



INVERSA FILTRADA: UMA SOLUÇÃO ALTERNATIVA PARA A CINEMÁTICA INVERSA DE MANIPULADORES ROBÓTICOS

Lucas Vares Vargas

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

Orientadores: Ramon Romankevicius Costa
Antonio Candea Leite

Rio de Janeiro
Março de 2013

INVERSA FILTRADA: UMA SOLUÇÃO ALTERNATIVA PARA A
CINEMÁTICA INVERSA DE MANIPULADORES ROBÓTICOS

Lucas Vares Vargas

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

Examinada por:

Prof. Ramon Romankevicius Costa, D.Sc.

Prof. Antonio Candea Leite, D.Sc.

Prof. Amit Bhaya, Ph.D.

Prof. Marco Antonio Meggiolaro, Ph.D.

RIO DE JANEIRO, RJ – BRASIL

MARÇO DE 2013

Vargas, Lucas Vares

Inversa filtrada: uma solução alternativa para a cinemática inversa de manipuladores robóticos/Lucas Vares Vargas. – Rio de Janeiro: UFRJ/COPPE, 2013.

XIV, 94 p.: il.; 29, 7cm.

Orientadores: Ramon Romankevicius Costa

Antonio Candea Leite

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

Referências bibliográficas: p. 76 – 81.

1. Cinemática inversa. 2. Singularidades. 3. Manipuladores robóticos. 4. Inversa filtrada. I. Costa, Ramon Romankevicius *et al.* II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia Elétrica. III. Título.

*À minha mãe Maria, ao meu pai
Edson e às minhas irmãs
Juliana e Gabriela, com muita
gratidão e carinho.*

Agradecimentos

A Deus, sempre presente em minha vida. Agradeço a Ele por me dar forças nesta jornada, por tudo que pude concretizar e, principalmente, pelas pessoas maravilhosas que tenho ao meu lado.

Aos meus orientadores, os professores Ramon Romankevicius Costa e Antonio Candea Leite, pela confiança, incentivo, disponibilidade e apoio que permitiram o desenvolvimento deste trabalho.

Aos professores Afonso Celso del Nero Gomes, Amit Bhaya, Fernando Lizarralde, João Carlos dos Santos Basílio, José Luís Lopes da Silveira e Liu Hsu, pelos valiosos ensinamentos, essenciais para a pesquisa realizada.

Aos colegas Alessandro Jacoud, Alex Neves, Andrei Battistel, Eduardo Leão, Evandro Costa, Fábio Soares, Gabriel Casulari, Gustavo Freitas, Ighor Marcovistz, Ivanko Yanque, Maurício Silva, Nerito Oliveira, Rafael Pereira, Roberto Calvet, Rodrigo Carneiro, Thiago Braga e Wanderley de Almeida, pelas conversas e por proporcionar um excelente ambiente de trabalho.

Aos amigos Alex Neves, Andrei Battistel e Ivanko Yanque, pela amizade, prestatividade e inestimáveis contribuições.

Aos amigos Anderson Sangreman, André Barbosa, Aurélio Lima, Daniel Vaz, Diogo Nomiya, Gabriel Gleizer, Gabriel Kyosuke, Henrique Ferraz, Leonardo Orenstein, Marcelo Prado, Maurício Silva, Raphael Cota, Thiago Camanho e Trevor Dobbin, pela amizade e companheirismo.

Aos membros da banca, os professores Amit Bhaya e Marco Antonio Meggiolaro, pelas valiosas sugestões e comentários que possibilitaram o aprimoramento da pesquisa desenvolvida.

À COPPE pela oportunidade de realizar o mestrado em uma instituição de pesquisa e ensino de excelência.

Ao CNPq e à CAPES, pelo suporte financeiro.

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

INVERSA FILTRADA: UMA SOLUÇÃO ALTERNATIVA PARA A CINEMÁTICA INVERSA DE MANIPULADORES ROBÓTICOS

Lucas Vares Vargas

Março/2013

Orientadores: Ramon Romankevicius Costa
Antonio Candea Leite

Programa: Engenharia Elétrica

Este trabalho apresenta uma solução alternativa para o problema de cinemática inversa de robôs manipuladores baseado na cinemática diferencial e em um algoritmo que calcula a inversa da matriz Jacobiana dinamicamente. A saída do algoritmo proposto pode ser interpretada como uma inversa filtrada da matriz Jacobiana. Uma propriedade interessante do algoritmo de inversão é a sua habilidade para lidar com o problema de singularidades cinemáticas, permitindo também a ponderação ou priorização dos diferentes objetivos de controle e a consideração de restrições adicionais. A análise de estabilidade e convergência é baseada na teoria de estabilidade de Lyapunov. Estudos de caso para diversos manipuladores são apresentados e resultados de simulação ilustram o desempenho da metodologia proposta.

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

FILTERED INVERSE: AN ALTERNATIVE SOLUTION TO INVERSE
KINEMATICS OF ROBOTIC MANIPULATORS

Lucas Vares Vargas

March/2013

Advisors: Ramon Romankevicius Costa
Antonio Candeia Leite

Department: Electrical Engineering

In this work we present an alternative solution to the problem of inverse kinematics for robot manipulators based on differential kinematics and on an algorithm that calculates the inverse of the Jacobian matrix dynamically. The output of the proposed algorithm can be interpreted as a filtered inverse of the Jacobian matrix. An interesting property of the inversion algorithm is its ability to deal with the problem of kinematic singularities, also allowing the prioritization or weighting of the different control objectives and the consideration of additional constraints. The convergence and stability analysis is based on the Lyapunov stability theory. Case studies for various manipulators are presented and simulation results illustrate the performance of the proposed methodology.

Sumário

Lista de figuras	xi
1 Introdução	1
1.1 Motivação	2
1.2 Revisão bibliográfica	3
1.3 Objetivo	5
1.4 Metodologia	5
1.5 Estrutura do trabalho	6
2 Cinemática	7
2.1 Posição e orientação de um corpo rígido	7
2.2 Cinemática direta	10
2.2.1 Cadeia cinemática aberta	12
2.2.2 Espaço das juntas e espaço operacional	13
2.3 Cinemática diferencial	14
2.3.1 Jacobiano geométrico	14
2.3.2 Jacobiano analítico	15
2.3.3 Jacobiano de representação	16
2.3.4 Singularidades cinemáticas	16
2.4 Cinemática inversa	16
2.4.1 Abordagem diferencial	17
2.4.2 Manipuladores não-redundantes	17
2.4.3 Erro de posição	19
2.4.4 Erro de orientação	19
2.4.5 Manipuladores redundantes	22
3 Singularidades e métodos alternativos	24
3.1 SVD e manipulabilidade	24
3.2 Singularidades em robôs antropomórficos	26
3.3 Algoritmo DLS	29
3.4 Algoritmo FIK	30

4	Inversa filtrada	37
4.1	Algoritmo proposto	37
4.2	Propriedades da inversa filtrada	39
4.3	Inversa filtrada de matrizes	42
4.3.1	Matriz de erro à direita	43
4.3.2	Matriz de erro à esquerda	43
4.3.3	Lei de atualização composta	44
4.4	Análise de convergência	46
4.5	Solução para cinemática inversa	50
4.6	Lei de controle modificada	62
4.7	Função objetivo e Jacobiano ampliado	68
4.7.1	Função f para limite de juntas	69
4.7.2	Função f para desvio de obstáculo	69
5	Conclusões e trabalhos futuros	74
5.1	Conclusões	74
5.2	Propostas de trabalhos futuros	75
	Referências bibliográficas	76
A	Conceitos básicos	82
A.1	Propriedades da função traço	82
A.2	Subespaços fundamentais	82
A.3	Decomposição de valor singular (SVD)	83
A.4	Produto de Kronecker	84
A.5	Operador vec	84
A.6	Equação de Sylvester	85
B	Zebra-ZERO	86
B.1	Cinemática direta	87
B.2	Modelo 3D	88
C	Trajетórias	89
C.1	Trajетória 1	89
C.2	Trajетória 2	90
C.3	Trajетória 3	90
C.4	Trajетória 4	91
C.5	Trajетória 5	91
C.6	Trajетória 6	92
C.7	Trajетória 7	92
C.8	Trajетória 8	93

C.9 Trajetória 9	93
C.10 Trajetória 10	94
C.11 Trajetória 11	94

Lista de figuras

1.1	(a) LUMA (COPPE/UFRJ), veículo operado remotamente para exploração submarina; e (b) Curiosity (NASA), utilizado para coleta de informações na superfície de Marte (foto retirada de <i>www.nasa.gov</i>).	2
2.1	Posição e orientação do corpo rígido.	8
2.2	Manipulador SCARA.	10
2.3	Descrição da orientação e posição do sistema de coordenadas do efetuador.	11
2.4	Transformação de coordenadas em uma cadeia cinemática aberta.	12
2.5	Algoritmo de cinemática inversa (caso não-redundante).	18
2.6	Robô planar não-redundante com 3 juntas de revolução.	21
2.7	Simulação 1: (a) norma do erro para diferentes ganhos λ ; (b) velocidades das juntas para $\lambda = 2$; e (c) ângulos das juntas para $\lambda = 2$	21
2.8	Algoritmo de cinemática inversa (caso redundante).	22
2.9	Robô planar redundante com 3 juntas de revolução.	23
2.10	Simulação 2: (a) norma do erro para diferentes ganhos λ ; (b) velocidades das juntas para $\lambda = 2$; e (c) ângulos das juntas para $\lambda = 2$	23
3.1	Elipsoides de manipulabilidade de velocidade para robô planar.	26
3.2	Manipulador antropomórfico com punho esférico.	26
3.3	Configurações singulares: (a)cotovelo, (b)ombro e (c)punho.	27
3.4	Simulação 3: (a) norma do erro; (b) condicionamento; (c) $\dot{\mathbf{q}}_p$ e (d) $\dot{\mathbf{q}}_o$	28
3.5	Simulação 3: Manipulador Zebra-ZERO em diferentes posturas: (a) afastada ($t \approx 1s$) e (b) próxima de uma singularidade ($t \approx 9s$).	28
3.6	Diagrama de blocos FIK.	30
3.7	Diagrama de blocos FIK com restrição adicional.	32
3.8	Simulação 4: (a) trajetórias; (b) norma do erro; e (c) condicionamento.	33
3.9	Simulação 4: variáveis (a) $\dot{\mathbf{q}}_{DLS}$; (b) \mathbf{q}_{DLS} ; (c) $\dot{\mathbf{q}}_{FIK}$; e (d) \mathbf{q}_{FIK}	33
3.10	Simulação 5: (a) trajetórias; (b) norma do erro; e (c) condicionamento.	34
3.11	Simulação 5: variáveis (a) $\dot{\mathbf{q}}_{DLS}$; (b) \mathbf{q}_{DLS} ; (c) $\dot{\mathbf{q}}_{FIK}$; e (d) \mathbf{q}_{FIK}	34
3.12	Simulação 6: (a) trajetórias; (b) norma do erro; e (c) condicionamento.	35

3.13	Simulação 6: variáveis (a) $\dot{\mathbf{q}}_{DLS}$; (b) \mathbf{q}_{DLS} ; (c) $\dot{\mathbf{q}}_{FIK}$; e (d) \mathbf{q}_{FIK}	35
3.14	Simulação 7: (a) trajetórias; (b) norma do erro; e (c) condicionamento.	36
3.15	Simulação 7: variáveis (a) $\dot{\mathbf{q}}_{DLS}$; (b) \mathbf{q}_{DLS} ; (c) $\dot{\mathbf{q}}_{FIK}$; e (d) \mathbf{q}_{FIK}	36
4.1	Plano $\theta \times k$ (diferentes condições iniciais).	39
4.2	Plano $\theta \times k$ (escape).	39
4.3	Simulação 8: evolução de (a) θ e (b) S para diferentes ganhos k	41
4.4	Simulação 9: variáveis (a) k e θ ; (b) S ; e (c) plano $\theta \times k$	41
4.5	Simulação 9: variáveis (a) y e (b) u para diferentes ganhos β	42
4.6	Plano $d_i \times \sigma_i$ (diferentes condições iniciais).	48
4.7	Diagrama de blocos IF.	51
4.8	Simulação 10: manipulador Zebra-ZERO e trajetória desejada em perspectiva e visão superior.	52
4.9	Simulação 10: trajetórias no espaço cartesiano.	52
4.10	Simulação 10: (a) norma do erro e (b) manipulabilidade.	53
4.11	Simulação 10: variáveis (a) $\dot{\mathbf{q}}_{IF}$; (b) $\dot{\mathbf{q}}_{DLS}$; (c) \mathbf{q}_{IF} ; e (d) \mathbf{q}_{DLS}	53
4.12	Simulação 11: manipulador Zebra-ZERO e trajetória desejada em perspectiva e visão frontal.	54
4.13	Simulação 11: trajetórias no espaço cartesiano.	54
4.14	Simulação 11: trajetórias no plano yz	55
4.15	Simulação 11: (a) norma do erro; (b) norma do erro (escala ampliada); e (c) manipulabilidade.	55
4.16	Simulação 11: variáveis (a) $\dot{\mathbf{q}}_{IF}$; (b) $\dot{\mathbf{q}}_{DLS}$; (c) \mathbf{q}_{IF} ; e (d) \mathbf{q}_{DLS}	56
4.17	Simulação 11: (a) função V_c e (b) δ	56
4.18	Simulação 11b: trajetórias no espaço cartesiano.	57
4.19	Simulação 11b: (a) norma do erro; (b) manipulabilidade; e (c) δ	57
4.20	Simulação 12: manipulador Zebra-ZERO e trajetória desejada em perspectiva e visão superior.	58
4.21	Simulação 12: trajetórias no espaço cartesiano.	58
4.22	Simulação 12: trajetórias no plano xy	59
4.23	Simulação 12: (a) norma do erro; (b) norma do erro (escala ampliada); e (c) manipulabilidade.	59
4.24	Simulação 12: variáveis (a) $\dot{\mathbf{q}}_{IF}$; (b) $\dot{\mathbf{q}}_{DLS}$; (c) \mathbf{q}_{IF} ; e (d) \mathbf{q}_{DLS}	60
4.25	Simulação 12: (a) função V_c e (b) δ	60
4.26	Simulação 12: trecho não-factível e diferentes ganhos γ	61
4.27	Simulação 12: variáveis (a) $\dot{\mathbf{q}}_{IF}$ e (b) \mathbf{q}_{IF} para $\gamma = 25$	61
4.28	Planos $\theta \times k$ e $\theta_M \times k$ (curvas típicas).	62
4.29	Simulação 13: trajetórias no plano xy	63
4.30	Simulação 13: trechos de transição no plano xy	64

4.31	Simulação 13: variáveis (a) q_1 ; (b) q_2 ; e (c) q_3 ($\gamma = 1$).	64
4.32	Simulação 13: $\dot{\mathbf{q}}$ em (a) escala normal e (b) ampliada ($5 < t < 10$).	65
4.33	Simulação 13: (a) manipulabilidade e (b) função V_c ($6 < t < 10$).	65
4.34	Simulação 14: (a) $\lambda_p > \lambda_o$; (b) $\lambda_p = \lambda_o$; e (c) $\lambda_p < \lambda_o$.	66
4.35	Simulação 14: trajetórias com prioridade para (a) posição e (b) orientação.	67
4.36	Simulação 14: diferentes pesos para erros em \mathbf{x} e \mathbf{z} .	67
4.37	Função objetivo para limite de juntas.	69
4.38	Função objetivo para desvio de obstáculo.	69
4.39	Simulação 15: normas dos erros e ângulos das juntas.	70
4.40	Simulação 15: norma do erro e ângulos das juntas para $\zeta_2 = 0.2 rad$.	70
4.41	Simulação 16: (a) trajetórias com e sem desvio; (b) dimensão ampliada (função f); e (c) desvio para diferentes ganhos α_f .	71
4.42	Obstáculo operacional e reconfiguração interna.	72
4.43	Ambiente industrial: manipuladores cooperativos.	73
4.44	Manipulador planar 3R e função f correspondente.	73
B.1	Manipulador Zebra-ZERO.	86
B.2	Modelo 3D para Simulink.	88
C.1	Trajetoária 1.	89
C.2	Trajetoária 2.	90
C.3	Trajetoária 3.	90
C.4	Trajetoária 4.	91
C.5	Trajetoária 5.	91
C.6	Trajetoária 6.	92
C.7	Trajetoária 7.	92
C.8	Trajetoária 8.	93
C.9	Trajetoária 9.	93
C.10	Trajetoária 10.	94
C.11	Trajetoária 11.	94

Abreviaturas

SISO:	Uma entrada e uma saída - escalar <i>(Single-input-single-output)</i>
MIMO:	Múltiplas entradas e múltiplas saídas - multivariável <i>(Multiple-input-multiple-output)</i>
SVD:	Decomposição de valor singular <i>(Singular value decomposition)</i>
DLS:	Mínimos quadrados amortecidos <i>(Damped least square)</i>
FIK:	Cinemática inversa realimentada <i>(Feedback inverse kinematics)</i>
IF:	Inversa filtrada <i>(Filtered inverse)</i>
PDS	Positiva definida simétrica <i>(Positive definite symmetric)</i>
nR:	n juntas de revolução <i>(n revolute joints)</i>
DoF:	Graus de liberdade <i>(Degrees of freedom)</i>
DH:	Denavit-Hartenberg

Capítulo 1

Introdução

A robótica teve seu desenvolvimento impulsionado pela busca de maior produtividade e qualidade de células industriais. A utilização de robôs permitiu uma maior eficiência principalmente em tarefas repetitivas, como soldagem ou movimentação de peças, e se mostrou uma alternativa para o trabalho humano em ambientes hostis e não estruturados. O estudo de sistemas robóticos evoluiu rapidamente e a aplicação destes sistemas não se limitou apenas às linhas de produção industriais. Neste contexto, algumas das inúmeras aplicações da robótica são:

- Processos industriais - a utilização de sistemas robóticos na linha de produção possibilitou a maior eficiência da célula industrial. Tarefas repetitivas, cansativas e/ou perigosas não são mais efetuadas por seres humanos [1];
- Robótica em áreas hostis - a robótica surge como alternativa para a realização de tarefas em ambientes onde a presença humana é inviável ou arriscada. Como exemplo, a robótica submarina aplicada à exploração de áreas profundas (Figura 1.1.a), que visa desde a obtenção de informações sobre a bio-diversidade de ambientes à coleta de recursos naturais e prospecção oceânica [2];
- Robótica espacial - sistemas robóticos são utilizados para coleta de informações na superfície de planetas (Figura 1.1.b), como os conhecidos *Mars Rovers* (2003) e *Curiosity* (2012), ou realizando serviços de manutenção em órbita [3];
- Medicina - a utilização de robôs em operações cirúrgicas a distância (telecirurgias) ou que exijam grande precisão, proporcionando destreza e conforto durante o tempo de operação além da regulação de sensibilidade e filtragem contra perturbações externas (e.g., um natural tremor das mãos) [4];
- Segurança - robôs são utilizados para localizar e desativar dispositivos explosivos em campos minados [5], reduzindo o risco à vida humana durante a operação.

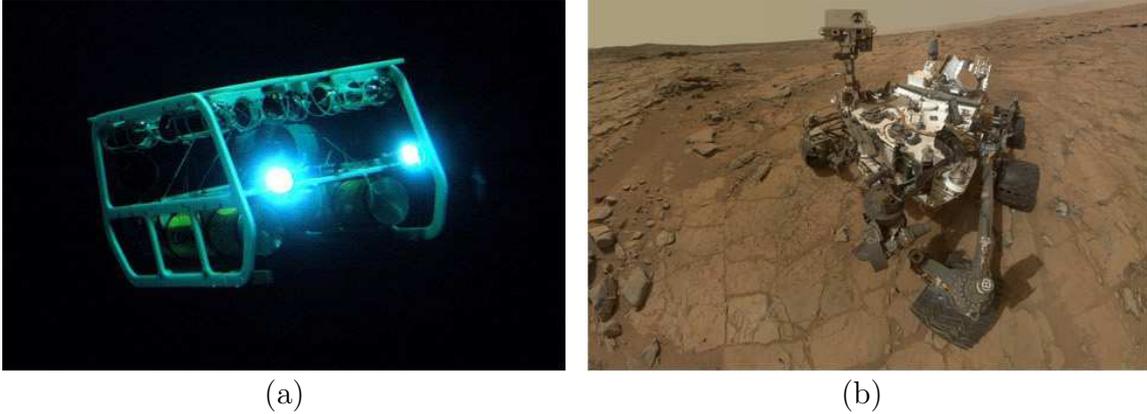


Figura 1.1: (a) LUMA (COPPE/UFRJ), veículo operado remotamente para exploração submarina; e (b) Curiosity (NASA), utilizado para coleta de informações na superfície de Marte (foto retirada de www.nasa.gov).

Nas últimas décadas, um esforço de pesquisa considerável tem sido realizado para identificar uma solução genérica e robusta para o problema de cinemática inversa, usualmente encontrado em sistemas de controle de manipuladores robóticos [6–8]. O problema de cinemática inversa de robôs manipuladores consiste em determinar as variáveis das juntas associadas a uma determinada posição e orientação do efetuador. Este problema constitui um tópico de pesquisa de grande relevância não apenas em robótica, mas em áreas como computação gráfica, na animação 3D de modelos virtuais [9–11], e biologia, em processos de dobramento de proteínas [12–15]. De forma geral, o problema consiste em, dada uma função $\mathbf{f} : \mathbb{R}^n \rightarrow \mathbb{R}^m$ e um valor desejado $\mathbf{y}_d \in \mathbb{R}^m$, resolver

$$\mathbf{y}_d = \mathbf{f}(\mathbf{x}) \quad (1.1)$$

para algum $\mathbf{x} \in \mathbb{R}^n$. A solução deste problema não é trivial uma vez que: nem sempre há solução; pode não haver solução na forma fechada; e podem existir infinitas soluções [16]. Para manipuladores robóticos, a dificuldade para solucionar o problema de cinemática inversa surge também quando estes se aproximam de configurações específicas, chamadas *singularidades cinemáticas*. Nestas configurações há mudança nos graus de liberdade da estrutura mecânica e sua mobilidade é reduzida [17].

1.1 Motivação

A motivação deste trabalho vem da dificuldade em se encontrar soluções viáveis para o problema de cinemática inversa de manipuladores em configurações singulares ou até mesmo na vizinhança destas. A existência de singularidades cinemáticas pode gerar instabilidade, como é o caso de soluções obtidas através da inversa ou pseudo-inversa da matriz Jacobiana, e já na vizinhança destas configurações, os métodos de solução podem acarretar grandes velocidades a nível de juntas.

1.2 Revisão bibliográfica

A solução do problema de cinemática inversa é de importância fundamental na área de robótica pois permite transformar as especificações de movimento, naturalmente prescritas ao efetuador no espaço operacional, em correspondentes movimentos no espaço das juntas, permitindo a execução bem sucedida do movimento desejado [16]. Em geral, o desempenho das soluções apresentadas é avaliado de acordo com a estabilidade, custo computacional e robustez às singularidades [18, 19].

Singularidades cinemáticas são as configurações do espaço das juntas onde a matriz Jacobiana possui posto incompleto e a sua determinação é fundamental na área de robótica, pois em uma configuração singular pequenas velocidades no espaço operacional podem causar grandes velocidades no espaço das juntas e, consequentemente, a saturação dos atuadores [20, 21]. Além disso, na vizinhança de uma singularidade, a mobilidade da estrutura mecânica é reduzida e podem existir infinitas soluções para o problema de cinemática inversa [17]. Uma análise mais rigorosa do comportamento da solução na vizinhança de uma configuração singular pode ser desenvolvida recorrendo-se à decomposição em valores singulares da matriz Jacobiana ou SVD (no inglês, *singular value decomposition*) [22]. Para lidar com o problema de singularidades em manipuladores redundantes, uma abordagem alternativa consiste em utilizar os graus de mobilidade extras da estrutura cinemática para evitar alguns tipos de singularidades [23–25], assim como atender a restrições adicionais [26, 27]. Uma abordagem interessante consiste em formular o problema de controle como um problema de otimização linear com restrições [16].

O problema de cinemática inversa apresenta soluções finitas na forma fechada apenas para manipuladores que possuem estrutura cinemática simples, por exemplo, os manipuladores de cadeia aberta formados tipicamente por um braço serial e um punho esférico [28]. A maioria das técnicas propostas são baseadas em métodos numéricos ou técnicas de otimização [22, 29]. Em [30], uma combinação de métodos analíticos e numéricos é utilizada para solucionar a cinemática inversa de robôs antropomórficos, incluindo os problemas de posição e orientação, assim como restrições adicionais. O método de sub-problemas de Paden-Kahan, apresentado originalmente em [31], formalizado em [32] e apresentado didaticamente em [33], considera identidades geométricas e permite tratar o problema através da resolução de subproblemas frequentes nas soluções de cinemática inversa de robôs manipuladores.

A dificuldade para obtenção de soluções admissíveis surge sempre que (i) o efetuador atinge uma posição ou orientação particular no espaço operacional onde, por exemplo, duas ou mais juntas de revolução são colineares, (ii) a estrutura cinemática do manipulador é complexa e não é possível relacionar a postura do efetuador com diferentes configurações de variáveis das juntas; (iii) o manipulador é cinemática-

mente redundante. Essas limitações são, em geral, devidas à relação não-linear entre as variáveis no espaço das juntas e as variáveis no espaço operacional [34]. Por outro lado, a equação de cinemática diferencial representa um mapeamento linear entre a velocidade no espaço das juntas e a velocidade no espaço operacional [35], embora esta relação, representada pela matriz Jacobiana [36], dependa da configuração atual dos ângulos das juntas. Este fato sugere que é possível utilizar a abordagem diferencial para resolver o problema de cinemática inversa [37]. Esta proposta é tipicamente citada na literatura como algoritmo iterativo.

A obtenção das variáveis de juntas a partir de um esquema em malha aberta baseado em técnicas de integração numérica está sujeita ao fenômeno de *drift* da solução e, como consequência, a postura do efetuador correspondente às variáveis das juntas computadas pode ser diferente da postura desejada. Uma solução alternativa consiste em utilizar esquemas em malha-fechada, baseados na pseudo-inversa e na transposta da matriz Jacobiana, que consideram o erro no espaço operacional entre as posturas desejada e atual do efetuador. Os métodos que utilizam a pseudo-inversa do Jacobiano podem falhar na busca de soluções admissíveis na vizinhança de uma singularidade, aumentando o erro de regime permanente e/ou produzindo elevadas velocidades das juntas [38]. Por outro lado, os métodos baseados no Jacobiano transposto são numericamente eficientes na presença de singularidades, mas não garantem a estabilidade assintótica do erro para tarefas de rastreamento [16].

Outros métodos para solucionar o problema de cinemática inversa recorrem a diferentes técnicas de otimização. Em [29], é proposta a combinação de duas técnicas de programação não-linear apresentadas em [39]. O algoritmo proposto utiliza o método CCD (no inglês, *cyclic coordinate descent*) para encontrar rapidamente um ponto próximo à solução e então utiliza o método de métrica variável BFS (Broyden-Fletcher-Shanno) para obter a solução com o desejado grau de precisão. O algoritmo é computacionalmente eficiente, robusto às singularidades e considera as limitações angulares das juntas, combinando as vantagens das duas técnicas consideradas. A aplicação do método é naturalmente estendida para manipuladores com cadeia fechada através de uma simples modificação.

O algoritmo de cinemática inversa amortecido ou DLS (no inglês, *damped least square*), baseado na pseudo-inversa da matriz Jacobiana, corresponde à abordagem mais tratada na literatura e emprega um fator de amortecimento para evitar as singularidades, estabelecendo um compromisso entre a exatidão e a viabilidade da solução [40, 41]. A escolha de valores adequados para o fator de amortecimento se torna essencial e, desta forma, a identificação de diferentes métodos de adaptação deste fator constitui um tema de pesquisa bastante relevante [42, 43]. A utilização do termo de amortecimento garante uma matriz bem condicionada para inversão [44] e, em [45], o método é utilizado com a inclusão de ponderações dinâmicas referentes

às diferentes variáveis de juntas. Em [46], é proposto um algoritmo adaptativo não-linear baseado no método da pseudo-inversa amortecida.

O método de cinemática inversa realimentado ou FIK (no inglês, *feedback inverse kinematics*) é o método mais recente na literatura para lidar com a presença de singularidades cinemáticas. Proposto inicialmente em [47], onde seu desempenho é comparado com o método DLS para um manipulador planar, e aplicado para o problema de posição de um manipulador 6DoF em [48], o método não requer o cálculo de um fator de amortecimento ou manipulações matriciais, como inversão ou decomposição em valores singulares, empregando um laço de realimentação para minimizar a discrepância entre as velocidades atual e demandada no espaço operacional. O método é robusto à presença de singularidades e computacionalmente eficiente, com grande redução da demanda computacional quando comparado aos métodos baseados na pseudo-inversa.

1.3 Objetivo

Neste trabalho, é proposto e estudado um método alternativo para solucionar o problema de cinemática inversa de manipuladores robóticos seriais, capaz de contornar a presença de singularidades cinemáticas nas trajetórias de referência. A estabilidade e a convergência do algoritmo devem ser analisadas e garantidas. A solução utilizará uma abordagem diferencial, tendo como base o mapeamento linear realizado pela matriz Jacobiana entre as velocidades no espaço das juntas e no espaço de operação, e o desempenho do método para diversos robôs manipuladores e condições iniciais deve ser ilustrado através de simulações.

1.4 Metodologia

O desenvolvimento tem início com a simples análise do movimento de partículas e pela decorrente análise do movimento coletivo destas, como em um corpo rígido. A descrição da posição e orientação de um corpo rígido é de grande importância para o entendimento do trabalho desenvolvido, constituindo um requisito básico e sendo apresentada com detalhes em [33]. A extensão para uma cadeia de corpos rígidos é natural, sendo obtidas as chamadas equações de cinemática direta, que relacionam o espaço das juntas e o espaço operacional, tendo como base a convenção de Denavit-Hartenberg [49]. A cinemática diferencial, por sua vez, relaciona as velocidades linear e angular do efetuador com as velocidades das juntas, sendo essencial para o desenvolvimento de uma solução iterativa para o problema inverso de determinação das variáveis das juntas a partir da configuração da estrutura no espaço operacional, chamado de cinemática inversa. Em [16], um capítulo é dedi-

cado ao estudo deste assunto e à obtenção da matriz de mapeamento, o Jacobiano. É apresentada uma análise do comportamento da solução para o problema de cinemática inversa na proximidade de configurações singulares, sendo abordado o caso específico de manipuladores antropomórficos com punho esférico. São apresentados dois métodos existentes na literatura para análise e comparação. O algoritmo DLS, bastante consolidado, utiliza um termo de amortecimento que torna a matriz Jacobi-ana melhor condicionada para inversão. O algoritmo FIK, por sua vez, corresponde à alternativa mais recente para lidar com singularidades cinemáticas, propondo um laço de realimentação e não exigindo a inversão matricial. Por fim, para solucionar o problema de singularidades cinemáticas de manipuladores robóticos, apresenta-se o método proposto. Inicialmente, apresenta-se sua ideia principal, aplicada ao caso escalar e, em seguida, sua generalização para o caso multivariável, utilizada na solução do problema de cinemática inversa através de uma abordagem diferencial. A comparação entre os casos escalar e multivariável é realizada, assim como a análise de convergência e estabilidade do método, baseada na teoria de Lyapunov [50]. Publicações envolvendo métodos de controle para manipuladores constituíram importante base para o trabalho. A utilização de produções acadêmicas também foi de enorme importância para o desenvolvimento deste trabalho, como [51].

1.5 Estrutura do trabalho

O trabalho desenvolvido é apresentado da seguinte forma:

- Capítulo 2 - são apresentados os fundamentos teóricos básicos e, portanto, essenciais para o trabalho. Neste capítulo são abordadas as equações de cinemática de robôs manipuladores.
- Capítulo 3 - é apresentada uma análise do comportamento da solução para a cinemática inversa na proximidade de configurações singulares e são apresentados dois métodos consolidados na literatura para análise e comparação.
- Capítulo 4 - é apresentada a ideia principal do algoritmo proposto. Inicialmente, trata-se o caso escalar e, em seguida, estende-se a análise para o caso de interesse, multivariável.
- Capítulo 5 - são apresentadas as considerações finais sobre o trabalho desenvolvido e o método proposto. São discutidas, também, propostas de trabalhos futuros e que possam utilizar os tópicos abordados neste documento.

Os resultados obtidos a partir de simulações são apresentados ao longo do trabalho, acompanhando o desenvolvimento teórico. Para uma melhor leitura, alguns conceitos básicos e dados adicionais são apresentados nos apêndices.

Capítulo 2

Cinemática

Um manipulador robótico pode ser esquematicamente representado sob o ponto de vista mecânico como uma cadeia de corpos rígidos, os elos, conectados por meio de juntas, prismáticas ou de revolução. No início desta cadeia é estabelecida a origem, tipicamente a referência inercial escolhida para o sistema e, ao seu final, é montado o efetuador. O movimento resultante de toda a estrutura é obtido pela composição dos movimentos elementares de cada elo com respeito ao elo anterior. De forma a possibilitar a manipulação de um objeto no espaço de trabalho pelo efetuador, torna-se necessária a descrição de sua posição e orientação. Este capítulo, elaborado a partir de [16] e [33], dedica-se à apresentação das bases teóricas para a obtenção da equação que determina a posição e a orientação do efetuador a partir das variáveis das juntas, denominada equação de cinemática direta, e da equação que relaciona as velocidades desenvolvidas no espaço das juntas e no espaço operacional, denominada equação de cinemática diferencial. O estudo da cinemática de manipuladores robóticos tem início, portanto, com o estudo do movimento do corpo rígido.

2.1 Posição e orientação de um corpo rígido

O movimento de uma partícula no espaço Euclidiano \mathbb{R}^3 é descrito por sua localização relativa a um sistema de coordenadas Cartesiano inercial $\mathbf{O}-xyz$ definido por três eixos ortonormais \mathbf{x} , \mathbf{y} e \mathbf{z} . A posição da partícula é especificada pela terna $(p_x, p_y, p_z) \in \mathbb{R}^3$, em que cada coordenada representa a projeção da posição \mathbf{p} nos eixos do sistema de coordenadas inercial. Assim, a trajetória efetuada pela partícula pode ser representada pela curva $\mathbf{p}(t) = (p_x(t), p_y(t), p_z(t)) \in \mathbb{R}^3$. O interesse da robótica consiste na análise do movimento coletivo destas partículas, como em um elo de um manipulador. Assim, define-se formalmente um corpo rígido como um conjunto de partículas tal que a distância entre elas permanece fixa para todo tempo, independente do movimento realizado e das forças exercidas sobre o corpo. De maneira menos formal, o corpo rígido nada mais é do que um corpo imune a

distorções. Os pontos \mathbf{p} e \mathbf{q} de um corpo rígido satisfazem

$$\|\mathbf{p}(t) - \mathbf{q}(t)\| = \|\mathbf{p}(0) - \mathbf{q}(0)\| = C \quad \forall t. \quad (2.1)$$

O corpo rígido é completamente descrito no espaço por sua posição e orientação relativas ao sistema de coordenadas inercial. A Figura 2.1 mostra o sistema de coordenadas de referência (ou inercial) e um corpo rígido, onde quaisquer dois pontos a ele pertencentes satisfazem (2.1).

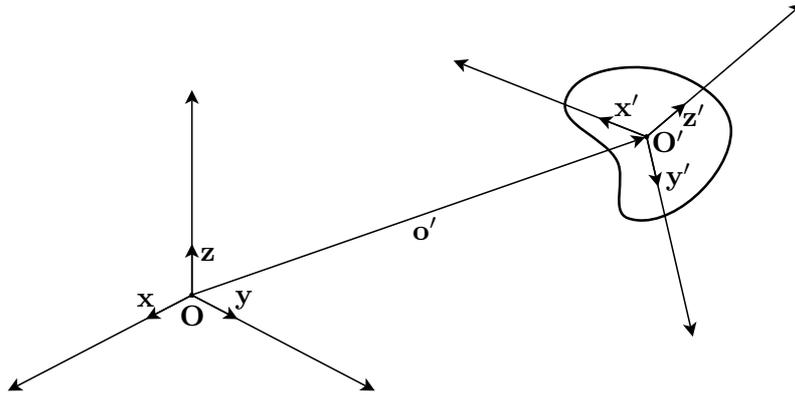


Figura 2.1: Posição e orientação do corpo rígido.

A posição de um ponto \mathbf{O}' pertencente ao corpo rígido com relação ao sistema de coordenadas $\mathbf{O}-xyz$ é expressa pela relação

$$\mathbf{o}' = o'_x \mathbf{x} + o'_y \mathbf{y} + o'_z \mathbf{z}, \quad (2.2)$$

em que o'_x , o'_y e o'_z correspondem as projeções do vetor \mathbf{o}' nos eixos do sistema de coordenadas inercial. De maneira mais compacta tem-se que

$$\mathbf{o}' = \begin{bmatrix} o'_x \\ o'_y \\ o'_z \end{bmatrix}. \quad (2.3)$$

A orientação do corpo é descrita através da orientação relativa entre um sistema de coordenadas ortonormal fixado ao corpo e o sistema de coordenadas inercial. Denomina-se $\mathbf{O}'-x'y'z'$ o sistema de coordenadas do corpo e \mathbf{x}' , \mathbf{y}' , $\mathbf{z}' \in \mathbb{R}^3$ as coordenadas dos eixos principais deste novo sistema relativos ao sistema de coordenadas $\mathbf{O}-xyz$ (ver Figura 2.1). Estes vetores são expressos como:

$$\begin{aligned} \mathbf{x}' &= x'_x \mathbf{x} + x'_y \mathbf{y} + x'_z \mathbf{z}, \\ \mathbf{y}' &= y'_x \mathbf{x} + y'_y \mathbf{y} + y'_z \mathbf{z}, \\ \mathbf{z}' &= z'_x \mathbf{x} + z'_y \mathbf{y} + z'_z \mathbf{z}. \end{aligned} \quad (2.4)$$

Adotando uma notação compacta, os vetores unitários em (2.4) descrevendo a orientação do corpo com respeito ao sistema de coordenadas de referência podem ser combinados em uma matriz

$$\mathbf{R} = \begin{bmatrix} \mathbf{x}' & \mathbf{y}' & \mathbf{z}' \end{bmatrix} = \begin{bmatrix} x'_x & y'_x & z'_x \\ x'_y & y'_y & z'_y \\ x'_z & y'_z & z'_z \end{bmatrix} = \begin{bmatrix} \mathbf{x}'^T \mathbf{x} & \mathbf{y}'^T \mathbf{x} & \mathbf{z}'^T \mathbf{x} \\ \mathbf{x}'^T \mathbf{y} & \mathbf{y}'^T \mathbf{y} & \mathbf{z}'^T \mathbf{y} \\ \mathbf{x}'^T \mathbf{z} & \mathbf{y}'^T \mathbf{z} & \mathbf{z}'^T \mathbf{z} \end{bmatrix}, \quad (2.5)$$

denominada matriz de rotação. As matrizes de rotação, portanto, descrevem a rotação em torno de um eixo no espaço necessária para alinhar os eixos do sistema de referência ao sistema do corpo. Há ainda outro significado geométrico, referente à representação de um mesmo vetor em sistemas de coordenadas diferentes. Seja \mathbf{p} a representação do ponto \mathbf{P} com respeito ao sistema de coordenadas $\mathbf{O}-xyz$ e \mathbf{p}' a representação deste mesmo ponto \mathbf{P} com respeito ao sistema de coordenadas $\mathbf{O}'-x'y'z'$. Como \mathbf{p} e \mathbf{p}' são representações para o mesmo ponto \mathbf{P} , obtém-se

$$\mathbf{p} = p'_x \mathbf{x}' + p'_y \mathbf{y}' + p'_z \mathbf{z}' = \begin{bmatrix} \mathbf{x}' & \mathbf{y}' & \mathbf{z}' \end{bmatrix} \begin{bmatrix} p'_x \\ p'_y \\ p'_z \end{bmatrix} = \mathbf{R} \mathbf{p}'. \quad (2.6)$$

As colunas das matrizes de rotação correspondem aos eixos de um sistema de coordenadas, que são ortogonais e de módulo unitário, o que implica na ortonormalidade da matriz \mathbf{R} . Desta forma, $\mathbf{R}^T \mathbf{R} = \mathbf{I}_{3 \times 3}$ e, portanto, $\det \mathbf{R} = \pm 1$. O determinante da matriz de rotação pode ser encontrado através da seguinte relação algébrica:

$$\det \mathbf{R} = \mathbf{x}'^T (\mathbf{y}' \times \mathbf{z}'). \quad (2.7)$$

Para sistemas de coordenadas construídos através da regra da mão direita, $\det \mathbf{R} = 1$, e as matrizes de rotação assim definidas são ditas pertencentes ao *Grupo Especial Ortonormal* $SO(3) = \{\mathbf{R} \in \mathbb{R}^{3 \times 3} : \mathbf{R}^T \mathbf{R} = \mathbf{I}, \det \mathbf{R} = 1\}$. As matrizes de rotação elementares, em torno dos eixos \mathbf{x} , \mathbf{y} e \mathbf{z} , são obtidas com facilidade e dadas por:

$$\mathbf{R}_x(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\alpha & -s\alpha \\ 0 & s\alpha & c\alpha \end{bmatrix}; \mathbf{R}_y(\beta) = \begin{bmatrix} c\beta & 0 & s\beta \\ 0 & 1 & 0 \\ -s\beta & 0 & c\beta \end{bmatrix}; \mathbf{R}_z(\gamma) = \begin{bmatrix} c\gamma & -s\gamma & 0 \\ s\gamma & c\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.8)$$

onde $cv = \cos(v)$ e $sv = \sin(v)$ para $v = \alpha, \beta, \gamma$. As colunas destas matrizes constituem os vetores unitários do sistema de coordenadas rotacionado com respeito ao sistema original. Uma representação compacta [16] da posição e orientação de um corpo rígido é realizada através das transformações homogêneas. Sejam $\mathbf{O}_1-x_1y_1z_1$ e $\mathbf{O}_2-x_2y_2z_2$ dois sistemas de coordenadas ortonormais. A matriz de transforma-

ção homogênea \mathbf{T}_{12} representa de forma compacta a transformação de coordenadas (translação e rotação) entre os dois sistemas de coordenadas por

$$\mathbf{T}_{12} = \begin{bmatrix} \mathbf{R}_{12} & (\mathbf{p}_{12})_1 \\ \mathbf{0} & 1 \end{bmatrix}, \quad (2.9)$$

e uma vez que $\mathbf{p}_{12} \in \mathbb{R}^3$ e $\mathbf{R}_{12} \in SO(3)$, esta matriz pertence ao *Grupo Especial Euclidiano* $SE(3) = \mathbb{R}^3 \times SO(3)$. No que se segue, assim como em (2.9), é utilizado um subíndice para explicitar o sistema de coordenadas escolhido como referência, sendo $(\mathbf{p})_i$ a representação do ponto \mathbf{p} no sistema de coordenadas $\mathbf{O}_i-x_iy_iz_i$. A equação (2.6) é então reescrita:

$$(\mathbf{p})_0 = \mathbf{R}_{0i}(\mathbf{p})_i. \quad (2.10)$$

Note que para a matriz de transformação homogênea a propriedade de ortogonalidade não permanece válida dado que, em geral, $\mathbf{T}^{-1} \neq \mathbf{T}^T$.

2.2 Cinemática direta

Como foi visto na seção anterior, um manipulador robótico consiste em uma cadeia cinemática de corpos rígidos, conectados por meio de juntas, tipicamente prismáticas ou de revolução. O início da estrutura mecânica é denominada base e a extremidade final é chamada efetuador, uma vez que corresponde ao ponto da cadeia que realizará efetivamente a manipulação de objetos no espaço de trabalho. Um exemplo de manipulador robótico é mostrado na Figura 2.2.

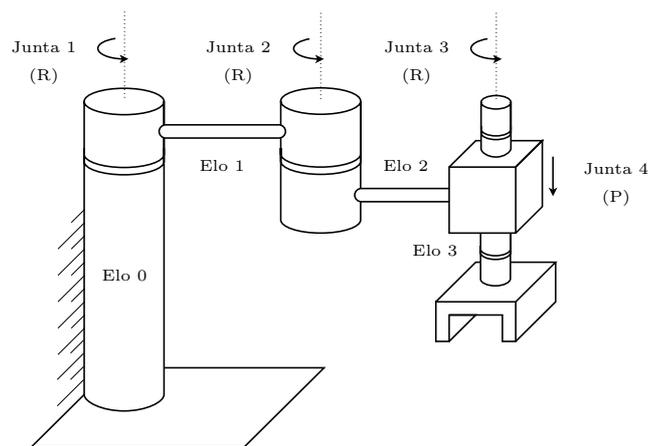


Figura 2.2: Manipulador SCARA.

A estrutura mecânica formada pela composição de elos pode ser caracterizada pelo seu número de graus de mobilidade, que determinam de forma única a configuração de todos os corpos rígidos da cadeia cinemática. Cada grau de mobilidade

está tipicamente associado a uma articulação de junta, definindo uma *variável de junta*. Ângulos constituem as variáveis de juntas de revolução e deslocamentos as variáveis de juntas prismáticas. As cadeias cinemáticas podem ainda ser caracterizadas como *fechadas* ou *abertas*. As cadeias fechadas correspondem a estruturas em que uma sequência de elos forma um laço, enquanto cadeias *abertas* correspondem a estruturas em que há apenas uma sequência de elos conectando suas extremidades. O objetivo da cinemática direta consiste em, a partir do conhecimento das variáveis das juntas, calcular a posição e orientação da extremidade final da cadeia cinemática com respeito a um sistema de coordenadas de referência, tipicamente posicionado na base da estrutura mecânica.

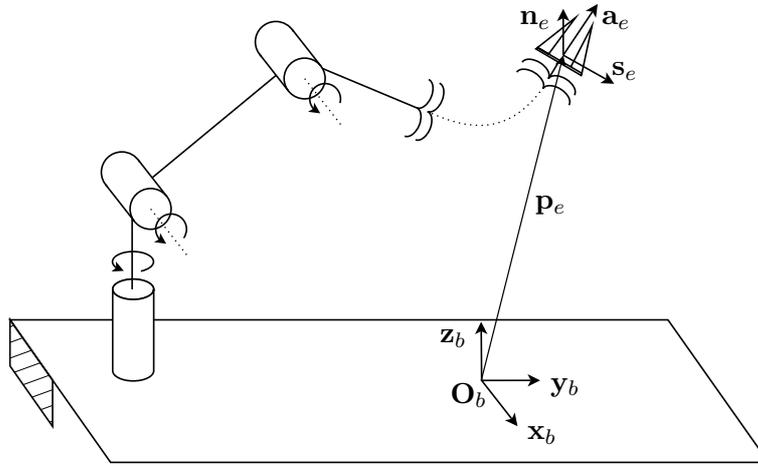


Figura 2.3: Descrição da orientação e posição do sistema de coordenadas do efetuador.

A configuração de um corpo rígido em relação a um determinado sistema de coordenadas de referência é completamente descrito pela posição da origem e pelos vetores unitários que definem o sistema de coordenadas do corpo. Assim, relativamente ao sistema de coordenadas de referência $\mathbf{O}_b-x_b y_b z_b$, a cinemática direta é expressa pela matriz de transformação homogênea

$$\mathbf{T}_{be}(\mathbf{q}) = \begin{bmatrix} \mathbf{n}_e(\mathbf{q}) & \mathbf{s}_e(\mathbf{q}) & \mathbf{a}_e(\mathbf{q}) & (\mathbf{p}_e(\mathbf{q}))_b \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (2.11)$$

onde \mathbf{q} corresponde ao vetor de variáveis das juntas, \mathbf{n}_e , \mathbf{s}_e e \mathbf{a}_e são os vetores unitários que definem os eixos do sistema de coordenadas do efetuador, convenientemente escolhido de acordo com a geometria da tarefa, e $(\mathbf{p}_e)_b$ corresponde ao vetor de posição da origem deste sistema de coordenadas em relação ao sistema de coordenadas de referência $\mathbf{O}_b-x_b y_b z_b$. Uma escolha típica para os vetores \mathbf{n}_e , \mathbf{s}_e e \mathbf{a}_e é apresentada na Figura 2.3. A cinemática direta pode então ser calculada a partir de uma análise geométrica da estrutura do manipulador em questão. Porém,

a complexidade da estrutura e o grande número de juntas pode tornar o cálculo extremamente trabalhoso. Para o caso de cadeias fechadas, tal análise se mostra ainda mais complexa. Desta forma, torna-se interessante a utilização de um procedimento sistemático e geral para obter a cinemática direta de um manipulador: a convenção de Denavit-Hartenberg [16, 49].

2.2.1 Cadeia cinemática aberta

Considera-se um manipulador de cadeia aberta constituído por $n + 1$ elos conectados por n juntas, onde o Elo 0 é convenientemente fixado ao solo. Assume-se que cada junta adiciona à estrutura mecânica um único grau de liberdade, correspondente à variável da junta. A construção de um procedimento operacional para o cálculo das equações de cinemática direta é naturalmente derivada de uma típica cadeia cinemática aberta da estrutura do manipulador. De fato, já que cada junta conecta dois elos consecutivos, é razoável considerar primeiramente a descrição da relação cinemática entre estes e, de maneira recursiva, obter a descrição total do manipulador. Assim, define-se um sistema de coordenadas em cada elo, do elo 0 para o elo n .

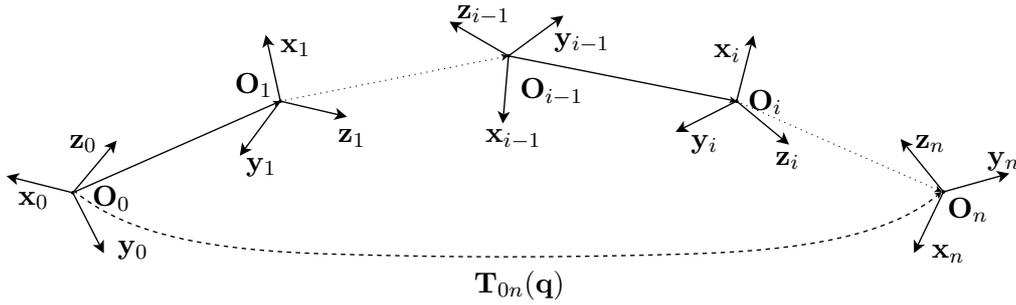


Figura 2.4: Transformação de coordenadas em uma cadeia cinemática aberta.

A transformação homogênea que descreve a posição e orientação do sistema de coordenadas n tendo o sistema de coordenadas 0 (Figura 2.4) como referência é obtida pelo produto de matrizes de transformação homogênea entre elos consecutivos

$$\mathbf{T}_{0n}(\mathbf{q}) = \mathbf{T}_{01}(q_1) \mathbf{T}_{12}(q_2) \dots \mathbf{T}_{n-1,n}(q_n), \quad (2.12)$$

onde $\mathbf{q} = [q_1 \ q_2 \ \dots \ q_n]^T$ é o vetor de variáveis das juntas. Como pretendido, o cálculo da cinemática direta é recursivo e obtido de maneira sistemática através de simples produtos de matriz de transformação homogênea $\mathbf{T}_{i-1,i}(q_i)$ (para $i = 1, 2, \dots, n$), onde cada matriz é função de uma única variável de junta. A transformação homogênea que descreve a posição e orientação do efetuador tendo como referência o sistema de coordenadas da base pode ser obtida, portanto, por

$$\mathbf{T}_{be}(\mathbf{q}) = \mathbf{T}_{b0} \mathbf{T}_{0n}(\mathbf{q}) \mathbf{T}_{ne}, \quad (2.13)$$

onde \mathbf{T}_{b_0} e \mathbf{T}_{n_e} são transformações homogêneas tipicamente constantes que descrevem, respectivamente, a posição e orientação do sistema 0 com respeito a base, e do sistema de coordenadas do efetuador com respeito ao sistema n .

2.2.2 Espaço das juntas e espaço operacional

A equação de cinemática direta para um determinado manipulador permite expressar a posição e orientação do sistema de coordenadas do efetuador relativas ao sistema de coordenadas da base como função das variáveis das juntas. Quando uma tarefa é especificada para o efetuador, é necessário associar a posição e orientação do efetuador, eventualmente como uma função do tempo. Para a posição, isto se torna simples e a trajetória desejada é facilmente expressa como função do tempo. Porém, para o caso da orientação, existe uma grande dificuldade em especificar 9 elementos de uma matriz de rotação e garantir que suas colunas satisfaçam as condições de ortonormalidade a cada instante de tempo. A solução natural para este problema consiste na utilização de representações mínimas para a orientação: ângulos de Euler ou ângulo e eixo [16]. Portanto, a posição é dada por um número mínimo de coordenadas em respeito a geometria da estrutura e a orientação pode ser especificada em termos de representações mínimas que descrevem a rotação do sistema de coordenadas do efetuador relativa ao sistema de coordenadas da base. Desta forma, torna-se possível descrever a postura do efetuador por

$$\mathbf{x}_e = \begin{bmatrix} \mathbf{p}_e \\ \phi_e \end{bmatrix} \in \mathbb{R}^m, \quad m \leq n, \quad (2.14)$$

onde \mathbf{p}_e descreve a posição do efetuador e ϕ_e sua orientação. A orientação pode também ser especificada através de uma representação adequada, como quaternions unitários. Esta representação de posição e orientação permite a descrição da tarefa designada ao efetuador, em termos de um número de parâmetros independentes. O vetor \mathbf{x}_e é definido no espaço onde a tarefa do manipulador é especificada, isto é, no espaço de operação do robô. Por isso, este espaço é tipicamente chamado *espaço operacional* (\mathcal{O}). Por outro lado, o espaço de configuração onde o vetor $\mathbf{q} \in \mathbb{R}^n$ é definido é chamado *espaço das juntas* (\mathcal{Q}). A cinemática direta pode ser escrita, de forma alternativa a (2.13), como:

$$\mathbf{x}_e = \mathbf{h}(\mathbf{q}), \quad (2.15)$$

onde a postura (posição e orientação) do efetuador é uma função das variáveis das juntas, geralmente não-linear. Tipicamente, variáveis de juntas de revolução são designadas por θ_i e variáveis de juntas prismáticas por d_i .

2.3 Cinemática diferencial

Na Seção 2.2 foi obtida uma descrição da postura (posição e orientação) do efetuador \mathbf{x}_e no instante t a partir do vetor \mathbf{q} das variáveis das juntas neste mesmo instante. A cinemática diferencial, por sua vez, estabelece a relação entre a variação temporal da postura do efetuador em função da variação temporal das variáveis das juntas. Isto é, deseja-se descrever a relação entre as velocidades linear e angular do efetuador e as velocidades das juntas. Este mapeamento, como será visto, é realizado pelo *Jacobiano geométrico*. A matriz Jacobiana constitui uma das principais ferramentas para o estudo de sistemas robóticos, sendo essencial para implementação do controle cinemático, planejamento e execução de trajetórias suaves, identificação de configurações singulares, análise de redundância e mapeamento de forças e torques realizados no efetuador para as juntas [16].

A postura do efetuador pode ser expressa, como visto na Seção 2.2, por seu vetor de posição e por uma representação mínima da orientação, como ângulos de Euler, ou apenas adequada, como é o caso de quaternions unitários. Estas relações podem ser diferenciadas em função das variáveis das juntas, sendo obtida uma relação entre a velocidade linear e a derivada da representação da orientação, e as velocidades das juntas. Este mapeamento, por sua vez, é realizado pelo *Jacobiano analítico*. A relação entre as duas matrizes de mapeamento também é obtida, sendo estabelecida por outra matriz, denominada *Jacobiano de representação*.

2.3.1 Jacobiano geométrico

O objetivo da cinemática diferencial é estabelecer uma relação entre as velocidades linear e angular do efetuador e as velocidades das juntas. É necessário analisar a contribuição da velocidade de cada junta para as velocidades angular e linear do efetuador. A contribuição de uma junta de revolução é dada por

$$\boldsymbol{\omega}_i = \dot{q}_i \mathbf{h}_i, \quad (2.16)$$

$$\mathbf{v}_i = \boldsymbol{\omega}_i \times \mathbf{p}_{ie} = \dot{q}_i \mathbf{h}_i \times \mathbf{p}_{ie}, \quad (2.17)$$

em que \mathbf{h}_i constitui o vetor unitário na direção do eixo de rotação da i -ésima junta e $q_i = \theta_i$. Para uma junta prismática

$$\boldsymbol{\omega}_i = \mathbf{0}, \quad (2.18)$$

$$\mathbf{v}_i = \dot{q}_i \mathbf{h}_i, \quad (2.19)$$

onde $q_i = d_i$. As velocidades linear e angular são obtidas considerando as contribuições de todas as juntas. Por exemplo, para um manipulador composto apenas por

juntas de revolução, podemos expressar os vetores \mathbf{v}_e e $\boldsymbol{\omega}_e$ por

$$\mathbf{V}_e = \begin{bmatrix} \mathbf{v}_e \\ \boldsymbol{\omega}_e \end{bmatrix} = \begin{bmatrix} \mathbf{h}_1 \times \mathbf{p}_{1e} & \mathbf{h}_2 \times \mathbf{p}_{2e} & \dots & \mathbf{h}_n \times \mathbf{p}_{ne} \\ \mathbf{h}_1 & \mathbf{h}_2 & \dots & \mathbf{h}_n \end{bmatrix} \dot{\mathbf{q}}, \quad (2.20)$$

ou ainda,

$$\mathbf{V}_e = \begin{bmatrix} \mathbf{J}_p(\mathbf{q}) \\ \mathbf{J}_o(\mathbf{q}) \end{bmatrix} \dot{\mathbf{q}} = \mathbf{J}_G(\mathbf{q}) \dot{\mathbf{q}}. \quad (2.21)$$

Nota-se que a matriz Jacobiana depende do sistema de coordenadas escolhido como referência. Assim, se for desejado representar o Jacobiano em um determinado sistema de coordenadas $\mathbf{O}_b-x_b y_b z_b$, basta utilizar a representação no eixo de interesse dos vetores presentes na matriz Jacobiana, conforme (2.10).

2.3.2 Jacobiano analítico

Como visto na Seção 2.2, a postura do efetuador pode ser descrita pelo seu vetor posição \mathbf{p} e uma representação mínima ou adequada da orientação $\boldsymbol{\phi}$. Esta configuração, como visto, depende das variáveis das juntas e pode ser expressa de forma compacta por (2.15). Com o intuito de relacionar a variação temporal da postura do efetuador com a variação temporal das variáveis de juntas, a diferenciação da equação de cinemática direta se mostra um passo natural ao desenvolvimento. A velocidade translacional do efetuador pode ser expressa como a derivada temporal de sua posição, ou seja,

$$\dot{\mathbf{p}} = \frac{\partial \mathbf{p}}{\partial \mathbf{q}} \dot{\mathbf{q}} = \mathbf{J}_p(\mathbf{q}) \dot{\mathbf{q}}. \quad (2.22)$$

Dada a representação da orientação $\boldsymbol{\phi}$, a sua derivada temporal é obtida:

$$\dot{\boldsymbol{\phi}} = \frac{\partial \boldsymbol{\phi}}{\partial \mathbf{q}} \dot{\mathbf{q}} = \mathbf{J}_\phi(\mathbf{q}) \dot{\mathbf{q}}. \quad (2.23)$$

Desta forma, a equação de cinemática diferencial é obtida analiticamente como

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{\mathbf{p}} \\ \dot{\boldsymbol{\phi}} \end{bmatrix} = \begin{bmatrix} \mathbf{J}_p(\mathbf{q}) \\ \mathbf{J}_\phi(\mathbf{q}) \end{bmatrix} \dot{\mathbf{q}} = \mathbf{J}(\mathbf{q}) \dot{\mathbf{q}}, \quad (2.24)$$

onde o *Jacobiano analítico* do manipulador é dado por

$$\mathbf{J} = \frac{\partial \mathbf{h}(\mathbf{q})}{\partial \mathbf{q}} \in \mathbb{R}^{m \times n}. \quad (2.25)$$

É importante observar que a matriz Jacobiana $\mathbf{J}_\phi(\mathbf{q})$ não é facilmente calculada, uma vez que a representação da orientação $\boldsymbol{\phi}$, em geral, não está na forma direta e requer o cálculo de elementos de uma matriz de rotação relativa.

2.3.3 Jacobiano de representação

Considera-se uma representação ϕ , mínima ou apenas adequada da orientação. A relação entre a velocidade angular $\boldsymbol{\omega}$ e a derivada temporal $\dot{\phi}$, ou velocidade rotacional, é estabelecida pelo *Jacobiano de representação* através da seguinte igualdade:

$$\dot{\phi} = \mathbf{J}_R(\phi) \boldsymbol{\omega}. \quad (2.26)$$

onde a função $\mathbf{J}_R(\phi)$ depende da representação escolhida. Estabelecida a relação entre $\boldsymbol{\omega}$ e $\dot{\phi}$, a relação entre o *Jacobiano analítico* e o *Jacobiano geométrico* é obtida:

$$\mathbf{J} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{J}_R(\phi) \end{bmatrix} \mathbf{J}_G = \mathbf{M}_R \mathbf{J}_G. \quad (2.27)$$

2.3.4 Singularidades cinemáticas

A matriz Jacobiana na equação de cinemática diferencial de um determinado manipulador robótico define um mapeamento

$$\mathbf{V}_e = \mathbf{J}_G(\mathbf{q})\dot{\mathbf{q}} \quad (2.28)$$

entre o vetor $\dot{\mathbf{q}}$ de velocidades das juntas e o vetor $\mathbf{V}_e = [\mathbf{v}_e^T \boldsymbol{\omega}_e^T]^T$ de velocidades do efetuador. O Jacobiano é geralmente uma função da configuração do manipulador e, para algumas configurações específicas, perde posto. Estas configurações, chamadas de *singularidades cinemáticas*, devem ser evitadas no momento do planejamento de trajetórias pois representam configurações em que a mobilidade da estrutura é reduzida, se tornando inviável mover o efetuador em algumas direções. Na proximidade de singularidades, grandes velocidades no espaço das juntas são desenvolvidas para pequenas velocidades no espaço operacional. Além disso, em tais configurações, podem haver infinitas soluções para o problema de cinemática inversa [16].

2.4 Cinemática inversa

O problema de cinemática inversa consiste em determinar as variáveis de juntas que resultam em uma determinada posição e orientação do efetuador. Dada uma função de cinemática direta $\mathbf{h} : \mathcal{Q} \rightarrow \mathcal{O}$ e uma postura desejada $\mathbf{x}_d \in \mathbb{R}^m$, deseja-se resolver a equação

$$\mathbf{h}(\mathbf{q}) = \mathbf{x}_d, \quad (2.29)$$

para algum $\mathbf{q} \in \mathbb{R}^n$. O problema de cinemática inversa é de maior complexidade quando comparado a cinemática direta por alguns motivos: (i) as equações para solucionar são, em geral, não lineares e por isso não é sempre possível encontrar

uma solução na forma fechada; (ii) podem existir múltiplas ou infinitas soluções, como em manipuladores redundantes; e (iii) podem existir soluções não admissíveis em vista da estrutura cinemática do manipulador. A solução é garantida somente se a postura dada pertence ao espaço de trabalho destro do manipulador [16]. Por outro lado, a existência de múltiplas soluções depende do número de graus de mobilidade e parâmetros DH não nulos. Em geral, quanto maior o número de parâmetros DH não nulos, maior o número de soluções admissíveis.

Tradicionalmente, as soluções para o problema de cinemática inversa são divididas em duas classes: soluções na forma fechada e soluções numéricas. Soluções na forma fechada são obtidas através do uso de identidades geométricas e algébricas para resolver o conjunto de equações não lineares que definem a cinemática inversa. As soluções numéricas, por sua vez, se baseiam na utilização de métodos iterativos para resolver a equação (2.29). Assim como no problema de cinemática direta, em que a solução completa resulta da análise recursiva de soluções menores, para a cinemática inversa uma ideia semelhante pode ser aplicada. O método de sub-problemas de Paden-Kahan, apresentado originalmente em [31], permite a solução do problema proposto através da solução de subproblemas frequentes nas soluções de cinemática inversa de manipuladores. Para maiores esclarecimentos sobre o método de sub-problemas de Paden-Kahan, o leitor é convidado a consultar [33].

2.4.1 Abordagem diferencial

Como foi visto na Seção 2.3, a equação de cinemática diferencial representa um mapeamento linear entre as velocidades desenvolvidas no espaço de juntas e as velocidades no espaço operacional, sugerindo a possibilidade de utilizar tal abordagem para resolver o problema de cinemática inversa [16]. A solução diferencial consiste em, dada uma configuração inicial $\mathbf{q}(t_0)$, definir $\dot{\mathbf{q}}$ de forma que

$$\mathbf{q}(t) \rightarrow \mathbf{q}^*, \quad (2.30)$$

onde \mathbf{q}^* corresponde a uma configuração de juntas que satisfaz (2.29). Nota-se que \mathbf{q}^* é desconhecido. Como será visto, a definição de $\dot{\mathbf{q}}$ a partir de variáveis de erro no espaço de operação \mathcal{O} e não no espaço das juntas \mathcal{Q} resultará na utilização da inversa (ou pseudo-inversa) da matriz Jacobiana.

2.4.2 Manipuladores não-redundantes

O objetivo de controle é conduzir a estrutura mecânica de uma postura inicial $\mathbf{x}(t_0) = (\mathbf{p}(t_0), \boldsymbol{\phi}(t_0))$ para uma postura desejada $\mathbf{x}_d(t) = (\mathbf{p}_d(t), \boldsymbol{\phi}_d(t))$. A postura desejada pode ser invariante no tempo, o que define um problema de *regulação* ou variante

no tempo, definindo um problema de *rastreamento*. Deseja-se, portanto, atuar nas juntas do manipulador de forma que o efetuador descreva uma trajetória desejada (posição e orientação). O objetivo de controle é então formalizado como:

$$\mathbf{x} \rightarrow \mathbf{x}_d \quad \text{ou} \quad \mathbf{e} := \mathbf{x}_d - \mathbf{x} \rightarrow \mathbf{0}, \quad (2.31)$$

onde $\mathbf{e} \in \mathbb{R}^m$ corresponde ao erro de rastreamento no espaço operacional. A obtenção da dinâmica do erro \mathbf{e} torna-se necessária para estabelecer a análise da convergência do mesmo a zero. Assim, a equação do erro é obtida através da diferenciação de ambos os lados de (2.31) e a partir da cinemática diferencial.

$$\dot{\mathbf{e}} = \dot{\mathbf{x}}_d - \dot{\mathbf{x}} = \dot{\mathbf{x}}_d - \mathbf{J}(\mathbf{q}) \dot{\mathbf{q}}. \quad (2.32)$$

Nota-se que a utilização de grandezas no espaço operacional implica naturalmente na utilização do *Jacobiano analítico* ao invés do *Jacobiano geométrico*. O objetivo de controle consiste em escolher $\dot{\mathbf{q}}$ tal que a variável de erro \mathbf{e} convirja para zero. A matriz $\mathbf{J}(\mathbf{q})$ é quadrada ($m = n$), e assumindo posto completo (não-singular e portanto inversível), a escolha

$$\dot{\mathbf{q}} = \mathbf{J}(\mathbf{q})^{-1} \underbrace{(\dot{\mathbf{x}}_d + \mathbf{\Lambda} \mathbf{e})}_{\nu} \quad (2.33)$$

conduz ao seguinte sistema linear

$$\dot{\mathbf{e}} + \mathbf{\Lambda} \mathbf{e} = 0. \quad (2.34)$$

A escolha de uma matriz $\mathbf{\Lambda} \in \mathbb{R}^{m \times m}$ *positiva definida* garante a estabilidade assintótica do erro, sendo suficiente uma escolha *-Λ Hurwitz*. O erro tende a zero ao longo da trajetória com uma taxa de convergência que depende dos autovalores da matriz de ganho $\mathbf{\Lambda}$. A utilização de $\mathbf{J}(\mathbf{q})^{-1}$ permite a linearização do sistema, compensando o termo não-linear $\mathbf{J}(\mathbf{q})$, e a utilização do termo *feedforward* $\dot{\mathbf{x}}_d$ garante a manutenção do erro em $\mathbf{0}$ ao longo da trajetória, independente do tipo de sinal de referência \mathbf{x}_d . O bloco correspondente ao algoritmo implementado é apresentado na Figura 2.5.

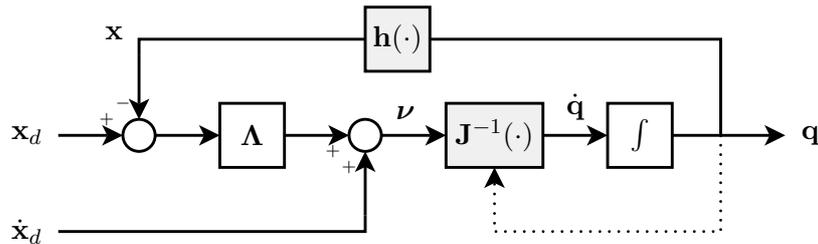


Figura 2.5: Algoritmo de cinemática inversa (caso não-redundante).

2.4.3 Erro de posição

A definição do erro de posição é bastante natural

$$\mathbf{e}_p = \mathbf{p}_d - \mathbf{p}(\mathbf{q}), \quad (2.35)$$

onde \mathbf{p}_d e $\mathbf{p}(\mathbf{q})$ correspondem a posição desejada e a atual do efetuador, respectivamente. A derivada temporal do erro de posição é dada por

$$\dot{\mathbf{e}}_p = \dot{\mathbf{p}}_d - \dot{\mathbf{p}}. \quad (2.36)$$

2.4.4 Erro de orientação

Por outro lado, a expressão do erro de orientação depende da representação particular escolhida para a orientação do efetuador, como ângulos de Euler ou quaternions unitários.

Ângulos de Euler

A expressão para o erro de orientação é escolhida de forma análoga à equação(2.35), ou seja,

$$\mathbf{e}_o = \boldsymbol{\phi}_d - \boldsymbol{\phi}(\mathbf{q}), \quad (2.37)$$

e sua dinâmica é expressa por

$$\dot{\mathbf{e}}_o = \dot{\boldsymbol{\phi}}_d - \dot{\boldsymbol{\phi}}. \quad (2.38)$$

Assim, assumindo que não haja singularidades cinemáticas ou de representação, a solução de controle é obtida de (2.33), isto é,

$$\dot{\mathbf{q}} = \mathbf{J}(\mathbf{q})^{-1} \begin{bmatrix} \dot{\mathbf{p}}_d + \boldsymbol{\Lambda}_p \mathbf{e}_p \\ \dot{\boldsymbol{\phi}}_d + \boldsymbol{\Lambda}_o \mathbf{e}_o \end{bmatrix}, \quad (2.39)$$

onde $\boldsymbol{\Lambda}_p$ e $\boldsymbol{\Lambda}_o$ são matrizes *positivas definidas*, garantindo a *estabilidade assintótica* do erro \mathbf{e} .

Quaternion unitário

A representação da orientação por quaternions requer a definição de um erro de orientação adequado. Assim, sejam $\mathbf{Q}_d = \{\eta_d, \boldsymbol{\epsilon}_d\}$ e $\mathbf{Q}_e = \{\eta_e, \boldsymbol{\epsilon}_e\}$ os quaternions associados a \mathbf{R}_d e \mathbf{R}_{0e} , respectivamente. O erro de orientação será descrito pela matriz de rotação (ou atitude) \mathbf{R}_ϕ . Esta matriz pode ser definida, através do enfoque inercial, como $\mathbf{R}_\phi = \mathbf{R}_d \mathbf{R}_{0e}^T$. O objetivo de controle é então formalizado como:

$$\mathbf{R}_{0e} \rightarrow \mathbf{R}_d(t) \quad \text{ou} \quad \mathbf{R}_\phi \rightarrow \mathbf{I}. \quad (2.40)$$

A representação da matriz \mathbf{R}_ϕ em quaternions é dada por $\Delta\mathbf{Q} = \{\Delta\eta, \Delta\boldsymbol{\epsilon}\}$.

$$\Delta\mathbf{Q} = \mathbf{Q}_d * \mathbf{Q}_e^{-1}. \quad (2.41)$$

A representação da matriz identidade \mathbf{I} em quaternions unitários é dada por $\mathbf{Q}_I = \{1, \mathbf{0}\}$. Assim, $\Delta\mathbf{Q} = \{1, \mathbf{0}\}$ se e somente se \mathbf{R}_d e \mathbf{R}_{0e} estiverem alinhadas. Portanto, é suficiente definir o erro de orientação como

$$\mathbf{e}_o = \Delta\boldsymbol{\epsilon} = \eta_e(\mathbf{q}) \boldsymbol{\epsilon}_d - \eta_d \boldsymbol{\epsilon}_e(\mathbf{q}) - \boldsymbol{\epsilon}_d \times \boldsymbol{\epsilon}_e(\mathbf{q}). \quad (2.42)$$

Desta forma, a solução para o controle pode ser computada como

$$\dot{\mathbf{q}} = \mathbf{J}_G(\mathbf{q})^{-1} \begin{bmatrix} \dot{\mathbf{p}}_d + \boldsymbol{\Lambda}_p \mathbf{e}_p \\ \boldsymbol{\omega}_d + \boldsymbol{\Lambda}_o \mathbf{e}_o \end{bmatrix}, \quad (2.43)$$

onde o *Jacobiano geométrico* foi utilizado. Substituindo (2.43) em (2.21), obtém-se:

$$\boldsymbol{\omega}_d - \boldsymbol{\omega}_e + \boldsymbol{\Lambda}_o \mathbf{e}_o = \mathbf{0}. \quad (2.44)$$

A equação obtida é não-linear em \mathbf{e}_o uma vez que contém a velocidade angular do efetuador ao invés da derivada temporal do erro de orientação. Assim, torna-se importante conhecer a relação entre a derivada temporal do quaternion \mathbf{Q}_e e a velocidade angular $\boldsymbol{\omega}_e$ ([16], pg. 140):

$$\dot{\eta}_e = -\frac{1}{2} \boldsymbol{\epsilon}_e^T \boldsymbol{\omega}_e, \quad (2.45)$$

$$\dot{\boldsymbol{\epsilon}}_e = \frac{1}{2} (\eta_e \mathbf{I}_{3 \times 3} - \boldsymbol{\epsilon}_e \times) \boldsymbol{\omega}_e, \quad (2.46)$$

que é a conhecida equação de propagação do quaternion, estando explícito o *Jacobiano de representação*. O estudo da estabilidade de (2.44) considera a seguinte função de Lyapunov [50]:

$$V(\Delta\eta, \Delta\boldsymbol{\epsilon}) = (\eta_d - \eta_e)^2 + (\boldsymbol{\epsilon}_d - \boldsymbol{\epsilon}_e)^T (\boldsymbol{\epsilon}_d - \boldsymbol{\epsilon}_e). \quad (2.47)$$

Considerando as equações (2.44), (2.45) e (2.46), a diferenciação temporal de (2.47) resulta em

$$\dot{V}(\Delta\eta, \Delta\boldsymbol{\epsilon}) = -\mathbf{e}_o^T \boldsymbol{\Lambda}_o \mathbf{e}_o < 0, \quad (2.48)$$

que será negativa definida uma vez que $\boldsymbol{\Lambda}_o$ é positiva definida. Desta forma, conclui-se a respeito da estabilidade assintótica do sistema e o erro \mathbf{e}_o converge para zero, como pretendido. A prova de estabilidade para o sistema completo (posição e orientação) se dá pela função de Lyapunov $V(\mathbf{e}_p, \Delta\mathbf{Q}) = \mathbf{e}_p^T \mathbf{e}_p + V(\Delta\eta, \Delta\boldsymbol{\epsilon})$, [52].

Simulação 1

Considere um manipulador planar com 3 juntas de revolução (Figura 2.6) cuja postura é descrita pela posição $\mathbf{p} \in \mathbb{R}^2$ e pela orientação $\phi \in \mathbb{R}$.

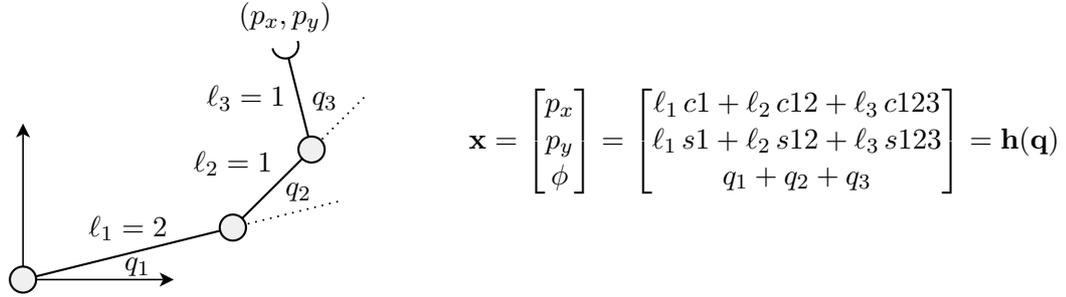


Figura 2.6: Robô planar não-redundante com 3 juntas de revolução.

No que se segue, são apresentados resultados de simulação para rastreamento de posição (Trajetória 1) e regulação da orientação ($\phi_d = \frac{\pi}{2}$). Considera-se uma configuração inicial $\mathbf{q}(0) = [\frac{\pi}{6} \frac{\pi}{6} \frac{\pi}{6}]^T rad$ e ganho $\mathbf{\Lambda} = \lambda \mathbf{I}$.

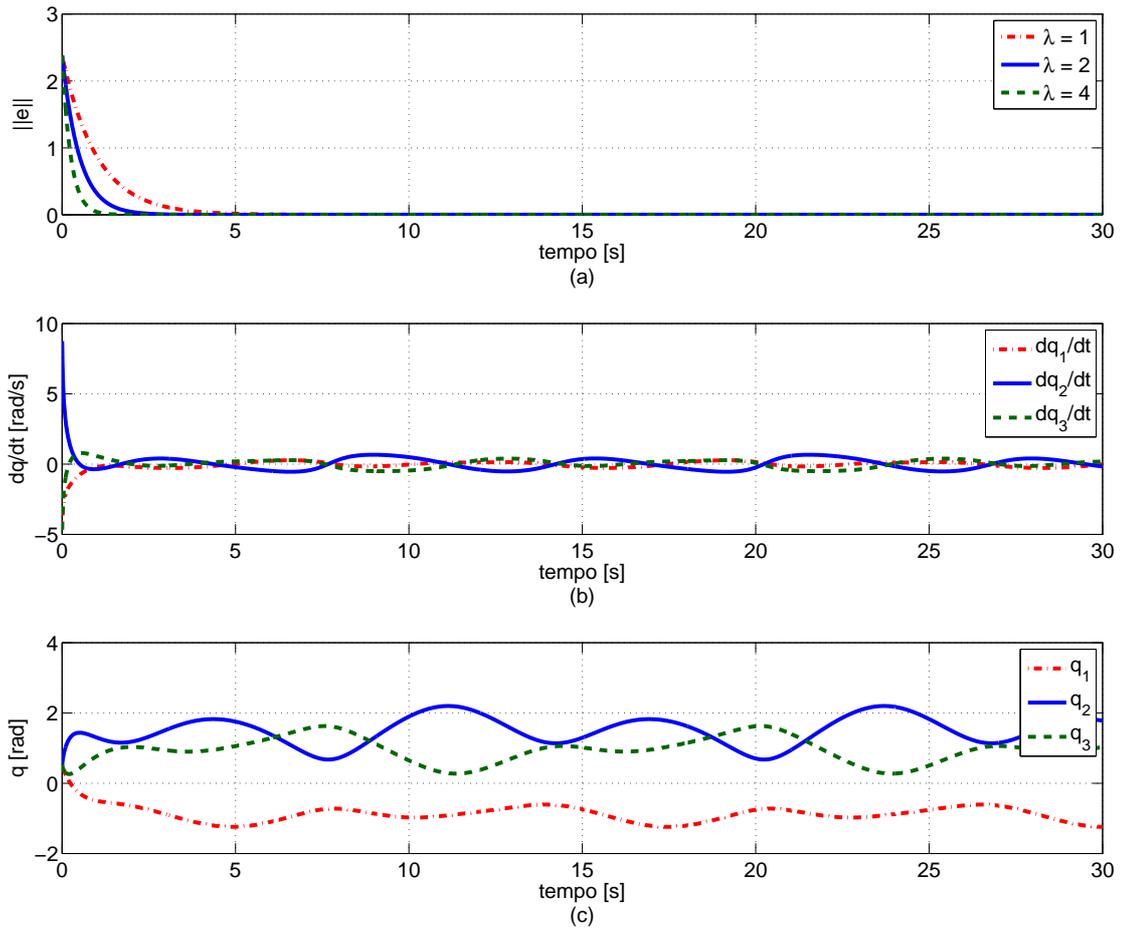


Figura 2.7: Simulação 1: (a) norma do erro para diferentes ganhos λ ; (b) velocidades das juntas para $\lambda = 2$; e (c) ângulos das juntas para $\lambda = 2$.

2.4.5 Manipuladores redundantes

Para o caso onde $m < n$, o sinal $\dot{\mathbf{q}}$ nas juntas pode ser obtido por

$$\dot{\mathbf{q}} = \mathbf{J}^\dagger(\mathbf{q}) \boldsymbol{\nu}, \quad (2.49)$$

onde $\mathbf{J}^\dagger = \mathbf{J}^T (\mathbf{J} \mathbf{J}^T)^{-1} \in \mathbb{R}^{n \times m}$ é a matriz pseudo-inversa à direita de \mathbf{J} . A solução obtida a partir de (2.49) minimiza localmente a norma das velocidades das juntas, desde que $\boldsymbol{\nu}(t)$ não conduza o robô para configurações singulares [53]. A falha desta última condição ainda é um tema de pesquisa relevante na área da robótica, e será discutido a seguir. Para uma lei de controle *feedforward* e proporcional

$$\boldsymbol{\nu} = \dot{\mathbf{x}}_d + \boldsymbol{\Lambda} \mathbf{e}, \quad (2.50)$$

a dinâmica do erro de postura é governada por $\dot{\mathbf{e}} + \boldsymbol{\Lambda} \mathbf{e} = \mathbf{0}$. Novamente, a partir de uma escolha apropriada de $\boldsymbol{\Lambda}$ como uma matriz positiva definida, o sistema em malha fechada é exponencialmente estável e, conseqüentemente, $\lim_{t \rightarrow \infty} \mathbf{e}(t) = \mathbf{0}$. Contudo, quando o robô manipulador é redundante ($m < n$) a matriz Jacobiana possui mais colunas do que linhas e, por isso, existem infinitas soluções para (2.29). Então, devido à presença de $n - m$ graus de liberdade redundantes, a inversão desejada (2.49) pode ser modificada introduzindo-se um termo de projeção no espaço nulo de \mathbf{J} , isto é,

$$\dot{\mathbf{q}} = \mathbf{J}^\dagger \boldsymbol{\nu} + (\mathbf{I} - \mathbf{J}^\dagger \mathbf{J}) \dot{\mathbf{q}}_0, \quad (2.51)$$

onde $\dot{\mathbf{q}}_0 \in \mathbb{R}^n$ é um vetor de velocidades arbitrárias das juntas que pode ser especificado para satisfazer uma restrição adicional ao problema, com prioridade secundária, como evitar singularidades ou desviar de obstáculos [26, 27, 38]. A solução obtida localmente minimiza a norma das velocidades das juntas e permite a geração de movimentos internos para reconfigurar a estrutura do manipulador sem modificar a postura do efetuador [16]. O algoritmo diferencial projetado para solucionar o problema de cinemática inversa é ilustrado no diagrama em blocos da Figura 2.8.

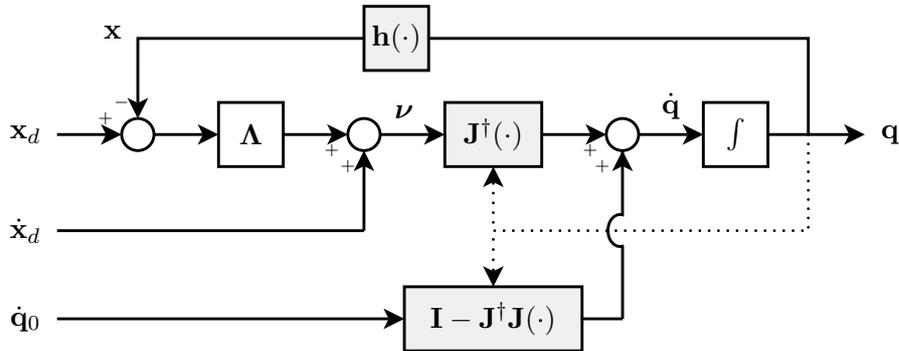


Figura 2.8: Algoritmo de cinemática inversa (caso redundante).

Simulação 2

Considere um manipulador planar com 3 juntas de revolução (Figura 2.9) cuja postura é descrita apenas pela posição $\mathbf{p} \in \mathbb{R}^2$, configurando um problema redundante.

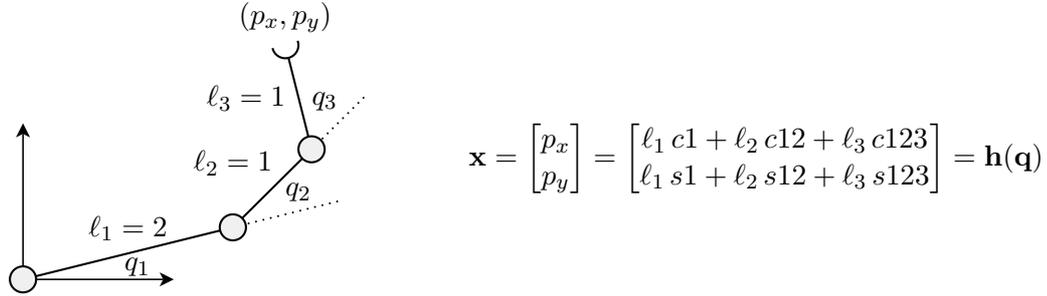


Figura 2.9: Robô planar redundante com 3 juntas de revolução.

A seguir, são apresentados os resultados de simulação obtidos para rastreamento de posição (Trajetória 1) a partir da lei de norma mínima (2.49). Considera-se novamente uma configuração inicial $\mathbf{q}(0) = [\frac{\pi}{6} \ \frac{\pi}{6} \ \frac{\pi}{6}]^T rad$ e ganho $\Lambda = \lambda \mathbf{I}$.

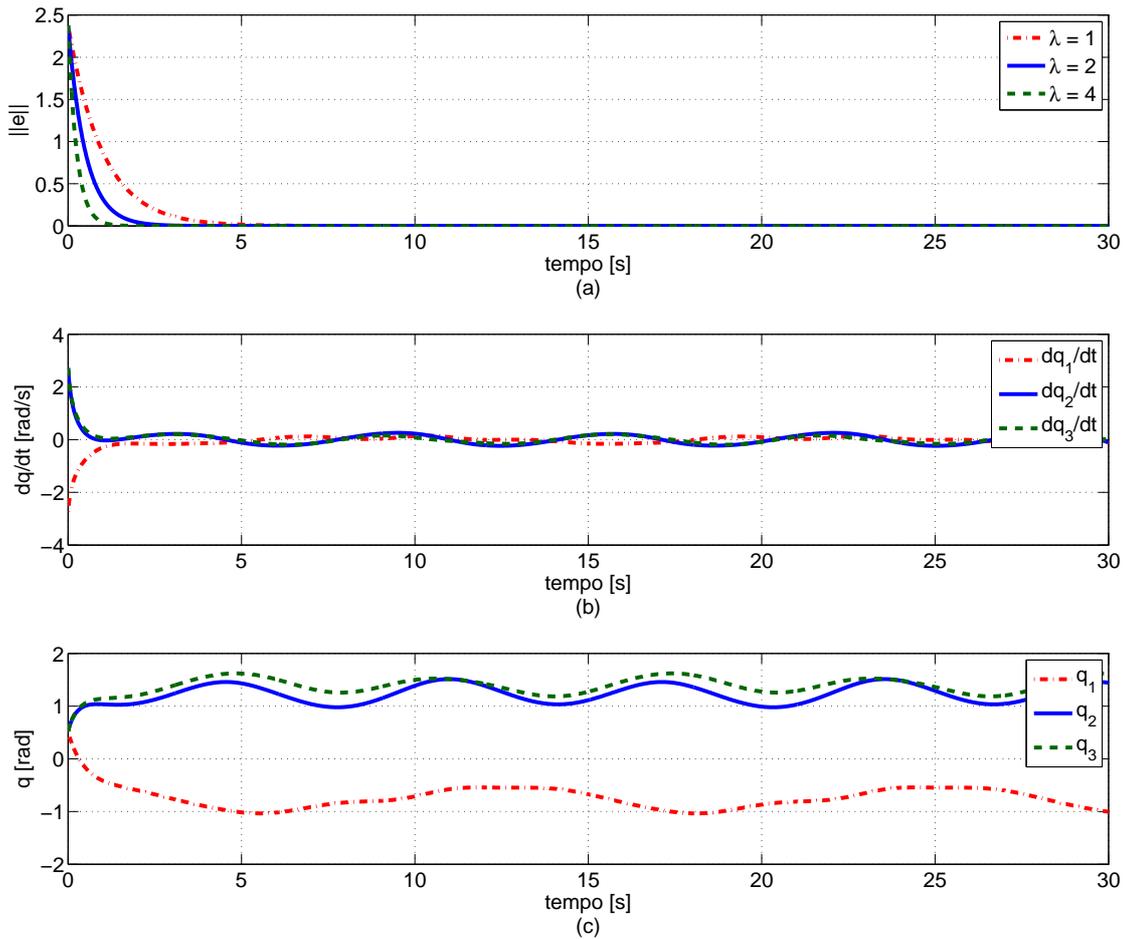


Figura 2.10: Simulação 2: (a) norma do erro para diferentes ganhos λ ; (b) velocidades das juntas para $\lambda = 2$; e (c) ângulos das juntas para $\lambda = 2$.

Capítulo 3

Singularidades e métodos alternativos

A solução para o problema de cinemática inversa baseada na inversa (ou pseudo-inversa) da matriz Jacobiana pode ser computada somente quando esta possui posto completo, tornando-se sem sentido quando o manipulador está em uma *configuração singular*. Neste caso, a matriz Jacobiana possui linhas/colunas linearmente dependentes. É importante ressaltar que a inversão da matriz Jacobiana pode representar um grande inconveniente não apenas em uma configuração singular mas também na vizinhança de uma singularidade, onde a matriz se torna mal condicionada, resultando em grandes valores de velocidades das juntas. Neste capítulo é apresentada uma análise de configurações singulares para manipuladores antropomórficos, cinematicamente simples e bastante empregados por tal facilidade. Serão apresentados também dois algoritmos gerais para solucionar o problema de cinemática inversa na presença de singularidades. O algoritmo DLS utiliza um fator de amortecimento para tornar a inversão melhor condicionada, estabelecendo um compromisso entre a viabilidade e a exatidão da solução [40, 41]. O algoritmo FIK, por sua vez, apresenta uma proposta alternativa que não exige inversão matricial, solucionando o problema sob o ponto de vista de controle ao empregar um laço de realimentação para minimizar a discrepância entre as velocidades demandada e atual [47, 48].

3.1 SVD e manipulabilidade

Uma análise mais rigorosa do comportamento da solução para a cinemática inversa na vizinhança de uma configuração singular pode ser desenvolvida recorrendo-se à *decomposição em valores singulares* da matriz Jacobiana [22]. Assim, seja

$$\mathbf{J} = \sum_{i=1}^m \sigma_i \mathbf{u}_i \mathbf{v}_i^T \quad (3.1)$$

onde \mathbf{v}_i e \mathbf{u}_i são os vetores singulares de entrada e saída, respectivamente, e σ_i são os valores singulares ordenados tal que $\sigma_1 \geq \sigma_2 \geq \dots \sigma_r > 0$, sendo r o posto de \mathbf{J} ¹. Quando \mathbf{J} multiplica uma coluna \mathbf{v}_j , produz σ_j vezes a coluna \mathbf{u}_j , isto é,

$$\mathbf{J} \mathbf{v}_j = \sigma_j \mathbf{u}_j, \quad (3.2)$$

o que permite uma análise da mobilidade da estrutura mecânica na direção \mathbf{u}_j através de σ_j . Para $\sigma_j \gg 0$, movimentos na direção \mathbf{u}_j podem ser obtidos com pequenas variáveis de entrada na direção \mathbf{v}_j . Por outro lado, para valores σ_j pequenos, movimentos nesta direção ainda são possíveis mas exigem grandes variáveis de entrada. Neste caso, a mobilidade da estrutura mecânica é reduzida e grandes velocidades devem existir no espaço de velocidades das juntas para que pequenas velocidades no espaço de operação sejam alcançadas. No caso limite, em que $\sigma_j = 0$, movimentos na direção \mathbf{u}_j tornam-se não-factíveis e configura-se uma singularidade cinemática.

A decomposição SVD permite caracterizar a manipulabilidade da estrutura em termos de velocidade. Considere o conjunto de velocidades de norma unitária e constante

$$\dot{\mathbf{q}}^T \dot{\mathbf{q}} = 1. \quad (3.3)$$

Esta igualdade descreve os pontos de uma esfera no espaço de velocidades das juntas. Para descrever as velocidades no espaço operacional que podem ser desenvolvidas por este conjunto de velocidades de juntas, utiliza-se a equação de cinemática diferencial. Para o caso geral de manipuladores redundantes, a solução de norma mínima $\dot{\mathbf{q}} = \mathbf{J}^\dagger \boldsymbol{\nu}$ pode ser considerada e ao substituir em (3.3) obtém-se

$$\boldsymbol{\nu}^T (\mathbf{J}(\mathbf{q})\mathbf{J}(\mathbf{q})^T)^{-1} \boldsymbol{\nu} = 1, \quad (3.4)$$

que define um elipsoide no espaço de velocidades do efetuador [16]. A forma e a orientação do elipsoide são determinadas pelo núcleo da forma quadrática e portanto pela matrix $\mathbf{J}\mathbf{J}^T$. A direção dos eixos principais são definidas pelos vetores \mathbf{u}_i , para $i = 1, 2, \dots, r$ e as dimensões do eixos são dadas pelos valores singulares de \mathbf{J} [54]. Uma medida bastante utilizada para a manipulabilidade é o volume do elipsoide definido em (3.4), proporcional a

$$\omega(\mathbf{q}) = \sqrt{\det(\mathbf{J}(\mathbf{q})\mathbf{J}^T(\mathbf{q}))}. \quad (3.5)$$

A Figura 3.1 apresenta elipsoides de manipulabilidade para uma braço planar 2R (2

¹A equação (3.1) apresenta a decomposição SVD de uma forma compacta e específica para manipuladores robóticos, onde $m \leq n$. Na forma geral, a decomposição SVD de uma matrix \mathbf{A} pode ser expressa por $\mathbf{A} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T$, incluindo os vetores \mathbf{v}_i (últimas $n - r$ colunas de \mathbf{V}) e \mathbf{u}_i (últimas $m - r$ colunas de \mathbf{U}) que fornecem bases ortonormais para os espaços nulo e nulo à esquerda de \mathbf{A} , respectivamente [54].

juntas de revolução) em diferentes posturas.

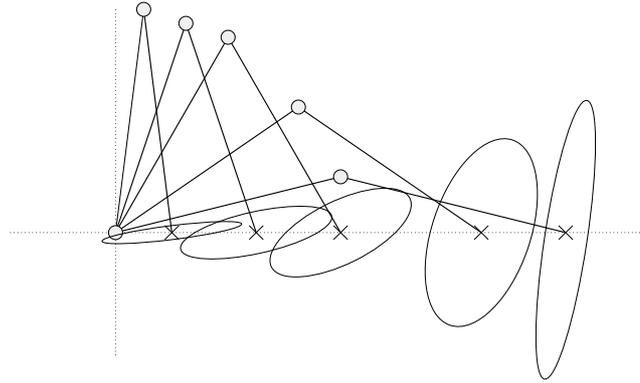


Figura 3.1: Elipsoides de manipulabilidade de velocidade para robô planar.

3.2 Singularidades em robôs antropomórficos

A maioria dos manipuladores existentes são cinematicamente simples, tipicamente constituídos por um braço serial e um punho esférico. Esta escolha é parcialmente motivada pela dificuldade de encontrar soluções para o problema de cinemática inversa no caso geral [28]. Felizmente, a estrutura simples de um manipulador antropomórfico permite formular o problema de cinemática inversa em dois subproblemas, uma vez que a solução para a posição do punho é desacoplada da orientação [55]. Neste contexto, manipuladores antropomórficos constituem uma classe de manipuladores de particular interesse para o trabalho desenvolvido. A estrutura mecânica desta classe de manipuladores é introduzida na Figura 3.2, recebendo este nome devido à sua similaridade com um braço humano.

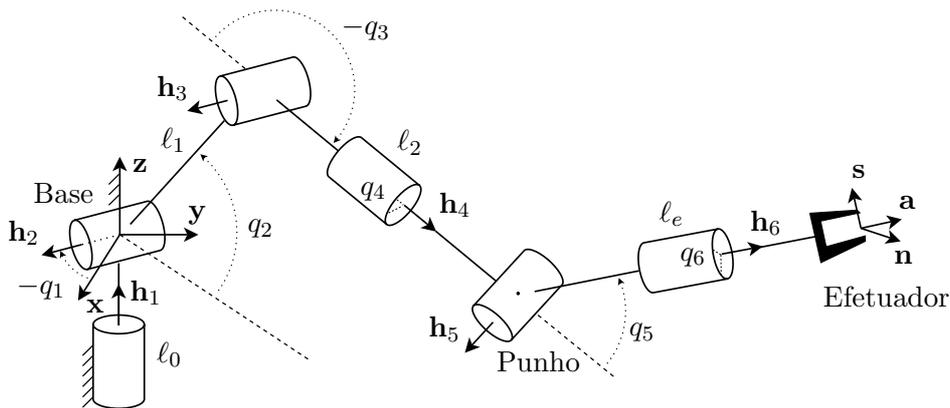


Figura 3.2: Manipulador antropomórfico com punho esférico.

Um *manipulador antropomórfico com punho esférico* é composto por seis juntas de revolução, cada uma adicionando um grau de liberdade ao movimento da estrutura mecânica. Uma característica física importante neste tipo de manipulador é o desacoplamento da posição do punho e da orientação.

Sejam \mathbf{p}_w e \mathbf{R}_{0e} a posição do punho e a orientação do efetuador com respeito ao sistema de coordenadas da base, respectivamente (neste caso, o sistema de coordenadas da base é coincidente com o sistema de coordenadas 0), dadas por

$$\mathbf{p}_w = f_1(q_1, q_2, q_3) \quad (3.6)$$

$$\mathbf{R}_{0e} = f_2(q_1, q_2, q_3, q_4, q_5, q_6), \quad (3.7)$$

onde $f_1 : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ e $f_2 : \mathbb{R}^6 \rightarrow \mathbb{R}^{3 \times 3}$. A relação entre a velocidade linear \mathbf{p}_w e a velocidade angular $\boldsymbol{\omega}_e$ com as velocidades das variáveis de juntas é dada por

$$\begin{bmatrix} \dot{\mathbf{p}}_w \\ \boldsymbol{\omega}_e \end{bmatrix} = \begin{bmatrix} \mathbf{J}_{11} & \mathbf{0}_{3 \times 3} \\ \mathbf{J}_{21} & \mathbf{J}_{22} \end{bmatrix} \begin{bmatrix} \dot{\mathbf{q}}_p \\ \dot{\mathbf{q}}_o \end{bmatrix}, \quad (3.8)$$

onde $\mathbf{q}_p \in \mathbb{R}^3$ corresponde ao vetor de velocidades das três primeiras juntas e $\mathbf{q}_o \in \mathbb{R}^3$ corresponde ao vetor de velocidades das três últimas juntas, ou juntas do punho. Nota-se a partir de (3.8) que o *Jacobiano geométrico* assume uma forma bloco triangular, em que a posição do punho depende apenas das velocidades das três primeiras juntas. Tendo em vista a equação (3.8), o Jacobiano é triangular por blocos e seu determinante pode então ser calculado a partir dos determinantes de cada bloco. Assim,

$$\det(\mathbf{J}_G) = \det(\mathbf{J}_{11}) \det(\mathbf{J}_{22}). \quad (3.9)$$

A igualdade $\det(\mathbf{J}_{11}) = 0$ corresponderá a uma singularidade de braço (singularidade de ombro e de cotovelo), resultante do movimento dos 3 primeiros elos, e $\det(\mathbf{J}_{22}) = 0$ a uma singularidade de punho [56–58]. Deve-se notar que o Jacobiano considerado não fornece a relação entre as velocidades das juntas e as velocidades do efetuador, e sim do punho. Porém, permite o cálculo simplificado das configurações singulares. Na Figura 3.3 são apresentados ambos os tipos de singularidades existentes para o manipulador em questão. Note a existência de infinitas soluções para a cinemática inversa em singularidades de ombro e punho (Figuras 3.3.b e 3.3.c, respectivamente).

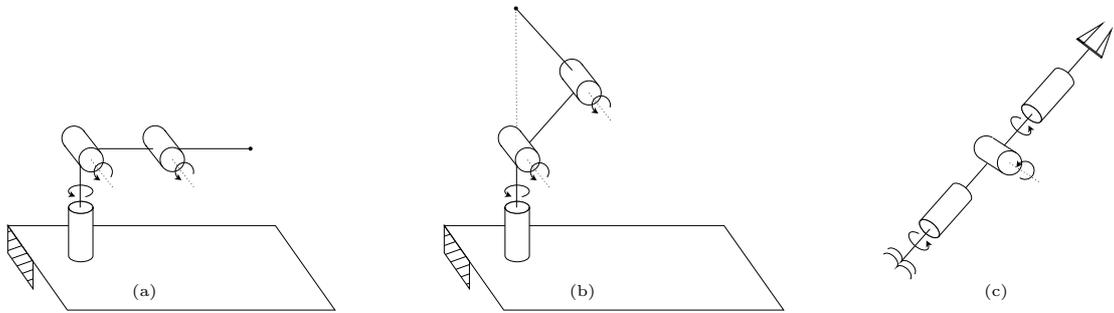


Figura 3.3: Configurações singulares: (a)cotovelo, (b)ombro e (c)punho.

Como visto, as singularidades cinemáticas são configurações que se deseja evitar. Caso contrário, algoritmos alternativos devem ser empregados.

Simulação 3

Considere o manipulador antropomórfico Zebra-ZERO (Apêndice B), e o rastreamento de posição do punho (Trajetória 10) e regulação da orientação ($\mathbf{R}_d = \mathbf{R}_y(\pi/2)$). A seguir, são apresentados resultados obtidos para $\Lambda_p = \Lambda_o = 5 \mathbf{I}$.

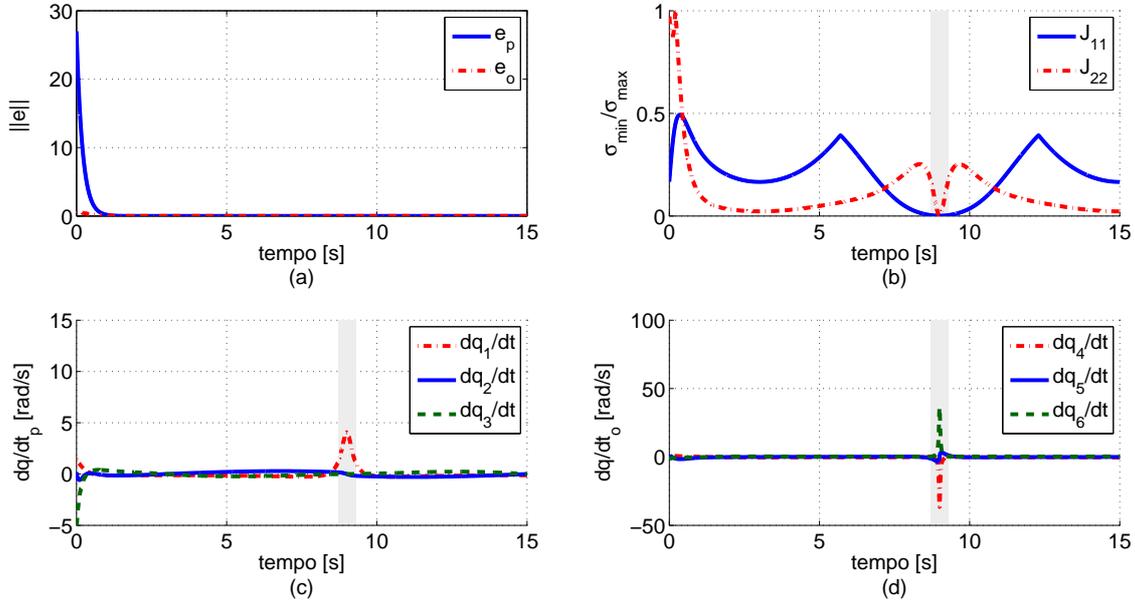


Figura 3.4: Simulação 3: (a) norma do erro; (b) condicionamento; (c) $\dot{\mathbf{q}}_p$ e (d) $\dot{\mathbf{q}}_o$.

Novamente, os erros de postura tendem exponencialmente para zero de acordo com os ganhos Λ_p e Λ_o (Figura 3.4.a). A partir da Figura 3.4.b, nota-se que para $t \approx 9$ s o manipulador atinge a vizinhança de uma singularidade, resultando em picos de velocidade no espaço das juntas (Figuras 3.4.c e 3.4.d).

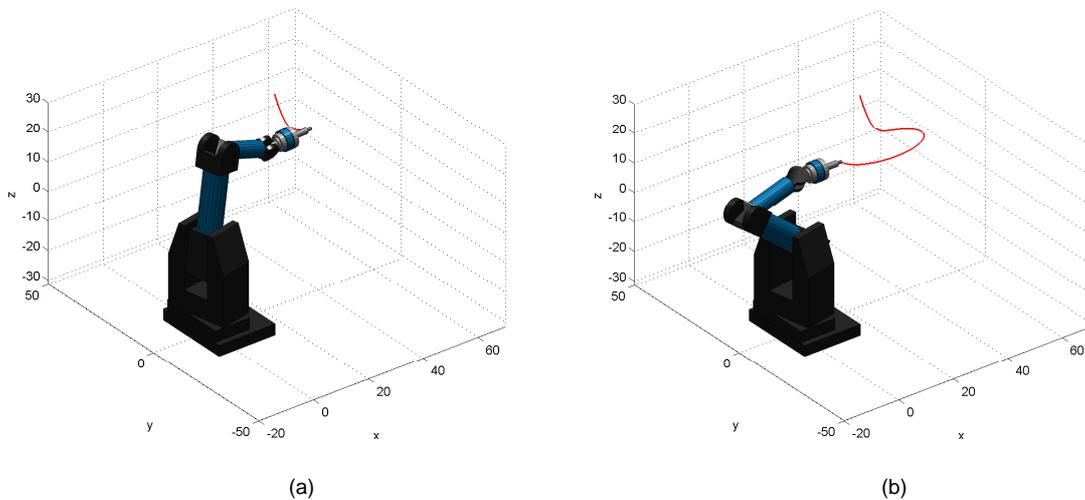


Figura 3.5: Simulação 3: Manipulador Zebra-ZERO em diferentes posturas: (a) afastada ($t \approx 1$ s) e (b) próxima de uma singularidade ($t \approx 9$ s).

3.3 Algoritmo DLS

Uma solução para superar o problema da inversão da equação de cinemática diferencial na vizinhança de uma singularidade é fornecida pelo método DLS (*Damped Least-Squares*) [40]. Nesta proposta, a inversa ou pseudo-inversa da matriz Jacobiana é substituída por

$$\mathbf{J}^* = \mathbf{J}^T(\mathbf{J}\mathbf{J}^T + \delta\mathbf{I})^{-1}, \quad \delta \geq 0, \quad (3.10)$$

onde δ é um fator de amortecimento (ou regularização, como tratado em [59]) que torna a inversão melhor condicionada de um ponto de vista numérico [44]. A solução depende deste fator de amortecimento, que estabelece um compromisso entre a exatidão ($\delta \approx 0$) e a viabilidade da solução ($\delta \gg 0$). Desta forma, a identificação de diferentes métodos de adaptação de δ constitui um tema de pesquisa bastante relevante, como em [40],[41] e [43]. O fator de amortecimento δ pode ser calculado como [40]:

$$\delta = \begin{cases} 0 & , \quad \omega \geq \omega_0, \\ \delta_0 \left(1 - \frac{\omega}{\omega_0}\right) & , \quad \omega < \omega_0, \end{cases} \quad (3.11)$$

onde $\omega = \sqrt{\det(\mathbf{J}(\mathbf{q})\mathbf{J}^T(\mathbf{q}))}$ é a medida de manipulabilidade do manipulador, δ_0 é uma constante que estabelece um fator de escalamento nas configurações singulares e ω_0 é um parâmetro que define o limite da vizinhança de uma configuração singular, sendo escolhido de acordo com a estrutura mecânica do manipulador. Uma forma alternativa, proposta em [43], utiliza δ^2 ao invés de δ . A principal ideia por trás do método é amortecer as velocidades das juntas não-factíveis na vizinhança de uma configuração singular, permitindo ao efetuador desviar da trajetória de referência desejada, estando o desvio na direção e magnitude do movimento [16]. Utilizando (3.10), a solução para a cinemática inversa pode ser calculada como

$$\dot{\mathbf{q}} = \mathbf{J}^T(\mathbf{J}\mathbf{J}^T + \delta\mathbf{I})^{-1}\boldsymbol{\nu}, \quad (3.12)$$

onde $\boldsymbol{\nu} = \dot{\mathbf{x}}_d + \boldsymbol{\Lambda}\mathbf{e}$. A solução (3.12) satisfaz a seguinte condição:

$$\min_{\dot{\mathbf{q}}} \|\boldsymbol{\nu} - \mathbf{J}(\mathbf{q})\dot{\mathbf{q}}\|^2 + \delta \|\dot{\mathbf{q}}\|^2, \quad (3.13)$$

estabelecendo uma ponderação entre as condições de mínimos quadrados e norma mínima, ou seja, considerando tanto a acurácia quanto a viabilidade da solução. A escolha de valores δ pequenos resulta em soluções precisas mas com baixa robustez na vizinhança de singularidades. Por outro lado, a escolha de valores δ elevados resulta em pouca precisão de rastreamento mesmo quando soluções factíveis e precisas são

possíveis [48]. Considerando (3.1), a solução (3.12) proposta pelo algoritmo DLS pode ser reescrita como

$$\dot{\mathbf{q}} = \sum_{i=1}^m \frac{\sigma_i}{\sigma_i^2 + \delta} \mathbf{v}_i \mathbf{u}_i^T \boldsymbol{\nu}. \quad (3.14)$$

Nota-se que para componentes associados a $\sigma_i \gg \sqrt{\delta}$, o fator de amortecimento tem pouca influência e $\sigma_i/(\sigma_i^2 + \delta) \approx 1/\sigma_i$. Por outro lado, na vizinhança de uma singularidade, o menor valor singular σ_r se aproxima de zero e o componente associado a este também tenderá a zero pelo fator σ_r/δ , reduzindo as velocidades de juntas que estão associadas às componentes de $\boldsymbol{\nu}$ na direção \mathbf{v}_r , em que a estrutura apresenta mobilidade reduzida. No caso limite em que $\sigma_i = 0$, tem-se que $\sigma_i/\delta = 0$ e não há atuação nas juntas [43].

3.4 Algoritmo FIK

A solução mais recente para tratar o problema de cinemática inversa na proximidade de configurações singulares é o algoritmo FIK (*Feedback Inverse Kinematics*), apresentado em [47]. O método apresenta uma proposta alternativa que não exige inversão matricial e soluciona o problema ao empregar um laço de realimentação para minimizar a diferença entre as velocidades demandada e atual.

A discrepância entre a velocidade atual $\dot{\mathbf{x}} \in \mathbb{R}^m$ e a velocidade demandada $\boldsymbol{\nu} \in \mathbb{R}^m$ pode ser representada pelo vetor de erro $\mathbf{e}_v \in \mathbb{R}^m$ dado por

$$\mathbf{e}_v = \boldsymbol{\nu} - \mathbf{J}(\mathbf{q})\dot{\mathbf{q}}. \quad (3.15)$$

Considere o laço de realimentação mostrado na Figura 3.6, onde $\mathbf{K}(s) \in \mathbb{R}^{n \times m}$ é uma função de transferência matricial completa. Existe uma forma adaptativa para $\mathbf{K}(s)$ que reduz o erro em (3.15) para um valor arbitrariamente pequeno, resultando na seguinte solução para o problema de cinemática inversa:

$$\dot{\mathbf{q}} = \mathbf{K}(s)(\mathbf{J}\mathbf{K}(s) + \mathbf{I})^{-1}\boldsymbol{\nu}. \quad (3.16)$$

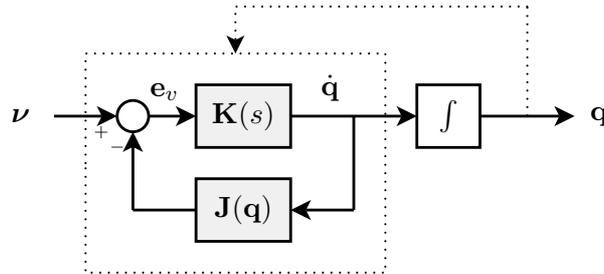


Figura 3.6: Diagrama de blocos FIK.

A matriz $\mathbf{K}(s)$ ótima apresenta a seguinte forma adaptativa:

$$\mathbf{K}(s) = \mathbf{J}^T(\mathbf{q})\mathcal{G}(s\mathbf{I} - \mathcal{A})^{-1}, \quad (3.17)$$

onde $\mathcal{G} \in \mathbb{R}^{m \times m}$ é uma matriz positiva definida simétrica e $\mathcal{A} \in \mathbb{R}^{m \times m}$ é uma matriz negativa definida diagonal. A forma ótima para $\mathbf{K}(s)$ é obtida a partir do seguinte problema de minimização:

$$\min_{\mathbf{K}(s)} \left\| \begin{pmatrix} \boldsymbol{\omega}_1(s)(\mathbf{J}_0\mathbf{K}(s) + \mathbf{I})^{-1} \\ \boldsymbol{\omega}_2(\mathbf{J}_0\mathbf{K}(s) + \mathbf{I})^{-1} \end{pmatrix} \right\|, \quad (3.18)$$

onde \mathbf{J}_0 é calculada a partir de um vetor fixo de variáveis não nulas de juntas, sendo utilizada para mapear as velocidades das juntas no espaço cartesiano. O termo $\boldsymbol{\omega}_1(s) \in \mathbb{R}^{m \times m}$ determina a largura de banda da lei de cinemática inversa, sendo obtido a partir de restrições dinâmicas do manipulador e da trajetória de referência, e $\boldsymbol{\omega}_2(s) \in \mathbb{R}^{m \times m}$ determina restrições para os atuadores, estabelecendo um limite superior para as velocidades $\dot{\mathbf{q}}$ desenvolvidas. Para a adaptação de $\mathbf{K}(s)$, a equação (3.17) pode ser reescrita no espaço de estados [48]:

$$\mathbf{K}(s) : \begin{cases} \dot{\mathbf{z}} = \mathcal{A}\mathbf{z} + \mathbf{e}_v, \\ \dot{\mathbf{q}} = (\mathcal{D}_{12}^T\mathcal{D}_{12})^{-1}\mathbf{J}(\mathbf{q})\mathcal{B}^T\mathcal{P}\mathcal{B}\mathbf{z}, \end{cases} \quad (3.19)$$

onde $\mathcal{G} = (\mathcal{D}_{12}^T\mathcal{D}_{12})^{-1}\mathcal{B}^T\mathcal{P}\mathcal{B}$ e $\mathbf{z} \in \mathbb{R}^m$ corresponde ao vetor de estados do controlador. A matriz \mathcal{P} é baseada na escolha de \mathbf{J}_0 e as matrizes \mathcal{A}, \mathcal{B} e \mathcal{D}_{12} são matrizes diagonais dependentes dos parâmetros de desempenho, portanto $\boldsymbol{\omega}_1$ e $\boldsymbol{\omega}_2$. A matriz de transferência $\boldsymbol{\omega}_1(s)$ é diagonal e dada por

$$\boldsymbol{\omega}_1(s) = \frac{b}{s + \alpha}\mathbf{I}, \quad (3.20)$$

onde b determina a largura de banda da lei de cinemática inversa e α é o ganho de saturação do integrador em baixas frequências. Assim, \mathcal{A} pode ser escrita como

$$\mathcal{A} = -\alpha\mathbf{I}. \quad (3.21)$$

O termo $\boldsymbol{\omega}_2$ está relacionado às restrições nos atuadores do manipulador e, dada uma velocidade máxima de junta ω , pode ser escrito como

$$\boldsymbol{\omega}_2 = \frac{1}{\omega}\mathbf{I}. \quad (3.22)$$

Nota-se que tanto $\boldsymbol{\omega}_1$ e $\boldsymbol{\omega}_2$ podem ser especificados independentemente em cada direção m ou grau de liberdade n , respectivamente.

Apesar de utilizar uma abordagem bastante diferente para solucionar o problema de cinemática inversa, o algoritmo FIK apresenta algumas similaridades com o algoritmo DLS apresentado anteriormente. Considerando

$$\mathbf{Q} = \mathcal{G}(s\mathbf{I} - \mathcal{A})^{-1}, \quad (3.23)$$

a solução apresentada em (3.16) se torna

$$\dot{\mathbf{q}} = \mathbf{J}^T \mathbf{Q} (\mathbf{J} \mathbf{J}^T \mathbf{Q} + \mathbf{I})^{-1} \boldsymbol{\nu}, \quad (3.24)$$

apresentando estrutura semelhante à solução (3.12), exceto pela diferença entre os termos δ e \mathbf{Q} . A solução proposta pelo método FIK utiliza uma função dinâmica de mapeamento $\mathbf{Q}(s)$ e considera $\delta = 1$, não exigindo portanto o cálculo dos valores singulares da matriz Jacobiana. Em comparação com outros métodos baseados na transposta da matriz Jacobiana, o método FIK proporciona robustez a singularidades devido à forma não-diagonal e dinâmica de $\mathbf{Q}(s)$. As velocidades das juntas $\dot{\mathbf{q}}$ estão relacionadas às velocidades demandadas no espaço cartesiano através da função de controle $\mathbf{J}^T \mathbf{Q} (\mathbf{J} \mathbf{J}^T \mathbf{Q} + \mathbf{I})^{-1}$ e, uma vez que a solução é obtida através de um laço de realimentação, não é necessária a inversão da matriz Jacobiana. Assim, o custo computacional do método é reduzido consideravelmente. De (3.24), a estabilidade interna da solução FIK é garantida desde que [60, 61]

$$\det(\mathbf{J} \mathbf{J}^T \mathbf{Q} + \mathbf{I}) \neq 0, \quad (3.25)$$

ou seja, para todas as configurações \mathbf{q} , a condição $\sigma(\mathbf{J} \mathbf{J}^T \mathbf{Q}) \neq -1$ deve ser verificada. Uma vez que $\mathbf{Q} = \mathbf{Q}^T \geq \mathbf{0}$, tem-se que $\sigma(\mathbf{J} \mathbf{J}^T \mathbf{Q}) \geq 0$ [62, 63]. Assim, a condição (3.25) é sempre satisfeita e não depende de \mathbf{q} .

O algoritmo FIK permite a utilização de um vetor de velocidades arbitrárias das juntas $\dot{\mathbf{q}}_0$ para satisfazer uma restrição adicional ao sistema redundante [47], similar a (2.51). O diagrama de blocos aumentado é apresentado na Figura 3.7.

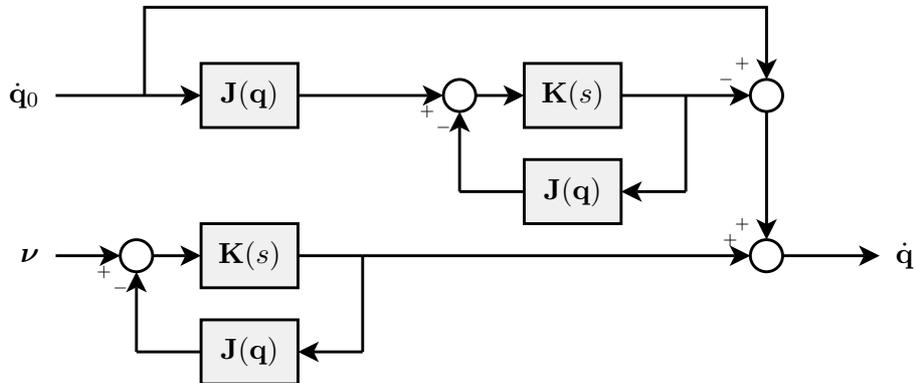


Figura 3.7: Diagrama de blocos FIK com restrição adicional.

Simulação 4

Considere o manipulador planar redundante apresentado na Figura 2.9 e, inicialmente, uma trajetória livre e afastada de singularidades (Trajetória 2). No que se segue são apresentados os resultados obtidos por simulação para os métodos DLS e FIK. Os ganhos de controle e parâmetros de simulação são obtidos de [40] e [47].

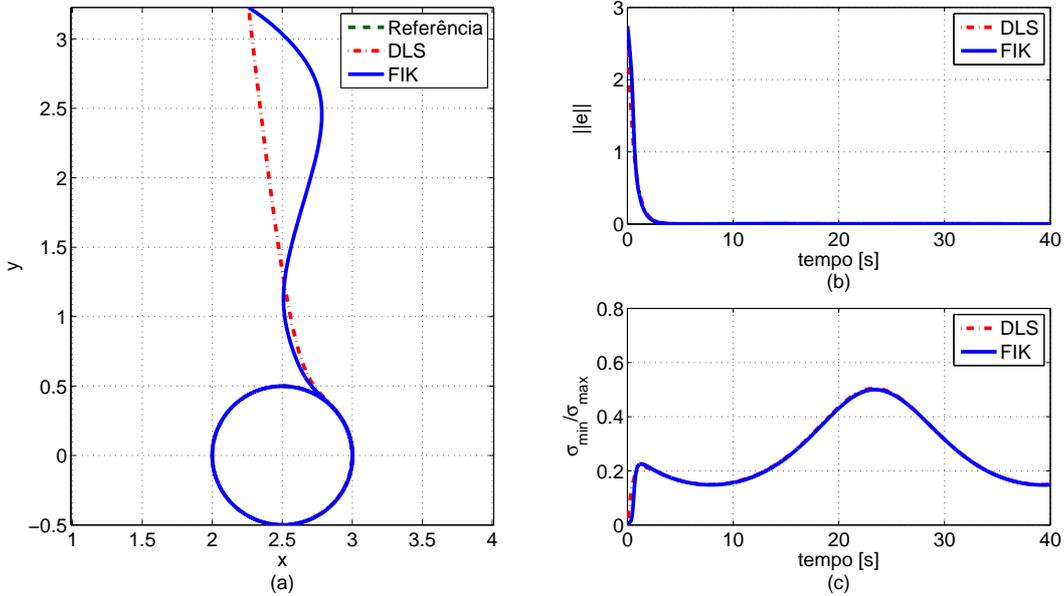


Figura 3.8: Simulação 4: (a) trajetórias; (b) norma do erro; e (c) condicionamento.

Ambos os algoritmos permitem o rastreamento da trajetória desejada. Para o algoritmo DLS, tem-se que $\delta = 0, \forall t$, uma vez que a trajetória está afastada da região de pouca manipulabilidade definida por ω_o , e $\lim_{t \rightarrow \infty} \mathbf{e}(t) = \mathbf{0}$.

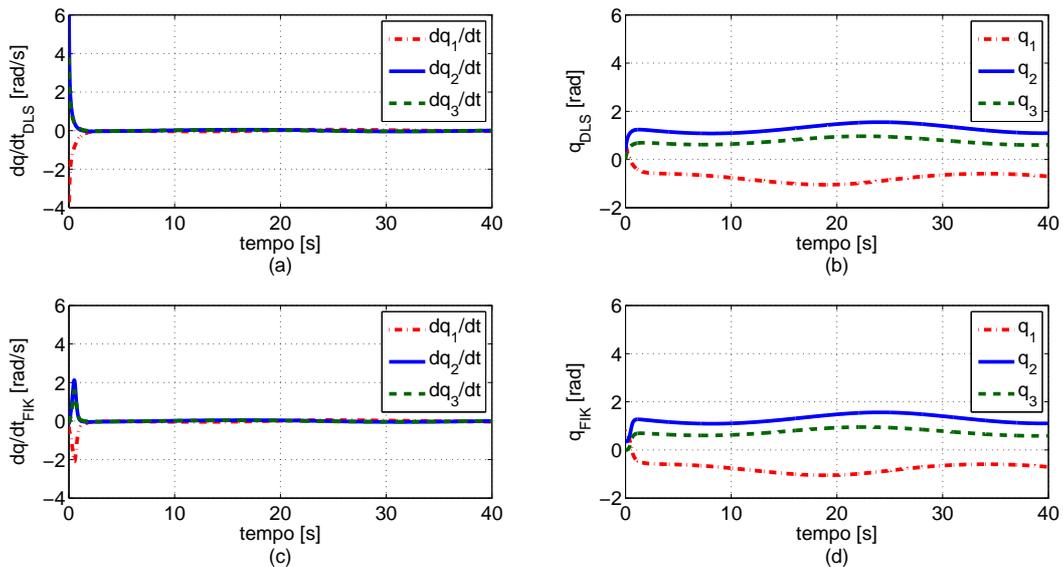


Figura 3.9: Simulação 4: variáveis (a) $\dot{\mathbf{q}}_{DLS}$; (b) \mathbf{q}_{DLS} ; (c) $\dot{\mathbf{q}}_{FIK}$; e (d) \mathbf{q}_{FIK} .

Simulação 5

Considere agora uma trajetória de referência que se aproxima de uma configuração singular (Trajetória 3). A seguir, são apresentados os resultados de simulação obtidos para os mesmos parâmetros de simulação considerados anteriormente.

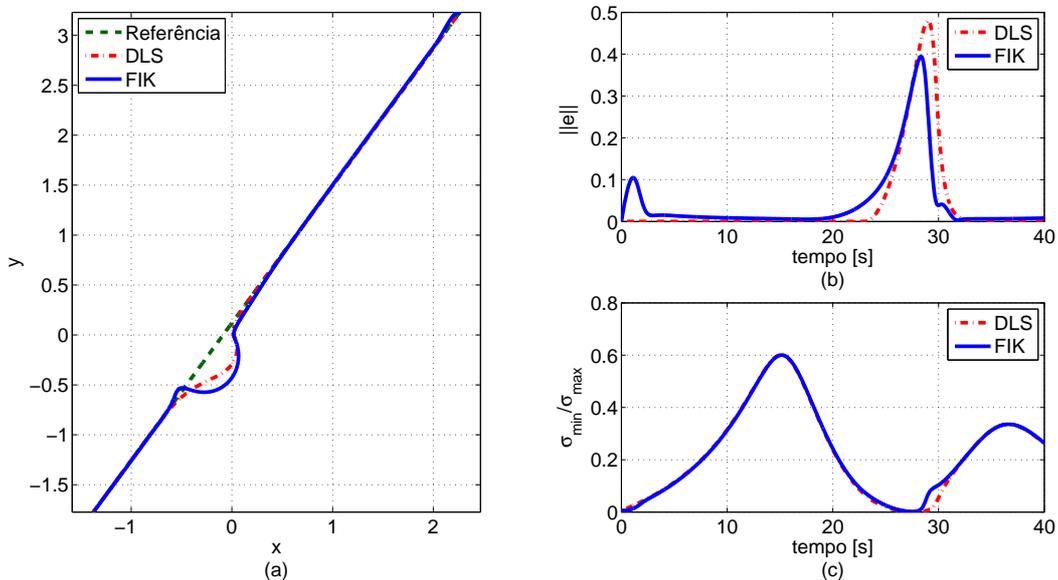


Figura 3.10: Simulação 5: (a) trajetórias; (b) norma do erro; e (c) condicionamento.

A partir da Figura 3.10.c, nota-se que, para $t \approx 26$ s, a trajetória desejada se aproxima da origem e os algoritmos conseguem evitar esta configuração singular.

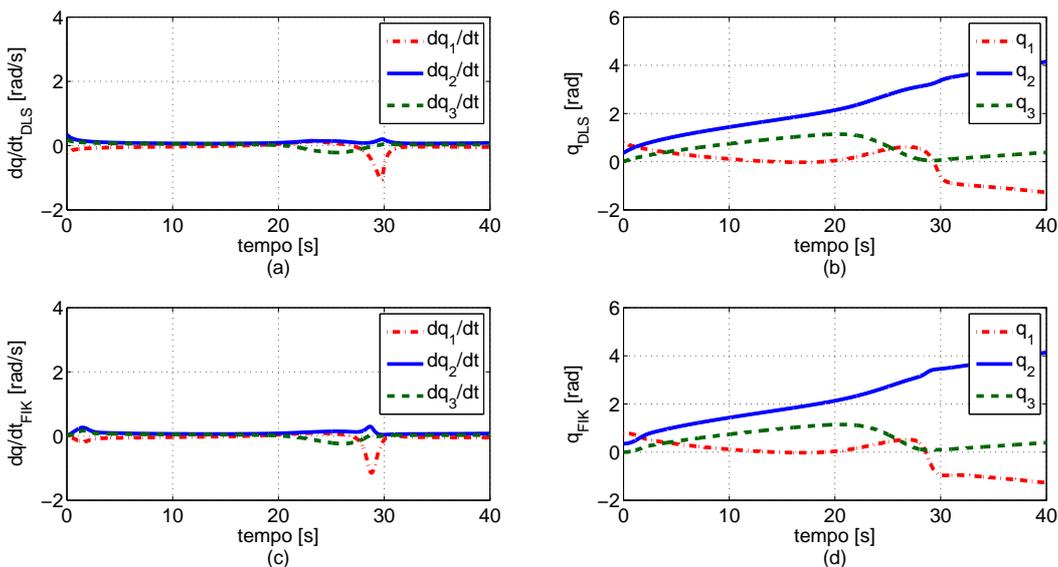


Figura 3.11: Simulação 5: variáveis (a) $\dot{\mathbf{q}}_{DLS}$; (b) \mathbf{q}_{DLS} ; (c) $\dot{\mathbf{q}}_{FIK}$; e (d) \mathbf{q}_{FIK} .

Como visto nas Figuras 3.11.a e 3.11.c, ambos algoritmos mantêm as velocidades $\dot{\mathbf{q}}$ das juntas em uma região adequada, abaixo de 1.2 rad/s .

Simulação 6

Neste caso, considera-se uma trajetória que se inicia ($\mathbf{q}(0) = [\frac{\pi}{2} \ 0 \ 0]^T rad$) e que passa por uma configuração singular (Trajetória 4).

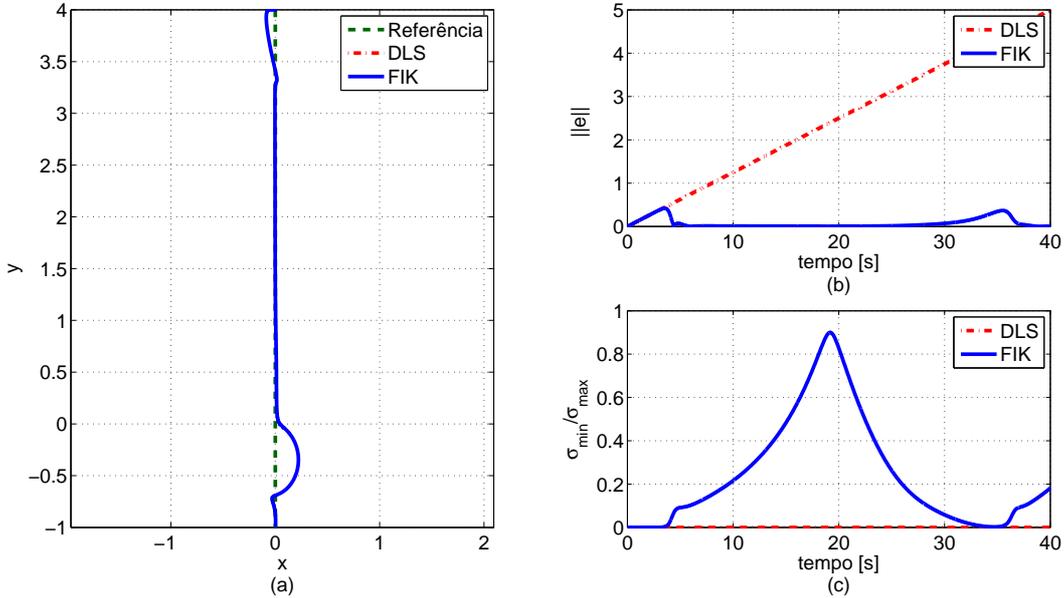


Figura 3.12: Simulação 6: (a) trajetórias; (b) norma do erro; e (c) condicionamento.

Uma vez que a postura inicial configura uma singularidade cinemática, todo sinal de velocidade na direção degenerada será anulado pelo algoritmo DLS e este se torna incapaz de alterar a configuração do manipulador. Por outro lado, o algoritmo FIK permite alterar a configuração inicial do manipulador, dado o formato não-diagonal da matriz \mathcal{G} , e mantém o erro de rastreamento pequeno ao longo da trajetória.

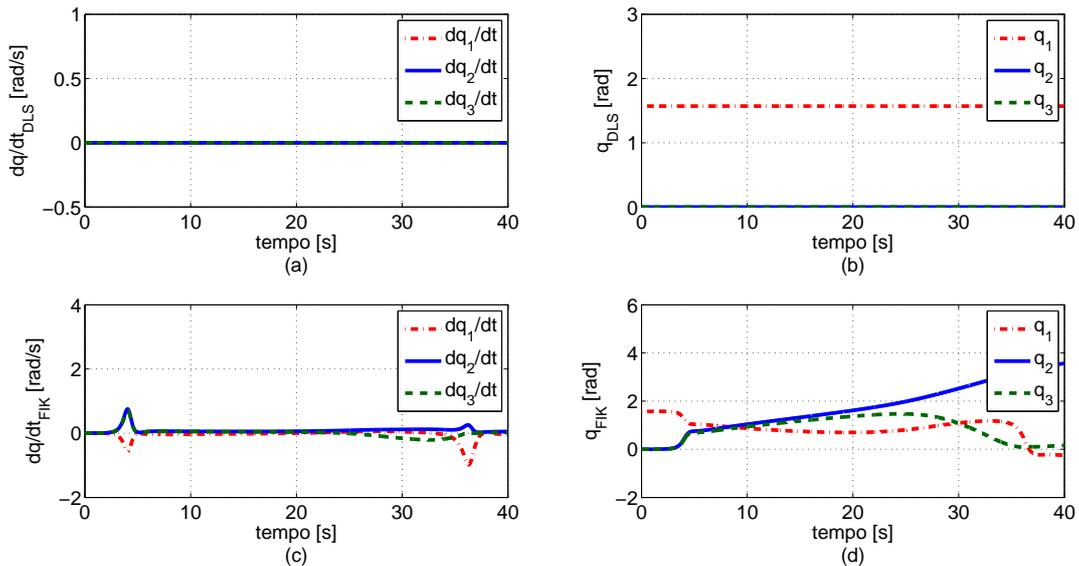


Figura 3.13: Simulação 6: variáveis (a) $\dot{\mathbf{q}}_{DLS}$; (b) \mathbf{q}_{DLS} ; (c) $\dot{\mathbf{q}}_{FIK}$; e (d) \mathbf{q}_{FIK} .

Simulação 7

Considere novamente a Trajetória 4. Para uma configuração inicial não-singular $\mathbf{q}(0) = [\frac{\pi}{2} \ \frac{1}{10} \ \frac{1}{10}]^T rad$, o desempenho dos algoritmos é novamente comparado.

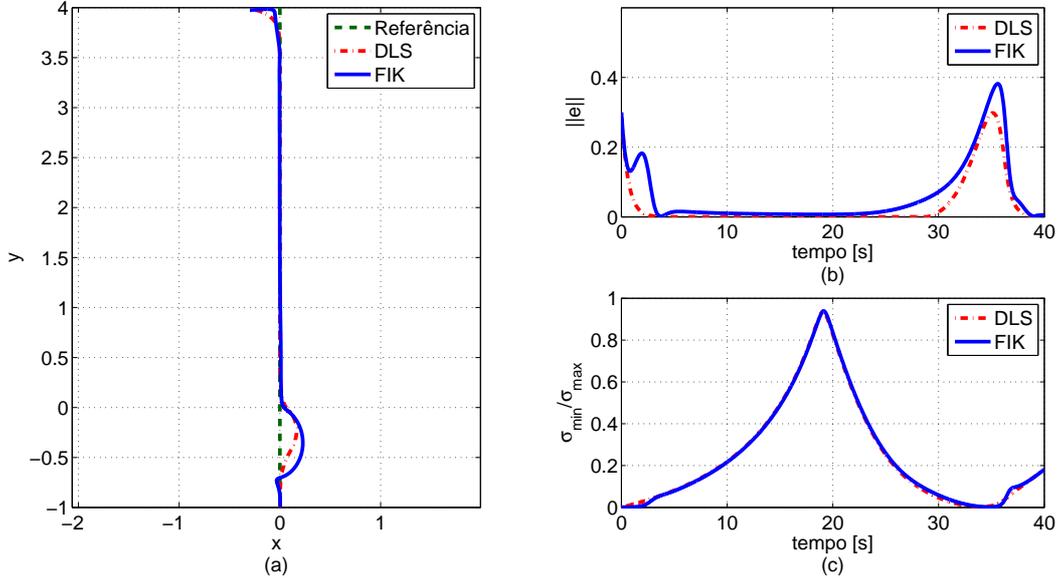


Figura 3.14: Simulação 7: (a) trajetórias; (b) norma do erro; e (c) condicionamento.

Nota-se que, para uma configuração inicial não-singular, o algoritmo DLS é capaz de rastrear a trajetória com erro nulo quando afastado da singularidade e permite, na vizinhança desta, o desvio da trajetória (Figuras 3.14.a e 3.14.b).

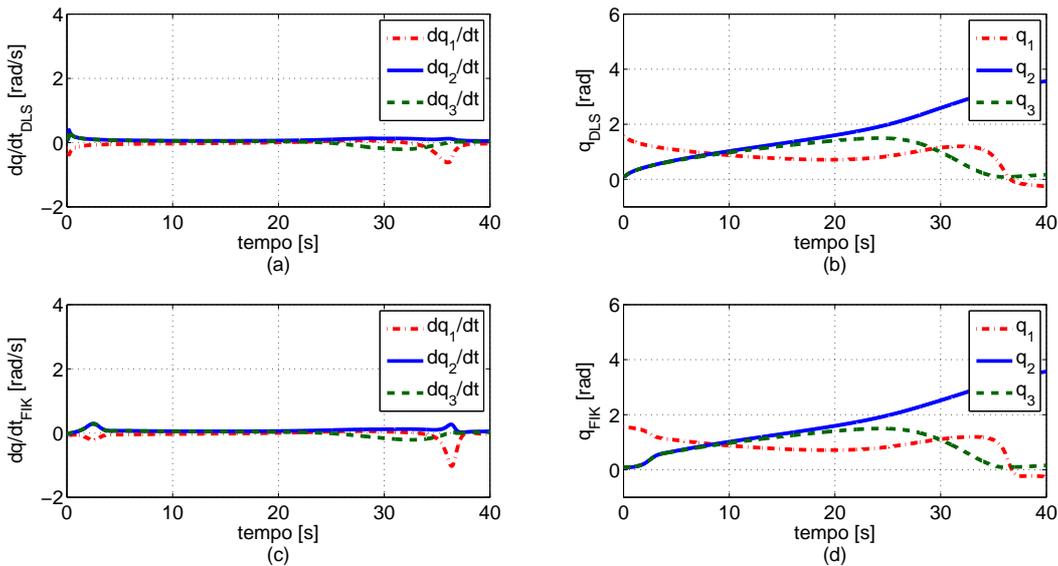


Figura 3.15: Simulação 7: variáveis (a) $\dot{\mathbf{q}}_{DLS}$; (b) \mathbf{q}_{DLS} ; (c) $\dot{\mathbf{q}}_{FIK}$; e (d) \mathbf{q}_{FIK} .

Novamente, os algoritmos DLS e FIK mantêm o vetor de velocidades $\dot{\mathbf{q}}$ em uma faixa adequada, abaixo de 0.7 rad/s , como visto nas Figuras 3.15.a e 3.15.c.

Capítulo 4

Inversa filtrada

No Capítulo 3 foram apresentados dois algoritmos alternativos e consolidados para solucionar o problema de cinemática inversa de manipuladores robóticos na presença de configurações singulares. O método DLS propõe a utilização de um termo de amortecimento, estabelecendo um compromisso entre a exatidão e a viabilidade da solução. O algoritmo FIK, por sua vez, emprega um laço de realimentação para minimizar a diferença entre as velocidades atual e desejada no espaço operacional, não exigindo inversões. Neste capítulo, é introduzido o algoritmo proposto para solucionar tal problema. A ideia principal por trás do método consiste em calcular a inversa dinamicamente. A análise tem início, portanto, com o caso mais simples, escalar, sendo natural sua extensão para o caso multivariável.

4.1 Algoritmo proposto

Considere uma planta SISO de 1ª ordem descrita por

$$\dot{y} = k(t) u, \quad (4.1)$$

onde $u \in \mathcal{L}_\infty$ é a entrada escalar da planta ou a variável de controle do sistema, y é a saída da planta e $k(t)$ é uma função escalar variante no tempo. Supondo que o objetivo de controle seja rastrear uma trajetória de referência desejada $y_d(t)$, uma lei de controle u que garanta a estabilidade assintótica do erro de rastreamento $e := y_d - y$ é dada por

$$u = k^{-1}(t) \nu, \quad \nu = \dot{y}_d + \lambda e, \quad (4.2)$$

onde $\lambda > 0$. Como foi visto, o erro de rastreamento tenderá exponencialmente a 0 de acordo com o valor do ganho λ utilizado e a presença do termo *feedforward* \dot{y}_d garante a manutenção do erro em 0 independente do tipo de sinal y_d (constante

ou variante no tempo). Agora, considere uma lei de controle u que não utilize a inversa, computada instantaneamente como $1/k(t)$, mas uma função $\theta(t)$ atualizada dinamicamente tal que

$$k\theta \rightarrow 1. \quad (4.3)$$

Para estabelecer uma dinâmica adequada para $\theta(t)$, satisfazendo (4.3), introduz-se o seguinte sinal de erro não-linear

$$S = k\theta - 1. \quad (4.4)$$

Considera-se a função de Lyapunov $2V(S) = S^2$ cuja derivada temporal é dada por

$$\dot{V}(S) = S \dot{S} = S(\dot{k}\theta + k\dot{\theta}). \quad (4.5)$$

Em vista de (4.5), escolhe-se a seguinte lei de atualização paramétrica

$$\dot{\theta} = -\beta S k, \quad (4.6)$$

onde $\beta > 0$, e como resultado obtém-se

$$\dot{V}(S) = S \dot{k}\theta - \beta S^2 k^2. \quad (4.7)$$

Note que, $\dot{V}(S)$ depende de \dot{k} e, devido ao termo $S\dot{k}\theta$, não tem sinal definido. Neste ponto, porém, pode-se investigar o comportamento do algoritmo para o caso particular em que $\dot{k} \equiv 0$. Neste caso, $\dot{V}(S) \leq 0$ e a análise é consideravelmente simplificada. A lei de atualização (4.6) pode ser reescrita como

$$\dot{\theta} = -\beta(k\theta - 1)k = -\beta k^2\theta + \beta k, \quad (4.8)$$

e, como k é constante, utiliza-se o operador diferencial s para obter

$$(s + \beta k^2)\theta = \beta k. \quad (4.9)$$

Para $k \neq 0$, pode-se explicitar θ como

$$\theta = \frac{\beta k}{s + \beta k^2} = \frac{1}{\tau s + 1} \frac{1}{k}, \quad (4.10)$$

onde $\tau = 1/\beta k^2$. Deste modo, θ pode ser interpretado como a saída de um filtro linear onde a entrada é a inversa $1/k$, isto é, θ tende exponencialmente para $1/k$ com rapidez determinada pela constante de tempo τ . Por isso, o sinal θ será denominado de *inversa filtrada* de k . Quanto menores são os valores de k e β , maior é a constante τ e, conseqüentemente, mais lento é o filtro.

4.2 Propriedades da inversa filtrada

Uma propriedade importante do algoritmo proposto é que para $k \equiv 0$ tem-se que $\dot{\theta} \equiv 0$ e, como consequência, $\theta(t) = \theta(0)$. A Figura 4.1 ilustra o plano $\theta \times k$ com o lugar geométrico de $S = 0$ e algumas trajetórias para diferentes condições iniciais $\theta(0)$. De acordo com a figura pode-se afirmar que para qualquer k constante a saída $\theta(t)$ é limitada.

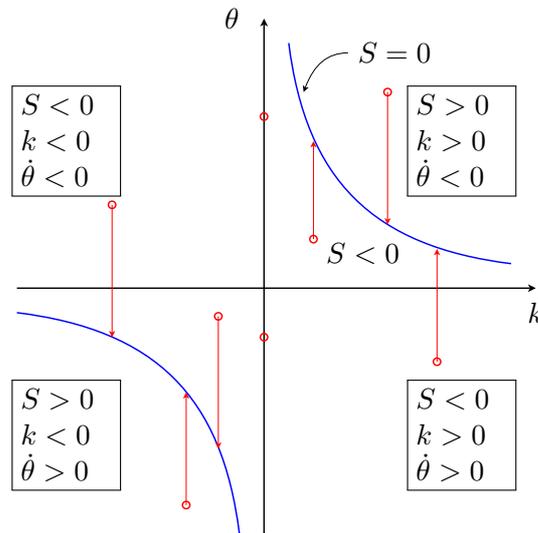


Figura 4.1: Plano $\theta \times k$ (diferentes condições iniciais).

Considere agora o caso de uma função $k(t)$ variante no tempo. A partir da figura é evidente que a única possibilidade para $\theta(t)$ crescer sem limites está na região delimitada por $-1 < S < 0$ (Figura 4.2). Note que, quanto mais próximo de zero é o valor de k , maior pode ser o valor de θ .

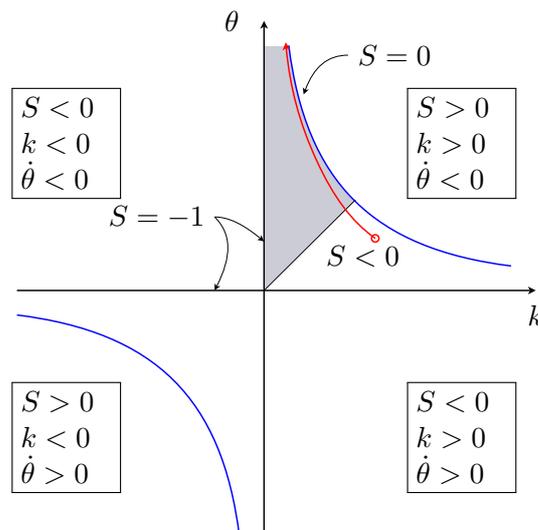


Figura 4.2: Plano $\theta \times k$ (escape).

Entretanto, na região sombreada no primeiro quadrante, pode-se verificar que $0 < |k| < 1$ e $0 < |\dot{\theta}| < \beta$, implicando que θ não apresenta escape em tempo finito. Considera-se agora o caso em que k tende para zero. A seguir será construído um exemplo mostrando que, neste caso, θ pode crescer sem limites mas de maneira muito lenta. Considere uma trajetória no interior da região sombreada satisfazendo $V = S^2 = f(t)$, onde $f(t)$ é uma função contínua tal que $0 < f(t) < 1, \forall t$. Esta constitui uma análise do pior cenário uma vez que $\dot{\theta} > 0, \forall t$. Como $-1 < S < 0$ e $k > 0$, segue de (4.4) que

$$\theta = \frac{1 - \sqrt{f}}{k}, \quad (4.11)$$

Por sua vez, como $\dot{V} = \dot{f}$, a partir de (4.5) e sabendo que $S \neq 0$ tem-se:

$$\dot{k} = \frac{\dot{f} + \beta f k^2}{S \theta}. \quad (4.12)$$

Então, substituindo (4.11) em (4.12) obtém-se:

$$\dot{k} = \frac{\dot{f} + \beta f k^2}{f - \sqrt{f}} k. \quad (4.13)$$

Para uma função constante $f(t) = c$, com $0 < c < 1$, a equação (4.13) assume a forma:

$$\dot{k} = -a k^3, \quad a = \frac{\beta \sqrt{c}}{1 - \sqrt{c}}. \quad (4.14)$$

Portanto, a solução de (4.14) é dada por:

$$k(t) = k(0) \sqrt{\frac{1}{2at k^2(0) + 1}}, \quad (4.15)$$

que converge para zero mais lentamente que qualquer exponencial. As propriedades do algoritmo (4.6) podem ser resumidas como:

(P1) $k \equiv 0 \Rightarrow \theta(t) = \theta(0)$.

(P2) k constante $\Rightarrow \theta \in \mathcal{L}_\infty$.

(P3) $k \in \mathcal{L}_\infty \Rightarrow \dot{\theta} \in \mathcal{L}_\infty$.

(P4) $\theta(t)$ não possui escape em tempo finito.

Como visto, a ideia do algoritmo é simples: calcular a inversa dinamicamente. A utilização de apenas um ganho para atualização de θ surge como uma facilidade de sintonia do método e a condição verificada para a aplicação do método é que k não tenda para 0 mais lentamente do que uma exponencial.

Simulação 8

Considere os ganhos $k_1 = 2$, $k_2 = 0.5$, $k_3 = 0$ e $k_4(t) = 2e^{-0.5t}$. Na Figura 4.3, são apresentados os sinais θ_i obtidos para $\beta = 5$ e $\theta_i(0) = 1$.

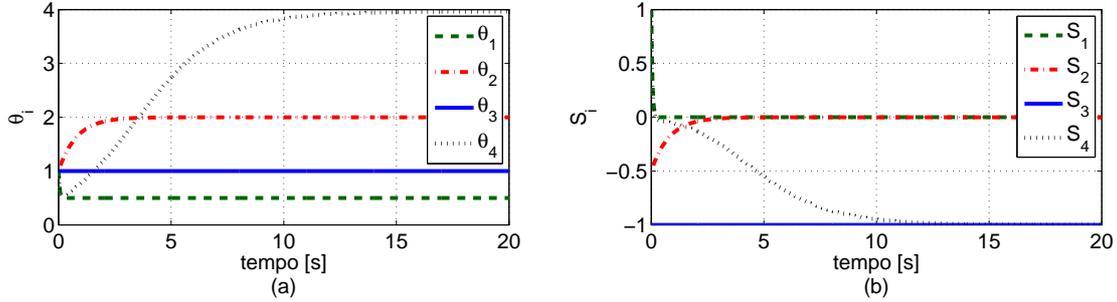


Figura 4.3: Simulação 8: evolução de (a) θ e (b) S para diferentes ganhos k .

Simulação 9

Considere agora um ganho variante no tempo $k(t) = \sin(0.4\pi t)$, $\theta(0) = 0$ e $\beta = \{10, 100, 1000\}$. Como visto nas Figuras 4.4.a e 4.4.b, a medida que o ganho β aumenta, menor é a constante de tempo do filtro e a variável θ atinge valores mais elevados para k próximo ou igual a zero. A Figura 4.4.c mostra o plano $\theta \times k$ para os diferentes ganhos utilizados e, uma vez que $\dot{k} < \infty$, é possível verificar que $\partial\theta/\partial k = 0$ para $S = 0$ e $k = 0$.

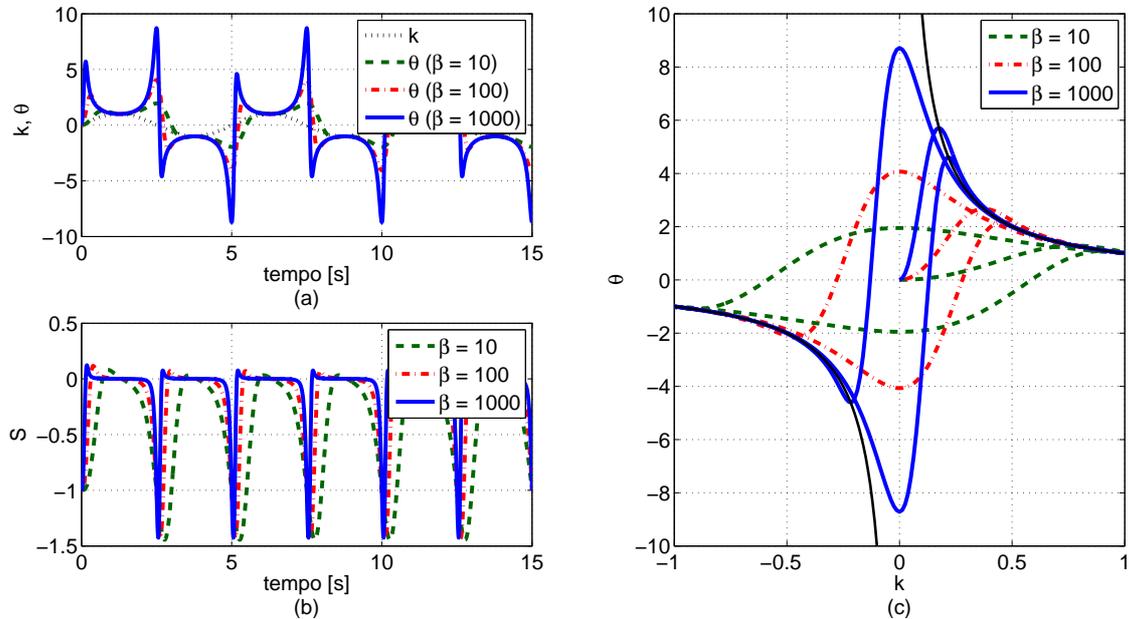


Figura 4.4: Simulação 9: variáveis (a) k e θ ; (b) S ; e (c) plano $\theta \times k$.

Os resultados obtidos para o rastreamento de $y_d = 4 + \cos(0.2\pi t)$, com ganho $\lambda = 10$ e $y(0) = 5$, são apresentados na Figura 4.5. Nota-se que quanto maior o ganho de atualização utilizado, melhor será o desempenho.

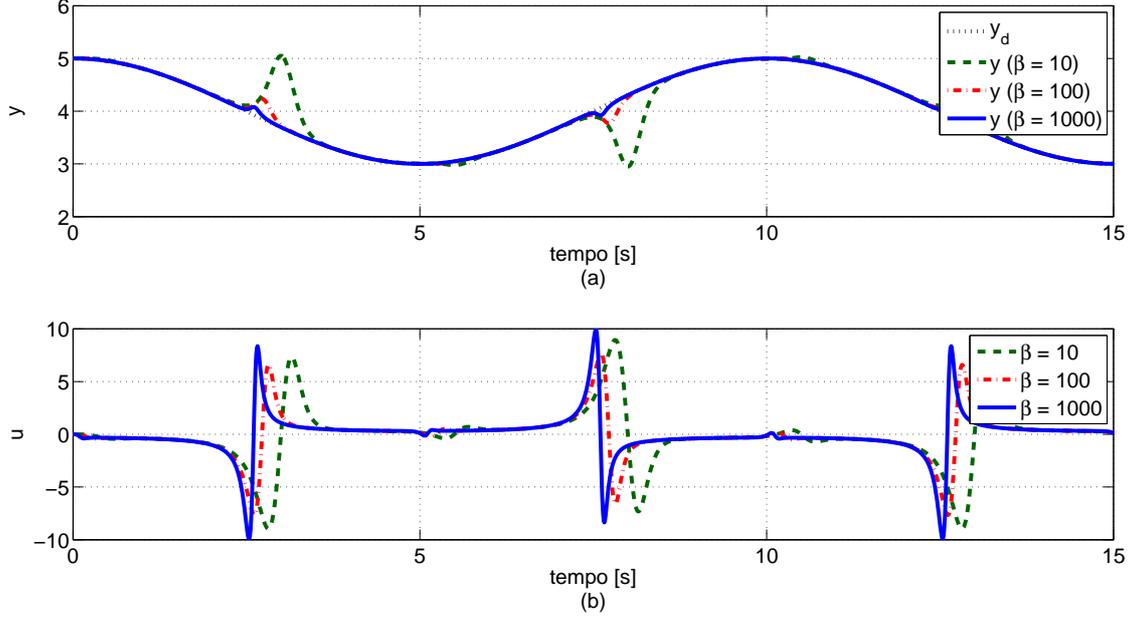


Figura 4.5: Simulação 9: variáveis (a) y e (b) u para diferentes ganhos β .

4.3 Inversa filtrada de matrizes

O algoritmo (4.6) pode ser generalizado para inverter matrizes. Considere a seguinte planta MIMO de 1ª ordem descrita por

$$\dot{\mathbf{y}} = \mathbf{K}(t) \mathbf{u}, \quad (4.16)$$

onde $\mathbf{u} \in \mathbb{R}^n$ é a entrada da planta ou a variável de controle do sistema, $\mathbf{y} \in \mathbb{R}^m$ é a saída da planta, $\mathbf{K}(t) \in \mathbb{R}^{m \times n}$ é uma função matricial variante no tempo e $m \leq n$. Supondo que o objetivo de controle seja rastrear uma trajetória desejada $\mathbf{y}_d(t)$, uma lei de controle \mathbf{u} que garanta a estabilidade assintótica do erro de rastreamento $\mathbf{e} := \mathbf{y}_d - \mathbf{y}$ é dada por

$$\mathbf{u} = \mathbf{K}^\dagger(t) \boldsymbol{\nu}, \quad \boldsymbol{\nu} = \dot{\mathbf{y}}_d + \boldsymbol{\Lambda} \mathbf{e}, \quad (4.17)$$

onde $\mathbf{K}^\dagger(t) \in \mathbb{R}^{n \times m}$ é a pseudo-inversa à direita da matriz $\mathbf{K}(t)$ e $\boldsymbol{\Lambda} = \boldsymbol{\Lambda}^T > 0$. Novamente, considera-se a substituição da pseudo-inversa por uma matriz $\boldsymbol{\Theta}(t)$ atualizada dinamicamente. Assim, para estabelecer uma dinâmica adequada para $\boldsymbol{\Theta}(t)$, introduz-se os seguintes sinais de erro não-lineares à direita

$$\mathbf{S}_r = \mathbf{K}\boldsymbol{\Theta} - \mathbf{I} \in \mathbb{R}^{m \times m}, \quad (4.18)$$

e à esquerda

$$\mathbf{S}_\ell = \boldsymbol{\Theta}\mathbf{K} - \mathbf{I} \in \mathbb{R}^{n \times n}. \quad (4.19)$$

4.3.1 Matriz de erro à direita

Considere inicialmente o erro (4.18) e a função de Lyapunov $2V_r = \text{tr}(\mathbf{S}_r^T \mathbf{S}_r)$, onde $\text{tr}(\cdot)$ corresponde à função traço. A partir de (4.18), a derivada temporal de V_r ao longo das trajetórias do sistema é dada por

$$2\dot{V}_r = \text{tr}(\dot{\mathbf{S}}_r^T \mathbf{S}_r + \mathbf{S}_r^T \dot{\mathbf{S}}_r) = 2 \text{tr}(\mathbf{S}_r^T \dot{\mathbf{K}} \Theta) + 2 \text{tr}(\mathbf{S}_r^T \mathbf{K} \dot{\Theta}). \quad (4.20)$$

Em vista de (4.20), escolhe-se a seguinte lei de atualização

$$\dot{\Theta} = -\Gamma \mathbf{K}^T \mathbf{S}_r, \quad (4.21)$$

onde $\Gamma = \Gamma^T > 0$ é uma matriz de ganho de atualização. Como resultado, tem-se

$$\dot{V}_r = \text{tr}(\mathbf{S}_r^T \dot{\mathbf{K}} \Theta) - \underbrace{\text{tr}(\mathbf{S}_r^T \mathbf{K} \Gamma \mathbf{K}^T \mathbf{S}_r)}_{\leq 0}, \quad (4.22)$$

que depende de $\dot{\mathbf{K}}$. Para $\dot{\mathbf{K}} \equiv \mathbf{0}$, tem-se que $\dot{V}_r \leq 0$. Neste caso, $\dot{\Theta} = \mathbf{0}$ e $\dot{V}_r = 0$ quando

$$\mathbf{K}^T \mathbf{S}_r = \mathbf{0}, \quad (4.23)$$

isto é, quando as colunas de \mathbf{S}_r pertencerem ao espaço nulo à esquerda de \mathbf{K} . Para uma matriz \mathbf{K} tal que $Nul(\mathbf{K}^T) = \{\mathbf{0}\}$, a equação $\mathbf{K} \Theta = \mathbf{I}_m$ tem ao menos uma solução Θ e a igualdade (4.23) é verificada apenas para $\mathbf{S}_r = \mathbf{0}$. Porém, para o caso em que o espaço nulo à esquerda possui dimensão igual ou superior a 1, a equação $\mathbf{K} \Theta = \mathbf{I}_m$ não tem solução Θ . Note que, neste caso, a igualdade $\mathbf{K}^T \mathbf{S}_r = \mathbf{0}$ corresponde à equação normal para as colunas de Θ , dada por

$$\mathbf{K}^T (\mathbf{K} \Theta) = \mathbf{K}^T (\mathbf{I}_m), \quad (4.24)$$

e o algoritmo fornecerá a melhor solução pelo critério dos mínimos quadrados.

4.3.2 Matriz de erro à esquerda

Similarmente, para o erro (4.19) considera-se a função de Lyapunov $2V_\ell = \text{tr}(\mathbf{S}_\ell^T \mathbf{S}_\ell)$, com derivada temporal ao longo das trajetórias do sistema dada por

$$2\dot{V}_\ell = \text{tr}(\dot{\mathbf{S}}_\ell^T \mathbf{S}_\ell + \mathbf{S}_\ell^T \dot{\mathbf{S}}_\ell) = 2 \text{tr}(\mathbf{S}_\ell^T \Theta \dot{\mathbf{K}}) + 2 \text{tr}(\mathbf{S}_\ell^T \dot{\Theta} \mathbf{K}). \quad (4.25)$$

Novamente, uma escolha para a lei de adaptação é:

$$\dot{\Theta} = -\Gamma \mathbf{S}_\ell \mathbf{K}^T, \quad (4.26)$$

onde $\mathbf{\Gamma} = \mathbf{\Gamma}^T > 0$ é a matriz de ganho de atualização, resultando em

$$\dot{V}_\ell = \text{tr}(\mathbf{S}_\ell^T \mathbf{\Theta} \dot{\mathbf{K}}) - \underbrace{\text{tr}(\mathbf{K} \mathbf{S}_\ell^T \mathbf{\Gamma} \mathbf{S}_\ell \mathbf{K}^T)}_{\leq 0}. \quad (4.27)$$

que também depende de $\dot{\mathbf{K}}$. Para $\dot{\mathbf{K}} \equiv \mathbf{0}$, tem-se que $\dot{V}_\ell \leq 0$. Neste caso, $\dot{\mathbf{\Theta}} = \mathbf{0}$ e $\dot{V}_\ell = 0$ quando

$$\mathbf{S}_\ell \mathbf{K}^T = \mathbf{0}, \quad (4.28)$$

ou seja, quando as linhas de \mathbf{S}_ℓ pertencerem ao espaço nulo de \mathbf{K} . Para uma matriz \mathbf{K} tal que $\text{Nul}(\mathbf{K}) = \{\mathbf{0}\}$, a equação $\mathbf{\Theta} \mathbf{K} = \mathbf{I}_n$ tem ao menos uma solução $\mathbf{\Theta}$ e a igualdade (4.28) é verificada apenas para $\mathbf{S}_\ell = \mathbf{0}$. Por outro lado, para o caso em que o espaço nulo possui dimensão igual ou superior a 1, a equação $\mathbf{\Theta} \mathbf{K} = \mathbf{I}_n$ não tem solução $\mathbf{\Theta}$. Note que, neste caso, a igualdade $\mathbf{S}_\ell \mathbf{K}^T = \mathbf{0}$ corresponde à equação normal para as linhas de $\mathbf{\Theta}$, dada por

$$(\mathbf{\Theta} \mathbf{K}) \mathbf{K}^T = (\mathbf{I}_n) \mathbf{K}^T. \quad (4.29)$$

Novamente, o algoritmo fornecerá a melhor solução pelo critério dos mínimos quadrados.

4.3.3 Lei de atualização composta

Para casos não-redundantes ($m = n$) e com posto completo, as leis de atualização (4.21) e (4.26) são capazes de resolver individualmente o problema de determinação da inversa, uma vez que as matrizes inversa à esquerda e inversa à direita são iguais. Para os casos em que $m \neq n$, configuram-se os casos de existência (posto completo por linhas) ou unicidade (posto completo por colunas). Nota-se que as próprias matrizes de erro \mathbf{S}_r e \mathbf{S}_ℓ apresentam dimensões diferentes, sugerindo também propriedades diferentes. A lei de atualização composta que envolve as duas matrizes de erro simultaneamente pode ser obtida a partir da análise de estabilidade da função de Lyapunov $2V_c = 2V_r + 2V_\ell$, com derivada temporal dada por

$$2\dot{V}_c = 2g(\dot{\mathbf{K}}) + 2\text{tr}(\mathbf{S}_r^T \mathbf{K} \dot{\mathbf{\Theta}}) + 2\text{tr}(\mathbf{S}_\ell^T \dot{\mathbf{\Theta}} \mathbf{K}), \quad (4.30)$$

onde $g(\dot{\mathbf{K}}) = \text{tr}(\mathbf{S}_\ell^T \mathbf{\Theta} \dot{\mathbf{K}}) + \text{tr}(\mathbf{S}_r^T \dot{\mathbf{K}} \mathbf{\Theta})$. Assim, de (4.30) tem-se

$$\dot{V}_c = g(\dot{\mathbf{K}}) + \text{tr}((\mathbf{S}_r^T \mathbf{K} + \mathbf{K} \mathbf{S}_\ell^T) \dot{\mathbf{\Theta}}). \quad (4.31)$$

Desta forma, a lei de atualização proposta é

$$\dot{\mathbf{\Theta}} = -\mathbf{\Gamma}(\mathbf{S}_\ell \mathbf{K}^T + \mathbf{K}^T \mathbf{S}_r). \quad (4.32)$$

onde $\mathbf{\Gamma} = \mathbf{\Gamma}^T > 0$. Como resultado, tem-se

$$\dot{V}_c = g(\dot{\mathbf{K}}) \underbrace{- \text{tr}((\mathbf{S}_r^T \mathbf{K} + \mathbf{K} \mathbf{S}_\ell^T) \mathbf{\Gamma} (\mathbf{S}_\ell \mathbf{K}^T + \mathbf{K}^T \mathbf{S}_r))}_{\leq 0}. \quad (4.33)$$

Para $\dot{\mathbf{K}} \equiv \mathbf{0}$, tem-se que $\dot{V}_c \leq 0$. Neste caso, $\dot{\mathbf{\Theta}} = \mathbf{0}$ e $\dot{V}_c = 0$ quando

$$\mathbf{K}^T \mathbf{S}_r + \mathbf{S}_\ell \mathbf{K}^T = \mathbf{0}. \quad (4.34)$$

De imediato, tem-se que a ocorrência simultânea das condições $\mathbf{K}^T \mathbf{S}_r = \mathbf{0}$ e $\mathbf{S}_\ell \mathbf{K}^T = \mathbf{0}$ apresentadas anteriormente implica em $\dot{\mathbf{\Theta}} = \mathbf{0}$. A prova no sentido contrário é obtida a seguir. Assumindo (4.34), tem-se que

$$(\mathbf{K}^T \mathbf{S}_r + \mathbf{S}_\ell \mathbf{K}^T) \mathbf{K} \mathbf{S}_\ell^T = \mathbf{0}. \quad (4.35)$$

Utilizando a função traço e uma vez que $\mathbf{S}_r \mathbf{K} = \mathbf{K} \mathbf{S}_\ell = \mathbf{K} \mathbf{\Theta} \mathbf{K} - \mathbf{K}$, obtém-se

$$\begin{aligned} 0 &= \text{tr}(\mathbf{K}^T \mathbf{S}_r \mathbf{K} \mathbf{S}_\ell^T + \mathbf{S}_\ell \mathbf{K}^T \mathbf{K} \mathbf{S}_\ell^T) \\ &= \text{tr}(\mathbf{K}^T \mathbf{K} \mathbf{S}_\ell \mathbf{S}_\ell^T + \mathbf{S}_\ell \mathbf{K}^T \mathbf{K} \mathbf{S}_\ell^T) \\ &= \text{tr}(\mathbf{K}^T \mathbf{K} \mathbf{S}_\ell \mathbf{S}_\ell^T) + \text{tr}(\mathbf{S}_\ell \mathbf{K}^T \mathbf{K} \mathbf{S}_\ell^T) \\ &= \underbrace{\text{tr}(\mathbf{S}_\ell^T \mathbf{K}^T \mathbf{K} \mathbf{S}_\ell)}_{\geq 0} + \underbrace{\text{tr}(\mathbf{S}_\ell \mathbf{K}^T \mathbf{K} \mathbf{S}_\ell^T)}_{\geq 0}. \end{aligned} \quad (4.36)$$

Como todos os termos da soma são positivos ou nulos (traço de matrizes semi-definidas positivas e simétricas), tem-se que $\text{tr}(\mathbf{S}_\ell^T \mathbf{K}^T \mathbf{K} \mathbf{S}_\ell) = \text{tr}(\mathbf{S}_\ell \mathbf{K}^T \mathbf{K} \mathbf{S}_\ell^T) = 0$. Desta forma, tem-se que $\mathbf{S}_\ell \mathbf{K}^T = \mathbf{0}$ e, considerando (4.34), chega-se a $\mathbf{K}^T \mathbf{S}_r = \mathbf{0}$. Assim,

$$\{\mathbf{K}^T \mathbf{S}_r + \mathbf{S}_\ell \mathbf{K}^T = \mathbf{0}\} \Leftrightarrow \{\mathbf{K}^T \mathbf{S}_r = \mathbf{0}\} \wedge \{\mathbf{S}_\ell \mathbf{K}^T = \mathbf{0}\}. \quad (4.37)$$

Como visto, a matriz $\mathbf{\Theta}$ evoluirá para uma solução que satisfaz (4.34), isto é,

$$\mathbf{K}^T \mathbf{K} \mathbf{\Theta} + \mathbf{\Theta} \mathbf{K} \mathbf{K}^T = 2 \mathbf{K}^T. \quad (4.38)$$

A equação (4.38) é uma equação de Sylvester e apresentará uma solução única se e somente se as matrizes $\mathbf{K}^T \mathbf{K}$ e $-\mathbf{K} \mathbf{K}^T$ não apresentarem autovalores em comum (ver Apêndice A.6). Esta condição é satisfeita para matrizes \mathbf{K} com o maior posto possível (completo por linhas ou colunas) e a solução converge para a pseudo-inversa de \mathbf{K} . Para os casos em que \mathbf{K} apresenta posto incompleto, a solução não é única e depende da condição inicial $\mathbf{\Theta}(0)$. Na próxima seção, a lei de atualização composta é analisada, sendo justificada a utilização das duas matrizes de erro \mathbf{S}_ℓ e \mathbf{S}_r . Além disso, será estabelecida uma relação entre os casos escalar e multivariável.

4.4 Análise de convergência

Considere a *decomposição de valor singular* (SVD) da matriz de ganho $\mathbf{K}(t) \in \mathbb{R}^{m \times n}$

$$\mathbf{K} = \mathbf{U} \mathbf{\Sigma} \mathbf{V}^T, \quad (4.39)$$

onde $\mathbf{U} \in \mathbb{R}^{m \times m}$, $\mathbf{\Sigma} \in \mathbb{R}^{m \times n}$ e $\mathbf{V} \in \mathbb{R}^{n \times n}$. As matrizes \mathbf{U} e \mathbf{V} são matrizes ortogonais que fornecem bases ortonormais para todos os subespaços fundamentais de \mathbf{K} , e a matriz $\mathbf{\Sigma}$ possui r valores singulares em sua diagonal, sendo r o posto de \mathbf{K} . A fatoração SVD escolhe as bases de forma especial e permite, para um sistema (4.16), relacionar a entrada \mathbf{u} e a saída $\dot{\mathbf{y}}$. Seja \mathbf{u} uma combinação linear das colunas de \mathbf{V} . A saída $\dot{\mathbf{y}}$ será a mesma combinação linear das colunas de \mathbf{U} , multiplicadas pelos valores singulares σ_i a elas associados. Isto se origina diretamente de $\mathbf{K} \mathbf{V} = \mathbf{U} \mathbf{\Sigma}$, observado uma coluna por vez [54]. Da mesma forma, pode-se expressar a matriz $\mathbf{\Theta}$ como

$$\mathbf{\Theta} = \mathbf{U}_\Theta \mathbf{\Sigma}_\Theta \mathbf{V}_\Theta^T. \quad (4.40)$$

A condição $\mathbf{K} \mathbf{\Theta} \rightarrow \mathbf{I}_m$ pode então ser dividida em:

$$\mathbf{U}_\Theta \rightarrow \mathbf{V}, \quad (4.41)$$

$$\mathbf{\Sigma} \mathbf{\Sigma}_\Theta \rightarrow \mathbf{I}_m, \quad (4.42)$$

$$\mathbf{V}_\Theta \rightarrow \mathbf{U}. \quad (4.43)$$

Note que (4.42) representa um ajuste de amplitudes, similar ao caso escalar, e que para a condição $\mathbf{\Theta} \mathbf{K} \rightarrow \mathbf{I}_n$ se torna $\mathbf{\Sigma}_\Theta \mathbf{\Sigma} \rightarrow \mathbf{I}_n$. Porém, adicionalmente a este problema, estão as condições (4.41) e (4.43), associadas ao alinhamento das bases ortonormais definidas por \mathbf{V} e \mathbf{U}_Θ , e também \mathbf{U} e \mathbf{V}_Θ . Isto é, para o caso multivariável, além do ajuste de amplitudes existente no caso escalar, há também a necessidade do ajuste de ângulos. A solução adotada para análise é expressar o mapeamento realizado por $\mathbf{\Theta}$ através das bases ortonormais definidas por \mathbf{U} e \mathbf{V} . Assim, assumamos que

$$\mathbf{\Theta} = \mathbf{V} \mathbf{\Psi} \mathbf{U}^T. \quad (4.44)$$

Intuitivamente, o que se faz é utilizar o mapeamento contrário, como o realizado pela inversa generalizada [54], razoável uma vez que $\mathbf{\Theta}$ deve evoluir para a inversa de \mathbf{K} . Note porém que a matriz $\mathbf{\Psi}$ não é diagonal e, portanto, quando $\mathbf{\Theta}$ multiplica uma coluna \mathbf{u}_j de \mathbf{U} , produz uma combinação linear de todas as colunas de \mathbf{V} . Considere o caso mais simples em que $\dot{\mathbf{K}} \equiv \mathbf{0}$ ($\dot{\mathbf{U}} \equiv \mathbf{0}$ e $\dot{\mathbf{V}} \equiv \mathbf{0}$). Neste caso, a dinâmica de $\mathbf{\Psi}$ é dada por

$$\dot{\mathbf{\Psi}} = \mathbf{V}^T \dot{\mathbf{\Theta}} \mathbf{U}. \quad (4.45)$$

Aplicando a lei de atualização composta, proposta em (4.32), e considerando um ganho escalar $\Gamma > 0$ tem-se

$$\begin{aligned}
\dot{\Psi} &= -\Gamma \mathbf{V}^T (\mathbf{S}_\ell \mathbf{K}^T + \mathbf{K}^T \mathbf{S}_r) \mathbf{U} \\
&= -\Gamma \mathbf{V}^T \mathbf{S}_\ell \mathbf{K}^T \mathbf{U} - \Gamma \mathbf{V}^T \mathbf{K}^T \mathbf{S}_r \mathbf{U} \\
&= -\Gamma \mathbf{V}^T \mathbf{S}_\ell \mathbf{V} \Sigma^T \mathbf{U}^T \mathbf{U} - \Gamma \mathbf{V}^T \mathbf{V} \Sigma^T \mathbf{U}^T \mathbf{S}_r \mathbf{U} \\
&= -\Gamma \mathbf{V}^T (\Theta \mathbf{K} - \mathbf{I}_n) \mathbf{V} \Sigma^T - \Gamma \Sigma^T \mathbf{U}^T (\mathbf{K} \Theta - \mathbf{I}_m) \mathbf{U} \\
&= -\Gamma (\mathbf{V}^T \Theta \mathbf{U} \Sigma - \mathbf{I}_n) \Sigma^T - \Gamma \Sigma^T (\Sigma \mathbf{V}^T \Theta \mathbf{U} - \mathbf{I}_m) \\
&= -\Gamma (\Psi \Sigma - \mathbf{I}_n) \Sigma^T - \Gamma \Sigma^T (\Sigma \Psi - \mathbf{I}_m).
\end{aligned} \tag{4.46}$$

A similaridade entre os casos escalar e multivariável pode ser verificada através da análise dos elementos diagonais e residuais de Ψ . Desta forma, a matriz Ψ é decomposta em

$$\Psi = \mathbf{D}_\Theta + \mathbf{R}_\Theta, \tag{4.47}$$

onde $\mathbf{D}_\Theta \in \mathbb{R}^{n \times m}$ possui os elementos diagonais de Ψ e $\mathbf{R}_\Theta \in \mathbb{R}^{n \times m}$ possui os elementos fora da diagonal de Ψ , denominados residuais, com zeros na diagonal. Por exemplo, $\Psi \in \mathbb{R}^{4 \times 3}$ pode ser expressa como

$$\Psi = \underbrace{\begin{bmatrix} d_1 & 0 & 0 \\ 0 & d_2 & 0 \\ 0 & 0 & d_3 \\ 0 & 0 & 0 \end{bmatrix}}_{\mathbf{D}_\Theta} + \underbrace{\begin{bmatrix} 0 & r_{12} & r_{13} \\ r_{21} & 0 & r_{23} \\ r_{31} & r_{32} & 0 \\ r_{41} & r_{42} & r_{43} \end{bmatrix}}_{\mathbf{R}_\Theta}. \tag{4.48}$$

A derivada temporal de (4.47) é dada por

$$\dot{\Psi} = \dot{\mathbf{D}}_\Theta + \dot{\mathbf{R}}_\Theta \tag{4.49}$$

e, substituindo (4.47) e (4.49) em (4.46),

$$\begin{aligned}
\dot{\mathbf{D}}_\Theta + \dot{\mathbf{R}}_\Theta &= -\Gamma (\mathbf{D}_\Theta \Sigma + \mathbf{R}_\Theta \Sigma - \mathbf{I}_n) \Sigma^T \\
&\quad - \Gamma \Sigma^T (\Sigma \mathbf{D}_\Theta + \Sigma \mathbf{R}_\Theta - \mathbf{I}_m).
\end{aligned} \tag{4.50}$$

Note que o produto de matrizes diagonais e matrizes residuais também é residual, apresentando zeros na diagonal principal. Desta forma, é possível separar a análise das dinâmicas de \mathbf{D}_Θ e \mathbf{R}_Θ , presentes em (4.50). Assim, para a componente diagonal de Ψ , tem-se

$$\dot{\mathbf{D}}_\Theta = -\Gamma (\mathbf{D}_\Theta \Sigma - \mathbf{I}_n) \Sigma^T - \Gamma \Sigma^T (\Sigma \mathbf{D}_\Theta - \mathbf{I}_m). \tag{4.51}$$

A dinâmica do elemento diagonal d_i é dada por

$$\dot{d}_i = -\Gamma(d_i\sigma_i - 1)\sigma_i - \Gamma\sigma_i(\sigma_id_i - 1) = -2\Gamma S_i\sigma_i, \quad (4.52)$$

apresentando uma forma similar ao caso escalar analisado inicialmente (note que $\sigma_i \in \mathbb{R}^+$). Note também que os elementos da diagonal convergem exponencialmente para $1/\sigma_i$ quando $\sigma_i \neq 0$. Para $\sigma_i = 0$, tem-se que $\dot{d}_i = 0$. A Figura 4.6 apresenta a evolução de d_i para diferentes condições iniciais.

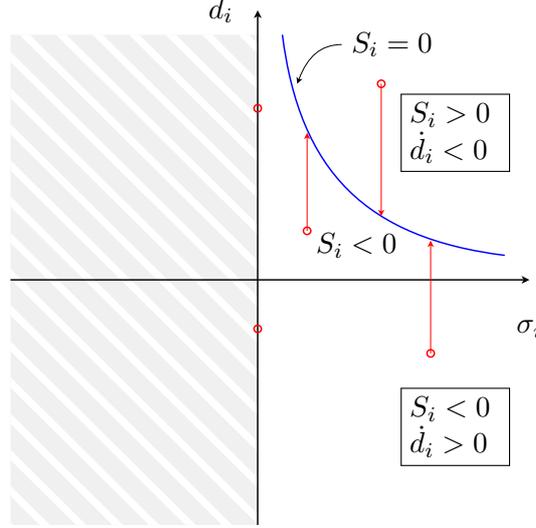


Figura 4.6: Plano $d_i \times \sigma_i$ (diferentes condições iniciais).

Para os elementos residuais tem-se

$$\dot{\mathbf{R}}_{\Theta} = -\underbrace{\Gamma \mathbf{R}_{\Theta} \Sigma \Sigma^T}_1 - \underbrace{\Gamma \Sigma^T \Sigma \mathbf{R}_{\Theta}}_2. \quad (4.53)$$

As matrizes $\Sigma \Sigma^T \in \mathbb{R}^{m \times m}$ e $\Sigma^T \Sigma \in \mathbb{R}^{n \times n}$ são matrizes diagonais cujos elementos são os quadrados dos valores singulares de \mathbf{K} . Para os casos em que $m \neq n$, apresentam dimensões diferentes e uma delas também apresentará zeros na diagonal. No termo 1 de (4.53), obtido quando considerada a matriz de erro \mathbf{S}_{ℓ} , os elementos diagonais multiplicam as colunas de \mathbf{R}_{Θ} . Por outro lado, no termo 2, obtido com a utilização de \mathbf{S}_r , os elementos diagonais são aplicados às linhas de \mathbf{R}_{Θ} . Assim, a dinâmica de um elemento residual r_{ij} pode ser expressa por

$$\dot{r}_{ij} + \Gamma(\sigma_i^{*2} + \sigma_j^{*2})r_{ij} = 0, \quad (4.54)$$

onde $\sigma_k^* = \sigma_k$ se $k \leq \min(m, n)$ e $\sigma_k^* = 0$ caso contrário. A utilização de ambas as matrizes de erro \mathbf{S}_r e \mathbf{S}_{ℓ} torna a convergência do elemento residual dependente de dois valores singulares distintos, associados à linha e à coluna.

Observação 1: Considere o caso onde $m = n$ e $\sigma_m = 0$. O uso de uma lei de atualização baseada apenas em \mathbf{S}_r (multiplicação das linhas) resultaria em $\dot{r}_{mj} = 0, \forall j$. Os elementos residuais de toda m -ésima linha não podem se alterar e portanto não se anulam. O algoritmo composto que também utiliza \mathbf{S}_ℓ (multiplicação das colunas) assegura a convergência desses elementos para 0 desde que $\sigma_j \neq 0$. Note que uma lei de atualização baseada apenas na matriz de erro \mathbf{S}_ℓ resultaria em $\dot{r}_{in} = 0, \forall i$. Neste caso, todos os elementos residuais da n -ésima coluna não podem se alterar e, conseqüentemente, não convergem para 0.

Observação 2: Para os casos onde $m \neq n$ a necessidade de uma lei de atualização composta se torna ainda mais evidente. Considere o caso $\mathbf{K} \in \mathbb{R}^{2 \times 3}$

$$\mathbf{K} = \begin{bmatrix} \vdots & \vdots \\ \mathbf{u}_1 & \mathbf{u}_2 \\ \vdots & \vdots \end{bmatrix} \begin{bmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \end{bmatrix} \begin{bmatrix} \vdots & \vdots & \vdots \\ \mathbf{v}_1 & \mathbf{v}_2 & \mathbf{v}_3 \\ \vdots & \vdots & \vdots \end{bmatrix}^T, \quad (4.55)$$

onde $\sigma_1 \neq 0$ e $\sigma_2 \neq 0$ (posto completo por linhas). A última coluna de \mathbf{V} , \mathbf{v}_3 , fornece uma base ortonormal para o espaço nulo de \mathbf{K} , isto é, seja $\mathbf{b} = \alpha \mathbf{v}_3$, tem-se $\mathbf{K} \mathbf{b} = \mathbf{0}, \forall \alpha$. A matriz Θ pode ser expressa, de acordo com (4.44), como

$$\Theta = \begin{bmatrix} \vdots & \vdots & \vdots \\ \mathbf{v}_1 & \mathbf{v}_2 & \mathbf{v}_3 \\ \vdots & \vdots & \vdots \end{bmatrix} \begin{bmatrix} d_1 & r_{12} \\ r_{21} & d_2 \\ r_{31} & r_{32} \end{bmatrix} \begin{bmatrix} \vdots & \vdots \\ \mathbf{u}_1 & \mathbf{u}_2 \\ \vdots & \vdots \end{bmatrix}^T. \quad (4.56)$$

Note que r_{31} e r_{32} estão associados a esta base (multiplicam \mathbf{v}_3) e, para $r_{31} \neq 0$ e $r_{32} \neq 0$, uma lei $\mathbf{b} = \Theta \mathbf{c}$ apresentará componentes no espaço nulo de \mathbf{K} , não representando portanto uma lei de norma mínima ou ótima. Como foi visto, apenas a utilização de \mathbf{S}_ℓ permitirá que estes elementos evoluam para zero, uma vez que torna a convergência dependente do valor singular associado à coluna.

Observação 3: Considere agora o caso $\mathbf{K} \in \mathbb{R}^{3 \times 2}$, com posto completo por colunas ($\sigma_1 \neq 0$ e $\sigma_2 \neq 0$). Da mesma forma, \mathbf{K} pode ser escrito como

$$\mathbf{K} = \begin{bmatrix} \vdots & \vdots & \vdots \\ \mathbf{u}_1 & \mathbf{u}_2 & \mathbf{u}_3 \\ \vdots & \vdots & \vdots \end{bmatrix} \begin{bmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \vdots & \vdots \\ \mathbf{v}_1 & \mathbf{v}_2 \\ \vdots & \vdots \end{bmatrix}^T. \quad (4.57)$$

Neste caso, a última coluna de \mathbf{U} , \mathbf{u}_3 , fornece uma base ortonormal para o espaço nulo à esquerda de \mathbf{K} . Isto é, para um sistema $\mathbf{K} \mathbf{b} = \mathbf{c}$, não existe solução \mathbf{b} para $\mathbf{c} = \alpha \mathbf{u}_3$. Representa portanto uma base para variáveis \mathbf{c} inalcançáveis ou inviáveis.

Considere novamente a expressão proposta em (4.44) para Θ

$$\Theta = \begin{bmatrix} \vdots & \vdots \\ \mathbf{v}_1 & \mathbf{v}_2 \\ \vdots & \vdots \end{bmatrix} \begin{bmatrix} d_1 & r_{12} & r_{13} \\ r_{21} & d_2 & r_{23} \end{bmatrix} \begin{bmatrix} \vdots & \vdots & \vdots \\ \mathbf{u}_1 & \mathbf{u}_2 & \mathbf{u}_3 \\ \vdots & \vdots & \vdots \end{bmatrix}^T. \quad (4.58)$$

Os elementos residuais r_{13} e r_{23} estão associados a \mathbf{u}_3 . Para $r_{13} \neq 0$, $r_{23} \neq 0$ e $\mathbf{c} = \alpha \mathbf{u}_3$, tem-se $\mathbf{b} = \Theta \mathbf{c} = r_{13} \mathbf{v}_1 + r_{23} \mathbf{v}_2 \neq \mathbf{0}$, embora \mathbf{c} seja uma saída inviável para o sistema. Neste caso, apenas a utilização da matriz de erro à direita \mathbf{S}_r garantirá que estes elementos residuais converjam para 0 e assim $\mathbf{b} = \Theta \mathbf{c} \rightarrow 0$.

4.5 Solução para cinemática inversa

Como foi visto no capítulo 2, a solução para o problema de cinemática inversa pode ser obtida de forma analítica, envolvendo o uso de identidades algébricas e geométricas, ou através de métodos numéricos e técnicas de otimização. A solução diferencial, também denominada algoritmo iterativo, foi apresentada nas seções 2.4.2 e 2.4.5, e tem como base o mapeamento linear realizado pela matriz Jacobiana entre as velocidades nos espaços das juntas e operacional. Como visto, a escolha

$$\dot{\mathbf{q}} = \mathbf{J}^{-1} \boldsymbol{\nu}, \quad (4.59)$$

para $\boldsymbol{\Lambda} = \boldsymbol{\Lambda}^T > 0$ e $\boldsymbol{\nu} = \dot{\mathbf{x}}_d + \boldsymbol{\Lambda} \mathbf{e}$ garante a convergência exponencial do erro \mathbf{e} para $\mathbf{0}$. Para manipuladores redundantes, a pseudo-inversa à direita $\mathbf{J}^\dagger = \mathbf{J}^T (\mathbf{J} \mathbf{J}^T)^{-1}$ é utilizada, fornecendo uma solução de norma mínima ou ótima desde que $\boldsymbol{\nu}(t)$ não conduza o manipulador para configurações singulares. Nestas configurações, a matriz \mathbf{J} perde posto e sua inversa não se define. Na vizinhança de singularidades, a matriz Jacobiana se torna mal condicionada, apresentando pivôs muito pequenos e resultando em velocidades de juntas elevadas, indesejadas na prática. A proposta de aplicação da inversa filtrada como solução para a cinemática inversa de manipuladores robóticos é dada por

$$\dot{\mathbf{q}} = \Theta \boldsymbol{\nu} \quad (4.60)$$

onde $\Theta \in \mathbb{R}^{n \times m}$ é a inversa filtrada da matriz Jacobiana \mathbf{J} , sendo atualizada pela lei composta apresentada em (4.32), logo,

$$\dot{\Theta} = -\Gamma (\mathbf{J}^T \mathbf{S}_r + \mathbf{S}_\ell \mathbf{J}^T) \quad (4.61)$$

onde $\Gamma = \gamma \mathbf{I} > 0$ é o ganho de atualização. As matrizes \mathbf{S}_r e \mathbf{S}_ℓ correspondem às matrizes de erro à direita e à esquerda, respectivamente. Considere a decomposição

em valores singulares da matriz Jacobiana, apresentada anteriormente em (3.1) e aqui reescrita por conveniência:

$$\mathbf{J} = \sum_{i=1}^m \sigma_i \mathbf{u}_i \mathbf{v}_i^T. \quad (4.62)$$

Em vista de (4.44) e (4.62), a solução (4.60) pode ser reescrita como

$$\dot{\mathbf{q}} = \sum_{i=1}^m d_i \mathbf{v}_i \mathbf{u}_i^T \boldsymbol{\nu} + \sum_{i \neq j} r_{ij} \mathbf{v}_i \mathbf{u}_j^T \boldsymbol{\nu}. \quad (4.63)$$

Para valores singulares não nulos, os elementos diagonais d_i convergem para $1/\sigma_i$. Os elementos residuais r_{ij} representam pareamentos incorretos entre as colunas de \mathbf{V} e \mathbf{U} . Como visto na Seção 4.4, a convergência destes termos para zero depende dos valores singulares associados tanto à linha quanto à coluna. O diagrama de blocos referente à utilização da inversa filtrada para solucionar o problema de cinemática inversa é apresentado na Figura 4.7.

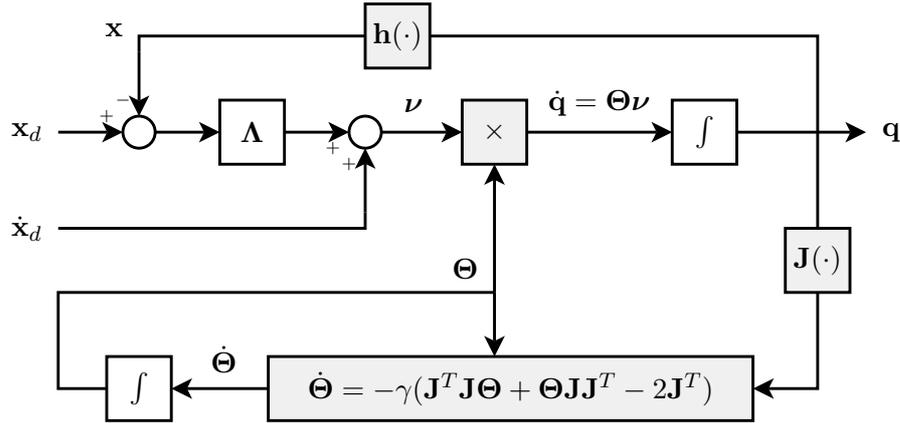


Figura 4.7: Diagrama de blocos IF.

A Tabela 4.1 resume as equações referentes à solução proposta, assim como apresenta suas correspondentes discretas para aplicações práticas.

Inversa filtrada contínua	Correspondente discreta (período T)
$\mathbf{e}(t) = \mathbf{x}_d(t) - \mathbf{x}(t)$	$\mathbf{e}(k) = \mathbf{x}_d(k) - \mathbf{x}(k)$
$\boldsymbol{\nu}(t) = \dot{\mathbf{x}}_d(t) + \boldsymbol{\Lambda} \mathbf{e}(t)$	$\boldsymbol{\nu}(k) = \dot{\mathbf{x}}_d(k) + \boldsymbol{\Lambda} \mathbf{e}(k)$
$\mathbf{S}_r(t) = \mathbf{J}(t) \boldsymbol{\Theta}(t) - \mathbf{I}_m$	$\mathbf{S}_r(k) = \mathbf{J}(k) \boldsymbol{\Theta}(k) - \mathbf{I}_m$
$\mathbf{S}_\ell(t) = \boldsymbol{\Theta}(t) \mathbf{J}(t) - \mathbf{I}_n$	$\mathbf{S}_\ell(k) = \boldsymbol{\Theta}(k) \mathbf{J}(k) - \mathbf{I}_n$
$\boldsymbol{\Theta}(t) = \boldsymbol{\Theta}(0) + \int_0^t \dot{\boldsymbol{\Theta}} dt$	$\boldsymbol{\Theta}(k) = \boldsymbol{\Theta}(k-1) + \boldsymbol{\Delta}_\Theta(k-1)T$
$\dot{\boldsymbol{\Theta}}(t) = -\gamma(\mathbf{J}^T(t) \mathbf{S}_r(t) + \mathbf{S}_\ell(t) \mathbf{J}^T(t))$	$\boldsymbol{\Delta}_\Theta(k) = -\gamma(\mathbf{J}^T(k) \mathbf{S}_r(k) + \mathbf{S}_\ell(k) \mathbf{J}^T(k))$
$\mathbf{q}(t) = \mathbf{q}(0) + \int_0^t \dot{\mathbf{q}} dt$	$\mathbf{q}(k) = \mathbf{q}(k-1) + \boldsymbol{\Delta}_q(k-1)T$
$\dot{\mathbf{q}}(t) = \boldsymbol{\Theta}(t) \boldsymbol{\nu}(t)$	$\boldsymbol{\Delta}_q(k) = \boldsymbol{\Theta}(k) \boldsymbol{\nu}(k)$

Tabela 4.1: Inversa filtrada e correspondente discreta.

Simulação 10

Considere novamente o manipulador antropomórfico Zebra-ZERO, porém sem atuação nas últimas 3 juntas. Desta forma, configura-se um caso 3DoF, não-redundante para posição $\mathbf{p} \in \mathbb{R}^3$. Inicialmente será avaliado o desempenho do algoritmo proposto para a Trajetória 6, livre e afastada de singularidades (Figura 4.8).

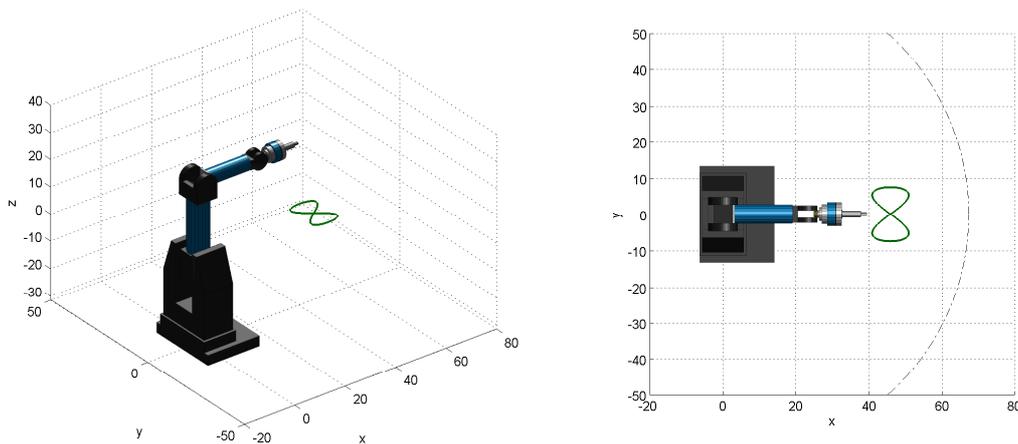


Figura 4.8: Simulação 10: manipulador Zebra-ZERO e trajetória desejada em perspectiva e visão superior.

Os resultados obtidos para $\mathbf{q}(0) = [0 \ \pi/2 \ -\pi]^T rad$, $\Theta(0) = \mathbf{0}$, $\Lambda = 2\mathbf{I}$ e $\Gamma = \mathbf{I}$ são apresentados a seguir e comparados com os obtidos pelo algoritmo DLS ($\delta_0 = 3 \times 10^2$ e $\omega_0 = 10^3$). Como esperado, os dois métodos possibilitaram o rastreamento com erro praticamente nulo. As trajetórias efetuadas e as normas dos erros de posição são mostradas nas Figuras 4.9 e 4.10.a, respectivamente.

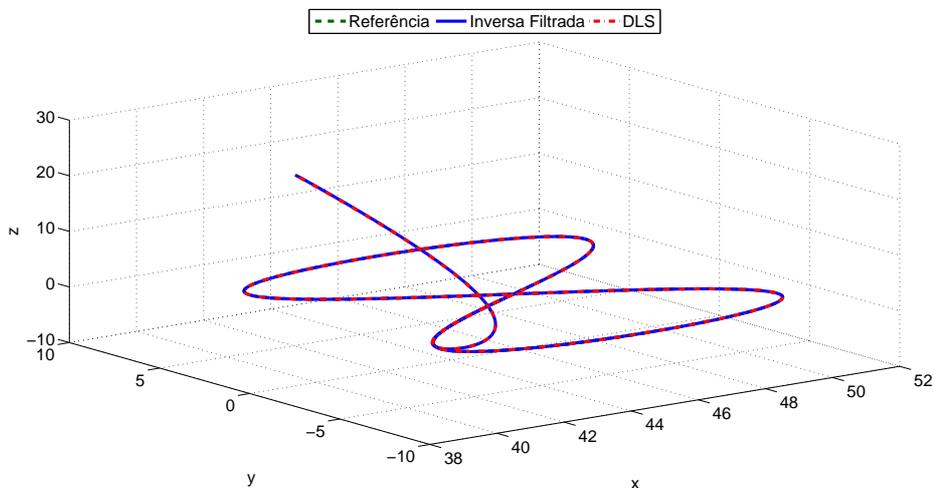


Figura 4.9: Simulação 10: trajetórias no espaço cartesiano.

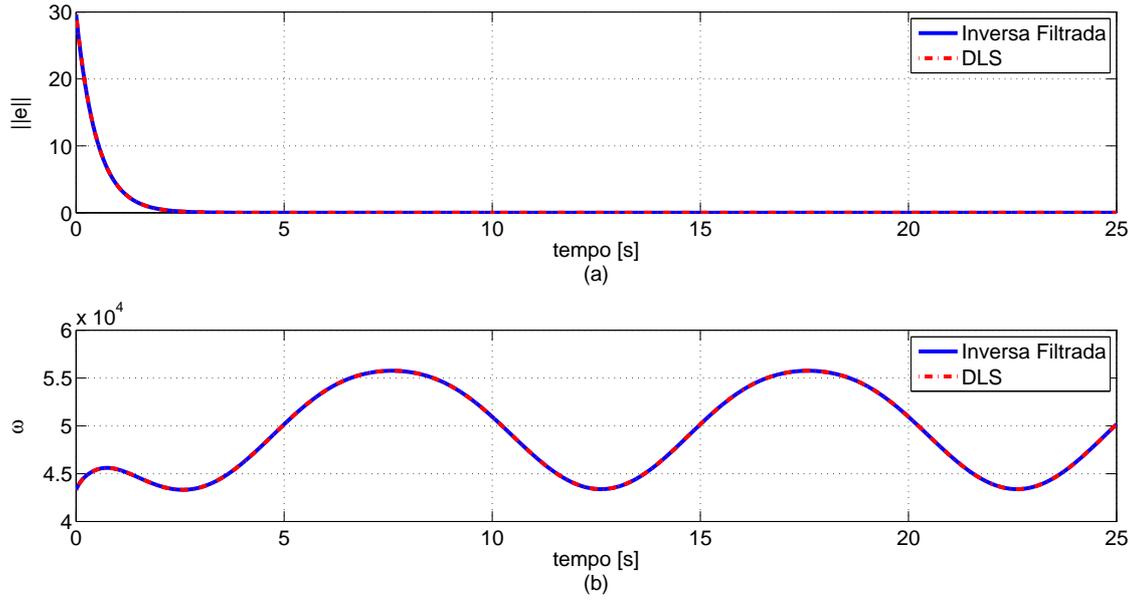


Figura 4.10: Simulação 10: (a) norma do erro e (b) manipulabilidade.

A medida de manipulabilidade $\omega(\mathbf{q}) = \sqrt{\det(\mathbf{J}(\mathbf{q})\mathbf{J}^T(\mathbf{q}))}$ é mostrada na Figura 4.10.b e evidencia o fato de que a trajetória desejada está afastada de configurações singulares. Desta forma, $\delta = 0$ para todo tempo t . A Figura 4.11 apresenta as variáveis $\dot{\mathbf{q}}$ e \mathbf{q} obtidas.

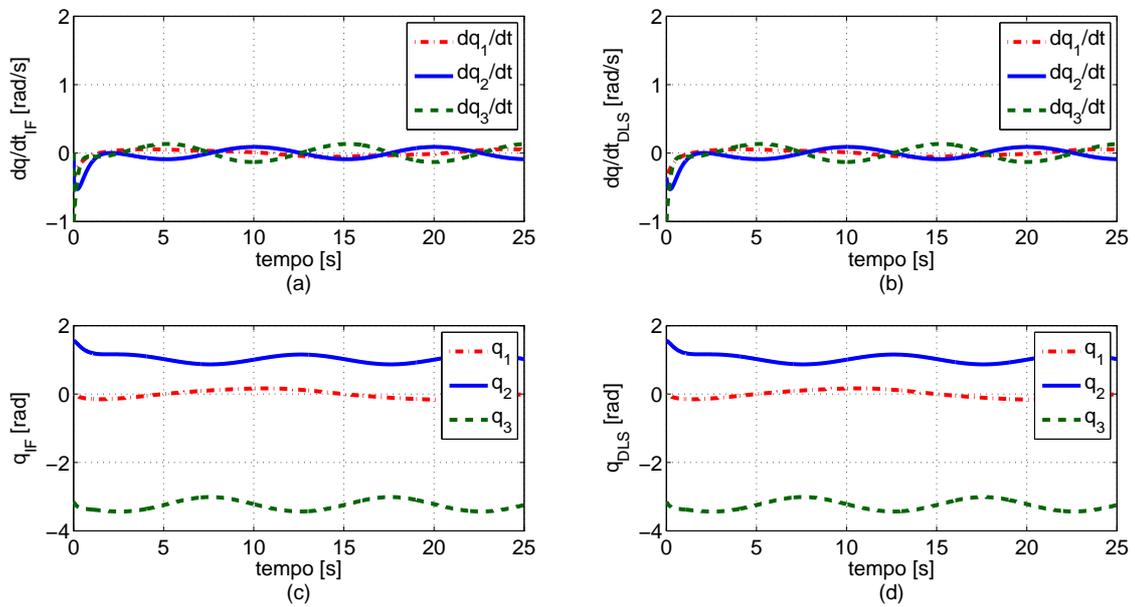


Figura 4.11: Simulação 10: variáveis (a) $\dot{\mathbf{q}}_{IF}$; (b) $\dot{\mathbf{q}}_{DLS}$; (c) \mathbf{q}_{IF} ; e (d) \mathbf{q}_{DLS} .

Note que os resultados são praticamente coincidentes e que as velocidades das juntas permanecem em uma faixa adequada, abaixo de 0.2 rad/s após o transitório.

Simulação 11

Considera-se agora uma trajetória que apresenta singularidades internas, isto é, configurações singulares que pertencem ao espaço de trabalho do manipulador robótico. A trajetória considerada (Trajetória 7) apresenta 3 pontos de singularidade de ombro, como evidenciado na Figura 4.12.

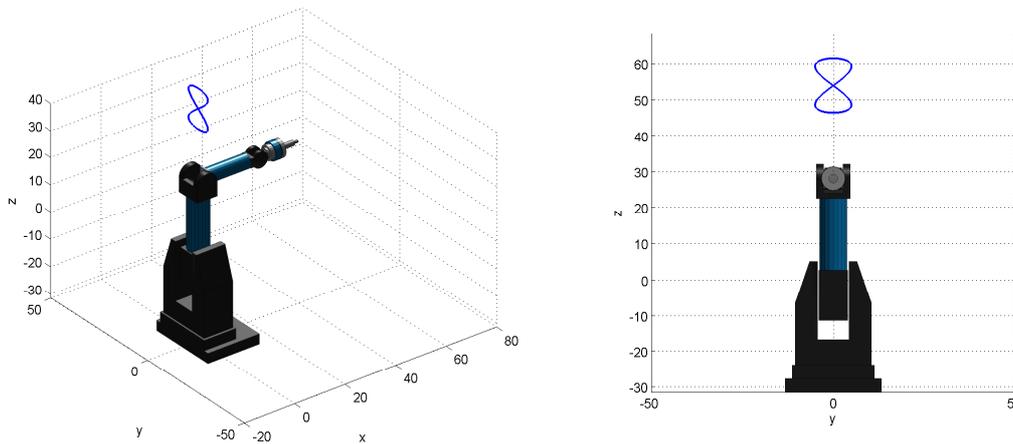


Figura 4.12: Simulação 11: manipulador Zebra-ZERO e trajetória desejada em perspectiva e visão frontal.

As trajetórias desenvolvidas são mostradas nas Figuras 4.13 e 4.14. Nota-se que, para os ganhos considerados, o algoritmo proposto apresentou um desempenho superior ao método DLS, permitindo o rastreamento da trajetória desejada com erro praticamente nulo, inclusive em configurações singulares. As normas dos erros de posição e as medidas de manipulabilidade são apresentados na Figura 4.15.

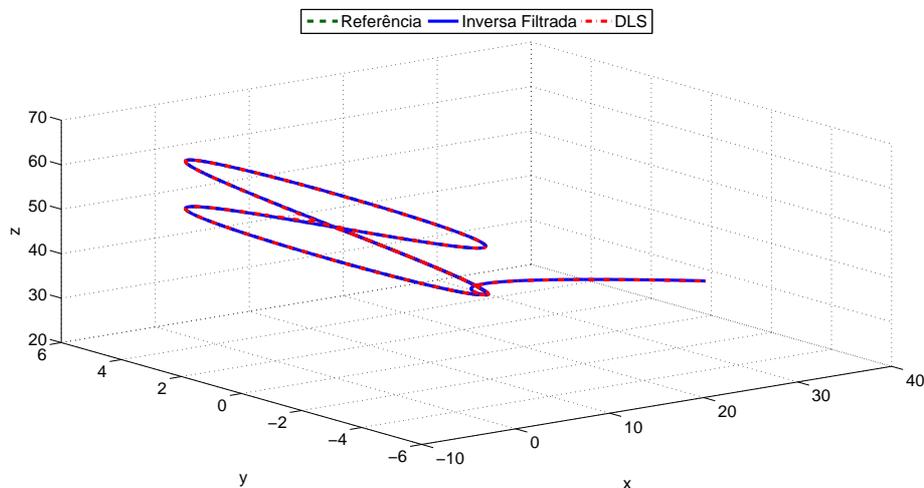


Figura 4.13: Simulação 11: trajetórias no espaço cartesiano.

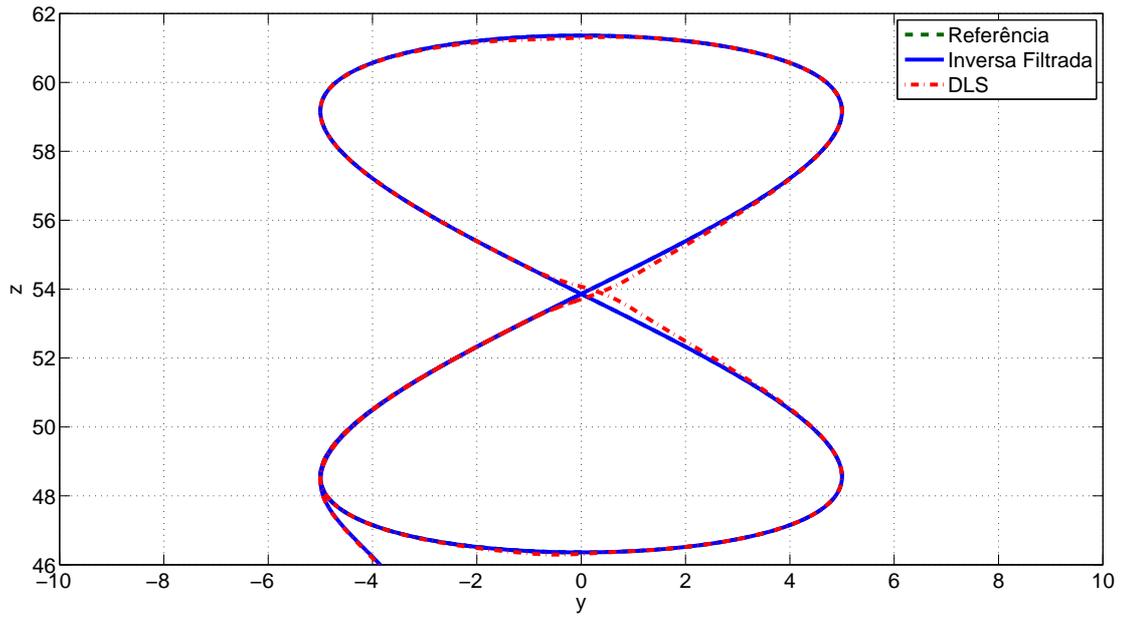


Figura 4.14: Simulação 11: trajetórias no plano yz .

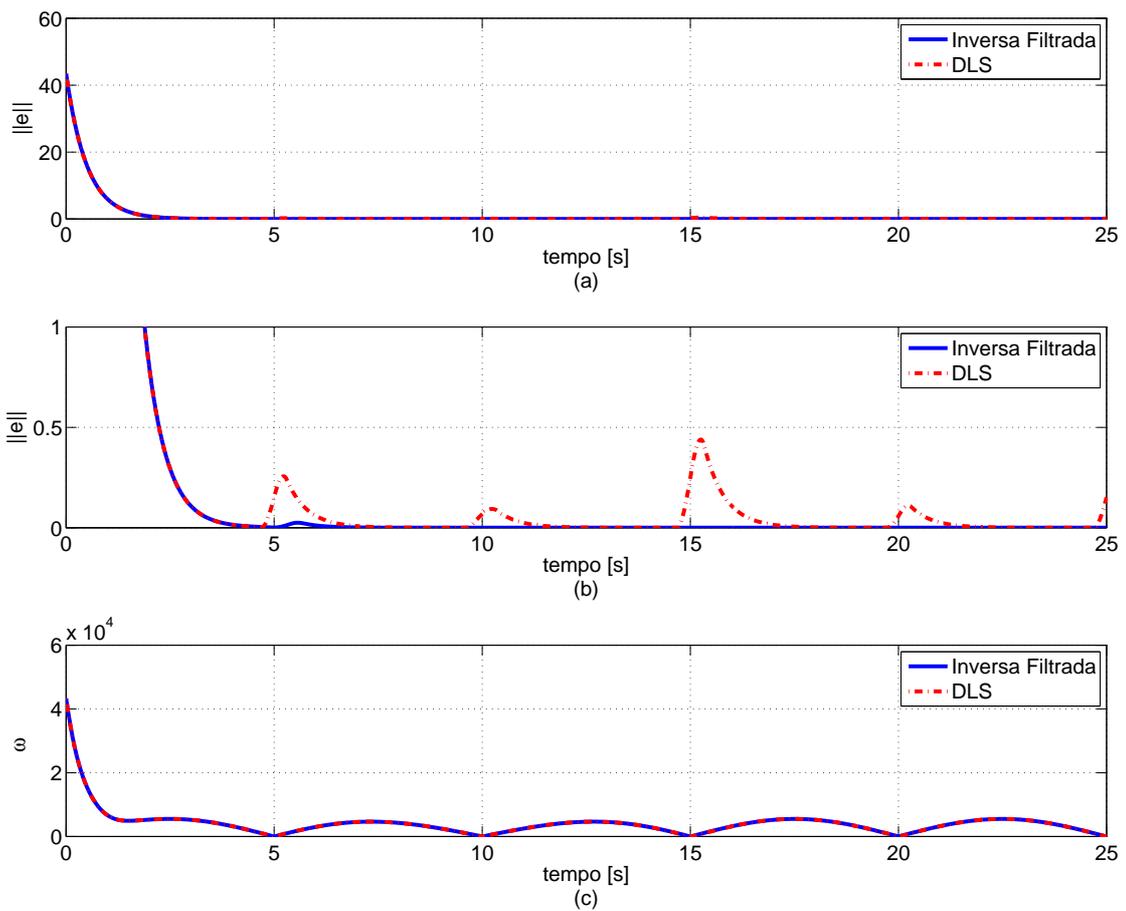


Figura 4.15: Simulação 11: (a) norma do erro; (b) norma do erro (escala ampliada); e (c) manipulabilidade.

As variáveis $\dot{\mathbf{q}}$ e \mathbf{q} obtidas são apresentadas na Figura 4.16. Note que as velocidades das juntas obtidas com a inversa filtrada são mais suaves, uma vez que não há o chaveamento de um termo de amortecimento.

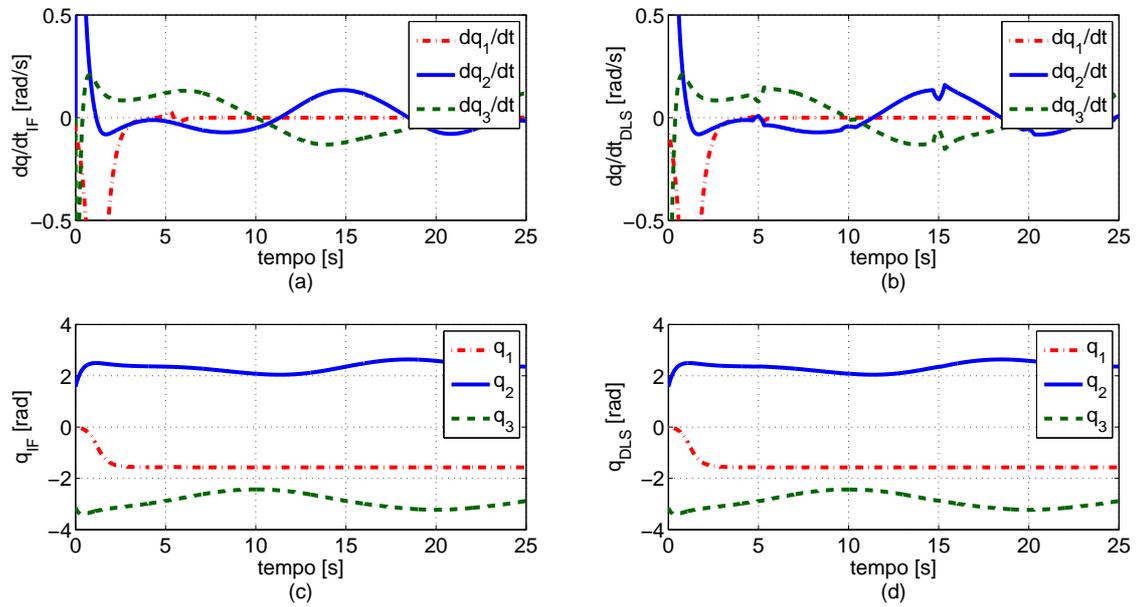


Figura 4.16: Simulação 11: variáveis (a) $\dot{\mathbf{q}}_{IF}$; (b) $\dot{\mathbf{q}}_{DLS}$; (c) \mathbf{q}_{IF} ; e (d) \mathbf{q}_{DLS} .

Em configurações de pouca manipulabilidade, onde o termo de amortecimento é adicionado, a convergência da inversa filtrada também se torna mais lenta, visto que estas configurações estão associadas a um ou mais valores singulares pequenos. A Figura 4.17 apresenta a variável de amortecimento δ utilizada pelo algoritmo DLS e o valor da função V_c , havendo uma similaridade entre ambos os sinais.

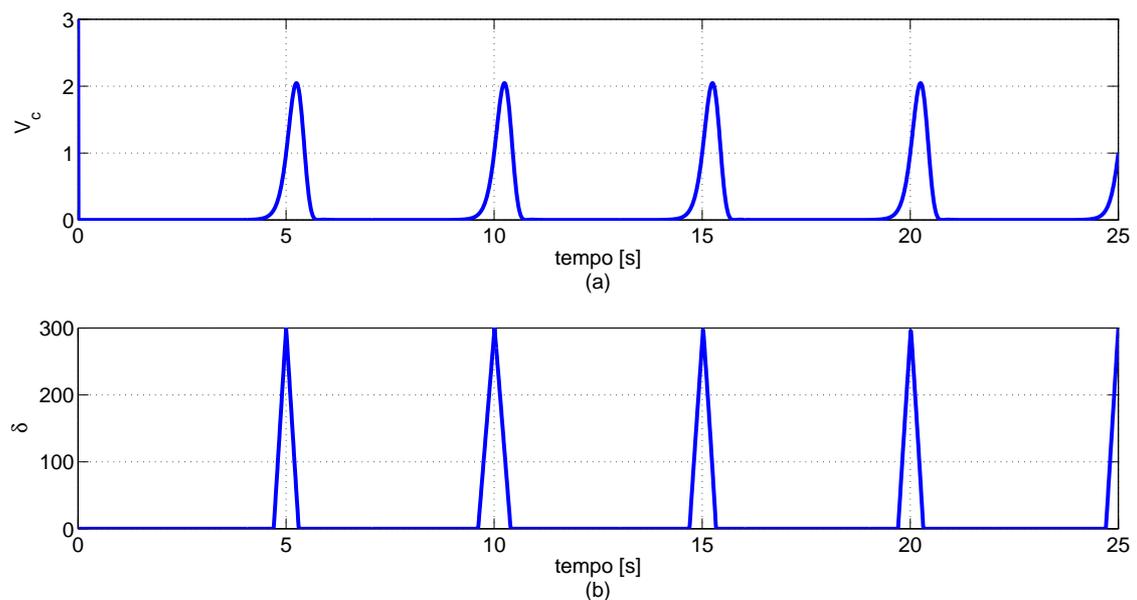


Figura 4.17: Simulação 11: (a) função V_c e (b) δ .

Para uma condição inicial $\mathbf{q}(0) = [0 \ (\pi/2 - 0.1) \ (0.1 - \pi/2)]^T \text{rad}$, dentro da região de pouca manipulabilidade definida por ω_0 , o algoritmo DLS apresenta desempenho pouco satisfatório, sendo incapaz de rastrear a posição desejada (Figuras 4.18 e 4.19). Neste caso, $\delta > 0$ para todo tempo t (Figura 4.19.c), garantindo uma solução viável porém pouco precisa.

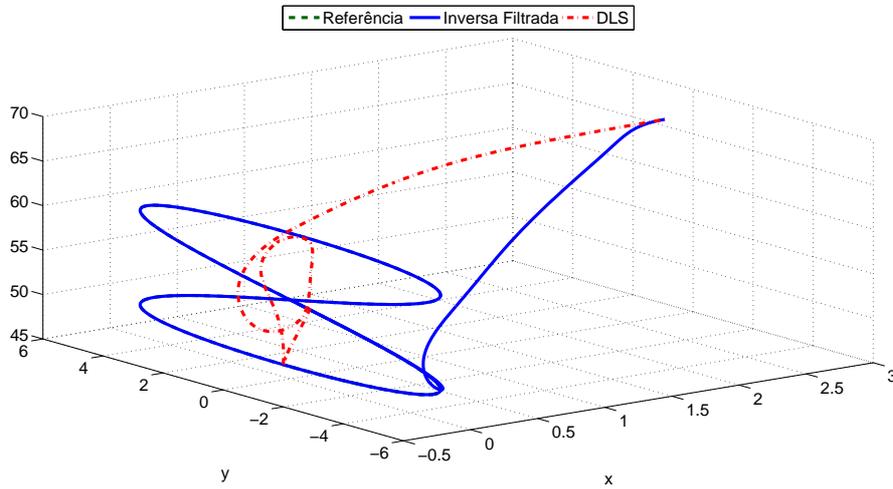


Figura 4.18: Simulação 11b: trajetórias no espaço cartesiano.

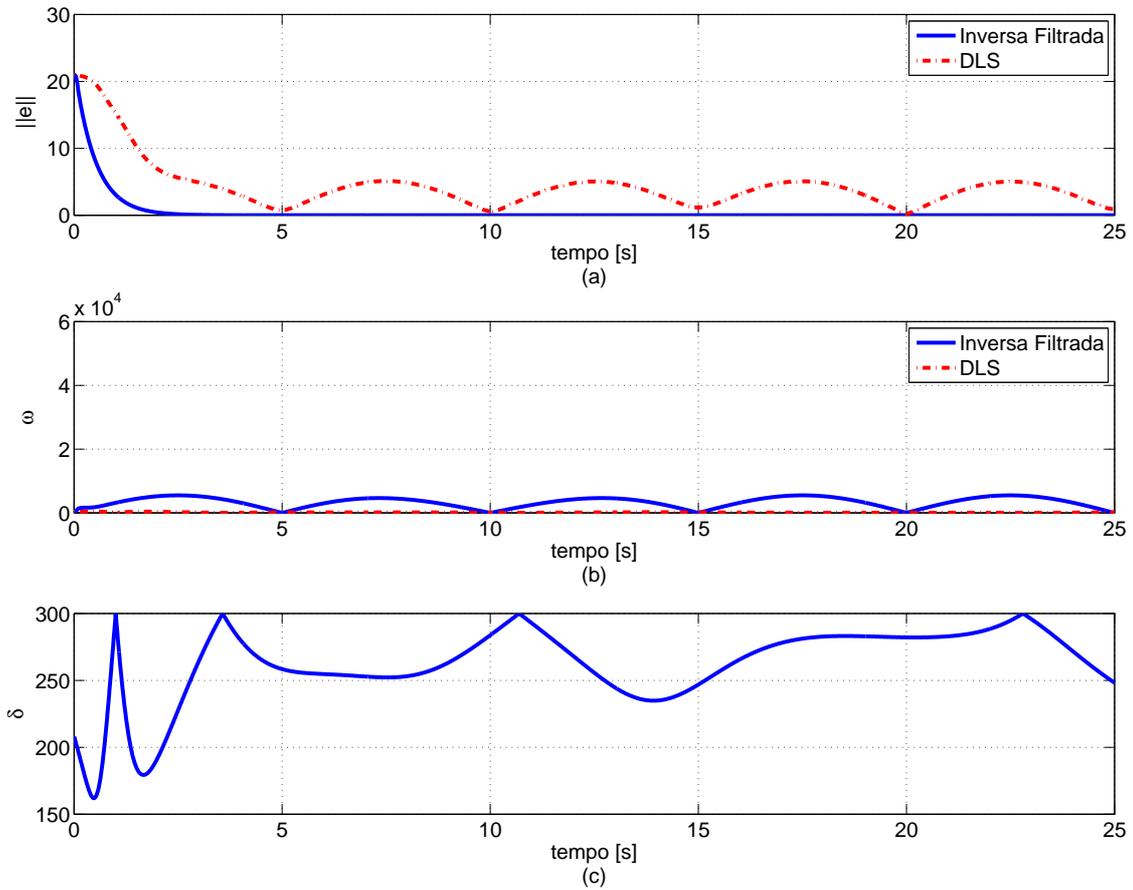


Figura 4.19: Simulação 11b: (a) norma do erro; (b) manipulabilidade; e (c) δ .

Simulação 12

Neste caso, analisa-se o desempenho dos métodos para um caso extremo, em que trechos da trajetória desejada não pertencem ao espaço de trabalho do manipulador, de forma que o erro não pode ser anulado. A trajetória considerada (Trajetória 8) é similar à considerada na Simulação 10, porém deslocada no eixo x .

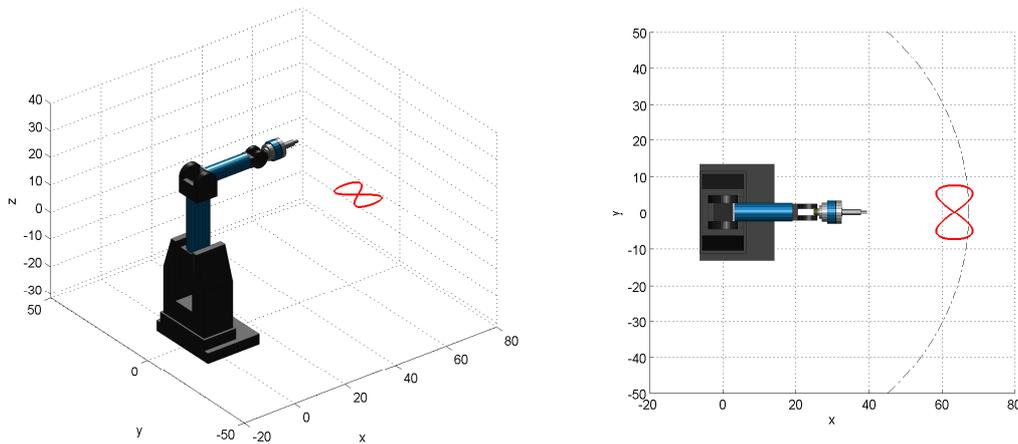


Figura 4.20: Simulação 12: manipulador Zebra-ZERO e trajetória desejada em perspectiva e visão superior.

As Figuras 4.21 e 4.22 apresentam as trajetórias desenvolvidas no espaço cartesiano. As normas dos erros e as medidas de manipulabilidade são apresentadas na Figura 4.23. Ambos os algoritmos permitem o rastreamento da parcela viável da trajetória desejada. Porém, em determinados instantes, esta se torna inalcançável e ambos os algoritmos levam o manipulador a configurações próximas de singularidades de fronteira.

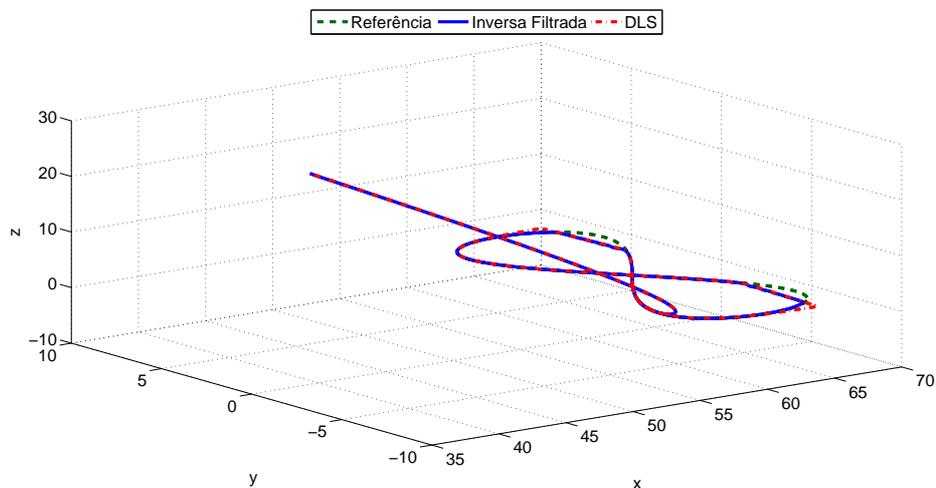


Figura 4.21: Simulação 12: trajetórias no espaço cartesiano.

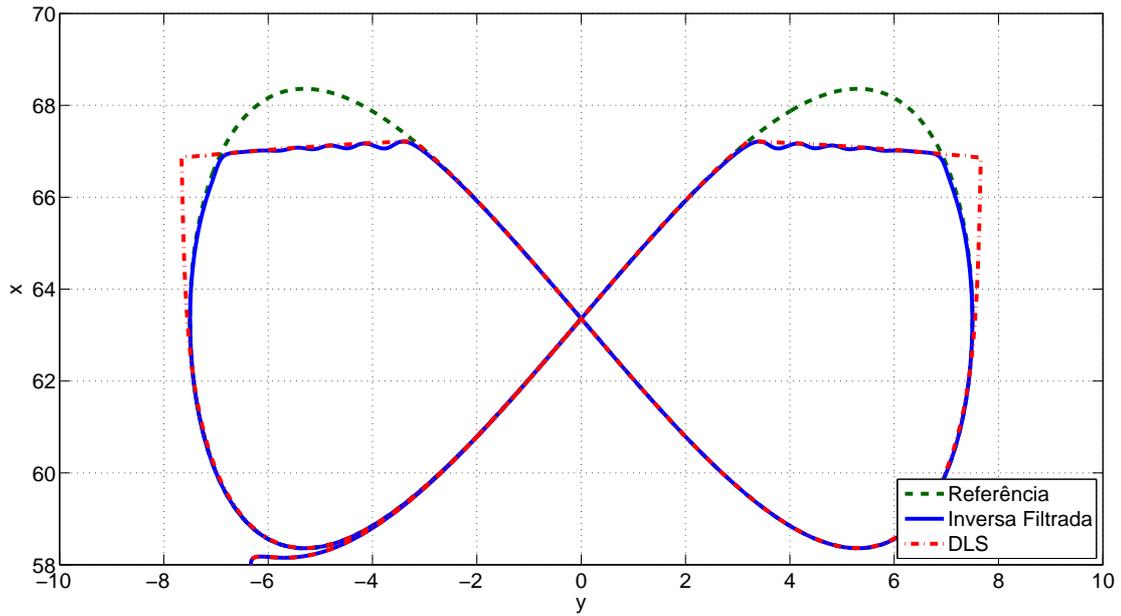


Figura 4.22: Simulação 12: trajetórias no plano xy .

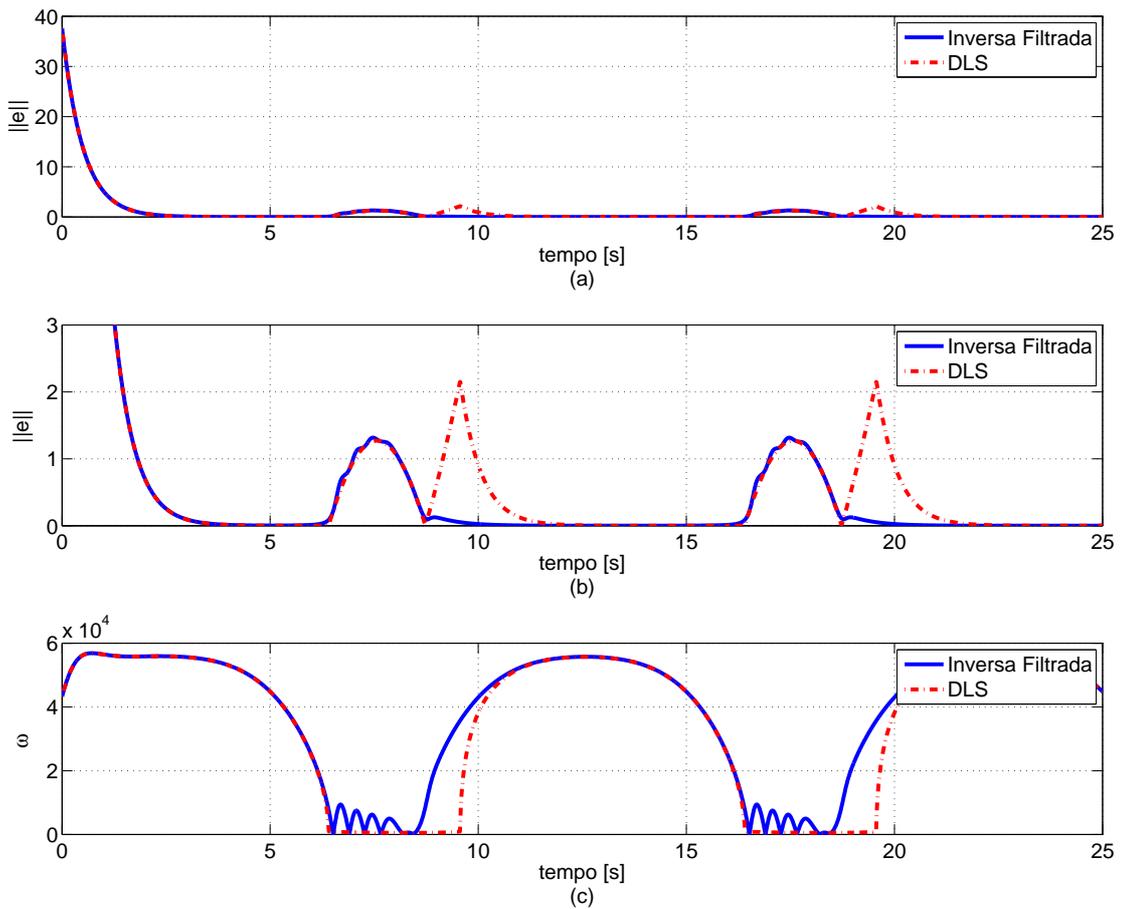


Figura 4.23: Simulação 12: (a) norma do erro; (b) norma do erro (escala ampliada); e (c) manipulabilidade.

As variáveis $\dot{\mathbf{q}}$ e \mathbf{q} obtidas são apresentadas na Figura 4.24. Nos trechos em que a trajetória se torna não-factível, a utilização da inversa filtrada resultou em oscilações indesejadas nas velocidades das juntas e, conseqüentemente, na posição do efetuador. O algoritmo DLS, por sua vez, resultou em picos de velocidade na vizinhança da região de chaveamento de δ .

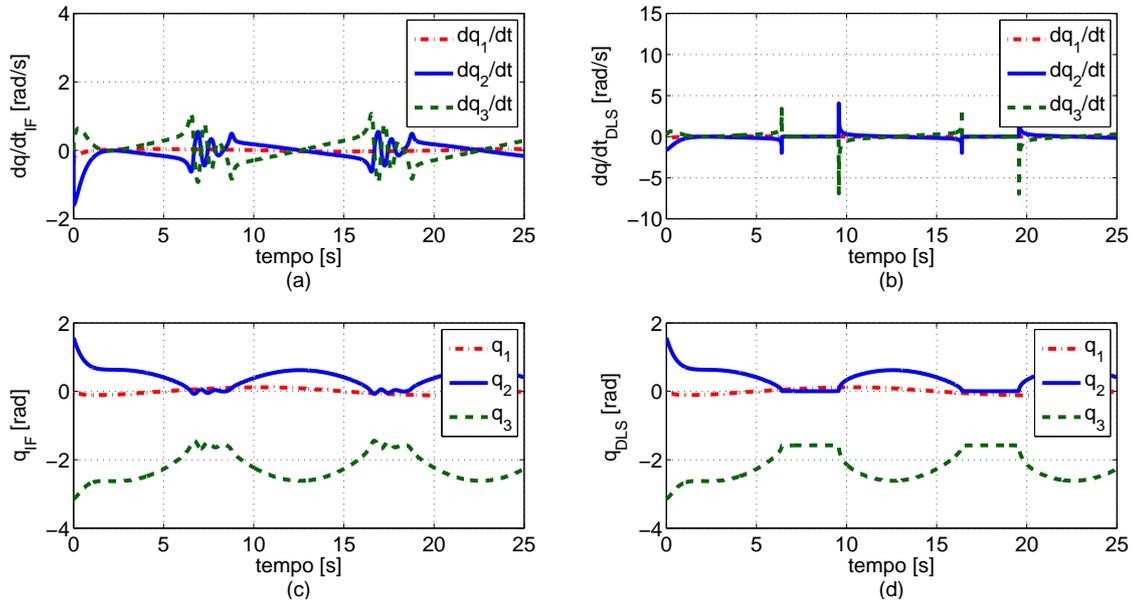


Figura 4.24: Simulação 12: variáveis (a) $\dot{\mathbf{q}}_{IF}$; (b) $\dot{\mathbf{q}}_{DLS}$; (c) \mathbf{q}_{IF} ; e (d) \mathbf{q}_{DLS} .

Nota-se também que, uma vez que a trajetória é novamente alcançável, o método DLS apresenta um desvio considerável. Este desvio pode ser atribuído ao fato de que o chaveamento de δ é realizado com base em medidas de manipulabilidade, isto é, elipsoides no espaço operacional. Os sinais V_c e δ são mostrados na Figura 4.25.

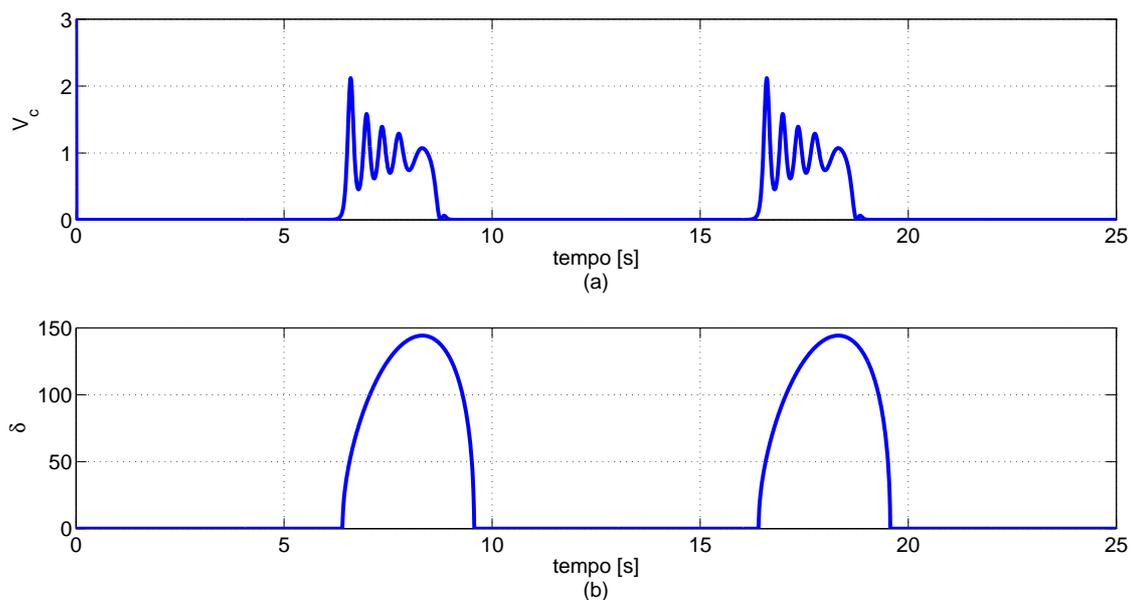


Figura 4.25: Simulação 12: (a) função V_c e (b) δ .

Na Figura 4.26 são apresentadas as trajetórias obtidas com o método da inversa filtrada para um trecho não-factível e diferentes ganhos $\mathbf{\Gamma} = \gamma \mathbf{I}$.

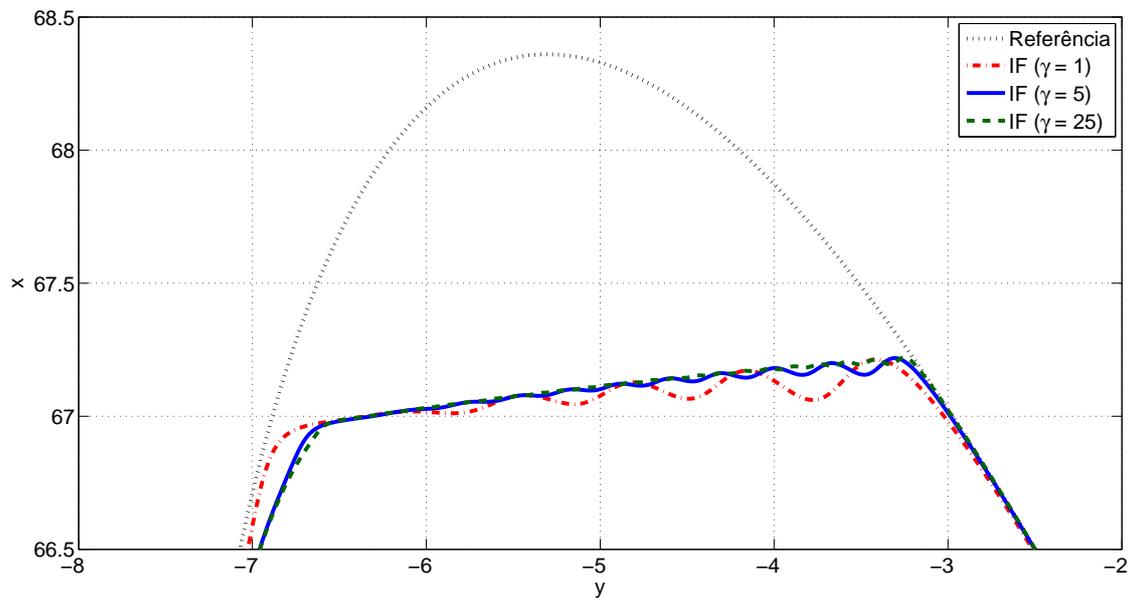


Figura 4.26: Simulação 12: trecho não-factível e diferentes ganhos γ .

O ganho γ utilizado está diretamente relacionado ao desempenho do algoritmo proposto e quanto maior este ganho, menor será a amplitude das oscilações observadas no espaço operacional. Porém, ao analisar as variáveis $\dot{\mathbf{q}}$ desenvolvidas a nível de juntas, verifica-se também um aumento considerável na frequência das oscilações, geralmente inviáveis e indesejadas em aplicações práticas. A Figura 4.27 apresenta as velocidades e as variáveis de juntas obtidas para $\gamma = 25$.

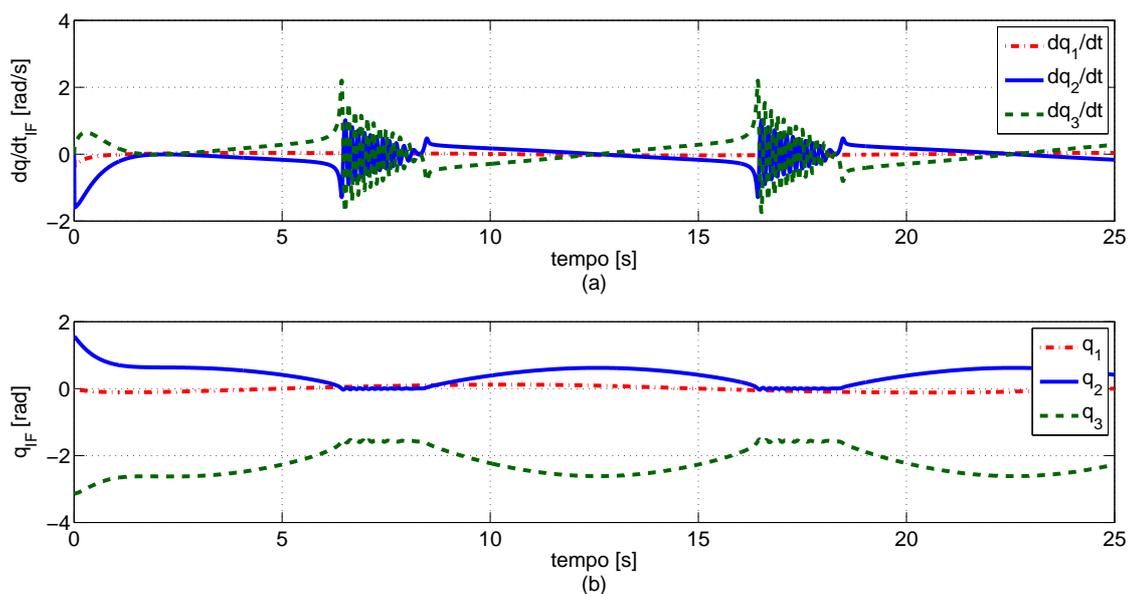


Figura 4.27: Simulação 12: variáveis (a) $\dot{\mathbf{q}}_{IF}$ e (b) \mathbf{q}_{IF} para $\gamma = 25$.

4.6 Lei de controle modificada

Considere inicialmente o ganho escalar k e sua função inversa k^{-1} . A inversa é uma função ímpar de k e possui o mesmo sinal do ganho k . Porém, esta igualdade de sinal não é verificada para a inversa filtrada θ . Com o objetivo de recuperar esta propriedade, é proposta uma modificação na forma como θ é aplicada na lei de controle u em (4.2). A nova proposta é dada por

$$\theta_M = \theta^2 k. \quad (4.64)$$

Desta forma, tem-se que $\text{sign}(\theta_M) = \text{sign}(k)$. A Figura 4.28 ilustra os comportamentos de θ e θ_M para um ganho k evoluindo do semi-plano positivo para o negativo.

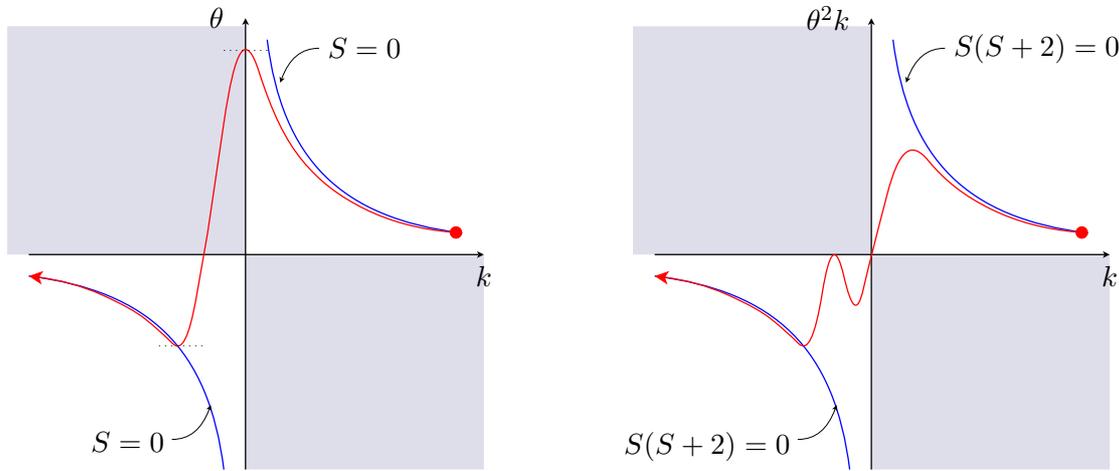


Figura 4.28: Planos $\theta \times k$ e $\theta_M \times k$ (curvas típicas).

A utilização da proposta modificada resulta no mapeamento de todo o eixo θ na origem do plano $\theta_M \times k$. Além disso, uma vez que os 2º e 4º quadrantes são mapeados nos 3º e 1º quadrantes, a hipérbole passa a representar também o lugar geométrico de $S = -2$. Note porém que para $S = -2$ e $k \neq 0$, $\dot{\theta} \neq 0$. Para o caso multivariável uma modificação similar é proposta. A matriz Θ_M é dada por

$$\Theta_M = \Theta \Theta^T \mathbf{K}^T. \quad (4.65)$$

Note que $\Theta \Theta^T$ é semi-definida positiva e simétrica. Considere o problema de controle dado pela equação $\dot{\mathbf{e}} = \dot{\mathbf{y}}_d - \mathbf{K} \mathbf{u}$ e a função de Lyapunov $2V_e = \mathbf{e}^T \mathbf{\Lambda}^T \mathbf{e}$, onde $\mathbf{\Lambda}$ é PDS. Sua derivada temporal ao longo das trajetórias do sistema é dada por $\dot{V}_e = \mathbf{e}^T \mathbf{\Lambda}^T (\dot{\mathbf{y}}_d - \mathbf{K} \mathbf{u})$. Assim, para $\mathbf{u} = \mathbf{u}_M = \Theta \Theta^T \mathbf{K}^T \boldsymbol{\nu}$, tem-se

$$\dot{V}_e = \mathbf{e}^T \mathbf{\Lambda}^T (\mathbf{I}_m - \mathbf{N} \mathbf{N}^T) \dot{\mathbf{y}}_d - \underbrace{\mathbf{e}^T \mathbf{\Lambda}^T \mathbf{N} \mathbf{N}^T \mathbf{\Lambda} \mathbf{e}}_{\Phi}, \quad (4.66)$$

onde $\mathbf{N} = \mathbf{K}\Theta$. O termo Φ corresponde a uma parcela semi-definida negativa, independente da qualidade da estimativa dada pelo algoritmo da inversa filtrada (4.32). Assim, a proposta modificada para solução do problema de cinemática inversa é dada por

$$\dot{\mathbf{q}}_M = \Theta_M \boldsymbol{\nu} = \Theta \Theta^T \mathbf{J}^T \boldsymbol{\nu}, \quad (4.67)$$

A relação entre as velocidades efetuada $\dot{\mathbf{x}}$ e demandada $\boldsymbol{\nu}$ é então obtida:

$$\dot{\mathbf{x}} = \mathbf{J}\Theta\Theta^T\mathbf{J}^T\boldsymbol{\nu} = \mathbf{M}\boldsymbol{\nu}. \quad (4.68)$$

Note que $\mathbf{M} = \mathbf{M}^T \geq 0$, apresentando forma similar à matriz de projeção no espaço coluna de \mathbf{J} , dada por $\mathbf{P} = \mathbf{J}(\mathbf{J}^T\mathbf{J})^{-1}\mathbf{J}^T$. Para $\boldsymbol{\nu} = \boldsymbol{\nu}_c + \boldsymbol{\nu}_n$, onde $\boldsymbol{\nu}_c \in \text{Col}(\mathbf{J})$ e $\boldsymbol{\nu}_n \in \text{Nul}(\mathbf{J}^T)$, tem-se $\dot{\mathbf{q}}_M(\boldsymbol{\nu}) = \dot{\mathbf{q}}_M(\boldsymbol{\nu}_c)$, isto é, a utilização de $\dot{\mathbf{q}}_M(\boldsymbol{\nu}) = \Theta_M \boldsymbol{\nu}$ cancela as componentes de $\boldsymbol{\nu}$ no espaço nulo à esquerda de \mathbf{J} . Note também que \dot{V}_c é função da variável de erro ponderada $\mathbf{e}_w = \mathbf{\Lambda}\mathbf{e}$. Desta forma, espera-se que para casos em que a variável de referência é inalcançável, a matriz $\mathbf{\Lambda}$ influencie na solução obtida, ponderando os diferentes objetivos de controle. Uma simulação envolvendo orientação e posição inatingíveis simultaneamente será realizada, servindo como análise preliminar desta propriedade. Porém, uma análise rigorosa é necessária e deve ser incluída em trabalhos futuros.

Simulação 13

Considere a mesma trajetória e os mesmos parâmetros da Simulação 12. As Figuras 4.29 e 4.30 apresentam as trajetórias obtidas com o algoritmo DLS e a partir da proposta modificada para dois ganhos de atualização $\mathbf{\Gamma} = \gamma\mathbf{I}$.

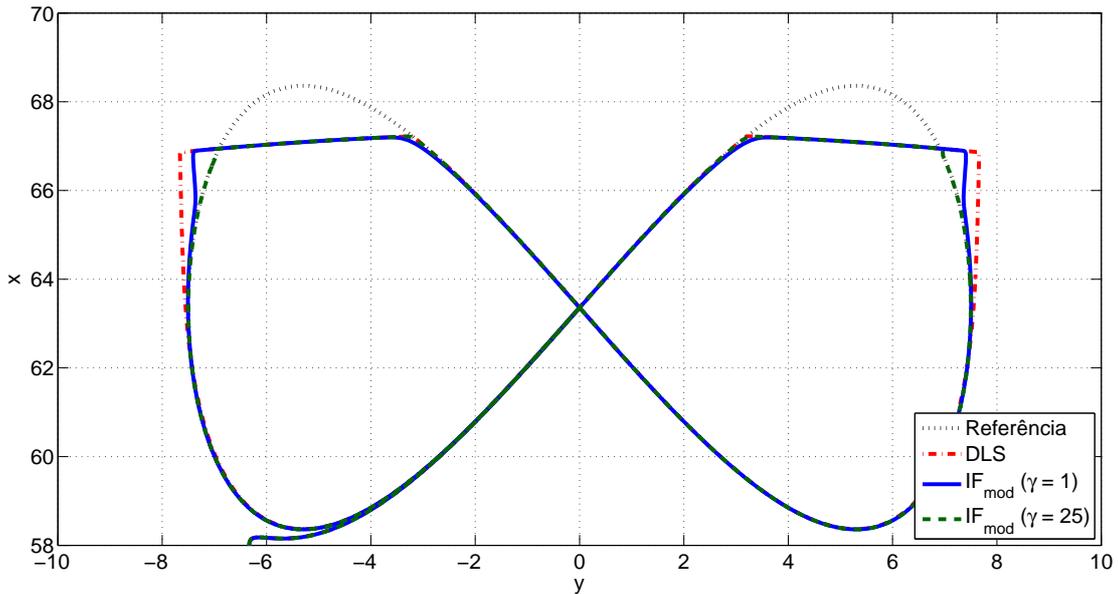


Figura 4.29: Simulação 13: trajetórias no plano xy .

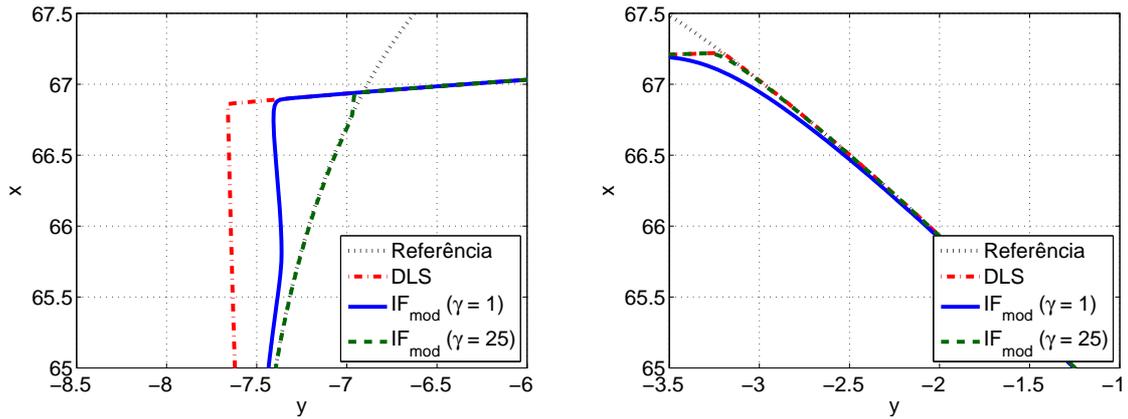


Figura 4.30: Simulação 13: trechos de transição no plano xy .

Para $\gamma = 1$, a solução obtida a partir da inversa filtrada é bastante similar à obtida pelo algoritmo DLS, apresentando um desvio considerável no momento em que a trajetória se torna novamente viável. Porém, nota-se que as variáveis $\dot{\mathbf{q}}$ obtidas com a inversa filtrada são mais suaves nos pontos de transição, notavelmente quando o manipulador se aproxima da singularidade (Figura 4.31). À medida que o manipulador se aproxima destas configurações singulares, a convergência de Θ torna-se mais lenta e os sinais de velocidade nas direções degeneradas são atenuados.

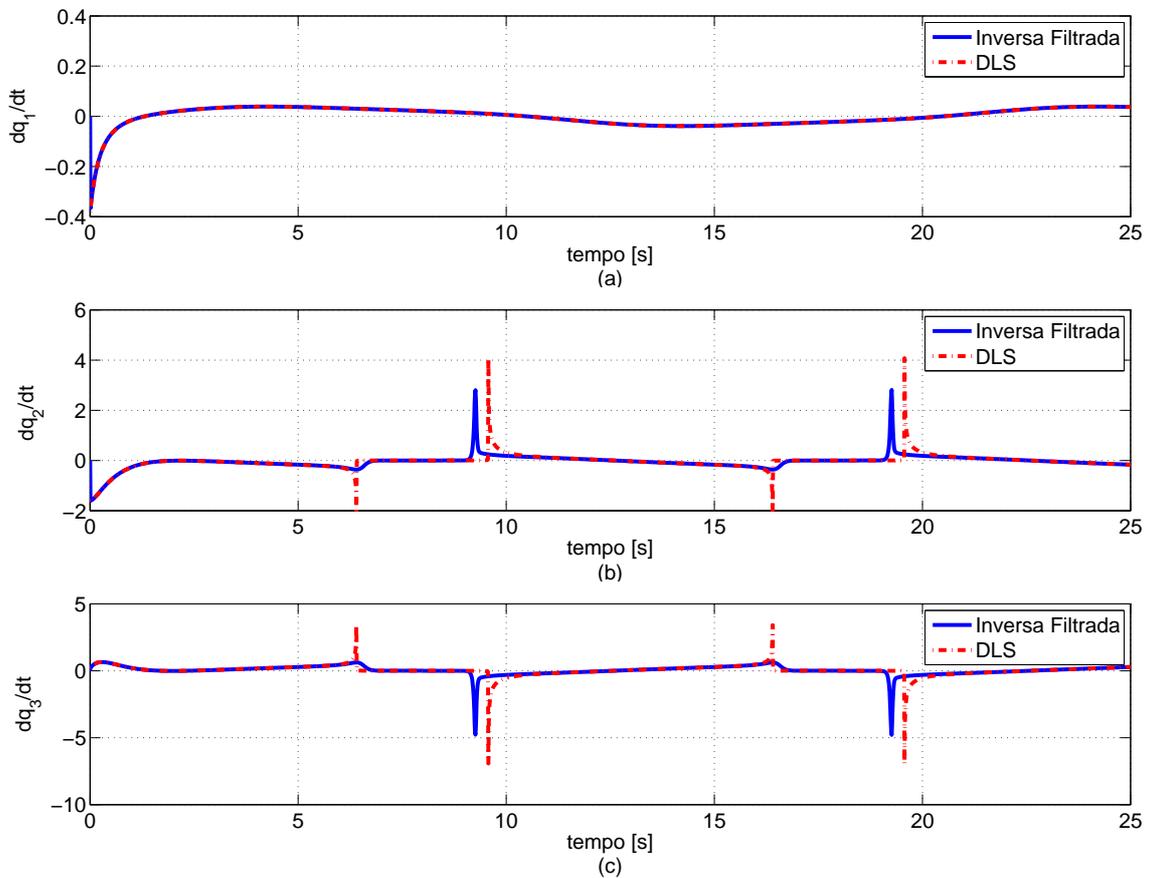


Figura 4.31: Simulação 13: variáveis (a) \dot{q}_1 ; (b) \dot{q}_2 ; e (c) \dot{q}_3 ($\gamma = 1$).

O ganho de atualização Γ está diretamente relacionado ao desempenho do algoritmo e, para ganhos elevados ($\gamma = 25$), os desvios de trajetória em pontos de transição são pequenos, como mostrado na Figura 4.30. Nota-se também que as velocidades das juntas obtidas não apresentam alterações significativas em suas amplitudes (Figuras 4.31 e 4.32), estando a principal alteração no instante de atuação.

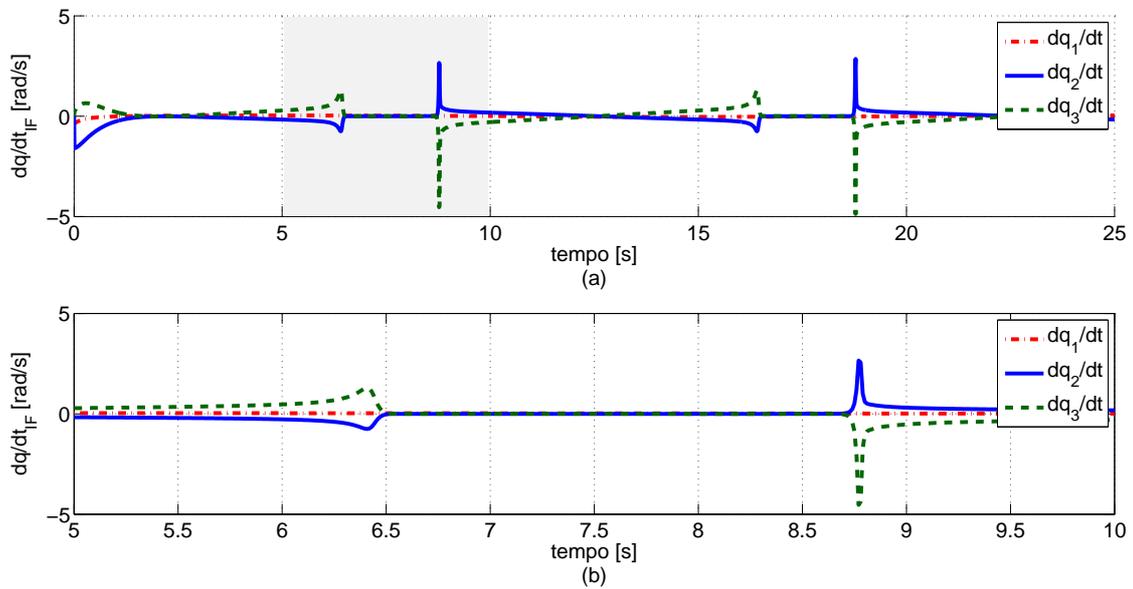


Figura 4.32: Simulação 13: $\dot{\mathbf{q}}$ em (a) escala normal e (b) ampliada ($5 < t < 10$).

Como esperado, a convergência das matrizes de erro se torna mais rápida para ganhos de atualização maiores (Figura 4.33.b). Desta forma, configurações de menor manipulabilidade são atingidas (Figura 4.33.a).

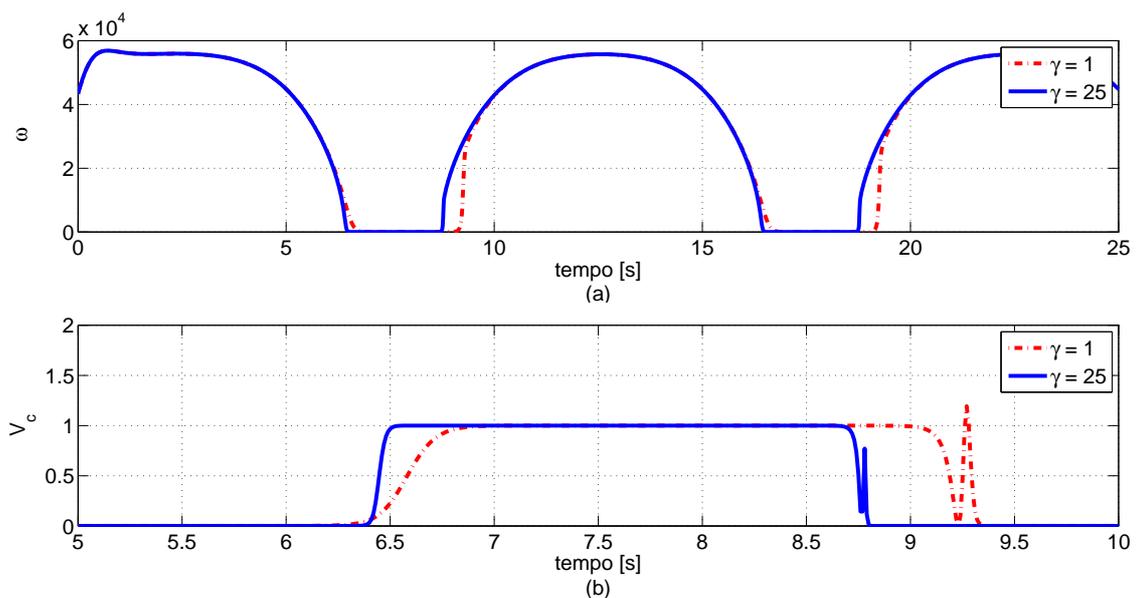


Figura 4.33: Simulação 13: (a) manipulabilidade e (b) função V_c ($6 < t < 10$).

Simulação 14

Considere novamente o manipulador antropomórfico Zebra-ZERO porém com atuação em todas as juntas (6DoF). No que se segue, são apresentados os resultados de simulação obtidos através do método proposto para regulação da orientação ($\mathbf{R}_d = \mathbf{R}_y(\pi)$) e rastreamento da posição do efetuador (Trajetória 11). Considera-se também $\mathbf{\Gamma} = 25\mathbf{I}$, $\mathbf{q}_0 = [0 \ \frac{\pi}{2} \ -\pi \ 0 \ -\frac{\pi}{2} \ 0]^T rad$, $\mathbf{\Lambda}_p = \lambda_p \mathbf{I}$, $\mathbf{\Lambda}_o = \lambda_o \mathbf{I}$ e a representação do erro de orientação através de quaternions unitários. A Figura 4.34 apresenta as normas dos erros para diferentes combinações de ganhos proporcionais λ_p e λ_o .

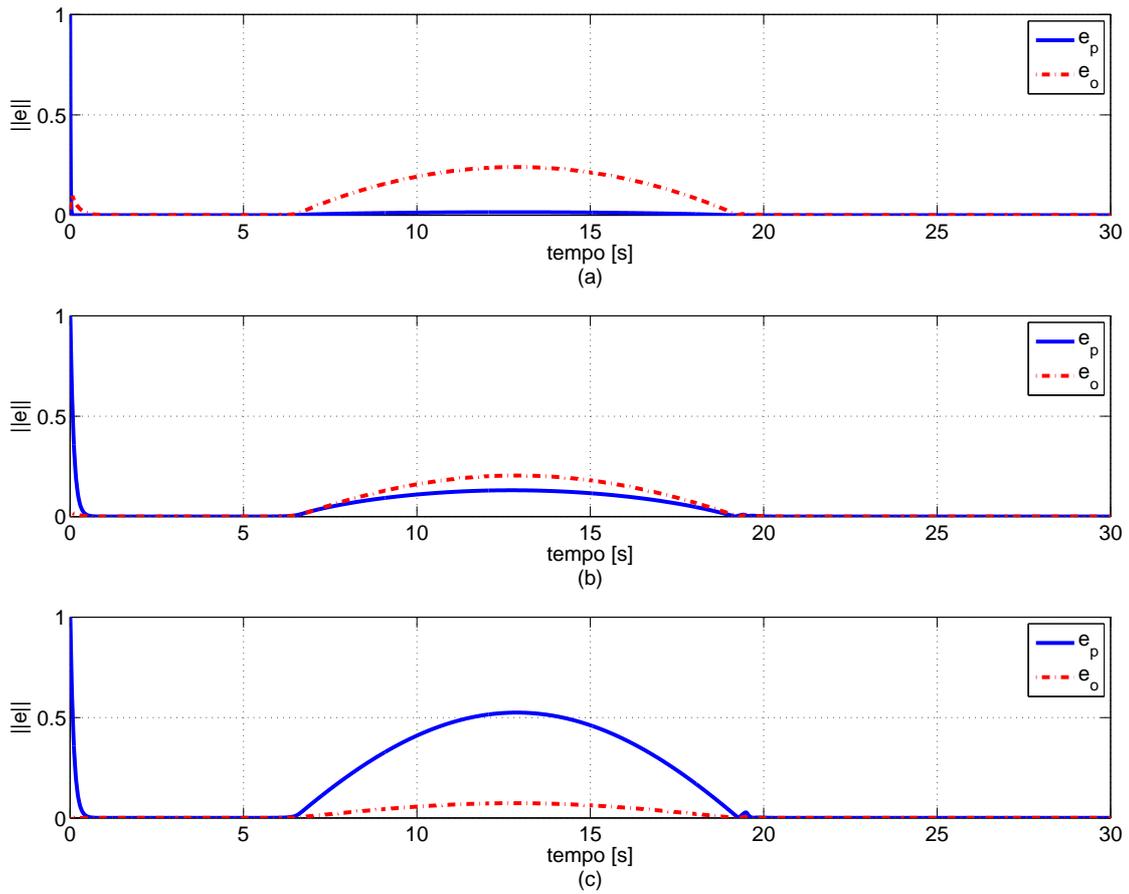


Figura 4.34: Simulação 14: (a) $\lambda_p > \lambda_o$; (b) $\lambda_p = \lambda_o$; e (c) $\lambda_p < \lambda_o$.

Para os trechos em que a posição e a orientação de referência não podem ser atingidas simultaneamente, a escolha dos ganhos permite uma ponderação dos erros obtidos no espaço operacional. Na Figura 4.34.a, o ganho para posição ($\lambda_p = 100$) é superior ao ganho para orientação ($\lambda_o = 10$) e, portanto, prioriza-se a posição nos trechos considerados. Por outro lado, na Figura 4.34.c, escolhe-se um ganho de orientação ($\lambda_o = 100$) superior ao ganho de posição ($\lambda_p = 10$), priorizando-se a orientação. A escolha de ganhos $\lambda_p = \lambda_o$ resulta em erros similares no espaço operacional (Figura 4.34.b). Note porém que os erros não são expressos pelas mesmas unidades e, com o intuito de tornar as grandezas similares para uma melhor análise

da relação de ganhos, a solução de cinemática inversa é obtida através da expressão em decímetros da posição e das dimensões do manipulador considerado. As Figuras 4.35.a e 4.35.b apresentam as trajetórias obtidas com prioridade para posição e orientação, respectivamente.

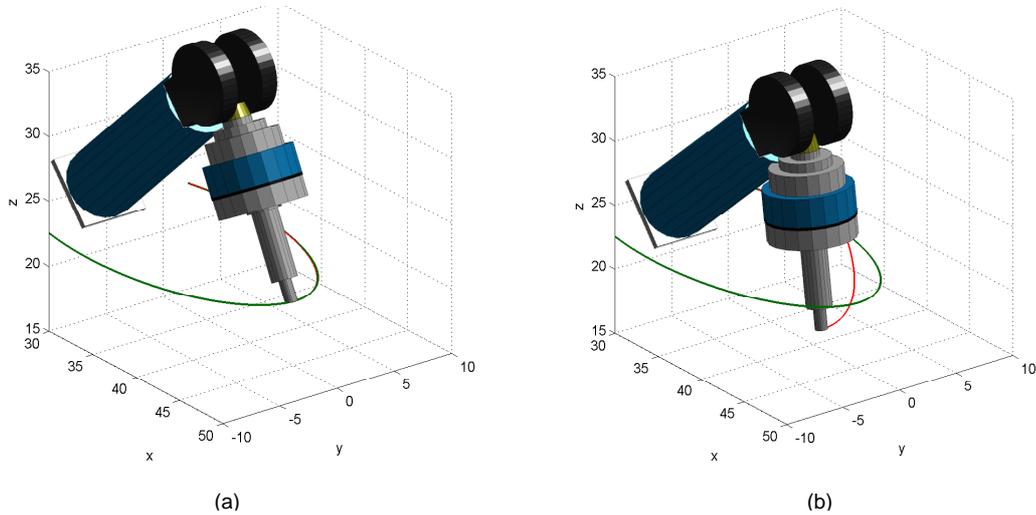


Figura 4.35: Simulação 14: trajetórias com prioridade para (a) posição e (b) orientação.

A priorização da orientação, a utilização de um ganho $\Lambda_p = \lambda_p \mathbf{I}$ (pesos iguais para cada direção) e uma posição de referência no plano $z_d = 20$ (cm) resultaram em uma trajetória que apresenta erros em todas as direções \mathbf{x} , \mathbf{y} e \mathbf{z} (Figura 4.35.b). Neste contexto, a utilização de uma matriz de ganho Λ_p diagonal com pesos diferentes para cada direção permitirá a priorização de uma componente da variável de erro de posição (Figura 4.36).

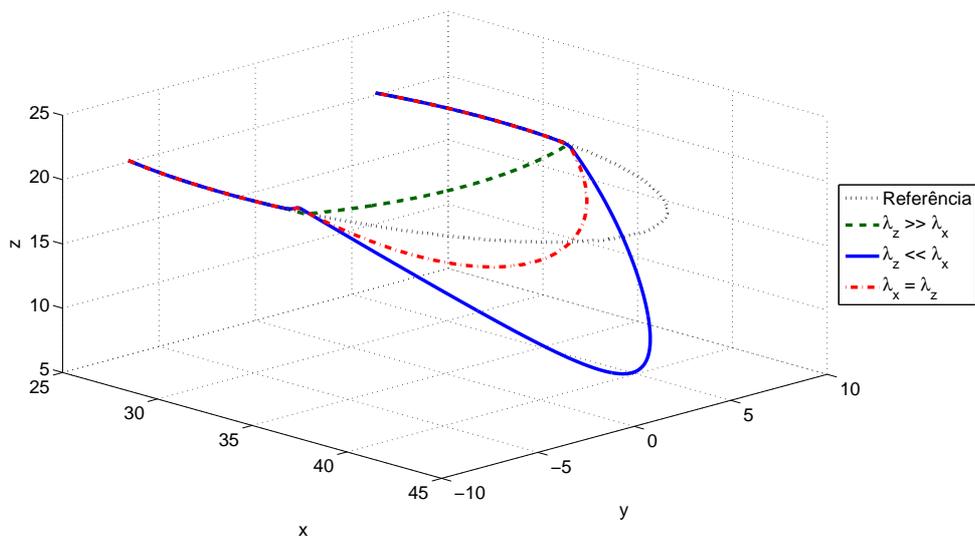


Figura 4.36: Simulação 14: diferentes pesos para erros em \mathbf{x} e \mathbf{z} .

4.7 Função objetivo e Jacobiano ampliado

O algoritmo da inversa filtrada fornece uma solução para o problema de cinemática inversa de manipuladores robóticos mesmo quando posturas desejadas inviáveis ou inalcançáveis são consideradas. Como visto, o método proposto permite priorizar um dos objetivos de controle através do ajuste de ganhos. Nesta seção, utiliza-se o método proposto para satisfazer também uma restrição adicional ao sistema, expressada pela função escalar $f(\mathbf{q})$. O objetivo de controle para o problema ampliado pode ser formalizado como

$$\begin{bmatrix} \mathbf{x} \\ f \end{bmatrix} \rightarrow \begin{bmatrix} \mathbf{x}_d \\ 0 \end{bmatrix}, \quad (4.69)$$

onde $\mathbf{x} \in \mathbb{R}^m$ e $\mathbf{x}_d \in \mathbb{R}^m$ correspondem às posturas atual e desejada do efetuador, respectivamente. A partir da derivação temporal de (4.69), tem-se

$$\begin{bmatrix} \dot{\mathbf{x}} \\ \dot{f} \end{bmatrix} = \underbrace{\begin{bmatrix} \mathbf{J}(\mathbf{q}) \\ \mathbf{J}_f(\mathbf{q}) \end{bmatrix}}_{\mathbf{J}_+(\mathbf{q})} \dot{\mathbf{q}}, \quad (4.70)$$

onde $\mathbf{J}_f(\mathbf{q}) = \partial f / \partial \mathbf{q} \in \mathbb{R}^{1 \times n}$. Note que $\mathbf{J}_+(\mathbf{q}) \in \mathbb{R}^{m' \times n}$ e $m' = m + 1$. Para problemas de cinemática inversa bem definidos¹, tem-se $m \leq n$. Desta forma, para o problema ampliado, é possível que $m' > n$, justificando a utilização da matriz de erro à direita, definida em (4.18). Assim, de forma similar à (4.67), a solução para o problema ampliado de cinemática inversa é dada por

$$\dot{\mathbf{q}} = \Theta_+ \Theta_+^T \mathbf{J}_+^T \boldsymbol{\nu}_+, \quad (4.71)$$

onde

$$\boldsymbol{\nu}_+ = \begin{bmatrix} \boldsymbol{\nu} \\ \nu_f \end{bmatrix} = \begin{bmatrix} \mathbf{x}_d + \boldsymbol{\Lambda} \mathbf{e} \\ -f \end{bmatrix} \quad (4.72)$$

é o sinal de referência. A matriz Θ_+ é a inversa filtrada de \mathbf{J}_+ e $\boldsymbol{\Lambda} = \boldsymbol{\Lambda}^T > 0$. Inicialmente, considerou-se $\nu_f = -\lambda_f f$, onde $\lambda_f \geq 0$ seria o ganho proporcional referente à função objetivo. Porém, em (4.72), escolhe-se $\lambda_f = 1$ e, como será visto, a ponderação será incluída no cálculo de f . A principal vantagem desta escolha alternativa é que, para os casos em que não há restrição ou esta apresenta baixa prioridade, os elementos de \mathbf{J}_f são nulos ou pequenos. No que se segue, são apresentadas duas propostas de funções objetivo, definidas a partir de variáveis de desvio angular ou de posição.

¹Um problema de rastreamento da posição $\mathbf{p}_d \in \mathbb{R}^3$ constitui um problema de cinemática inversa mal definido para um manipulador planar, por exemplo.

4.7.1 Função f para limite de juntas

A função objetivo é definida a partir de potências pares das variáveis de desvio das juntas normalizadas $(\Delta q_i/\zeta_i)$, sendo apresentada e ilustrada abaixo. O objetivo é manter a variável de junta q_i no intervalo $[q_{id} - \zeta_i, q_{id} + \zeta_i]$.

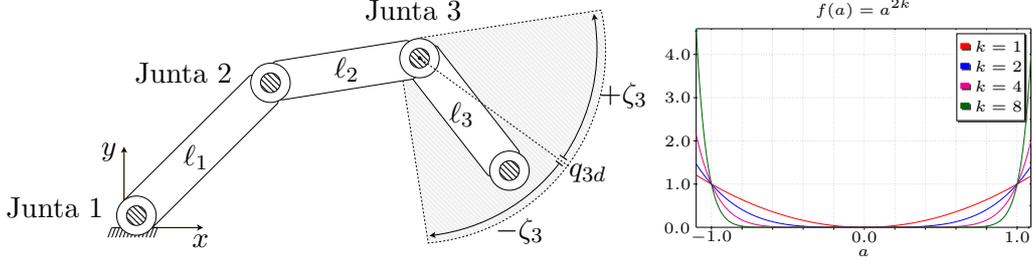


Figura 4.37: Função objetivo para limite de juntas.

$$f(q_1, \dots, q_n) = \sum_{i=1}^n \alpha_i \left(\frac{q_i - q_{id}}{\zeta_i} \right)^{2k_i} = \sum_{i=1}^n \alpha_i \left(\frac{\Delta q_i}{\zeta_i} \right)^{2k_i}, \quad (4.73)$$

onde $\alpha_i \geq 0$ e $\zeta_i > 0$. Ganhos k_i elevados impõem pouca restrição aos ângulos das juntas, desde que permaneçam dentro das faixas determinadas, e uma ação de controle acentuada uma vez fora destas faixas.

4.7.2 Função f para desvio de obstáculo

Neste caso, a função objetivo é inspirada na função Gaussiana e definida a partir das distâncias dos pontos \mathbf{p}_j ao centro do obstáculo $\boldsymbol{\mu}$. As curvas de nível são elipsoides, definidos pela matriz $\mathbf{M}_f = \mathbf{M}_f^T = \mathbf{R}_f \mathbf{D}_f^{-1} \mathbf{R}_f^T > 0$. A matriz de rotação \mathbf{R}_f define os eixos do elipsoide e a matriz \mathbf{D}_f apresenta elementos positivos em sua diagonal, cujas raízes são proporcionais ao alongamento da curva de nível nesses eixos.

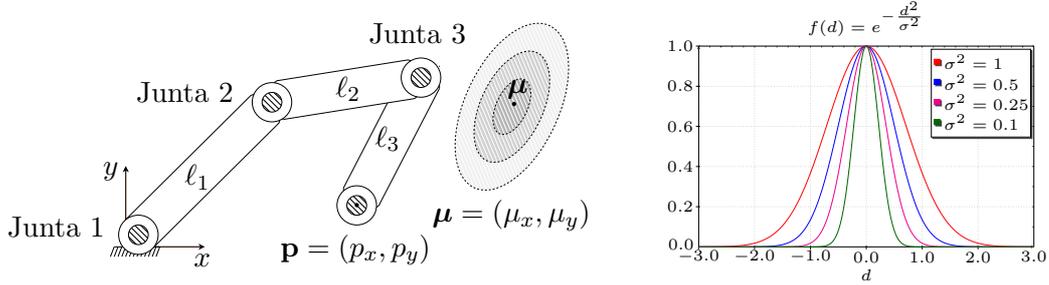


Figura 4.38: Função objetivo para desvio de obstáculo.

$$f(\mathbf{p}_1, \dots, \mathbf{p}_c) = \sum_{j=1}^c \alpha_j e^{-(\mathbf{p}_j - \boldsymbol{\mu})^T \mathbf{M}_f (\mathbf{p}_j - \boldsymbol{\mu})} = \sum_{j=1}^c \alpha_j e^{-(\Delta \mathbf{p}_j)^T \mathbf{D}_f^{-1} (\Delta \mathbf{p}_j)}, \quad (4.74)$$

onde $\alpha_j \geq 0$ e $(\Delta \mathbf{p}_j)' = \mathbf{R}_f^T \Delta \mathbf{p}_j$ é a representação de $\Delta \mathbf{p}_j = \mathbf{p}_j - \boldsymbol{\mu}$ no sistema de coordenadas rotacionado por \mathbf{R}_f , conforme (2.6).

Simulação 15

Considere o problema de rastreamento de posição (Trajetória 1) para o manipulador planar 3R da Figura 2.9 e uma função f para a 2ª junta, definida por $q_{2d} = -1 \text{ rad}$, $\zeta_2 = 0.5 \text{ rad}$, $\alpha_{f2} = 5$ e $k = 10$. Os resultados obtidos com o método proposto para $\Gamma = 5\mathbf{I}$ e $\Lambda_p = 5\mathbf{I}$ são apresentados a seguir.

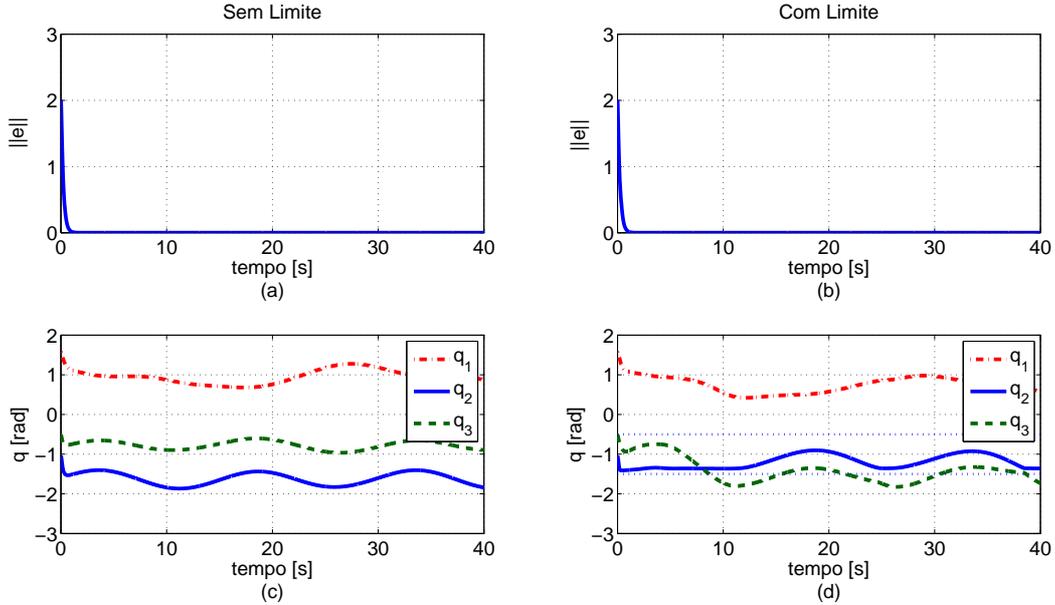


Figura 4.39: Simulação 15: normas dos erros e ângulos das juntas.

A utilização de $k = 10$ resulta em f praticamente nula para $q_2 \in [q_{2d} - \zeta_2, q_{2d} + \zeta_2]$ e elevada para ângulos fora desta faixa. Como visto na Figura 4.39, o método proposto permite manter o ângulo da 2ª junta na faixa desejada com erro de rastreamento praticamente nulo para todo tempo t . Para a trajetória considerada, a redução de ζ_2 torna o problema mais restrito e o rastreamento da posição se torna inviável sem que f atinja valores elevados. A Figura 4.40 apresenta a norma do erro e as variáveis de juntas obtidas para $\zeta_2 = 0.2 \text{ rad}$. Nota-se que o ângulo da 2ª junta permanece na faixa definida, porém são observados pequenos erros de rastreamento.

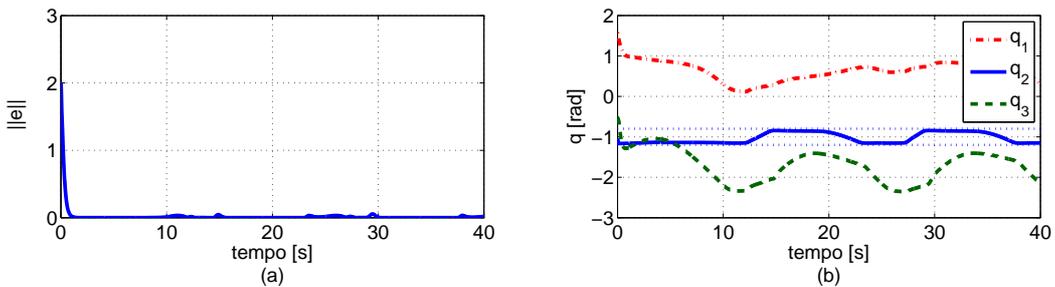


Figura 4.40: Simulação 15: norma do erro e ângulos das juntas para $\zeta_2 = 0.2 \text{ rad}$.

Simulação 16

Considere agora uma função objetivo definida no espaço operacional por $\boldsymbol{\mu} = (\mu_x, \mu_y) = (2, 0.75)$, $\mathbf{R}_f = \mathbf{I}$ e $\mathbf{D}_f = (0.2)^2\mathbf{I}$, avaliada no efetuador. Os resultados obtidos com o algoritmo da inversa filtrada para $\boldsymbol{\Gamma} = 5\mathbf{I}$, $\boldsymbol{\Lambda} = 5\mathbf{I}$ e rastreamento da Trajetória 1 são apresentados na Figura 4.41.

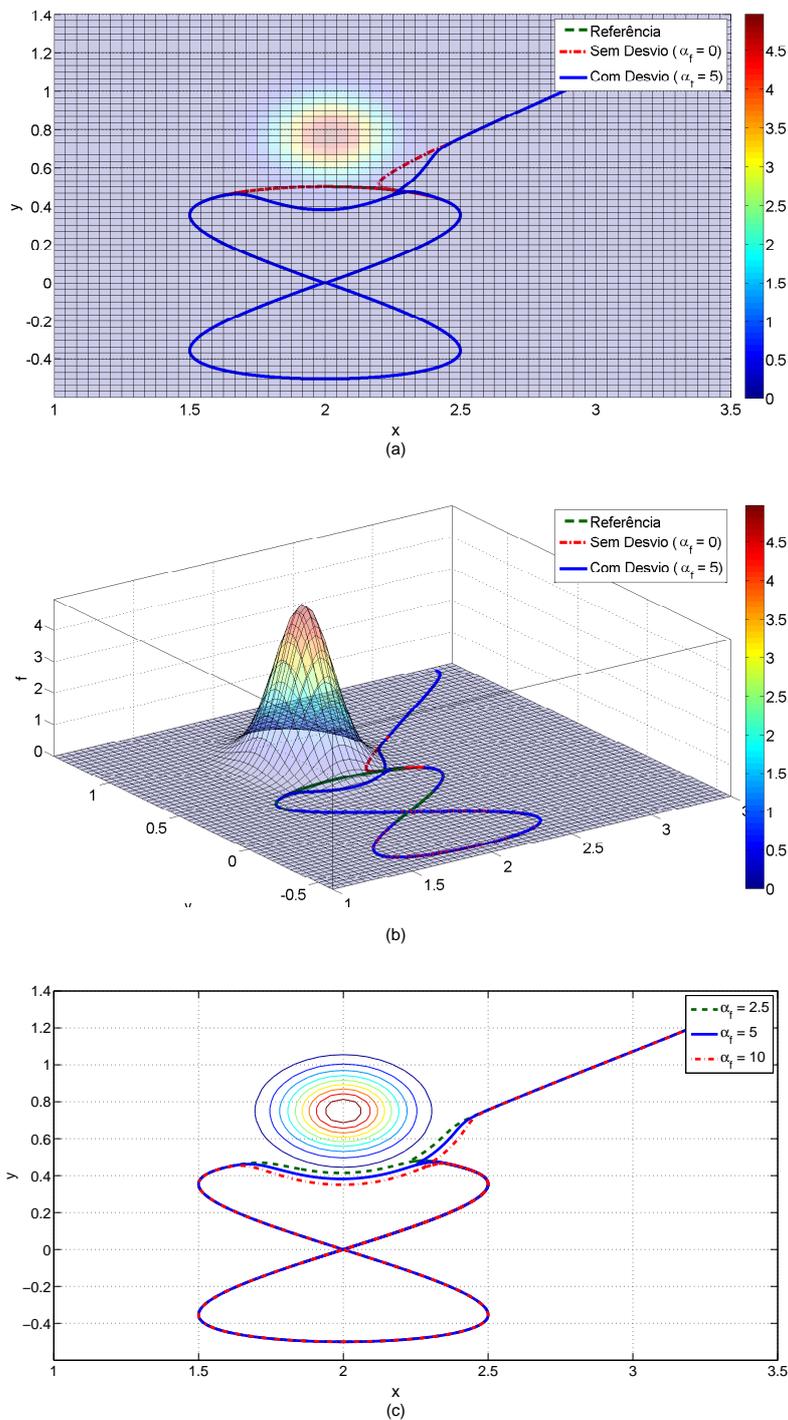


Figura 4.41: Simulação 16: (a) trajetórias com e sem desvio; (b) dimensão ampliada (função f); e (c) desvio para diferentes ganhos α_f .

A utilização de $\alpha_f > 0$ permite que o efetuador desvie do obstáculo definido pela função f , sendo observado um erro de posição (Figuras 4.41.a e 4.41.b). A Figura 4.41.c apresenta as trajetórias obtidas para diferentes ganhos α_f e curvas de nível da função f normalizada (f/α_f). Como esperado, quanto maior o ganho α_f considerado, maior o desvio. A função objetivo f pode ser avaliada em pontos intermediários, ao longo dos elos do robô, permitindo sua reconfiguração interna e evitando colisões dos elos com obstáculos. A Figura 4.42 apresenta resultados de simulação para rastreamento da Trajetória 5 e diferentes posições μ .

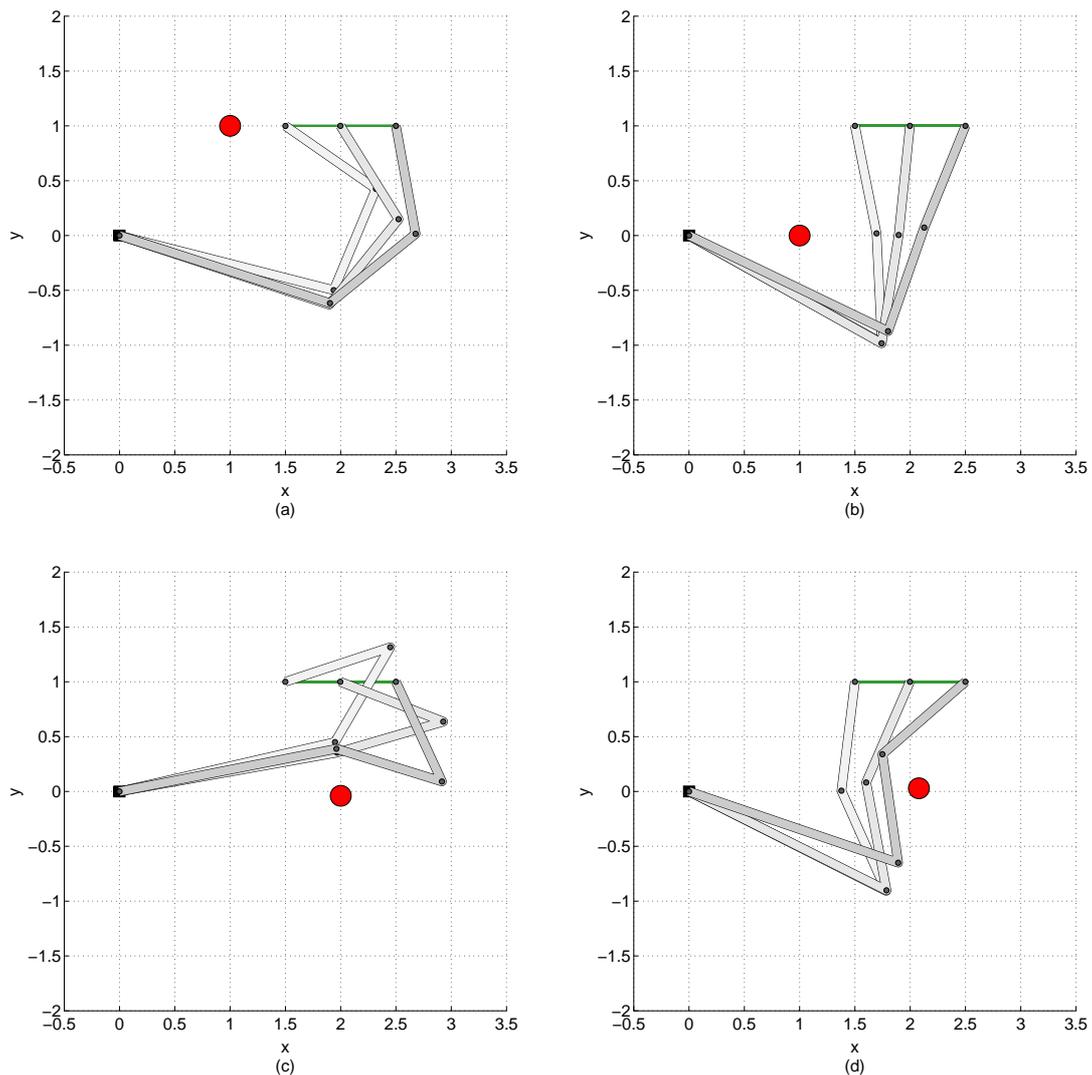


Figura 4.42: Obstáculo operacional e reconfiguração interna.

Como previsto, o manipulador se reconfigura internamente, permitindo o rastreamento com erro praticamente nulo desde que o obstáculo esteja afastado da posição de referência do efetuador. Para obstáculos pequenos, o número de pontos intermediários em que f é avaliada deve ser elevado de forma a garantir o bom desempenho do algoritmo proposto.

Em ambientes industriais, manipuladores cooperativos dividem um espaço de trabalho ou uma parcela deste. Neste caso, colisões são possíveis e representam eventos indesejados. Uma estratégia para evitar tais colisões é definir para um dos manipuladores uma função objetivo $f = \sum f_i$ (variante no tempo) que represente o outro manipulador (Figuras 4.43 e 4.44).

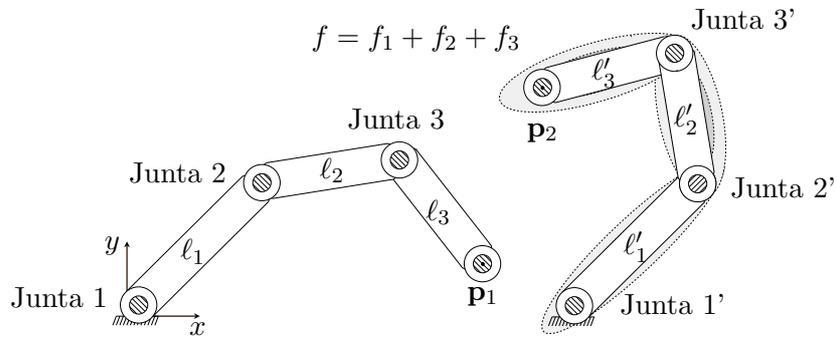


Figura 4.43: Ambiente industrial: manipuladores cooperativos.

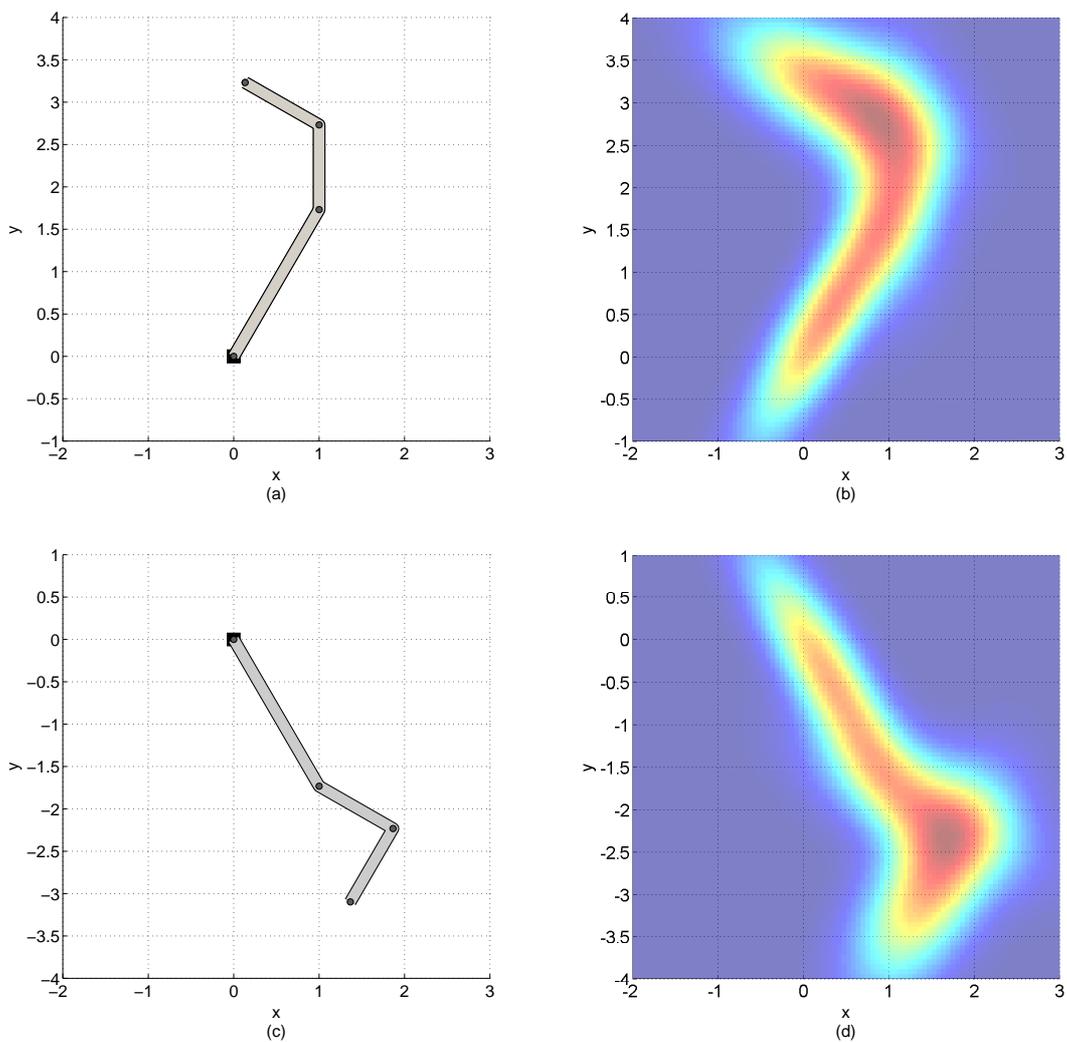


Figura 4.44: Manipulador planar 3R e função f correspondente.

Capítulo 5

Conclusões e trabalhos futuros

5.1 Conclusões

- Este trabalho apresenta um algoritmo alternativo para o cálculo da cinemática inversa de robôs manipuladores, que pode ser interpretado como uma inversa filtrada. A ideia básica do algoritmo proposto consiste em calcular a inversa dinamicamente, com rapidez determinada pelo condicionamento da matriz. Foram apresentados diversos resultados de simulação que ilustram o bom desempenho do algoritmo proposto.
- Neste trabalho, o algoritmo da inversa filtrada é generalizado para matrizes retangulares com a utilização de uma lei de atualização composta, garantindo ao método uma forma geral independente das dimensões da matriz considerada.
- A utilização do algoritmo proposto e o ajuste de ganhos permite a priorização de um dos objetivos de controle quando estes são inalcançáveis (e.g., posição e orientação). Um problema de dimensão ampliada também pode ser definido, considerando uma restrição adicional cuja prioridade está da mesma forma relacionada ao ajuste de ganhos. São propostas duas funções objetivo, associadas a limite de juntas e desvio de obstáculos.
- Comparado com outros algoritmos descritos na literatura, uma vantagem do algoritmo proposto é sua facilidade de sintonia pois possui apenas um parâmetro de ajuste: um ganho de adaptação. Este ganho está diretamente relacionado ao desempenho do método. Do ponto de vista computacional, o método proposto se mostra eficiente por não requerer a inversão da matriz Jacobiana, assim como o cálculo de valores singulares.
- O desenvolvimento de um ambiente de realidade virtual para *Simulink* também constitui uma contribuição deste trabalho, permitindo a visualização em tempo real do movimento da estrutura mecânica.

5.2 Propostas de trabalhos futuros

Com o intuito de aprimorar a pesquisa desenvolvida, no que se segue são apresentados tópicos para trabalhos futuros:

- Análise de uma proposta alternativa $\dot{\mathbf{q}} = \Theta_P \mathbf{J}^T \boldsymbol{\nu}$, onde Θ_P é a inversa filtrada de $\mathbf{J}^T \mathbf{J}$. Esta análise deve incluir as vantagens da utilização de uma lei de atualização composta, uma comparação das matrizes Θ_P e $\Theta \Theta^T$, onde Θ é a inversa filtrada de \mathbf{J} , e o estudo do condicionamento do bom desempenho do método à escolha inicial $\Theta_P(0)$.
- Inclusão de um termo *feedforward* na lei de atualização. O termo $\mathbf{F}_\Theta = -\Theta \dot{\mathbf{J}} \Theta$ foi considerado durante a pesquisa, porém não foi incluído na proposta final para a lei de atualização de Θ , apenas proporcional às matrizes de erro à esquerda e à direita. Para o caso escalar, a utilização do termo correspondente $f_\theta = -\theta^2 \dot{k}$ ($\dot{\theta}$ para $\theta = k^{-1}$) permite que o erro escalar S permaneça nulo após o transitório. Desta forma, a trajetória no plano $\theta \times k$ se desenvolve na hipérbole onde $S = 0$, impedindo que o ganho k passe por 0 e, portanto, troque de sinal. No caso multivariável, a utilização do termo \mathbf{F}_Θ resulta no cálculo exato da matriz inversa (após o transitório), impedindo que a matriz Jacobiana passe por um ponto de singularidade, principal motivação deste trabalho. Por outro lado, sua consideração permite um melhor desempenho para matrizes bem condicionadas e afastadas de singularidades, sugerindo o chaveamento deste termo como tópico futuro de pesquisa.
- Realização de ensaios experimentais. A implementação prática do algoritmo proposto é de grande relevância e deve ser considerada em trabalhos futuros. A aplicação destes estudos de caso deve ser acompanhada de uma análise da aproximação discreta da inversa filtrada, de seu custo computacional e da influência do período de amostragem no desempenho do método proposto. Durante a pesquisa, resultados experimentais foram obtidos para o problema de posição e orientação de um manipulador planar 3R, porém através da proposta escalar para $k = \sin(q_2)$.
- Aplicação em tópicos de controle adaptativo. Um tópico de pesquisa a ser desenvolvido consiste na utilização da abordagem proposta para solucionar o problema de cinemática inversa de manipuladores com incertezas em seu modelo cinemático (e.g., robô manipulando um objeto de tamanho desconhecido com ângulo desconhecido). Em geral, uma solução adaptativa indireta utiliza a inversão da matriz Jacobiana estimada, cuja invertibilidade pode não ser garantida.

Referências bibliográficas

- [1] P. Ránky and C. H. Ho. *Robot Modelling: Control and Applications with Software*. IFS (Publications) Ltd, 1985.
- [2] A. Pascoal, P. Oliveira, C. Silvestre, A. Bjerrum, A. Ishoy, J. Pignon, G. Ayela, and C. Petzelt. Marius: An autonomous underwater vehicle for coastal oceanography. *IEEE Robotics and Automation Magazine*, 4(4):46-59, 1997.
- [3] E. Coleshill, L. Oshinowo, R. Rembala, B. Bina, D. Rey, and S. Sindelar. Dextre: Improving maintenance operations on the international space station. *Acta Astronautica*, 64(9-10):869-874, 2009.
- [4] P. Dario, E. Guglielmelli, B. Allotta, and M. C. Carrozza. Robotics for medical applications. *IEEE Robotics and Automation Magazine*, 3(3):44-56, 1996.
- [5] S. Thrun, S. Thayer, W. Whittaker, C. Baker, W. Burgard, D. Ferguson, D. Hanel, M. Montemerlo, A. Morris, Z. Omohundro, and C. Reverte. Autonomous exploration and mapping of abandoned mines. *IEEE Robotics and Automation Magazine*, 11(4):79-91, 2004.
- [6] K. Tchoń and J. Jakubiak. Extended jacobian inverse kinematics algorithm for nonholonomic mobile robots. *International Journal of Control*, 79(8):895–909, 2006.
- [7] S. Chiaverini, G. Oriolo, and I. D. Walker. Kinematically redundant manipulators. In B. Siciliano and O. Khatib, editors, *Springer Handbook of Robotics*, pages 245–268. Springer-Verlag Ltd., 1st edition, 2008.
- [8] W. A. Wolovich and H. Elliot. A computational technique for inverse kinematics. *The 23rd IEEE Conference on Decision and Control*, 23:1359–1363, 1984.
- [9] T. Komura, A. Kuroda, S. Kudoh, T. C. Lan, and Y. Shinagawa. An inverse kinematics method for 3D figures with motion data. In *Proceedings of the Computer Graphics International*, pages 226–271, Tokyo, Japan, 2003.

- [10] S. Li, M. Okuda, and S. Takahashi. Improved kinematics based motion compression for human figure animation. In *Proceedings of the 30th IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 1077–1080, Philadelphia, PA, USA, 2005.
- [11] L. Unzueta, M. Peinado, R. Boulic, and A. Suescun. Full-body performance animation with sequential inverse kinematics. *Graphical Models*, 20(5):87–104, 2008.
- [12] D. Manocha, Y. Zhu, and W. Wright. Conformational analysis of molecular chains using nano-kinematics. *Computer Application of Biological Sciences*, 11(1):71–86, 1995.
- [13] M. Zhang and L. E. Kavraki. Approximating solutions of molecular inverse kinematics problems by subdivision. In *Proceedings of the Second Joint EMBS/BMES Conference*, pages 2182–2183, Houston, TX, USA, 2002.
- [14] A. A. Canutescu and R. L. Dunbrack Jr. Cyclic coordinate descent: A robotics algorithm for protein loop closure. *Protein Science*, 12:963–972, 2003.
- [15] R. Kolodny, L. Guibas, M. Levitt, and P. Koehl. Inverse kinematics in biology: The protein loop closure problem. *The International Journal of Robotics Research*, 24(2–3):151–163, 2005.
- [16] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo. *Robotics: Modelling, Planning and Control*. Springer Publishing Company, Inc., 2009.
- [17] P. Donelan. Kinematic singularities of robot manipulators. In Ernest Hall, editor, *Advances in Robot Manipulators*, pages 401–415. InTech, 2010.
- [18] K. Tchoń and R. Muszynski. Singular inverse kinematic problem for robotic manipulators: A normal form approach. *IEEE Transactions on Robotics and Automation*, 14(1):93–104, 1998.
- [19] D. N. Nenchev, Y. Tsumaki, and M. Uchiyama. Singularity-consistent parameterization of robot motion and control. *The International Journal of Robotics Research*, 19(2):159–182, 2000.
- [20] D. K. Pai and M. C. Leu. Genericity and singularities of robot manipulators. *IEEE Transactions on Robotics and Automation*, 8(5):545–559, 1992.
- [21] K. Sugimoto, J. Duffy, and K. H. Hunt. Special configurations of spatial mechanisms and robot arms. *Mechanism and Machine Theory*, 17(2):119–132, 1982.

- [22] A. A. Maciejewski and C. A. Klein. The singular value decomposition: Computation and applications to robotics. *The International Journal of Robotics Research*, 8(6):63–79, 1989.
- [23] K. Tchoń and A. Matuszok. On avoiding singularities in redundant robot kinematics. *Robotica*, 13:599–606, 1995.
- [24] S. Chiaverini. Singularity-robust task-priority redundancy resolution for real-time kinematic control of robot manipulators. *IEEE Transactions on Robotics and Automation*, 13(3):398–410, 1997.
- [25] A. A. Maciejewski and J. M. Reagin. A parallel algorithm and architecture for the control of kinematically redundant manipulators. *IEEE Transactions on Robotics and Automation*, 10(4):405–414, 1994.
- [26] C. A. Klein. Use of redundancy in the design of robotic systems. In *Second International Symposium on Robotics Research*, pages 207–214, Cambridge MA, 1985. MIT Press.
- [27] Y. Nakamura and H. Hanafusa. Task priority based redundancy control of robot manipulators. In *Second International Symposium on Robotics Research*, pages 155–162, Cambridge, MA, 1985. MIT Press.
- [28] R. P. Paul, B. E. Shimano, and G. Mayer. Kinematic control equations for simple manipulators. *IEEE Transactions on Systems, Man, and Cybernetics*, 11(6):449–455, 1981.
- [29] L-C. T. Wang and C. C. Chen. A combined optimization method for solving the inverse kinematics problem of mechanical manipulators. *IEEE Transactions on Robotics and Automation*, 7(4):489–499, 1991.
- [30] D. Tolani, A. Goswami, and N. I. Badler. Real-time inverse kinematics techniques for anthropomorphic limbs. Academic Press, 2000.
- [31] B. Paden. *Kinematics and Control Robot Manipulators*. PhD thesis, Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, 1986.
- [32] W. Kahan. *Lectures on computational aspects of geometry*. Department of Electrical Engineering and Computer Sciences, University of California, Berkeley. Unpublished, 1983.
- [33] R. Murray, Z. Li, and S. S. Sastry. *A Mathematical Introduction to Robotic Manipulation*. CRC Press, 1994.

- [34] R. P. Paul. *Robot Manipulators: Mathematics, Programming, and Control*. MIT Press, Cambridge, MA, 1981.
- [35] R. Featherstone. Position and velocity transformations between robot end-effector coordinates and joint angles. *International Journal of Robotics Research*, 2(2):35–45, 1983.
- [36] K. J. Waldron, S. L. Wang, and S. J. Bolin. A study of the jacobian matrix of serial manipulators. *Transmission and Automation in Design*, 107(2):230–238, 1985.
- [37] D. E. Whitney. Resolved motion rate control of manipulators and human prostheses. *IEEE Transactions on Man-Machine Systems*, 10(2):47–53, 1969.
- [38] C. A. Klein and C. S. Huang. Review of pseudoinverse control for use with kinematically redundant manipulators. *IEEE Transactions on Systems, Man and Cybernetics*, 13:245–250, 1983.
- [39] D. G. Luenberger. *Linear and nonlinear programming*. Reading, MA: Addison-Wesley, 1984.
- [40] Y. Nakamura and H. Hanafusa. Inverse kinematic solution with singularity robustness for robot manipulator control. *ASME Journal of Dynamic Systems, Measurement and Control*, 108(3):163–171, 1986.
- [41] C. W. Wampler. Manipulator inverse kinematic solutions based on vector formulations and damped least-squares methods. *IEEE Transactions on System, Man and Cybernetics*, 16(1):93–101, 1986.
- [42] S. Chiaverini. Estimate of the two smallest singular values of the jacobian matrix: Application to damped least-squares inverse kinematics. *Journal of Robotic Systems*, 10(8):991–1008, 1993.
- [43] S. Chiaverini, B. Siciliano, and O. Egeland. Review of the damped least-squares inverse kinematics with experiments on an industrial robot manipulator. *IEEE Transactions on Control Systems Technology*, 2(2):123–134, 1994.
- [44] R. V. Mayorga, N. Milano, and A. K. C. Wong. A fast procedure for manipulator inverse kinematics computation and singularities prevention. *Journal of Robotic Systems*, 10(1):45–72, 1993.
- [45] D. E. Schinstock, T. N. Faddis, and R. B. Greenway. Robust inverse kinematics using damped least squares with dynamic weighting. American Institute of Aeronautics and Astronautics, Inc., 1994.

- [46] A. S. Deo and I. D. Walker. Adaptive non-linear least squares for inverse kinematics. In *IEEE International Conference on Robotics and Automation*, pages 186–193, 1993.
- [47] A. N. Pechev. Inverse kinematics without matrix inversion. In *IEEE International Conference on Robotics and Automation*, pages 2005–2012, Pasadena, CA, USA, 2008.
- [48] S. M. Ahmed and A. N. Pechev. Performance analysis of FIK and DLS inverse kinematics using six degree of freedom manipulator. In *IEEE International Conference on Robotics and Biomimetics*, pages 1405–1410, Guilin, China, 2009.
- [49] J. Denavit and R. S. Hartenberg. A kinematic notation for lower-pair mechanisms based on matrices. *ASME Journal of Applied Mechanics*, 22:215–221, 1955.
- [50] J. J. E. Slotine and W. Li. *Applied Nonlinear Control*. Prentice Hall, 1991.
- [51] A. C. Leite. Controle híbrido de força e visão de um manipulador robótico sobre superfícies desconhecidas. Master’s thesis, Universidade Federal do Rio de Janeiro, UFRJ, 2005.
- [52] J. S. Yuan. Close-loop manipulator control using quaternion feedback. *IEEE Journal of Robotics and Automation*, 4(4):434–440, 1988.
- [53] K. Tchoń. Differential topology of the inverse kinematic problem for redundant robot manipulators. *International Journal of Robotics Research*, 10(5):492–504, 1991.
- [54] G. Strang. *Linear Algebra and Its Applications*. Cengage Learning, 4th edition, 2009.
- [55] D. L. Pieper. *The Kinematics of Manipulators Under Computer Control*. PhD thesis, Stanford University, 1968.
- [56] R. P. Paul and C. N. Stevenson. Kinematics of robot wrists. *The International Journal of Robotics Research*, 2(1):31–38, 1983.
- [57] M. M. Stanišić and J. W. Engelberth. A geometric description of manipulator singularities in terms of singular surfaces. In *Advances in Robot Kinematics*, pages 132–141, Ljubljana, Slovenia, 1988.

- [58] M. M. Stanišić and C. M. Goehler. Singular planes of serial wrist-partitioned manipulators and their singularity metrics. *Mechanism and Machine Theory*, 42(8):889–902, 2007.
- [59] A. V. Tikhonov and V. Y. Arsenin. *Solution of Ill-posed Problems*. Wiley, New York, NY, 1977.
- [60] H. H. Rosenbrock. The stability of multivariable systems. *IEEE Transactions on Automatic Control*, 17(1):105–107, 1972.
- [61] J. Doyle and G. Stein. Multivariable feedback design: concepts for a classical/modern synthesis. *IEEE Transactions on Automatic Control*, 26(1):4–16, 1981.
- [62] H. Flanders. Elementary divisors of AB and BA . *Proceedings of the American Mathematical Society*, 2(6):871–874, 1951.
- [63] S. J. Bernau and A. Abian. Jordan canonical forms of matrices AB and BA . *Rendiconti dell'Istituto di Matematica dell'Università di Trieste*, 20(1):101–108, 1988.
- [64] J. R. Magnus and H. Neudecker. *Matrix Differential Calculus with Applications in Statistics and Econometrics*. John Wiley & Sons, West Sussex, England, 2007.
- [65] A. J. Laub. *Matrix Analysis for Scientists and Engineers*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 2005.
- [66] R. Bhatia and P. Rosenthal. How and why to solve the operator equation $AX - XB = Y$. *Bulletin of the London Mathematical Society*, 29(1):1–21, 1997.
- [67] W. E. Roth. The equations $AX - YB = C$ and $AX - XB = C$ in matrices. *Proceedings of the American Mathematical Society*, 3(3):392–396, 1952.

Apêndice A

Conceitos básicos

A.1 Propriedades da função traço

Considere uma matriz quadrada $\mathbf{A} \in \mathbb{R}^{m \times m}$. A função traço de \mathbf{A} , denotada por $\text{tr}(\mathbf{A})$, corresponde à soma dos elementos de sua diagonal principal [64], podendo ser escrita como

$$\text{tr}(\mathbf{A}) = a_{11} + a_{22} + \dots + a_{mm} = \sum_{i=1}^m a_{ii}, \quad (\text{A.1})$$

onde a_{ij} denota o elemento da i -ésima linha e j -ésima coluna de \mathbf{A} . As propriedades da função $\text{tr}(\cdot)$ são apresentadas a seguir:

1. $\text{tr}(\alpha \mathbf{A}) = \alpha \text{tr}(\mathbf{A})$, onde $\alpha \in \mathbb{R}$;
2. $\text{tr}(\mathbf{A} + \mathbf{B}) = \text{tr}(\mathbf{A}) + \text{tr}(\mathbf{B})$;
3. $\text{tr}(\mathbf{A}) = \text{tr}(\mathbf{A}^T)$;
4. $\frac{d}{dt} \text{tr}(\mathbf{A}) = \text{tr}(\frac{d\mathbf{A}}{dt})$;
5. $\text{tr}(\mathbf{BC}) = \text{tr}(\mathbf{CB})$, onde $\mathbf{B} \in \mathbb{R}^{m \times n}$ e $\mathbf{C} \in \mathbb{R}^{n \times m}$;
6. $\text{tr}(\mathbf{A}) = \sum \lambda_i$, onde λ_i é autovalor de \mathbf{A} .

A partir das propriedades 1 e 2, tem-se

$$2'. \text{tr}(\alpha \mathbf{A} + \beta \mathbf{B}) = \alpha \text{tr}(\mathbf{A}) + \beta \text{tr}(\mathbf{B}), \text{ onde } \alpha \in \mathbb{R} \text{ e } \beta \in \mathbb{R}. \quad (\text{linearidade})$$

A.2 Subespaços fundamentais

Considere uma matriz $\mathbf{A} \in \mathbb{R}^{m \times n}$ com posto r . Os subespaços fundamentais de \mathbf{A} são apresentados a seguir [54]:

1. O **espaço-coluna** de \mathbf{A} é denotado por $Col(\mathbf{A})$ e é gerado pelas colunas de \mathbf{A} . Sua dimensão é r ;
2. O **espaço nulo** de \mathbf{A} é denotado por $Nul(\mathbf{A})$. Ele contém todos os vetores \mathbf{x} tal que $\mathbf{Ax} = \mathbf{0}$. Sua dimensão é $n - r$;
3. O **espaço-linha** de \mathbf{A} é o espaço-coluna de \mathbf{A}^T . Ele é denotado por $Col(\mathbf{A}^T)$ e é gerado pelas linhas de \mathbf{A} . Sua dimensão também é r ;
4. O **espaço nulo à esquerda** de \mathbf{A} é o espaço nulo de \mathbf{A}^T . Ele é denotado por $Nul(\mathbf{A}^T)$ e contém todos os vetores \mathbf{y} tal que $\mathbf{y}^T\mathbf{A} = \mathbf{0}$. Sua dimensão é $m - r$.

Teorema fundamental de ortogonalidade: O espaço nulo $Nul(\mathbf{A})$ e o espaço-linha $Col(\mathbf{A}^T)$ são ortogonais e subespaços de \mathbb{R}^n . O espaço nulo à esquerda $Nul(\mathbf{A}^T)$ e o espaço-coluna $Col(\mathbf{A})$ são ortogonais e subespaços de \mathbb{R}^m .

A.3 Decomposição de valor singular (SVD)

A decomposição de valor singular, ou simplesmente SVD, constitui uma ferramenta extremamente poderosa para análise em álgebra linear. Qualquer matriz $\mathbf{A} \in \mathbb{R}^{m \times n}$ pode ser fatorada em [54]

$$\mathbf{A} = \mathbf{U} \mathbf{\Sigma} \mathbf{V}^T = (\text{ortogonal})(\text{diagonal})(\text{ortogonal}). \quad (\text{A.2})$$

Os r valores singulares (denotados por σ) na diagonal de $\mathbf{\Sigma} \in \mathbb{R}^{m \times n}$ são as raízes quadradas de autovalores não nulos de \mathbf{AA}^T e $\mathbf{A}^T\mathbf{A}$. As colunas de $\mathbf{U} \in \mathbb{R}^{m \times m}$ são autovetores de \mathbf{AA}^T e as colunas de $\mathbf{V} \in \mathbb{R}^{n \times n}$ são autovetores de $\mathbf{A}^T\mathbf{A}$. As matrizes \mathbf{U} e \mathbf{V} fornecem bases ortonormais para todos os quatros subespaços fundamentais apresentados na Seção A.2:

Primeiras	r	colunas de \mathbf{U} :	espaço-coluna de \mathbf{A}
Últimas	$m - r$	colunas de \mathbf{U} :	espaço nulo à esquerda de \mathbf{A}
Primeiras	r	colunas de \mathbf{V} :	espaço-linha de \mathbf{A}
Últimas	$n - r$	colunas de \mathbf{V} :	espaço-nulo de \mathbf{A}

A partir de (A.2), tem-se

$$\mathbf{AV} = \mathbf{U} \mathbf{\Sigma}, \quad (\text{A.3})$$

e, portanto, quando a matriz \mathbf{A} multiplica uma coluna \mathbf{v}_j de \mathbf{V} , produz σ_j vezes uma coluna de \mathbf{U} .

A.4 Produto de Kronecker

Considere as matrizes $\mathbf{A} \in \mathbb{R}^{m \times n}$ e $\mathbf{B} \in \mathbb{R}^{p \times q}$. O produto de Kronecker (ou produto tensor) de \mathbf{A} e \mathbf{B} é definido como a matriz:

$$\mathbf{A} \otimes \mathbf{B} = \begin{bmatrix} a_{11}\mathbf{B} & \cdots & a_{1n}\mathbf{B} \\ \vdots & \ddots & \vdots \\ a_{m1}\mathbf{B} & \cdots & a_{mn}\mathbf{B} \end{bmatrix} \in \mathbb{R}^{mp \times nq}. \quad (\text{A.4})$$

As propriedades básicas do produto de Kronecker são apresentadas a seguir [65]:

1. $(\alpha\mathbf{A}) \otimes \mathbf{B} = \mathbf{A} \otimes (\alpha\mathbf{B}) = \alpha(\mathbf{A} \otimes \mathbf{B})$, onde $\alpha \in \mathbb{R}$;
2. $\mathbf{A} \otimes (\mathbf{B} + \mathbf{C}) = (\mathbf{A} \otimes \mathbf{B}) + (\mathbf{A} \otimes \mathbf{C})$;
3. $(\mathbf{A} + \mathbf{B}) \otimes \mathbf{C} = (\mathbf{A} \otimes \mathbf{C}) + (\mathbf{B} \otimes \mathbf{C})$;
4. $(\mathbf{A} \otimes \mathbf{B}) \otimes \mathbf{C} = \mathbf{A} \otimes (\mathbf{B} \otimes \mathbf{C})$;
5. $(\mathbf{A} \otimes \mathbf{B})^* = \mathbf{A}^* \otimes \mathbf{B}^*$, $(\mathbf{A} \otimes \mathbf{B})^T = \mathbf{A}^T \otimes \mathbf{B}^T$ e $(\mathbf{A} \otimes \mathbf{B})^\dagger = \mathbf{A}^\dagger \otimes \mathbf{B}^\dagger$;
6. $(\mathbf{A} \otimes \mathbf{B})(\mathbf{C} \otimes \mathbf{D}) = \mathbf{AC} \otimes \mathbf{BD}$;
7. $(\mathbf{A} \otimes \mathbf{B}) = (\mathbf{U}_A \Sigma_A \mathbf{V}_A^T \otimes \mathbf{U}_B \Sigma_B \mathbf{V}_B^T) = (\mathbf{U}_A \otimes \mathbf{U}_B)(\Sigma_A \otimes \Sigma_B)(\mathbf{V}_A \otimes \mathbf{V}_B)^T$;
8. $\text{posto}(\mathbf{A} \otimes \mathbf{B}) = \text{posto}(\mathbf{A})\text{posto}(\mathbf{B}) = \text{posto}(\mathbf{B} \otimes \mathbf{A})$.

Para matrizes quadradas $\mathbf{A} \in \mathbb{R}^{m \times m}$ e $\mathbf{B} \in \mathbb{R}^{n \times n}$, tem-se que:

9. $\text{tr}(\mathbf{A} \otimes \mathbf{B}) = \text{tr}(\mathbf{A})\text{tr}(\mathbf{B})$ e $\det(\mathbf{A} \otimes \mathbf{B}) = (\det(\mathbf{A}))^n (\det(\mathbf{B}))^m$.

A.5 Operador vec

O operador $\text{vec}(\cdot)$ cria um vetor coluna a partir de uma matriz $\mathbf{A} \in \mathbb{R}^{m \times n}$ ao empilhar as colunas de $\mathbf{A} = [\mathbf{a}_1 \ \mathbf{a}_2 \ \cdots \ \mathbf{a}_n]$ uma embaixo da outra [65]:

$$\text{vec}(\mathbf{A}) = \begin{bmatrix} \mathbf{a}_1 \\ \mathbf{a}_2 \\ \vdots \\ \mathbf{a}_n \end{bmatrix}. \quad (\text{A.5})$$

Para $\mathbf{A} \in \mathbb{R}^{m \times n}$, $\mathbf{X} \in \mathbb{R}^{n \times p}$ e $\mathbf{B} \in \mathbb{R}^{p \times q}$, tem-se que:

$$\text{vec}(\mathbf{AXB}) = (\mathbf{B}^T \otimes \mathbf{A})\text{vec}(\mathbf{X}). \quad (\text{A.6})$$

A.6 Equação de Sylvester

Considere as matrizes $\mathbf{A} \in \mathbb{R}^{n \times n}$, $\mathbf{B} \in \mathbb{R}^{m \times m}$ e $\mathbf{C} \in \mathbb{R}^{n \times m}$. Uma equação matricial linear dada por

$$\mathbf{A}\mathbf{X} + \mathbf{X}\mathbf{B} = \mathbf{C} \quad (\text{A.7})$$

é denominada uma equação de Sylvester em homenagem a J. J. Sylvester, que estudou equações matriciais lineares descritas por

$$\sum_{i=1}^k \mathbf{A}_i \mathbf{X}_i \mathbf{B}_i = \mathbf{C}. \quad (\text{A.8})$$

A equação (A.7) pode ser reescrita utilizando o operador $\text{vec}(\cdot)$ e o produto de Kronecker da seguinte forma [65]:

$$[(\mathbf{I}_m \otimes \mathbf{A}) + (\mathbf{B}^T \otimes \mathbf{I}_n)] \text{vec}(\mathbf{X}) = \text{vec}(\mathbf{C}). \quad (\text{A.9})$$

Uma única solução $\mathbf{X} \in \mathbb{R}^{n \times m}$ existirá para (A.9) se e somente se $[(\mathbf{I}_m \otimes \mathbf{A}) + (\mathbf{B}^T \otimes \mathbf{I}_n)]$ for não-singular e, portanto, não apresentar autovalores nulos. Sejam $\lambda_1, \dots, \lambda_n$ e μ_1, \dots, μ_m os autovalores de \mathbf{A} e \mathbf{B} , respectivamente. A matriz $[(\mathbf{I}_m \otimes \mathbf{A}) + (\mathbf{B} \otimes \mathbf{I}_n)]$, também denominada soma de Kronecker ($\mathbf{A} \oplus \mathbf{B}$), apresenta mn autovalores

$$\lambda_1 + \mu_1, \dots, \lambda_1 + \mu_m, \lambda_2 + \mu_1, \dots, \lambda_2 + \mu_m, \dots, \lambda_n + \mu_1, \dots, \lambda_n + \mu_m. \quad (\text{A.10})$$

Assim, uma condição necessária e suficiente para que a equação $\mathbf{A}\mathbf{X} + \mathbf{X}\mathbf{B} = \mathbf{C}$ apresente uma solução única é dada por

$$\sigma(\mathbf{A}) \cap \sigma(-\mathbf{B}) = \emptyset, \quad (\text{A.11})$$

onde $\sigma(\mathbf{A})$ e $\sigma(-\mathbf{B})$ correspondem aos espectros de \mathbf{A} e $-\mathbf{B}$, respectivamente. Isto é, existe uma única solução \mathbf{X} para (A.7) se e somente se \mathbf{A} e $-\mathbf{B}$ não apresentarem autovalores em comum [66]. Uma condição para a existência da solução é que

$$\begin{bmatrix} \mathbf{A} & \mathbf{C} \\ \mathbf{0} & -\mathbf{B} \end{bmatrix} \text{ e } \begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{0} & -\mathbf{B} \end{bmatrix} \quad (\text{A.12})$$

sejam similares, como demonstrado em [67]. Supondo \mathbf{A} e \mathbf{B} assintoticamente estáveis (autovalores com parcelas reais estritamente negativas), a solução para a equação de Sylvester pode ser escrita como [65]

$$\mathbf{X} = - \int_0^{+\infty} e^{t\mathbf{A}} \mathbf{C} e^{t\mathbf{B}} dt. \quad (\text{A.13})$$

Apêndice B

Zebra-ZERO

O manipulador Zebra-ZERO (Figura B.1) é um manipulador antropomórfico com punho esférico, composto portanto por seis juntas de revolução. A atuação é realizada por seis motores DC *Pittman* e a leitura dos ângulos das juntas é realizada por encoders incrementais HEDS-5500 (*Hewlett Packard*). A acessibilidade a um exemplar deste manipulador e, portanto, uma possível aplicação prática justificam a utilização de suas dimensões para simulações, assim como o desenvolvimento de um modelo tridimensional próprio. No que se segue, são apresentadas as equações de cinemática direta para este manipulador e o ambiente de realidade virtual desenvolvido em *Simulink*.

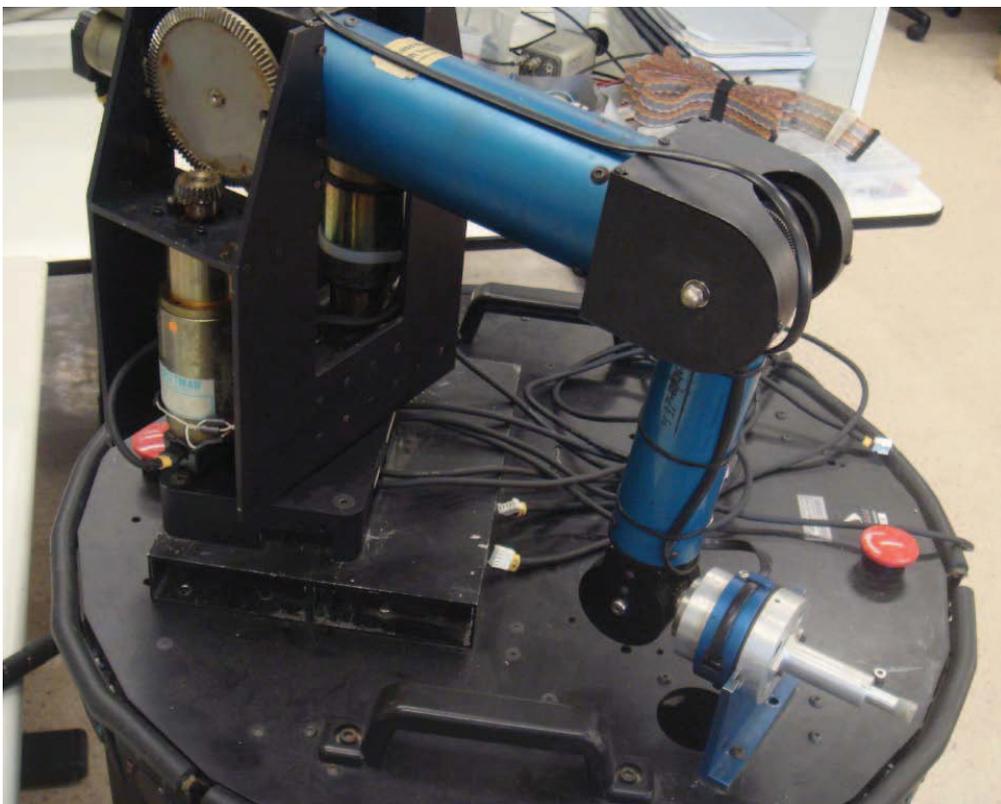


Figura B.1: Manipulador Zebra-ZERO.

B.1 Cinemática direta

Os parâmetros Denavit-Hartenberg (DH) do manipulador em questão são apresentados na Tabela B.1.

i	a_i (mm)	α_i (rad)	d_i (mm)	q_i (rad)
1	0	$\frac{\pi}{2}$	0	q_1
2	279.4	0	0	q_2
3	0	$-\frac{\pi}{2}$	0	q_3
4	0	$\frac{\pi}{2}$	228.6	q_4
5	0	$-\frac{\pi}{2}$	0	q_5
6	0	0	165	q_6

Tabela B.1: Parâmetros Denavit-Hartenberg standard

A matriz de rotação \mathbf{R}_{0e} é obtida de forma trivial e recursiva a partir de

$$\mathbf{R}_{0e} = \mathbf{R}_{01} \mathbf{R}_{12} \dots \mathbf{R}_{56}, \quad (\text{B.1})$$

onde $\mathbf{R}_{i-1,i}$ é dada por

$$\mathbf{R}_{i-1,i} = \mathbf{R}_z(q_i) \mathbf{R}_x(\alpha_i). \quad (\text{B.2})$$

Desta forma,

$$(\mathbf{n}_e)_0 = \begin{bmatrix} c_1(c_{23}(c_4c_5c_6 - s_4s_6) - s_{23}s_5c_6) - s_1(s_4c_5c_6 + c_4s_6) \\ s_1(c_{23}(c_4c_5c_6 - s_4s_6) - s_{23}s_5c_6) + c_1(s_4c_5c_6 + c_4s_6) \\ s_{23}(c_4c_5c_6 - s_4s_6) + c_{23}s_5c_6 \end{bmatrix}, \quad (\text{B.3})$$

$$(\mathbf{s}_e)_0 = \begin{bmatrix} c_1(-c_{23}(c_4c_5s_6 + s_4c_6) + s_{23}s_5s_6) + s_1(s_4c_5s_6 - c_4c_6) \\ s_1(-c_{23}(c_4c_5s_6 + s_4c_6) + s_{23}s_5s_6) - c_1(s_4c_5s_6 - c_4c_6) \\ -s_{23}(c_4c_5s_6 + s_4c_6) - c_{23}s_5s_6 \end{bmatrix}, \quad (\text{B.4})$$

$$(\mathbf{a}_e)_0 = \begin{bmatrix} -c_1(c_{23}c_4s_5 + s_{23}c_5) + s_1s_4s_5 \\ -s_1(c_{23}c_4s_5 + s_{23}c_5) - c_1s_4s_5 \\ -s_{23}c_4s_5 + c_{23}c_5 \end{bmatrix}. \quad (\text{B.5})$$

A posição do punho pode ser obtida pela análise geométrica da estrutura mecânica. Assim,

$$(\mathbf{p}_w)_0 = \begin{bmatrix} l_1c_1c_2 - l_2c_1s_{23} \\ l_1s_1c_2 - l_2s_1s_{23} \\ l_1s_2 + l_2c_{23} \end{bmatrix}. \quad (\text{B.6})$$

A posição do efetuador, obtida a partir de (B.5) e (B.6), é dada por

$$(\mathbf{p}_e)_0 = (\mathbf{p}_w)_0 + l_e(\mathbf{a}_e)_0. \quad (\text{B.7})$$

B.2 Modelo 3D

O modelo 3D desenvolvido tem como principal objetivo a visualização em tempo real do movimento da estrutura mecânica. Desta forma, alguns aspectos geralmente despercebidos em resultados gráficos podem ser avaliados. Por exemplo, pode-se notar se o braço está quase totalmente esticado, apresentando pouca manipulabilidade. O desenvolvimento do modelo tridimensional é realizado através de duas funções básicas: ‘patch’ e ‘makehgtform’. A função ‘patch’ recebe como argumentos as coordenadas dos vértices e vetores contendo os índices dos vértices pertencentes à cada face. Já a função ‘makehgtform’ é utilizada para manipulação das figuras geométricas criadas pela função ‘patch’, recebendo argumentos referentes à rotação e ao deslocamento. A construção do modelo preciso do manipulador se inicia com a modelagem individual das peças que o constituem, havendo entre elas relações do tipo *pai/filho*. Ao fim do processo, o modelo de realidade virtual receberá como argumentos os ângulos das juntas, mostrando em tempo real a configuração da estrutura mecânica. Outras formas de utilização são possíveis, como *joystick*, e a utilização do renderizador *OpenGL* torna o desempenho extremamente satisfatório, com atrasos imperceptíveis. A Figura B.2 apresenta o modelo final e a trajetória efetuada (Trajetória 9).

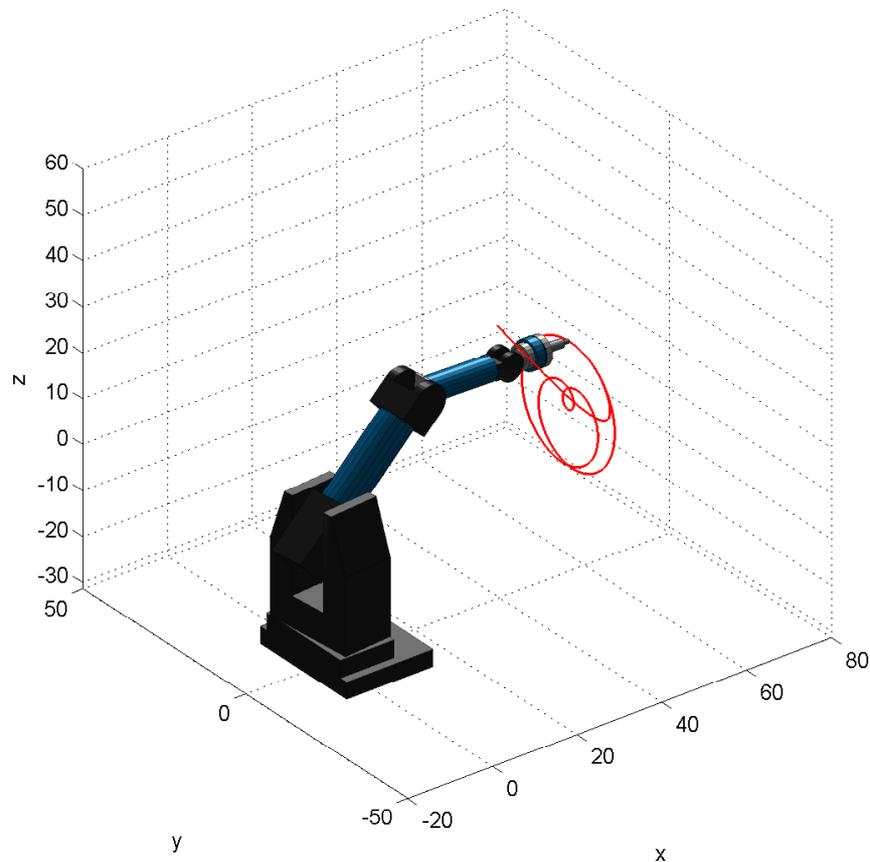


Figura B.2: Modelo 3D para Simulink.

Apêndice C

Trajétórias

No que se segue são apresentadas as trajetórias utilizadas ao longo do trabalho documentado. As trajetórias 1 a 5, desenvolvidas no plano xy , são abordadas nas simulações envolvendo manipuladores planares, redundantes e não-redundantes. As trajetórias 3 e 4 são retiradas de [48] e a abordagem destas visa a reprodução de resultados existentes na literatura. Por fim, as trajetórias 6 a 11 são utilizadas nas simulações envolvendo o manipulador antropomórfico Zebra-ZERO, apresentado anteriormente no Apêndice B.

C.1 Trajetória 1

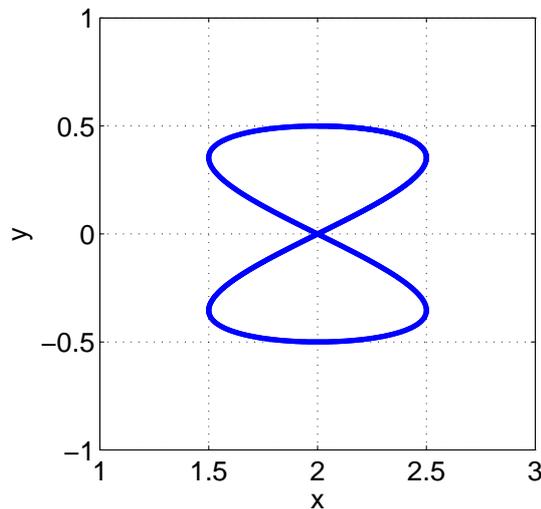


Figura C.1: Trajetória 1.

$$\mathbf{p}_d(t) = \begin{bmatrix} x_d(t) \\ y_d(t) \end{bmatrix} = \begin{bmatrix} 2 + 0.5 \sin(0.4 t) \\ 0.5 \cos(0.2 t) \end{bmatrix}. \quad (\text{C.1})$$

C.2 Trajetória 2

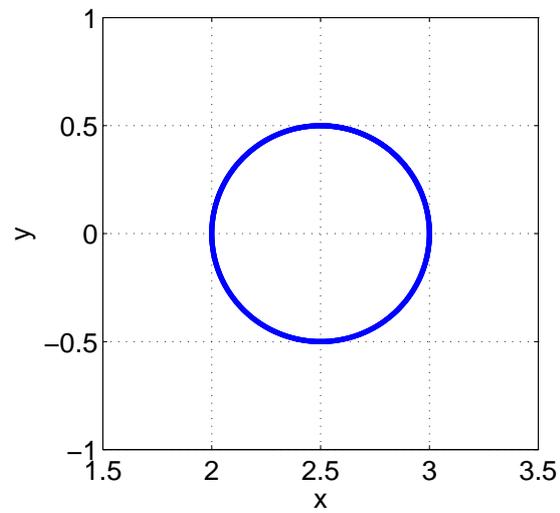


Figura C.2: Trajetória 2.

$$\mathbf{p}_d(t) = \begin{bmatrix} x_d(t) \\ y_d(t) \end{bmatrix} = \begin{bmatrix} 2.5 + 0.5 \sin(0.2t) \\ 0.5 \cos(0.2t) \end{bmatrix}. \quad (\text{C.2})$$

C.3 Trajetória 3

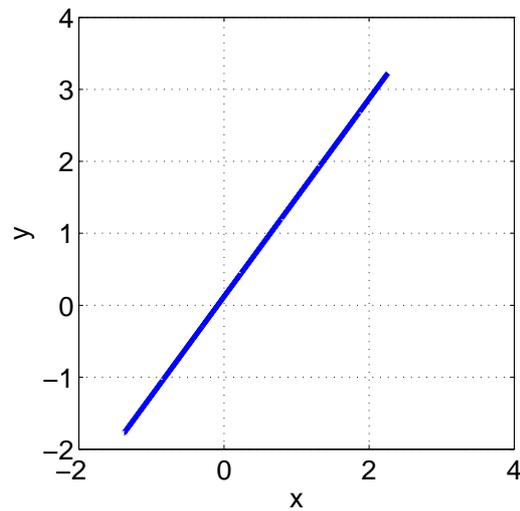


Figura C.3: Trajetória 3.

$$\mathbf{p}_d(t) = \begin{bmatrix} x_d(t) \\ y_d(t) \end{bmatrix} = \begin{bmatrix} 2.26 - t/11 \\ 3.23 - t/8 \end{bmatrix}, \text{ para } 0 < t < 40. \quad (\text{C.3})$$

C.4 Trajetória 4

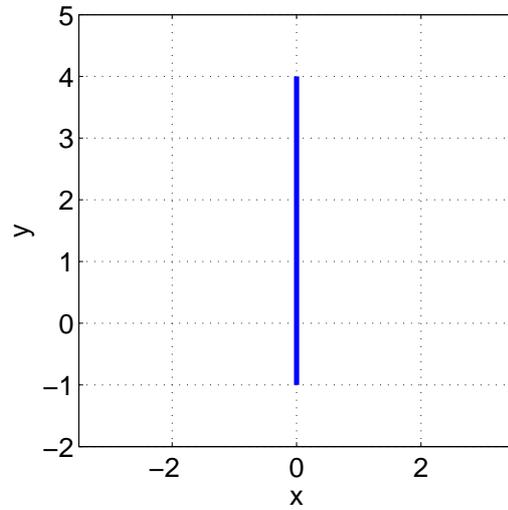


Figura C.4: Trajetória 4.

$$\mathbf{p}_d(t) = \begin{bmatrix} x_d(t) \\ y_d(t) \end{bmatrix} = \begin{bmatrix} 0 \\ 4 - t/8 \end{bmatrix}, \text{ para } 0 < t < 40. \quad (\text{C.4})$$

C.5 Trajetória 5

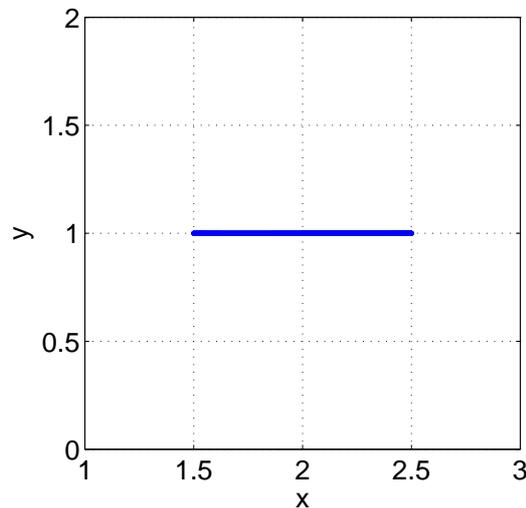


Figura C.5: Trajetória 5.

$$\mathbf{p}_d(t) = \begin{bmatrix} x_d(t) \\ y_d(t) \end{bmatrix} = \begin{bmatrix} 2 + 0.5 \sin(0.4t) \\ 1 \end{bmatrix}. \quad (\text{C.5})$$

C.6 Trajetória 6

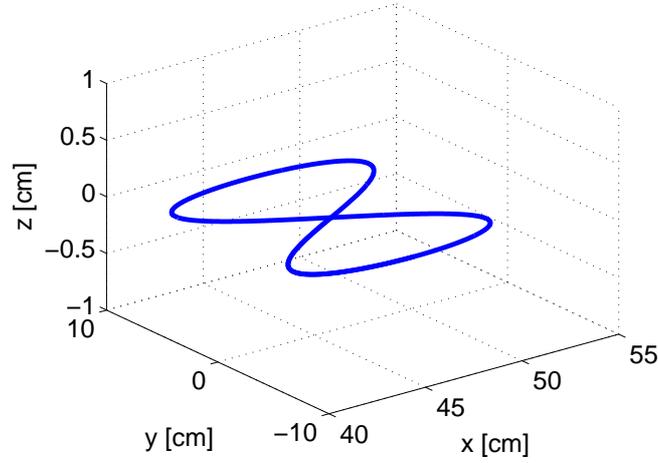


Figura C.6: Trajetória 6.

$$\mathbf{p}_d(t) = \begin{bmatrix} x_d(t) \\ y_d(t) \\ z_d(t) \end{bmatrix} = \begin{bmatrix} 5 \sin(0.2\pi t) + 45.86 \\ 7.5 \sin(0.1\pi t) \\ 0 \end{bmatrix} \text{ cm.} \quad (\text{C.6})$$

C.7 Trajetória 7

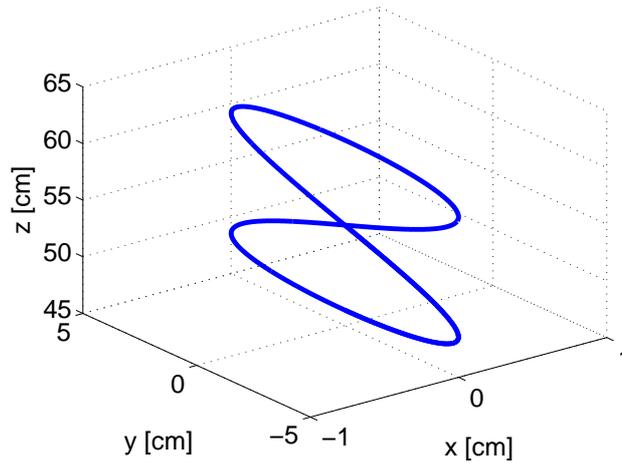


Figura C.7: Trajetória 7.

$$\mathbf{p}_d(t) = \begin{bmatrix} x_d(t) \\ y_d(t) \\ z_d(t) \end{bmatrix} = \begin{bmatrix} 0 \\ 5 \sin(0.2\pi t) \\ 7.5 \sin(0.1\pi t) + 53.86 \end{bmatrix} \text{ cm.} \quad (\text{C.7})$$

C.8 Trajetória 8

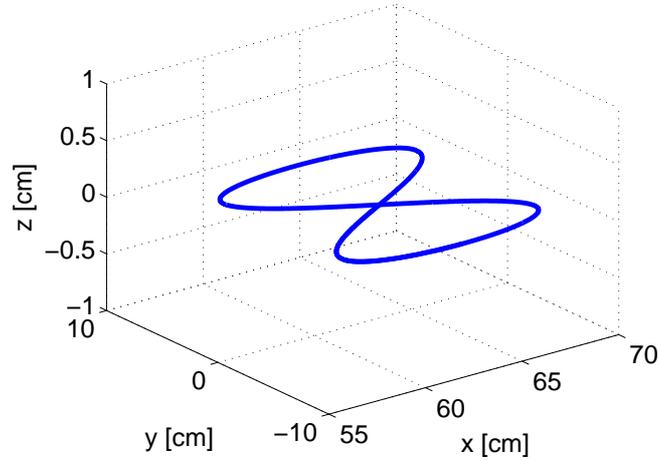


Figura C.8: Trajetória 8.

$$\mathbf{p}_d(t) = \begin{bmatrix} x_d(t) \\ y_d(t) \\ z_d(t) \end{bmatrix} = \begin{bmatrix} 5 \sin(0.2\pi t) + 63.36 \\ 7.5 \sin(0.1\pi t) \\ 0 \end{bmatrix} \text{ cm.} \quad (\text{C.8})$$

C.9 Trajetória 9

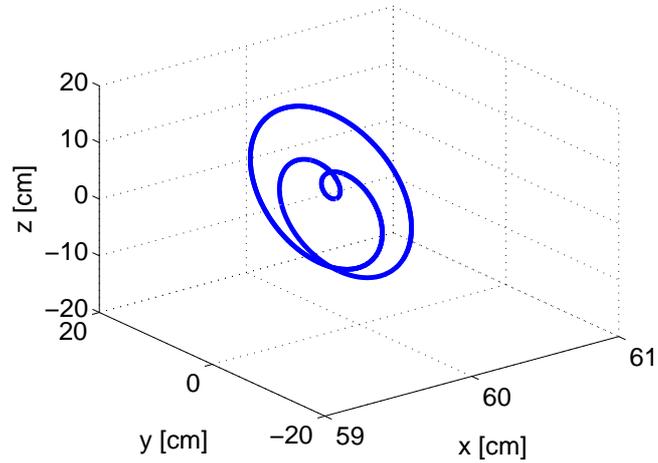


Figura C.9: Trajetória 9.

$$\mathbf{p}_d(t) = \begin{bmatrix} x_d(t) \\ y_d(t) \\ z_d(t) \end{bmatrix} = \begin{bmatrix} 60 \\ 7.5 \sin(0.1\pi t) + 7.5 \sin(0.15\pi t) + 5 \\ 7.5 \sin(0.1\pi t + 1.6) + 7.5 \sin(0.15\pi t + 1.6) \end{bmatrix} \text{ cm.} \quad (\text{C.9})$$

C.10 Trajetória 10

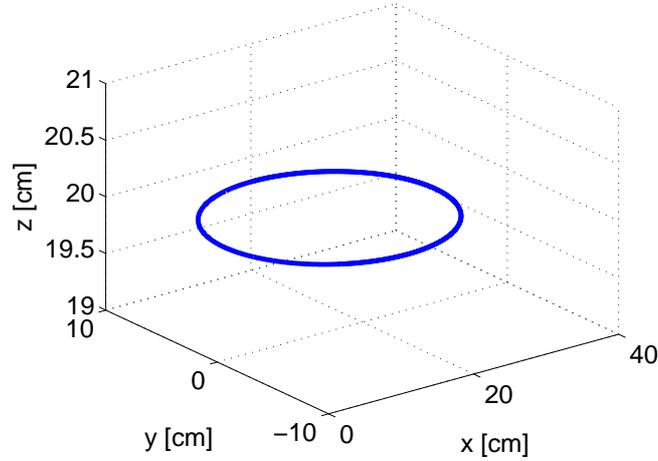


Figura C.10: Trajetória 10.

$$\mathbf{p}_d(t) = \begin{bmatrix} x_d(t) \\ y_d(t) \\ z_d(t) \end{bmatrix} = \begin{bmatrix} 14.6 \sin(0.2t) + 15.5 \\ 7 \cos(0.2t) \\ 20 \end{bmatrix} \text{ cm.} \quad (\text{C.10})$$

C.11 Trajetória 11

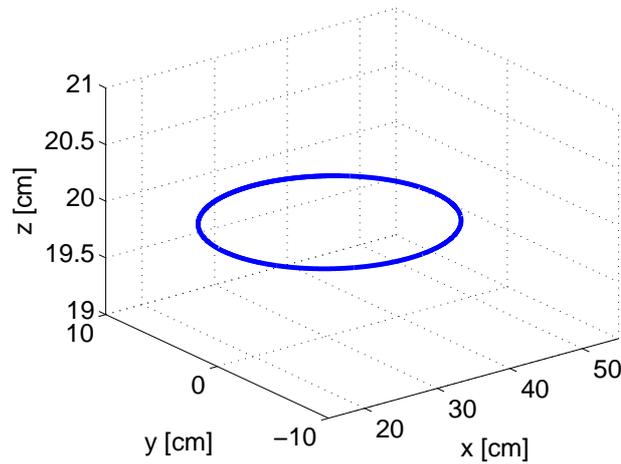


Figura C.11: Trajetória 11.

$$\mathbf{p}_d(t) = \begin{bmatrix} x_d(t) \\ y_d(t) \\ z_d(t) \end{bmatrix} = \begin{bmatrix} 14.6 \sin(0.2t) + 30.5 \\ 7 \cos(0.2t) \\ 20 \end{bmatrix} \text{ cm.} \quad (\text{C.11})$$