TATIANA DE FIGUEIREDO PEREIRA ALVES TAVEIRA PAZELLI

***Montagem e Controle $\mathcal{H}_\infty$ Não Linear de
Manipuladores Espaciais com Base Flutuante***

São Carlos

2011

TATIANA DE FIGUEIREDO PEREIRA ALVES TAVEIRA PAZELLI

***Montagem e Controle $\mathcal{H}_\infty$ Não Linear de
Manipuladores Espaciais com Base Flutuante***

Tese apresentada à Escola de Engenharia de
São Carlos da Universidade de São Paulo,
como parte dos requisitos para obtenção do
título de Doutora em Ciências pelo Programa
de Engenharia Elétrica.

Área de Concentração: Sistemas Dinâmicos
Orientador: Prof. Dr. Marco Henrique Terra

São Carlos

2011

**Trata-se da versão corrigida da tese. A versão original se encontra disponível na
EESC/USP que aloja o Programa de Pós-Graduação de Engenharia Elétrica.**

AUTORIZO A REPRODUÇÃO E DIVULGAÇÃO TOTAL OU PARCIAL DESTE TRABALHO, POR QUALQUER MEIO CONVENCIONAL OU ELETRÔNICO, PARA FINS DE ESTUDO E PESQUISA, DESDE QUE CITADA A FONTE.

Ficha catalográfica preparada pela Seção de Tratamento
da Informação do Serviço de Biblioteca – EESC/USP

P348m

Pazelli, Tatiana de Figueiredo Pereira Alves Taveira.
Montagem e controle H^∞ não-linear de manipuladores
espaciais com base flutuante. / Tatiana de Figueiredo
Pereira Alves Taveira Pazelli ; orientador Marco Henrique
Terra. São Carlos, 2011.

Tese (Doutorado - Programa de Pós-Graduação em
Engenharia Elétrica e Área de Concentração em Sistemas
Dinâmicos)-- Escola de Engenharia de São Carlos da
Universidade de São Paulo, 2011.

1. Robótica. 2. Manipuladores espaciais. 3. Controle
 H^∞ não-linear. 4. Redes neurais. I. Título.

FOLHA DE JULGAMENTO

Candidata:

Engenheira **TATIANA DE FIGUEIREDO PEREIRA ALVES TAVEIRA PAZELLI.**

Título da tese: "Montagem e controle H_∞ não-linear de manipuladores espaciais com base flutuante."

Data da defesa: 13/01/2012

Comissão Julgadora:

Prof. Associado **Marco Henrique Terra (Orientador)**
(Escola de Engenharia de São Carlos/EESC)

Prof. Dr. **Adriano Almeida Gonçalves Siqueira**
(Escola de Engenharia de São Carlos/EESC)

Prof. Titular **José Jaime da Cruz**
(Escola Politécnica/USP)

Prof. Associado **Eduardo Aoun Tannuri**
(Escola Politécnica/USP)

Dr. **Luiz Carlos Gadelha de Souza**
(Instituto Nacional de Pesquisas Espaciais/INPE)

Resultado:

Dpro vado

APROVADO

APROVADA

APROVADO

Aprovado

Coordenador do Programa de Pós-Graduação em Engenharia Elétrica:
Prof. Titular **Denis Vinicius Coury**

Presidente da Comissão de Pós-Graduação:
Prof. Associado **Paulo Cesar Lima Segantine**

ET 000000
Serviço de Pós-Graduação
Protocolado em 13/01/2012
13/01/2012

Aos meus avós,
pela inspiração na busca pelo conhecimento.

Aos meus filhos,
por me indicarem o caminho.

Agradecimentos

A Deus, pela constante iluminação.

Aos meus filhos Marcelo e Theo, por darem significado à minha vida e suportarem com resignação a ausência da mamãe para que este trabalho fosse concluído.

Ao meu marido Henrique, pelo amor incondicional, por acreditar na minha capacidade e alimentar minha confiança todos os dias.

Aos meus pais Luís Carlos e Maria Beatriz, pelo empenho na minha educação, por todo amor, compreensão e incentivo que sempre guiaram os meus passos e me motivaram a prosseguir, mesmo nos momentos mais difíceis.

Aos meus irmãos Rafael e Cairo, pela amizade, pelo suporte e pelo carinho desde sempre.

Aos meus sogros Altino e Haida, pela disponibilidade da orientação técnica e pela preocupação constante com o êxito deste trabalho.

A todos os familiares e amigos, que se interessaram, se disponibilizaram e colaboraram de muitas formas para que as dificuldades fossem enfrentadas com mais leveza e a conclusão deste trabalho pudesse ser comemorada.

Aos meus colegas do LASI, que pela alegria da convivência em todos esses anos de trabalho acabaram por se tornar grandes amigos. Em especial, aos engenheiros Rafael e Samuel que atuaram no desenho mecânico do robô, e ao engenheiro José Miguel que implementou o software do sistema embarcado.

Ao Professor Marco Henrique Terra, pela orientação, dedicação e paciência em todas as fases deste trabalho.

Ao Professor Adriano Almeida Gonçalves Siqueira, pelo apoio na realização deste trabalho em todos os momentos de necessidade.

Aos professores e funcionários do Departamento de Engenharia Elétrica da Escola de Engenharia de São Carlos, pela contribuição efetiva na minha formação como engenheira, pesquisadora e como pessoa em todos estes anos.

À FAPESP, pelo suporte financeiro que possibilitou o desenvolvimento desta pesquisa.

“Energy and persistence conquer all things.”

Benjamin Franklin

Sumário

Resumo

Abstract

Publicações

Lista de Figuras

Lista de Tabelas

Lista de Siglas

Lista de Símbolos

Introdução p. 1

I Plataforma Experimental 13

1 Esquema Geral p. 17

1.1 Especificação p. 18

1.2 Concepção p. 19

2 Sistema Mecânico p. 27

2.1 Projeto p. 27

2.2 Implementação p. 34

2.3 Modelagem p. 39

3	Sistema Eletrônico	p. 43
3.1	Projeto	p. 43
3.2	Implementação	p. 53
4	Sistema Computacional	p. 59
4.1	Ambiente de Simulação e Controle de Manipuladores Espaciais	p. 59
II	Controle $\mathcal{H}_\infty$ Não Linear	75
5	Manipulador Dinamicamente Equivalente	p. 79
5.1	UARM	p. 81
5.2	UARM-E	p. 81
6	Controle $\mathcal{H}_\infty$ Não Linear no Espaço da Tarefa	p. 83
6.1	Formulação do Problema	p. 83
6.2	Controlador Adaptativo $\mathcal{H}_\infty$ Não Linear baseado em Redes Neurais	p. 87
7	Resultados	p. 95
7.1	Simulação	p. 97
7.2	Implementação	p. 104
7.3	Análise dos Resultados	p. 108
	Conclusão	p. 113
	Referências Bibliográficas	p. 115
	Apêndice A – UnderActuated Robot Manipulator	p. 121
	Apêndice B – Descrição Técnica dos Equipamentos	p. 123
	Apêndice C – Dimensionamento das Peças Mecânicas	p. 127

Apêndice D – Conexões Elétricas	p. 143
Apêndice E – Esquemáticos do Sistema Eletrônico	p. 147
Apêndice F – Matrizes do Modelo Dinâmico	p. 155

Resumo

PAZELLI, T. F. P. A. T. (2011). *Montagem e Controle $\mathcal{H}_\infty$ Não Linear de Manipuladores Espaciais com Base Flutuante*. Tese (Doutorado), Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos.

Robôs manipuladores espaciais serão aplicados, em um futuro próximo, em serviços de resgate e manutenção de naves e satélites em órbita. O estudo e o desenvolvimento de controladores para esse tipo de sistema é fundamental para que essas aplicações se tornem realidade. Nesta tese, uma plataforma experimental é construída para possibilitar a avaliação comportamental desse tipo de sistema. Baseada em um módulo de flutuação por colchões de ar, é composta por uma base livre, elos conectados por juntas e efetadores. Duas possibilidades de flutuação foram definidas para tornar a estrutura mais versátil, a primeira utiliza uma câmara de ar na mesa de apoio e a segunda utiliza câmaras de ar na base e em cada junta do robô. Sua estrutura mecânica modular permite diversas configurações, com um ou dois braços compostos por elos rígidos ou flexíveis. Toda a eletrônica de comando e a alimentação dos componentes do robô são alocadas em sua base flutuante, baseando a comunicação do sistema com o computador remoto em um padrão de comunicação sem fio. O software de controle, desenvolvido em Matlab e residente no computador remoto, apresenta uma interface amigável e intuitiva, possibilitando a utilização tanto do UARM como do robô de base livre flutuante para testes simulados e experimentais de sistemas de controle. A principal característica dos manipuladores espaciais é o acoplamento dinâmico entre a base e o braço robótico. A fim de evitar as complicações envolvidas no mapeamento cinemático desses sistemas, o problema de acompanhamento de trajetória é formulado diretamente no espaço da tarefa. Assim as posições do efetador do manipulador são diretamente controladas. O equacionamento dinâmico do manipulador de base livre flutuante é descrito a partir do conceito do Manipulador Dinamicamente Equivalente. Propõe-se uma solução de controle adaptativo robusto baseado no critério $\mathcal{H}_\infty$ para lidar com o problema de acompanhamento de trajetória sujeito a incertezas no modelo e distúrbios externos. A adaptabilidade das redes neurais é aliada à robustez definida por um controlador $\mathcal{H}_\infty$ não linear, compondo diferentes técnicas desenvolvidas de acordo com o conhecimento e a disponibilidade do modelo do robô para o controlador. A análise de resultados de simulação e de experimentos realizados no UARM mostraram a aplicabilidade dos métodos, assim como sua capacidade de robustez. Gráficos ilustraram o procedimento do acompanhamento de trajetória realizado pelo efetador do manipulador espacial identificando a ação das leis de controle propostas. Uma comparação numérica entre as estratégias foi estabelecida por índices de desempenho relacionados ao consumo de energia e ao erro de acompanhamento.

Palavras-chave: Robótica, manipuladores espaciais, controle $\mathcal{H}_\infty$ não linear, redes neurais.

Abstract

PAZELLI, T. F. P. A. T. (2011). *Assembly and Nonlinear $\mathcal{H}_\infty$ Control of Free-Floating Base Space Manipulators*. Thesis (Doctoral), Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos.

Space manipulators robots will be applied, in a near future, in rescue services and maintenance of spacecraft and satellites in orbit. The study and development of controllers for this type of system is crucial to ensure that those applications become reality. At this thesis, an experimental platform is built to enable behavioral assessment of this type of system. Based on a floating module by air bearings, it is composed by a free base, links connected by joints and end-effectors. Two possibilities of fluctuation were set to make the structure more versatile. The first uses an air chamber in the support desk and the second uses air chambers at the base and in each joint of the robot. Its modular mechanical structure allows a variety of configurations, with one or two arms which may be composed of flexible or rigid links. The entire command electronics and the power of the robot's components are allocated in its floating base, basing the system communication with the remote computer in a wireless communication standard. The control software, developed in Matlab and residing on the remote computer, presents a friendly and intuitive interface, enabling the use of both the UARM and the free-floating base robot for simulated and experimental testing of control systems. The main characteristic of space manipulators is the dynamic coupling between the base and the robotic arm. In order to avoid the complications involved in kinematic mapping of these systems, the problem of trajectory tracking is formulated directly in task space. So the positions of the manipulator end-effector are directly controlled. The dynamic equation of the free-floating manipulator is described from the concept of Dynamically Equivalent Manipulator. A solution of adaptive robust control is proposed, based on $\mathcal{H}_\infty$ criterion to deal with the problem of trajectory tracking subject to uncertainties in the model and external disturbances. The adaptability of neural networks is combined with robustness defined by a nonlinear $\mathcal{H}_\infty$ controller composing different techniques developed in accordance with the knowledge and the availability of the robot's model to the controller. The analysis of results of simulation and experiments performed in UARM showed the applicability of the methods, as well as its capacity for robustness. Graphs have illustrated the trajectory tracking procedure conducted by the end-effector of the space manipulator identifying the action of control laws proposed. A numerical comparison between the strategies was provided by performance indices related to energy consumption and the tracking error.

Keywords: Robotics, space manipulators, nonlinear $\mathcal{H}_\infty$ control, neural networks.

Publicações

PAZELLI, T. F. P. A. T.; TERRA, M. H.; SIQUEIRA, A. A. G. . Adaptive Robust Controller for a Constrained Free-Floating Space Manipulator. In: *Anais do X Congresso Brasileiro de Inteligência Computacional (CBIC)*, Fortaleza - CE, 2011.

NOGUEIRA, S. L.; SIQUEIRA, A. A. G.; PAZELLI, T. F. P. A. T.; TERRA, M. H. . Controladores Fuzzy Adaptativos $\mathcal{H}_\infty$ Não Lineares Aplicados a Manipuladores com Restrições de Posição e Força. In: *Anais do X Congresso Brasileiro de Inteligência Computacional (CBIC)*, Fortaleza - CE, 2011.

PAZELLI, T. F. P. A. T.; TERRA, M. H.; SIQUEIRA, A. A. G. . Experimental Investigation on Adaptive Robust Controller Designs Applied to a Free-Floating Space Manipulator. *Control Engineering Practice*, v. 9, n. 4, p. 395-408, 2011.

PAZELLI, T. F. P. A. T.; TERRA, M. H.; SIQUEIRA, A. A. G. . Adaptive Robust Controller Designs Applied to Free-Floating Space Manipulators in Task Space, *Robot Manipulators, Trends and Development*, Agustin Jimenez and Basil M Al Hadithi (Ed.), ISBN: 978-953-307-073-5, Vienna: In-Tech, p. 231-248, 2010.

PAZELLI, T. F. P. A. T.; TERRA, M. H.; SIQUEIRA, A. A. G. . Framework for the Development of Control Systems for Free-Floating Space Manipulators. In: *Anais do XVIII Congresso Brasileiro de Automática (CBA)*, Bonito - MS, 2010, p. 696-703.

PAZELLI, T. F. P. A. T.; TERRA, M. H.; SIQUEIRA, A. A. G. . Neural Network Robust Controllers Applied to Free-Floating Space Manipulators in Task Space. In: *Proceedings of 2010 American Control Conference (ACC)*, Baltimore - MD - USA, 2010, p. 3648-3653.

PAZELLI, T. F. P. A. T.; TERRA, M. H.; SIQUEIRA, A. A. G. . Plataforma de Pesquisa Experimental para Manipuladores de Base Livre Flutuante. In: *Proceedings of 4th Workshop in Applied Robotics and Automation (ROBOCONTROL)*, Bauru - SP, 2010.

CERRI, J. P.; TERRA, M. H.; SIQUEIRA, A. A. G.; PAZELLI, T. F. P. A. T. . Measurement-Feedback Control for Discrete-Time State-Space Systems. In: *Anais do IX Simpósio Brasileiro de Automação Inteligente (SBAI)*, Brasília - DF, 2009, p. 1-6.

PAZELLI, T. F. P. A. T.; SIQUEIRA, A. A. G.; TERRA, M. H. . Mixed Model Based/ANFIS Adaptive Robust Controller with H-infinity Criterion Applied to Free-Floating Space Manipulators. In: *Proceedings of IX International Symposium on Artificial Intelligence, Robotics and Automation in Space (iSAIRAS)*, Los Angeles - CA - USA, 2008.

NOGUEIRA, S. L.; PAZELLI, T. F. P. A. T.; SIQUEIRA, A. A. G.; TERRA, M. H. . Adaptive Fuzzy Nonlinear H-infinity Tracking Control Design of a Constrained Robot System. In: *Proceedings of 16th Mediterranean Conference on Control and Automation (MED08)*, Ajaccio - France, 2008. p. 362-367.

INOUE, R. S.; PAZELLI, T. F. P. A. T.; SIQUEIRA, A. A. G.; TERRA, M. H. . Mixed Model Based/Fuzzy Adaptive Robust Controller with H-infinity Criterion Applied to Wheeled Mobile Robots. In: *Proceedings of 3rd IFAC Symposium on System, Structure and Control (SSSC'07)*, Foz do Iguaçu - PR - Brasil, 2007.

PAZELLI, T. F. P. A. T.; INOUE, R. S.; SIQUEIRA, A. A. G.; TERRA, M. H. . Mixed Model Based/Fuzzy Adaptive Robust Controller with H-infinity Criterion Applied to Free-Floating Space Manipulators. In: *Anais do VIII Simpósio Brasileiro de Automação Inteligente (SBAI)*, Florianópolis - SC, 2007.

INOUE, R. S.; PAZELLI, T. F. P. A. T.; SIQUEIRA, A. A. G.; TERRA, M. H. . Controlador Robusto H-infinito Baseado em Redes Neurais Aplicado em Robôs Móveis com Rodas. In: *Anais do VIII Simpósio Brasileiro de Automação Inteligente (SBAI)*, Florianópolis - SC, 2007.

ZAMAMI, A. C.; PAZELLI, T. F. P. A. T.; TERRA, M. H. . Especificação da Metodologia de Comunicação no Controle de um Manipulador de Base Livre Flutuante. In: *Anais do 15º Simpósio Internacional de Iniciação Científica da Universidade de São Paulo (SIICUSP)*, São Carlos - SP - Brasil, 2007.

Lista de Figuras

1	Primeiro voo do SRMS.	p. 1
2	Construção da ISS.	p. 2
3	Manutenção do telescópio espacial Hubble.	p. 2
4	Estrutura do SSRMS.	p. 2
5	JEMRMS.	p. 3
6	Concepção artística do ETS-VII.	p. 4
7	<i>Orbital Express</i>	p. 5
8	Concepção artística da missão DEOS.	p. 5
9	Plataformas experimentais com sistema de sustentação por colchões de ar. . .	p. 11
10	Plataforma experimental com sistema de suspensão por cabos.	p. 11
11	Esquema Geral.	p. 17
12	Módulo de Flutuação - Mesa Ealing.	p. 19
13	Módulo de Atuação - Motor Maxon EC 90 flat.	p. 20
14	Módulo de Sensoriamento - (a) Esquema de funcionamento do encoder; (b) Encoder E6S: encapsulamento e disco.	p. 21
15	Módulo de Sensoriamento - Câmera SMX-160C.	p. 22
16	Módulo de Sensoriamento - Sensor de força: vista interna e aplicação. . . .	p. 22
17	Módulo de Comunicação - RCM4510W: Módulo RabbitCore com Zigbee. . .	p. 24
18	Módulo de Energia - Bateria de polímero de lítio (LiPo) e carregador. . . .	p. 25
19	Módulo de Controle - ASCME.	p. 26
20	Projeto Mecânico - Base.	p. 28
21	Projeto Mecânico - Câmara de ar.	p. 28

22	Projeto Mecânico - Detalhes de projeto da câmara de ar.	p. 28
23	Projeto Mecânico - Manipulador.	p. 29
24	Projeto Mecânico - Junta + Elo.	p. 29
25	Projeto Mecânico - Junta.	p. 30
26	Projeto Mecânico - Junta: Corte longitudinal e vista explodida.	p. 30
27	Projeto Mecânico - Copo: Vista e corte longitudinal.	p. 30
28	Projeto Mecânico - Base da junta.	p. 31
29	Projeto Mecânico - Tampa da junta.	p. 31
30	Projeto Mecânico - Fim de curso.	p. 31
31	Projeto Mecânico - Elo.	p. 32
32	Projeto Mecânico - Elo: Vista explodida.	p. 32
33	Projeto Mecânico - Conexão Junta-Elo.	p. 32
34	Projeto Mecânico - Elo.	p. 33
35	Projeto Mecânico - Conexão Elo-Junta.	p. 33
36	Projeto Mecânico - Conexão em L.	p. 33
37	Projeto Mecânico - Robô espacial.	p. 34
38	Sistema Mecânico - Base (foto).	p. 34
39	Sistema Mecânico - Câmara de ar: Vistas superior e inferior (foto).	p. 35
40	Sistema Mecânico - Manipulador (foto).	p. 35
41	Sistema Mecânico - Junta: Montagem e vista explodida (foto).	p. 35
42	Sistema Mecânico - Copo: Vistas superior e inferior (foto).	p. 36
43	Sistema Mecânico - Tampa da junta (foto).	p. 36
44	Sistema Mecânico - Fim de curso (foto).	p. 36
45	Sistema Mecânico - Elo: Montagem e vista explodida (foto).	p. 36
46	Sistema Mecânico - Conexão Junta-Elo (foto).	p. 37
47	Sistema Mecânico - Elo (foto).	p. 37

48	Sistema Mecânico - Conexão Elo-Junta (foto).	p. 37
49	Sistema Mecânico - Conexão em L (foto).	p. 37
50	Montagem - Robô espacial: (a) 1 braço, 1 junta; (b) 1 braço, 2 juntas.	p. 38
51	Montagem - Robô espacial: 1 braço, 3 juntas.	p. 38
52	Montagem - Robô espacial: 1 braço, 4 juntas.	p. 38
53	Montagem - Robô espacial: (a) 1 braço, 5 juntas; (b) 1 braço, 6 juntas.	p. 39
54	Montagem - Robô espacial: (a) 2 braços, 1 junta; (b) 2 braços, 2 juntas.	p. 39
55	Montagem - Robô espacial: (a) 2 braços, 3 juntas; (b) 2 braços, 1/5 juntas.	p. 39
56	Modelagem - Corpo livre 1: Base espacial	p. 41
57	Modelagem - Corpo livre 2: Junta	p. 41
58	Modelagem - Corpo livre 3: Efetuador	p. 41
59	Projeto Eletrônico - Módulo de Atuação.	p. 43
60	Projeto Eletrônico - (a) DEC 50/5 e (b) Configuração para controle de velocidade em malha aberta.	p. 44
61	Projeto Eletrônico - Módulo de Sensoriamento.	p. 47
62	Projeto Eletrônico - Operação do decodificador de quadratura.	p. 47
63	Projeto Eletrônico - Módulo de Energia.	p. 51
64	Projeto Eletrônico - Configuração do conversor CC-CC.	p. 51
65	Projeto Eletrônico - Supervisor de tensão.	p. 52
66	Sistema Eletrônico - Diagrama Geral.	p. 54
67	Sistema Eletrônico - Layout: Alocação de componentes e roteamento - Camada superior.	p. 55
68	Sistema Eletrônico - Layout: Alocação de componentes e roteamento - Camada inferior.	p. 55
69	Sistema Eletrônico - Placa eletrônica: Vistas superior e inferior.	p. 56
70	Sistema Eletrônico - Cabeamento: (a) Base; (b) Junta.	p. 57

71	Sistema Eletrônico - Cabeamento: Placa eletrônica, baterias, drivers, encoders e motores.	p. 57
72	ASCME - Ambiente de Simulação e Controle de Manipuladores Espaciais. .	p. 61
73	ASCME - Seleção do robô: UARM ou UARM-E.	p. 61
74	ASCME - Seleção do procedimento: Simulação (<i>Simulation</i>) ou Experimento (<i>Experiment</i>).	p. 62
75	ASCME - Comandos do robô.	p. 62
76	ASCME - Posição [0 0 ... 0].	p. 63
77	ASCME - Comandos do usuário: (a) Simulação, (b) Experimento.	p. 63
78	ASCME - Parâmetros do procedimento: (a) Robô, (b) Tarefa.	p. 64
79	ASCME - Espaço de trabalho.	p. 64
80	ASCME - Resultados numéricos.	p. 65
81	ASCME - Resultados gráficos.	p. 65
82	ASCME - Configuração do UARM.	p. 66
83	ASCME - Configuração do UARM-E: (a) 2 juntas, (b) 5 juntas.	p. 67
84	ASCME - Configuração de hardware do UARM-E: (a) 2 juntas, (b) 5 juntas. .	p. 67
85	ASCME - Configuração do UARM-E: Número de juntas.	p. 68
86	ASCME - Configuração da tarefa: (a) 2 juntas, (b) 5 juntas.	p. 69
87	ASCME - Configuração da tarefa: Seleção do espaço de controle.	p. 69
88	ASCME - Configuração da tarefa: Seleção da trajetória para controle no espaço da tarefa.	p. 69
89	ASCME - Configuração da tarefa: Seleção do controlador.	p. 70
90	ASCME - Laço de controle: Experimento no UARM.	p. 71
91	ASCME - Laço de controle: Simulação de controle do UARM no espaço das juntas.	p. 71
92	ASCME - Laço de controle: Experimento no UARM-E.	p. 72
93	ASCME - Laço de controle: Simulação de controle do UARM-E no espaço da tarefa.	p. 72

94	Modelagem - ME e seu MDE correspondente.	p. 79
95	Controle - Estrutura de uma rede neural de saída única.	p. 87
96	Resultados - Simulação: Distúrbio de torque.	p. 97
97	Resultados - UARM - Simulação: Caso nominal: Trajetória, erros de posição e velocidades do efetuador - Torque calculado.	p. 100
98	Resultados - UARM - Simulação: Caso nominal: Trajetória, erros de posição e velocidades do efetuador - $\mathcal{H}_\infty$ não linear.	p. 100
99	Resultados - UARM - Simulação: Caso nominal: Trajetória, erros de posição e velocidades do efetuador - $\mathcal{H}_\infty$ não linear baseado em redes neurais (1). . .	p. 100
100	Resultados - UARM - Simulação: Caso nominal: Trajetória, erros de posição e velocidades do efetuador - $\mathcal{H}_\infty$ não linear baseado em redes neurais (2). . .	p. 100
101	Resultados - UARM - Simulação: Caso com distúrbios: Trajetória, erros de posição e velocidades do efetuador - Torque calculado.	p. 101
102	Resultados - UARM - Simulação: Caso com distúrbios: Trajetória, erros de posição e velocidades do efetuador - $\mathcal{H}_\infty$ não linear.	p. 101
103	Resultados - UARM - Simulação: Caso com distúrbios: Trajetória, erros de posição e velocidades do efetuador - $\mathcal{H}_\infty$ não linear baseado em redes neurais (1).	p. 101
104	Resultados - UARM - Simulação: Caso com distúrbios: Trajetória, erros de posição e velocidades do efetuador - $\mathcal{H}_\infty$ não linear baseado em redes neurais (2).	p. 101
105	Resultados - UARM-E - Simulação: Caso nominal: Trajetória, erros de posição e velocidades do efetuador - Torque calculado.	p. 102
106	Resultados - UARM-E - Simulação: Caso nominal: Trajetória, erros de posição e velocidades do efetuador - $\mathcal{H}_\infty$ não linear.	p. 102
107	Resultados - UARM-E - Simulação: Caso nominal: Trajetória, erros de posição e velocidades do efetuador - $\mathcal{H}_\infty$ não linear baseado em redes neurais (1).	p. 102

108	Resultados - UARM-E - Simulação: Caso nominal: Trajetória, erros de posição e velocidades do efetuador - $\mathcal{H}_\infty$ não linear baseado em redes neurais (2).	p. 102
109	Resultados - UARM-E - Simulação: Caso com distúrbios: Trajetória, erros de posição e velocidades do efetuador - Torque calculado.	p. 103
110	Resultados - UARM-E - Simulação: Caso com distúrbios: Trajetória, erros de posição e velocidades do efetuador - $\mathcal{H}_\infty$ não linear.	p. 103
111	Resultados - UARM-E - Simulação: Caso com distúrbios: Trajetória, erros de posição e velocidades do efetuador - $\mathcal{H}_\infty$ não linear baseado em redes neurais (1).	p. 103
112	Resultados - UARM-E - Simulação: Caso com distúrbios: Trajetória, erros de posição e velocidades do efetuador - $\mathcal{H}_\infty$ não linear baseado em redes neurais (2).	p. 103
113	Resultados - UARM - Implementação: Distúrbio de torque.	p. 104
114	Resultados - UARM - Implementação: Caso nominal: Trajetória, erros de posição e velocidades do efetuador - Torque calculado.	p. 106
115	Resultados - UARM - Implementação: Caso nominal: Trajetória, erros de posição e velocidades do efetuador - $\mathcal{H}_\infty$ não linear.	p. 106
116	Resultados - UARM - Implementação: Caso nominal: Trajetória, erros de posição e velocidades do efetuador - $\mathcal{H}_\infty$ não linear baseado em redes neurais (1).	p. 106
117	Resultados - UARM - Implementação: Caso nominal: Trajetória, erros de posição e velocidades do efetuador - $\mathcal{H}_\infty$ não linear baseado em redes neurais (2).	p. 106
118	Resultados - UARM - Implementação: Caso com distúrbios: Trajetória, erros de posição e velocidades do efetuador - Torque calculado.	p. 107
119	Resultados - UARM - Implementação: Caso com distúrbios: Trajetória, erros de posição e velocidades do efetuador - $\mathcal{H}_\infty$ não linear.	p. 107
120	Resultados - UARM - Implementação: Caso com distúrbios: Trajetória, erros de posição e velocidades do efetuador - $\mathcal{H}_\infty$ não linear baseado em redes neurais (1).	p. 107

121	Resultados - UARM - Implementação: Caso com distúrbios: Trajetória, erros de posição e velocidades do efetuador - $\mathcal{H}_\infty$ não linear baseado em redes neurais (2).	p. 107
122	Resultados - UARM-E - Simulação: Caso com distúrbios: Torques nas juntas - (a) $\mathcal{H}_\infty$ não linear, (b) $\mathcal{H}_\infty$ não linear baseado em redes neurais (1), (c) $\mathcal{H}_\infty$ não linear baseado em redes neurais (2).	p. 108
123	Resultados - UARM-E - Simulação: Caso com distúrbios: Posição das juntas - $\mathcal{H}_\infty$ não linear baseado em redes neurais (1).	p. 109
124	Resultados - UARM-E - Simulação: Caso com incertezas: Adaptação dos parâmetros e atuação das redes neurais - (a) $\mathcal{H}_\infty$ não linear baseado em redes neurais (1), (b) $\mathcal{H}_\infty$ não linear baseado em redes neurais (2).	p. 110
125	Resultados - UARM - Implementação: Caso com distúrbios: Torques nas juntas - (a) Torque calculado, (b) $\mathcal{H}_\infty$ não linear, (c) $\mathcal{H}_\infty$ não linear baseado em redes neurais (1), (d) $\mathcal{H}_\infty$ não linear baseado em redes neurais (2).	p. 111
126	Apêndice A - UARM.	p. 122
127	Apêndice C - Base(2).	p. 128
128	Apêndice C - Prateleira.	p. 129
129	Apêndice C - Acessórios.	p. 130
130	Apêndice C - Câmara de ar.	p. 131
131	Apêndice C - Conexão Junta-Elo.	p. 132
132	Apêndice C - Copo.	p. 133
133	Apêndice C - Tampa da junta.	p. 134
134	Apêndice C - Base da junta.	p. 135
135	Apêndice C - Base da câmara de ar.	p. 136
136	Apêndice C - Fim de curso.	p. 137
137	Apêndice C - Elo da junta.	p. 138
138	Apêndice C - Elo do efetuador.	p. 139
139	Apêndice C - Conexão Elo-Junta.	p. 140
140	Apêndice C - Conexão em L.	p. 141

141	Apêndice E - Esquemático geral.	p. 148
142	Apêndice E - Esquemático do módulo de energia.	p. 149
143	Apêndice E - Esquemático do módulo de atuação.	p. 150
144	Apêndice E - Esquemático do módulo de comando.	p. 151
145	Apêndice E - Esquemático do módulo de comunicação.	p. 152
146	Apêndice E - Esquemático do módulo de sensoriamento.	p. 153

Lista de Tabelas

1	Módulo de Comunicação - Comparação entre os principais padrões de comunicação sem fio.	p. 23
2	Módulo de Energia - Comparação entre as principais tecnologias de bateria. .	p. 24
3	Módulo de Energia - Pontos positivos e negativos de cada tecnologia de bateria.	p. 25
4	Modelagem - Massa	p. 40
5	Modelagem - Parâmetros do robô	p. 42
6	Projeto Eletrônico - Operação do decodificador de quadratura.	p. 47
7	Projeto Eletrônico - Processo de leitura do decodificador de quadratura. . . .	p. 49
8	Projeto Eletrônico - Operação do conversor LVTTTL.	p. 50
9	Projeto Eletrônico - Análise de demanda de potência.	p. 50
10	Modelagem - UARM: MDE.	p. 81
11	Modelagem - UARM: ME.	p. 81
12	Modelagem - UARM-E: ME.	p. 82
13	Modelagem - UARM-E: MDE.	p. 82
14	Resultados - Simulação: Matrizes de ponderação selecionadas.	p. 98
15	Resultados - Simulação: Índices de desempenho.	p. 99
16	Resultados - UARM - Implementação: Matrizes de ponderação selecionadas.	p. 105
17	Resultados - UARM - Implementação: Índices de desempenho.	p. 105
18	Apêndice A - Parâmetros do UARM.	p. 122
19	Apêndice B - Descrição técnica do atuador.	p. 123
20	Apêndice B - Descrição técnica do encoder.	p. 123
21	Apêndice B - Descrição técnica do sistema de visão.	p. 124

22	Apêndice B - Descrição técnica do dispositivo de comunicação.	p. 124
23	Apêndice B - Descrição técnica do dispositivo de comando.	p. 124
24	Apêndice B - Descrição técnica da bateria.	p. 125
25	Apêndice B - Descrição técnica do amplificador.	p. 125
26	Apêndice B - Descrição técnica do DAC.	p. 125
27	Apêndice B - Descrição técnica do demultiplexador.	p. 125
28	Apêndice B - Descrição técnica do CPLD.	p. 126
29	Apêndice B - Descrição técnica do decodificador de quadratura.	p. 126
30	Apêndice B - Descrição técnica do conversor LVTTL.	p. 126
31	Apêndice B - Descrição técnica do conversor CC-CC.	p. 126
32	Apêndice B - Descrição técnica do supervisor de tensão.	p. 126
33	Apêndice D - Interface Driver - Motor	p. 143
34	Apêndice D - Interface DAC - Driver.	p. 143
35	Apêndice D - Interface DEMUX - Driver.	p. 144
36	Apêndice D - Interface μ C - DEMUX.	p. 144
37	Apêndice D - Interface Encoder - DecodQuad.	p. 145
38	Apêndice D - Interface DecQ - CPLD - μ C.	p. 145
39	Apêndice D - Interface μ C - DEMUX.	p. 145
40	Apêndice D - Interface μ C - Conversor LVTTL.	p. 145
41	Apêndice D - Interface DEMUX - Conversor LVTTL.	p. 145
42	Apêndice D - Interface Conversor LVTTL - DecodQuad.	p. 146
43	Apêndice D - Interface μ C - Chaves.	p. 146

Lista de Siglas

ANFIS - Adaptive Network-based Fuzzy Inference System

ARAMIS - do inglês *Automation, Robotics And Machine Intelligence Systems*

ASCME - Ambiente de Simulação e Controle de Manipuladores Espaciais

ASTRO - do inglês *Autonomous Space Transport Robotic Operations*

CAD - do inglês *Computer-Aided Design*

CC - Corrente Contínua

CI - Circuito Integrado

CMOS - do inglês *Complementary Metal-Oxide Semiconductor*

CPLD - do inglês *Complex Programmable Logic Device*

CSA - do inglês *Canadian Space Agency*

DAC - do inglês *Digital-to-Analog Converter*

DARPA - do inglês *Defense Advanced Research Projects Agency*

DEOS - do alemão *DEutsche Orbitale Servicing mission*

DLR - do alemão *Deutsches zentrum für Luft-und Raumfahrt*

EFFORTS - do inglês *Experimental Free-Floating RoboT Satellite*

ERA - do inglês *European Robotic Arm*

ESA - do inglês *European Space Agency*

ETS-VII - do inglês *Engineering Test Satellite*

EVA - do inglês *ExtraVehicular Activities*

ISS - do inglês *International Space Station*

JAXA - do inglês *Japan Aerospace eXploration Agency*

JEMRMS - do inglês *Japanese Experiment Module Remote Manipulator System*

LiPo - do inglês *Lithium Polymer battery*

LSB - do inglês *Less Significant Bit*

LVTTL - do inglês *Low Voltage Transistor Transistor Logic*

MBS - do inglês *Mobile Base System*

MDE - Manipulador Dinamicamente Equivalente

ME - Manipulador Espacial

MSB - do inglês *Most Significant Bit*

NASA - do inglês *National Aeronautics and Space Administration*

PAA - configuração Passiva-Ativa-Ativa

PIW - do inglês *Path Independent Workspace*

ROKVISS - do inglês *Robotic Components Verification on the ISS*

SPDM - do inglês *Special Purpose Dexterous Manipulator*

SRMS - do inglês *Shuttle Remote Manipulator System*

SSRMS - do inglês *Space Station Remote Manipulator System*

STS - do inglês *Space Transportation System*

TTL - do inglês *Transistor Transistor Logic*

UARM - do inglês *UnderActuated Robot Arm*

UARM-E - sigla UARM acrescida do sufixo *E* com o objetivo de caracterizar um sistema de mesmo tipo como *Espacial*

WATFLEX - do inglês *WATerloo Flexible Link EXperimental*

Lista de Símbolos

Sistema Eletrônico

M_B - torque do motor

V_{CC} - tensão de alimentação em corrente contínua

K_n - constante de velocidade do motor

$\Delta n / \Delta M$ - gradiente de velocidade/torque do motor

V_{in} - nível de tensão V_{in} na entrada *Velocidade* do driver do motor

n - velocidade de rotação do eixo do motor

R_{fase} - resistência entre as fases do motor

J_R - inércia do rotor do motor

J_L - inércia da carga no eixo do motor

chA - sinal de onda quadrada de saída do encoder

chB - sinal de onda quadrada de saída do encoder defasado em 90° de chA

T_e - largura de pulso do encoder

t_{CLK} - período do clock de entrada do decodificador de quadratura

f_{ENC} - frequência de operação do encoder

Parâmetros e Coordenadas do ME

n - número de elos do ME

m'_i - massa do elo i

C'_i - centro de massa do elo i

I'_{REFi} - momento de inércia do elo i em relação ao eixo de referência do elo

I'_{CMi} - momento de inércia do elo i em relação ao centro de massa do elo

J'_i - junta que conecta os elos $(i - 1)$ e i

L_i - vetor que conecta J'_i a C'_i

R_i - vetor que conecta C'_i a J'_{i+1}

M'_i - massa total

θ'_i - rotação do elo i em torno da junta J'_i

(ϕ', θ', ψ') - ângulos de Euler (Z-Y-Z) com respeito à orientação da base

q' - coordenadas generalizadas

p' - coordenadas cartesianas do efetuador

$(x'_{ef}, y'_{ef}, \theta'_{ef})$ - coordenadas cartesianas do efetuador no plano xy

(x'_b, y'_b, θ'_b) - coordenadas cartesianas da base no plano xy

Parâmetros e Coordenadas do MDE

n - número de juntas ativas do MDE

m_i - massa do elo i

C_i - centro de massa do elo i

I_{REFi} - momento de inércia do elo i em relação ao eixo de referência do elo

I_{CMi} - momento de inércia do elo i em relação ao centro de massa do elo

J_i - junta que conecta os elos $(i - 1)$ e i

W_i - vetor que conecta J_i a J_{i+1}

l_{ci} - vetor que conecta J_i a C_i

M_t - massa total

θ_i - rotação do elo i em torno da junta J_i

(ϕ, θ, ψ) - ângulos de Euler (Z-Y-Z) com respeito à orientação da base (J_1)

q - coordenadas generalizadas

p - coordenadas cartesianas do efetuador

$(x_{ef}, y_{ef}, \theta_{ef})$ - coordenadas cartesianas do efetuador no plano xy

(x_b, y_b, θ_b) - coordenadas cartesianas da base (J_1) no plano xy

Modelo do MDE

M - matriz de inércia simétrica definida positiva

ΔM - porção incerta da matriz de inércia

C - matriz de Coriolis e forças centrífugas

ΔC - porção incerta da matriz de Coriolis e forças centrífugas

J - Jacobiano da relação de velocidades entre juntas e efetuador

ΔJ - porção incerta do Jacobiano

τ - vetor de torques atuando nas juntas do MDE

F - vetor de forças generalizadas do efetuador no espaço inercial

b - índice referente à junta passiva (base)

m - índice referente às juntas ativas (manipulador)

Formulação do Problema

u - índice referente às variáveis livres

a - índice referente às variáveis controladas

p_a^d - trajetória de referência para variáveis controladas do efetuador

$\tilde{x}_{ef}$ - erro de estado

δ - vetor representante da soma de incertezas

ω - vetor referente ao erro de estimativa degradado pelas redes neurais e a distúrbios externos

Controle $\mathcal{H}_\infty$ Não Linear

γ - nível desejado de atenuação do distúrbio

$\bar{u}$ - saída de controle

Q - matriz de ponderação de $\tilde{x}_m$ simétrica definida positiva

R - matriz de ponderação de $\bar{u}$ simétrica definida positiva

V - função de Lyapunov

P - solução da equação de Riccati

T_0 - parte da solução da equação algébrica de Riccati

K - parte da solução da equação algébrica de Riccati

Controle Adaptativo $\mathcal{H}_\infty$ Não Linear baseado em Redes Neurais

$H(x_e, \Phi)$ - conjunto de n redes neurais

Φ_k - vetor de pesos ajustáveis da camada de saída

w^k - vetor de pesos constantes da camada de entrada

b^k - vetor de bias constantes da camada intermediária

$G = \tanh$ - função de ativação dos neurônios da camada intermediária

Z - matriz de ponderação da lei de atualização dos parâmetros

Resultados

θ_{traj} - auxiliar para geração da trajetória de referência

t_f - tempo de execução da tarefa

t_r - tempo gasto pelo robô para atingir a posição desejada

$\mathcal{L}_2[\tilde{x}_{ef}]$ - norma $\mathcal{L}_2$ do vetor de estados

$E[\tau_a]$ - soma dos torques aplicados

(K_p, K_d, K_i) - vetor de ganhos do controlador por torque calculado

Introdução

A construção e o controle de manipuladores robóticos para atuar no ambiente espacial tem sido objeto de intensa pesquisa há algumas décadas. Com o objetivo de auxiliar, ou mesmo substituir, astronautas em atividades extra-veiculares - EVAs (do inglês *ExtraVehicular Activities*), a utilização de robôs manipuladores em órbita envolve tarefas como posicionamento, inspeção, transporte de materiais, acoplamento e montagem, [1].

O primeiro manipulador robótico utilizado em ambiente espacial foi o Canadarm ou SRMS (do inglês *Shuttle Remote Manipulator System*). A bordo do ônibus espacial Columbia, o qual realizava sua segunda missão em órbita, STS-2, em 1981, o Canadarm demonstrou sua operabilidade, Figura 1. Capaz de transportar cargas de até 29 toneladas no espaço, o Canadarm está em operação até os dias atuais, já tendo desempenhado importante papel na construção da Estação Espacial Internacional (ISS - do inglês *International Space Station*) e nas missões de manutenção do telescópio espacial Hubble, Figuras 2 e 3. Após o acidente com o Columbia durante a missão STS-107 em 2003, a agência espacial americana NASA (do inglês *National Aeronautics and Space Administration*) equipou o Canadarm com sensores em um sistema prolongador do braço para inspecionar o exterior da nave em busca de danos no sistema de proteção termal, [3].

Figura 1: Em novembro de 1981, o Canadarm flutua no espaço pela primeira vez em operação da missão STS-2, [2].

Desde 2001, a ISS, por sua vez, conta com seu próprio sistema robótico. O Canadarm2 ou SSRMS (do inglês *Space Station Remote Manipulator System*) move equipamentos e suprimentos, auxilia astronautas em EVAs e opera instrumentos anexados à ISS. Sendo uma evolução do Canadarm, o braço robótico da ISS tem a capacidade de se locomover em torno da estrutura fixando sua extremidade livre em um dos vários conectores especiais distribuídos estrategicamente ao longo da superfície da ISS, e então desconectando a base, girando para a frente. Assim, as duas extremidades do manipulador são atuantes e através da conexão com a estrutura

Figura 2: Em fevereiro de 2001, o Canadarm retira o módulo de laboratórios *Destiny* do compartimento de carga do ônibus espacial Atlantis para compor a ISS, STS-98, [2].

Figura 3: Canadarm utilizado em uma EVA como plataforma para o astronauta na missão STS-61 de manutenção do telescópio espacial Hubble em dezembro de 1993 a bordo do ônibus espacial Endeavour, [2].

da ISS recebem energia e transmitem dados e vídeo. Além disso, o Canadarm2, acoplado à base de um sistema móvel (MBS, do inglês *Mobile Base System*), também é capaz de se movimentar por trilhos em uma estrutura de treliças, cobrindo toda a extensão da ISS sob responsabilidade dos Estados Unidos. Em 2008, o Canadarm2 foi equipado com uma “mãozinha” para executar operações mais delicadas e substituir astronautas em algumas tarefas, [4]. O Dextre ou SPDM (do inglês *Special Purpose Dexterous Manipulator*) é composto por dois braços manipuladores menores cuja base é acoplada à extremidade livre do Canadarm2. Cada braço tem uma ferramenta dedicada e é teleoperado por um astronauta dentro da ISS. A estrutura do sistema robótico da ISS é ilustrada na Figura 4.

Figura 4: Estrutura do SSRMS, [5].

O módulo japonês da ISS também possui seu próprio manipulador robótico, o JEMRMS (do inglês *Japanese Experiment Module Remote Manipulator System*). Composto por dois

braços, um grande e um pequeno que se encaixa no efetuador do maior, o equipamento forma um sistema do tipo macro-mini, [3]. Desenvolvido pela agência espacial japonesa JAXA (do inglês *Japan Aerospace eXploration Agency*), o braço principal do JEMRMS foi acoplado ao módulo experimental Kibo em 2008, Figura 5, e o braço menor em janeiro de 2010, [6].

Figura 5: Kibo e JEMRMS instalados na ISS durante missão STS-124, [6].

Um braço robótico europeu, o ERA (do inglês *European Robotic Arm*), está sendo desenvolvido pela agência espacial européia, ESA (do inglês *European Space Agency*), e será lançado em março de 2012 pela nave russa Proton. O ERA possui configuração e habilidades similares às do Canadarm2, mas será acoplada ao segmento russo da ISS, [7]. Anteriormente, em 2004, a agência espacial alemã, DLR (do alemão *Deutsches zentrum für Luft-und Raumfahrt*), já havia enviado à ISS um precursor do ERA. Foram conduzidos experimentos de telepresença nos quais operadores humanos na Terra operavam o braço com base em imagens estéreo e sensores de força bilateral, utilizando um joystick com refletor de força e um sistema embarcado de controle de torque, [8, 9]. O ROKVISS (do inglês *Robotic Components Verification on the ISS*) foi removido da ISS em novembro de 2010.

Sob um ponto de vista diferente, está o desenvolvimento de sistemas com base livre, ou seja, robôs manipuladores espaciais móveis que possam ser aplicados em serviços de resgate e manutenção de naves e satélites em órbita. Nos últimos anos, EVAs foram conduzidas para executar o procedimento de captura-reparo-alocação de satélites em mal-funcionamento (por exemplo Anik-B e Intelsat 6) e também para a manutenção do telescópio espacial Hubble (STS-61, 82, 103 e 109) utilizando como base os ônibus espaciais. No entanto, missões de serviço não-tripuladas ainda não estão operacionais, [3]. Especificamente, essas missões incluem tarefas como substituição de componentes, recarga de suprimentos, reabastecimento de combustível,

reparos, resgate e reposicionamento do satélite na órbita correta. Esse cenário apresenta um potencial para operações mais flexíveis de satélites, incluindo a ampliação de sua vida útil e sua transferência de órbita quando não estiver mais funcional. Desde 1981, através do projeto ARAMIS (do inglês *Automation, Robotics And Machine Intelligence Systems*), a NASA vem discutindo propostas e a aplicabilidade desse tipo de intervenção, [10].

Figura 6: Concepção artística do ETS-VII. O experimento de manipulação do satélite alvo foi conduzido em março de 1999. O objetivo do experimento era demonstrar a capacidade de manipulação de cargas grandes e pesadas por um braço robótico acoplado a um satélite, [6].

Voos experimentais como o realizado no final da década de 90 pelo satélite de testes de engenharia ETS-VII (do inglês *Engineering Test Satellite*), desenvolvido no Japão, forneceram bons resultados. O ETS-VII era composto por duas partes: um satélite de serviço e um satélite alvo, Figura 6. O satélite de serviço era equipado com um braço robótico de 2m de comprimento e foi capaz de executar tarefas de captura e acoplamento com o satélite alvo, além de experimentos que envolviam teleoperação, substituição de peças e o controle coordenado da reação do manipulador e da resposta em atitude do satélite, [11]. A missão espacial *Orbital Express*, gerenciada pela NASA e pela agência americana de pesquisas em projetos de defesa,

DARPA (do inglês *Defense Advanced Research Projects Agency*), foi lançada em 2007, Figura 7. A missão teve por objetivo validar a viabilidade técnica de operações robóticas em órbita para reabastecimento e reconfiguração de satélites. O sistema americano também era composto por dois agentes: o satélite ASTRO (do inglês *Autonomous Space Transport Robotic Operations*), equipado com um manipulador robótico capaz de capturar o satélite alvo e realizar a substituição de peças; e o satélite NextSat, um protótipo modular de uma próxima geração de satélites padronizados para receber serviços de manutenção. Experimentos como inspeção visual, captura, acoplamento, transferência de combustível e substituição de peças, foram executados com sucesso.

A DLR, por sua vez, vem trabalhando desde 2007 no projeto DEOS (do alemão *DEutsche Orbitale Servicing mission*), que tem por objetivo qualificar em órbita elementos mecatrônicos (hardware e software) para sistemas espaciais avançados de manutenção, [12, 13]. A missão consiste nas seguintes fases: aproximação, voo de inspeção, voo em formação, captura, estabilização e calibração da composição, manobra de voo da composição, controle ativo por telepresença, monitoramento de operações autônomas, e reposicionamento da composição na órbita,

Figura 7: Foto do sistema *Orbital Express* tirada pela câmera do braço robótico, [2].

Figura 8. Um dos objetivos desses experimentos é investigar diferentes estratégias de controle e os diferentes modos do sistema de controle de atitude e orientação, assim como determinar as manobras mais adequadas para um acoplamento suave e a subsequente estabilização do sistema composto.

Figura 8: Concepção artística da missão DEOS, com o satélite cliente (à esquerda) e o satélite de serviço (à direita), [9].

Tais operações exibem uma dinâmica complexa e diferenciada, resultando em problemas de modelagem e controle exclusivos dessa área. A principal característica dos manipuladores espaciais é o acoplamento dinâmico entre a base (satélite) e o braço robótico. O movimento do braço provoca movimentos coordenados na base espacial e vice-versa. A partir da análise apresentada em [14], podemos considerar dois modos representativos de atuação de manipuladores robóticos espaciais identificando alguns de seus problemas característicos de planejamento e controle: manipuladores de base livre controlada e manipuladores de base livre flutuante.

Em um manipulador de base livre controlada, a orientação da base espacial é ativamente controlada por rodas de reação e/ou jatos propulsores, limitando ou mesmo eliminando o movimento da base. Neste caso, a base espacial se mantém estacionária no espaço operacional e a maioria das leis de controle para manipuladores terrestres pode ser diretamente aplicada, [14]. No entanto, a utilização desses mecanismos aumenta o consumo de energia elétrica e combus-

tível, seja adicionando peso e complexidade ou por utilizarem o combustível diretamente, além de provocar perturbações no sistema. O consumo de grandes quantidades de combustível limita a vida útil do robô no espaço. Assim, grande parte das técnicas de controle desenvolvidas para esse modo de atuação busca o planejamento de trajetória para otimizar o consumo de energia, [15, 16]. O controle por teleoperação é mais simples, mas é necessário observar as restrições de movimento da base, limitado pela comunicação, [9].

Por outro lado, manipuladores com base livre flutuante são sistemas que permitem que a base espacial se mova livremente em resposta aos movimentos do braço, evitando a adição de perturbações e conservando combustível e energia elétrica, [17]. Considerando essa configuração, o controle coordenado de base e manipulador tem sido abordado no caso de manipuladores redundantes, [18, 19], e algoritmos de planejamento de trajetória têm sido desenvolvidos com o objetivo de minimizar a reação da base espacial durante a execução da tarefa do manipulador, [15, 20, 21, 22, 23].

Embora uma combinação desses dois modos de atuação dos sistemas robóticos espaciais possa ser empregada durante diferentes fases de uma missão, cada modo demanda um projeto de controle específico. Assim, o trabalho desenvolvido nesta tese é focado na otimização do controle do movimento dos manipuladores de base livre flutuante. Propõe-se uma tarefa de acompanhamento de trajetória considerando uma trajetória pré-definida para o efetuador.

Deve-se notar que quando o satélite está operando no modo flutuante, o mapeamento cinemático do espaço da tarefa para o espaço das juntas, onde o controle é executado, pode levar a múltiplas soluções devido à conservação do momento angular não-integrável, e ainda à inexistência de uma trajetória de referência no espaço das juntas. Com o objetivo de evitar as complicações envolvidas no mapeamento cinemático nesse caso, este trabalho formula o problema de acompanhamento de trajetória diretamente no espaço da tarefa. Assim as posições do efetuador do manipulador são diretamente controladas.

A dificuldade de recriar as condições do espaço em laboratórios terrestres e a necessidade de simulações precisas para o conhecimento do comportamento dos sistemas espaciais fazem de sua modelagem dinâmica um recurso importante.

Um método analítico de modelagem para manipuladores espaciais chamado Manipulador Virtual foi desenvolvido em [24]. O Manipulador Virtual é um robô com base fixa no espaço inercial cuja primeira junta é esférica e passiva, representando a natureza flutuante da base espacial. No entanto, somente a equivalência cinemática é considerada nessa abordagem.

Com base no conceito do Manipulador Virtual, os autores de [25] mapearam um manipulador espacial em um manipulador convencional, com base fixa, e mostraram que as propriedades cinemáticas e dinâmicas do manipulador espacial são preservadas neste mapeamento. Este manipulador é chamado Manipulador Dinamicamente Equivalente. O Manipulador Dinamicamente Equivalente vai além do conceito do Manipulador Virtual, pois modela o sistema espacial tanto cinematicamente quanto dinamicamente. Assim, o Manipulador Dinamicamente Equivalente pode ser fisicamente construído como um manipulador subatuado convencional e utilizado para estudar experimentalmente o desempenho dinâmico e a execução de tarefas de um manipulador espacial, sem que seja necessário recorrer a montagens experimentais complexas para simular o ambiente espacial.

O trabalho em [26], por sua vez, propõe um modelo de robô estendido para manipuladores espaciais composto por um pseudo-braço com seis juntas passivas, representando o movimento da base, e o manipulador. Esse modelo permite definir o robô espacial como um sistema subatuado e revela propriedades intrínsecas à dinâmica interna desse sistema. Em uma outra técnica, apresentada em [27], o sistema é considerado de forma invertida, uma vez que o manipulador espacial não possui base fixa. Assim, o sistema robótico é modelado a partir do efetuador até a base. Essa abordagem é chamada abordagem da cadeia invertida e descreve explicitamente o acoplamento dinâmico entre o efetuador e o braço do robô.

Entretanto, é preciso observar que uma vez que operam em condições extremas, manipuladores espaciais estão geralmente sujeitos a dinâmicas desconhecidas ou não-modeladas. Quando diferentes cargas são manipuladas, características de massa e inércia são difíceis de precisar. O ambiente de trabalho hostil pode causar distúrbios externos ao sistema e incertezas paramétricas podem aparecer devido ao desgaste mais acentuado dos componentes mecânicos e eletrônicos nessas condições. A maioria dos manipuladores espaciais são projetados para serem leves e, dessa forma, de baixa potência, em coerência com o ambiente de gravidade nula e com a necessidade de eficiência energética. Nesse caso, o atrito das juntas, o amortecimento, e outras incertezas não lineares do sistema são muito mais significativas em robôs espaciais do que em robôs industriais. Além disso, incertezas paramétricas aparecem não somente na equação dinâmica, mas também no mapeamento cinemático do espaço das juntas para o espaço da tarefa devido à ausência de uma base fixa. Portanto, é real a necessidade de sistemas de controle adequados e robustos o suficiente para que as chances de erro sejam minimizadas. Duas abordagens são geralmente utilizadas nos projetos de controladores para sistemas incertos sujeitos a distúrbios externos: controle adaptativo e controle robusto.

Controladores adaptativos estimam e compensam as dinâmicas incertas do sistema, [28]. Esse método têm sido direcionado ao controle de movimento de manipuladores de base livre flutuante em diversos trabalhos na literatura, [26, 29, 30, 31]. Porém, a maioria das abordagens requer as medidas de orientação, velocidade e aceleração da base livre flutuante, as quais são difíceis de se obter na prática. Em [30] e [31] os autores modelaram o sistema espacial como um braço estendido e aplicaram um controlador por torque calculado baseado em estimativas dos parâmetros desconhecidos para seguir a trajetória desejada. Os primeiros optaram por realizar estimativas *on-line*, enquanto em [31] foram utilizados valores de parâmetros identificados por um estimador *off-line*. Ambas estratégias não utilizam medidas de aceleração da base e do braço, no entanto em [31] aplica-se um filtro para obter seus valores a partir das medidas de posição e velocidade e então aplicá-los na lei de controle. O procedimento adaptativo é geralmente baseado na propriedade de parametrização linear, a qual demanda um conhecimento preciso da estrutura do modelo e considera que os parâmetros desconhecidos são constantes ou variam lentamente. Além disso, a presença de dinâmicas não-modeladas é comum em aplicações reais e seus efeitos diminuem o desempenho dessa técnica.

Outra linha de pesquisa em controle adaptativo é definida pelos sistemas inteligentes. Abordagens baseadas em sistemas fuzzy ou redes neurais são capazes de aprender e estimar conceitos do mundo real, construindo uma base de conhecimento que pode ser interpretada e modificada pelo usuário. Com essas abordagens é possível construir mecanismos que necessitem operar em sistemas cujos parâmetros variem rapidamente. Assim, sistemas inteligentes têm sido aplicados com sucesso na literatura para estimar universalmente modelos matemáticos de sistemas dinâmicos, [32, 33, 34, 35, 36]. Dentro do escopo deste trabalho, aplicações de sistemas inteligentes para identificar e controlar manipuladores espaciais de base livre flutuante podem ser encontradas em [37, 38, 39, 40, 41, 42].

Por outro lado, controladores robustos têm sido projetados para lidar com os efeitos de incertezas paramétricas e distúrbios externos. No caso de robôs espaciais, perturbações podem ser geradas, por exemplo, pelos movimentos da base, por ruídos nos sensores e atuadores devido a extremos de temperatura e luz, ou ainda pelos efeitos da microgravidade e dos impactos durante os procedimentos de captura e acoplamento.

O controle por modo deslizante é uma das principais abordagens para lidar com esse problema. No entanto, essa estratégia geralmente leva à ocorrência de vibrações, que podem aumentar o consumo de energia e danificar os atuadores. Além disso, devido ao conteúdo de alta frequência dessas vibrações, elas podem facilmente estimular modos flexíveis que, por sua vez, podem gerar instabilidade. Particularmente nas aplicações espaciais, esse efeito torna-se um

problema grave. Tendo sido tratado extensivamente na literatura, várias estratégias já foram aplicadas para minimizá-lo. Dentro do escopo desta tese, os autores em [43] propõem, por exemplo, um controlador por modo deslizante com redes neurais de função radial para eliminar o fenômeno das vibrações no problema de acompanhamento de trajetória de um robô espacial de base livre flutuante.

Outro método de controle robusto foi apresentado em [44] para um sistema espacial de dois braços com base na modelagem proposta em [26]. Em [45], um observador de distúrbios foi empregado em cada junta do manipulador de base flutuante para estimar incertezas na planta, dinâmicas não-modeladas e distúrbios externos. Nesta estratégia, um filtro passa-baixa realiza papel importante na supressão dos distúrbios ao assegurar a robustez e o desempenho do sistema. Em [46] um conjunto de controladores lineares invariantes no tempo é acoplado ao modelo do sistema espacial através do ajuste de ganhos de alimentação direta e de realimentações definidas adequadamente. Os autores em [46] atestam a robustez do método com base no fato de a estrutura do controlador não exigir o conhecimento dos parâmetros do robô. Na referência [47], por sua vez, propõe-se um controlador por realimentação através da Jacobiana transposta composto com um termo de compensação dinâmica. Para evitar o uso de medidas de aceleração da base, os autores definem uma nova velocidade de referência.

Uma técnica diferente de controle robusto, baseada no critério $\mathcal{H}_\infty$, tem sido amplamente utilizada em aplicações robóticas por garantir boas propriedades de rejeição de distúrbio. A norma $\mathcal{H}_\infty$ é uma medida da robustez aplicada à planta pelo sistema de controle. Ela define o nível de atenuação na relação entrada/saída entre as perturbações e a saída controlada. Aplicações espaciais dessa estratégia também estão presentes no controle da atitude de satélites e veículos espaciais, [48, 49, 50, 51, 52]. Com relação ao controle de movimento de manipuladores de base livre flutuante, no entanto, trabalhos que utilizaram esse método são difíceis de se encontrar além de [38, 41, 42, 53].

Nesse sentido, com a proposta de lidar com o problema de acompanhamento de trajetória de manipuladores de base livre flutuante sujeitos a incertezas paramétricas e distúrbios externos, o trabalho desenvolvido nessa tese apresenta uma solução de controle adaptativo robusto baseado no critério $\mathcal{H}_\infty$. A versatilidade das redes neurais é aliada à robustez aplicada por um controlador $\mathcal{H}_\infty$ não linear, compondo um conjunto de técnicas desenvolvidas de acordo com o conhecimento e a disponibilidade do modelo do robô para o controlador.

Desenvolver um sistema de controle que garanta o sucesso desejado nas ações do robô depende de uma plataforma para testes que possa ser facilmente modificada para avaliar os controladores em desenvolvimento. O objetivo deste trabalho, portanto, inclui também a construção

de uma plataforma de pesquisa experimental que possa ser utilizada na validação de técnicas de controle desenvolvidas para robôs espaciais com base livre flutuante.

Plataformas de teste para robôs espaciais na Terra devem garantir, [54]:

1. Os efeitos da gravidade devem ser compensados para simular o ambiente espacial;
2. A base do robô e o braço robótico devem ser livres para se movimentar em qualquer direção;
3. Os movimentos da base e do efetuador devem ser medidos e acompanhados a uma taxa adequada para que todo o estado do sistema seja reconstruído.

Segundo [54] e [55], uma mesa com sistema de sustentação por colchões de ar (*air-bearing table*) é o método mais eficiente para o teste de robôs espaciais devido à sua simplicidade. Embora, seja uma plataforma experimental planar, as mesas de ar tem sido amplamente utilizadas no estudo de braços flexíveis e de manipuladores espaciais. Uma vez que o atrito é praticamente eliminado pelo colchão de ar, a ausência de gravidade é simulada para robôs planares. Esse tipo de plataforma experimental é tipicamente aplicada em testes de sistemas pequenos e leves e é mais adequada para a fase de estudo de sistemas do que para as fases de teste de voo. Centros espaciais e universidades desenvolveram instalações onde os experimentos são executados nesse tipo de sistema. Pesquisas na Universidade de Waterloo, no Canadá, por exemplo, foram realizadas utilizando a plataforma WATFLEX (do inglês *WATerloo Flexible Link EXperimental*) sob contrato da CSA, [56]. A construção da plataforma WATFLEX, por sua vez, é baseada em [57], trabalho desenvolvido na Naval Postgraduate School. A plataforma experimental EFFORTS (do inglês *Experimental Free-Floating Robot Satellite*) foi utilizada para a concepção do sistema ETS-VII no Instituto de Tecnologia de Tóquio, [11]. Pode-se citar ainda as plataformas experimentais utilizadas na Universidade de Stanford, [58] e no Instituto de Tecnologia de Massachusetts - MIT, [59]. A Figura 9 apresenta fotos dos sistemas citados. Mais exemplos e referências podem ser encontrados em [3], [54] e [55].

A partir de experimentos realizados em plataformas experimentais baseadas no sistema de sustentação por colchões de ar, pesquisadores em [54] estabeleceram que esse tipo de sistema é robusto, simula eficientemente a condição de micro-gravidade em 2D, possibilitando testes de longa duração e, portanto, é adequado para provar resultados teóricos. No entanto, não permite avaliações de movimentos fora do plano e demanda a construção de protótipos dedicados.

No caso de testes de sistemas tridimensionais, uma possibilidade é a utilização de sistemas de suspensão por cabos, [54]. Nesse tipo de plataforma experimental, um sistema de base fixa

Figura 9: Plataformas experimentais com sistema de sustentação por colchões de ar: WAT-FLEX, [56], EFFORTS, [11], Universidade de Stanford, [58], e MIT, [59].

gera uma força com a mesma amplitude e na direção oposta à força peso de cada subsistema do robô (base e elos). O sistema aplica essas forças nos centros de massa de cada subsistema enquanto seus valores se mantêm constantes durante o movimento tridimensional do braço robótico espacial. Essa técnica, no entanto pode gerar instabilidades e vibrações. Uma alternativa é a utilização de plataformas passivas, baseadas em cabos com menor rigidez ou molas. A Figura 10 ilustra uma plataforma experimental baseada no sistema de suspensão por cabos.

Figura 10: Plataforma experimental com sistema de suspensão por molas passivas durante uma sessão de testes na Universidade de Pádova, Itália, [54].

Inicialmente introduzidos no treinamento de astronautas, os vãos parabólicos tem se tornado uma ferramenta útil para a realização de experimentos científicos e tecnológicos de curta duração em ambiente de microgravidade. As condições de microgravidade dentro de um avião em voo parabólico são ideais para o teste de cargas antes de operações em órbita. No entanto,

muitos robôs espaciais construídos para missões de serviço em EVAs não podem ser testados durante vôos parabólicos devido a seu peso e dimensões. Nesse caso, protótipos miniaturizados são necessários. Além disso, experimentos que demandem muito tempo de operação não são adequados para testes em vôos parabólicos.

Testes em ambientes subaquáticos, por outro lado, permitem a realização de experimentos por tempo ilimitado em condições de flutuação neutra semelhantes à gravidade nula do espaço. No entanto, exigem a construção de protótipos dedicados, resistentes à água, e o tratamento dos efeitos dos distúrbios que o movimento do robô provoca no ambiente.

O trabalho desenvolvido nesta tese visa a aplicação e a avaliação de técnicas de controle para manipuladores espaciais de base livre flutuante, estando então situado na fase de estudos do sistema. Assim, considerando também a possibilidade de comparação direta com os resultados obtidos em [60] e a viabilidade da técnica, uma plataforma experimental baseada no sistema de sustentação por colchões de ar foi projetada.

A sequência do texto é dividida em 2 partes: a primeira parte descreve a concepção e a implementação do projeto da plataforma experimental para manipuladores de base livre flutuante, a segunda parte apresenta o desenvolvimento dos controladores adaptativos $\mathcal{H}_\infty$ não lineares baseados em redes neurais com a formulação do problema no espaço da tarefa. Resultados são apresentados no Capítulo 7 considerando um manipulador espacial com duas juntas. O capítulo final apresenta as conclusões da tese e um prospectivo para trabalhos futuros.

Parte I

Plataforma Experimental para Manipuladores Espaciais de Base Flutuante

A primeira parte desta trabalho apresenta o projeto de um sistema mecatrônico concebido para ser utilizado na validação de técnicas de controle para robôs espaciais com base livre flutuante.

O robô manipulador de base flutuante projetado é planar e composto por uma base, juntas conectadas por elos e um efetuador. A base não possui qualquer sistema de atuação, apenas flutua livremente. Suas juntas podem ser configuradas como ativas ou passivas, possibilitando também o estudo de sistemas subatuados. A estrutura do robô está projetada para possibilitar diversas configurações do manipulador com um braço ou com dois. O efetuador está no extremo do manipulador e, entre outras aplicações, poderá ser substituído por um sensor de força.

As especificações de projeto enfatizam a necessidade de que nenhuma conexão física, cabos ou fios, exista entre o robô e o ambiente, possibilitando sua flutuação sem qualquer interferência mecânica indesejada. Nesse sentido, toda a eletrônica de comando e a alimentação dos componentes do robô são alocadas em sua base flutuante, baseando a comunicação do sistema com o computador remoto em um padrão de comunicação sem fio.

A sequência do texto descreve os módulos que compõem a plataforma de pesquisa experimental, apresentando um esquema geral e o desenvolvimento dos sistemas mecânico, eletrônico e computacional.

1 *Esquema Geral*

O robô manipulador de base flutuante proposto neste trabalho é essencialmente composto por um sistema mecânico, um sistema eletrônico e um ambiente de simulação e controle estabelecido em um computador remoto. A estrutura geral do projeto foi definida em módulos e é apresentada na Figura 11.

Figura 11: Esquema Geral.

O módulo de flutuação é responsável por manter toda a estrutura física do robô flutuando sobre um colchão de ar formado entre as bases de suas juntas e uma plataforma de apoio.

O módulo de atuação é composto pelos motores presentes em cada junta e é responsável pelo movimento dos elos do manipulador.

O módulo de sensoriamento é responsável por realimentar o módulo de controle, fornecendo dados de posicionamento da base e das juntas. É composto por duas partes, uma no robô e outra externa. O módulo de sensoriamento no robô é baseado em encoders rotativos localizados em cada junta que fornecem informações da posição angular de cada elo. O módulo de

sensoriamento externo é definido por uma câmera de alta resolução posicionada no teto sobre a mesa de apoio e fornece imagens do robô. Dados de posicionamento da base e das juntas são extraídos dessas imagens após processamento pelo módulo de controle. O módulo de sensoriamento pode ainda ser complementado por um sensor de força instalado no efetuator em caso de aplicações que envolvam restrições de posição e controle de força.

O módulo de comunicação estabelece um sistema de comunicação sem fio responsável pela transferência de dados entre o robô e o módulo de controle, e vice-versa.

O módulo de comando realiza as interfaces entre os módulos internos do robô. Atua no tratamento dos dados advindos do módulo de comunicação, transformando-os em sinais elétricos compatíveis ao módulo de atuação. No sentido inverso, realiza a leitura dos encoders e formata esses dados para o módulo de comunicação. Além disso, monitora, regula e aciona o módulo de energia do robô.

O módulo de energia gera e distribui o nível de tensão característico a cada módulo interno do robô. Baterias recarregáveis de alto desempenho definem a fonte de energia do sistema.

O módulo de controle está situado em um computador remoto que se comunica com o robô através de um padrão de comunicação sem fio. É responsável pela interpretação dos dados do módulo de sensoriamento e pela conseguinte implementação da lei de controle em estudo, fornecendo ao robô valores de torque para cada uma de suas juntas. Apresenta uma interface amigável para o usuário, permitindo que toda a caracterização do experimento seja realizada e ilustra o comportamento do robô em tempo real.

1.1 Especificação

De maneira geral as especificações do sistema como um todo são dadas por:

- Nenhuma conexão física, cabos ou fios, deve existir entre o robô e o ambiente, possibilitando sua flutuação sem qualquer interferência mecânica indesejada;
- A arquitetura do sistema deve ser configurável possibilitando montagens do manipulador com um braço ou dois, com número variável de juntas.

1.2 Concepção

O propósito da fase de concepção do sistema é uma descrição da tecnologia, do princípio de funcionamento e empacotamento mecânico de cada parte constituinte da plataforma. Através de pesquisas realizadas no estado da arte das tecnologias envolvidas e com a experiência de projetos anteriores desenvolvidos no LASI (Laboratório de Sistemas Inteligentes - USP - São Carlos), foram identificadas alternativas de concepção. As melhores soluções foram definidas a partir de análises de viabilidade e custo.

1.2.1 Módulo de flutuação

O objetivo do módulo de flutuação é suportar a estrutura do braço robótico de forma a minimizar os efeitos gravitacionais. Considerando que não deve haver um acoplamento mecânico entre o robô e qualquer equipamento externo, para evitar interferências no movimento, a solução adotada é baseada na técnica de flutuação sobre um filme de ar utilizando uma mesa do tipo *Ealing*, Figura 12. A mesa de ar é caracterizada por criar uma fina película de ar pressurizado sob um corpo de prova com base plana e distribuição de massa uniforme que deve flutuar com mínimo atrito e perturbação. Um compressor de fluxo contínuo alimenta a câmara sob o tampo furado da mesa, gerando um fluxo de ar com direção mesa-corpo de prova em cada furo, criando o filme de ar entre a mesa e a base do corpo de prova. O nivelamento e a aspereza da plataforma são características importantes para garantir a minimização dos atritos e dos distúrbios gravitacionais, o que justifica a estrutura complexa de nivelamento e o tipo de material selecionado para o tampo da mesa.

Figura 12: Módulo de Flutuação - Mesa Ealing.

Como solução alternativa, uma adaptação no projeto mecânico das juntas e da base do robô foi concebido para possibilitar a formação do colchão de ar a partir de um fluxo de ar vindo das juntas e da base do robô, além do sistema para formação do colchão de ar a partir do fluxo de ar vindo da mesa. Nesse caso, um compressor de fluxo contínuo alimenta as câmaras de ar em cada junta/base, os tampos inferiores das juntas e da base são furados para permitir o fluxo de ar das juntas e da base para a mesa. No entanto, as mangueiras de conexão entre as juntas/base e o compressor formam uma conexão física com o ambiente e são dispostas de forma a minimizar qualquer interferência no movimento do robô (utiliza-se a suspensão por um cabo). No projeto original, os tampos inferiores das juntas e da base são sólidos e planos, o fluxo de ar vem da mesa e não existe qualquer conexão física entre o robô e o ambiente.

Portanto, o módulo de flutuação é definido pela técnica de flutuação sobre um colchão de ar. A estrutura mecânica proposta para o robô possibilita duas configurações distintas visando a formação do colchão de ar. A configuração básica utiliza um fluxo de ar vindo da mesa de apoio. A configuração opcional permite a constituição do fluxo de ar a partir das juntas e da base do robô.

1.2.2 Módulo de atuação

A especificação dos atuadores das juntas rotativas do manipulador inclui capacidade de torque em baixa velocidade de rotação, mínimo acoplamento físico entre o rotor e o estator e inexistência de sistema de redução para possibilitar a realização de experimentos de subatuação. Considerando ainda a metodologia de flutuação já definida, é interessante atentar para o fato de que a estrutura física do atuador deve permitir sua caracterização em formato de *puck*. Características como base larga e distribuição de massa uniforme são essenciais. Nesse sentido, o motor elétrico *Maxon EC 90 flat*, ilustrado na Figura 13 e descrito na Tabela 19 (Apêndice B), foi selecionado para compor as juntas rotativas do manipulador.

Figura 13: Módulo de Atuação - Motor Maxon EC 90 flat.

1.2.3 Módulo de sensoriamento

Sensoriamento interno

Considerando o conhecimento já estabelecido pela equipe em montagens anteriores, definiu-se pela utilização de encoders incrementais na função de sensoriamento de posição. Encoders são sensores de posição que fornecem um sinal de onda quadrada para posterior processamento pelo módulo de comando. São caracterizados por fornecerem um sinal de posição relativa apropriado para tarefas de posicionamento, reconhecimento da direção de rotação, informação de velocidade através do número de pulsos por unidade de tempo, além de ser a solução padrão para diferentes aplicações.

O princípio opto-eletrônico é baseado na aplicação de um sinal luminoso de LED através de uma roda codificada rigidamente conectada ao eixo do motor, Figura 14(a). O receptor (fóto-transistor) transforma sinais luminosos claro/escuro em impulsos elétricos correspondentes, que são amplificados e processados. A Figura 14(b) ilustra o encoder adquirido cujas características principais são identificadas na Tabela 20 (Apêndice B).

Figura 14: Módulo de Sensoriamento - (a) Esquema de funcionamento do encoder; (b) Encoder E6S: encapsulamento e disco.

Sensoriamento externo

A localização da base livre flutuante é realizada encontrando a posição de centros de círculos em etiquetas fixadas sobre o robô em imagens bi-dimensionais. Tais imagens são adquiridas através de um sistema de visão computacional e a metodologia aplicada é baseada na Transformada de Hough Circular. A Tabela 21 (Apêndice B) apresenta as características técnicas do sistema utilizado. As especificações de campo de visão e da precisão de posicionamento foram definidas considerando o tamanho da plataforma de apoio e a precisão de posicionamento fornecida pelos encoders. Essas especificações determinaram a seleção da câmera a partir de sua capacidade de resolução em imagens em tempo real e a seleção do conjunto de lentes. As

rotinas de processamento das imagens são implementadas como parte do software de controle no computador remoto. A Figura 15 mostra o equipamento adquirido.

Figura 15: Módulo de Sensoriamento - Câmera SMX-160C.

Sensor de força

O sensor de força é aplicado entre o efetuator e o ambiente de trabalho; ou seja o sensor mede as forças e momentos exercidos pelo efetuator do braço robótico no ambiente (ou vice-versa). Baseado em unidades piezoelétricas, as quais são caracterizadas por um sinal elétrico que decai no tempo, o sensor de força é utilizado para medição de grandezas dinâmicas (forças dinâmicas, impactos e acelerações) em três eixos ortogonais. As aplicações previstas para o robô espacial de base livre flutuante incluem a implementação de controladores onde o sensoriamento da força no efetuator torna-se fundamental: controle de impedância [61, 62]; controle de impedância múltiplo [63] e controle de restrição de posição e força [64, 65, 66].

A Figura 16 apresenta um modelo com vista interna da estrutura do sensor e sua aplicação. A parte identificada em amarelo no modelo representa a parte móvel do sensor, onde as forças de reação entre o efetuator e o ambiente são aplicadas. A parte fixa do sensor sustenta as unidades de medição de força (sensores de força piezoelétricos unidirecionais) e também acopla o sensor ao efetuator do braço robótico.

Figura 16: Módulo de Sensoriamento - Sensor de força: vista interna e aplicação.

1.2.4 Módulo de comunicação

O módulo de comunicação estabelece uma rede de transferência de dados entre o manipulador espacial, seus sensores e atuadores, e o computador remoto que implementa o módulo de controle. A configuração completa do manipulador permite até 6 juntas rotacionais, conseqüentemente, 6 motores e 6 encoders. A informação de torque que é transmitida do computador para cada motor possui até 64 bits. O posicionamento de cada junta é captado pelo encoder e enviado para o computador, esta informação possui até 16 bits. A especificação é de que esses dados sejam transmitidos a cada 15 ms. Assim, tem-se:

$$[(64\text{bits} \times 6) + (16\text{bits} \times 6)]/15\text{ms} = 32000\text{bits/s}.$$

Ou seja, é necessário um sistema de comunicação sem fio que garanta a taxa de transmissão mínima de 32 Kbits/s e que demande o mínimo de energia, já que o robô é alimentado por baterias. A Tabela 1 mostra uma comparação entre os padrões de comunicação mais atuais encontrados na literatura.

Tabela 1: Módulo de Comunicação - Comparação entre os principais padrões de comunicação sem fio.

	Taxa de transmissão	Alcance	Consumo
Wi-Fi	11Mbits/s - 54Mbits/s	100m - 50m	alto
Bluetooth	1Mbits/s	10m	médio
Zigbee	20Kbits/s - 250Kbits/s	10m - 100m	muito baixo

Com base em fatores como consumo de energia, taxa de transmissão e simplicidade de implementação, optou-se pelo sistema de comunicação baseado no padrão Zigbee. O dispositivo selecionado é composto por um módulo microcontrolador da marca *Rabbit* que traz o modem Zigbee como componente *on-board*, Figura 17. O kit fornece ferramentas de hardware e software necessárias para aplicação direta de suas funcionalidades. A Tabela 22 (Apêndice B) identifica as principais características técnicas do dispositivo adquirido.

1.2.5 Módulo de comando

O microcontrolador Rabbit[®] 4000 é utilizado como centro de comando do robô, Figura 17. É responsável por receber os sinais enviados pelo computador remoto através do modem Zigbee, tratá-los e transmiti-los a amplificadores que acionam os motores. No sentido contrário, recebe a informação dos encoders através de decodificadores de quadratura e a transmite ao computador utilizando o protocolo Zigbee. As interfaces elétricas implementadas pelo módulo

Figura 17: Módulo de Comunicação - RCM4510W: Módulo RabbitCore com Zigbee.

de comando são descritas no Capítulo 3. As rotinas internas do microcontrolador são programadas em *Dynamic C* e a estrutura do software embarcado é detalhada no Capítulo 4. No Apêndice B, a Tabela 23 identifica as características principais do módulo.

1.2.6 Módulo de energia

Considerando que o projeto consiste na construção de um equipamento para pesquisa e desenvolvimento, propondo-se à realização de experimentos sequenciais em períodos indeterminados com uma longa vida útil, fez-se necessária a especificação de baterias recarregáveis como fonte de alimentação, uma vez que a conexão com fontes de alimentação externas não é desejável. São aspectos determinantes para a escolha do tipo de bateria: o peso, a capacidade de energia, a vida útil e a disponibilidade.

Entre os vários tipos de baterias recarregáveis, podem ser citadas as cinco tecnologias mais utilizadas: prata-zinco (AgZn), lítio-ion, níquel-cádmio (NiCd), níquel-metal hidreto (NiMH) e chumbo ácido (Pb-Ácido). As Tabelas 2 e 3 apresentam dados quantitativos e qualitativos que caracterizam cada tipo de bateria, permitindo a comparação entre as tecnologias.

Tabela 2: Módulo de Energia - Comparação entre as principais tecnologias de bateria.

Tecnologia	AgZn	Lítio-ion	NiCd	NiMH	Chumbo
Tensão nominal da célula (V/célula)	1.5	3.5	1.2	1.2	2.0
Tensão 100% de descarga (V/célula)	0.9	2.5	1.0	1.0	1.7
Densidade de energia (Wh/kg)	75	110	50	60	35
Densidade de energia (Wh/l)	200	180	100	130	80
Taxa de auto descarga (%/mês)	1	1	15	30	2
Vida útil (em ciclos) (25% descarga)	100	1000	2000	500	1200
Vida útil (em ciclos) (100%descarga)	50	300	500	500	300
Custo/Wh (US\$/Wh)	2,00	1,00	1,00	1,50	0,50

Tabela 3: Módulo de Energia - Pontos positivos e negativos de cada tecnologia de bateria.

Prata Zinco (AgZn)	
Positivo:	Alta confiabilidade
Negativo:	Alto custo
Lítio-íon	
Positivo:	Baixo peso e tensão elevada
Negativo:	Alto custo e carga crítica
Níquel Cádmio (NiCd)	
Positivo:	Alto pico de corrente e alta vida útil
Negativo:	Efeito memória e tóxico
Níquel-Metal Hidrido (NiMH)	
Positivo:	Não agride o ambiente
Negativo:	Alto custo, auto descarga e carga complicada
Chumbo Ácido (Pb-Ácido)	
Positivo:	Baixo custo, longa vida de armazenamento e tecnologia dominada
Negativo:	Elevado peso

Baterias de Lítio-íon apresentam baixo peso e tensão elevada, e embora exijam alto investimento, suas características energéticas e físicas se enquadram perfeitamente na aplicação. A bateria de polímero de lítio (LiPo) utiliza um eletrólito sólido (polímero) no lugar do eletrólito líquido, proporcionando baterias mais seguras (em relação ao risco de explosão) e com dimensões menores. Por ser amplamente utilizada como fonte de energia de aeromodelos, sua obtenção é facilitada em casas de *hobbistas*. Carregadores-balanceadores especializados são encontrados comercialmente, capazes de carregar separadamente cada célula do conjunto com a precisão requerida, evitando oscilações e sobrecarga. Faz parte também do Módulo de Energia, um circuito eletrônico ligado ao Módulo de Comando que regula e monitora a energia disponibilizada pelas baterias (a ser descrito no Capítulo 3). A Figura 18 mostra os equipamentos adquiridos e a Tabela 24 (Apêndice B) apresenta sua descrição técnica.

Figura 18: Módulo de Energia - Bateria de polímero de lítio (LiPo) e carregador.

1.2.7 Módulo de controle

O módulo de controle é responsável pela interface entre as técnicas de controle e o robô manipulador de base livre flutuante. Essa interface é definida pelo Ambiente de Simulação e Controle de Manipuladores Espaciais (ASCME), Figura 19. O ASCME contempla a implementação de um manipulador espacial de 2 elos baseado em um manipulador de base fixa (UARM - Apêndice A) e apresenta a possibilidade da comparação visual entre o comportamento do UARM e o comportamento do manipulador espacial modelado segundo [25]. O simulador está implementado em linguagem nativa do Matlab[®] (*m-file*), mas funções diretamente relacionadas à interface com o robô são escritas em linguagem C para assegurar a comunicação em tempo real. No caso do manipulador de base livre flutuante, rotinas de comunicação com a porta USB do computador para transmissão das mensagens em protocolo Zigbee foram desenvolvidas. A plataforma possui em sua interface diversas funcionalidades que permitem ao usuário configurar o sistema e entender melhor o comportamento do experimento. O Capítulo 4 descreve com detalhes a interface e suas funcionalidades.

Figura 19: Módulo de Controle - ASCME.

2 *Sistema Mecânico*

2.1 Projeto

Sob o ponto de vista do projeto mecânico, duas estruturas distintas são consideradas: a base e o manipulador. O desenho das peças foi desenvolvido utilizando o sistema CAD 2D/3D Solid Edge[®]. O Apêndice C apresenta as dimensões de todas as peças em detalhe.

2.1.1 Base

A estrutura da base espacial é formada sob o conceito de prateleiras, Figura 20. Sua arquitetura abriga os módulos de alimentação, comando e comunicação. Além disso, permite o acoplamento de uma ou duas juntas, possibilitando as configurações com um ou dois braços, respectivamente. Na prateleira inferior estão acomodadas as baterias pois representam a maior parte da massa da estrutura. São fixadas por braçadeiras prafusadas na base. As duas prateleiras superiores sustentam os *drivers* dos motores, fixados por quatro parafusos cada. No topo se encaixará a placa de circuito impresso que comporta todo o circuito eletrônico do sistema de comando e de comunicação. O material selecionado para as prateleiras e para a base é acrílico cristal por ser de fácil moldagem, possibilitando a planicidade requerida para a base da estrutura. As dimensões da estrutura foram definidas de modo a possibilitar a equivalência cinemática e dinâmica com o UARM.

A Figura 20(a) mostra o projeto da base para operação no modo de flutuação original, com o fluxo de ar vindo da mesa. A Figura 20(b) ilustra a estrutura da base espacial no caso de o fluxo de ar para o sistema de flutuação ser alimentado pelo robô. Dessa forma, faz parte de sua estrutura uma câmara de ar identificada na Figura 21. O ar vindo de um compressor entra através de um cânulo e sai pelos furos da base da câmara, formando um colchão de ar com a mesa de apoio. O caminho de saída de ar da câmara foi baseado no desenho de rolamentos a ar (*air-bearings*), como os utilizados nas juntas do UARM. Na Figura 22 (a) e (b) são mostrados em detalhe o interior da câmara de ar e a base furada, respectivamente. As peças da câmara, usi-

nadas em alumínio, estão conectadas por parafusos de pressão laterais, e a vedação é garantida pelo posicionamento de *o-rings*¹ nas interfaces mecânicas.

Figura 20: Projeto Mecânico - Base.

Figura 21: Projeto Mecânico - Câmara de ar.

Figura 22: Projeto Mecânico - Detalhes de projeto da câmara de ar.

¹ Anel de borracha projetado para ser encaixado em um sulco e comprimido durante a montagem entre duas ou mais partes, vedando a interface.

2.1.2 Manipulador

Figura 23: Projeto Mecânico - Manipulador.

O manipulador espacial é composto por uma cadeia de juntas conectadas por elos e pelo efetuador, Figura 23. Sua estrutura é, então, baseada em módulos como o da Figura 24 compostos por uma junta e um elo.

Figura 24: Projeto Mecânico - Junta + Elo.

Junta

A junta é essencialmente composta por um motor e um encoder. Dessa forma o projeto mecânico da junta visa acomodar esses equipamentos da forma mais adequada possível. As juntas do robô são, a princípio, rotacionais. No entanto, a característica modular do sistema permite que, se houver necessidade de aplicação futura, uma ou mais juntas rotacionais sejam substituídas por juntas prismáticas.² A Figura 25 ilustra a montagem da junta projetada. Vistas em corte e explodida como na Figura 26 facilitam o entendimento da montagem da estrutura.

²A utilização de juntas prismáticas no braço do robô exigirá pequenas adaptações mecânicas para a conexão com os elos definidos nesse projeto, enquanto o sistema eletrônico (a ser descrito no Capítulo 3) é facilmente adaptável.

Figura 25: Projeto Mecânico - Junta.

Figura 26: Projeto Mecânico - Junta: Corte longitudinal e vista explodida.

O copo, que pode ser visto em corte na Figura 27, é usinado em alumínio, abriga o motor e serve como estrutura para fixação do elo. É interessante observar que essa peça é um prolongamento da câmara de ar ilustrada na Figura 21, possibilitando sua utilização nos dois processos de flutuação. Dessa forma, a câmara de ar faz parte de sua estrutura, sendo simplesmente desprezada quando o sistema de flutuação estiver operando no modo de projeto original (Seção 1.2.1). Uma base em acrílico como na Figura 28(a) é utilizada nesse caso.

Figura 27: Projeto Mecânico - Copo: Vista e corte longitudinal.

Figura 28: Projeto Mecânico - Base da junta.

Na parte superior do copo, uma tampa, usinada em alumínio, faz a interface de fixação entre o motor e o encapsulamento do encoder. Na Figura 29 pode ser visto o seu desenho.

Figura 29: Projeto Mecânico - Tampa da junta.

Com intuito de prover segurança mecânica, batedores de fim de curso, feitos em borracha, foram acoplados ao copo para limitar o movimento do elo que se segue à junta, Figura 30.

Figura 30: Projeto Mecânico - Fim de curso.

Elo

O elo é composto por um conjunto de conexões que unem uma junta à outra, Figura 31. Sua montagem é ilustrada na Figura 32.

Figura 31: Projeto Mecânico - Elo.

Figura 32: Projeto Mecânico - Elo: Vista explodida.

A primeira conexão une a junta ao elo anterior e é fixada no copo da junta, Figura 33. Observe que a entrada do cânulo de ar é feita através dessa junta quando o pulmão de ar do copo for utilizado.

Figura 33: Projeto Mecânico - Conexão Junta-Elo.

O elo propriamente dito é a peça ilustrada na Figura 34. É feita em acrílico e seu desenho visou a redução de massa sem comprometer a resistência da peça. Os elos intermediários, entre juntas, têm, a princípio, o mesmo comprimento. O elo entre a última junta e o efetuator tem um comprimento diferenciado para manter a equivalência dinâmica proposta com o UARM. Optou-se por projetar os elos como peças totalmente separadas das conexões para possibilitar

configurações com outros comprimentos de braço, bastando para isso construir essa peça no comprimento desejado.

Figura 34: Projeto Mecânico - Elo.

A conexão ilustrada na Figura 35 é responsável por fixar o elo à junta em L, Figura 36. A junta em L é responsável por transferir o movimento do eixo do motor da junta anterior ao elo. Seu desenho também foi projetado para reduzir a massa.

Figura 35: Projeto Mecânico - Conexão Elo-Junta.

Figura 36: Projeto Mecânico - Conexão em L.

2.1.3 Montagem

Como foi dito, a estrutura do robô vem sendo projetada para possibilitar diversas configurações do manipulador. O robô com um braço pode ser montado com uma, duas, três, quatro, cinco ou seis juntas atuadas. E dessa forma, o robô terá quantos graus de liberdade quantas juntas atuadas tiver. O número de seis juntas foi proposto visando o controle de posição simultâneo da base e do efetuador, de forma que com seis graus de liberdade é possível controlar as

posições x_{ef} , y_{ef} e a orientação θ_{ef} do efetuador e ainda as posições x_b , y_b e a orientação θ_b da base espacial, atentando para o fato de o robô ser planar. O robô com dois braços pode ser configurado com combinações de juntas atuadas, desde que o total delas não exceda seis. Essa configuração possibilitará também o estudo de manipuladores cooperativos.

Figura 37: Projeto Mecânico - Robô espacial.

2.2 Implementação

Esta Seção apresenta fotos das peças usinadas conforme o projeto apresentado na Seção anterior.

2.2.1 Base

(a)

(b)

Figura 38: Sistema Mecânico - Base (foto).

Figura 39: Sistema Mecânico - Câmara de ar: Vistas superior e inferior (foto).

2.2.2 Manipulador

Figura 40: Sistema Mecânico - Manipulador (foto).

Junta

Figura 41: Sistema Mecânico - Junta: Montagem e vista explodida (foto).

Figura 42: Sistema Mecânico - Copo: Vistas superior e inferior (foto).

Figura 43: Sistema Mecânico - Tampa da junta (foto).

Figura 44: Sistema Mecânico - Fim de curso (foto).

Elo

Figura 45: Sistema Mecânico - Elo: Montagem e vista explodida (foto).

Figura 46: Sistema Mecânico - Conexão Junta-Elo (foto).

Figura 47: Sistema Mecânico - Elo (foto).

Figura 48: Sistema Mecânico - Conexão Elo-Junta (foto).

Figura 49: Sistema Mecânico - Conexão em L (foto).

2.2.3 Montagem

Figura 50: Montagem - Robô espacial: (a) 1 braço, 1 junta; (b) 1 braço, 2 juntas.

Figura 51: Montagem - Robô espacial: 1 braço, 3 juntas.

Figura 52: Montagem - Robô espacial: 1 braço, 4 juntas.

(a)

(b)

Figura 53: Montagem - Robô espacial: (a) 1 braço, 5 juntas; (b) 1 braço, 6 juntas.

(a)

(b)

Figura 54: Montagem - Robô espacial: (a) 2 braços, 1 junta; (b) 2 braços, 2 juntas.

(a)

(b)

Figura 55: Montagem - Robô espacial: (a) 2 braços, 3 juntas; (b) 2 braços, 1/5 juntas.

2.3 Modelagem

O processo de modelagem do robô demanda o conhecimento de algumas características estruturais de suas peças e também da forma como essas peças se relacionam.

A Tabela 4 a seguir lista individualmente a massa das peças produzidas e dos dispositivos. O valor apresentado foi medido em uma balança de precisão, considerando uma amostra de cada peça. Parafusos, porcas e arroelas utilizados na fixação dos conjuntos foram pesados junto com a peça que têm conexão direta. À massa do elo que conecta as juntas foi acrescentada a massa da cablagem que passa por ele (vide Figura 70).

Tabela 4: Modelagem - Massa

Peça	Massa (g)	Peça	Massa (g)
Base (1)	908	Tampa da junta	112
Base (2)	307	Base da junta	102
Prateleira	178	Base da câmara de ar	235
Braçadeira	50	Fim de curso	3
Poste	26	Elo da junta	57
Câmara de ar	185	Elo do efetuator	11
Conexão Junta-Elo	35	Conexão Elo-Junta	21
Copo	228	Conexão em L	56
Dispositivos	Massa (g)	Dispositivos	Massa (g)
Placa eletrônica	308	Motor	709
Driver	150	Encoder	57
Bateria	652		

Com base em princípios consagrados nas leis de movimento de Newton e no princípio da compatibilidade (não existem espaços livres entre elementos conectados), [67], o modelo mecânico do robô projetado nesse trabalho pode ser analisado considerando três corpos livres diferentes. Cada corpo livre é representado por um ponto de massa ideal que se move em relação a um eixo de referência fixo a ele.

O primeiro corpo representa a base espacial e a parte fixa da primeira junta, Figura 56(a). O eixo de referência é fixado no centro de massa do conjunto, Figura 56(b). Verifica-se nesse momento que é necessário separar o motor em dois corpos: o rotor e o estator. Uma vez que o rotor se move em relação ao estator, apenas o estator faz parte do conjunto fixo da junta, enquanto o rotor será considerado no segundo corpo livre. Dada a massa medida do motor, determinou-se através do modelo mecânico em Solid Edge[®] as respectivas massas do estator: 264g, e do rotor: 445g.

Figura 56: Modelagem - Corpo livre 1: Base espacial

O segundo corpo livre representa as juntas 2 a n, e é composto pelo rotor do motor da junta anterior, conexões do elo e parte fixa da junta, Figura 57(a). O eixo de referência é fixado no centro do eixo do rotor, Figura 57(b).

Figura 57: Modelagem - Corpo livre 2: Junta

O último corpo característico representa o efetuator e é composto pelo rotor do motor da junta anterior, conexões do elo e montagem do copo do efetuator, Figura 58(a). O eixo de referência é fixado no centro do eixo do rotor, Figura 58(b).

Figura 58: Modelagem - Corpo livre 3: Efetuator

Analisando os conjuntos no software Solid Edge[®], considerando as dimensões das peças e os valores de massa medidos, determinou-se os parâmetros característicos dos corpos, Tabela

5. Observando que o sistema é planar e simétrico em relação à coordenada y , a medida do centro de massa de cada conjunto é tomada ao longo do eixo x , a partir do eixo de referência. Os momentos de inércia, I , foram tomados em relação ao eixo de coordenadas de referência e em relação ao centro de massa. Os parâmetros estáticos de cada corpo foram determinados a partir do posicionamento do centro de massa, tal que L é a distância do eixo de rotação ao centro de massa do conjunto e R é a distância entre o centro de massa e o ponto de conexão do próximo conjunto, vide Figuras 56, 57, 58(b). Devido à simetria em y , assume-se que L e R são colineares. Na tabela a seguir, a legenda **Robô 1** diz respeito ao sistema de flutuação baseado na mesa de ar e **Robô 2** indica os valores dos conjuntos baseados no sistema de flutuação alternativo.

Tabela 5: Modelagem - Parâmetros do robô

Robô 1					
<i>Corpo</i>	<i>m(kg)</i>	<i>I_{REF}(kgm²)</i>	<i>I_{CM}(kgm²)</i>	<i>L(m)</i>	<i>R(m)</i>
1	4.780	0.0404	0.0404	0	0.150
2	1.380	0.0467	0.0182	0.144	0.111
3	1.011	0.0222	0.0115	0.103	0.112
Robô 2					
<i>Corpo</i>	<i>m(kg)</i>	<i>I_{REF}(kgm²)</i>	<i>I_{CM}(kgm²)</i>	<i>L(m)</i>	<i>R(m)</i>
1	5.616	0.0503	0.0503	0	0.155
2	1.516	0.0544	0.0194	0.152	0.103
3	1.144	0.0283	0.0130	0.116	0.099

3 *Sistema Eletrônico*

O sistema eletrônico implementa a interface entre o sistema mecânico e o módulo de controle. Contempla os módulos de comunicação e comando, realizando as interfaces exigidas pelos módulos de atuação, sensoriamento e energia.

3.1 Projeto

3.1.1 Módulo de atuação

O módulo de atuação, composto pelos motores de cada junta, deve ser capaz de aplicar aos elos os torques cujos valores são enviados pelo módulo de controle através do módulo de comunicação. Uma interface entre o módulo de comunicação e o módulo de atuação é implementada em três blocos, Figura 59.

Figura 59: Projeto Eletrônico - Módulo de Atuação.

O primeiro bloco, mais próximo aos motores, é composto por seis drivers. O controlador digital mostrado na Figura 60 (a), é um amplificador de 1-quadrante para controle digital de motores CC eletronicamente comutados e foi selecionado por sua compatibilidade com o motor EC flat 90. A Tabela 25 (Apêndice B) ressaltava suas características gerais. Pode ser configurado para diferentes modos de operação: controle de velocidade, controle de corrente, controle de velocidade em malha aberta. Optou-se pela configuração de controle de velocidade em malha aberta. Para tal, o posicionamento das chaves de configuração em cada driver deve ser como na Figura 60 (b). O potenciômetro **P3** ajusta o limite de corrente de saída do driver em uma faixa de 0...10A. Em nossa aplicação, **P3** deve estar posicionado em 100%. **P1** e **P2** não atuam nesse modo de operação.

Figura 60: Projeto Eletrônico - (a) DEC 50/5 e (b) Configuração para controle de velocidade em malha aberta.

A alimentação de cada driver é provida pela bateria e pode variar de 18 a 22.2 V. É interessante observar que a tensão nominal do motor EC flat 90 é de 48 V. No entanto, os experimentos a que o robô será submetido leva os motores a serem acionados com tensões muito inferiores à tensão nominal, visto que velocidades de rotação relativamente baixas são exigidas. Dessa forma, um nível V_{cc} de até 22.2 V é suficiente. A interface elétrica entre cada motor e seu respectivo driver é caracterizada na Tabela 33 (Apêndice D).

Considerando o posicionamento dos drivers na base espacial e a necessidade de facilitar a possibilidade de configurações variáveis do robô, a conexão entre os drivers e os motores é feita através de conectores na placa de comando na base espacial. Dessa forma, atuadores podem ser conectados ou desconectados sem qualquer alteração na estrutura da base flutuante.

O segundo bloco da interface na Figura 59 conecta os drivers ao microcontrolador Rabbit 4000[®]. É um bloco composto por um conversor digital-analógico (DAC) de 8 canais e demultiplexadores de saída *latch*. O DAC tem resolução de 10 bits e fornece a entrada de tensão para

os drivers segundo a velocidade requerida, operando na faixa de 0 a 5V. Os demultiplexadores fornecem os sinais de controle dos 6 drivers: *Desabilita* (contato 11), *Direção* (contato 12) e *Freio* (contato 13). Optou-se por implementar os circuitos demultiplexadores em um CPLD (do inglês *Complex Programmable Logic Device*) de forma a facilitar o desenvolvimento da placa de circuito impresso e permitir que qualquer ajuste seja de fácil execução. As Tabelas 26, 27 e 28 (Apêndice B) apresentam as características gerais dos CIs selecionados. As Tabelas 34 e 35 (Apêndice D) identificam a interface elétrica entre os drivers e estes CIs.

Com o driver operando no modo de controle de velocidade em malha aberta, a velocidade requerida para o motor é controlada pela entrada *Velocidade* (contato 15) do driver. A faixa de valores de 0...5V nessa entrada é equivalente à aplicação de tensão no motor na faixa de 0V... V_{CC} . A entrada de *Velocidade* tem resolução de 1024 passos, o que determinou a escolha da resolução do DAC de 10 bits. A velocidade máxima é determinada pela tensão de alimentação e pela constante de velocidade do motor. O menor passo de torque possível com essa resolução pode ser calculado por:

$$M_{B_{min}}[mNm] = \frac{V_{CC}K_n}{1024(\Delta n/\Delta M)}, \quad (3.1)$$

sendo $\Delta n/\Delta M$ o gradiente de velocidade/torque e K_n a constante de velocidade do motor (vide Tabela 19 (Apêndice B)). Através da equação 3.2 a seguir pode-se calcular o nível de tensão V_{in} na entrada *Velocidade* para um determinado torque M_B desejado:

$$V_{in}[V] = \frac{5(\Delta n/\Delta M)}{V_{CC}K_n}M_B. \quad (3.2)$$

O sinal *Desabilita* é responsável por bloquear o estágio de potência do driver. Quando o sinal *Freio* é ativado, o eixo do motor desacelera até o travamento, curto-circuitando os enrolamentos do motor. Os enrolamentos se mantêm curto-circuitados até que a função *Freio* seja desativada. O sinal *Direção* indica a direção de rotação do eixo do motor. Quando o nível muda, o motor é freiado e depois acelerado na direção oposta até que a velocidade requerida seja atingida novamente. Uma rampa é aplicada somente durante o procedimento de aceleração. Para os casos de frenagem do motor e mudança de direção deve-se observar a velocidade máxima permitida para o motor no momento da sinalização. A velocidade máxima permitida para frenagem é limitada pela corrente de curto-circuito máxima, $I_{cc} \leq 30A$, e pela máxima energia cinética, $W_k \leq 20Ws$. Assim, o valor da velocidade máxima permitida é calculado por:

$$n_{max}[rpm] = \min \left\{ \begin{array}{l} 30K_n(R_{fase} + 0.05) \\ \sqrt{\frac{365}{J_R + J_L}} 10000 \end{array} \right.$$

sendo R_{fase} a resistência entre as fases, J_R a inércia do rotor e J_L a inércia da carga.

Demultiplexadores foram designados para encaminhar cada um desses sinais de controle a todos os drivers com o intuito de economizar portas no microcontrolador. O CI SN74LS259 tem quatro modos de operação. No modo de endereçamento de *latch*, os dados na porta de entrada *D* (pino 13) são escritos no *latch* endereçado. O *latch* endereçado vai seguir o dado de entrada e todos os *latches* restantes que não foram endereçados manterão seu estado anterior. Essa característica foi decisiva para a seleção desse modelo de demultiplexador, pois devem manter o sinal de cada uma de suas portas de saída se eles não forem modificados pelo sistema de comando. As linhas de endereçamento dos três demultiplexadores são curto-circuitadas de forma que o mesmo endereço designa o mesmo driver para os três sinais de controle. O sinal de controle $\bar{C}$ (pino 14) dos demultiplexadores é mantido em nível alto.

O terceiro bloco na Figura 59 é o microcontrolador Rabbit 4000[®]. Ele é responsável por receber as mensagens do modem Zigbee enviadas pelo sistema de controle, traduzí-las para o formato adequado ao DAC e aos demultiplexadores, endereçar e controlar a liberação desse dados por parte do segundo bloco. A Tabela 36 (Apêndice D) identifica a interface entre o Rabbit 4000[®] e os demultiplexadores no CPLD.

Convém observar que o CPLD selecionado apresenta a característica de interface *Multi-VoltI/O*, ou seja, opera simultaneamente com dispositivos atuados por 3.3V e com dispositivos atuados por 5V. As entradas V_{CCINT} (pinos 3 e 43) devem estar sempre alimentados a 5 VCC, o que faz com que suas portas de entrada sejam compatíveis com níveis 3.3V e 5V (LVTTTL, TTL e CMOS). Os pinos V_{CCIO} (pinos 13, 26, 38, 53, 66 e 78) podem ser alimentados a 3.3 VCC ou 5 VCC, dependendo dos requisitos das portas de saída. Em nossa aplicação, $V_{CCIO} = 3.3$ VCC, ou seja o nível alto das portas de saída é de 3.3V e portanto é compatível com sistemas LVTTTL (Low Voltage Transistor Transistor Logic - 3.3V) e TTL (Transistor Transistor Logic - 5V). Dessa forma, o CPLD também faz a transição de nível de tensão na interface entre os dispositivos utilizados e o microcontrolador Rabbit 4000[®].

3.1.2 Módulo de sensoriamento

O sistema de sensoriamento no robô, composto pelos encoders em cada junta, deve ser capaz de realizar a leitura de posição angular dos eixos do motores para que essa informação seja enviada ao sistema de controle através do sistema de comunicação. A interface entre o sistema de comunicação e o sistema de sensoriamento é feita em três blocos, Figura 61.

Cada encoder fornece um par de sinais de onda quadrada, chA e chB, defasados em 90°. O primeiro bloco da interface, mais próximo aos encoders, é composto por seis decodificadores de quadratura. A Figura 62 e a Tabela 6 mostram os estados de quadratura dos sinais chA e

Figura 61: Projeto Eletrônico - Módulo de Sensoriamento.

chB.

Figura 62: Projeto Eletrônico - Operação do decodificador de quadratura.

Tabela 6: Projeto Eletrônico - Operação do decodificador de quadratura.

chA	chB	estado	decoder
1	0	1	pulso
1	1	2	pulso
0	1	3	pulso
0	0	4	pulso

Cada decodificador de quadratura realiza a decodificação dos sinais gerados por um encoder, transformando a informação em um dado de 16 bits resultante da contagem dos pulsos. O circuito multiplica a resolução dos sinais de entrada por 4. Com base no estado anterior e no

estado atual dos sinais chA e chB, o decodificador atualiza o sinal de um contador. Um pulso de contagem é gerado a cada transição de estado. Se o chA vem antes do chB a contagem é positiva. Se o chB estiver na frente do chA a contagem é negativa. A Tabela 29 (Apêndice B) ressaltas as características gerais do decodificador de quadratura selecionado e a Tabela 37 (Apêndice D) identifica a interface entre o encoder e o decodificador de quadratura.

Observação 1 *Um pulso de entrada deve estar estável por pelo menos 3 subidas de clock, portanto a largura de pulso do encoder (T_e - baixo ou alto) deve ser maior que 3 períodos de clock, $T_e > 3t_{CLK}$. O circuito interno do decodificador de quadratura impõe ainda outra restrição: deve haver pelo menos um período de clock entre dois estados de quadratura consecutivos, $T_{es} > t_{CLK}$. Para essa análise devemos observar que a velocidade máxima do motor é de 5000 rpm ou 30 kHz, e a velocidade máxima de leitura do encoder é de 100 kHz, logo, o encoder selecionado é capaz de ler a posição do eixo do motor até em velocidade máxima com folga. Então, considerando $T_e = 2T_{es}$, temos $T_{es} = \frac{t_{ENC}}{4} > \frac{3}{2}t_{CLK}$ e portanto, $f_{CLK} > 6f_{ENC}=600$ kHz. O CI selecionado pode operar em até 14MHz.*

Os níveis de tensão das portas dos decodificadores de quadratura não são compatíveis com os níveis de tensão praticados pelo microcontrolador. Portanto faz-se necessário a utilização de um circuito de deslocamento de nível entre os decodificadores de quadratura e o microcontrolador. Considerando que a saída dos decodificadores de quadratura é *tristate*, optou-se por conectar o barramento de saída de todos os decodificadores de quadratura a um só barramento de entrada. Assim, esse duto de dados passará também pelo CPLD, que alimentado com 3.3V, gera sinais de saída neste nível, porém suporta 5V em suas portas de entrada, reveja a Figura 61. A Tabela 38 (Apêndice D) identifica a interface entre o barramento de saída dos decodificadores de quadratura e o microcontrolador, passando pelo CPLD.

O controle de ativação das saídas dos decodificadores de quadratura é realizado através do pino $\overline{OE}$. Também com o intuito de economizar pinos no microcontrolador, o sinal $\overline{OE}$ é demultiplexado no CPLD, da mesma forma que os sinais de controle dos drivers vistos na Seção 3.1.1. A Tabela 39 (Apêndice D) identifica a interface entre o demultiplexador implementado no CPLD e o microcontrolador. Os sinais de inicialização dos decodificadores de quadratura, RST , são definidos como um único sinal comum a todos os decodificadores. Da mesma forma, os sinais de seleção de byte dos dados de saída, SEL , também são definidos como um único sinal comum a todos os decodificadores, uma vez que somente um dos CIs estará habilitado pelo sinal $\overline{OE}$. O processo de leitura da saída de dados do decodificador de quadratura é controlado como na Tabela 7 a seguir.

Uma lógica de inibição (sinal interno de inibição) no HCTL-2017 interrompe a transferên-

Tabela 7: Projeto Eletrônico - Processo de leitura do decodificador de quadratura.

Passo	$\overline{OE}$	SEL	Inibição	Saída
0	1	X	0	Z
1	0	0	1	MSB
2	0	1	1	LSB
3	1	X	0	Z

cia de dados do contador para o *latch* de saída do CI durante o tempo em que os bytes estão sendo lidos pelo microcontrolador (passando pelo CPLD). Essa lógica de inibição possibilita que o microcontrolador leia o MSB (byte mais significativo) e depois o LSB (byte menos significativo), uma vez que o barramento de saída do CI opera com 8 bits. Quando o sinal de inibição está baixo, os dados do contador são continuamente transferidos para o *latch* de saída (Passo 0). Ao ativar o sinal $\overline{OE}$, o dado do *latch* é habilitado ao barramento de saída. Se o indicador **SEL** está em 0, os sinais do byte mais significativo do dado são direcionados ao barramento. Se o indicador **SEL** está em 1, os sinais do byte menos significativo do dado são direcionados ao barramento de saída. Quando o CI detecta ambos os sinais de controle $\overline{OE}$ e **SEL** em nível baixo, o sinal interno de inibição é ativado (Passo 1). Esse processo bloqueia a transferência de novos dados para o *latch* de saída. O contador, no entanto, continua acompanhando os estados de quadratura dos sinais de entrada, chA e chB. O barramento de saída apresenta os dados estabilizados segundo a seleção de **SEL** (Passos 1-2). A lógica de inibição é reiniciada sob duas condições sequenciais identificadas na descida dos pulsos de *clock*: 1) $\overline{OE}=0$ e **SEL**=1; 2) $\overline{OE}=1$ (Passos 2-3).

Os níveis de tensão das portas de entrada de controle dos decodificadores de quadratura (CMOS) não são compatíveis com os níveis de tensão praticados pelo microcontrolador (LVTTL). Nesse caso, e ainda considerando que a saída do CPLD também é LVTTL, os sinais de controle *RST*, *SEL* e $\overline{OE}$ precisam passar por um circuito elevador de nível.

O CI SN74LVCH16T245 é um conversor LVTTL projetado para realizar uma comunicação assíncrona entre dois barramentos de dados. Os níveis lógicos de controle de direção, *DIR*, e de habilitação da saída, $\overline{OE}$, ativam as portas A ou as portas B ou colocam as duas portas em modo de alta impedância. O dispositivo transmite dados do barramento A para o barramento B quando as portas B estão ativadas, e do barramento B para o barramento A quando as portas A estão ativadas, veja Tabela 8. O circuito de entrada em ambas portas A e B está sempre ativado. Características gerais do CI selecionado são apresentadas na Tabela 30 (Apêndice B).

Em nossa aplicação, a direção de transmissão de dados no conversor LVTTL será sempre mantida, do CPLD e do microcontrolador (3.3V) para os decodificadores de quadratura (5V), e

Tabela 8: Projeto Eletrônico - Operação do conversor LVTTL.

Controle		Saída		Operação
$\overline{OE}$	DIR	Porta A	Porta B	
0	0	Habilitada	Z	Dados da porta B para o barramento A
0	1	Z	Habilitada	Dados da porta A para o barramento B
1	X	Z	Z	Portas isoladas

portanto, o sinal de direção do conversor LVTTL será mantido em 5V, **DIR** = V_{CCB} . As Tabelas 40 e 41 (Apêndice D) identificam as interfaces entre o microcontrolador e o conversor LVTTL e entre o demultiplexador implementado no CPLD e o conversor LVTTL, respectivamente. A Tabela 42 (Apêndice D) especifica a interface entre o conversor de nível e os decodificadores de quadratura.

3.1.3 Módulo de energia

O módulo de energia é responsável por fornecer energia a todos os sistemas internos do robô. É composto por duas partes distintas. A parte de fornecimento é responsável por captar a saída da bateria e gerar sinais regulados de 5 V e 3.3 V. A Tabela 9 a seguir mostra uma estimativa da potência máxima consumida por cada módulo que compõe os sistemas do robô identificados pelo nível da alimentação requerida, 5 V e 3.3 V. A parte de monitoração objetiva supervisionar o nível de tensão de saída da bateria. Seu intuito principal é identificar quando a bateria estiver descarregada a um nível intolerável e controlar o processo de troca de baterias nesse caso. Ainda, monitora o nível de tensão que é enviado aos drivers para controle da alimentação dos motores, entrada *Velocidade*. A Figura 63 mostra um esquema do módulo de energia.

Tabela 9: Projeto Eletrônico - Análise de demanda de potência.

5 V		3.3 V	
Encoder E6S	$6 \times 88 \text{ mA}$	Módulo RCM4510W	150 mA
DecodQ HCTL 2017	$6 \times 100 \mu\text{A}$	Conversor SN74LVCH16T245	20 μA
Conversor SN74LVCH16T245	20 μA		
CPLD EPM7128SLC84	300 mA		
DAC TLV5631	21 mA		
Chaves	$6 \times 500 \mu\text{A}$		
Total Corrente	852.62 mA	Total Corrente	150.02 mA
Total Potência	4.26 W	Total Potência	0.50 W

Figura 63: Projeto Eletrônico - Módulo de Energia.

Fornecimento

As baterias que estão sendo utilizadas têm como tensão de saída nominal 22.2V. A tensão de saída da bateria será utilizada diretamente para alimentação dos drivers do sistema de atuação. Os sistemas de comando, comunicação e sensoramento no robô operam, no entanto, com alimentação de 5 V ou 3.3 V. Considerando que a queda de tensão de 22.2 V para 5 V ou ainda 3.3 V é relativamente alta, provocaria uma perda considerável de potência térmica dissipada por um simples regulador. Nesse caso, optou-se por utilizar um conversor CC-CC para executar tal tarefa.

O CI TPS54283 é um conversor CC-CC assíncrono de duas saídas. A configuração da tensão de saída é realizada através da composição de elementos externos como na Figura 64. Dessa forma, com um único CI é possível fornecer os sinais de 5 V e 3.3 V. Características gerais do conversor CC-CC selecionado estão na Tabela 31 (Apêndice B).

Figura 64: Projeto Eletrônico - Configuração do conversor CC-CC.

Observação 2 Para que a inicialização do sistema de alimentação não tenha atraso, os pinos $\overline{EN1}$ e $\overline{EN2}$ devem ser conectados em 0 V.

Monitoração

As baterias de polímero de lítio apresentam uma condição especial em que a carga de suas células não pode estar abaixo de 3V, sendo 3.7 V sua carga nominal. Considerando que a bateria utilizada nesse projeto é uma composição de 6 células, totalizando a tensão de saída nominal de 22.2 V, deve-se supervisionar a saída do conjunto para que o nível de saída da bateria não seja inferior a 18 V.

O CI TL7757 é um supervisor projetado para monitorar a tensão de alimentação para condições de subtensão. Se a tensão de alimentação cai abaixo do nível de tensão determinado, $V_{IT-} = 4.55$ V, a saída $\overline{RESET}$ (pino 1) torna-se ativa até que a condição de subtensão do pino de alimentação seja eliminada. A Tabela 32 (Apêndice B) exibe as características gerais do supervisor. A saída da bateria entra no pino V_{CC} (pino 2) do circuito supervisor através de um divisor de tensão como na Figura 65. O sinal $\overline{RESET}$ do TL7757 é conectado à porta PE0 (pino 32) no microcontrolador. O microcontrolador envia um sinal ao sistema de controle através do sistema de comunicação para alertar o usuário. Caso, a outra bateria esteja disponível e carregada quando o sinal de $\overline{RESET}$ for ativado, uma lógica implementada no microcontrolador acionará o relé da segunda bateria, desligando o relé da bateria que precisa ser recarregada.

Figura 65: Projeto Eletrônico - Supervisor de tensão.

Uma outra função desse sistema de monitoração da bateria é fornecer ao microcontrolador o conhecimento do nível real de saída da bateria a cada instante para que o valor do nível de tensão que controla os drivers dos motores seja gerado de forma correta, Seção 3.1.1. Um divisor resistivo passa ao microcontrolador essa informação através de portas de entrada analógica: Bateria 1 $\rightarrow$ ADC0 (pino 1 - J4) e Bateria 2 $\rightarrow$ ADC1 (pino 2 - J4).

3.1.4 Interface com o usuário

A interface com o usuário em hardware é composta por conjuntos de chaves. Um conjunto de seis chaves indica ao sistema de comando quais conjuntos do sistema de atuação e do sistema de sensoramento no robô estão ativados. Dessa forma o software embarcado é capaz de controlar o fluxo de dados de saída e de entrada especificamente. A Tabela 43 (Apêndice D) especifica a interface entre as chaves seletoras e o microcontrolador.

Uma chave liga/desliga com fusíveis é posicionada entre a bateria e o sistema de fornecimento e monitoração, servindo também como chave de emergência.

3.2 Implementação

A Figura 66 apresenta um diagrama geral do sistema sob o ponto de vista do projeto eletrônico. Nas Figuras 67 e 68 pode ser vista a disposição dos componentes e conectores, apresentando a melhor estrutura para o cabeamento entre os diversos sistemas (baterias, drivers, motores e encoders), além do roteamento das trilhas que foi ajustado considerando a diferença entre sinais de alta potência (22.2V) e os sinais de baixa potência. A Figuras 69 mostra as camadas superior e inferior da placa eletrônica implementada. A Figura 70 mostra em (a) as conexões dos cabos de bateria e drivers na base, e em (b) a conexão do motor e do encoder na junta. A Figura 71 mostra o robô montado com duas juntas, com toda a estrutura mecânica e eletrônica.

Figura 66: Sistema Eletrônico - Diagrama Geral.

Figura 67: Sistema Eletrônico - Layout: Alocação de componentes e roteamento - Camada superior.

Figura 68: Sistema Eletrônico - Layout: Alocação de componentes e roteamento - Camada inferior.

Figura 69: Sistema Eletrônico - Placa eletrônica: Vistas superior e inferior.

Figura 70: Sistema Eletrônico - Cabeamento: (a) Base; (b) Junta.

Figura 71: Sistema Eletrônico - Cabeamento: Placa eletrônica, baterias, drivers, encoders e motores.

4 *Sistema Computacional*

O sistema computacional envolvido no projeto da plataforma experimental é dividido em duas partes, uma residente no computador remoto e outra embarcada no módulo de comando do robô. O sistema computacional no computador remoto é um ambiente de aplicação desenvolvido em Matlab[®] cujo objetivo é integrar as fases de projeto, teste e validação experimental de algoritmos de controle desenvolvidos para manipuladores espaciais de base livre flutuante. Este sistema estabelece a interface com o usuário e será descrito neste capítulo. Por outro lado, o sistema computacional embarcado contempla a programação do microcontrolador para realizar o interfaceamento entre os equipamentos eletrônicos e controlar a comunicação de dados com o computador remoto, além de definir a estrutura programável de hardware no CPLD, e já foi detalhado na descrição do projeto eletrônico.

4.1 Ambiente de Simulação e Controle de Manipuladores Espaciais

O Ambiente de Simulação e Controle de Manipuladores Espaciais (ASCME) é uma plataforma de software proposta para simular o comportamento dinâmico de um manipulador espacial, assim como aplicar a um robô experimental as estratégias de controle sendo investigadas em tempo real.

Com base no conceito do Manipulador Dinamicamente Equivalente, [25], a ser tratado com mais detalhes no Capítulo 5, o aplicativo possibilita que o laço de controle seja executado levando em consideração sistemas baseados em manipuladores de base fixa dinamicamente equivalentes a manipuladores espaciais. Com esta técnica, o controlador é projetado com referência ao modelo de um manipulador de base fixa, já bem estabelecido, simplificando o processo. Neste software, os comportamentos dinâmicos do manipulador espacial e de seu MDE correspondente são avaliados simultaneamente.

Por outro lado, o conceito do MDE permite também que manipuladores experimentais de

base fixa, cuja junta da base possa ser configurada como passiva, sejam aplicados para avaliar o comportamento de manipuladores espaciais dinamicamente equivalentes.

Dessa forma o software de controle definido pelo ASCME está preparado não somente para atuar no robô experimental de base livre flutuante descrito nos capítulos anteriores (que será tratado daqui em diante neste texto por UARM-E), como também para atuar no UARM (Apêndice A).

A plataforma inclui uma interface gráfica através da qual o usuário pode ajustar os parâmetros do manipulador espacial e seu MDE correspondente, e observar a evolução simultânea do movimento durante a execução do controle. Possibilita a inclusão rápida de uma variedade de métodos de controle, desde que sejam implementados em arquivos .m. Alguns controladores já estão implementados em dois modos diferentes de formulação para o acompanhamento de trajetória: (1) Formulação no espaço das juntas (a posição de cada junta deve acompanhar uma trajetória); e (2) Formulação no espaço da tarefa (a posição inercial do efetuador é controlada). Janelas gráficas apresentam a evolução no tempo das posições das juntas, da trajetória do efetuador, dos erros de acompanhamento, velocidades e dos torques aplicados.

As seções a seguir descrevem as funcionalidades da interface gráfica, as possibilidades de configuração do sistema e a implementação do laço de controle.

4.1.1 Interface gráfica

A interface gráfica apresenta a possibilidade de comparação visual entre o comportamento do manipulador espacial e o comportamento do manipulador de base fixa dinamicamente equivalente. Algumas funcionalidades permitem ao usuário configurar o sistema e entender melhor os resultados da simulação ou do ensaio experimental sendo realizados. Quando executado de dentro do Matlab[®], o ambiente de desenvolvimento de sistemas de controle aparece como na Figura 72. Com o intuito de ampliar o público usuário e internacionalizar o sistema, os textos da interface gráfica foram escritos em inglês.

A interface é dividida em três áreas distintas: o quadro de comandos e dados, o quadro do espaço de trabalho e o quadro de resultados.

O quadro de comandos e dados está posicionado na parte superior da interface e é composto por cinco seções que determinam os comandos e configurações do procedimento, além de apresentar os dados do robô espacial e da tarefa a ser executada.

A seção identificada por **ROBOT** e ilustrada na Figura 73 possibilita ao usuário selecio-

Figura 72: ASCME - Ambiente de Simulação e Controle de Manipuladores Espaciais.

nar o tipo de sistema que ele deseja trabalhar. No caso pode-se selecionar entre o UARM (o manipulador de base fixa configurado para atuar dinamicamente equivalente a um manipulador espacial) e o UARM-E (o manipulador de base livre desenvolvido neste trabalho).

Figura 73: ASCME - Seleção do robô: UARM ou UARM-E.

Selecionado o sistema, o próximo passo é determinar o tipo de procedimento a ser executado: Simulação ou Experimental, Figura 74. Como os botões de comando e configuração, assim como os dados a serem apresentados são dependentes dessas escolhas, somente após tendo sido definido o sistema e o procedimento, as outras seções do quadro de comandos e dados são apresentadas.

O conjunto de botões apresentado na Figura 75, **UARM-E COMMANDS**, estabelece comandos ao robô UARM-E. Com exceção do botão de configuração do robô, *Robot Configuration*, os botões estão ativos somente quando o procedimento experimental é selecionado. Suas ações são como segue:

Figura 74: ASCME - Seleção do procedimento: Simulação (*Simulation*) ou Experimento (*Experiment*).

Figura 75: ASCME - Comandos do robô.

- *Robot Configuration*: abre uma janela para configuração dos parâmetros do sistema robótico experimental ou que se deseja simular. Detalhes de configuração são descritos na Seção 4.1.2.
- *Start UARM-E*: abre uma janela de mensagem com a lista de procedimentos a serem realizados pelo usuário para ligar o robô.
- *Calibration*: determina a posição de referência nula para o robô. Inicialmente, anula os torques nos motores e libera os freios. Uma janela de mensagem pede ao usuário que posicione o robô em uma certa posição de referência (conhecida pelo procedimento de calibração). Quando o botão *OK* é pressionado pelo usuário, a plataforma inicia um procedimento para posicionar o robô na posição $[0\ 0\ \dots\ 0]$ como na Figura 76, os freios são travados e todos os encoders são, então, zerados. Assim, os encoders estão calibrados.
- *Read Position*: lê a posição atual do robô e exibe em uma janela de mensagem.
- *Brakes ON*: trava os freios.
- *Brakes OFF*: libera os freios.

Um quadro semelhante é apresentado quando o robô UARM é selecionado. Porém é interessante observar que as funções executadas pelos botões, embora tenham as mesmas finalidades,

Figura 76: ASCME - Posição $[0\ 0\ \dots\ 0]$.

executam programas diferentes, relacionados à interface com o UARM.

Os botões da seção de comando do usuário, **USER COMMANDS**, executam o controle do procedimento de controle, além de possibilitar a configuração da tarefa a ser executada. A Figura 77 ilustra o conjunto de botões apresentado nos casos de simulação e procedimento experimental.

Figura 77: ASCME - Comandos do usuário: (a) Simulação, (b) Experimento.

- *Task Configuration*: abre uma janela para configuração da tarefa e do procedimento de controle. Detalhes de configuração são descritos na Seção 4.1.2.
- *Start Simulation/Experiment*: inicia a execução do laço de controle em simulação ou em ensaio experimental.
- *Stop Simulation/Experiment*: interrompe a execução do laço de controle em simulação ou em ensaio experimental.
- *Reset Position*: posiciona o manipulador na posição Posição $[0\ 0\ \dots\ 0]$, Figura 76.
- *Initial Position*: posiciona o manipulador na posição inicial definida nas configurações da tarefa.

- *Save Results*: grava as variáveis tempo, posições controladas, velocidades controladas, posições controladas desejadas, torque aplicado e resultados numéricos relacionados à última execução, ao arquivo *SimuData.mat* ou *ExperData.mat*.

As seções de apresentação de dados são ilustradas na Figura 78 a seguir. O quadro **SM PARAMETERS** exibe os parâmetros físicos configurados para o robô espacial, ou seja, massa, momento de inércia, comprimento dos elos, posição do centro de massa, além do vetor que determina as incertezas nesses parâmetros. O quadro **TASK PARAMETERS** por sua vez, apresenta as características do procedimento a ser executado tais como: posição inicial do robô, posição final desejada, tempo desejado para execução da tarefa, controlador selecionado, tempo de amostragem, e características da perturbação como tempo de início da perturbação, tipo de perturbação e quantidade de energia inserida pelo distúrbio.

Figura 78: ASCME - Parâmetros do procedimento: (a) Robô, (b) Tarefa.

O quadro do espaço de trabalho fornece a ilustração do manipulador espacial, em preto e azul, definido pelos parâmetros de configuração do robô, sobreposta pela ilustração de seu MDE correspondente, em rosa e verde¹. A evolução do movimento dos robôs é apresentada *online* e em tempo real no caso de ensaios experimentais. A Figura 79 mostra o posicionamento final do UARM e do manipulador espacial equivalente no espaço de trabalho.

Figura 79: ASCME - Espaço de trabalho.

¹O asterisco (*) indica a posição do centro de massa do MDE.

Ao término ou após uma interrupção do procedimento de controle, os principais resultados são mostrados no quadro de resultados. Resultados numéricos indicam o erro de estado estacionário para cada junta controlada (controle no espaço das juntas) ou posição do efetuador (controle no espaço da tarefa) e o tempo que cada junta (ou posição inercial do efetuador) gastou para alcançar o *set-point*, dentro de um limite de erro pré-definido. Valores de índices de desempenho que caracterizam o procedimento pela quantidade de erro gerado ao longo da execução da tarefa de acompanhamento de trajetória e pela quantidade de energia despendida pelo controlador também são apresentados, Figura 80. Além disso, quando operando no modo de controle no espaço das juntas, o quadro de resultados, fornece gráficos apresentado a evolução da posição da base e das juntas ao longo do tempo, a velocidade da base e das juntas ao longo do tempo e os torques aplicados $\tau(t)$, Figura 81. Na formulação em espaço da tarefa, os gráficos gerados ilustram a evolução das posições do efetuador ao longo do tempo $x_{ef}(t)$ e $y_{ef}(t)$, a trajetória do efetuador no espaço inercial $y_{ef}(x_{ef})$ e os torques aplicados $\tau(t)$.

RESULTS			
Steady-state error:	-0.05	0.04	L2 error norm: 0.000
Convergence time (s):	2.77	2.85	Energy spent (Nm): 0.032

Figura 80: ASCME - Resultados numéricos.

Figura 81: ASCME - Resultados gráficos.

4.1.2 Configurações

UARM

Quando o robô UARM é selecionado, a janela de configurações do robô, aberta ao clicar o botão *Robot Configuration*, é como na Figura 82. Duas tabelas, uma referente ao ME (**SM Parameters**) e outra referente ao MDE (**DEM Parameters**), possibilitam a edição dos valores dos parâmetros cinemáticos e dinâmicos que caracterizam o robô. No caso do UARM em especial, como sua estrutura é fixa, só é possível trabalhar com manipuladores de três elos, não havendo a possibilidade de modificar essa característica do sistema. A tabela referente aos parâmetros do MDE deve ser preenchida com os valores dos parâmetros do UARM, em especial no caso de procedimento experimental, enquanto os campos da tabela referente aos parâmetros do ME são preenchidos de acordo com a equivalência demonstrada no Capítulo 5. O botão *Calculate SM from DEM* realiza o cálculo dos parâmetros que caracterizam o manipulador espacial equivalente. Neste caso, o usuário também deve fornecer o valor da massa total do manipulador espacial solicitado. O botão *Calculate DEM from SM*, por outro lado, realiza a operação inversa, fornecendo os parâmetros do MDE para um dado manipulador espacial. É interessante observar que, embora somente duas casa decimais estejam visíveis nos valores apresentados nas tabelas, valores mais precisos podem ser inseridos e serão considerados nos cálculos. O botão *Default Values* cancela todas as modificações e retorna os valores dos parâmetros às características do UARM e de seu ME correspondente. O botão *Cancel* ignora as modificações executadas na configuração atual, enquanto ao clicar no botão *OK* o sistema acata as configurações executadas pelo usuário e fecha a janela. A ilustração dos robôs no quadro do espaço de trabalho é atualizada de acordo com as novas configurações, assim como os valores apresentados na seção **SM PARAMETERS**.

The screenshot shows the 'UARM Configuration' window. At the top, there's a title bar with the window name and standard OS controls. Below the title bar, there's a section for 'Number of joints:' with a dropdown menu set to '3'. The main area contains two tables: 'SM Parameters' and 'DEM Parameters'. Each table has columns for joint parameters (J1, J2, J3) and rows for Mass(kg), Inertia(kgm^2), Length(m), and Center of mass(m). Below the tables are two buttons: 'Calculate DEM from SM' and 'Calculate SM from DEM'. At the bottom of the window are three buttons: 'Default Values', 'Cancel', and 'OK'.

SM Parameters				DEM Parameters			
	J1	J2	J3		J1	J2	J3
Mass(kg)	4.82	0.62	0.57	Mass(kg)	1.93	0.85	0.63
Inertia(kgm ²)	0.02	0.01	0.01	Inertia(kgm ²)	0.02	0.01	0.01
Length(m)	0.25	0.24	0.21	Length(m)	0.20	0.20	0.20
Center of mass(m)	0.00	0.12	0.09	Center of mass(m)	0.00	0.10	0.08

Figura 82: ASCME - Configuração do UARM.

UARM-E

Já no caso de ter-se selecionado o sistema UARM-E a janela de configuração dos parâmetros do robô é apresentada como na Figura 83. A estrutura do UARM-E possibilita sua montagem com número variável de juntas, e portanto essa é uma característica configurável nesse caso. Assim, em uma lista de 1 a 6, o usuário pode selecionar o número de juntas do robô de base livre flutuante com que ele deseja trabalhar. O formato das tabelas de parâmetros, **SM PARAMETERS** e **DEM PARAMETERS**, é modificado de acordo com o número de juntas definido para que os parâmetros de todos os elos sejam configurados (verificar Figura 83(a) e (b)).

Figura 83: ASCME - Configuração do UARM-E: (a) 2 juntas, (b) 5 juntas.

Nesta janela nota-se o botão *Hardware Configuration*. Sua execução abre uma segunda janela de configurações como as mostradas na Figura 84 (a) e (b) para os casos do robô com duas e cinco juntas. Nesta janela é possível identificar para o software de controle qual motor conectado na placa de comando está relacionado a cada junta. Uma vez que a montagem do robô pode ser modificada, esses dados são imprescindíveis para que o software de controle envie os dados ao módulo de comando na ordem correta. Como teste de segurança, os dados dessa tabela são confrontados com os valores das chaves selecionadas no sistema eletrônico. Caso o teste falhe, uma mensagem de erro pede ao usuário que verifique as configurações de hardware.

Figura 84: ASCME - Configuração de hardware do UARM-E: (a) 2 juntas, (b) 5 juntas.

O funcionamento dos outros botões é análogo ao que foi descrito no caso anterior. Da mesma forma, a ilustração dos robôs no quadro do espaço de trabalho é atualizada de acordo com as novas configurações. A Figura 85 ilustra as possíveis configurações do UARM-E e o MDE correspondente.

Figura 85: ASCME - Configuração do UARM-E: Número de juntas.

Tarefa

A janela de configuração da tarefa, *Task Configuration*, permite a caracterização do procedimento de controle. É dividida em quatro seções. A Figura 86 ilustra a janela fornecida no caso de um robô configurado com duas juntas em (a), e para o caso de um robô configurado com cinco juntas em (b).

Na primeira seção o usuário pode selecionar o tipo de controle que será executado: Espaço das juntas (*Joint space*) ou Espaço da tarefa (*Task space*), Figura 87.

Ao selecionar o controle no espaço da tarefa, abre-se a possibilidade de escolher o tipo de trajetória a ser executada pelo efetuator. A princípio estão disponíveis uma trajetória retilínea e uma trajetória de meio-círculo, Figura 88. Novas trajetórias podem ser facilmente implementadas no arquivo *TaskTraj.m*, bastando observar que a trajetória proposta deve estar inserida em um espaço de trabalho livre de singularidades (PIW) para trabalhar com os controladores já implementados. No entanto, se for da intenção do usuário utilizar a plataforma para projetar e testar um sistema de controle que trate esse tipo de problema, então a trajetória pode ser gerada livremente.

Figura 86: ASCME - Configuração da tarefa: (a) 2 juntas, (b) 5 juntas.

Figura 87: ASCME - Configuração da tarefa: Seleção do espaço de controle.

Figura 88: ASCME - Configuração da tarefa: Seleção da trajetória para controle no espaço da tarefa.

O segundo campo nesta seção lista os controladores disponíveis, Figura 89. Um novo controlador pode ser inserido com a criação de um arquivo `cont_NOVOCONTROLADOR.m` na pasta do software de controle.

A segunda seção na janela de configurações da tarefa diz respeito aos parâmetros de posição. São definidos nesse espaço os valores de posição inicial e posição final desejada, para cada

Figura 89: ASCME - Configuração da tarefa: Seleção do controlador.

tipo de procedimento de controle selecionado, das juntas ou do efetuador. Na terceira seção são configurados os parâmetros de tempo da tarefa: tempo desejado para execução da tarefa e tempo de amostragem para a simulação. No caso de procedimentos experimentais, a variação de tempo é medida a cada iteração do laço de controle (comandos `tic` e `toc`) para que o sistema atue em tempo real. Nesse caso o tempo de amostragem definido na configuração da tarefa é utilizado como valor padrão quando a medida apresenta ruídos. A última seção da janela *Task Configuration* apresenta a possibilidade de inserção de incertezas no modelo e distúrbios de torque na malha do controlador. O sistema já oferece a possibilidade da seleção de um distúrbio no formato de senóides amortecidas cujos parâmetros podem ser ajustados nessa janela de configurações, reveja a Figura 86.

4.1.3 Laço de controle

Quatro situações podem ser consideradas para definir a estrutura do laço de controle:

1. Simulação no UARM,
2. Experimento no UARM,
3. Simulação no UARM-E,
4. Experimento no UARM-E.

Nos casos 1 e 2, o laço de controle é definido pelo diagrama de blocos na Figura 90. Sejam as variáveis controladas do robô definidas pela velocidade e pela posição do robô em um espaço de trabalho determinado (juntas ou tarefa). Em estratégias dependentes do modelo, o controlador considera o modelo nominal (MDE_0) do manipulador de base fixa dinamicamente equivalente ao manipulador espacial em análise. A saída do controlador é um sinal de torque τ . Este sinal pode ser perturbado por um distúrbio externo τ_d , e torna-se a entrada das plantas do MDE e do ME. Quando um ensaio experimental está sendo executado, o sinal $\tau + \tau_d$ é direcionado ao UARM ao invés de seguir ao modelo do MDE. A saída do ME e do MDE é um vetor de

posição das juntas, uma vez que cada junta do UARM possui um encoder. Este sinal de saída também é enviado à interface gráfica do ASCME para atualizar a ilustração dos manipuladores. Velocidades das juntas são obtidas através de diferenciação numérica e filtragem, $L(s)$. Os modelos são atualizados com os novos dados de posição e velocidade e uma nova iteração se inicia até que a trajetória seja completada e o *set-point* alcançado. Na Figura 91 são ilustrados seis passos do procedimento de acompanhamento de trajetória no espaço das juntas em uma simulação do UARM.

Figura 90: ASCME - Laço de controle: Experimento no UARM.

Figura 91: ASCME - Laço de controle: Simulação de controle do UARM no espaço das juntas.

Nos casos 3 e 4 a situação é diferente pois o sistema real no laço é o manipulador de base livre. Observe no diagrama de blocos da Figura 92 que a realimentação do laço de controle é

- todas as juntas ou posições do efetuador alcançam seus *set-points*: a simulação/experimento continuará executando por mais $0.1 * (\text{tempo de tarefa desejado})$ segundos e é interrompida;
- a simulação/experimento dura mais do que um máximo tempo pré-definido: a simulação/experimento é interrompida quando o tempo máximo é atingido;
- o usuário pressiona o botão *Stop Simulation/Experiment*: a simulação/experimento para imediatamente;
- um erro numérico ocorre: a simulação/experimento para imediatamente.

Quando a simulação/experimento para, os torques são zerados e os freios acionados. Nos três primeiros casos, assim que a simulação/experimento termina, o botão *Start Experiment* torna-se ativo novamente. Os laços de controle implementados no ASCME podem ser generalizados no seguinte algoritmo:

Algoritmo 1: ASCME - Laço de controle.

```

Captura posição inicial; for  $t = 0, \dots, \text{max\_tempo}$  do
  Captura tempo;
  Gera próximo set-point de acordo com a trajetória a ser seguida;
  Calcula erros de posição;
  Calcula erros de velocidade;
  if  $\text{erros} < \text{max\_erro}$  then
    break
  Controlador;
  if distúrbio then
    break
  if  $\text{torque} > \text{max\_torque}$  then
    break
  Aplicar torque no robô;
  Capturar posição atual das juntas do robô em análise;
  Calcular e filtrar velocidade das juntas;
  Calcular e filtrar aceleração das juntas;
  Calcular posição do efetuador;
  Atualizar modelo do robô em análise;
  Calcular posição do modelo equivalente;
  Calcular e filtrar velocidade;
  Calcular e filtrar aceleração;
  Atualizar modelo do sistema equivalente;
  if Stop Simulation/Experiment then
    break
  Atualizar ilustração no espaço de trabalho;
Gera resultados;
  
```

Parte II

Controle $\mathcal{H}_\infty$ Não Linear de Manipuladores Espaciais de Base Flutuante

A segunda parte deste texto apresenta a investigação de diferentes soluções para o problema de controle robusto de um manipulador de base livre flutuante no espaço da tarefa.

A fim de se evitar as complicações envolvidas em um mapeamento cinemático, neste trabalho, a formulação do problema foi desenvolvida diretamente no espaço da tarefa. Assim, as posições do efetuador são controladas diretamente. O modelo do manipulador espacial é descrito através do conceito do Manipulador Dinamicamente Equivalente (MDE). Propõe-se realizar o tratamento dos efeitos de incertezas paramétricas no modelo, de distúrbios e de dinâmicas não-modeladas com base no critério $\mathcal{H}_\infty$. Uma primeira abordagem é proposta considerando uma estrutura bem definida para a planta, sendo as incertezas paramétricas representadas como distúrbios. A segunda abordagem é adaptativa e define dois métodos de controle baseados em redes neurais. O primeiro método aplica o sistema inteligente para aprender o comportamento dinâmico do sistema robótico, considerado completamente desconhecido. O segundo método inteligente considera que o modelo nominal do sistema é bem definido e as redes neurais são aplicadas para estimar somente o comportamento dinâmico resultante de incertezas paramétricas e da base espacial, cuja dinâmica é considerada não-modelada. O critério $\mathcal{H}_\infty$ é aplicado para atenuar o efeito dos erros de estimativa e distúrbios externos.

Uma tarefa de acompanhamento de trajetória do efetuador é definida para o manipulador espacial em simulações e ensaios experimentais. Um manipulador planar de três elos com base fixa, cuja primeira junta é configurada passiva é empregado como referência dinamicamente equivalente a um manipulador espacial de base livre flutuante com dois elos. Incertezas no modelo e distúrbios de torque são incluídos no sistema. Análises gráficas e numéricas dos resultados são fornecidas para validar e comparar as estratégias propostas.

A sequência do texto apresenta a formulação do problema no espaço da tarefa e as soluções dos problemas de controle $\mathcal{H}_\infty$ baseados em modelo e em redes neurais.

5 Manipulador Dinamicamente Equivalente

Um manipulador rígido com n elos montado sobre uma base livre flutuante, sem atuação de forças externas ou torques, pode ser modelado pelo conceito do Manipulador Dinamicamente Equivalente (MDE) proposto em [25]. O MDE é um manipulador de base fixa com $n + 1$ elos, sendo a primeira junta esférica e passiva e, cujo modelo mapeia, sob ação de qualquer lei de controle, a dinâmica de um Manipulador Espacial (ME) de forma idêntica, possibilitando um entendimento completo do comportamento de um manipulador espacial e sua base através de experimentos com um robô de base fixa convencional.

Figura 94: Modelagem - ME e seu MDE correspondente.

A Figura 94 mostra a representação e a notação de parâmetros para o ME e para o MDE. Sendo os parâmetros do ME identificados por ', os elos do manipulador são numerados de 2 a $n + 1$ e J_i é a junta que conecta os elos $(i - 1)$ e i , θ_i é a rotação do elo i em torno da junta J_i , e os ângulos de Euler Z-Y-Z (φ, θ, ψ) representam os ângulos da primeira junta, passiva, do MDE e a orientação da base do ME. A convenção Z-Y-Z dos ângulos de Euler é utilizada nesse trabalho como em [25]. Essa convenção representa uma rotação φ do sistema XYZ em torno do eixo Z, uma rotação θ do sistema X'Y'Z' em torno do eixo Y', e uma rotação ψ do sistema X''Y''Z'' em torno do eixo Z''.

Sejam C_i o centro de massa do elo i , L_i o vetor que conecta J'_i e C'_i , R_i o vetor que conecta C'_i e J'_{i+1} , l_{ci} o vetor que conecta J_i e C_i , e W_i o vetor que conecta J_i e J_{i+1} . Considerando que o MDE atua na ausência de gravidade e que sua base está localizada no centro de massa do ME, os parâmetros cinemáticos e dinâmicos do MDE podem ser calculados a partir dos parâmetros do ME da forma:

$$\begin{aligned}
 m_i &= \frac{M_t^2 m'_i}{\sum_{k=1}^{i-1} m'_k \sum_{k=1}^i m'_k}, \\
 I_1 &= I'_1, \\
 I_i &= I'_i, \\
 W_1 &= \frac{R_1 m'_1}{M_t}, \\
 W_i &= \frac{R_i}{M_t} \sum_{k=1}^i m'_k + \frac{L_i}{M_t} \sum_{k=1}^{i-1} m'_k, \\
 l_{c1} &= 0, \\
 l_{ci} &= \frac{L_i}{M_t} \sum_{k=1}^{i-1} m'_k,
 \end{aligned} \tag{5.1}$$

sendo $i = 2, \dots, n+1$ e M_t a massa total do ME. Observe que a massa da base (junta passiva), m_1 , não é definida pelas propriedades de equivalência, podendo assumir qualquer valor positivo não nulo.

Sejam as coordenadas generalizadas $q = [\varphi \quad \theta \quad \psi \quad \theta_2 \quad \dots \quad \theta_{n+1}]^T \in \mathbb{R}^{(n+3)}$ particionadas como $q = [q_b^T \quad q_m^T]^T$, sendo b e m os índices representando a junta esférica passiva (base) e as juntas ativas (manipulador), respectivamente. Admite-se que o sistema não apresenta componentes elásticas e que a aceleração gravitacional no espaço é nula. Da teoria de Lagrange, as equações dinâmicas do MDE são dadas por um conjunto de equações diferenciais não lineares acopladas como

$$M(q_m)\ddot{q} + C(q_m, \dot{q})\dot{q} = \tau, \tag{5.2}$$

sendo $M(q_m) \in \mathbb{R}^{(n+3) \times (n+3)}$ a matriz de inércia simétrica definida positiva dependente somente das coordenadas do manipulador, $C(q_m, \dot{q}) \in \mathbb{R}^{(n+3) \times (n+3)}$ a matriz de Coriolis e de forças centrífugas, e $\tau = [0 \quad 0 \quad 0 \quad \tau_2 \quad \dots \quad \tau_{n+1}]^T \in \mathbb{R}^{(n+3)}$ o vetor de torques atuando sobre as juntas do MDE.

Do ponto de vista prático, o ME precisa manipular diferentes cargas cujas características de massa e inércia não são bem definidas, atritos nas juntas, amortecimentos, vibrações e outras incertezas não lineares não são fáceis de modelar ou medir e produzem efeitos que não devem ser negligenciados. Além disso, o ambiente de trabalho hostil e o movimento da base

flutuante são fontes de distúrbios que também podem gerar erros de trajetória no ME. Nesse sentido, define-se τ_d como um distúrbio de torque de energia finita e incertezas paramétricas são introduzidas no modelo dividindo-se as matrizes paramétricas $M(q_m)$ e $C(q_m, \dot{q})$ em uma parte nominal e uma parte incerta como

$$\begin{aligned} M(q_m) &= M_0(q_m) + \Delta M(q_m), \\ C(q_m, \dot{q}) &= C_0(q_m, \dot{q}) + \Delta C(q_m, \dot{q}), \end{aligned}$$

sendo $M_0(q_m)$ e $C_0(q_m, \dot{q})$ matrizes nominais e $\Delta M(q_m)$ e $\Delta C(q_m, \dot{q})$ incertezas paramétricas do modelo.

5.1 UARM

O manipulador UARM (Apêndice A) pode ser utilizado como manipulador dinamicamente equivalente de um ME de 2 elos. Aplicando a configuração PAA, o elo 1 (junta passiva) se relaciona à base espacial no ME e os elos 2 e 3 correspondem ao manipulador espacial. Alguns ajustes estruturais, descritos em [60], são necessários para assegurar a condição $l_{c1} = 0$ em (5.1). Assim, os parâmetros físicos modificados do UARM são apresentados na Tabela 10. A Tabela 11 apresenta os respectivos parâmetros do manipulador espacial calculados utilizando as relações em (5.1).

Tabela 10: Modelagem - UARM: MDE.

	m_i	I_i	W_i	l_{ci}
	(kg)	(kgm ²)	(m)	(m)
Junta 1	1.932	0.0153	0.203	0
Junta 2	0.850	0.0075	0.203	0.096
Junta 3	0.625	0.0060	0.203	0.077

Tabela 11: Modelagem - UARM: ME.

	m_i'	I_i'	R_i	L_i
	(kg)	(kgm ²)	(m)	(m)
Base	4.816	0.0153	0.253	0
Elo 2	0.618	0.0075	0.118	0.120
Elo 3	0.566	0.0060	0.126	0.085

5.2 UARM-E

Por outro lado, dado um ME, o conceito do MDE também pode ser aplicado com o objetivo de simplificar o projeto de controladores para manipuladores espaciais. O processo de modelagem de manipuladores espaciais com base livre flutuante é complexo, pois deve considerar o acoplamento dinâmico, singularidades dinâmicas e restrições não-holonômicas inerentes a esse tipo de sistema. Dessa forma, o MDE pode ser utilizado como uma ferramenta para o desenvolvimento do modelo dinâmico do ME, utilizando formulações simples e bem estabelecidas na literatura para a modelagem de manipuladores de base fixa, [25].

Portanto, com base nas relações definidas em (5.1), determina-se um MDE para o manipulador planar de base livre flutuante descrito na Parte I dessa tese. Na Tabela 12 são identificados os parâmetros físicos do robô experimental com um braço e duas juntas ativas (obtidos na Seção 2.3), enquanto na Tabela 13 estão relacionados os parâmetros do respectivo MDE.

Tabela 12: Modelagem - UARM-E: ME.

Robô 1				
	m_i	I_i	L_i	R_i
	(kg)	(kgm ²)	(m)	(m)
Base	4.780	0.0404	0	0.150
Elo 2	1.380	0.0182	0.144	0.111
Elo 3	1.011	0.0115	0.103	0.112
Robô 2				
	m_i	I_i	L_i	R_i
	(kg)	(kgm ²)	(m)	(m)
Base	5.616	0.0503	0	0.155
Elo 2	1.516	0.0194	0.152	0.103
Elo 3	1.144	0.0130	0.116	0.099

Tabela 13: Modelagem - UARM-E: MDE.

Robô 1				
	m_i	I_i	W_i	lc_i
	(kg)	(kgm ²)	(m)	(m)
Junta 1	4.780	0.0404	0.100	0
Junta 2	2.410	0.0182	0.191	0.096
Junta 3	1.177	0.0115	0.201	0.089
Robô 2				
	m_i	I_i	W_i	lc_i
	(kg)	(kgm ²)	(m)	(m)
Junta 1	5.616	0.0503	0.105	0
Junta 2	2.592	0.0194	0.192	0.103
Junta 3	1.328	0.0130	0.199	0.100

6 *Controle $\mathcal{H}_\infty$ Não Linear no Espaço da Tarefa*

6.1 Formulação do Problema

Por se tratar de um manipulador espacial com base livre flutuante, considere que apenas as juntas ativas do MDE são controladas, e que a junta esférica passiva é livre. Neste caso, a dinâmica da junta passiva interfere no controle das juntas ativas do manipulador.

A posição e a orientação inercial do efetuador de um ME,

$$p = [\varphi_{ef} \quad \theta_{ef} \quad \psi_{ef} \quad x_{ef} \quad y_{ef} \quad z_{ef}],$$

é função da posição e orientação de sua base espacial e das coordenadas generalizadas das juntas do manipulador. Segundo [17], a dependência das coordenadas do efetuador com a posição da base pode ser eliminada escrevendo a cinemática do sistema em relação ao centro de massa do sistema e integrando sua equação de momento linear. A modelagem baseada no Manipulador Virtual em [24] segue essa técnica, uma vez que a origem do eixo inercial, referência nesta abordagem, está localizada no centro de massa do sistema espacial. A dependência com a orientação da base, q_b , não pode ser eliminada pois o momento angular do sistema não pode ser integrado analiticamente de forma a fornecer a orientação da base como função das variáveis das juntas do manipulador, q_m . Portanto, $p = f(q_b, q_m)$.

Seja $J(q)$ o Jacobiano que relaciona as velocidades das coordenadas das juntas, $\dot{q}$, e as velocidades do efetuador do MDE, $\dot{p}$, como

$$\dot{p} = J(q)\dot{q}. \quad (6.1)$$

Admita que $J(q) = J_0(q) + \Delta J(q)$, sendo $J_0(q)$ o Jacobiano nominal e $\Delta J(q)$ a parcela resultante de incertezas paramétricas.

Considerando que $J(q)$ é não-singular para qualquer q pertencente à trajetória definida para

as juntas do manipulador, aplicando (6.1) e sua respectiva derivada, $\ddot{p} = \dot{J}(q)\dot{q} + J(q)\ddot{q}$, em (5.2), as equações dinâmicas para o MDE no espaço da tarefa são dadas por

$$\tau = M_{ef}(q)\ddot{p} + C_{ef}(q, \dot{q})\dot{p}, \quad (6.2)$$

com

$$\begin{aligned} M_{ef}(q) &= M(q_m)J^{-1}(q), \\ C_{ef}(q, \dot{q}) &= (C(q_m, \dot{q}) - M(q_m)J^{-1}(q)\dot{J}(q, \dot{q}))J^{-1}(q). \end{aligned}$$

Nesta estratégia de controle de movimento, os valores de orientação da base espacial, q_b , estão presentes nas matrizes da equação dinâmica do sistema, M_{ef} e C_{ef} , pois foram inseridos pelo Jacobiano $J(q)$. Isso não acontece quando o problema é formulado no espaço das juntas, [38]. É preciso atentar também para o fato de que $M_{ef}(q)$ não é simétrica definida positiva e que a matriz $N_{ef}(q, \dot{q}) = \dot{M}_{ef}(q, \dot{q}) - 2C_{ef}(q, \dot{q})$ não é anti-simétrica. Para que a equação dinâmica do sistema no espaço inercial preserve as características da dinâmica formulada no espaço das juntas, uma transformação de forças é aplicada a (6.2) como em [68] de forma que

$$\tau = J^T(q)F, \quad (6.3)$$

sendo F o vetor de forças generalizadas do efetuador no espaço inercial. Portanto,

$$F = \bar{M}_{ef}(q)\ddot{p} + \bar{C}_{ef}(q, \dot{q})\dot{p}, \quad (6.4)$$

com

$$\begin{aligned} \bar{M}_{ef}(q) &= J^{-T}(q)M_{ef}(q), \\ \bar{C}_{ef}(q, \dot{q}) &= J^{-T}(q)C_{ef}(q, \dot{q}). \end{aligned}$$

Nesse formato a equação dinâmica do sistema formulada no espaço inercial mantém a estrutura e as propriedades encontradas no espaço das juntas. A matriz $\bar{M}_{ef}(q)$ é simétrica definida positiva e a matriz $\bar{N}_{ef}(q, \dot{q}) = \dot{\bar{M}}_{ef}(q, \dot{q}) - 2\bar{C}_{ef}(q, \dot{q})$ é anti-simétrica.

Uma característica do controle de manipuladores subatuados, que na formulação em espaço das juntas foi absorvida de forma natural, deve, agora, ser explicitada para o problema no espaço inercial: em um manipulador com n_a atuadores, somente n_a graus de liberdade podem ser controlados por vez, veja [69]. O MDE apresenta n juntas ativas, logo $n_a = n$. Assim sendo, considere $p = [p_u^T \ p_a^T]^T$ o vetor de coordenadas generalizadas do sistema, com $p_u \in \mathbb{R}^{(6-n) \times 1}$ e $p_a \in \mathbb{R}^{n \times 1}$, sendo os índices u e a referentes às variáveis de p que estarão livres (passivas) durante o procedimento de controle e às variáveis de p a serem controladas, respectivamente.

A equação (6.4) pode ser particionada como

$$\begin{bmatrix} F_u \\ F_a \end{bmatrix} + \delta + F_d = \begin{bmatrix} \bar{M}_{efuu}(q) & \bar{M}_{efua}(q) \\ \bar{M}_{efau}(q) & \bar{M}_{efaa}(q) \end{bmatrix} \begin{bmatrix} \ddot{p}_u \\ \ddot{p}_a \end{bmatrix} + \begin{bmatrix} \bar{C}_{efuu}(q, \dot{q}) & \bar{C}_{efua}(q, \dot{q}) \\ \bar{C}_{efau}(q, \dot{q}) & \bar{C}_{efaa}(q, \dot{q}) \end{bmatrix} \begin{bmatrix} \dot{p}_u \\ \dot{p}_a \end{bmatrix}, \quad (6.5)$$

sendo $\delta = [\delta_u^T \ \delta_a^T]^T$ definido como um vetor representando a soma das incertezas paramétricas do sistema tal que

$$\begin{bmatrix} \delta_u(q, \dot{q}, \dot{p}, \ddot{p}) \\ \delta_a(q, \dot{q}, \dot{p}, \ddot{p}) \end{bmatrix} = - \begin{bmatrix} \Delta \bar{M}_{efuu}(q) \ddot{p}_u + \Delta \bar{M}_{efua}(q) \ddot{p}_a + \Delta \bar{C}_{efuu}(q, \dot{q}) \dot{p}_u + \Delta \bar{C}_{efua}(q, \dot{q}) \dot{p}_a \\ \Delta \bar{M}_{efau}(q) \ddot{p}_u + \Delta \bar{M}_{efaa}(q) \ddot{p}_a + \Delta \bar{C}_{efau}(q, \dot{q}) \dot{p}_u + \Delta \bar{C}_{efaa}(q, \dot{q}) \dot{p}_a \end{bmatrix},$$

F_d um distúrbio externo de energia finita, $\bar{M}_{efuu}, \bar{C}_{efuu} \in \mathbb{R}^{(6-n) \times (6-n)}$, $\bar{M}_{efua}, \bar{C}_{efua} \in \mathbb{R}^{(6-n) \times n}$, $\bar{M}_{efau}, \bar{C}_{efau} \in \mathbb{R}^{n \times (6-n)}$, $\bar{M}_{efaa}, \bar{C}_{efaa} \in \mathbb{R}^{n \times n}$, $F_u \in \mathbb{R}^{(6-n) \times 1}$ e $F_a \in \mathbb{R}^{n \times 1}$. Por simplicidade de notação, o índice 0 referente ao sistema nominal foi suprimido e também o argumento das funções será omitido (ex.: $M(\cdot)$, $C(\cdot, \cdot)$, etc) daqui por diante. Essa composição deve também preservar as propriedades da equação dinâmica para as matrizes $\bar{M}_{efaa}$ e $\bar{C}_{efaa}$ tal que $\bar{M}_{efaa} = \bar{M}_{efaa}^T > 0$ e $\bar{N}_{efaa} = \dot{\bar{M}}_{efaa} - 2\bar{C}_{efaa}$ é anti-simétrica.

Observação 3 O controlador é projetado para o ME considerando o modelo do MDE.

Sejam $p_a^d, \dot{p}_a^d, \ddot{p}_a^d \in \mathbb{R}^n$ a trajetória de referência desejada e as correspondentes velocidade e aceleração para as variáveis controladas do efetuador, respectivamente. Admite-se que as variáveis $p_a^d, \dot{p}_a^d$ e $\ddot{p}_a^d$ fazem parte da região do espaço de trabalho do ME que é independente da trajetória, [17], e, portanto, não levarão a nenhuma singularidade dinâmica, ou seja, $\det(J) \neq 0$ durante todo o caminho percorrido. O erro de acompanhamento de trajetória é definido por

$$\tilde{x}_{ef} := \begin{bmatrix} \dot{\tilde{p}}_a \\ \tilde{p}_a \end{bmatrix}, \quad (6.6)$$

sendo $\dot{\tilde{p}}_a = \dot{p}_a - \dot{p}_a^d$ e $\tilde{p}_a = p_a - p_a^d$.

A equação dinâmica do erro de acompanhamento em espaço de estados fica

$$\dot{\tilde{x}}_{ef} = A_{ef} \tilde{x}_{ef} + B_{ef} (F_a - \bar{M}_{efaa} \ddot{p}_a^d - \bar{C}_{efaa} \dot{p}_a^d - \bar{M}_{efau} \ddot{p}_u - \bar{C}_{efau} \dot{p}_u + \delta_a + F_d), \quad (6.7)$$

com

$$A_{ef}(q, \dot{q}) = \begin{bmatrix} -\bar{M}_{efaa}^{-1} \bar{C}_{efaa} & 0 \\ I & 0 \end{bmatrix} \quad \text{e} \quad B_{ef}(q) = \begin{bmatrix} \bar{M}_{efaa}^{-1} \\ 0 \end{bmatrix}.$$

Visando minimizar os erros de posição e velocidade atuando com um mínimo de torque e

consumo de energia, uma entrada de controle é escolhida, como em [70], tal que

$$\bar{u} = \begin{bmatrix} \bar{M}_{efaa} & \bar{C}_{efaa} \end{bmatrix} \begin{bmatrix} \dot{\tilde{z}}_1 \\ \tilde{z}_1 \end{bmatrix} = \bar{M}_{efaa} T_1 \dot{\tilde{x}}_{ef} + \bar{C}_{efaa} T_1 \tilde{x}_{ef}, \quad (6.8)$$

com $\tilde{z}$ e T_1 introduzidos pela seguinte transformação em $\tilde{x}_{ef}$ dada por

$$\tilde{z} = \begin{bmatrix} \tilde{z}_1 \\ \tilde{z}_2 \end{bmatrix} = T_0 \tilde{x}_{ef} = \begin{bmatrix} T_1 \\ T_2 \end{bmatrix} \tilde{x}_{ef} = \begin{bmatrix} T_{11} & T_{12} \\ 0 & I \end{bmatrix} \begin{bmatrix} \dot{\tilde{p}}_a \\ \tilde{p}_a \end{bmatrix}, \quad (6.9)$$

sendo $T_{11}, T_{12} \in \mathbb{R}^{n \times n}$ matrizes constantes a serem determinadas adequadamente. A entrada de controle definida dessa forma inclui torques seletivos que afetam somente a energia cinética (uma vez que durante a execução da tarefa não é necessário considerar o consumo devido à energia potencial ou otimizar forças dependentes da gravidade), trajetórias de referência (6.6) e uma transformação em espaço de estados (6.9) que elimina as redundâncias entre coordenadas de posição e velocidade sem alterar a ordem do sistema.

Assim

$$\begin{aligned} \bar{u} &= \bar{M}_{efaa} \begin{bmatrix} T_{11} & T_{12} \end{bmatrix} \begin{bmatrix} \ddot{\tilde{p}}_a \\ \dot{\tilde{p}}_a \end{bmatrix} + \bar{C}_{efaa} \begin{bmatrix} T_{11} & T_{12} \end{bmatrix} \begin{bmatrix} \dot{\tilde{p}}_a \\ \tilde{p}_a \end{bmatrix} \\ &= \bar{M}_{efaa} (T_{11} \ddot{\tilde{p}}_a + T_{12} \dot{\tilde{p}}_a) + \bar{C}_{efaa} (T_{11} \dot{\tilde{p}}_a + T_{12} \tilde{p}_a), \end{aligned} \quad (6.10)$$

levando à expressão para a aceleração que rege o acompanhamento de trajetória através da lei de controle aplicada, descrita por

$$\ddot{p}_a = \ddot{p}_a^d - T_{11}^{-1} T_{12} \dot{\tilde{p}}_a - T_{11}^{-1} \bar{M}_{efaa}^{-1} (\bar{C}_{efaa} B^T T_0 \tilde{x}_{ef} - \bar{u}), \quad (6.11)$$

com $B^T = \begin{bmatrix} I & 0 \end{bmatrix}$.

De (6.5) e (6.11), forma-se a relação entre a variável de controle $\bar{u}$ e a força realizada pelo efetuator no espaço inercial em termos dos parâmetros nominais,

$$F_a = \bar{M}_{efaa} (\ddot{p}_a^d - T_{11}^{-1} T_{12} \dot{\tilde{p}}_a - T_{11}^{-1} \bar{M}_{efaa}^{-1} (\bar{C}_{efaa} B^T T_0 \tilde{x}_{ef} - \bar{u})) + \bar{C}_{efaa} \dot{\tilde{p}}_a + \bar{M}_{efaa} \ddot{p}_u + \bar{C}_{efaa} \dot{\tilde{p}}_u. \quad (6.12)$$

A equação dinâmica do erro de acompanhamento de trajetória modificada com base em (6.12) é, conseqüentemente, dada por

$$\dot{\tilde{x}}_{ef} = A_{T_{ef}} \tilde{x}_{ef} + B_{T_{ef}} T_{11} (F_a - \bar{F}(x_{ef}) - \bar{E}(x_{eb}) + \delta_a + F_d), \quad (6.13)$$

sendo

$$\begin{aligned}
 A_{T_{ef}}(q, \dot{q}) &= T_0^{-1} \begin{bmatrix} -\bar{M}_{efaa}^{-1} \bar{C}_{efaa} & 0 \\ T_{11}^{-1} & -T_{11}^{-1} T_{12} \end{bmatrix} T_0, \\
 B_{T_{ef}}(q) &= T_0^{-1} \begin{bmatrix} \bar{M}_{efaa}^{-1}(q) \\ 0 \end{bmatrix}, \\
 x_{ef} &= \begin{bmatrix} q^T & \dot{q}^T & p_a^T & \dot{p}_a^T & (p_a^d)^T & (\dot{p}_a^d)^T & (\ddot{p}_a^d)^T \end{bmatrix}^T, \\
 \bar{F}(x_{ef}) &= \bar{M}_{efaa}(\ddot{p}_a^d - T_{11}^{-1} T_{12} \dot{\ddot{p}}_a) + \bar{C}_{efaa}(\dot{p}_a^d - T_{11}^{-1} T_{12} \dot{\ddot{p}}_a), \\
 x_{eb} &= \begin{bmatrix} q^T & \dot{q}^T & \dot{p}_u^T & \ddot{p}_u^T \end{bmatrix}^T, \\
 \bar{E}(x_{eb}) &= \bar{M}_{efau} \ddot{p}_u + \bar{C}_{efau} \dot{p}_u.
 \end{aligned}$$

6.2 Controlador Adaptativo $\mathcal{H}_\infty$ Não Linear baseado em Redes Neurais

6.2.1 Estimativa do modelo via redes neurais

Seja $H(x_e, \Phi)$ um conjunto de n redes neurais de saída única definidas por $H_k(x_e, \Phi_k)$, $k = 1, \dots, n$, tal que $x_e = [x_{e_j}] \in \mathbb{R}^{q_k}$ é um vetor de entradas e $\Phi_k = [\phi_{k_i}] \in \mathbb{R}^{p_k}$ é um vetor de pesos ajustáveis. Composta por uma camada de entrada com q_k neurônios, uma camada intermediária com p_k neurônios e um neurônio de saída, a estrutura de $H_k(x_e, \Phi_k)$ é ilustrada na Figura 95.

Figura 95: Controle - Estrutura de uma rede neural de saída única.

Os valores do vetor de pesos da camada de entrada $W^k = [w_{ij}^k] \in \mathbb{R}^{p_k \times q_k}$ e do vetor de bias $b^k = [b_i^k] \in \mathbb{R}^{p_k}$ são admitidos constantes e especificados pelo projetista. A função de ativação para os neurônios da camada intermediária, $G_k(\cdot)$, é escolhida como a função tangente hiperbólica, ver [71]. Assim, o ajuste das redes neurais se restringe apenas à atualização dos vetores Φ_k .

A saída de $H_k(x_e, \Phi_k)$ é descrita por

$$H_k(x_e, \Phi_k) = \sum_{i=1}^{p_k} \phi_{k_i} G_k \left(\sum_{j=1}^{q_k} w_{ij}^k x_{e_j} + b_i^k \right), \quad (6.14)$$

tal que, definindo

$$\xi_k^T := \left[\tanh \left(\sum_{j=1}^{q_k} w_{1j}^k x_{e_j} + b_1^k \right) \quad \tanh \left(\sum_{j=1}^{q_k} w_{2j}^k x_{e_j} + b_2^k \right) \quad \cdots \quad \tanh \left(\sum_{j=1}^{q_k} w_{p_k j}^k x_{e_j} + b_{p_k}^k \right) \right],$$

pode-se reescrever (6.14) no formato

$$H_k(x_e, \Phi_k) = \xi_k \Phi_k. \quad (6.15)$$

Assim, o conjunto completo de redes neurais é dado por

$$H(x_e, \Phi) = \begin{bmatrix} H_1(x_e, \Phi_1) \\ H_2(x_e, \Phi_2) \\ \vdots \\ H_n(x_e, \Phi_n) \end{bmatrix} = \begin{bmatrix} \xi_1^T & 0 & \cdots & 0 \\ 0 & \xi_2^T & \vdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \xi_n^T \end{bmatrix} \begin{bmatrix} \Phi_1 \\ \Phi_2 \\ \vdots \\ \Phi_n \end{bmatrix} = \Xi \Phi. \quad (6.16)$$

Considere uma primeira abordagem em que o termo $\bar{F}(x_{e_{ef}}) + \bar{E}(x_{e_b}) - \delta_a$ em (6.13) é completamente desconhecido em estrutura e valores de parâmetros. A rede neural definida em (6.16) é aplicada para aprender o comportamento dinâmico do sistema robótico:

$$\bar{F}(x_{e_{ef}}) + \bar{E}(x_{e_b}) - \delta_a \approx H(x_e, \Phi), \quad (6.17)$$

sendo o vetor de entrada x_e definido por

$$x_e = \left[q^T \quad \dot{q}^T \quad p_a^T \quad \dot{p}^T \quad \ddot{p}_u^T \quad (p_a^d)^T \quad (\dot{p}_a^d)^T \quad (\ddot{p}_a^d)^T \right]^T.$$

No entanto, os valores de q_b , $\dot{q}_b$, $\dot{p}_u$ e $\ddot{p}_u$ seriam necessários na composição desse vetor, mas não são fáceis de serem obtidos na prática. Considerando que uma abordagem baseada em redes neurais é geralmente utilizada quando não é possível fornecer ao modelo do sistema todos os valores dos parâmetros, define-se o vetor x_e como

$$x_e = \left[q_m^T \quad \dot{q}_m^T \quad p_a^T \quad \dot{p}_a^T \quad (p_a^d)^T \quad (\dot{p}_a^d)^T \quad (\ddot{p}_a^d)^T \right]^T, \quad (6.18)$$

evitando a necessidade de qualquer dado dinâmico da base livre flutuante ou relacionado às variáveis passivas. Resultados experimentais ilustrarão a praticabilidade desse argumento.

Definindo o seguinte problema de otimização

$$\Phi^* = \arg \min_{\Phi \in \Omega_\Phi} \max_{x_e \in \Omega_{x_e}} \|H(x_e, \Phi) - (\bar{F}(x_{ef}) + \bar{E}(x_{eb}) - \delta_a)\|_2,$$

a equação de erro modificada (6.13) pode ser reescrita como

$$\begin{aligned} \dot{\tilde{x}}_{ef} &= A_{T_{ef}} \tilde{x}_{ef} + B_{T_{ef}} T_{11} (F_a - \bar{F}(x_{ef}) - \bar{E}(x_{eb}) + \delta_a + F_d + H(x_e, \Phi) - H(x_e, \Phi)) \\ &= A_{T_{ef}} \tilde{x}_{ef} + B_{T_{ef}} T_{11} (F_a - H(x_e, \Phi)) + B_{T_{ef}} T_{11} (H(x_e, \Phi) - \bar{F}(x_{ef}) - \bar{E}(x_{eb}) + \delta_a + F_d) \\ &= A_{T_{ef}} \tilde{x}_{ef} + B_{T_{ef}} u + B_{T_{ef}} \omega, \end{aligned} \quad (6.19)$$

com

$$u = T_{11} (F_a - H(x_e, \Phi)), \quad (6.20)$$

$$\omega = T_{11} (H(x_e, \Phi) - \bar{F}(x_{ef}) - \bar{E}(x_{eb}) + \delta_a + F_d), \quad (6.21)$$

sendo ω referente ao erro de estimativa gerado pelas redes neurais e a distúrbios externos. Uma vez que u é a lei de controle fornecida pelo controlador $\mathcal{H}_\infty$ não linear, F_a pode ser calculada pela seguinte expressão

$$F_a = H(x_e, \Phi) + T_{11}^{-1} u. \quad (6.22)$$

Agora, considere que a estrutura do modelo do manipulador e valores nominais para o termo $\bar{F}(x_{ef})$ estão bem definidos e disponíveis para o controlador. Então, uma segunda abordagem pode ser proposta. Neste caso, as redes neurais são aplicadas para estimar somente o comportamento dinâmico das incertezas paramétricas e da base livre flutuante (considerada uma dinâmica não-modelada), tal que

$$\bar{E}(x_{eb}) - \delta_a \approx H(x_e, \Phi). \quad (6.23)$$

De modo semelhante, x_e é definido por (6.18), o vetor ótimo de parâmetros estimados é dado por

$$\Phi^* = \arg \min_{\Phi \in \Omega_\Phi} \max_{x_e \in \Omega_{x_e}} \|H(x_e, \Phi) - (\bar{E}(x_{eb}) - \delta_a)\|_2,$$

e a equação de erro modificada (6.13) pode ser reescrita como

$$\begin{aligned} \dot{\tilde{x}}_{ef} &= A_{T_{ef}} \tilde{x}_{ef} + B_{T_{ef}} T_{11} (F_a - \bar{F}(x_{ef}) - H(x_e, \Phi)) + B_{T_{ef}} T_{11} (H(x_e, \Phi) - \bar{E}(x_{eb}) + \delta_a + F_d) \\ &= A_{T_{ef}} \tilde{x}_{ef} + B_{T_{ef}} u + B_{T_{ef}} \omega, \end{aligned} \quad (6.24)$$

com

$$u = T_{11}(F_a - \bar{F}(x_{ef}) - H(x_e, \Phi)), \quad (6.25)$$

$$\omega = T_{11}(H(x_e, \Phi) - \bar{E}(x_{eb}) + \delta_a + F_d), \quad (6.26)$$

sendo ω referente ao erro de estimativa gerado pelas redes neurais e a distúrbios externos. Analogamente, seja u a lei de controle fornecida pelo controlador $\mathcal{H}_\infty$ não linear, F_a pode ser então determinada por

$$F_a = \bar{F}(x_{ef}) + H(x_e, \Phi) + T_{11}^{-1}u. \quad (6.27)$$

6.2.2 Controle $\mathcal{H}_\infty$ não linear

A solução para o problema de controle $\mathcal{H}_\infty$ não linear foi desenvolvida em [72] a partir da teoria dos jogos diferenciais, para robôs manipuladores convencionais totalmente atuados. A estratégia consiste em escolher uma função de Lyapunov dependente do estado, tal que o problema minimax gerado pela teoria dos jogos reduz-se a uma equação de Riccati, cuja solução pode ser facilmente encontrada escolhendo-se apropriadamente a função P contida na função de Lyapunov e utilizando a propriedade de anti-simetria das matrizes dinâmicas do sistema. Esta equação é uma particularização da equação de Hamilton-Jacobi para o caso não linear variante no tempo. A mesma técnica foi utilizada em [73] para resolver o problema de controle adaptativo $\mathcal{H}_\infty$ não linear com redes neurais. A proposta neste capítulo é estender essa solução para o caso de manipuladores espaciais com base livre flutuante atuando sobre o erro de acompanhamento de trajetória definido no espaço da tarefa. As equações dinâmicas do erro de estado assim definido, equações (6.19) e (6.24), satisfazem as condições necessárias de positividade e simetria da matriz P e de anti-simetria das matrizes dinâmicas. Utilizando a análise de estabilidade de Lyapunov em [73] verifica-se que o erro $\tilde{x}_{ef}$ é assintoticamente estável com a escolha apropriada da lei de adaptação dos parâmetros da rede neural.

Assim, dado um nível desejado de atenuação de distúrbios $\gamma > 0$, o problema de controle adaptativo $\mathcal{H}_\infty$ não linear é definido para a aplicação proposta pelo seguinte índice de desempenho:

$$\min_u \max_{\omega \in \mathcal{L}_2} \int_0^T (\tilde{x}_{ef}^T Q \tilde{x}_{ef} + u^T R u) dt \leq \tilde{x}_{ef}^T(0) P_0 \tilde{x}_{ef}(0) + \tilde{\Phi}^T(0) Z_0 \tilde{\Phi}(0) + \gamma^2 \int_0^T (\omega^T \omega) dt, \quad (6.28)$$

sendo $\tilde{\Phi} = \Phi^* - \Phi$ o erro de estimativa dos parâmetros do conjunto de redes neurais e, $Q = Q^T > 0, R = R^T > 0, P_0 = P_0^T > 0$ e $Z_0 = Z_0^T > 0$ matrizes de ponderação definidas adequadamente pelo projetista.

A teoria dos jogos diferenciais define a seguinte equação de Bellman-Isaacs a partir de (6.28):

$$-\frac{\partial V(\tilde{x}_{ef}, t)}{\partial t} = \min_u \max_{\omega \in \mathcal{L}_2} \left\{ (\tilde{x}_{ef}^T Q \tilde{x}_{ef} + u^T R u - \gamma^2 \omega^T \omega) + \left(\frac{\partial V(\tilde{x}_{ef}, t)}{\partial \tilde{x}_{ef}} \right)^T \dot{\tilde{x}}_{ef} \right\}, \quad (6.29)$$

definindo o Lagrangeano do sistema por $H(u, \omega) = \tilde{x}_{ef}^T Q \tilde{x}_{ef} + u^T R u - \gamma^2 \omega^T \omega$.

Para qualquer $Z = Z^T > 0$, seja

$$V(\tilde{x}_{ef}, t) = \frac{1}{2} \tilde{x}_{ef}^T P(\tilde{x}_{ef}, t) \tilde{x}_{ef} + \frac{1}{2} \tilde{\Phi}^T Z \tilde{\Phi}, \quad (6.30)$$

a função de Lyapunov selecionada que transforma (6.29) em uma equação de Riccati.

Utilizando o método de completar quadrados, a solução segue determinando-se a expressão para o pior caso de distúrbio ω^* que maximize $H(u, \omega)$:

$$\omega^* = \frac{1}{\gamma^2} B_{T_{ef}}^T P(\tilde{x}_{ef}, t) \tilde{x}_{ef}. \quad (6.31)$$

Aplicando em $H(u, \omega^*)$ as propriedades de simetria e anti-simetria das matrizes dinâmicas do sistema, buscando minimizar $H(u, \omega^*)$, obtém-se as expressões para a lei de controle ótima u^* e para a lei de adaptação dos parâmetros do controlador adaptativo $\dot{\Phi}$:

$$u^* = -R^{-1} B_{T_{ef}}^T P(\tilde{x}_{ef}, t) \tilde{x}_{ef}, \quad (6.32)$$

$$\dot{\Phi} = -Z^{-T} \Xi^T T_{11} B_{T_{ef}}^T P(\tilde{x}_{ef}, t) \tilde{x}_{ef}. \quad (6.33)$$

Dessa forma, (6.29) resulta na equação de Riccati dada por

$$\dot{P}(\tilde{x}_{ef}, t) + P(\tilde{x}_{ef}, t) A_{T_{ef}} + A_{T_{ef}}^T P(\tilde{x}_{ef}, t) - P(\tilde{x}_{ef}, t) B_{T_{ef}} \left(R^{-1} - \frac{1}{\gamma^2} I \right) B_{T_{ef}}^T P(\tilde{x}_{ef}, t) + Q = 0, \quad (6.34)$$

cujas solução é determinada algebricamente escolhendo-se

$$P(\tilde{x}_{ef}, t) = T_0^T \begin{bmatrix} \bar{M}_{efaa} & 0 \\ 0 & K \end{bmatrix} T_0. \quad (6.35)$$

A equação de Riccati (6.34) torna-se então simplificada em uma equação matricial algébrica independente das matrizes paramétricas do sistema na forma

$$\begin{bmatrix} 0 & K \\ K & 0 \end{bmatrix} - T_0^T B \left(R^{-1} - \frac{1}{\gamma^2} I \right)^{-1} B^T T_0 + Q = 0, \quad (6.36)$$

cujas soluções $[K, T_0]$ é dada por

$$K = \frac{1}{2}(Q_1^T Q_2 - Q_2^T Q_1) - \frac{1}{2}(Q_{21}^T - Q_{12})$$

e

$$T_0 = \begin{bmatrix} T_{11} & T_{12} \\ 0 & I \end{bmatrix} = \begin{bmatrix} R_1^T Q_1 & R_1^T Q_2 \\ 0 & I \end{bmatrix},$$

considerando R_1 o resultado da fatoração de Cholesky

$$R_1^T R_1 = \left(R^{-1} - \frac{1}{\gamma^2} I \right)^{-1}$$

e a matriz Q simétrica definida positiva fatorada como

$$Q = \begin{bmatrix} Q_1^T Q_1 & Q_{12} \\ Q_{12}^T & Q_2^T Q_2 \end{bmatrix},$$

tal que $K = K^T > 0$ e $R < \gamma^2 I$.

Substituindo (6.35) em (6.32) e (6.33), a entrada de controle ótima, u^* , definida para o pior caso de distúrbio e a lei de adaptação dos parâmetros do controlador adaptativo, $\dot{\Phi}$, são dadas por

$$u^* = -R^{-1} B^T T_0 \tilde{x}_{ef}, \quad (6.37)$$

$$\dot{\Phi} = -Z^{-T} \Xi^T T_{11} B^T T_0 \tilde{x}_{ef}. \quad (6.38)$$

E dessa forma, pelo critério de Lyapunov, se $\omega \in \mathcal{L}_2[0, T]$, para todo $T \geq 0$, então $\tilde{x}_{ef}(T)$, $\Phi(T)$ e $F_a(T)$ são limitadas.

Portanto, um controlador dinâmico de realimentação de estados dado por

$$\dot{\Phi} = \alpha(t, \tilde{x}_{ef}) = -Z^{-T} \Xi^T T_{11} B^T T_0 \tilde{x}_{ef}, \quad (6.39)$$

$$F_a = F_a(t, \Phi, \tilde{x}_{ef}) = \Xi \Phi - T_{11}^{-1} R^{-1} B^T T_0 \tilde{x}_{ef}, \quad (6.40)$$

é a solução do problema de controle adaptativo $\mathcal{H}_\infty$ não linear baseado em redes neurais sujeito a (6.19) e satisfaz (6.28) para qualquer condição inicial.

E ainda, para a segunda abordagem, um controlador dinâmico de realimentação de estados na forma

$$\dot{\Phi} = \alpha(t, \tilde{x}_{ef}) = -Z^{-T} \Xi^T T_{11} B^T T_0 \tilde{x}_{ef}, \quad (6.41)$$

$$F_a = F_a(t, \Phi, \tilde{x}_{ef}) = \bar{F}(x_{ef}) + \Xi \Phi - T_{11}^{-1} R^{-1} B^T T_0 \tilde{x}_{ef}, \quad (6.42)$$

é a solução do problema de controle adaptativo $\mathcal{H}_\infty$ não linear baseado em redes neurais sujeito a (6.24) e também satisfaz (6.28) para qualquer condição inicial.

Considerando $J(q) = \begin{bmatrix} J_{uu} & J_{ua} \\ J_{ua} & J_{aa} \end{bmatrix}$, na prática, para as duas abordagens, os torques aplicados às juntas ativas do manipulador são determinados por

$$\tau_a = J_{aa}^T F_a. \quad (6.43)$$

Observação 4 *As estratégias de controle adaptativo $\mathcal{H}_\infty$ não linear baseado em redes neurais apresentadas não demandam valores de velocidade e aceleração para a base livre flutuante. O sistema inteligente é aplicado para estimar a dinâmica da base espacial assim como dinâmicas resultantes de incertezas paramétricas e outras dinâmicas não-modeladas. O critério de desempenho $\mathcal{H}_\infty$ por sua vez é aplicado para atenuar os efeitos de distúrbios externos e do erro de estimativa do conjunto de redes neurais.*

7 *Resultados*

Para fins de validação, as soluções de controle adaptativo $\mathcal{H}_\infty$ não linear propostas no Capítulo 6 são aplicadas a um manipulador espacial planar com dois elos e base livre flutuante. Duas situações distintas são propostas para avaliar o desempenho dos controladores formulados no espaço da tarefa, além de validar a aplicação do conceito do MDE.

No primeiro caso, o MDE correspondente é um manipulador planar de base fixa com três elos e com a primeira junta passiva baseado na estrutura do manipulador experimental UARM (Apêndice A). O ME equivalente, simulado, é caracterizado a partir dos parâmetros nominais identificados na Seção 5.1.

No segundo caso, o ME é um manipulador planar de base livre flutuante com dois elos baseado na estrutura do manipulador experimental desenvolvido na Parte I deste trabalho. O MDE, simulado, é caracterizado a partir dos parâmetros nominais identificados na Seção 5.2. Nesta seção será utilizado o sistema identificado por **Robô 1** nos capítulos anteriores.

O software ASCME, apresentado no Capítulo 4, é aplicado para gerar as simulações e interfacear experimentos reais. As matrizes de modelo para o MDE, $M(q)$, $C(q, \dot{q})$ e $J(q)$, estão descritas no Apêndice F.

Uma tarefa de acompanhamento de trajetória é definida para o efetuador do manipulador espacial. As posições cartesianas $p_a = [x_{ef} \ y_{ef}]^T$ do manipulador são escolhidas variáveis controladas, enquanto a orientação do efetuador φ_{ef} é deixada livre.

A trajetória de referência é definida por um semi-círculo iniciado na posição inicial do efetuador e caracterizado por raio = 5 cm. A varredura dos ângulos (0 a π) que determinam a trajetória de referência do semi-círculo segue um polinômio de 5° grau dado por

$$\theta_{traj}(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4 + a_5 t^5, \quad (7.1)$$

sendo

$$\begin{aligned} a_0 &= \theta_{traj}(0), \\ a_1 &= a_2 = [0 \ 0]^T, \\ a_3 &= \frac{10(\theta_{traj}(t_f) - \theta_{traj}(0))}{t_f^3}, \\ a_4 &= \frac{15(\theta_{traj}(t_f) - \theta_{traj}(0))}{t_f^4}, \\ a_5 &= \frac{6(\theta_{traj}(t_f) - \theta_{traj}(0))}{t_f^5}, \end{aligned}$$

satisfazendo as seguintes condições:

$$\begin{aligned} \theta_{traj}(0) &= 0, \\ \theta_{traj}(t_f) &= \pi, \\ \dot{\theta}_{traj}(0) &= \dot{\theta}_{traj}(t_f) = [0 \ 0]^T, \\ \ddot{\theta}_{traj}(0) &= \ddot{\theta}_{traj}(t_f) = [0 \ 0]^T. \end{aligned}$$

A trajetória proposta deve ser executada pelo efetuator em $t_f = 3s$.

Durante a execução da tarefa, um distúrbio de torque de energia limitada, começando em $t = 1s$, é introduzido na forma de senóides amortecidas:

$$\tau_d = \begin{bmatrix} \beta_1 e^{-5t} \sin(6\pi t) \\ \beta_2 e^{-5t} \sin(6\pi t) \end{bmatrix},$$

sendo $\beta = [\beta_1 \ \beta_2]$ definido na Seção 7.1 a seguir.

Para uma comparação quantitativa dos resultados obtidos pelos controladores avaliados, dois índices de desempenho são definidos: a norma $\mathcal{L}_2$ do vetor de estados

$$\mathcal{L}_2[\tilde{x}_{ef}] = \left(\frac{1}{(t_r - t_0)} \int_{t_0}^{t_r} \|\tilde{x}_{ef}(t)\|_2^2 dt \right)^{1/2},$$

sendo $\|\cdot\|_2$ a norma Euclidiana, e a soma dos torques aplicados

$$E[\tau_a] = \sum_{i=1}^2 \left(\int_{t_0}^{t_r} |\tau_{a_i}(t)| dt \right),$$

sendo t_r o tempo gasto pelo robô para alcançar a posição desejada. A norma $\mathcal{L}_2$ do erro de acompanhamento da trajetória é um índice de desempenho amplamente utilizado na literatura

[74, 75, 76, 77, 78, 79, 80]. A análise da soma dos torques aplicados está diretamente relacionada ao consumo de energia, uma preocupação em aplicações espaciais.

7.1 Simulação

O distúrbio introduzido durante as simulações foi caracterizado por $\beta = [0.025 \ 0.015]$ e é exibido na Figura 96. Comparado ao torque aplicado em casos em que não há inserção de distúrbios, o distúrbio τ_d atinge picos da ordem de 70% do máximo valor dos torques nominais e insere no sistema uma quantidade de energia da ordem de 5% da energia gasta para executar a tarefa sem distúrbios.

Figura 96: Resultados - Simulação: Distúrbio de torque.

Incertezas paramétricas foram aplicadas aos valores de massa m , momento de inércia I , comprimento dos elos W e posição do centro de massa dos elos l_c como

$$\delta = \begin{bmatrix} 0.7m & 1.2I & 1.7W & 0.5l_c \end{bmatrix}.$$

Os valores multiplicativos foram definidos de forma arbitrária.

Para claramente identificar a atuação dos controladores, resultados obtidos sem adicionar distúrbios externos ou incertezas ao modelo simulado (caso nominal) são apresentados nas Figuras 97 a 100 para o caso de simulação com o UARM e nas Figuras 105 a 108 para o caso de simulação com o UARM-E. As Figuras 101 a 104 e 109 a 112 ilustram os resultados para a situação com distúrbios de torque e incertezas paramétricas (caso com distúrbios).

Para estabelecer uma base de comparação, seja um controlador por torque calculado definido por

$$\ddot{p}_a = K_p \tilde{p}_a + K_d \dot{\tilde{p}}_a + K_i \int \tilde{p}_a + \ddot{p}_a^d, \quad (7.2)$$

com $K_p = 50I_2$, $K_d = \begin{bmatrix} 50\sqrt{|K_p|} & 0 \\ 0 & 1.5\sqrt{|K_p|} \end{bmatrix}$ e $K_i = I_2$ selecionados empiricamente. Apli-

cando (7.2) à equação das forças relacionadas às variáveis ativas resultante de (6.5), obtém-se F_a e os torques nas juntas são calculados por (6.43). É importante ressaltar que essa técnica é fortemente baseada em modelo, e portanto, exige o conhecimento preciso da estrutura do modelo dinâmico da base livre flutuante e do manipulador, assim como dos valores de seus parâmetros. Resultados de simulação são apresentados pelas Figuras 97, 101, 105 e 109.

O nível desejado de atenuação de distúrbios definido para os controladores $\mathcal{H}_\infty$ não lineares propostos é $\gamma = 2$. As matrizes de ponderação R e Q são selecionadas empiricamente atendendo às restrições impostas para existência da solução de (6.36). A matriz R pondera inversamente a força de atuação da lei de controle $\mathcal{H}_\infty$. As matrizes Q_1 e Q_2 , que compõem a matriz de ponderação Q , são ajustadas para cada controlador visando seu melhor desempenho em termos dos erros de acompanhamento de trajetória (que é o objetivo do sistema de controle), observando a aparência dos gráficos obtidos e os valores da norma $\mathcal{L}_2$ do vetor de estados. Observa-se que Q_1 atua diretamente no tratamento do erro de velocidade enquanto pondera inversamente a força de atuação da lei de controle $\mathcal{H}_\infty$. Já Q_2 atua no ajuste do erro de posição. A matriz Z é determinada para otimizar a velocidade de aprendizado da dinâmica desconhecida. As matrizes de ponderação selecionadas são apresentadas na Tabela 14.

Tabela 14: Resultados - Simulação: Matrizes de ponderação selecionadas.

UARM				
$\gamma = 2$	R	Q_1	Q_2	Z
$\mathcal{H}_\infty$ baseado em modelo	$\begin{bmatrix} 3 & 0 \\ 0 & 3 \end{bmatrix}$	$\begin{bmatrix} 0.16 & 0 \\ 0 & 0.16 \end{bmatrix}$	$\begin{bmatrix} 49 & 0 \\ 0 & 9 \end{bmatrix}$	-
Adaptativo $\mathcal{H}_\infty$ RN (1)	$\begin{bmatrix} 3 & 0 \\ 0 & 3 \end{bmatrix}$	$\begin{bmatrix} 0.04 & 0 \\ 0 & 0.04 \end{bmatrix}$	$\begin{bmatrix} 100 & 0 \\ 0 & 64 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$
Adaptativo $\mathcal{H}_\infty$ RN (2)	$\begin{bmatrix} 3 & 0 \\ 0 & 3 \end{bmatrix}$	$\begin{bmatrix} 0.16 & 0 \\ 0 & 0.16 \end{bmatrix}$	$\begin{bmatrix} 4 & 0 \\ 0 & 9 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$
UARM-E				
$\gamma = 2$	R	Q_1	Q_2	Z
$\mathcal{H}_\infty$ baseado em modelo	$\begin{bmatrix} 3 & 0 \\ 0 & 3 \end{bmatrix}$	$\begin{bmatrix} 0.16 & 0 \\ 0 & 0.16 \end{bmatrix}$	$\begin{bmatrix} 144 & 0 \\ 0 & 144 \end{bmatrix}$	-
Adaptativo $\mathcal{H}_\infty$ RN (1)	$\begin{bmatrix} 1.5 & 0 \\ 0 & 1.5 \end{bmatrix}$	$\begin{bmatrix} 0.0625 & 0 \\ 0 & 0.16 \end{bmatrix}$	$\begin{bmatrix} 144 & 0 \\ 0 & 144 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$
Adaptativo $\mathcal{H}_\infty$ RN (2)	$\begin{bmatrix} 0.015 & 0 \\ 0 & 0.015 \end{bmatrix}$	$\begin{bmatrix} 0.0625 & 0 \\ 0 & 0.0625 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$

As Figuras 98, 102, 106 e 110 apresentam os gráficos obtidos para o controlador $\mathcal{H}_\infty$ não linear baseado em modelo.

Para os controladores adaptativos $\mathcal{H}_\infty$ não lineares baseados em redes neurais propostos, seja $n = 2$ a dimensão de p_a determinada pelo número de juntas do manipulador espacial (juntas ativas no MDE). Seja o conjunto de redes neurais definido por

$$H(x_e, \Phi) = [H_1(x_e, \Phi_1) \quad H_2(x_e, \Phi_2)]^T$$

com $p_1 = p_2 = 7$ neurônios na camada intermediária, o vetor de bias dado por $b_1 = b_2 = [-3 \quad -2 \quad -1 \quad 0 \quad 1 \quad 2 \quad 3]$ e a matriz de pesos da camada de entrada selecionada como por

$$W_i^1 = W_i^2 = [w_{ij}^1] = [w_{ij}^2] = [1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad -1 \quad -1 \quad -1 \quad -1 \quad -1 \quad -1].$$

O vetor de parâmetros desconhecidos Φ é definido por $\Phi = [\Phi_1^T \quad \Phi_2^T]^T$, com

$$\Phi_1^T = [\phi_{11} \quad \cdots \quad \phi_{17}],$$

$$\Phi_2^T = [\phi_{21} \quad \cdots \quad \phi_{27}],$$

e a matriz Ξ pode ser calculada $\xi_1^T = [\xi_{11} \quad \cdots \quad \xi_{17}]$ e $\xi_2^T = [\xi_{21} \quad \cdots \quad \xi_{27}]$. Os resultados de simulação para os controladores adaptativos $\mathcal{H}_\infty$ não lineares baseados em redes neurais são mostrados na Figuras 99, 103, 107 e 111. Para a segunda abordagem, os resultados são mostrados nas Figuras 100, 104, 108 e 112.

A Tabela 15 apresenta os valores obtidos para os índices de desempenho.

Tabela 15: Resultados - Simulação: Índices de desempenho.

UARM				
	Caso nominal		Caso com distúrbios	
	$\mathcal{L}_2[\tilde{x}_{ef}]$	$E[\tau_a](\text{Nms})$	$\mathcal{L}_2[\tilde{x}_{ef}]$	$E[\tau_a](\text{Nms})$
Torque calculado	1.24×10^{-4}	0.0449	210×10^{-4}	0.0592
$\mathcal{H}_\infty$ baseado em modelo	1.39×10^{-4}	0.0449	24×10^{-4}	0.0458
Adaptativo $\mathcal{H}_\infty$ RN (1)	3.63×10^{-4}	0.0452	20×10^{-4}	0.0559
Adaptativo $\mathcal{H}_\infty$ RN (2)	16×10^{-4}	0.0442	22×10^{-4}	0.0444
UARM-E				
	Caso nominal		Caso com distúrbios	
	$\mathcal{L}_2[\tilde{x}_{ef}]$	$E[\tau_a](\text{Nms})$	$\mathcal{L}_2[\tilde{x}_{ef}]$	$E[\tau_a](\text{Nms})$
Torque calculado	1.2×10^{-4}	0.1023	190×10^{-4}	0.1123
$\mathcal{H}_\infty$ baseado em modelo	0.5×10^{-4}	0.1022	1.5×10^{-4}	0.1022
Adaptativo $\mathcal{H}_\infty$ RN (1)	7.8×10^{-4}	0.1043	5.7×10^{-4}	0.1065
Adaptativo $\mathcal{H}_\infty$ RN (2)	8.9×10^{-4}	0.1020	12×10^{-4}	0.1020

Figura 97: Resultados - UARM - Simulação: Caso nominal: Trajetória, erros de posição e velocidades do efetuador - Torque calculado.

Figura 98: Resultados - UARM - Simulação: Caso nominal: Trajetória, erros de posição e velocidades do efetuador - $\mathcal{H}_\infty$ não linear.

Figura 99: Resultados - UARM - Simulação: Caso nominal: Trajetória, erros de posição e velocidades do efetuador - $\mathcal{H}_\infty$ não linear baseado em redes neurais (1).

Figura 100: Resultados - UARM - Simulação: Caso nominal: Trajetória, erros de posição e velocidades do efetuador - $\mathcal{H}_\infty$ não linear baseado em redes neurais (2).

Figura 101: Resultados - UARM - Simulação: Caso com distúrbios: Trajetória, erros de posição e velocidades do efetuador - Torque calculado.

Figura 102: Resultados - UARM - Simulação: Caso com distúrbios: Trajetória, erros de posição e velocidades do efetuador - $\mathcal{H}_\infty$ não linear.

Figura 103: Resultados - UARM - Simulação: Caso com distúrbios: Trajetória, erros de posição e velocidades do efetuador - $\mathcal{H}_\infty$ não linear baseado em redes neurais (1).

Figura 104: Resultados - UARM - Simulação: Caso com distúrbios: Trajetória, erros de posição e velocidades do efetuador - $\mathcal{H}_\infty$ não linear baseado em redes neurais (2).

Figura 105: Resultados - UARM-E - Simulação: Caso nominal: Trajetória, erros de posição e velocidades do efetuador - Torque calculado.

Figura 106: Resultados - UARM-E - Simulação: Caso nominal: Trajetória, erros de posição e velocidades do efetuador - $\mathcal{H}_\infty$ não linear.

Figura 107: Resultados - UARM-E - Simulação: Caso nominal: Trajetória, erros de posição e velocidades do efetuador - $\mathcal{H}_\infty$ não linear baseado em redes neurais (1).

Figura 108: Resultados - UARM-E - Simulação: Caso nominal: Trajetória, erros de posição e velocidades do efetuador - $\mathcal{H}_\infty$ não linear baseado em redes neurais (2).

Figura 109: Resultados - UARM-E - Simulação: Caso com distúrbios: Trajetória, erros de posição e velocidades do efetuador - Torque calculado.

Figura 110: Resultados - UARM-E - Simulação: Caso com distúrbios: Trajetória, erros de posição e velocidades do efetuador - $\mathcal{H}_\infty$ não linear.

Figura 111: Resultados - UARM-E - Simulação: Caso com distúrbios: Trajetória, erros de posição e velocidades do efetuador - $\mathcal{H}_\infty$ não linear baseado em redes neurais (1).

Figura 112: Resultados - UARM-E - Simulação: Caso com distúrbios: Trajetória, erros de posição e velocidades do efetuador - $\mathcal{H}_\infty$ não linear baseado em redes neurais (2).

7.2 Implementação

O distúrbio introduzido durante os experimentos foi caracterizado por $\beta = [0.12 \ 0.08]$ e é exibido na Figura 113. Observa-se que a intensidade dos distúrbios nesse caso é maior do que a apresentada na seção de simulações, Figura 96. O intuito é manter a proporção do nível de distúrbios em relação ao torque utilizado experimentalmente para executar a tarefa quando não há inserção de distúrbios. Deve-se também levar em consideração que o modelo nominal do MDE apresenta incertezas paramétricas reais, uma vez que o UARM é um robô real atuando em um ambiente real. Assim, é desnecessário aplicar incertezas ao modelo nas implementações experimentais.

Figura 113: Resultados - UARM - Implementação: Distúrbio de torque.

O controlador por torque calculado aplicado como base de comparação é caracterizado por $K_p = 200I_2$, $K_d = 90I_2$ e $K_i = 20I_2$ em (7.2). Estes ganhos foram ajustados empiricamente, observando o melhor desempenho do controlador para o caso nominal e buscando atingir resultados comparáveis aos obtidos pelo controlador $\mathcal{H}_\infty$ não-linear baseado em modelo. Dessa forma, entende-se que é possível identificar a capacidade de robustez das estratégias propostas.

O nível de atenuação de distúrbios definido para os controladores $\mathcal{H}_\infty$ propostos é $\gamma = 2$. As matrizes de ponderação R , Q e Z foram ajustadas empiricamente para cada controlador visando seu melhor desempenho, como no caso das simulações. As matrizes selecionadas são mostradas na Tabela 16. Como na Seção 7.1, para que a atuação dos controladores seja claramente identificada as Figuras 116 e 117 mostram resultados obtidos sem a inserção de distúrbios de torque (caso nominal¹) enquanto as Figuras 120 e 121 mostram os resultados para a situação com perturbações de torque (caso com distúrbios) para ambos os controladores baseados em redes neurais. Uma marca vermelha ressalta, nas Figuras 120 e 121, o instante em que o distúrbio é iniciado ($t = 1s$). A Tabela 17 mostra os valores médios para os índices de desempenho

¹Deve-se atentar que nos experimentos realizados incertezas paramétricas são inerentes ao sistema no estudo do caso nominal.

obtidos de 5 experimentos.

Tabela 16: Resultados - UARM - Implementação: Matrizes de ponderação selecionadas.

$\gamma = 2$	R	Q_1	Q_2	Z
$\mathcal{H}_\infty$ baseado em modelo	$\begin{bmatrix} 0.07 & 0 \\ 0 & 0.07 \end{bmatrix}$	$\begin{bmatrix} 4 & 0 \\ 0 & 4 \end{bmatrix}$	$\begin{bmatrix} 324 & 0 \\ 0 & 324 \end{bmatrix}$	-
Adaptativo $\mathcal{H}_\infty$ RN (1)	$\begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 225 & 0 \\ 0 & 225 \end{bmatrix}$	$\begin{bmatrix} 10 & 0 \\ 0 & 10 \end{bmatrix}$
Adaptativo $\mathcal{H}_\infty$ RN (2)	$\begin{bmatrix} 0.2 & 0 \\ 0 & 0.2 \end{bmatrix}$	$\begin{bmatrix} 4 & 0 \\ 0 & 4.84 \end{bmatrix}$	$\begin{bmatrix} 100 & 0 \\ 0 & 100 \end{bmatrix}$	$\begin{bmatrix} 0.3 & 0 \\ 0 & 0.3 \end{bmatrix}$

Tabela 17: Resultados - UARM - Implementação: Índices de desempenho.

	Caso nominal		Caso com distúrbios	
	$\mathcal{L}_2[\tilde{x}_{ef}]$	$E[\tau_a](\text{Nms})$	$\mathcal{L}_2[\tilde{x}_{ef}]$	$E[\tau_a](\text{Nms})$
Torque calculado	0.0079	0.2354	0.0105	0.2801
$\mathcal{H}_\infty$ baseado em modelo	0.0074	0.2314	0.0082	0.2560
Adaptativo $\mathcal{H}_\infty$ RN (1)	0.0099	0.1641	0.0100	0.2093
Adaptativo $\mathcal{H}_\infty$ RN (2)	0.0090	0.2107	0.0084	0.2006

Figura 114: Resultados - UARM - Implementação: Caso nominal: Trajetória, erros de posição e velocidades do efetuador - Torque calculado.

Figura 115: Resultados - UARM - Implementação: Caso nominal: Trajetória, erros de posição e velocidades do efetuador - $\mathcal{H}_\infty$ não linear.

Figura 116: Resultados - UARM - Implementação: Caso nominal: Trajetória, erros de posição e velocidades do efetuador - $\mathcal{H}_\infty$ não linear baseado em redes neurais (1).

Figura 117: Resultados - UARM - Implementação: Caso nominal: Trajetória, erros de posição e velocidades do efetuador - $\mathcal{H}_\infty$ não linear baseado em redes neurais (2).

Figura 118: Resultados - UARM - Implementação: Caso com distúrbios: Trajetória, erros de posição e velocidades do efetuador - Torque calculado.

Figura 119: Resultados - UARM - Implementação: Caso com distúrbios: Trajetória, erros de posição e velocidades do efetuador - $\mathcal{H}_\infty$ não linear.

Figura 120: Resultados - UARM - Implementação: Caso com distúrbios: Trajetória, erros de posição e velocidades do efetuador - $\mathcal{H}_\infty$ não linear baseado em redes neurais (1).

Figura 121: Resultados - UARM - Implementação: Caso com distúrbios: Trajetória, erros de posição e velocidades do efetuador - $\mathcal{H}_\infty$ não linear baseado em redes neurais (2).

7.3 Análise dos Resultados

Uma comparação entre os resultados gráficos mostra que os controladores propostos apresentam a capacidade de rejeitar distúrbios, atenuando seus efeitos na execução do acompanhamento de trajetória eficientemente. O controle por torque calculado, por sua vez, apresenta um resultado adequado na Figura 97, quando o modelo nominal da planta é completamente definido e nenhuma incerteza ou distúrbio está presente. No entanto, sua ação de controle não é capaz de rejeitar os efeitos dos distúrbios e das incertezas quando estes ocorrem, Figura 101. Por estes resultados a característica de robustez do critério $\mathcal{H}_\infty$ aplicado pode ser claramente verificada. Considerando o ambiente hostil onde um robô espacial opera, o que pode deteriorar sua estrutura e suas características físicas, os resultados obtidos pelos controladores robustos adaptativos propostos são bastante interessantes.

Uma análise quantitativa baseada nas Tabelas 15 e 17 ratifica os resultados verificados nos gráficos. A robustez proporcionada pelo controlador $\mathcal{H}_\infty$ não linear é ilustrada pelo desempenho alcançado no índice $\mathcal{L}_2[\tilde{x}_{ef}]$ quando comparado aos valores apresentados pelo método do torque calculado. Os resultados apresentados pelas abordagens baseadas em redes neurais exibem sua eficiência em estimar o efeito das incertezas e a dinâmica desconhecida da base livre flutuante. Para a primeira abordagem (1), nota-se nos experimentos que uma quantidade de energia foi gasta na atenuação dos efeitos dos distúrbios. Por outro lado, a segunda estratégia proposta (2), a qual considera conhecido um modelo nominal para a dinâmica do manipulador, mantém sua capacidade de minimização de erros sem elevar seus esforços de torque.

Observando o formato das curvas de torque na Figura 122 é possível identificar a forma de atuação dos controladores.

Figura 122: Resultados - UARM-E - Simulação: Caso com distúrbios: Torques nas juntas - (a) $\mathcal{H}_\infty$ não linear, (b) $\mathcal{H}_\infty$ não linear baseado em redes neurais (1), (c) $\mathcal{H}_\infty$ não linear baseado em redes neurais (2).

Verifica-se que os sinais enviados pelos controladores baseados em modelo são suaves, enquanto que a saída do controlador identificada na curva (b) apresenta um padrão oscilatório que não é exclusivamente uma reação ao distúrbio, mas é modificado por ele. No entanto, como pode ser visto na Figura 123, esse sinal não produz oscilação no posicionamento das juntas do robô.

Figura 123: Resultados - UARM-E - Simulação: Caso com distúrbios: Posição das juntas - $\mathcal{H}_\infty$ não linear baseado em redes neurais (1).

Poderia-se atribuir esse comportamento à atividade de adaptação dos parâmetros das redes neurais, e consequentemente à contribuição de sua saída ao sinal total do torque fornecido pelo controlador $\mathcal{H}_\infty$ não linear baseado em redes neurais (1). Porém, como pode ser visto na Figura 124 (a), os sinais de saída das redes neurais (RN - Junta 2 e RN - Junta 3) não apresentam oscilação. Dessa forma, a oscilação é proveniente da lei de controle $u(t)$. O nível de atuação de $u(t)$ é ponderado inversamente por R . Ao elevarmos o valor de R a oscilação no sinal de torque diminui, mas a reação ao distúrbio também diminui, gerando maiores erros de estado. Sendo assim, deve-se sempre observar a reação do sistema à oscilação do sinal de torque e definir o melhor ponto de projeto entre a quantidade de erro permitida e a possibilidade de desgaste dos equipamentos devido a essa atuação oscilatória.

Figura 124: Resultados - UARM-E - Simulação: Caso com incertezas: Adaptação dos parâmetros e atuação das redes neurais - (a) $\mathcal{H}_\infty$ não linear baseado em redes neurais (1), (b) $\mathcal{H}_\infty$ não linear baseado em redes neurais (2).

A Figura 124 apresenta também o processo de adaptação dos parâmetros das redes neurais, cuja saída deve estimar o comportamento dinâmico desconhecido do robô. Portanto, no tipo de problema estudado neste trabalho, acompanhamento de trajetória, os parâmetros não ficarão estáveis em um valor fixo, pois parametrizam o acompanhamento dinâmico proposto e não os parâmetros físicos do robô. Deve-se acrescentar que a ordem de grandeza dos valores desses parâmetros, assim como a influência do sinal de saída das redes neurais no sinal total de torque aplicado às juntas, é inversamente proporcional à matriz de ponderação Z .

Os sinais de torque aplicados ao UARM nos procedimentos experimentais são apresentados nos gráficos da Figura 125. A oscilação de alta frequência observada nestes gráficos é resultado do ruído dos sensores de posição e dos sinais de velocidade provenientes da diferenciação numérica. Ao ajustar o filtro do cálculo de velocidade é possível atenuar essa oscilação ruidosa, porém é gerado um atraso no sinal de entrada do controlador que pode levar à instabilidade.

Figura 125: Resultados - UARM - Implementação: Caso com distúrbios: Torques nas juntas - (a) Torque calculado, (b) $\mathcal{H}_\infty$ não linear, (c) $\mathcal{H}_\infty$ não linear baseado em redes neurais (1), (d) $\mathcal{H}_\infty$ não linear baseado em redes neurais (2).

Uma comparação entre as estratégias propostas, através dos resultados gráficos e numéricos, com e sem distúrbios, aponta para um desempenho equivalente entre as estratégias robustas. Portanto, o projetista deve ponderar entre a demanda de energia e a disponibilidade do modelo da planta em termos dos requisitos da aplicação e da presença ou não de grandes distúrbios. Deve-se enfatizar que nenhuma das abordagens $\mathcal{H}_\infty$ propostas exige a medida de valores de aceleração da base ou do braço.

Conclusão

Satélites de telecomunicações custam pelo menos 250 milhões de dólares e são projetados para uma vida útil média em órbita de 10-15 anos. Uma vez que sua carga de propelente tenha sido totalmente consumida, os satélites são dispensados, mesmo quando sua carga funcional de telecomunicações continua ativa, [8]. Entretanto, sistemas capacitados para executar missões de serviço autônomas ou tele-operadas estão em desenvolvimento nas principais agências espaciais, buscando modificar esse cenário. Com os avanços nas aplicações espaciais, a robotização dos procedimentos tende a tornar as missões espaciais mais seguras e baratas. O objetivo é aperfeiçoar a utilização de equipamentos em órbita e evitar a exposição de astronautas ao ambiente hostil do espaço. Assim, o trabalho apresentado nessa tese se enquadra na demanda do desenvolvimento tecnológico espacial em pauta.

Neste trabalho, uma plataforma de pesquisa experimental foi projetada e construída para possibilitar o estudo de comportamento e de técnicas de controle de manipuladores de base livre flutuante. Definida como um sistema planar, cuja flutuação é baseada na técnica de colchões de ar, a plataforma experimental construída se distingue dos sistemas existentes por ser bastante versátil. Sua estrutura mecânica modular possibilita diversas configurações, de uma a seis juntas ativas, com um ou dois braços. Experimentos com elos rígidos ou flexíveis, de comprimentos variados, poderão ser realizados. Além disso, o tipo de motor selecionado possibilita também o estudo de manipuladores com juntas passivas. Essa característica é interessante no âmbito das aplicações espaciais, uma vez que sistemas subatuados podem ser aplicados em procedimentos de economia de energia, ou ainda, de tratamento de falhas. Ainda vale ressaltar que a construção de mesas de teste que possibilitem o estudo de sistemas dinâmicos que não estão sujeitos à força da gravidade tem importância também para a pesquisa em controle de satélites e sistemas subaquáticos. A interface com o usuário é apresentada em um ambiente gráfico que possibilita a configuração do sistema e dos experimentos de forma amigável. Sendo o núcleo do laço de controle implementado com base na teoria de modelagem descrita em [25], o sistema computacional possibilita também a atuação no robô de base fixa UARM, configurado como dinamicamente equivalente a um manipulador espacial. A obtenção de resultados experimentais nesse tipo de sistema é um diferencial do trabalho em relação ao existente na literatura.

Uma vez que operam em um ambiente hostil e não-controlado, os manipuladores espaciais

estão sujeitos a uma série de fontes de perturbação que podem afetar diretamente seu desempenho. A capacidade de amenizar o efeito desses distúrbios é um resultado muito interessante para esses sistemas. Além dos benefícios óbvios dessa propriedade de robustez, pode-se ressaltar que ao manter a execução da tarefa com bom desempenho mesmo sujeito a incertezas de modelagem e perturbações externas, o sistema robótico espacial pode manter sua condição de flutuante, evitando a necessidade de atuação na base espacial durante a tarefa, economizando combustível e energia. Dessa forma, a aplicação de técnicas de controle robusto baseadas no critério $\mathcal{H}_\infty$ para manipuladores espaciais desenvolvida nesta tese constitui outra contribuição efetiva deste trabalho. O problema de acompanhamento de trajetória de manipuladores espaciais de base livre flutuante foi solucionado no espaço da tarefa mediante a presença de incertezas no modelo e perturbações externas, além de considerar a dinâmica das variáveis não-controladas do efetuador como totalmente desconhecidas. Formulada com base no conceito de modelagem definido em [25] aliado à introdução de redes neurais para estimar as parcelas incertas ou desconhecidas do modelo do sistema espacial, a aplicação dos controladores não lineares $\mathcal{H}_\infty$ geraram resultados promissores.

Considerando trabalhos futuros, o sistema construído define uma poderosa ferramenta para a análise comportamental de suas possíveis configurações, assim como para a implementação experimental de sistemas de controle em manipuladores de base livre flutuante, tornando interessante sua utilização em cursos de graduação e pós-graduação em robótica. A continuação da pesquisa utilizando essa plataforma abordará a implementação das técnicas de controle robusto desenvolvidas em configurações diferentes, buscando em especial o controle simultâneo de posicionamento da base e do efetuador. A modelagem do sistema com dois braços utilizando o conceito do MDE ampliará as possibilidades para outra gama de pesquisas a serem desenvolvidas no âmbito dos manipuladores cooperativos com base livre flutuante. Além disso o desenvolvimento de técnicas de controle híbrido de posição e força deve também ser avaliado para esses sistemas, considerando a aplicação do sensor de força no efetuador. Ainda no sentido do desenvolvimento de controladores para manipuladores de base livre flutuante, pode-se considerar incluir o movimento da base livre no índice de desempenho $\mathcal{H}_\infty$, buscando atenuar os efeitos do acoplamento dinâmico.

Referências Bibliográficas

- [1] PEDERSEN, L. et al. A survey of space robotics. In: *International Symposium on Artificial Intelligence, Robotics and Automation in Space (iSAIRAS)*. Nara, Japan: [s.n.], 2003. (7th). Disponível em: <<http://robotics.estec.esa.int/i-SAIRAS/isairas2003/index.html>>.
- [2] NASA. 2004–2010. Disponível em: <<http://www.nasa.gov>>.
- [3] SICILIANO, B.; KHATIB, O. *Handbook of Robotics*. Berlin, Germany: Springer-Verlag, 2008. ISBN:978-3-540-23957-4, e-ISBN:978-3-540-30301-5.
- [4] CSA. 2004–2010. Disponível em: <<http://www.asc-csa.gc.ca/eng/iss/overview.asp>>.
- [5] NASA. International space station assembly: A construction site in orbit. In: LYNDON B. JOHNSON SPACE CENTER. *NASA Facts*. Houston, USA, 1999.
- [6] JAXA. *Kibo - Japanese Experiment Module*. 2010. Disponível em: <<http://kibo.jaxa.jp/en/index.html>>.
- [7] ESA. 2004–2010. Disponível em: <<http://www.esa.int/TEC/Robotics/>>.
- [8] LANDZETTEL, K. et al. Robotic on-orbit servicing - DLR's experience and perspective. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*. Beijing, China: [s.n.], 2006. p. 4587 – 4594. Doi: 10.1109/IROS.2006.282164.
- [9] DLR. *Ongoing Space Robotic Missions*. 2008–2010. Disponível em: <<http://www.dlr.de/rm/en/desktopdefault.aspx/tabid-3825/>>.
- [10] MILLER, R. H.; MINSKY, M. L.; SMITH, D. B. S. *Space Applications of Automation, Robotics and Machine Intelligence Systems (ARAMIS)*. [S.l.], 1982.
- [11] YOSHIDA, K. Engineering Test Satellite VII flight experiments for space robot dynamics and control: Theories on laboratory test beds ten years ago, now in orbit. *International Journal of Robotics Research*, v. 22, n. 5, p. 321–335, 2003.
- [12] RUPP, T. et al. Flight dynamics challenges of the german on-orbit servicing mission DEOS. In: *International Symposium on Space Flight Dynamics*. Toulouse, France: [s.n.], 2009. (21st). Disponível em: <http://www.weblab.dlr.de/rbrt/pdf/ISSFD_09f.pdf>.
- [13] SELLMAIER, F. et al. On-orbit servicing missions: Challenges and solutions for spacecraft operations. In: *AIAA SpaceOps Conference*. Huntsville, USA: [s.n.], 2010. Disponível em: <http://www.weblab.dlr.de/rbrt/pdf/SpaceOps_10b.pdf>.
- [14] DUBOWSKY, S.; PAPADOPOULOS, E. The kinematics, dynamics, and control of free-flying and free-floating space robotic systems. *IEEE Transactions on Robotics and Automation*, v. 9, n. 5, p. 531–543, Oct 1993. Doi:10.1109/70.258046.

- [15] TORRES, M. A.; DUBOWSKY, S. Minimizing spacecraft attitude disturbances in space manipulators systems. *AIAA Journal of Guidance, Control, and Dynamics*, v. 15, n. 4, p. 1010–1017, Jul-Aug 1992. Doi:10.2514/3.20936.
- [16] SAKAWA, Y. Trajectory planning of a free-flying robot by using the optimal control. *Optimal Control Applications and Methods*, v. 20, n. 5, p. 235–248, Sep/Oct 1999. Doi:10.1002/(SICI)1099-1514(199909/10)20:5<235::AID-OCA658>3.0.CO;2-I.
- [17] PAPADOPOULOS, E.; DUBOWSKY, S. On the nature of control algorithms for free-floating space manipulators. *IEEE Transactions on Robotics and Automation*, v. 7, n. 6, p. 750–758, Dec 1991. Doi:10.1109/70.105384.
- [18] DUBOWSKY, S.; TORRES, M. A. Path planning for space manipulators to minimize spacecraft attitude disturbances. In: *IEEE International Conference on Robotics and Automation (ICRA)*. Sacramento, USA: [s.n.], 1991. v. 3, p. 2522–2528. Doi:10.1109/ROBOT.1991.132005.
- [19] CACCAVALE, F.; SICILIANO, B. Quaternion-based kinematic control of redundant spacecraft/manipulator systems. In: *IEEE International Conference on Robotics and Automation (ICRA)*. Seoul, Korea: [s.n.], 2001. v. 1, p. 435–440. Doi:10.1109/ROBOT.2001.932589.
- [20] PAPADOPOULOS, E.; TORTOPIDIS, I.; NANOS, K. Smooth planning for free-floating space robots using polynomials. In: *IEEE International Conference on Robotics and Automation (ICRA)*. Barcelona, Spain: [s.n.], 2005. p. 4272–4277.
- [21] HUANG, P.; XU, Y. Pso-based time-optimal trajectory planning for space robot with dynamic constraints. In: *IEEE International Conference on Robotics and Biomimetics (ROBIO)*. Kunming, China: [s.n.], 2006. p. 1402–1407. Doi:10.1109/ROBIO.2006.340134.
- [22] TORTOPIDIS, I.; PAPADOPOULOS, E. Point-to-point planning: Methodologies for underactuated space robots. In: *IEEE International Conference on Robotics and Automation (ICRA)*. Orlando, USA: [s.n.], 2006. p. 3861–3866. Doi:10.1109/ROBOT.2006.1642293.
- [23] LIU, Z. et al. Multi-objective genetic algorithms for trajectory optimization of space manipulator. In: *IEEE Conference on Industrial Electronics and Applications (ICIEA)*. Xi'an, China: [s.n.], 2009. (4th), p. 2810–2815. Doi:10.1109/ICIEA.2009.5138722.
- [24] VAFA, Z.; DUBOWSKY, S. The kinematics and dynamics of space manipulators: The virtual manipulator approach. *International Journal of Robotics Research*, MIT Press, v. 9, n. 4, p. 3–21, Ago 1990. Doi:10.1177/0278364990000900401.
- [25] LIANG, B.; XU, Y.; BERGERMAN, M. Dynamically equivalent manipulator for space manipulator system: Part 1. In: *IEEE International Conference on Robotics and Automation (ICRA)*. Albuquerque, USA: [s.n.], 1997. v. 4, p. 2765–2770. Doi:10.1109/ROBOT.1997.606705.
- [26] GU, Y. L.; XU, Y. A normal form augmentation approach to adaptive control of space robot systems. In: *IEEE International Conference on Robotics and Automation (ICRA)*. Atlanta, USA: [s.n.], 1993. v. 2, p. 731–737. Doi:10.1109/ROBOT.1993.291872.
- [27] ABIKO, S.; HIRZINGER, G. Frontiers in adaptive control. In: _____. Vienna, Austria: InTech, 2009. cap. An Adaptive Control for a Free-Floating Space Robot by Using Inverted Chain Approach, p. 1–16.

- [28] CRAIG, J. *Adaptive Control of Mechanical Manipulators*. Reading, MA: Addison-Wesley, 1988.
- [29] WOERKOM, P. L. M.; GUELMAN, M.; EHRENWALD, L. Integrated adaptive control for space manipulators. *Acta Astronautica*, v. 38, n. 3, p. 161–174, Feb 1996. Doi:10.1016/0094-5765(96)00067-7.
- [30] PARLAKTUNA, O.; OZKAN, M. Adaptive control of free-floating space robots in cartesian coordinates. *Advanced Robotics*, v. 18, n. 9, p. 943–959, 2004. Doi:10.1163/1568553042225732.
- [31] FU, Y. et al. A dynamic control method for free-floating space manipulator in task space. In: *IEEE International Conference on Robotics and Biomimetics*. Sanya, China: [s.n.], 2007. p. 1230–1235. Doi:10.1109/ROBIO.2007.4522340.
- [32] TAKAGI, T.; SUGENO, M. Fuzzy identification of systems and its applications to modeling and control. *IEEE Transactions on Systems, Man and Cybernetics*, v. 15, p. 116–132, 1985.
- [33] HORNIK, K.; STINCHCOMBE, M.; WHITE, H. Multilayer feedforward networks are universal approximators. *Neural Networks*, v. 2, n. 5, p. 359–366, 1989. Doi:10.1016/0893-6080(89)90020-8.
- [34] NARENDRA, K. S.; PARTHASARATHY, K. Identification and control of dynamical systems using neural networks. *IEEE Transactions on Neural Networks*, v. 1, n. 1, p. 4–27, Mar 1990. Doi:10.1109/72.80202.
- [35] JANG, J. S. R. ANFIS - Adaptive Network-based Fuzzy Inference System. *IEEE Transactions on Systems, Man and Cybernetics*, v. 23, n. 3, p. 665–685, May-Jun 1993. Doi:10.1109/21.256541.
- [36] POLYCARPOU, M. M. Stable adaptive neural control scheme for nonlinear systems. *IEEE Transactions on Automatic Control*, v. 41, n. 3, p. 447–451, Mar 1996. Doi:10.1109/9.486648.
- [37] SANNER, R. M.; VANCE, E. E. Adaptive control of free-floating space robots using neural networks. In: *American Control Conference (ACC)*. Seattle, USA: [s.n.], 1995. v. 4, p. 2790–2794.
- [38] TAVEIRA, T. F. P. A.; SIQUEIRA, A. A. G.; TERRA, M. H. Adaptive nonlinear $\mathcal{H}_\infty$ controllers applied to a free-floating space manipulator. In: *IEEE International Conference on Control Applications (CCA)*. Munich, Germany: [s.n.], 2006. p. 1476–1481. Doi:10.1109/CCA.2006.286094.
- [39] HUANG, D.; CHEN, L. Inverse kinematic control of free-floating space robot system based on a mutual mapping neural network. In: *World Congress on Intelligent Control and Automation (WCICA)*. Chongqing, China: [s.n.], 2008. (7th), p. 8666–8670. Doi:10.1109/WCICA.2008.4594293.
- [40] GUO, Y.; CHEN, L. Adaptive neural network control of coordinated motion of dual-arm space robot system with uncertain parameters. In: *World Congress on Intelligent Control and Automation (WCICA)*. Chongqing, China: [s.n.], 2008. (7th), p. 4842–4846. Doi:10.1109/WCICA.2008.4594534.

- [41] PAZELLI, T. F. P. A. T.; SIQUEIRA, A. A. G.; TERRA, M. H. Mixed model based/ANFIS adaptive robust controller with $\mathcal{H}_\infty$ criterion applied to free-floating space manipulators. In: JPL - NASA. *International Symposium on Artificial Intelligence, Robotics and Automation in Space (iSAIRAS)*. Los Angeles, USA, 2008. (9th).
- [42] PAZELLI, T. F. P. A. T.; TERRA, M. H.; SIQUEIRA, A. A. G. Neural network robust controllers applied to free-floating space manipulators in task space. In: AACC, IFAC. *American Control Conference (ACC)*. Baltimore, USA, 2010. p. 3648–3653.
- [43] FENG, B. et al. Robust chattering-free sliding mode control of space robot in task space. *Journal of Control Theory and Applications*, v. 6, n. 2, p. 146–152, 2008. Doi:10.1007/s11768-008-6058-6.
- [44] GUO, Y.; CHEN, L. Robust control of dual-arm space robot system with two objects in joint space. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*. Beijing, China: [s.n.], 2006. p. 5091–5095. Doi:10.1109/IROS.2006.282598.
- [45] ZHONGYI, C.; FUCHUN, S.; JING, C. Disturbance observer-based robust control of free-floating space manipulators. *IEEE Systems Journal*, v. 2, n. 1, p. 114–119, Mar 2008. Doi:10.1109/JSYST.2007.914871.
- [46] PATHAK, P. M. et al. A scheme for robust trajectory control of space robots. *Simulation Modelling Practice and Theory*, v. 16, n. 9, p. 1337–1349, Oct 2008. Doi:10.1016/j.simpat.2008.06.011.
- [47] WANG, H.; XIE, Y. Passivity based adaptive jacobian tracking for free-floating space manipulators without using spacecraft acceleration. *Automatica*, v. 45, n. 6, p. 1510–1517, Jun 2009. Doi:10.1016/j.automatica.2009.02.013.
- [48] KANG, W. Nonlinear $\mathcal{H}_\infty$ control and its application to rigid spacecraft. *IEEE Transactions on Automatic Control*, v. 40, n. 7, p. 1281–1285, Jul 1995. Doi:10.1109/9.400476.
- [49] DALSMO, M.; EGELAND, O. State feedback $\mathcal{H}_\infty$ -suboptimal control of a rigid spacecraft. *IEEE Transactions on Automatic Control*, v. 42, n. 8, p. 1186–1191, Aug 1997. Doi:10.1109/9.618253.
- [50] CHEN, B. S.; WU, C. S.; JAN, Y. W. Adaptive fuzzy mixed $\mathcal{H}_2/\mathcal{H}_\infty$ attitude control of spacecraft. *IEEE Transactions on Aerospace and Electronic Systems*, v. 36, n. 4, p. 1343–1359, Oct 2000. Doi:10.1109/7.892682.
- [51] SKULLESTAD, A.; GILBERT, J. M. $\mathcal{H}_\infty$ control of a gravity gradient stabilized satellite. *Control Engineering Practice*, v. 8, n. 9, p. 975–983, Sep 2000. Doi:10.1016/S0967-0661(00)00044-7.
- [52] YANG, C. D.; SUN, Y. P. Mixed $\mathcal{H}_2/\mathcal{H}_\infty$ state-feedback design for microsatellite attitude control. *Control Engineering Practice*, v. 10, n. 9, p. 951–970, Sep 2002. Doi:10.1016/S0967-0661(02)00049-7.
- [53] PAZELLI, T. F. P. A. T.; SIQUEIRA, A. A. G.; TERRA, M. H. Controle robusto de manipuladores espaciais no espaço das juntas e no espaço cartesiano. In: *Congresso Brasileiro de Automática (CBA)*. Salvador, Brasil: [s.n.], 2006. (XVI), p. 2957–2962. (in portuguese).

- [54] MENON, C. et al. Issues and new solutions for testing free-flying robots. In: *International Astronautical Congress*. Vancouver, Canada: [s.n.], 2004. (55).
- [55] SCHWARTZ, J. L.; PECK, M. A.; HALL, C. D. Historical review of air-bearing spacecraft simulators. *Journal of Guidance, Control, and Dynamics*, v. 26, n. 4, p. 513–522, 2003.
- [56] HEPPLER, G. R.; MCPHEE, J. J.; SHI, P. *Modelling of a Free-Floating Robotic Manipulator*. Waterloo, Canada, 1999.
- [57] MADDOX, D. L. *An Experimental Testbed for a Free-Floating Manipulator*. Tese (Mestrado) — Naval Postgraduate School, Monterey, USA, 1993. 44p.
- [58] SCHUBERT, H.; HOW, J. Space construction: an experimental testbed to develop enabling technologies. In: *Telemanipulator and Telepresence Technologies*. Pittsburgh, USA: [s.n.], 1997. (IV).
- [59] BONING, P. et al. An experimental study of the control of space robot teams assembling large flexible space structures. In: . [S.l.: s.n.], 2008.
- [60] PAZELLI, T. F. P. A. T. *Controladores Adaptativos Não Lineares com Critério $\mathcal{H}_\infty$ Aplicados em Robôs Espaciais*. Tese (Mestrado) — Escola de Engenharia de São Carlos - Universidade de São Paulo, São Carlos, 2006. 109p.
- [61] PATHAK, P. M.; MUKHERJEE, A.; DASGUPTA, A. Impedance control of space robots using passive degrees of freedom in controller domain. *Journal of Dynamic Systems Measurement and Control - Transactions of the ASME*, v. 127, n. 4, p. 564–578, Dec 2005.
- [62] NAKANISHI, J.; MISTRY, M.; SCHAAL, S. Inverse dynamics control with floating base and constraints. In: *IEEE International Conference on Robotics and Automation (ICRA)*. Roma, Italy: [s.n.], 2007. p. 1942–1947. Doi:10.1109/ROBOT.2007.363606.
- [63] MOOSAVIAN, S. A. A.; HOMAEINEJAD, M. R. Control of space free-flying robots using regulated sliding mode controller. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*. Beijing, China: [s.n.], 2006. p. 3131–3136. Doi:10.1109/IROS.2006.282333.
- [64] CHANG, Y. C.; CHEN, B. S. Adaptive tracking control design of constrained robot systems. *International Journal of Adaptive Control Signal Processing*, v. 12, p. 495–526, 1998.
- [65] PETRONILHO, A.; SIQUEIRA, A. A. G.; TERRA, M. H. Adaptive $\mathcal{H}_\infty$ tracking control design via neural networks of a constrained robot system. In: *IEEE Conference on Decision and Control*. Seville, Spain: [s.n.], 2005. (44th), p. 5528–5533.
- [66] NOGUEIRA, S. L. et al. Adaptive fuzzy nonlinear $\mathcal{H}_\infty$ tracking control design of a constrained robot system. In: *Mediterranean Conference on Control and Automation (MED)*. Ajaccio, France: [s.n.], 2008. (16th), p. 362–367. Doi:10.1109/MED.2008.4602181.
- [67] KULAKOWSKI, B. T.; GARDNER, J. F.; SHEARER, J. L. *Dynamic Modeling and Control of Engineering Systems*. 3. ed. New York, USA: Cambridge University Press, 2007.
- [68] LEWIS, F. L.; ABDALLAH, C. T.; DAWSON, D. M. *Control of Robot Manipulators*. New York: Macmillan Publishing Company, 1993.

- [69] SIQUEIRA, A. A. G.; TERRA, M. H. Nonlinear and markovian $\mathcal{H}_\infty$ controls of underactuated manipulators. *IEEE Transactions on Control Systems Technology*, v. 12, n. 6, p. 811–826, Nov 2004. Doi:10.1109/TCST.2004.833626.
- [70] JOHANSSON, R. Quadratic optimization of motion coordination and control. *IEEE Transactions on Automatic Control*, v. 35, n. 11, p. 1197–1208, Nov 1990. Doi:10.1109/9.59805.
- [71] HAYKIN, S. *Redes Neurais: Princípios e Prática*. 2. ed. Porto Alegre, Brasil: Bookman, 2001. Traduzido por Paulo Martins Engel.
- [72] CHEN, B. S.; LEE, T. S.; FENG, J. H. A nonlinear $\mathcal{H}_\infty$ control design in robotic systems under parameter perturbation and external disturbance. *International Journal of Control*, v. 59, n. 2, p. 439–461, 1994. Doi:10.1080/00207179408923085.
- [73] CHANG, Y. C.; CHEN, B. S. A nonlinear adaptive $\mathcal{H}_\infty$ tracking control design in robotic systems via neural networks. *IEEE Transactions on Control Systems Technology*, v. 5, n. 1, p. 13–29, 1997. Doi:10.1109/87.553662.
- [74] WHITCOMB, L. L.; RIZZI, A. A.; KODISTSCHEK, D. E. Comparative experiments with a new adaptive controller for robot arms. *IEEE Transactions on Robotics and Automation*, v. 9, n. 1, p. 59–70, Feb 1993. Doi:10.1109/70.210795.
- [75] BERGHUIS, H.; ROEBBERS, H.; NIJMEIJER, H. Experimental comparison of parameter estimation methods in adaptive robot control. *Automatica*, v. 31, n. 9, p. 1275–1285, Sept 1995. Doi:10.1016/0005-1098(95)00046-Y.
- [76] JARITZ, A.; SPONG, M. W. An experimental comparison of robust control algorithms on a direct drive manipulator. *IEEE Transactions on Control Systems Technology*, v. 4, n. 6, p. 627–640, Nov 1996. Doi:10.1109/87.541692.
- [77] REYES, F.; KELLY, R. Experimental evaluation of model-based controllers on a direct-drive robot arm. *Mechatronics*, v. 11, n. 3, p. 267–282, Apr 2001. Doi:10.1016/S0957-4158(00)00008-8.
- [78] SIQUEIRA, A. A. G.; TERRA, M. H. Nonlinear $\mathcal{H}_\infty$ controllers for underactuated cooperative manipulators. *Robotica*, v. 25, n. 1, p. 1–8, Jan 2007. Doi:10.1017/S0263574706003201.
- [79] INOUE, R. S.; SIQUEIRA, A. A. G.; TERRA, M. H. Experimental results on the nonlinear $\mathcal{H}_\infty$ control via quasi-lpv representation and game theory for wheeled mobile robots. *Robotica*, v. 27, n. 4, p. 547–553, 2009. Doi:10.1017/S0263574708004931.
- [80] SIQUEIRA, A. A. G.; TERRA, M. H. Neural network-based $\mathcal{H}_\infty$ control for fully actuated and underactuated cooperative manipulators. *Control Engineering Practice*, v. 17, n. 3, p. 418–425, Mar 2009. Doi:10.1016/j.conengprac.2008.09.007.

APÊNDICE A – UnderActuated Robot Manipulator

O UARM é um manipulador planar horizontal com 3 elos rígidos em série conectados por juntas rotacionais. Cada junta é composta por um motor CC *brushless*, um freio pneumático e um encoder óptico com decodificador de quadratura. O encoder fornece as posições e a velocidade da junta é obtida por diferenciação numérica e filtragem. As juntas e elos são numerados de 1 a 3, sendo a junta 1 e o elo 1 os mais próximos à base do manipulador, que é fixa a uma mesa de granito, horizontal e plana, Figura 126. Um sistema de ar comprimido produz um filme de ar entre a mesa e a base das juntas e do efetuador, fazendo com que estes componentes flutuem sobre a mesa, reduzindo consideravelmente o atrito entre estas partes além de anular o efeito da gravidade. Os motores podem ser configurados para girar livremente, caracterizando uma junta passiva. Assim, o UARM foi construído especialmente para a aplicação de técnicas de controle com subatuação e pode se apresentar nas seguintes configurações:

- AAA - Três juntas ativas;
- PAA - Junta 1 passiva, juntas 2 e 3 ativas;
- APA - Junta 2 passiva, juntas 1 e 3 ativas;
- AAP - Junta 3 passiva, juntas 1 e 2 ativas;
- PPA - Juntas 1 e 2 passivas, junta 3 ativa;
- APP - Juntas 2 e 3 passivas, junta 1 ativa;
- PAP - Juntas 1 e 3 passivas, junta 2 ativa.

O modelo dinâmico do manipulador de base fixa planar é obtido através da teoria de Lagrange, e a estrutura de suas matrizes características $M(q)$, $C(q, \dot{q})$ e $J(q)$ são dadas no Apêndice F. Como o manipulador UARM é horizontal o termo gravitacional do modelo é nulo. Os parâmetros nominais cinemáticos e dinâmicos do robô são apresentados na Tabela 18.

Figura 126: Apêndice A - UARM.

Tabela 18: Apêndice A - Parâmetros do UARM.

i	m_i	I_i	W_i	lc_i
	(kg)	(kgm ²)	(m)	(m)
Junta 1	0.850	0.0153	0.203	0.096
Junta 2	0.850	0.0100	0.203	0.096
Junta 3	0.625	0.0100	0.203	0.077

APÊNDICE B – Descrição Técnica dos Equipamentos

Tabela 19: Apêndice B - Descrição técnica do atuador.

Motor Elétrico	EC 90 flat
Tipo	Motor elétrico <i>brushless</i>
Fabricante	Maxon
Potência	90 W
Tensão nominal	48 VCC
Corrente nominal (sem carga)	130 mA
Máxima corrente contínua @ 1640rpm	2.12 A
Velocidade nominal (sem carga)	2080 rpm
Máxima velocidade permitida	5000 rpm
Máximo torque contínuo @ 1640rpm	0.494 Nm
Torque contínuo travado	4530 mNm
Constante de torque, K_M	217 mNm/A
Constante de velocidade, K_n	44 rpm/V
Gradiente velocidade/torque, $\Delta n / \Delta M$	0.466 rpm/mNm
Resistência de enrolamento	2.30 Ohm (entre fases)
Inércia do rotor	3060 gcm ²
Massa do motor	648 g

Tabela 20: Apêndice B - Descrição técnica do encoder.

Encoder	E6S
Fabricante	US Digital
Princípio de funcionamento	óptico
Tensão de alimentação	5 VCC
Linhas	2000/rev
Linhas com quadratura	8000/rev
Precisão de posicionamento	0.045°
Número de canais	2
Sinal de saída	TTL

Tabela 21: Apêndice B - Descrição técnica do sistema de visão.

Sistema de visão	SMX-160C
Tipo de sensor	CMOS colorido
Resolução da imagem	até 2208 x 3000 pixels
Taxa de amostragem	31 quadros/s @ 1280 x 1024
Conjunto de lentes	NMV-12HC, 12mm f/1.4
Campo de visão	2 m x 2 m
Precisão de posicionamento	5 mm
Interface	USB 2.0
Tensão de alimentação	5 V (via interface USB)
Corrente de alimentação	360 mA
Máximo consumo de energia	2.28 W

Tabela 22: Apêndice B - Descrição técnica do dispositivo de comunicação.

Dispositivo de comunicação	
Módulo de RF	MaxStream XBee Series 2
Padrão	802.15.4
Frequência	2.4 GHz
Alcance - indoor	30 m
Alcance - outdoor	90 m
Potência do sinal de saída	1mW
Taxa de transmissão	250 Kbits/s
Sensibilidade de recepção	-92 dBm
Antena	chip
Topologias de rede	Point-to-point; Point-to-multipoint; peer-to-peer; mesh
Número de canais	16
Consumo - transmissão	40 mA @ 3.3 VCC
Consumo - recepção	40 mA @ 3.3 VCC

Tabela 23: Apêndice B - Descrição técnica do dispositivo de comando.

Módulo de comando	
Microprocessador	Rabbit© 4000 @ 29.49 MHz
Memória Flash	512K
SRAM	512K
I/O	até 49 linhas de I/O paralelas digitais
Entradas analógicas	4 canais 0-1.2 VCC
Resolução do conversor A/D	10 bits
Barramento de dados auxiliar	8 linhas de dados e 6 linhas de endereço
Portas seriais	6 portas CMOS de alta velocidade
Taxa de comunicação serial	até CLK/8
Timers	10 de 8 bits, 1 de 10 bits, 1 de 16 bits
PWM	4 canais
Decodificador de quadratura	2
Alimentação	3.3 VCC
Consumo com Zigbee	150 mA transmitindo; 80 mA não-transmitindo; 20 μ A em repouso
Módulo de RF	MaxStream XBee Series 2

Tabela 24: Apêndice B - Descrição técnica da bateria.

Bateria	
Tipo	Polímero de lítio
Tensão	22.2V
Capacidade	4100mAh
Taxa de descarga	15C (61500mA)
Número de células	6 (conectadas em série)
Dimensão	135mm x 44mm x 60mm
Peso	649g

Tabela 25: Apêndice B - Descrição técnica do amplificador.

Amplificador	Maxon DEC 50/5
Tensão de alimentação (V_{CC})	10-50 VCC
Máxima tensão de saída	0.95 V_{CC}
Máxima corrente de saída	10 A
Corrente contínua de saída	5 A
Nível alto de tensão de entrada digital, (V_{IH})	> 2.4 V
Nível baixo de tensão de entrada digital, (V_{IL})	< 0.8 V
Massa	155g
Dimensões (L x W x H)	95 x 75 x 24 mm

Tabela 26: Apêndice B - Descrição técnica do DAC.

DAC	TLV5631
Tensão de alimentação (DV_{DD}, AV_{DD})	5 VCC
Corrente de alimentação	21 mA
Nível alto de tensão de entrada digital, (V_{IH})	> 2.4 V
Nível baixo de tensão de entrada digital, (V_{IL})	< 1 V
Máxima tensão de saída analógica	$AV_{DD}-0.4$ V
Corrente de saída	1 mA
Canais de saída analógica	8
Interface serial de entrada digital	SPI
Resolução	10 bits

Tabela 27: Apêndice B - Descrição técnica do demultiplexador.

Demultiplexador	SN74LS259
Tensão de alimentação (V_{CC})	5 VCC
Corrente de alimentação	36 mA
Nível alto de tensão de entrada, (V_{IH})	> 2 V
Nível baixo de tensão de entrada, (V_{IL})	< 0.8 V
Nível alto de tensão de saída, (V_{OH})	3.5 V
Nível baixo de tensão de saída, (V_{OL})	0.35 V
Canais de saída endereçáveis	8

Tabela 28: Apêndice B - Descrição técnica do CPLD.

CPLD	EPM7128S
Alimentação interna (V_{CCINT})	5 VCC
Alimentação I/O de saída (V_{CCIO})	5 VCC ou 3.3 VCC
Quantidade de portas lógicas	2500
Quantidade de macrocélulas	128
Quantidade máxima de I/Os disponíveis	100

Tabela 29: Apêndice B - Descrição técnica do decodificador de quadratura.

Decodificador de quadratura	HCTL2017
Tensão de alimentação (V_{DD})	5 VCC
Nível alto de tensão de entrada, (V_{IH})	> 3.5 V
Nível baixo de tensão de entrada, (V_{IL})	< 1.5 V
Nível alto de tensão de saída, (V_{OH})	4.5 V
Nível baixo de tensão de saída, (V_{OL})	0.4 V
Resolução do contador	16 bits
Clock de operação	14 MHz
Interface de saída	Tristate

Tabela 30: Apêndice B - Descrição técnica do conversor LVTTL.

Conversor LVTTL	SN74LVCH16T245
Tensão de alimentação (V_{CCA})	3.3 VCC
Nível alto de tensão de entrada - A, (V_{IHA})	> 2 V
Nível baixo de tensão de entrada - A, (V_{ILA})	< 0.8 V
Nível alto de tensão de saída - A, (V_{OHA})	> 2.4 V, -24 mA
Nível baixo de tensão de saída - A, (V_{OLA})	< 0.55 V, 24 mA
Tensão de alimentação (V_{CCB})	5 VCC
Nível alto de tensão de entrada - B, (V_{IHB})	> 3.5 V
Nível baixo de tensão de entrada - B, (V_{ILB})	< 1.5 V
Nível alto de tensão de saída - B, (V_{OHB})	> 3.8 V, -32 mA
Nível baixo de tensão de saída - B, (V_{OLB})	< 0.55 V, 32 mA
Interface de saída	Tristate

Tabela 31: Apêndice B - Descrição técnica do conversor CC-CC.

Conversor CC-CC	TPS54283
Tensão de alimentação (V_{PVDD1}, V_{PVDD2})	4.5 - 28 VCC
Canais	2
Capacidade de corrente por canal	2.0 A
Frequência de chaveamento	300 kHz

Tabela 32: Apêndice B - Descrição técnica do supervisor de tensão.

Supervisor	TL7757
Tensão de alimentação (V_{CC})	1 - 7 VCC
Tensão de <i>threshold</i> (V_{IT-})	4.55 VCC

APÊNDICE C – Dimensionamento das Peças Mecânicas

Figura 127: Apêndice C - Base(2).

- Furos com diâmetro = 3mm -> Profundidade = 5mm (não vaza ou parafuso interno)
- Furos com diâmetro = 4mm -> Profundidade = 7mm (vaza)
- Furos com diâmetro = 5mm -> Profundidade = 5mm (não vaza ou parafuso interno)

Figura 128: Apêndice C - Prateleira.

Figura 129: Apêndice C - Acessórios.

Figura 130: Apêndice C - Câmara de ar.

Figura 131: Apêndice C - Conexão Junta-Elo.

Figura 132: Apêndice C - Copo.

Figura 133: Apêndice C - Tampa da junta.

Figura 134: Apêndice C - Base da junta.

Figura 135: Apêndice C - Base da câmara de ar.

Figura 136: Apêndice C - Fim de curso.

Figura 137: Apêndice C - Elo da junta.

Figura 138: Apêndice C - Elo do efetuator.

Figura 139: Apêndice C - Conexão Elo-Junta.

Figura 140: Apêndice C - Conexão em L.

APÊNDICE D – Conexões Elétricas

Tabela 33: Apêndice D - Interface Driver - Motor

Sinais Driver → Motor	Pin - Driver	Pin - Motor
Alimentação dos Sensores Hall	9	3
Terra	10	6
Enrolamento do motor (1)	5	7
Enrolamento do motor (2)	4	8
Enrolamento do motor (3)	3	4
Sinais Motor → Driver	Pin - Driver	Pin - Motor
Sensor Hall (1)	8	1
Sensor Hall (2)	7	2
Sensor Hall (3)	6	5

Tabela 34: Apêndice D - Interface DAC - Driver.

Sinais DAC → Driver	Pin - DAC	Pin - Driver
Velocidade (Driver 1)	12 (OUTA)	15
Velocidade (Driver 2)	13 (OUTB)	15
Velocidade (Driver 3)	14 (OUTC)	15
Velocidade (Driver 4)	15 (OUTD)	15
Velocidade (Driver 5)	6 (OUTE)	15
Velocidade (Driver 6)	7 (OUTF)	15

Tabela 35: Apêndice D - Interface DEMUX - Driver.

Sinais DEMUX1 → Driver	Pin - DEMUX1	Pin - CPLD	Pin - Driver
Desabilita (Driver 1)	4 (Q0)	15	11
Desabilita (Driver 2)	5 (Q1)	18	11
Desabilita (Driver 3)	6 (Q2)	22	11
Desabilita (Driver 4)	7 (Q3)	11	11
Desabilita (Driver 5)	9 (Q4)	8	11
Desabilita (Driver 6)	10 (Q5)	4	11
Sinais DEMUX2 → Driver	Pin - DEMUX2	Pin - CPLD	Pin - Driver
Direção (Driver 1)	4 (Q0)	16	12
Direção (Driver 2)	5 (Q1)	20	12
Direção (Driver 3)	6 (Q2)	33	12
Direção (Driver 4)	7 (Q3)	12	12
Direção (Driver 5)	9 (Q4)	9	12
Direção (Driver 6)	10 (Q5)	5	12
Sinais DEMUX3 → Driver	Pin - DEMUX3	Pin - CPLD	Pin - Driver
Freio (Driver 1)	4 (Q0)	17	13
Freio (Driver 2)	5 (Q1)	21	13
Freio (Driver 3)	6 (Q2)	24	13
Freio (Driver 4)	7 (Q3)	81	13
Freio (Driver 5)	9 (Q4)	10	13
Freio (Driver 6)	10 (Q5)	6	13

Tabela 36: Apêndice D - Interface μ C - DEMUX.

Sinais μC → DEMUX1	Pin - μC	Pin - CPLD	Pin - DEMUX1
Endereçamento (lsb)	33 (PE1)	49	1 (A0)
Endereçamento	34 (PE2)	50	2 (A1)
Endereçamento (msb)	35 (PE3)	51	3 (A2)
Entrada de dados	21 (PB5)	55	13 (D)
Habilita entrada	20 (PB4)	52	14 ($\bar{E}$)
Sinais μC → DEMUX2	Pin - μC	Pin - CPLD	Pin - DEMUX2
Endereçamento (lsb)	33 (PE1)	49	1 (A0)
Endereçamento	34 (PE2)	50	2 (A1)
Endereçamento (msb)	35 (PE3)	51	3 (A2)
Entrada de dados	22 (PB6)	56	13 (D)
Habilita entrada	20 (PB4)	52	14 ($\bar{E}$)
Sinais μC → DEMUX3	Pin - μC	Pin - CPLD	Pin - DEMUX3
Endereçamento (lsb)	33 (PE1)	49	1 (A0)
Endereçamento	34 (PE2)	50	2 (A1)
Endereçamento (msb)	35 (PE3)	51	3 (A2)
Entrada de dados	23 (PB7)	54	13 (D)
Habilita entrada	20 (PB4)	52	14 ($\bar{E}$)

Tabela 37: Apêndice D - Interface Encoder - DecodQuad.

Sinais Encoder → DecQ	Pin - Encoder	Pin - DecQ
Canal A	3	7 (chA)
Canal B	5	6 (chB)

Tabela 38: Apêndice D - Interface DecQ - CPLD - μC .

Sinais DecQ → μC	Pin - DecQ	Pin - CPLD(i)	Pin - CPLD(o)	Pin - μC
D0 (DecQ1 - DecQ6)	1	39	69	8 (PA0)
D1 (DecQ1 - DecQ6)	15	46	68	9 (PA1)
D2 (DecQ1 - DecQ6)	14	45	67	10 (PA2)
D3 (DecQ1 - DecQ6)	13	44	65	11 (PA3)
D4 (DecQ1 - DecQ6)	12	41	64	12 (PA4)
D5 (DecQ1 - DecQ6)	11	40	63	13 (PA5)
D6 (DecQ1 - DecQ6)	10	37	61	14 (PA6)
D7 (DecQ1 - DecQ6)	9	36	60	15 (PA7)

Tabela 39: Apêndice D - Interface μC - DEMUX.

Sinais μC → DEMUX4	Pin - μC	Pin - CPLD	Pin - DEMUX4
Endereçamento (lsb)	33 (PE1)	49	1 (A0)
Endereçamento	34 (PE2)	50	2 (A1)
Endereçamento (msb)	35 (PE3)	51	3 (A2)
Entrada de dados	36 (PE4)	57	13 (D)
Habilita entrada	39 (PE7)	58	14 ($\overline{E}$)

Tabela 40: Apêndice D - Interface μC - Conversor LVTTL.

Sinais μC → Conv	Pin - μC	Pin - Conv
Seleciona saída (DecQ1 - DecQ6)	37	17
Reset (DecQ1 - DecQ6)	38	16

Tabela 41: Apêndice D - Interface DEMUX - Conversor LVTTL.

Sinais DEMUX4 → Conv	Pin - DEMUX4	Pin - CPLD	Pin - Conv
Habilita (DecQ1)	4 (Q0)	73	23
Habilita (DecQ2)	5 (Q1)	74	22
Habilita (DecQ3)	6 (Q2)	76	21
Habilita (DecQ4)	7 (Q3)	75	20
Habilita (DecQ5)	9 (Q4)	77	19
Habilita (DecQ6)	10 (Q5)	79	18

Tabela 42: Apêndice D - Interface Conversor LVTTL - DecodQuad.

Sinais Conv → DecQ	Pin - Conv	Pin - DecQ
Habilita (DecQ1)	10	4
Habilita (DecQ2)	9	4
Habilita (DecQ3)	8	4
Habilita (DecQ4)	7	4
Habilita (DecQ5)	6	4
Habilita (DecQ6)	5	4
Seleciona saída (DecQ1 - DecQ6)	4	3
Reset (DecQ1 - DecQ6)	3	5

Tabela 43: Apêndice D - Interface μC - Chaves.

Sinais μC → Chaves	Pin - μC	Pin - Chaves
Chave 1	40	14
Chave 2	41	13
Chave 3	43	12
Chave 4	44	11
Chave 5	45	10
Chave 6	46	9

APÊNDICE E – Esquemáticos do Sistema Eletrônico

Figura 142: Apêndice E - Esquemático do módulo de energia.

Figura 145: Apêndice E - Esquemático do módulo de comunicação.

APÊNDICE F – Matrizes do Modelo Dinâmico

Sejam as variáveis auxiliares:

$$\begin{aligned} t_1 &= -m_2 W_1 l c_2 \sin(q_2), \\ t_2 &= -m_3 W_1 W_2 \sin(q_2), \\ t_3 &= -m_3 W_1 l c_3 \sin(q_2 + q_3), \\ t_4 &= -m_3 W_2 l c_3 \sin(q_3). \end{aligned}$$

A matriz de inércia é dada por

$$M(q) = \begin{bmatrix} M_{11}(q) & M_{12}(q) & M_{13}(q) \\ M_{21}(q) & M_{22}(q) & M_{23}(q) \\ M_{31}(q) & M_{32}(q) & M_{33}(q) \end{bmatrix}$$

$$\begin{aligned} M_{11}(q) &= I_1 + I_2 + I_3 + m_1 l c_1^2 + m_2 (W_1 + l c_2^2 + 2W_1 l c_2 \cos(q_2)) + \\ &+ m_3 (W_1^2 + W_2^2 + l c_3^2 + 2W_1 W_2 \cos(q_2) + 2W_2 l c_3 \cos(q_3) + \\ &+ 2W_1 l c_3 \cos(q_2 + q_3)), \\ M_{12}(q) &= I_2 + I_3 + m_2 (l c_2^2 + W_1 l c_2 \cos(q_2)) + \\ &+ m_3 (W_2^2 + l c_3^2 + W_1 W_2 \cos(q_2) + 2W_2 l c_3 \cos(q_3) + W_1 l c_3 \cos(q_2 + q_3)), \\ M_{13}(q) &= I_3 + m_3 (l c_3^2 + W_2 l c_3 \cos(q_3) + W_1 l c_3 \cos(q_2 + q_3)), \\ M_{21}(q) &= M_{12}(q), \\ M_{22}(q) &= I_2 + I_3 + m_2 l c_2^2 + m_3 (W_2^2 + l c_3^2 + 2W_2 l c_3 \cos(q_3)), \\ M_{23}(q) &= I_3 + m_3 (l c_3^2 + W_2 l c_3 \cos(q_3)), \\ M_{31}(q) &= M_{13}(q), \\ M_{32}(q) &= M_{23}(q), \\ M_{33}(q) &= I_3 + m_3 l c_3^2. \end{aligned}$$

O vetor de torques não-inerciais é dado por

$$h(q, \dot{q}) = \begin{bmatrix} h_1(q, \dot{q}) \\ h_2(q, \dot{q}) \\ h_3(q, \dot{q}) \end{bmatrix}$$

$$\begin{aligned} h_1(q, \dot{q}) &= 2(t_1 + t_2 + t_3)\dot{q}_1\dot{q}_2 + 2(t_3 + t_4)\dot{q}_1\dot{q}_3 + 2(t_1 + t_2 + t_3)\dot{q}_2\dot{q}_2 + \\ &\quad + 2(t_3 + t_4)\dot{q}_2\dot{q}_3 + (t_3 + t_4)\dot{q}_3\dot{q}_3, \\ h_2(q, \dot{q}) &= -(t_1 + t_2 + t_3)\dot{q}_1\dot{q}_1 + 2t_4\dot{q}_1\dot{q}_3 + 2t_4\dot{q}_2\dot{q}_3 + t_4\dot{q}_3\dot{q}_3, \\ h_3(q, \dot{q}) &= -(t_3 + t_4)\dot{q}_1\dot{q}_1 - 2t_4\dot{q}_1\dot{q}_2 - t_4\dot{q}_2\dot{q}_2. \end{aligned}$$

Escrevendo $h(q, \dot{q}) = C(q, \dot{q})\dot{q}$ tem-se:

$$C(q, \dot{q}) = \begin{bmatrix} C_{11}(q, \dot{q}) & C_{12}(q, \dot{q}) & C_{13}(q, \dot{q}) \\ C_{21}(q, \dot{q}) & C_{22}(q, \dot{q}) & C_{23}(q, \dot{q}) \\ C_{31}(q, \dot{q}) & C_{32}(q, \dot{q}) & C_{33}(q, \dot{q}) \end{bmatrix}$$

$$\begin{aligned} C_{11}(q, \dot{q}) &= 2(t_1 + t_2 + t_3)\dot{q}_2 + 2(t_3 + t_4)\dot{q}_3, \\ C_{12}(q, \dot{q}) &= (t_1 + t_2 + t_3)\dot{q}_2 + 2(t_3 + t_4)\dot{q}_3, \\ C_{13}(q, \dot{q}) &= (t_3 + t_4)\dot{q}_3, \\ C_{21}(q, \dot{q}) &= -(t_1 + t_2 + t_3)\dot{q}_1 + 2t_4\dot{q}_3, \\ C_{22}(q, \dot{q}) &= t_4\dot{q}_3, \\ C_{23}(q, \dot{q}) &= t_4(\dot{q}_2 + \dot{q}_3), \\ C_{31}(q, \dot{q}) &= -(t_3 + t_4)\dot{q}_1 - 2t_4\dot{q}_2, \\ C_{32}(q, \dot{q}) &= -t_4\dot{q}_2, \\ C_{33}(q, \dot{q}) &= 0. \end{aligned}$$

O Jacobiano é dado por

$$J(q) = \begin{bmatrix} J_{11}(q) & J_{12}(q) & J_{13}(q) \\ J_{21}(q) & J_{22}(q) & J_{23}(q) \\ J_{31}(q) & J_{32}(q) & J_{33}(q) \end{bmatrix}$$

$$J_{11}(q) = J_{12}(q) = J_{13}(q) = 1,$$

$$J_{21}(q) = -W_1 \sin(q_1) - W_2 \sin(q_1 + q_2) - W_3 \sin(q_1 + q_2 + q_3),$$

$$J_{22}(q) = -W_2 \sin(q_1 + q_2) - W_3 \sin(q_1 + q_2 + q_3),$$

$$J_{23}(q) = -W_3 \sin(q_1 + q_2 + q_3),$$

$$J_{31}(q) = W_1 \cos(q_1) + W_2 \cos(q_1 + q_2) + W_3 \cos(q_1 + q_2 + q_3),$$

$$J_{32}(q) = W_2 \cos(q_1 + q_2) + W_3 \cos(q_1 + q_2 + q_3),$$

$$J_{33}(q) = W_3 \cos(q_1 + q_2 + q_3).$$