

INSTITUTO MILITAR DE ENGENHARIA

FÁBIO SUIM CHAGAS

MANIPULADOR BILATERAL COM REALIMENTAÇÃO
HÁPTICA

Dissertação de Mestrado apresentada ao Curso de Mestrado em Sistemas e Computação do Instituto Militar de Engenharia, como requisito parcial para obtenção do título de Mestre em Ciências em Sistemas e Computação.

Orientador: Prof. Paulo Fernando Ferreira Rosa - PhD.

Rio de Janeiro
2005

c2005

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

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

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

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

C433m Fábio Suim Chagas

Manipulador Bilateral com Realimentação Háptica / Fábio Suim

Chagas. - Rio de Janeiro : Instituto Militar de Engenharia, 2005.

130 p.: il, graf., tab.

Dissertação (mestrado) - Instituto Militar de Engenharia- Rio de Janeiro, 2005

1. Teleoperação (Robótica). 2. Manipulador Bilateral. 3. Realimentação Háptica. 4. Controle híbrido. I. Fábio Suim Chagas II. Instituto Militar de Engenharia. III. Título.

CDD 629.892

INSTITUTO MILITAR DE ENGENHARIA

FÁBIO SUIM CHAGAS

MANIPULADOR BILATERAL COM REALIMENTAÇÃO HÁPTICA

Dissertação de Mestrado apresentada ao Curso de Mestrado em Sistemas e Computação do Instituto Militar de Engenharia, como requisito parcial para obtenção do título de Mestre em Ciências em Sistemas e Computação.

Orientador: Prof. Paulo Fernando Ferreira Rosa - PhD.

Aprovada em 24 de Agosto de 2005 pela seguinte Banca Examinadora:

Prof. Paulo Fernando Ferreira Rosa - PhD. do IME - Presidente

Prof. Max Suell Dutra - Dr. Ing da COPPE-UFRJ

Prof. Jauvane Cavalcante de Oliveira - PhD. do LNCC/IME

Prof. Edison Ishikawa - D.Sc do IME

Rio de Janeiro
2005

”Acreditar em tudo e não acreditar em nada são duas posturas igualmente cômodas que nos dispensam da árdua tarefa de pensar” - Voltaire (1694-1778).

À minha esposa Camila Frutuoso Chagas.

AGRADECIMENTOS

À Camila, minha adorável esposa, pelo carinho, dedicação, respeito, sinceridade, confiança, por sempre está ao meu lado e principalmente pela disponibilidade.

Ao Professor Paulo Fernando Ferreira Rosa, por ter me aceitado em sua pesquisa, pelo incentivo, pela orientação e principalmente pelo tempo dedicado a mim.

Aos Professores do Departamento de Sistemas e Computação do IME, pela contribuição no meu crescimento acadêmico.

À CAPES por ter financiado parcialmente a pesquisa.

Ao Tenente Sandro S. Lima, por seu indispensável conhecimento de eletrônica e principalmente pelo tempo dedicado ao projeto.

Aos meus Pais Carlos e Maria da Penha e aos meus irmãos Marcelo, Adriana e Janieli pela confiança.

Ao Wagner Tanaka Botelho, pela disponibilidade.

Ao aluno da graduação em Engenharia de Computação do IME Antônio Fernando Rocha, por sua ajuda na construção do protótipo do simulador.

Ao Coronel Nei Bruno e ao Capitão Antônio Eduardo Carrilho, pelo interesse e disposição em ajudar no trabalho.

Ao Evandro Carvalho, por sua ajuda na elaboração gráfica do simulador.

Ao Ronaldo Lanna, por sua contribuição.

Ao José Renato de Alvarenga Bravo pelo apoio, no decorrer do trabalho.

Aos Colegas de trabalho Sra. Graziella Brenner, Rita Rollin, Regina Navarro, Elita Pereira, Jorge de Souza C. Filho, Mônica Loisse, Marcos Viana e Jorge Vallim pelo interesse no trabalho.

Aos professores do departamento de Informática da UFV pela contribuição quando no decorrer de minha graduação.

E a todos que de alguma forma contribuíram para a execução deste trabalho.

SUMÁRIO

LISTA DE ILUSTRAÇÕES	10
LISTA DE TABELAS	14
LISTA DE ABREVIATURAS E SÍMBOLOS	15
1 INTRODUÇÃO GERAL	19
1.1 Contexto e motivação	19
1.2 Objetivos da dissertação	19
1.3 Trabalhos realizados - organização do documento	19
2 TÓPICOS TUTORIAIS SOBRE TELEMANIPULAÇÃO	21
2.1 Introdução	21
2.2 Manipulador Robótico Mestre-Escravo	21
2.3 Teleoperação	22
2.4 Percepção Háptica	23
2.5 Realidade Virtual	23
2.6 Telemedicina	25
2.7 Conclusão	26
3 REVISÃO DE LITERATURA	27
3.1 Sistemas de telemanipulação robótica	27
3.2 Sistemas de Percepção Háptica	29
3.2.1 A importância dos sensores	29
3.3 Sistemas de telemanipulação e a Internet	30
3.4 Conclusão	31
4 TRABALHOS PRECURSORES	32
4.1 Manipulador Virtual Teleoperado para 1 GDL	32
4.1.1 Estudo e definição do problema	32
4.1.2 O protocolo de comunicação	33
4.1.3 Implementação física do sistema	34
4.2 Manipulador do tipo Mestre-Escravo 3 GDL com realimentação háptica ...	36
4.2.1 Definição do problema	37

4.2.2	A comunicação do sistema	37
4.2.3	Simulação e validação do problema	38
5	PROJETO DO MANIPULADOR ROBÓTICO	42
5.1	Introdução	42
5.2	Modelo do manipulador robótico	42
5.3	Equações cinemáticas do manipulador	43
5.3.1	Transformações homogêneas	43
5.3.2	Solução das equações cinemáticas para o manipulador	47
5.4	O Jacobiano do manipulador	51
5.5	Equações Dinâmicas do manipulador.....	51
5.6	Conclusão	57
6	DEFINIÇÃO DA ARQUITETURA DE CONTROLE	58
6.1	Introdução	58
6.2	Modelo Mecânico	58
6.3	Arquitetura de Controle.....	63
6.4	Conclusão	72
7	COMUNICAÇÃO ENTRE OS SISTEMAS MESTRE-ESCRAVO	73
7.1	Introdução	73
7.2	Modelo de referência TCP/IP	74
7.2.1	Principais funcionalidades das camadas do modelo TCP/IP	74
7.3	User Datagram Protocol - UDP	76
7.3.1	Características do protocolo UDP	77
7.4	Transmission Control Protocol - TCP	78
7.4.1	Estabelecimento da conexão	80
7.4.2	Encerramento da conexão	81
7.4.3	Características do TCP	82
7.5	Definição do protocolo de comunicação do sistema mestre-escravo	83
7.6	Conclusão	86
8	DESENVOLVIMENTO DO PROTÓTIPO FÍSICO E DO SIMU- LADOR DO SISTEMA	88
8.1	Protótipo do sistema	88
8.1.1	Especificação do dispositivo de captura/escrita de dados	90

8.1.2	Funcionamento da Placa de captura/escrita de dados	90
8.1.2.1	Entrada Analógica	91
8.1.2.2	Saída Analógica	91
8.1.3	Detalhes da pinagem da porta paralela	93
8.2	Modelo gráfico para o sistema mestre-escravo	96
8.2.1	Definição da linguagem de programação	96
8.2.2	Escolha da plataforma para implementação	99
8.2.3	Conclusão	101
9	RESULTADOS EXPERIMENTAIS	103
9.1	Análise de Desempenho e Estabilidade	103
9.1.1	Diagrama de Bode	103
9.1.2	Teste da arquitetura de controle	105
9.2	Simulação do dispositivo físico	109
9.3	Conclusão	111
10	COMENTÁRIOS GERAIS	113
11	REFERÊNCIAS BIBLIOGRÁFICAS	116
12	APÊNDICES	120
12.1	APÊNDICE 1: O Jacobiano do Manipulador	121
12.2	APÊNDICE 2: Lagrangiano	122
12.3	APÊNDICE 3: Variáveis de Estados	124
12.4	APÊNDICE 4: Esquema das Ligações Elétricas do MVT	126
12.5	APÊNDICE 5: Valores para o Diagrama de Blocos	127
12.6	APÊNDICE 6: Cálculo da Função de Transferência para os blocos em contato	128

LISTA DE ILUSTRAÇÕES

FIG.2.1	Esquema do funcionamento de um sistema de teleoperação do tipo mestre-escravo que permite a interação do operador humano com um ambiente remoto.	22
FIG.3.1	Diagrama de blocos representando um sistema de teleoperação.	28
FIG.3.2	Representação de um sistema de teleoperação utilizando a Internet como meio de comunicação.	30
FIG.4.1	Diagrama de blocos representando o esquema do manipulador proposto, onde T_H é torque de entrada, T_D é um ruído, T_E é o torque sentido no manipulador escravo, θ_M e θ_E são os ângulos medidos no mestre e escravo, respectivamente.	33
FIG.4.2	Representação dos motores com seus respectivos momentos de inércia (J_M e J_C e fator de redução das engrenagens ($N : 1$)).	34
FIG.4.3	Diagrama de blocos para o manipulador em termos de função de transferência (FERNANDES, 2002).	34
FIG.4.4	Esboço mecânico do manipulador mestre, onde o simulador pode ser observado sob uma vista lateral e uma frontal (FERNANDES, 2002).	36
FIG.4.5	Esboço mecânico do manipulador escravo, onde o simulador pode ser observado sob uma vista lateral e uma frontal (FERNANDES, 2002).	36
FIG.4.6	(a) Esboço do manipulador de 3 GDL. (b) Representação esquemática do manipulador, onde os cilindros representam os elos de rotação (CORREIA, 2003).	37
FIG.4.7	Diagrama de sequência caracterizando o protocolo de comunicação. (a) Transmissão da informação sem a presença de obstáculos. (b) Simulação da ocorrência de obstáculos na transmissão (CORREIA, 2003).	39
FIG.4.8	Sequência representativa de uma das baterias de testes executadas. (a) Visualização do mestre em movimento. (b) Visualização do escravo executando os comandos enviados pelo mestre (CORREIA, 2003).	40

FIG.5.1	Esquema do manipulador proposto.	43
FIG.5.2	Esquema do manipulador no R^3	44
FIG.5.3	Modelo de uma garra com a indicação do sistema de coordenadas.	45
FIG.6.1	Modelo mecânico, que simula um braço robótico.	58
FIG.6.2	Modelo mecânico, simula um sistema de teleoperação para 1 GDL.	59
FIG.6.3	Diagrama de blocos para o modelo mecânico da FIG. 6.2.	60
FIG.6.4	Modelo mecânico representando uma situação na qual o mestre está diretamente em contato com o escravo.	61
FIG.6.5	Representação de um sistema de teleoperação para 1 GDL, com as respectivas forças que atuam no mesmo.	62
FIG.6.6	Diagrama de blocos do modelo mecânico apresentado na FIG. 6.5.	63
FIG.6.7	Pseudo diagrama de blocos que representa a arquitetura de controle para um sistema de teleoperação de 1 GDL.	65
FIG.6.8	Modelo de duas portas, i_1 e i_2 são fontes de corrente, v_1 e v_2 são fontes de tensão.	66
FIG.6.9	Representação de um sistema bilateral de teleoperação através um modelo de duas portas.	66
FIG.6.10	Diagrama de blocos da FIG. 6.7 com os respectivos valores das funções de transferência.	71
FIG.7.1	Representação da comunicação entre as camadas. Observa-se que a atuação dos protocolos ocorrem em mesmo nível.	73
FIG.7.2	Modelo de referência TCP/IP.	75
FIG.7.3	Formato do segmento UDP.	76
FIG.7.4	Formato do segmento TCP.	78
FIG.7.5	Representação do estabelecimento de uma conexão TCP entre duas estações.	81
FIG.7.6	Descrição do encerramento de uma conexão TCP entre duas es- tações. As setas indicam a sequência em que o processo ocorre.	82
FIG.7.7	Composição do sistema: fluxo da informação.	85
FIG.7.8	Fluxograma representando a comunicação entre os sistemas mestre- escravo.	85
FIG.8.1	Representação em 3D para o protótipo real de 1 GDL.	88
FIG.8.2	Modelo real do sistema desenvolvido para 1 GDL.	89

FIG.8.3	Pseudo diagrama representando as ligações entre os sistemas mestre e escravo e a comunicação entre ambos.	89
FIG.8.4	Fluxograma detalhando o caminho percorrido pelo sinal nos sistemas mestre e escravo.	92
FIG.8.5	Esquema do circuito elétrico da placa desenvolvida.	93
FIG.8.6	Esquema da pinagem do dispositivo (DB-25) da porta paralela.	93
FIG.8.7	Curva de representação da palavra (eixo x) para uma determinada tensão (eixo y).	94
FIG.8.8	Ambiente de simulação real, consistindo de duas placas de leitura e escrita de dados conectadas a dois PC(s) conectados à Internet com uma taxa de transmissão local de 100Mb/s.	95
FIG.8.9	Exemplo da syntax de um documento VRML.	97
FIG.8.10	Código da página HTML.	98
FIG.8.11	Interligação da linguagem JAVA com VRML utilizando EAI.	98
FIG.8.12	Manipulador 3 GDL executado a partir de um browser, utilizando o plugin Cortona	99
FIG.8.13	manipulador 3 GDL em detalhes: (A) movimento de rotação à direita (B) rotação à esquerda do braço superior; (C) movimento de rotação à direita; (D) rotação à esquerda do braço inferior. Em (E) e (F) observa-se o movimento de translação.	100
FIG.8.14	Visualização gráfica do manipulador mestre utilizando o Java 1.4.2, o Conectiva Linux 9 e o navegador Mozilla	101
FIG.8.15	Visualização gráfica do manipulador mestre, executando um movimento de rotação do braço superior, utilizando o Java 1.4.2, o Conectiva Linux 9 e o navegador Mozilla	102
FIG.9.1	Diagrama de Bode, (magnitude) x (tempo).	104
FIG.9.2	Diagrama de Bode, (fase) x (tempo).	104
FIG.9.3	Diagrama de Bode, magnitude e fase mostradas em um só gráfico.	105
FIG.9.4	Resposta do sistema à função degrau.	106
FIG.9.5	Resposta do sistema à função impulso.	106
FIG.9.6	Alimentação do sistema através da função seno. O eixo das abcissas fornecem a informação sobre o tempo e o eixo das ordenadas fornecem a informação sobre a amplitude.	107

FIG.9.7	Resposta do sistema à entrada feita com a função seno. Resposta obtida no sistema escravo.	107
FIG.9.8	Alimentação do sistema através da função de números aleatórios.	108
FIG.9.9	Resposta do sistema à função de números aleatórios. Resposta obtida no sistema escravo.	108
FIG.9.10	Curva de calibração. Os pontos representam a resposta do escravo para uma entrada de tensão no mestre. A reta contínua representa a resposta ideal para o sistema.	110
FIG.9.11	Gráfico representando o fluxo de dados do sistema; o eixo das abscissas mostra o byte analisado e o eixo das ordenadas mostra o tempo gasto para transmití-lo e recebê-lo.	111
FIG.12.1	Sistema mola-massa-amortecedor.	124
FIG.12.2	Representação da caracterização de um sistema de controle no qual $u_i(t)$ são as variáveis de entrada em função do tempo e $y_i(t)$ são as variáveis de saída.	124
FIG.12.3	Esquema das ligações elétricas do Manipulador Virtual Teleoperado - MVT.	126
FIG.12.4	Diagrama de blocos para o modelo mecânico da FIG. 6.4	128
FIG.12.5	Diagrama de blocos rearranjado a partir da FIG. 12.4	128
FIG.12.6	Representação do laço inferior do diagrama da FIG. 12.5	129
FIG.12.7	Diagrama de blocos modificado, resultado da eliminação do laço inferior da FIG. 12.5	129
FIG.12.8	Laço com retroação, extraído da FIG. 12.7.	129
FIG.12.9	Diagrama de blocos representando a EQ. 12.40.	130
FIG.12.10	Diagrama de blocos final para a função de transferência dada pela EQ. 12.41.	130

LISTA DE TABELAS

TAB.5.1	Parametros de DENAVIT-HARTEMBERG	46
---------	--	----

LISTA DE ABREVIATURAS E SÍMBOLOS

S_i -	seno(θ_i)
C_i -	cosseno(θ_i)
sen -	seno
cos -	cosseno
Tan -	tangente
$ArcTan$ -	arco tangente

SIGLAS

CAS -	Computer Assisted Surgery
CGI -	Common Gateway Interface
DHCP -	Dynamic Host Configuration Protocol
DNS -	Domain Name System
EAI -	External Authoring Interface
FTP -	File Transport Protocol
GDL -	Graus de Liberdade
HIC -	Hybrid Impedance Control
HTTP -	HyperText Transfer Protocol
IA -	Inteligencia Artificial
IANA -	Internet Assigned Numbers Authority
IMAP -	Internet Message Access Protocol
IP -	Internet Protocol
LAN -	Local Area Network
MIS -	Minimally Invasive Surgery
MSS -	Maximum Segment Size
MVT -	Manipulador Virtual Teleoperado
NFS -	Network File System
PC -	Personal Computer
ROV -	Remotely Operator Vehicle
SMTP -	Simple Mail Transfer Protocol
SNMP -	System network management protocol

VRML - Virtual Reality Modeling Language

RESUMO

Esta dissertação trata do desenvolvimento de um sistema de telemanipulação com realimentação háptica, visto que este tema tem despertado grande interesse na comunidade científica nas últimas décadas devido à sua interdisciplinaridade e à sua vasta gama de aplicações.

O desenvolvimento do sistema foi elaborado em várias etapas a saber: (i) a primeira, foi a elaboração do projeto do manipulador que consistiu na obtenção das equações cinemáticas e dinâmicas; (ii) a segunda, consistiu na elaboração da arquitetura de controle para o sistema; (iii) a terceira, foi a elaboração de um protocolo de comunicação desenvolvido utilizando-se programação de sockets; (iv) a quarta, consistiu na elaboração de um simulador que foi implementado na linguagem JAVA combinada com VRML (Virtual Reality Modeling Language) através da Java EAI (External Authoring Interface); (v) a quinta, foi o desenvolvimento de uma placa de captura e escrita de dados, utilizada como interface para interligar dois manipuladores reais de um grau de liberdade à fim de validar o sistema robótico proposto.

ABSTRACT

This thesis deals with the development of a telemanipulation robotic system with haptic feedback. This topic has been of great interest in the scientific robotics community due to its interdisciplinary nature and wide range of applications.

The development of the system was carried out in several stages, namely: (i) the elaboration of the manipulator's project, it consisted of the obtaining kinematics and dynamics equations; (ii) the elaboration of the control architecture system; (iii) the elaboration of a communication protocol that was developed using socket programming; (iv) the elaboration of a simulator that was implemented in the JAVA language in conjunction with VRML (Virtual Reality Modeling Language) and Java EAI (External Authoring Interface); (v) the hardware implementation to validate the overall system.

1 INTRODUÇÃO GERAL

1.1 CONTEXTO E MOTIVAÇÃO

Existem situações em que a intervenção humana é limitada por fatores tais como condições ambientais, incapacidade física, necessidade de alta precisão e longas distâncias, dentre outros aspectos. A intervenção humana fica limitada quando fatores ambientais apresentam situações que põem em risco a integridade física ou até mesmo a vida das pessoas. A incapacidade física pode ser relacionada a situações em que seja necessário ampliar a força humana a uma escala bem superior ao convencional; por exemplo, máquinas de ponto de solda utilizadas na indústria automobilística podem pesar até o equivalente a três pessoas adultas, o que inviabiliza a sua manipulação por apenas um único operador. Existem também situações que envolvem precisão, como a perfuração de um crânio para realizar uma cirurgia no cérebro, ou quando um especialista necessita realizar tarefas em um sistema localizado remotamente. Dentro deste contexto, a utilização de sistemas robóticos surge como uma solução e é isto que motiva o desenvolvimento e/ou aperfeiçoamento de sistemas que visam a auxiliar o homem, para que este possa superar seus limites.

1.2 OBJETIVOS DA DISSERTAÇÃO

Este trabalho tem como principal objetivo desenvolver um sistema de teleoperação do tipo mestre-escravo, que seja capaz de detectar uma colisão em um ambiente desconhecido e transmiti-la a quem o estiver operando pela Internet. Para tal, será necessário determinar o tipo de controle a ser utilizado; elaborar o projeto do manipulador robótico; desenvolver uma implementação via software para validar o sistema (manipulador mestre, manipulador escravo e comunicação entre ambos) e, para finalizar, construir fisicamente o sistema, avaliando-o experimentalmente.

1.3 TRABALHOS REALIZADOS - ORGANIZAÇÃO DO DOCUMENTO

Esta dissertação está organizada de forma a permitir um maior entendimento do assunto a ser tratado. Inicialmente foi feita uma breve introdução com a motivação desta pesquisa; o capítulo dois apresenta os assuntos sobre manipulador robótico mestre-

escravo, teleoperação, percepção háptica, realidade virtual e telemedicina. O capítulo três trata da manipulação bilateral com realimentação háptica, manipulação à distância, sensores e utilização da internet na teleoperação. No capítulo quatro é mostrado resumidamente dois outros trabalhos que serviram de base para este. No capítulo cinco é definido o projeto do manipulador robótico assim como as equações cinemáticas e dinâmicas. No capítulo seis é mostrada a definição da arquitetura de controle. O capítulo sete trata da comunicação entre os sistemas mestre e escravo. O capítulo oito mostra o desenvolvimento do protótipo físico e do simulador gráfico para o sistema mestre-escravo. O capítulo nove mostra os resultados experimentais. E, para finalizar, chega-se aos comentários finais sobre todo o sistema.

2 TÓPICOS TUTORIAIS SOBRE TELEMANIPULAÇÃO

Neste capítulo, serão tratados assuntos relacionados a sistemas robóticos do tipo mestre-escravo, teleoperação, percepção háptica, realidade virtual e telemedicina. E, ao final do capítulo, será apresentada uma conclusão do mesmo.

2.1 INTRODUÇÃO

Sistemas de manipulação mestre-escravo surgem da necessidade do ser humano de executar tarefas, ou realizar trabalhos, que até então não seriam factíveis pelo mesmo, como: limitações física, precisão, estar presente em ambientes insalubres, etc. Em suma, o homem necessita ganhar tempo, necessita de maior precisão em suas tarefas e, principalmente, não quer correr riscos que venham a ser prejudiciais à sua integridade física. Dessa forma, o desenvolvimento de sistemas que atendam como um todo, ou apenas em parte essas necessidades, são justificáveis.

2.2 MANIPULADOR ROBÓTICO MESTRE-ESCRAVO

A utilização de sistemas do tipo mestre-escravo surge da necessidade de se trabalhar com manipuladores robóticos sem que estes tenham uma configuração pré-definida¹, ou seja, neste caso existe a necessidade de se ter um controle humano sobre o robô e não deixar que este seja feito via software. Em princípio, a idéia parece não ser palpável, visto que o manipulador foi construído para "substituir o homem" e agora este necessita intervir diretamente naquele. E ainda existe o problema da ação direta do ser humano junto à máquina, pois, às vezes, esta encontra-se em ambientes hostis. Além disso existe o problema de escala (o robô ser muito maior ou muito menor que o operador). No entanto, uma boa justificativa para a intervenção direta do homem é que existem casos em que o poder de decisão humano é indispensável; como por exemplo, a execução de uma cirurgia feita com a intervenção de um robô. Devido a fatores como este, surge a idéia de trabalhar-se com um sistema de manipulação do tipo mestre-escravo.

Um sistema do tipo mestre-escravo pode ser descrito como a seguir: suponha-se um

¹Entende-se aqui como configuração pré-definida um programa que foi embarcado no manipulador, o que faz com que este execute somente as instruções programadas.

cenário onde exista um ser humano, um robô cuja função é fazer a soldagem de peças e que seja necessário realizar uma solda em uma tubulação localizada a uma grande profundidade no oceano - onde a alta pressão inviabiliza a presença humana. Diante deste cenário, o que pode ser feito? Uma solução é construir dois sistemas robóticos, com um deles programado para realizar a tarefa diretamente; este sistema é chamado de sistema escravo. O outro sistema é o que permitiria a interface humana com o robô que está executando a tarefa e este será chamado de sistema mestre. Caberá ao mestre enviar sinais ao escravo para que este execute a tarefa em questão. A FIG. 2.1 ilustra o sistema descrito.

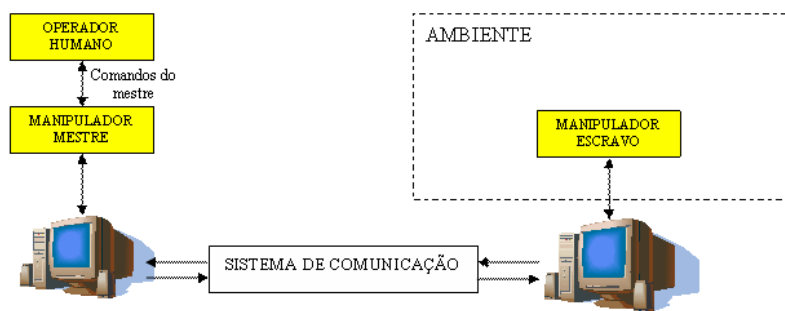


FIG 2.1: Esquema do funcionamento de um sistema de teleoperação do tipo mestre-escravo que permite a interação do operador humano com um ambiente remoto.

2.3 TELEOPERAÇÃO

Sistemas de teleoperação como o próprio nome sugere, dizem respeito a um conjunto de dispositivos que permitem ser operados ou manipulados à distância. Pois, *tele* do radical grego significa *distância*. Geralmente, estes tipos de sistemas são utilizados em explorações oceanográficas ou geológicas em geral, aplicações militares e de resgate para a defesa civil, ocupações de plantas nucleares contaminadas, medicina e explorações espaciais, dentre outras.

Existem vários tipos de sistemas de teleoperação, dentre os quais podem ser citados: os veículos operados remotamente ou *ROVs* (LARKUM, 1994; CONTE, 1995); Sistemas como o Zeus (BUTNER, 2003; BALLANTYNE, 2002) que são utilizados em cirurgias robóticas. E, robôs como o *Sojourner*, que fez parte da missão "Mars Pathfinder" ao planeta Marte, (TEAM, 1997; MISHKIN, 1998) e consistia em um teste, onde o *Sojourner* era controlado a partir de fotos e sinais enviados por ele mesmo para a terra; devido ao grande retardo ele também possuía um sistema autônomo de controle.

Outros inúmeros sistemas poderiam ser citados, mas acredita-se que os exemplos dados são bastante representativos no que se refere a Sistemas de Teleoperação.

2.4 PERCEPÇÃO HÁPTICA

Muitos trabalhos que tratam de sistemas de teleoperação fazem apenas referência à percepção de força isoladamente. Dessa forma, não levam em consideração outros parâmetros envolvidos, como a percepção de textura, temperatura e, inclusive, a sensação de dor. Em vista disto, o termo háptico pode ser definido como a capacidade de avaliar parâmetros como a aspereza, temperatura, e a força aplicada em um objeto, ou seja, é a análise de toda a capacidade tátil do ser humano.

A importância da percepção háptica para a teleoperação está na tentativa de simular da forma mais real possível a imersão do ser humano em um ambiente onde ele precisa atuar sem necessariamente estar presente (telepresença). Ou seja, a percepção háptica irá melhorar a eficiência do ser humano ao realizar uma tarefa remotamente, ou ainda dar-lhe-a maior satisfação quando se trata de entretenimento. Por exemplo, luvas que possuem sensores táteis, manipuladores utilizados em cirurgias, etc.

Muitos dispositivos que provêm a percepção tátil têm sido desenvolvidos e estudados, dentre os quais podem ser citados trabalhos como (ROSA, 2004a), onde se encontra o desenvolvimento de um dispositivo de telemanipulação com *1 GDL*, que fornece realimentação háptica e pode ser operado via web. Em (NIKI, 2000) é mostrado um dispositivo háptico simples que trabalha juntamente com uma base de dados que contém informações como temperatura e textura do objeto analisado. Em (ELHAJJ, 2001) é possível verificar a transmissão da informação háptica via Internet. Em (RICHARD, 1999) foi avaliada a habilidade da percepção háptica humana quando presente em ambientes virtuais; este trabalho resultou de uma parceria entre o *Man-Machine Interface Laboratory, NJ, USA* da *Rutgers University* com o laboratório de robótica de Paris (*CRIIF-LRP*). Outras pesquisas onde se trabalhou no desenvolvimento ou aperfeiçoamento de dispositivos para fornecer a habilidade de percepção háptica podem ser encontrados em (HUI, 1995; SITTI, 2003; HER, 2001).

2.5 REALIDADE VIRTUAL

Realidade Virtual, pode ser descrita como um ambiente de simulação interativo no qual o homem participa como uma peça da simulação. O ambiente, "geralmente" é uma

simulação gráfica que pode ser feita por computador, que pode ser um PC ou um super computador; a escolha do hardware depende da complexidade e finalidade da aplicação. O ser humano participa do cenário através de uma conexão feita por componentes físicos como capacetes, luvas, roupas, etc., ou pode ser introduzido através de uma estrutura mais sofisticada chamada *The Cave (Cave:Audio-Visual Experience/Automatic Environment)*. O Cave foi construído pelo *Electronic Visualization Laboratory* da Universidade de Illinois, e consiste em um cubo com suas paredes revestidas por telas que servem para projetar imagens, dando assim maior sensação de realidade ao usuário. Em resumo, pode-se dizer que Realidade Virtual é o processo em que o homem participa de um ambiente simulado num computador, sendo que este ambiente é capaz de interagir com o ser humano a ponto de parecer real (SCHALAKE, 1993).

Historicamente, o termo *Realidade Virtual*, é creditado a Jaron Lainer (ZHENG, 1998). Dentro da Realidade Virtual existem dois conceitos importantes que podem ser utilizados para um maior entendimento do assunto, são os conceitos de *presença* e *imersão*. O termo presença é utilizado no sentido de que o usuário sente-se parte do cenário em que está atuando. Já o termo imersão refere-se ao grau de complexidade do sistema que, além de fornecer estímulos visuais, também pode fornecer a sensação de movimento, cheiro, calor, medo, felicidade, estímulos auditivos, dentre outros.

A Realidade Virtual possui uma enorme gama de sistemas e processos em que pode ser utilizada, como por exemplo, na indústria do entretenimento, na medicina na educação, em treinamento, na engenharia e na indústria aeroespacial. A indústria do entretenimento na era atual talvez seja a mais representativa da utilização desta tecnologia; isto pode ser verificado nos inúmeros jogos de computador e no cinema que já há algum tempo a tem explorado, em filmes como *I Robot*, *Matrix*, *Minority Report*, *Jurassic Park*, *Final Fantasy*, dentre inúmeros outros.

Na medicina, a Realidade Virtual surge como um fator de aprimoramento dos processos cirúrgicos no qual o cirurgião de posse de um mundo virtual pode praticar a fim de melhorar o seu desempenho, quando exposto ao mundo real. Um sistema deste tipo vem sendo desenvolvido pelo Instituto Militar de Engenharia, onde um ambiente virtual colaborativo tridimensional não-imersivo é proposto com a finalidade de possibilitar a análise pré-operativa e o estudo de técnicas e planejamento do procedimento cirúrgico, auxiliando na identificação dos locais de intervenção, na visualização da anatomia local e na verificação de estruturas e pontos sensíveis envolvidos no procedimento, além de testar a habilidade de médicos cirurgiões e de estudantes da área (ALBERIO, 2004). Em se

tratando de processos industriais, uma simulação virtual pode melhorar a qualidade dos produtos, reduzir custos de fabricação e melhorar o desempenho dos operários. Como exemplo pode-se citar a Ford Motor Co. e a Chrysler Corp que conseguiram reduzir o tempo de desenvolvimento de seus produtos em doze meses através da utilização de um ambiente de desenvolvimento virtual (ZHENG, 1998). A aplicação de técnicas e métodos de computação gráfica na área da robótica é discutida em (BEJCZY, 1996), onde o trabalho consistiu em verificar como a utilização de um sistema virtual ajudaria a aumentar o desempenho de uma determinada tarefa.

Atualmente, devido às facilidades em adquirir processadores rápidos e placas de captura de imagens a um preço acessível, tornou esta área de pesquisa mais próxima do público, ou seja, dos desenvolvedores de sistemas; e isto é constatado no dia a dia, pois constantemente um jogo interativo é lançado no mercado, ou um novo filme que inova nos efeitos virtuais aparece. Em princípio, esta parece ser uma área onde ainda há muito para ser explorado, principalmente nos processos fabris, em robótica e medicina.

2.6 TELEMEDICINA

Em se tratando de telemedicina, faz-se necessário definir alguns termos como cirurgia minimamente invasiva ou MIS do inglês (*Minimally Invasive Surgery*), cirurgia laparoscópica e outras que podem ser resumida pela sigla CAS (*Computer Assisted Surgery*). Cirurgias minimamente invasivas são aquelas em que os instrumentos cirúrgicos são inseridos no corpo do paciente por pequenas incisões. Já uma cirurgia laparoscópica é uma modalidade das MIS, onde a cavidade abdominal é expandida por gases, então instrumentos especificamente desenhados para tal processo, são inseridos através de pequenas incisões feitas no abdome. As vantagens, deste tipo de cirurgia estão no menor trauma pós-cirúrgico, na recuperação dos pacientes fazer-se de forma mais rápida, o tempo de hospitalização ser menor o que reduz o perigo de infecção hospitalar, com a conseqüente diminuição dos custos.

Apesar de todas as vantagens citadas, este tipo de cirurgia possui fatores agravantes como o cansaço, stress e perda de sensibilidade de tocar os órgãos por parte do cirurgião. Essas desvantagens podem ser explicadas devido ao restrito espaço de trabalho, pois a incisão é mínima, e ao formato dos instrumentos cirúrgicos. No sentido de solucionar ou melhorar estes agravantes, pesquisas vêm sendo desenvolvidas; e dentre estas pode-se citar (ÇAVUŞOĞLU, 1999; BICCHI, 1996; YAMASHITA, 2003).

Existem dois sistemas muito importantes no que se refere à telecirurgia; o sistema

”da Vinci” e o sistema Zeus. O sistema da Vinci surgiu como uma generalização de um outro sistema que foi desenvolvido pelo Instituto de Pesquisa de Stanford (SRI) (STOIANOVICI, 2000). O sistema original incluía alimentação de áudio e vídeo em três dimensões, dois manipuladores teleoperados, ambos com 7 GDL. O sistema da Vinci compõe-se de dois componentes principais, uma interface de controle para o cirurgião e uma unidade cirúrgica, que é um braço capaz de manipular os instrumentos cirúrgicos. O sistema vem operando desde janeiro de 2000 e tem sido utilizado pelos centros cardíacos europeus com a função endoscópica. O sistema Zeus, também foi projetado para realizar procedimentos cirúrgicos laparoscópicos, e vem sendo utilizado com sucesso. Um exemplo bem sucedido da utilização desse sistema pode ser visto em (MARESCAUX, 2001), onde 25 pacientes com sintomas de cholecystiasis (pedra na vesícula) foram submetidos a uma experiência clínica. Em média o tempo de duração da cirurgia era de 25 a 109 minutos. Um engenheiro participou de todo o procedimento, pois sua presença fazia-se necessário para suprir eventuais problemas técnicos que pudessem ocorrer. Pode-se dizer que a experiência foi bem sucedida, pois 24 dos 25 pacientes tiveram boa recuperação sendo que apenas um deles apresentou maiores problemas. Em (BUTNER, 2003), foi feita uma alteração no sistema Zeus, para que este se tornasse capaz de executar uma telecirurgia humana a longas distâncias. Como teste de prova foi executada uma cirurgia transatlântica denominada *Operação Lindbergh*. A operação foi realizada em 7 de setembro de 2001, em uma paciente de 68 anos em Strasbourg na França com os cirurgiões localizados em Nova York. A cirurgia durou 54 minutos e consistiu de uma colecistectomia²; após dois dias, a paciente pôde deixar o hospital.

2.7 CONCLUSÃO

Neste capítulo, foram feitas algumas apresentações a respeito de como funcionam um sistema mestre-escravo, o que é um sistema de teleoperação; também foi feita uma apresentação do termo percepção háptica e, para finalizar, foi feita uma prévia a respeito de telemedicina que é uma área da robótica que tem despertado grande interesse de diversos pesquisadores nos últimos tempos.

²Colecistectomia é a cirurgia para a retirada da vesícula biliar.

3 REVISÃO DE LITERATURA

Motivadas pela necessidade de estar presente em ambientes que trazem riscos para a vida do ser humano, várias pesquisas vêm sendo desenvolvidas sobre sistemas robóticos de teleoperação. Algumas dessas pesquisas serviram de suporte para o desenvolvimento deste trabalho e neste capítulo elas serão citadas de forma a permitir ao leitor descobrir mais detalhes a respeito dos sistemas de tele-robótica e percepção háptica que são o enfoque principal desta pesquisa.

3.1 SISTEMAS DE TELEMANIPULAÇÃO ROBÓTICA

Um sistema de teleoperação pode ser representado por um diagrama de blocos, composto de três sub-blocos [vide FIG. 3.1], onde o bloco da esquerda representa o subsistema mestre, o bloco do meio representa o subsistema de comunicação e o bloco mais à direita representa o subsistema escravo. F e V são as variáveis força e velocidade que são entradas para o sistema; estas são propagadas pelo sistema tornando-se F' e V' até atingirem o escravo que fica em contato com o ambiente externo. Após receber a informação do mestre, o escravo a executa e a retransmite para o mestre. Em um sistema ideal de teleoperação, a força de ação no ambiente é sentida integralmente pelo operador. Sistemas como o apresentado pela FIG. 3.1 são chamados de bilaterais, pois a informação flui em duas direções.

(HANNAFORD, 1989) e (LAWRENCE, 1992) trabalharam com sistemas bilaterais de teleoperação implementados juntamente com controle híbrido. Nestes trabalhos, encontra-se uma boa representação da teoria de controle híbrido, pois ambos propõem que um sistema de teleoperação seja modelado através de uma rede de duas portas. Este trabalho também propõe o modelo de um sistema de teleoperação através de uma rede de duas portas, pois isto permite trabalhar com força e velocidade, diferindo assim dos controles clássicos de posição em que apenas um parâmetro é avaliado.

Em (LAWRENCE, 1992) é proposta uma arquitetura de teleoperação baseada em quatro canais de comunicação, cujo objetivo é aumentar a "transparência"³ sobre um controle de posição. Em (ZHU, 1995), a proposta de (LAWRENCE, 1992) é utilizada para mostrar que uma perfeita transparência pode ser alcançada por um sistema de

³veja seção 6.3.

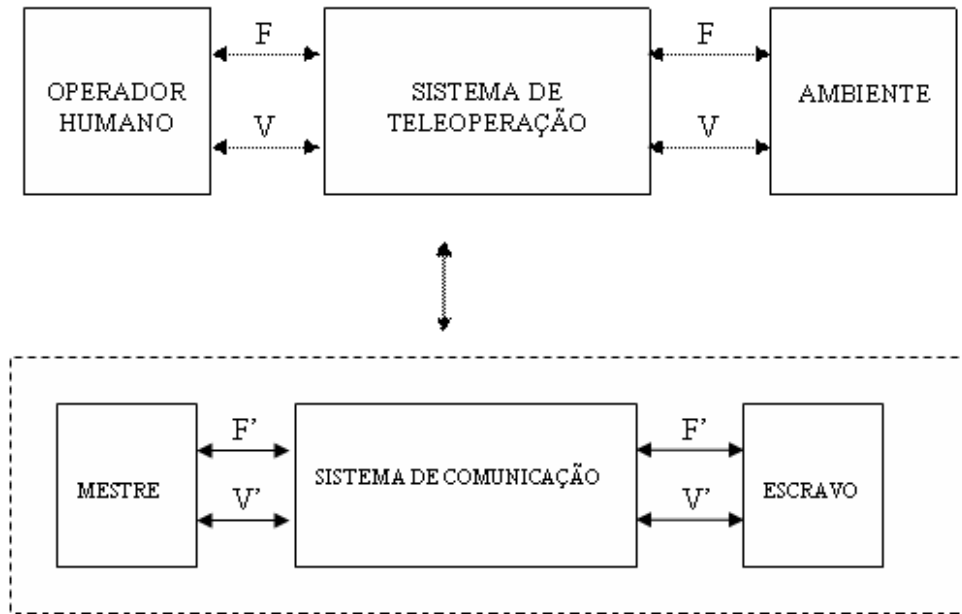


FIG 3.1: Diagrama de blocos representando um sistema de teleoperação.

teleoperação mas não necessariamente sobre um modelo baseado em controle de posição, mas sim através de um modelo que combina velocidade e posição.

(ANDERSON, 1988) propôs um tipo de arquitetura de controle chamada HIC, do inglês "Hybrid Impedance Control" que consiste em um híbrido entre o controle de impedância e o controle híbrido. Para testar essa proposta foi utilizado um manipulador simples composto de dois elos. A conclusão a que eles chegaram a respeito da arquitetura foi que apesar de ser uma arquitetura geral e de obterem bons resultados algumas pendências permaneceram. Existe também dificuldade em modelar uma função de aproximação para a impedância humana devido às variações; a programação se torna mais difícil visto que é difícil entender a execução de uma tarefa em termos das forças aplicadas à tarefa. Sendo assim, eles chegam a sugerir alguma técnica de IA para a determinação destas forças. E além de tudo existem os problemas de ruídos, atrito estático e restrição de torque que degradam o desempenho do sistema de forma a não torná-lo ideal.

Para este trabalho é proposta uma arquitetura baseada em controle híbrido; a opção de utilizar este tipo de controle foi baseada em diversos trabalhos existentes na área como (LAWRENCE, 1992; HANNAFORD, 1989; ADAMS, 1998; ÇAVUŞOĞLU, 2001; SHERMAN, 2000), onde este tipo de arquitetura tem-se mostrado uma boa solução para sistemas que envolvem precisão. Sua principal característica está no fato de levar em

consideração dois tipos de parâmetros diferentes, que são força e posição. A utilização destes dois tipos de parâmetros permite aumentar a eficiência do mecanismo controlado.

3.2 SISTEMAS DE PERCEPÇÃO HÁPTICA

Na seção 2.4 foi feita uma introdução a respeito da percepção háptica e alguns trabalhos foram citados com o intuito de familiarizar o termo. Este item trata dos sensores, pois eles são de suma importância para os sistemas hápticos, visto que são os dispositivos responsáveis por simular as sensações. Para efeito de ilustração serão mencionadas pesquisas bastante relevantes sobre o assunto.

3.2.1 A IMPORTÂNCIA DOS SENSORES

Em princípio, pensar em percepção háptica, quase de imediato nos vem à mente o termo sensor. Realmente vários trabalhos buscam o desenvolvimento destes dispositivos de forma que eles atendam perfeitamente à necessidade de cada aplicação. Bons trabalhos podem ser encontrados na literatura, dentre os quais podem ser citados (ROSA, 1995; VOYLES, 1996; BICCHI, 1996; MASCARO, 1999; FASSE, 1994).

Em (ROSA, 1995) foi proposto o desenvolvimento de um dispositivo denominado SCROLLIC, que consiste em duas garras paralelas com aderência, que têm como objetivo fornecer múltiplos contatos e permitir a manipulação de objetos de formas variadas desde que suas dimensões estejam dentro do espaço de trabalho da garra. Em (VOYLES, 1996) foi desenvolvido um sensor tátil modular e um sistema de atuador para observar demonstrações de tarefas de contato. O sistema consiste em três partes: um sensor tátil intrínseco para medir força líquida/torque, um sensor tátil extrínseco para medir distribuições de contato, e um atuador tátil para exibir distribuições táteis. Os componentes modernos são o sensor extrínseco e o atuador tátil que é "simétrico por dentro e por fora" para um ou outro e emprega um gel eletroreológico para atuação. A maior vantagem de usar o gel eletroreológico é que ele pode ser controlado eletricamente; o que é conveniente, pois requer pequena carga embora as voltagens sejam muito altas, e ele possa ser feito de uma forma bem compacta. (MASCARO, 1999) trabalhou no desenvolvimento de um sensor para detectar força e toque, quando o operador tem a intenção de pressionar botões ou interruptores. Este sensor permite que dispositivos físicos normais como interruptores presos em paredes sejam substituídos por "dispositivos virtuais", que contêm circuitos elétricos e partes mecânicas, mas que são apenas desenhos.

O desenvolvimento de dispositivos relacionados à simulação, transmissão de sensações

em ambientes virtuais também é muito comum já que tais trabalhos transmitem a necessidade de medir o quão real seja o ambiente simulado. Devido a este fato, (FASSE, 1994) em seu trabalho propôs um modelo teórico com o objetivo de medir quantitativamente dispositivos de percepção háptica embebidos em ambientes virtuais.

3.3 SISTEMAS DE TELEMANIPULAÇÃO E A INTERNET

Com o decorrer dos anos, a Internet tem se tornado a grande aliada para as mais variadas tarefas, pois o que inicialmente tinha caráter apenas científico popularizou-se e, atualmente, o seu domínio vai desde uma simples consulta de texto até o controle de um dispositivo sofisticado. Com o aperfeiçoamento dos meios de transmissão, ela tem sido testada como auxílio para os sistemas de teleoperação, pois possui várias características que vêm simplificar o projeto dos sistemas de telemanipulação. A FIG. 3.2 mostra um esquema representando um sistema de teleoperação comunicando-se através da Internet.

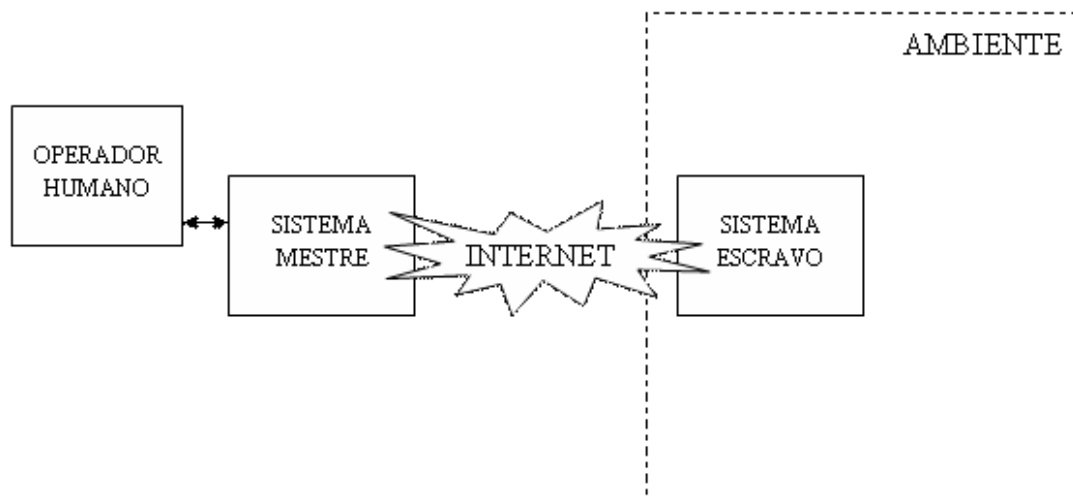


FIG 3.2: Representação de um sistema de teleoperação utilizando a Internet como meio de comunicação.

A primeira demonstração de um sistema de telemanipulação via Web data de setembro de 1994, onde um robô ASEA IRb-6 foi conectado à Internet através de um servidor Web, na Universidade Western na Austrália. Quatro semanas após, Ken Goldberg, na Universidade da Califórnia, em Berkeley, conectou um robô do tipo SCARA via Internet (TAYLOR, 2000).

OBOE (1999) propôs um sistema de teleoperação baseado no acesso remoto de um

robô comandado pela Internet com a linguagem JAVA, utilizando servlet⁴ ao invés de CGI. Tal como o trabalho de Oboe, este trabalho também propõe-se a trabalhar com a linguagem JAVA, visto que ela tem se demonstrado como uma boa ferramenta; principalmente pelo fato da interoperabilidade com várias plataformas diferentes.

MALINOWSKI (1999) descreve o controle de um robô via Internet utilizando um circuito simples de microcontrolador utilizando os protocolos TCP/IP. Em seu trabalho, Malinowski faz a utilização de sensores e câmeras de vídeo para fornecer a realimentação para o operador. Vale citar também que a utilização de câmeras de vídeo teleoperadas tem sido de grande utilidade em sistemas de vigilância via Internet. Para o presente trabalho não é feita a utilização de câmeras de vídeo para realimentação visual, pois a principal prioridade é fornecer uma realimentação háptica da forma mais precisa possível e a alimentação visual neste caso poderia degradar o desempenho do sistema, visto que a transmissão de vídeo em tempo-real não é trivial.

A teleoperação de dispositivos robóticos via Web tem sido um campo de grande interesse em robótica, pois o aperfeiçoamento dos meios de transmissão (ex. fibras óticas) tem motivado o desenvolvimento destes dispositivos; além disso, este tipo de dispositivo é uma solução econômica, o que permite beneficiar um enorme número de usuários, como a realimentação tátil, além de áudio e vídeo.

3.4 CONCLUSÃO

Diferente do capítulo 2, este capítulo teve como principal objetivo promover o entendimento deste trabalho no sentido de apresentar subsídios para que outros trabalhos sirvam como termo de comparação. Assim, foram examinados vários outros tentando sempre compará-los a este de forma a melhorá-lo e descobrir aspectos que não foram explorados ou qualidades que este possa adicionar aos outros, quando for o caso.

⁴Servlet é um programa que reside em um servidor capaz de responder requisições vindas de um programa JAVA hospedado em um cliente; diferente do CGI, a conexão com o cliente fica sempre aberta.

4 TRABALHOS PRECURSORES

Neste capítulo, serão mostrados resumidamente dois outros trabalhos que antecederam e servirão de base para este. Inicialmente, será mostrado o trabalho desenvolvido por (FERNANDES, 2002), onde foi desenvolvido um protótipo para um manipulador de 1 GDL. Em seguida, será exibido o trabalho desenvolvido por (CORREIA, 2003), onde foi desenvolvido um protótipo para um simulador de 3 GDL; este sistema foi proposto para ser operado através de uma LAN "Local Area Network".

4.1 MANIPULADOR VIRTUAL TELEOPERADO PARA 1 GDL

O trabalho de FERNANDES (2002) consistiu em 3 tópicos principais que podem ser definidos a seguir: (1) estudo e definição do problema; (2) definição do protocolo de comunicação; (3) construção de um protótipo para um grau de liberdade, e simulação.

4.1.1 ESTUDO E DEFINIÇÃO DO PROBLEMA

O problema consistiu em construir um sistema de teleoperação de um grau de liberdade. Para que o sistema fosse concebido foram estabelecidas algumas premissas que serão listadas a seguir:

- ambos os manipuladores devem ter o mesmo número de graus de liberdade
- a área de trabalho do mestre deve estar livre de obstáculos, o mesmo não é necessário no escravo, já que este pode estar em um ambiente desconhecido.
- um movimento qualquer gerado no mestre deve ser transmitido ao escravo e este deve executá-lo precisamente, exceto quando da presença de um obstáculo. Caso ocorra uma colisão com o obstáculo, ela deve ser transmitida ao mestre dando assim a sensação de realimentação háptica.

O sistema foi definido conforme mostra a FIG. 4.1. Esta figura mostra o diagrama de blocos de um sistema de controle de posição da i -ésima junta do manipulador. Nota-se também, que o sistema foi definido em três sub-sistemas distintos: o sub-sistema mestre, o sub-sistema escravo e o sub-sistema de comunicação, que une o mestre ao escravo.

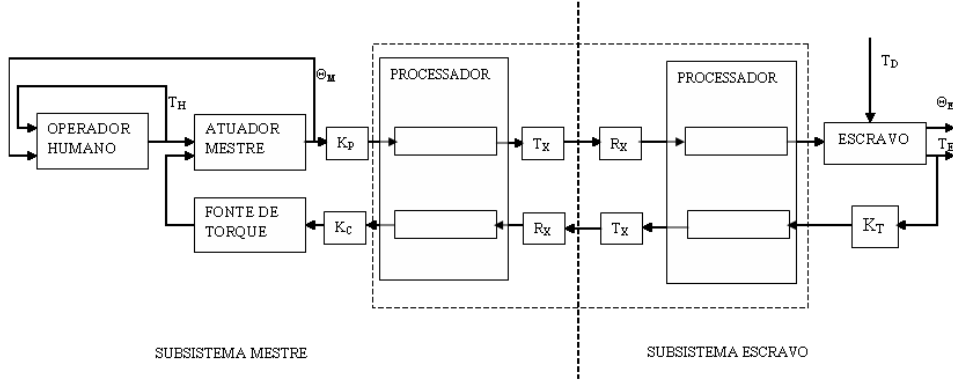


FIG 4.1: Diagrama de blocos representando o esquema do manipulador proposto, onde T_H é torque de entrada, T_D é um ruído, T_E é o torque sentido no manipulador escravo, θ_M e θ_E são os ângulos medidos no mestre e escravo, respectivamente.

A leitura da FIG. 4.1 é feita da seguinte forma: o operador humano aplica um torque T_H executando assim um deslocamento angular θ_M no mestre; esse deslocamento é convertido de ângulo para tensão por K_P e é entregue ao meio físico para ser transmitido ao escravo. Como o escravo está em um ambiente remoto, fica sujeito a forças externas que neste caso são representadas por T_D ; assim que o escravo recebe o sinal, ele executa um deslocamento angular θ_E e um torque T_E é gerado; este torque é convertido para tensão por K_T e entregue ao canal de comunicação; antes de chegar no mestre, ele passa por um fator de conversão de tensão para corrente K_C e o processo continua até que o mestre não seja mais requisitado.

4.1.2 O PROTOCOLO DE COMUNICAÇÃO

O estudo do protocolo de comunicação feito por (FERNANDES, 2002), teve como enfoque o fato de que a manutenção das posições angulares do sistema deviam ser feitas em tempo-real; assim, o projeto do protocolo foi realizado de tal forma que transmitisse os dados de um modo dedicado e foi descrito com segue:

- eliminação de procedimentos existentes nos protocolos padrões.
- a comunicação sempre será realizada de um mestre para um escravo e vice-versa.
- a comunicação deve ser feita utilizando um canal dedicado, e isto elimina a necessidade de tabela de roteamento.

De uma forma resumida o protocolo foi definido como sendo uma tripla, contendo (1) bloco de pedido de conexão; (2) bloco de pedido de conexão recebido e; (3) bloco de dados.

4.1.3 IMPLEMENTAÇÃO FÍSICA DO SISTEMA

A implementação física do sistema foi feita levando-se em consideração um sistema de 1 GDL. A FIG. 4.2 e a FIG. 4.3 mostram uma representação do motor e do diagrama de blocos do sistema.

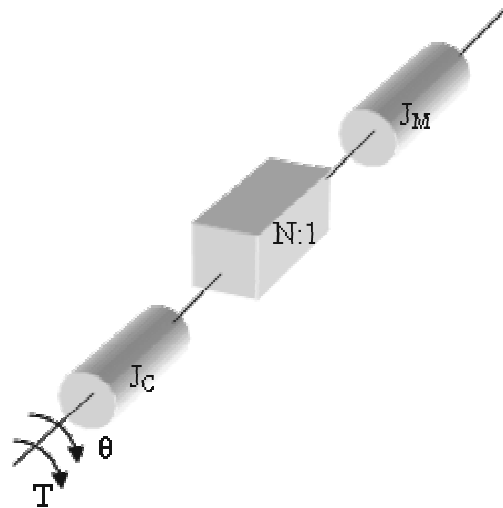


FIG 4.2: Representação dos motores com seus respectivos momentos de inércia (J_M e J_C e fator de redução das engrenagens ($N : 1$)).

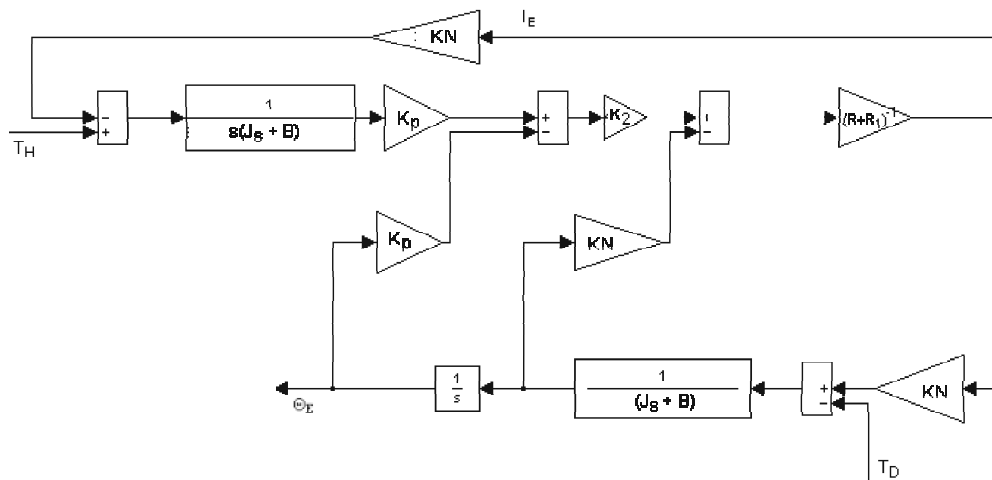


FIG 4.3: Diagrama de blocos para o manipulador em termos de função de transferência (FERNANDES, 2002).

- J_C é o momento de inércia do escravo

- J_M é o momento de inércia do mestre.
- N é a relação das engrenagens que ligam o eixo do motor ao eixo da carga.
- K_P constante do potenciômetro.
- A_1 ganho do amplificador de sinais.
- A_2 ganho do amplificador de potências.
- R_m resistência do rotor do motor.
- R Resistência colocada em série com o rotor para sensoriamento da corrente que circula no motor escravo.
- K constante de torque e/ou constante de tensão.
- T_D perturbação externa (torque).

Na FIG. 4.2, θ é o deslocamento angular. As equações no domínio do tempo são mostrados na EQ. 4.1, EQ. 4.2, EQ. 4.3, e EQ. 4.4. B é o coeficiente de atrito do motor.

$$T(t) = (J_C + N^2 J_M) \frac{d^2 \theta}{dt^2} \quad (4.1)$$

$$em(t) = R_m i_m(t) + K \frac{d\theta}{dt} \quad (4.2)$$

$$T_m(t) = K i_m(t) \quad (4.3)$$

$$T_m(t) = J \frac{d^2 \theta}{dt^2} + B J \frac{d\theta}{dt} + T_D(t) \quad (4.4)$$

A FIG. 4.4 e a FIG. 4.5 mostram o esboço mecânico do mestre e do escravo, respectivamente. Na FIG. 4.5, é possível notar que foram colocados obstáculos móveis a fim de simular um ambiente real. Na seção 8.1 do Capítulo 9 que será descrita a posteriori, é possível ver uma maquete eletrônica e o sistema real.

A seguir, será feita uma breve apresentação do trabalho de CORREIA (2003), que também serviu de base para este trabalho.

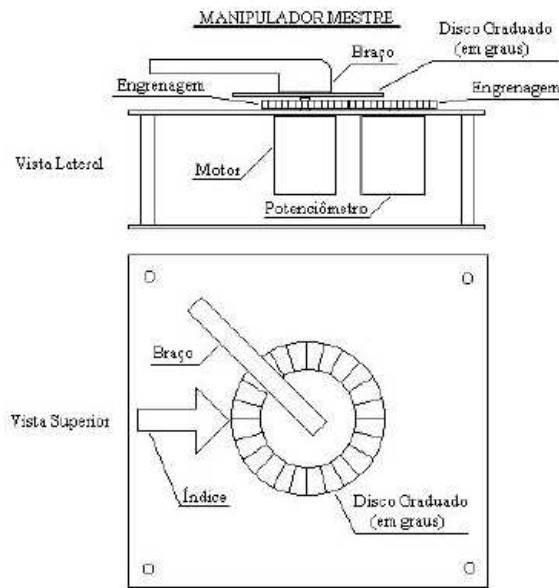


FIG 4.4: Esboço mecânico do manipulador mestre, onde o simulador pode ser observado sob uma vista lateral e uma frontal (FERNANDES, 2002).

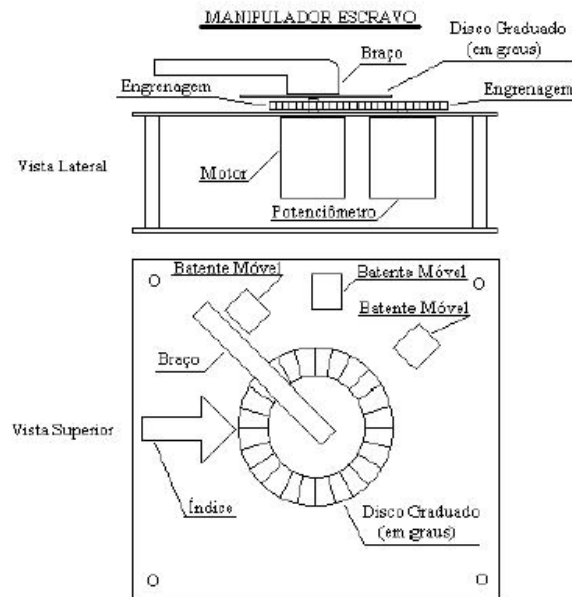


FIG 4.5: Esboço mecânico do manipulador escravo, onde o simulador pode ser observado sob uma vista lateral e uma frontal (FERNANDES, 2002).

4.2 MANIPULADOR DO TIPO MESTRE-ES CRAVO 3 GDL COM REALIMENTAÇÃO HÁPTICA

Esta seção, tratará brevemente do trabalho desenvolvido por CORREIA (2003).

4.2.1 DEFINIÇÃO DO PROBLEMA

O trabalho de (CORREIA, 2003) consistiu no projeto de um manipulador de 3 GDL, a implementação de um protocolo de comunicação entre os sistemas mestre-escravo e a validação deste através de um simulador desenvolvido na linguagem C++. A FIG. 4.6 (a) e (b) mostra em detalhes o manipulador proposto, onde ele consiste em uma seqüência de elos, de rotação, L_A , L_B e L_C , dispostos em série e conectados por juntas atuantes, $J1$, $J2$ e $J3$. Para descrever a relação entre os elos, é assinalado um sistema de coordenadas para cada um. A variável θ_n é o ângulo de cada junta do manipulador.

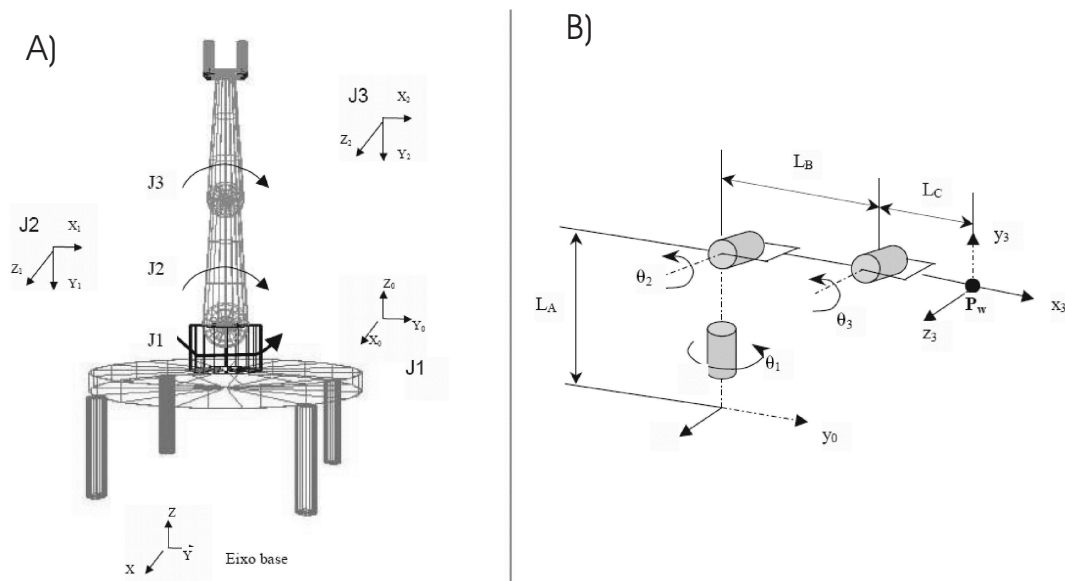


FIG 4.6: (a) Esboço do manipulador de 3 GDL. (b) Representação esquemática do manipulador, onde os cilindros representam os elos de rotação (CORREIA, 2003).

4.2.2 A COMUNICAÇÃO DO SISTEMA

O protocolo de comunicação proposto em (CORREIA, 2003) tem como objetivo integrar, à distância, dois sistemas robóticos com mais de um grau de liberdade, tendo como premissas o fato de que o protocolo deveria fornecer um serviço confiável de transmissão de dados e alta velocidade de transmissão. O protocolo foi baseado na arquitetura cliente/servidor, pois este tipo de arquitetura tem como característica um nível inicial de segurança e confiabilidade na troca de informações. O funcionamento do protocolo pode ser descrito pela EQ. 4.5, onde p é a posição inicial, p_f é a posição final e p' é uma função $F(p, p_f)$ e ϕ é um parâmetro do sistema, cujo o valor é selecionado pelo usuário, e que determina a quantidade de fragmentos em que a trajetória completa será dividida.

O processo inicia-se quando o usuário do sistema informa o deslocamento desejado. O sistema robótico mestre elabora todo o cálculo da trajetória a ser percorrida. No passo seguinte, o sistema mestre determina o primeiro intervalo em ângulo adicionado à posição atual p' (que corresponde a $(p \times \phi) + p$), para ser enviado através do protocolo. Mas, o sistema só enviará p' através do protocolo, como calculado, caso este intervalo de ângulo não seja igual ou superior o valor de p_f ; pois caso isto aconteça, p' será transmitido com o valor de p_f .

$$p' = F(p, p_f) = \begin{cases} se[(p \times \Phi) + p] \geq p_f \rightarrow p_f \\ se[(p \times \Phi) + p] < p_f \rightarrow [(p \times \Phi) + p], \end{cases} \quad (4.5)$$

Após executar os passos descritos, o sistema mestre entra em um estado de espera; isto é, ele "adormece" até o momento da confirmação do recebimento do pacote enviado para o escravo. A mensagem de confirmação recebida, além de conter a informação de recebimento, trás também a informação do deslocamento executado pelo escravo. Desta forma, é possível saber se ocorreu alguma obstrução durante a execução do movimento. Caso o movimento esteja livre de obstruções, o sistema mestre executa o deslocamento até uma posição p' , que foi enviada pelo escravo, e calcula uma nova posição e a envia para o sistema escravo. Esta nova posição é constituída do novo deslocamento adicionando o valor de p' , ou seja, $p'' = F(p', x)$. [Vide FIG. 4.7 (a)].

Para o caso da ocorrência de uma obstrução, o sistema mestre interpreta as informações do pacote contendo a posição exata em que o sistema parou. Dessa forma, o sistema mestre executa o deslocamento até o ponto em que ocorreu a obstrução. Sendo assim, ao invés do sistema mestre posicionar-se na posição p' , ele se posicionará em uma posição $(p' + \delta)$, onde δ é o intervalo percorrido pelo escravo até o ponto onde a obstrução foi detectada. [Vide FIG. 4.7 (b)].

4.2.3 SIMULAÇÃO E VALIDAÇÃO DO PROBLEMA

Para a validação do sistema, foi elaborado um simulador desenvolvido na linguagem C++ que representa um manipulador mestre e um manipulador escravo, ambos com 3 GDL. O simulador foi dividido em dois módulos, ambos com 3 GDL, juntas de revolução e elos com dimensões proporcionais a um braço humano. A FIG. 4.8 mostra uma interface gráfica que representa os manipuladores mestre e escravo, que é o resultado da aplicação das equações cinemáticas desenvolvidas, juntamente com a utilização do padrão gráfico Open Gl, respectivamente. A FIG. 4.8, também, ilustra um exemplo de utilização do

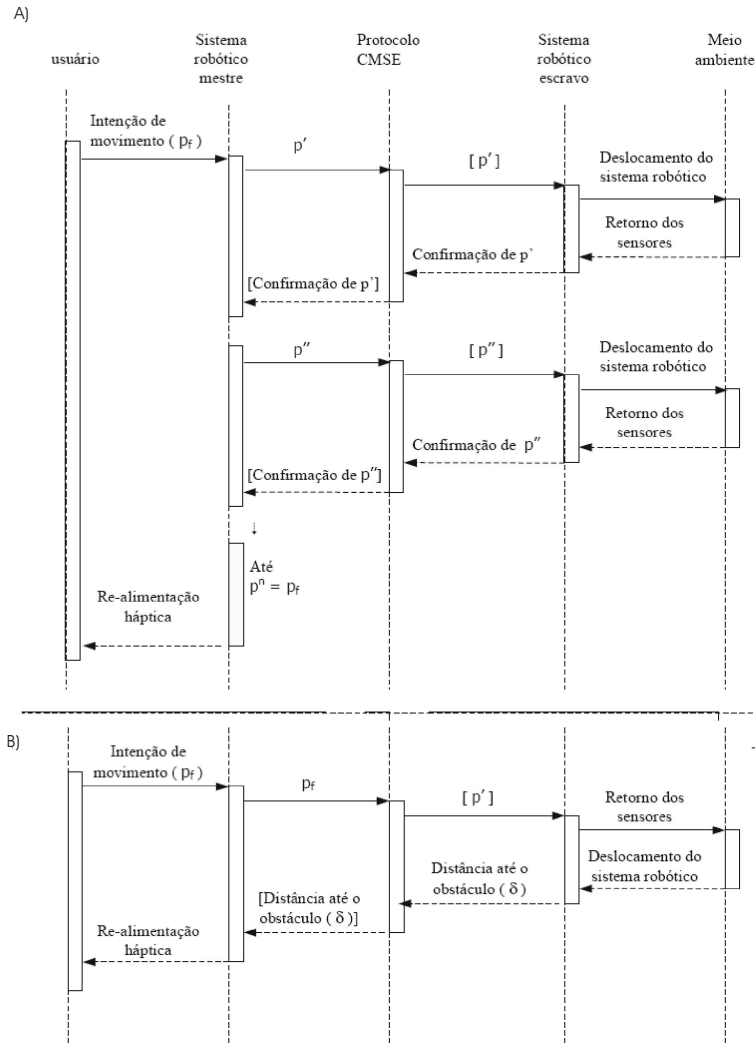


FIG 4.7: Diagrama de sequência caracterizando o protocolo de comunicação. (a) Transmissão da informação sem a presença de obstáculos. (b) Simulação da ocorrência de obstáculos na transmissão (CORREIA, 2003).

simulador, onde cada movimento executado pelo mestre é transmitido ao sistema escravo que tem como objetivo repetir fielmente este movimento.

A simulação constituiu-se, basicamente, de dois testes distintos: O primeiro deles, teve por objetivo verificar a consistência da troca de informações, através do simulador construído. Foi feita uma bateria de testes, que consistia em escolher posições aleatórias para cada elo do manipulador; no total, foram escolhidas 50 posições. O segundo teste, compôs-se do sistema robótico percorrendo 5 (cinco) trajetórias distintas, sendo que a variável Φ , EQ. 4.5, foi associada a 4 (quatro) valores: um para cada uma das trajetórias elaboradas. Vale frisar que um dos principais requisitos do sistema consiste na repetição dos movimentos executados pelo sistema mestre por parte do escravo. Sendo assim,

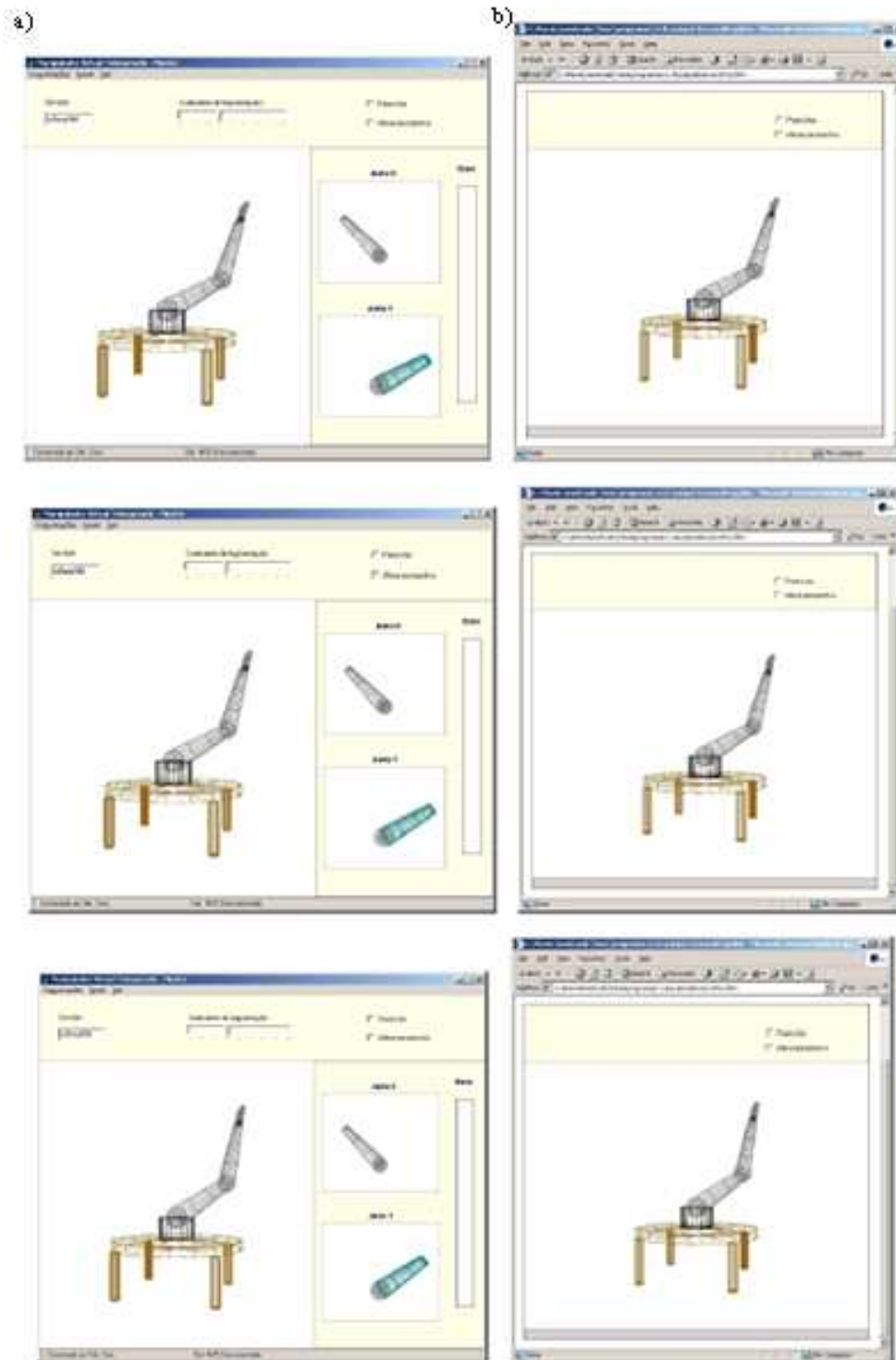


FIG 4.8: Sequência representativa de uma das baterias de testes executadas. (a) Visualização do mestre em movimento. (b) Visualização do escravo executando os comandos enviados pelo mestre (CORREIA, 2003).

o principal objetivo a ser verificado neste teste foi o comportamento do protocolo de comunicação utilizado entre o mestre e o escravo, com relação aos fatores tempo real e

precisão. Ficou provado que quanto maior o valor de Φ , maior é o sincronismo entre o sistema mestre e o sistema escravo. Por outro lado, quanto menor for o valor da variável Φ maior é o coeficiente do fator de tempo-real. Ou seja, os sistemas robóticos têm o seu desempenho degradado, isto porque, a variável Φ interfere diretamente no número de fragmentos que percorre a rede de comunicação.

5 PROJETO DO MANIPULADOR ROBÓTICO

Neste capítulo, serão estudadas as equações matemáticas que compõem o manipulador robótico. Inicialmente, serão abordados, de uma forma bem sucinta, os manipuladores robóticos, e logo após serão propostos o modelo e o esquema do manipulador. Em seguida, será feito o equacionamento cinemático e dinâmico para o manipulador proposto. O capítulo é concluído com uma discussão sobre a viabilidade do projeto.

5.1 INTRODUÇÃO

Os manipuladores robóticos em sua maioria são constituídos por um conjunto de elos rígidos ligados em série através de juntas rotacionais e/ou prismáticas. O número de juntas de um manipulador informa o grau de liberdade do mesmo, e a disposição em que as juntas estão dispostas informam o espaço de trabalho que este descreve. Comercialmente, a maioria dos manipuladores possui seis graus de liberdade (GDL), pois com esta configuração é possível posicionar e orientar o efetuador final em qualquer ponto do espaço.

Neste trabalho, será elaborado o projeto de um manipulador de 3 (três) GDL; este manipulador não segue os padrões comerciais existentes, ele será projetado com o objetivo de detectar colisões em um ambiente desconhecido e transmití-las a um outro manipulador, dando assim a informação de realimentação háptica a quem estiver operando o sistema.

5.2 MODELO DO MANIPULADOR ROBÓTICO

De uma forma geral, a cinemática do manipulador robótico pode ser escrita conforme a EQ. 5.1, (PAUL, 1981b). Onde $x \in R^m$, R^m representa um vetor de coordenadas no espaço operacional. $q \in R^n$, R^n representa um vetor de coordenadas no espaço das juntas e f é uma aplicação de $R^n \mapsto R^m$, n indica o grau de liberdade do manipulador e m é a dimensão do espaço operacional.

$$x = f(q), \tag{5.1}$$

O problema cinemático divide-se em problema cinemático direto e cinemático inverso; o primeiro determina a posição do efetuador final, para uma configuração pré-definida do manipulador. Já o problema cinemático inverso, consiste em determinar toda a configuração dos parâmetros de um manipulador, a partir de uma posição determinada. Dessa forma, o problema cinemático inverso é, em muitos casos, difícil ou mesmo impossível de se obter analiticamente.

5.3 EQUAÇÕES CINEMÁTICAS DO MANIPULADOR

A FIG. 5.1 mostra o esquema do manipulador proposto. Este possui 3 (três) GDL, duas juntas rotacionais, com $0 \leq \theta_i \leq 2\pi$, e uma junta prismática. A FIG. 5.1 mostra um esquema do manipulador, onde os cilindros representam as juntas rotacionais e o cubo representa a junta prismática; ainda na FIG. 5.1, observa-se que foram marcados eixos de orientação, que servem de auxílio para a obtenção das variáveis que são utilizadas no projeto do manipulador. Na FIG. 5.2, é possível ver uma representação no R^3 , sob as vistas perspectiva, frontal, lateral e topo.

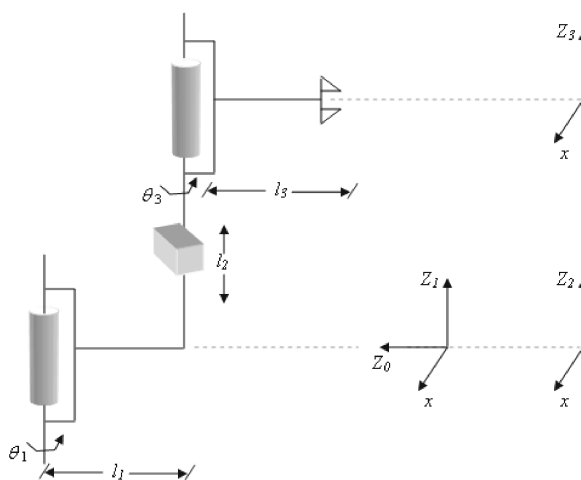


FIG 5.1: Esquema do manipulador proposto.

5.3.1 TRANSFORMAÇÕES HOMOGÊNEAS

As equações cinemáticas para o manipulador proposto na FIG. 5.1 são obtidas utilizando-se a convenção de Denavit-Hartenberg⁵. Uma matriz de transformação ho-

⁵A convenção de Denavit-Hartenberg descreve o relacionamento entre dois sistemas de referência locais. Para o caso dos elos do manipulador, cada um deles é descrito em relação ao seu elo adjacente por 4 parâmetros, [vide TAB. 5.1].

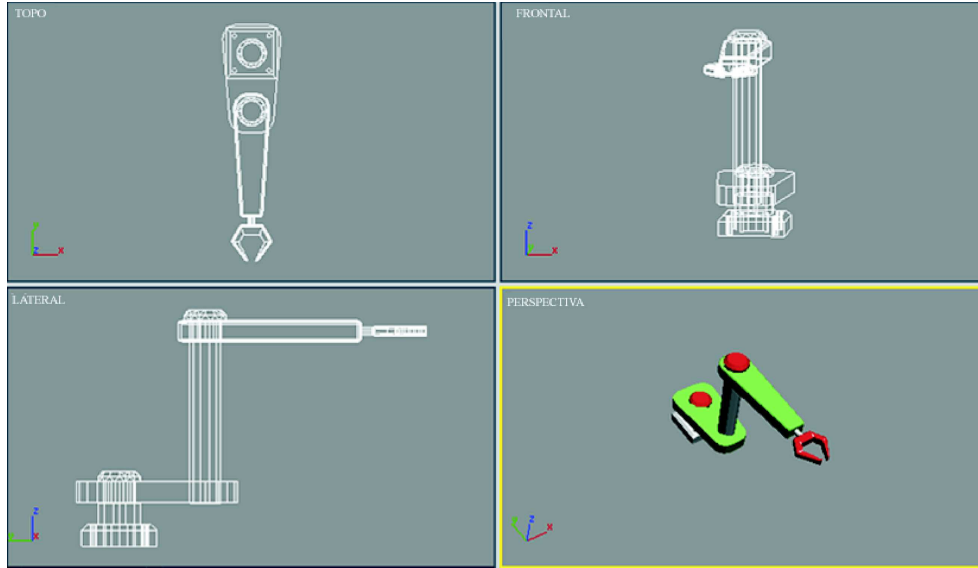


FIG 5.2: Esquema do manipulador no R^3 .

homogênea indica a posição e orientação do efetuador final, EQ. 5.2.

$$\begin{bmatrix} p_x \\ p_y \\ p_z \\ 1 \end{bmatrix} = \begin{bmatrix} & & d_x \\ & R_{3 \times 3} & d_y \\ & & d_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} p_u \\ p_v \\ p_w \\ 1 \end{bmatrix}, \quad (5.2)$$

O vetor $\begin{bmatrix} p_x & p_y & p_z & 1 \end{bmatrix}^T$ faz referência ao sistema de coordenadas fixo, já o vetor $\begin{bmatrix} p_u & p_v & p_w & 1 \end{bmatrix}^T$ faz referência ao sistema de coordenadas móvel. $R_{3 \times 3}$ é a matriz que fornece informações sobre rotações e translações. O vetor $\begin{bmatrix} dx & dy & dz \end{bmatrix}^T$ é o vetor de posição da origem do sistema referenciado, tal qual modelado de acordo com a origem do sistema de referência. A matriz de transformações homogêneas também é conhecida como matriz T , que pode ser vista na EQ. 5.3.

$$T = \begin{bmatrix} n_x & o_x & a_x & d_x \\ n_y & o_y & a_y & d_y \\ n_z & o_z & a_z & d_z \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (5.3)$$

Observando-se a FIG. 5.3, pode-se ver um sistema coordenado, rotulado como n , o e a que são três vetores ortogonais usados para especificar a orientação da garra. O vetor $\vec{a} = \begin{bmatrix} a_x & a_y & a_z \end{bmatrix}^T$ que segue a orientação do eixo z e aponta na direção em que a garra

aproxima-se de um objeto. O vetor $\vec{o} = \begin{bmatrix} o_x & o_y & o_z \end{bmatrix}^T$ que está na orientação do eixo y , especifica a orientação da garra de dedo para dedo, e o vetor $\vec{n} = \begin{bmatrix} n_x & n_y & n_z \end{bmatrix}^T$ que é o vetor normal, pode ser escrito conforme a EQ. 5.4. A FIG. 5.3 mostra a disposição dos vetores \vec{n} , \vec{o} e \vec{a} em uma garra.

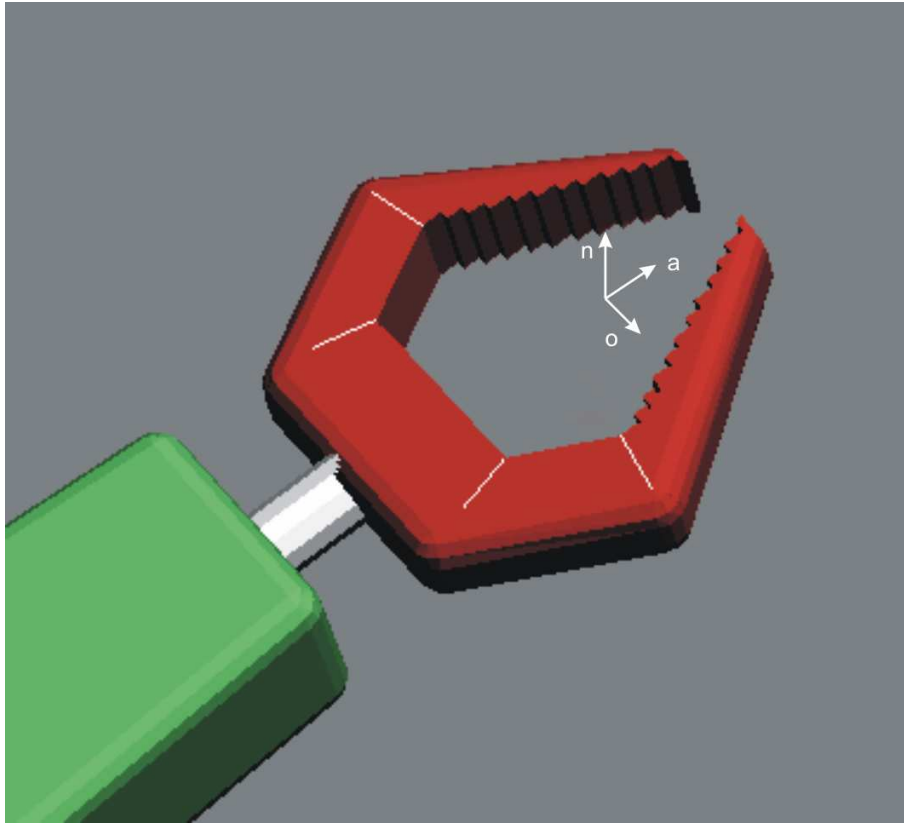


FIG 5.3: Modelo de uma garra com a indicação do sistema de coordenadas.

$$n = o \times a, \quad (5.4)$$

Após definir o formato da matriz T , pode-se obter as equações cinemáticas do manipulador através do método proposto por Denavit-Hartenberg (PAUL, 1981a; SCIAVICCO, 1996; FU, 1987), o que conduz às matrizes ${}^{i-1}A_i$, para $i = 1, \dots, n$, onde n é o número de graus de liberdade do manipulador. A matriz T é obtida pela multiplicação das matrizes A (ou matrizes de transformação homogêneas), como pode ser visto na EQ. 5.5.

$$T = {}^0 A_1 \cdot {}^1 A_2 \cdot \dots \cdot {}^{n-1} A_n, \quad (5.5)$$

Com base nas informações da FIG. 5.1, pode-se extrair os dados para o preenchimento

TAB 5.1: Parametros de DENAVIT-HARTEMBERG

ELoS	θ_i	α_i	a_i	d_i
1	θ_1	-90	0	0
2	0	0	0	l_1
3	θ_3	0	0	l_2

da TAB. 5.1, onde a é a menor distância entre os eixos, α representa a torção de um eixo em relação a outro, d é a distância entre as juntas e θ é o ângulo entre as juntas. Matrizes de transformações homogêneas para o sistema são mostradas na EQ. 5.6, EQ. 5.7 e EQ. 5.8.

$$A_1 = \begin{bmatrix} C_1 & 0 & -S_1 & 0 \\ S_1 & 0 & C_1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (5.6)$$

$$A_2 = \begin{bmatrix} C_2 & -S_2 & 0 & 0 \\ S_2 & C_2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (5.7)$$

$$A_3 = \begin{bmatrix} C_3 & -S_3 & 0 & l_2 C_3 \\ S_3 & C_3 & 0 & l_2 S_3 \\ 0 & 0 & 1 & l_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (5.8)$$

A matriz de transformação T_3 procurada é obtida conforme a EQ. 5.9, sendo assim a matriz T_3 pode ser vista na EQ. 5.10.

$$T_3 = {}^0 A_1 \cdot {}^1 A_2 \cdot {}^2 A_3, \quad (5.9)$$

$$T_3 = \begin{bmatrix} C_1 C_2 C_3 - C_1 S_2 S_3 & -C_1 C_2 S_3 - C_1 S_2 C_3 & -S_1 & C_1 C_2 l_2 C_3 - C_1 S_2 l_2 S_3 - S_1 l_2 \\ S_1 C_2 C_3 - S_1 S_2 S_3 & -S_1 C_2 S_3 - S_1 S_2 C_3 & C_1 & S_1 C_2 l_2 C_3 - S_1 S_2 l_2 S_3 + C_1 l_2 \\ -S_2 C_3 - C_2 S_3 & S_2 S_3 - C_2 C_3 & 0 & -S_2 l_2 C_3 - C_2 l_2 S_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.10)$$

Igualando-se a 5.3 com a EQ. 5.10, obtém-se as equações (5.11), (5.12), (5.13), (5.14), (5.15), (5.16), (5.17), (5.18), (5.19), (5.34), (5.35) e (5.22).

$$n_x = C_1 C_2 C_3 - C_1 S_2 S_3, \quad (5.11)$$

$$n_y = S_1 C_2 C_3 - S_1 S_2 S_3, \quad (5.12)$$

$$n_z = -S_2 C_3 - C_2 S_3, \quad (5.13)$$

$$o_x = -C_1 C_2 S_3 - C_1 S_2 C_3, \quad (5.14)$$

$$o_y = -S_1 C_2 S_3 - S_1 S_2 C_3, \quad (5.15)$$

$$o_z = S_2 S_3 - C_2 C_3, \quad (5.16)$$

$$a_x = -S_1, \quad (5.17)$$

$$a_y = C_1, \quad (5.18)$$

$$a_z = 0, \quad (5.19)$$

$$d_x = C C_1 C_2 l_2 C_3 - C_1 S_2 l_2 S_3 - S_1 l_2, \quad (5.20)$$

$$d_y = S_1 C_2 l_2 C_3 - S_1 S_2 l_2 S_3 + C_1 l_2, \quad (5.21)$$

$$d_z = -S_2 l_2 C_3 - C_2 l_2 S_3, \quad (5.22)$$

5.3.2 SOLUÇÃO DAS EQUAÇÕES CINEMÁTICAS PARA O MANIPULADOR

A equação da matriz T_3 é obtida pela sucessiva mutiplicação das matrizes da EQ. 5.10 pelo inverso das matrizes A . Dessa forma obtêm-se as equações descritas na EQ. (5.23) e na EQ. (5.24).

$$A_1^{-1} T_3 = {}^1 T_3, \quad (5.23)$$

$$A_2^{-1} A_1^{-1} T_3 = {}^2 T_3, \quad (5.24)$$

A EQ. 5.23, também podem ser escritas da forma como mostrado na EQ. 5.25.

$$A_1^{-1} T_3 = A_2 A_3, \quad (5.25)$$

O lado esquerdo da EQ. 5.23 é dado pela EQ. 5.26.

$$A_1^{-1} T_3 = \begin{bmatrix} C_1 & S_1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ -S_1 & C_1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} n_x & o_x & a_x & d_x \\ n_y & o_y & a_y & d_y \\ n_z & o_z & a_z & d_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.26)$$

Para facilitar a solução da EQ. 5.25, será adotada a EQ. 5.27 como sugerido em (PAUL, 1981a), onde, g_{ij} é uma função em que o índice i fornece informação a respeito da junta a qual se está referindo, e o índice j refere-se ao eixo de referência.

$$A_1^{-1}T_3 = \begin{bmatrix} g_{11}(n) & g_{11}(o) & g_{11}(a) & g_{11}(d) \\ g_{12}(n) & g_{12}(o) & g_{12}(a) & g_{12}(d) \\ g_{13}(n) & g_{13}(o) & g_{13}(a) & g_{13}(d) \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.27)$$

Igualando-se a EQ. 5.26 com a EQ. 5.27 obtêm-se as equações descritas na EQ. 5.28, EQ. 5.29 e na EQ. 5.30.

$$g_{11} = C_1x + S_1y, \quad (5.28)$$

$$g_{12} = z, \quad (5.29)$$

$$g_{13} = -S_1x + C_1y. \quad (5.30)$$

A matriz 1T_3 é mostrada na EQ. 5.31.

$${}^1T_3 = A_2A_3 = \begin{bmatrix} C_2C_3 - S_2S_3 & -S_2C_3 - C_2S_3 & 0 & C_2l_2C_3 - S_2l_2S_3 \\ S_2C_3 + C_2S_3 & C_2C_3 - S_2S_3 & 0 & S_2l_2C_3 + C_2l_2S_3 \\ 0 & 0 & 1 & l_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (5.31)$$

A EQ. 5.33 pode ser obtida utilizando-se a EQ. 5.31 e a EQ. 5.30 sendo possível determinar o valor da variável θ_1 .

$$g_{13}(d) = l_2. \quad (5.32)$$

$$-S_1x + C_1y = l_2. \quad (5.33)$$

Para resolver a EQ. 5.33 pode-se utilizar as relações trigonométricas mostradas na EQ. 5.34 e na EQ. 5.35.

$$d_x = r.\cos(\phi). \quad (5.34)$$

$$d_y = r.\sen(\phi). \quad (5.35)$$

Utilizando-se a EQ. 5.34 e a EQ. 5.35 pode-se obter as equações mostradas na EQ. 5.36 e na EQ. 5.37.

$$r = \sqrt{d_x^2 + d_y^2}. \quad (5.36)$$

$$\phi = \text{ArcTan} \left(\frac{d_y}{d_x} \right). \quad (5.37)$$

Substituindo a EQ. 5.36 e a EQ. 5.37 na EQ. 5.33 obtém-se a EQ. 5.38.

$$\text{sen}(\phi)\cos(\theta_1) - \cos(\phi)\text{sen}(\theta) = (l_2)/r. \quad (5.38)$$

Com,

$$0 < (l_2) \leq 1 \quad (5.39)$$

A EQ. 5.38 pode ser reduzida a EQ. 5.40

$$\text{sen}(\phi - \theta_1) = (l_2)/r, \quad (5.40)$$

com,

$$0 < (\phi - \theta_1) < 2\pi \quad (5.41)$$

O cosseno é dado pela EQ. 5.42.

$$\cos(\phi - \theta_1) = \pm\sqrt{1 - (l_2)^2}, \quad (5.42)$$

A tangente é dada pela EQ. 5.45

$$\tan(\phi - \theta_1) = \frac{(l_2)}{\pm\sqrt{r^2(1 - (l_2)^2)}}, \quad (5.43)$$

$$\phi - \theta_1 = \text{ArcTan} \left(\frac{(l_2)}{\pm\sqrt{r^2 - r^2(l_2)^2}} \right), \quad (5.44)$$

Substituindo a EQ. 5.37 na EQ. 5.45 e rearranjando os termos obtém-se o valor de θ_1 .

$$\theta_1 = \text{ArcTan} \left(\frac{d_y}{d_x} \right) - \text{ArcTan} \left(\frac{(l_2)}{\pm\sqrt{r^2 - r^2(l_2)^2}} \right), \quad (5.45)$$

Após determinar θ_1 e l_2 pode-se utilizar a mesma sequência de passos para determinar o valor θ_3 ; dessa forma, substituindo-se os valores na EQ. 5.24 obtém-se a EQ. 5.46, onde: S_i é o seno de θ_i e C_i é cosseno de θ_i .

$$A_2^{-1}A_1^{-1}T_3 = \begin{bmatrix} C_2 & S_2 & 0 & 0 \\ -S_2 & C_2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} C_1 & S_1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ -S_1 & C_1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} n_x & o_x & a_x & d_x \\ n_y & o_y & a_y & d_y \\ n_z & o_z & a_z & d_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.46)$$

Resolvendo parcialmente o lado esquerdo da EQ. 5.46 obtém-se a EQ. 5.47

$$A_2^{-1}A_1^{-1}T_3 = \begin{bmatrix} C_1C_2 & S_1C_2 & -S_2 & 0 \\ -C_1S_2 & -S_1S_2 & -C_2 & 0 \\ -S_1 & C_1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} n_x & o_x & a_x & d_x \\ n_y & o_y & a_y & d_y \\ n_z & o_z & a_z & d_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.47)$$

$$A_2^{-1}A_1^{-1}T_3 = \begin{bmatrix} g_{31}(n) & g_{31}(o) & g_{31}(a) & g_{31}(d) \\ g_{32}(n) & g_{32}(o) & g_{32}(a) & g_{32}(d) \\ g_{33}(n) & g_{33}(o) & g_{33}(a) & g_{33}(d) \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.48)$$

Tomando por base a EQ. 5.47 e a EQ. 5.48, obtém-se as equações 5.49, 5.50 e 5.51.

$$g_{31} = C_1C_2x + S_1C_2y - S_2z, \quad (5.49)$$

$$g_{32} = -(C_1S_2)x - S_1S_2y - C_2z, \quad (5.50)$$

$$g_{33} = -S_1x + C_1y \quad (5.51)$$

Sabe-se que ${}^2T_3 = A_3$, dessa forma substituindo-se os valores obtém-se a EQ. 5.52.

$${}^2T_3 = A_3 = \begin{bmatrix} C_3 & -S_3 & 0 & l_2C_3 \\ S_3 & C_3 & 0 & l_2S_3 \\ 0 & 0 & 1 & l_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (5.52)$$

Utilizando-se a EQ. 5.52 pode-se obter uma expressão para θ_3 , basta tomar os elementos diferentes de zero das linhas 1 e 2 e coluna 1 e compará-los com a EQ. 5.49 e a EQ. 5.50, respectivamente. Dessa forma, obtém-se a EQ. 5.53 que é a solução para θ_3 .

$$\theta_3 = \text{ArcTan} \left(\frac{(C_1C_2)d_x + (S_1C_2)d_y - (S_2)d_z}{-(C_1S_2)d_x - (S_1S_2)d_y - (C_2)d_z} \right), \quad (5.53)$$

Até o momento foi determinada a matriz T_3 e os valores para θ_1 , θ_2 e θ_3 . A próxima seção tratará do jacobiano do manipulador, de como é feita a sua obtenção e a sua importância.

5.4 O JACOBIANO DO MANIPULADOR

Sabe-se que as equações da cinemática direta definem apenas o espaço cartesiano das posições das juntas, ou seja as relações entre as velocidades não são consideradas. O jacobiano surge na forma de uma função matricial que supre esta deficiência.

O jacobiano pode ser imaginado como uma versão vetorial da derivada ordinária de uma função escalar. Em outras palavras, ele é uma matriz que possui informações sobre as velocidades escalares e angulares das juntas do manipulador.

A importância do Jacobiano para o manipulador é que o mesmo aparece em situações como:

- planejamento de trajetórias;
- determinação das singularidades;
- obtenção das equações dinâmicas do movimento; e
- transformação de forças e torques do efetuador final para forças e torques atuantes nas juntas do manipulador.

O jacobiano para o manipulador proposto é uma matriz $J_{6 \times 3}$, que pode ser vista na EQ. 5.54. [vide APÊNDICE 12.1 para maiores detalhes].

$$J = \begin{bmatrix} -l_2[(C_2 + C_3) - 2(C_1 C_2 C_3 S_1 S_2 S_3) - S_2 S_3 (S_1^2 - C_1^2)] & C_2 C_3 - S_2 S_3 & 0 \\ l_2(C_2 C_3 + S_2 C_3) & -S_2 C_3 - C_2 S_3 & l_2 \\ l_2(C_2 S_3 + S_2 C_3) & 0 & 0 \\ -S_2 C_3 - C_2 S_3 & 0 & 0 \\ S_2 S_3 - C_2 C_3 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (5.54)$$

Nesta seção, foi dada uma breve explicação sobre a função que o jacobiano desempenha no projeto da construção de sistemas mecânicos e posteriormente encontrou-se o jacobiano para o manipulador proposto. A próxima seção trata da obtenção das equações dinâmicas para o manipulador proposto neste trabalho.

5.5 EQUAÇÕES DINÂMICAS DO MANIPULADOR

Após ter determinado as equações cinemáticas do manipulador, faz-se necessário determinar a dinâmica do mesmo, pois esta fornece informações a respeito dos torques

que são aplicados às juntas, dos efeitos gravitacionais e das velocidades. Para este, trabalho será utilizado o método de Lagrange-Euler⁶ para a determinação das equações dinâmicas, pois é um método que possui como característica o fato de permitir a solução de sistemas dinâmicos complexos de uma forma bem simples.

Sejam: $L \implies$ Lagrangiano.

$E_c \implies$ Energia Cinética.

$E_p \implies$ Energia Potencial.

Então:

$$L = E_c - E_p, \quad (5.55)$$

As equações dinâmicas, em termos de coordenadas usadas para expressar a energia cinética e potencial, são obtidas como:

$$F_i = \frac{d}{dt} \frac{\partial L}{\partial \dot{q}_i} - \frac{\partial L}{\partial q_i} \quad (5.56)$$

sendo:

$q_i \implies$ São as coordenadas em que a E_c e E_p são expressadas.

$\dot{q}_i \implies$ é a velocidade correspondente ao i -ésimo elo.

$F_i \implies$ é a força ou torque.

A seguir, será definida a relação das forças e velocidades entre as juntas; essa relação será descrita pela letra D , dessa forma, tem-se:

$D_{ii} \implies$ é a efetiva inércia na junta i (como uma aceleração na junta i causa um torque na junta i igual a $D_{ii}\dot{\theta}_i$).

$D_{ij} \implies$ é a efetiva inércia na junta i ou j (como uma aceleração na junta i ou j causa um torque na junta i ou j igual a $D_{ij}\dot{\theta}_i$ ou $D_{ij}\dot{\theta}_j$).

$D_i \implies$ é a a força da gravidade agindo na junta i .

A seguir, serão obtidos os valores de D_{ii} , D_{ij} e D_i . A formulação matemática para obtê-los está descrita no apêndice 12.2.

Para calcular D_{33} serão necessários utilizar 2T_3 , 3d_3 e ${}^3\delta_3$ que já foram calculados anteriormente na EQ. 5.52 e na EQ. 5.54.

⁶Equações de Lagrange-Euler, são um conjunto de equações diferenciais capazes de descrever a natureza da trajetória do movimento de um objeto que está sujeito a forças.

$${}^3\delta_3 = \begin{bmatrix} 0 \\ l_2 \\ 0 \\ 1 \end{bmatrix}, \quad (5.57)$$

$${}^3d_3 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad (5.58)$$

Substituindo os valores de 3d_3 e ${}^3\delta_3$ na EQ. 12.10 [vide Apêndice 12.2] obtêm-se a EQ. 5.59.

$$D_{33} = m_3 l_2^2 \quad (5.59)$$

Para calcular D_{22} , será necessário a utilização de 1T_3 , previamente obtida na EQ. 5.31 pela permutação de 2T_3 por A_2 e 1T_2 , que é A_2 [vide EQ. 5.7]. Através dessas transformações, pode-se obter os vetores 3d_2 , ${}^3\delta_2$, 2d_2 e ${}^2\delta_2$.

$${}^3d_2 = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}, \quad (5.60)$$

$${}^3\delta_2 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad (5.61)$$

$${}^2d_2 = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}, \quad (5.62)$$

$${}^2\delta_2 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad (5.63)$$

Substituindo-se os valores de 3d_2 , ${}^3\delta_2$, 2d_2 e ${}^2\delta_2$ na EQ. 12.10 [vide Apêndice 12.2], obtêm-se a EQ. 5.64.

$$D_{22} = m_2 + m_3 \quad (5.64)$$

Para movimentar o elo 1, é necessário utilizar-se T_3 , T_2 e T_1 , que são obtidas pela pré-multiplicação das transformações obtidas pelos elos antecessores pela matriz A_1 , sendo que $A_1 = T_1$. Através de cada uma dessas transformações, serão obtidos os vetores diferenciais de rotações e translações. Para o cálculo do D_{11} , será necessário determinar os respectivos centros de massa que são dados pela EQ. 5.74, EQ. 5.75 e EQ. 5.76.

Logo:

$$T_3 = A_1A_2A_3, \quad (5.65)$$

$$T_2 = A_1A_2, \quad (5.66)$$

$$T_1 = A_1 \quad (5.67)$$

$${}^3d_1 = \begin{bmatrix} l_2(S_2S_3 + C_2C_3) \\ l_2(S_2C_3 + C_2S_3) \\ l_2(S_2S_3 + C_2C_3) \\ 1 \end{bmatrix}, \quad (5.68)$$

$${}^3\delta_1 = \begin{bmatrix} -S_2C_3 + C_2S_3 \\ S_2S_3 - C_2C_3 \\ 0 \\ 0 \end{bmatrix}, \quad (5.69)$$

$${}^2d_1 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}, \quad (5.70)$$

$${}^2\delta_1 = \begin{bmatrix} -S_2 \\ -C_2 \\ 0 \\ 0 \end{bmatrix}, \quad (5.71)$$

$${}^1d_1 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}, \quad (5.72)$$

$${}^1\delta_1 = \begin{bmatrix} 0 \\ -1 \\ 0 \\ 0 \end{bmatrix}, \quad (5.73)$$

$${}^1\bar{r}_1 = \begin{bmatrix} \bar{x}_1 \\ \bar{y}_1 \\ \bar{z}_1 \\ 1 \end{bmatrix}, \quad (5.74)$$

$${}^2\bar{r}_2 = \begin{bmatrix} \bar{x}_2 \\ \bar{y}_2 \\ \bar{z}_2 \\ 1 \end{bmatrix}, \quad (5.75)$$

$${}^3\bar{r}_3 = \begin{bmatrix} \bar{x}_3 \\ \bar{y}_3 \\ \bar{z}_3 \\ 1 \end{bmatrix}, \quad (5.76)$$

Substituindo-se os valores da EQ. 5.68, EQ. 5.69, EQ. 5.70, EQ. 5.71, EQ. 5.72, EQ. 5.73 e EQ. 5.76 na EQ. 12.10 [vide Apêndice 12.2], obtêm-se a EQ. 5.77.

$$\begin{aligned} D_{11} = & m_1(-K_{2xx}^2) + m_2(-S_{2xx}^2 - C_{2yy}^2) + \\ & m_3[-(S_2C_3 + C_2S_3)^2k_{3xx}^2 + (S_2S_3 - C_2C_3)^2k_{3yy}^2] + \\ & 2[(\bar{x}_3 + \bar{y}_3 + \bar{z}_3)(l_2(S_2S_3 + C_2C_3)(-S_2S_3 - C_2C_3))], \end{aligned} \quad (5.77)$$

Para encontrar os termos D_{ij} e D_i , será necessário calcular os vetores ${}^{i-1}g$ e ${}^{i-1}\bar{r}_p$. Para o cálculo dos vetores g que representam a aceleração da gravidade, inicialmente, será considerado que o manipulador está na posição vertical [vide FIG. 5.1], sendo assim, a equação que o representa pode ser vista na EQ. 5.81.

$$g = \begin{bmatrix} 0 & 0 & -g & 0 \end{bmatrix} \quad (5.78)$$

$${}^0g = \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix} \quad (5.79)$$

$${}^1g = \begin{bmatrix} -g & 0 & 0 & 0 \end{bmatrix} \quad (5.80)$$

$${}^2g = \begin{bmatrix} C_2g & S_2g & 0 & 0 \end{bmatrix} \quad (5.81)$$

$${}^1\bar{r}_3 = T_3^3\bar{r}_3 \quad (5.82)$$

$${}^1\bar{r}_3 = \begin{bmatrix} \bar{x}_3C_1(C_2C_3 - S_2S_3) - \bar{y}_3C_1(C_2S_3 + S_2C_3) - \bar{z}_3(S_1) + l_2(C_1(C_2C_3 - S_2S_3) - S_1) \\ \bar{x}_3S_1(C_2C_3 - S_2S_3) - \bar{y}_3S_1(C_2S_3 + S_2C_3) + \bar{z}_3(C_1) + l_2(S_1(C_2C_3 - S_2S_3) + C_1) \\ \bar{x}_3(-S_2C_3 - C_2S_3) + \bar{y}_3(S_2S_3 - C_2C_3) + (-S_2l_2C_3 - C_2l_2S_3) \\ 1 \end{bmatrix}, \quad (5.83)$$

$${}^2\bar{r}_3 = {}^2T_3^3\bar{r}_3 \begin{bmatrix} \bar{x}_3S_3 - \bar{y}_3C_3 + l_2C_3 \\ \bar{x}_3S_3 - \bar{y}_3C_3 + l_2S_3 \\ \bar{z}_3 + l_2 \\ 1 \end{bmatrix}, \quad (5.84)$$

$${}^0\bar{r}_3 = {}^1T_3^3\bar{r}_3 \begin{bmatrix} \bar{x}_3(C_2C_3 - S_2S_3) - \bar{y}_3(S_2C_3 + C_2S_3) + l_2(C_2C_3 - S_2S_3) \\ \bar{x}_3(S_2C_3 - C_2S_3) + \bar{y}_3(C_2C_3 - S_2S_3) + l_2(S_2C_3 + C_2S_3) \\ \bar{z}_3 + l_2 \\ 1 \end{bmatrix}, \quad (5.85)$$

Para o segundo elo, os centros de massa são dados pela EQ. 5.86 e pela EQ. 5.87.

$${}^1\bar{r}_2 = {}^1T_2^2\bar{r}_2 \begin{bmatrix} \bar{x}_2C_1 - \bar{z}_2S_1 \\ \bar{x}_2S_1 + \bar{z}_2C_1 \\ -\bar{y}_2 \\ 1 \end{bmatrix}, \quad (5.86)$$

$${}^0\bar{r}_2 = T_2^2\bar{r}_2 \begin{bmatrix} \bar{x}_2C_1C_2 - \bar{y}_2C_1S_2 + \bar{z}_2S_1 \\ \bar{x}_2S_1C_2 + \bar{y}_2C_1C_2 + \bar{z}_2C_1 \\ -\bar{x}_2S_2 \\ 1 \end{bmatrix}, \quad (5.87)$$

De posse dos centros de massa e dos vetores dos termos de gravidade, torna-se possível calcular os valores de D_{12} , D_{13} , D_{23} , D_1 , D_2 e D_3 , através da aplicação direta da EQ. 12.12 e da EQ. 12.13 [vide Apêndice 12.2].

$$D_{12} = m_2 + m_2[2^2\bar{r}_2({}^2d_2 \times {}^2\delta_1)] \quad (5.88)$$

$$D_{13} = m_3[l_2^2(S_2C_3 + C_2S_3)] \quad (5.89)$$

$$D_{23} = m_3l_2 \quad (5.90)$$

$$D_1 = {}^0g[m_1 + m_2 + m_3]{}^0\bar{r}_1 \quad (5.91)$$

$$D_2 = {}^1g[m_2{}^1\bar{r}_2 + m_3{}^2\bar{r}_2] \quad (5.92)$$

$$D_3 = {}^2g(m_3{}^2\bar{r}_3) \quad (5.93)$$

5.6 CONCLUSÃO

Neste capítulo, foram otidas as equações cinemáticas e dinâmicas para um manipulador de 3(três) graus de liberdade, sendo que este possui duas juntas de rotação e uma junta prismática. O manipulador não tem o propósito e nem foi projetado com o intuito de atuar no setor industrial, comercial ou doméstico. O simulador serviu para testar o protocolo de comunicação do sistema, verificar o sincronismo do mesmo. O próximo passo é implementá-lo via software com a finalidade de testá-lo para tornar possível a implementação física.

6 DEFINIÇÃO DA ARQUITETURA DE CONTROLE

6.1 INTRODUÇÃO

Neste capítulo será detalhado o desenvolvimento de uma arquitetura de controle passo a passo; o objetivo é partir de um modelo mecânico até a análise de estabilidade. Desta forma, inicialmente será exposto um modelo mecânico; logo após, serão mostradas as equações diferenciais do modelo e em seguida será extraída a função de transferência do sistema.

6.2 MODELO MECÂNICO

Um sistema de teleoperação pode ser modelado mecanicamente através de um sistema mola-massa com amortecimento. Em (VOLPE, 1990) é feita uma análise minuciosa do desenvolvimento de uma estrutura de controle. O principal objetivo do trabalho não é ater-se somente ao desenvolvimento de uma arquitetura de controle; assim, neste trabalho será dada uma explicação não muito detalhada sobre o assunto, apenas o suficiente para a compreensão do mesmo. Esta será iniciada com o modelo de um simples braço (ou manipulador) robótico isoladamente e, no decorrer da explicação serão acrescentados outros componentes como o ambiente e a comunicação entre os manipuladores. Um braço robótico pode ser visto mecanicamente como um simples sistema mola-massa com amortecimento, [vide FIG. 6.1].

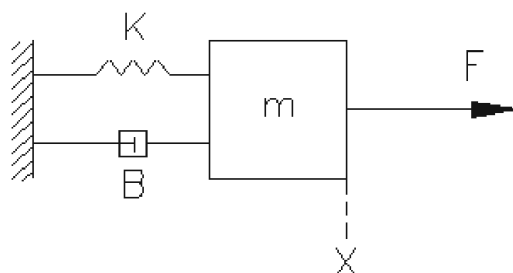


FIG 6.1: Modelo mecânico, que simula um braço robótico.

K é o coeficiente de elasticidade da mola, B é o coeficiente de atrito viscoso (amortecimento), m é a massa do braço robótico, e X é o deslocamento do sistema após a

aplicação de uma força F . A equação que modela o sistema pode ser vista na EQ. 6.1. Seja, $\frac{d}{dt}x = \dot{x}$, logo:

$$F_r = m\ddot{x} + B\dot{x} + Kx, \quad (6.1)$$

Escrevendo-se a EQ. 6.1 em termos de função de transferência, obtêm-se a EQ. 6.2.

$$\begin{cases} F = F'(t) \\ X = Y(t), \end{cases} \quad (6.2)$$

F e X foram escritas em função do tempo, daí,

$$F'(t) = m\ddot{Y}(t) + B\dot{Y}(t) + KY(t), \quad (6.3)$$

logo,

$$F'(s) = ms^2\ddot{Y}(s) + Bs\dot{Y}(s) + KY(s), \quad (6.4)$$

$$F'(s) = Y(s)[ms^2 + Bs + K], \quad (6.5)$$

$$\frac{Y(s)}{F'(s)} = \frac{1}{ms^2 + Bs + K}, \quad (6.6)$$

Continuando o desenvolvimento, tem-se que o modelo da FIG. 6.1 modela apenas um braço robótico sem levar em consideração parâmetros como interação com o ambiente e a comunicação entre os manipuladores. Em (ROSA, 2004b), foi proposto que o modelo mecânico para um sistema de teleoperação constitui-se de duas massas conectadas por uma mola e um coeficiente de elasticidade. [Vide FIG. 6.2].

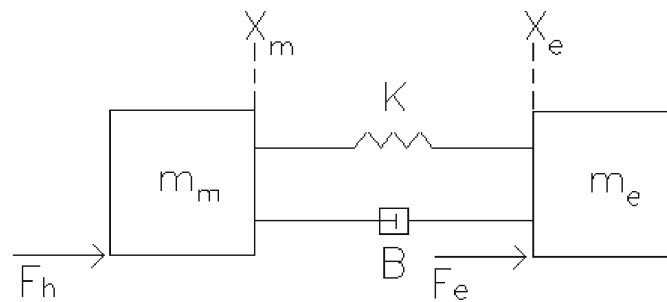


FIG 6.2: Modelo mecânico, simula um sistema de teleoperação para 1 GDL.

F_h é a força exercida pelo operador humano.

F_e é a força exercida pelo escravo quando está em contato com o ambiente.

X_m é o deslocamento do manipulador mestre.

X_e é o deslocamento do manipulador escravo.

m_m é a massa (ou momento de inércia) do manipulador mestre.

m_e é a massa (ou momento de inércia) do manipulador escravo.

As equações para o modelo apresentado na FIG. 6.2. são:

$$m_m \ddot{X} = F_h + B(\dot{X}_e - \dot{X}_m) + K(X_e - X_m), \quad (6.7)$$

$$m_e \ddot{X} = F_e + B(\dot{X}_m - \dot{X}_e) + K(X_m - X_e), \quad (6.8)$$

A FIG. 6.3 mostra o diagrama de blocos que representa o modelo mecânico exposto na FIG. 6.2. Este diagrama representa um modelo simplificado de controle de erro; onde o erro é a diferença de posição entre o momento de inércia (massa) do mestre e do escravo, respectivamente. Uma vantagem deste tipo de arquitetura é que o erro pode ser medido antes do sinal ser realimentado no mestre. Desta forma, é possível controlar a quantidade de força a ser aplicada na realimentação.

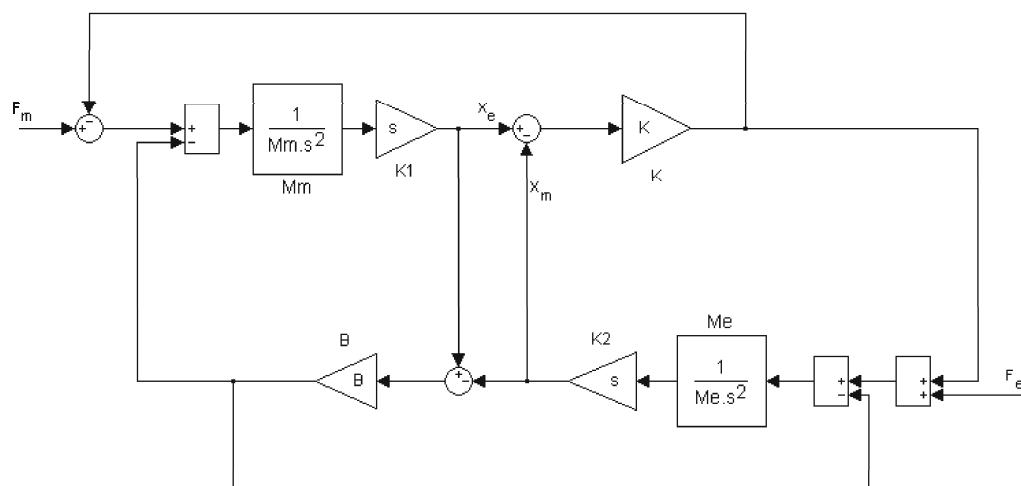


FIG 6.3: Diagrama de blocos para o modelo mecânico da FIG. 6.2.

O modelo mecânico mostrado na FIG. 6.2 representa parcialmente um sistema de teleoperação, pois não leva em consideração alguns parâmetros, como já foi exposto. Dessa forma, este modelo não será adotado neste trabalho; a seguir, será feita uma análise

e o desenvolvimento de um modelo mecânico mais completo do que aqueles apresentados até o momento.

Como um dos objetivos deste trabalho é verificar a interação de força de um manipulador escravo com o ambiente, o modelo a ser adotado deve permitir fazê-lo de uma forma concisa, coesa e ampla. Devido a fatores como estes, será feita uma análise inicial, tomando por base o modelo mecânico exposto na FIG. 6.4. Este é um modelo de segunda ordem e descreve a interação do ambiente com o manipulador, onde m_e , B_e e K_e representam a massa, a viscosidade e a rigidez do manipulador escravo, respectivamente; F_R é a força de reação exercida pelo ambiente no escravo. Neste exemplo, a análise é feita levando em consideração que o manipulador está diretamente em contato com o ambiente. Dessa forma, tem-se: $F_m = F_R$ e $X = X_m = X_e$. O diagrama de blocos para o modelo pode ser visto na FIG. 12.4 [Apêndice 12.6].

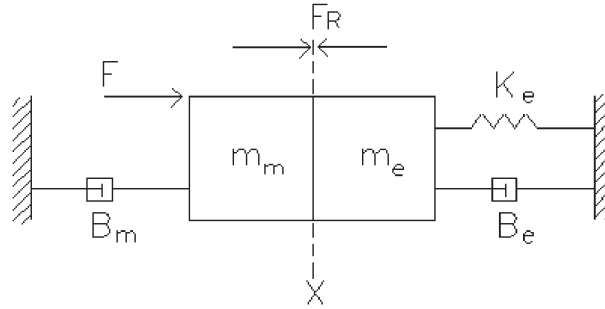


FIG 6.4: Modelo mecânico representando uma situação na qual o mestre está diretamente em contato com o escravo.

As equações para o modelo mecânico mostrado na FIG. 6.4 são dadas por (6.9), (6.10), (6.11) e (6.12)

$$m_m \ddot{X} = F - F_R + B_m(\ddot{X}), \quad (6.9)$$

$$m_e \dot{X} = F_R - B_e \dot{X} - K_e X, \quad (6.10)$$

$$F = (m_m + m_e) \ddot{X} + (-B_m + B_e) \dot{X} + K_e X, \quad (6.11)$$

$$F_R = m_e \ddot{X} + B_e \dot{X} + K_e X, \quad (6.12)$$

O desenvolvimento da função de transferência para o sistema mostrado na FIG. 6.4 encontra-se no apêndice 12.6.

O modelo mecânico apresentado na FIG. 6.4 leva em consideração que os blocos estão diretamente em contato; isto foge ao propósito deste trabalho, pois é necessário usar um controle de posição para que o escravo sinta e transmita ao operador humano as interações com o ambiente. A FIG. 6.5. apresenta um sistema mecânico mais sofisticado do que aquele apresentado na FIG. 6.1. Neste sistema foram incluídos uma mola de constante elástica K_f e uma viscosidade B_f ; X_m e X_e são os deslocamentos do mestre e do escravo, respectivamente.

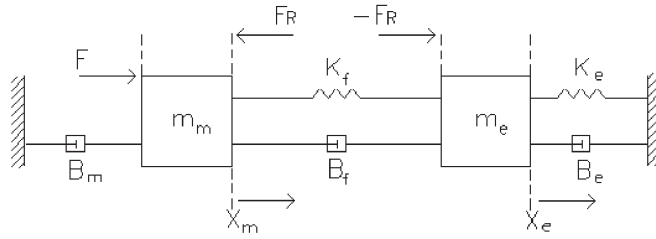


FIG 6.5: Representação de um sistema de teleoperação para 1 GDL, com as respectivas forças que atuam no mesmo.

As equações que modelam o sistema da FIG. 6.5 são dadas pela EQ. (??) e pela EQ. (6.14).

$$m_m \ddot{X}_m = F - B_m \dot{X}_m - K_f(X_m - X_e) - B_f(\dot{X}_m - \dot{X}_e), \quad (6.13)$$

$$m_e \ddot{X}_e = K_f(X_e - X_m) - B_f(\dot{X}_e - \dot{X}_m) - K_e X_e - B_e \dot{X}_e, \quad (6.14)$$

Ainda tomando por base o sistema da FIG. 6.5, encontra-se a EQ. 6.15 e a EQ. 6.17.

$$F - F_R = m_m \dot{X}_m + B_m \dot{X}_m, \quad (6.15)$$

$$F_R = m_e \dot{X}_e + B_e \dot{X}_e + K_e X_e, \quad (6.16)$$

$$F_R = K_f(X_m - X_e) + B_f(\dot{X}_m + \dot{X}_e), \quad (6.17)$$

Escrevendo-se a EQ. 6.15, a EQ. 6.16 e a EQ. 6.17 em termos de função de transferência obtêm-se a EQ. 6.18, EQ. 6.19 e a EQ. 6.20:

$$M = \frac{X_m}{F - F_R} = \frac{1}{m_m s^2 + B_m s} \quad (6.18)$$

$$E = \frac{X_e}{F_R} = \frac{1}{m_e s^2 + B_e s + K_e} \quad (6.19)$$

$$G = \frac{F_R}{X_m - X_e} = B_f + K_f, \quad (6.20)$$

Na FIG. 6.6 é mostrado o diagrama de blocos do modelo mostrado na FIG. 6.5.

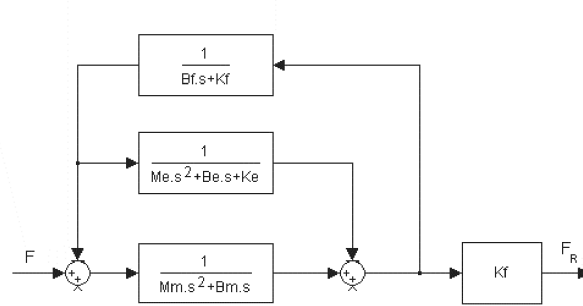


FIG 6.6: Diagrama de blocos do modelo mecânico apresentado na FIG. 6.5.

A função de transferência para o diagrama da FIG. 6.6 é dada pela EQ. 6.21:

$$\frac{F_m}{F} = \frac{MK_f}{1 + MG + EG}, \quad (6.21)$$

Substituindo-se os valores de M, E e G na EQ. 6.21, obtêm-se a a EQ. 6.22.

$$\frac{F_m}{F} = \frac{(m_e s^2 + B_e s + K_e) K_f}{[m_e s^2 + (B_f + B_e) s + (K_f + K_e)](m_m s^2 + B_m s) + (m_e s^2 + B_e s + K_e)(B_f s + K_f)}, \quad (6.22)$$

6.3 ARQUITETURA DE CONTROLE

Como este trabalho não tem por objetivo desenvolver uma arquitetura de controle e sim verificar a interação de força de um manipulador escravo com o ambiente, será adotada uma arquitetura de controle já pré-definida e serão feitas as modificações adequadas a fim de garantir estabilidade e transparência. Estabilidade é a característica do sistema de manter-se íntegro quando exposto a condições adversas. Já o termo transparência pode ser definido como a "habilidade" de o operador do sistema poder sentir diretamente as interações de força com o ambiente, para o caso, um ambiente remoto.

Para o problema da teleoperação, obter um sistema completamente transparente não é possível; o que se tem feito é analisar qual nível de transparência é possível de ser

obtido. Em (LAWRENCE, 1992), o problema é discutido e é proposta uma arquitetura de controle que utiliza 4 (quatro) canais para a transmissão de dados. Este tipo de arquitetura pode ser usado quando se trata de transmissões a longas distâncias na presença de ruídos.

Vários tipos de arquitetura de controle têm sido propostos na literatura ao longo dos anos. Dentre estes pode-se citar: (HANNAFORD, 1989), onde é proposta a utilização de um modelo baseado em controle híbrido para um sistema de teleoperação do tipo mestre-escravo com realimentação háptica. O uso deste tipo de controle representa, de uma forma intuitiva, o desempenho de um sistema de teleoperação ideal, e que pode ser aplicado a vários outros tipos de arquitetura; isto significa que os valores dos parâmetros híbridos podem ser utilizados para comparar desempenho com um sistema ideal.

(ÇAVUŞOĞLU, 2001) e (SHERMAN, 2000), mencionam três tipos distintos de arquiteturas: arquitetura de erro de posição (PERR); arquitetura de realimentação de força cinética (KFF), e a arquitetura de erro de posição com realimentação de força cinética (P+FF). Esta última é um híbrido formado através da junção da PERR com a KFF. O objetivo desta comparação foi o de escolher um tipo de arquitetura que seja capaz de perceber pequenas variações ao interagir com um objeto não rígido. Neste trabalho, não se faz necessário tal tipo de arquitetura, visto que o escravo estará interagindo com objetos rígidos.

A FIG. 6.7 mostra o diagrama de blocos da arquitetura de controle proposta para este trabalho. Esta arquitetura é um modelo adaptado de (LAWRENCE, 1992). F_h^* e F_a^* são as forças exteriores do operador humano e do escravo interagindo com o ambiente, respectivamente; Z_m , Z_e e Z_a são as impedâncias do mestre, do escravo e do ambiente; C_1 e C_2 representam o canal de comunicação entre o mestre e o escravo. Todas as funções, inclusive C_m e C_e serão especificadas a posteriori.

Para este trabalho, o tipo de controle a ser utilizado será o controle híbrido, por permitir ao projetista trabalhar com dois tipos de informação; no caso, força e posição. É importante ressaltar o objetivo de se trabalhar com estes tipos de variáveis. A justificativa para a utilização do controle de força está no fato do operador ter um maior controle quanto à intensidade do torque aplicado; a entrada de posição permite que se aumente ou diminua a velocidade com que o movimento é executado. Como um exemplo prático, pode-se citar o fato de um robô manipulador para lavar janelas de vidro; neste caso, é importante que aquele não exceda a resistência do vidro da janela. Além do problema da força é necessário que o manipulador se movimente (variação de posição) a uma velocidade

variável.

Pelo explicitado até o momento, torna-se óbvio que a utilização de um controle do tipo híbrido adequa-se perfeitamente à solução do problema, que é a de verificar as interações de forças do manipulador escravo em um ambiente remoto.

A nomenclatura do diagrama da FIG. 6.7 é mostrada nas equações 6.23, 6.24, 6.25, 6.26 e 6.27. Os valores destas provêm do modelo mecânico mostrado na FIG. 6.5 e pela adição de outros ganhos como Z_h e Z_a , que são impedância de entrada e impedância de saída, ao sistema.

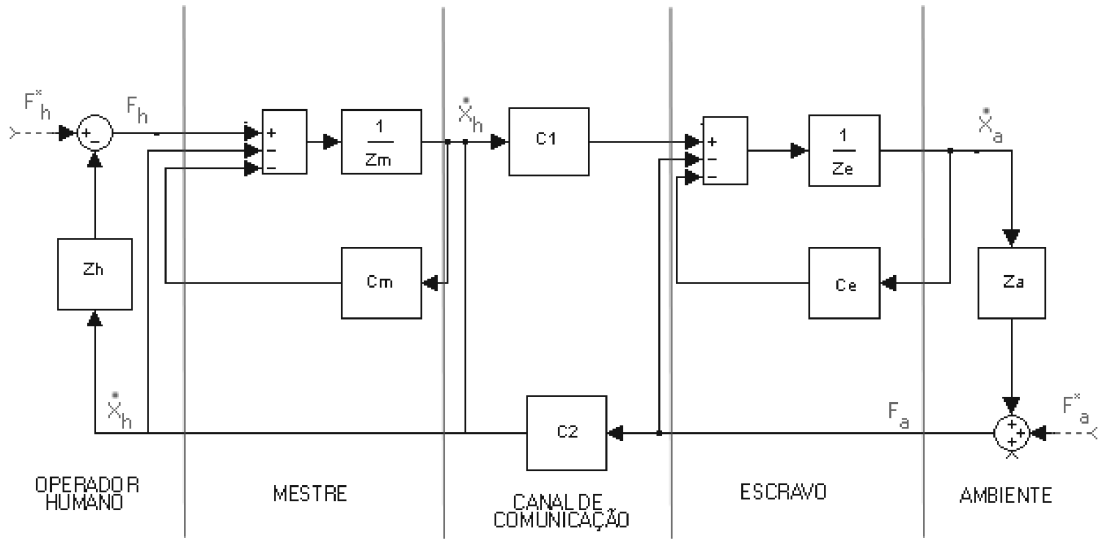


FIG 6.7: Pseudo diagrama de blocos que representa a arquitetura de controle para um sistema de teleoperação de 1 GDL.

$$Z_m = m_m s, \quad (6.23)$$

$$Z_e = m_e s, \quad (6.24)$$

$$C_m = B_m s, \quad (6.25)$$

$$C_e = C_1 = B_e + \left(\frac{K_e}{s}\right), \quad (6.26)$$

$$C_2 = K_f, \quad (6.27)$$

O sistema de teleoperação pode ser modelado como uma rede de quatro terminais ou redes ou modelo de duas portas (two-ports), ou ainda quadripolos. Para este trabalho,

será adotada a terminologia do modelo de duas portas. Este tipo de dispositivo é baseado no teorema de Norton-Thévenin⁷. A FIG. 6.8 mostra esquematicamente o modelo de duas portas (ROSA, 2004c), visto sob o aspecto eletrônico, pois é perceptível que os parâmetros envolvidos são tensão e corrente. O mesmo modelo, sob o aspecto mecânico, pode ser visto na FIG. 6.9; a utilização da FIG. 6.8 faz-se necessária, pois será feita uma analogia entre o sistema elétrico e o sistema mecânico, a fim de determinar a matriz de parâmetros híbridos.

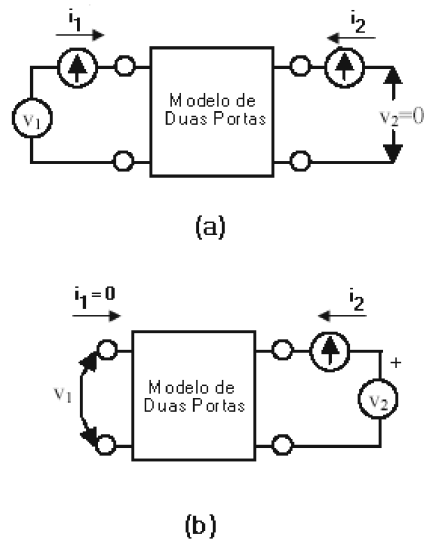


FIG 6.8: Modelo de duas portas, i_1 e i_2 são fontes de corrente, v_1 e v_2 são fontes de tensão.

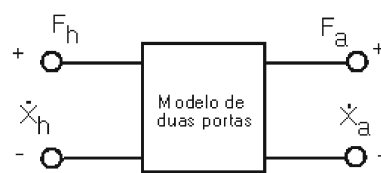


FIG 6.9: Representação de um sistema bilateral de teleoperação através um modelo de duas portas.

A FIG. 6.8 mostra o modelo de duas portas visto sob o aspecto eletrônico, onde os parâmetros v_1 e v_2 são fontes de tensão e i_1 e i_2 são fontes de corrente. Os parâmetros

⁷O teorema de Thévenin, assim como o de Norton, são utilizados para simplificar a análise de circuitos com várias fontes e vários resistores.

podem ser escritos na forma matricial, como pode ser visto na EQ. 6.28.

$$\begin{bmatrix} v_1 \\ i_2 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} \\ h_{21} & h_{22} \end{bmatrix} \begin{bmatrix} i_1 \\ v_2 \end{bmatrix}, \quad (6.28)$$

Como os parâmetros referem-se às fontes de tensão e corrente, são denominados híbridos. As fontes de tensão podem ser representadas pelas forças e as fontes de corrente podem ser representadas pela derivação da posição. Dessa forma, a EQ. 6.28 pode ser reescrita utilizando-se os parâmetros de entrada da FIG. 6.9 assim, tem-se a EQ. 6.29.

$$\begin{bmatrix} F_h \\ \dot{X}_h \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} \\ h_{21} & h_{22} \end{bmatrix} \begin{bmatrix} \dot{X}_a \\ -F_a \end{bmatrix}, \quad (6.29)$$

A determinação dos parâmetros híbridos pode ser feita da seguinte forma: tomando-se a FIG. 6.8 por base, primeiramente faz-se um "curto-circuito" nos terminais de saída. Em outras palavras faz-se $v_2 = 0$. Assim, tem-se que:

$$v_1 = h_{11}i_1, \quad (6.30)$$

daí,

$$h_{11} = \left. \frac{v_1}{i_1} \right|_{v_2=0} \quad (6.31)$$

Ainda, pode-se escrever:

$$i_2 = h_{21}i_1, \quad (6.32)$$

assim,

$$h_{21} = \left. \frac{i_2}{i_1} \right|_{v_2=0} \quad (6.33)$$

Para a obtenção dos parâmetros h_{12} ou h_{22} , abrem-se os terminais de entrada, obtendo-se: $i_1 = 0$. Assim:

$$v_1 = h_{12}v_2, \quad (6.34)$$

daí,

$$h_{12} = \left. \frac{v_1}{v_2} \right|_{i_1=0} \quad (6.35)$$

Por analogia, pode-se obter os parâmetros para a matriz da EQ. 6.29. Logo:

$$h_{11} = \left. \frac{F_h}{\dot{X}_a} \right|_{-F_a=0} \quad (6.36)$$

$$h_{21} = \left. \frac{\dot{X}_h}{\dot{X}_a} \right|_{-F_a=0} \quad (6.37)$$

$$h_{12} = \left. \frac{F_h}{-F_a} \right|_{\dot{X}_a=0} \quad (6.38)$$

$$h_{22} = \left. \frac{\dot{X}_h}{F_a} \right|_{\dot{X}_a=0} \quad (6.39)$$

Após ter definido como obter os parâmetros híbridos da EQ. 6.29, agora pode-se aplicar a definição das equações 6.36, 6.38, 6.37 e 6.39 sobre o diagrama da FIG. 6.7. Através do diagrama da FIG. 6.7, pode-se obter as seguintes equações:

$$\dot{X}_h = \frac{F_h - \dot{X}_a - F_a C_2}{Z_m + C_m}, \quad (6.40)$$

$$\dot{X}_a = \frac{\dot{X}_e C_1 - F_a}{Z_e + C_e}, \quad (6.41)$$

$$F_h = \dot{X}_h (Z_m + C_m) + F_a C_2, \quad (6.42)$$

Substituindo a EQ. 6.42 na EQ. 6.41, tem-se:

$$\dot{X}_a (Z_e + C_e) = \dot{X}_h (Z_m + C_m) + F_a C_2 - F_a + \dot{X}_h C_1, \quad (6.43)$$

$$\frac{\dot{X}_h}{\dot{X}_a} = \frac{Z_e + C_e}{(Z_m + C_m) + C_1} + \frac{F_a (C_2 - 1)}{\dot{X}_a (Z_m + C_m) + C_1}, \quad (6.44)$$

$$h_{11} = \left. \frac{F_h}{\dot{X}_a} \right|_{-F_a=0} = \frac{\dot{X}_h (Z_m + C_m) + F_a C_2}{\dot{X}_a}, \quad (6.45)$$

Rearranjando-se a EQ. 6.45, tem-se:

$$h_{11} = \frac{\dot{X}_h}{\dot{X}_a} (Z_m + C_m) + \frac{F_a C_2}{\dot{X}_a}, \quad (6.46)$$

Substituindo-se a EQ. 6.44 na EQ. 6.46, e fazendo $F_a = 0$, obtém-se a EQ. 6.47 que é o valor do parâmetro, h_{11} , que está sendo procurado.

$$h_{11} = \frac{(Z_e + C_e)(Z_m + C_m)}{(Z_m + C_m) + C_1}, \quad (6.47)$$

Para o cálculo do parâmetro, toma-se por base a EQ. 6.37 e a EQ. 6.44, daí:

$$h_{21} = \left. \frac{\dot{X}_h}{\dot{X}_a} \right|_{-F_a=0} = \frac{(Z_e + C_e)}{(Z_m + C_m) + C_1} + \frac{F_a (C_2 - 1)}{\dot{X}_a (Z_m + C_m) + C_1}, \quad (6.48)$$

Sendo $F_a = 0$, a EQ. 6.48 reduzir-se-á a:

$$h_{21} = \frac{(Z_e + C_e)}{(Z_m + C_m) + C_1}, \quad (6.49)$$

A obtenção dos parâmetros h_{11} e h_{21} foi quase imediata. Para facilitar o cálculo dos parâmetros h_{12} e h_{22} , faz-se necessário um rearranjo na EQ. 6.41, que pode ser visto na EQ. 6.50.

$$-F_a = \dot{X}_a(Z_e + C_e) - \dot{X}_h C_1 - F_h, \quad (6.50)$$

Substituindo-se a EQ. 6.42, na EQ. 6.50, tem-se:

$$\begin{aligned} -F_a &= \dot{X}_a(Z_e + C_e) - \dot{X}_h C_1 - [\dot{X}_h(Z_m + C_m) + F_a C_2], \\ -F_a &= \dot{X}_a(Z_e + C_e) - \dot{X}_h C_1 - \dot{X}_h(Z_m + C_m) - F_a C_2, \\ -F_a(1 - C_2) &= \dot{X}_a(Z_e + C_e) - \dot{X}_h(C_1 + Z_m + C_m), \\ \dot{X}_a(Z_e + C_e) &= -F_a(1 - C_2) + \dot{X}_h(C_1 + Z_m + C_m), \\ \dot{X}_h(C_1 + Z_m + C_m) &= -F_a(1 - C_2) + \dot{X}_a(Z_e + C_e), \end{aligned} \quad (6.51)$$

$$\frac{\dot{X}_h}{-F_a} = \frac{1 - C_2}{C_1 + Z_m + C_m} + \frac{\dot{X}_a(Z_e + C_e)}{-F_a(C_1 + Z_m + C_m)} \quad (6.52)$$

Portanto,

$$h_{22} = \left. \frac{\dot{X}_h}{F_a} \right|_{\dot{X}_a=0} = \frac{1 - C_2}{C_1 + Z_m + C_m} + \frac{\dot{X}_a(Z_e + C_e)}{-F_a(C_1 + Z_m + C_m)}, \quad (6.53)$$

Fazendo $\dot{X}_a = 0$, tem-se:

$$h_{22} = \frac{1 - C_2}{C_1 + C_m + Z_m}, \quad (6.54)$$

O parametro h_{12} é obtido através da EQ. 6.38.

$$h_{12} = \left. \frac{F_h}{F_a} \right|_{\dot{X}_a=0} = \frac{\dot{X}_h}{-F_a}(Z_m + C_m) + \frac{F_a C_2}{-F_a}, \quad (6.55)$$

Substituindo-se a EQ. 6.51 na EQ. 6.55 tem-se:

$$h_{12} = \left[\frac{1 - C_2}{C_1 + Z_m + C_m} + \frac{\dot{X}_a(Z_e + C_2)}{-F_a(C_1 + Z_m + C_m)} \right] (Z_m + C_m) + \frac{F_a C_2}{-F_a}, \quad (6.56)$$

Fazendo $\dot{X}_a = 0$, tem-se:

$$h_{12} = \frac{(1 - C_2)(Z_m + C_m)}{C_1 + Z_m + C_m} - C_2, \quad (6.57)$$

Os valores para as funções de cada bloco que compõe o diagrama de blocos na FIG. 6.7, encontram-se no apêndice 12.5.

Os valores de Z_h e Z_a foram aproximados a fim de levar o sistema à estabilidade, devido ao fato da complexidade que é modelar a interação humana e do ambiente no sistema. A EQ. 6.58 e a EQ. 6.59 mostram as funções encontradas para Z_h e Z_a .

$$Z_h = 0.5s + 20.0 + \frac{285.72}{s}, \quad (6.58)$$

$$Z_a = 100.0s + 10000.0 + \frac{10000.0}{s}, \quad (6.59)$$

Substituindo-se os valores da EQ. 6.58 e da EQ. 6.59 no diagrama da FIG. 6.7 tem-se o diagrama da FIG. 6.10.

A seguir será definida a matriz de variáveis de estado. Para maiores informações sobre as variáveis de estado, referir-se ao Apêndice 12.3.

$$\frac{d_x}{d_t} = Ax + Bu \quad (6.60)$$

$$y = Cx + Du \quad (6.61)$$

$$\frac{d_x}{d_t} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \\ \dot{x}_5 \\ \dot{x}_6 \\ \dot{x}_7 \end{bmatrix} \quad (6.62)$$

$$A = \begin{bmatrix} 0.2371 & -8.297 & -0.0005056 & 6.195e^{-005} & -0.002979 & -7.504e^{-005} & 7.504e^{-005} \\ 0.0007316 & -0.02561 & 5.938e^{-005} & 3.121e^{-009} & -1.501e^{-007} & 8.812e^{-006} & -8.812e^{-006} \\ 283.1 & -9910 & -0.6039 & 0.001208 & -0.05808 & 3.41 & -3.41 \\ 283.1 & -9910 & -0.6039 & 0.001208 & -3.558 & -0.08962 & 0.08962 \\ 0.2371 & -8.297 & -0.0005056 & 6.195e^{-005} & -0.002979 & -7.504e^{-005} & 7.504e^{-005} \\ 0.2371 & -8.297 & -0.0005056 & 6.195e^{-005} & -0.002979 & -7.504e^{-005} & 7.504e^{-005} \\ 0.0007316 & -0.02561 & 5.938e^{-005} & 3.121e^{-009} & -1.501e^{-007} & 8.812e^{-006} & -8.812e^{-006} \end{bmatrix} \quad (6.63)$$

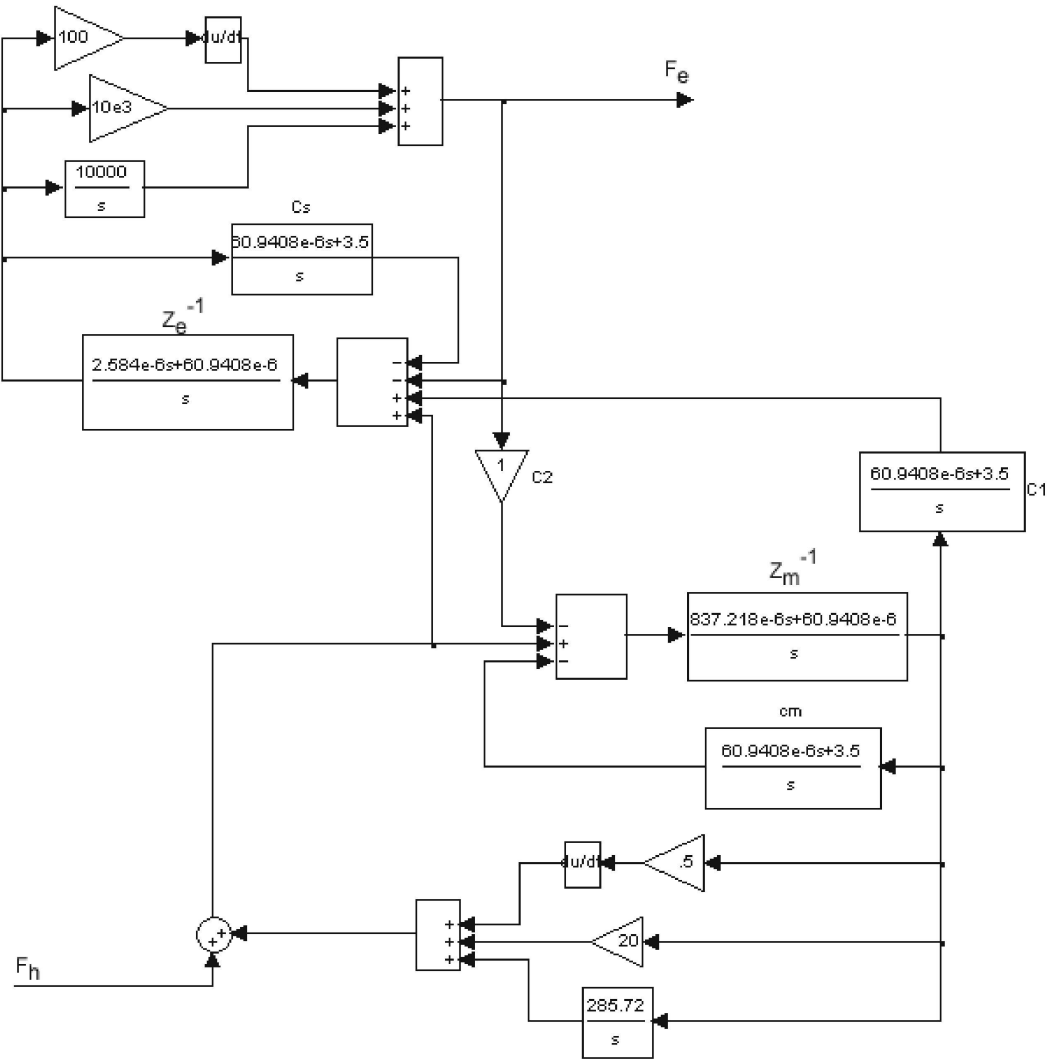


FIG 6.10: Diagrama de blocos da FIG. 6.7 com os respectivos valores das funções de transferência.

$$B = \begin{bmatrix} 0.0008297 \\ 2.561e^{-006} \\ 0.991 \\ 0.991 \\ 0.0008297 \\ 0.0008297 \\ 2.561e^{-006} \end{bmatrix} \quad (6.64)$$

$$C = \begin{bmatrix} 7.316 & 9744 & 0.5938 & 3.121e^{-005} & -0.001501 & 0.08812 & -0.08812 \end{bmatrix} \quad (6.65)$$

$$D = \left[0.02561 \right] \quad (6.66)$$

A função de transferência do sistema pode ser vista na EQ. 6.67, onde $R(s)$ e $Y(s)$, são a saída e entrada do sistema, respectivamente.

$$\frac{F_e}{F_h} = \frac{0.0256s^6 + 0.6297s^5 + 0.6076s^4 + 0.0038s^3 + 0.0003s^2}{1.0000s^6 + 0.3943s^5 + 0.5908s^4 + 0.0038s^3 + 0.0003s^2} \quad (6.67)$$

Nesta seção, foi definida a arquitetura de controle do sistema e ao final foi obtida a função de transferência para o mesmo. O item referente à análise de estabilidade será tratado no capítulo 9.

6.4 CONCLUSÃO

Neste capítulo, foi apresentado o desenvolvimento da arquitetura de controle para o sistema; para isso, partiu-se de um modelo mecânico, e desse modelo foram extraídas equações diferenciais que resultaram em um diagrama de blocos. O próximo passo será fazer a análise de estabilidade do sistema, e testar a arquitetura com diversas funções de entrada, como: degrau, impulso, seno, etc, para verificar o tempo de "convergência" do sistema à estabilidade.

Após ter definido a arquitetura do sistema e as equações cinemáticas e as equações dinâmicas, o próximo passo será definir o protocolo de comunicação do sistema mestre-escravo.

7 COMUNICAÇÃO ENTRE OS SISTEMAS MESTRE-ESCRAVO

Neste capítulo, será mostrado como é feita a comunicação entre os sistemas mestre e escravo, como ocorre o processo de captura, conversão e transmissão dos sinais elétricos.

Inicialmente, é feita uma breve descrição sobre como as redes de computadores são organizadas, depois serão apresentados dois protocolos de comunicação, UDP (User Datagram Protocol) e o TCP (Transmission Control Protocol). Após a descrição dos protocolos, será realizada uma análise sobre a escolha de um deles para servir de suporte para a implementação da comunicação. Em seguida, será feita uma descrição sobre programação com *socket* e, para finalizar, será mostrado o detalhamento da comunicação entre os sistemas.

7.1 INTRODUÇÃO

Um grande problema na implementação de sistemas que se comunicam à distância é o da transmissão dos dados, pois estes devem ser feitos ordenadamente, de forma confiável e às vezes em tempo real. Muitos protocolos de comunicação têm sido propostos, mas em sua maioria estes protocolos são particulares e atendem à aplicações específicas.

Para minimizar a complexidade do projeto de rede, a mesma foi organizada em camadas interdependentes, cada uma utilizando os serviços da camada imediatamente inferior e prestando serviço para a camada imediatamente superior. Com relação à comunicação entre as camadas, para estabelecer uma conversa entre uma camada n de uma máquina com outra utiliza-se o protocolo da camada n . Entende-se por protocolo, o conjunto de regras utilizadas nesta conversa, conforme pode ser visto na FIG. 7.1.

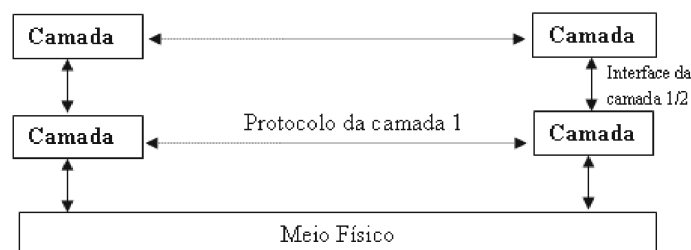


FIG 7.1: Representação da comunicação entre as camadas. Observa-se que a atuação dos protocolos ocorrem em mesmo nível.

À partir deste modelo de camadas, foram definidos modelos de referência com o objetivo de padronizar a implementação de protocolos e permitir a interoperação de equipamentos de fabricantes diferentes. Os 2 (dois) principais modelos de referência são o modelo OSI (Open Systems Interconnection) e TCP/IP (KUROSE, 2003; TANENBAUM, 1996; GASPARINI, 1993). Neste trabalho, não serão feitas referências ao modelo OSI, atendo-se somente ao modelo TCP/IP.

7.2 MODELO DE REFERÊNCIA TCP/IP

O modelo TCP/IP possui apenas quatro camadas [vide FIG. 7.2]. Este modelo teve sua origem na ARPANET⁸. Ainda na FIG. 7.2 pode-se ver que na segunda coluna tem-se a citação de alguns protocolos que atuam nestas camadas. Por exemplo, na coluna 1 linha 1 tem-se a camada de Aplicação e na coluna 2 ainda linha 1 tem-se Telnet e SMTP, que são exemplos de protocolos que atuam nesta camada.

7.2.1 PRINCIPAIS FUNCIONALIDADES DAS CAMADAS DO MODELO TCP/IP

As funcionalidades das camadas do modelo TCP/IP são descritas a seguir:

- Camada Física: É a camada onde acontece a transmissão dos bits através de um meio de comunicação. Resumidamente pode-se dizer que esta camada está relacionada aos componentes mecânicos e eletrônicos envolvidos no processo de transmissão de sinais.
- Camada Internet: Esta camada possibilita que os *hosts* coloquem o pacote na rede e os roteiam até o seu destino. O protocolo IP (*Internet Protocol*) é o responsável por identificar cada máquina na rede, este protocolo também é independente de outras formas de endereçamento que possam existir nos níveis inferiores. Outros protocolos desta camada são: Protocolo de controle e erro: ICMP (*Internet Control Message Protocol*); Protocolo de controle de grupo de endereços IGMP (*Internet Group Management Protocol*); Protocolos de controle de informações de roteamento.
- Camada de Transporte: É a camada onde estão localizados os protocolos que realizam as funções de transporte de dados fim-a-fim, ou seja, leva-se em consideração

⁸Foi criada nos anos 60 pelo Departamento de Defesa dos Estados Unidos. A partir dela começou-se a estimular pesquisas sobre redes de computadores em departamentos de computação em diversas universidades americanas.

apenas a origem e o destino da comunicação, dessa forma, não há necessidade de preocupar-se com elementos intermediários. A camada de transporte possui dois protocolos que são o UDP (User Datagram Protocol) e TCP (Transmission Control Protocol). O protocolo UDP realiza apenas a multiplexação para que várias aplicações possam acessar o sistema de comunicação de forma coerente. O protocolo TCP realiza, além da multiplexação, uma série de funções para tornar a comunicação entre origem e destino mais confiável. São responsabilidades do protocolo TCP: o controle de fluxo, o controle de erro, a sequenciação e a multiplexação de mensagens.

- Camada de aplicação: A camada de aplicação dentre as camadas é a que fornece uma interface quase direta com o usuário. Dentre os protocolos desta camada pode-se citar: FTP, HTTP, Telnet, SMTP, IMAP, NFS, ICQ, SNMP, DNS, DHCP e outros

APLICAÇÃO	Telnet, SMTP, DHCP, HTTP, etc.
TRANSPORTE	TCP, UDP
REDE	IP
FÍSICA	802.3, FDDI, etc.

FIG 7.2: Modelo de referência TCP/IP.

Neste ponto, será dada mais ênfase à Camada de Aplicação e de Transporte, pois serão estas duas camadas que estarão diretamente relacionadas a este trabalho. A Camada de Aplicação é o mais alto nível da arquitetura TCP/IP, pois é onde se pode desenvolver as aplicações que utilizam os protocolos desta camada e das camadas subjacentes. As aplicações mais conhecidas que atuam nesta são: correio eletrônico que utiliza o SMTP (Simple Mail Transfer Protocol); Emulação de terminais que utilizam Telnet e os Browsers que utilizam o HTTP (Hiper Text Transfer Protocol). A camada de transporte é subjacente à de aplicação e ambas possuem uma correspondência quase direta.

Esta camada tem por função fornecer às aplicações um processo de transporte de dados. Nesta camada, encontram-se os protocolos TCP e UDP, os quais serão detalhados nos próximos itens.

7.3 USER DATAGRAM PROTOCOL - UDP

O protocolo UDP está definido na RFC 768 e atua na Camada de transporte do modelo TCP/IP. É um protocolo simplificado, sendo que tem como função a multiplexação/demultiplexação e verificação de erros. A FIG. 7.3 mostra a estrutura do segmento UDP.

Porta de origem UDP	Porta de destino UDP
Tamanho da mensagem UDP	Soma de verificação de erro (CHECKSUM)
DADOS	

FIG 7.3: Formato do segmento UDP.

- Porta de origem UDP:
 - Este campo, é composto por 16 bits, é onde se especifica o número de porta⁹ da aplicação de origem.
- Porta de destino UDP:
 - Este campo é utilizado para especificar o endereço da porta adotado pela aplicação de destino.
- Tamanho da mensagem UDP:
 - Este campo, é composto de 16 bits e está relacionado ao tamanho da mensagem.
- Soma de verificação de erro (CHECKSUM):
 - O campo de verificação de erro é um campo opcional e sua utilização depende do protocolo da aplicação.

⁹Porta é uma interface entre a camada de aplicação e a camada de transporte, também, pode ser descrita como uma interface de programação na qual as aplicações de rede são disponibilizadas na Internet.

- Dados:

- Este campo trata do armazenamento dos dados da mensagem; seu tamanho varia de acordo com a necessidade da aplicação que o utiliza.

O conceito de porta está associado a um serviço específico. Assim, cada um dos processos comunicantes na conexão dispõe de uma porta associada (um valor de 16 bit). Alguns serviços (que fazem uso de protocolos específicos) são tipicamente acessíveis em portas fixas, que se encontram no intervalo de 1 à 1023. Além destas, existem outras duas classes de portas, registradas e privadas ou dinâmicas. As portas com endereços fixos são estabelecidas pela Internet Assigned Numbers Authority (IANA). Estas portas encontram-se em escuta passiva a serviços como: HTTP, SSH, FTP, etc.

7.3.1 CARACTERÍSTICAS DO PROTOCOLO UDP

À seguir, são dadas algumas características do protocolo UDP.

- a) Não existe estabelecimento de conexão. O UDP envia mensagens sem a necessidade de uma comunicação formal que preceda este fato. Sendo assim, ele também não possui o problema de atrasos devido ao estabelecimento de conexão.
- b) Não existe controle de congestionamento. Sendo assim, ele não precisa manter parâmetros de controle de congestionamento, parâmetros numéricos de seqüência e de reconhecimento.
- c) Não existe grande "overhead" no cabeçalho do pacote, pois o UDP anexa, além dos dados da mensagem, somente 8 bytes de cabeçalho para cada segmento.
- d) Não existe limitação para a taxa de transmissão de dados. Dessa forma, o limite superior para a transmissão de dados fica limitado somente pela taxa em que a aplicação está trabalhando e pela vazão do canal de transmissão. É importante lembrar que se a rede encontra-se congestionada, dados podem ser perdidos devido a limitações dos roteadores.

O próximo item trata da transmissão confiável de dados pela utilização do protocolo TCP.

7.4 TRANSMISSION CONTROL PROTOCOL - TCP

O TCP ou protocolo de controle de transmissão está definido nos RFCs (Request for Comments) 793, 1122, 1323, 2018 e 2581 (KUROSE, 2003). Diferente do UDP, o TCP é um protocolo mais completo devido a implementações de serviços como orientação à conexão, mecanismo de controle de congestionamento e mecanismo de retransmissão. A transferência de dados é feita em um canal bidirecional. A FIG. 7.4 mostra a estrutura do segmento TCP. A descrição dos campos são as seguintes:

Porta de origem TCP				Porta de destino TCP				
Número de seqüência (32 bits)								
Número de reconhecimento (32 bits)								
Tamanho Cabeçalho	Reservado	U R G	A C K	P S H	R S T	S Y N	F I N	Tamanho da janela (16 bits)
Soma de verificação de erro (CHECKSUM)						Ponteiro para dados urgentes		
OPÇÕES						PADDING		
DADOS								

FIG 7.4: Formato do segmento TCP.

- Porta de origem e destino TCP:
 - Estes campos obedecem à mesma descrição do UDP.
- Número de seqüência:
 - Este campo atua no controle de fluxo e refinamento do protocolo TCP, numerando os dados no pacote a ser transmitido.
- Número de reconhecimento:
 - Este campo refere-se à confirmação dos dados recebidos. O número de reconhecimento é gerado a partir do número de seqüência, adicionando um valor. Dessa forma o número de reconhecimento gera o próximo número seqüência que o transmissor deve enviar.
- Tamanho do cabeçalho:

- Este campo é composto por 4 bits e refere-se ao tamanho do cabeçalho do TCP. Esse campo torna-se importante porque o campo Opções é variável. Devido a este fato, um receptor pode efetuar a leitura correta do cabeçalho TCP.
- Reservado:
 - Este campo possui tamanho igual a 16 bits e está reservado para futuras implementações.
- URG:
 - Quando este campo está com o valor 1, significa que ele deve ser lido, pois contém dados válidos.
- ACK:
 - Quando este campo está com o valor 1, significa que o cabeçalho envia uma confirmação válida.
- PSH:
 - Quando este campo possui valor igual a 1, significa que o emissor avisa o receptor que os dados podem ser entregues diretamente à aplicação.
- SYN:
 - Quando possui valor igual a 1, significa que um host está fazendo um pedido de conexão a outro TCP.
- FIN:
 - Quando este campo possui valor igual a 1 significa que o host está fazendo um pedido de encerramento de conexão.
- Tamanho da janela:
 - Este campo especifica o tamanho da janela que o emissor pode trabalhar. Numa transmissão o valor assumido para este campo é sempre o valor do menor tamanho da janela dos hosts envolvidos na transmissão.
- Soma de verificação de erro (CHECKSUM):

- Este campo é composto por 16 bits e é utilizado na checagem de erros.
- Ponteiro para dados urgentes:
 - Este campo possui 16 bits, e seu objetivo é sinalizar a aplicação da existência de uma informação urgente contida no campo DADOS, que tenha prioridade sobre as outras informações.
- Opções:
 - Este campo possui tamanho variável, de acordo com a opção utilizada. Tem por função ajudar na negociação entre dois processos comunicantes TCP. Uma das facilidades utilizadas na negociação é o MSS (Maximum Segment Size), que define o tamanho máximo dos segmentos TCP.
- Padding (preenchimento de espaços):
 - Este campo possui tamanho variável e é utilizado para se garantir que o comprimento do cabeçalho seja sempre um múltiplo inteiro de 32 bits.
- Dados:
 - Este campo é destinado a armazenar os dados que serão transportados.

No próximo item, será feita uma breve abordagem sobre como ocorre o estabelecimento e o encerramento de uma conexão TCP.

7.4.1 ESTABELECIMENTO DA CONEXÃO

O TCP fornece um serviço orientado à conexão sobre os pacotes trocados na rede. Um serviço orientado à conexão significa que um canal virtual foi estabelecido entre dois pontos finais. Existem três fases na conexão virtual. São elas: estabelecimento da conexão, transferência de dados e encerramento da conexão.

Para dois hosts se comunicarem utilizando TCP, primeiramente, eles necessitam estabelecer uma conexão para a troca de mensagens; este processo é conhecido como aperto de mão triplo, do inglês (*three-way handshake*). A FIG. 7.5 mostra como é feito o processo. Inicialmente, o host A inicia a conexão enviando um segmento TCP, com o bit SYN ativo e com um número de seqüência que está representado pela variável θ no campo número de seqüência. Em seguida, o host B recebe o segmento SYN, executa-o e responde com o seu próprio segmento TCP. A resposta do host B contém o bit de controle SYN ativado

e o seu número de seqüência inicial representado pela variável θ . O host B também ativa o bit de controle ACK para indicar o próximo byte esperado do host A que pode conter dados iniciais com o número de seqüência $\theta + 1$. Quando o host A recebe o número de seqüência inicial do host B e ACK, aquele finaliza a conexão e envia um ACK final para o host B. Neste caso, o host A ativa o bit de controle ACK e indica o próximo byte vindo do host B pela colocação de um número de reconhecimento $\delta + 1$ no campo ACK.

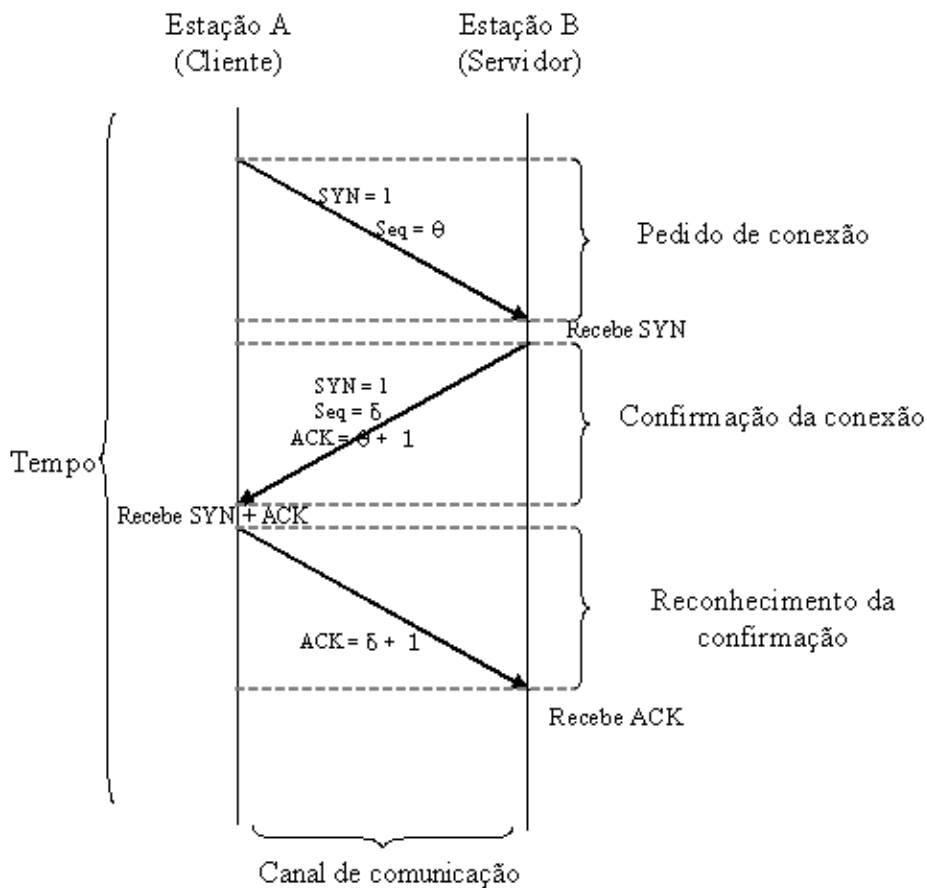


FIG 7.5: Representação do estabelecimento de uma conexão TCP entre duas estações.

7.4.2 ENCERRAMENTO DA CONEXÃO

Para encerrar a conexão, quatro fases são necessárias. Como o TCP trabalha em um canal bidirecional, cada encerramento necessita ser feito de forma independente, conforme observa-se na FIG. 7.6. Analisando a FIG. 7.6 observa-se que a aplicação executada no host A sinaliza para o TCP finalizar a conexão. Isto gera o primeiro segmento FIN do host A para o host B. Quando o host B recebe o segmento inicial FIN, ele imediatamente

reconhece o segmento e notifica a aplicação destino à finalização da requisição. Uma vez que a aplicação no host B também decida encerrar a conexão, essa envia seu segmento FIN, que o host A irá processar e responder com um ACK.

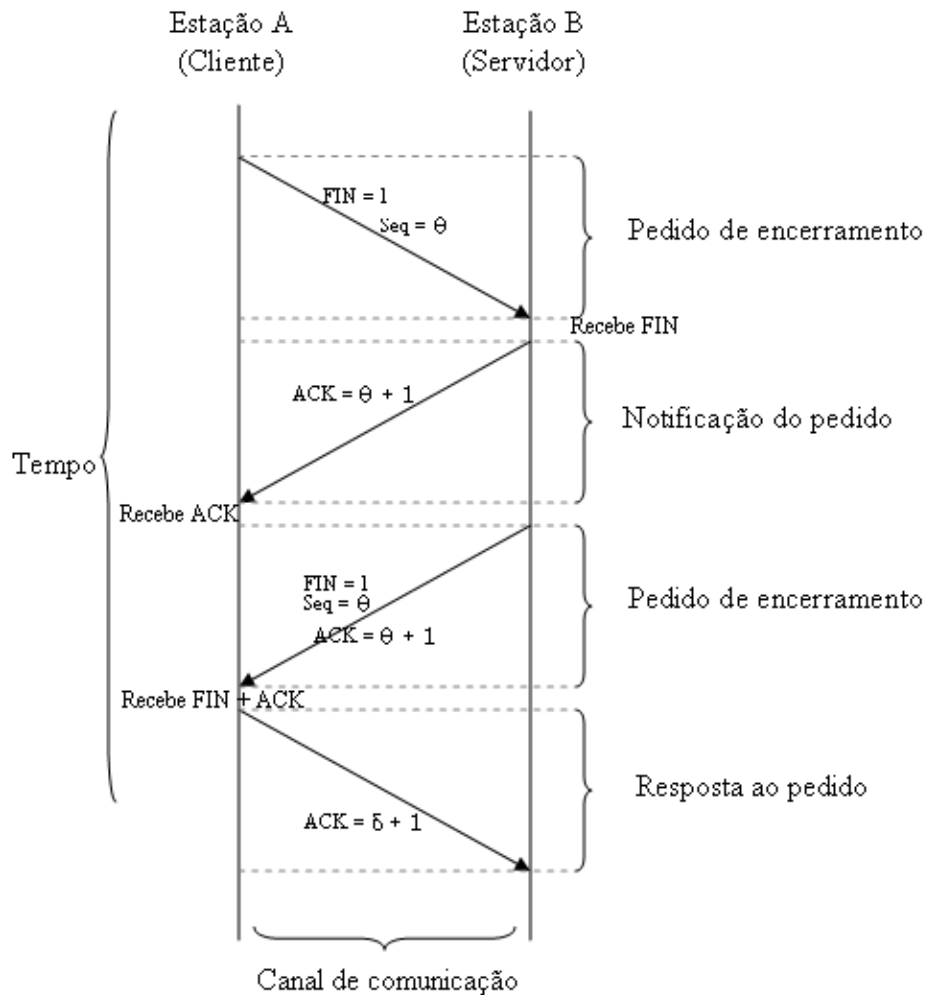


FIG 7.6: Descrição do encerramento de uma conexão TCP entre duas estações. As setas indicam a sequência em que o processo ocorre.

7.4.3 CARACTERÍSTICAS DO TCP

A seguir serão descritas algumas características do TCP e na sessão seguinte será feita a definição do protocolo de comunicação proposto para este trabalho.

- Os dados são enviados e recebidos em fluxo (stream).
- Os eventuais erros de transmissão são corrigidos através de uma soma verificadora dos dados transmitidos (checksum). Assim o destinatário verifica a soma com o

cálculo dos dados recebidos.

- Existe o controle de fluxo que é feito através de um sistema de baferização (buffering) denominada janela deslizante. O TCP envia uma série de pacotes sem aguardar o reconhecimento de cada um deles. Na medida em que recebe a confirmação de cada bloco enviado, atualiza o buffer, caso a confirmação seja positiva, ou reenvia, caso a confirmação seja negativa ou caso não haja uma confirmação após um tempo de espera (timeout);
- É garantido que os dados cheguem ao destino na mesma ordem em que foram enviados.
- Assim como no UDP, existe o conceito de portas, onde números são definidos para multiplexar o acesso ao TCP por várias aplicações residentes na mesma máquina.
- Uma conexão TCP é definida por um conjunto de quatro, parâmetros que são:
 - O número IP do Cliente;
 - O número IP do Servidor;
 - O número de porta do Cliente e;
 - O número de porta do Servidor.

Se comparado ao UDP, o TCP é um protocolo complexo, pois implementa vários serviços que inexistem no UDP. Sendo assim, as características listadas nesta sessão o tornam um bom candidato para compor um sistema de teleoperação. Na próxima sessão será feita a definição do protocolo de comunicação para o sistema proposto.

7.5 DEFINIÇÃO DO PROTOCOLO DE COMUNICAÇÃO DO SISTEMA MESTRE-ESCRAVO

O protocolo de comunicação atuará no nível da camada de aplicação e utilizará os protocolos e os recursos das camadas subjacentes que o protocolo TCP/IP fornece. O protocolo deve ser projetado de forma que dois manipuladores robóticos de n-GDL comuniquem-se a longas distâncias. Algumas premissas são desejáveis para que isto aconteça:

- Segurança com relação à taxa de transmissão;
- Ininterruptibilidade dos processos comunicantes;

- Garantia de entrega da informação;
- Garantia da ordem de entrega da informação;
- Transmissão em tempo real.

Analisando os requisitos estabelecidos para a implementação da comunicação entre os sistemas, verifica-se que o protocolo TCP obedece em 80% aos requisitos, deixando a desejar no item referente à comunicação em tempo real. Dessa forma, o protocolo da camada de transporte a ser escolhido para a aplicação (protocolo de comunicação entre os sistemas robóticos) será o TCP. A princípio, o item que trata da comunicação em tempo real poderá ser compensado com um canal de comunicação que trabalhe com alta largura de banda.

Observando a FIG. 7.7 verifica-se que o sistema compõe-se de dois manipuladores robóticos, um mestre e um escravo. O mestre gera um sinal analógico vindo da execução de um movimento por um operador e é transmitido a uma estação (computador). A partir daí, o sinal é convertido para digital. Um programa residente nesta estação, chamado de *programa mestre*, capta o sinal e o envia para uma outra estação, através de uma conexão TCP. Nesta outra estação, existe um programa chamado de *programa escravo*, que fica em constante alerta, à espera de um comando vindo do programa mestre. O programa escravo captura o sinal vindo do mestre, converte-o de digital para analógico e, então, o transmite para o manipulador escravo. O manipulador escravo executa o movimento enviado pelo mestre, um sinal é gerado, este é retransmitido para o programa escravo que o retransmite para o programa mestre. O programa mestre de posse do sinal o envia para o manipulador mestre, dando assim a sensação do operador humano estar agindo no próprio manipulador escravo.

É fácil perceber que os sistemas trabalham de forma semelhante a um sistema Cliente/Servidor, onde o cliente pode ser representado pelo programa mestre e o servidor, pelo sistema escravo, visto que este deve sempre estar à espera de uma requisição. Dessa forma, a comunicação entre os sistemas foi implementada utilizando este modelo. A FIG. 7.8 mostra o pseudocódigo do programa mestre e do programa escravo.

A leitura da FIG. 7.8 deve ser feita da seguinte forma: quando o programa escravo é ativado, ele inicialmente cria uma porta virtual e atribui um valor a esta. Em seguida, fica em modo de espera aguardando por uma requisição, neste caso, uma requisição do programa mestre. No lado do cliente ou do programa mestre, este, quando é ativado, também cria uma porta e, em seguida, é estabelecida uma conexão TCP com o programa

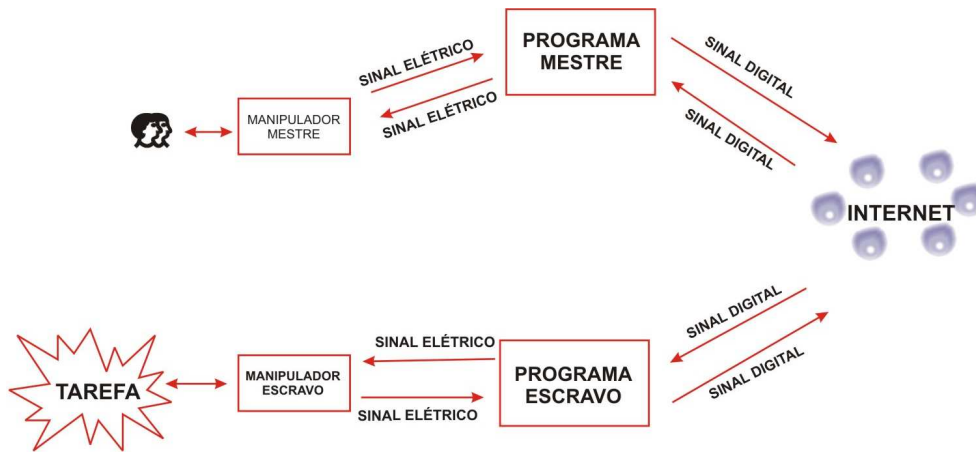


FIG 7.7: Composição do sistema: fluxo da informação.

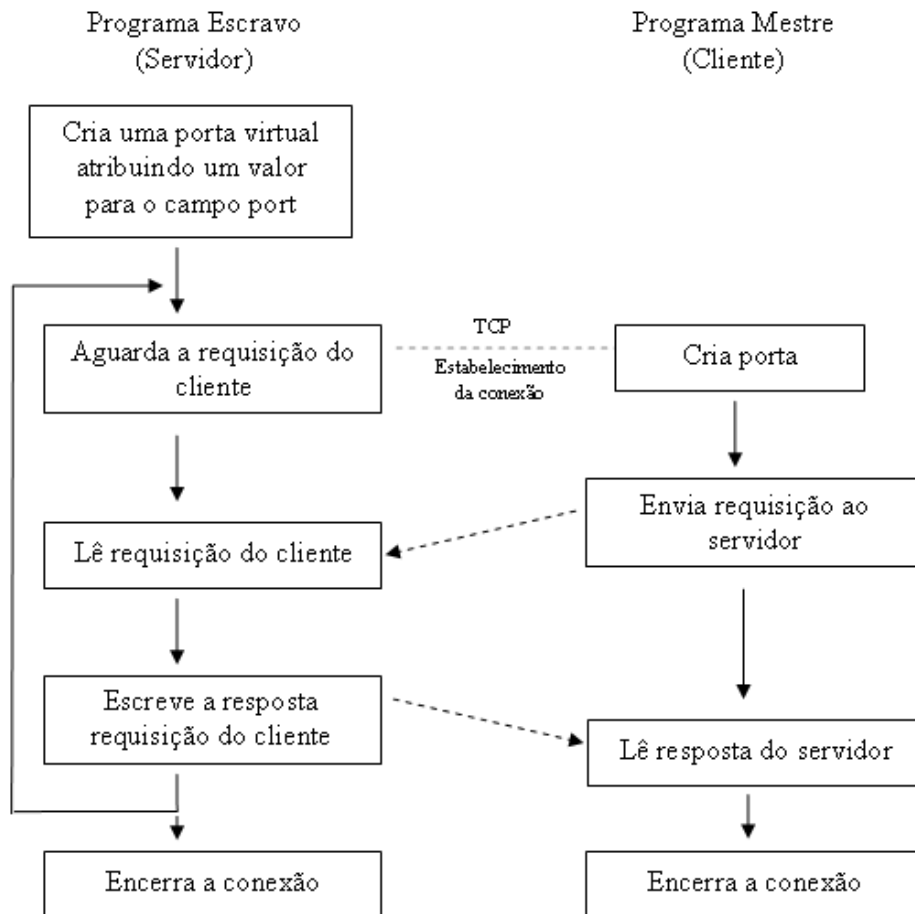


FIG 7.8: Fluxograma representando a comunicação entre os sistemas mestre-escravo.

escravo, cuja porta tem de ser conectada. Após estabelecida a conexão, o programa mestre envia um comando ao programa escravo. Este comando é lido e, em seguida, o

programa escravo escreve uma resposta para a requisição e volta ao estado de espera. Do lado do cliente, a resposta é lida e em seguida a conexão é encerrada.

7.6 CONCLUSÃO

Neste capítulo, foi apresentada uma introdução a respeito da arquitetura de rede de computadores TCP/IP. Foram analisados dois protocolos da camada de transporte deste modelo. Foi proposto um sistema de comunicação para que dois manipuladores robóticos se comuniquem a longas distâncias. Com relação à arquitetura de redes, o modelo TCP/IP está dividido em quatro camadas: física, rede, transporte e aplicação. Para este trabalho, foram estudados dois protocolos da camada de transporte, o UDP e o TCP. O UDP é um protocolo simples, de fácil utilização e que fornece bons resultados para aplicações que envolvam o conceito de tempo real, mas que não sofram alterações com perdas de dados; por exemplo: transmissão de áudio e vídeo descomprimido. O TCP, por sua vez, mostra-se como um protocolo mais complexo, que possui características como: estabelecimento de um canal de comunicação confiável, mecanismo de retransmissão de dados, entrega de dados na ordem em que estes lhe foram confiados e controle de congestionamento. Contudo, o TCP possui o problema de não conseguir transmitir os seus pacotes a uma taxa constante, devido ao mecanismo de controle de congestionamento. Para a implementação do sistema de comunicação foram definidos alguns pré-requisitos, como:

- A transmissão dos dados deve ser feita a uma taxa constante;
- Não pode haver interrupção enquanto os processos estejam se comunicando;
- Os dados devem ser entregues na ordem em que estes lhe foram confiados e;
- A transmissão dos dados deve ser feita em tempo real.

Depois de estabelecidos os pré-requisitos e observado o comportamento do sistema, optou-se por implementar a comunicação destes baseando-se no modelo *Cliente/Servidor*, onde o sistema mestre seria o cliente e o sistema escravo seria o servidor. Com relação à implementação física do sistema foi utilizado o conceito de programação de soquets (do inglês socket programming). O protocolo escolhido para atuar ao nível da camada de transporte foi o TCP, pois ele atendeu aos pré-requisitos em sua grande maioria. Em princípio, fez-se a comunicação utilizando o modelo de cliente servidor. O próximo passo a ser tomado é tentar implementar um protocolo que torne a aplicação independente

da utilização de um canal de comunicação de alta capacidade. Outras soluções devem ser testadas, como: a utilização do protocolo UDP, pela implementação de serviços adicionais, na camada de aplicação, àquilo que o UDP provê; e o RTP (Real Time Protocol), que é um protocolo utilizado em aplicações de tempo real.

8 DESENVOLVIMENTO DO PROTÓTIPO FÍSICO E DO SIMULADOR DO SISTEMA

Neste capítulo, será mostrado como foi desenvolvido o dispositivo físico que permite conectar dois manipuladores reais, para que estes possam ser teleoperados via web. Em seguida será mostrado o simulador gráfico desenvolvido para o sistema de teleoperação proposto neste trabalho.

8.1 PROTÓTIPO DO SISTEMA

A simulação por hardware baseou-se na utilização de um manipulador de 1 GDL descrito em (FERNANDES, 2004; ROSA, 2004c). A FIG. 8.1 mostra a representação do manipulador feita em 3D, este é composto de dois manipuladores idênticos, um manipulador mestre e um manipulador escravo, compostos por um braço robótico que foi posicionado horizontalmente sobre uma plataforma de acrílico [vide FIG. 8.2], e um sistema de engrenagens que são movidas por um servo-motor.

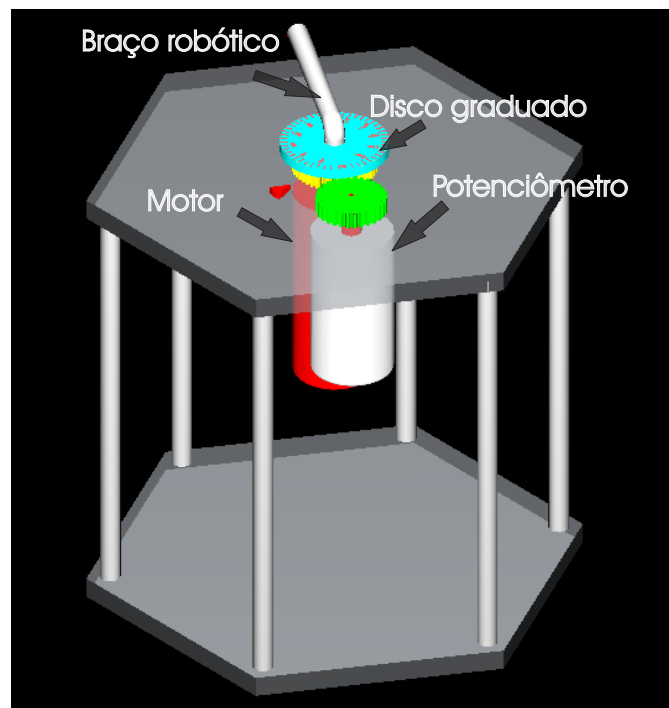


FIG 8.1: Representação em 3D para o protótipo real de 1 GDL.

Para interligar o protótipo via web, foi projetada e construída uma placa de captura e escrita de sinais, que se comunica via porta paralela do computador. A FIG. 8.3, (que é uma variação da FIG. 12.3 apresentada no Apêndice 12.4), mostra esquematicamente a posição em que o dispositivo de captura deverá ser acoplado aos manipuladores. A posição das placas deve ser aquela em que os blocos são nomeados de conversor A/D e D/A.

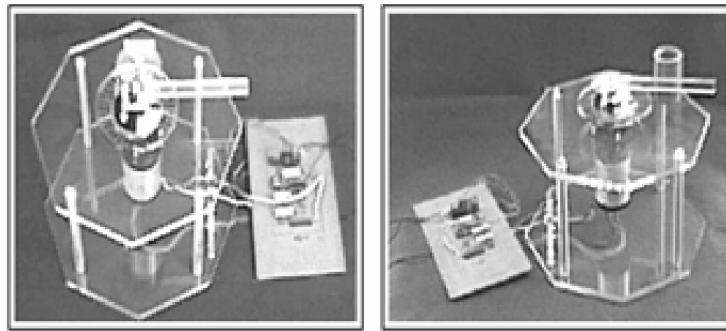


FIG 8.2: Modelo real do sistema desenvolvido para 1 GDL.

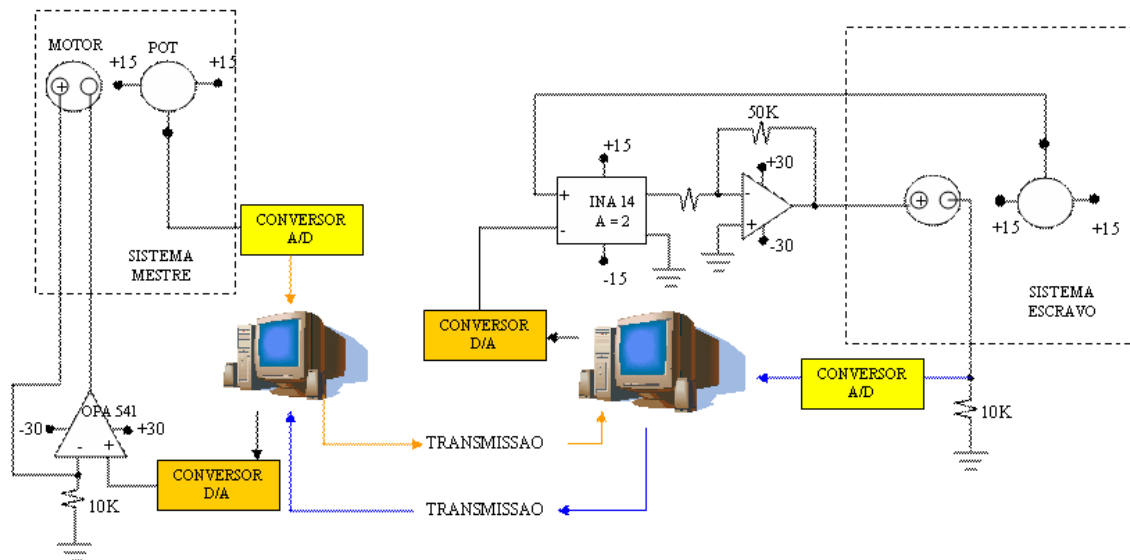


FIG 8.3: Pseudo diagrama representando as ligações entre os sistemas mestre e escravo e a comunicação entre ambos.

O manipulador opera com um intervalo de tensão de $\pm 15V$. Quando um operador humano aplica uma força no braço robótico este gera um sinal elétrico. Este sinal é cap-

turado pela placa de captura/escrita de dados, que o converte para digital e o escreve na porta paralela do computador. O simulador desenvolvido captura o sinal e o transmite via web, a um outro computador que hospeda o programa escravo e este escreve o sinal enviado na porta paralela do computador; assim, o sinal é lido pela placa de captura/escrita de sinais, que converte o sinal para analógico e o entrega ao manipulador escravo, que executa o movimento desejado. Então, a placa é realimentada, o sinal é capturado pelo programa escravo e entregue ao programa mestre, que o entrega ao manipulador mestre. Este processo foi descrito detalhadamente na seção 7.5. O protótipo real pode ser visto na FIG. 8.2.

8.1.1 ESPECIFICAÇÃO DO DISPOSITIVO DE CAPTURA/ESCRITA DE DADOS

O dispositivo de captura de dados, como já foi mencionado na seção 7, foi projetado para trabalhar com um intervalo de tensão variando entre -15 V a +15 V. Para o detalhamento do sistema tomou-se por base o sistema mostrado no Apêndice 12.4. O esquema elétrico do dispositivo é mostrado na FIG. 8.5. O sistema foi projetado para receber e enviar dados através da utilização da porta paralela do computador, [vide FIG. 8.6]. Ao todo foram construídas duas placas de leitura/escrita de dados. A FIG. 8.4 mostra resumidamente como ocorre o fluxo do sinal do mestre para o escravo e vice-versa. Das duas placas construídas, uma foi acoplada ao sistema mestre (placa-mestre) e outra, acoplada ao sistema escravo (placa-escravo). Na FIG. 8.4 (A) a leitura de tensão de saída do mestre é feita pela placa-mestre e transmitida ao computador executando o programa mestre que transmite o sinal capturado pela rede. Na FIG. 8.4 (B), a placa-escravo captura um sinal digital vindo do mestre e o converte para analógico. Na FIG. 8.4 (C), a placa-escravo coleta a tensão do motor, converte-a de analógico para digital e a transmite através da rede para o computador-mestre. Na FIG. 8.4 (D), o sinal provindo do escravo é convertido de digital para analógico, acionando assim o motor mestre.

8.1.2 FUNCIONAMENTO DA PLACA DE CAPTURA/ESCRITA DE DADOS

Nesta seção, será detalhada uma parte do hardware construído para o desenvolvimento do sistema de teleoperação proposto. Este hardware consiste na construção de duas placas de captura/escrita de dados. Em síntese uma das placas deve ser conectada ao sistema mestre e outra deverá estar conectada ao sistema escravo, sendo ambas responsáveis por converter os sinais analógicos em digitais e vice-versa.

8.1.2.1 ENTRADA ANALÓGICA

O conversor AD 7821 é um conversor analógico para digital de 8 bits, que converte para uma palavra digital de oito bits a tensão que lhe é apresentada no pino 1. Para que possa operar no modo bipolar, faz-se necessário alimentar os pinos conforme mostrado no diagrama esquemático da FIG. 8.5. Neste tipo de operação, a tensão que pode ser aplicada ao conversor varia no intervalo $[-2,5V, 2,5V]$. Desta forma, para que o conversor possa operar com tensões no intervalo $[-10V, 10V]$, faz-se necessário utilizar um circuito com o amplificador operacional com um ganho ajustado para $V_{out} = -\left(\frac{R2}{R1}\right) \cdot V_{in}$ [vide LM741(2) na FIG. 8.5], onde o ganho $R2/R1$ deve ser ajustado de modo que se tenha $V_{out} = -\frac{V_{in}}{4}$. O amplificador operacional LM741(1) foi utilizado somente para isolar o circuito cuja tensão será a medida do circuito de medição. Na configuração em que foi montado, o conversor AD 7821 efetuará uma conversão todas as vezes que ocorrer uma borda negativa no seu pino 6. Isto é obtido através do pino 14 da porta paralela. Como o valor do pino 14 é invertido, e seu valor na inicialização é alto (5V), para que possa ser obtida a borda negativa, primeiramente deve-se escrever o bit 1 no pino 14 e, logo em seguida, o bit 0. Concluído o comando para a conversão da tensão apresentado ao AD 7821, o processo de leitura da tensão convertida é efetuado de 4 em 4 bits; para isso, os 4 bits mais significativos da palavra (D7-D4) são conectados às entradas 1A e os bits menos significativos (D3-D0) são conectados às entradas 2A do buffer de 3 estados 74LS244. A seleção dos bits que serão lidos pelo computador é efetuada através de um comando enviado ao pino 1 da porta paralela, da seguinte forma: um sinal baixo (0 volts) habilitará a entrada 1G do buffer e, como o pino 2 também está conectado à entrada inversora do 74LS04, a entrada 2G do buffer estará desabilitada. Desta forma, os bits mais significativos poderão ser lidos através da porta paralela, pelos pinos 10, 12, 13 e 15. Um sinal alto (5 volts) desabilitará a entrada 1G e habilitará a entrada 2G, permitindo a leitura dos bits menos significativos.

8.1.2.2 SAÍDA ANALÓGICA

O conversor AD 7224 é um conversor digital para analógico de 8 bits. Na FIG. 8.5, ele pode ser visto operando em conjunto com o amplificador operacional LM 741(3), o circuito é capaz de fornecer uma tensão de saída que varia no intervalo de $[-2,5V, 2,5V]$. O circuito logo a seguir, composto pelo amplificador operacional LM 741(4), está configurado para um ganho $V_{out} = -4V_{in}$, de modo que a saída analógica do sistema possa variar em um intervalo de $[-10V, 10V]$. Para se obter um valor de tensão na saída

analgógica do sistema, basta escrever nos bits de dados da porta paralela (pinos 2 a 9, sendo o pino 9 o mais significativo), a palavra desejada.

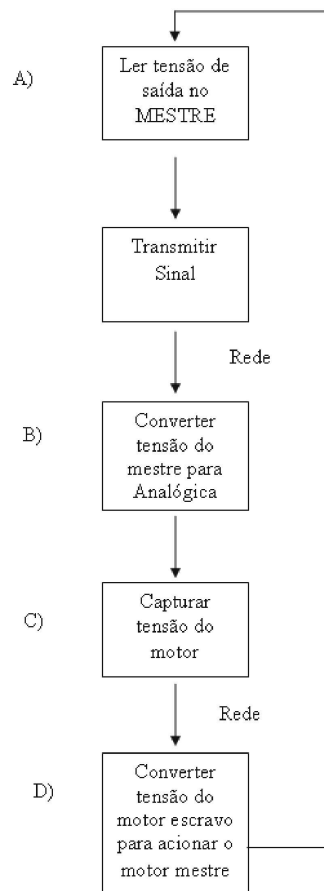


FIG 8.4: Fluxograma detalhando o caminho percorrido pelo sinal nos sistemas mestre e escravo.

A FIG. 8.6 mostra o esquema de pinagem da porta paralela DB-25 que foi utilizada neste trabalho para fazer a interface entre os programas-mestre e escravo com os sistemas mecânicos do mestre e do escravo, respectivamente. A pinagem do DB-25 é dividida em três grupos, descritos a seguir:

- Pinos de Dados;
- Pinos de Controle; e
- Pinos de Status.

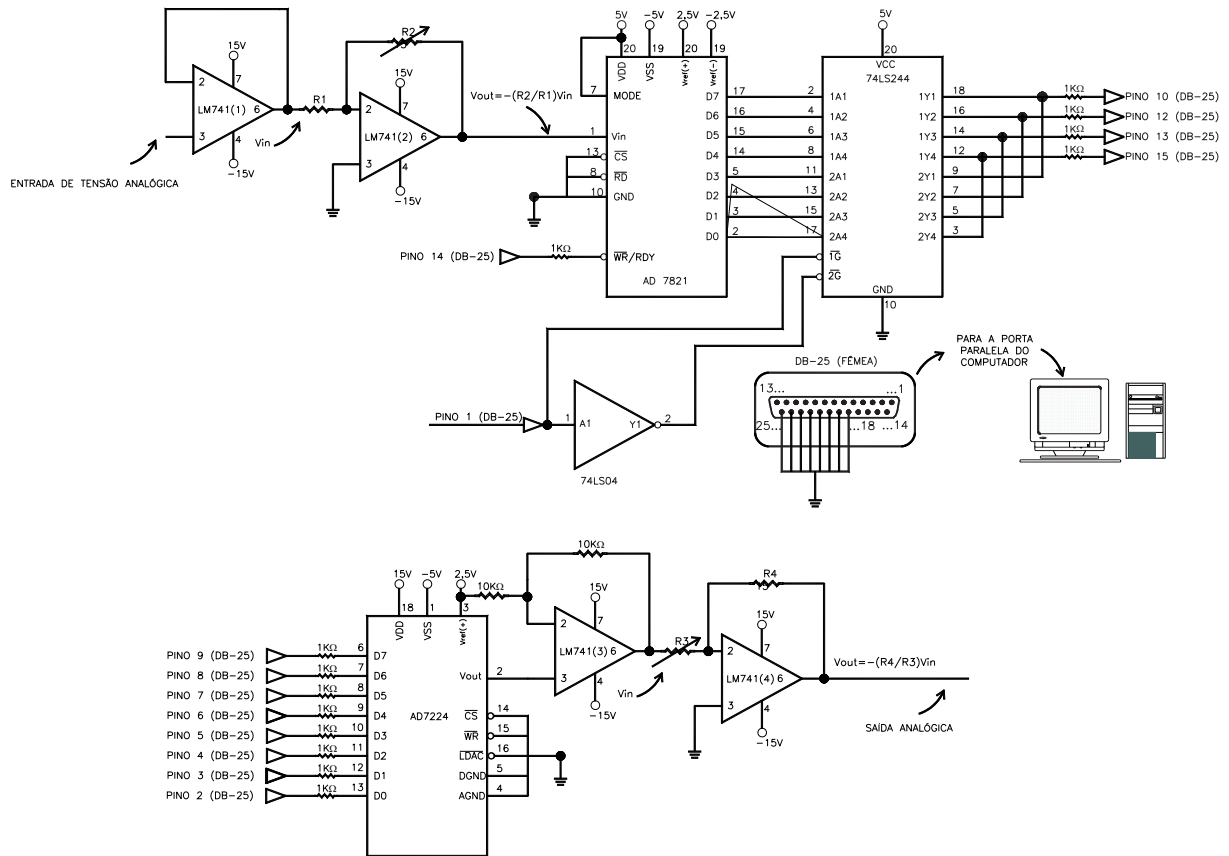


FIG 8.5: Esquema do circuito elétrico da placa desenvolvida.

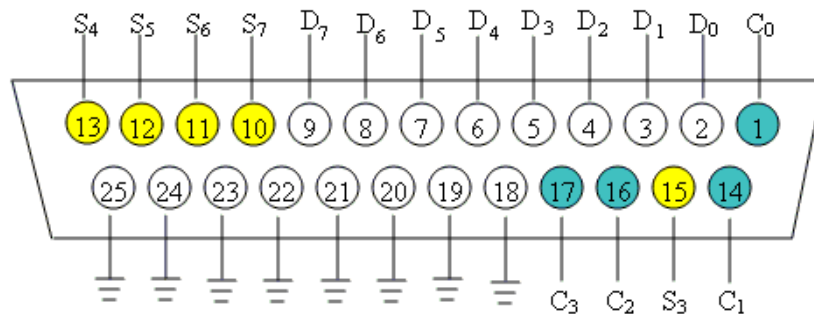


FIG 8.6: Esquema da pinagem do dispositivo (DB-25) da porta paralela.

8.1.3 DETALHES DA PINAGEM DA PORTA PARALELA

Os pinos de dados são os pinos que variam de 2 a 9 ou D_0 a D_7 , sendo que o D_0 é o pino 2. Os pinos de controle são os pinos 1, 14, 16 e 17, sendo: C_0 à C_7 , $C_0 = 1$, $C_1 = 14$, $C_2 = 16$ e $C_3 = 17$. Os pinos de status são os pinos 10, 11, 12, 13, 15 sendo que $S_7 = 11$,

$S_6 = 10$, $S_5 = 12$, $S_4 = 13$ e $S_3 = 15$. Todos os pinos possuem uma representação de 8 bits (1 byte) dessa forma os valores de saída podem variar de $00000000_2(0_{10})$ à $11111111_2(255_{10})$.

Tendo o conhecimento da saída da porta paralela e do intervalo de tensão que o sistema suporta, foi montada uma curva mostrada na FIG. 8.7. O objetivo dessa curva é extrair uma função que seja capaz de corrigir digitalmente a alimentação analógica do sistema.

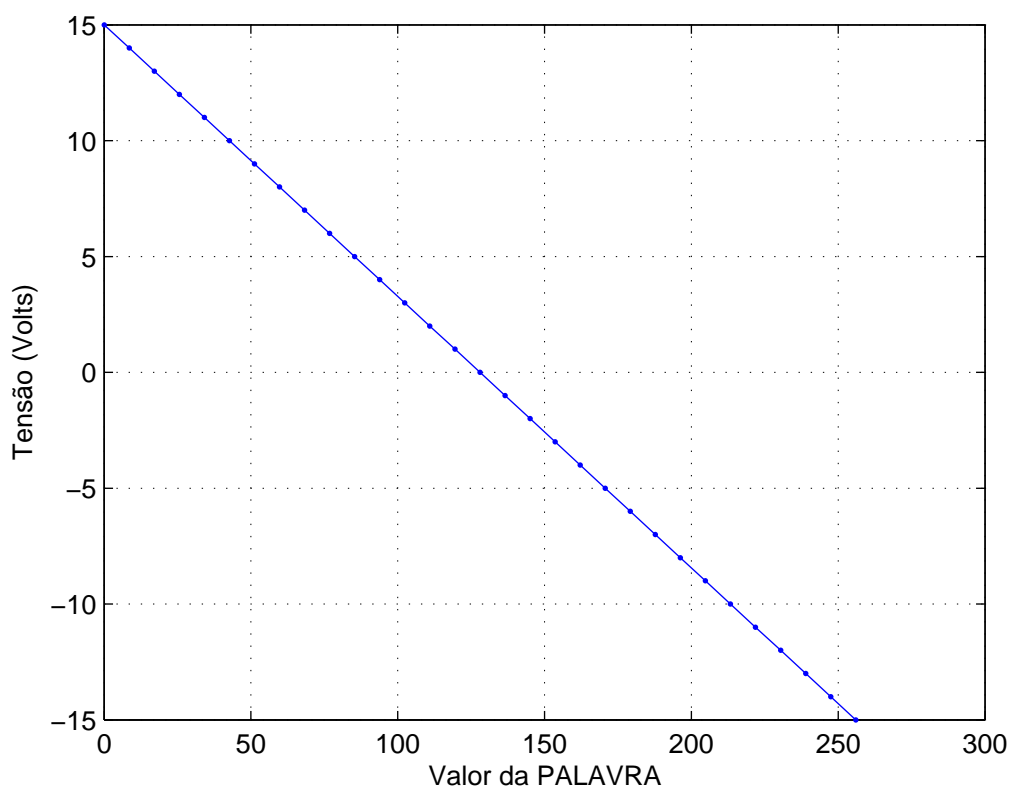


FIG 8.7: Curva de representação da palavra (eixo x) para uma determinada tensão (eixo y).

Com base na curva descrita na FIG. 8.7, foi encontrada a função que está mostrada na equação 8.1.

$$V(x) = \left[\left(-\frac{128}{15} \right) * x \right] + 128 \quad (8.1)$$

Após encontrar a curva de tensão foi desenvolvido um protocolo de comunicação com o objetivo de fazer uma interface entre a placa desenvolvida e a porta paralela do computador. O algoritmo do programa para leitura de dados na placa está detalhado a seguir:

Passo 1: Fazer o pino 14 (controle) ir para 5 volts, depois ir para 0 volts e retorná-lo a 5 volts.

Passo 2: Fazer o pino 1 ir para 0 volts, ler os 4 bits superiores dos pinos de status 10, 12, 13 e 15 e armazenar o valor em uma variável.

Passo 3, Fazer o pino 1 ir para 5 volts, ler os 4 bits inferiores dos pinos de status 10,12,13 e 15 e armazenar o valor em uma variável.

Passo 4: Formar a palavra pela concatenação dos bits, inferiores com os superiores.

A escrita é feita de uma forma simples, pois basta escrever dados nos pinos de dados que são os que variam de 2 a 9.

A FIG. 8.8 mostra o ambiente de simulação que consiste de dois computadores ligados a web. Na FIG. 8.8 ,também, é possível observar as placas de dados conectadas aos computadores.

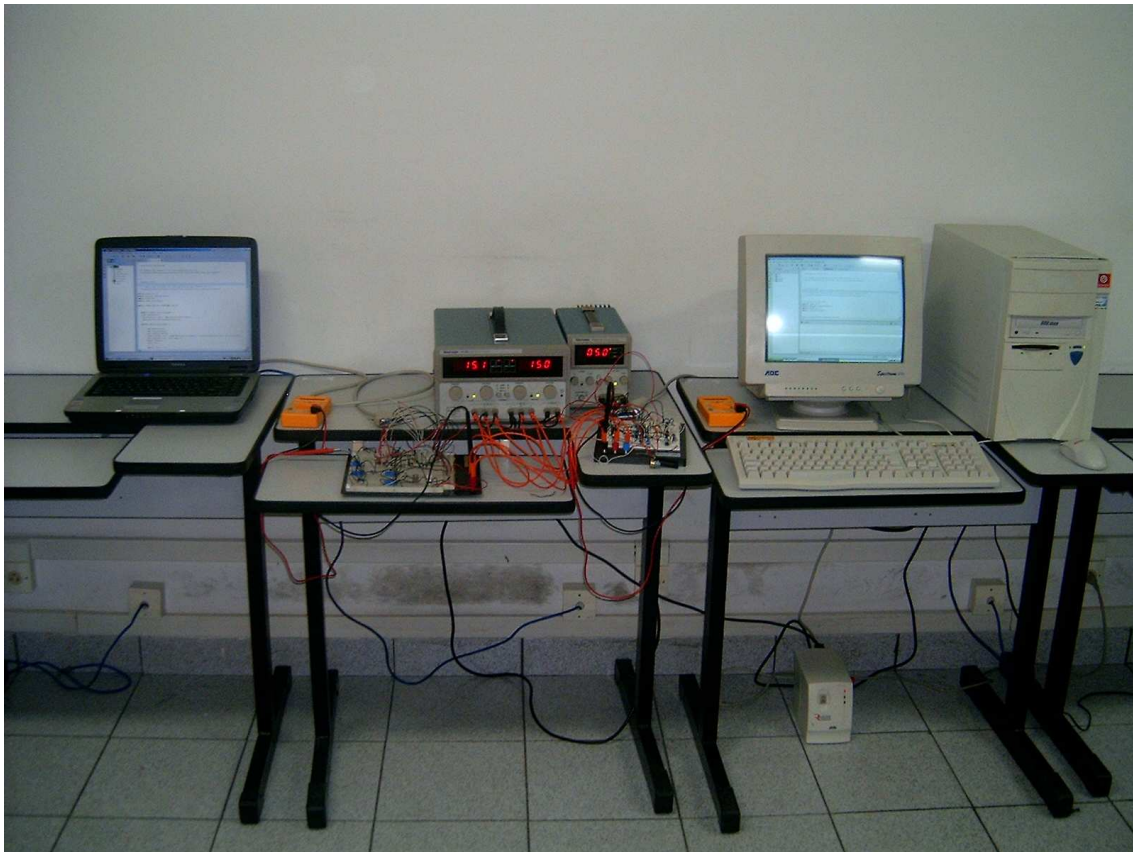


FIG 8.8: Ambiente de simulação real, consistindo de duas placas de leitura e escrita de dados conectadas a dois PC(s) conectados à Internet com uma taxa de transmissão local de 100Mb/s.

8.2 MODELO GRÁFICO PARA O SISTEMA MESTRE-ESCRAVO

Nesta seção, será mostrado o simulador gráfico desenvolvido para o trabalho, serão tratados assuntos relacionados à escolha da linguagem e do ambiente de programação. O software gráfico foi construído de forma a simular sistema mestre-escravo a fim de permitir testar o protocolo de comunicação proposto, e verificar o sincronismo do sistema.

8.2.1 DEFINIÇÃO DA LINGUAGEM DE PROGRAMAÇÃO

Inicialmente, para a construção do simulador foi feito um estudo sobre a tecnologia que deveria ser adotada. Primeiramente, optou-se por trabalhar com a linguagem C++ em conjunto com uma biblioteca gráfica OpenGL, como feito em (CORREIA, 2003). A linguagem C++ não foi adotada, pois como foi definido anteriormente, o manipulador deveria se comunicar através da web e neste caso a linguagem java se tornou mais atrativa pois além de ser mais fácil a programação ela possui as seguintes características:

- Alta Portabilidade: significa que independe da plataforma.
- Livre acesso: É uma linguagem pública e de código aberto.

Esses dois fatores foram cruciais para a escolha da linguagem JAVA. Após ser definida a linguagem foi feito um estudo sobre como desenvolver o simulador e descobriu-se que existiam várias formas de implementá-lo. Dentre estas, pode-se citar:

- Utilização de JAVA 3D;
- Utilização de VRML.

Entre as opções mencionadas, optou-se pela utilização da linguagem VRML (Virtual Reality Modeling Language), pois esta possui como principal característica o fato de permitir ao usuário criar uma cena ou um cenário em 3D, que é formado por vários objetos (ou objetos VRML), cujas características específicas permitem rotacioná-los, transladá-los e alterar a sua estrutura. Além disso, os arquivos VRML podem ser combinados com HTML ou JavaScript, permitindo a sua utilização por um browser. Em (JUNIOR, 2005) é possível encontrar informações detalhadas a respeito da utilização de VRML em associação com JAVA. Na FIG. 8.9 é possível ver um exemplo de documento VRML.

Conhecidas as vantagens da linguagem JAVA e a facilidade de VRML, fez-se necessário integrar as duas tecnologias. Para tal, foi escolhido o uso de uma API desenvolvida para a linguagem JAVA, que é denominada EAI (External Authoring Interface).


```

#VRML V2.0 utf8

WorldInfo {
  title "EXAMPLE"
  info ["R.T. - 2005"]
}

Transform {
  translation 0 -2 0
  children [
    Shape {
      appearance Appearance {
        texture ImageTexture {
          url "Fig.jpg"
          repeatS TRUE
          repeatT TRUE
        }
      }
      geometry Box {
        size 4 1 4
      }
    }
  ]
}

```

FIG 8.9: Exemplo da syntax de um documento VRML.

A EAI é uma especificação que permite a um programa externo (geralmente um applet java) acessar nós em uma cena VRML utilizando o modelo de eventos VRML. Neste modelo um *eventOut* de um dado nó pode ser redirecionado para um *eventIn* de outro nó. Quando um *eventOut* gera um evento, o *eventIn* é notificado e seu nó processa o evento. Adicionalmente, se um script possui um ponteiro para um dado nó ele poderá enviar eventos diretamente para o *eventIn* e obter o último valor enviado para um *eventOut*. Esse mecanismo de acesso é utilizado quando se envia e captura eventos utilizando JAVA. A EAI possibilita três tipos de acesso para uma cena VRML:

- a) Enviar eventos para nós *eventIns* no interior da cena.
- b) Ler o último valor enviado para um nó *eventOut* da cena.
- c) Capturar uma notificação quando um evento for enviado para um nó *eventOut*

Basicamente, podemos obter um link entre a página WEB e o applet através de um objeto Browser e com este objeto podemos manipular os eventos. O código HTML da página pode ser visto na FIG. 8.10 e na FIG. 8.11 mostra-se como acontece a interligação entre as linguagens.

Na FIG. 8.12, é possível verificar o protótipo do simulador executado através de um browser da web. Na FIG. 8.13 (A) e FIG. 8.13 (B), pode-se observar os movimentos de rotação à esquerda e à direita do braço superior, respectivamente; na FIG. 8.13 (C) e FIG. 8.13 (D) é possível ver os mesmos movimentos, mas no braço da base. Na FIG. 8.13

```

<html>
<head><title>Manipulador</title></head>
<body>
<param name="src" value="resources/vrml/manipulador.wrl">
<param name="vrml_splashscreen" value="true">
<param name="vrml_dashboard" value="true">
<object codebase="Master.vrml" width=700 height=700>
<applet code="master/MasterApplet.class" codebase="classes"
width=700 height=30 mayscript = true>
<PARAM NAME = "BrowserIndex" VALUE="0">
</APPLET>
</body>
</html>

```

FIG 8.10: Código da página HTML.

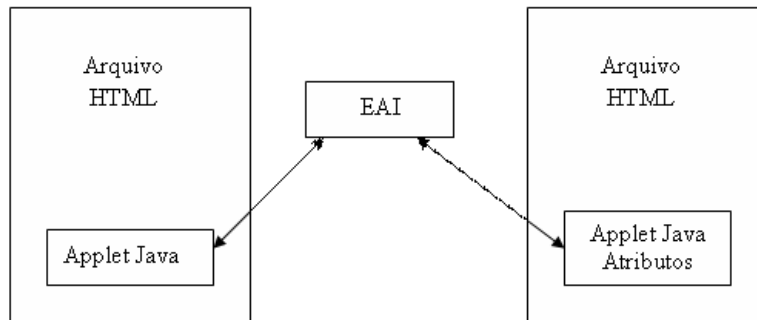


FIG 8.11: Interligação da linguagem JAVA com VRML utilizando EAI.

(E) e (F), observa-se o movimento de translação. Vale lembrar que esta demonstração foi feita utilizando-se uma parte do simulador, pois o sistema completo compõe-se de um mestre e um escravo.

O manipulador apresentado na FIG. 8.12 e na FIG. 8.13, serviu de teste para a elaboração do protocolo de comunicação, onde os dados transmitidos constituíam 3-Tuplas que continham as informações de rotação do braço da base, translação e rotação do braço superior. A FIG. 8.12 mostra o simulador em execução através de um browser

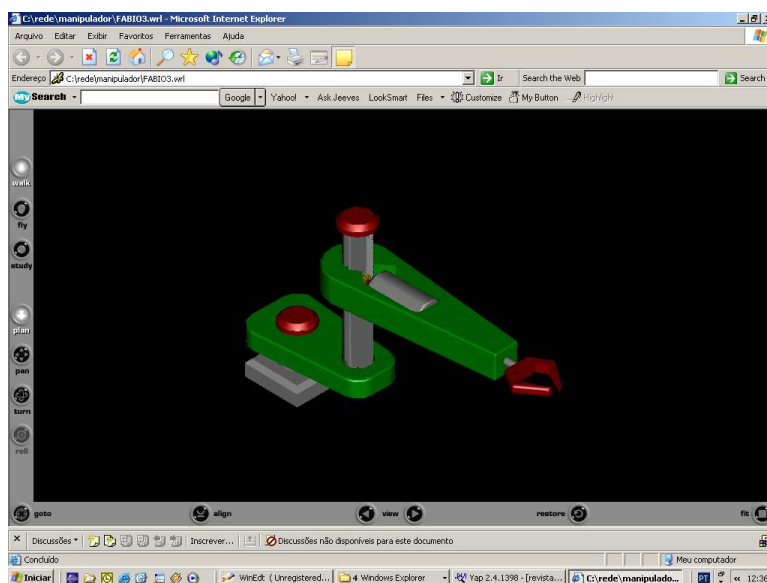


FIG 8.12: Manipulador 3 GDL executado a partir de um browser, utilizando o plugin Cortona

utilizando o Cortona¹⁰.

8.2.2 ESCOLHA DA PLATAFORMA PARA IMPLEMENTAÇÃO

Inicialmente tentou-se utilizar o sistema operacional Windows mas neste a utilização de EAI fica limitada à máquina virtual da microsoft (MSJVM). A utilização da MSJVM tornou-se um impecilho para o sistema visto que, atualmente, a Microsoft não mais oferece suporte e/ou ferramentas de desenvolvimento para esta máquina virtual, que deixou de ser desenvolvida.

Na tentativa de tornar o sistema independente da plataforma Windows e consequentemente da MSJVM, partiu-se para uma implementação utilizando o sistema operacional LINUX. O passo inicial foi adquirir um plugin para LINUX que possibilitasse o uso da máquina virtual da SUN e da segunda geração do Java (1.2.x, 1.3.x, 1.4.x, 1.5.x). Os plugins para Linux ainda encontram-se em estágio inicial de desenvolvimento, sendo a

¹⁰Cortona é um plugin utilizado para visualizar arquivos VRML através de um browser e pode ser encontrado no seguinte endereço: <http://www.parallelgraphics.com/products/cortona/>.

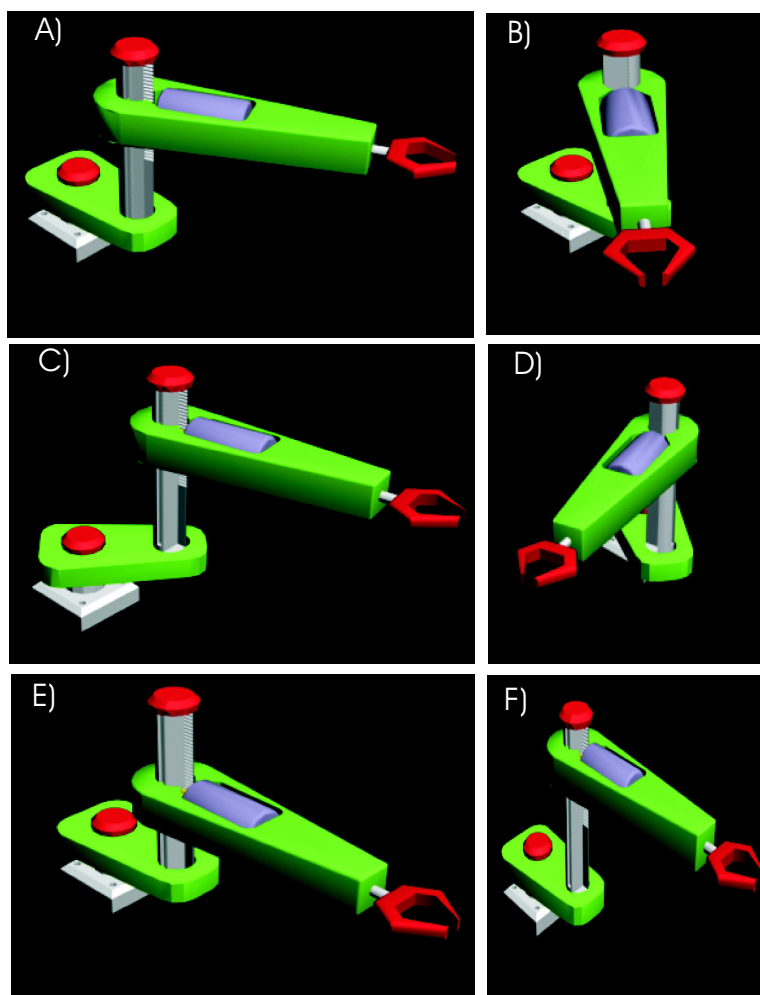


FIG 8.13: manipulador 3 GDL em detalhes: (A) movimento de rotação à direita (B) rotação à esquerda do braço superior; (C) movimento de rotação à direita; (D) rotação à esquerda do braço inferior. Em (E) e (F) observa-se o movimento de translação.

maioria deles desenvolvidos pela comunidade da internet. Como opção para a implementação foi escolhido o FreeWRL¹¹.

A implementação do EAI do FreeWRL ainda encontra-se em estágio de desenvolvimento e devido a este fato a comunicação JAVA/VRML é muito lenta, mesmo quando implementado em uma máquina que utiliza um processador Pentium IV. A utilização de placas gráficas 3D ainda não foi implementada devido a este fato, a taxa de exibição dos modelos tridimensionais fica prejudicada. A FIG. 8.14 mostra o manipulador mestre, estabelecendo conexão com o escravo já a FIG. 8.15 mostra o manipulador mestre realizando uma rotação em seu braço inferior.

¹¹FreeWRL é um plugin para LINUX e pode ser encontrado no site <http://freewrl.sourceforge.net/>.

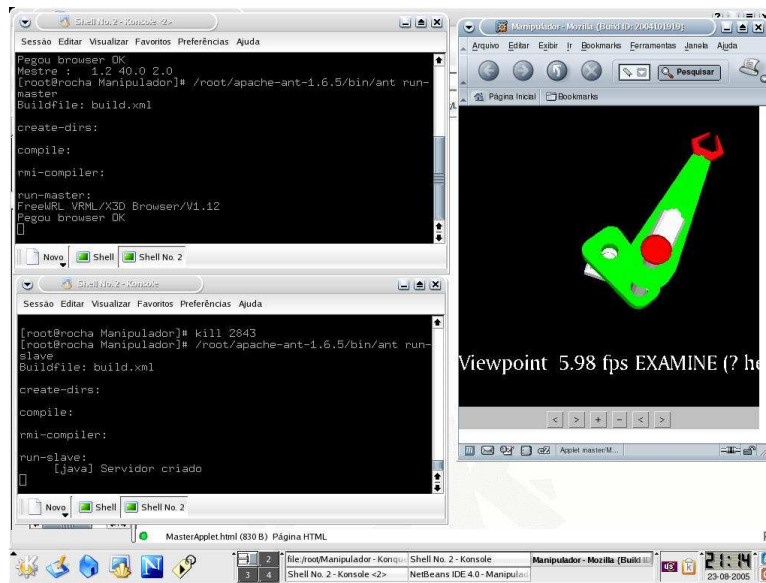


FIG 8.14: Visualização gráfica do manipulador mestre utilizando o Java 1.4.2, o Conectiva Linux 9 e o navegador Mozilla

8.2.3 CONCLUSÃO

Este capítulo, consistiu de duas etapas: a primeira foi a elaboração e desenvolvimento de uma interface física capaz de permitir que dois manipuladores reais sejam conectados à web via porta paralela de um PC. A segunda consistiu na elaboração e implementação de um simulador gráfico com 3 GDL desenvolvido com o objetivo de testar o protocolo de comunicação proposto neste trabalho e para verificar o sincronismo do mesmo.

Inicialmente, foi elaborado o desenvolvimento de uma interface capaz de permitir que dois manipuladores robóticos possam se comunicar via web. Esta elaboração consistiu no desenvolvimento de dois protocolos de comunicação, escritos na linguagem JAVA; e no projeto e construção de duas placas de leitura/escrita de dados. Também, foi necessário desenvolver dois protocolos de comunicação devido ao fato de um ser utilizado para transmissão dos dados via TCP/IP e um outro para leitura e escrita de dados na porta paralela de um PC.

Apos concluir a etapa da construção do dispositivo físico, foi elaborado um simulador baseando-se em uma tecnologia envolvendo as linguagens Java e VRML. Para a comunicação entre estas foi utilizada uma API chamada EAI, cuja finalidade era de diminuir o esforço de programação e utilizar os recursos já implementados. Este tipo de tecnologia atendeu em parte o objetivo do simulador, pois o sistema não teve uma boa resposta em relação ao fluxo de dados em tempo-real. Devido a problemas encontrados com a máquina virtual da Microsoft o sistema foi migrado para LINUX; este fato implicou em uma limi-

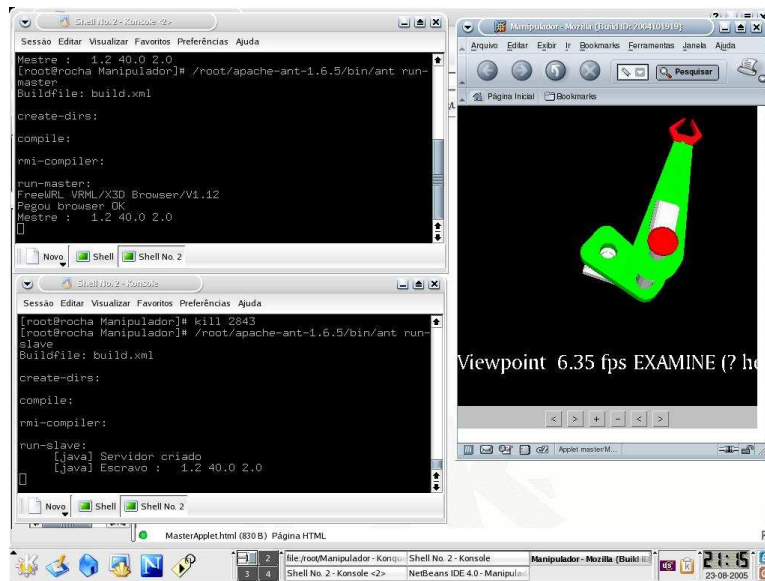


FIG 8.15: Visualização gráfica do manipulador mestre, executando um movimento de rotação do braço superior, utilizando o Java 1.4.2, o Conectiva Linux 9 e o navegador Mozilla

tação para o bom funcionamento do simulador, pois este não teve um bom desempenho com relação à utilização de recursos gráficos devido ao fato do plugin, FreeWrl, utilizado para interligar a linguagem JAVA e o VRML, estar em fase inicial de desenvolvimento.

A vantagem do sistema está no fato de poder ser executado em plataformas diferentes, sem a necessidade de grandes modificações. Outro aspecto que atrai a implementação deste sistema é a possibilidade de utilizar os recursos disponíveis na Internet a fim de reduzir os custos dos sistemas teleoperados.

9 RESULTADOS EXPERIMENTAIS

Neste capítulo, serão descritos os resultados experimentais que servirão para validar todas as etapas descritas anteriormente. Inicialmente, foi testada a arquitetura de controle através do *Simulink*TM e em seguida serão mostrados os testes feitos com o protótipo desenvolvido para conectar o sistema à web.

9.1 ANÁLISE DE DESEMPENHO E ESTABILIDADE

No Capítulo 8 foi proposta uma arquitetura de controle para o sistema e neste item será feita a análise de estabilidade desta através dos diagramas de Bode, também serão feitos testes experimentais com diversas funções a fim de verificar a estabilidade do sistema. O objetivo é que este continui estável quando exposto a oscilações, ou melhor, é desejável que o sistema tenha uma resposta limitada para uma entrada limitada.

Inicialmente, faz-se necessário mostrar o conceito de estabilidade, que na verdade resume-se a um sistema com resposta limitada, ou seja, o sistema não deve exceder os valores de magnitude limitados. Se o sistema for submetido a uma excitação limitada ou a um ruído limitado e a resposta for limitada em magnitude o sistema é estável. Para o presente trabalho, a análise de estabilidade será feita utilizando-se o diagrama de Bode.

9.1.1 DIAGRAMA DE BODE

Os diagramas de Bode, ou gráficos logarítmicos são assim denominados em homenagem a H. W. Bode, que os utilizou incansavelmente em seus estudos sobre amplificadores com retroação. As FIG. 9.1 e FIG. 9.2. mostram o diagrama de Bode, margem de fase e margem de ganho, onde dB (decibel), é o ganho logarítmico. Para verificar se o sistema é instável, basta observar se o diagrama de Bode possui algum ponto comum, onde o ganho $dB = 0$ e o ângulo de fase $\Phi = 180^\circ$. Observando-se a FIG. 9.1 e a FIG. 9.2, pode-se concluir que no espaço amostral destas não existe nenhum ponto que convirja para a situação de instabilidade. Em vista disto, diz-se que o sistema tende à estabilidade para os valores apresentados. Para efeitos de uma melhor visualização, a FIG. 9.3 mostra uma composição dos gráficos de magnitude e fase postos lado a lado.

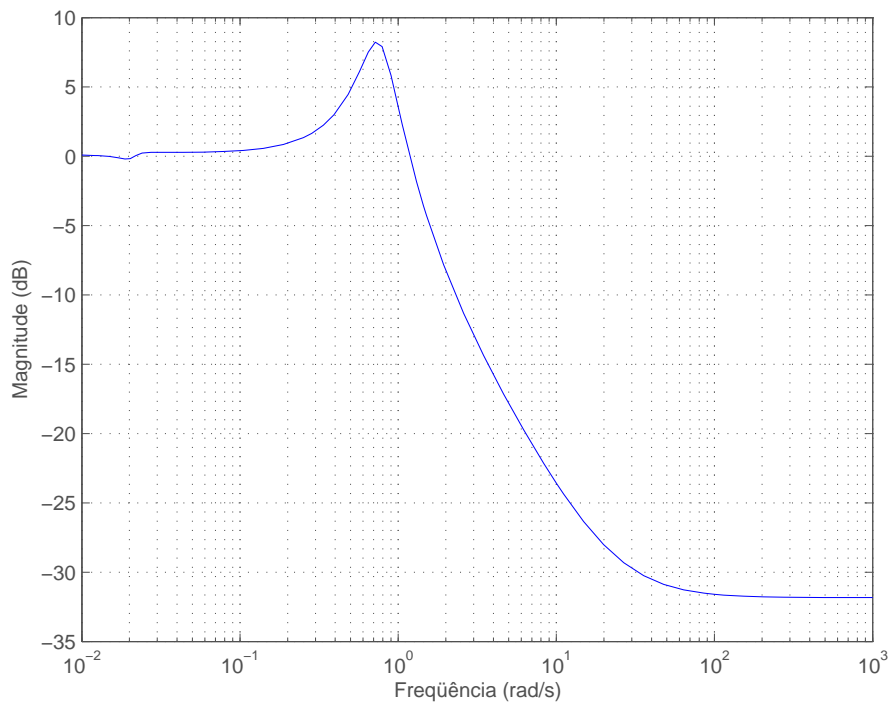


FIG 9.1: Diagrama de Bode, (magnitude) x (tempo).

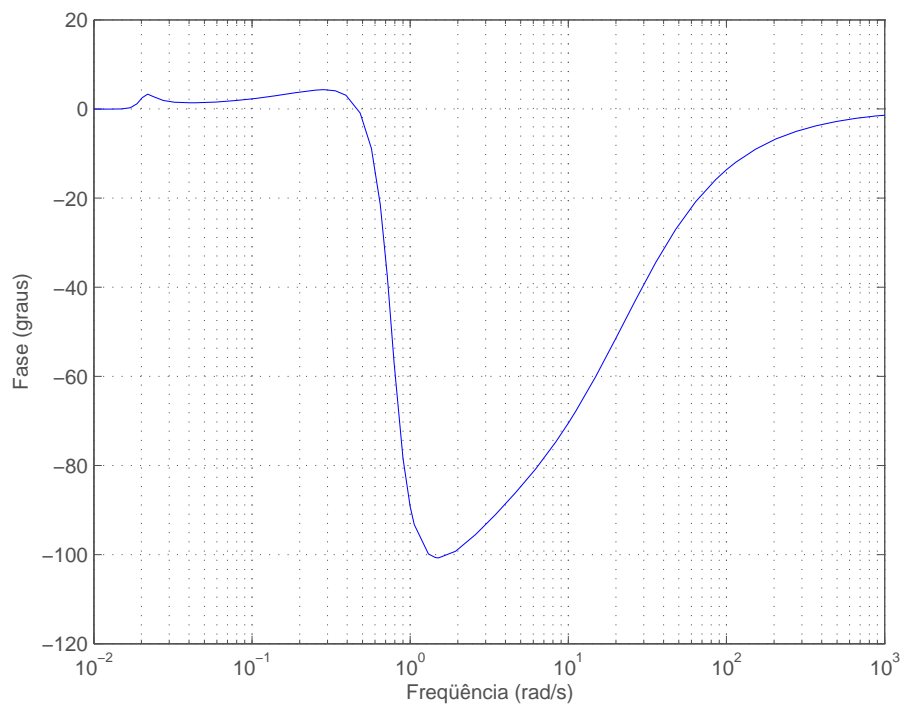


FIG 9.2: Diagrama de Bode, (fase) x (tempo).

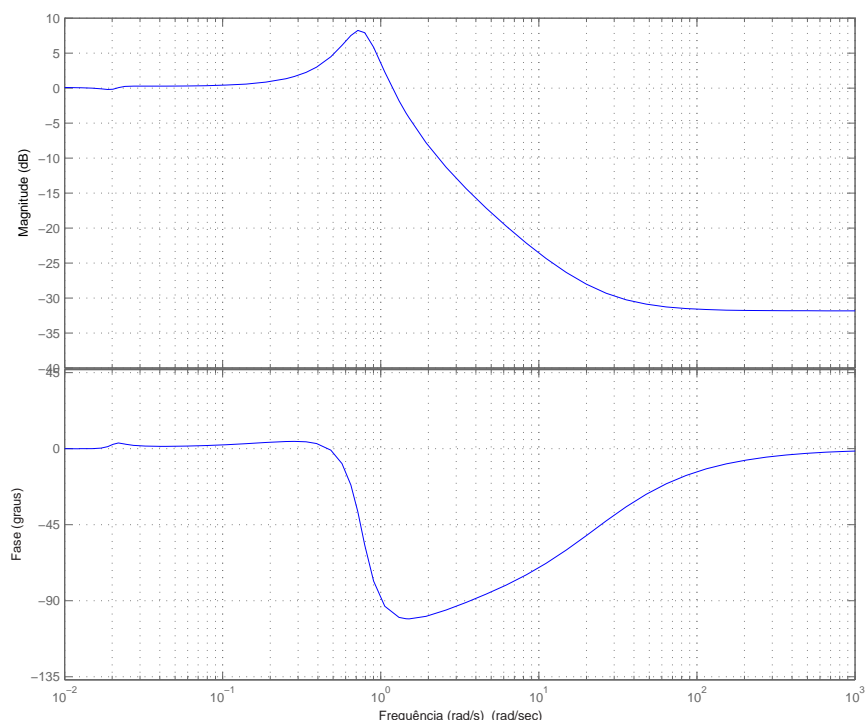


FIG 9.3: Diagrama de Bode, magnitude e fase mostradas em um só gráfico.

9.1.2 TESTE DA ARQUITETURA DE CONTROLE

Após ter extraído o diagrama de Bode, foram feitos alguns testes utilizando-se o *SimulinkTM*. O primeiro teste está representado na FIG. 9.4 e consistiu em fornecer uma função degrau como entrada para o sistema. Através do gráfico da FIG. 9.4, observa-se que o sistema possui boa resposta quando excitado, pois tende rapidamente a se estabilizar. Na FIG. 9.5, tem-se o gráfico para a função impulso que também fornece uma boa resposta ao sistema; quando excitado ele estabiliza-se em questões de milésimo de segundos.

Um segundo teste foi feito utilizando-se a função seno como entrada; o comportamento desta é mostrado na FIG. 9.6 e na FIG. 9.7. O teste foi feito utilizando-se uma onda com amplitude de 0.01 *Nm* e frequência de 0.1 *rad/s*. O período de amostragem foi de 100 segundos. Na FIG. 9.7, é possível verificar a resposta do sistema no qual a onda de saída é muito próxima da onda de entrada.

O terceiro e último teste foi feito utilizando-se uma função geradora de números randômicos, cuja saída normalmente é uma distribuição Gaussiana do sinal. Para a execução deste seguiu-se o padrão já adotado anteriormente. O teste está representado na

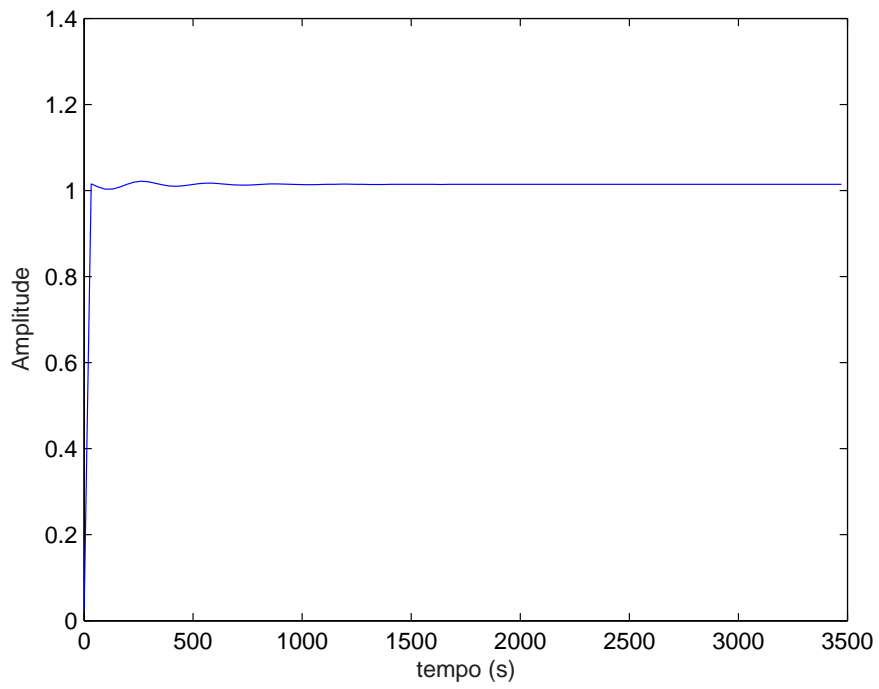


FIG 9.4: Resposta do sistema à função degrau.

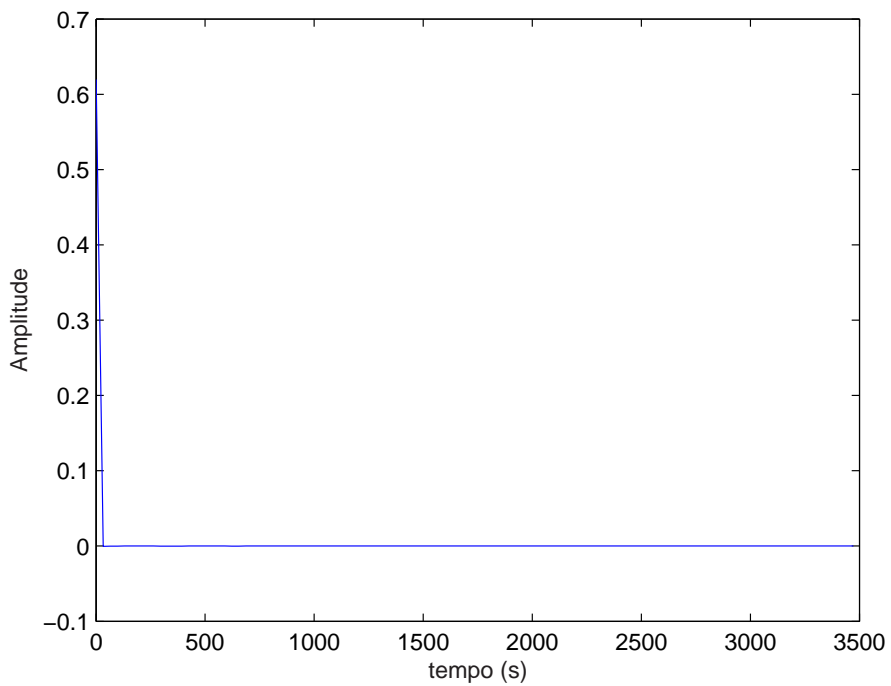


FIG 9.5: Resposta do sistema à função impulso.

FIG. 9.8 e na FIG. 9.9. Para o teste, foi utilizada uma variância de 0.01 e o tempo de amostragem foi de 100 segundos.

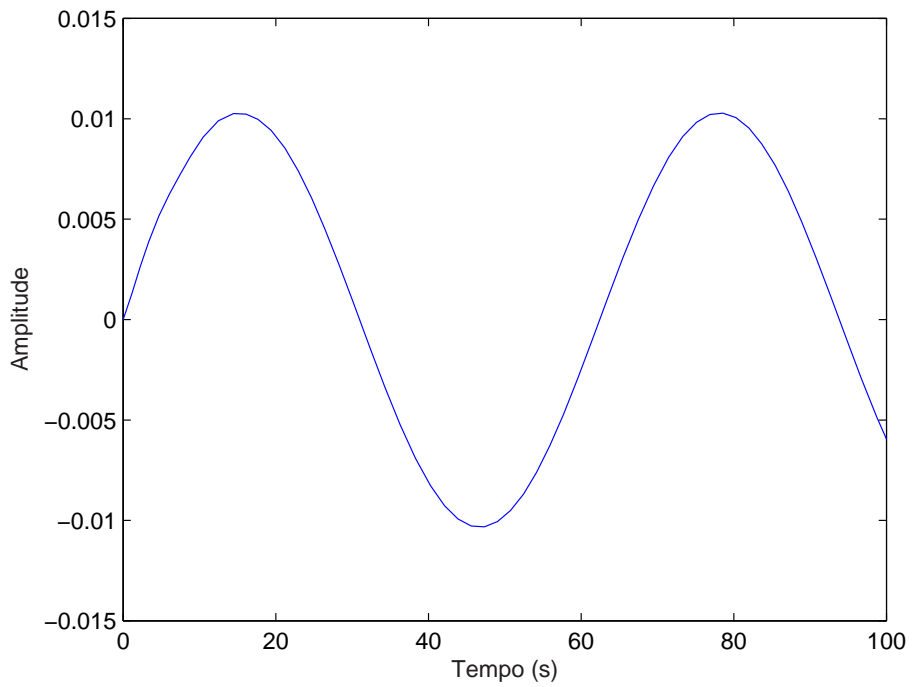


FIG 9.6: Alimentação do sistema através da função seno. O eixo das abscissas fornecem a informação sobre o tempo e o eixo das ordenadas fornecem a informação sobre a amplitude.

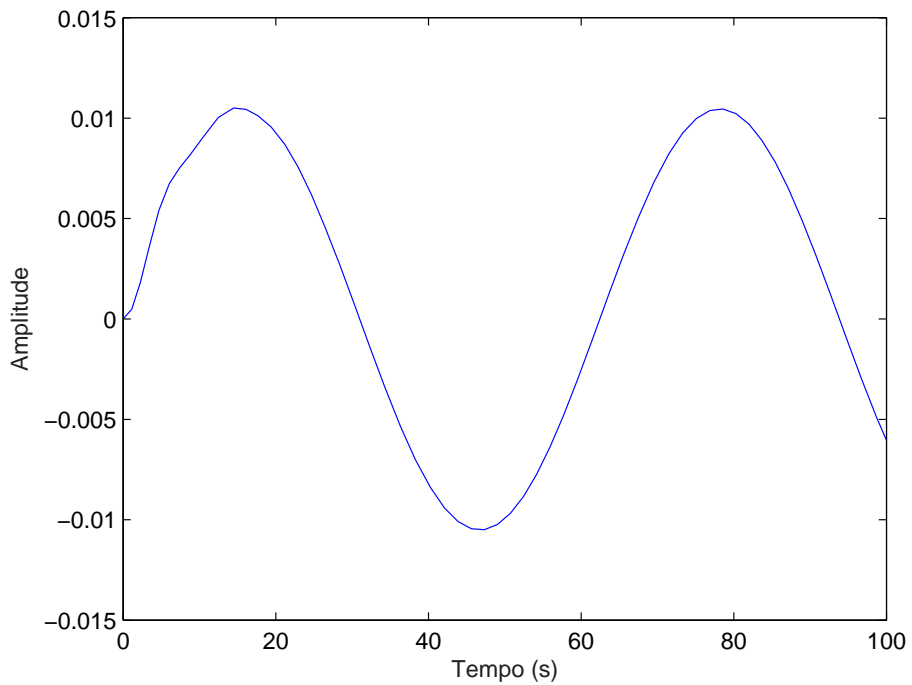


FIG 9.7: Resposta do sistema à entrada feita com a função seno. Resposta obtida no sistema escravo.

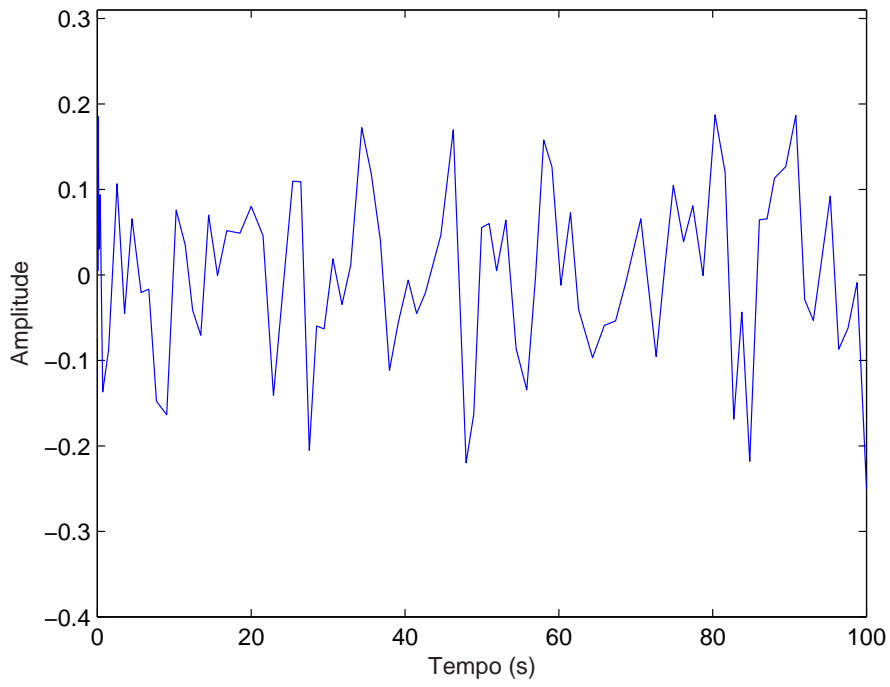


FIG 9.8: Alimentação do sistema através da função de números aleatórios.

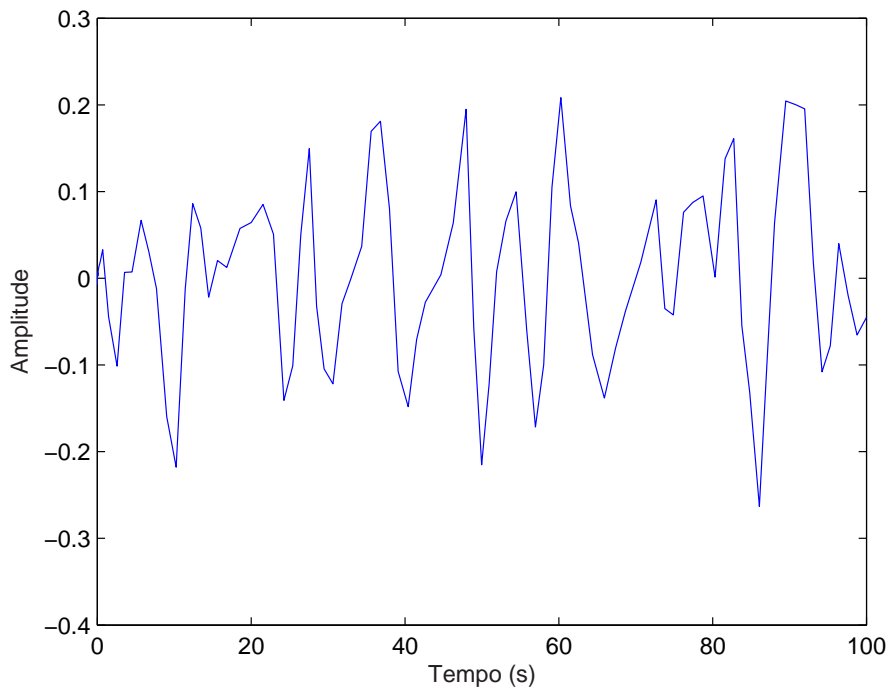


FIG 9.9: Resposta do sistema à função de números aleatórios. Resposta obtida no sistema escravo.

9.2 SIMULAÇÃO DO DISPOSITIVO FÍSICO

Após construir as placas de captura/escrita de dados e definir o algoritmo de leitura foram desenvolvidos dois programas, um programa-mestre e um programa-escravo. O programa mestre e o programa-escravo são uma combinação dos dois protocolos propostos neste trabalho: protocolo de comunicação descrito no capítulo 7 e protocolo de comunicação da porta paralela com as placas desenvolvidas.

Inicialmente, o sistema foi testado pela adição de um potenciômetro na placa-mestre. O objetivo foi simular a variação de tensão que poderia ocorrer durante a simulação com os manipuladores mestre e escravo.

O desenvolvimento de sistemas como este requer uma fase onde se deve fazer ajustes a fim de tornar esses sistemas confiáveis. Para tal, foi realizada uma simulação com o objetivo de calibrar o sistema. A simulação consistiu de uma leitura de -5 a 5, volts feita na saída analógica de cada placa. Utilizando um potenciômetro acoplado à placa-mestre foi feita a variação de tensão, sendo que a leitura nas placas foi coletada a cada variação de 0,5 volts. A curva encontrada pode ser vista na FIG. 9.10. A configuração dos computadores utilizados na simulação é a seguinte: para executar o sistema escravo foi utilizado um computador com processador celeron 2.7 GHz com 256 MB de memória RAM, 40 Giga bytes de HD, 1 entrada paralela, 3 entradas USB, placa de rede de 100Mb/s e Windows 2000. Para executar o sistema mestre, foi utilizado um Pentium III, 64 MB RAM, 1 entrada paralela, uma placa de rede 10/100 Mb/s e Windows XP.

Na FIG. 9.10, é possível observar os pontos que são as respostas do escravo, dada uma determinada entrada de tensão no mestre. A reta mostrada na FIG. 9.10 representa a situação ideal para o sistema e foi traçada para fazer um contraste entre a leitura realizada e o que seria uma situação ótima para o sistema. Com esta curva, é possível extrair uma função de correção para que a mesma tensão fornecida no mestre seja lida no escravo.

Uma outra simulação executada foi a análise do tempo de transmissão do sistema; esta simulação consistiu em manter os sistemas conectados e fixar uma determinada quantidade de bytes a serem transmitidos. Dessa forma, conseguiu-se um espaço amostral e dentro deste intervalo foram analisados alguns elementos tomados ao acaso. Destes elementos foram analisados o tempo de ida e volta dos dados com escrita na placa e o tempo de ida e volta sem escrita na placa. O tempo que o sistema mestre ou escravo utiliza é de 0,01 segundos, que foi o tempo de atraso inserido no programa para que este pudesse ativar os pinos de controle que habilitam o software a ler a porta paralela. Esta

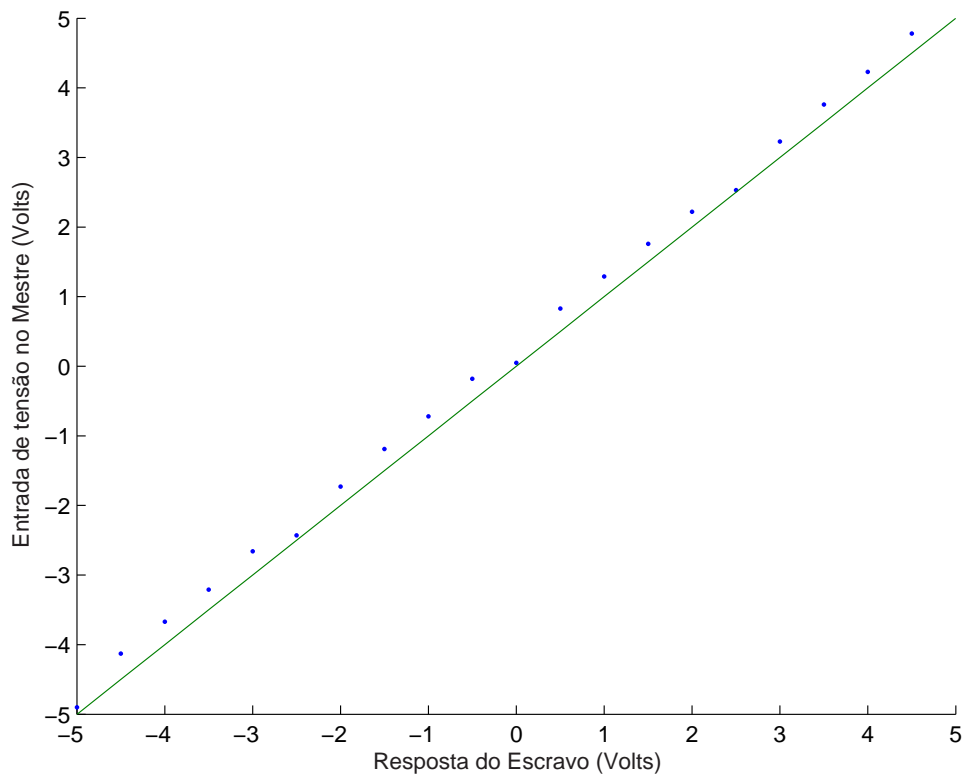


FIG 9.10: Curva de calibração. Os pontos representam a resposta do escravo para uma entrada de tensão no mestre. A reta contínua representa a resposta ideal para o sistema.

simulação foi feita no lado do sistema mestre, da seguinte forma:

- foi medido o tempo que o sistema leva pra escrever e ler um byte na placa;
- foi definido um espaço amostral de 100 bytes a ser transmitido;
- desses 100 bytes, 10 foram escolhidos ao acaso;
- foi medido o tempo de ida e volta do byte com o mestre escrevendo e lendo na placa;
- foi medido o tempo de envio e recebimento do byte sem o mestre estar escrevendo ou lendo dados na placa.

O resultado da simulação está mostrado na FIG. 9.11. As barras pintadas de preto fornecem a informação do tempo gasto para transmitir e receber um determinado byte sem o programa-mestre estar escrevendo ou lendo na placa-mestre. Já as barras pintadas de branco fornecem o tempo, considerando a placa de leitura/escrita utilizada pelo sis-

tema mestre. O tempo total da simulação foi de 36,442 segundos. Dessa forma, a média para transmitir e receber um byte foi de 0,36442 segundos.

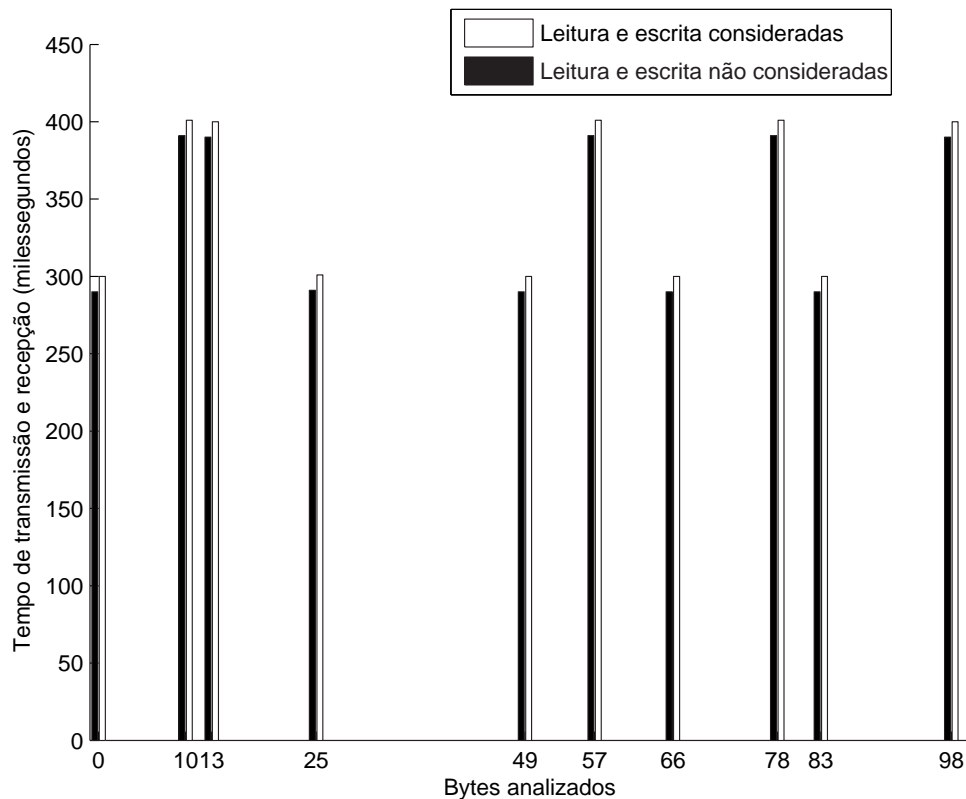


FIG 9.11: Gráfico representando o fluxo de dados do sistema; o eixo das abscissas mostra o byte analisado e o eixo das ordenadas mostra o tempo gasto para transmití-lo e recebê-lo.

9.3 CONCLUSÃO

Neste capítulo são exibidos os resultados experimentais obtidos durante o desenvolvimento do sistema. Inicialmente, foi testada a arquitetura de controle proposta para o trabalho e logo após foi testado um protótipo real capaz de simular a transmissão da leitura e escrita de sinais em um manipulador real.

Para fazer a análise de estabilidade do sistema, lançou-se mão do recurso dos gráficos logarítmicos ou Diagrama de Bode. Ainda, com relação à estabilidade foram analisadas as funções, degrau e impulso, seno e números aleatórios para verificar o tempo de "convergência" do sistema à estabilidade.

Após testar a arquitetura de controle, foram feitos os testes do dispositivo desenvolvido, que é capaz de permitir que dois manipuladores robóticos possam se comunicar

via web. Esta simulação consistiu em desenvolver uma aplicação do tipo cliente/servidor feito na linguagem JAVA, onde o mestre, é representado pelo cliente; e o servidor, simula o funcionamento do escravo. Após o desenvolvimento da aplicação cliente/servidor o sistema foi ligado via web através de uma conexão TCP/IP.

A grande vantagem do sistema desenvolvido está no fato de poder ser executado em plataformas diferentes, sem a necessidade de grandes modificações. Isto foi possível porque utilizou-se uma linguagem de programação JAVA, que é multiplataforma e um recurso (porta paralela) que está disponível em qualquer PC (Personal Computer).

Futuramente, pretende-se migrar o sistema para a linguagem C ou C++, pois apesar da facilidades disponibilizadas pela linguagem JAVA, esta não se mostrou eficaz no que tange ao desenvolvimento de aplicações que envolvam tempo real. Como exemplo, pode-se citar o recurso chamado garbage collection que a linguagem JAVA implementa o que implica em uma queda de performance da aplicação que a está utilizando.

10 COMENTÁRIOS GERAIS

Este trabalho descreve o projeto de um sistema bilateral de telemanipulação com realimentação háptica capaz de ser operado via web. A elaboração do projeto consistiu de várias etapas: o estudo de diversos sistemas de teleoperação existentes; a definição do manipulador, assim como a elaboração das equações cinemáticas e dinâmicas; o estudo e o desenvolvimento da arquitetura de controle para o sistema; a elaboração e implementação da comunicação do sistema; a construção de um simulador gráfico a fim de testar o sistema de comunicação; o projeto e construção de um dispositivo de leitura e escrita de dados que utiliza a porta paralela do computador; e, para finalizar, foram feitos vários testes a fim de ajustar o sistema.

Inicialmente, foram obtidas as equações cinemáticas e dinâmicas para um manipulador de 3(três) graus de liberdade, com duas juntas de rotação e uma junta prismática. O manipulador foi elaborado com o propósito de testar o protocolo de comunicação desenvolvido, o número de graus de liberdade e, para o caso, tem o propósito de aumentar o espaço operacional(ou espaço de trabalho) do mesmo.

O desenvolvimento da arquitetura de controle do sistema foi feito a partir de um modelo mecânico, e desse modelo foram extraídas equações diferenciais que resultaram em um diagrama de blocos. Após a estruturação desse diagrama foi possível fazer a análise de estabilidade da arquitetura proposta. Para esta análise lançou-se mão do recurso dos gráficos logarítmicos ou Diagramas de Bode. Além do diagrama de Bode, foram analisadas funções que serviram para verificar o tempo de "convergência" do sistema que leva à estabilidade.

Para o desenvolvimento do sistema de comunicação, foi realizado um estudo sobre a arquitetura de rede de computadores TCP/IP, visando adquirir informações sobre os protocolos que se adequariam melhor ao sistema proposto.

Analisando os protocolos da camada de transporte do modelo TCP/IP, constatou-se que os protocolos UDP e/ou TCP poderiam ser utilizados para a implementação do sistema de comunicação. O UDP é um protocolo simples, de fácil utilização e que fornece bons resultados para aplicações que envolvam o conceito de tempo real, mas que não sofram alterações com perdas de dados, por exemplo: transmissão de áudio e vídeo descomprimido. O TCP, por sua vez, mostra-se como um protocolo mais complexo

que possui características como o estabelecimento de um canal de comunicação confiável, mecanismo de retransmissão de dados, entrega de dados na ordem em que estes lhe foram confiados, e controle de congestionamento. Contudo, o TCP possui o problema de não conseguir transmitir os seus pacotes a uma taxa constante, devido ao mecanismo de controle de congestionamento. Para a implementação do sistema de comunicação, foram definidos alguns pré-requisitos, como:

- A transmissão dos dados deve ser feita a uma taxa constante;
- Não pode haver interrupção enquanto os processos estejam se comunicando;
- Os dados devem ser entregues na ordem em que estes lhe foram confiados; e
- A transmissão dos dados deve ser feita em tempo real.

Depois de estabelecidos os pré-requisitos e observando-se o comportamento do sistema, optou-se por implementar a comunicação dos sistemas baseando-se no modelo de *Cliente/Servidor*, onde o sistema mestre foi o cliente e o sistema escravo foi o servidor. Para a implementação do sistema foi utilizado o conceito de programação de soquets (do inglês: socket programming). O protocolo escolhido para atuar ao nível da camada de transporte foi o TCP pois ele atendeu aos pré-requisitos em sua grande maioria. A princípio, fazer a comunicação utilizando o conceito de cliente/servidor pareceu ser a melhor opção. O próximo passo a ser tomado, será tentar implementar um protocolo que torne a aplicação independente da utilização de um canal de comunicação de alta capacidade. Outras soluções devem ser testadas, como a utilização do protocolo UDP, pela implementação de serviços adicionais, na camada de aplicação, àquilo que o UDP provê; e pode-se testar o RTP (Real Time Protocol) que é um protocolo utilizado em aplicações de tempo real.

Foi elaborado um simulador, baseando-se em uma tecnologia envolvendo as linguagens JAVA e VRML. Para a comunicação entre estas foi utilizada uma API chamada EAI, cuja finalidade foi diminuir o esforço de programação e utilizar os recursos já implementados. Este tipo de tecnologia atendeu em parte o objetivo do simulador, pois o sistema não obteve uma boa resposta com relação ao fluxo de dados em tempo-real. Um outro problema encontrado foi em relação à utilização do sistema em plataformas diferentes, pois a utilização de VRML mostrou-se restrita a aplicações antigas e proprietárias, Como solução para este problema futuramente pretende-se reconstruir o simulador utilizando Java 3D.

Para o sistema proposto desenvolveu-se um dispositivo de leitura e escrita de dados na porta paralela do computador e construiu-se um protocolo para a comunicação via esta porta. Esse desenvolvimento, em trabalhos futuros permitirá, que se acople os manipuladores para que estes funcionem via TCP/IP.

A grande vantagem do desenvolvimento deste sistema foi a possibilidade de ser executado em plataformas diferentes, sem a necessidade de grandes modificações. Isto foi possível devido à utilização de uma linguagem de programação que é multi-plataforma e de um recurso (porta paralela) que está disponível em qualquer PC.

O próximo passo a ser dado é o acoplamento do MVT (Manipulador Virtual Teleoperado) ao sistema desenvolvido. Com relação ao simulador gráfico será necessário alterar a implementação para que este também possa ser acoplado ao sistema.

11 REFERÊNCIAS BIBLIOGRÁFICAS

- ADAMS, RICHARD J.; HANNAFORD, B. **A Two-Port Framework for the Design of Unconditionally Stable Haptic Interfaces.** *Proceedings of the 1998 IEEE/RSJ International Conference on Intelligent Robots and Systems*, págs. 1254–1259, 1998.
- ALBERIO, M. D. V. e OLIVEIRA, J. C. **ACOnTECe: um Ambiente COlaborativo para TrEinamento Cirúrgico.** *Workshop de Teses e Dissertações do VII Symposium on Virtual Reality (SVR2004)*, São Paulo - SP. *Anais do Symposium on Virtual Reality*, 1, Outubro 2004.
- ANDERSON, ROBERT. J., S. M. W. **Hybrid Impedance Control of Robotic Manipulators.** *IEEE Journal of Robotics and Automation*, 4(3):549–556, October 1988.
- BALLANTYNE, G. **TRobotic surgery, telerobotic surgery, telepresence, and telementoring, Surgical Endoscopy.** *Springer Verlag New York, LLC*, 6:1389–1402, October 2002.
- BEJCZY, A. K. **Virtual Reality In Robotics.** *Jet Propulsion Laboratory*, págs. 7–15, 1996.
- BICCHI, A., C. G. R. D. D. L. P. e SCILINGO, E. P. **A Sensorized Minimally Invasive Surgery Tool for Detecting Tissuatal Elastic Properties.** *Proceedings of the 1996 IEEE International Conference on Robotics and Automation Minneapolis, Minnesota*, págs. 884–888, April 1996.
- BUTNER, S. E. e GHODOUSSI, M. **Transforming a Surgical Robot for Human Telesurgery.** *IEEE Transactions on Robotics and Automation*, 19(5):818–824, October 2003.
- ÇAVUŞOĞLU, M. C.; SHERMAN, A. e TENDICK, F. **Bilateral controller design for telemanipulation in soft environments.** *In Proceedings of the IEEE International Conference on Robotics and Automation, Seoul, Korea*, May 2001.
- ÇAVUŞOĞLU, M. C.; TENDICK, F. C. M. e S., S. **A Laparoscopic Telesurgical Workstation.** *IEEE Transactions on Robotics and Automation*, 15(4):728–739, August 1999.
- CONTE, G., PERDON, A. e ZANOLI, S. **A navigation and inspection systems for ROVs.** *OCEANS '95. MTS/IEEE. 'Challenges of Our Changing Global Environment'. Conference Proceedings.*, 1:354–359, October 1995.
- CORREIA, F. A. C. **Um Sistema de Tele-Operação para um Manipulador Mestre-Escravo 3 GDL com Realimentação Háptica.** Tese de Doutorado, Instituto Militar de Engenharia - IME-RJ, 2003.

- ELHAJJ, I., XI, N., LIU, Y. H. e ET AL. **Haptic Information in Internet-Based Teleoperation.** *IEEE/ASME Transactions on Mechatronics*, 6(3):295–304, Setember 2001.
- FASSE, E. D. e HOGAN, N. **Quantitative Measurement of Haptic Perception.** *IEEE*, págs. 3199–3204, 1994.
- FERNANDES, J. A. S. **Manipulador Virtual Teleoperado: Transmissão da percepção háptica à distância.** Tese de Doutorado, Instituto Militar de Engenharia - IME-RJ, 2002.
- FERNANDES, J. A. S., R. P. F. F. e N., B. **Teleoperated Manipulator: The Remote Haptic Perception.** *WCETE World Congress on Engineering and Tecnology Education*, págs. 375–379, March 2004.
- FU, K. S., GONZALEZ, R. C. e LEE, C. S. G. **ROBOTICS: Control, Sensing, Vision and Intelligence.** McGraw-Hill Book Companies, 1987.
- GASPARINI, ANTEU FABIANO L; BARRELA, F. E. **TCP/IP Solução para Conectividade.** Editora Érica, 1993.
- HANNAFORD, B. **A design framework for teleoperators with kinesthetic feedback.** *IEEE Transactions on Robotics and Automation*, 5(4):426–434, August 1989.
- HER, M. G., HSU, K. S., YU, W. S. e KARKOUB, M. **Analysis and design of haptic telerobotic system.** *IEE Proc. -Control Theory Appl.*, 148(4):306–314, July 2001.
- HUI, R., OUELLET, A., WANG, A. e ET AL. **Mechanisms for Haptic Feedback.** *IEEE International Conference on Robotics and Automation*, págs. 2138–2143, 1995.
- JUNIOR, A. FERNADO DA ROCHA; ROSA, P. F. F. **Manipulador Bilateral Teleoperado: Transmissão da percepção háptica a distância.** Technical report, 2005.
- KUROSE, J. F. **Redes de Computadores e a Internet: uma nova abordagem.** Addison Wesley, 2003.
- LARKUM, T. e BROOME, D. **Advanced controller for an underwater manipulator.** *Control Applications, 1994., Proceedings of the Third IEEE Conference on*, 2: 1081–1086, August 1994.
- LAWRENCE, D. A. **Designing Teleoperator Architectures for Tranparency.** *in Proceedings of the IEEE International Conference on Robotics and Automation, (Nice, France)*, págs. 1406–1411, May 1992.
- MALINOWSKI, A.; KONETSKI, T. D. e B. SCHERTZ, D. **Web-Controlled Robotic Manipulator using Java and Client-Server Architecture.** *IEEE*, págs. 827–830, 1999.
- MARESCAUX, J.; SMITH, M. K., D., F. e AT AL. **Telerobotic Laparoscopic Cholecystectomy: Initial Clinical Experience With 25 Patients.** *Annals of Surgery*, 234(1):1–7, July 2001.

- MASCARO, STEPHEN.; ASADA, H. H. **Virtual Switch Human-Machine Interface Using Fingernail Touch Sensors.** *Proceedings of the 1999 IEEE International Conference on Robotics & Automation Detroit, Michigan*, págs. 2533–2538, May 1999.
- MISHKIN, A., MORRISON, J., NGUYEN, T. e ET AL. **Experiences with operations and autonomy of the Mars Pathfinder Microrover.** *Aerospace Conference, 1998. Proceedings., IEEE*, 2:337–351, March 1998.
- NIKI, T. e SHIMJO, M. **A development of simple haptic display.** *IEEE Transactions on Systems, Man, and Cybernetics*, págs. 1001–1006, 2000.
- OBOE, R. e PIOVAN, S. **Sensorless force reflecting teleoperation for low cost web-interfaced systems.** *IEEE, ISIE'99 - Bled, Slovenia*, págs. 123–128, 1999.
- PAUL, R. P. **Robot Manipulators: Mathematics, Programming and Control.** The MIT Press, Cambridge, Massachusetts and London, England, 1981a.
- PAUL, RICHARD P., S. B. e MAYER, G. E. **Control Equations for Simple Manipulators.** *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-11(6), June 1981b.
- RICHARD, P. e COIFFET, P. **Dextrous Haptic Interaction in Virtual Environments: Human Performance Evaluations.** *Proceedings of the 1999 IEEE International Workshop on Robot and Human Interaction Pisa, Italy*, págs. 315–320, Setember 1999.
- ROSA, PAULO F. F.; CHAGAS, F. S. C. F. A. C. **CMSE-3GDL: Development of a robotic telemanipulator bilateral system.** *In: VI IEEE/Induscon, 2004, Joinville-SC. VI Conferência Internacional de Aplicações Industriais*, 1:1415–1420, 2004a.
- ROSA, PAULO F. F.; CHAGAS, F. S. C. F. A. C. **CMSE-3GDL: O Desenvolvimento de um Sistema Bilateral de Telemanipulação Robótica.** *In: SBC/IEEnRi - I Encontro de Robótica Inteligente, Salvador. SBC2004. São Paulo-SP : Tec Art Editora Ltda*, 1:1727–1736, 5 e 6 Agosto 2004b.
- ROSA, PAULO F. F.; CHAGAS, F. S. F. J. A. S. B. N. **Development of a bilateral device for telemanipulation with tactile feedback.** *In: VI IEEE/Induscon, 2004, Joinville-SC. VI Conferência Internacional de Aplicações Industriais*, 1:1399–1403, 2004c.
- ROSA, PAULO. F. F.; OKADA, T. **3-D Firm Grasps with the Scrollic Gripper.** *IEEE International conference on Robotics and Automation*, págs. 1662–1667, 1995.
- SCHALAKE, B. e NARAYANAN, M. **Virtual Reality Applications.** *Department of Manufacturing Engineering, Miami University, School of Applied Science*, págs. 76–81, 1993.
- SCIAVICCO, L. e SICILIANO, B. **Modeling and Control of Robot Manipulators.** The McGraw-Hill Companies, Inc., Università degli Studi di Napoli Federico II Naples, Italy, 1996.

- SHERMAN, A.; ÇAVUŞOĞLU, M. e TENDICK, F. **Comparison of teleoperator control architectures for palpation task.** *Proceedings on IMECE'00, Symp. on Haptic Interfaces for Virtual Environments and Teleoperator Systems, Orlando, Florida*, November 2000.
- SITTI, M. e HASHIMOTO, H. **Teleoperated Touch Feedback From the Surfaces at the Nanoscale: Modeling and Experiments.** *IEEE/ASME Transactions on Mechatronics*, 8(2):287–298, June 2003.
- STOIANOVICI, D. **Robotic Surgery.** *World J Urol*, págs. 289–295, 2000.
- TANENBAUM, A. S. **Computer Networks.** Upper Saddle River: Prentice Hall, 1996.
- TAYLOR, K., D. B. **Internet Robots: A New Robotics Niche from Interfaces to Operator Behavior Technology Considerations for Expanding the Robotics "Web".** *IEEE Robotics & Automation Magazine*, págs. 27–34, March 2000.
- TEAM, R. **Characterization of the Martian Surface Deposits by the Mars Pathfinder Rover, Sojourner.** *Science*, 278(5344):1765–1768, December 1997.
- VOLPE, R. A. **Real and Artificial Forces in the Control of Manipulators: Theory and Experiments.** Tese de Doutorado, Department of Physics, The Robotics Institute - Carnegie Mellon University, 1990.
- VOYLES, RICHARD. M.; FEDDER, G. e KHOSLA, P. K. **Design of a Modular Tactile Sensor and Actuator Based on an Electrorheological Gel.** *In Proceedings of the 1996 IEEE International Conference on Robotics and Automation, Minneapolis, MN*, págs. 13–17, April 1996.
- YAMASHITA, H.; KIM, D. H. N. e T., D. **Multi-Slider Linkage Mechanism for Endoscopic Forceps Manipulador.** *Inc Proc. of 2003 IEEE/RSJ, Intl. Convergence on Intelligent Robots and Systems*, 3:2577–2582, October 2003.
- ZHENG, J. M.; CHAN, K. W. e I., G. **Virtual Reality, A real World on a somewhat touchy subject.** *IEEE Potentials*, págs. 20–23, April/May 1998.
- ZHU, M.; SALCUDEAN, S. E. **Achieving Transparency for Teleoperator Systems under Position and Rate Control.** *IEEE/RSJ International Conference on Intelligent Robots and Systems, Pittsburgh, PA*, 5-9 August 1995.

12 APÊNDICES

12.1 APÊNDICE 1: O JACOBIANO DO MANIPULADOR

Para o manipulador proposto, as mudanças diferenciais na posição e orientação em T_3 são causadas pelas mudanças diferenciais nas coordenadas das juntas dq_i . No caso das juntas de revolução, dq_i corresponde a rotação diferencial $d\theta_i$, e para o caso de uma junta prismática, dq_i corresponde a mudança diferencial na distância da junta dd_i , PAUL (1981a).

A EQ. 12.1, mostra a matriz $J_{6 \times 3}$ que indica as relações diferenciais das coordenadas das juntas.

$$\begin{bmatrix} T_3 d_x \\ T_3 d_y \\ T_3 d_z \\ T_4 \delta_x \\ T_5 \delta_y \\ T_6 \delta_z \end{bmatrix} = \begin{bmatrix} T_3 d_{1x} & T_3 d_{2x} & T_3 d_{3x} \\ T_3 d_{1y} & T_3 d_{2y} & T_3 d_{3y} \\ T_3 d_{1z} & T_3 d_{2z} & T_3 d_{3z} \\ T_4 \delta_{1x} & T_4 \delta_{2x} & T_4 \delta_{3x} \\ T_5 \delta_{1y} & T_5 \delta_{2y} & T_5 \delta_{3y} \\ T_6 \delta_{1z} & T_6 \delta_{2z} & T_6 \delta_{3z} \end{bmatrix} \begin{bmatrix} dq_1 \\ dq_2 \\ dq_3 \\ dq_4 \\ dq_5 \\ dq_6 \end{bmatrix} \quad (12.1)$$

As EQ. 12.2 e EQ. 12.3 são utilizadas para juntas de rotação e as EQ. 12.4 e EQ. 12.5 são utilizadas quando as juntas são prismáticas.

$${}^{T_3}d_i = (-n_x p_y + n_y p_x)i(-o_x p_y + o_y p_x)j(-a_x p_y + a_y p_x)k \quad (12.2)$$

$${}^{T_3}\delta_i = n_z i + o_z j + a_z k \quad (12.3)$$

$${}^{T_3}d_i = n_z i + o_z j + a_z k \quad (12.4)$$

$${}^{T_3}\delta_i = 0i + 0j + 0k \quad (12.5)$$

A primeira e a terceira coluna do jacobiano correspondem a $d\frac{T_3}{\theta_1}$ e $d\frac{T_3}{\theta_3}$ e são obtidas através das EQ. 12.2 e EQ. 12.3 aplicadas às matrizes T_3 e $d\frac{T_3}{\theta_3}$, respectivamente. A segunda coluna corresponde a $d\frac{T_3}{l_2}$ e é obtida através das EQ. 12.4 e EQ. 12.5 aplicadas a matriz 1T_3 .

12.2 APÊNDICE 2: LAGRANGIANO

Para as juntas de rotação os vetores p_i^d e p_i^δ são obtidos conforme as EQ. 12.6 e EQ. 12.7. Para as juntas prismáticas utilizam-se as EQ. 12.8 e EQ. 12.9

$${}^p d_i = \begin{bmatrix} -(^{i-1}n_x)(^{i-1}p_y) + (^{i-1}n_y)(^{i-1}p_x) \\ -(^{i-1}o_x)(^{i-1}p_y) + (^{i-1}o_y)(^{i-1}p_x) \\ -(^{i-1}a_x)(^{i-1}p_y) + (^{i-1}a_y)(^{i-1}p_x) \\ 1 \end{bmatrix}, \quad (12.6)$$

$${}^p \delta_i = \begin{bmatrix} -(^{i-1}n_z) \\ -(^{i-1}o_z) \\ -(^{i-1}a_z) \\ 0 \end{bmatrix} \quad (12.7)$$

$${}^p d_i = \begin{bmatrix} -(^{i-1}n_z) \\ -(^{i-1}o_z) \\ -(^{i-1}a_z) \\ 1 \end{bmatrix}, \quad (12.8)$$

$${}^p \delta_i = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (12.9)$$

$$D_{ii} = \sum_{p=i}^3 = m_p [{}^p \delta_{ix}^2 {}^p k_{pxx}^2 + {}^p \delta_{iy}^2 {}^p k_{pyy}^2 + {}^p \delta_{iz}^2 {}^p k_{pzz}^2] + [{}^p d_i \cdot {}^p d_i] + [2^p \bar{r}_p ({}^p d_i \times {}^p \delta_i)] \quad (12.10)$$

$$\Delta_{ij} = 2^p \bar{r}_p ({}^p d_i \times {}^p \delta_j + {}^p d_j \times {}^p \delta_i) \quad (12.11)$$

$$D_{ij} = \sum_{p=i}^3 = m_p [{}^p \delta_{ix} {}^p k_{pxx}^2 {}^p \delta_{jx} + {}^p \delta_{iy} {}^p k_{pyy}^2 {}^p \delta_{jy} + {}^p \delta_{iz} {}^p k_{pzz}^2 {}^p \delta_{jz}] + [{}^p d_i \cdot {}^p d_j] + [\Delta_{ij}] \quad (12.12)$$

$$D_i = {}^{i-1} g \sum_{p=i}^3 = m_p ({}^{i-1} r_p) \quad (12.13)$$

$${}^{i-1}\bar{r}_p = {}^{i-1}T_i({}^p\bar{r}_p) \quad (12.14)$$

Sendo que $\bar{r} \implies$ é a posição do centro de massa do elo. E $g \implies$ é o vetor que representa a aceleração da gravidade.

Sejam,

$$a = a_x i + a_y j + a_z k \quad (12.15)$$

$$b = b_x i + b_y j + b_z k \quad (12.16)$$

O produto interno ($a \cdot b$) é dado pela EQ. 12.17

$$a \cdot b = a_x b_x + a_y b_y + a_z b_z \quad (12.17)$$

O produto externo ($a \times b$) é dado pela EQ. 12.18

$$a \times b = (a_y b_z - a_z b_y)i + (a_z b_x - a_x b_z)j + (a_x b_y - a_y b_x)k \quad (12.18)$$

12.3 APÊNDICE 3: VARIÁVEIS DE ESTADOS

Para um sistema mola-massa-amortecedor, FIG. 12.1, onde B é o coeficiente de atrito viscoso, K é a constante de elasticidade da mola, m é a massa do bloco e X é o deslocamento que ocorre quando uma força F é aplicada ao sistema.

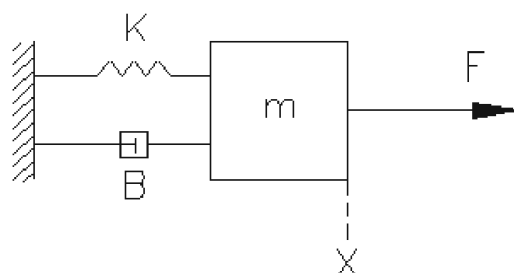


FIG 12.1: Sistema mola-massa-amortecedor.

A equação diferencial do sistema é dado pela EQ. 12.19.

$$F = m\ddot{x} + B\dot{x} + Kx, \quad (12.19)$$

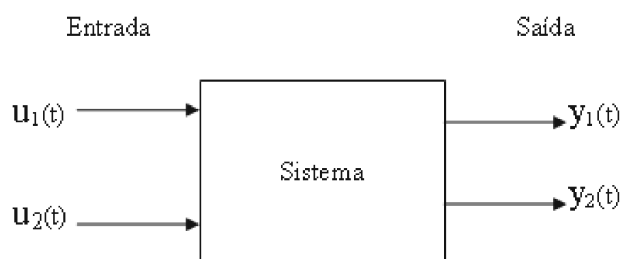


FIG 12.2: Representação da caracterização de um sistema de controle no qual $u_i(t)$ são as variáveis de entrada em função do tempo e $y_i(t)$ são as variáveis de saída.

O sistema da FIG. 12.1 pode ser escrito na forma de variáveis de estado como descrito na EQ. 12.2, sendo assim pode-se escrever as EQ. 12.20, EQ. 12.21, EQ. 12.22 e EQ. 12.25. seja:

$$F = u(t), \quad (12.20)$$

$$x = y(t), \quad (12.21)$$

$$x_1 = y(t), \quad (12.22)$$

$$x_2 = \frac{dy(t)}{dt} = \dot{x}, \quad (12.23)$$

Por analogia a EQ. 12.19 pode ser escrita como mostra a EQ. 12.24

$$u(t) = m\ddot{x}_2 + B\dot{x}_2 + Kx_1, \quad (12.24)$$

A EQ. 12.24 pode ser escrita como a EQ. 12.25.

$$\dot{x}_2 = -\frac{B}{m}x_2 - \frac{Kx_1}{m} + \frac{u(t)}{m} \quad (12.25)$$

Assim, as EQ. 12.22 e EQ. 12.25 são equações de saída e entrada do sistema.

As equações diferenciais de estado são escritas como mostradas na EQ. 12.26 e na EQ. 12.27.

$$\dot{x} = Ax + Bu \quad (12.26)$$

$$y = Cx + Du \quad (12.27)$$

Dessa forma, o sistema mola-massa-amortecimento mostrado na 12.1, pode ser escrito na forma matricial como mostram as EQ. 12.28 e EQ. 12.29.

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ -\frac{k}{m} & -\frac{B}{m} \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix} u \quad (12.28)$$

$$y = \begin{bmatrix} 1 & 0 \end{bmatrix} x \quad (12.29)$$

12.4 APÊNDICE 4: ESQUEMA DAS LIGAÇÕES ELÉTRICAS DO MVT

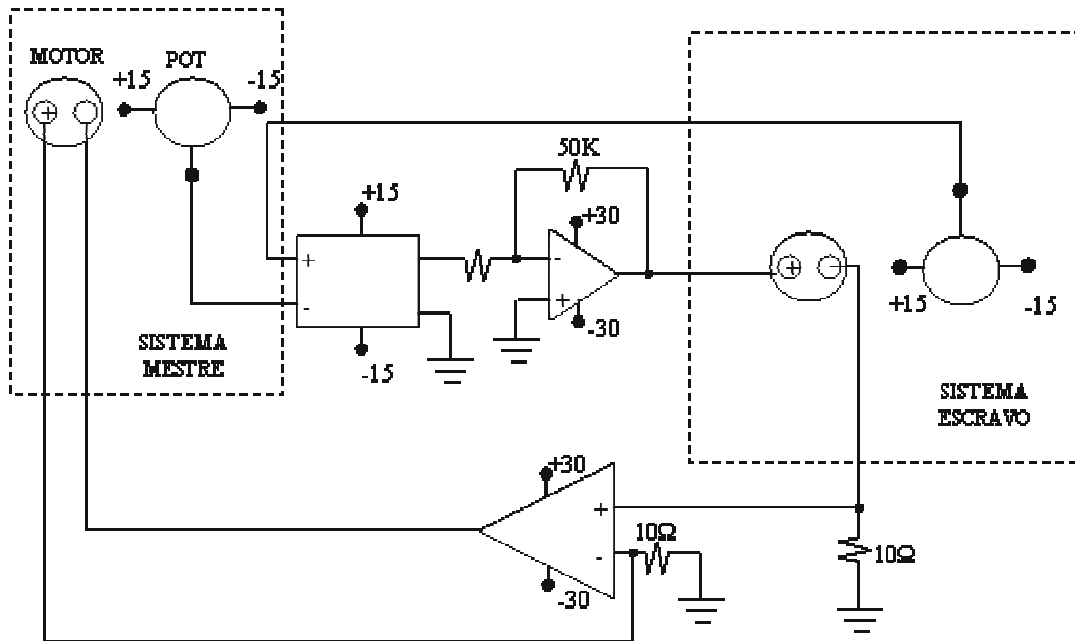


FIG 12.3: Esquema das ligações elétricas do Manipulador Virtual Teleoperado - MVT.

12.5 APÊNDICE 5: VALORES PARA O DIAGRAMA DE BLOCOS

Os valores para as funções de cada bloco que compõem o diagrama de blocos na FIG. 6.7.

$$m_m = 837.218e^{-6} Kg.m^2, \quad (12.30)$$

$$m_e = 2.584e^{-6} Kg.m^2, \quad (12.31)$$

$$B_m = 60.9408e^{-6} N.m/rad, \quad (12.32)$$

$$B_e = 60.9408e^{-6} N.m/rad, \quad (12.33)$$

$$K_e = 3.5 N.m/rad, \quad (12.34)$$

$$K_f = 1.0, \quad (12.35)$$

$$C_s = 60.9408e^{-6} + \frac{3.5}{s}, \quad (12.36)$$

$$Z_m = 837.218e^{-6} + \frac{60.9408e^{-6}}{s}, \quad (12.37)$$

$$Z_e = 2.584e^{-6} + \frac{60.9408e^{-6}}{s}, \quad (12.38)$$

12.6 APÊNDICE 6: CÁLCULO DA FUNÇÃO DE TRANSFERÊNCIA PARA OS BLOCOS EM CONTATO

A partir da EQ. 6.9, EQ. 6.10, EQ. 6.11, EQ. 6.12 e da EQ. 6.9 obtêm-se o diagrama de blocos da FIG. 12.4. Para encontrar a função de transferência, adota-se deslocar o ponto de soma rotulado como (2) para a esquerda do bloco, obtendo-se o diagrama de blocos modificado, como mostra a FIG. 12.5.

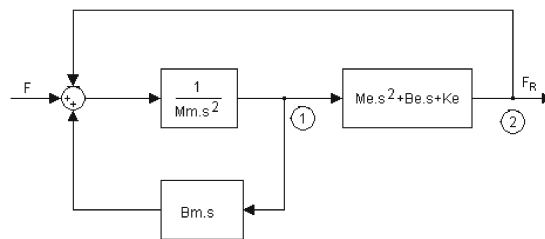


FIG 12.4: Diagrama de blocos para o modelo mecânico da FIG. 6.4

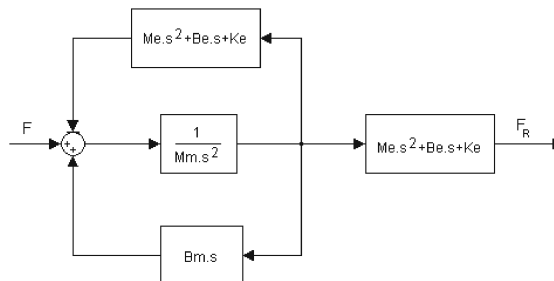


FIG 12.5: Diagrama de blocos rearranjado a partir da FIG. 12.4

Resolvendo o laço inferior, tem-se o diagrama da FIG. 12.7

Eliminando o laço de retroação do diagrama apresentado na FIG. 12.7. tem-se o diagrama da FIG. 12.10.

$$\frac{F_R}{F} = \frac{\frac{1}{m_m s^2}}{1 + \frac{1}{m_m s^2} B_m s} = \frac{1}{m_m s^2 + B_m s}, \quad (12.39)$$

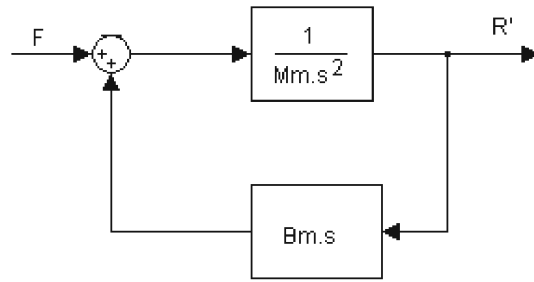


FIG 12.6: Representação do laço inferior do diagrama da FIG. 12.5

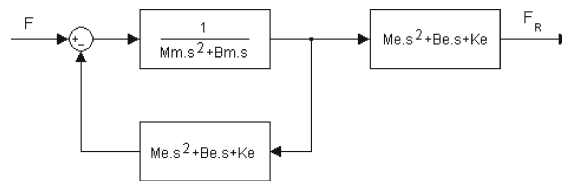


FIG 12.7: Diagrama de blocos modificado, resultado da eliminação do laço inferior da FIG. 12.5

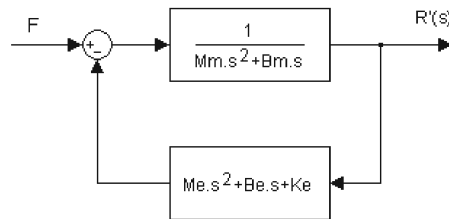


FIG 12.8: Laço com retroação, extraído da FIG. 12.7.

$$\frac{F_R}{F} = \frac{\frac{1}{m_m s^2 + B_m s}}{1 + \frac{1}{[m_m s^2 + B_m s]}(m_e s^2 + B_e s + K_e)} = \frac{1}{m_m s^2 + B_m s + m_e s^2 + B_e s + K_e}, \quad (12.40)$$

logo, a a função de transferência para o sistema mecânico apresentado na FIG. 6.1 é dada pela EQ. 12.41.

$$\frac{F_R}{F} = \frac{m_e s^2 + B_e s + K_e}{(m_m + m_e) s^2 + (B_m + B_e) s + K_e}, \quad (12.41)$$

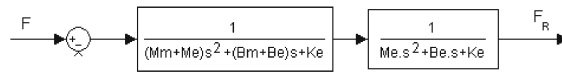


FIG 12.9: Diagrama de blocos representando a EQ. 12.40.

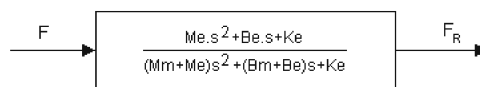


FIG 12.10: Diagrama de blocos final para a função de transferência dada pela EQ. 12.41.