetor das velocidades nas juntas e J o Jacobiano geométrico para o manipulador SCARA, dado por (A.4):

$$J_A(q) = \begin{bmatrix} -l_0 s_0 - l_1 s_1 & -l_1 s_1 & 0 & 0 \\ l_0 c_0 + l_1 c_1 & l_1 c_1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 1 & 1 & 0 & -1 \end{bmatrix} \quad (\text{A.4})$$

Aceleração

$$\ddot{x} = J_A(q, \dot{q})\ddot{q} + \dot{J}_A(q)\dot{q} \quad (\text{A.5})$$

sendo:

$\ddot{x} = [\ddot{x}_0 \quad \ddot{y}_0 \quad \ddot{z}_0 \quad \ddot{\phi}_0]^T$, o vetor das acelerações no efetuador final, $\ddot{q} = [\ddot{\theta}_0 \quad \ddot{\theta}_1 \quad \ddot{d}_2 \quad \ddot{\theta}_3]^T$, o vetor das acelerações nas juntas e $\dot{J}_A(q, \dot{q})$ é a derivada do Jacobiano para o manipulador, dado por (A.6):

$$\dot{J}_A(q, \dot{q}) = \begin{bmatrix} -l_0 c_0 \dot{\theta}_0 - l_1 c_1 (\dot{\theta}_0 + \dot{\theta}_1) & -l_1 c_1 (\dot{\theta}_0 + \dot{\theta}_1) & 0 & 0 \\ -l_0 s_0 \dot{\theta}_0 - l_1 s_1 (\dot{\theta}_0 + \dot{\theta}_1) & -l_1 s_1 (\dot{\theta}_0 + \dot{\theta}_1) & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (\text{A.6})$$

A 3.2. CINEMÁTICA INVERSA

A cinemática inversa determina as posições, velocidades e acelerações das juntas do robô, a partir das correspondentes posições, velocidades e acelerações do efetuador final. Inicialmente definiremos a distância D , entre a origem dos sistemas de coordenadas do efetuador final e da base, como:

$$D = \frac{x_0^2 + y_0^2 - l_0^2 - l_1^2}{2l_0l_1} \quad (\text{A.7})$$

Para o manipulador SCARA temos as seguintes relações:

Posição

$$\theta_0 = \tan^{-1}\left(\frac{y_0}{x_0}\right) - \tan^{-1}\left(\frac{l_1 \text{sen}\theta_1}{l_0 + l_1 \cos\theta_1}\right) \quad (\text{A.8})$$

$$\theta_1 = \tan^{-1}\left(\frac{\pm \sqrt{1 - D^2}}{D}\right) \quad (\text{A.9})$$

$$d_2 = d_0 - z_0 \quad (\text{A.10})$$

$$\theta_3 = \theta_0 + \theta_1 - \phi_0 \quad (\text{A.11})$$

com ϕ_0 definido em (A.2).

Velocidade

$$\dot{q} = J_A^{-1}(q)\dot{x} \quad (\text{A.12})$$

onde $J_A^{-1}(q)$ é a inversa do Jacobiano Geométrico para o manipulador SCARA, dado por (A.13):

$$J_A^{-1}(q) = \begin{bmatrix} \frac{c01}{l_0s1} & \frac{s01}{l_0s1} & 0 & 0 \\ \frac{-l_0c0 - l_1c01}{l_0l_1s1} & \frac{-l_0s0 - l_1s01}{l_0l_1s1} & 0 & 0 \\ 0 & 0 & -1 & 0 \\ -\frac{c0}{l_1s1} & -\frac{s0}{l_1s1} & 0 & -1 \end{bmatrix} \quad (\text{A.13})$$

Aceleração

$$\ddot{q} = J_A^{-1}(q)(\ddot{x} - \dot{J}_A(q)\dot{q}) \quad (\text{A.14})$$

onde \ddot{x} é o vetor das acelerações no efetuador final, \ddot{q} é o vetor das acelerações nas juntas.

A 4. MODELAGEM DINÂMICA

O modelo dinâmico de um robô relaciona as posições, velocidades e acelerações das juntas, ou do efetuador final, com os torques que devem ser aplicados pelos atuadores, de acordo com a tarefa a ser executada, considerando as características de massa, geometria e inércia do sistema. (SPONG e VIDYASAGAR, 1989), (SCIAVICCO e SICILIANO, 1996).

Considere a equação diferencial que representa o modelo dinâmico do robô no espaço de juntas apresentado na seção 2:

$$M(q)\ddot{q} + V(q, \dot{q})\dot{q} + F_a\dot{q} + G(q) = u - J^T(q)h = \tau \quad (\text{A.15})$$

onde $M(q)$ é a matriz de inércia do manipulador, representada por (A.16):

$$M(q) = \begin{bmatrix} h_0 & h_1 & 0 & -I_3 \\ h_4 & h_5 & 0 & -I_3 \\ 0 & 0 & h_{10} & 0 \\ -I_3 & -I_3 & 0 & I_3 \end{bmatrix} \quad (\text{A.16})$$

com os termos:

$$\begin{aligned}
h_0 &= m_0 l_{c_0}^2 + (m_1 + h_{10}) l_0^2 + m_1 l_{c_1}^2 + m_2 + m_3 + I_0 + I_1 + I_2 + I_3 + 2m_1 l_0 l_{c_1} c_1 + (m_2 + m_3) 2l_0 l_1 c_1 \\
h_1 &= h_4 = (m_1 l_{c_1} + h_{10} l_1) l_0 c_1 + m_1 l_{c_1}^2 + (m_2 + m_3) l_1^2 + I_1 + I_2 + I_3 \\
h_5 &= m_1 l_{c_1}^2 + (m_2 + m_3) l_1^2 + I_1 + I_2 + I_3 \\
h_{10} &= m_2 + m_3
\end{aligned}$$

$V(q, \dot{q})\dot{q}$ é o vetor das forças de Coriolis e centrífugas, dado por (A.17):

$$V(q, \dot{q})\dot{q} = \begin{bmatrix} -C_1 \text{sen}(\theta_1) (2\dot{\theta}_0 \dot{\theta}_1 + \dot{\theta}_1^2) \\ C_1 \text{sen}(\theta_1) \dot{\theta}_0^2 \\ 0 \\ 0 \end{bmatrix} \quad (\text{A.17})$$

F_a é a matriz diagonal de atritos do sistema representado por (A.18):

$$F_a = \begin{bmatrix} \mu_1 & 0 & 0 & 0 \\ 0 & \mu_2 & 0 & 0 \\ 0 & 0 & \mu_3 & 0 \\ 0 & 0 & 0 & \mu_4 \end{bmatrix} \quad (\text{A.18})$$

onde os termos μ_1, μ_2, μ_3 e μ_4 representam os coeficientes de atrito viscoso de cada junta.

$G(q)$ é o vetor dos termos gravitacionais do sistema, representado por (A.19):

$$G(q) = \begin{bmatrix} 0 \\ 0 \\ (m_2 + m_3)g \\ 0 \end{bmatrix} \quad (\text{A.19})$$

u é o vetor (4x1) dos torques de controle aplicados às juntas do robô.

APÊNDICE B – PROGRAMAS DESENVOLVIDOS PARA O MATLAB E SIMULINK

Apresentam-se neste apêndice os códigos fonte dos programas desenvolvidos para o Matlab e os diagramas desenvolvidos para o simulink.

B.1 DIAGRAMAS PARA O AMBIENTE SIMULINK

Inicialmente será apresentado o diagrama geral do sistema de controle de impedância, exibido na figura B.1, onde pode-se notar a divisão do controle em blocos, que serão discutidos neste anexo.

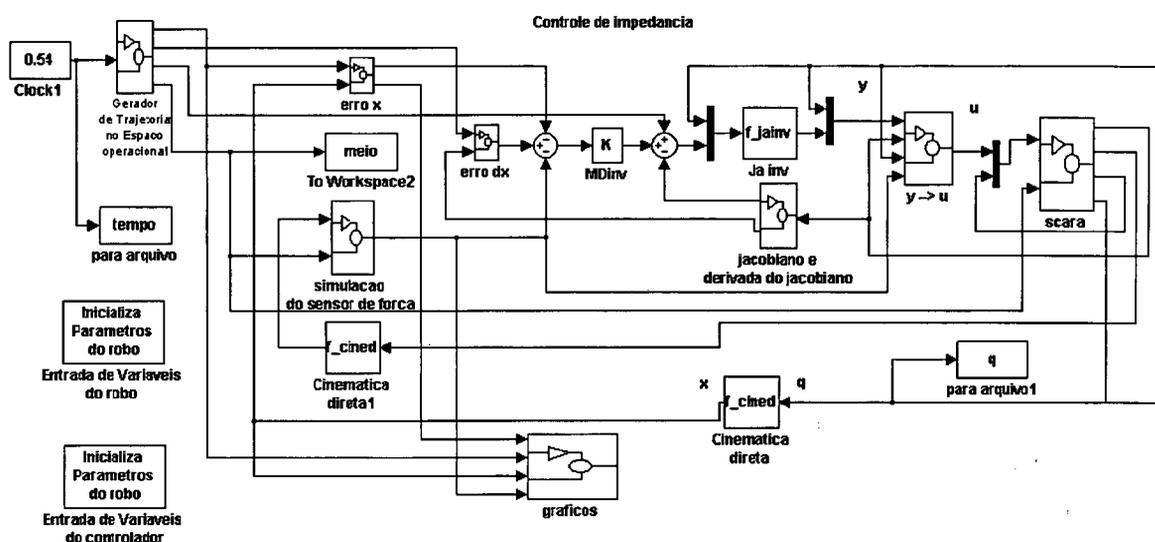


figura B.1: sistema de controle de impedância aplicado a um robô SCARA

Os principais blocos desenvolvidos e que serão discutidos a seguir são:

Módulo gerador de trajetória no espaço operacional

Módulo de cálculo do Jacobiano Analítico e sua derivada

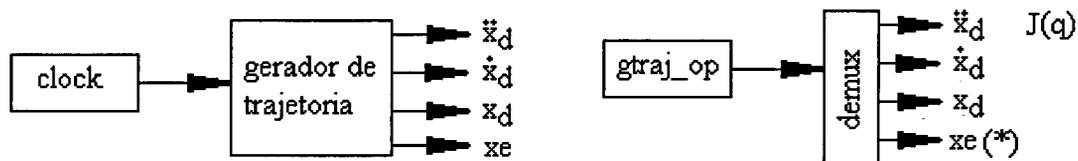
Módulo de simulação do sensor de força

Módulo de simulação do robô SCARA

Módulo de cálculo da linearização $y \rightarrow u$

Módulo de simulação do robô SCARA

A geração da trajetória desejada no espaço operacional, que servirá de referência ao controle, é feita pelo módulo `gtraj_op`, e que tem sua chamada dentro do bloco gerador de trajetória no espaço operacional, exibido na figura B.1 e detalhado na figura B.2:



(*) o sinal x_e é utilizado no bloco de simulação do sensor de força

figura B.2: diagrama esquemático do bloco “gerador de trajetória no espaço operacional”

O módulo de cálculo do Jacobiano e sua derivada contém, conforme mostrado na figura B.3, as chamadas para as funções `f_ja` e `f_jad`.

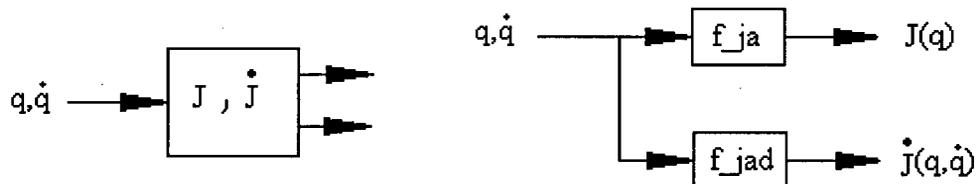


figura B.3: diagrama esquemático do bloco “cálculo do Jacobiano e sua derivada”

O terceiro módulo descrito é o de simulação do sensor de força (figura B.4), o qual apresenta uma função muito importante, pois quanto mais preciso for a sua simulação, melhor será a comparação com uma situação real. Este módulo recebe de entrada o valor teórico que seria atingido pelo manipulador, caso não ocorresse contato com o meio e o valor representativo da posição do meio. No caso em que não ocorre contato, a posição teórica a ser atingida é a própria posição do efetuador.

Nas situações de contato, a posição teórica que seria atingida pelo efetuador, caso não existisse contato, é calculada e enviada para o simulador do sensor de força e este valor comparado com o valor referencial do meio. A diferença entre os dois valores é multiplicada pela rigidez do meio, que é determinada através da definição dos parâmetros de simulação.

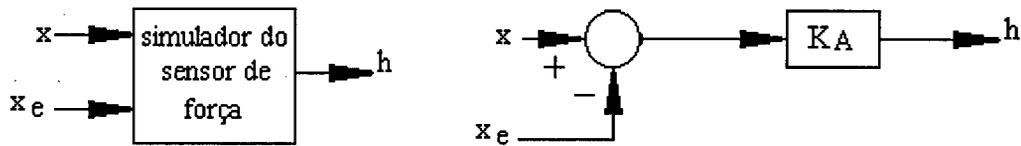


figura B.4: diagrama esquemático do bloco “simulação do sensor de força”

O módulo de cálculo da linearização $y \rightarrow u$, representado pela figura B.5, é responsável pela implementação de parte da estrutura da lei de controle, a qual permite fazer a substituição do sinal de controle u , pelo sinal de controle y , o qual efetua a compensação das não linearidades do manipulador.

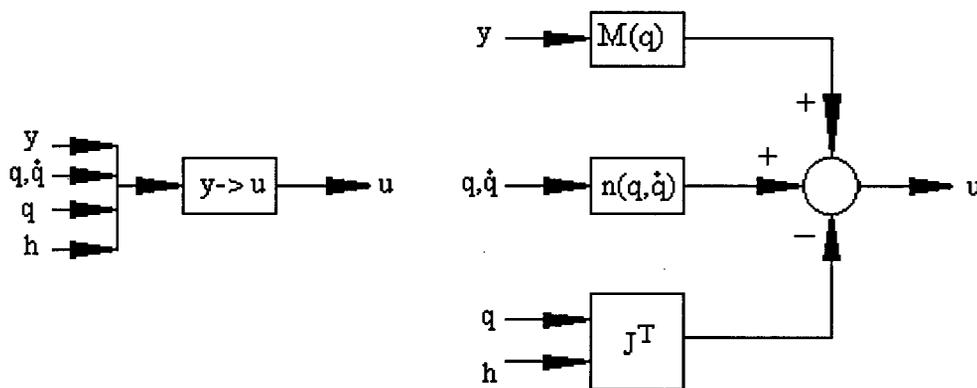


figura B.5: diagrama esquemático do bloco “cálculo da linearização $y \rightarrow u$ ”

O módulo de simulação do robô SCARA mostrado na figura B.6, engloba além da função de cálculo das equações diferenciais que representam o comportamento do robô, o cálculo da posição que seria atingida pelo efetuador, caso não ocorresse contato entre o manipulador e o meio. Para isto, é feita a aplicação da cinemática direta sobre o valor de posição das juntas, em seguida, a posição a ser atingida é comparada com a posição do meio, no espaço operacional, caso exista contato, o valor de deslocamento é alterado e tem seu valor limitado ao meio, e após feito a limitação, retorna-se para variáveis no espaço das juntas, utilizando para tal, a cinemática inversa do robô manipulador e a aplicação do Jacobiano, para permitir o cálculo da velocidade no espaço das juntas.

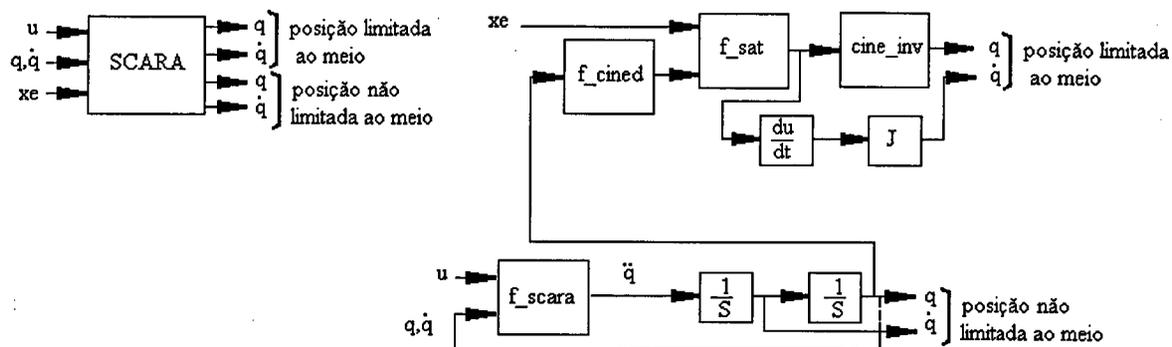


figura B.6: diagrama esquemático do bloco de "simulação do robô SCARA"

Os demais blocos representados na figura B.1 e que não foram especificados anteriormente, estão presentes nas bibliotecas de blocos e funções do Simulink e portanto não é necessário uma descrição formal.

B.2 FUNÇÕES DESENVOLVIDAS PARA O MATLAB

Serão agora brevemente comentadas e a listagem exibida, das principais funções desenvolvidas, necessárias para a implementação do controle de impedância, e que podem ser utilizadas para outras estruturas de controle.

A função **f_cined** calcula a cinemática direta do manipulador, tendo como objetivo retornar a posição do efetuador final no espaço cartesiano, nas coordenadas X, Y e Z, a partir dos valores de posição das juntas q_0 , q_1 e q_2 .

Listagem:

```
function [sys, x0] = f_cined(t,x,u,flag,a1,a2);
% programa que a partir de valores de posição desejados
% no espaço de juntas calcula a posição no espaço operacional
% (cinemática direta)
% entrada no formato (t,x,u[3],flag,a1,a2)
% u[teta1 teta2 teta3]

if abs(flag) == 3 % outputs

    % inicializa o vetor
    teta1 = u(1);
    teta2 = u(2);
    teta3 = u(3);

    % Cálculo das posições desejadas

    c1 = cos(teta1);
    c12 = cos(teta1+teta2);
    s1 = sin(teta1);
    s12 = sin(teta1+teta2);

    x = a1*c1 + a2*c12;
    y = a1*s1 + a2*s12;
    z = teta3;

    % Atribuição dos valores
    p(1) = x;
    p(2) = y;
    p(3) = z;

    sys = [p];

elseif flag == 0 % initial conditions

    sys = [0 0 3 3 0 1];
    x0 = [];

else

    sys = [];

end
```

A função `f_cineinv` calcula a cinemática inversa do manipulador, tendo como objetivo retornar a posição do efetuador final no espaço das juntas, nas coordenadas q_0 , q_1 e q_2 , a partir dos valores de posição do efetuador final X , Y e Z no espaço cartesiano.

Listagem:

```
function [sys, x0] = cineinv(t,x,u,flag,a1,a2);

% programa que a partir dos valores de posição no espaço
% cartesiano, calcula a posição no espaço das juntas
% para o robô Scara especificado para 3 juntas
%
% (cinemática inversa)
%

if abs(flag) == 3 % outputs (número de saidas)

    % inicializa as variáveis

    x = u(1);
    y = u(2);
    z = u(3);

    % Cálculo das posições desejadas

    d = (x^2+y^2-a1^2-a2^2)/(2*a1*a2);

    theta2 = atan2(sqrt(1-d^2),d);
    theta1 = atan2(y,x)-atan2(a2*sin(theta2), (a1+a2*cos(theta2)));
    theta3 = z;

    % Atribuição dos valores de saída

    q(1) = theta1;
    q(2) = theta2;
    q(3) = theta3;

    sys = [q];

elseif flag == 0 % initial conditions

    sys = [0 0 3 3 0 1];
    x0 = [];

else

    sys = [];

end
```

A função `f_ja` calcula e aplica o Jacobiano Analítico, tendo como objetivo retornar a velocidade do efetuador final no espaço cartesiano, a partir dos valores de velocidade no espaço das juntas.

Listagem:

```
function[sys, x0] = f_ja(t,x,u,flag,a1,a2);

% Este programa calcula o Jacobiano para transformação de velocidade
% do espaço de juntas para o sistema cartesiano
% Realizado por:
% Francisco Javier Triveno Vargas
% Revisado por Cesar Motejunas - 31/05/2000
%-----

if abs(flag) == 3 % outputs

    teta1 = u(1);
    teta2 = u(2);
    teta3 = u(3);
    dteta(1) = u(4);
    dteta(2) = u(5);
    dteta(3) = u(6);

    s1 = sin(teta1);
    s2 = sin(teta2);
    c1 = cos(teta1);
    c2 = cos(teta2);
    s12 = sin(teta1+teta2);
    c12 = cos(teta1+teta2);

    % Calculo do Jacobiano do Scara

    J = [-a1*s1-a2*s12  -a2*s12  0
          a1*c1+a2*c12  a2*c12  0
           0             0      1];

    dx = J*dteta';

    sys = [dx'];

elseif flag == 0 % initial conditions

    sys = [0 0 3 6 0 1];
    x0 = [];

else

    sys = [];
end

return
```

A função `f_jad` calcula e aplica a derivada do Jacobiano Analítico, tendo como objetivo retornar o produto $J_A(q, \dot{q})\dot{q}$, utilizado no controle de impedância.

Listagem:

```
function[sys, x0] = f_jad(t,x,u,flag,a1,a2);

% Este programa calcula a derivada do Jacobiano Analítico
%-----

if abs(flag) == 3 % outputs

    teta1 = u(1);
    teta2 = u(2);
    teta3 = u(3);
    dteta(1) = u(4);
    dteta(2) = u(5);
    dteta(3) = u(6);
    dq1 = dteta(1);
    dq12 = dteta(1)+dteta(2);
    s1 = sin(teta1);
    s2 = sin(teta2);
    c1 = cos(teta1);
    c2 = cos(teta2);
    s12 = sin(teta1+teta2);
    c12 = cos(teta1+teta2);

    % Calculo da Derivada do Jacobiano Analítico

    Jad = [-a1*c1*dq1-a2*c12*dq12 -a2*c12*dq12      0
           -a1*s1*dq1-a2*s12*dq12 -a2*s12*dq12      0
           0                        0                0];

    dx = Jad*dteta';

    sys = [dx'];

elseif flag == 0 % initial conditions

    sys = [0 0 3 6 0 1];
    x0 = [];

else

    sys = [];

end

return
```

A função `f_jainv` calcula e aplica a inversa do Jacobiano Analítico, tendo como objetivo retornar o produto $J_A^{-1}(q, \dot{q})\dot{x}$, utilizado no controle de impedância.

Ao invés de utilizar o comando de inversão do matlab, optou-se por montar a matriz inversa do jacobiano, a partir do princípio:

$J_A \cdot J_A^{-1} = I$, tem-se:

$$\begin{bmatrix} A & B & 0 \\ C & D & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} E & F & G \\ H & I & J \\ K & L & N \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Obtém-se para a matriz inversa as seguintes relações:

$$E = 1/(-B*C/D+A);$$

$$H = -C*E/D;$$

$$I = 1/(-B*C/A+D);$$

$$F = -B*I/A;$$

$$G = 0;$$

$$J = 0;$$

$$K = 0;$$

$$L = 0;$$

$$M = 1;$$

Listagem:

```
function[sys, x0] = f_jainv(t,x,u,flag,a1,a2);
```

```
% Este programa calcula a inversa do Jacobiano Analítico
```

```
if abs(flag) == 3 % outputs
```

```
    teta1 = u(1);
```

```
    teta2 = u(2);
```

```
    teta3 = u(3);
```

```
    dx(1) = u(4);
```

```
    dx(2) = u(5);
```

```
    dx(3) = u(6);
```

```
    s1 = sin(teta1);
```

```
    s2 = sin(teta2);
```

```
    c1 = cos(teta1);
```

```
    c2 = cos(teta2);
```

```
    s12 = sin(teta1+teta2);
```

```
c12 = cos(teta1+teta2);

%atribuição de variáveis auxiliares, para montagem da inversa do
%Jacobiano

A = -a1*s1-a2*s12;
B = -a2*s12;
C = a1*c1+a2*c12;
D = a2*c12;
E = 1/(-B*C/D+A);
H = -C*E/D;
I = 1/(-B*C/A+D);
F = -B*I/A;
G = 0;
J = 0;
K = 0;
L = 0;
M = 1;

Jinv = [E F G;H I J;K L M];

dteta = Jinv*dx';

sys = [dteta'];

elseif flag == 0 % initial conditions

    sys = [0 0 3 6 0 1];
    x0 = [];

else

    sys = [];

end

return
```

A função `f_m` tem por objetivo calcular a matriz de inércias, $M(q)$ do manipulador e retornar o produto $M(q).y$, para uso na simulação do robô e na realimentação de linearização da lei de controle utilizada.

```
function[sys, x0] = f_m(t,x,u,flag,g,li,mi,HB,Lc,mc,Ic);
% Este programa calcula a matriz M(q) e determina M(q).y
if abs(flag) == 3 % outputs
    q = u(1:3);
    y = u(4:6);

    m1 = mi(1);
    m2 = mi(2);
    m3 = mi(3);
    m4 = mi(4);
    ma = m3+m4+mc;
    I1 = HB(1);
    I2 = HB(2);
    I3 = HB(3);
    I4 = HB(4);
    l1 = li(1);
    l2 = li(2);
    lc1 = Lc(1);
    lc2 = Lc(2);
    aux1=I1+I2+I3+I4+Ic+(m2+ma)*l1^2+(m2*lc2^2)+ma*l2^2+(m1*lc1^2);
    aux2=(m2*l1*lc2)+ma*(l2*l1);
    aux3=I2+I3+I4+Ic+(m2*lc2^2)+ma*l2^2;

    M(1,1) = aux1+(2*aux2)*cos(q(2));
    M(1,2) = aux3+aux2*cos(q(2));
    M(1,3) = 0.0;
    M(2,1) = M(1,2);
    M(2,2) = aux3;
    M(2,3) = 0.0;
    M(3,1) = 0.0;
    M(3,2) = 0.0;
    M(3,3) = (m3+m4+mc);

    u = M*y;
    sys = [u];

elseif flag == 0 % initial conditions
    sys = [0 0 3 6 0 1];
    x0 = [];

else
    sys = [];

end
return
```

A função f_n tem por objetivo o cálculo da matriz $n(q, \dot{q})$, das forças de Coriolis e centrífugas, onde $n(q, \dot{q}) = V(q, \dot{q})\dot{q} + F\dot{q} + G(q)$, e é utilizado na formação do sinal de controle $u = M(q)y + n(q, \dot{q})$.

```
function[sys, x0] = f_n(t,x,u,flag,g,li,mi,HB,Lc,mc,Ic);
% Este programa calcula as forcas de coriolis e centrifuga
if abs(flag) == 3 % outputs
    q = u(1:3);
    dq = u(4:6);
    m2 = mi(2);
    m3 = mi(3);
    m4 = mi(4);
    ma = m3+m4+mc;
    l1 = li(1);
    l2 = li(2);
    lc1 = Lc(1);
    lc2 = Lc(2);
    aux2=(m2*l1*lc2)+ma*(l2*l1);

    %A matriz de forca de Coriolis e Centrífuga

    centrif1=(aux2 * dq(1) * dq(1)) *sin(q(2));
    centrif2=(2 * aux2 * dq(1) * dq(2)) * sin(q(2));
    coriol12=(aux2 * dq(2) * dq(2)) * sin(q(2));
    centrif3 = 0.0;

    C(1,1) = centrif2+coriol12;
    C(2,1) = centrif1;
    C(3,1) = centrif3;

    sys = [C];

elseif flag == 0

    sys = [0 0 3 6 0 1];
    x0 = [];

else

    sys = [];

end

return
```

A função `f_scara` é responsável pela simulação do comportamento do robô SCARA, a partir da aplicação de um sinal de controle `u`, através das equações diferenciais que modelam o robô.

Listagem:

```
function[sys, x0] = f_scara(t,x,u,flag,g,li,mi,HB,Lc,mc,Ic);
global q_inicial;
```

```
% Este programa simula um manipulador scara
```

```
if abs(flag) == 3 % outputs
```

```
    controle = u(1:3);
        q = u(4:6);
        dq = u(7:9);
```

```
    [M,C,G] = f_mtrMCG(q,dq,g,li,mi,HB,Lc,mc,Ic);
```

```
    qpp = inv(M) * ( -( C + G ) + controle );
```

```
    x1 = q;
    x2 = dq;
    xd1 = x2;
    xd2 = qpp;
    xd = [xd1; xd2];
    sys = [qpp];
```

```
elseif flag == 0 % initial conditions
```

```
    sys = [0 0 3 9 0 1];
    x0 = [];
```

```
else
```

```
    sys = [];
```

```
end
```

```
return
```

A função **f_jacobt** calcula e aplica a transposta do Jacobiano Geométrico do robô SCARA, tendo como objetivo retornar o produto $J^T(q)h$, utilizado no controle de impedância.

Listagem:

```
function[sys, x0] = f_jacobt(t,x,u,flag,a1,a2);

% Este programa calcula o Jacobiano geométrico transposto
% para geração do termo de compensação J'*h

if abs(flag) == 3 % outputs

    teta1 = u(1);
    teta2 = u(2);
    teta3 = u(3);
    h(1) = u(4);
    h(2) = u(5);
    h(3) = u(6);
    s1 = sin(teta1);
    s2 = sin(teta2);
    c1 = cos(teta1);
    c2 = cos(teta2);
    s12 = sin(teta1+teta2);
    c12 = cos(teta1+teta2);

    % Calculo do Jacobiano geométrico do Scara

    J = [-a1*s1-a2*s12  -a2*s12  0
          a1*c1+a2*c12  a2*c12  0
           0             0    -1];

    dx = J'*h';

    sys = [dx'];

elseif flag == 0

    sys = [0 0 3 6 0 1];
    x0 = [];

else

    sys = [];

end

return
```

A função **gtraj_op** gera os valores de posição, velocidade e aceleração desejados, a partir do valor de entrada de tempo, nas direções X, Y e Z no espaço operacional.

Listagem:

```
function [sys, x0] = gtraj_op(t,x,u,flag, xi, xf, yi, yf, zi, zf,tini,
ta, tfin);
```

```
% Gera curvas de posição, velocidade e aceleração desejadas
% para o ESPACO OPERACIONAL. Aceleração constante, velocidade linear
% e posição parabólica
```

```
if abs(flag) == 3 % outputs
```

```
    tt = u;
```

```
    tax=ta(1);      %Definição dos tempos de acelerações
```

```
    tay=ta(2);
```

```
    taz=ta(3);
```

```
    tf=tfin;
```

```
    % cálculo da trajetória no espaco operacional
```

```
    ac_x = 2*(xf-xi)/(tax^2);
```

```
    ac_y = 2*(yf-yi)/(tay^2);
```

```
    ac_z = 2*(zf-zi)/(taz^2);
```

```
    pos_x = xi + (ac_x*tt^2/2);
```

```
    vel_x = ac_x*tt;
```

```
    acc_x = ac_x;
```

```
    pos_y = yi + (ac_y*tt^2/2);
```

```
    vel_y = ac_y*tt;
```

```
    acc_y = ac_y;
```

```
    pos_z = zi + (ac_z*tt^2/2);
```

```
    vel_z = ac_z*tt;
```

```
    acc_z = ac_z;
```

```
% cálculo da posição do meio em relação a posição desejada do robô.
```

```
    e_x = pos_x-0.001; % posição desejada está 1mm para dentro do meio
```

```
    e_y = pos_y-0.002; % posição desejada está 2mm para dentro do meio
```

```
    e_z = pos_z+0.001; % posição desejada está 1mm para fora do meio
```

```
    p(1)=pos_x; p(2)=pos_y; p(3)=pos_z;
```

```
    p(4)=vel_x; p(5)=vel_y; p(6)=vel_z;
```

```
    p(7)=acc_x; p(8)=acc_y; p(9)=acc_z;
```

```
    p(10)=e_x; p(11)=e_y; p(12)=e_z;
```

```
    sys = [p];
```

```
elseif flag == 0
```

```
    sys = [0 0 12 1 0 1];
```

```
    x0 = [];
```

```
else
```

```
    sys = [];
```

```
end
```

A rotina **init_r** é chamada diretamente do ambiente de simulação desenvolvido no Simulink, e tem a função de carregar as variáveis globais, contendo os dados relativos ao robô SCARA que serão utilizados pelas demais funções. Desta maneira, caso ocorra uma alteração de um parâmetro do robô, uma vez efetuada dentro da rotina **init_r**, passará a estar disponível para todas as demais funções, eliminando assim o risco de uma função não estar com valores atualizados.

Listagem:

```
% init_r.m
% Inicialização de variáveis globais do robô SCARA

clear global all;
clear all;

global g li mi HB Lc mc lc Ic a1 a2

% Dados reais do robô SCARA
g      = 9.81;
li = [0.25; 0.25; 0.0];
mad = 1.0;
mi = [11.4; 19.5; 2.0; (1.5+mad)];
HB = [0.23; 0.16; 0.0; +0.1];
Lc  = [0.118; 0.116; 0.0];
mc = 0;
Ic = 0;
a1 = li(1);
a2 = li(2);
```

Para a inclusão e alteração das variáveis de controle, criou-se uma estrutura de menus e submenus que possibilitam que os valores de ganho, posição inicial e final desejada e tempo de movimentação sejam alterados diretamente do ambiente de simulação, sem a necessidade de salvar arquivos ou edição no ambiente de trabalho do Matlab.

O menu principal é chamado a partir do programa `init_ctr` e é exibido na figura B.7.

Listagem:

```
% init_ctr      Menu que permite visualizar e alterar os parâmetros do
% controlador em uso
while 1,
    principal = menu('Controlador por Impedância', ...
        ' constantes ', ...
        'posição inicial', ...
        ' posição final ', ...
        ' Kp ', ...
        ' Kd ', ...
        ' Ka ', ...
        ' Md ', ...
        ' Exit ');
    switch principal,
    case 1
        valctd %submenu para inserção de constantes
    case 2,
        posini %submenu para inserção da posição inicial
    case 3,
        posfim %submenu para inserção da posição final
    case 4,
        valkp %submenu para inserção dos ganhos kp
    case 5,
        valkd %submenu para inserção dos ganhos kd
    case 6,
        valka %submenu para inserção dos ganhos ka
    case 7,
        valmd %submenu para inserção dos ganhos md
    case 8,
        break;
    end
end
```

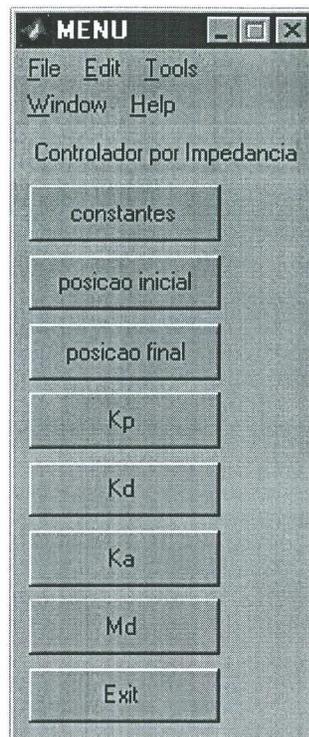


figura B.7: menu de seleção principal

O primeiro sub menu (figura B.8) permite a definição do tempo total de simulação, tempos de aceleração em cada uma das direções: X, Y e Z.

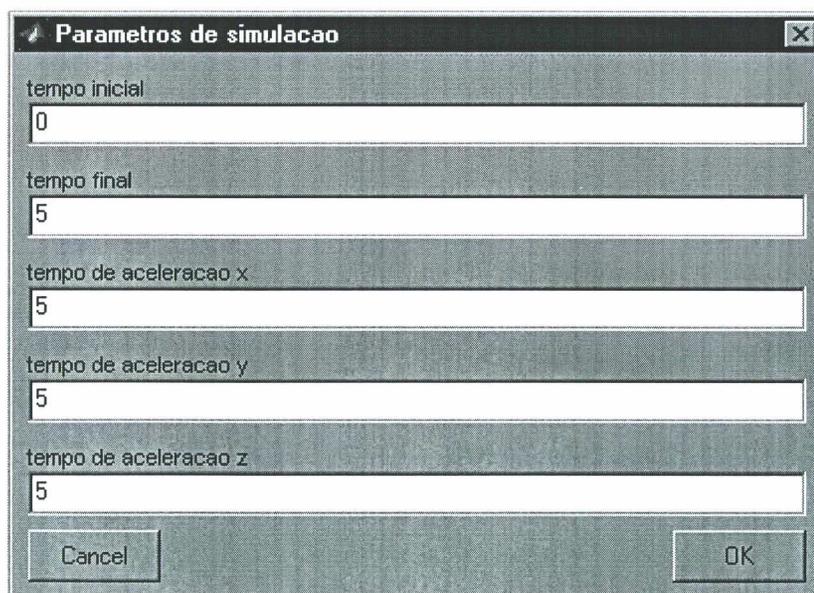
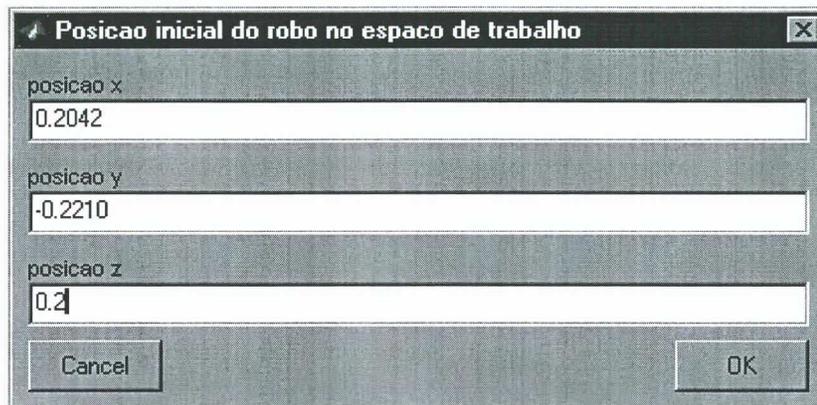


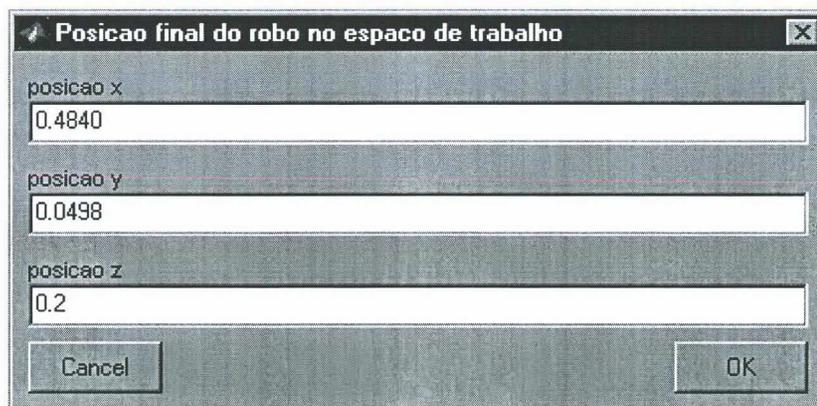
figura B.8: sub menu de seleção dos parâmetros de simulação

O segundo e terceiro sub menus, exibidos nas figuras B.9 e B.10, são utilizados para a entrada dos valores iniciais e finais de posição definindo a trajetória a ser seguida pelo robô, nas direções X, Y e Z. Os valores são comparados com os limites físicos do robô, e em caso de valores fora dos padrões, uma mensagem de erro é exibida (figura B.11) alertando para o erro e um novo valor deve ser digitado.



A screenshot of a dialog box titled "Posicao inicial do robo no espaco de trabalho". It contains three input fields: "posicao x" with the value "0.2042", "posicao y" with the value "-0.2210", and "posicao z" with the value "0.2". At the bottom, there are "Cancel" and "OK" buttons.

figura B.9: sub menu para escolha da posição inicial nas direções X, Y e Z



A screenshot of a dialog box titled "Posicao final do robo no espaco de trabalho". It contains three input fields: "posicao x" with the value "0.4840", "posicao y" with the value "0.0498", and "posicao z" with the value "0.2". At the bottom, there are "Cancel" and "OK" buttons.

figura B.10: sub menu para escolha da posição final nas direções X, Y e Z

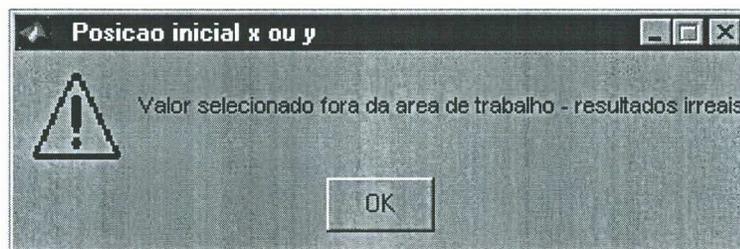


figura B.11: mensagem alertando para valores fora da faixa de trabalho

Os submenus para entrada dos ganhos M_d , B_d e K_d , utilizados no controle de impedância, e ainda o ganho K_a , que representará a rigidez do meio são similares ao da figura B.12, alterando somente o título e nome da variável a ser alterada.

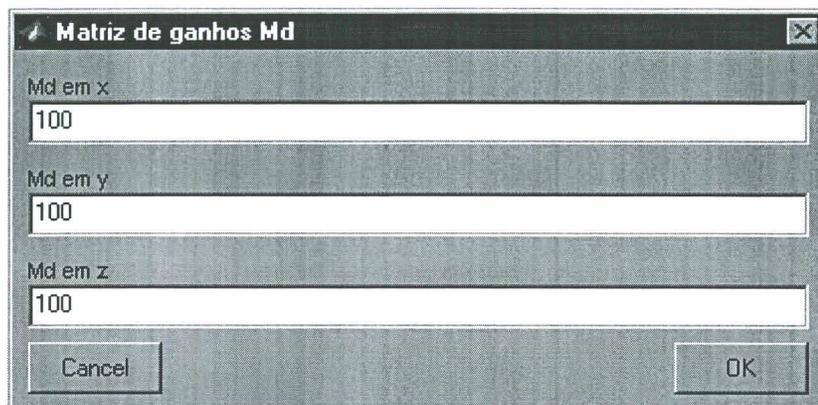


figura B.12: sub menu para escolha dos ganhos da matriz M_d .

Através do uso de menus, é possível agilizar o processo de alteração e verificação de novos ganhos, além de permitir a utilização de modo didático, pois não tendo a necessidade de editar arquivos, pode-se trabalhar todo o tempo no ambiente de simulação.

A adoção de rotinas de validação de valores reduz a possibilidade de ocorrência de erro, evitando assim simulações com valores irrealistas. Ao fazer a escolha do tempo inicial e final para a simulação, caso o valor final for menor do que o inicial, os dois serão permutados e uma mensagem de alerta será exibida (figura B.13).

Durante a escolha dos tempos de aceleração, caso o valor escolhido em uma das direções for menor do que o tempo de simulação, será exibido um alerta (figura B.14) e adotado como valor de aceleração o máximo possível, ou seja, o próprio tempo de simulação.

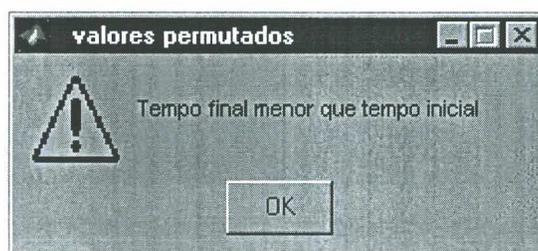


figura B.13: alerta de tempo final menor do que tempo inicial

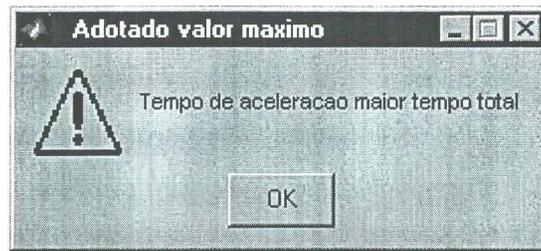


figura B.14: alerta de tempo de aceleração maior do que o tempo de simulação

APÊNDICE C – PROGRAMAS DESENVOLVIDOS PARA O XOBERON

Apresentam-se neste apêndice as alterações feitas nas rotinas já existentes do XOBERON, para possibilitar a implementação do controle de impedância.

SCARACarthKin.Mod

Foram acrescentadas 3 procedures para cálculo do Jacobiano Inverso e Jacobiano Transposto

```

MODULE SCARACarthKin;

(....)

IMPORT Math:=MathL, Base, O:=Objects, Parser:=ConfigP, Drive, GD:= GlobalDefs;

CONST
  Version = 960914;
  q0*=0; q1*=1; q2*=2; q3*=3;
  x*=0; y*=1; z*=2; phi*=3;

TYPE
  CarthKin* = POINTER TO CarthKinDesc;
  CarthKinDesc* = RECORD (Base.ObjDesc)
    z0*, l1*, l2*, l1l1*, l2l2*, l1l2*, rMin*, rMax* : LONGREAL;
    d* : ARRAY 4 OF Drive.Drive;
    minq-, maxq- : GD.Vector4;
  END;

  JakobiMatrix* = ARRAY 2,2 OF LONGREAL;

  Trigotyp* = RECORD
    q0*          : LONGREAL;
    q1*          : LONGREAL;
    cosq0* : LONGREAL;
    sinq0* : LONGREAL;
    sinq1* : LONGREAL;
    cosq0q1*   : LONGREAL;
    sinq0q1*   : LONGREAL;
  END;

  Matrix* = ARRAY OF GD.Vector4;

VAR
  version- : LONGINT;
  llver, t1, t2, t3, t4, t5, t6, t7, t8, t9, t10 : LONGREAL;

PROCEDURE (k : CarthKin) TrigoCalc*( q0,q1 : LONGREAL; VAR t:Trigotyp );
BEGIN
  t.cosq0:=Math.cos(q0);
  t.sinq0:=Math.sin(q0);
  t.sinq1:=Math.sin(q1);
  t.sinq0q1:=Math.sin(q0+q1);
  t.cosq0q1:=Math.cos(q0+q1);
END TrigoCalc ;

PROCEDURE (k : CarthKin) JakobiCalc*( t : Trigotyp ; VAR jak : JakobiMatrix);
VAR minusL2sinq0q1, L2cosq0q1:LONGREAL;
BEGIN
  minusL2sinq0q1:=-k.l2*t.sinq0q1;
  L2cosq0q1:=k.l2*t.cosq0q1;

```

```

        jak[0,0]:=-k.11*t.sinq0+minusL2sinq0q1; jak[0,1]:=minusL2sinq0q1;
        jak[1,0]:=k.11*t.cosq0+L2cosq0q1;
        jak[1,1]:=L2cosq0q1;
END JakobiCalc;

(* Acrescentado por cesarm - Cálculo da Derivada do Jacobiano - Juntas 0 e 1*)
PROCEDURE (k : CarthKin) DJakobiCalc*( t : Trigotyp; istspeedR: GD.Vector4;
VAR Djak : JakobiMatrix);
VAR dq0q1: LONGREAL;
BEGIN
    dq0q1:=istspeed[0] + istspeed[1];
    Djak[0,0]:= -k.11*t.cosq0*istspeed[0] - k.12*t.cosq0q1*dq0q1;
    Djak[0,1]:= -k.12*t.cosq0q1*dq0q1;
    Djak[1,0]:= -k.11*t.sinq0*istspeed[0] - k.12*t.sinq0q1*dq0q1;
    Djak[1,1]:= -k.12*t.sinq0q1*dq0q1;
END DJakobiCalc;

(* Acrescentado por cesarm - Cálculo da Inversa do Jacobiano - Juntas 0 e 1 *)
PROCEDURE (k : CarthKin) JakInvCalc01*( t : Trigotyp; istposR: GD.Vector4;
VAR JakInv01 : JakobiMatrix);
VAR l1sinq1, l2sinq1: LONGREAL;
BEGIN
    l1sinq1:= k.11*t.sinq1;      l2sinq1:= k.12*t.sinq1;
    JakInv01[0,0]:= t.cosq0q1 / l1sinq1;
    JakInv01[0,1]:= t.sinq0q1 / l1sinq1;
    JakInv01[1,0]:= (-k.11*t.cosq0 - k.12*t.cosq0q1) / (k.11*l2sinq1);
    JakInv01[1,1]:= (-k.11*t.sinq0 - k.12*t.sinq0q1) / (k.11*l2sinq1);
END DJakobiCalc;

(* Acrescentado por cesarm - Cálculo da Inversa do Jacobiano - juntas 2 e 3 *)
(* Eq. (90) de (Bier, 2000) *)
PROCEDURE (k : CarthKin) JakInvCalc23*( t : Trigotyp; istposR: GD.Vector4;
VAR JakInv23 : JakobiMatrix);
VAR l2sinq1: LONGREAL;
BEGIN
    l2sinq1:= k.12*t.sinq1;
    JakInv23[0,0]:= 0;
    JakInv23[0,1]:= 0;
    JakInv23[1,0]:= -t.cosq0 / l2sinq1;
    JakInv23[1,1]:= -t.sinq0 / l2sinq1;
END DJakobiCalc;

PROCEDURE (k : CarthKin) PosRobotToCart*(pr : GD.Vector4; t : Trigotyp; VAR pc
: GD.Vector4; VAR rechts: BOOLEAN);
BEGIN
    pc[x]:=k.11*t.cosq0+k.12*t.cosq0q1;
    pc[y]:=k.11*t.sinq0+k.12*t.sinq0q1;
    pc[z]:=-pr[q2] + k.z0 ;
    pc[phi]:=pr[q0]+pr[q1]-pr[q3];
    rechts:=pr[q1]>0;
END PosRobotToCart;

PROCEDURE (k : CarthKin) SpeedRobotToCarth*( sr : GD.Vector4; jak :
JakobiMatrix; VAR sc : GD.Vector4 );
BEGIN
    sc[x]:=jak[0,0]*sr[q0]+jak[0,1]*sr[q1];
    sc[y]:=jak[1,0]*sr[q0]+jak[1,1]*sr[q1];
    sc[z]:=-sr[q2];
    sc[phi]:=sr[q0]+sr[q1]-sr[q3];
END SpeedRobotToCarth;

```

(...)

Módulo StateCtrl.Mod

```

(...)
PROCEDURE(c: StateCtrl) Assign*(o: Base.Object; name: ARRAY OF CHAR):BOOLEAN;

VAR   value : Base.Value;

BEGIN
  IF (o=NIL) THEN RETURN FALSE
  ELSIF (o IS Base.Value) THEN      value:=o(Base.Value);
    IF name = 'Md' THEN c.iPara.Md:=SHORT(value.x);
    ELSIF name = 'Kd' THEN c.iPara.Kd:=SHORT(value.x);
    ELSIF name = 'Kp' THEN c.iPara.Kp:=SHORT(value.x);
    ELSIF name = 'clock' THEN c.iPara.clock:=SHORT(value.x);
    ELSE RETURN c.Assign^(o, name);
    END
  ELSE RETURN c.Assign^(o, name)
  END; RETURN TRUE;
END Assign;

PROCEDURE AttrMsg(obj: StateCtrl; VAR M: O.AttrMsg)
BEGIN
  IF M.id = O.get THEN
  IF M.name = "Gen"
    THEN M.class:=O.String; M.s:='ImpCtrl.NewICtrl'; M.res:=0
  ELSIF M.name = "Md"
    THEN M.class:=O.Real; M.x:=obj.iPara.Md; M.res:=0
  ELSIF M.name = "Kd"
    THEN M.class:=O.Real; M.x:=obj.iPara.Kd; M.res:=0
  ELSIF M.name = "Kp"
    THEN M.class:=O.Real; M.x:=obj.iPara.Kp; M.res:=0
  ELSIF M.name = "clock"
    THEN M.class:=O.Real; M.x:=obj.iPara.clock; M.res:=0
  ELSE Base.Handler(obj, M)
    END
  ELSIF M.id = O.set
    THEN Base.Handler(obj, M)
  ELSIF M.id = O.enum
    THEN M.Enum("Md"); M.Enum("Kd"); M.Enum("Kp"); M.Enum("clock");
    Base.Handler(obj, M)
    END;
  END AttrMsg;

PROCEDURE Handler*(obj: O.Object; VAR M: O.ObjMsg);
BEGIN
  IF obj IS StateCtrl THEN
    WITH obj: StateCtrl DO
      IF M IS O.AttrMsg
        THEN AttrMsg(obj, M(O.AttrMsg))
        ELSE Base.Handler(obj, M) END;
    END
  END
  END Handler;

(...)

PROCEDURE (c : StateCtrl) ChangePara* (VAR para: IPara);
(* modificado para controle por impedancia *)
VAR p : IPara;
BEGIN
  XOK.NotInterruptible;
  p:= c.iPara;
  c.iPara:= para;
  para:= p;
  XOK.Interruptible;
END ChangePara;

```

```
PROCEDURE (c : StateCtrl) Algo*( junta: LONGINT; istforceTool, istspeedCarth,
dq, istposCarth, solaccCarth, solspeedCarth, solposCarth: REAL; Jad, Inv01,
Inv23, Jac : SCK.JakobiMatrix ) : REAL;
```

```
VAR
```

```
force, epsilon, C, K, s: REAL;
```

```
BEGIN
```

```
(*Lei de controle por impedancia*)
```

```
(* dot = derivada; inv = inversa; ddot = segunda derivada
```

```
y=inv(Jacob).{ddot(xd)-dot(Jacob).dot(q)+inv(MD).[Kd.dot(x-xd)+Kp.(x-xd)-ha]}
```

```
u=y+DynamicCompens+FrictionCompens-TranspJacob.h*)
```

```
(* ----- significado das variaveis -----
```

```
xd = pos desej x y z phi
```

```
dxd = vel desej x y z phi
```

```
accxd = solaccx; accyd = solaccy; accz = solaccz; accphi= solaccphi;
```

```
Obs: todos os valores no espaço cartesiano.
```

```
-----*)
```

```
xd:= solposCarth; dxd:= solspeedCarth; aacxd:= solaccCarth; hx:= istForce;
```

```
Mdx:=c[0].iPara.Md; Mdy:=c[1].iPara.Md; Mdz:=c[2].iPara.Md;
```

```
Mdphi:=c[3].iPara.Md;
```

```
Kdx:=c[0].iPara.Kd; Kdy:=c[1].iPara.Kd; Kdz:=c[2].iPara.Kd;
```

```
Kdphi:=c[3].iPara.Kd;
```

```
Kpx:=c[0].iPara.Kp; Kpy:=c[1].iPara.Kp; Kpz:=c[2].iPara.Kp;
```

```
Kpphi:= c[3].iPara.Kp;
```

```
(*-----
```

```
Jad := Derivada do Jacobiano
```

```
x[0]= pos atual x x[1]= pos atual y x[2]=pos atual z x[3]= pos atual phi
```

```
dx[0]= vel atual x dx[1]= vel atual y dx[2]=vel atual z dx[3]= vel atual phi
```

```
hx[0]= forca em x hx[1]= forca em y hx[2]=forca em z hx[3]= forca em phi
```

```
aux= accxd - Jad . dq + Md. (Kd . (dx - dxd) + Kp . (x-xd) -hx)]
```

```
-----*)
```

```
x:= istposCarth;
```

```
dx:= istspeedCarth;
```

```
aux1:= accxd[0]-(Jad[0,0]*dq[0]+Jad[0,1]*dq[1])+Mdix*(Kdx*(dx[0]-
```

```
dxd[0])+Kpx*(x[0]-xd[0])-hx[0]);
```

```
aux2:= accxd[1]-(Jad[1,0]*dq[0]+Jad[1,1]*dq[1])+Mdiy*(Kdx[1]*(dx[1]-
```

```
dxd[1])+Kpy*(x[1]-xd[1])-hx[1]);
```

```
aux3:= accxd[2]+Mdiz*(Kdz*(dx[2]-dxd[2])+Kpz*(x[2]-xd[2])-hx[2]);
```

```
aux4:= accxd[3]+Mdiphi*(Kdphi*(dx[3]-dxd[3])+Kpphi*(x[3]-xd[3])-hx[3]);
```

```
(* Inv01 = Jacobiano Inversa qoq1
```

```
Inv 23 = Jacobiano Inverso q2q3
```

```
*)
```

```
(* Calcula ycontrol = inv(Jacob) . aux *)
```

```
ycontrolx:= Inv01[0,0] * aux1 + Inv01[0,1] * aux2;
```

```
ycontroly:= Inv01[1,0] * aux1 + Inv01[1,1] * aux2;
```

```
ycontrolz:= Inv23[0,0] * aux1 + Inv23[0,1] * aux2 - aux3;
```

```
ycontrolphi:= Inv23[1,0] * aux1 + Inv23[1,1] * aux2 - aux4;
```

```
(* Calcula u = ycontrol + dynnamicCompens + FrictionCompens - TranspJacob . h
```

```
*)
```

```

u[0] = ycontrolx + (Jac[0,0] * hx[0] + Jac[1,0] * hx[1] + hx[3]);
u[1] = ycontroly + (Jac[0,1] * hx[0] + Jac[1,1] * hx[1] + hx[3]);
u[2] = ycontrolz - hx[ 2];
u[3] = ycontrolphi - hx[3];

```

```

RETURN u[junta]

```

```

END Algo;

```

```

(...)

```

```

NEW(c);
c.handle:=Handler;
NEW(p);
c.iPara:= p;
(* Inicializa ganhos controlador por impedancia *)
(* Valores obtidos de simulacoes no Matlab *)
c.iPara.Md:= 0.01;
c.iPara.Kd:= 500.0;
c.iPara.Kp:= 2500.0;
O.NewObj:=c;

```

```

(...)

```

Módulo GolPontosBier

(* Este Modulo carrega um arquivo de dados de posicao, velocidade e aceleracao provenientes do MatLab (Traj.mat), converte-os em LONGREAL e manda-os para o controlador do robo. O posicionamento inicial é feito pelo Gerador de Trajetorias ponto a ponto antigo (Bahnplaner.Mod) em baixa velocidade. Atualmente esta preparado para ler ate 10.000 linhas de pontos e ainda nao foi determinado o numero maximo de linhas possiveis. *)

```

MODULE GolPontosBier;

```

```

IMPORT

```

```

F:= XFiles, XT:= XTexts,
XO:=XOberon, StrConv,
GD:=GlobalDefs, XOK:=PPCXOKernel,
M := Main3, BP := Bahnplaner, MF := MatlabFile2;

```

```

CONST

```

```

Clock = 1; Clockms = 0.001;
(* NoPlot = 0; *)

```

```

VAR

```

```

convOk,
dummy, dummyMF, (* Verifica se a conversao foi feita de acordo *)
primeiraVez, (* Boolean utilizado sem proposito especial *)
leituraOk, (* Utilizado para posicionar o Rider *)
fileOk, (* Verificacao dos dados do Arquivo *)
autoritOk, (* Procura pelo Arquivo Traj.mat *)
controlOk, (* Verifica se o controlador esta livre *)
ptIniOk, (* Troca o tipo de controlador para StateCtrl *)
trajetoria (* Verifica se o robo foi para o ponto inicial da

```

```

trajetoria *)

```

```

started : BOOLEAN;
tamanhoFile: LONGINT; (* Tamanho do Arquivo em caracteres *)
m: LONGINT;
R: F.Rider; (* Posicionador para leitura do Arquivo *)
posDes, velDes, accDes: GD.Vector4;
pos0, vel0, acc0,

```

```

Acc.*)
pos1, vel1, acc1, (* Valores Convertidos de Posicao, Velocidade e

```

```

pos2, vel2, acc2,
pos3, vel3, acc3 : ARRAY 10100 OF LONGREAL;
coluna, (* Numero da coluna do Arquivo *)
linha, (* Numero da linha *)

```

```

envio: LONGINT;      (* Numero de linhas enviadas      *)
mainevent : XOK.MainEvent;
ctrlevent : XOK.EveryEvent;
matauthority : LONGINT;
mlab: LONGINT;
contador,      (* Conta o no. de vezes que EnvDados foi executado *)
amostragem : SHORTINT;      (* Define o periodo para amostragem de dados
no Matlab *)
time : REAL;
q, vq : GD.Vector4;      (* plottype : LONGINT; *)

PROCEDURE LerTodosVal*(): BOOLEAN;
VAR
  f: F.File;      (* Arquivo carregado      *)
  Valor: ARRAY 32 OF LONGREAL;      (* Valor transformado dos caracteres *)
  a: ARRAY 1024 OF CHAR;      (* ARRAY com os caracteres que sao numeros *)
  i, j, x: LONGINT;
  ch: CHAR;      (* Caracter carregado para verificacao      *)

BEGIN
  f:= F.Old("Traj.mat");
  IF f#NIL THEN
    i:=0; x:=0;
    tamanhoFile:=F.Length(f);
    IF primeiraVez THEN F.Set(R,f,0);
    ELSE F.Set(R,f,F.Pos(R));
    END;
    F.Read(R,ch);
    WHILE ~R.eof DO
      IF ((ch>2AX)&(ch<2FX))OR((ch>29X)&(ch<40X))OR(ch=65X) THEN
        WHILE ((ch=2DX)OR(ch=2EX)OR((ch>29X)&(ch<40X))OR(ch=65X)) DO
          a[i]:= ch; INC(i);
          F.Read(R,ch);
        END;
        convOk:= StrConv.StrToReal(a, Valor[x]);
        INC(x);
        IF (x=12) THEN
          coluna:=0;
          pos0[linha]:=Valor[coluna]; INC(coluna);      (* Coleta dos
nove valores *)
          pos1[linha]:=Valor[coluna]; INC(coluna);      (* de posicao,
vel e acel *)
          pos2[linha]:=Valor[coluna]; INC(coluna);
          pos3[linha]:=Valor[coluna]; INC(coluna);;
          vel0[linha]:=Valor[coluna]; INC(coluna);
          vel1[linha]:=Valor[coluna]; INC(coluna);
          vel2[linha]:=Valor[coluna]; INC(coluna);
          vel3[linha]:=Valor[coluna]; INC(coluna);;
          acc0[linha]:=Valor[coluna]; INC(coluna);
          acc1[linha]:=Valor[coluna]; INC(coluna);
          acc2[linha]:=Valor[coluna]; INC(coluna);
          acc3[linha]:=Valor[coluna]; INC(coluna);;
          primeiraVez:=FALSE;
          x:=0;
          INC(linha);
        END;
        FOR j:=0 TO i DO a[j]:=" " END;
        i:=0;
      ELSE F.Read(R,ch);
      END;
    END;
    i:=0;
    F.Close(f);
  ELSE fileOk:=FALSE;
  END;
  fileOk:=TRUE;
  IF primeiraVez THEN RETURN FALSE;

```

```

ELSE RETURN TRUE;
END;
END LerTodosVal;

PROCEDURE EnvDados ( e: XOK.Event);
VAR
BEGIN
  IF m < linha THEN
    posDes[0]:=pos0[m];
    posDes[1]:=pos1[m];
    posDes[2]:=pos2[m];
    posDes[3]:=pos3[m];;
    velDes[0]:=vel0[m];
    velDes[1]:=vel1[m];
    velDes[2]:=vel2[m];
    velDes[3]:=vel3[m];;
    accDes[0]:=acc0[m];
    accDes[1]:=acc1[m];
    accDes[2]:=acc2[m];
    accDes[3]:=acc3[m];;
    INC(m);

  (* foi acrescentado velDesc para compatibilidade com o Main3 *)
  (* Poderiam ser transmitidos ao Main3 os valores das aceleracoes *)

  dummy := M.SetRobPosSpeed(mataauthority, posDes, velDes, accDes);

  (* IF ~started THEN INC(MF.oversamples) END; *)
  IF contador = amostragem THEN
  CASE MF.plotttype OF MF.NoPlot : (* do nothing *);
  |MF.Q0Q1 :
    M.GetRobPosSpeed(q, vq);
    dummyMF:= MF.Sample(SHORT(posDes[0]), SHORT(q[0]),
    SHORT(posDes[1]), SHORT(q[1]), SHORT(velDes[0]), SHORT(vq[0]),
    SHORT(velDes[1]), SHORT(vq[1]), time);
  |MF.Q2Q3 :
    M.GetRobPosSpeed(q, vq);
    dummyMF:= MF.Sample(SHORT(posDes[2]), SHORT(q[2]),
    SHORT(posDes[3]),SHORT(q[3]), SHORT(velDes[2]), SHORT(vq[2]),
    SHORT(velDes[3]), SHORT(vq[3]), time);
  |MF.Tau01 :
    M.GetRobPosSpeed(q, vq);
    dummyMF:= MF.Sample(M.outValues[0].force, SHORT(accDes[0]),
    M.outValues[1].force, SHORT(accDes[1]), SHORT(velDes[0]),
    SHORT(vq[0]), SHORT(velDes[1]), SHORT(vq[1]), time);
  |MF.Tau23 : M.GetRobPosSpeed(q, vq);
    dummyMF:= MF.Sample(M.outValues[2].force, SHORT(accDes[2]),
    M.outValues[3].force, SHORT(accDes[3]), SHORT(velDes[2]),
    SHORT(vq[2]), SHORT(velDes[3]), SHORT(vq[3]), time)
  ELSE END;
    time:=time + amostragem*Clockms;
    contador := 0;
    ELSE (* nao faz nada *)
    END;
    INC(contador);

  ELSE leituraOk:=FALSE;
  END;
END EnvDados;

PROCEDURE Verificar( e: XOK.Event);
BEGIN
  WHILE leituraOk DO
  END;
  ctrlevent.UnInstall;
  IF MF.plotttype#MF.NoPlot THEN
    dummyMF:=MF.StopSample();
    dummyMF:=MF.WriteFile();
  
```

```

END;
IF ~ M.AuthorityFree(mataauthority) THEN HALT(120) END;
END Verificar;

PROCEDURE Instalar*;
VAR wt: XT.Writer;
BEGIN
  ASSERT(XO.Writer(wt));
  leituraOk:=TRUE;
  fileOk:= FALSE;
  convOk:= TRUE;
  primeiraVez:=TRUE;
  ptIniOk := FALSE;
  linha:=0; envio:=0; amostragem:=10; contador := amostragem;
  leituraOk:= LerTodosVal();
  ptIniOk:= BP.MoveAbsAllJoint( SHORT(pos0[0]), SHORT(pos1[0]),
SHORT(pos2[0]),
      SHORT(pos3[0]), 0.5, 0.5);
  autoritOk := FALSE;
  controlOk := FALSE;
  controlOk:= M.ChangeCtrl(GD.SStateCtrl);
  IF MF.plottype#MF.NoPlot THEN
    time:=0.0; dummyMF:=MF.StartSample(); MF.oversamples:=0;
  END;
  IF( controlOk & leituraOk & ptIniOk & fileOk ) THEN
    autoritOk:=M.GetAuthority(mataauthority);
    IF autoritOk THEN
      XT.WriteLine(wt);XT.WriteLine(wt);
      XT.WriteString(wt, " Iniciado o movimento");
      m:=0;
      XOK.InitMain(mainevent); mainevent.Install(Verificar);
mainevent.Notify;
      XOK.InitEvery(ctrlevent, Clock, XOK.ONEm);
ctrlevent.Install(EnvDados);
      ELSE
        XT.WriteLine(wt);XT.WriteLine(wt);
        XT.WriteString(wt, " O controlador esta sendo utilizado por
outro modulo");
      END;
    ELSE
      XT.WriteLine(wt);XT.WriteLine(wt);
      IF ~controlOk THEN
        XT.WriteString(wt, " Nao foi possivel alterar o
controlador!!");
        XT.WriteLine(wt);
      END;
      IF ~leituraOk THEN
        XT.WriteString(wt, " Encontrou o arquivo, mas nao converteu
os dados corretamente!!");
        XT.WriteLine(wt);
        XT.WriteString(wt, " Verifique se todas as linhas estao
completas.");
        XT.WriteLine(wt);
      END;
      IF ~ptIniOk THEN
        XT.WriteString(wt, " Nao foi possivel movimentar o robo
para a posicao inicial!!");
        XT.WriteLine(wt);
      END;
      IF ~fileOk THEN
        XT.WriteString(wt, " Arquivo nao encontrado!!");
        XT.WriteLine(wt);
      END;
    END;
    XT.WriteLine(wt); XT.Append(XO.XLog(), wt.buf);
  END Instalar;

```

```
BEGIN
    NEW(mainevent);
    NEW(ctrlevent);
END GolPontosBier.

XSystem.Call GolPontosBier.Instalar ~
XSystem.Call GolPontosBier.Resultado ~

XSystem.Call MatlabFile2.SetQ0Q1Plot ~
XSystem.Call MatlabFile2.SetQ2Q3Plot ~
XSystem.Call MatlabFile2.SetTau01Plot ~
XSystem.Call MatlabFile2.SetTau23Plot ~
```

Referências Bibliográficas

- [1] AMARAL, S. do. “Controle a Estrutura Variável Interagindo com Ambientes Passivos”. Florianópolis – SC. Tese de Doutorado, Pós Graduação em Engenharia Elétrica – UFSC, (2000)
- [2] ASADA, H.; SLOTINE, J.J.E.. **Robot Analysis and Control**. New York: John Wiley and Sons, (1986)
- [3] ASIMOV, Isaac.. **Visão de Robô**. Rio de Janeiro: Record, (1994)
- [4] BARCZAK, C. L.. **Controle Digital de sistemas Dinâmicos, Projeto e Análise**. São Paulo: Ed. Edgard Blucher LTDA, (1995)
- [5] BATISTELA, Sandro.. **Controle de Força e Posição de Robôs Manipuladores Utilizando Redes Neurais Artificiais**. Florianópolis – SC. Dissertação de mestrado, Pós Graduação em Engenharia Elétrica – UFSC, (1999)
- [6] BARRIENTOS, A.; Penin, L. F.; Balaguer, C.; Aracil, R.. **Fundamentos de Robótica**. Universidade Politécnica de Madri : Ed. McGraw-Hill, (1997)
- [7] BIER, Carlos C.; Golin, Júlio; FERNANDES Junior, C.. **Estudo e Implementação de um Gerador Off-Line de Trajetórias de Referência para um Robô industrial SCARA**. Porto Alegre – RS. In. II Congresso Sul-Americano de Engenharia de Controle e Automação (CONET SUL – 99), março (1999)
- [8] BIER, Carlos C.. **Implementação de um Algoritmo de Controle de Força em um Manipulador SCARA**. Florianópolis – SC. Dissertação de Mestrado, Pós Graduação em Engenharia Mecânica, (2000)
- [9] BISSO, Carlos J. V.. **Controle de Posição de Robôs Manipuladores Rígidos e com Transmissões Flexíveis utilizando Controladores nas Estruturas de Dois Graus de Liberdade**. Florianópolis – SC. Dissertação de Mestrado, Pós Graduação em Engenharia Elétrica – (1999)

- [10] CABRERA, Mário A. Análise do Controle de Impedância Aplicado ao Controle de Robôs em Tarefas de Contato. Campinas. Dissertação de Mestrado, Pós Graduação em Engenharia Elétrica – UNICAMP – (1990)
- [11] CACCAVALE, Fabrizio; Natale, Ciro; SICILIANO, Bruno; VILLANI, Luigi. **SIX-DOF IMPEDANCE CONTROL BASED ON ANGLE/AXIS REPRESENTATIONS.** IEEE Transactions on Robotic and Automation, Vol. 15, No. 2, Abril, (1999)
- [12] CAI, L. and SONG, G. **“Robust Position/Force Control of Robot Manipulators during Contact Tasks”**, American Control Conference, Baltimore, maryland, USA, pp.216-220, (1994)
- [13] CHEAH, C. C., WANG, D.. **“Learning Impedance Control for Robotic Manipulators”** IEEE Transactions on Robotics and Automation, Vol. 14, No. 3, June, pp 452-465.(1998)
- [14] CHIAVERINI, S.; SCIAVICCO, L.. **The Parallel Approach to Force- Position Control of Robotic Manipulators.** IEEE Trans. Robotics and Automation, vol. 9, n.4, pp. 361-373, (1993)
- [15] CRAIG, J. J.. **Introduction to Robotics: Mechanics & Control.** New York: Addison-Wesley, (1986)
- [16] De SCHUTTER, J.. **“A Study of Active Compliant Motion Control Methods for Rigid Manipulators based on a Generic Scheme”** IEEE International Conference on Robotics and Automation, pp. 1060-1065.
- [17] De SCHUTTER, J., VAN BRUSSEL, H.. **“Compliant Robot Motion II, A Control Approach Based on External Control Loops”** International Journal of Robotics Research 7, No. 4, pp. 18-33,(1988)

- [18] DOEBELIN, E.O.. **Measurement Systems: Application and Design**. New York: McGraw-Hill, (1990)
- [19] FRANKLIN, G. F.; POWELL, J. D.; EMANI-NAEINI, Abbas. **Feedback Control of Dynamic Systems**. New York: Addison-Wesley Publishing Company, (1994)
- [20] FREUND, E., PESARA, J., “**High- bandwidth Force and Impedance Control for Industrial Robots**” *Robotica*, Vol. 16, pp 75-87,(1998)
- [21] GOLIN, Júlio. **Desenvolvimento e Testes de um Gerador de Trajetórias para o robô inter – SCARA**. Florianópolis – SC. Projeto de fim de curso, Graduação em Engenharia de Controle e Automação industrial – UFSC, (1999)
- [22] GOLIN, Júlio; GUENTER, Raul; WEIHMANN, Lucas. **Manual do usuário do Robô INTER**. Florianópolis: Laboratório de Robótica – UFSC, (1998)
- [23] GORINEVSKY, D. M.; FORMALSKY, A. M.; SCHNEIDER, A. YU. **Force Control of Robotic Systems**. New York: McGraw-Hill, (1997).
- [24] GROOVER, M. et al. **Robótica: Tecnologia e Programação**. São paulo: McGraw-Hill, (1997)
- [25] HAMEL, W.. “www.utk.edu/~whamel”. Endereço na internet, acessado em 05 de agosto de 2002, atualizado em maio.(2002)
- [26] HIRZINGER, G.; LANDZETTEL, K.. “**Sensory Feedback Structures for Robots with supervised learning**” *IEEE International Conference on Robotics and Automation*, pp. 627-635, (1985)
- [27] HOGAN, N.. “**Impedance Control: Na Approach to Manipulation, Parts I-III**” *ASME Journal of Dynamic Systems, Measurement and Control*, Vol. 107, No. 1, pp 1-24, (1985)

- [28] HOGAN, N.. “www.web.mit.edu/be/people/hogan.html”. Endereço na internet, acessado em 12 de agosto de 2002, atualizado em abril (2002).
- [29] KHATIB, O.. “**A Unified Approach for Motion and Force Control of Robot Manipulators: The Operational Space Formulation**” IEEE Journal of Robotics and Automation 3, No. 1, pp. 43-53, (1987)
- [30] KANSEL, T. M.. **Mobile Robotics. State of the Art Review.** Robotics 2, North-Holland, (1986)
- [31] KRAEUTER, C.; BLACKWELL, B.. “**ROBOLEG: A Robotics Soccer-ball Kicking Leg**” IEEE Int. Conf. On Robotics and Automation, Nagoya, Japão, junho (1995)
- [32] KWAN, C. M.. **Robusts Adaptative Force/Motion Control of Constraned Robots**, IEEE Procedures – Control Theory Appl. 143. pp. 104-109. (1998)
- [33] LAWRENCE, D. A.. **Impedance Control Stability Properties im Common Implementations**, IEEE International Conference om Robotics and Automation, pp. 1185-1190, (1988)
- [34] LEE, W. Y.; **Spatial Semantic Hierarchy for a Physical Mobile Robot**, tese de doutorado, Departamento de Ciências da Computação, Universidade do Texas, Dezembro (1996).
- [35] LEE, W. Y.; **A Guide to Programming Spot, a Mobile Robot of the University of Texas at Austin**, Laboratório de Inteligência Artificial, Universidade do Texas, AI95-235, junho (1995).
- [36] LEWIS, F. L.; ABDALLAH, C. T.; DAWSON, D.. **Control of Robot Manipulators.** New York: Macmillan Publishing Company, (1993)

- [37] LU, Z. and GOLDEMBERG, A. A.. **Roust Impedance Control and Force Regulation: Theory and Experiments**. International Journal of Robotics Research 14(3): pp. 225-254, (1995)
- [38] MAPLES, J. A., BECKER, J.J.. **“Experiments in Force Control of Robotic Manipulators”** IEEE International Conference on Robotics and Automation, pp. 695-702, (1996)
- [39] MASSON, T. M.. **Compliance and Force Control for Computer Controlled Manipulators**, IEEE Trans. Systems, Man and Cybernetics, v. SMC-11, n.6, pp 418-432, (1981)
- [40] MEGGIOLARO, M.A.. **“Controle de Alta Precisão de Manipuladores de Longo Alcance com Aplicação em Sistemas Médicos”**, 25th Biennial Mechanisms Conference - ASME. (1997).
- [41] MENDES, Marcos F.. **Controle de Força de Robôs Manipuladores Interagindo com Ambientes de Elasticidade não Linear**. Florianópolis – SC. Dissertação de Mestrado, Pós Graduação em Engenharia Elétrica – UFSC, (1999)
- [42] MOSSENBOCK, H.. **Object-Oriented Programming in Oberon-2**. Berlin Heidelberg: Springer-Verlag, (1993)
- [43] NARENDA, K. S.. **Neural Networks for Real-Time Control**. 36th IEEE Conference on Decision and Control – CDC’97, San diego, Califórnia, USA, pp.1026-1031. (1997)
- [44] NOSTRAND, J.W.; SAMPSON, E.L..**Robots in Service Industry**. In Concise International Encyclopedia of Robotics, Editor R.C. Dorf, John Wiley & Sons, (1990)
- [45] REISER, M.. **The Oberon System: User Guide and Programmer’s Manual**. New York: Addison-Wesley Publishing Company, (1991)

- [46] SALISBURY, K. J.. **Active Stiffness Control of a Manipulator in Cartesian Coordinates**. 19th IEEE Conference on Decision and Control, Albuquerque, pp. 95-100, (1980)
- [47] SCIAVICCO, L.; SICILIANO, B.. **Modeling and Control of Robot Manipulator**. New York: Mcgraw-Hill, (1996)
- [48] SICILIANO, B.. **“Robot Force Control “**. Workshop on Industrial Robotics, Portugal, (1999).
- [49] SILVEIRA, Carlos A. B.; GUENTER, R. Contribuição ao Estudo e Projeto de Controladores de Força para Robôs Manipuladores. Florianópolis, SC. Dissertação de Mestrado, Pós Graduação em Engenharia Elétrica – UFSC, (1998).
- [50] SLOTINE, J. J. E.. **Sliding Controller Design for Nonlinear Systems**. International Journal of Control, 40(2), pp. 912-921, (1984)
- [51] SMALLWOOD, D.; BACHMAYER, R.; WHITCOMB, L.. **“A New Remotely Operated Underwater Vehicle for Dynamics and Control Research”**. Proceeding of the 11th International Symposium on Unmanned Untethered Submersible Technology, Durham, NH, páginas 370-377. Setembro (1999).
- [52] SPONG, M. W.; VIDYASAGAR, M.. **Robot Dynamics and Control**. Singapore: John Wiley & Sons, (1989)
- [53] SURDILOVIC, D.; KIRCHHOF, J.. **A New Position Based Force/Impedance Control for Industrial Robots**. Minneapolis, Minnesota: In International Conference on Robotics and Automation – IEEE, April (1996)
- [54] TORRALVA, C., **“Nueva Clasificación de la IFR para los Robots”**, Automatización industrial y revista de robótica, 67, abril (1992).

- [55] VARGAS, F. J. T. PIERI, E. R. CASTELAN NETO, E. Controle Força-Posição de Robôs Manipuladores. Florianópolis – SC. Dissertação de Mestrado. Pós Graduação em Engenharia Elétrica – UFSC, (2000)
- [56] WEIHMANN, Lucas. **Descrição, Instalação, Programação e Funcionamento de um Robô Manipulador do tipo SCARA.** Florianópolis – SC. Dissertação de Mestrado, Pós Graduação em Engenharia Mecânica – UFSC, (1999)
- [57] WIT, Canudas; SICILIANO, Bruno; BASTIN, Georges. **Theory of robot Control.** London: Spring-Verlag, (1996)
- [58] YOERGER, D. **“Coordinated Vehicle/Manipulator Design and Control Issues for Underwater Telemanipulation”.** International Federation of Automatic Control, CAMS’92, Genova, Itália, Abril (1992).
- [59] ZENG, G. and HEMANI, A.. **An Overview of Robot Force Control.** Robótica, Vol. 15, pp. 473-482, (1997)