



TELEOPERAÇÃO BILATERAL DE MÚLTIPLOS ROBÔS APLICADA AO TRANSPORTE DE CARGA

Gabriel Casulari da Motta Ribeiro

Dissertação de Mestrado apresentada ao Programa de Pós-graduação em Engenharia Elétrica, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Mestre em Engenharia Elétrica.

Orientador: Fernando Cesar Lizarralde

Rio de Janeiro
Agosto de 2013

TELEOPERAÇÃO BILATERAL DE MÚLTIPLOS ROBÔS APLICADA AO
TRANSPORTE DE CARGA

Gabriel Casulari da Motta Ribeiro

DISSERTAÇÃO SUBMETIDA AO CORPO DOCENTE DO INSTITUTO ALBERTO LUIZ COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE ENGENHARIA (COPPE) DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIAS EM ENGENHARIA ELÉTRICA.

Examinada por:

Prof. Fernando Cesar Lizarralde, D.Sc

Prof. Liu Hsu, Dr. d'État

Prof. Guilherme Augusto Silva Pereira, D.Sc.

RIO DE JANEIRO, RJ – BRASIL
AGOSTO DE 2013

Ribeiro, Gabriel Casulari da Motta

Teleoperação Bilateral de Múltiplos Robôs aplicada ao Transporte de Carga/Gabriel Casulari da Motta Ribeiro.
– Rio de Janeiro: UFRJ/COPPE, 2013.

XVIII, 124 p.: il.; 29, 7cm.

Orientador: Fernando Cesar Lizarralde

Dissertação (mestrado) – UFRJ/COPPE/Programa de Engenharia Elétrica, 2013.

Referências Bibliográficas: p. 81 – 88.

1. Teleoperação. 2. Robôs Cooperativos. 3. Transporte de carga. I. Lizarralde, Fernando Cesar. II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia Elétrica. III. Título.

Agradecimentos

Primeiramente a minha família que sempre me apoiou e continuará apoiando, mesmo com oscilações de humor e nervos a flor da pele.

Um agradecimento especial ao meu orientador neste trabalho Fernando Lizarralde pelo apoio na definição do tema e para que o conteúdo atingisse a consistência agora apresentada. Parte de seus conselhos vão seguir comigo nos próximos passos de meu desenvolvimento. Com este mesmo pensamento, ofereço ao membros da banca, Liu e Guilherme, um muito obrigado pelos elogios e críticas. Estas contribuíram para melhorar o trabalho final e irão contribuir para trabalhos futuros.

Uma lembrança muito especial aos amigos de graduação que serviram de suporte acadêmico e foram companheiros para os momentos de descontração, sem esta turma o curso de Controle e Automação não teria a mesma validade. Aqui, entram também os colegas de mestrado, um desafio que não deve ser enfrentado sozinho. Menções especiais vão para o Thiago, que foi companheiro no projeto da PETROBRAS que correu junto à dissertação, e ao Fernando Coutinho. Este chegou no final do caminho, mas foi um qualificado ajudante para os testes experimentais.

Agradeço ao CNPq, a FAPERJ e a CAPES pelo suporte financeiro, com a bolsa e o apoio aos projetos desenvolvidos no laboratório e que de alguma forma contribuíram para realização do presente trabalho.

Agradeço, também, a equipe COPPE \TeX por criar a classe com o modelo COPPE de forma que eu não precisei olhar para aquele manual de formatação ao escrever esta dissertação.

Por fim, mas não menos importante, aos demais membros do LABCON e do LEAD, sejam alunos, técnicos, porteiros ou professores todos contribuíram de alguma maneira durante a pesquisa.

Resumo da Dissertação apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

TELEOPERAÇÃO BILATERAL DE MÚLTIPLOS ROBÔS APLICADA AO TRANSPORTE DE CARGA

Gabriel Casulari da Motta Ribeiro

Agosto/2013

Orientador: Fernando Cesar Lizarralde

Programa: Engenharia Elétrica

Este trabalho apresenta uma proposta para a teleoperação bilateral cooperativa de múltiplos manipuladores transportando uma carga, considerando atraso nos canais de comunicação. A teleoperação cooperativa exige, além da transparência e estabilidade, a garantia da restrição estabelecida pela interação entre robôs durante a tarefa. Entre as diferentes estratégias de teleoperação, considera-se uma arquitetura em duas camadas verticais. Uma camada é utilizada para tratar a transparência, isto é, a coordenação entre mestres e escravos e a realimentação de força para o operador, e a outra camada garante que não seja gerada nenhuma energia virtual. A cooperação entre os escravos é tratada como um problema de formação, onde as forças de contato entre os robôs e a carga são modeladas como gradientes de uma função potencial. A força de pegada necessária para manter o objeto seguro é controlada diretamente no ambiente local, onde se realiza a tarefa, não sendo afetada por atrasos na comunicação. São propostas leis de controle cinemática e dinâmica, de forma a controlar robôs industriais ou diretamente atuados. A teleoperação é testada através de simulação e procedimentos experimentais. Neste caso, utiliza-se uma arquitetura aberta para prototipagem rápida de aplicações de teleoperação desenvolvida durante o trabalho.

Abstract of Dissertation presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

BILATERAL TELEOPERATION OF MULTIPLE COOPERATIVE ROBOTS FOR LOAD TRANSPORT

Gabriel Casulari da Motta Ribeiro

August/2013

Advisor: Fernando Cesar Lizarralde

Department: Electrical Engineering

In this work the bilateral teleoperation of multiple cooperative robots is addressed. A control framework for the teleoperation between multiple master/slave robots is proposed for collaboratively perform a task. A recently proposed two-layer approach where the control architecture is separated into two layers is considered. The top layer is used to address the transparency, that is, the coordination between master and slave and the correct operator force feedback, and the lower layer guarantees that no virtual energy is generated. As a cooperative task, the load transportation problem is considered which is treated as a formation control problem, where the contact force between robots and the payload are modeled as gradients of nonlinear potentials. A decentralized ultimately bounded control scheme is proposed for controlling each one of the robot manipulators. Without explicit communication between the robots, all slave manipulators and the payload track a velocity reference with bounded contact forces. Numerical simulation and experimental results using a Phantom Omni robot or Kinect device and a Motoman dual-arm robot illustrate the feasibility of the proposed approach.

Sumário

Lista de Figuras	x
Lista de Tabelas	xiv
Lista de Símbolos	xv
Lista de Abreviaturas	xvii
1 Introdução	1
1.1 Motivação	4
1.2 Revisão da Literatura	6
1.2.1 Teleoperação	6
1.2.2 Manipuladores Cooperativos	16
1.2.3 Transporte de Cargas	19
1.3 Objetivos	22
1.4 Metodologia	22
1.5 Estrutura do Texto	23
2 Modelagem de Manipuladores	25
2.1 Modelo Cinemático	25
2.1.1 Cinemática diferencial	28
2.2 Modelo Dinâmico	29
2.3 Múltiplos manipuladores	30
3 Teleoperação Cooperativa	31
3.1 Definição do problema	31
3.2 Camada de Transparência	33
3.3 Camada de Passividade	35
4 Controle Cooperativo	39
4.1 Modelagem	40
4.2 Controle Cinemático	42
4.2.1 Condição de estabilidade	42

4.2.2	Estratégia de controle	43
4.3	Controle Dinâmico de Posição	48
4.3.1	Estratégia de Controle	49
4.4	Controle Dinâmico de Posição e Orientação	51
4.4.1	Condição de estabilidade	52
4.4.2	Estratégia de controle	53
5	Resultados Experimentais	59
5.1	Simulação	59
5.1.1	Modelagem	59
5.1.2	Resultados	62
5.2	Testes Experimentais	68
5.2.1	Montagem experimental	68
5.2.2	Operação autônoma	69
5.2.3	Teleoperação unilateral	72
6	Conclusões	77
6.1	Limitações	78
6.2	Trabalhos futuros	79
	Referências Bibliográficas	81
A	Arquitetura de Teleoperação	89
A.1	Comunicação	90
A.1.1	Robot Raconteur	91
A.2	Infraestrutura Local	93
A.2.1	Sistema de controle	94
A.2.2	Câmera	95
A.3	Infraestrutura Remota	95
A.3.1	Kinect	96
A.3.2	Phantom Omni	97
A.3.3	Servovisão	98
A.3.4	Interface Gráfica	99
A.4	Testes Experimentais	101
A.5	Observações dos Experimentos	103
B	Teoremas auxiliares	106
B.1	Prova do Teorema 1	106
B.2	Lema 4.3 de Khalil (2002)	107
B.3	Prova do Teorema 4	108

C	Manual de utilização - Motoman DIA10	109
C.1	Características cinemáticas	110
C.2	Instalação	112
C.3	Controle via <i>Teach Pendant</i>	114
C.4	Controle via computador	115
C.4.1	HSC	116
C.4.2	Robot Raconteur	117
C.5	Operação	118
C.5.1	Inicialização	118
C.5.2	Conexão com computador	121

Lista de Figuras

1.1	Exemplos de robôs industriais executando tarefas pré-programadas. . .	1
1.2	Exemplos de aplicação de teleoperação robótica.	2
1.3	Primeiro dispositivo de teleoperação, descrito por Goertz em 1954. . .	3
1.4	Proposta de novo modelo de plataforma de exploração de petróleo em áreas remotas.	5
1.5	Exemplo de operação em plataformas de petróleo que necessita de cooperação entre dispositivos robóticos.	6
1.6	Diagramas de blocos para teleoperação com controle supervisorio. . .	7
1.7	Diagramas de blocos para teleoperação com controle compartilhado. .	7
1.8	Diagramas de blocos para teleoperação com controle direto.	8
1.9	Diagramas com representações de teleoperações na modalidade direta.	8
1.10	Representação de teleoperação como um conjunto de sistemas de 4 portas interligados.	9
1.11	Representação de teleoperação utilizando a teoria de espalhamento como um conjunto de sistemas de 4 portas interligados.	10
1.12	Diagrama de blocos do controle de teleoperação com passividade no domínio do tempo.	12
1.13	Diagrama de blocos da estratégia de teleoperação em duas camadas. .	14
1.14	Diagrama representando os sistemas de controle centralizado e descentralizado com comunicação direta e indireta.	19
1.15	Exemplos de condições de pegada.	20
1.16	Exemplo de transporte de carga utilizando enjaulamento.	21
2.1	Representação esquemática de um manipulador robótico.	25
2.2	Representação de dois sistemas de coordenadas ortonormais.	27
2.3	Exemplo de um manipulador que possui mais de uma solução de cinemática inversa para a mesma pose.	28
3.1	Representação do problema de transporte de carga abordado.	31
3.2	Diagrama representando as camadas de controle.	32
3.3	Representação dos manipuladores em uma teleoperação com $N = 4$. .	34

3.4	Diagrama representando as camadas de transparência e passividade.	38
4.1	Diagrama de blocos representando o controle cooperativo descentralizado.	39
4.2	Representação da interação entre um manipulador e o objeto.	40
4.3	Diagrama de blocos da malha de controle cinemático de um manipulador na tarefa cooperativa.	44
4.4	Diagrama de blocos representando o controle dinâmico.	50
4.5	Diagrama de blocos representando a estratégia de controle em cascata.	51
4.6	Diagrama de blocos representando a proposta de controle em cascata em um manipulador do sistema cooperativo.	57
5.1	Representação do sistema escravo simulado ($N=2$), indicando os parâmetros do modelo.	60
5.2	Erro de velocidade da carga durante simulação de teleoperação com movimento circular, considerando $N = 2$ e $N = 4$	62
5.3	Erro de velocidade durante simulação de teleoperação com movimento circular.	62
5.4	Representação dos robôs mestre e escravo em diferentes instantes de tempo com e sem presença de amortecimento pelo nível de energia.	63
5.5	Magnitude das forças desejada e aplicada a carga, por cada robô, durante movimento circular.	64
5.6	Magnitude das forças desejada e aplicada a carga, por cada robô, durante movimento com velocidade constante.	64
5.7	Orientação dos manipuladores escravo durante a simulação de teleoperação com movimento circular.	65
5.8	Erro de velocidade durante simulação de teleoperação com movimento circular e controle em cascata.	65
5.9	Magnitude das forças desejada e aplicada a carga, por cada robô, durante movimento circular e controle em cascata.	66
5.10	Orientação dos manipuladores escravo durante a simulação de teleoperação com movimento circular e controle em cascata.	66
5.11	Representação dos robôs mestre e escravo em diferentes instantes de tempo para a simulação de controle em cascata.	67
5.12	Motoman DIA10 segurando uma caixa.	68
5.13	Erros de velocidade do efetuador do escravo 1 durante o controle autônomo com referência linear e circular.	70
5.14	Módulo das força virtuais nos efetuadores durante o controle autônomo com referência linear e circular.	70

5.15	Trajetória dos braços do Motoman DIA10 rastreando uma trajetória helicoidal.	71
5.16	Representação da orientação durante a trajetória helicoidal com dois ganhos para a tarefa secundária.	71
5.17	Forças virtuais medidas nos efetuadores durante a teleoperação utilizando Phantom Omni como mestre.	73
5.18	Referência de velocidade gerada pelo operador durante teleoperação unilateral com Phantom Omni.	74
5.19	Operador comandando o robô através do Kinect. A imagem a direita serve de auxílio, representando os pontos de esqueleto e uma caixa virtual.	74
5.20	Erro de velocidade dos efetuadores durante teleoperação utilizando o Kinect como mestre.	75
5.21	Forças medidas nos efetuadores durante a teleoperação utilizando Kinect como mestre.	76
A.1	Diagrama de interfaces entre as diferentes modalidades de entrada e o sistema de controle	90
A.2	Diagrama representativo das vias de comunicação para a arquitetura de teleoperação proposta	91
A.3	Diagrama representando os elementos da arquitetura do Robot Raconteur	92
A.4	Diagrama do procedimento de conexão entre nós do Robot Raconteur.	92
A.5	Diagrama esquemático de um sistema de controle utilizando RR e <i>XPC-Target</i>	95
A.6	Pontos do corpo que formam o esqueleto fornecido pelo Kinect.	96
A.7	Pontos do esqueleto fornecido pelo Kinect e utilizados para controle do manipulador	97
A.8	Phantom Omni com os eixos de rotação indicados.	98
A.9	Exemplo de imagem apresentada para o operador durante controle por servovisão.	98
A.10	Interface gráfica nas quatro configurações programadas.	100
A.11	Exemplos de figuras para auxílio do operador durante o uso do Kinect	101
A.12	Sistema de dois robôs PUMA560 sobre trilhos do CATS/RPI.	102
A.13	Exemplo do experimento com as diferentes modalidades de controle	103
A.14	Trajetórias das juntas 1, 2 e 3 do PUMA560 durante a movimentação mestre escravo com o Kinect e Phantom	104
A.15	Erro em pixels nos eixos x e y durante a servovisão.	105
C.1	Nomes das juntas e sistema de referência do robô DIA10.	110

C.2	Dimensões dos elos e limites de rotação das juntas do robô DIA10. . .	111
C.3	Controlador industrial NX100.	112
C.4	Diagrama para conexão dos cabos entre o robô e o controlador NX100.	113
C.5	<i>Teach Pendant</i>	115
C.6	Diagrama de conexão entre o HSC, o controlador NX100 e um computador.	116
C.7	Exemplo de malha de controle de juntas implementada em Simulink para o Motoman DIA10.	118
C.8	Inicialização do NX100.	119
C.9	Localização das ranhuras de referência e exemplo para o procedimento de calibração.	121

Lista de Tabelas

3.1	Referências calculadas no exemplo de camada de transparência.	35
5.1	Parâmetros do modelo de manipulador 3 <i>DoF</i> utilizado na simulação.	61
5.2	Parâmetros para simulação dos tanques de energia.	61
5.3	Trajetórias para os testes experimentais do controle de manipuladores cooperativos.	69
C.1	Parâmetros de Denavit-Hartenberg <i>standard</i> para o DIA10.	111
C.2	Parâmetros cinemáticos segundo a teoria de <i>Screw</i> para o DIA10.	112
C.3	Dados compartilhados entre o computador e o NX100 via HSC.	117
C.4	Ordem de apresentação das posições das juntas no serviço RR.	123

Lista de Símbolos

C	Matriz de Coriolis e forças centrípetas, p. 29
J	Jacobiano geométrico, p. 29
$J^\#$	Matriz de projeção no espaço nulo do Jacobiano, p. 43
J^\dagger	Pseudo-inversa do Jacobiano, p. 43
J_a	Jacobiano analítico, p. 28
J_o	Jacobiano de orientação, p. 29
J_p	Jacobiano de posição, p. 29
J_r	Jacobiano da representação, p. 28
M	Matriz de inércia, p. 29
M_c	Massa da carga, p. 41
\dot{x}	Velocidade linear, p. 28
ω	Velocidade angular, p. 28
τ	Torque aplicado as juntas do manipulador, p. 29
θ	Posição das juntas do manipulador, p. 27
ξ	Erro de velocidade, p. 44
a_i	Posição do efetuador sem deformar a carga, p. 40
f^{dm}	Referência de força do usuário, p. 33
f_i	Força de contato, p. 40
f_i^d	Referência de força, p. 32
g	Vetor de forças gravitacionais, p. 29

p	Pose do efetuador de um manipulador, p. 27
q_e	Parte escalar do quaternion, p. 26
q_v	Parte vetorial do quaternion, p. 26
r_i	Vetor entre x_c e a_i , p. 40
v^d	Referência de velocidade, p. 32
x_c	Posição do centro de massa da carga, p. 40
z_i	Deformação da carga, p. 40

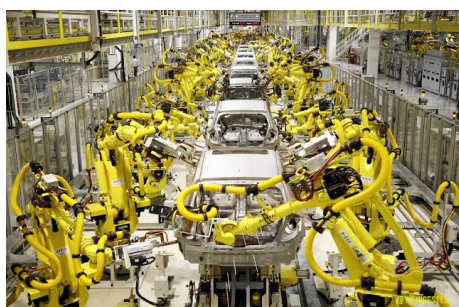
Lista de Abreviaturas

GUI	Interface gráfica (do inglês <i>Graphical User Interface</i>), p. 95
MMMS	do inglês, Multiple Masters Multiple Slaves, p. 18
RR	Robot Raconteur TM , p. 89
SMMS	do inglês, Single Master Multiple Slaves, p. 18
UB	do inglês Ultimately Bounded, p. 42

Capítulo 1

Introdução

Classicamente, dispositivos robóticos são utilizados na indústria para substituir trabalhadores humanos em tarefas repetitivas, que necessitam de grande precisão ou esforço físico, porém com baixo grau de exigência intelectual. Assim, o projeto de um robô industrial costuma ser realizado em conjunto com a planta de produção de forma que o dispositivo executa somente um pequeno conjunto de tarefas de forma contínua. Como exemplo tem-se a fabricação de automóveis e a montagem em série de circuitos eletrônicos, mostrados na Figura 1.1.



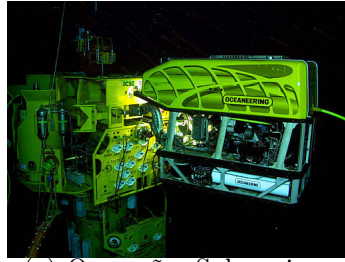
(a) Indústria automotiva



(b) Indústria eletrônica

Figura 1.1: Exemplos de robôs industriais executando tarefas pré-programadas.

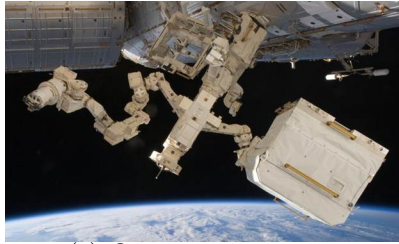
Outra aplicação para robótica consiste em utilizar robôs para operar ou realizar manutenção em áreas de difícil acesso, com baixas condições de segurança ou com risco a saúde do trabalhador. Neste segundo caso, é comum que um mesmo dispositivo tenha que realizar diferentes tarefas, nem sempre previstas e programadas com antecedência. Exemplos de robôs teleoperados são robôs submarinos, para cirurgias remotas (Ballantyne 2002, Cho et al. 2012), para operações espaciais (Hirzinger et al. 1994, Marescaux et al. 2002, Preusche et al. 2006) e utilizados em operações de despoluição ou controle ambiental, como o robô ambiental híbrido, proposto pela Petrobras (Freitas et al. 2010). Estes são apresentados na Figura 1.2.



(a) Operações Submarinas



(b) Áreas de difícil acesso



(c) Operações espaciais



(d) Telecirurgia

Figura 1.2: Exemplos de aplicação de teleoperação robótica.

Para obter um sistema capaz de se adaptar a situações adversas de um ambiente não estruturado ou de uma tarefa com características variáveis, é utilizada a teleoperação robótica, como nos casos da Figura 1.2. Esta consiste em um operador humano controlando diretamente o robô escravo a partir de um outro ambiente, possivelmente distante da área de trabalho. Este controle pode ser em alto nível, através de trajetórias desejadas e sequências de ações ou direto, onde os movimentos do escravo são transmitidos durante a execução, por exemplo, através de um outro dispositivo robótico denominado mestre (Niemeyer et al. 2008). Para que o operador realize a tarefa de forma correta, o mesmo é realimentado com informações visuais, auditivas, de força e outros sensores de forma que possa conhecer o ambiente onde o robô está presente.

Considera-se que os primeiros trabalhos sobre dispositivos para teleoperação são dois artigos publicados por Goertz em 1954 (Niemeyer et al. 2008). Ambos são motivados pela indústria nuclear, de forma a diminuir a exposição do trabalhador a radiação. O primeiro trabalho descreve um dispositivo mecânico, mostrado na Figura 1.3, onde os dispositivos mestre e escravo são acoplados diretamente em um mecanismo de quatro barras semelhante a um pantógrafo. O segundo trabalho inicia a maior separação entre ambiente local e remoto apresentando um mecanismo de teleoperação controlado eletronicamente.

Após este primeiro trabalho, a pesquisa em teleoperação nos anos 60 foi concentrada na relação entre o operador humano e o dispositivo robótico e na consequência da introdução desta nova ferramenta para a realização da tarefa (Hokayem & Spong 2006). Como exemplo, cita-se o trabalho de Sheridan & Ferrell (1963) que investiga experimentalmente a influencia da presença de atraso na comunicação na realização



Figura 1.3: Primeiro dispositivo de teleoperação, descrito por Goertz em 1954. O acoplamento entre mestre e escravo é mecânico, através de um mecanismo de quatro barras. Retirado de (Niemeyer et al. 2008)

da tarefa, observando a adoção por parte do operador de um movimento pausado. No fim dos anos 60, motivado pelo uso de robôs no espaço, Ferrell & Sheridan (1967) propõe uma alternativa para melhorar a performance da teleoperação na presença de atraso através do controle supervisorio de um sistema com capacidade de completar um pequeno conjunto de tarefas autonomamente.

Uma visão para teleoperação mais voltada a teoria de controle começou a ser introduzida em 1980, mas a partir da década de 90, com o advento da internet, o interesse sobre a teleoperação aumentou com as novas possibilidades proporcionadas por essa rede de comunicação (Hokayem & Spong 2006). O uso da internet, entretanto, acrescentou novos desafios para o projeto de leis de controle para a teleoperação. Devido à estrutura do canal de comunicação, o atraso é variável, podem haver perdas de pacotes e a transmissão digital apresenta quantização.

Como solução para a instabilidade resultante do canal de comunicação, são propostas leis de controle baseadas na teoria de redes, em passividade, controle adaptativo e controle ótimo (Hokayem & Spong 2006). Abordando outra questão relacionada a teleoperação, a capacidade de transparecer o ambiente onde ocorre a tarefa ao operador, Lawrence (1993), Yokokohji & Yoshikawa (1994) mostraram que esse objetivo é conflitante com a estabilidade. Em uma abordagem mais recentes, Franken et al. (2009) utilizam uma estrutura em duas camadas verticais, separando o projeto da estabilização e da transparência tentando obter a maior transparência possível sem perder a estabilidade.

Uma aplicação ainda pouco explorada em teleoperação é a utilização de múltiplos escravos, com um ou múltiplos mestres (Hokayem & Spong 2006). Essa tarefa cooperativa apresenta novos desafios, já que é preciso garantir a coordenação entre os robôs, entretanto possibilita novas operações e um aumento da carga a ser manipulada. Aplicações típicas de manipuladores cooperativos são transporte de cargas

grandes e/ou pesadas, montagem de múltiplas partes simultaneamente e manipulação de objetos flexíveis ou com partes móveis (Caccavale & Uchiyama 2008).

Fora da teleoperação, a cooperação entre robôs é uma assunto estudado desde 1970, com diferentes estratégias de controle propostas. Dentre estas, destacam-se o controle híbrido, o controle de força/complacência e a linearização por realimentação. Uma visão diferente para a cooperação robótica é apresentada por Bai & Wen (2010) que utilizam uma abordagem de formação, onde os robôs atingem uma mesma velocidade constante enquanto aplicam uma força sobre um objeto deformável.

Para este trabalho, a estrutura em duas camadas (Franken et al. 2009) para teleoperação bilateral será utilizada em uma tarefa cooperativa de transporte de carga, onde podem ser empregados múltiplos dispositivos mestre. Para a coordenação entre os escravos utiliza-se a estratégia de formação (Bai & Wen 2010), com o controle de pegada realizado localmente no ambiente da tarefa, não sofrendo diretamente a ação do atraso. São propostas leis de controle cinemático e dinâmico para contemplar robôs industriais e diretamente atuados. Destaca-se que esta integração representa uma extensão de ambos os trabalhos, considerando o caso cinemático para as camadas de teleoperação e o rastreamento de trajetórias de velocidade para o controle cooperativo.

1.1 Motivação

A teleoperação robótica tem uma motivação clara no aumento da segurança do trabalhador e redução de custos de logística. Isto porque pode evitar a necessidade da presença de humanos em tarefas e ambientes que apresentam risco a saúde, dificuldade de acesso e condições climáticas adversas.

No contexto brasileiro atual, uma área para aplicação deste tipo de tecnologia são as novas fronteiras de exploração de petróleo na região do pré-sal. A atividade de exploração apresenta riscos devido a presença de produtos inflamáveis, explosivos e tóxicos em uma região com poucas alternativas de evacuação e alta taxa de corrosão pela maresia. No pré-sal, devido a grande distância entre os poços e a plataforma continental, ocorre ainda um aumento no custo e no tempo de transporte de trabalhadores, não sendo possível em certas áreas uma viagem completa de ida e volta de um helicóptero sem a realização de escalas. Com isso, além do aumento dos custos de operação e reparo, a capacidade de resposta a emergências é deteriorada.

Uma possível alternativa para a exploração nesta região remota é reduzir o número de trabalhadores e o tempo de permanência destes na plataforma, utilizando robôs para operação da mesma (Anisi et al. 2010, 2011, From 2010, Graf & Pfeiffer 2007, Transeth et al. 2010). De forma extrema propõe-se um projeto de plataforma dividida e duas áreas: uma permanentemente não habitada por humanos, onde

ocorre a exploração, e outra parcialmente habitada que recebe trabalhadores para operações de manutenção e reparo, como na ilustração da Figura 1.4. Este conceito foi criado para ser utilizado em novos poços na Noruega, onde além dos riscos comuns a operação, o frio é intenso (Transeth et al. 2010).

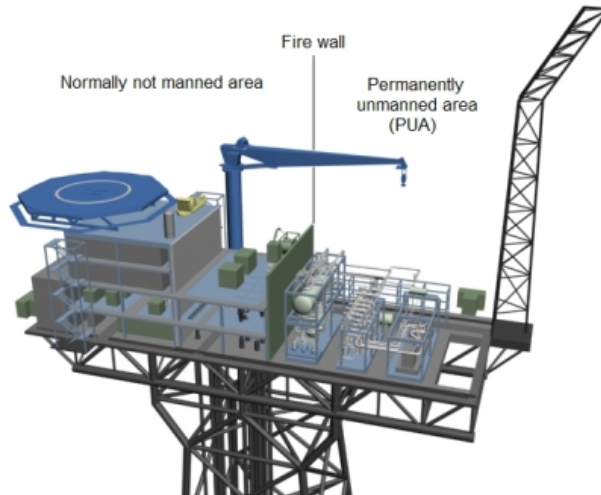


Figura 1.4: Proposta de novo modelo de plataforma de exploração de petróleo em áreas remotas. Apresenta uma divisão em duas áreas, uma constantemente não habitada e outra parcialmente habitada. Retirado de (Transeth et al. 2010).

No caso mais específico da teleoperação de múltiplos manipuladores em cooperação, a motivação inicial é teórica, visto que apresenta desafios adicionais ao caso de um único robô. Por exemplo, a coordenação entre os múltiplos mestres e escravos e o cumprimento da restrição. Apesar desse tópico ser apontado como uma tendência por (Hokayem & Spong 2006), somente quatro trabalhos foram encontrados na literatura (Aldana et al. 2012, Bacocco & Melchiorri 2009, Lee & Spong 2005, Sirouspour 2005).

Além da motivação teórica, a teleoperação cooperativa estende as possíveis aplicações da teleoperação. No contexto da exploração de petróleo, o desenho atual de plataformas exige que diversas operações sejam realizadas utilizando as duas mãos ou por dois trabalhadores em conjunto. A Figura 1.5 apresenta um exemplo com um operador acionando uma válvula.



Figura 1.5: Exemplo de operação em plataformas de petróleo que necessita de cooperação entre dispositivos robóticos. Devido a posição da válvula é necessário utilizar duas mãos para acioná-la de forma eficiente. Retirado de (Remiro 2009).

1.2 Revisão da Literatura

Neste trabalho, será desenvolvida uma estratégia de teleoperação cooperativa, onde múltiplos robôs escravos são comandados através de múltiplos robôs mestres para transportar uma carga em conjunto.

Nesta seção, será apresentada uma revisão de literatura sobre os principais pontos abordados. Primeiramente, uma descrição sobre teleoperação, considerando questões de estabilidade e diferentes arquiteturas utilizadas. Como a estratégia adotada no trabalho é baseada em passividade, esta categoria será abordada de forma mais detalhada. Na sequência, é realizada uma revisão sobre controle cooperativo e, por fim, é abordado o caso específico de cooperação para de transporte de carga.

1.2.1 Teleoperação

A teleoperação robótica consiste na realização de uma tarefa utilizando recursos dispostos em dois ambiente distintos, o local e o remoto. No primeiro estão presentes os dispositivos robóticos, sensores, atuadores e demais objetos necessários para a realização da tarefa. O operador, que planeja e executa a tarefa, está no ambiente remoto junto a dispositivos que fornecem realimentação sobre o estado do ambiente local. Desta forma, é possível se utilizar da capacidade de adaptação e percepção do homem para realização de tarefas complexas sem expor o mesmo a situações de risco e esforço extremo ou ambientes insalubres.

A teleoperação pode ser classificada pelo nível de controle exercido pelo humano em três modalidades: supervisório, compartilhado e direto. Na primeira modalidade o operador observa a ação do robô programando tarefas a serem cumpridas baseado nas leituras de sensores instalados no ambiente local (Niemeyer et al. 2008), como no diagrama da Figura 1.6. Assim, o manipulador é autônomo, com todas as malhas de controle implementadas no mesmo ambiente. Um exemplo são os robôs de exploração enviados a Marte. A equipe de terra define um caminho e pontos de

interesse para coleta de amostra, o robô executa este trajeto e envia as informações em pacotes.

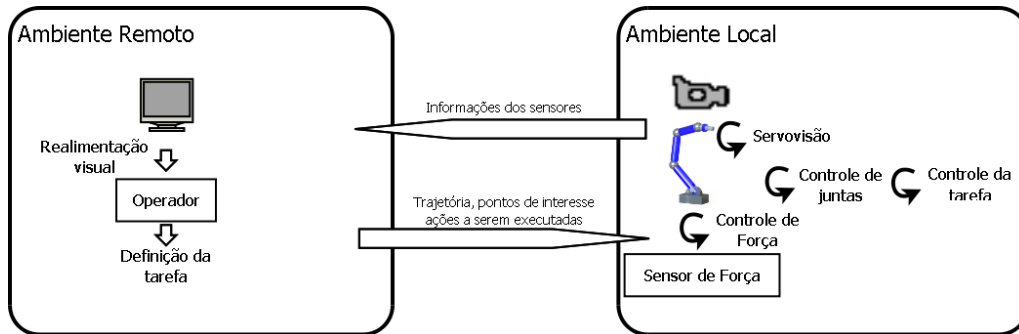


Figura 1.6: Diagramas de blocos para teleoperação com controle supervisorio.

Na modalidade de controle compartilhado (ver Figura 1.7) o operador atua diretamente para a execução da tarefa. Entretanto, os sensores no ambiente local são utilizados de forma a obter uma malha local de maior precisão (Niemeyer et al. 2008). Desta forma trajetórias executadas pelo operador são refinadas pelo robô. Pode ser estabelecido um limite de força ou artefatos virtuais podem ser inseridos na realimentação real de forma a auxiliar a realização da operação. Como exemplo, Cho et al. (2012) propõe uma estrutura em camadas para a realização de uma cirurgia remota. Entre os manipuladores mestre e escravo uma camada intermediária modifica os comandos para garantir a segurança da cirurgia e a realimentação para aumentar a atenção do operador.

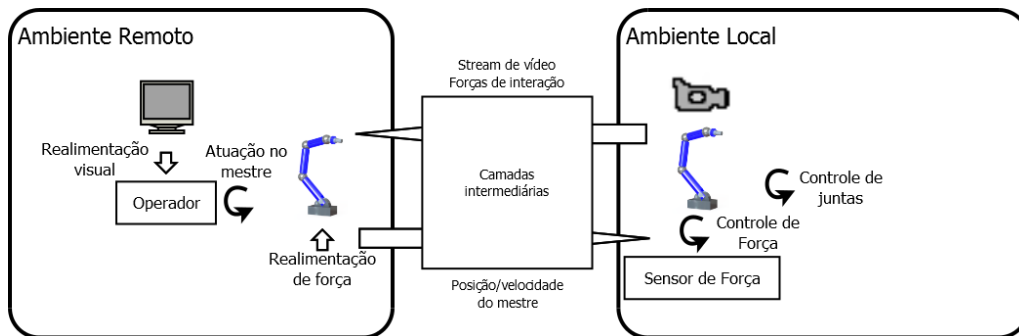


Figura 1.7: Diagramas de blocos para teleoperação com controle compartilhado.

No controle direto o operador realiza a tarefa diretamente, ocorrendo no ambiente local somente a malha de controle de juntas e atuadores dos dispositivos robóticos, ver Figura 1.8. Esta modalidade é dividida em duas categorias de acordo com a presença de realimentação de força. Quando esta existe, operação bilateral, o operador tem a percepção das forças de interação do manipulador com o ambiente. No caso contrário, operação unilateral, somente a realimentação visual é utilizada como guia para a realização da tarefa (Niemeyer et al. 2008).

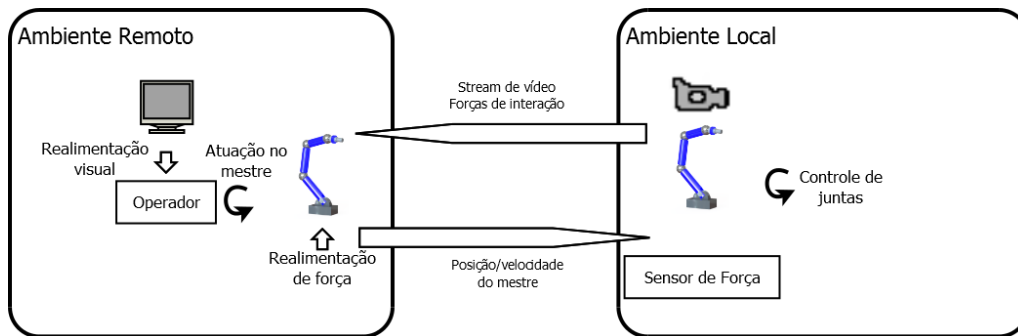
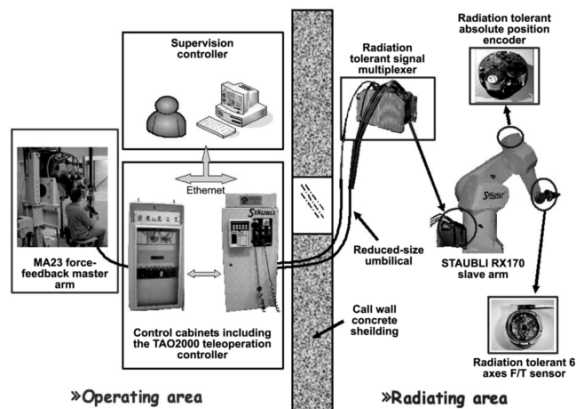


Figura 1.8: Diagramas de blocos para teleoperação com controle direto.

Um exemplo de teleoperação unilateral são as operações de ROV (veículos remotamente operados) submarinos utilizados para inspeções em usinas hidrelétricas, tubulações de plataformas de petróleo, operações de bancos de válvulas (árvore de natal) submersos, entre outros. Estes dispositivos são operados de um navio através de um cabo umbilical, sendo comandados por velocidade através de um *joystick*, como na ilustração da Figura 1.9(a). A modalidade bilateral, por exemplo, é utilizada em usinas nucleares para a manutenção de áreas com presença de radiação, mantendo o operador seguro em uma sala com uma visão direta do manipulador (Desbats et al. 2006), utilizando uma estrutura como a ilustrada na Figura 1.9(b).



(a) Operação de ROV.



(b) Plantas de energia nuclear. Adaptado de (Desbats et al. 2006)

Figura 1.9: Diagramas com representações de teleoperações na modalidade direta.

Neste trabalho, utiliza-se teleoperação com controle compartilhado. No ambiente remoto, parte da realimentação do operador é calculada diretamente sem informações sobre os escravos. Enquanto no ambiente local, parte do controle de cooperação é realizada sem a ação do operador.

Estabilidade, passividade e transparência

Para analisar a estabilidade dos sistemas de controle em teleoperação, considere a interligação de diferentes subsistemas de duas portas, como na Figura 1.10. Seguindo um fluxo da esquerda para direita, inicia-se no ambiente remoto com o operador interagindo com o dispositivo mestre ($\dot{x}_h \rightarrow \dot{x}_m$). Transmite-se um comando de velocidade ($\dot{x}_m \rightarrow \dot{x}_s$) para o ambiente local, este é passado para o controlador e o robô escravo, agindo sobre o ambiente ($\dot{x}_s \rightarrow \dot{x}_e$). A força resultante desta operação ($f_e \rightarrow f_s$) é enviada como realimentação para o operador ($f_s \rightarrow f_m \rightarrow f_h$). O canal de comunicação pode conter atraso variável e distinto em cada sentido.

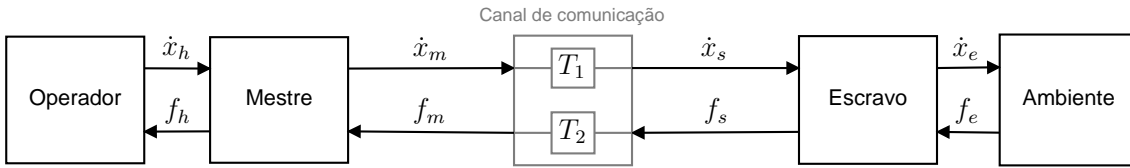


Figura 1.10: Representação de teleoperação como um conjunto de sistemas de 4 portas interligados. Entre os sistemas são transmitidas forças (f) e velocidades (\dot{x}), considerando a possibilidade de atraso no canal de comunicação. Os subíndices representam: h - operador; m - mestre; s - escravo; e - ambiente.

Através da teoria de passividade, temos que um sistema com entrada u e saída y é passivo, e por tanto estável, se a potência (P) que entra neste é tal que

$$P = u^T y = \frac{dH}{dt} + P_{diss},$$

ou seja, o sistema armazena ($\frac{dH}{dt}$) ou dissipa (P_{diss}) a potência (Khalil 2002). Uma condição que, geralmente, é mais simples de verificar é observar a energia ($H(t)$) que o sistema pode fornecer. Tem-se a cada instante

$$\int_0^t P(\tau) d\tau = H(t) - H(0) + \int_0^t P_{diss}(\tau) d\tau \geq -H(0),$$

que é limitada à energia inicialmente disponível. Desta condição, é simples concluir que conectando dois sistemas passivos o resultante mantém a mesma propriedade, pois em cada etapa a energia é conservada ou dissipada (Khalil 2002), inclusive em sistemas realimentados.

Na teleoperação, é comum supor que o operador e o ambiente se comportam como sistemas passivos, desde que apresentem complacência ao movimento. Os controladores para o mestre e o escravo serão projetados de forma estável, assim o ponto a ser analisado é o canal de comunicação. A energia neste é obtida da

diferença entre a potência na entrada (P_{in}) e na saída (P_{out}),

$$\int_0^t P = \int_0^t P_{in} - P_{out} d\tau = \int_0^t \dot{x}_m^T f_m - \dot{x}_s^T f_s d\tau. \quad (1.1)$$

Se o canal de comunicação é ideal, sem atrasos e sem perda de pacotes, então tem-se $\dot{x}_m = \dot{x}_s$ e $f_m = f_s$ e o sistema é estável. Como o aumento da distância entre os ambientes local e remoto leva a uma inevitável presença de atraso nesta comunicação, este é o principal ponto de investigação teórica no assunto.

A primeira solução apresentada utilizando a teoria de controle considera apenas a presença de atrasos constantes e baseia-se em passividade. Como o sistema pode ser tratado como uma rede de subsistemas de duas portas, Anderson & Spong (1989) aplicaram a teoria de dispersão (*Scattering*) sugerindo uma transformação de variáveis de estados com a qual a linha de transmissão se comporta como um sistema passivo. A nova topologia é apresentada na Figura 1.11, nota-se que são transmitidos sinais transformados que representam uma combinação da força e velocidade em cada ambiente.

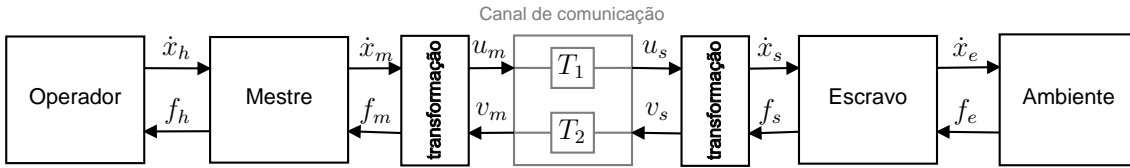


Figura 1.11: Representação de teleoperação utilizando a teoria de espalhamento como um conjunto de sistemas de 4 portas interligados. Entre os sistemas são transmitidas combinações de força (f) e velocidade (\dot{x}) através de um canal de comunicação com atraso constante. Os subíndices representam: h - operador; m - mestre; s - escravo; e - ambiente.

Com as mesmas condições no canal de comunicação e também considerando transformações de estado, Niemeyer & Slotine (1991) empregam a teoria de ondas, a qual permite observar novos aspectos deste canal, como a reflexão gerada por desbalanceamentos de impedância. As transformações propostas são semelhantes, porém a notação de (Niemeyer & Slotine 1991) é mais utilizada. Com esta, os sinais transmitidos são

$$u = \frac{b\dot{x} + F}{\sqrt{2b}} \quad e \quad v = \frac{b\dot{x} - F}{\sqrt{2b}}, \quad (1.2)$$

sendo parametrizados por b .

Utilizando a transformação (1.2) e considerando a energia inicial nula. A energia acumulada (1.1), é obtida de

$$\int_0^t P = \int_0^t \dot{x}_m^T f_m - \dot{x}_s^T f_s d\tau = \frac{1}{2} \int_0^t u_m^T u_m - v_m^T v_m + v_s^T v_s - u_s^T u_s d\tau.$$

Devido a presença de atraso tem-se que $u_s(t) = u_m(t - T_1)$ e $v_m(t) = v_s(t - T_2)$, assim a energia dada por

$$\begin{aligned}
& \frac{1}{2} \int_0^t u_m^\top u_m - v_s(\tau - T_2)^\top v_s(\tau - T_2) + v_s^\top v_s - u_m(\tau - T_1)^\top u_m(\tau - T_1) d\tau \\
&= \frac{1}{2} \int_0^t u_m(\tau)^\top u_m(\tau) d\tau - \frac{1}{2} \int_0^{t-T_1} u_m(\theta_1)^\top u_m(\theta_1) d\theta_1 + \\
&\quad + \frac{1}{2} \int_0^t v_s(\tau)^\top v_s(\tau) d\tau - \frac{1}{2} \int_0^{t-T_2} v_s(\theta_2)^\top v_s(\theta_2) d\theta_2 \\
&= \frac{1}{2} \int_{t-T_1}^t u_m(\tau)^\top u_m(\tau) d\tau + \frac{1}{2} \int_{t-T_2}^t v_s(\tau)^\top v_s(\tau) d\tau \tag{1.3}
\end{aligned}$$

é positiva, e portanto estável, independentemente da magnitude dos atrasos (Anderson & Spong 1989, Niemeyer & Slotine 1991). Destaca-se que o cálculo acima considerou a substituição de variáveis $\theta_i = \tau - T_i$, $d\theta_i = d\tau$ ($i = 1, 2$).

A partir da década de 90, a estrutura da internet se mostrou como a principal candidata a linha de transmissão para a teleoperação. Entretanto, devido as características desta rede, o atraso encontrado não é constante ($T_i = T_i(t)$, $i = 1, 2$) e (1.3) não é mais válida. A transformação de variáveis utilizada leva a $d\theta_i = (1 - \dot{T}_i)d\tau$ e, assim, a energia no canal de comunicação passa a ser dada por

$$\begin{aligned}
& \frac{1}{2} \int_0^t u_m(\tau)^\top u_m(\tau) d\tau - \frac{1}{2} \int_0^{t-T_1} \left(\frac{u_m(\theta_1)^\top u_m(\theta_1)}{1 - \dot{T}_1} \right) d\theta_1 + \\
&\quad + \frac{1}{2} \int_0^t v_s(\tau)^\top v_s(\tau) d\tau - \frac{1}{2} \int_0^{t-T_2} \left(\frac{v_s(\theta_2)^\top v_s(\theta_2)}{1 - \dot{T}_2} \right) d\theta_2 \\
&= \frac{1}{2} \int_{t-T_1}^t u_m(\tau)^\top u_m(\tau) d\tau - \frac{1}{2} \int_0^{t-T_1} \left(\frac{\dot{T}_1}{1 - \dot{T}_1} \right) u_m(\theta_1)^\top u_m(\theta_1) d\theta_1 + \\
&\quad + \frac{1}{2} \int_{t-T_2}^t v_s(\tau)^\top v_s(\tau) d\tau - \frac{1}{2} \int_0^{t-T_2} \left(\frac{\dot{T}_2}{1 - \dot{T}_2} \right) v_s(\theta_2)^\top v_s(\theta_2) d\theta_2. \tag{1.4}
\end{aligned}$$

Neste caso o sistema é estável somente se a variação do atraso for negativa, podendo se tornar instável se o atraso aumentar com o tempo, ou seja, $\dot{T}_i > 0$. Para comportar esta variação positiva do atraso Lozano et al. (2002) propõe uma modificação na transformação de estados (1.2) utilizando um ganho proporcional a variação de T_i ($k_i(T_i)$). Com estes ganhos os termos negativos em (1.4) se tornam

$$\frac{1}{2} \int_0^{t-T_i} \left(\frac{k_i^2 - 1 + \dot{T}_i}{1 - \dot{T}_i} \right) v_s(\theta_i)^\top v_s(\theta_i) d\theta_i,$$

levando à condição de estabilidade $k_i^2 \leq 1 - \dot{T}_i$.

Assim, é necessário conhecer-se somente o limite superior para variação do atraso. Entretanto a escolha deste ganho interfere também na performance do sistema, de modo que esta não deve ser muito conservadora, com valores pequenos para

comportar condições adversas no canal de comunicação.

Recentemente, uma transformação de variáveis alternativa às teorias de *scattering* e ondas foi proposta em (Chopra & Spong 2005). Esta consiste em uma combinação da velocidade e posição das juntas do dispositivo,

$$r_s = \dot{q}_s + \lambda q_s \quad \text{e} \quad r_m = \dot{q}_m + \lambda q_m ,$$

como no controle adaptativo de Slotine & Li (1991). Utilizando então uma lei proporcional ao erro desta nova variável, é possível demonstrar que, com atraso constante, os robôs em ambos ambientes convergem para a mesma posição e velocidade.

Ainda baseando-se na passividade do sistema Ryu et al. (2004), para atraso constante, e Ryu & Preusche (2007), para atraso variável, propõem um método no domínio do tempo em que não se utiliza uma transformação de estados. Ao invés de passivar o canal de comunicação, a energia gerada neste devido ao atraso é dissipada por elementos ativos antes dos controladores local e remoto. A nova configuração do sistema de teleoperação é apresentada na Figura 1.12.

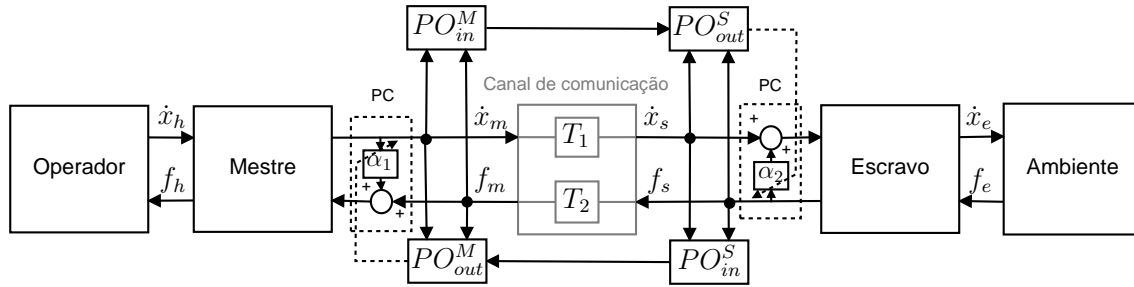


Figura 1.12: Diagrama de blocos do controle de teleoperação com passividade no domínio do tempo. PO: estimador de energia; PC: controlador de energia.

Outro método baseado em passividade que não se utiliza de transformações de estado foi proposto em (Lee & Spong 2006). Este considera atraso constante no sistema e uma lei de controle proporcional e derivativa do erro de posição, mais um termo de amortecimento. Assim, a lei de controle completa, para o mestre, é

$$f_m = -K_d \dot{x}_m(t) - K_p(x_m(t) - x_s(t - T_2)) - K_v(\dot{x}_m(t) - \dot{x}_s(t - T_2)),$$

e uma expressão semelhante é utilizada para o robô escravo. Os ganhos K_p e K_d são matrizes positivas definidas e a estabilidade é obtida pela dissipação resultante do termo de amortecimento, dependendo assim da escolha correta do ganho K_d .

Para determinar este ganho, Lee & Spong (2006) impõe a condição no domínio da frequência

$$K_d \geq \left(\frac{\sin \frac{w(T_1+T_2)}{2}}{w} \right)^2 K_p K_d^{-1} K_p, \quad \forall w \in \mathbb{R}.$$

Uma possível solução para esta condição é $K_d = \frac{(\bar{T}_1 + \bar{T}_2)}{2} K_p$, que depende somente do limite superior dos atrasos (\bar{T}_i). Esta lei de controle garante uma transparência ideal durante o regime permanente.

Uma comparação entre diferentes estratégias baseadas em passividade para estabilizar a teleoperação é realizada através de uma estrutura teórica unificada, baseada em funções de Lyapunov, em (Nuño et al. 2011). Um trabalho menos atual, de Arcara & Melchiorri (2002), realiza uma comparação mais ampla, incluindo estratégias que não se baseiam em passividade.

Dentre as técnicas que não se baseiam em passividade se destacam aquelas que utilizam controle ótimo, estimadores de estados e técnicas de controle avançado. As técnicas de controle ótimo utilizam informações de limite superior dos atrasos no projeto dos controladores, obtendo leis estáveis até estes limites (Leung et al. 1995). A síntese é realizada em duas etapas, na primeira um controlador para movimentação livre é projetado via \mathcal{H}_∞ , em seguida utiliza-se μ -synthesis para um controlador que rejeite os atrasos como perturbações durante o movimento restrito pela interação com o ambiente, considerado uma impedância conhecida. Para aumentar o intervalo de aplicação das leis projetadas e gerar resultados menos conservadores, Boukhnifer et al. (2004) empregam um parâmetro de escalonamento para alterar os ganhos de acordo com o nível de atraso.

O uso de estimadores de estados visa, no ambiente remoto, estimar o ambiente local de forma a obter uma informação de contato, e portanto força, livre de atraso. A solução mais empregada é a utilização do preditor de Smith, o qual é utilizado para controle de diversas plantas com atraso e utiliza um modelo do sistema em uma malha de realimentação interna. Na teleoperação, este método foi classicamente empregado com modelos a parâmetros fixos e aproximações lineares. Mais recentemente, técnicas não lineares passaram a ser empregadas para aumentar a robustez desta estimativa. Como exemplo, Smith & Hashtrudi-Zaad (2006) utilizam redes neurais para adaptar de forma contínua o modelo do escravo e seu ambiente.

Recentemente, Denasi et al. (2013) substituem a arquitetura clássica do preditor de Smith por uma estratégia que combina o princípio do modelo interno e o controle com modelo interno (na sigla em inglês, IMPACT). Ao agregar o princípio do modelo interno, a IMPACT (Tsytkin & Holmberg 1995) aumenta, em relação a técnica de Smith, a robustez e a rejeição de perturbações. A eficácia do método é demonstrada em simulação e testes experimentais para teleoperação baseada em posição.

No campo de controle avançado, Sheng & Spong (2004) propõem a utilização de modelos dos ambientes local e remoto em uma versão modificada do controle preditivo (MPC). Com isso, obtêm uma teleoperação com estabilidade implícita, ajustando o horizonte de predição de acordo com o atraso no canal de comunicação e impondo as condições necessárias como restrições de otimização. A modificação

adotada utiliza um termo de erro entre o valor predito e medido, de forma a aumentar a robustez para parâmetros desconhecidos de atraso.

Além da estabilidade, um outro aspecto importante para a teleoperação é a transparência. Esta é a capacidade do sistema de controle reproduzir para o operador a interação entre o manipulador escravo e o ambiente, idealmente causando a sensação de atuação direta. Os dois principais trabalhos neste assunto são os de Lawrence (1993), Yokokohji & Yoshikawa (1994) que estabeleceram que a transparência total exige o cancelamento das impedâncias do mestre e escravo, necessitando a transmissão de informações por quatro canais. Além disso, estabelecem que transparência e estabilidade são conflitantes, exigindo sempre o balanço entre as duas condições.

Uma medida para a transparência do sistema é estabelecida pela chamada matriz híbrida

$$\begin{bmatrix} f_m(s) \\ \dot{x}_m(s) \end{bmatrix} = \begin{bmatrix} h_{11}(s) & h_{12}(s) \\ h_{21}(s) & h_{22}(s) \end{bmatrix} \begin{bmatrix} \dot{x}_s(s) \\ -f_s(s) \end{bmatrix}$$

que relaciona, no domínio da frequência, força e velocidade no mestre e no escravo (Lawrence 1993). A transparência ideal é então representada por $h_{11} = h_{22} = 0$, $h_{12} = -1$ e $h_{21} = 1$, na qual a inércia do sistema não é transmitida entre os ambientes.

Recentemente, uma nova abordagem para o projeto de controle na teleoperação foi proposta em (Franken et al. 2009). Esta considera o projeto em duas camadas, uma responsável pela transparência e outra pela estabilidade. A Figura 1.13 mostra o diagrama de blocos representando esta estratégia.

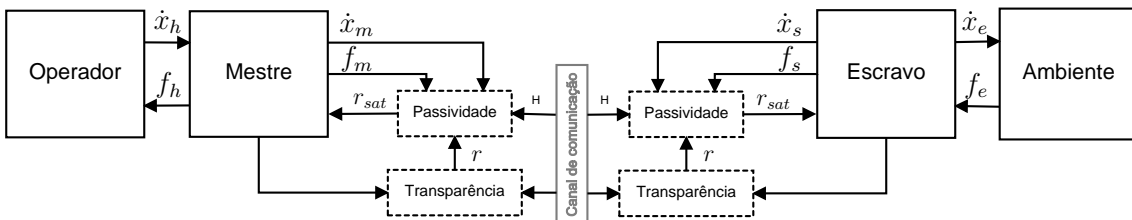


Figura 1.13: Diagrama de blocos da estratégia de teleoperação em duas camadas. Na camada de passividade são transmitidos pacotes de energia entre os ambientes, na de transparência transmiti-se os sinais utilizados para controle. O sinal r representa a referência ideal enviada pelo controle de teleoperação para o controlador do robô, entretanto esta é alterada para manter a passividade gerando r_{sat} .

Na camada de transparência são implementadas as leis de controle que para que o manipulador escravo siga os movimentos do mestre e este reproduza para o operador as forças de interação no ambiente local. Essas leis podem ser projetadas sem considerar a estabilidade total do sistema, podendo inclusive ser instáveis (Franken et al. 2009). Antes de enviar as referências para o controlador dos robôs essas passam através da camada de estabilidade. Nessa se mantém um tanque de energia

que monitora os movimentos dos manipuladores e sua interação com o ambiente e o operador, assegurando que o mesmo se comporte de forma passiva. Para tal, anula-se a referência sempre que não há mais energia disponível podendo-se também determinar outras formas de saturação de acordo com a aplicação (Franken et al. 2009). Para manter a energia fluindo no sistema e possibilitar que o mesmo se recupere após possíveis movimentos instáveis, esta camada também mantém um canal de comunicação por onde são enviados pacotes de energia.

Apesar de também se basear no monitoramento e dissipação da energia nos ambientes local e remoto, a estratégia de Franken et al. (2009) é diferente daquela de Ryu & Preusche (2007). Isto porque aquela não limita o tipo de controlador bilateral a ser utilizado, permite que se atua no sistema antes da perda de passividade para evitar-la e comporta diretamente camada intermediária para melhoria da performance do operador, como em (Cho et al. 2012).

Arquiteturas para teleoperação

Em Taylor et al. (1999) e Bambang (2007) apresentam-se interfaces para operação de manipuladores através da internet limitadas a operação ponto a ponto, onde o operador fornece uma posição e deve esperar o fim do movimento para indicar um novo alvo. As posições podem ser definidas no espaço das juntas por meio de um formulário (Bambang 2007) ou no espaço cartesiano também através de um formulário, clicando em uma imagem ou em um modelo virtual (Taylor et al. 1999). Já na interface de Safaric et al. (2003) a tarefa pode ser completamente planejada em um modelo virtual do manipulador, sendo posteriormente enviada e executada no robô real. Em todos os trabalhos a interface gráfica é desenvolvida para navegadores utilizando CGI, HTML e Java.

A SMART, arquitetura modular de controle apresentada por Anderson (1995), é implementada em uma rede VME (*Virtual Module Europe*) de sensores, atuadores e computadores utilizando sistema operacional VxWorks. A interface de comando está em uma máquina que se conecta à rede através do padrão Ethernet. É possível criar malhas para teleoperação autônoma ou bilateral através da interligação de blocos pré-programados. Após conectar todos os blocos, são gerados códigos em linguagem C, que podem ser modificados para atender especificações da tarefa e então compilados e enviados para os respectivos computadores ou dispositivos na rede.

Uma arquitetura que combina operação bilateral e servovisão é apresentada por Goto et al. (2010). O mestre e o escravo são controlados por dois computadores rodando Linux de tempo real. A comunicação entre o ambiente remoto e o local, das posições dos manipuladores, é realizada por protocolo TCP (do inglês, *Transmission Control Protocol*), com previsão para implementação em internet, mas

testado somente em rede local. A servovisão é responsável pelo posicionamento fino utilizando uma câmera no punho que é alinhada a um alvo de forma automática após este entrar no quadro da imagem. O processamento é local em um terceiro computador se comunicando via TCP com o controlador do manipulador escravo.

Realizando somente o controle bilateral Oboe & Fiorini (1998) utilizam um sistema em tempo real criado pelos autores que se comunica com a internet através de uma *socket*. Munir & Book (2003) utilizam um processo em tempo real no Windows NT com o auxílio do Hyperkernel, um programa que cria um *kernel* paralelo ao do sistema operacional, realizando a comunicação com a internet através do NT que acessa uma memória compartilhada com o Hyperkernel. Em ambos trabalhos, um computador remoto retransmite o sinal de controle do manipulador mestre, através da internet, para uma simulação do mesmo utilizada como escravo, que realimenta o sistema de controle diretamente.

Hirche et al. (2004) controlam um manipulador de 7 graus de mobilidade em um espaço operacional de 2 graus e Stanczyk & Buss (2004) estendem o controle através de um manipulador mestre com 6 graus de mobilidade. Estes dois trabalhos realizam o controle dos dispositivos robóticos utilizando a combinação Matlab/Simulink e *XPC Target*, porém não especificam como é realizada a comunicação UDP (do inglês, *User Datagram Protocol*) entre as máquinas.

É válido ressaltar que essas diferentes interfaces propostas não apresentam visualização *online* do ambiente de operação. As imagens são mostradas no início e fim do movimento (Taylor et al. 1999), um vídeo da tarefa é mostrado após a sua execução (Safaric et al. 2003), apenas um modelo virtual é apresentado ao operador (Bambang 2007) ou o mestre e o escravo estão no mesmo ambiente (Goto et al. 2010, Munir & Book 2003, Oboe & Fiorini 1998).

1.2.2 Manipuladores Cooperativos

A utilização de múltiplos manipuladores operando em cooperação foi proposta para superar desafios encontrados pela estratégia clássica de um único manipulador. A coordenação entre mais de um robô permite a manipulação de objetos de maiores dimensões e peso, objetos flexíveis ou a montagem de diferentes partes simultaneamente.

Um desafio nesta área é que a cooperação impõe restrições ao movimento, alterando a cinemática e dinâmica dos manipuladores quando considerados individualmente. Estas restrições decorrem do ambiente variável, já que o movimento de um manipulador é refletido no efetuador dos demais, e da necessidade de manter o objeto seguro ou uma determinada posição relativa ao demais robôs.

Desta forma, uma solução natural para este problema são as estratégias de mestre-escravo e líder-seguidor. Na primeira, o manipulador mestre é controlado por posição de forma robusta a perturbações externas, como os outros manipuladores, enquanto os escravos são controlados por força para obter um comportamento complacente, com seus movimentos impostos pela interação com o mestre (Caccavale & Uchiyama 2008). Já na segunda estratégia, o líder é semelhante ao mestre, mas os seguidores tem seus movimentos determinados considerando o sistema como uma cadeia fechada (Luh & Zheng 1987).

Entretanto, estas estratégias possuem dificuldades de implementação, pois é difícil obter a complacência necessária para os manipuladores que não são mestre ou líder. Além disso, de acordo com a tarefa, pode ser necessário alterar os papéis dos robôs (Caccavale & Uchiyama 2008). Para contornar estas limitações, as propostas mais atuais buscam calcular a trajetória individual de cada manipulador de forma a alcançar o objetivo em conjunto.

Dentro destas novas estratégias, destacam-se aquelas baseadas em controle híbrido e a linearização por realimentação. A primeira adota uma proposta de controle para a interação entre robôs e ambientes rígidos. Esta consiste em separar o controle de posição e de força em graus de liberdade distintos, através de uma matriz seletora (Uchiyama & Dauchez 1992). Assim, cada manipulador segue uma referência de posição e força adequada para a tarefa.

A estratégia de de linearização por realimentação é semelhante ao caso com um único manipulador, onde o modelo dinâmico do sistema é computado de forma a compensar a não linearidade existente no mesmo. No caso cooperativo, entretanto, o modelo utilizado é do sistema aumentado, onde o sistema completo, manipuladores e objeto, é considerado. A lei de controle é então projetada com um termo de posição e outro de força, computando o torque (τ) nas juntas através de (Hsu 1993):

$$\tau = J^T W^\dagger \left[M_o \left(\ddot{x}_e^d + K_v \left(\dot{x}_E^d - \dot{x}_E \right) + K_p \left(x_E^d - x_E \right) + c_o \right) \right] + J^T V \left(h_I^d + K \int h_I^d - h_I \right).$$

O modelo do sistema é representado pela matriz M de inércia, o vetor c de forças de Coriolis, centrípetas e gravitacionais, e a matriz Jacobiano J utilizada para transformação do espaço Cartesiano para o espaço das juntas. As matrizes K são ganhos diagonais positivas, x_E é a posição do objeto e h_I são as forças internas, com o sobrescrito d representando a referência.

As forças internas são aquelas que não contribuem para o movimento do objeto, somente para segura-lo. Matematicamente, são forças projetadas no espaço nulo da matriz W , que transforma as forças e momentos no efetuador dos manipuladores para o centro de massa do objeto (Caccavale & Uchiyama 2008). Na equação acima, esta projeção no espaço nulo de W é representada pela matriz V . O uso das forças internas permite, então, desacoplar os controles de movimento e força. Entretanto,

necessita de informações da distribuição de carga do objeto e da localização dos pontos de contato em relação ao centro de massa.

Teleoperação

A teleoperação de manipuladores cooperativos é um problema ainda pouco abordado na literatura. Destacam-se três abordagens considerando um ou múltiplos mestres (Aldana et al. 2012, Bacocco & Melchiorri 2009, Lee & Spong 2005, Sirouspour 2005).

Em uma estratégia SMMS (do inglês, *Single Master Multiple Slaves*), Lee & Spong (2005) propõe um único mestre comandando a posição de um objeto firmemente seguro por múltiplos manipuladores. A estabilidade da teleoperação é garantida utilizando-se uma transformação de estados, através da teoria de *scattering*. Já a coordenação entre os robôs escravos é realizada através de decomposição passiva, onde o sistema formado por todas as juntas é desacoplado em um grupo responsável pela pegada do objeto e outro pelo movimento restrito devido a essa pegada, sem que se altere a passividade. Posteriormente, Aldana et al. (2012), considerando que não existe medição de força, utilizam a mesma lei de controle para o manipulador mestre, alterando o controle dos escravos para um controle adaptativo no espaço Cartesiano. Esta lei de controle considera tanto o modelo dos manipuladores quanto o do objeto, adaptando ambos. Com a atraso constante e sem interação entre a carga e o ambiente, esta e o manipulador mestre convergem para a mesma posição.

Com um conceito de múltiplos mestres, MMMS (do inglês, *Multiple Masters Multiple Slaves*), Sirouspour (2005) propõe um controlador em que as informações de posição e força são compartilhadas por todos os agentes. Entretanto, é mantido um pareamento entre mestres e escravos. No ambiente local utiliza-se um controlador ótimo baseado no método μ -*synthesis* que estabelece estabilidade robusta para a teleoperação. Utilizando diferentes funções objetivo é possível também otimizar a coordenação entre os agentes.

Com outra abordagem para o caso MMMS, Bacocco & Melchiorri (2009) consideram o uso de múltiplos pares de robôs locais e remotos teleoperados individualmente compartilhando a mesma ferramenta ou atuando sobre o mesmo objeto. Cada um dos pares utiliza uma estratégia baseada em variáveis de onda como em (Niemeyer & Slotine 1991), sem que ocorra uma ação explícita de coordenação entre os escravos. Os robôs são modelados como sistemas massa-amortecedor de um grau de liberdade, sendo controlados por posição através de um controle proporcional e derivativo ajustado por síntese linear quadrática. A validação da estratégia é obtida por simulação dos escravos compartilhando uma ferramenta ou transportando um objeto.

1.2.3 Transporte de Cargas

O problema de transporte de carga por dispositivos robóticos é típico da robótica móvel e se assemelha ao problema de cooperação. O objetivo é carregar um objeto que é maior que cada um dos robôs ou supera o limite individual de carga destes.

Arquitetura

Existem diferentes arquiteturas para o sistema de controle e comunicação entre os robôs no transporte de carga. Estas podem ser divididas entre centralizadas e descentralizadas.

Em um sistema de controle centralizado todos os agentes obedecem aos comandos de uma única malha de controle, como na Figura 1.14(a). Esta malha central detêm o conhecimento total do sistema, com as leituras de todos os sensores e a posição de todos os agentes, podendo assim determinar os movimentos individuais que garantem o objetivo coletivo.

No caso do controle descentralizado cada agente possui sua própria malha de controle. A coordenação pode ser realizada através de informações passadas diretamente entre os agentes, exigindo um canal de comunicação entre estes (ver Figura 1.14(b)), ou indiretamente, sendo obtida da própria tarefa (ver Figura 1.14(c)).

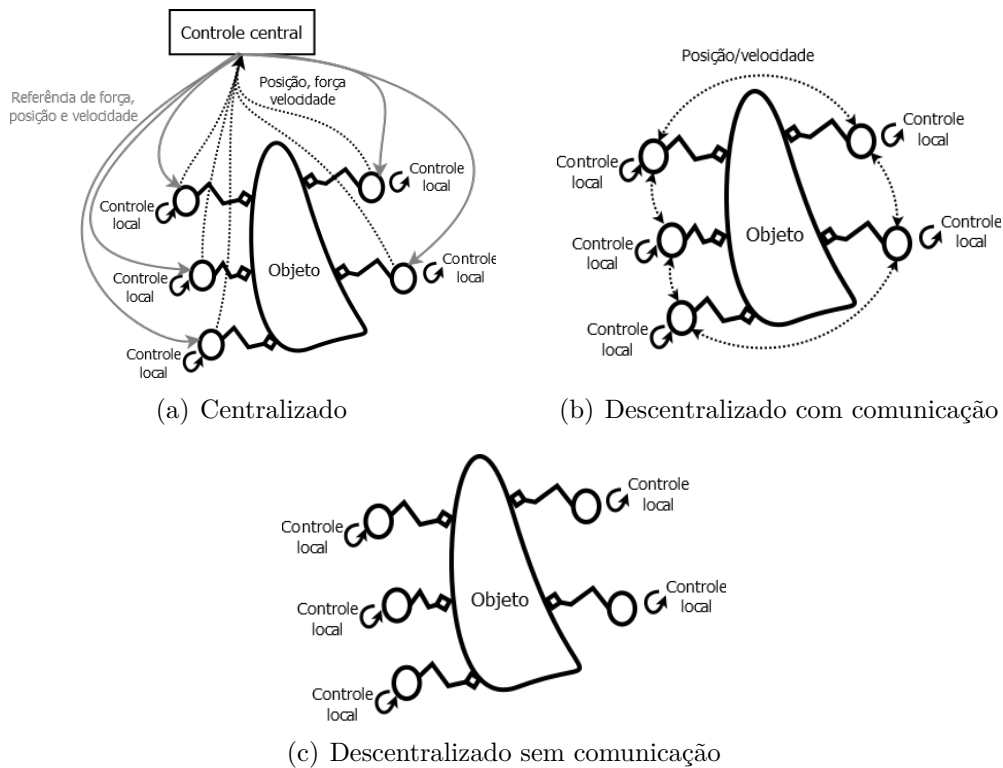


Figura 1.14: Diagrama representando os sistemas de controle centralizado (a) e descentralizado com comunicação direta (b) e indireta (c).

Pegada

Ao transportar a carga é necessário garantir que a mesma permaneça segura e contida pelos robôs, essa condição é denominada de pegada (no inglês, *grasp*). O desenvolvimento teórico sobre esta condição é decorrente do estudo de mãos robóticas. Neste se estabelecem duas modalidades de pegada: os fechamentos de força e de forma (Prattichizzo & Trinkle 2008).

No fechamento de força, os robôs atuam sobre a carga de forma que o esforço exercido mantenha um determinado número de graus de liberdade da mesma fixos. Um exemplo de fechamento de força é mostrado na Figura 1.15(a). Já no fechamento de forma os movimentos da carga são restritos pela configuração dos robôs durante a pegada, como no exemplo da Figura 1.15(b).

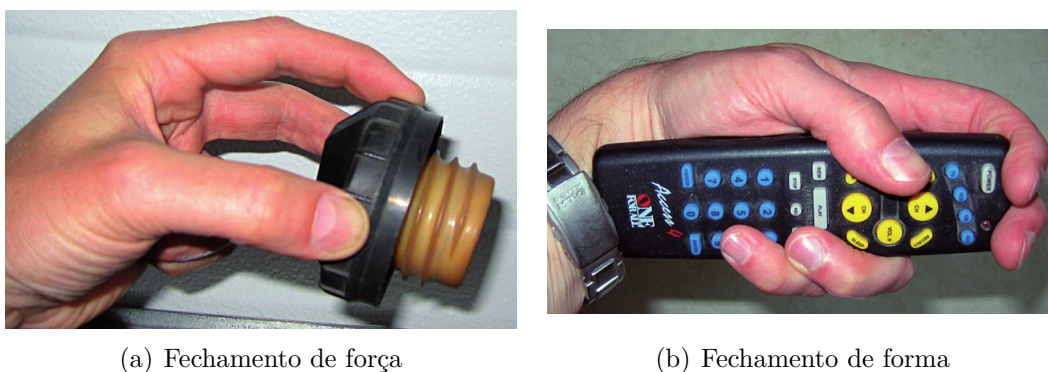


Figura 1.15: Exemplos de condições de pegada. Retirado de (Prattichizzo & Trinkle 2008).

Para o caso de robôs móveis transportando uma carga, pode-se acrescentar uma terceira condição denominada fechamento de objeto. Neste caso o objeto fica restrito a uma determinada área formada pelos robôs como se estivesse em uma gaiola (no inglês, *cage*), podendo, desta forma, se movimentar livremente neste espaço e perder o contato temporariamente com os agentes (Pereira et al. 2004). Para ilustrar este conceito, considere um grupo de robôs formando uma circunferência com um objeto no centro. Se os agentes da circunferência começarem a se deslocar coordenadamente, o objeto do centro terá que seguir o mesmo caminho, já que será empurrado por um ou mais robôs. A Figura 1.16 representa este caso exemplo.

Controle

O problema de transportar uma carga, possivelmente flexível, utilizando um grupo de robôs móveis é tratado por Sugar & Kumar (2002a), são considerados diferentes leis de controle descentralizadas com comunicação entre os agentes. As soluções propostas mantêm autonomia dos robôs permitindo alterações na formação apresentada pelos mesmos, utilizando uma estratégia de líder-seguidor com os papéis



Figura 1.16: Exemplo de transporte de carga utilizando enjaulamento. O objeto pode se mover dentro da circunferência de robôs, mas não pode sair deste espaço².

podendo ser alterados durante a trajetória. Para resolver as dificuldades decorrentes da necessidade de controle de forças e momentos sobre a carga, e de erros pelo uso de veículos não-holonômicos, um novo manipulador paralelo com efetuator ativo, proposto em (Sugar & Kumar 2002b), é utilizado em um dos robôs.

Uma outra estratégia de controle descentralizada, agora sem comunicação explícita, é proposta em (Kume et al. 2002). Os pontos de contato são modelados com uma dinâmica semelhante a rodas castor em 3D. O conjunto é controlado por uma estratégia líder-seguidor, proposta em (Kosuge & Oosumi 1996), em que os seguidores estimam a referência através dos estímulos recebidos via carga, permitindo o transporte da carga sem que os robôs conheçam as relações geométricas com os demais. Os agentes são dispositivos móveis com um manipulador acoplado. O manipulador é responsável pela pegada, e os graus de liberdade excedentes são utilizados para movimentação.

Utilizando controle descentralizado e uma condição de enjaulamento Pereira et al. (2004) transportam um objeto em uma trajetória planar utilizando um conjunto de robôs móveis. São apresentados resultados teóricos com uma condição para garantir o fechamento de objeto e um controle em três etapas: criar a pegada, manter-la e se deslocar para o objetivo. Cada um dos agentes transita independentemente por estas etapas, de forma que o conjunto atinge o objetivo de carregar o objeto. Os resultados são exemplificados em um experimento onde os robôs são controlados utilizando realimentação visual.

Sem utilizar uma estratégia de líder-seguidor, Montemayor & Wen (2005) propõe um controle descentralizado com fechamento de força. A estratégia utilizada realiza o controle simultâneo de deslocamento e força em cada um dos agentes sem uma comunicação explícita de nenhum parâmetro. Primeiramente, considera-se, de forma independente, o controle de movimento, projetando-se, em seguida, o controle de força que considera como perturbações as forças geradas pelo outro controlador. Os controladores não utilizam explicitamente a projeção no espaço nulo para obter

²Esta imagem é um quadro do vídeo <http://www.youtube.com/watch?v=SBLIYAHdYLM>

a realimentação das forças internas e o desacoplamento dos controles. Entretanto, a estabilidade é garantida, pois o controle de movimento tem convergência exponencial. A limitação é que se consideram somente translações.

Em (Bai & Wen 2010) o problema de transporte de carga é abordado através do controle de formação. O objeto transportado é deformável, de modo que a força exercida por cada robô sobre o objeto é modelado como uma função da deformação provocada no mesmo. Com isso, controlar uma força desejada é equivalente a controlar a distância entre os efetuadores de dois robôs. Considerando que cada robô é um agente em uma formação em movimento, manter o objeto seguro é equivalente a manter esta formação. Este problema é solucionado de forma que os agentes, considerados uma massa pontual, e a carga adquirem uma mesma velocidade constante, utilizando controladores descentralizado com comunicação através da força exercida na carga (Bai & Wen 2010).

Diferente dos demais trabalhos, Hacıoglu et al. (2011) propõem uma lei de controle projetada diretamente para manipuladores robóticos. É considerado o caso de dois robôs planares movimentando uma carga, controlados através de modos deslizantes e uma lógica *fuzzy* multivariável. O sistema é robusto a alteração da carga e a ruídos nos sinais de medição.

1.3 Objetivos

O objetivo principal deste trabalho é desenvolver uma metodologia para a teleoperação bilateral de uma tarefa de transporte de cargas de maneira estável considerando múltiplos mestres e múltiplos escravos em um canal de comunicação com atrasos variáveis. Como resultados parciais destacam-se: i. propor uma arquitetura de teleoperação flexível com prototipagem rápida. ii. integrar o dispositivo Phantom Omni ao Matlab. iii. estabelecer a estrutura para controlar o robô Motoman DIA10 através de um computador.

1.4 Metodologia

O problema de cooperação será resolvido utilizando-se rastreamento de velocidade associado a regulação de força considerando as seguintes hipóteses:

1. o contato entre o efetuator dos manipuladores e a carga é pontual com atrito, transmitindo forças e não torques;
2. antes de iniciar a teleoperação os manipuladores já estão segurando o objeto;
3. as referências de forças de contato são projetadas de forma a garantir uma condição de fechamento de força;

4. os robôs mestres são manipuladores, ou dispositivos hápticos, que possuem um mecanismo para atuar com a força transmitida do ambiente local para realimentação do operador;
5. os movimentos de cada um dos manipuladores escravo não leva a configurações singulares.

O controle de pegada é realizado diretamente no ambiente local, não sendo afetado pelo atraso no canal de comunicação e não dependendo da ação do operador. Este controle é realizado em conjunto com o de velocidade para movimentar a carga, realizando a cooperação entre os manipuladores através da abordagem de formação proposta por Bai & Wen (2010). Para tal, esta é estendida para ao caso de movimento espacial e rastreamento de trajetória de velocidade, apresentando uma prova teórica da estabilidade da nova lei de controle proposta. Esta estratégia permite considerar os manipuladores em separado, sem calcular explicitamente restrições de cadeia fechada.

Para tornar a solução proposta mais geral, são desenvolvidas três leis de controle: cinemática, dinâmica sem controle de orientação e dinâmica com controle de orientação, considerando o modelo completo dos manipuladores. Assim podem ser utilizados robôs industriais e com dinâmica não desprezível, utilizando ou não sensor de força no efetuador.

Para garantir a estabilidade na teleoperação, utiliza-se a estratégia em duas camadas proposta em (Franken et al. 2009). Devido a estratégia de cooperação utilizada, o sinal de referência de velocidade é igual para todos os manipuladores. Assim, é possível utilizar um número de mestres e escravos distintos, possibilitando a realização de teleoperação SMMS (Aldana et al. 2012, Lee & Spong 2005) ou MMMS (Bacocco & Melchiorri 2009, Sirouspour 2005). Para manter esta possibilidade, a estratégia de realimentação de força é projetada de forma a não parear diretamente um robô mestre a um escravo, utilizando um agrupamento pela direção da mesma. Diferentemente dos trabalhos anteriores de (Aldana et al. 2012, Lee & Spong 2005, Sirouspour 2005), transmite-se a força de contato para o operador, assim o caso SMMS é reduzido à teleoperação unilateral.

1.5 Estrutura do Texto

A sequência do trabalho irá apresentar as estratégias de teleoperação e controle cooperativos propostos, utilizando a seguinte estrutura.

Capítulo 2 Apresenta uma introdução para a modelagem de manipuladores robóticos, no nível cinemático e dinâmico. São apresentados os principais conceitos e equações necessários para o desenvolvimento dos capítulos seguintes.

Capítulo 3 Descrição da estrutura em duas camadas utilizada para a teleoperação. Cada uma das camadas é apresentada com suas particularidades para o ambiente remoto e o local, de forma independente à lei de controle aplicada para os manipuladores.

Capítulo 4 São propostas três leis de controle para os manipuladores escravos, uma cinemática e duas dinâmicas. Desta forma a estratégia abordada neste trabalho pode ser aplicada a manipuladores industriais e aqueles com dinâmica não desprezível. Cada uma é projetada para rastrear uma referência de velocidade enquanto regula uma referência de força, seguindo uma abordagem de controle cooperativo descentralizado semelhante ao problema de formação.

Capítulo 5 A lei de controle cinemático e a estrutura de teleoperação são testadas através de simulação e procedimentos experimentais. As simulações contemplam a teleoperação bilateral, enquanto os experimentos abordam o controle autônomo e a teleoperação unilateral, utilizando como mestre um Phnatom Omni ou os movimentos do operador capturados através do Kinect.

Capítulo 6 No último capítulo são apresentadas as conclusões.

Além do corpo principal, informações complementares são apresentadas em três seções em anexo. No Apêndice A é apresentada a arquitetura para prototipagem rápida de aplicações de teleoperação desenvolvida durante este trabalho (Motta-Ribeiro et al. 2012), no Apêndice B são apresentados teoremas e provas auxiliares, e no Apêndice C encontra-se um manual para operação do robô Motoman DIA10 utilizado para os testes experimentais.

Capítulo 2

Modelagem de Manipuladores

Manipuladores robóticos são cadeias de corpos rígidos, denominados elos, conectados por mecanismos móveis, denominados juntas (ver Figura 2.1). A relação entre o estado de cada junta da cadeia com o estado de cada um dos elos pode ser modelada através de uma abordagem cinemática ou dinâmica. A primeira aborda posição e orientação dos corpos, enquanto a segunda considera as forças e momentos angulares.

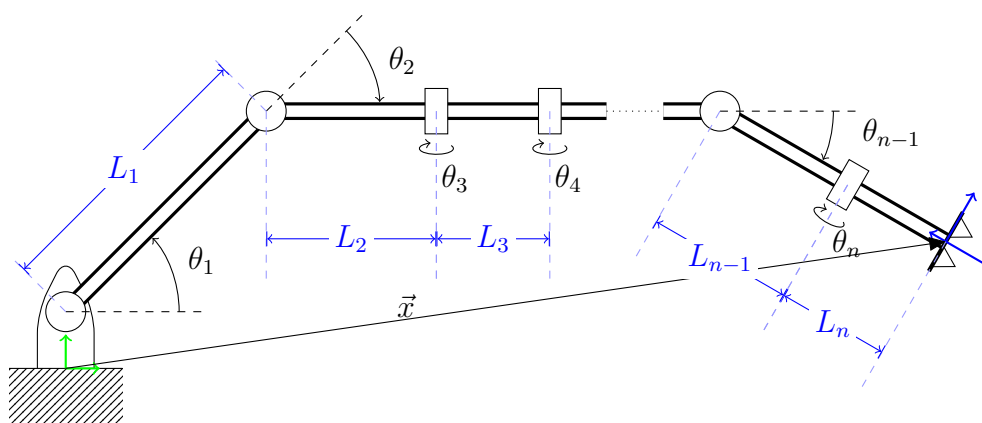


Figura 2.1: Representação esquemática de um manipulador robótico. Em vermelho estão indicados os movimentos das juntas e em azul o comprimento dos elos. Os sistemas de coordenadas representam a referência inercial na base (verde) e o efetuator (azul).

2.1 Modelo Cinemático

A cinemática estuda a descrição da posição e orientação, bem como suas derivadas temporais, de cada um dos corpos rígidos da cadeia que descreve o manipulador. Por simplicidade quando posição e orientação forem consideradas simultaneamente serão denominadas pose.

A pose é definida a partir de um sistema de coordenadas ortonormal $\bar{E} = [\vec{e}_1 \ \vec{e}_2 \ \vec{e}_3]$ que satisfaz as propriedades

- Normalidade: $\|\vec{e}_i\| = 1, \quad i = 1, 2, 3;$
- Ortogonalidade: $\vec{e}_i \cdot \vec{e}_j = 0, \quad i \neq j;$
- Produto Vetorial: $\vec{e}_3 = \vec{e}_1 \times \vec{e}_2.$

sendo assim os vetores unitários \vec{e}_i são uma base para um espaço vetorial normado tridimensional (\mathcal{V}). Considerando dois sistemas $\bar{E}_l = [\vec{i}_l \ \vec{j}_l \ \vec{k}_l]$ ($l = 0, 1$), como na Figura 2.2(a), a pose de E_1 em relação a E_0 é representada por uma matriz de transformação homogênea

$$T = \begin{bmatrix} R_{01} & \vec{x}_{01} \\ 0 & 1 \end{bmatrix}.$$

Esta matriz é formada pelo vetor de posição

$$\vec{x}_{01} = x_i \vec{i}_0 + x_j \vec{j}_0 + x_k \vec{k}_0$$

e pela matriz de rotação $R_{01} \in \mathbb{R}^{3 \times 3}$ de classe $SO(3)$ (Siciliano et al. 2009).

Nota-se que a matriz de rotação representa a orientação através de nove parâmetros, com seis restrições de ortogonalidade. Uma alternativa mais sintética é considerar o uso do quaternion unitário $q = (q_e, q_v)$, definido como

$$\begin{aligned} q_e &= \cos(\alpha/2) \in \mathbb{R} \\ q_v &= \sin(\alpha/2) \vec{v} \in \mathbb{R}^3. \end{aligned}$$

Esta representação utiliza o teorema de Euler¹ para tratar a orientação como equivalente a somente uma rotação por um ângulo α em torno de um vetor \vec{v} no sistema de coordenadas E_0 , como mostrado na Figura 2.2(b). A parcela q_e é denominada parte escalar, enquanto a parcela q_v parte vetorial, do quaternion. Ambas são submetidas a restrição $q_e^2 + q_v^T q_v = 1$ (Siciliano et al. 2009).

O quaternion pode ser relacionado diretamente à matriz de rotação através da chamada fórmula de Rodrigues (Lizarralde & Wen 1996)

$$R = I + 2q_e(q_v \times) + 2(q_v \times)(q_v \times),$$

na qual $I \in \mathbb{R}^{3 \times 3}$ é a matriz identidade e o operador \times representa a matriz produto vetorial. Já o mapeamento inverso, de $R \mapsto q$, acarreta em uma ambiguidade de sinal, pois q e $-q$ representam a mesma rotação. Uma possível solução, considerando

¹O teorema das rotações de Euler diz que qualquer sequência de rotações em torno de um ponto fixo é equivalente a uma única rotação por um dado ângulo α em torno de um eixo \vec{v} que passa pelo ponto.

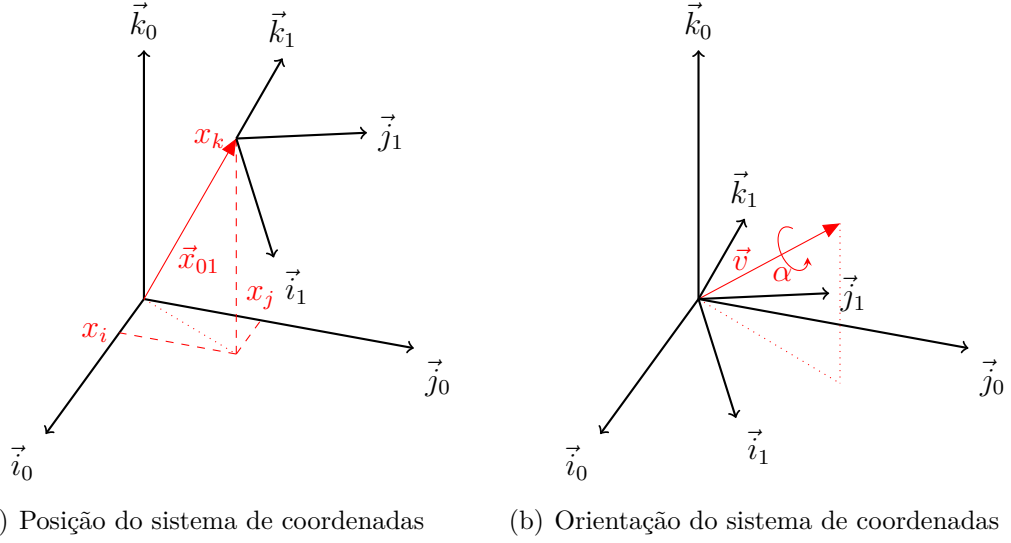


Figura 2.2: Representação de dois sistemas de coordenadas ortonormais.

$q_e \geq 0$, é dada por

$$q_e = \frac{1}{2} \sqrt{r_{11} + r_{22} + r_{33} + 1}, \quad q_v = \frac{1}{2} \begin{bmatrix} \operatorname{sgn}(r_{32} - r_{23}) \sqrt{r_{11} - r_{22} - r_{33} + 1} \\ \operatorname{sgn}(r_{13} - r_{31}) \sqrt{r_{22} - r_{33} - r_{11} + 1} \\ \operatorname{sgn}(r_{21} - r_{12}) \sqrt{r_{33} - r_{11} - r_{22} + 1} \end{bmatrix}$$

com a função sinal definida por $\operatorname{sgn}(u) = 1$, se $y \geq 0$, ou $\operatorname{sgn}(u) = -1$, para os demais casos (Siciliano et al. 2009).

O quaternion não é a única representação alternativa para a orientação, como destaque pode-se citar as representações mínimas de ângulos de Euler e *Roll-Pitch-Yaw* (Siciliano et al. 2009). A opção por utilizar o quaternion resulta de sua eficiência computacional e ausência de singularidades (Yuan 1988).

A cinemática pode ser classificada como direta ou inversa. No primeiro caso a pose (p) do efetuador (sistema de coordenadas azul na Figura 2.1), em relação a base (sistema de coordenadas verde), é determinada em a partir do conjunto de posições das juntas (θ) (ver Figura 2.1),

$$p = \begin{bmatrix} x \\ q \end{bmatrix} = h(\theta). \quad (2.1)$$

A expressão analítica de $h(\theta)$ é obtida através das relações geométricas entre as juntas e os comprimentos dos elos, próprias de cada manipulador. Uma alternativa são os métodos sistemáticos que podem ser aplicados de maneira padrão para qualquer configuração de robô, como por exemplo o método de Denavit-Hartenberg (Siciliano et al. 2009) ou a teoria de *Screw* (Murray et al. 1994).

O caso inverso, claramente, recupera as posições das juntas para uma dada pose do efetuador. Neste caso podem haver uma ou múltiplas soluções para uma mesma pose, como mostrado na Figura 2.3. Assim, para grande parte dos manipuladores, não existe solução algébrica fechada para a cinemática inversa (Murray et al. 1994, Siciliano et al. 2009).

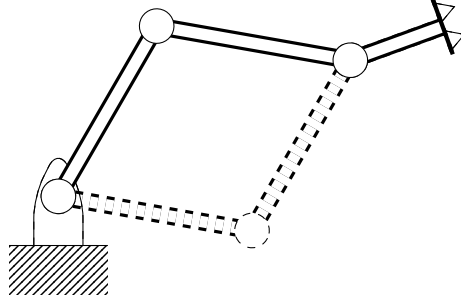


Figura 2.3: Exemplo de um manipulador que possui mais de uma solução de cinemática inversa para a mesma pose. Explicitamente, com cotovelo para cima ou para baixo.

2.1.1 Cinemática diferencial

A cinemática diferencial estabelece a relação entre a velocidade das juntas e as velocidades linear (\dot{x}) e angular (ω) de um ponto no corpo rígido. Esta relação é estabelecida por uma matriz chamada de Jacobiano geométrico, obtida através das relações geométricas inerentes da configuração do robô. Entretanto se considerarmos a expressão de cinemática direta (2.1), a velocidade pode ser obtida por derivação direta como

$$\dot{p} = \begin{bmatrix} \dot{x} \\ \dot{q} \end{bmatrix} = \frac{\partial h(\theta)}{\partial \theta} \dot{\theta} \quad (2.2)$$

onde a derivada parcial em relação as juntas é denominada Jacobiano analítico, representada por J_a e pertencente ao espaço $\mathbb{R}^{7 \times n}$. A relação entre a velocidade da representação por quaternions e a velocidade angular é dada pela expressão

$$\dot{q} = \frac{1}{2} J_q(q) \omega = \frac{1}{2} \begin{bmatrix} -q_v^\top \\ q_e I - q_v \times \end{bmatrix} \omega$$

na qual o operador \times representa o produto vetorial. A matriz tipo Jacobiano J_q possui colunas ortogonais, $J_q J_q^\top = \frac{1}{4} I$ e $J_q(q)^\top q = 0$ (Lizarralde & Wen 1996).

Se também for considerada a velocidade linear, \dot{p} e $v = \begin{bmatrix} \dot{x} & \omega \end{bmatrix}^\top$ são relacionados através do Jacobiano da representação (J_r)

$$\dot{p} = J_r v = \begin{bmatrix} I & 0 \\ 0 & \frac{1}{2} J_q \end{bmatrix} v. \quad (2.3)$$

Levando esta relação em (2.2), obtêm-se

$$v = J_r J_a \dot{\theta} = J \dot{\theta} = \begin{bmatrix} J_p \\ J_o \end{bmatrix} \dot{\theta} \quad (2.4)$$

que relaciona J_a com o Jacobiano geométrico ($J \in \mathbb{R}^{6 \times n}$), que por sua vez é dividido em duas parcelas $J_p \in \mathbb{R}^{3 \times n}$ e $J_o \in \mathbb{R}^{3 \times n}$ que relacionam as velocidades da juntas a velocidade linear e angular respectivamente. Destaca-se que a dimensão de J_a e a expressão de J_r variam de acordo com a representação escolhida para a orientação. Entretanto, o J é inerente ao manipulador e permanece igual, independente da representação. Assim como no caso da cinemática direta os métodos sistemáticos de Denavit-Hartenberg e *Twist* fornecem expressões fechadas para obter o Jacobiano geométrico de um manipulador genérico.

2.2 Modelo Dinâmico

Na dinâmica estuda-se a relação entre os torques e forças aplicados nas juntas e os consequentes forças lineares e momentos angulares nos elos. Para tal, considera-se que um robô pode ser representado pelo seguinte modelo de Euler-Lagrange para manipuladores seriais de cadeia aberta com n juntas (Siciliano et al. 2009)

$$M(\theta) \ddot{\theta} + C(\theta, \dot{\theta}) \dot{\theta} + g(\theta) = \tau + J^T(\theta) f, \quad (2.5)$$

no qual $M(\theta) \in \mathbb{R}^{n \times n}$ é a matriz de inércias, $C(\theta, \dot{\theta}) \dot{\theta} \in \mathbb{R}^n$ representa os efeitos centrífugos e de Coriolis, $g(\theta) \in \mathbb{R}^n$ são os torques devido a gravidade, $\tau \in \mathbb{R}^n$ é o vetor dos torques aplicados as juntas e $f \in \mathbb{R}^m$ é um vetor de forças e momentos aplicados no efetuador, com m sendo a dimensão do espaço operacional.

Ressalta-se que o modelo de (2.5) possui, dentre outras, as seguintes propriedades (Siciliano et al. 2009):

- P1.** $M(\theta)$ é simétrica positiva definida e limitada, $\lambda_{\min} \leq \|M\| \leq \lambda_{\max}$, para todo $\theta \in \mathbb{R}^n$;
- P2.** $u^T (\dot{M} - 2C) u = 0$, $\forall u \in \mathbb{R}^n$;
- P3.** Existe um vetor de parâmetros dinâmicos constantes $a \in \mathbb{R}^{n_d}$ e uma matriz regressora dinâmica $Y \in \mathbb{R}^{n \times n_d}$, tal que $Y(\theta, \dot{\theta}, \ddot{\theta}) a = \tau$.

Observação 1. A matriz $C(\theta, \dot{\theta})$ não é unicamente definida, assim a propriedade P2 é válida somente com uma escolha adequada desta. Um exemplo de representação que garante a característica de anti-simetria são os símbolos de Christoffel (Murray et al. 1994, Siciliano et al. 2009).

Se considerarmos a relação (2.2) de cinemática diferencial e a dualidade entre velocidade e força

$$\tau = J^T F,$$

é possível reescrever (2.5) no espaço Cartesiano como

$$\bar{M}(\theta)\dot{v} + \bar{C}(\theta, \dot{\theta})v + \bar{G}(\theta) = F - f. \quad (2.6)$$

Sendo a relação entre v e a derivada temporal da pose (\dot{p}), obtida por (2.3). As novas matrizes de inércia $\bar{M} \in \mathbb{R}^{6 \times 6}$, de Coriolis $\bar{C} \in \mathbb{R}^{6 \times 6}$ e de forças gravitacionais $\bar{G} \in \mathbb{R}^6$ são definidas por

$$\bar{M} = (J^\dagger)^T \bar{M} J^\dagger \quad \bar{C} = (J^\dagger)^T (\bar{M} \dot{J}^\dagger + \bar{C} J^\dagger) \quad \bar{G} = (J^\dagger)^T \bar{G}$$

e mantêm as propriedades do modelo no espaço das juntas. Os termos F e f representam, respectivamente, as forças e momentos resultantes do torque aplicado as juntas e da interação com o ambiente no efetuador do manipulador.

2.3 Múltiplos manipuladores

A interação entre dois ou mais manipuladores através de um objeto manipulado, ferramenta de trabalho ou contato direto, acarreta em restrições de movimento que alteram a dinâmica e a cinemática dos robôs. Em particular, se os manipuladores estão segurando firmemente o objeto, a pose do centro de massa deste pode ser descrita considerando-a com um efetuador virtual para cada um dos manipuladores. Com isso, o conjunto forma uma cadeia fechada e cada efetuador possui uma junta passiva no seu efetuador real.

Entretanto, como a estratégia de cooperação utilizada neste trabalho permite que se considere cada manipulador de forma independente, mesmo com o objeto em comum. Nenhuma das estratégias de modelagem diferente do apresentado anteriormente será abordada, sugerindo-se a consulta à (Caccavale & Uchiyama 2008, Murray et al. 1994, Siciliano et al. 2009).

Capítulo 3

Teleoperação Cooperativa

Utilizando os conceitos apresentados no Capítulo 1 será definida uma estratégia de teleoperação MMMS para múltiplos manipuladores transportando uma carga. Esta estratégia é apresentada de forma independente à lei de controle que atua sobre os robôs escravos e considera os mestres como dispositivos hápticos.

3.1 Definição do problema

Na teleoperação, os robôs que executam a tarefa e o operador que a comanda estão em locais distintos. O ambiente onde a tarefa é realizada será denominado local, enquanto o ambiente onde está o operador será o remoto.

No ambiente local, considere N robôs segurando uma carga deformável parcialmente desconhecida através de um contato pontual com atrito, como representado na Figura 3.1. Neste tipo de contato somente forças, e não torques, são transmitidas.

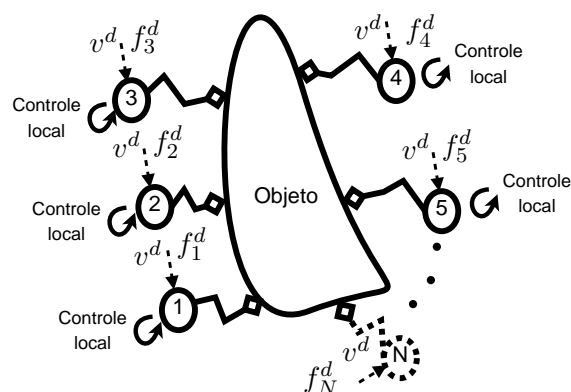


Figura 3.1: Representação do problema de transporte de carga abordado.

Deseja-se movimentar a carga rastreado uma trajetória de velocidade sem que nenhum contato seja perdido, mantendo-se assim o objeto seguro entre os robôs. Supondo uma rede local com banda limitada é necessário minimizar o seu uso,

optando-se por um controle descentralizado sem comunicação direta entre os agentes. Como entradas, cada robô recebe uma referência de força (f_i^d) determinada de forma a manter uma pegada segura e uma referência de velocidade comum (v^d), definindo a trajetória do objeto.

As referências de força devem ser escolhidas de forma a garantir um critério de pegada, por força ou por forma. Os dois critérios podem exigir o conhecimento de propriedades do objeto transportado, como geometria e peso (Prattichizzo & Trinkle 2008). Este problema não é diretamente considerado neste trabalho.

A referência de velocidade é gerada a partir da movimentação realizada pelo operador sobre os mestres. Este também transmite um sinal de referência de força adicional a força de pegada, de forma a apertar o objeto. Como realimentação para o operador, os mestres são mantidos sobre uma restrição de distância e o ambiente local transmite os erros de força referentes ao controle de pegada.

Como visto na Seção 1.2.1, a teleoperação bilateral pode apresentar instabilidade devido a presença de atraso no canal de comunicação. Para garantir a estabilidade, e portanto a segurança, o controle será implementado seguindo uma estratégia de duas camadas, representada na Figura 3.2. Como apresentado na Seção 1.2.1, o projeto em camadas permite que a estabilidade seja alcançada independentemente da lei de controle, que deve garantir a transparência (Franken et al. 2009, Stramigioli et al. 2010).

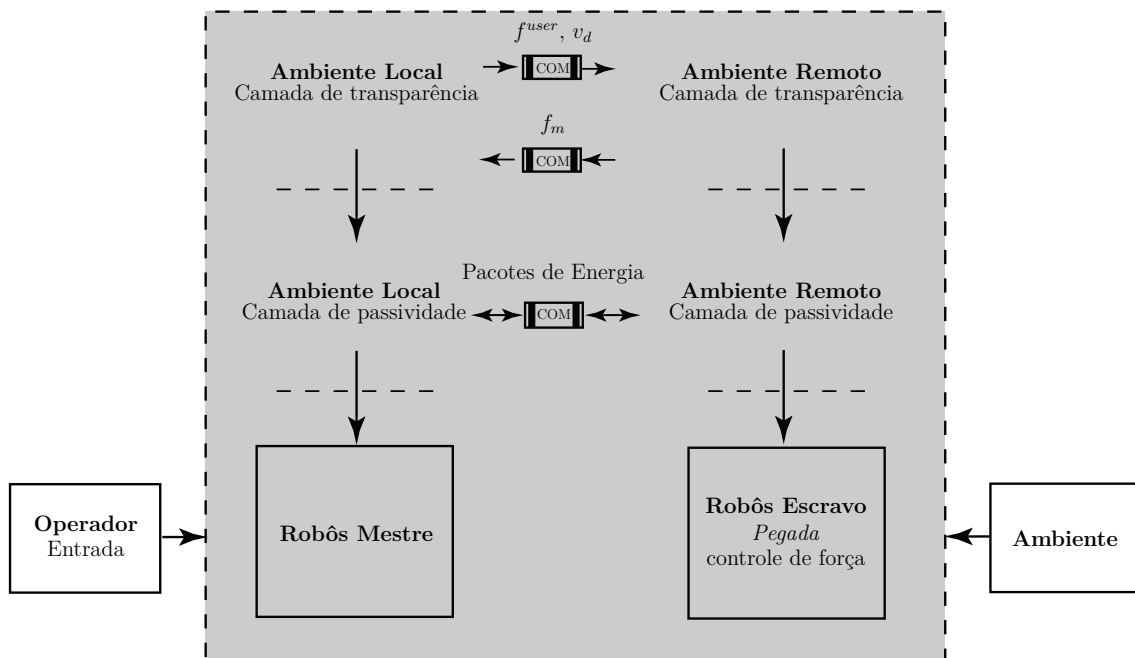


Figura 3.2: Diagrama representando as camadas de controle. COM: canal de comunicação; v_d : referência de velocidade; f^{user} : referência de força do usuário; e f_m : força a ser realimentada ao operador.

A seguir serão apresentadas as camadas de transparência e de passividade, projetadas para o uso com múltiplos robôs e considerando a possibilidade de controle cinemático ou dinâmico dos manipuladores.

3.2 Camada de Transparência

A camada de transparência fornece as ferramentas para que a tarefa de teleoperação seja cumprida, estabelecendo as referências para o movimento dos robôs escravos e a força para realimentação do operador. Como os mestres são considerados dispositivos hápticos, os seus movimentos são impostos diretamente pelo operador e já possuem um mecanismo para impor a força de realimentação, não sendo necessário projetar uma lei de controle. Para os escravos, a estratégia de teleoperação considera somente o projeto das referências de força e de velocidade para o efetuador. As leis de controle para seguir estas referências são projetadas no Capítulo 4.

Para gerar a referência de velocidade transmitida aos escravos, considera-se uma carga virtual sem massa sendo transportada pelos mestres. A carga do ambiente local deve então se movimentar como esta carga virtual mestre, levando há

$$v^d = \frac{1}{N} \sum_{i=1}^N \dot{x}_i^m \quad (3.1)$$

onde o índice sobrescrito m indica as variáveis referentes aos robôs mestres. A força de referência a ser aplicada a carga por cada robô escravo é

$$f_i^d = f_i^{min} + f_i^{dm} \quad (3.2)$$

composta por uma força constante f_i^{min} e uma força determinada pelo operador f_i^{dm} que permite ao mesmo comprimir a carga. O termo constante é determinado de forma a garantir a condição de pegada independentemente do usuário.

A força aplicada pelo operador, determinada no ambiente remoto, é obtida pela deformação da carga virtual mestre. Essa deformação é modelada como um conjunto de molas, com coeficiente de elasticidade k_m e comprimento L , ligando os efetuadores de cada mestre ao centro da carga. A força então é dada por

$$f_{\vec{d}}^{dm} = \frac{k_m}{n_{\vec{d}}} \sum_{j \in \Omega_{\vec{d}}} (\|x_j^m - x_c^m\| - L) \vec{d} \quad (3.3)$$

onde os manipuladores são agrupados nos conjuntos $\Omega_{\vec{d}}$, com $n_{\vec{d}}$ elementos, de acordo com a direção \vec{d} da força que exercem sobre a carga. A posição x_c^m da carga virtual mestre é definida como a média das posições dos efetuadores de todos os manipuladores mestre. Como, por definição, a força aplicada pelo operador pode somente

comprimir a carga uma camada intermediária no ambiente remoto divide f_d^{dm} de acordo com o sinal do termo $\|x_j^m - x_c^m\| - L$. Quando este é negativo, a força é enviada ao ambiente local para ser aplicada à carga. Quando positivo, a força é diretamente realimentada ao operador, pois esta violaria a condição de pegada estabelecida no sistema escravo.

Como f_i^{min} é projetada para garantir a segurança da carga, sendo aplicada independente da movimentação da mesma, a realimentação de força considera somente o erro relativo a este patamar. De forma semelhante a f^{dm} , a realimentação é realizada de acordo com a direção da força exercida sobre a carga considerando uma medida da deformação nesta direção. Assim temos

$$f_d^m = \frac{1}{n_d} \sum_{j \in \Omega_d} \|f_j\| - \|f_i^{min}\| \quad (3.4)$$

aplicada aos respectivos robôs mestre com o sentido determinado de acordo com a posição do efetuador em relação ao centro da carga (ver exemplo).

Destaca-se que f_i^{min} é determinada no ambiente local, enquanto f_i^{dm} é uma referência transmitida a partir do ambiente remoto. Assim, a pegada que garante que o objeto se mantenha entre os robôs é totalmente tratada no local de realização da tarefa, não sendo influenciada por atrasos, perda de pacotes e outros efeitos do canal de comunicação. A estabilidade de f_i^{dm} , por outro lado é garantida pela presença da camada de passividade, descrita a seguir.

Exemplo. Como um exemplo para aplicação da camada de transparência considere a teleoperação com $N = 4$, na qual em cada ambiente dois robôs estão alinhados ao eixo x e dois ao eixo y , como na Figura 3.3. Neste caso, existiriam dois conjuntos para os cálculos das referências: $\Omega_{\bar{y}}$ com os robôs de número 1 e 2; e $\Omega_{\bar{x}}$ com os robôs 3 e 4.

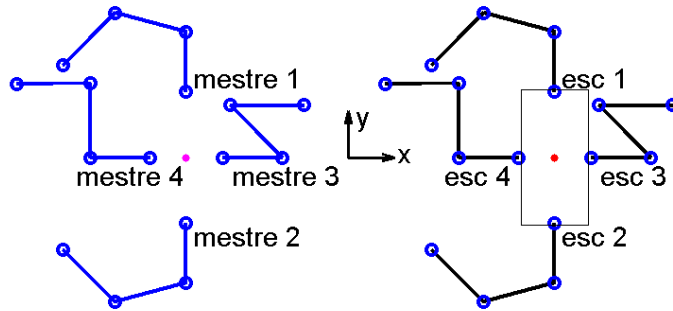


Figura 3.3: Representação dos manipuladores mestre (linhas azuis) e escravo (linhas pretas) em uma teleoperação com $N = 4$. Juntos aos escravos a carga é representada pelo retângulo com x_c no ponto vermelho. O centro da carga virtual (x_c^m) é representado pelo ponto magenta junto aos mestres.

As referências calculadas em cada um dos ambientes são mostradas na Tabela 3.1. Pode-se observar o agrupamento dos manipuladores e a definição dos sinais para a aplicação nos robôs de acordo com a posição deste em relação ao centro de massa do objeto, real ou virtual.

Tabela 3.1: Referências calculadas no exemplo de camada de transparência.

Ambiente Remoto	
Referência de velocidade	$v^d = \frac{1}{4} (\dot{x}_1^m + \dot{x}_2^m + \dot{x}_3^m + \dot{x}_4^m)$
Referência de força	$f_{\vec{x}}^{dm} = \frac{k_m}{2} [(\ x_3^m - x_c^m\ - L) + (\ x_4^m - x_c^m\ - L)] \vec{x}$ $f_{\vec{y}}^{dm} = \frac{k_m}{2} [(\ x_1^m - x_c^m\ - L) + (\ x_2^m - x_c^m\ - L)] \vec{y}$
Força aplicada ao operador	$f_3^m = f_{\vec{x}}^m \quad f_4^m = -f_{\vec{x}}^m$ $f_1^m = f_{\vec{y}}^m \quad f_2^m = -f_{\vec{y}}^m$
Ambiente Local	
Força de realimentação	$f_{\vec{x}}^m = \frac{1}{2} (\ f_3\ - \ f_3^{min}\ + \ f_4\ - \ f_4^{min}\)$ $f_{\vec{y}}^m = \frac{1}{2} (\ f_1\ - \ f_1^{min}\ + \ f_2\ - \ f_2^{min}\)$
Referência de força	$f_3^d = f_3^{min} + f_{\vec{x}}^{dm}, \quad f_4^d = f_4^{min} - f_{\vec{x}}^{dm}, \quad f_3^{min} = -f_4^{min}$ $f_1^d = f_1^{min} + f_{\vec{y}}^{dm}, \quad f_2^d = f_2^{min} - f_{\vec{y}}^{dm}, \quad f_1^{min} = -f_2^{min}$

3.3 Camada de Passividade

Abaixo da camada de transparência, a segunda camada de controle garante a estabilidade do sistema. Essa camada monitora a energia dissipada e acumulada durante a movimentação dos robôs, em cada ambiente, e então restringe os sinais de controle aplicados a esses de forma a garantir que o sistema se comporte de forma passiva.

A monitoração da energia é realizada considerando o controle como um sistema discreto de acordo com o proposto em (Franken et al. 2009). A cada instante de amostragem k a energia disponível para a movimentação dos robôs é estimada como

$$H_j(k) = H_j(k-1) + H_{j+} - H_{j-} - \Delta H_j \quad (3.5)$$

sendo o subíndice j igual m para o mestre e s para o escravo. $H_j(k-1)$ representa o nível de energia presente na amostragem anterior e os termos H_{j+} e H_{j-} são usados para troca de energia entre os ambientes local e remoto. Uma forma simples para essa transferência de energia é mostrada em (Stramigioli et al. 2010) e consiste em enviar sempre uma parcela da energia disponível, desta forma

$$H_{j-} = \beta H(k-1) \quad (3.6)$$

é transmitida a cada instante k . Outras estratégias para troca de energia são encontradas em (Stramigioli et al. 2010).

No lado onde se recebe o pacote é formada uma fila (Q_j) até a ocorrência de um novo passo do controlador. Neste instante, o termo aditivo de energia é computado a partir dos pacotes na fila através de

$$H_{j+} = \sum_{H \in Q_j} H_- \quad (3.7)$$

e a fila é esvaziada. Essa estratégia tende a manter os tanques de energia em cada um dos ambientes com o mesmo nível, mantendo assim os mestres e os escravos com possibilidade de movimentação.

O último termo de (3.5), ΔH_j , representa a energia trocada entre o robô e o ambiente ou operador. Esse termo pode adicionar ou remover energia do tanque. Essa interação pode ser estimada a partir da força de interação medida e posição Cartesiana calculada através da cinemática direta. Considerando o controle como um sistema Port-Hamiltoniano discreto entre duas amostras a energia é obtida de

$$\Delta H = f_j^T(x_j(k) - x_j(k-1)) , \quad (3.8)$$

na qual f_j e x_j são vetores com as forças e posições pareadas de todos os robôs em um dos ambientes. Destaca-se que (3.8) é uma representação Cartesiana da estimativa de energia obtida em (Stramigioli et al. 2005). Note que no caso de robôs sob controle dinâmico a força medida pode ser substituída pelo torque de comando, utilizando assim somente a medida das posições das juntas.

A estabilidade é imposta ao sistema assegurando-se que os robôs são atuados respeitando o nível de energia disponível no tanque correspondente. Nos trabalhos originais de Franken et al. (2009) e Stramigioli et al. (2010) essa restrição é aplicada utilizando-se um sinal de força saturado

$$F_j^{sat} = \min(F_{max1}, \|F_j\|) \frac{F_j}{\|F_j\|} \quad (3.9)$$

com $\min(\cdot)$ representando a função que seleciona o menor valor e F_{max1} igual a zero quando não há energia no tanque ou ∞ , caso contrário.

Nota-se, entretanto, de (3.8) que um efeito semelhante pode ser obtido se a saturação for realizada no sinal de velocidade, no lugar da força. Deste modo, em uma malha de controle cinemático, (3.9) será substituída por

$$\dot{x}_j^{sat} = \min(\dot{x}_{max1}, \|\dot{x}_j\|) \frac{\dot{x}_j}{\|\dot{x}_j\|}, \quad (3.10)$$

onde \dot{x}_{max1} corresponde a F_{max1} . Acrescentando-se novos termos a função mínimo é possível considerar limites específicos de acordo com a aplicação, por exemplo

restringir uma força máxima ou liberar um objeto de forma suave (Franken et al. 2009). No presente caso, onde o ambiente remoto transmite referências de velocidade e de força, ambas sofrem saturação na camada de passividade do ambiente local.

Até aqui, a camada de passividade monitora a energia de forma a garantir que não ocorra restrição a condição de passividade. Entretanto, essa ação tende a levar o sistema a condição de energia armazenada nula, onde, sem movimento, a estabilidade é garantida. Para permitir que o sistema possa recuperar energia quando necessário, permitindo o movimento, o operador é utilizado como fonte. Isto é realizado através dos robôs mestres aplicando-se uma pequena força (F_{TLC}), adicional a de realimentação, se opondo ao movimento do operador. Essa força é definida como

$$F_{TLC} = -\alpha \cdot (H^d - H_m(k))\dot{x}_m, \quad (3.11)$$

sendo aplicada sempre que o tanque está abaixo de um nível H^d de referência.

A estabilidade gerada pela estratégia proposta é intuitiva, considerando balanço de energia e a restrição de movimento imposta de maneira a manter a condição de passividade válida. Validações experimentais desta estratégia são realizadas em Franken et al. (2009) e Stramigioli et al. (2010). É mostrado que o sistema permanece estável mesmo com grandes atrasos no canal de comunicação, contatos rígidos entre o escravo e o ambiente, e diferentes graus de rigidez na pegada do operador no efetuador do dispositivo mestre.

A definição da camada de transparência não requer o uso de múltiplos mestres, como no experimento apresentado no Capítulo 5. Entretanto, no caso de um único mestre a realimentação de força é realizada a partir da interação da carga com o ambiente, que nem sempre pode ser medida. Além disso, não se faz necessário o uso de um número igual de robôs mestres e escravos, somente as direções, em relação ao sistema de coordenadas inercial, das forças de contato devem ser pareadas.

Ao considerar a presença de múltiplos mestres é possível realizar a teleoperação dos escravos de forma independente antes de pegar o objeto, facilitando a obtenção da condição de pegada firme existente em (Aldana et al. 2012, Lee & Spong 2005, Sirouspour 2005) e suposta durante o projeto de controle. Entretanto, diferente da proposta de múltiplos manipulares em (Bacocco & Melchiorri 2009), o uso de uma estratégia de cooperação garante a pegada independentemente da ação do operador, sujeita a atraso, perda de pacotes, falha na comunicação, entre outros efeitos. Outra diferença para outros trabalhos na área teleoperação cooperativa é que não se considera um contato rígido entre efetuadores e carga, acomodando-se a complacência natural do objeto, do sensor de força e de outros componentes do sistema.

Na Figura 3.4 é apresentado um resumo das camadas de transparência e passividade. São representados os dois ambientes com as respectivas equações, sinais transmitidos e informações recebidas e enviadas aos manipuladores.

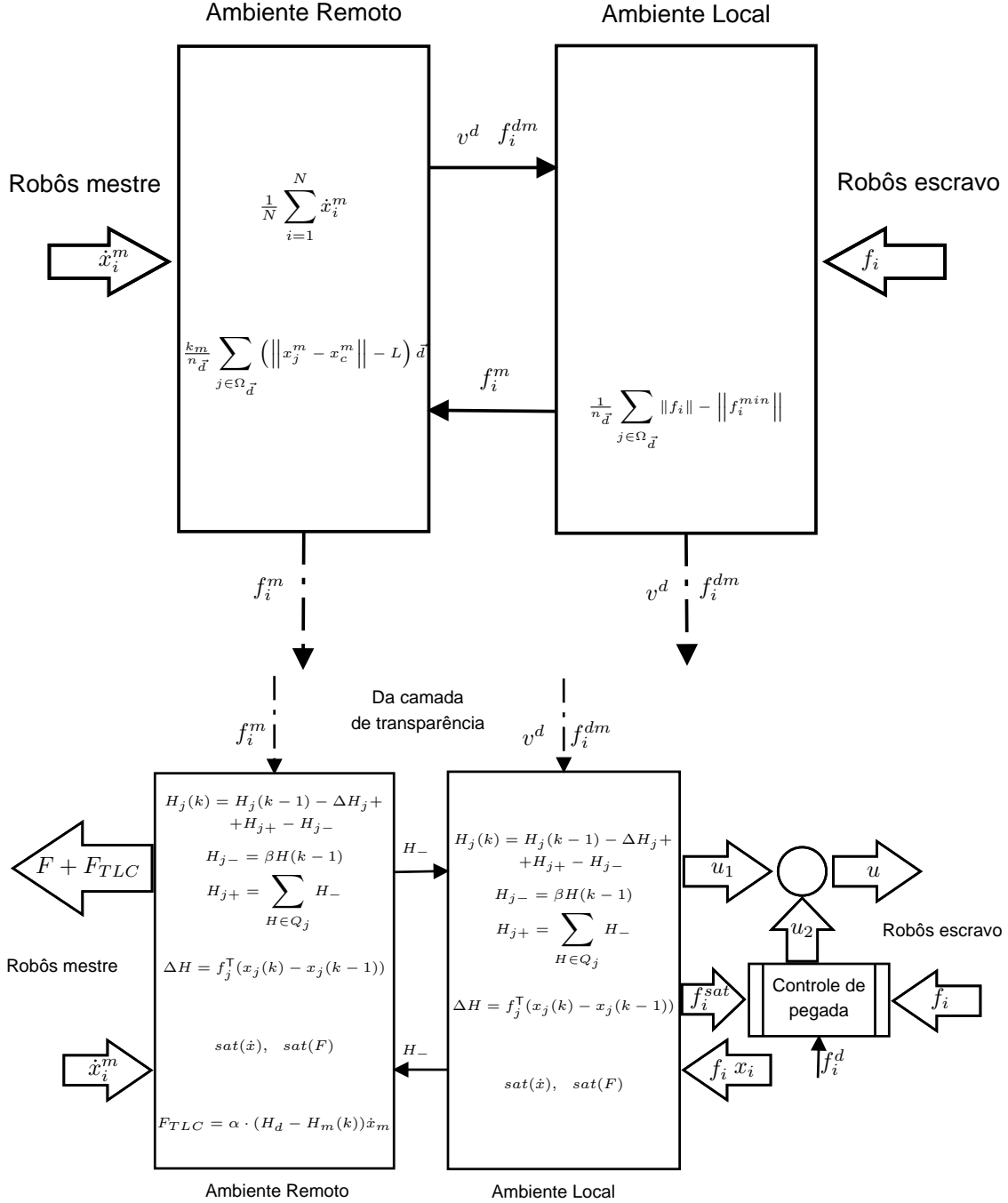


Figura 3.4: Diagrama representando as camadas de transparência e passividade. Na camada de transparência, são calculadas as referências de velocidade (v^d) e força (f^{dm}) para os manipuladores escravos a partir de uma carga virtual deformável de massa nula. No ambiente local são calculadas as forças para a realimentação do operador, estas são as forças de contato que diferem da força de pegada (f^{min}). Na camada de passividade, estima-se a energia ($H(k)$) trocada entre manipuladores e ambiente, considera-se uma parcela a ser enviada para (H_-) e outra recebida (h_+) do outro ambiente, além da estimativa de energia gasta ou obtida com o movimento (ΔH). A partir de $H(k)$ os sinais da camada de transparência são modificados para garantir a condição de passividade. No ambiente remoto aplica-se ao operador uma força oposta ao movimento (F_{TLC}) para recuperar energia.

Capítulo 4

Controle Cooperativo

A estratégia de teleoperação proposta no Capítulo 3 impõe ao ambiente local somente as referências de velocidade e de força, sendo necessário projetar uma lei de controle que garanta que os efetadores do escravos sigam estas referências. Neste capítulo serão descritas três alternativas para controlar os manipuladores escravos, uma para o caso cinemático, comum em robôs industriais, e duas para robôs com dinâmica não desprezível.

Os controladores propostos são descentralizados, no sentido de que cada robô utiliza somente as informações de seus próprios sensores para fechar a malha de controle, como representado no diagrama de blocos da Figura 4.1. Nota-se, entretanto, que o controle da força de contato com a carga representa uma comunicação indireta de forma a estabelecer a cooperação.

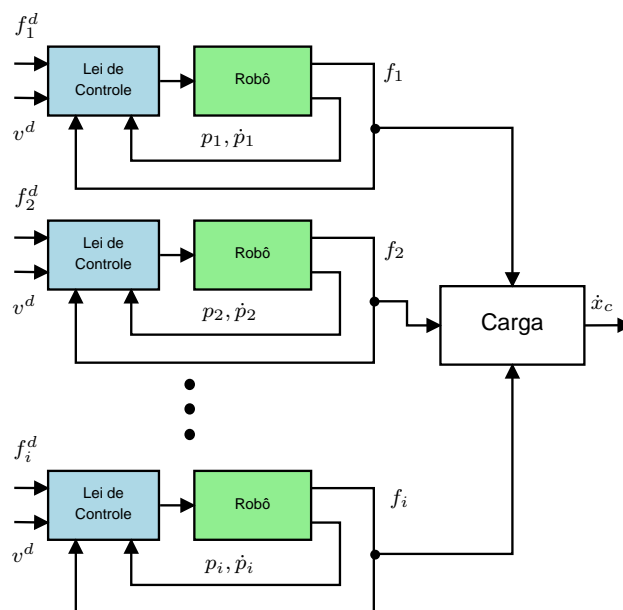


Figura 4.1: Diagrama de blocos representando o controle cooperativo descentralizado. Cada manipulador é controlado utilizando somente informações de seus próprios sensores, sem comunicação direta com os demais.

4.1 Modelagem

Para o projeto do controlador do sistema cooperativo, considera-se primeiramente a carga sem deformação com o robô i em contato no ponto a_i . Se a orientação do objeto é representada por uma matriz de rotação $R(\theta_c)$ então:

$$a_i(t) = x_c(t) + R(\theta_c) r_i \quad \text{e} \quad \dot{a}_i(t) = \dot{x}_c(t) + \dot{R}(\theta_c) r_i$$

sendo x_c a posição inercial do centro de massa da carga e r_i um vetor fixo no sistema de coordenadas inercial, E_0 (ver Figura 3.1).

A posição $a_i(t)$ representa, então, o ponto de contato do i -ésimo robô sem que este deforme a carga em um dado tempo t . Como os robôs se movem, a carga pode ser, eventualmente, comprimida com $x_i \neq a_i$. Pode-se assim ser definida uma deformação $z_i = x_i - a_i$, como representado na Figura 4.2, responsável por uma força f_i no contato entre a carga e o efetuator do i -ésimo robô. Sob a hipótese de que não ocorre rotação da carga ($\dot{R}(\theta_c) = 0$), a variação da deformação ao longo do tempo é dada por

$$\dot{z}_i = \dot{x}_i - \dot{a}_i = \dot{x}_i - \dot{x}_c \quad (4.1)$$

e assume-se que a mesma é limitada, *i.e.*, existe $c < \infty$ constante tal que $\|\dot{z}_i\| \leq c$.

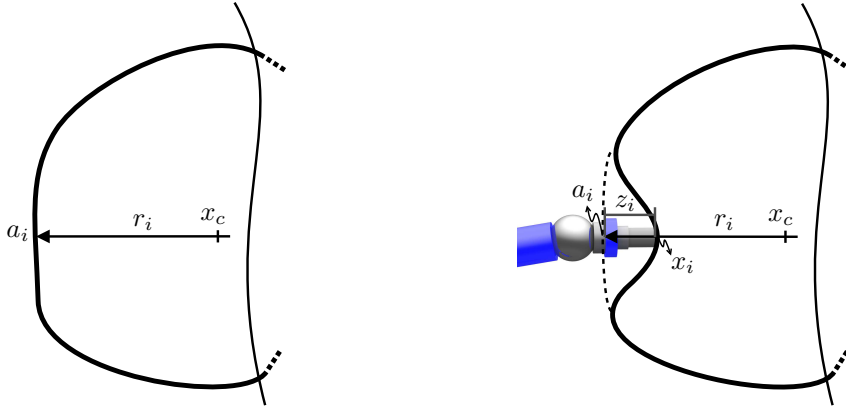


Figura 4.2: Representação da interação entre um manipulador e o objeto.

A força f_i pode ser considerada como o gradiente ($\nabla P(z_i)$) de uma função potencial positiva definida da deformação. Por definição, quando $z_i = 0$ o i -ésimo robô não deforma o objeto, então nenhuma força de reação é gerada e as condições

$$P(z_i) = 0 \iff z_i = 0 \quad (4.2)$$

$$\nabla P(z_i) = 0 \iff z_i = 0 \quad (4.3)$$

devem ser satisfeitas por $P(z_i)$ (Bai & Wen 2010). Nota-se que como a_i reside sobre a superfície externa do objeto não deformado, z_i e portanto f_i são limitados

inferiormente por zero. Isto representa a condição de que o efetuador de um robô não é capaz de puxar a carga.

Neste trabalho, diferentemente de Bai & Wen (2010), os seis graus de liberdade do efetuador são considerados no sistema de controle. Os três graus de posição formam um objetivo principal de movimentação translacional da carga. Enquanto, a orientação é controlada com o objetivo secundário de manter os efetuadores perpendiculares à face do objeto.

As equações de movimento que regem o sistema cooperativo com os robôs e a carga, representam um sistema com $N + 1$ agentes. Os manipuladores são modelados como no Capítulo 2, de acordo com o tipo de controle considerado e a dinâmica da carga é dada por

$$M_c \ddot{x}_c = \sum_{i=1}^N f_i, \quad (4.4)$$

sendo M_c a massa da carga e f_i as forças que os robôs impõem sobre a mesma.

De (4.4) tem-se que para a carga se movimentar com uma velocidade desejada, as referências para as forças de interação devem atender, além das condições de pegada (Prattichizzo & Trinkle 2008), a condição de proporcionalidade

$$\sum_{i=1}^N f_i^d \propto \dot{v}_d$$

de forma que, quando a velocidade desejada é constante a restrição

$$\sum_{i=1}^N f_i^d = 0 \quad (4.5)$$

é atendida. Apesar de na teleoperação as velocidades do sistema não permanecerem sempre constantes, por simplificação, este trabalho supõe que as forças de referência para os efetuadores respeitam (4.5). Assim, existe um resíduo na força aplicada para comportar o acoplamento com o controle de velocidade.

Nota-se que as forças de reação f_i dependem das deformações z_i , desta forma regular a força para um valor desejado é semelhante a regular a deformação provocada por cada robô. Isto é similar ao problema de formação, onde a posição relativa entre agentes é regulada para um valor desejado (Bai & Wen 2010). Para este fim, assume-se que a força desejada, assim com a medida, é modelada por $f_i^d = \nabla P(z_i^d)$, o gradiente de uma função potencial da deformação desejada (z_i^d).

4.2 Controle Cinemático

Em robôs industriais é comum que não se tenha acesso direto as malhas de controle de torque das juntas, sendo normalmente empregado um controle cinemático. Nesta seção, o problema de transporte de carga descrito anteriormente será resolvido considerando esta modalidade de controle. Para tal, considera-se que o movimento dos robôs é modelado através da cinemática diferencial por

$$\dot{x}_i = J_{pi}(\theta_i)u_i, \quad i = 1, \dots, N. \quad (4.6)$$

Na qual $J_{pi}(\theta_i)$ considera somente a parte referente às velocidades lineares da matriz Jacobiano (ver Capítulo 2), $\theta_i \in \mathbb{R}^n$ são as posições das juntas e $u_i \approx \dot{\theta}_i \in \mathbb{R}^n$ são os sinais de controle no espaço das juntas. Com isso, assume-se que o manipulador possui uma malha de controle de velocidade com desempenho muito bom.

4.2.1 Condição de estabilidade

A seguir será apresentada a lei de controle para a tarefa de cooperação. Com ela, será mostrado que o sistema é UB (do inglês *ultimately bounded*) em relação a uma parcela dos estados. Seguindo esta condição, o sistema é limitado para qualquer condição inicial dentro de uma determinada vizinhança. De maneira mais formal considere o sistema não linear

$$\dot{y}_1(t) = f_1(y_1(t), y_2(t)), \quad y_1(0) = y_{10}, \quad t \in \mathcal{I}_{y_0} \quad (4.7)$$

$$\dot{y}_2(t) = f_2(y_1(t), y_2(t)), \quad y_2(0) = y_{20} \quad (4.8)$$

no qual $y_1 \in \mathcal{D}$, $\mathcal{D} \in \mathbb{R}^{n_1}$ é um conjunto aberto com $0 \in \mathcal{D}$ e $y_2 \in \mathbb{R}^{n_2}$. A função $f_1 : \mathcal{D} \times \mathbb{R}^{n_2} \rightarrow \mathbb{R}^{n_1}$ é tal que $\forall y_2 \in \mathbb{R}^{n_2}$, $f_1(0, y_2) = 0$ e $f_1(\cdot, y_2)$ é localmente Lipschitz em y_1 e a função $f_2 : \mathcal{D} \times \mathbb{R}^{n_2} \rightarrow \mathbb{R}^{n_2}$ é tal que $\forall y_1 \in \mathcal{D}$, $f_2(y_1, \cdot)$ é localmente Lipschitz em y_2 . $\mathcal{I}_{y_0} := [0, \tau_{y_0})$, $0 < \tau_{y_0} \leq \infty$, é o maior intervalo de existência para solução $(y_1(t), y_2(t))$, $t \in \mathcal{I}_{y_0}$. A condição UB é dada pela seguinte definição (Haddad & Chellaboina 2011):

Definição 1. Um sistema dinâmico não linear representado por (4.7) e (4.8) é UB com respeito a y_1 uniformemente em y_2 com limite ϵ , se existe $\gamma > 0$ tal que, para todo $\delta \in (0, \gamma)$, existe $T = T(\delta, \epsilon) > 0$ e $\|y_{10}\| < \delta$ implica em $\|y_1(t)\| < \epsilon$, para $t \geq T$. O sistema (4.7) e (4.8) é globalmente UB com respeito a y_1 uniformemente em y_2 com limite ϵ , se para todo $\delta \in (0, \infty)$ existe $T = T(\delta, \epsilon) > 0$ tal que $\|y_{10}\| < \delta$ implica em $\|y_1(t)\| < \epsilon$, para $t \geq T$.

Para caracterizar um sistema dinâmico não linear como UB, o seguinte teorema baseado em funções de Lyapunov é utilizado¹.

Teorema 1. *Considere o sistema dinâmico não linear em (4.7) e (4.8). Assume-se que existe uma função continuamente diferenciável $V : \mathcal{D} \times \mathbb{R}^{n_2} \rightarrow \mathbb{R}$ e funções $\alpha(\cdot)$, $\beta(\cdot)$ da classe \mathcal{K} , tal que*

$$\alpha(\|y_1\|) \leq V(y_1, y_2) \leq \beta(\|y_1\|), \quad y_1 \in \mathcal{D}, \quad y_2 \in \mathbb{R}^{n_2} \quad (4.9)$$

Se existe uma função contínua $W : \mathcal{D} \rightarrow \mathbb{R}$ tal que $W(y_1) > 0$, $\|y_1\| > \mu$, e

$$\dot{V}(y_1, y_2) \leq -W(y_1), \quad y_1 \in \mathcal{D}, \quad \|y_1\| \geq \mu, \quad y_2 \in \mathbb{R}^{n_2} \quad (4.10)$$

com $\mu > 0$ sendo tal que $\mathcal{B}_{\alpha^{-1}(\eta)}(0) \subset \mathcal{D}$ para algum $\eta \leq \beta(\mu)$. Então o sistema dinâmico não linear é UB com respeito a y_1 uniformemente em y_2 com limite $\epsilon := \alpha^{-1}(\eta)$. Ademais, $\limsup_{t \rightarrow \infty} \|y_1(t)\| \leq \alpha^{-1}(\beta(\mu))$. Se, adicionalmente, $\mathcal{D} = \mathbb{R}^{n_1}$ e $\alpha(\cdot)$ é uma função da classe \mathcal{K}_∞ , então (4.7) e (4.8) representam um sistema globalmente UB com respeito a y_1 uniformemente em y_2 e para todo $y_{10} \in \mathbb{R}^{n_1}$, $\|y_1(t)\| \leq \epsilon$, $t \geq 0$.

Prova. A prova para este teorema está no Apêndice B.1. □

4.2.2 Estratégia de controle

Para alcançar o objetivo de movimentar a carga enquanto a pegada é garantida, é proposta uma lei de controle cinemático no espaço Cartesiano. Esta lei leva em consideração que ao realizar somente a translação da carga, existe redundância nos manipuladores para cumprir tarefas secundárias. Assim, a lei de controle, descentralizada em relação a cada robô, é

$$u_i = J_{pi}^\dagger \phi_i + J_{pi}^\# \mu_i \quad (4.11)$$

na qual $J_{pi}^\dagger = J_p^\top (J_p J_p^\top)^{-1}$ é a matriz pseudo inversa de Moore-Penrose do Jacobiano de posição e $J_{pi}^\# = (I - J_{pi}^\dagger J_{pi})$ a matriz de projeção no espaço nulo de J_{pi} (Siciliano et al. 2009). Os sinais de controle ϕ_i e μ_i são projetados para satisfazer os objetivos principal e secundário, a saber controle da posição e da orientação do efetuador, respectivamente. A malha de controle para um manipulador é representada pelo diagrama de blocos da Figura 4.3.

¹Este teorema corresponde ao 4.14 de (Haddad & Chellaboina 2011).

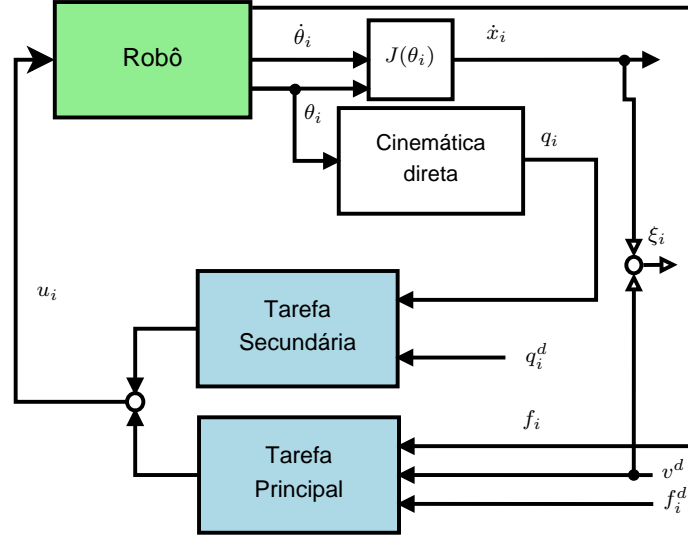


Figura 4.3: Diagrama de blocos da malha de controle cinemático de um manipulador na tarefa cooperativa.

Tarefa Principal

A tarefa principal do sistema cooperativo é garantir o rastreamento de uma referência de velocidade $v^d(t)$ pelos efetadores dos N robôs e pela carga, enquanto esta é mantida segura através de uma força que varia de forma limitada. Para tal é proposto o sinal de controle cinemático

$$\phi_i = v^d + K_i (f_i^d - f_i) \quad (4.12)$$

sendo K_i uma matriz positiva definida de ganho proporcional para o controle de força.

Considerando essa lei de controle em (4.6) e (4.11) e os erros de velocidade para os efetadores e a carga definidos como

$$\xi_i(t) = \dot{x}_i(t) - v^d(t) \quad \text{e} \quad \xi_c(t) = \dot{x}_c(t) - v^d(t)$$

o modelo em malha fechada com manipuladores cinemáticos e uma carga dinâmica é dado por

$$\xi_i = K_i (f_i^d - \nabla P(z_i)) \quad (4.13)$$

$$\dot{\xi}_c = -\dot{v}^d(t) + M_c^{-1} \sum_{i=1}^N \nabla P(z_i). \quad (4.14)$$

Destaca-se que a velocidade de deformação da carga (4.1) pode ser redefinida em termos destes erros de velocidade como

$$\dot{z}_i = \xi_i - \xi_c. \quad (4.15)$$

A interação entre os $N + 1$ agentes exibe então uma topologia de estrela com a carga no centro. Dessa maneira, controlar as forças, em consequência as deformações, entre os robôs e a carga simultaneamente garante que as posições relativas entre agentes são firmemente mantidas. Ressalta-se ainda que a carga é um agente passivo sem uma ação de controle direta. A informação da referência de velocidade é recuperada através das forças de contato f_i , que formam um canal de comunicação indireto entre os agentes.

Teorema 2. *Considere um sistema de robôs cooperativos cinemáticos (4.6) transportando uma carga deformável dinâmica (4.4). As forças de interação são consideradas como o gradiente de uma função potencial positiva definida da deformação da carga, submetida a condição (4.3). Suponha que os robôs são controlados de acordo com (4.11) utilizando o sinal de controle (4.12) para a tarefa principal. Se a velocidade de referência $v^d(t)$, sua derivada $\dot{v}^d(t)$ e a velocidade de deformação da carga $\dot{z}_i(t)$ são limitadas e a força desejada f^d satisfaz (4.5), então o sistema completo em malha fechada ((4.13) a (4.15)) na prática² rastreia (i.e. é UB) $v^d(t)$ com forças de interação f_i variando de forma limitada.*

Prova. Considere o sistema completo em malha fechada descrito por (4.13) a (4.15). Este, sistema, pode ser dividido em dois grupos $y_1 = z_i$ e $y_2 = \xi_c$. A seguir será mostrado que o sistema é UB com respeito a y_1 .

Para tal, a seguinte candidata a função de Lyapunov é proposta

$$V(z_i, \xi_c) = \sum_{i=1}^N [P_i(z_i) - P_i(z_i^d) - (f^d)^\top (z_i - z_i^d)] + \frac{1}{2} \xi_c^\top M_c \xi_c. \quad (4.16)$$

Como V é positiva definida e radialmente ilimitada, o Lema 4.3 de (Khalil 2002) (ver Apêndice B.2) pode ser evocado para garantir que a condição

$$\alpha(y_1) \leq V(y_1, y_2) \leq \beta(y_1)$$

seja válida.

²Aqui, prática é utilizado como na estabilidade prática, onde o sistema permanece em uma vizinhança do ponto de equilíbrio.

A derivada de V ao longo das trajetórias do sistema é dada por

$$\begin{aligned}
\dot{V} &= \sum_{i=1}^N [f_i^\top \dot{z}_i - (f_i^d)^\top \dot{z}_i^d - (f_i^d)^\top \dot{z}_i + (f_i)^\top \dot{z}_i^d] + \xi_c^\top M_c \dot{\xi}_c \\
&= \sum_{i=1}^N (f_i - f_i^d)^\top \dot{z}_i + \xi_c^\top \sum_{i=1}^N f_i - \xi_c^\top M_c \dot{v}^d \\
&= \sum_{i=1}^N (f_i - f_i^d)^\top (\xi_i - \xi_c) + \xi_c^\top \sum_{i=1}^N f_i - \xi_c^\top M_c \dot{v}^d \\
&= \sum_{i=1}^N (f_i - f_i^d)^\top \xi_i + \xi_c^\top \sum_{i=1}^N f_i^d - \xi_c^\top \sum_{i=1}^N f_i + \xi_c^\top \sum_{i=1}^N f_i - \xi_c^\top M_c \dot{v}^d \\
&= \sum_{i=1}^N (f_i - f_i^d)^\top \xi_i - \xi_c^\top M_c \dot{v}^d = - \sum_{i=1}^N (f_i - f_i^d)^\top K_i (f_i - f_i^d) - \xi_c^\top M_c \dot{v}^d \quad (4.17)
\end{aligned}$$

na qual a condição sobre a referência de força (4.5) é considerada. Nota-se que o segundo termo de (4.17) não possui um sinal definido, entretanto é possível encontrar um limitante superior para \dot{V} .

Para determinar o limitante superior será considerada a existência de um limite superior para a referência de aceleração (\dot{v}^d) e para a velocidade de deformação da carga (\dot{z}_i), além da relação (4.15), assim

$$\begin{aligned}
\dot{V} &\leq \sum_{i=1}^N -k_1 \|f_i - f_i^d\|^2 + k_2 \|\xi_c\| \|\dot{v}^d\| \\
&\leq \sum_{i=1}^N -k_1 \|f_i - f_i^d\|^2 + k_3 \sum_{i=1}^N \|\xi_i - \dot{z}_i\| \|\dot{v}^d\| \\
&\leq \sum_{i=1}^N -k_1 \|f_i - f_i^d\|^2 + k_3 \|K_i (f_i - f_i^d)\| \|\dot{v}^d\| + k_3 \|\dot{z}_i\| \|\dot{v}^d\| \\
&\leq \sum_{i=1}^N -k_1 \|f_i - f_i^d\|^2 + k_3 k_1 \|f_i - f_i^d\| \|\dot{v}^d\| + k_3 \|\dot{z}_i\| \|\dot{v}^d\| \\
&\leq \sum_{i=1}^N -k_1 \|f_i\|^2 + k_1 \|f_i^d\|^2 + k_4 \|f_i\| + k_4 \|f_i^d\| + k_5 \\
&\leq \sum_{i=1}^N -k_1 \|f_i\|^2 + k_4 \|f_i\| + k_6 := W(y_1). \quad (4.18)
\end{aligned}$$

Note que W é uma função de y_1 , pois $f_i = \nabla P(z_i)$. As constantes k_1 , k_4 e k_6 são

$$k_1 = \lambda_{\max}(K_i), \quad k_4 = \frac{\|M_c\| \|\dot{v}^d\|_{\max} \lambda_{\max}(K_i)}{N}, \quad k_6 = \frac{k_4}{k_1} \|\dot{z}_i\|_{\max} + k_1 \|f_i^d\| \left(\|f_i^d\| + \frac{k_4}{k_1} \right).$$

A condição para o sistema ser UB é que $W(y_1)$ seja negativa semi-definida fora de uma determinada vizinhança no espaço formado por y_1 . Para tal, observa-se que

as raízes de (4.18) são

$$\|f_i\| = \frac{k_4 \pm \sqrt{(k_4)^2 + 4k_1 \cdot k_6}}{2k_1}$$

sendo uma solução negativa e outra positiva. Assim, o limite da vizinhança é dado pela raiz positiva levando a condição

$$\|f_i\| > \frac{k_4}{k_1} + k \quad (4.19)$$

na qual k é uma constante em função de k_1 e k_6 . □

Observação 2. Destaca-se que a condição de y_1 limitado implica em ξ_i e ξ_c limitados devido as relações (4.13) e (4.15).

Tarefa Secundária

Devido a definição da forças de referência no sistema de coordenadas inercial e a limitação da não transmissão de momento, de acordo com o modelo de contato, os efetuadores dos N robôs devem se manter perpendiculares a superfície de contato da carga transportada.

Essa restrição é mantida pelo controlador da tarefa secundária que utiliza a movimentação livre das juntas no espaço nulo do Jacobiano para preservar a orientação adequada. Para isso, considere uma função custo, $h(\theta) : \mathbb{R}^n \mapsto \mathbb{R}$, convexa em relação a θ_i , escolhida de forma que o seu mínimo representa a orientação desejada. Desta forma, a tarefa secundária consiste em minimizar $h(\theta)$.

Para alcançar este objetivo, o sinal μ_i em (4.11) é proposto como

$$\mu_i = -K_n \frac{\partial h(\theta_i)}{\partial \theta_i} \quad (4.20)$$

sendo K_n uma matriz de ganho positiva definida. Este sinal consiste em uma velocidade na mesma direção e com o sentido oposto ao gradiente da função custo, apontando assim para o ponto de mínimo, onde então se torna nula.

Observação 3. Essa tarefa secundária consiste em um problema de otimização local com restrições representadas pela tarefa principal. Não há garantia de existência de uma solução global, em especial quando o manipulador está em uma configuração singular, onde há perda de posto da matriz Jacobiano (Murray et al. 1994, Siciliano et al. 2009), não existe projeção no espaço nulo dessa matriz.

Observação 4. A lei de controle (4.20) representa uma solução geral para o problema de realizar uma subtarefa que consista em minimizar uma função convexa.

Um exemplo simples de função custo adequada para esta tarefa é

$$h(\theta_i) = (\psi(\theta_i) - \psi_d)^\top (\psi(\theta_i) - \psi_d) \quad (4.21)$$

na qual $\psi(\theta_i)$ é uma representação vetorial da orientação do efetuador em termos das posições das juntas e ψ_d é a orientação desejada.

Uma escolha evidente para $\psi(\theta_i)$ consiste na parte relativa a orientação na expressão de cinemática direta (2.1). Desta maneira, a derivada parcial de h utilizará a parcela relativa a velocidade angular do Jacobiano analítico, J_{ao} (ver Seção 2.1.1). A seguir, demonstra-se que esta escolha garante a estabilidade da regulação caso os parâmetros do modelo cinemático sejam conhecidos. Para tal, considere que a velocidade das juntas é $\dot{\theta}_i = \mu_i$ e a candidata a função de Lyapunov $2V = h(\theta_i)$.

A derivada temporal de V ao longo das trajetórias do sistema é dada por

$$\begin{aligned} \dot{V} &= \frac{dh(\theta_i)}{dt} = \frac{d(\psi(\theta_i) - \psi_d)^\top}{dt} (\psi(\theta_i) - \psi_d) = \left(\frac{d\psi(\theta_i)}{d\theta_i} \dot{\theta}_i \right)^\top (\psi(\theta_i) - \psi_d) \\ &= \left(-J_{ao} K_n J_{ao}^\top (\psi(\theta_i) - \psi_d) \right)^\top (\psi(\theta_i) - \psi_d) \\ &= -(\psi(\theta_i) - \psi_d)^\top J_{ao} K_n^\top J_{ao}^\top (\psi(\theta_i) - \psi_d) \\ &= -\|A(\psi(\theta_i) - \psi_d)\| \leq 0 \end{aligned}$$

e é semi-definida negativa. Considerando o teorema de LaSalle (Khalil 2002) e que o único ponto de equilíbrio é $V = 0$, a estabilidade da regulação é garantida.

Com a escolha da representação da orientação através de quaternions, é sempre possível obter o valor J_a a partir de $J_r^\top J$, e como o Jacobiano geométrico pode ser obtido de forma sistemática (por exemplo através de parâmetros de Denavit-Hartenberg (Siciliano et al. 2009)), a expressão $\frac{\partial h(\theta)}{\partial \theta}$ pode sempre ser avaliada.

4.3 Controle Dinâmico de Posição

Quando a dinâmica de um manipulador não pode ser desprezada, o controle puramente cinemático não é recomendado. No trabalho de Bai & Wen (2010) que propõe a abordagem para transporte de carga aqui utilizada, considera-se este caso de dinâmica não desprezível com agentes pontuais modelados como uma massa. Nesta seção será mostrado que a lei de controle proposta em (Bai & Wen 2010), se associada a uma lei de torque computado, pode ser estendida para um manipulador com dinâmica não linear completa e com uma referência de velocidade variável, de forma dual ao caso cinemático. Como no trabalho original, será considerado somente o controle de posição, na hipótese de que a orientação é mantida por uma outra técnica.

O modelo de interação entre os manipuladores e a carga se mantém como apresentado na Seção 4.1. Entretanto, agora, o modelo de cada robô considera a sua dinâmica. Como a tarefa consiste em movimentos translacionais no espaço cartesiano os manipuladores são representados através de (2.6) considerando somente a parte do Jacobiano de posição, J_p . A equação do movimento é então dada por

$$\bar{M}_i(\theta_i)\ddot{x}_i + \bar{C}_i(\theta_i, \dot{\theta}_i)\dot{x}_i + \bar{G}_i(\theta_i) = F_i - f_i \quad (4.22)$$

com $\bar{M}_i \in \mathbb{R}^{3 \times 3}$, $\bar{C}_i \in \mathbb{R}^{3 \times 3}$, $\bar{G}_i \in \mathbb{R}^3$, $F_i \in \mathbb{R}^3$ e $f_i \in \mathbb{R}^3$ consistindo somente de forças lineares, modeladas como funções potenciais (4.3).

Assim como no caso cinemático, a redundância alcançada ao considerar somente a posição do efetuador pode ser explorada para controlar a orientação em uma tarefa secundária. Como considera-se o manipulador modelado no espaço operacional a resolução da redundância é realizada através de uma referência de força, utilizando o espaço nulo do Jacobiano transposto (Khatib 1987, Nakanishi et al. 2008). Entretanto, o controle de orientação com dinâmica não desprezível será considerado somente na próxima seção, utilizando uma abordagem em cascata.

4.3.1 Estratégia de Controle

Como observado em (4.22) o sinal de controle dinâmico (F_i) consiste em uma força no efetuador equivalente ao torque aplicado as juntas do manipulador. Para projetar esta força, considere uma lei de controle clássica por torque computado com os parâmetros do modelo totalmente conhecidos. Com as matrizes que descrevem o modelo conhecidas o projeto se separa em duas partes, um sinal de *feedforward* do modelo avaliado na trajetória desejada e um sinal de controle para seguir esta trajetória. Dado o objetivo de controle

$$\dot{x}_i(t) \rightarrow v^d(t) \quad \text{e} \quad f_i \rightarrow f_i^d$$

de rastrear uma velocidade mantendo uma força de contato constante. A seguinte lei de controle é proposta

$$F_i = \bar{M}_i\dot{v}^d + \bar{C}_iv^d + \bar{G}_i - \Gamma_i\xi_i + f_i^d. \quad (4.23)$$

Nesta, os três primeiros termos representam a compensação do modelo dinâmico e os dois últimos um termo para controle de velocidade e outro para aplicação da força. A Figura 4.4 apresenta um diagrama de blocos da estratégia de controle proposta.

Destaca-se que, a parte dos termos de *feedforward*, a lei de controle (4.23) é o dual da lei de controle do objetivo principal no caso cinemático, (4.11). Aqui, o

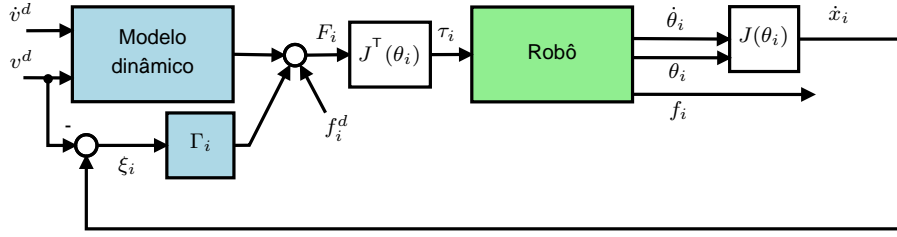


Figura 4.4: Diagrama de blocos representando o controle dinâmico.

termo proporcional é relativo a velocidade, já que se controla diretamente a força.

Observação 5. Como a força de contato é imposta diretamente sobre o efetuador, esta lei de controle dinâmico não requer a utilização de um sensor de força para realimentação. Entretanto, este fato considera que o Jacobiano do manipulador é completamente conhecido, de forma que a utilização de um sensor pode aumentar a robustez do sistema.

Levando esta lei de controle no modelo (4.22) o sistema em malha fechada é representado por

$$\dot{\xi}_i = \bar{M}_i^{-1} \left(- \left(\bar{C}_i + \Gamma_i \right) \xi_i + \nabla P(z_i^d) - \nabla P(z_i) \right) \quad (4.24)$$

(4.14) e (4.15), para os manipuladores, as forças e a carga. A estabilidade deste sistema, dentro do conceito de UB, é garantida pelo seguinte teorema.

Teorema 3. *Considere um sistema de robôs cooperativos modelados por (4.22), com parâmetros conhecidos, transportando uma carga deformável, (4.4). As forças de interação são consideradas como o gradiente de uma função potencial positiva definida da deformação da carga, submetida a condição (4.3). Suponha que os robôs são controlados de acordo com (4.23). Se a velocidade de referência $v^d(t)$ e sua derivada $\dot{v}^d(t)$ são limitadas e a força desejada f^d satisfaz (4.5), então o sistema completo em malha fechada ((4.14), (4.15) e (4.24)) na prática rastreia (i.e. é UB) $v^d(t)$ com forças de interação f_i variando de forma limitada entorno do valor desejado.*

Prova. Considerando o sistema dividido em $y_1 = [\xi_i^T \ z_i^T]^T$ e $y_2 = [\xi_c]$, o objetivo é demonstrar que o mesmo é UB com respeito a y_1 .

A seguinte função positiva definida e radialmente ilimitada

$$V(\xi_i, z_i, \xi_c) = \sum_{i=1}^N \left[P_i(z_i) - P_i(z_i^d) - (f^d)^T (z_i - z_i^d) \right] + \frac{1}{2} \sum_{i=1}^N \xi_i^T \bar{M}_i \xi_i + \frac{1}{2} \xi_c^T M_c \xi_c \quad (4.25)$$

é uma candidata a função de Lyapunov para o Teorema 1. Considerando as trajetórias do sistema e a propriedade P2 do modelo dinâmico, a derivada temporal de

$V(y_1, y_2)$ é igual a

$$\begin{aligned}
\dot{V} &= \sum_{i=1}^N [f_i^T \dot{z}_i - (f_i^d)^T \dot{z}_i^d - (f_i^d)^T \dot{z}_i + (f_i^d)^T \dot{z}_i^d] + \sum_{i=1}^N \xi_i^T \bar{M}_i \dot{\xi}_i + \frac{1}{2} \sum_{i=1}^N \xi_i^T \dot{M}_i \xi_i + \xi_c^T M_c \dot{\xi}_c \\
&= \sum_{i=1}^N (f_i - f_i^d)^T (\xi_i - \xi_c) + \sum_{i=1}^N \xi_i^T (-(\Gamma_i + \bar{C}_i) \xi_i + f_i^d - f_i + \frac{1}{2} \dot{M}_i \xi_i) + \xi_c^T \sum_{i=1}^N f_i - \xi_c^T M_c \dot{v}^d \\
&= - \sum_{i=1}^N \xi_i^T \Gamma_i \xi_i + \sum_{i=1}^N \xi_i^T (\frac{1}{2} \dot{M}_i - \bar{C}_i) \xi_i - \xi_c^T \sum_{i=1}^N f_i + \xi_c^T \sum_{i=1}^N f_i^d + \xi_c^T \sum_{i=1}^N f_i - \xi_c^T M_c \dot{v}^d \\
&= - \sum_{i=1}^N \xi_i^T \Gamma_i \xi_i - \xi_c^T M_c \dot{v}^d,
\end{aligned}$$

que é estruturalmente igual a (4.17). Assim, o restante da demonstração segue a do Teorema 2, com as constantes k_1 , k_4 e k_5 dadas por

$$k_1 = \lambda_{\max}(\Gamma_i), \quad k_4 = \frac{\|M_c\| \|\dot{v}^d\|_{\max}}{N}, \quad k_5 = \frac{\|M_c\| \|\dot{v}^d\|_{\max} \|\dot{z}_i\|_{\max}}{N}. \quad \square$$

4.4 Controle Dinâmico de Posição e Orientação

Nesta seção é considerado o controle de posição e orientação para o caso com dinâmica não desprezível. Para tal é considerada uma estratégia em cascata, a qual permite alterar a dinâmica aparente do sistema robótico para um modelo simplificado (Guenther & Hsu 1993, Lizarralde et al. 2008, Zachi, Hsu, Ortega & Lizarralde 2006).

O controle em cascata consiste de duas malhas de controle interconectadas como representado na Figura 4.5. Uma malha é responsável por resolver a dinâmica do manipulador e a outra por fornecer o sinal de controle necessário para atingir o objetivo da tarefa.

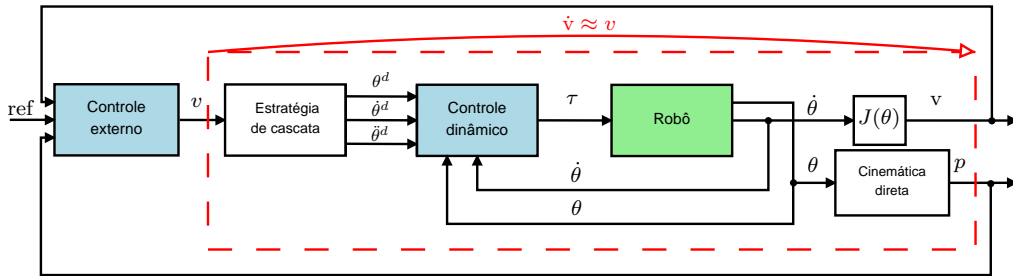


Figura 4.5: Diagrama de blocos representando a estratégia de controle em cascata.

O projeto do controlador considera então três etapas, propondo um controle dinâmico no espaço das juntas conduzido por um cinemático de segunda ordem. A utilização do espaço das juntas simplifica a implementação de estratégias de controle adaptativo. As etapas podem ser resumidas em:

S1. Projeto de uma malha interna como uma lei de controle $\tau \in \mathbb{R}^n$ descrita por

$$\tau = h(\theta, \dot{\theta}, \theta^d, \dot{\theta}^d, \ddot{\theta}^d) + J^T(\theta)f, \quad (4.26)$$

que garante o objetivo de controle definido por

$$\theta \rightarrow \theta^d(t), \quad e = \theta - \theta^d(t) \rightarrow 0,$$

onde $e \in \mathbb{R}^n$ é o erro de posição angular e $\theta^d \in \mathbb{R}^n$ denota a trajetória de referência expressa no espaço das juntas e assumida ser uniformemente limitada.

S2. Projeto de uma estratégia de cascata que conduz a malha interna através da trajetória desejada θ^d e de suas derivadas $\dot{\theta}^d, \ddot{\theta}^d$. Esta trajetória é determinada a partir de um sinal de controle externo v de tal forma que tem-se o efetuador modelado por

$$\dot{v} = v + J(\theta)L(s)e, \quad (4.27)$$

onde o segundo termo converge assintoticamente para zero, com $L(\cdot)$ um operador linear, possivelmente não-causal, e s o operador diferencial.

S3. Projeto de uma lei de controle para o novo sistema formado pelo manipulador controlado pela lei de controle dinâmico conduzida pela estratégia em cascata, de forma que esta cumpra a tarefa.

Ressalta-se que (4.27) determina um desacoplamento entre o controle de posição e de orientação do efetuador, que é evidente se a rescrevermos na forma

$$\dot{v} = \begin{bmatrix} v_p \\ v_o \end{bmatrix} + \begin{bmatrix} J_p(\theta) \\ J_o(\theta) \end{bmatrix} L(\cdot)e,$$

como subíndices p e o representando posição e orientação, respectivamente. O acoplamento existente em termos de torques nas juntas é resolvido pelo controle dinâmico em (4.26).

4.4.1 Condição de estabilidade

Considerando que nos passos acima os controladores são projetados de forma passiva. A estabilidade do controle em cascata pode ser estabelecida fazendo uso da teoria de passividade através do seguinte resultado geral para sistemas interconectados (Hsu et al. 2007)³.

³Teorema 2 no trabalho original (Hsu et al. 2007).

Teorema 4. *Considere os seguintes sistemas interconectados onde Σ_1 é conduzido por Σ_2 :*

$$\begin{aligned} \Sigma_1 : & \begin{cases} \dot{\chi}_1 = f_1(\chi, t) + g_1(\chi, t) \varphi_2, \\ \varphi_1 = h_1(\chi_1), \end{cases} \\ \Sigma_2 : & \begin{cases} \dot{\chi}_2 = f_2(\chi, t) + \nu, \\ \varphi_2 = h_2(\chi_2), \end{cases} \end{aligned} \quad (4.28)$$

sendo f_1, f_2 contínuas por partes em t e localmente Lipschitz em χ para todo $t \geq 0$ e $\chi \in \mathcal{D}$, e $\mathcal{D} \subset \mathbb{R}^n$ é um domínio que contém a origem $\chi=0$. h_1, h_2 e g_1 são contínuos.

Assumindo que $\|g_1(\chi, t)\| \leq c$, para algum $c > 0$, se o sistema Σ_1 é estritamente passivo de $\varphi_2 \mapsto \varphi_1$ com uma função de armazenamento positiva definida $V_1(\chi_1)$ tal que

$$\dot{V}_1 \leq -\lambda_1 \|\varphi_1\|^2 + c_1 \varphi_2^T \varphi_1, \quad \lambda_1 > 0 \quad (4.29)$$

e o sistema Σ_2 é estritamente passivo com relação a $\nu \mapsto \varphi_2$ com uma função positiva definida $V_2(\chi_2)$ tal que

$$\dot{V}_2 \leq -\lambda_2 \|\varphi_2\|^2 + c_2 \nu^T \varphi_2, \quad \lambda_2 > 0. \quad (4.30)$$

Então, para $\nu = 0$ as seguintes propriedades são válidas: (i) $\chi_1, \chi_2 \in \mathcal{L}_\infty$; (ii) $\lim_{t \rightarrow \infty} \varphi_1(t) = 0, \lim_{t \rightarrow \infty} \varphi_2(t) = 0$.

Prova. A demonstração pode ser encontrada no Apêndice B.3. □

4.4.2 Estratégia de controle

A primeira etapa consiste em projetar o controle dinâmico seguindo S1. Para tal, considera-se que os parâmetros que definem as matrizes M, C e g do modelo (2.5) são conhecidos de forma exata, projetando-se um controle por torque computado

$$\tau = M(\theta)\ddot{\theta}_r + C(\theta, \dot{\theta})\dot{\theta}_r + g(\theta) - K_d\sigma + u_d + J^T(\theta)f \quad (4.31)$$

no qual $K_d > 0$ é uma matriz de ganho para um controlador proporcional do erro virtual de velocidade definido por

$$\sigma := \dot{\theta} - \dot{\theta}_r = \dot{e} + \Lambda e, \quad \dot{\theta}_r := \dot{\theta}^d - \Lambda e, \quad (4.32)$$

com Λ uma matriz diagonal constante estritamente positiva. O termo u_d representa um sinal de controle externo que atua sobre o sistema em malha fechada.

A representação do sistema em malha fechada é obtida a partir de (2.5) e (4.31) como

$$M(\theta)\dot{\sigma} + (C(\theta, \dot{\theta}) + K_d)\sigma = u_d, \quad (4.33)$$

e representa um mapeamento $u_d \mapsto \sigma$ estritamente passivo. Esta condição é facilmente reconhecida considerando-se a candidata a função de Lyapunov $2V = \sigma^T M \sigma$, cuja derivada sobre as trajetórias do sistema é

$$\dot{V} = \sigma^T u_d - \sigma^T K_d \sigma.$$

Como os sinais são todos limitados, se $u_d = 0 \Rightarrow \dot{V} < 0$ e o lema de Barbalat (Khalil 2002) garante que $\lim_{t \rightarrow \infty} \sigma(t) = 0$. Ressalta-se que pelas definições de σ e $\dot{\theta}_r$ (em (4.32)) tem-se $\sigma = \dot{e} + \Lambda e$ e, portanto, $\lim_{t \rightarrow \infty} \dot{e}(t) = 0$ e $\lim_{t \rightarrow \infty} e(t) = 0$.

Seguindo os passos para o projeto do controle em cascata, na segunda etapa (S2) projeta-se a estratégia que determina as referências para a malha interna de controle dinâmico. Como pretende-se considerar um modelo de segunda ordem para sistema aparente entre a entrada dessa estratégia de cascata e a posição do efetuador, o sinal $\ddot{\theta}_r$ será projetado diretamente, obtendo os demais por meio integração. Para isso, considera-se a derivada temporal da relação de cinemática diferencial (2.4)

$$\dot{v} = J\ddot{\theta} + \dot{J}\dot{\theta}$$

levando ao sinal de referência

$$\ddot{\theta}_r = J^\dagger(v - \dot{J}\dot{\theta} + J\Lambda\sigma), \quad (4.34)$$

que leva o sistema formado pelo robô e a malha de controle dinâmico a seguir a referência cinemática v .

Por fim, segue-se o terceiro passo do projeto em cascata (S3) projetando o sinal v . Este consiste das acelerações linear e angular do efetuador do robô. Como ambas estão desacopladas cada uma será considerada separadamente.

O controle de posição segue a estrutura utilizada no caso de primeira ordem, de um termo *feedforward* mais um termo proporcional, desta vez contando com o erro de força, levando a

$$v_p = \dot{v}_i^d - K_v \xi_i + (f^d - f_i), \quad (4.35)$$

no qual $K_v > 0$ e $\xi_i = \dot{x}_i - v^d$. Substituindo (4.35) na parte referente a posição do modelo de (4.27), lembrando que o segundo termo decai assintoticamente a zero, a malha fechada é dada por

$$\dot{\xi}_i = -K_v \xi_i + f^d - f_i, \quad (4.36)$$

que, assim como no caso anterior de primeira ordem, é UB em relação a ξ_i e z_i , como mostrado no teorema a seguir.

Teorema 5. *Considere um sistema de robôs cooperativos com cada um dos efetadores modelados como um sistema de segunda ordem transportando uma carga deformável modelada por (4.4). As forças de interação são consideradas como o gradiente de uma função potencial positiva definida da deformação da carga, submetida a condição (4.3). Suponha que os robôs são controlados de acordo com (4.35). Se a velocidade de referência $v^d(t)$ e sua derivada $\dot{v}^d(t)$ são limitadas e a força desejada f^d satisfaz (4.5), então, para o caso de controle de posição, o sistema completo em malha fechada ((4.14), (4.15) e (4.36)) na prática rastreia (i.e. é UB) $v^d(t)$ com forças de interação f_i variando de forma limitada.*

Prova. Considere o sistema completo dividido em $y_1 = [\xi_i^T \ z_i^T]^T$ e $y_2 = [\xi_c]$, com a candidata a Lyapunov

$$V_p(\xi_i, z_i, \xi_c) = \sum_{i=1}^N [P_i(z_i) - P_i(z_i^d) - (f^d)^T(z_i - z_i^d)] + \frac{1}{2} \sum_{i=1}^N \xi_i^T \xi_i + \frac{1}{2} \xi_c^T M_c \xi_c \quad (4.37)$$

positiva definida e radialmente ilimitada. A derivada temporal de V_p ao longo das trajetórias do sistema é dada por

$$\begin{aligned} \dot{V}_p &= \sum_{i=1}^N [f^T \dot{z}_i - (f^d)^T \dot{z}_i^d - (f^d)^T \dot{z}_i + (f^d)^T \dot{z}_i^d] + \sum_{i=1}^N \xi_i^T \dot{\xi}_i + \xi_c^T M_c \dot{\xi}_c \\ &= \sum_{i=1}^N [(f_i - f^d)^T (\xi_i - \xi_c) + \xi_i^T (K_v \xi_i + f^d - f_i)] + \xi_c^T \sum_{i=1}^N f_i - \xi_c^T M_c \dot{v}^d \\ &= \sum_{i=1}^N [-(f^d - f_i)^T \xi_i + \xi_i^T (f^d - f_i) - \xi_i^T K_v \xi_i] - \xi_c^T M_c \dot{v}^d \\ &= - \sum_{i=1}^N \xi_i^T K_v \xi_i - \xi_c^T M_c \dot{v}^d \end{aligned} \quad (4.38)$$

que é semelhante a (4.17). Com isso o resto da prova segue a demonstração do Teorema 2. \square

Note que a parte relativa a orientação no modelo em (4.27) considera a aceleração angular $\dot{\omega}$, enquanto a representação escolhida para cinemática direta é a de quaternions. A compatibilização entre as dimensões destes dois sinais para estratégias de comando por aceleração é conhecida, considerando somente a parte vetorial do erro de quaternion (Caccavale et al. 1998, Nakanishi et al. 2007, Yuan 1988). Assim, a lei para regulação da orientação é dada por

$$a_o = -K_{vo}\omega - K_{po}e_{qv} \quad (4.39)$$

na qual $K_{vo} > 0$ e $K_{po} = k_{po}I > 0$ são os ganhos de velocidade e posição, e o termo $e_{qv} = (q^d * q^{-1})_v$ é a parte vetorial do erro de quaternion, com q^d representando a orientação desejada e o operador $*$ o produto de quaternions. Ressalta-se que as velocidades linear e angular do efetuador utilizadas nas leis de controle (4.35) e (4.39) podem ser obtidas através da cinemática diferencial, assumindo-se a medição da velocidade das juntas.

Teorema 6. *Considere o movimento do efetuador de um manipulador modelado como um sistema de segunda ordem. A orientação do mesmo pode ser regulada em uma configuração fixa de maneira assintoticamente estável através da lei de controle (4.39).*

Prova. Para demonstrar a estabilidade de (4.39), considere a seguinte candidata a função de Lyapunov

$$V_o = k_{po} [(e_{qe} - 1)^2 + e_{qv}^T e_{qv}] + \frac{1}{2} \omega^T \omega. \quad (4.40)$$

Dado que, com uma velocidade angular ω , a derivada temporal da orientação é obtida através da chamada propagação do quaternion (Yuan 1988)

$$\dot{q} = \begin{bmatrix} \dot{q}_e \\ \dot{q}_v \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 0 & -\omega^T \\ \omega & -\omega \times \end{bmatrix} \begin{bmatrix} q_e \\ q_v \end{bmatrix},$$

a derivada temporal de V_o ao longo das trajetórias do sistema é

$$\begin{aligned} \dot{V}_o &= k_{po} [2(e_{qe} - 1)\dot{e}_{qe} + 2e_{qv}^T \dot{e}_{qv}] + \omega^T \dot{\omega} = \\ &= k_{po} [-(e_{qe} - 1)\omega^T e_{qv} + e_{qv}^T (\omega e_{qe} - \omega \times e_{qv})] + \omega^T (-K_{vo}\omega - K_{po}e_{qv}) = \\ &= -\omega^T K_{vo}\omega \leq 0. \end{aligned} \quad (4.41)$$

Como \dot{V}_o é negativa semi-definida, a estabilidade deve ser analisada com base no teorema de LaSalle (Khalil 2002). Existem dois conjuntos invariantes para este sistema nos quais $\omega = 0$, $e_{qv} = 0$ e $e_{qe} = \pm 1$. O caso em que $e_{qe} = -1$ o equilíbrio é instável (Caccavale et al. 1998), de forma que o objetivo de controle $q \rightarrow q^d$ (equivalentemente $e_q \rightarrow [1 \ 0 \ 0 \ 0]^T$) é atingido assintoticamente. \square

A estratégia em cascata em cada manipulador para o caso do controle cooperativo descentralizado é então representada no diagrama de blocos da Figura 4.6.

Para a analisar a estabilidade da estratégia de cascata aplica-se o Teorema 4, como a malha dinâmica é passiva, é necessário estabelecer a passividade da malha cinemática. Considere, então, o seguinte sistema conduzido por um sinal fictício

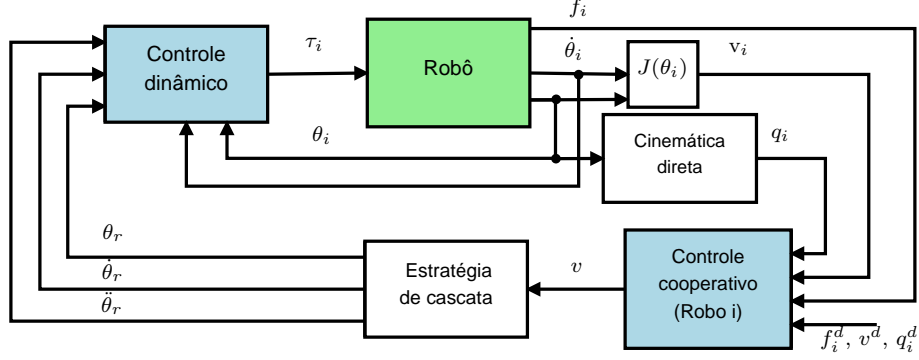


Figura 4.6: Diagrama de blocos representando a proposta de controle em cascata em um manipulador do sistema cooperativo.

$$v = \begin{bmatrix} v_v & v_o \end{bmatrix}^T$$

$$\dot{\xi}_i = -K_v \xi_i + f^d - f_i + v_p \quad (4.42)$$

$$\dot{e}_o = \dot{\omega} = -K_{vo} \omega - K_{po} e_{qv} + v_o \quad (4.43)$$

sendo $v_p = J_p \sigma$ e $v_o = J_o \sigma$.

Corolário 1. Os erros definidos por (4.42) e (4.43) definem um mapeamento localmente passivo de saída $\sigma \mapsto \sum_{i=1}^N J_p^T \xi_i + J_o^T \omega$ com função de armazenagem positiva definida $V = V_p + V_o$ (em (4.37) e (4.40)).

Prova. A partir de (4.38) e (4.41) é fácil perceber que

$$\dot{V} = \dot{V}_p + \dot{V}_o \leq \sum_{i=1}^N -k_1 \|y_1\|^2 + k_4 \|y_1\| + k_5 - \omega^T K_{vo} \omega + \sum_{i=1}^N \xi_i^T v_p + \omega^T v_o,$$

na qual a parte independente de v é, ao menos, localmente negativa semi-definida segundo os Teoremas 5 e 6. A passividade do mapeamento $\sigma \mapsto \sum_{i=1}^N J_p^T \xi_i + J_o^T \omega$ decorre diretamente da definição de passividade (Khalil 2002). \square

O sistema interconectado, considerando as malhas de controle dinâmico e cinemático, é representado pelas variáveis de estado

$$\begin{aligned} \chi_1 &= \begin{bmatrix} \xi_i^T & \xi_c^T & z_i^T & e_{qv}^T \end{bmatrix}^T & \varphi_1 &= J_p^T \xi_i + J_o^T \omega \\ \chi_2 &= \begin{bmatrix} e^T & \dot{e}^T \end{bmatrix}^T & \varphi_2 &= \sigma \end{aligned}$$

cuja estabilidade decorre diretamente do Corolário 1 e do Teorema 4 e da passividade do controle por torque computado.

Destaca-se que devido as propriedades de passividade, a lei de controle por torque computado (4.31) pode ser substituída por uma estratégia adaptativa ou adaptativa robusta, como as propostas por Slotine & Li (1991). Assim, a hipótese de parâmetros

conhecidos de forma exata pode ser flexibilizada, já que esta não é satisfeita em aplicações práticas.

Neste capítulo foram apresentadas três leis de controle descentralizadas para manipuladores robóticos transportando uma carga de forma cooperativa. São abordados os casos de robôs com dinâmica desprezível e não desprezível. Para o primeiro caso utilizou-se o controle de posição e orientação separados em duas tarefas. A principal considera um controle de velocidade e força para transladar a carga, enquanto a secundária é utilizada para que os efetuadores do robôs permaneçam perpendiculares a esta.

No caso com dinâmica não desprezível, mostrou-se primeiramente que a lei de controle proposta por (Bai & Wen 2010) para o transporte de carga sem rotações, quando associada a uma lei semelhante a torque computado, pode ser estendida para manipuladores com dinâmica completa e rastreando velocidades variáveis com o tempo. Essa lei é dual a tarefa principal do caso cinemático e, a princípio, não necessita de um sensor de força no efetuator. Entretanto, é necessário utilizar a aceleração de referência na compensação da dinâmica do sistema e, como o operador determina a trajetória durante a execução, esta informação tem que ser obtida de forma numérica.

Por fim considerou-se o problema de controle de orientação e posição do efetuator para o caso de dinâmica não desprezível. Para isso, foi utilizada uma estratégia considerando uma malha cinemática e uma dinâmica em cascata. Os controles de posição e orientação podem ser projetados de forma independente, sem que se considere nenhuma redundância. Novamente, entretanto, utiliza-se a aceleração de referência.

A utilização de controle de velocidade pode gerar diferenças na posição entre o mestre e o escravo (Hokayem & Spong 2006). Entretanto, o problema abordado não se propõem a este sincronismo, principalmente por considerar como mestres dispositivos com configuração distinta da dos escravos e a presença de realimentação visual. O objetivo da tarefa aqui exposta é posicionar a carga, assim a visão do ambiente local permite que o operador corrija os erros de forma gradual.

Capítulo 5

Resultados Experimentais

Para validar a nova proposta de estratégia para teleoperação cooperativa em relação a estabilidade garantida pelos tanques de energia e a coordenação entre os escravos, são apresentados resultados de simulação da mesma. Na simulação considera-se o uso de manipuladores planares com 3 DoF, utilizando-se controle cinemático ou em cascata para os robôs escravos. O caso cinemático também é implementado em um robô industrial Motoman DIA10 que possui dois manipuladores com 7 DoF cada. A implementação experimental considera o controle cooperativo local de trajetórias pré-estabelecidas, sem a ação de um operador, e devido a limitações de dispositivos o caso de controle unilateral com somente um mestre. O mestre pode ser um dispositivo háptico Phantom Omni (Sensable, EUA) ou um dispositivo de interface natural Kinect (Microsoft Corp., EUA).

5.1 Simulação

A verificação da eficácia da arquitetura de controle proposta foi realizada através de simulação da teleoperação bilateral. As simulações consideraram manipuladores planares em formações com dois ou quatro agentes.

5.1.1 Modelagem

A presente proposta de solução para o problema de teleoperação cooperativa foi simulada utilizando Simulink e considerando mestres e escravos planares com 3 juntas de revolução, como na Figura 5.1. Os mestres são modelados como manipuladores com dinâmica completa

$$M(\theta)\ddot{\theta} + C(\theta, \dot{\theta})\dot{\theta} + G(\theta) = J^T F$$

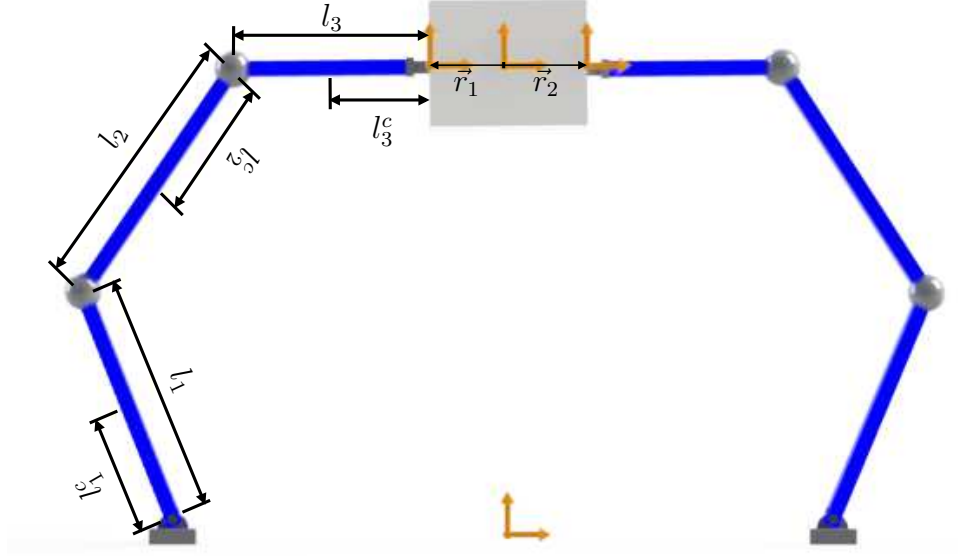


Figura 5.1: Representação do sistema escravo simulado ($N=2$), indicando os parâmetros do modelo.

atuados através da força F . Seguindo a modelagem de (Aldana et al. 2012) e utilizando os parâmetros da Figura 5.1 as matrizes M e C são definidas em função das variáveis auxiliares

$$\begin{aligned}
 p_1 &= m_1 l_1^c l_1^c + I_1 + m_2 l_1 l_1 + m_2 l_2^c l_2^c + I_2 + m_3 l_1 l_1 + m_3 l_2 l_2 + m_3 l_3^c l_3^c + I_3 \\
 p_2 &= m_2 l_1 l_2^c + m_3 l_1 l_2 & p_3 &= m_3 l_2 l_3^c & p_4 &= m_3 l_1 l_3^c \\
 p_5 &= m_2 l_2^c l_2^c + I_2 + m_3 l_2 l_2 + m_3 l_3^c l_3^c + I_3 & p_6 &= m_3 l_3^c l_3^c + I_3
 \end{aligned}$$

Sendo a matriz de inércia simétrica

$$M = \begin{bmatrix} p_1 + 2p_2 c\theta_2 + 2p_3 c\theta_3 + 2p_4 c\theta_{23} & * & * \\ p_5 + p_2 c\theta_2 + 2p_3 c\theta_3 + p_4 c\theta_{23} & p_5 + 2p_3 c\theta_3 & * \\ p_6 + p_3 c\theta_3 + p_4 c\theta_{23} & p_6 + p_3 c\theta_3 & p_6 \end{bmatrix} \quad (5.1)$$

na qual $c\theta_{ijk}$ e $s\theta_{ijk}$ representam $\cos(\theta_i + \theta_j + \theta_k)$ e $\sin(\theta_i + \theta_j + \theta_k)$ respectivamente. A matriz de forças de Coriolis e centrífugas é

$$C = \begin{bmatrix} \bar{c}_{11} & \bar{c}_{12} & \bar{c}_{13} \\ p_2 s\theta_2 \dot{\theta}_1 - p_3 s\theta_3 \dot{\theta}_3 + p_4 s\theta_{23} \dot{\theta}_1 & -p_3 s\theta_3 \dot{\theta}_3 & -p_3 s\theta_3 (\dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3) \\ p_3 s\theta_3 (\dot{\theta}_1 + \dot{\theta}_2) + p_4 s\theta_{23} \dot{\theta}_1 & p_3 s\theta_3 (\dot{\theta}_1 + \dot{\theta}_2) & 0 \end{bmatrix} \quad (5.2)$$

com os termos \bar{c}_{1j} definidos por

$$\begin{aligned}
 \bar{c}_{11} &= -p_2 s\theta_2 \dot{\theta}_2 - p_3 s\theta_3 \dot{\theta}_3 - p_4 s\theta_{23} (\dot{\theta}_2 + \dot{\theta}_3) \\
 \bar{c}_{12} &= -p_2 s\theta_2 (\dot{\theta}_1 + \dot{\theta}_2) - p_3 s\theta_3 \dot{\theta}_3 - p_4 s\theta_{23} (\dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3) \\
 \bar{c}_{13} &= -(p_3 s\theta_3 + p_4 s\theta_{23}) (\dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3) .
 \end{aligned}$$

O vetor de forças referentes a gravidade é dado por

$$G = \begin{bmatrix} (m_2gl_2^c + m_3gl_2)c\theta_1 + (m_2gl_2^c + m_3gl_2)c\theta_{12} + m_3gl_3^c c\theta_{123} \\ (m_2gl_2^c + m_3gl_2)c\theta_{12} + p_9c\theta_{123} \\ m_3gl_3^c c\theta_{123} \end{bmatrix}, \quad (5.3)$$

sendo g a magnitude da aceleração da gravidade no ambiente local, considerando esta atuando ao longo da direção y .

Os escravos, por sua vez, são manipuladores cinemáticos representados pelo Jacobiano de posição

$$J_p = \begin{bmatrix} -l_1s\theta_1 - l_2s\theta_{12} - l_3s\theta_{123} & -l_2s\theta_{12} - l_3s\theta_{123} & -l_3s\theta_{123} \\ l_1c\theta_1 + l_2c\theta_{12} + l_3c\theta_{123} & l_2c\theta_{12} + l_3c\theta_{123} & l_3c\theta_{123} \end{bmatrix} \quad (5.4)$$

como no modelo (4.6) quando considerado o caso de dinâmica desprezível. Para manipuladores com dinâmica não desprezível, utiliza-se como modelo a dinâmica do erro virtual de velocidade em (4.33).

Para a simulação considera-se que os mestres e os escravos tem as mesmas dimensões, de acordo com os parâmetros da Tabela 5.1.

Tabela 5.1: Parâmetros do modelo de manipulador 3 *DoF* utilizado na simulação.

Parâmetro	Símbolo	Elo (k)		
		1	2	3
Massa	m_k	2	2	1.5
Momento de inércia	I_k	0.5	0.5	0.25
Comprimento	l_k	0.8	0.8	0.65

Os pacotes referentes as camadas de transparência e de passividade são transmitidos pelo mesmo canal de comunicação, sem perdas e com atraso variável distinto em cada sentido. Os manipuladores mestre rastreiam uma referência de velocidade Cartesiana através de um controlador proporcional e integral. A carga transportada possui uma rigidez de $500 N/m$, o ganho de controle é $K_i = 0.02$ e $K_v = ?$ para os todos os escravos e a referência de força para pegada é $f^d = [0 \ \pm 20]^T$ ou $f^d = [\pm 20 \ 0]^T$, de acordo com a orientação do efetuador.

Os parâmetros utilizados para implementar os tanques de energia são apresentados na Tabela 5.2.

Tabela 5.2: Parâmetros para simulação dos tanques de energia.

β_m	0.05	H_d	0.05
α	100	β_s	0.05
k_{ss}	5		

5.1.2 Resultados

Os resultados serão apresentados para a simulação da teleoperação com $N = 2$. A escolha por este caso decorre, exclusivamente, do menor número de dados a serem apresentados, já que o desempenho é semelhante para quatro ou dois agentes, como apresentado na Figura 5.2. A numeração dos robôs segue a ordem estabelecida na Figura 3.3.

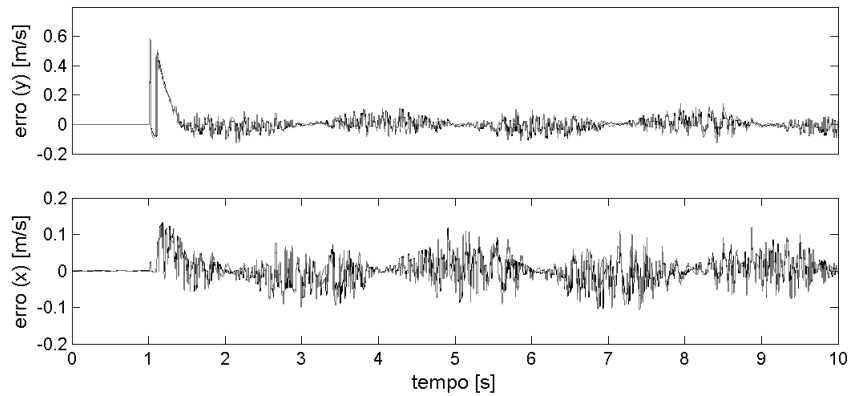


Figura 5.2: Erro de velocidade da carga durante simulação de teleoperação com movimento circular, considerando $N = 2$ e $N = 4$. Cinza - $N = 4$; preto - $N = 2$.

A influencia da saturação do sinal de controle na camada de passividade pode ser claramente observada no caso em que se considera uma função da energia atual no tanque para limitar os sinais de comando. Os manipuladores não seguem diretamente a referência até existir um nível de energia adequado, como observado na Figura 5.3 que representado erro de velocidade para os manipuladores, com e sem a presença da saturação. Os dois casos convergem para o mesmo erro variável, presente devido a velocidade não constante e o acoplamento do controle de força.

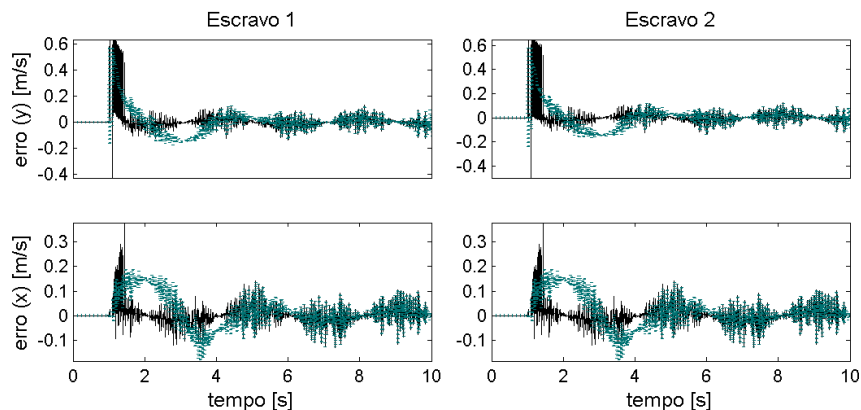


Figura 5.3: Erro de velocidade durante simulação de teleoperação com movimento circular. Verde tracejado - presença de saturação considerando o nível de energia atual do tanque; preto contínuo - sem a saturação pelo nível de energia.

A deformação na trajetória dos manipuladores escravo é melhor apresentada na Figura 5.4, onde se observam retratos da configuração dos robôs mestre e escravo em diferentes instantes de tempo.

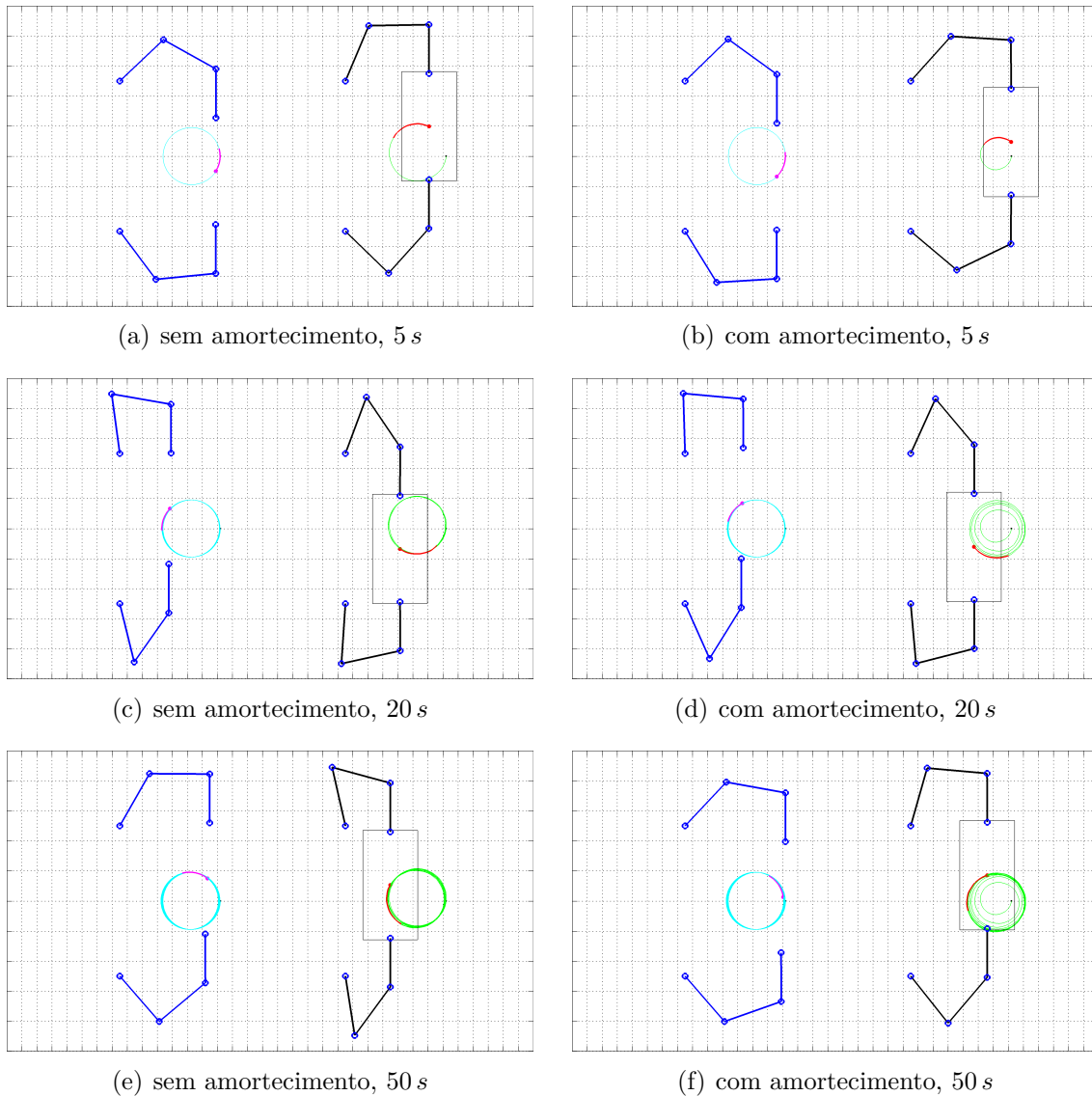


Figura 5.4: Representação dos robôs mestre e escravo em diferentes instantes de tempo com e sem presença de amortecimento pelo nível de energia. As linhas azuis representam os mestres. A trajetória dos escravos sofre um deslocamento ao se utilizar o amortecimento, atingindo o mesmo raio centrado em outro ponto.

A Figura 5.5 apresenta as magnitudes da força de pegada durante a simulação de movimento circular. Nesta, é claro que a pegada é mantida, com ambos os manipuladores mantendo a força mínima e rastreando a força do usuário. Nota-se que como demonstrado no Teorema 2 o controle de pegada mantém uma força limitada, a menos que a velocidade de referência seja constante, levando a um erro de força nulo.

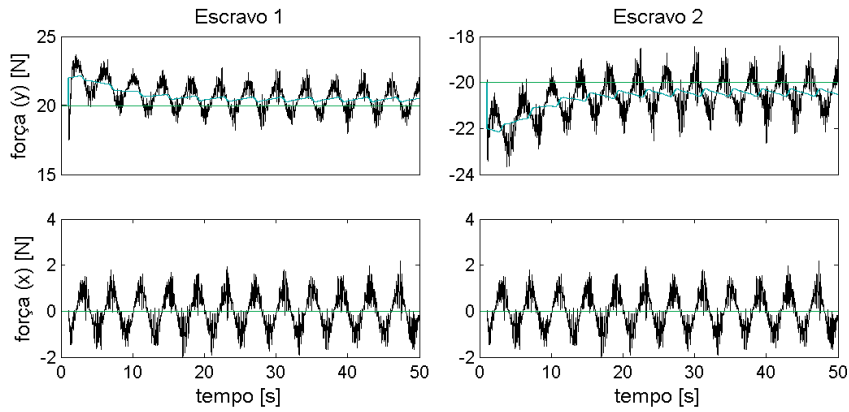


Figura 5.5: Magnitude das forças desejada e aplicada a carga, por cada robô, durante movimento circular. Preto - força medida pelo robô; Verde - força de referência de pegada; e Azul - referência incluindo a força do usuário.

O caso em que o sistema cooperativo se move com velocidade constante foi simulado considerando o deslocamento na direção \vec{x} . O erro de força correspondente converge para zero como mostrado na Figura 5.6, mesmo com a presença de uma força variável acima do nível mínimo exercida pelo operador.

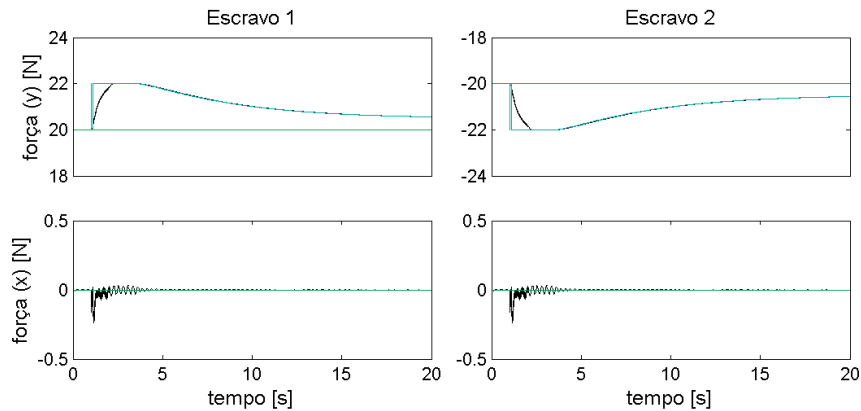


Figura 5.6: Magnitude das forças desejada e aplicada a carga, por cada robô, durante movimento com velocidade constante. Preto - força medida pelo robô; Verde - força de referência de pegada; e Azul - referência incluindo a força do usuário.

A Figura 5.7 apresenta os resultados do controle da tarefa secundária, através da orientação de cada um dos manipuladores. Nota-se que estas permanecem próximas a $\pi/2$, mantendo assim os efetuadores perpendiculares a carga, como visto nos instantâneos da Figura 5.4.

A variação em alta frequência observada nos resultados de simulação apresentados é explicada pela implementação de um canal de comunicação com atraso variável. Esse atraso é considerado como uma distribuição uniforme em torno de um valor médio com frequência de 10 Hz .

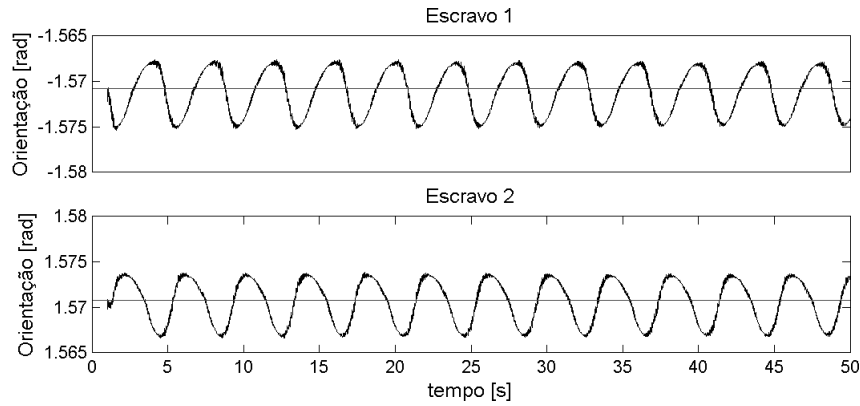


Figura 5.7: Orientação dos manipuladores escravo durante a simulação de teleoperação com movimento circular. Preto: orientação calculada; e cinza escuro: referência.

Os resultados obtidos com a estratégia em cascata são semelhantes ao caso em que a dinâmica do manipulador é desprezível. A Figura 5.8 apresenta o erro de rastreamento de velocidade, este é limitado com amplitude da mesma ordem de grandeza para o controle cinemático. Destaca-se que por considerar atraso constante a curva é suave.

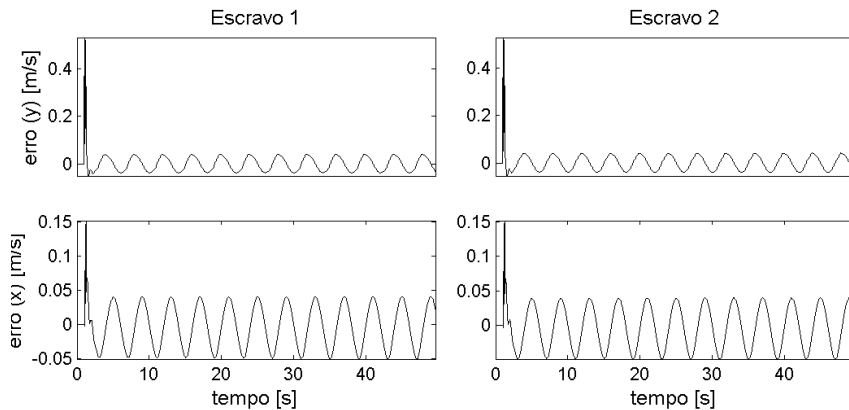


Figura 5.8: Erro de velocidade durante simulação de teleoperação com movimento circular e controle em cascata.

Para o controle de força a Figura 5.9 mostra que a amplitude da oscilação sobre o valor desejado é maior do que a do caso de controle cinemático. Isso indica a presença de uma maior influência do controle de velocidade sobre a regulação de força, que pode ser explicado pelo ganho unitário neste segundo controle.

A principal diferença entre a resposta do sistema com controle em cascata e apenas cinemático está na orientação do manipulador. De maneira diferente ao apresentado anteriormente na Figura 5.10, com a estratégia de cascata a orientação não oscila em torno do valor desejado, dado que não é mais um objetivo secundário. Nota-se que ao final da simulação esta orientação tem uma tendência a divergir do valor de referência. Isto pode ser explicado como uma questão de precisão do Matlab

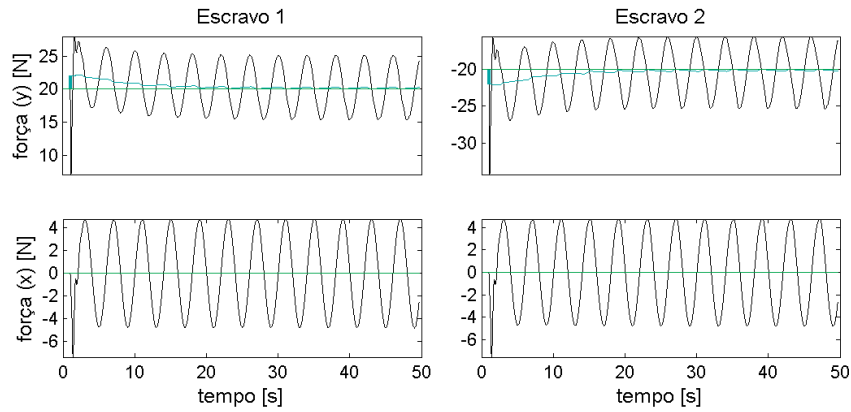


Figura 5.9: Magnitude das forças desejada e aplicada a carga, por cada robô, durante movimento circular e controle em cascata. Preto - força medida pelo robô; Verde - força de referência de pegada; e Azul - referência incluindo a força do usuário.

ou, mais provavelmente, pelo *drift* de posição devido ao controle de velocidade. Este *drift* é claro ao observar-se os instantâneos da Figura 5.11.

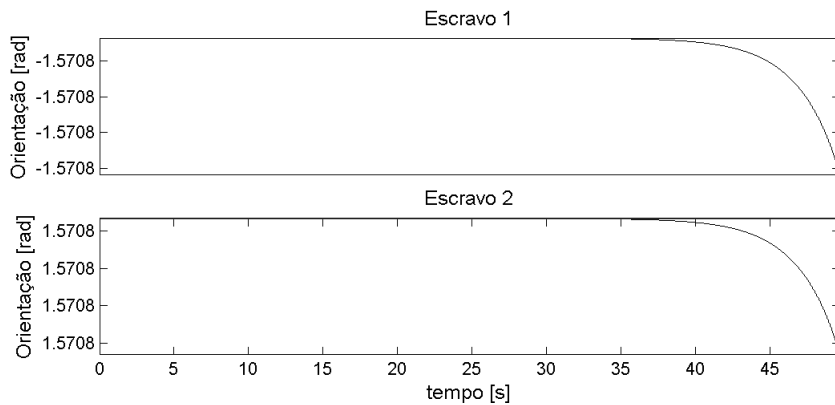


Figura 5.10: Orientação dos manipuladores escravo durante a simulação de teleoperação com movimento circular e controle em cascata. Preto: orientação calculada; e cinza escuro: referência.

Na Figura 5.11 são apresentados os mestres e os escravos em diferentes instantes da simulação. Nota-se que devido ao atraso de 1 s os escravos permanecem na mesma posição na primeira e segunda imagens. Além disso, o *drift* na posição é observado ao longo da trajetória do centro da carga, representada pela linha verde.

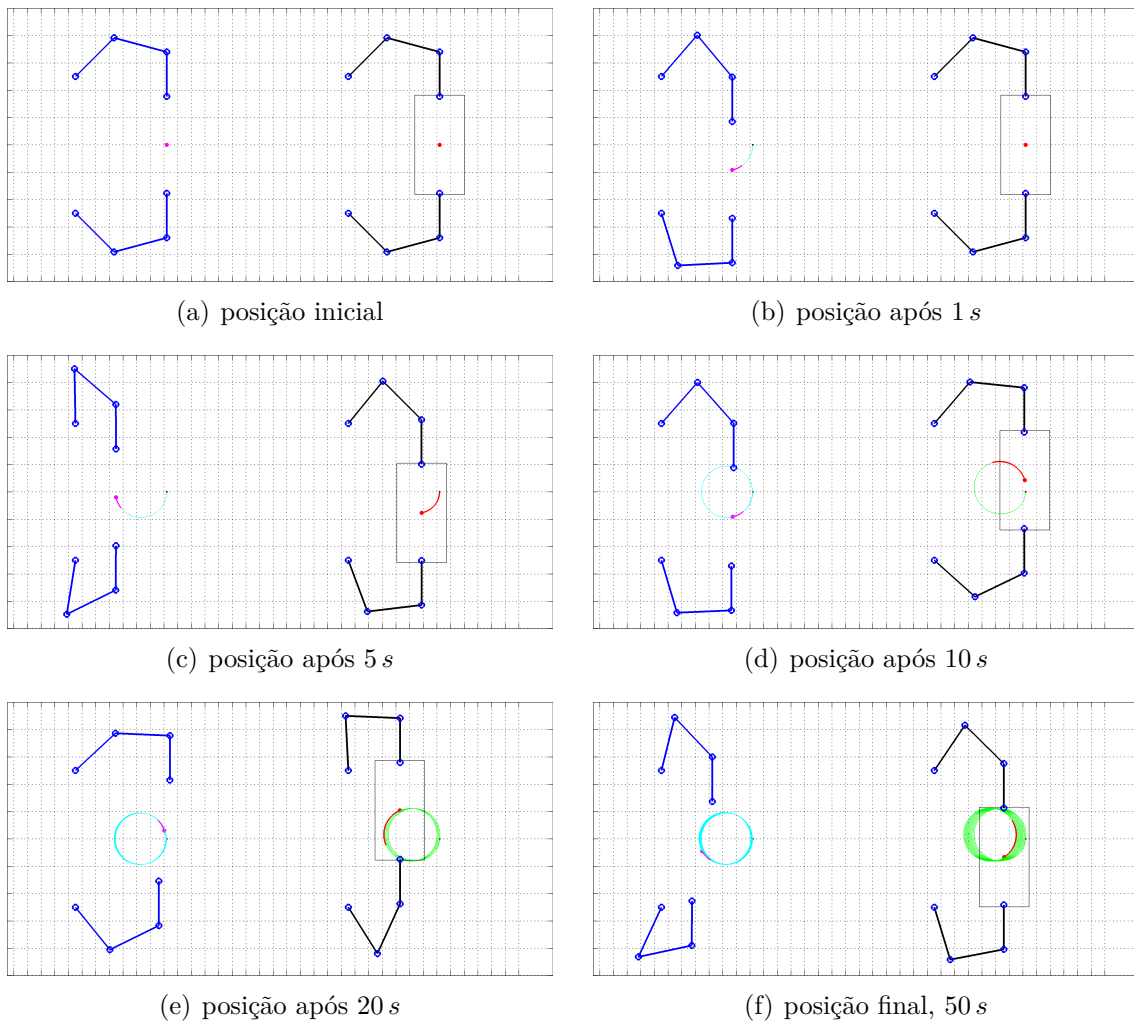


Figura 5.11: Representação dos robôs mestre e escravo em diferentes instantes de tempo para a simulação de controle em cascata. As linhas azuis representam os mestres. A trajetória da carga no ambiente local, linha verde, sofre um deslocamento devido ao controle somente de velocidade.

5.2 Testes Experimentais

Os testes experimentais foram realizados utilizando a estrutura presente no LEAD¹ e a arquitetura apresentada no Apêndice A. Foram considerados o controle autônomo de cooperação e a teleoperação unilateral, utilizando um único mestre que pode ser um dispositivo háptico Phantom Omni ou um Kinect.

5.2.1 Montagem experimental

O robô utilizado para os experimentos é um Motoman DIA10 com dois braços (7 DoF cada), como na Figura 5.12, controlado em baixo nível por um controlador industrial NX100. As referências para este são geradas em um computador pessoal conectado com o NX100 através de um controlador de alta velocidade Motoma HSC.

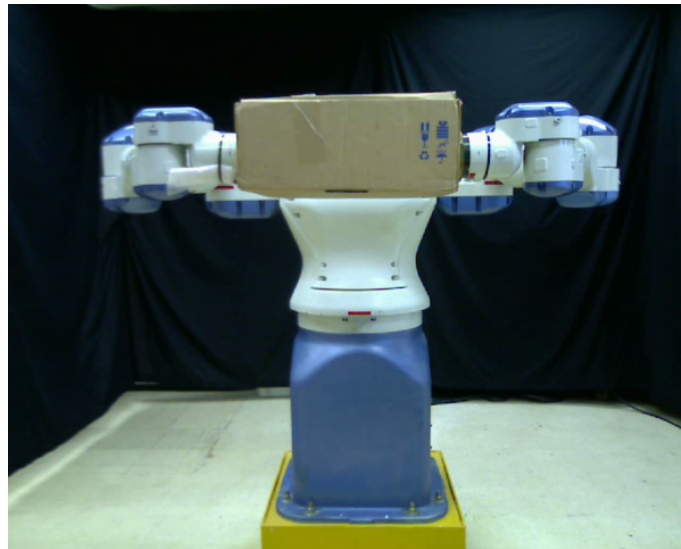


Figura 5.12: Motoman DIA10 segurando uma caixa.

Para acessar o HSC a partir do Matlab, onde é implementada a malha de controle de cooperação, é utilizado o RR (ver Apêndice A.1.1) (Wason & Wen 2011). O serviço projetado para esta comunicação fornece informações sobre as posições e torques das juntas, além das medidas de força virtuais. Como entrada, o serviço recebe posições de referência para juntas e implementa um malha de controle de posição intermediária, a malha de deslocamento relativo do controlador industrial. Esta malha é projetada em Simulink, sendo então compilada para rodar em um processo exclusivo com amostragem de 2 ms , seguindo o HSC. O Apêndice C apresenta uma descrição mais detalhada deste sistema.

Como o robô não possui sensor de força no efetuador, um objeto virtual com rigidez 500 N/m e um sensor de força ideal foram implementados na malha de con-

¹Laboratório de Controle e Automação, Engenharia de Aplicação e Desenvolvimento

trole intermediário. O acesso à medida de força é também realizado através do RR, de maneira similar a um sensor real.

Foram realizados dois grupos de testes utilizando o Motoman: controle autônomo e teleoperação. No primeiro é considerado somente o controle cooperativo rastreando trajetórias pré-definidas sem a presença de um operador na malha. No segundo considera-se a teleoperação unilateral com a arquitetura de controle completa. A seguir cada um dos testes é descrito, sendo apresentados seus respectivos resultados.

5.2.2 Operação autônoma

Para os testes do controle de cooperação, foram utilizadas duas modalidades de trajetória de referência, planar e tridimensional. As trajetórias foram determinadas de forma a melhor aproveitar o espaço de trabalho considerando as restrições de pegada, distância entre os efetuadores e orientação. As expressões das curvas discretas utilizadas em cada teste são apresentadas na Tabela 5.3.

Tabela 5.3: Trajetórias para os testes experimentais do controle de manipuladores cooperativos.

Teste	Plano	Velocidade de referência		
		x	y	z
1	xy	-0,02	0,02	0
2	xz	-0,02	0	0,02
3	yz	0	-0,02	0,02
4	xy	$-0,08 \sin\left(\frac{2\pi}{600}k\right)$	$0,08 \cos\left(\frac{2\pi}{600}k\right)$	0
5	xz	$-0,08 \sin\left(\frac{2\pi}{600}k\right)$	0	$0,08 \cos\left(\frac{2\pi}{600}k\right)$
6	yz	0	$0,08 \cos\left(\frac{2\pi}{600}k\right)$	$-0,08 \sin\left(\frac{2\pi}{600}k\right)$
7	—	$-0,08 \sin\left(\frac{2\pi}{600}k\right)$	$0,08 \cos\left(\frac{2\pi}{600}k\right)$	-0,02

Com as trajetórias planares pode-se observar o comportamento do sistema com referências de velocidade constante e variáveis. Na Figura 5.13 são apresentados os erros de velocidade para ambos os casos. A velocidade de referência é obtida pela lei de controle (4.11) utilizando (4.12) e (4.20), enquanto a velocidade real dos efetuadores é estimada através do modelo cinemático diferencial e da medida de velocidade das juntas a cada instante de amostragem do controlador. Por simplicidade somente o escravo 1 é mostrado, sendo o comportamento do escravo 2 semelhante.

Como esperado o erro de velocidade com referência constante é menor que durante o rastreamento de referência circular. Nota-se, ainda, que no primeiro caso o erro é constante, porém não nulo como esperado a partir do desenvolvimento teórico. Isto pode ser explicado por pequenas diferenças no modelo cinemático, a presença

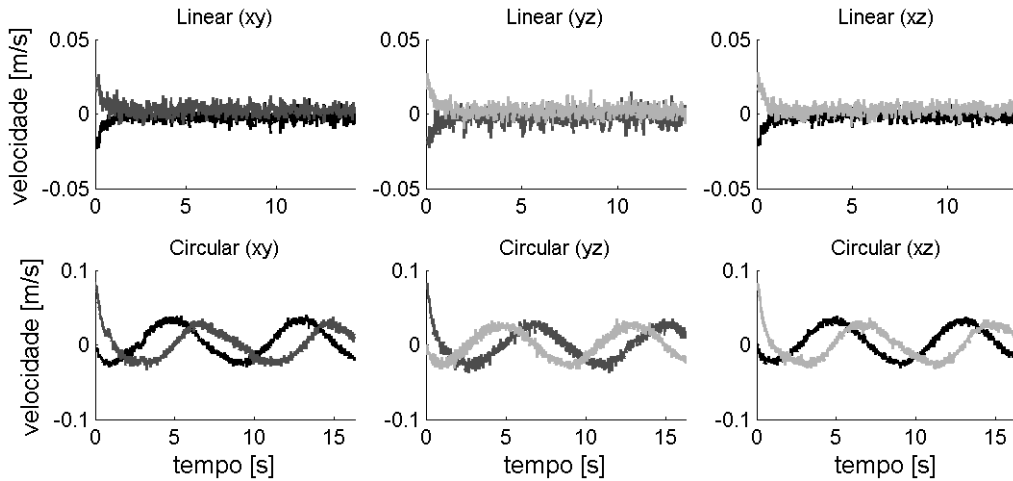


Figura 5.13: Erros de velocidade do efetuador do escravo 1 durante o controle autônomo com referência linear e circular. Preto: eixo x ; cinza escuro: eixo y ; e cinza claro: eixo z .

de atraso entre o comando e o movimento real das juntas e a utilização de um controle proporcional para a malha intermediária de controle de posição das juntas (ver Apêndice C).

Por se tratar de um problema de transporte de carga cooperativo, além do seguimento de trajetória deve-se garantir a força de pegada do objeto. A magnitude desta é mostrada na Figura 5.14 para cada um dos testes. Destaca-se que apesar do erro de regulação não houve perda de contato.

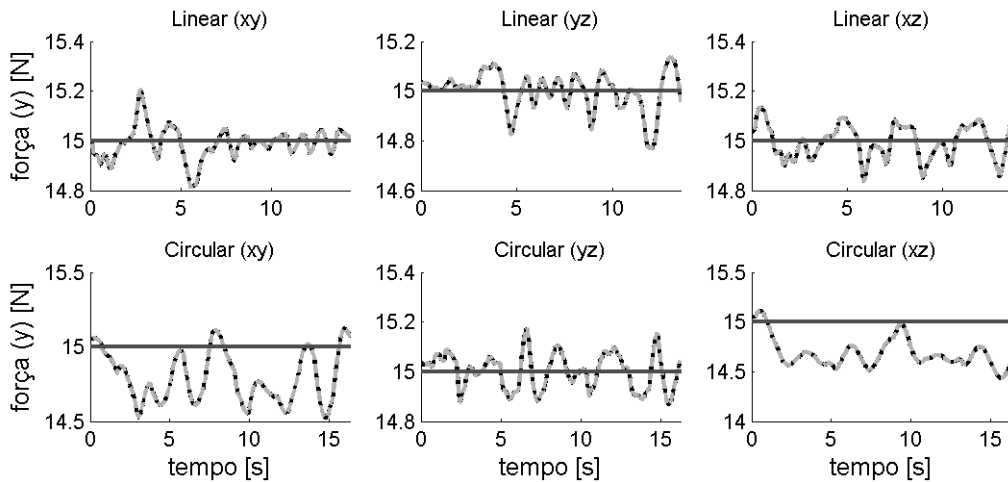


Figura 5.14: Módulo das força virtuais nos efetuadores durante o controle autônomo com referência linear e circular. Preto: escravo 1; cinza escuro: referência; e cinza claro: escravo 2.

Para o caso da referência tridimensional, a Figura 5.15 apresenta a trajetória dos dois manipuladores. São representadas as vista em perspectiva, superior (plano xy) e lateral (plano xz). Destaca-se que esta mantém o formato circular de maneira consistente, entretanto o movimento na direção x apresenta uma tendência, como

visto no painel inferior direito, levando ao deslocamento do centro da trajetória, no painel inferior esquerdo.

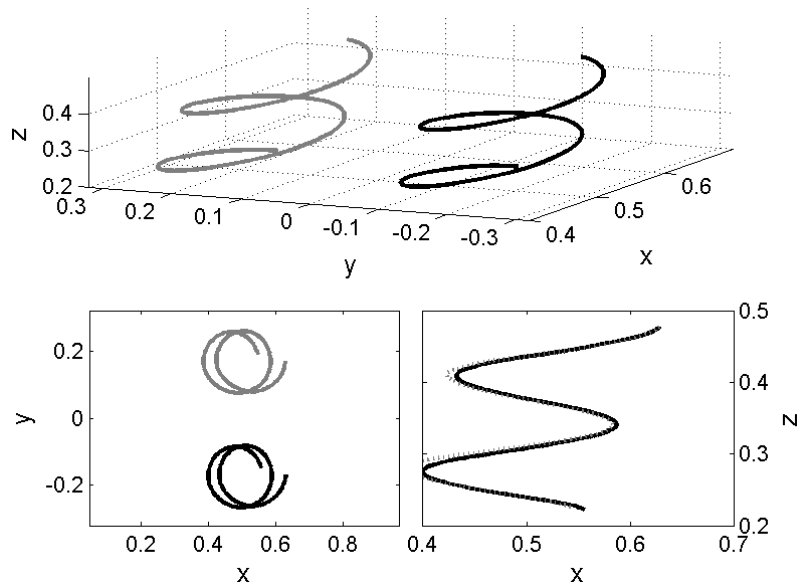


Figura 5.15: Trajetória dos braços do Motoman DIA10 rastreado uma trajetória helicoidal. Preto: escravo 1; e cinza: escravo 2.

A orientação dos efetuadores durante o movimento helicoidal é mostrada na Figura 5.16. Ao invés de representar os quatro elementos do quaternion, que não tem representação física, são utilizados sistemas de coordenadas. Assim o objetivo de controle é manter os sistemas que representam a orientação real, em azul, igual a desejada, em verde. Nota-se que um ganho mais elevado, como no painel direito, tende a aumentar a força da tarefa secundária e melhorar o resultado da regulação.

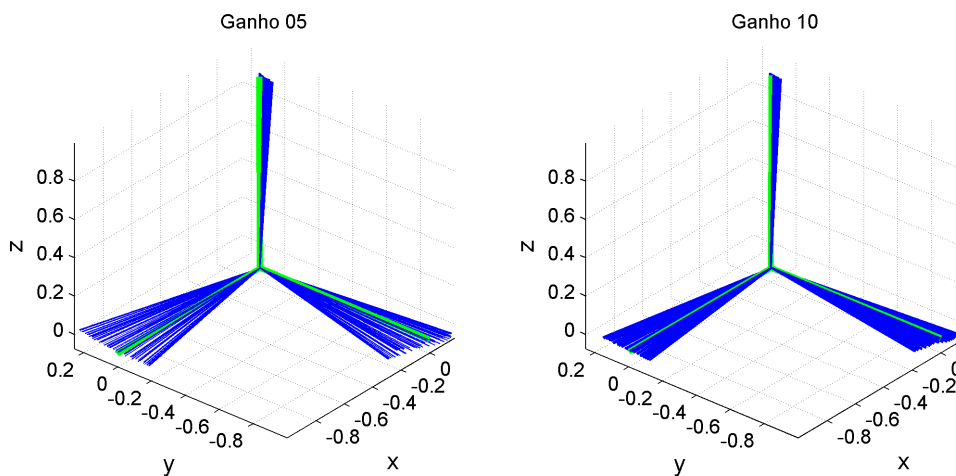


Figura 5.16: Representação da orientação durante a trajetória helicoidal com dois ganhos para a tarefa secundária. Verde: referência de orientação; e azul: orientação do efetuador.

5.2.3 Teleoperação unilateral

Para o teste de teleoperação unilateral, considerou-se como mestre um dispositivo háptico Phantom Omni (Apêndice A.3.2) ou os braços do operador captados através do Kinect. Ambos os dispositivos são acessados no Matlab através de nós do RR, como apresentado no Apêndice A. As especificações de cada teste e os resultados são apresentados a seguir.

Implementação

A estratégia de teleoperação descrita no Capítulo 3 é utilizando a estrutura proposta em Motta-Ribeiro et al. (2012), apresentada no Apêndice A. Cada camada de controle utiliza seu próprio canal de comunicação, num total de 4 *sockets*. Em cada ambiente um *script* implementa a transmissão e recepção de dados e o controle dos robôs utilizando dois laços aninhados como esquematizado no Código 5.1.

Código 5.1: Pseudocódigo Matlab para implementação da teleoperação cooperativa.

```
while it<limite || usuário finaliza
    tic % Inicia contagem de tempo para próximo passo do controlador
    while 1
        recebe pacotes de passividade
        acumula energia no buffer
        if (t+toc)>=periodo) break; end % Verifica se deve ocorrer o próximo passo
    end
    tic % Inicia a contagem de tempo utilizado em um passo do controlador
    recebe pacote de transparência
    calcula energia disponível
    calcula ação de controle
    satura ação de controle
    envia comando para o robô
    envia pacote de energia
    it=it+1; % Incrementa número de iterações
    t=toc; % Salva tempo utilizado por um passo do controlador
end
```

O laço principal é executado até que um número limite de iterações seja atingido ou o usuário envie um comando para terminá-lo. Os pacotes referentes a camada de transparência são amostrados a uma taxa fixa, no laço principal, determinada pela malha de controle, a variável *periodo*. Esta mesma taxa é utilizada para o envio de pacotes de energia, porém o recebimento destes é realizado de forma assíncrona entre dois intervalos de amostragem formando o *buffer* Q_j , dentro do laço interno.

Destaca-se que a taxa de amostragem mantida nestes laços é aproximada, utilizando as funções *tic* e *toc* de leitura relativa do *clock*. Entretanto a teleoperação representa uma camada superior de controle, sendo as camadas relativas ao acesso a *hardware* realizadas em tempo real.

Phantom Omni

O nó do RR para o Phantom fornece a posição do dispositivo no espaço Cartesiano e recebe comandos de força a serem aplicados para operador em uma malha de controle de baixo nível com amostragem de 1 kHz , adequada para a estabilidade e fluidez da realimentação de força. A referência de velocidade enviadas ao ambiente local são geradas a partir da diferença entre duas posições consecutivas do Phantom, amostradas na malha de controle da camada de transparência.

A Figura 5.17 apresenta as forças virtuais medidas durante o experimento de teleoperação unilateral. A lei de controle se mostra capaz de manter as forças de interação fluando próximas ao valor desejado, mostrado como linhas tracejadas. Observa-se que as forças de referência representam uma compressão do objeto, de forma que mesmo estas pequenas diferenças ainda mantém a carga segura.

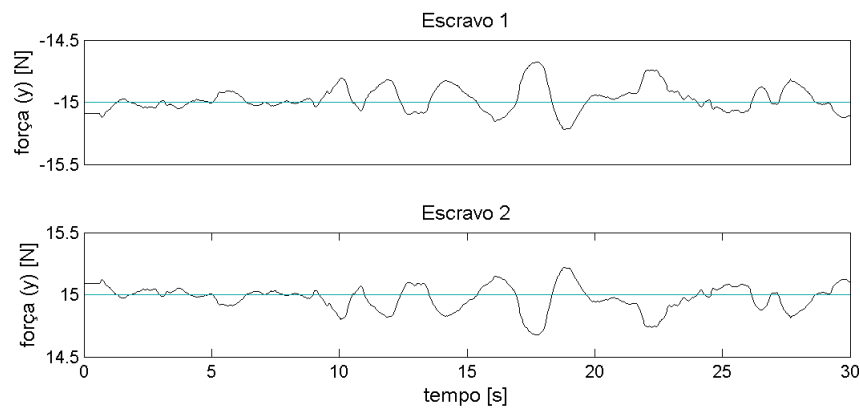


Figura 5.17: Forças virtuais medidas nos efetadores durante a teleoperação utilizando Phantom Omni como mestre. Preto: valor medido; e cinza: referência.

Esse erro de regulação de força é esperado, pelo Teorema 2, devido a presença de acoplamento entre os controles de força e velocidade, sendo resultado de uma referência de velocidade variável, ver Figura 5.18. Destaca-se que o erro é limitado.

Kinect

Na operação considerando como mestre os braços do operador, diferentemente do apresentado na Apêndice A.4, não se calcula a cinemática inversa dos pontos de esqueleto, sendo utilizado diretamente a referência de posição das mãos. Como no caso do Phantom a referência de velocidade é obtida da diferença de posições de duas amostras consecutivas. Para auxiliar o operador a manter suas mãos posicionadas de maneira consistente o esqueleto fornecido pelo Kinect é apresentado junto a uma carga, representada por um cubo, entre as suas mãos como na Figura 5.19.

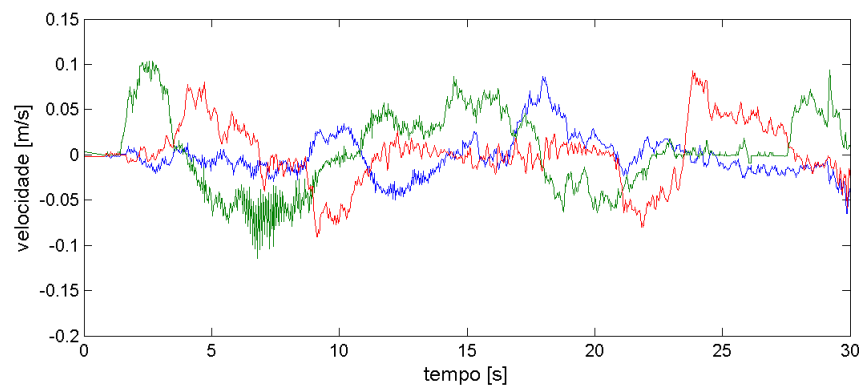


Figura 5.18: Referência de velocidade gerada pelo operador durante teleoperação unilateral com Phantom Omni. Azul: direção y ; verde: direção x ; e vermelho: direção z .

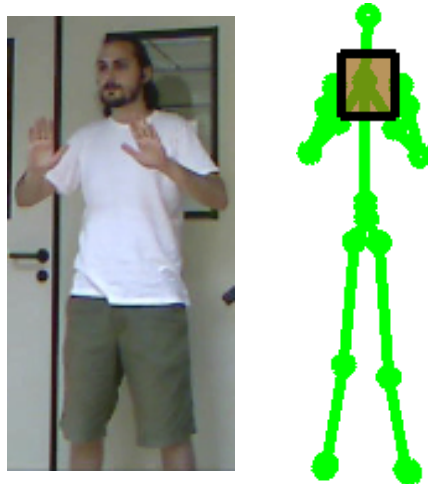
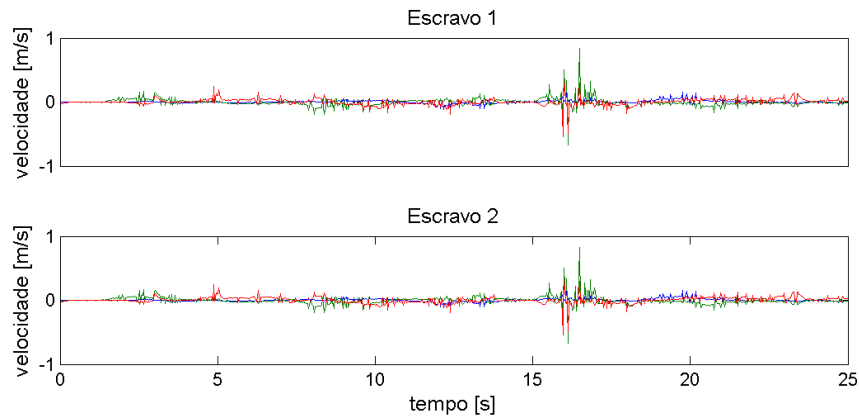
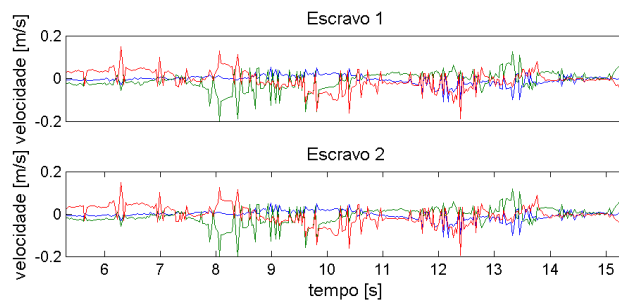


Figura 5.19: Operador comandando o robô através do Kinect. A imagem a direita serve de auxílio, representando os pontos de esqueleto e uma caixa virtual.

Assim como no caso autônomo os dois manipuladores rastreiam a referência de velocidade com error limitado, como mostrado na Figura 5.20. Observam-se alguns picos de erro presentes devido a acelerações mais intensas para mudança de sentido do movimento e possíveis falhas de rastreamento do kinect. A inércia devido a camada intermediária de controle de posição faz com que o sistema responda de forma lenta, levando aos picos.



(a) Experimento completo.



(b) Detalhamento entre 5 e 15 s.

Figura 5.20: Erro de velocidade dos efetadores durante teleoperação utilizando o Kinect como mestre. Azul: direção x ; verde: direção y ; e vermelho: direção z .

Essas mudanças de sentido do movimento mais abruptas presentes neste experimento, não tem um maior reflexo sobre a força de contato entre os efetadores e a carga, como apresentado na Figura 5.21.

Apesar de uma realimentação de força direta na operação com Kinect exigir equipamentos grandes e de auto custo, descaracterizando a naturalidade obtida com uso deste tipo de interface, é possível utilizar-se de formas indiretas de realimentação. Como exemplo destas técnicas, tem-se o uso de indicadores visuais de força (Kitagawa et al. 2005, Reiley et al. 2008) ou dispositivos de realimentação cutânea, os quais estimulam o tato em diferentes regiões da mão com intensidade proporcional a força (Pacchierotti et al. 2012, Prattichizzo et al. 2012, Schoonmaker & Cao 2006).

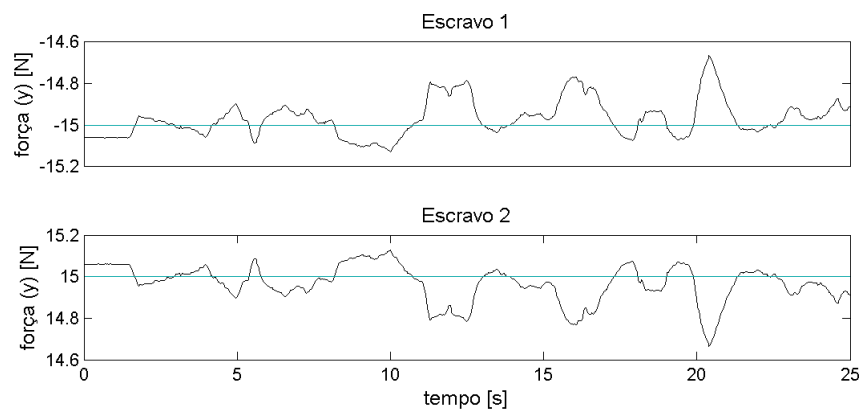


Figura 5.21: Forças medidas nos efetadores durante a teleoperação utilizando Kinect como mestre. Preto: valor medido; e cinza: referência.

Capítulo 6

Conclusões

Neste trabalho foi proposta uma metodologia para teleoperação bilateral cooperativa com múltiplos mestres comandando múltiplos robôs escravos para o problema de transporte de carga. Para garantir a estabilidade no sistema de teleoperação devido a presença de atraso no canal de comunicação utiliza-se uma estratégia em duas camadas (Franken et al. 2009). A primeira camada, de transparência, implementa as leis de controle de forma que o sistema escravo siga a referência comandada pelo sistema mestre, enquanto este reproduz para o operador as forças de interação dos escravos com o objeto. A segunda camada monitora a energia recebida e dissipada pelos sistemas robóticos durante a interação com o operador e a carga. A partir da energia monitorada, o sinal de controle é alterado para que a condição de passividade seja atendida.

A camada de transparência, aqui, é implementada considerando que o controle de cooperação dos escravos utiliza referências de velocidade e de força para os efetuadores. A referência de força é a soma de um valor constante para garantir a pegada do objeto e um valor determinado pela interação dos robôs metres com uma carga virtual. Os manipuladores mestres e escravos são agrupados de acordo com a direção, no sistema de coordenadas inercial, da força de interação com o objeto. Desta forma, não há uma necessidade de se igualar o número de manipuladores em cada ambiente. O controle de pegada é realizado diretamente no ambiente da tarefa, não sendo afetado pelo atraso no canal de comunicação.

A cooperação entre os manipuladores se baseia na estratégia de controle cooperativo proposta em (Bai & Wen 2010), tratando o transporte de carga de maneira similar ao problema de formação. Para tal, considera-se uma carga deformável em que a força de contato é modelada como função desta deformação, assim controlar a força é equivalente a controlar a distância entre os efetuadores de cada agente. Para aplicar a teleoperação de manipuladores, o trabalho original foi estendido de forma a considerar o rastreamento de uma referência velocidade. A malha de controle con-

sidera todas as forças no efetuador e não só as internas, existindo um acoplamento com o controle de velocidade.

São propostas três leis de controle: uma cinemática para manipuladores com dinâmica desprezível e duas considerando o caso de robôs com dinâmica não desprezível. No controle cinemático os seis graus de liberdade do efetuador são considerados, a posição é controlada em uma tarefa principal para movimentar o objeto, enquanto a orientação é tratada em uma tarefa secundária para manter os efetuadores perpendiculares a carga. A primeira lei de controle para manipuladores com dinâmica não desprezível apresenta um controle dual ao caso cinemático para a posição do efetuador considerando um modelo no espaço cartesiano.

O segundo caso com modelo dinâmico utiliza uma estratégia em cascata que considera uma malha interna de controle dinâmico no espaço das juntas para transformar o manipulador não linear em um novo sistema com modelo simplificado. Neste trabalho considera-se que a cascata leva a um modelo linear de segunda ordem para o efetuador. A posição e orientação podem ser tratadas de forma independente e duas leis de controle são propostas para transladar a carga mantendo os efetuadores perpendiculares a mesma. Por considerar o espaço das juntas, a estratégia em cascata facilita a implementação de técnicas adaptativas.

Para cada uma das leis de controle é apresentada uma prova teórica de que os sistemas em malha fechada são UB, garantindo a estabilidade da tarefa de cooperação. A estratégia completa de teleoperação é verificada utilizando simulação numérica considerando o controle cinemático de manipuladores planares. Essa lei de controle também é utilizada em experimentos de controle autônomo, onde não há um operador, e com teleoperação unilateral. Os experimentos foram realizados utilizando uma arquitetura flexível para teleoperação (Motta-Ribeiro et al. 2012), utilizando conjuntamente o Phantom Omni ou o Kinect e um robô Motoman DIA10 através do Matlab.

6.1 Limitações

Considera-se com principais limitações do presente trabalho:

- Os controladores para dinâmica não desprezível, projetados para sistemas de segunda ordem, utilizam a aceleração de referência. Entretanto, como o movimento é realizado diretamente pelo operador, essa informação não pode ser conhecida *a priori*. Considerando, ainda, que pode existir atraso variável no canal de comunicação um sinal de aceleração transmitido do ambiente remoto para o local, não corresponderia ao sinal de velocidade enviado em conjunto. Uma possibilidade, então, é considerar um modelo de referência no ambiente local de forma a estimar esta aceleração.

- Nos casos que consideram manipuladores com dinâmica não desprezível, os controladores propostos supõem que os parâmetros do modelo dinâmico são totalmente conhecidos, podendo assim serem compensados. Em um manipulador real essa hipótese é forte, pois incertezas na medição e simplificações na modelagem geram imprecisão neste modelo. Uma alternativa é considerar um esquema adaptativo que respeite a condição de passividade como o de Slotine & Li (1991).
- Durante os experimentos, não foram utilizados sensores de força para medir a interação entre os efetuadores e a carga. É conhecido na literatura o fato de atrasos na medida de força poderem levar a instabilidade do sistema de controle (Montemayor & Wen 2004), o que pode ser uma nova fonte de erros ao se utilizar um sensor real.

6.2 Trabalhos futuros

A seguir são apresentadas sugestões de trabalhos futuros seguindo o tema abordado. São propostas temas para desenvolvimento teórico e complementação experimental do trabalho.

- Investigar o modelo de contato e as hipóteses necessárias para controlar a orientação da carga. Esse novo objetivo de controle exige que se transmitam para a carga os torques dos efetuadores e altera a análise de estabilidade, uma vez que $\dot{R}(\theta_c) \neq 0$. A rotação da carga também altera o controle de orientação dos efetuadores, passando de um problema de regulação para um de rastreamento, ou mesmo alterando-se a restrição de perpendicularidade.
- Na presença de um sensor de força e torque no efetuador do manipulador, o controle de orientação pode ser realizado utilizando-se essas medidas. Desta maneira, não se faz necessário o conhecimento prévio da posição do efetuador em relação a carga e nem sobre a geometria da superfície da mesma.
- A hipótese de parâmetros completamente conhecidos para o modelo dinâmico, utilizada nos casos de dinâmica não desprezível, pode ser relaxada fazendo-se uso de uma estratégia de controle adaptativo. Neste caso, o controlador em cascata é favorecido por utilizar o controle no espaço das juntas e pelas propriedades de passividade da estratégia proposta e de controles adaptativos, como o de Slotine & Li (1991). Esse uso de cascata com adaptação da lei de controle dinâmico foi explorada em Lizarralde et al. (2008), Zachi, Hsu, Ortega & Lizarralde (2006), entre outros.
- Para obter a referência de aceleração utilizada nas leis de controle para manipuladores com dinâmica não desprezível, pode-se explorar o uso de filtragem do sinal de velocidade ou estimadores de maior ordem no ambiente local.

- Ao se utilizar o controle de velocidade, simplifica-se o mapeamento entre os espaços de trabalho dos mestres e dos escravos. Entretanto não é garantida a sincronia entre a posição dos mesmos. Na literatura são propostas diferentes estratégias para realizar esta sincronia durante a teleoperação, assim, dependendo da tarefa pretendida, pode-se adotar novos termos na lei de controle para o rastreamento das posições dos mestres.
- Realização dos experimentos de controle autônomo com um sensor de força.
- Realização de experimento com teleoperação bilateral, utilizando dois dispositivos hápticos como mestre.

Referências Bibliográficas

- Aldana, C., Nuno, E. & Basanez, L. (2012), Bilateral teleoperation of cooperative manipulators, *in* ‘Proc. IEEE Int. Conf. Robotics and Automation’, pp. 4274–4279.
- Anderson, R. (1995), ‘Smart: a modular control architecture for telerobotics’, *IEEE Robot. Automat. Mag.* **2**(3), 10–18.
- Anderson, R. & Spong, M. (1989), ‘Bilateral control of teleoperators with time delay’, *IEEE Trans. Automat. Contr.* **34**(5), 494–501.
- Anisi, D., Gunnar, J., Lillehagen, T. & Skourup, C. (2010), Robot automation in oil and gas facilities: Indoor and onsite demonstrations, *in* ‘Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems’, pp. 4729–4734.
- Anisi, D., Persson, E. & Heyer, C. (2011), Real-world demonstration of sensor-based robotic automation in oil and gas facilities, *in* ‘Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems’, pp. 235–240.
- Arcara, P. & Melchiorri, C. (2002), ‘Control schemes for teleoperation with time delay: A comparative study’, *Robotics and Autonomous Systems* **38**(1), 49–64.
- Bacocco, R. & Melchiorri, C. (2009), Lq control design of cooperative teleoperation systems, *in* ‘Proc. IEEE Int. Conf. Advanced Robotics’, pp. 1–6.
- Bai, H. & Wen, J. T. (2010), ‘Cooperative load transport: a formation-control perspective’, *IEEE Trans. Robot.* **26**(4), 742–750.
- Ballantyne, G. (2002), ‘Robotic surgery, telerobotic surgery, telepresence, and telerobotics’, *Surgical Endoscopy And Other Interventional Techniques* **16**(10), 1389–1402.
- Bambang, R. (2007), ‘Development of architectures for internet telerobotics systems’, *J. Bionic Eng.* **4**(4), 291–297.

- Boukhnifer, M., Ferreira, A. & Fontaine, J. (2004), Scaled teleoperation controller design for micromanipulation over internet, *in* ‘Proc. IEEE Int. Conf. Robotics and Automation’, Vol. 5, pp. 4577–4583 Vol.5.
- Caccavale, F., Natale, C., Siciliano, B. & Villani, L. (1998), ‘Resolved-acceleration control of robot manipulators: A critical review with experiments’, *Robotica* **16**, 565–573.
URL: http://journals.cambridge.org/article_S0263574798000290
- Caccavale, F. & Uchiyama, M. (2008), Cooperative manipulators, *in* B. Siciliano & O. Khatib, eds, ‘Springer Handbook of Robotics’, Springer Berlin Heidelberg, pp. 701–718.
- Cho, J. H., From, P. J., Annerstedt, M., Robertsson, A. & Johansson, R. (2012), Design of an intermediate layer to enhance operator awareness and safety in telesurgical systems, *in* ‘Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems’, pp. 805–822.
- Chopra, N. & Spong, M. W. (2005), On synchronization of networked passive systems with time delays and application to bilateral teleoperation, *in* ‘Annual Conference of Society of Instrument and Control Engineers of Japan’, pp. 6–10.
- Denasi, A., Kostic, D. & Nijmeijer, H. (2013), ‘Time delay compensation in bilateral teleoperations using impact’, *IEEE Trans. Contr. Syst. Technol.* **21**(3), 704–715.
- Desbats, P., Geffard, F., Piolain, G. & Coudray, A. (2006), ‘Force-feedback teleoperation of an industrial robot in a nuclear spent fuel reprocessing plant’, *Industrial Robot: An International Journal* **33**(3), 178–186.
- Ferrell, W. & Sheridan, T. (1967), ‘Supervisory control of remote manipulation’, *Spectrum, IEEE* **4**(10), 81–88.
- Franken, M., Stramigioli, S., Reilink, R., Secchi, C. & Macchelli, A. (2009), Bridging the gap between passivity and transparency, *in* ‘Proc. Robotics: Science and Systems’, Seattle, USA.
- Freitas, G., Gleizer, G., Lizarralde, F., Hsu, L. & dos Reis, N. R. S. (2010), ‘Kinematic reconfigurability control for an environmental mobile robot operating in the amazon rain forest’, *Journal of Field Robotics* **27**(2), 197–216.

- From, P. J. (2010), Off-shore Robotics: Robust and Optimal Solutions for Autonomous Operation, PhD thesis, Norwegian University of Science and Technology.
- Goto, S., Naka, T., Matsuda, Y. & Egashira, N. (2010), Teleoperation system of robot arms combined with remote control and visual servo control, *in* ‘Proc. SICE Annual Conf.’, pp. 1975–1981.
- Graf, B. & Pfeiffer, K. (2007), Mobile robots for offshore inspection and manipulation, *in* ‘Proc. Int. Petroleum Technology Conf.’.
- Guenther, R. & Hsu, L. (1993), Variable structure adaptive cascade control of rigid-link electrically-driven robot manipulators, *in* ‘Proc. IEEE Conf. Dec. and Contr.’, San Antonio, pp. 2137–2142.
- Hacioglu, Y., Arslan, Y. Z. & Yagiz, N. (2011), ‘MIMO fuzzy sliding mode controlled dual arm robot in load transportation’, *Journal of the Franklin Institute* **348**(8), 1886 – 1902.
- Haddad, W. M. & Chellaboina, V. (2011), *Nonlinear dynamical systems and control: a Lyapunov-based approach*, Princeton University Press, New Jersey, EUA.
- Hirche, S., Stanczyk, B. & Buss, M. (2004), Haptic tele-assembly over the internet, *in* ‘Proc. EuroHaptics Conf.’, pp. 417–421.
- Hirzinger, G., Brunner, B., Dietrich, J. & Heindl, J. (1994), Rotex-the first remotely controlled robot in space, *in* ‘Proc. IEEE Int. Conf. Robotics and Automation’, pp. 2604–2611 vol.3.
- Hokayem, P. & Spong, M. (2006), ‘Bilateral teleoperation: An historical survey’, *Automatica* **42**(12), 2035–2057.
- Hsu, L., Costa, R. & Lizarralde, F. (2007), ‘Lyapunov/passivity-based adaptive control of relative degree two MIMO systems with an application to visual servoing’, *IEEE Trans. Automat. Contr.* **52**(2), 364–371.
- Hsu, P. (1993), ‘Coordinated control of multiple manipulator systems’, *IEEE Trans. Robot. Automat.* **9**(4), 400–410.
- Hutchinson, S., Hager, G. & Corke, P. (1996), ‘A tutorial on visual servo control’, *IEEE Trans. Robot. Automat.* **12**(5), 651–670.
- Khalil, H. (2002), *Nonlinear Systems*, Prentice Hall, New Jersey, EUA.
 URL: http://books.google.com.br/books?id=t_d1QgAACAAJ

- Khatib, O. (1987), ‘A unified approach for motion and force control of robot manipulators: The operational space formulation’, *IEEE Trans. Robot. Automat.* **3**(1), 43–53.
- Kitagawa, M., Dokko, D., Okamura, A. M. & Yuh, D. D. (2005), ‘Effect of sensory substitution on suture-manipulation forces for robotic surgical systems’, *The Journal of Thoracic and Cardiovascular Surgery* **129**(1), 151 – 158.
- Kosuge, K. & Oosumi, T. (1996), Decentralized control of multiple robots handling an object, *in* ‘Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems’, Vol. 1, pp. 318–323.
- Kume, Y., Hirata, Y., dong Wang, Z. & Kosuge, K. (2002), Decentralized control of multiple mobile manipulators handling a single object in coordination, *in* ‘Proc. IEEE/RSJ Int. Conf. Intelligent Robots Systems’, Vol. 3, pp. 2758–2763 vol.3.
- Lawrence, D. (1993), ‘Stability and transparency in bilateral teleoperation’, *IEEE Trans. Robot. Automat.* **9**(5), 624–637.
- Lee, D. & Spong, M. (2005), Bilateral teleoperation of multiple cooperative robots over delayed communication networks: Theory, *in* ‘Proc. IEEE Int. Conf. Robotics and Automation’, pp. 360–365.
- Lee, D. & Spong, M. (2006), ‘Passive bilateral teleoperation with constant time delay’, *IEEE Trans. Robot.* **22**(2), 269–281.
- Leung, G., Francis, B. & Apkarian, J. (1995), ‘Bilateral controller for teleoperators with time delay via μ -synthesis’, *IEEE Trans. Robot. Automat.* **11**(1), 105–116.
- Lizarralde, F., Hsu, L. & Costa, R. (2008), Adaptive visual servoing of robot manipulators without measuring the image velocity, *in* ‘Proc. of the 17th IFAC World Congress’, Seoul (South Korea), pp. 4108–4113.
- Lizarralde, F. & Wen, J. (1996), ‘Attitude control without angular velocity measurement: a passivity approach’, *IEEE Trans. Automat. Contr.* **41**(3), 468–472.
- Lozano, R., Chopra, N. & Spong, M. (2002), Passivation of force reflecting bilateral teleoperators with time varying delay, *in* ‘Proc. 8 Mechatronics Forum’, pp. 24–26.
- Luh, J. & Zheng, Y. (1987), ‘Constrained relations between two coordinated industrial robots for motion control’, *Int. J. Robotics Res.* **6**(3), 60–70.

- Marescaux, J., Leroy, J., Rubino, F., Smith, M., Vix, M., Simone, M. & Mutter, D. (2002), ‘Transcontinental robot-assisted remote telesurgery: feasibility and potential applications.’, *Ann Surg* **235**(4), 487–92.
- Montemayor, G. & Wen, J. (2004), Multiple robot force control with delayed force measurements, *in* ‘Control Applications, 2004. Proceedings of the 2004 IEEE International Conference on’, Vol. 1, pp. 327–332 Vol.1.
- Montemayor, G. & Wen, J. (2005), Decentralized collaborative load transport by multiple robots, *in* ‘Proc. IEEE Int. Conf. Robotics and Automation’, pp. 372–377.
- Motta-Ribeiro, G. C., Leite, A. C., From, P. J., Lizarralde, F. & Hsu, L. (2012), Uma arquitetura para teleoperação integrando interface natural, realimentação de força e servovisão, *in* ‘Proc. Congresso Brasileiro de Automatica’.
- Munir, S. & Book, W. J. (2003), ‘Control techniques and programming issues for time delayed internet based teleoperation’, *J. DYN. SYST.* **125**(2), 205–214.
- Murray, R. M., Li, Z. & Sastry, S. S. (1994), *A mathematical introduction to robotic manipulation*, Boca raton, Florida, EUA.
- Nakanishi, J., Cory, R., Mistry, M., Peters, J. & Schaal, S. (2008), ‘Operational space control: A theoretical and empirical comparison’, *Int. J. Robotics Res.* **27**(6), 737–757.
- Nakanishi, J., Mistry, M., Peters, J. & Schaal, S. (2007), Towards compliant humanoids-an experimental assessment of suitable task space position/orientation controllers, *in* ‘Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems’, pp. 2520–2527.
- Niemeyer, G., Preusche, C. & Hirzinger, G. (2008), Telerobotics, *in* B. Siciliano & O. Khatib, eds, ‘Springer Handbook of Robotics’, Springer Berlin Heidelberg, pp. 741–757.
- Niemeyer, G. & Slotine, J.-J. (1991), ‘Stable adaptive teleoperation’, *IEEE J. Oceanic Eng.* **16**(1), 152–162.
- Nuño, E., Basañez, L. & Ortega, R. (2011), ‘Passivity-based control for bilateral teleoperation: A tutorial’, *Automatica* **47**(3), 485 – 495.
- Oboe, R. (2001), ‘Web-interfaced, force-reflecting teleoperation systems’, *IEEE Trans. Ind. Electron.* **48**(6), 1257–1265.

- Oboe, R. & Fiorini, P. (1998), ‘A design and control environment for internet-based telerobotics’, *Int. J. Robot. Res.* **17**(4), 433–449.
- Pacchierotti, C., Chinello, F., Malvezzi, M., Meli, L. & Prattichizzo, D. (2012), Two finger grasping simulation with cutaneous and kinesthetic force feedback., *in* P. Isokoski & J. Springare, eds, ‘EuroHaptics (1)’, Vol. 7282 of *Lecture Notes in Computer Science*, Springer, pp. 373–382.
URL: <http://dblp.uni-trier.de/db/conf/haptics/eurohaptics2012-1.html#PacchierottiC MMP12>
- Pereira, G. A. S., Campos, M. F. M. & Kumar, V. (2004), ‘Decentralized algorithms for multi-robot manipulation via caging.’, *Int. J. Robotics Res.* **23**(7-8), 783–795.
URL: <http://dblp.uni-trier.de/db/journals/ijrr/ijrr23.html#PereiraCK04>
- Prattichizzo, D., Pacchierotti, C. & Rosati, G. (2012), ‘Cutaneous force feedback as a sensory subtraction technique in haptics’, *IEEE Trans. Haptics* **5**(4), 289–300.
- Prattichizzo, D. & Trinkle, J. (2008), Grasping, *in* B. Siciliano & O. Khatib, eds, ‘Springer Handbook of Robotics’, Springer Berlin Heidelberg, pp. 671–700.
- Preusche, C., Reintsema, D., Landzettel, K. & Hirzinger, G. (2006), Robotics component verification on iss rokvis - preliminary results for telepresence, *in* ‘Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems’, pp. 4595–4601.
- Quigley, M., Conley, K., Gerkey, B. P., Faust, J., Foote, T., Leibs, J., Wheeler, R. & Ng, A. Y. (2009), Ros: an open-source robot operating system, *in* ‘ICRA Workshop on Open Source Software’.
- Reiley, C. E., Akinbiyi, T., Burschka, D., Chang, D. C., Okamura, A. M. & Yuh, D. D. (2008), ‘Effects of visual force feedback on robot-assisted surgical task performance’, *The Journal of Thoracic and Cardiovascular Surgery* **135**(1), 196–202.
- Remiro, R. R. L. (2009), Determinantes da carga física de trabalho em plataformas de petróleo: o caso da operação do sistema de pig, Master’s thesis, Universidade Federal do Rio de Janeiro.
- Ryu, J.-H., Dong-Soo, K. & Hannaford, B. (2004), ‘Stable teleoperation with time-domain passivity control’, *IEEE Trans. Robot.* **20**(2), 365–373.

- Ryu, J.-H. & Preusche, C. (2007), Stable bilateral control of teleoperators under time-varying communication delay: Time domain passivity approach, *in* ‘Proc. IEEE Int. Conf. Robotics and Automation’, pp. 3508–3513.
- Safaric, R., Hedrih, I., Klobucar, R. & Sorgo, B. (2003), Remote controlled robot arm, *in* ‘Proc. ICIT’, Vol. 2, pp. 1202–1207.
- Schoonmaker, R. & Cao, C. (2006), Vibrotactile force feedback system for minimally invasive surgical procedures, *in* ‘Proc. IEEE Int. Conf. Systems, Man and Cybernetics’, Vol. 3, pp. 2464–2469.
- Sheng, J. & Spong, M. (2004), Model predictive control for bilateral teleoperation systems with time delays, *in* ‘Canadian Conf. Electrical and Computer Engineering’, Vol. 4, pp. 1877–1880.
- Sheridan, T. & Ferrell, W. (1963), ‘Remote manipulative control with transmission delay’, *IEEE Trans. Hum. Factors Electron.* **HFE-4**(1), 25–29.
- Siciliano, B., Sciavicco, L., Villani, L. & Oriolo, G. (2009), *Robotics: modelling, planning and control*, Springer Verlag.
- Silva, A., Ramirez, O., Vega, V. & Oliver, J. (2009), Phantom omni haptic device: Kinematic and manipulability, *in* ‘Proc. Electronics, Robotics and Automotive Mechanics Conf.’, IEEE, pp. 193–198.
- Sirouspour, S. (2005), ‘Modeling and control of cooperative teleoperation systems’, *IEEE Trans. Robot.* **21**(6), 1220 – 1225.
- Slotine, J. & Li, W. (1991), *Applied Nonlinear Control*, Prentice Hall, New Jersey, EUA.
- Smith, A. C. & Hashtrudi-Zaad, K. (2006), ‘Smith predictor type control architectures for time delayed teleoperation’, *Int. J. Robotics Res.* **25**(8), 797–818.
- Stanczyk, B. & Buss, M. (2004), Development of a telerobotic system for exploration of hazardous environments, *in* ‘Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems’, Vol. 3, pp. 2532–2537.
- Stramigioli, S., Mohony, R. & Corke, P. (2010), A novel approach to haptic teleoperation of aerial robot vehicles, *in* ‘Proc. IEEE Int. Conf. Robotics and Automation’, Anchorage, USA.
- Stramigioli, S., Secchi, C., Schaft, A. J. & Fantuzzi, C. (2005), ‘Sampled data systems passivity and discrete port-hamiltonian systems’, *IEEE Trans.*

Robot. **21**(4), 574–587.

URL: <http://doc.utwente.nl/54220/>

- Sugar, T. G. & Kumar, V. (2002a), ‘Control of cooperating mobile manipulators’, *IEEE Trans. Robot. Automat.* **18**(1), 94–103.
- Sugar, T. G. & Kumar, V. (2002b), ‘Design and control of a compliant parallel manipulator’, *Journal of Mechanical Design* **124**, 676–683.
- Taylor, K., Dalton, B. & Trevelyan, J. (1999), ‘Web-based telerobotics’, *Robotica* **17**(1), 49–57.
- Transth, A., Skotheim, O., Schumann-Olsen, H., Johansen, G., Thielemann, J. & Kyrkjebo, E. (2010), A robotic concept for remote maintenance operations: A robust 3d object detection and pose estimation method and a novel robot tool, in ‘Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems’, pp. 5099–5106.
- Tsytkin, Y. Z. & Holmberg, U. (1995), ‘Robust stochastic control using the internal mode principle and internal model control’, *Int. J. Contr.* **61**(4), 809–822.
- Uchiyama, M. & Dauchez, P. (1992), ‘Symmetric kinematic formulation and non-master/slave coordinated control of two-arm robots’, *Advanced Robotics* **7**(4), 361–383.
- Wason, J. & Wen, J. (2011), Robot raconteur: A communication architecture and library for robotic and automation systems, in ‘Proc. CASE’, pp. 761–766.
- Yokokohji, Y. & Yoshikawa, T. (1994), ‘Bilateral control of master-slave manipulators for ideal kinesthetic coupling-formulation and experiment’, *IEEE Trans. Robot. Automat.* **10**(5), 605–620.
- Yuan, J. (1988), ‘Closed-loop manipulator control using quaternion feedback’, *IEEE J. Robot. Automat.* **4**(4), 434–440.
- Zachi, A. R. L., Hsu, L., Lizarralde, F. & Leite, A. C. (2006), ‘Adaptive control of nonlinear visual servoing systems for 3D cartesian tracking’, *Automation & Control Brazilian Mag.* **17**(4), 381–390.
- Zachi, A. R. L., Hsu, L., Ortega, R. & Lizarralde, F. (2006), ‘Dynamic control of uncertain manipulators through immersion and invariance adaptive visual servoing’, *Int. J. Robotics Res.* **25**(11), 1149–1159.

Apêndice A

Arquitetura de Teleoperação

De forma a dispor de uma plataforma para realização de experimentos em teleoperação com a capacidade de prototipagem rápida e com flexibilidade para expansão e adição de diferentes equipamentos, Motta-Ribeiro et al. (2012) propõem uma arquitetura baseada em Matlab e Robot RaconteurTM(RR). Essa arquitetura permite a realização das seguintes ações, individuais ou combinadas:

1. Movimentação mestre/escravo e realimentação visual;
2. Controle de posição via servovisão;
3. Alteração do estado da garra;
4. Realimentação de força,

a fim de garantir a realização bem sucedida de uma tarefa teleoperada (e.g., *pick-and-place*). Como um exemplo de implementação, neste trabalho a primeira operação pode ser realizada com o Phantom Omni (Sensable, EUA) ou o Kinect (Microsoft Corp., EUA).

As diferentes interfaces de comando do robô localizadas no ambiente remoto são integradas através do Matlab sendo enviados informações de comando, por exemplo posições no espaço das juntas ou erro no espaço da imagem, através da internet para o ambiente local, onde estão as malhas de controle do manipulador robótico. O ambiente local, por sua vez, fornece informações para realimentação do operador, que pode ser visual e háptica. Os fluxos de dados desta arquitetura são apresentados na Figura A.1, sendo a infraestrutura local e remota detalhadas nas seções seguintes.

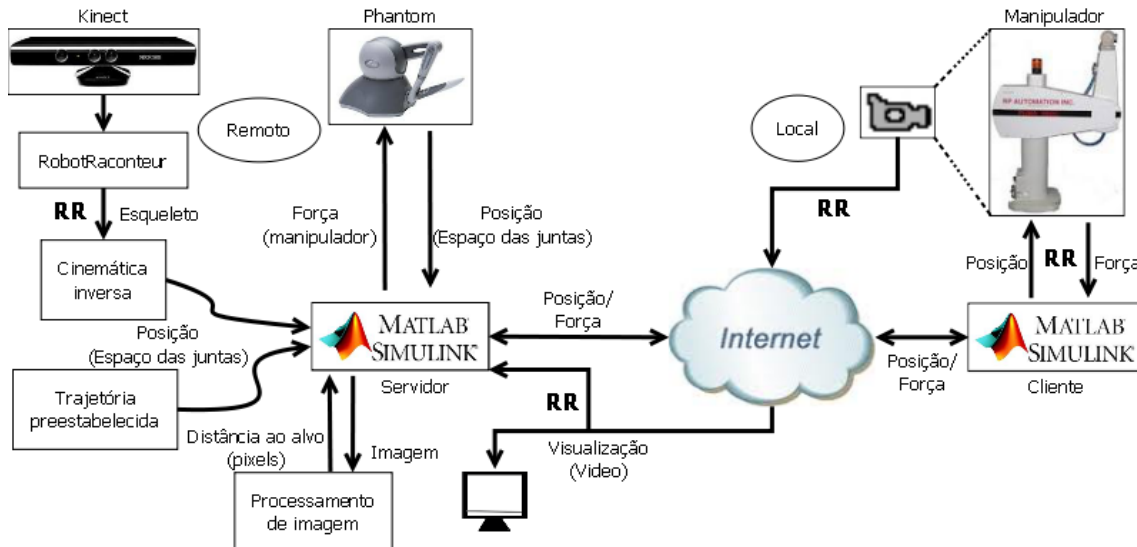


Figura A.1: Diagrama de interfaces entre as diferentes modalidades de entrada (ambiente remoto) e o sistema de controle (ambiente local). RR indica a comunicação através do Robot Raconteur. Retirado de (Motta-Ribeiro et al. 2012)

A.1 Comunicação

Como mostrado na Figura A.1 existem duas vias de comunicação entre os ambientes remoto e local, uma para operação do manipulador e outra para transmissão de vídeo. A primeira exige a maior taxa de transmissão possível, podendo ser robusta a desordenamento de mensagens se existir identificação e rejeição de informações atrasadas, aceitando-se sempre a mais recente. Deste modo, optou-se por comunicação UDP, com menor *overhead* e sem repetição de pacotes (Oboe 2001). O servidor e o cliente são implementados em Matlab através do *TCP/UDP toolbox*.

Para estabelecer a conexão, o cliente envia um comando para o servidor, conhecendo previamente o seu endereço IP, este então reconhece o endereço do cliente e inicia o envio de comandos de acordo com o modo de operação estabelecido. O controle de acesso ao manipulador é realizado pelo ambiente local, que define a conexão com o servidor UDP remoto (ver Figura A.2). Com isso, mesmo que o sistema de controle esteja em uma rede virtual local (ver Apêndice A.2), ao se requisitar a conexão, o roteador é capaz de resolver as questões de roteamento interno de portas. As possíveis barreiras de *firewall* estabelecidas pela rede do ambiente local podem ser ultrapassadas através da utilização de uma VPN (em inglês, *virtual private network*).

A segunda via de comunicação é utilizada para o envio das imagens do ambiente local como realimentação ao operador remoto, necessitando de ordenamento e baixa taxa de perda de pacotes. Optou-se então pelo protocolo TCP com a retransmissão automática de pacotes (Oboe 2001). Neste caso o servidor é local, pois o

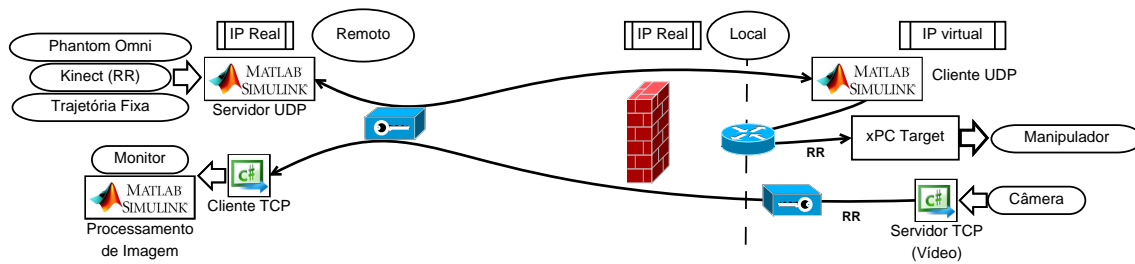


Figura A.2: Diagrama representativo das vias de comunicação para a arquitetura de teleoperação proposta. **RR** indica a comunicação através do Robot Raconteur. Retirado de (Motta-Ribeiro et al. 2012)

recurso acessado pode ser utilizado de forma segura por múltiplos clientes, no ambiente remoto de operação ou não. O gerenciamento de conexões e a transmissão da imagem são realizados através de um programa de *stream* utilizando a arquitetura RR (Wason & Wen 2011).

A.1.1 Robot Raconteur

O RR¹ é uma arquitetura e biblioteca de comunicação para aplicações de robótica e automação. É baseada em nós, tipicamente processos em uma máquina, que se conectam através de canais em um modelo servidor/cliente, ver Figura A.3. A conexão é realizada através de uma requisição do nó cliente que conhece previamente o endereço IP do nó servidor. Este então expõe suas funcionalidades por meio de um texto com as definições de suas estruturas e objetos em um formato independente de linguagem. Desta forma, o cliente não necessita conhecer previamente as características do servidor, podendo o mesmo ser alterado de forma independente (Wason & Wen 2011).

A conexão entre um cliente e um servidor de um serviço é realizada em duas etapas, a requisição de conexão e a criação da interface entre os nós. A Figura A.4 representa este processo. O cliente recebe a descrição das funcionalidades do serviço em formato de texto. A partir desta descrição, a implementação da interface ocorre de maneira distinta de acordo com a linguagem em que é implementado o nó cliente. Em linguagens fortemente tipadas, como C# ou C++, a interface com o serviço é criada em tempo de compilação e validada através da comparação da definição deste na máquina local e a recebida durante a conexão. Para linguagens como o Matlab, uma classe específica é automaticamente gerada, em tempo de execução, para cada serviço, utilizando uma camada de comunicação única.

Para o lado do servidor, a definição em formato texto da interface, chamada de arquivo *robdef*, é utilizada para gerar automaticamente o código para expor as suas

¹A época da realização desta dissertação o RR estava disponível em robotraconteur.com

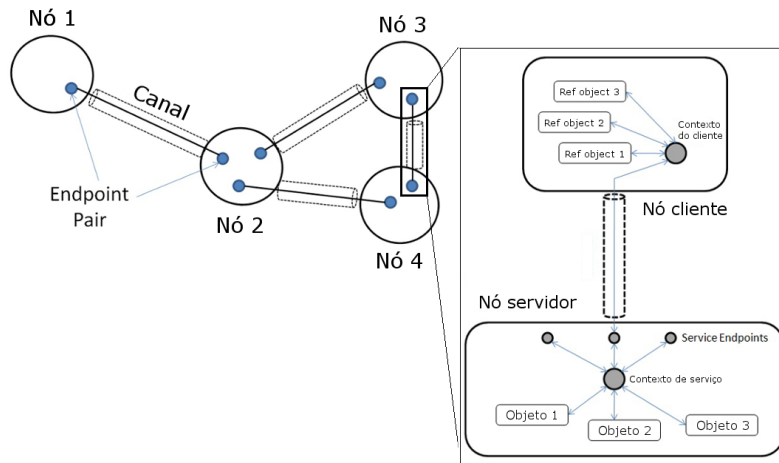


Figura A.3: Diagrama representando os elementos da arquitetura do Robot Raconteur. Adaptado de (Wason & Wen 2011).

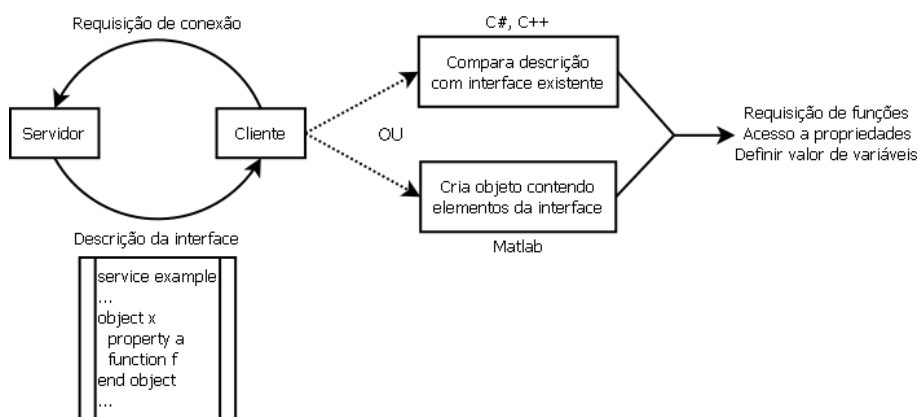


Figura A.4: Diagrama do procedimento de conexão entre nós do Robot Raconteur.

propriedade e funções. Resta ao programador escrever o código que implementa as funcionalidades do serviço. Um exemplo de robdef é apresentado no Código A.1.

Código A.1: Exemplo de arquivo robdef que define a interface para um serviço de interação com o Phantom Omni.

```
service PhantomOmniRemote_interface

struct SpatialOrientation
  # Orientação por quaternion
  field double[4] orientation
  # Posição cartesiana
  field double[3] position
end struct

object OmniHost
  objref Omni[] o
end object

object Omni
  property SpatialOrientation pose
  function void SetForce(double[3] force)
end object
```

O Código A.1 define um serviço para interagir com o Phantom Omni mostrando os diferentes tipos de elementos que podem ser definidos. Com este serviço é possível utilizar, através da rede, um ou mais dispositivos conectados em um mesmo computador. Cada dispositivo corresponde a um objeto `Omni`, sendo armazenado como um elemento do array `OmniHost`. Após a conexão, o cliente pode interagir com cada um dos objetos separadamente sendo capaz de acessar a pose do Phantom, através da propriedade `pose`, e enviar vetores de força para serem aplicados como realimentação, utilizando a função `SetForce`.

Os canais de transporte de mensagem implementados permitem a comunicação através de TCP/IP para uma rede local ou internet, de uma *pipe* para processos internos em uma máquina Windows e da porta serial para processadores embarcados de baixo consumo. O RR considera sistemas em tempo real ou baseados em eventos e permite programação em diferentes linguagens. Atualmente são permitidos nós desenvolvidos em linguagens C#, C++ e Matlab ou que utilizem a plataforma Arduino (Wason & Wen 2011). A possibilidade de integração com o Matlab e o reconhecimento de funcionalidades no ato da conexão, tornam o RR mais adequado a prototipagem rápida que ferramentas mais gerais como o ROS (Quigley et al. 2009).

A.2 Infraestrutura Local

O ambiente de operação local deve conter ao menos um manipulador robótico, com uma malha de controle de juntas e equipado com um efetuator. Como itens adicionais podem estar presentes um sensor de força acoplado ao efetuator, para teleoperação bilateral ou controle local de força, e uma câmera de vídeo, para realimentação

visual e servovisão. Devido a utilização do Matlab para a construção da interface de teleoperação, um requisito para todos os componentes presentes é a capacidade de integração com esta ferramenta, de forma direta ou através do RR. Esta integração indireta é apresentada a seguir para os casos da malha de controle e da câmera.

A.2.1 Sistema de controle

As malhas de controle de sistemas robóticos geralmente necessitam de uma velocidade e precisão na temporização dos ciclos que é dificilmente garantida pelo Matlab, requerendo em determinados casos ação em tempo real. A interface entre esta camada de alta precisão e o controle de alto nível no Matlab é realizada através de um serviço implementado com RR.

Há duas maneiras de implementar a malha de controle em um serviço RR: utilizando um computador dedicado e a ferramenta XPC-Target ou em um processo de alta prioridade em Windows.

A primeira solução se destina a sistemas robóticos em que deve ser implementado malhas de controle atuando diretamente no *hardware*. Consiste da combinação de uma distribuição Unix de tempo real rodando programas gerados automaticamente a partir de modelos do Simulink. Estes podem utilizar blocos especiais como, por exemplo, entradas e saídas digitais ou analógicas. A comunicação entre o sistema em tempo real e o computador rodando Matlab pode ser realizada por canais TCP, UDP, serial, entre outros. Para o controle do manipulador um serviço do RR é implementado diretamente no modelo do Simulink, mantendo uma estrutura sincronizada entre os dois computadores, assemelhando-se a uma memória compartilhada.

Um exemplo de aplicação é mostrado no diagrama da Figura A.5. Trata-se de um sistema com dois manipuladores PUMA 560 montados sobre duas juntas de revolução e uma prismática, do laboratório de robótica do *Center for Automation Technologies and Systems* (CATS) da *Rensselaer Politechnic Institute* (RPI) (Wason & Wen 2011), utilizado nos teste da arquitetura na Apêndice A.4.

No caso de utilização de um processo de alta prioridade, o robô já possui um sistema de controle em tempo real que atua sobre o *hardware*. Assim o controle presente no computador possui uma exigência de tempo real não crítica. Como no caso anterior a malha é implementada em Simulink com um código sendo gerado automaticamente para rodar em uma aplicação diferente do Matlab. No lugar do computador externo com sistema operacional de tempo real, um programa em C# executa o código gerado. São utilizados dois *treads*, um de alta prioridade para a malha de controle e outro normal para o serviço RR que realiza a interface com o Matlab, ou qualquer outro nó RR. Um exemplo de utilização desta solução é controle do robô Motoman DIA10 utilizado nos experimentos de cooperação na Seção 5.2.

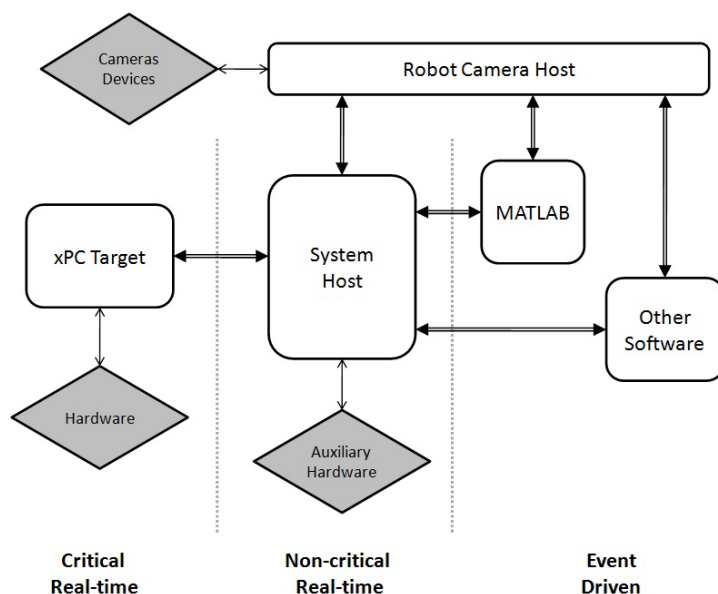


Figura A.5: Diagrama esquemático de um sistema de controle utilizando RR e *XPC-Target* para controle de dois manipuladores PUMA 560 sobre um sistema de trilho. Adptado de (Wason & Wen 2011)

A.2.2 Câmera

Para acessar as imagens da câmera local no ambiente remoto, estas são transmitidas utilizando um servidor desenvolvido com pacote básico do RR. Esta aplicação chamada de *RobotCameraHost*, permite que qualquer câmera compatível com a biblioteca *DirectShow* do Windows tenha suas imagens disponibilizadas para todos os computadores de uma rede. É possível utilizar mais de um equipamento de captura de vídeo em um mesmo computador selecionando-se um ou mais pra serem transmitidos ou acessados no momento da execução. A configuração do programa e dos dispositivos é realizada através de um arquivo XML, podendo-se alterar taxa de transmissão, resolução de captura e transmissão, configuração de filtro, entre outros.

Por se tratar de um servidor RR é possível realizar múltiplos acessos simultâneos a mesmo serviço com diferentes linguagens e aplicações. Desta forma, a realimentação visual é realiza através de uma aplicação C# fornecida com o pacote *RobotCameraHost* com visualização do *stream* de forma contínua. Enquanto a servovisão se utiliza de imagens amostradas diretamente do Matlab através de um nó nativo.

A.3 Infraestrutura Remota

No ambiente remoto estão presentes os dispositivos mestre integrados por uma interface gráfica (GUI). Assim como no ambiente local, a única limitação para a agregar novas ferramentas a teleoperação é integração destas com o Matlab, o que é facilitado

pelo uso do RR ou da ferramenta *MEX-function*, nativa do Matlab, que permite a implementação direta de funções em C/C++. Atualmente estão presentes no sistema o Kinect, o Phantom e um controle para rastreamento através de servovisão. As implementações destas interfaces são apresentadas nas seções seguintes junto descrição da GUI.

A.3.1 Kinect

O Kinect é um dispositivo composto por uma câmera RGB e uma câmera e emissor infravermelho. Este gera uma nuvem de pontos cuja deformação permite ao dispositivo fornecer informações de profundidade junto a imagem do ambiente, sendo classificado como um dispositivo RGBD (em inglês, *red, green, blue and depth*). O Kinect foi desenvolvido para fornecer interface natural a jogos eletrônicos a partir do rastreamento dos movimentos do usuário. O rastreamento é realizado através de 20 pontos compreendendo: as articulações dos braços e pernas, mãos, pés, cabeça, centro da linha dos ombros, centro de massa do corpo (próximo ao umbigo) e um ponto intermediário entre os dois últimos, como mostrado na Figura A.6. Para estender esta capacidade a plataforma Windows foram utilizados o *driver* e kit de desenvolvimento oficiais da Microsoft, sendo a interface com Matlab realizada por um servidor TCP utilizando o RR.

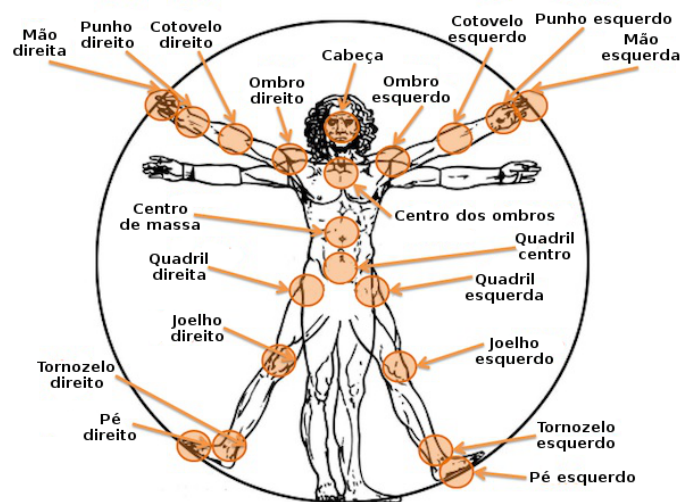


Figura A.6: Pontos do corpo que formam o esqueleto fornecido pelo Kinect.

O servidor fornece, quando requisitado, um vetor de 80 elementos com a posição e a qualidade da identificação de cada um dos 20 pontos, para até dois esqueletos. Para a aplicação atual somente três pontos, mostrados na Figura A.7, referentes ao braço direito são utilizados.

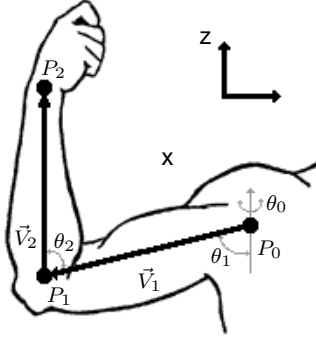


Figura A.7: Pontos (P_0 , P_1 e P_2) do esqueleto fornecido pelo Kinect e utilizados para controle do manipulador. Retirado de (Motta-Ribeiro et al. 2012).

Para movimentar o manipulador são utilizados os ângulos do ombro (θ_0 e θ_1) e do cotovelo (θ_2), onde o braço é modelado por meio dos vetores:

$$\begin{aligned}\vec{V}_1 &= R_z(\theta_0)R_y(\theta_1)(-\vec{y}) \\ \vec{V}_2 &= R_z(\theta_0)R_y(\theta_1)R_y(\theta_2)(-\vec{z})\end{aligned}\tag{A.1}$$

onde $R_z(\theta_0)$ é uma rotação elementar de θ_0 em torno do eixo \vec{z} e $R_y(\theta_i)$ é uma rotação elementar de θ_i em torno do eixo \vec{y} para $i = 1, 2$. Note que, os vetores \vec{V}_1 e \vec{V}_2 , definidos na Figura A.7, podem ser obtidos por cinemática inversa através dos subproblemas 2 e 1 de Paden-Kahan (Murray et al. 1994). Este modelo considera a posição zero do manipulador com o elo 1 alinhado ao eixo $-\vec{y}$ e o elo 2 ao eixo \vec{z} , além de um movimento inverso para os ângulos θ_1 e θ_2 . Esta inversão é justificada para se obter melhor ergonomia em operações com o cotovelo do robô para o alto, por exemplo coletando objetos sobre uma mesa.

A.3.2 Phantom Omni

O Phantom é um mecanismo robótico com 6 juntas de revolução, 3 atuadas e 3 não atuadas (Silva et al. 2009), mostrado na Figura A.8. Possui uma biblioteca de desenvolvimento em C/C++ com a qual é possível programar aplicações gráficas que o utilizam como dispositivo de entrada de 6 graus de mobilidade com realimentação em 3 graus, ou controlar diretamente o *hardware*.

Para integrar o Phantom no sistema, é possível utilizar a capacidade do Matlab de reproduzir funções escritas em linguagem C pré-compiladas em arquivos MEX. Neste caso, a função desenvolvida inicializa o laço de controle do dispositivo e permite a leitura das posições das juntas e a escrita de um vetor de forças de forma assíncrona. Para ajustar a leitura dos ângulos à cinemática de diferentes manipuladores, pode-se realizar rotações e translações independentes para cada junta.



Figura A.8: Phantom Omni com os eixos de rotação indicados. Em verde as juntas atuadas e em vermelho as não atuadas.

Uma segunda opção é utilizar o serviço RR utilizado como exemplo no Apêndice A.1.1 (Código A.1). Com este é possível utilizar um ou mais dispositivos Phantom, possivelmente em computadores diferentes do que roda o servidor da teleoperação, com as mesmas funcionalidades da integração direta ao Matlab.

A.3.3 Servovisão

Servovisão consiste em controlar a posição de um manipulador (ou de um robô móvel) a partir da realimentação visual obtida por um ou mais dispositivos de captura de imagem (Hutchinson et al. 1996). Neste trabalho, propõe-se uma implementação transparente ao operador, o qual somente indica a posição inicial de um marcador colorido (vermelho, verde ou azul) no punho do manipulador e a posição final desejada (alvo).

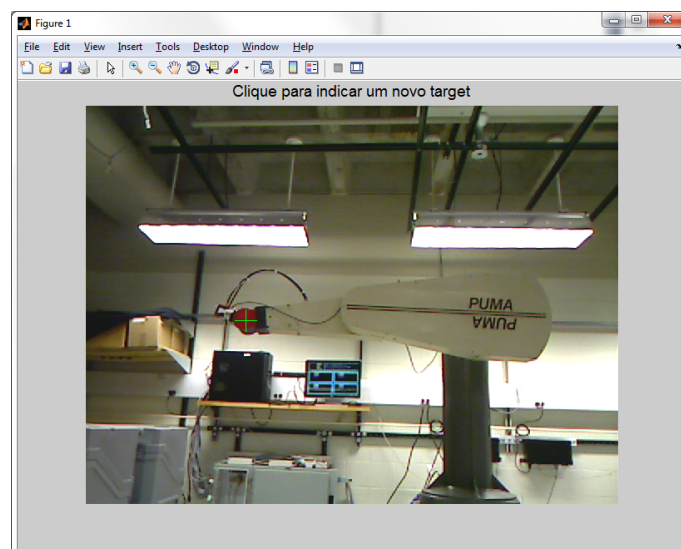


Figura A.9: Exemplo de imagem apresentada para o operador durante controle por servovisão.

A posição inicial é necessária devido a execução do processamento de imagem em uma sub-janela, considerando-se um movimento lento em relação a amostragem das imagens. Este processamento é realizados em cinco etapas:

1. subtração da imagem em tons de cinza da camada da cor do marcador;
2. filtragem através de filtro da mediana;
3. transformação de escala de cinza para preto e branco;
4. remoção de objetos pequenos;
5. cálculo do centroide do objeto identificado.

Uma vez obtido o centroide do marcador no punho, o erro de imagem (em pixels), obtido a partir da diferença entre a posição desejada e a posição do centroide, é transmitido para o ambiente local. Todo processamento é realizado em Matlab fazendo o uso do pacote *Image Processing* através das linhas de código apresentadas no Código A.2.

Código A.2: Processamento da imagem.

```
function [Xc,Yc,falha]=getcentroid(data,Xci,Yci,L)

Xci=round(Xci); Yci=round(Yci);
subwin_x=max(Xci-L,1):min(Xci+L,480);
subwin_y=max(Yci-L,1):min(Yci+L,640);
diff_im=imsubtract(data(subwin_x,subwin_y,1),rgb2gray(data(subwin_x,subwin_y,:)));

diff_im=medfilt2(diff_im,[4 4]);

diff_im=im2bw(diff_im,0.20);

diff_im=bwareaopen(diff_im,30);

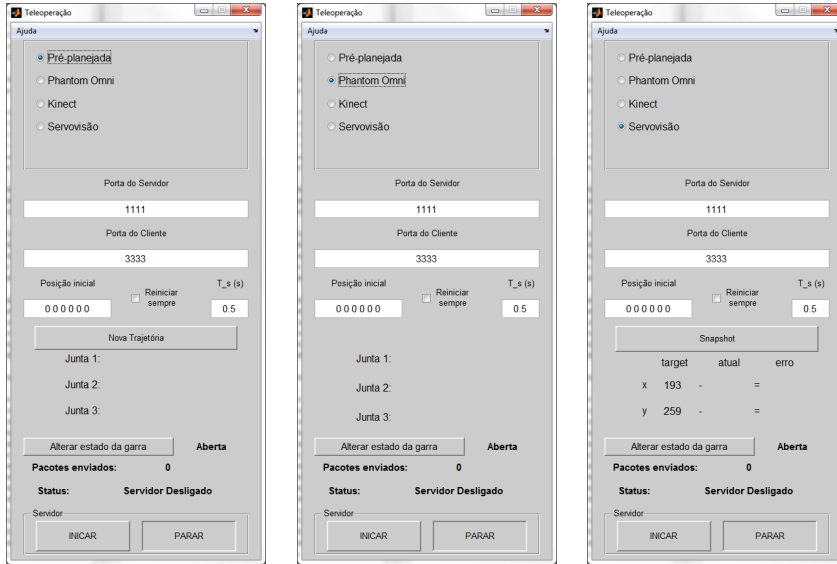
stats=regionprops(diff_im,'Centroid');

for object=1:length(stats)
    bc=stats(object).Centroid;
    bc=bc+[max(Yci-L,1)-1 max(Xci-L,1)-1];
end
if exist('bc','var')
    Xc=round(bc(1)); Yc=round(bc(2));
    falha=0;
else
    falha=1;
    Xc=Yci; Yc=Xci;
end
```

A.3.4 Interface Gráfica

A operação no ambiente remoto é centralizada em uma GUI desenvolvida em Matlab permitindo escolher a modalidade de controle, configurar o servidor e controlar o estado da garra no efetuador, além de fornecer informações para auxiliar o operador. A Figura A.10 apresenta a GUI nas diferentes configurações de acordo com o modo de controle, selecionado na parte superior. Logo abaixo configuram-se as portas do

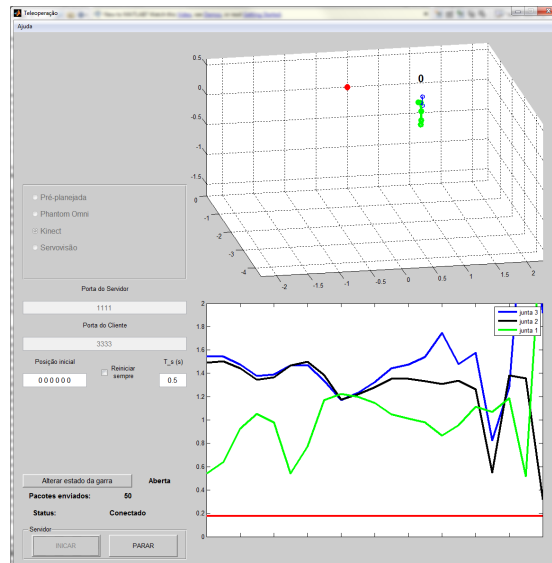
cliente e do servidor, a posição inicial das juntas do manipulador e o período de amostragem. A opção de reiniciar sempre retorna a posição inicial escolhida cada vez que a operação é parada e reiniciada, caso contrário isto ocorre apenas se o servidor for desligado.



(a) Trajetória preestabelecida

(b) Phantom Omni

(c) Servovisão



(d) Kinect

Figura A.10: Interface gráfica nas quatro configurações programadas.

Na região inferior da interface estão o controle da garra, uma indicação do estado da mesma, informações sobre as mensagens enviadas, o estado do servidor (desligado, esperando conexão ou conectado) e os botões para ligar e desligar o mesmo. A região central apresenta configurações e informações referentes à modalidade de controle selecionada, sendo alterada de acordo com cada uma. Na Figura A.10(a) é apresentada a interface para o envio de uma trajetória preestabelecida e, neste caso,

existe um botão para alterar a trajetória de cada junta e a cada instante é apresentado ao operador o valor do ângulo enviado. Para a operação com Phantom (ver Figura A.10(b)) o painel mostra somente os ângulos de cada junta e para a servovisão o painel apresenta as posição do alvo e do marcador no punho do manipulador além do erro de imagem em pixels, como representado na Figura A.10(c).

Para a servovisão também é apresentada uma figura com uma foto do ambiente local adquirida a partir da requisição do operador pelo botão *snapshot* ou ao se iniciar o procedimento. É através desta figura que a posição inicial do marcador no punho e o alvo são informados. As coordenadas do alvo podem ser alteradas a qualquer instante de tempo bastando selecionar um novo local da imagem, enquanto o marcador deve ser realocado apenas quando o processamento da imagem não o faz automaticamente.

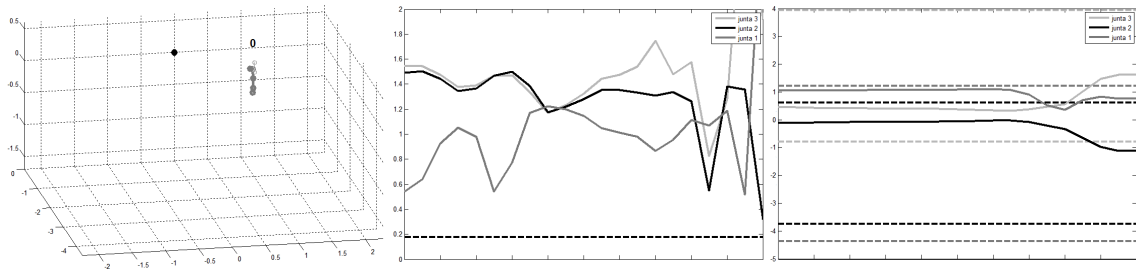


Figura A.11: Exemplos de figuras para auxílio do operador durante o uso do Kinect. Painel esquerdo - reconstrução do braço através dos pontos fornecidos pelo dispositivo; painel central - diferença entre ângulos das juntas no Kinect e no manipulador antes de iniciar a operação; painel direito - ângulos enviados ao manipulador (linhas sólidas) e limites (linhas tracejadas) para cada junta. Nos dois últimos os eixos são radianos e amostras. Retirado de (Motta-Ribeiro et al. 2012)

Na operação através do Kinect nenhuma informação é mostrada no painel principal e o operador é realimentado por duas figuras, como no exmplo da Figura A.10(d). A primeira apresenta uma reconstrução do braço através dos pontos de esqueleto recebidos do dispositivo. A segunda varia de acordo com o momento de operação: ao ser iniciada apresenta a diferença absoluta em relação a última posição (ou a posição inicial informada na interface). Após esta diferença atingir o limite de 10° em todas as juntas, a figura é trocada para o ângulo e os limites, superior e inferior, de cada uma. Exemplos destas figuras são apresentados na Figura A.11, da esquerda para direita.

A.4 Testes Experimentais

Para o teste experimental da arquitetura proposta o ambiente de operação local utilizado pertence ao Rensselaer Polytechnic Institute (Troy, NY, EUA). O ambiente

experimental é composto por dois manipuladores robóticos PUMA 560 montados sobre um sistema de trilho contendo uma junta prismática e duas de revolução (Wason & Wen 2011), mostrado na Figura A.12. Entretanto, nos ensaios experimentais será utilizado apenas um dos manipuladores com a base fixa, uma câmera CCD e um sensor de força JR3 (JR3 Inc.). Para avaliar a viabilidade da estrutura estabelecida, duas tarefas teste foram desenvolvidas, uma abrangendo os três métodos de comando do manipulador e a outra para a realimentação de força.



Figura A.12: Sistema de dois robôs PUMA560 sobre trilhos do CATS/RPI.

A primeira tarefa teste consiste em segurar um objeto alvo sobre um suporte em uma mesa e deixa-lo dentro de uma cesta (*pick-and-place*). O manipulador é inicializado em uma posição esticada para frente, com vetor de ângulos das juntas dado por $\theta = [0 \ 0 \ \frac{\pi}{2} \ 0 \ \frac{\pi}{3} \ 0]^T rad$. Em seguida ele é alinhado ao objeto alvo em um posicionamento lateral com $\theta_1 = -\frac{\pi}{2} rad$ de forma a obter o marcador do punho no quadro da câmera. Então, o manipulador é levado até o objeto e, após agarrar o mesmo, o caminho inverso é realizado. Os grandes deslocamentos são realizados através do Phantom ou Kinect, enquanto o posicionamento fino utiliza a servovisão. As etapas estão exemplificadas na Figura A.13 e os números representam as ações enumeradas no início deste capítulo.

Durante o experimento a etapa de servovisão foi realizada através de um algoritmo iterativo do tipo *look-and-move* (Hutchinson et al. 1996). O robô é controlado no espaço das juntas, em uma malha de posição com a referência (q_d) dada por:

$$q_d(k) = q(k-1) + K J_p^{-1}(q(k-1))\epsilon(k), \quad (A.2)$$

sendo J_p o Jacobiano de posição do manipulador, K uma matriz de ganhos positivos, q a posição atual das juntas e ϵ o erro de imagem. Neste caso se o plano de trabalho

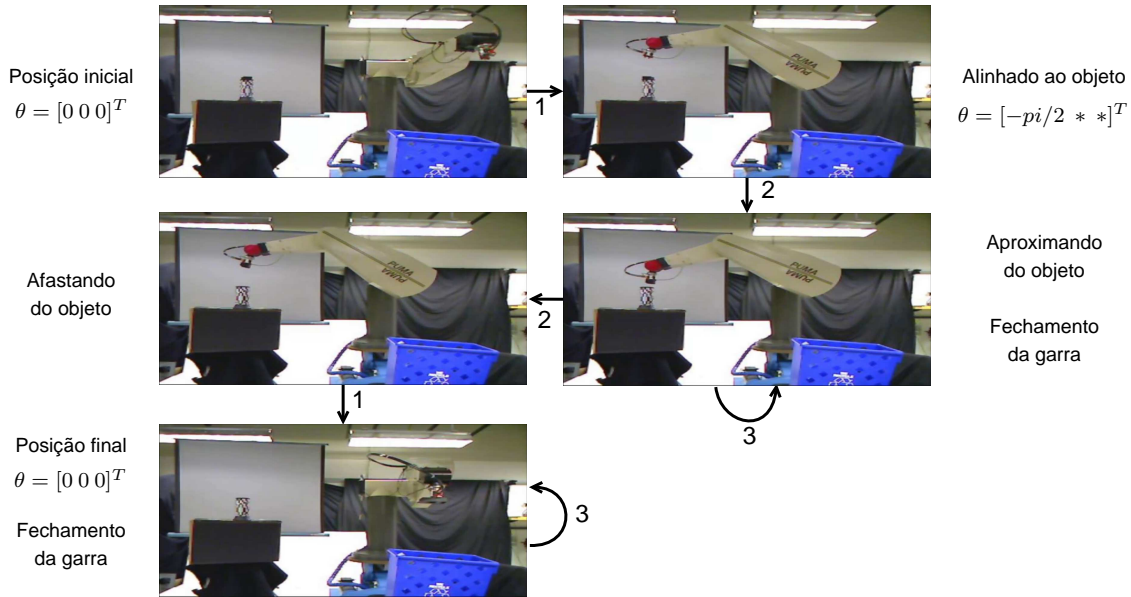


Figura A.13: Exemplo do experimento com as diferentes modalidades de controle. Adaptado de (Motta-Ribeiro et al. 2012)

e o plano da câmera são paralelos com uma rotação ϕ , $|\phi| < \frac{\pi}{2}$, é possível garantir que o erro na imagem converge para zero (Zachi, Hsu, Lizarralde & Leite 2006).

A tarefa referente a realimentação de força é dividida em duas etapas: regulação e rastreamento. Em ambas, um operador local atua diretamente no sensor de força instalado no efetuador do manipulador e as leituras deste são enviadas ao ambiente remoto servindo como entrada de força para o Phantom, cujos ângulos das juntas são enviados para manipulador local. Na primeira etapa o operador remoto deve manter a posição do dispositivo fixa, enquanto na segunda o dispositivo pode se movimentar, resultando em um movimento do manipulador.

A.5 Observações dos Experimentos

A seguir são apresentados resultados para avaliação da eficácia da estrutura proposta. Primeiramente, destaca-se que a estrutura utilizada permitiu a realização da tarefa de *pick and place*², porém devido ao grande atraso (próximo a 1 s), a baixa taxa de amostragem (0,5 s) e a operação unilateral, somente com realimentação visual, os movimentos devem ser realizados em baixa velocidade. A influência do atraso no tempo de execução da tarefa de teleoperação na modalidade unilateral é conhecida desde o trabalho de Sheridan & Ferrell (1963) mostrando que o operador adota uma estratégia de movimentos pausados, esperando pela reação do manipulador, quando há atraso na comunicação.

²Um vídeo com a tarefa de *pick and place* pode ser visto em http://www.youtube.com/watch?v=J5vRgD_kieY.

Na Figura A.14 são apresentados os ângulos enviados para o ambiente local durante a movimentação mestre/escravo, tanto com Phantom quanto com o Kinect. Observa-se que inicialmente os valores são constantes durante o período em que o operador ajusta o controlador a posição atual do manipulador, de forma a não ocorrerem saltos na posição das juntas. Após este ajuste a junta 1 é levada a posição $-\frac{\pi}{2} rad$ durante a etapa inicial e retorna próximo a zero na etapa final, assim como definido na tarefa. Ressalta-se que a operação com o Kinect se mostrou mais difícil, com maiores oscilações e tempo de execução, pois o PUMA 560 não possui todos os graus de mobilidade do braço humano tornando o movimento pouco natural, com o operador virando todo o corpo e dificultando a visualização da tela de operação. Além disso, apesar das inversões consideradas na cinemática inversa tornarem mais ergonômica a posição de cotovelo do manipulador para cima, elas tornam também os movimentos do braço menos intuitivos.

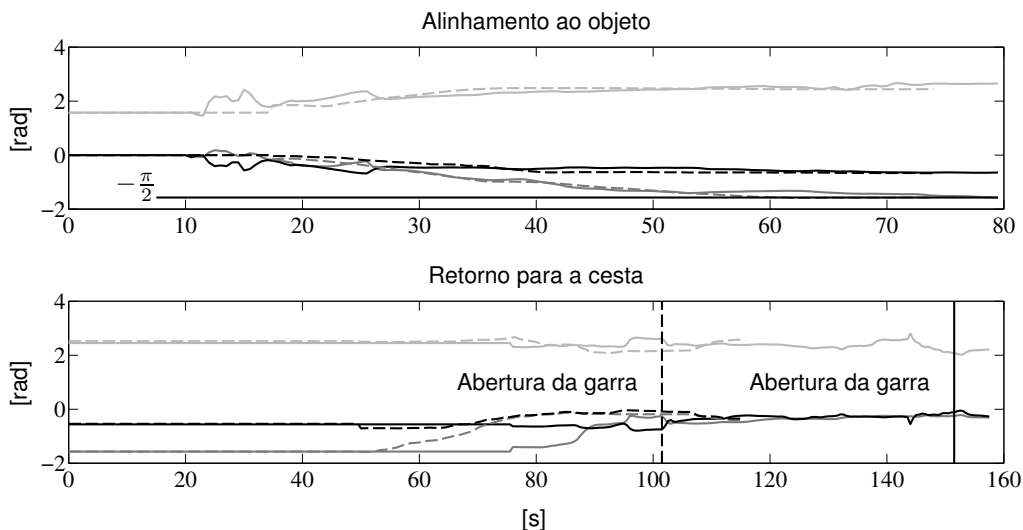


Figura A.14: Trajetórias da juntas 1 (cinza escuro), 2 (preto) e 3 (cinza claro) durante a movimentação mestre escravo com o Kinect (contínuo) e Phantom (traçado). Retirado de (Motta-Ribeiro et al. 2012)

Durante a servovisão, devido ao procedimento de obtenção da imagem pelo nó Matlab do RR, o intervalo de tempo entre o envio de informações para o ambiente local aumenta, podendo aproximar-se de 3 s. Com isso o deslocamento do manipulador deve ser realizado com velocidade ainda menor de forma a manter o marcador dentro da sub-janela de processamento da imagem. Este atraso na atualização do erro, porém, não inviabiliza o controle como pode ser observado na Figura A.15. A aproximação é realizada de forma gradual, enquanto a execução do trajeto de volta, após o fechamento da garra, ocorre em uma etapa única. Uma alternativa para aumentar a velocidade do controlador é realizar o processamento local, com o operador remoto capturando a imagem somente para indicar o marcador e o alvo e sendo realimentado pela informação de erro de imagem.

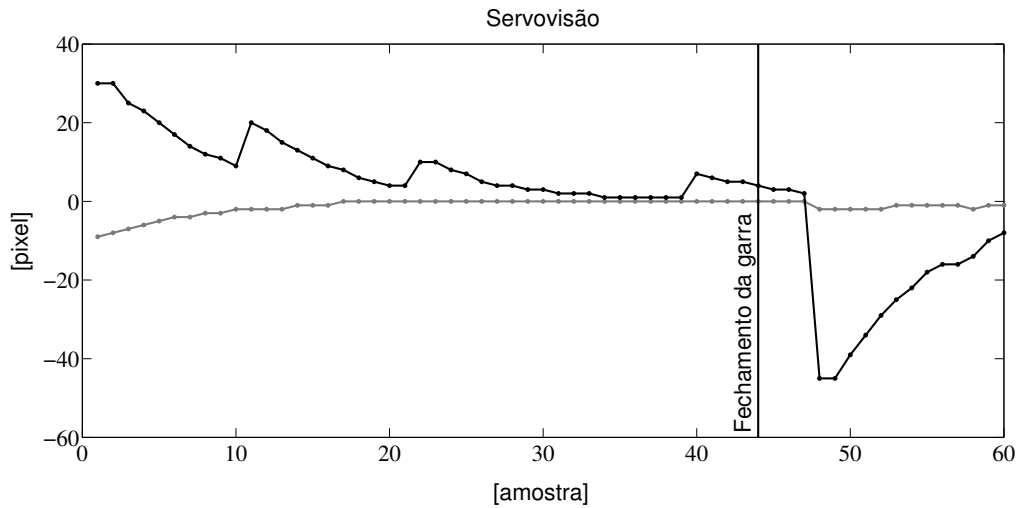


Figura A.15: Erro em pixels nos eixos x (cinza) e y (preto) durante a servovisão. Retirado de (Motta-Ribeiro et al. 2012)

Como a estratégia de servovisão adotada considera somente o movimento planar, ao fim da etapa inicial de movimentação mestre/escravo o manipulador deve estar alinhado ao objeto. Este alinhamento, porém, é dificultado pela falta da informação de profundidade na realimentação visual do operador remoto. A profundidade da imagem pode ser acrescentada através de uma visão não-frontal a partir de outras câmeras (Bambang 2007), de duas câmeras realizando estereoscopia (Hutchinson et al. 1996) ou utilizando a medida da área do marcador no punho (Zachi, Hsu, Lizarralde & Leite 2006), entre outros.

O teste de realimentação de força mostrou que é possível reproduzir no Phantom as forças medidas no efetuador do manipulador. Além disso, observa-se que devido a pequena amplitude do movimento na etapa de regulação, a operação é estável mesmo com o atraso na comunicação, o mesmo não ocorrendo durante o rastreamento.

Apêndice B

Teoremas auxiliares

Neste apêndice são apresentadas as provas para os resultados dos Teoremas 1 e 4, bem como o Lema 4.3 de (Khalil 2002) utilizado na prova dos Teoremas 2 e 3.

B.1 Prova do Teorema 1

A demonstração a seguir para o Teorema 1 é uma adaptação da prova original para o teorema 4.14 de (Haddad & Chellaboina 2011).

Prova. Primeiramente, considere $\delta \in (0, \mu]$ e assumamos $\|y_{10}\| \leq \delta$. Se $\|y_1(t)\| \leq \mu$, para $t \geq 0$, então de (4.9) segue que $\|y_1(t)\| \leq \mu \leq \alpha^{-1}(\beta(\mu)) \leq \alpha^{-1}(\eta) = \epsilon$, para $t \geq 0$.

Considere então que $\delta \in (\mu, \gamma)$, sendo $\gamma := \sup\{r > 0 : \mathcal{B}_{\alpha^{-1}(\beta(r))}(0) \subset \mathcal{D}\}$, e assumamos que $y_{10} \in \mathcal{B}_\delta(0)$ e $\|y_{10}\| > \mu$. Agora, para todo $\hat{t} > 0$ tal que $\|y_1(t)\| \geq \mu$, para $t \in [0, \hat{t}]$, segue de (4.9) and (4.10) que

$$\alpha(\|y_1(t)\|) \leq V(y_1(t), y_2(t)) \leq V(x_{10}, x_{20}) \leq \beta(\delta), \quad t \geq 0,$$

que implica em $\|y_1(t)\| \leq \alpha^{-1}(\beta(\delta))$, para $t \in [0, \hat{t}]$. Se, ainda, existe $T > 0$ tal que $\|y_1(t)\| \leq \mu$, então segue como no primeiro caso que $\|y_1(t)\| \leq \alpha^{-1}(\eta)$, para $t \geq T$. Consequentemente, se $y_{10} \in \mathcal{B}_\delta(0) \setminus \mathcal{B}_\mu(0)$, então $\|y_1(t)\| \leq \alpha^{-1}(\beta(\delta))$, para $t \geq 0$.

Suponha, *ad absurdum*, que $\|y_1(t)\| \geq \beta^{-1}(\eta)$, para $t \geq 0$, ou, de forma equivalente, $y_1(t) \in \mathcal{O} := \mathcal{B}_{\alpha^{-1}(\beta(\delta))}(0) \setminus \mathcal{B}_{\beta^{-1}(\eta)}(0)$, para $t \geq 0$. Como $\overline{\mathcal{O}}$ é compacto, $W(\cdot)$ é contínuo e $W(y_1) > 0$, para $\|y_1\| \geq \beta^{-1}(\eta) > \mu$, existe $k := \min_{y_1 \in \overline{\mathcal{O}}} W(y_1) > 0$. Assim, segue de (4.10) que

$$V(y_1(t), y_2(t)) \leq V(y_{10}, y_{20}) - kt, \quad t \geq 0$$

o que implica em

$$\alpha(\|y_1(t)\|) \leq \beta(\|y_{10}\|) - kt \leq \beta(\delta) - kt, \quad t \geq 0.$$

Agora, considerando $t > \beta(\delta)/k$ tem-se $\alpha(\|y_1(t)\|) < 0$, o que é uma contradição. Desta forma, existe $T = T(\delta, \eta) > 0$ tal que $\|y_1(T)\| < \beta^{-1}(\eta)$. Como no primeiro caso, por (4.9) tem-se $\|y_1(t)\| \leq \alpha^{-1}(\beta(\beta^{-1}(\eta))) = \alpha^{-1}(\eta)$, $t \geq T$. Portanto, o sistema não linear de (4.7) e (4.8) é UB com respeito a y_1 uniformemente em y_2 com limite $\epsilon = \alpha^{-1}(\eta)$. Ademais, $\limsup_{t \rightarrow \infty} \|y_1(t)\| \leq \alpha^{-1}(\beta(\mu))$. Finalmente, se $\mathcal{D} = \mathbb{R}^{n_1}$ e $\alpha(\cdot)$ é uma função de classe \mathcal{K}_∞ segue que $\beta(\cdot)$ é uma função de classe \mathcal{K}_∞ , e então, $\gamma = \infty$. Desta forma, o sistema não linear de (4.7) e (4.8) é globalmente UB com respeito a y_1 uniformemente em y_2 com limite ϵ . \square

B.2 Lema 4.3 de Khalil (2002)

As prova dos Teoremas 2 e 3 utilizam o resultado do Lema 4.3 de (Khalil 2002), assim este é aqui transcrito para completude do trabalho.

Lema. *Considere $V : D \rightarrow \mathbb{R}$ uma função positiva definida no domínio $D \subset \mathbb{R}^n$ que contém a origem e $B_r \subset D$ para algum $r > 0$. Então, existem funções α_1 e α_2 da classe \mathcal{K} , definidas em $[0, r]$, de tal modo que*

$$\alpha_1(\|x\|) \leq V(x) \leq \alpha_2(\|x\|)$$

para todo $x \in B_r$. Se $D = \mathbb{R}^n$ e $V(x)$ é radialmente ilimitada, então existem funções α_1 and α_2 de classe \mathcal{K}_∞ de tal forma que a inequação a cima é válida para todo $x \in \mathbb{R}^n$.

Prova. Seja $\psi(s)$, definida como,

$$\psi(s) = \inf_{s \leq \|x\| \leq r} V(x) \quad \text{for } 0 \leq s \leq r$$

uma função continua, definida positiva e crescente (mas não necessariamente estritamente crescente). Ademais, $V(x) \geq \psi(\|x\|)$ para $0 \leq \|x\| \leq r$. Considere a função $\alpha_1(s)$ de classe \mathcal{K} , tal que $\alpha_1(s) \leq k\psi(s)$ com $0 < k < 1$. Então,

$$V(x) \geq \psi(\|x\|) \geq \alpha_1(\|x\|) \quad \text{para } \|x\| \leq r$$

Por outro lado, defina $\phi(s)$ como

$$\phi(s) = \sup_{\|x\| \leq s} V(x) \quad \text{for } 0 \leq s \leq r$$

um função continua, definida positiva e crescente (mas não necessariamente estritamente crescente). Ademais, $V(x) \leq \phi(\|x\|)$ para $\|x\| \leq r$. Considere $\alpha_2(s)$ uma função de classe \mathcal{K} , tal que $\alpha_2(s) \geq k\phi(s)$ com $k > 1$. Então

$$V(x) \leq \phi(\|x\|) \leq \alpha_2(\|x\|) \quad \text{para } \|x\| \leq r$$

Se $D = \mathbb{R}^n$, as definições de $\psi(s)$ e $\phi(s)$ são alteradas para

$$\psi(s) = \inf_{\|x\| \geq s} V(x), \quad \phi(s) = \sup_{\|x\| \leq s} V(x), \quad \text{para } s \geq 0$$

Ambas são continuas, definidas positiva, crescentes, e

$$\psi(\|x\|) \leq V(x) \leq \phi(\|x\|), \quad \forall x \in \mathbb{R}^n$$

As funções α_1 e α_2 podem ser escolhidas como antes. Se $V(x)$ é radialmente ilimitada, $\psi(s)$ e $\phi(s)$ tendem ao infinito quando $s \rightarrow \infty$. Deste modo, pode-se escolher α_1 e α_2 como funções de classe \mathcal{K}_∞ . \square

B.3 Prova do Teorema 4

A demonstração a seguir para o Teorema 4 é semelhante a prova presente em (Hsu et al. 2007).

Prova. Considere a função positiva definida

$$V(\chi) = V_2(\chi_2) + \alpha V_1(\chi_1)$$

A sua derivada temporal ao longo das trajetórias do sistema

$$\dot{V} \leq -\lambda_2 \|\varphi_2\|^2 - \alpha \lambda_1 \|\varphi_1\|^2 + \lambda_1 \|\varphi_1\| \|\varphi_2\| \leq 0$$

é negativa definida com respeito a φ_1, φ_2 para um α suficientemente pequeno (pelo complemento de Schur). Então, $\chi_1, \chi_2 \in L_\infty$ e $\dot{\chi}_1, \dot{\chi}_2 \in L_\infty$.

A desigualdade a cima implica em $\varphi_1, \varphi_2 \in L_2 \cap L_\infty$ e, partir do lema de Barbalat (Khalil 2002), $\varphi_1, \varphi_2 \rightarrow 0$ as $t \rightarrow \infty$. \square

Apêndice C

Manual de utilização - Motoman DIA10

Sumário

C.1	Características cinemáticas	110
C.2	Instalação	112
C.3	Controle via <i>Teach Pendant</i>	114
C.4	Controle via computador	115
C.4.1	HSC	116
C.4.2	Robot Raconteur	117
C.5	Operação	118
C.5.1	Inicialização	118
C.5.2	Conexão com computador	121

Este manual tem o intuito de ser um guia para conhecer e operar o Motman DIA10. Algumas situações diferentes da operação padrão são apresentadas, entretanto, não substitui completamente os manuais fornecidos pelo fabricante. Estes devem ser consultados caso uma situação aqui não abordada ocorra. O mesmo pode ser dito sobre o controlador de alta velocidade (HSC), cujo funcionamento aqui também é resumido. Sugere-se manter uma cópia deste no local de operação para que seja utilizado para consultas rápidas.

C.1 Características cinemáticas

O Motoman DIA10 é um robô constituído de dois manipuladores antropomórficos (7 DoF) e um tronco, resultando em 15 DoF, como mostrado no desenho da Figura C.1. Cada um dos braços suporta uma carga de 10 kg no efetuador. Todas as juntas são de revolução e controladas de forma independente.

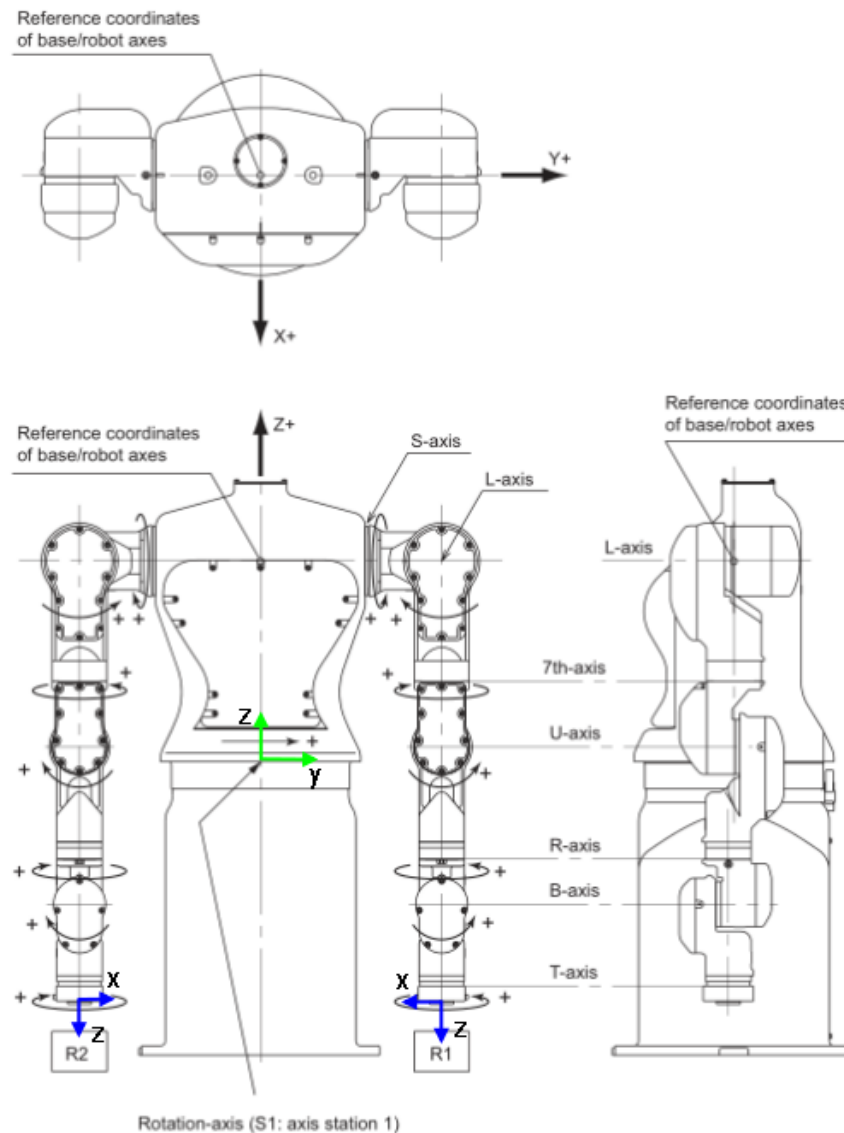


Figura C.1: Nomes das juntas e sistema de referência do robô DIA10. O eixo verde representa a referência e os azuis os sistemas de coordenadas finais para a cinemática direta via Denavit-Hartenberg

A partir das dimensões dos elos, na Figura C.2, e dos sentidos de rotação das juntas, apresentados na Figura C.1, pode-se levantar o modelo cinemático de cada um dos braços. Para tal, considera-se como posição inicial o robô com os dois braços abertos, semelhante ao padrão do NX100. Na Tabela C.1 são apresentados os parâmetros de Denavit-Hartenberg.

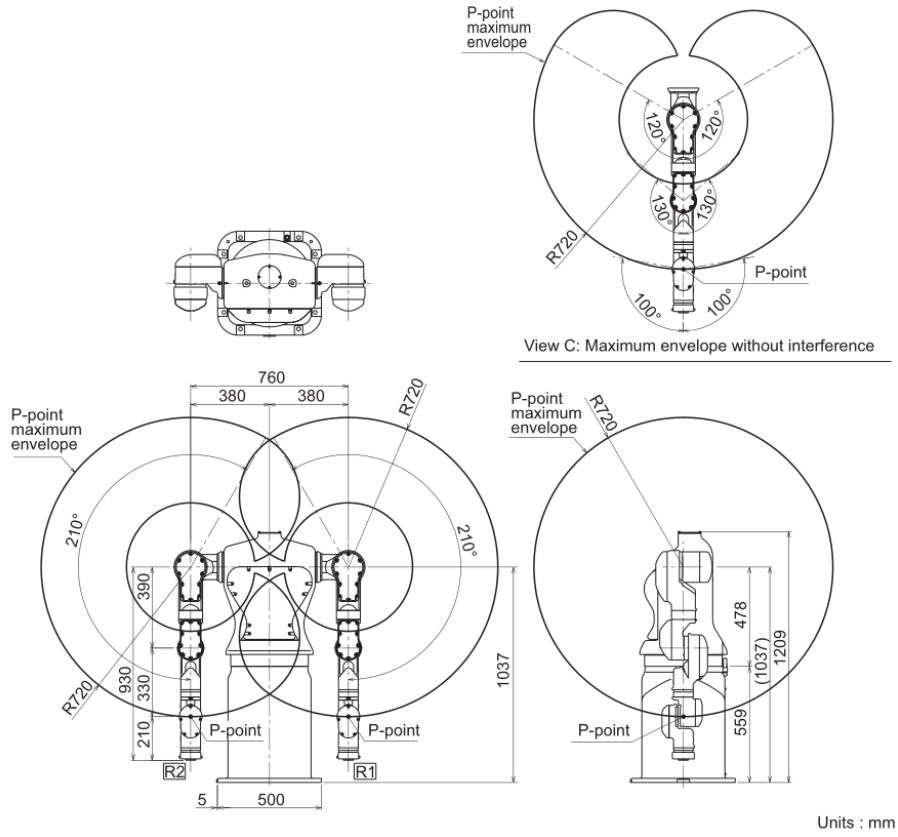


Figura C.2: Dimensões dos elos e limites de rotação das juntas do robô DIA10.

Tabela C.1: Parâmetros de Denavit-Hartenberg *standard* para o DIA10. Esta representação considera como posição inicial o robô com os dois braços abertos, semelhante ao padrão do NX100.

Junta	R1				R2			
	α	a	θ	d	α	a	θ	d
Tronco	$-\frac{\pi}{2}$	0.0	0.0	0.478	$-\frac{\pi}{2}$	0.0	0.0	0.478
S	$-\frac{\pi}{2}$	0.0	0.0	0.380	$\frac{\pi}{2}$	0.0	0.0	-0.380
L	$-\frac{\pi}{2}$	0.0	0.0	0.000	$-\frac{\pi}{2}$	0.0	0.0	0.000
7	$-\frac{\pi}{2}$	0.0	0.0	-0.390	$-\frac{\pi}{2}$	0.0	0.0	-0.390
U	$-\frac{\pi}{2}$	0.0	0.0	0.000	$-\frac{\pi}{2}$	0.0	0.0	0.000
R	$\frac{\pi}{2}$	0.0	0.0	0.330	$\frac{\pi}{2}$	0.0	0.0	0.330
B	$-\frac{\pi}{2}$	0.0	0.0	0.000	$-\frac{\pi}{2}$	0.0	0.0	0.000
T	0.0	0.0	0.0	0.210	0.0	0.0	0.0	0.210

A representação cinemática dos braços segundo a teoria de *Screw* é apresentada na Tabela C.2.

Tabela C.2: Parâmetros cinemáticos segundo a teoria de *Screw* para o DIA10.

Junta	R1			R2		
	$\vec{\omega}$	\vec{q}		$\vec{\omega}$	\vec{q}	
Tronco	$\begin{bmatrix} 0 & 0 & 1 \end{bmatrix}$	0	0.000	0.478	$\begin{bmatrix} 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 0 & 0.000 & 0.478 \end{bmatrix}$
S	$\begin{bmatrix} 0 & 1 & 0 \end{bmatrix}$	0	0.380	0.478	$\begin{bmatrix} 0 & 1 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & -0.380 & 0.478 \end{bmatrix}$
L	$\begin{bmatrix} 0 & 0 & -1 \end{bmatrix}$	0	0.000	0.478	$\begin{bmatrix} 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 0 & 0.000 & 0.478 \end{bmatrix}$
7	$\begin{bmatrix} 0 & -1 & 0 \end{bmatrix}$	0	0.770	0.478	$\begin{bmatrix} 0 & 1 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & -0.770 & 0.478 \end{bmatrix}$
U	$\begin{bmatrix} 0 & 0 & 1 \end{bmatrix}$	0	0.000	0.478	$\begin{bmatrix} 0 & 0 & -1 \end{bmatrix}$	$\begin{bmatrix} 0 & 0.000 & 0.478 \end{bmatrix}$
R	$\begin{bmatrix} 0 & 1 & 0 \end{bmatrix}$	0	1.100	0.478	$\begin{bmatrix} 0 & -1 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & -1.100 & 0.478 \end{bmatrix}$
B	$\begin{bmatrix} 0 & 0 & 1 \end{bmatrix}$	0	0.000	0.478	$\begin{bmatrix} 0 & 0 & -1 \end{bmatrix}$	$\begin{bmatrix} 0 & 0.000 & 0.478 \end{bmatrix}$
T	$\begin{bmatrix} 0 & 1 & 0 \end{bmatrix}$	0	1.310	0.478	$\begin{bmatrix} 0 & -1 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & -1.320 & 0.478 \end{bmatrix}$

C.2 Instalação

Para utilizar o Motoman DIA10 é necessário conectar-lo ao controlador industrial NX100, mostrado na Figura C.3, onde se encontram os *drivers* dos motores.

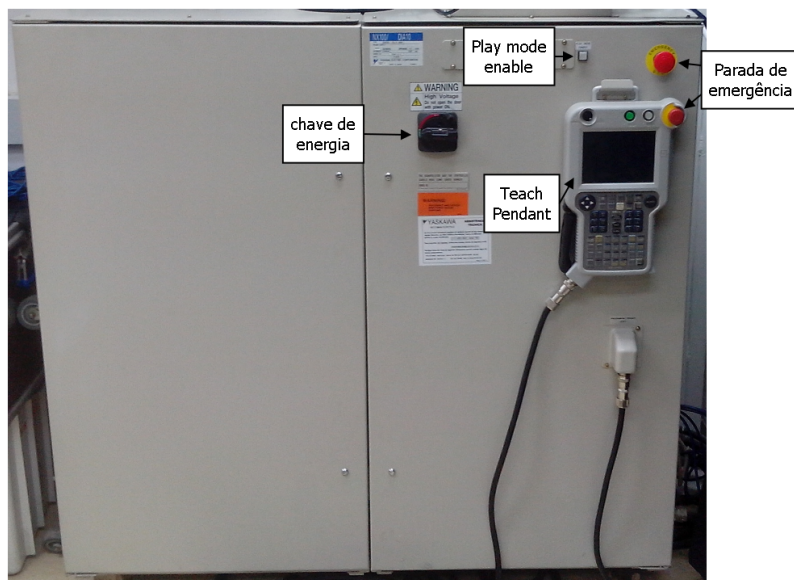


Figura C.3: Controlador industrial NX100.

A conexão é realizada através de três cabos de potência e dois para leitura de *encoders*. Os cabos são identificados nas duas extremidades, devendo ser conectados de acordo com esta identificação, como representado no diagrama da Figura C.4. Os conectores estão localizados na parte inferior traseira da base do robô e na lateral direita do NX100.

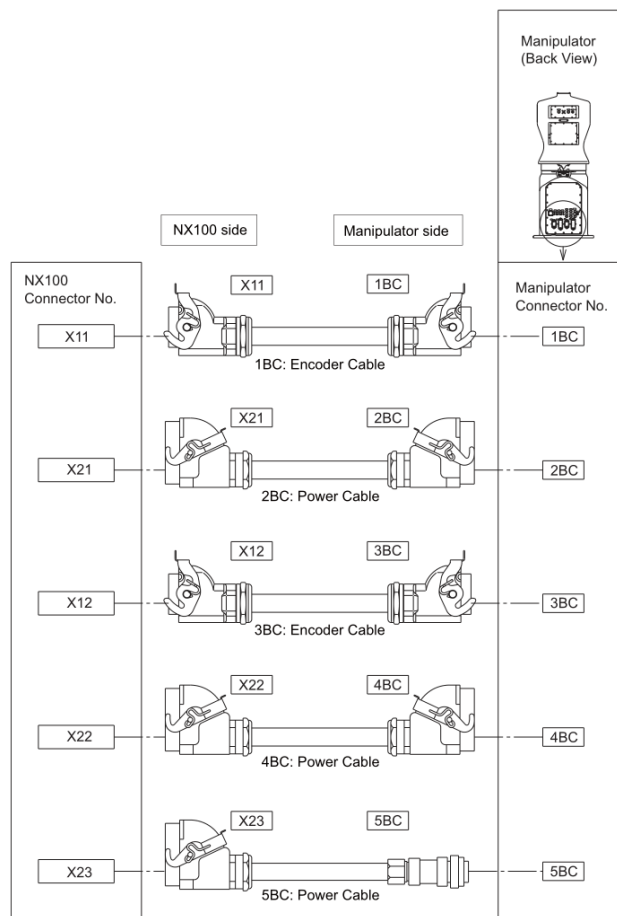


Figura C.4: Diagrama para conexão dos cabos entre o robô e o controlador NX100.

O NX100 é alimentado através de uma tomada industrial ligada a rede elétrica com 248 V, 4,5 kVA trifásico. Recomenda-se realizar a conexão dos cabos entre o robô e o controlador com este desenergizado. Para configurar o NX100 e controlar o DIA10 utiliza-se o *Teach Pendant*, mostrado na Figura C.3 e descrito na próxima seção. Para instalar este dispositivo basta conectar-lo a entrada na frente do controlador, o cabo para o *Teach Pendant* é o único com dois conectores redondos.

Além do conector do *Teach Pendant*, a parte da frente do NX100 contém a chave para para ligar-lo e mais dois botões:

Play mode enable - utilizado para colocar o controlador no modo automático, onde repete movimentos pré-programados ou aceita comandos externos de um computador via HSC (ver Apêndices C.3 e C.4).

Parada de emergência - desliga instantaneamente os servomotores fazendo com que o robô pare. Este botão fica travado impedindo qualquer novo movimento até que seja rotacionado em sentido horário.

C.3 Controle via *Teach Pendant*

O *Teach Pendant* é o dispositivo utilizado para configuração do NX100 e controle do robô ligado a este. É possível movimentar as juntas individualmente ou programar movimentos para serem reproduzidos repetidamente. A Figura C.5 mostra o *Teach Pendant* discriminando suas principais partes, apresentadas a seguir.

Chave seletora de modo - utilizada para alterar entre o modo de programação (*Teach*) e o modo de reprodução dos programas (*Play*). O primeiro modo também permite que as juntas sejam posicionadas individualmente, ou retornem para a posição inicial conjuntamente. O segundo modo permite o controle pelo computador via HSC (ver Apêndice C.4).

Start - inicia a reprodução de um programa no modo *Play*.

Parada de emergência - desliga instantaneamente os servomotores fazendo com que o robô pare. Este botão fica travado impedindo qualquer novo movimento até que seja rotacionado em sentido horário.

Servo on Ready - coloca os servomotores em estado de espera no modo *Teach* ou ligado no modo *Play*.

Page - alterna entre diferentes páginas de determinadas funções do *Teach Pendant*.

Select - utilizado para selecionar diferentes opções no menu apresentado na tela, a qual também é sensível ao toque.

Cancel - cancela ações ou mensagens de erro.

Comando de juntas - utilizados para movimentar as juntas individualmente. Cada junta possui um botão para deslocamento no sentido positivo e negativo,

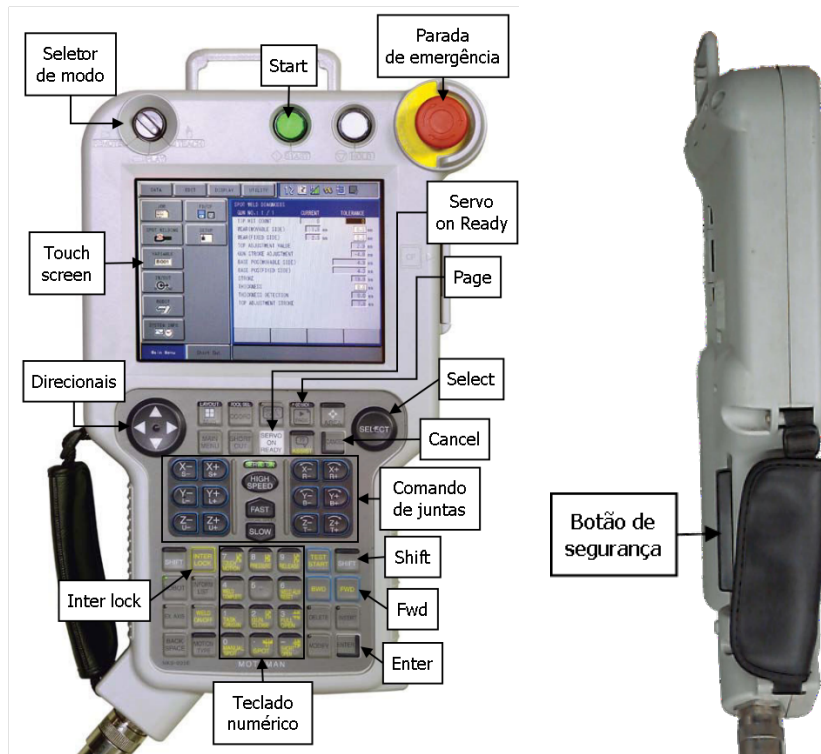


Figura C.5: *Teach Pendant*.

começando no canto superior esquerdo do quadro está a S seguindo para baixo até a T. Para a junta 7 é preciso manter pressionado *Shift* e acionar a S. Entre estes botões estão o controle de velocidade e o led indicador de estado dos servomotores.

Enter - utilizado para confirmar determinadas ações.

Inter lock - utilizado em conjunto com *Select* e o botão de segurança no procedimento de liberação dos freios.

Direcionais - navega entre diferentes opções nos menus da tela.

Botão de segurança - utilizado para acionar os servomotores no modo *Teach*, quando acionado até a metade. Igual a parada de emergência se acionado completamente, sem o travamento.

C.4 Controle via computador

Para comandar o DIA10 utilizando um computador é necessário conectar o NX100 a um controlador de alta velocidade (HSC, na sigla em inglês). Este é responsável por conciliar a malha de tempo real do controlador industrial e a malha de controle implementada no computador. Adicionalmente neste manual será considerado o uso do Robot Raconteur (RR) para disponibilizar o acesso ao robô por qual quer computador em uma mesma rede local.

C.4.1 HSC

O HSC conecta o NX100 e um computador através de uma rede industrial Star Fabric. Esta utiliza cabeamento Ethernet comum com duas vias, uma para transmissão e outra para recepção, necessitando de uma placa de rede especial. Na Figura C.6 é mostrado um diagrama da conexão entre os dispositivos, a ponta da seta indica o sentido da envio dos pacotes. Nota-se que no NX100 existem duas placas de rede, uma para cada braço do robô, mas não há uma correspondência específica nas porta do HSC. Respeitando, evidentemente, que o RX e TX de uma placa estejam ligados no mesmo Link.

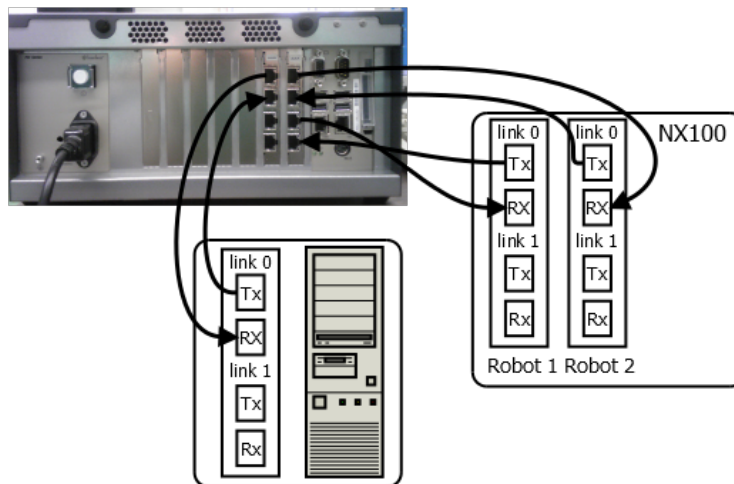


Figura C.6: Diagrama de conexão entre o HSC, o controlador NX100 e um computador.

OBSERVAÇÃO - A placa de rede Star Fabric possui suporte somente para sistemas operacional Linux e Windows XP.

O HSC estabelece um sistema semelhante a memória compartilhada permitindo que o NX100 e o computador acessem as mesmas informações, mesmo com temporizações distintas. Os dados acessados e enviados pelo computador estão dispostos na Tabela C.3. Variáveis terminadas em `_st` representam informações de uma possível estação, com até 2 DoF, na qual o robô está montado.

Como visto na Tabela C.3 o controlador NX100 considera as posições, atual e desejada, das juntas em pulsos do *encoder*. A conversão de pulso para graus é obtida através de um parâmetro próprio de cada junta. Na estrutura de controle considerada neste manual, essa conversão é realizada diretamente pra radianos em uma biblioteca de comunicação com o HSC escrita segundo as recomendações da Motoman. Essa biblioteca é acessada por qualquer computador de uma rede utilizando um programa em RR, apresentado na seção seguinte.

Tabela C.3: Dados compartilhados entre o computador e o NX100 via HSC.

	Descrição	Variável (tipo nome)	Unidade
<i>read</i> HSC → PC HSC2PC[2]	Contador síncrono	ulong sync_cnt	-
	Estado (robô)	ulong system_status	Bit
	Posição	long cur_pls[8] long cur_pls_st[2]	Pulsos Pulsos
	Torque comandado	long trq_ref_data[8] long trq_ref_data_st[2]	% máx. torque % máx. torque
	Velocidade	long spd_fb[8] long spd_fb_st[2]	Pulsos/s Pulsos/s
	Corrente (d)	long a_fb_d[8] long a_fb_d_st[2]	Graus (0 = 90°) Graus (0 = 90°)
	Corrente (q)	long a_fb_q[8] long a_fb_q_st[2]	% corrente nom. % corrente nom.
	Tool number	long tool_no	- (0 - 23)
	Grupo de junta ativas (JOB)	long grp	-
	Informações do comando (JOB)	ushort data[11]	-
<i>write</i> PC → HSC PC2HSC[2]	Contador síncrono	ulong sync_cnt	(não usado)
	Estado (PC)	ulong status	Bit
	Δposição	long seg_pls[8] long seg_pls_st[2]	Pulsos Pulsos
	Δposição (flag)	long seg_flag	1: ativo 0: inativo
	Comando de parada	ushort stop_req[2]	- (1: stop)
	Alarmes	ushort alarm[2]	-

C.4.2 Robot Raconteur

O RR¹ é uma arquitetura e biblioteca de comunicação para aplicações de robótica e automação. Cada programa é um nó em uma rede formada em um único computador ou distribuída em diferentes dispositivos. Cada nó pode possuir diferentes objetos, propriedades e métodos que podem ser acessados e invocados pelos demais nós. O RR suporta diferentes linguagens, entre elas Matlab, C# e C++.

Para se conectar a um nó é necessário conhecer o endereço (IP e porta) do mesmo. Durante a conexão o nó cliente recebe uma descrição das características do servidor através de uma mensagem de texto. O Código C.1 apresenta a descrição do nó que disponibiliza a interface com o HSC.

Código C.1: Definição do serviço para interação com o Motoman DIA10.

```

service MotomanDIA10_service

object MotomanDIA10

property double[15] ActualRobotJointAngles
property double[15] DesiredRobotJointAngles

end object
    
```

O nó possui um objeto com duas propriedades. Uma para acesso as posições atuais das juntas (`ActualRobotJointAngles`) e outra para definir os ângulos desejados (`DesiredRobotJointAngles`). Esta referência é passada para uma malha de controle

¹Disponível em robotraconteur.com

de posição que é projetada em *Simulink*, compilada e executada pelo nó em uma frequência de 500 Hz . Um exemplo para esta malha é apresentado na Figura C.7.

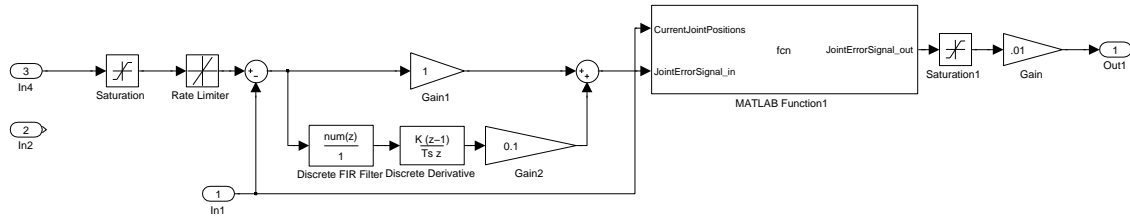


Figura C.7: Exemplo de malha de controle de juntas implementada em Simulink para o Motoman DIA10.

No exemplo tem-se uma malha proporcional-integral com a entrada 1 recebendo a posição atual e a 3 a referência. A entrada 2 não é utilizada, mas é um requisito do nó RR. A saída da malha é um a correção em radianos, que será transformada para pulsos e enviada fazendo uso da biblioteca de comunicação com o HSC.

C.5 Operação

A seguir serão apresentados diferentes procedimentos para a operação do robô Motoman DIA10. Os procedimentos estão divididos em duas etapas: inicialização e conexão com computador. Na primeira considera-se o caso normal e a correção de erros comuns. Para a segunda são apresentados os procedimento comuns e exemplos específicos para operação já desenvolvidas.

C.5.1 Inicialização

A inicialização do robô é simples, sendo realizada em apenas um etapa:

1. Ligar o NX100 colocando a chave do painel frontal na posição ON, como mostrado na Figura C.8(a).
 - (a) Esperar até que o NX100 esteja completamente inicializado. O *Teach Pendant* apresentará a tela inicial mostrada na Figura C.8(b).

Entretanto durante esta inicialização podem ocorrer falhas ocasionadas, principalmente, por falta de carga nas baterias do circuito de memória dos *encoders* absolutos. Esses erros e suas soluções são apresentados a seguir.

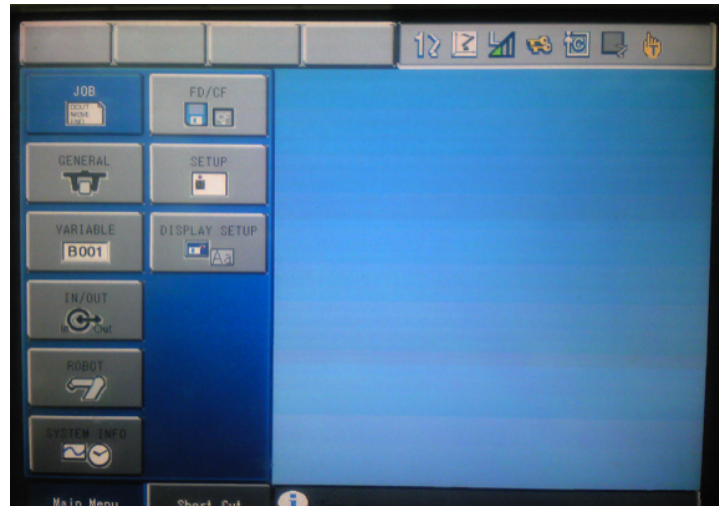
1. ENCODER BATTERY WEAK

Robot 2 [SLURBT7]

- É na realidade um aviso, não impede a operação do robô.



(a) Chave de energia



(b) Tela do *Teach Pendant* após inicialização

Figura C.8: Inicialização do NX100.

2. Alarm 4311 SV#2

ENCODER BACK-UP ERROR

Robot 2 [SLURBT7]

Alarm 4314 SV#2

SERIAL ENCODER BATTERY ERROR

Robot 2 [SLURBT7]

Alarm 4107

OUT OF RANGE (ABSO DATA)

Robot 2 [SLURBT7]

- Esse conjunto de erros permite operar o robô sem que seja necessária uma reinicialização do NX100. Entretanto é necessário realizar um procedimento para calibrar a referência para os *encoders* absolutos. Para isso, siga o procedimento abaixo.

- (a) Limpe os erros utilizando a opção *reset* na tela ou o botão *cancel*.
- (b) Entre no modo de administrador:
 - i. Utilize os menus da tela para navegar para: *System Info* → *Security*;
 - ii. Usando os botões direcionais selecione o modo *management*;
 - iii. Aperte *select* para alterar para este modo;
 - iv. Digite a senha: 99999999 (8 noves) e tecle *enter*.
- (c) Utilize os menus da tela para navegar para: *Robot* → *Second Home Position*.
- (d) Ligue os servo motores pressionando *Servo on Ready*.

- (e) Segurar o botão de segurança atrás do *Teach Pendant* na posição de trabalho (metade da excursão total).
 - O alarme soará novamente, deve-se pressionar o *Reset* na tela do *Teach Pendant*.
- (f) Com o botão de segurança na posição de trabalho desloque as juntas indicadas na mensagem de erro (esta está repetida na região inferior da tela). Para isso, utilize os botões azuis equivalentes a cada junta.
 - Para ajustar a junta 7, deve-se utilizar os botões de ajuste da junta S com o botão *Shift* pressionado.
 - Para uma calibração mais precisa da posição zero de cada junta deve-se utilizar a barra metálica de referência², alinhando as ranhuras referentes a junta (ver Figura C.9).
- (g) Após o ajuste de todas as juntas indicadas. Utilize os menus da tela para navegar para: *Robot* → *Home Position*.
 - i. Selecione cada junta que apresenta asteriscos no local do valor e pressione *Select*;
 - ii. Uma janela de pergunta pede para confirmar se deseja-se definir a posição atual como nova posição inicial, confirme.
- (h) Retorne ao modo de usuário comum:
 - i. Utilize os menus da tela para navegar para: *System Info* → *Security*;
 - ii. Usando os botões direcionais selecione o modo *editing*;
 - iii. Aperte *select* para alterar para este modo.

3. Alarm 1326 SV#2

DEFECTIVE ENCODER ABSOLUTE DATA

Robot 2 [SLURBT7]

- Este é um erro de alta prioridade que não ser apagado. Para operar o robô é necessário reiniciar o NX100, mas uma reinicialização direta tende a repetir o erro. A solução é movimentar as juntas que estão indicadas no erro através do procedimento de liberação dos freios descrito a seguir.
- Esta operação é simplificada se realizada por duas pessoas.

- (a) Utilize os menus da tela para navegar para: *Robot* → *Manual Break Release*.
 - Uma janela de pergunta pede para confirmar se deseja-se controlar manualmente os freios, confirme.

²a barra de referência está fixada na parte de trás do tronco do robô

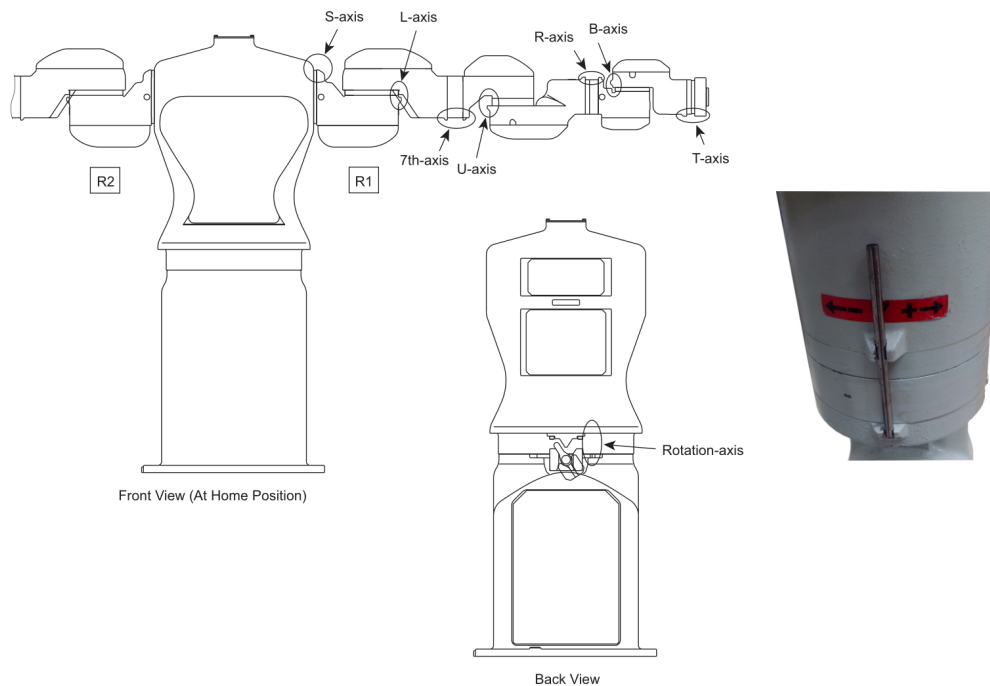


Figura C.9: Localização das ranhuras de referência e exemplo para o procedimento de calibração.

- (b) O desligamento do freio pode ser realizado individualmente para cada junta ou para cada braço como todo. Sugere-se utilizar o modo individual. Assim, para cada junta indicada no alarme:
 - i. Usando os botões direcionais selecione a junta desejada;
 - ii. Segure o botão de segurança na posição de trabalho + o botão *interlock* + o botão *select*. Um pequeno estalo indica que o feio está livre.
 - iii. Com os três botões pressionados deve-se deslocar a junta manualmente nos dois sentidos de movimento.
- (c) Desligue o NX100 e religue-o.
- (d) Caso o erro persista repita a mesma operação.

OBSERVAÇÃO - O erro mais comum ao ligar o NX100 é o 2 para as juntas B, T e 7. Recomenda-se que antes de desligar o NX100 desloque-se estas juntas para a posição zero. Desta forma, o passo 2f pode ser pulado.

C.5.2 Conexão com computador

Nesta seção é listado o procedimento para realizar a conexão entre o computador e o NX100 via HSC, como descrito na Seção C.4. Primeiramente apresenta-se a criação do serviço RR que disponibiliza o acesso as posições reais das juntas e permite designar a referência para a malha de controle de posição. A partir deste serviço

qualquer programa ou dispositivo que possa utiliza o RR (*e.g.* Matlab, Labview), com acesso a rede, pode comandar o robô.

ATENÇÃO - Ao movimentar o robô mantenha o *Teach Pendant* ao alcance para acionar o botão de parada de emergência caso necessário. Recomenda-se que o robô seja operado sempre em dupla, assim um está responsável pelo botão de emergência, enquanto o outro executa os programas necessários.

1. Mude o modo de trabalho do robô rotacionando a chave do *Teach Pendant* para a posição *Play*.
 - Um aviso indicará *Push play mode enable*.
2. Pressione o botão branco (*Play mode enable*) no NX100 (ver Fig.).
 - Este botão vai estar aceso devido ao passo anterior;
 - Após pressionar-lo o botão se apagará;
 - As funções do *Teach Pendant* permanecem bloqueadas até o acionamento deste botão. Retorne ao modo *Teach* se for necessário.
3. Ligue o HSC através do botão redondo do mesmo. Após esta ação o botão se torna verde.
4. Espere até que o HSC carregue completamente.
5. No *Teach Pendant*: selecione, através dos menus da tela, *Job* → *Select Job*.
 - (a) Com os botões direcionais selecione a tarefa *Computer*;
 - (b) Utilize a tecla *select* para confirmar.
6. Ligue os servo motores pressionando *Servo On Ready*.
7. No computador, execute o programa *DIA10SimulinkHost*.
8. Carregue a biblioteca contendo a malha de controle gerada no *Simulink*:
 - (a) Na janela do programa clique em *Load*;
 - (b) Busque pelo local do arquivo na janela que se abrirá. Esta abre no local utilizado na última execução;
 - (c) Selecione a biblioteca desejada e clique em *Abrir*.
9. Pressione *Start* e espere a conexão com o HSC se completar, ao lado dos botões irão aparecer as posições atuais das juntas.
10. Pressione *Start* no *Teach Pendant* (as juntas do Motoman devem estar próximas da posição zero).

OBSERVAÇÃO - Após pressionar *Start* no programa *DIA10SimulinkHost* para se conectar ao HSC, o tempo decorrido para realizar esta conexão não deve ser maior que *2 min*. Se este limite for excedido pode ter ocorrido uma falha. Feche o *DIA10SimulinkHost* e abra o programa *HSCTestMonitor*. Espere até que apareça a frase "*Finish Enter Exit*", aperte *Enter* e então repita o procedimento anterior a partir do passo 7.

A baixo é descrito o procedimento para acessar no Matlab o serviço RR de controle do Motoman DIA10 criado com o procedimento acima.

1. Abra o Matlab (versão 2011 ou superior) em que o RR está instalado.
2. O conexão com o serviço RR é realizada através do comando

```
motoman = RobotRaconteur.Connect('tcp://ip:5203/MotomanSimulinkHost/MotomanDIA10');
```

- `ip` deve ser substituído pelo endereço IP da máquina em que o serviço está rodando;
 - Dependendo da versão de RR utilizada a primeira tentativa de conexão pode resultar em erro, pois tenta-se utilizar IPv6. Tente novamente;
 - Se ocorrer um erro durante a conexão verifique o endereço de IP utilizado.
3. Após a conexão, a variável `motoman` é a representação do robô. Para acessá-lo utiliza-se duas propriedades:
 - `ActualRobotJointAngles` - fornece um vetor coluna com a posição das quinze juntas do robô;
 - `DesiredRobotJointAngles` - recebe um vetor coluna com as posições desejadas para as quinze juntas do robô.
 - A ordem das juntas neste vetor é apresentada na Tabela C.4.
 4. Sendo necessário encerrar a conexão com o serviço sem fechar o mesmo, utilize o seguinte comando:

```
RobotRaconteur.Disconnect(motoman);
```

Tabela C.4: Ordem de apresentação das posições das juntas no serviço RR.

	Tronco	R2							R1						
índice	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
junta	S1	S	L	7	U	R	B	T	S	L	7	U	R	B	T

O código abaixo apresenta um exemplo simples demonstrando o procedimento listado a cima. Após realizar a conexão com o robô, supondo o serviço no endereço local, a posição inicial é adquirida e realiza-se uma trajetória a partir desta. A trajetória consiste de pequenos movimentos nas junta R, B e T dos dois braços, retornando a posição inicial.


```

motoman=RobotRaconteur.Connect('tcp://localhost:5203/MotomanSimulinkHost/
    MotomanDIA10');

initial=motoman.ActualRobotJointAngles;
for i=0:15
    s= sin(i*pi/30);
    motoman.DesiredRobotJointAngles=initial+[zeros(5,1);s*pi/2;-s*pi/4;s*pi/2;...
        zeros(4,1);s*pi/2;-s*pi/4;s*pi/2];

    fprintf('%0.3f\n',motoman.ActualRobotJointAngles)
    fprintf('\n')
    pause(1)
end
motoman.DesiredRobotJointAngles=initial;

RobotRaconteur.Disconnect(motoman);

```