

# Projeto de Graduação



12 de Dezembro de 2017

## **METODOLOGIA DE CONTROLE PARA MECANISMOS PARALELOS NA PRESENÇA DE SINGULARIDADES CINEMÁTICAS**

André de Freitas Serapião



[www.ele.puc-rio.br](http://www.ele.puc-rio.br)

Projeto de Graduação



## **METODOLOGIA DE CONTROLE PARA MECANISMOS PARALELOS NA PRESENÇA DE SINGULARIDADES CINEMÁTICAS**

**Aluno: André de Freitas Serapião**

**Orientador: Prof. Antonio Candea Leite**

Trabalho apresentado com requisito parcial à conclusão do curso de Engenharia Elétrica na Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, Brasil.

## **Agradecimentos**

Agradeço primeiramente ao suporte do meu amigo pessoal e orientador, prof Antonio Candea Leite.  
Ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) por ter financiado parte deste trabalho como iniciação científica e tecnológica.  
Finalmente, agradeço a todos que estiveram torcendo e desejando o sucesso deste trabalho.

## Resumo

Neste projeto, considera-se a formulação teórica desenvolvido pelo método da inversa filtrada para obter uma solução eficiente e robusta para o problema de controle cinemático de mecanismos paralelos na presença de configurações singulares. Este método é baseado na equação de cinemática diferencial e consiste em um algoritmo que estime dinamicamente a inversa da matriz jacobiana ao invés de computá-la instantaneamente.

**palavras-chave:** Singularidades, Controle, Cinemática, Matriz Jacobiana, Robótica

## METHODOLOGY OF CONTROL FOR PARALLEL MECHANISMS IN THE PRESENCE OF KINEMATIC SINGULARITIES

### Abstract

In this paper, the theoretical formulation developed by the filtered inverse method is considered to obtain an efficient and robust solution for the kinematic control problem of parallel mechanisms in the presence of singular configurations. This method is based on the differential kinematics equation and consists of an algorithm that dynamically estimates the inverse of the Jacobian matrix instead of computing it instantaneously.

**Keywords:** Singularities, Control, Cinematic, Jacobian Matrix, Robotic

## Sumário

<b>1</b>	<b>Introdução</b>	<b>1</b>
<b>2</b>	<b>Metodologia</b>	<b>2</b>
a	Estrutura da Dissertação . . . . .	2
b	Conceitos fundamentais . . . . .	2
c	Singularidades . . . . .	5
<b>3</b>	<b>Detalhamento</b>	<b>7</b>
a	Posição e orientação de um corpo rígido . . . . .	7
b	Cinemática direta . . . . .	9
c	Cadeia cinemática aberta . . . . .	10
d	Espaço das juntas e espaço operacional . . . . .	11
e	Cinemática diferencial . . . . .	11
f	Jacobiano geométrico . . . . .	11
g	Jacobiano analítico . . . . .	12
h	Jacobiano de representação . . . . .	13
i	Singularidades cinemáticas . . . . .	13
j	Cinemática inversa . . . . .	13
k	Abordagem diferencial . . . . .	14
l	Manipuladores não-redundantes . . . . .	14
m	Erro de posição . . . . .	15
n	Erro de orientação . . . . .	15
o	Quaternion unitário . . . . .	15
p	Manipuladores redundantes . . . . .	16
<b>4</b>	<b>Algoritmos de controle</b>	<b>18</b>
a	Singularidades em manipuladores seriais . . . . .	18
b	Algoritmo DLS . . . . .	19
c	Algoritmo da Inversa Filtrada . . . . .	20
d	Inversa Filtrada em manipuladores paralelos . . . . .	21
e	Modelagem e controle cinemático de manipuladores paralelos . . . . .	22
<b>5</b>	<b>Simulações preliminares</b>	<b>24</b>
a	Ferramentas computacionais . . . . .	24
b	Simulação - Cinemática Direta . . . . .	24
c	Simulação - Cinemática Inversa . . . . .	25
d	Simulação - Manipulador serial . . . . .	26
e	Simulação - Controle de mecanismos seriais em configurações de singularidade . . . . .	28
<b>6</b>	<b>Controle de mecanismos paralelos em configurações de singularidade</b>	<b>32</b>
a	Manipulador paralelo Slider-Crank . . . . .	32
b	Simulações - Slider Crank em Simulink . . . . .	35
c	Manipulador paralelo FourBar Linkage . . . . .	36
<b>7</b>	<b>Conclusão</b>	<b>39</b>

## Lista de Figuras

1	Posição e orientação do corpo rígido. . . . .	7
2	Manipulador SCARA . . . . .	9
3	Descrição da orientação e posição do sistema de coordenadas do efetuador . . . . .	9
4	Transformação de coordenadas em uma cadeia cinemática aberta . . . . .	10
5	Algoritmo de cinemática inversa (caso não-redundante) . . . . .	15
6	Algoritmo de cinemática inversa (caso redundante) . . . . .	17
7	Manipulador antropomórfico com punho esférico . . . . .	18
8	Configurações singulares: (a)cotovelo, (b)ombro e (c)punho . . . . .	19
9	Manipulador serial 6 DoF. . . . .	27
10	Erro de posição e velocidades do efetuador . . . . .	27
11	Posição angular e velocidades das juntas . . . . .	28
12	Trajatória em controle por DLS . . . . .	29
13	Trajatória em controle por Inversa Filtrada . . . . .	29
14	Comparativo do sinal de controle . . . . .	30
15	Manipulabilidade em DLS . . . . .	30
16	Manipulabilidade em Inversa Filtrada . . . . .	31
17	Manipulador paralelo Slider Crank . . . . .	32
18	Dinâmica da singularidade no efetuador . . . . .	33
19	Comparativo do erro de trajetória em controle DLS e Inversa Filtrada . . . . .	33
20	Índice de manipulabilidade da matriz jacobiana de restrição das juntas passivas (IF). . . . .	34
21	Índice de manipulabilidade da matriz jacobiana do manipulador (IF). . . . .	34
22	Índice de manipulabilidade da matriz jacobiana de restrição das juntas passivas (DLS). . . . .	35
23	Índice de manipulabilidade da matriz jacobiana do manipulador (DLS). . . . .	35
24	Simulação do mecanismo paralelo Slider Crank . . . . .	36
25	Modelagem de Blocos - Slider Crank em Simulink . . . . .	36
26	Modelos 26.a 26.b 26.c e 26.d para FourBar Linkage . . . . .	36
27	Erro de trajetória - FourBar Linkage . . . . .	38

## 1 Introdução

A importância do desenvolvimento dos sistemas robóticos e a aplicabilidade da robótica tem sido notória, permitindo maior eficiência na realização de tarefas repetitivas como soldagem ou movimentação de peças, e vem se mostrando 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 como, por exemplo, montadoras de veículos, mas se expandiu significativamente e podemos contemplar hoje a robótica como elemento crucial em locais onde o ser humano não poderia atuar devido à exposição ao perigo. Além disso, a robótica também tem sido utilizada na exploração espacial, em aplicações submarinas submetidas à alta pressão, na medicina, no desarmamento de bombas e cada vez mais se faz presente no cotidiano para a segurança e otimização de processos.

A abordagem de controle cinemático pode ser aplicada para sistemas robóticos de maneira eficiente quando a tarefa de interesse não requer velocidades e acelerações elevadas. Nesse enfoque, podemos considerar dois problemas de interesse: o problema de cinemática direta e o problema de cinemática inversa. O primeiro consiste em calcular a posição e a orientação (pose) do robô a partir do conhecimento da posição das juntas e das dimensões dos elos do robô. O segundo consiste no problema inverso, isto é, obter a configuração das juntas do robô a partir do conhecimento da pose do robô no espaço operacional [16].

Uma solução alternativa para solucionar o problema de cinemática inversa consiste em explorar os graus de mobilidade redundantes dos mecanismos. Nesse contexto, os novos algoritmos de controle procuram evitar configurações singulares e otimizar índices de desempenho relevantes, tais como a distância dos limites mecânicos das juntas, as medidas de manipulabilidade, a distância de obstáculos e consumo de energia

Este trabalho dedica-se, portanto à realização de um estudo de casos por intermédio de simulações numéricas de um novo algoritmo de controle que minimize os impactos do problema de cinemática inversa em manipuladores robóticos paralelos, procurando determinar assim a validade operacional da implementação deste algoritmo em comparação à outro método de controle em malha fechada já utilizado no mercado.



## 2 Metodologia

### a Estrutura da Dissertação

---

#### Algorithm 1 Apresentação

---

Este projeto de conclusão de curso está organizado em 7 capítulos, sendo o primeiro deles esta breve introdução, que procura dar a ambientação do trabalho realizado.

No capítulo 2 são apresentados alguns conceitos básicos relacionados à robótica, tais como a modelagem de um robô manipulador, cinemática, e controle cinemático. Além disso, serão discutidos algumas características específicas da robótica relacionada aos manipuladores paralelos, onde dedica-se a parte principal deste trabalho.

No capítulo 3 expõe-se mais detalhadamente a teoria da cinemática.

No capítulo 4 descreve-se os algoritmos de controle simulados e a modelagem do algoritmo estudado para manipuladores robóticos seriais e paralelos, bem como a sistemática do processo de simulação.

No capítulo 5 apresentam-se as simulações preliminares com objetivo principal de avaliar o método reproduzindo resultados já obtidos na literatura.

No capítulo 6 apresentam-se contribuições à pesquisa ainda não realizadas na literatura por meios de resultados de simulações voltadas para a mecanismos paralelos em configuração de singularidade.

E finalmente no capítulo 7 apresentam-se os comentários, análises finais e a conclusão do trabalho tomando por base o projeto apresentado.

---

### b Conceitos fundamentais

#### 1 - Modelagem

Um manipulador robótico pode ser esquematicamente representado por uma série de corpos rígidos, os elos, em cadeia e conectados por meio de juntas, prismáticas ou de revolução. Cada junta proporciona um grau de mobilidade ao robô. No início desta cadeia é estabelecida a origem, tipicamente a referência inercial escolhida para o sistema e, ao seu final, é montado o efetuador (ou ferramenta). 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

Através dos parâmetros de Denavit-Hartenberg (D-H) [16], é possível modelar o braço robótico. Nesta convenção, 4 (quatro) parâmetros permitem obter o conjunto de equações que descreve a cinemática de uma junta com relação a junta seguinte e vice-versa:

- i. Ângulo de rotação da junta,  $\theta$  ;
- ii. Ângulo de torção da junta,  $\alpha$  ;
- iii. Comprimento do elo,  $A$ ;
- iv. Deslocamento da junta,  $d$ ;

Para um robô manipulador com  $N$  juntas numerados de 1 a  $N$ , há  $N + 1$  elos conectando 2 (duas) juntas, numerados de 0 a  $N$ . O elo 0 é a base do manipulador e o elo  $N$  tem o nome de efetuador. A junta  $j$  conecta o elo  $j - 1$  ao elo  $j$ , e assim a junta  $j$  é responsável por mover o elo  $j$ . Um elo pode ser especificado por 2 parâmetros: seu comprimento  $a_j$  e seu ângulo de torção  $\alpha_j$ . As juntas também podem ser caracterizadas por 2 parâmetros: seu deslocamento  $d_j$  e seu ângulo de rotação  $\theta_j$ .

A transformação do quadro de coordenados de um elo  $j - 1$  para o quadro do elo seguinte  $j$  pode ser calculado através de elementos rotacionais e translacionais da seguinte maneira:

$${}^{j-1}A_j(\theta_j, d_j, a_j, \alpha_j) = T_{Rz}(\theta_j)T_z(d_j)T_x(a_j)T_{Rx}(\alpha_j) \quad , \quad (1)$$

no qual pode ser expandido para:

$${}^{j-1}A_j = \begin{bmatrix} \cos\theta_j & -\sin\theta_j \cos\alpha_j & \sin\theta_j \sin\alpha_j & a_j \cos\theta_j \\ \sin\theta_j & \cos\theta_j \cos\alpha_j & -\cos\theta_j \sin\alpha_j & a_j \sin\theta_j \\ 0 & \sin\alpha_j & \cos\alpha_j & d_j \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (2)$$

Como esperado, o cálculo da função da cinemática direta é recursiva e é obtida de uma maneira sistemática pelo simples produto da contribuição de cada grau de mobilidade. Este procedimento pode ser aplicado em qualquer cadeia cinemática aberta, e pode ser facilmente reescrito em uma operação formal. O espaço de trabalho de um robô manipulador é definido como o conjunto de todas as configurações na qual pode ser alcançado por alguma escolha dos ângulos de junta. O espaço de trabalho é utilizado para planejar uma tarefa para o manipulador executar.

Para controlar um robô, geralmente se faz uso de modelos matemáticos e alguma forma de inteligência para atuar no modelo. O modelo matemático de um robô pode ser obtido através de leis da física que governam o movimento. Com relação a inteligência, isto requer o uso de capacidade sensorial para atuar e reagir a variáveis que foram medidas. Essas ações e reações do robô são o que resultam no design do controlador.

## 2 - Cinemática

Cinemática (do grego , movimento) é o ramo da física que se ocupa da descrição dos movimentos dos corpos, sem se preocupar com a análise de suas causas (dinâmica) [14]. O movimento é um conceito relativo, já que um objeto pode estar em repouso em relação a um primeiro referencial, mas em movimento em relação a um segundo referencial. Um robô pode ser considerado como um corpo rígido, isto é, todas as partes do robô mantêm sempre as mesmas distâncias relativas às outras partes independentemente da nova posição e orientação do robô. A posição de um corpo rígido em qualquer instante de tempo pode ser determinada indicando a posição de um ponto do corpo, a orientação de um eixo fixo em relação ao corpo e um ângulo de rotação em volta a esse eixo. A maioria dos manipuladores modernos consiste de um conjunto de elos rígidos conectados juntamente a uma série de juntas. Motores são acoplados às juntas para que o movimento do mecanismo seja controlado a fim de realizar determinada tarefa. Uma ferramenta, geralmente uma garra, é acoplada ao fim do robô para interagir com o ambiente. Geralmente, deseja-se controlar a pose do efetuador de um robô e sua trajetória para determinado instante de tempo. A pose do efetuador pode ser entendida pela descrição de sua posição e orientação com relação a determinado referencial. Uma importante característica de trajetória a ser seguida tem a ver com suavidade (smoothness), isto é, a posição e orientação devem variar suavemente com o decorrer do tempo. Isto quer dizer, as primeiras derivadas temporais devem ser contínuas.

### i. Cinemática Direta

O problema da cinemática direta está em calcular a posição e orientação da extremidade final da cadeia cinemática (efetuador) com respeito a um sistema de coordenadas de referência, tipicamente posicionado na base da estrutura mecânica, a partir do conhecimento das variáveis de juntas (ângulos ou deslocamentos). No nosso caso, restringiremos o estudo da cinemática direta para manipuladores de cadeia aberta (serial), onde cada par de elos é conectado por uma junta de revolução. As cadeias cinemáticas abertas correspondem a estruturas em que há apenas uma sequência de elos conectando suas extremidades, enquanto cadeias fechadas correspondem a estruturas em que uma sequência de elos forma um laço.

A solução da cinemática direta é única e pode ser obtida tanto na forma analítica quanto na forma numérica através de um processo sistemático. A pose do efetuador de um manipulador tem 6 (seis) graus de liberdade, onde 3 (três) para translação e 3 (três) para rotação. Com base nestas definições, a cinemática direta pode ser expressa através da seguinte equação:

$$\xi_E = \chi(q), \quad (3)$$

onde a pose do efetuador é uma função das variáveis de junta. A pose pode ser calculada, assim, através do produto da matriz de transformação de cada elo para um manipulador com N-DoF:

$$\xi_E \sim {}^0T_E = {}^0A_1 {}^1A_2 \dots {}^{N-1}A_N, \quad (4)$$

ii. Redundância cinemática O manipulador é redundante quando o número de graus de mobilidade é maior que a quantidade de variáveis (graus de liberdade) necessárias para descrever uma tarefa. Dado que cada junta adiciona somente um grau de mobilidade, um manipulador de n juntas tem n graus de mobilidade. Desta forma, considerando

uma tarefa com  $m$  graus de liberdade, um manipulador é redundante se  $m < n$

$m < n$  Neste caso, uma dada configuração do manipulador pode ser atingida com diversos valores de juntas (em geral infinitos).

iii. Cinemática Inversa O problema da cinemática inversa consiste em determinar as variáveis de juntas que resultam em uma determinada posição e orientação do efetuador. A cinemática inversa é um problema muito mais difícil de ser resolvido do que a cinemática direta. Isso se deve pelo fato da cinemática inversa ser altamente não linear. Descobrir o inverso da velocidade e aceleração são problemas lineares, e mais simples de serem resolvidos assim que o problema de descobrir a posição inversa já foi solucionado. O maior problema da cinemática inversa são as múltiplas soluções para determinada posição e orientação do efetuador, pois o sistema deve ser capaz de escolher somente uma. O critério para que se possa tomar uma decisão pode variar, mas a mais comumente utilizada é a solução mais próxima. A cinemática inversa pode ser expressa através da seguinte equação:

$$q = \chi^{-1}(\xi) \quad , \quad (5)$$

Em geral essa função não apresenta soluções únicas e para determinada classe de robôs manipuladores, não existem soluções de forma fechada, necessitando-se assim realizar uma análise numérica. Muitos pesquisadores preferem métodos numéricos para resolver o problema da cinemática indireta por evitar a dificuldade de encontrar soluções analíticas [15][16][17]. Normalmente, a abordagem analítica é apropriada para aplicações em tempo real, porque todas soluções podem ser encontradas e é computacionalmente mais rápida quando comparada com abordagens numéricas. Cada manipulador deve ser estudado previamente para que se possa entender seu espaço de trabalho. Para um manipulador com menos de seis graus de liberdade, pode não ser possível obter determinada posição e orientação no espaço tridimensional.

#### iv. Cinemática Diferencial

Nas seções anteriores, as equações de cinemática direta e inversa estabeleceram as relações entre as coordenadas de juntas e a posição e orientação do efetuador. A cinemática diferencial 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 é realizado através da matriz Jacobiana, a qual é dependente da configuração do manipulador. O Jacobiano do manipulador é uma das relações mais importantes para a análise e controle do movimento de um robô manipulador, pois permite:

i. Planejamento e execução de trajetórias suaves; ii. Determinação de singularidades; iii. Cálculo das equações dinâmicas de movimento; iv. Transformação de forças e torques do efetuador para as juntas.

Descrevemos a cinemática direta como sendo uma função do tipo:

$$\xi = \chi(q) \quad , \quad (6)$$

Derivando os dois lados da equação, temos:

$$v = J(q)\dot{q} \quad , \quad (7)$$

, onde  $V = (V_x V_y V_z V_x V_y)$  é o espaço de velocidades e representa as componentes de velocidade translacionais e rotacionais, e  $\dot{q} = [\dot{q}_1, \dots, \dot{q}_n]$  é o vetor de velocidade  $n \times 1$  da junta. A matriz  $J(q)$  é o manipulador Jacobiano ou Jacobiano geométrico.

A dimensão geral da matriz Jacobiana é  $m \times n$ , onde  $m$  é o número de linhas, que é igual ao número de graus de liberdade do campo de trabalho do robô, e  $n$  é o número de colunas, que é igual ao número de articulações do robô. Assim, observa-se que o número de linhas desta matriz não é fixa, podendo variar de acordo com cada problema, em função do robô a ser utilizado.

O problema inverso, de mapear as velocidades do efetuador a partir das velocidades nas juntas, pode ser descrito por:

$$\dot{q} = J(q)^{-1}v \quad , \quad (8)$$

Porém,  $J(q)$  deve ser uma matriz quadrada e não-singular. Caso o Jacobiano seja singular, há estratégias de manipulação da matriz para que sejam calculadas as velocidades. Uma solução para o problema da inversão da equação de cinemática diferencial na vizinhança de uma singularidade é fornecida pelo método DLS (damped inverse Jacobian) [18]. Nesta proposta, a inversa ou pseudo-inversa da matriz Jacobiana é substituída por:

$$j^* = J^T(JJ^T + \delta I)^{-1} \quad , \quad (9)$$

, assim  $\dot{q} = (J(q) + pI)^{-1}V$  onde  $p$  é uma pequena constante que multiplica a matriz identidade. O Jacobiano pseudo-inverso pode ser escrito como  $J^+ = (J^T J)^{-1} J^T$  Assim podemos:

$$\dot{q} = J(q)^+ v \quad , \quad (10)$$

## c Singularidades

### v. Configuração Singular

Para qualquer robô, redundante ou não, é possível descobrir algumas configurações, chamadas de configuração singular, no qual o número de graus de liberdade do efetuador é inferior à dimensão na qual ele está operando. Configurações singulares podem ocorrer quando:

- i. Dois eixos de juntas prismáticas se tornam paralelos;
- ii. Dois eixos de juntas de revolução se tornam idênticos.

Em configurações singulares, o efetuador perde um ou mais grau de liberdade, já que equações cinemáticas se tornam linearmente independentes. Posições singulares devem ser evitadas, pois as velocidades requeridas para mover o efetuador se tornam, teoricamente, infinitas. É importante ressaltar também 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 de juntas. Configurações singulares podem ser determinadas através da matriz Jacobiana. A matriz Jacobiana,  $J$ , relaciona o deslocamento infinitesimal do efetuador com as variáveis infinitesimais das juntas. A dimensão da matriz  $J$  é  $m \times n$ , onde  $m$  é o número de graus de liberdade do efetuador, e  $n$  é o número de juntas. Configurações onde o Jacobiano não tem posto completo, correspondem às singularidades do robô, que podem ser classificadas em dois tipos:

- i. Singularidade do espaço de fronteira: Aqueles que ocorrem quando o manipulador está esticado, ou dobrado sobre si mesmo
  - ii. Singularidade do espaço interior (ou internas): Aqueles que ocorrem quando dois ou mais eixos de alinham
- Se o manipulador tem menos de seis graus de liberdade, a configuração singular corresponde a configuração na qual o número de graus de liberdade decai. Isso é caracterizado pelo manipulador Jacobiano caindo de posto, isto é, duas ou mais colunas da matriz Jacobiana  $J$  se tornam linearmente dependentes

- i. A manipulabilidade de um robô pode ser descrita como sua habilidade de mover-se livremente em todas direções dentro do espaço de trabalho. A manipulabilidade pode ser classificada em:
- ii. A habilidade de alcançar certas posições ou conjunto de posições;
- iii. A habilidade de mudar de posição ou orientação para determinada configuração.

### vi. Jacobiano pseudo-inverso

Um método eficiente de se obter o Jacobiano para matrizes não quadradas, é calcular o Jacobiano pseudo-inverso. O cálculo desta matriz depende do seu número de linhas e colunas, e pode ser determinado da seguinte maneira para o caso em que o Jacobiano apresenta posto completo:

- i. Para o caso underconstrained (se tem mais colunas do que linhas, isto é,  $m < n$ );

$$J^\# = J^T (J J^T)^{-1} \quad , \quad (11)$$

- ii. Para o caso overconstrained (se há mais linhas do que colunas, isto é,  $n < m$ );

$$J^\# = (J^T J)^{-1} J^T \quad , \quad (12)$$

- iii. Para números iguais de linhas e colunas ( $m = n$ ).

$$J^\# = J^{-1} \quad , \quad (13)$$

## MÉTODO DLS INVERSO

$$\mathbf{J}^\dagger(\mathbf{q}) \approx \mathbf{J}_{DLS}^*(\mathbf{q}), \quad \mathbf{J}_{DLS}^*(\mathbf{q}) := (\mathbf{J}(\mathbf{q}) \mathbf{J}^\top(\mathbf{q}) - \alpha \mathbf{I})^{-1}, \quad (14)$$

$\alpha \in \mathbb{R}$ : fator de amortecimento;

$$\alpha = \begin{cases} 0, & \omega \geq \omega_0 \\ \alpha_0 \left(1 - \frac{\omega}{\omega_0}\right)^2, & \omega < \omega_0 \end{cases}, \quad (15)$$

$\alpha_0 \in \mathbb{R}$ : fator de escalonamento;  $\omega_0 \in \mathbb{R}$ : limite da vizinhança de singularidade;

$$\omega = \sqrt{\det(\mathbf{J}(\mathbf{q}) \mathbf{J}^\top(\mathbf{q}))}. \quad (16)$$

$\omega \in \mathbb{R}$ : índice de manipulabilidade;

Uma outra solução, recente e promissora, é o método da Inversa Filtrada que é baseado na equação de cinemática diferencial e consiste em um algoritmo que estima dinamicamente a inversa da matriz Jacobiana, ao invés de computar a inversa instantaneamente [69]. O algoritmo possui a característica notável de ser robusto a singularidades e possuir apenas um parâmetro (i.e., um ganho de atualização) a ser sintonizado. Além disso, é possível estender o método incluindo na estimação dinâmica uma função custo a ser otimizada (e.g., limite mecânico das juntas, desvio de obstáculos). Então, a ideia chave consiste em substituir a inversa do Jacobiano no algoritmo iterativo pela sua inversa filtrada e atualizar a estimativa da matriz inversa utilizando uma lei de atualização tipo gradiente.

## MÉTODO DA INVERSA FILTRADA

$$\mathbf{J}^\dagger(\mathbf{q}) \approx \mathbf{J}_{FI}^*(\mathbf{q}), \quad \mathbf{J}_{FI}^*(\mathbf{q}) := \Theta \Theta^\top \mathbf{J}^\top(\mathbf{q}). \quad (17)$$

- Sinais de Erro à Direita  $S_r \in \mathbb{R}^{m \times m}$  e de Erro à Esquerda  $S_\ell \in \mathbb{R}^{n \times n}$ :

$$S_r := \mathbf{J}(\mathbf{q}) \Theta - \mathbf{I}_m, \quad S_\ell := \Theta \mathbf{J}(\mathbf{q}) - \mathbf{I}_n. \quad (18)$$

- Lei de Atualização:

$$\dot{\Theta} = -(\Gamma_r \mathbf{J}^\top(\mathbf{q}) S_r + \Gamma_\ell S_\ell \mathbf{J}^\top(\mathbf{q})). \quad (19)$$

$\Gamma_r \in \mathbb{R}^{n \times n}$ ,  $\Gamma_\ell \in \mathbb{R}^{n \times n}$ : matrizes de ganho de atualização à direita e à esquerda;

### 3 Detalhamento

#### 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.

#### a 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 Oxyz definido por três eixos ortonormais x, y e 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 p nos eixos do sistema de coordenadas inercial. Assim, a trajetória efetuada pela partícula pode ser representada pela curva  $p(t) = (p_x(t), p_y(t), p_z(t)) \in \mathbb{R}^3$ . 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 p e q de um corpo rígido satisfazem

$$\|p(t) - q(t)\| = \|p(0) - q(0)\| = C \quad \forall t, \quad (20)$$

O corpo rígido é completamente descrito no espaço por sua posição e orientação relativas ao sistema de coordenadas inercial. A Figura 1 mostra o sistema de coordenadas de referência (ou inercial) e um corpo rígido, onde quaisquer dois pontos a ele pertencentes satisfazem (1).

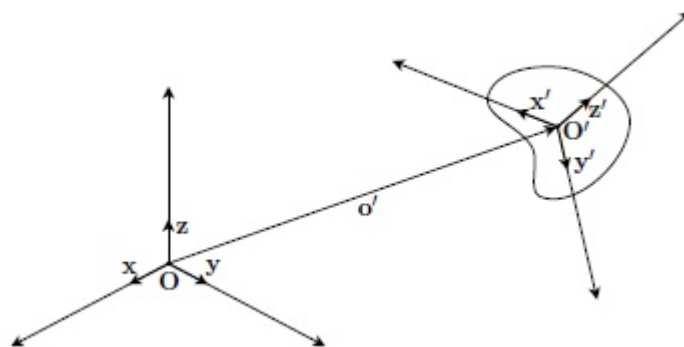


Figura 1: Posição e orientação do corpo rígido.

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

$$o' = o'_x + o'_y + o'_z, \quad (21)$$

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

$$o' = \begin{bmatrix} o'_x \\ o'_y \\ o'_z \end{bmatrix}, \quad (22)$$

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  $O' - o'_x, o'_y, o'_z$  o sistema de coordenadas do corpo e  $o'_x, o'_y, o'_z \in \mathbb{R}^3$  as coordenadas dos eixos principais deste novo sistema relativos ao sistema de coordenadas Oxyz (ver Figura 1). Estes vetores são expressos como:

$$\begin{aligned} x' &= x'_x x + x'_y z + x'_z z \\ y' &= y'_x x + y'_y z + y'_z z \\ z' &= z'_x x + z'_y z + z'_z z \end{aligned}, \quad (23)$$

Adotando uma notação compacta, os vetores unitários em (23) descrevendo a orientação do corpo com respeito ao sistema de coordenadas de referência podem ser combinados em uma matriz

$$R = \begin{bmatrix} x' & y' & z' \end{bmatrix} = \begin{bmatrix} x'_x & y'_x & z'_x \\ x'_y & y'_y & z'_y \\ x'_z & y'_z & z'_z \end{bmatrix} = \begin{bmatrix} x'^T x & y'^T x & z'^T x \\ x'^T y & y'^T y & z'^T y \\ x'^T z & y'^T z & z'^T z \end{bmatrix}, \quad (24)$$

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  $p$  a representação do ponto P com respeito ao sistema de coordenadas Oxyz e  $p'$  a representação deste mesmo ponto P com respeito ao sistema de coordenadas  $O'x'y'z'$ . Como  $p$  e  $p'$  são representações para o mesmo ponto P, obtém-se

$$P = p'_x x' + p'_y y' + p'_z z' = \begin{bmatrix} x' & y' & z' \end{bmatrix} \begin{bmatrix} p'_x \\ p'_y \\ p'_z \end{bmatrix} = R p', \quad (25)$$

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 R. Desta forma,  $R^T R = I_{3 \times 3}$  e, portanto,  $\det R = \pm 1$ . O determinante da matriz de rotação pode ser encontrado através da seguinte relação algébrica:

$$\det R = x'^T (y' \times z') \quad (26)$$

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

$$R_x(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\alpha & -s\alpha \\ 0 & s\alpha & c\alpha \end{bmatrix}; R_y(\beta) = \begin{bmatrix} c\beta & 0 & s\beta \\ 0 & 1 & 0 \\ -s\beta & 0 & c\beta \end{bmatrix}; R_z(\gamma) = \begin{bmatrix} c\gamma & -s\gamma & 0 \\ s\gamma & c\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad (27)$$

onde  $c\alpha = \cos(\alpha)$  e  $s\alpha = \sin(\alpha)$  para  $\alpha = \beta, \gamma$ . 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  $O_1 x_1 y_1 z_1$  e  $O_2 x_2 y_2 z_2$  dois sistemas de coordenadas ortonormais. A matriz de transformação homogênea  $T_{12}$  representa de forma compacta a transformação de coordenadas (translação e rotação) entre os dois sistemas de coordenadas por

$$T_{12} = \begin{bmatrix} R_{12} & (p_{12})_1 \\ 0 & 1 \end{bmatrix}, \quad (28)$$

e uma vez que  $p_{12} \in \mathbb{R}^3$  e  $R_{12} \in SO(3)$ , esta matriz pertence ao Grupo Especial Euclidiano  $SE(3) = \mathbb{R}^3 \times SO(3)$ . No que se segue, assim como em (28), é utilizado um subíndice para explicitar o sistema de coordenadas escolhido como referência, sendo  $(p)_i$  a representação do ponto p no sistema de coordenadas  $O_i - x_i y_i z_i$ . A equação (25) é então reescrita:

$$(p)_0 = R_{0i}(p)_i, \quad (29)$$



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

## b 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 realiza efetivamente a manipulação de objetos no espaço de trabalho. Um exemplo de manipulador robótico é mostrado na Figura 2.

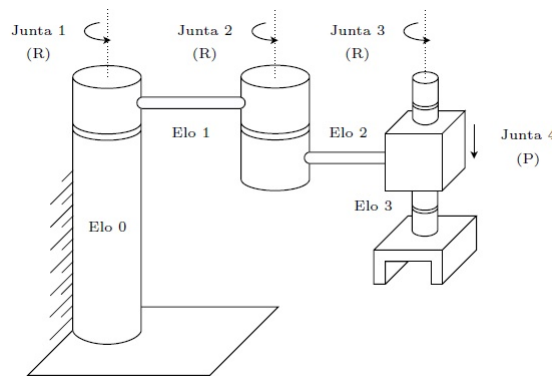


Figura 2: Manipulador SCARA

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.

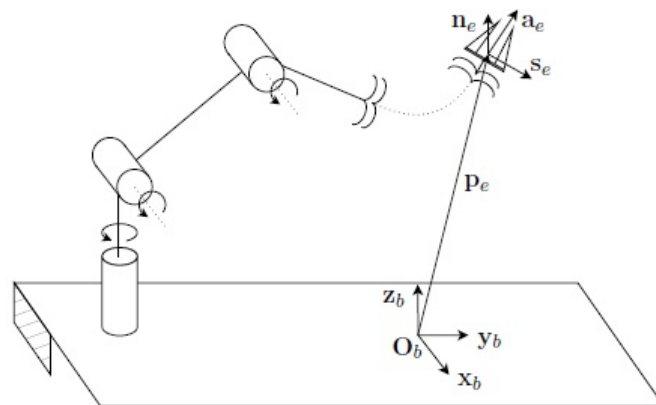


Figura 3: Descrição da orientação e posição do sistema de coordenadas do efetuador



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  $O_b - x_b y_b z_b$ , a cinemática direta é expressa pela matriz de transformação homogênea

$$T_{be}(q) = \begin{bmatrix} n_e(q) & s_e(q) & a_e(q) & (p_e(q))_b \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (30)$$

onde  $q$  corresponde ao vetor de variáveis das juntas,  $n_e$ ,  $s_e$  e  $a_e$  são os vetores unitários que definem os eixos do sistema de coordenadas do efetuador, convenientemente escolhido de acordo com a geometria da tarefa, e  $(p_e)_b$  corresponde ao vetor de posição da origem deste sistema de coordenadas em relação ao sistema de coordenadas de referência  $O_b - x_b y_b z_b$ . Uma escolha típica para os vetores  $n_e$ ,  $s_e$  e  $a_e$  é apresentada na Figura 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].

### c 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  $n$ .

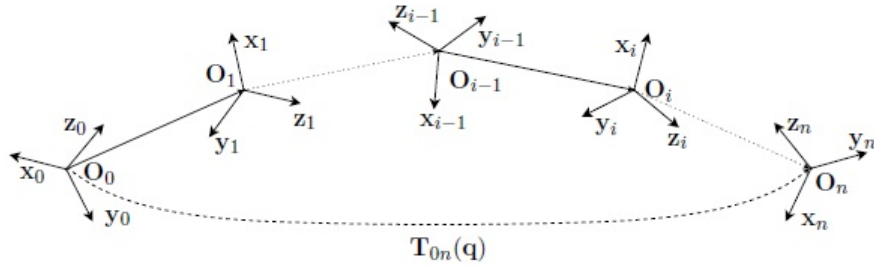


Figura 4: Transformação de coordenadas em uma cadeia cinemática aberta

A transformação homogênea que descreve a posição e orientação do sistema de coordenadas  $n$  tendo o sistema de coordenadas 0 (Figura 4) como referência é obtida pelo produto de matrizes de transformação homogênea entre elos consecutivos

$$T_{0n}(q) = T_{01}(q_1)T_{12}(q_2) \cdots T_{n-1,n}(q_n), \quad (31)$$

onde  $q = [q_1 q_2 \cdots q_n]^T$  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  $T_{i-1,i}(q_i)$  (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

$$T_{be}(q) = T_{b0}T_{0n}(q)T_{ne}, \quad (32)$$

onde  $T_{b0}$  e  $T_{ne}$  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  $n$ .

## d 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

$$x_e = \begin{bmatrix} p_e \\ \phi_e \end{bmatrix} \in \mathbb{R}^m, m \leq n, \quad (33)$$

onde  $p_e$  descreve a posição do efetuador e  $\phi_e$  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  $x_e$  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 ( $O$ ). Por outro lado, o espaço de configuração onde o vetor  $q \in \mathbb{R}^n$  é definido é chamado espaço das juntas ( $Q$ ). A cinemática direta pode ser escrita, de forma alternativa a (32), como:

$$x_e = h(q), \quad (34)$$

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  $\theta_i$  e variáveis de juntas prismáticas por  $d_i$ .

## e Cinemática diferencial

Na Seção anterior foi obtida uma descrição da postura (posição e orientação) do efetuador  $x_e$  no instante  $t$  a partir do vetor  $q$  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 anterior, 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.

## f 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

$$\omega_i = \dot{q}_i h_i, \quad (35)$$

$$v_i = \omega_i \times p_{ie} = \dot{q}_i h_i \times p_{ie}, \quad (36)$$

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

$$\omega_i = 0 \quad , \quad (37)$$

$$v_i = \dot{q}_i h_i \quad , \quad (38)$$

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  $v_e$  e  $\omega_e$  por

$$V_e = \begin{bmatrix} v_e \\ \omega_e \end{bmatrix} = \begin{bmatrix} h_1 \times p_{1e} & h_2 \times p_{2e} & \cdots & h_n \times p_{ne} \\ h_1 & h_2 & \cdots & h_n \end{bmatrix} \dot{q} \quad , \quad (39)$$

ou ainda

$$V_e = \begin{bmatrix} J_p(q) \\ J_o(q) \end{bmatrix} \dot{q} = J_G(q) \dot{q} \quad , \quad (40)$$

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  $O_b - x_b y_b z_b$ , basta utilizar a representação no eixo de interesse dos vetores presentes na matriz Jacobiana, conforme (29).

## g Jacobiano analítico

Como visto na Seção anterior, a postura do efetuador pode ser descrita pelo seu vetor posição  $p$  e uma representação mínima ou adequada da orientação  $\phi$ . Esta configuração, como visto, depende das variáveis das juntas e pode ser expressa de forma compacta por (24). 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,

$$\dot{p} = \frac{\partial p}{\partial q} \dot{q} = J_p(q) \dot{q} \quad , \quad (41)$$

Dada a representação da orientação  $\phi$ , a sua derivada temporal é obtida

$$\dot{\phi} = \frac{\partial \phi}{\partial q} \dot{q} = J_\phi(q) \dot{q} \quad , \quad (42)$$

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

$$\dot{x} = \begin{bmatrix} \dot{p} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} J_p(q) \\ J_\phi(q) \end{bmatrix} \dot{q} = J(q) \dot{q} \quad , \quad (43)$$

onde o Jacobiano analítico do manipulador é dado por

$$J = \frac{\partial h(q)}{\partial q} \in \mathbb{R}^{m \times n} \quad , \quad (44)$$

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

## h Jacobiano de representação

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

$$\dot{\phi} = J_R(\phi)\omega \quad , \quad (45)$$

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

$$J = \begin{bmatrix} I & 0 \\ 0 & J_R(\phi) \end{bmatrix} J_G = M_R J_G \quad , \quad (46)$$

## i Singularidades cinemáticas

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

$$V_e = J_G(q)\dot{q} \quad , \quad (47)$$

entre o vetor  $\dot{q}$  de velocidades das juntas e o vetor  $V_e = [v_e^T \ \omega_e^T]^T$  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].

## j 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  $h : Q \rightarrow O$  e uma postura desejada  $x_d \in \mathbb{R}^m$ , deseja-se resolver a equação

$$h(q) = x_d \quad , \quad (48)$$

para algum  $q \in \mathbb{R}^n$ . 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 destre 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 (48). 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 subproblemas 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 subproblemas de Paden-Kahan, o leitor é convidado a consultar [33].

## k Abordagem diferencial

Como foi visto na Seção anterior, 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  $q(t_0)$ , definir  $\dot{q}$  de forma que

$$q(t) \rightarrow q^* \quad , \quad (49)$$

onde  $q^*$  corresponde a uma configuração de juntas que satisfaz (48). Nota-se que  $q$  é desconhecido. Como será visto, a definição de  $\dot{q}$  a partir de variáveis de erro no espaço de operação  $O$  e não no espaço das juntas  $Q$  resultará na utilização da inversa (ou pseudo-inversa) da matriz Jacobiana

## l Manipuladores não-redundantes

O objetivo de controle é conduzir a estrutura mecânica de uma postura inicial  $x(t_0) = (p(t_0), \phi(t_0))$  para uma postura desejada  $x_d(t) = (p_d(t), \phi_d(t))$ . 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:

$$x \rightarrow x_d, \quad e := x_d - x \rightarrow 0, \quad , \quad (50)$$

onde  $e \in \mathbb{R}^m$  corresponde ao erro de rastreamento no espaço operacional. A obtenção da dinâmica do erro  $e$  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 (50) e a partir da cinemática diferencial.

$$\dot{e} = \dot{x}_d - \dot{x} = \dot{x}_d - J(q)\dot{q} \quad , \quad (51)$$

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  $\dot{q}$  tal que a variável de erro  $e$  convirja para zero.

A matriz  $J(q)$  é quadrada ( $m = n$ ), e assumindo posto completo (não-singular e portanto inversível), a escolha

$$\dot{q} = J(q)^{-1}v, \quad v = \dot{x}_d + \Lambda e, \quad , \quad (52)$$

conduz ao seguinte sistema linear:

$$\dot{e} + \Lambda e = 0. \quad , \quad (53)$$

A escolha de uma matriz  $\Lambda \in \mathbb{R}^{m \times m}$  positiva definida garante a estabilidade assintótica do erro, sendo suficiente uma escolha  $-\Lambda$  Hurwitz. O erro tende a zero ao longo da trajetória com uma taxa de convergência que depende dos autovalores da matriz de ganho  $\Lambda$ . A utilização de  $J(q)^{-1}$  permite a linearização do sistema, compensando o termo não-linear  $J(q)$ , e a utilização do termo feedforward  $\dot{x}_d$  garante a manutenção do erro em 0 ao longo da trajetória, independente do tipo de sinal de referência  $x_d$ . O bloco correspondente ao algoritmo implementado é apresentado na Figura 5.

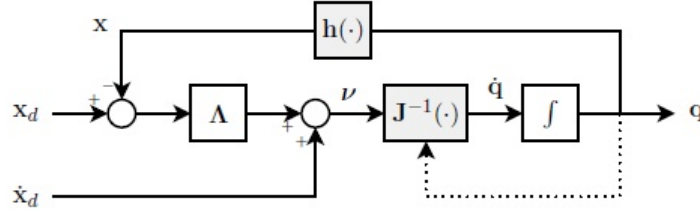


Figura 5: Algoritmo de cinemática inversa (caso não-redundante)

## m Erro de posição

A definição do erro de posição é bastante natural

$$e_p = p_d - p(q) \quad , \quad (54)$$

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

$$\dot{e}_p = \dot{p}_d - \dot{p} \quad , \quad (55)$$

## n 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.

A expressão para o erro de orientação é escolhida de forma análoga à equação (54), ou seja,

$$e_o = \phi_d - \phi(q) \quad , \quad (56)$$

e sua dinâmica é expressa por

$$\dot{e}_o = \dot{\phi}_d - \dot{\phi} \quad , \quad (57)$$

Assim, assumindo que não haja singularidades cinemáticas ou de representação, a solução de controle é obtida de (52), isto é

$$\dot{q} = J(q)^{-1} \begin{bmatrix} \dot{p}_d + \Lambda_p e_p \\ \dot{\phi}_d + \Lambda_o e_o \end{bmatrix} \quad , \quad (58)$$

onde  $\Lambda_p$  e  $\Lambda_o$  são matrizes positivas definidas, garantindo a estabilidade assintótica do erro  $e$ .

## o Quaternion unitário

A representação da orientação por quaternions requer a definição de um erro de orientação adequado. Assim, sejam  $Q_d = \{\eta_d, \epsilon_d\}$  e  $Q_e = \{\eta_e, \epsilon_e\}$  os quaternions associados a  $R_d$  e  $R_{0e}$ , respectivamente. O erro de orientação será descrito pela matriz de rotação (ou atitude)  $R_\phi$ . Esta matriz pode ser definida, através do enfoque inercial, como  $R_\phi = R_d R_{0e}^T$ . O objetivo de controle é então formalizado como:

$$R_\phi \rightarrow I \quad , \quad (59)$$

A representação da matriz  $R_\phi$  em quaternions é dada por

$$\Lambda Q = Q_d * Q_e^{-1} \quad , \quad (60)$$

A representação da matriz identidade  $I$  em quaternions unitários é dada por  $Q_I = \{1, 0\}$ . Assim,  $\Lambda Q = \{1, 0\}$  se e somente se  $R_d$  e  $R_{0e}$  estiverem alinhadas. Portanto, é suficiente definir o erro de orientação como

$$e_0 = \Lambda e = \eta_e(q)\epsilon_d - \eta_d\epsilon_e(q) - \epsilon_d \times \epsilon_e(q) \quad , \quad (61)$$

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

$$\dot{q} = J_G(q)^{-1} \begin{bmatrix} \dot{p}_d + \Lambda_p e_p \\ \omega_d + \Lambda_0 e_0 \end{bmatrix} \quad , \quad (62)$$

onde o Jacobiano geométrico foi utilizado. Substituindo (62) em (40), obtém-se

$$\omega_d - \omega_e + \Lambda_0 e_0 = 0 \quad , \quad (63)$$

A equação obtida é não-linear em  $e_o$  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  $Q_e$  e a velocidade angular  $\omega_e$  ([16], pg. 140):

$$\dot{\eta}_e = -\frac{1}{2}\epsilon_e^T \omega_e \quad , \quad (64)$$

$$\dot{\epsilon}_e = \frac{1}{2}(\eta_e I_{3 \times 3} - \epsilon_e) \times \omega_e \quad , \quad (65)$$

que é a conhecida equação de propagação do quaternion, estando explícito o Jacobiano de representação. O estudo da estabilidade de (63) considera a seguinte função de Lyapunov [50]:

$$V(\Delta\eta, \Delta\epsilon) = (\eta_d - \eta_e)^2 + (\epsilon_d - \epsilon_e)^T (\epsilon_d - \epsilon_e) \quad , \quad (66)$$

Considerando as equações (63), (64) e (65), a diferenciação temporal de (66) resulta em

$$\dot{V}(\Delta\eta, \Delta\epsilon) = -e_o^T \Lambda_o e_o < 0 \quad , \quad (67)$$

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

## p Manipuladores redundantes

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

$$\dot{q} = J^\dagger(q)v \quad , \quad (68)$$

onde  $J^\dagger = J^T(JJ^T)^{-1} \in \mathbb{R}^{n \times m}$  é a matriz pseudo-inversa à direita de  $J$ . A solução obtida a partir de (68) minimiza localmente a norma das velocidades das juntas, desde que  $(t)$  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

$$v = \dot{x}_d + \Lambda e \quad , \quad (69)$$

a dinâmica do erro de postura é governada por  $\dot{e} + \Lambda e = 0$ . Novamente, a partir de uma escolha apropriada de  $\Lambda$  como uma matriz positiva definida, o sistema em malha fechada é exponencialmente estável e, consequentemente,  $\lim_{t \rightarrow \infty} e(t) = 0$ . 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 (68). Então, devido à presença de  $n - m$  graus de liberdade redundantes, a inversão desejada (68) pode ser modificada introduzindo-se um termo de projeção no espaço nulo de  $J$ , isto é,

$$\dot{q} = J^\dagger v + (I - J^\dagger J) \dot{q}_0 \quad , \quad (70)$$

onde  $\dot{q}_0 \in \mathbb{R}^n$  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 6.

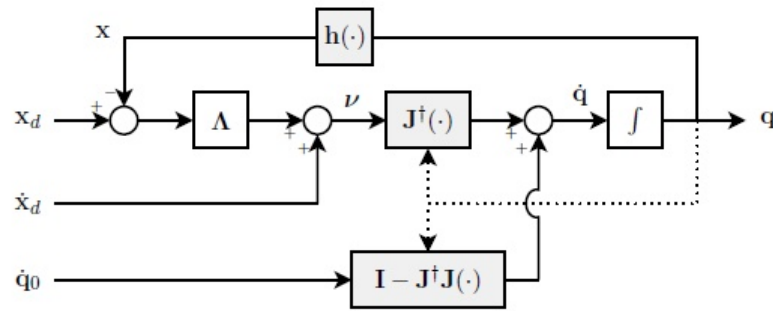


Figura 6: Algoritmo de cinemática inversa (caso redundante)



## 4 Algoritmos de controle

A solução para o problema de cinemática inversa baseada na inversa (ou pseudoinversa) 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 seriais e paralelos. Serão apresentados também dois algoritmos 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]. e o algoritmo da inversa filtrada [69].

### a Singularidades em manipuladores seriais

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 7, recebendo este nome devido à sua similaridade com um braço humano.

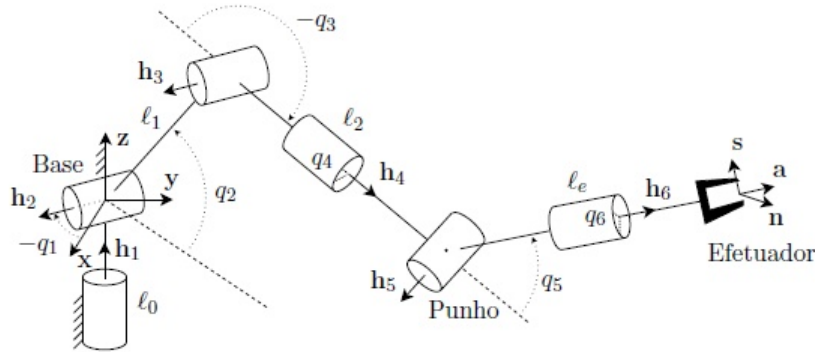


Figura 7: Manipulador antropomórfico com punho esférico

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  $p_\omega$  e  $R_{0e}$  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

$$p_\omega = f_1(q_1, q_2, q_3) \quad , \quad (71)$$

$$R_{0e} = f_2(q_1, q_2, q_3, q_4, q_5, q_6) \quad , \quad (72)$$

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  $p_\omega$  e a velocidade angular  $\omega_e$  com as velocidades das variáveis de juntas sendo dada por

$$\begin{bmatrix} \dot{p}_\omega \\ \omega_e \end{bmatrix} = \begin{bmatrix} J_{11} & 0_{3 \times 3} \\ J_{21} & J_{22} \end{bmatrix} \begin{bmatrix} \dot{q}_p \\ \dot{q}_0 \end{bmatrix} \quad , \quad (73)$$

onde  $q_p \in \mathbb{R}^3$  corresponde ao vetor de velocidades das três primeiras juntas e  $q_0 \in \mathbb{R}^3$  corresponde ao vetor de velocidades das três últimas juntas, ou juntas do punho. Nota-se a partir de (73) 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 (73), o Jacobiano é triangular por blocos e seu determinante pode então ser calculado a partir dos determinantes de cada bloco. Assim,

$$\det(J_G) = \det(J_{11})\det(J_{22}) \quad , \quad (74)$$

A igualdade  $\det(J_{11}) = 0$  corresponderá a uma singularidade de braço (singularidade de ombro e de cotovelo), resultante do movimento dos 3 primeiros elos, e  $\det(J_{22}) = 0$  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 8 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 8.b e 8.c, respectivamente).

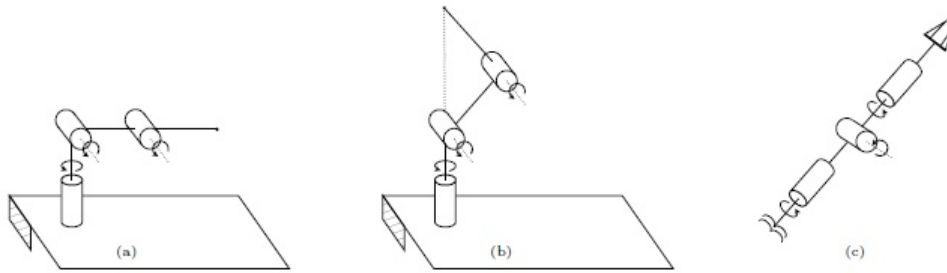


Figura 8: Configurações singulares: (a)cotovelo, (b)ombro e (c)punho

## b 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

$$J^* = J^T(JJ^T + \delta I)^{-1}, \quad \delta \geq 0 \quad , \quad (75)$$

onde  $\delta$  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 constitui um tema de pesquisa bastante relevante, como em [40],[41] e [43]. O fator de amortecimento pode ser calculado como [40]:

$$\delta = \begin{cases} 0 & , \quad \omega \geq \omega_0 \\ \delta_0(1 - \frac{\omega}{\omega_0}) & , \quad \omega < \omega_0 \end{cases} \quad , \quad (76)$$

onde  $\omega = \sqrt{\det(J(q)J^T(q))}$  é a medida de manipulabilidade do manipulador,  $\delta_0$  é uma constante que estabelece um fator de escalamento nas configurações singulares e  $\omega_0$  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  $\omega^2$  ao invés de  $\delta$ . 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 (75), a solução para a cinemática inversa pode ser calculada como

$$\dot{q} = J^T(JJ^T + \delta I)^{-1}v \quad , \quad (77)$$

onde  $v = \dot{x}_d + \Lambda e$ . A solução (77) satisfaz a seguinte condição:

$$\min \quad \dot{q} \quad \|v - J(q)\dot{q}\|^2 + \delta\|\dot{q}\|^2 \quad , \quad (78)$$

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 pequenos resulta em soluções precisas mas com baixa

robustez na vizinhança de singularidades. Por outro lado, a escolha de valores  $\delta$  elevados resulta em pouca precisão de rastreamento mesmo quando soluções factíveis e precisas são possíveis [48].

## c Algoritmo da Inversa Filtrada

Considere uma planta SISO de 1a ordem descrita por

$$\dot{y} = k(t)u \quad , \quad (79)$$

onde  $u \in \Lambda_\infty$  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

$$u = k^{-1}(t)v, \quad v = \dot{y}_d + \lambda e \quad , \quad (80)$$

onde  $\lambda > 0$ . Como foi visto, o erro de rastreamento tenderá exponencialmente a 0 de acordo com o valor do ganho  $\lambda$  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  $\frac{1}{k(t)}$ , mas uma função  $\theta(t)$  atualizada dinamicamente tal que

$$k\theta \rightarrow 1 \quad , \quad (81)$$

Para estabelecer uma dinâmica adequada para  $\theta(t)$ , satisfazendo (81), introduz-se o seguinte sinal de erro não-linear

$$S = k\theta - 1 \quad , \quad (82)$$

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

$$\dot{V}(S) = SS' = S(\dot{k}\theta + k\dot{\theta}) \quad , \quad (83)$$

Em vista de (83), escolhe-se a seguinte lei de atualização paramétrica

$$\dot{\theta} = -\beta S k \quad , \quad (84)$$

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

$$\dot{V}(S) = S\dot{k}\theta - \beta S^2 k^2 \quad , \quad (85)$$

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  $k \equiv 0$ . Neste caso,  $\dot{V}(S) \leq 0$  e a análise é consideravelmente simplificada. A lei de atualização (84) pode ser reescrita como

$$\dot{\theta} = -\beta(k\theta - 1)k = -\beta k^2 \theta + \beta k \quad , \quad (86)$$

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

$$(s + \beta k^2)\theta = \beta k \quad , \quad (87)$$

Para  $k \neq 0$ , usando a transformada de Laplace pode-se explicitar  $\theta$  como

$$\Theta = \frac{\beta k}{s + \beta k^2} = \frac{1}{\tau s + 1} \frac{1}{k} \quad , \quad (88)$$

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

## d Inversa Filtrada em manipuladores paralelos

O algoritmo da inversa filtrada pode ser generalizado para inverter matrizes. Considere a seguinte planta MIMO de 1ª ordem descrita por

$$\dot{y} = K(t)u \quad , \quad (89)$$

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

$$u = K^\dagger(t)v, \quad v = \dot{y}_d + \Lambda e \quad , \quad (90)$$

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

$$S_r = K\Theta - I \in \mathbb{R}^{m \times m} \quad , \quad (91)$$

e à esquerda

$$S_l = \Theta K - I \in \mathbb{R}^{n \times n} \quad , \quad (92)$$

Para casos não-redundantes ( $m = n$ ) e com posto completo, as leis de atualização 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  $S_r$  e  $S_l$  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  $2V_c = 2V_r + 2V_l$ , com derivada temporal dada por

$$2\dot{V}_c = 2g(\dot{K}) + 2tr(S_r^T K \dot{\Theta}) + 2tr(S_l^T \dot{\Theta} K) \quad , \quad (93)$$

onde  $g(\dot{K}) = tr(S_l^T \dot{\Theta} K) + tr(S_r^T K \dot{\Theta})$ . Assim, de (93) tem-se

$$\dot{V}_c = g(\dot{K}) + tr((S_r^T K + K S_l^T) \dot{\Theta}) \quad , \quad (94)$$

Desta forma, a lei de atualização proposta é

$$\dot{\Theta} = -\Gamma(S_l K^T + K^T S_r) \quad , \quad (95)$$

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

$$\dot{V}_c = g(\dot{K}) - tr((S_r^T K + K S_l^T) \Gamma (S_l K^T + K^T S_r)) \quad , \quad (96)$$

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

$$S_l K^T + K^T S_r = 0 \quad , \quad (97)$$

De imediato, tem-se que a ocorrência simultânea das condições  $K^T S_r = 0$  e  $S_l K^T = 0$  apresentadas anteriormente implica em  $\dot{\Theta} = 0$ . A prova no sentido contrário é obtida a seguir. Assumindo (97), tem-se que

$$(S_l K^T + K^T S_r) K S_l^T = 0 \quad , \quad (98)$$

## e Modelagem e controle cinemático de manipuladores paralelos

Considere o par  $\{p, R\}$  como sendo o vetor posição e a matriz de orientação do quadro  $F_e$  do efetuador em respeito ao quadro da base  $F_b$ . A pose do efetuador pode ser obtida pelo mapeamento da cinemática direta como:

$$\{p, R\} = k(\theta) \quad , \quad (99)$$

Onde  $\theta \in \mathbb{R}^n$  é o vetor de posição das juntas e  $k(\cdot) : \mathbb{R}^n \rightarrow \{R^3, SO(3)\}$  é um mapeamento não linear entre as juntas e o espaço cartesiano. Agora, fazendo com que  $\theta_a \in \mathbb{R}^{n_a}$  seja o vetor de posição das juntas atuadas ou ativas e  $\theta_p \in \mathbb{R}^{n_p}$  como o vetor posição das juntas não atuadas ou passivas, onde  $n = n_a + n_p$ . Então, é possível rearranjar o vetor posição das juntas como  $\theta = [\theta_a^T \quad \theta_p^T]^T$ .

As restrições cinemáticas podem ser localmente representadas em termos de limites algébricos expressos no espaço das juntas como:

$$c(\theta) = 0 \quad , \quad (100)$$

De maneira que  $c(\cdot) : \mathbb{R}^n \rightarrow \mathbb{R}^r$ . Note que o mecanismo paralelo tem  $n_e := (n - r)$  graus de liberdade efetivos e assumindo que o número das juntas ativas é igual ao número efetivo de graus de liberdade, que é,  $n_a = n_e$ ; o número de juntas passivas é igual ao número de restrições, onde  $n_p = r$ . Agora, considerando que as restrições algébricas (100) possam ser escritas em termos de vetor de velocidade das juntas  $\dot{\theta} \in \mathbb{R}^n$  como:

$$J_c(\theta)\dot{\theta} = 0 \quad , \quad (101)$$

Onde  $J_c \in \mathbb{R}^{r \times n}$  é a matriz jacobiana de restrição.

O vetor de velocidades do efetuador  $v \in \mathbb{R}^6$  pode ser relacionado ao vetor de velocidades das juntas  $\dot{\theta} \in \mathbb{R}^n$  conforme a equação de cinemática diferencial:

$$v = \begin{bmatrix} \dot{p} \\ \omega \end{bmatrix} = J_m(\theta)\dot{\theta} \quad , \quad (102)$$

Onde  $\dot{p} \in \mathbb{R}^3$  é o vetor de velocidade linear e  $\omega \in \mathbb{R}^3$  é o vetor de velocidade angular que são definidos como  $R = \omega \times R$  e  $J_m \in \mathbb{R}^{m \times n}$  é a matriz jacobiana do manipulador [68]

É possível separar as matrizes  $J_c$  e  $J_m$  de acordo com as variáveis das juntas ativas e passivas  $\theta_a$  e  $\theta_p$  como  $J_c = [J_{ca} \quad J_{cp}]$  e  $J_m = [J_{ma} \quad J_{mp}]$ . E ainda sem perda de generalidade é possível reescrever (101) e (102) como:

$$0 = J_{ca}\dot{\theta}_a + J_{cp}\dot{\theta}_p \quad , \quad (103)$$

$$v = J_{ma}\dot{\theta}_a + J_{mp}\dot{\theta}_p \quad , \quad (104)$$

Onde  $J_{ca} \in \mathbb{R}^{n_p \times n_a}$ ,  $J_{cp} \in \mathbb{R}^{n_p \times n_p}$ ,  $J_{ma} \in \mathbb{R}^{m \times n_a}$  e  $J_{mp} \in \mathbb{R}^{m \times n_p}$ . A partir de (104) é possível calcular  $\dot{\theta}_p$  em termos da velocidade das juntas ativas  $\dot{\theta}_a$ :

$$\dot{\theta}_p = -J_{cp}^{-1}J_{ca}\dot{\theta}_a \quad , \quad (105)$$

Onde assume-se que  $J_{cp}$  é invertível. E substituindo (105) em (104) é possível reescrever a equação de cinemática diferencial como:

$$v = (J_{ma} - J_{mp}J_{cp}^{-1}J_{ca})\dot{\theta}_a = \tilde{J}(\theta)\dot{\theta}_a \quad , \quad (106)$$

Onde  $\tilde{J} \in \mathbb{R}^{m \times n_e}$  também é chamada de matriz jacobiana modificada.

O controle da pose em manipuladores paralelos pode ser feito por controle cinemático a nível de velocidades dada a simplicidade de implementação e desempenho satisfatório quando uma tarefa dada não requer movimentos muito rápidos das juntas. Neste contexto é possível obter o sistema de controle:

$$v = \tilde{J}(\theta)u, \quad u := \dot{\theta}_a \quad , \quad (107)$$

Onde  $u \in \mathbb{R}^n$  é o sinal de controle de velocidade aplicado aos atuadores das juntas correspondentes. Note que se  $\tilde{J}$  possui rank completo, o sinal de controle de velocidade das juntas é semelhante ao caso de controle cinemático de manipuladores seriais.

Agora supondo que o objetivo do controle seja rastreamento, a posição desejada  $p_d(t)$  da posição atual  $p$  do efetuador assegura a convergência assintótica do erro  $e_p := p_d - p$  para zero. Uma lei de controle proporcional associada a feedforward baseada na pseudo-inversa da matriz jacobiana pode ser proposto como:

$$u = \tilde{J}_p^+(\theta)v_p, \quad v_p = \dot{p}_d + \Lambda_p e_p, \quad (108)$$

Onde  $\tilde{J}_p \in \mathbb{R}^{3n_e}$  denota a parte posição na matriz jacobiana modificada e  $\Lambda_p = \Lambda_p^T > 0$  é a matriz de ganho de posição.

Para o caso de controle de orientação  $R \in SO(3)$  deverá ser considerado uma unidade quaternion que seja livre de singularidades e computacionalmente mais eficiente.

Conclui-se, portanto que as configurações singulares em manipuladores paralelos ocorrerão quando alguma das matrizes ou ambas as matrizes  $J_{cp}, \tilde{J}_p$  possuírem rank incompleto ou se tornem numericamente não invertíveis. A análise intuitiva conduzida pelo arranjo das matrizes:

$$\begin{bmatrix} 0 & J_{ca} & J_{cp} \\ -I & \tilde{J}_p & 0 \end{bmatrix} \begin{bmatrix} \dot{p} \\ \dot{\theta}_a \\ \dot{\theta}_p \end{bmatrix} = 0, \quad (109)$$

Se o rank da matriz  $J_{cp}$  cair implica que  $\det(J_{cp}) = 0$  e neste caso existirão velocidades não nulas no efetuador  $\dot{p}$  cujas quais as velocidades nas juntas ativas  $\dot{\theta}_a = 0$ , chamadas de singularidades paralelas. Por outro lado se o rank da matriz  $\tilde{J}_p$  cair implica que  $\det(\tilde{J}_p) = 0$  e neste caso existirão velocidades não nulas nas juntas ativas  $\dot{\theta}_a$  para velocidades nulas no efetuador  $\dot{p} \neq 0$ . E finalmente se os ranks de ambas as matrizes  $J_{cp}$  e  $\tilde{J}_p$  ou se tornarem não invertíveis simultaneamente haverá velocidades não nulas nas juntas passivas  $\dot{\theta}_p$  para velocidades nulas das juntas ativas e no efetuador  $\dot{\theta}_a = 0$  e  $\dot{p} = 0$  ao mesmo tempo conduzindo a uma configuração de singularidade.

## 5 Simulações preliminares

### a Ferramentas computacionais

Em um programa de simulação é possível criar ambientes virtuais para testar a sequência de comandos programados para os robôs, para somente depois aplicá-los no robô real.

Uma grande parte das universidades de engenharia de todo o mundo utilizam como software de simulação e análise de controle de sistemas o MATLAB. Dentro deste programa existem os chamados toolboxes (ferramentas) que são funções diferentes das básicas, originalmente presentes no programa. Estas funções estão disponíveis em bibliotecas externas, podendo ser constantemente atualizadas à medida que forem desenvolvidas novas aplicações. Neste software há um toolbox especialmente destinado à simulação de manipuladores robóticos, chamada Robotics Toolbox for Matlab (ferramentas robóticas para o Matlab).

Esta ferramenta fornece funções em que é abordada uma série de questões relativas aos robôs, como por exemplo, cinemática, dinâmica e geração de trajetória. Esta ferramenta é extremamente útil para simulação e análise de resultados a partir de experimentos virtuais, baseados em modelos de manipuladores reais. Muitas indústrias têm implantado o programa em sua linha de produção (MATHWORKS, 2011).

MATLAB (abreviação de Matrix Laboratory - laboratório de matrizes) é um programa de computador desenvolvido na década de 80 por Cleve Moler, no Departamento de Ciência da Computação da Universidade de Novo México, Estados Unidos. Foi projetado inicialmente para efetuar cálculos científicos de engenharia, otimizando o tempo dos usuários. Seu conceito inicial foi baseado em operações matriciais (daí a origem de seu nome), porém com o passar dos anos foi se tornando um sistema flexível, através da implantação de novas funções que tornam possível a resolução de qualquer problema técnico (CHAPMAN, 2003).

O software implementa uma linguagem de programação: MATLAB, com funções predefinidas e que admite ferramentas adicionais com as mais diversas finalidades. A unidade fundamental de dados neste programa é a matriz: uma coleção de dados organizados em linhas e colunas e que podem ser utilizados mediante a utilização de um nome específico. Uma variável em MATLAB é uma região utilizada pela memória para o conteúdo de uma matriz específica (CHAPMAN, 2003).

Conforme mencionado no início deste capítulo, o software tem a sua versão básica, composta de um pacote de funções matemáticas necessárias à resolução de qualquer problema técnico. Para a solução de questões mais complexas, como processamento de sinais em tempo real ou simulação e controle de robôs, é necessário à obtenção de uma ferramenta específica a esta função (CHAPMAN, 2003).

No caso deste trabalho a ferramenta utilizada para controle e simulação de robô é a Robotic Toolbox for MATLAB (Ferramenta Robótica para MATLAB) e foi desenvolvida pelo Phd. australiano em engenharia mecânica, da Universidade de Queensland, Peter Corcke.

Esta ferramenta possui um conjunto de funções que permite operar e simular os robôs obtidos a partir do Matlab. Entre as principais atribuições deste aplicativo, destacam-se a cinemática, dinâmica e geração de trajetória de um robô. Além disto, ela possui funções necessárias à manipulação de dados, (como por exemplo, transformação homogênea), necessários à representação dos manipuladores em três dimensões. Além disso, possui os modelos de dois manipuladores prontos: PUMA 560 e Stanford Arm [71]

### b Simulação - Cinemática Direta

Para resolver o problema da cinemática direta o MATLAB cria, (com base na notação de DH), uma matriz  $T$  chamada matriz de transformação homogênea, onde cada linha da matriz relaciona um elo do manipulador e suas colunas são representadas conforme as seguintes funções: [70].

Coluna 1:  $\alpha_i$  = ângulo de rotação da junta.

Coluna 2:  $A_i$  = deslocamento da junta.

Coluna 3:  $\theta_i$  = ângulo de rotação da junta.

Coluna 4:  $D_i$  = comprimento da junta.

Coluna 5:  $\sigma_i$  = tipo da junta (0 se for prismática ou 1 de revolução).

A partir dos valores do manipulador e seus parâmetros de DH, sua matriz de transformação homogênea, (com base no último elo do manipulador em sua posição inicial), descrita pelo vetor  $q$  é obtido por exemplo:

$$q = [0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0] \quad , \quad (110)$$

$$T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0.4170 \\ 0 & -1 & 0 & 0.4170 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad , \quad (111)$$

As primeiras três colunas da matriz correspondem à rotação, enquanto que a última coluna da matriz corresponde à translação do manipulador em torno das coordenadas fornecidas pelo segundo argumento da função *fkine* [71].

A relevância deste conceito está no fato de que a partir de um ponto qualquer dado por suas coordenadas angulares (sempre em radianos), o operador pode descobrir a posição do elo final do manipulador. Ou seja, a movimentação do manipulador é feita de forma gradativa e minuciosa, movimentos são executados através do incremento do valor exato do ângulo relativo à uma determinada junta [71].

Modificando o conjunto de coordenadas angulares dado por  $q$ , obtém-se uma nova matriz de transformação homogênea  $T$  e uma nova configuração geométrica para o manipulador.

E após isto é efetuada uma nova simulação, a partir de um novo vetor  $q$ ,

$$q = [0 \quad -\pi/4 \quad -\pi/4 \quad 0 \quad \pi/8 \quad 0] \quad , \quad (112)$$

$$T = \begin{bmatrix} 0.6533 & 0.7071 & 0.2706 & 0.6265 \\ 0.3827 & 0.0000 & 0.9239 & 0.3970 \\ 0.6533 & 0.7071 & 0.2706 & -0.0722 \\ 0.0000 & 0.0000 & 0.0000 & 1.0000 \end{bmatrix} \quad , \quad (113)$$

## c Simulação - Cinemática Inversa

Para uma posição e orientação do efetuador final, determinam-se quais as posições das juntas que satisfazem à posição do mesmo. Trata-se de uma questão mais complexa que a cinemática direta, já que o algoritmo de solução é diferente para cada robô por apresentar diferença nos graus de liberdade.

Supondo um vetor de posições angulares:

$$q = [0 \quad \pi/12 \quad \pi/8 \quad 0 \quad \pi/6 \quad 0] \quad , \quad (114)$$

Neste caso, sua matriz de transformação homogênea será:

$$T = \begin{bmatrix} 0.8365 & 0.2588 & 0.4830 & 0.2287 \\ 0.5000 & 0.0000 & 0.8660 & 0.3818 \\ 0.2241 & 0.9659 & 0.1294 & 0.7573 \\ 0.0000 & 0.0000 & 0.0000 & 1.0000 \end{bmatrix} \quad , \quad (115)$$

Aplicando a função *ikine*, o vetor encontrado será, em radianos,

$$q_i = [0.0000 \quad 0.2618 \quad 0.3927 \quad 0.0000 \quad 0.5236 \quad 0.0000] \quad , \quad (116)$$



exatamente igual a  $q$  (114).

Esta mesma solução pode ser obtida através da solução de complexas equações diferenciais. Além de ter uma solução para cada robô (em decorrência de diferentes robôs terem diferentes graus de liberdade), também compromete a escolha de valor inicial, bem como o tempo gasto pelo projetista em buscar a solução ideal, levando em conta todas estas questões. E como último dificultador, um mesmo manipulador pode ter várias posições que resultem na mesma matriz de transformação homogênea para a última junta. Para a questão da cinemática inversa, nem mesmo a escolha do zero como ponto de partida do robô é uma boa escolha, pois ele limita o controle sobre a solução particular da equação diferencial.

Uma solução não será possível se a matriz de transformação homogênea descrever um ponto fora do alcance do manipulador (Singularidade). Esta toolbox oferece, em termos da cinemática inversa, apenas uma descrição generalizada do robô, em termos de seus parâmetros cinemáticos.

Visando eliminar as indeterminações causadas pela cinemática inversa utiliza-se o conceito da matriz jacobiana, que serve para relacionar as velocidades das coordenadas articulares às posições e orientações do efetuador no robô, controlando assim a cinemática dos mecanismos. Em outras palavras ela calcula as velocidades necessárias nas juntas (no plano cartesiano) para que o efetuador alcance uma determinada velocidade [16].

Para uma configuração inicial, a partir de pequenos deslocamentos associados às articulações calcula-se esta matriz a partir das relações cinemáticas da arquitetura do manipulador. Para estas coordenadas de seus efetuadores são descritas pelo vetor de posição  $q$  [16]. A partir das velocidades de cada articulação, com o cálculo da matriz jacobiana inversa obtém-se a velocidade de cada uma das articulações.

Se a singularidade do manipulador (fornecida pelo determinante da matriz jacobiana  $J$ ) for igual à zero, é impossível controlar a velocidade da garra do robô (neste caso, a matriz não admite inversão) [16].

Quando a determinante da matriz jacobiana é nula (logo ela não é inversível e consequentemente a velocidade do manipulador não pode ser controlada), esta servirá apenas para informar o quão bem posicionado o manipulador está para efetuar determinados movimentos. Esta é a função *maniply* (manipulabilidade), e baseia-se nos parâmetros cinemáticos. Como ela apenas informa o quão ‘bem posicionado’ o manipulador está o argumento de entrada é uma posição qualquer fornecida pelo operador, e seu retorno determinará o quão longe o efetuador está da posição desejada [70].

A toolbox do MATLAB simula, de acordo com as matrizes de transformação homogêneas, a trajetória percorrida, a variação de velocidade e aceleração, descritas por um manipulador no plano cartesiano, a partir de um ângulo zero de acordo com o intervalo de tempo fornecido por um vetor  $t$  [70].

#### d Simulação - Manipulador serial

Para familiarização com as ferramentas computacionais e características particulares dos algoritmos de controle para o estudo de casos em manipuladores seriais, torna-se necessário reproduzir resultados já obtidos na literatura, como implementar o controle cinemático em manipuladores seriais, sobretudo aqueles que possuam a dimensão do espaço das juntas diferente da dimensão do espaço de trabalho resultando em uma matriz jacobiana retangular.

As simulações iniciais de controle de manipuladores seriais em configurações de singularidade cinemática foram realizadas para diversos arranjos estruturais e para diferentes graus de liberdade. A seguir é apresentada a simulação de controle de um manipulador serial composto por dois links, seis juntas ativas de revolução que pode ser plotado utilizando o comando *robo.plot*.

Em um primeiro momento de simulações procura-se determinar a estabilidade do sistema e a convergência do erro de posição para zero, ainda sem lidar com problemas de singularidade, ou seja o objetivo inicial é realizar a simulação de controle dentro do espaço de trabalho do efetuador para levar da posição inicial das juntas à posição desejada na trajetória utilizando a metodologia clássica de controle onde a matriz jacobiana é retangular e sua inversa substituída pela pseudo-inversa. Obtemos:

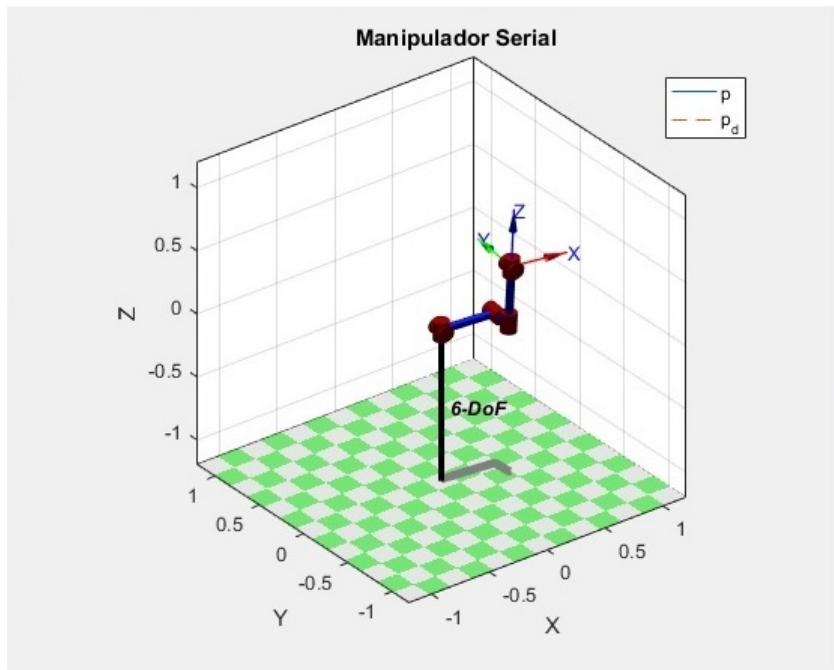


Figura 9: Manipulador serial 6 DoF.

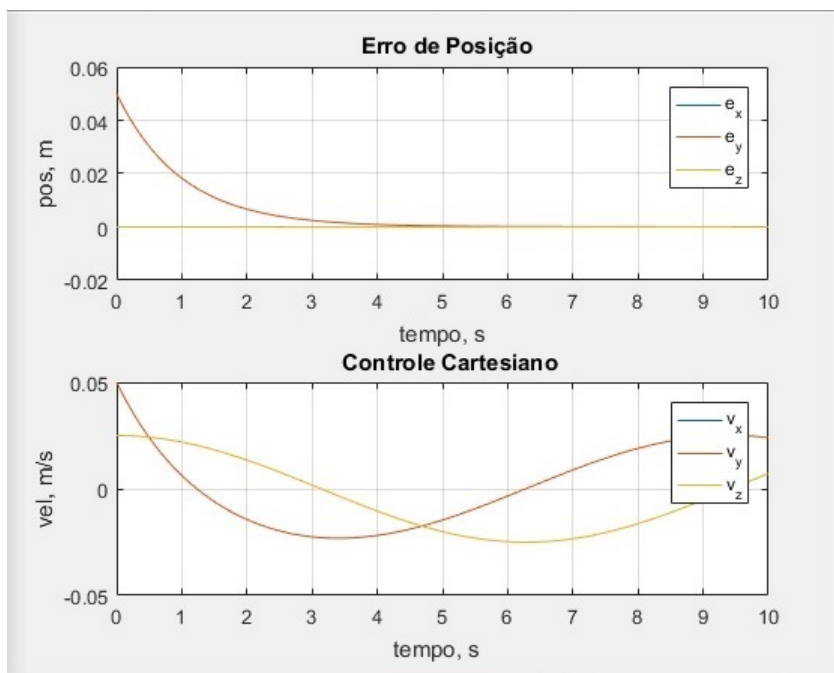


Figura 10: Erro de posição e velocidades do efetuador

A ideia do controle em malha fechada deste modelo é utilizar o erro de posição no espaço de trabalho do efetuador como elemento da retroalimentação. Sobre ele é aplicado um ganho e inserido na entrada do sistema juntamente com a posição desejada para que sirva de ajuste para a convergência do erro.

Sabe-se que em ambiente de simulação o erro de posição não é resultado de um feedback de sensores no manipulador, mas como resultado de uma estimativa matemática, que neste caso, é obtido pela cinemática direta.

Não somente o espaço de trabalho do efetuador é monitorado, espera-se de igual modo um comportamento típico também no espaço das juntas:

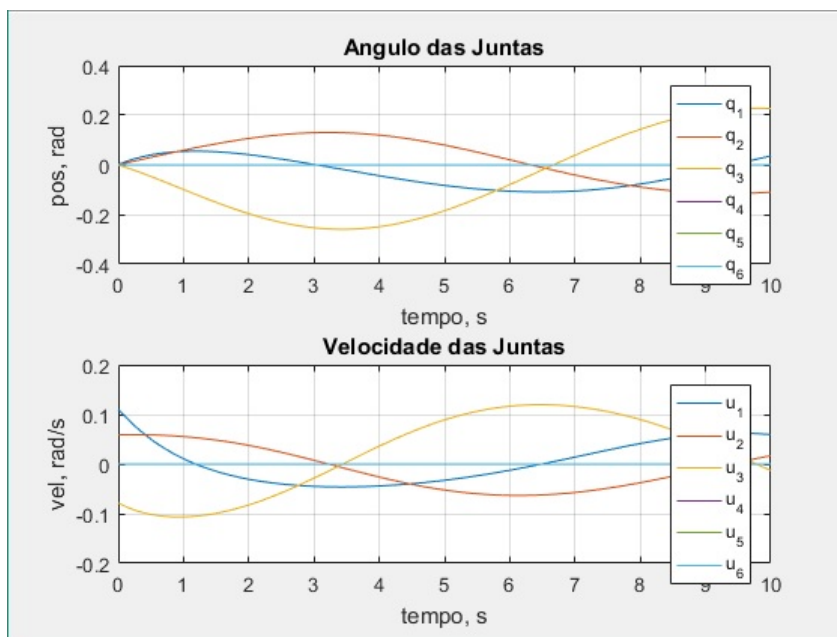


Figura 11: Posição angular e velocidades das juntas

### e Simulação - Controle de mecanismos seriais em configurações de singularidade

Para simplificação dos processos matemáticos a implementação de simulações dos algoritmos de controle de manipuladores seriais em condições de singularidade foi feita em robôs com um número reduzido de juntas, uma vez que a metodologia é a mesma.

Deste modo, um estudo comparativo dos algoritmos de controle foi feito em um manipulador contendo duas juntas de revolução, dois links, o primeiro link com 279.4 mm e o segundo com 228.6 mm modelados em série com uma margem de 10 por cento de incerteza nas dimensões. Esta configuração de manipulador é pertinente pois representa uma condição onde a dimensão do efetuador é maior que o número de juntas ativas e isso resulta em uma matriz jacobiana retangular, o que não é invertível. Como visto, pelo método clássico quando o tratamento inverso não é possível utiliza-se como recurso a pseudo inversa.

A trajetória proposta para este manipulador excede o espaço de trabalho do efetuador e explora a condição de singularidade de fronteira como visto nos capítulos anteriores.

Implementando inicialmente o algoritmo de controle DLS para lidar com a singularidade com  $\omega_0 = 10^3$  e  $\alpha_0 = 30$  o resultado é mostrado na Figura 12:

E paralelamente implementando o algoritmo de controle da inversa filtrada sob as mesmas condições, mas agora com um único parâmetro, o ganho de atualização dinâmica  $\gamma = 10^{-6}$  é mostrado na Figura 13:

É possível observar pela Figura 14 que a velocidade das juntas, que representa diretamente o sinal de controle nas juntas ativas, quando sob controle por DLS possui variações não factíveis do ponto de vista físico pois altíssimas velocidades instantâneas são solicitadas pelo controle. Ainda é possível perceber que quando o sinal de controle é determinado pelo algoritmo da inversa filtrada, variações suaves e razoavelmente factíveis são propostas uma vez que a velocidade das juntas não chegam 5rad/s

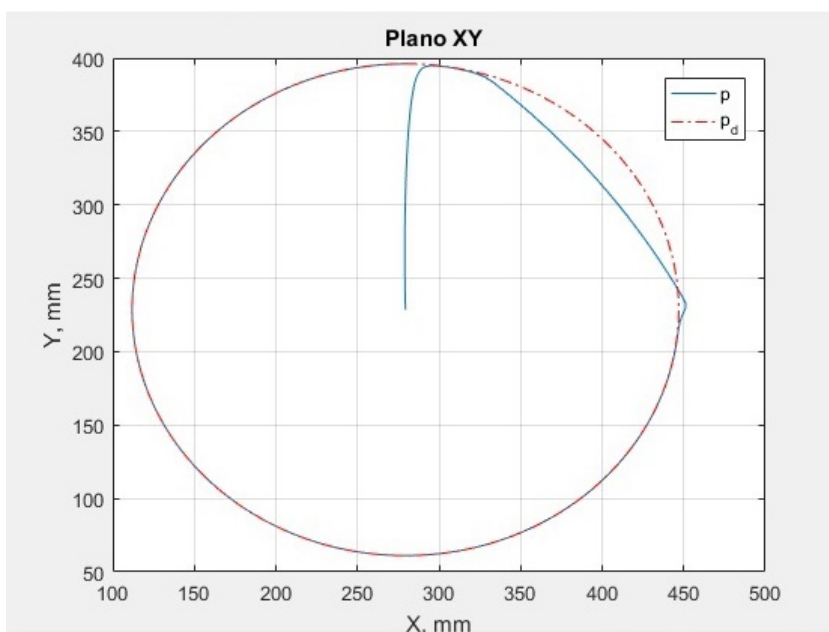


Figura 12: Trajetória em controle por DLS

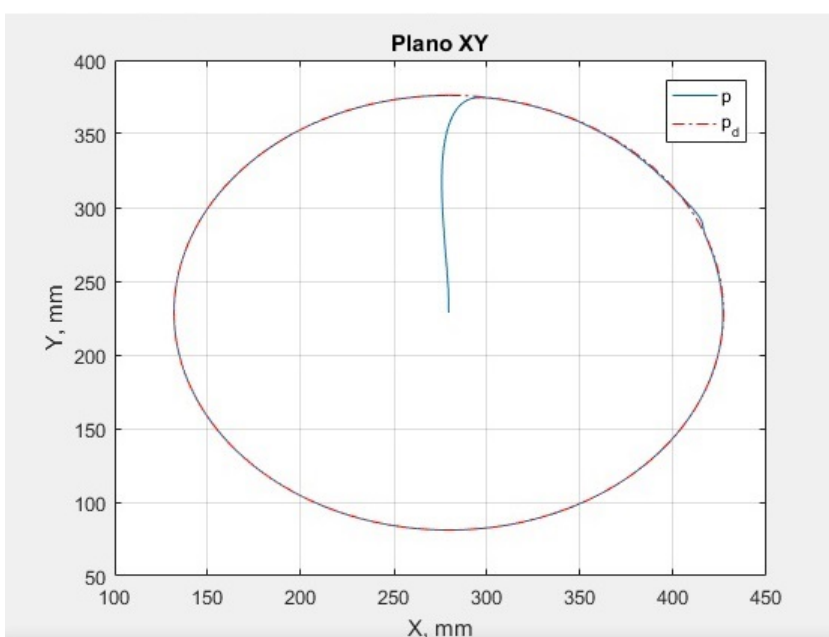


Figura 13: Trajetória em controle por Inversa Filtrada

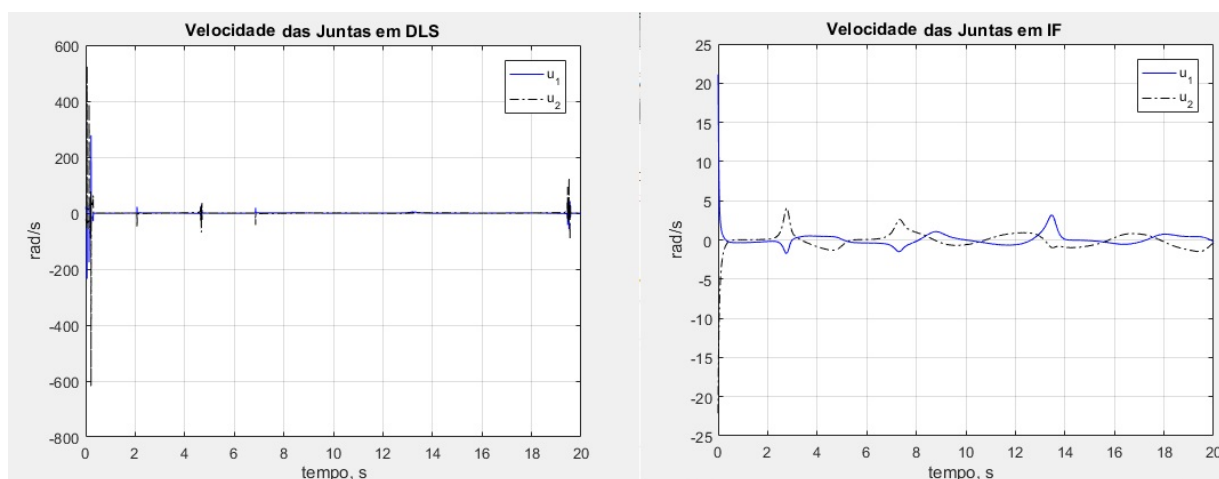


Figura 14: Comparativo do sinal de controle

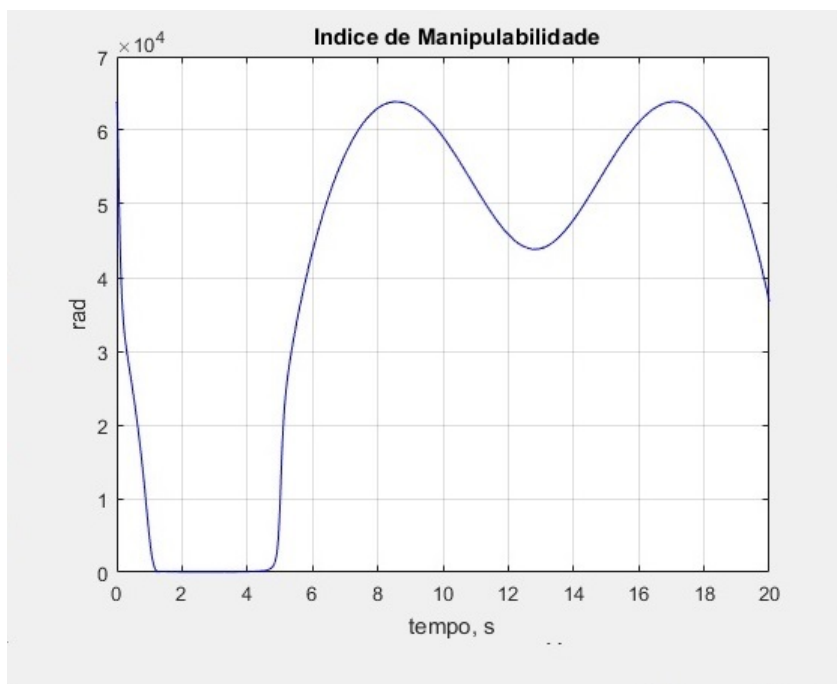


Figura 15: Manipulabilidade em DLS

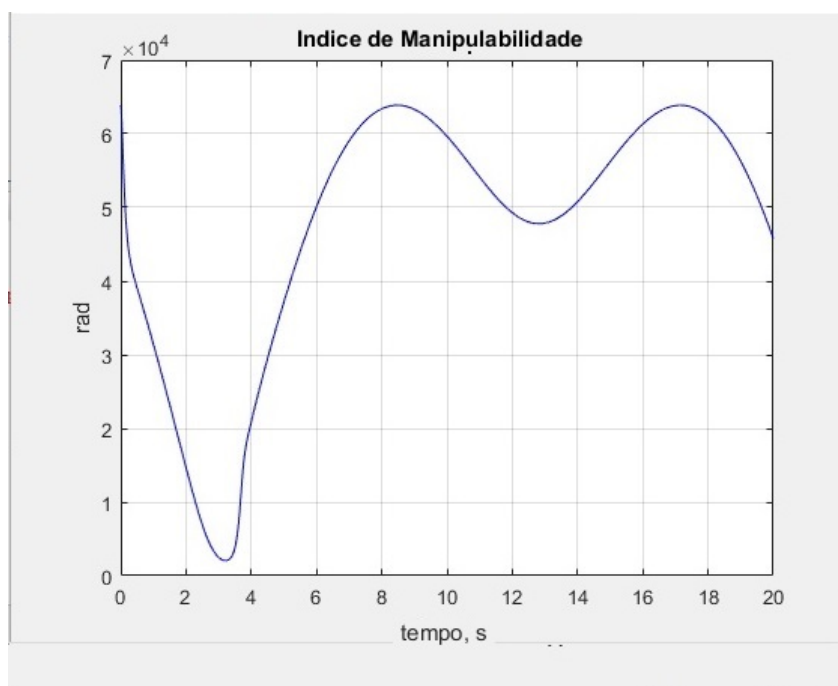


Figura 16: Manipulabilidade em Inversa Filtrada

## 6 Controle de mecanismos paralelos em configurações de singularidade

### a Manipulador paralelo Slider-Crank

A próxima etapa do estudo de casos fora implementar o controle por inversa filtrada em manipuladores paralelos, onde articuladores possuem juntas atuadas e não atuadas que representam o caso MIMO e, em tese, são aplicados dois algoritmos de inversa filtrada em que um é parte integrante do outro.

Fora escolhido para estudo de caso o manipulador paralelo composto inicialmente por uma única junta ativa e outra junta passiva. Onde efetuador está ligado à uma junta prismática ou carretinho que limita seu deslocamento em uma trajetória unidirecional, este modelo é conhecido como Slider Crank. Em uma configuração singular serial, as juntas podem ter velocidade não nula enquanto o efetuador está em posição de repouso, o que acontece quando o efetuador perde graus de liberdade no espaço de trabalho e a matriz Jacobiana perde posto e não é mais invertível. Entretanto em uma configuração singular de manipulador paralelo, podem existir velocidades não nulas do efetuador no espaço de trabalho quando as juntas ativas estão em posição de repouso. Neste caso o efetuador ganha graus de liberdade e o robô perde controlabilidade, e ainda, forças inercias podem levar a quebra da estrutura robótica.

Como fora detalhado, a modelagem clássica de um manipulador paralelo exige a inversão de duas matrizes, e por consequência de eventual perda de posto, a qual resulta na não invertibilidade de uma delas, temos por resultado uma configuração de singularidade cinemática. E fisicamente, a simples proximidade das configurações singulares seriam já suficientes para causar grandes danos na estrutura robótica.

Manipuladores paralelos são caracteristicamente mais precisos e suportam maior carga em comparação as estruturas seriais devido a distribuição das forças nas cadeias cinemáticas.

Do ponto de vista físico, se faz necessário para melhor compreensão das configurações de singularidade em manipuladores paralelos, uma análise geométrica que relacione o problema matemático aos efeitos estruturais do manipulador.

As singularidades em mecanismos paralelos podem estar relacionadas ao efetuador, ao atuador ou singularidade conjunta.

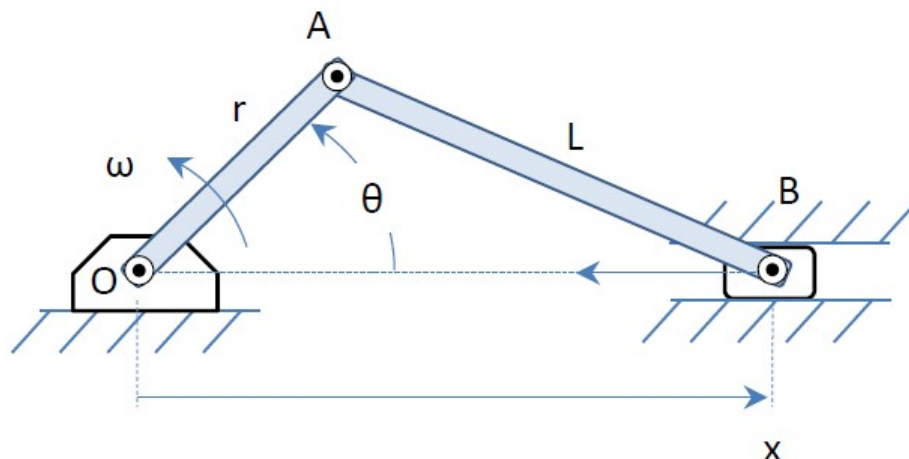


Figura 17: Manipulador paralelo Slider Crank

Um Slider Crank que possui o link  $r > L$  terá uma posição onde a estrutura poderá configurar-se semelhante a um triângulo retângulo. Neste ponto limite a junta ativa terá chegado ao seu limite máximo de abertura angular e, ao tentar retornar desta posição irá transferir pelo o link  $L$  uma força perpendicular ao eixo de deslizamento de modo que o efetuador  $B$  não poderá se deslocar em nenhum dos dois sentidos produzindo uma singularidade de efetuador conforme ilustrado a seguir:

Nesta simulação a trajetória é unidimensional, e para estudar as singularidades, fora proposto uma trajetória que passasse pelos pontos de singularidade se estendendo para fora do espaço de trabalho. E o objetivo na implementação do algoritmo de controle é evitar, não somente o ponto, mas também a proximidade das configurações singulares, de



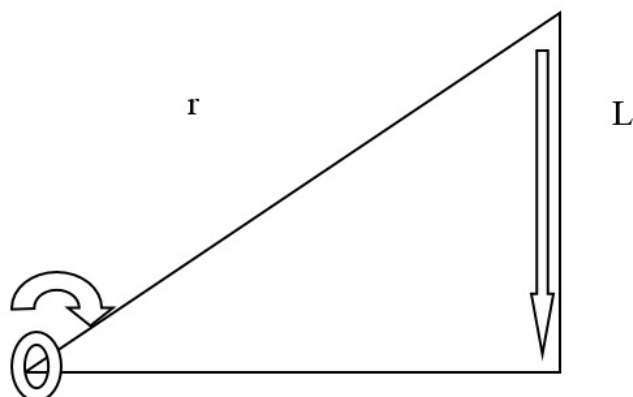


Figura 18: Dinâmica da singularidade no efetuador

modo que, quando a trajetória sugerir posição fora da área de trabalho do efetuador, seja proposto para configuração das juntas uma posição que permita ao efetuador estar o mais próximo da trajetória desejada quanto possível e ainda esteja confinado ao movimento entre as posições de configuração singular.

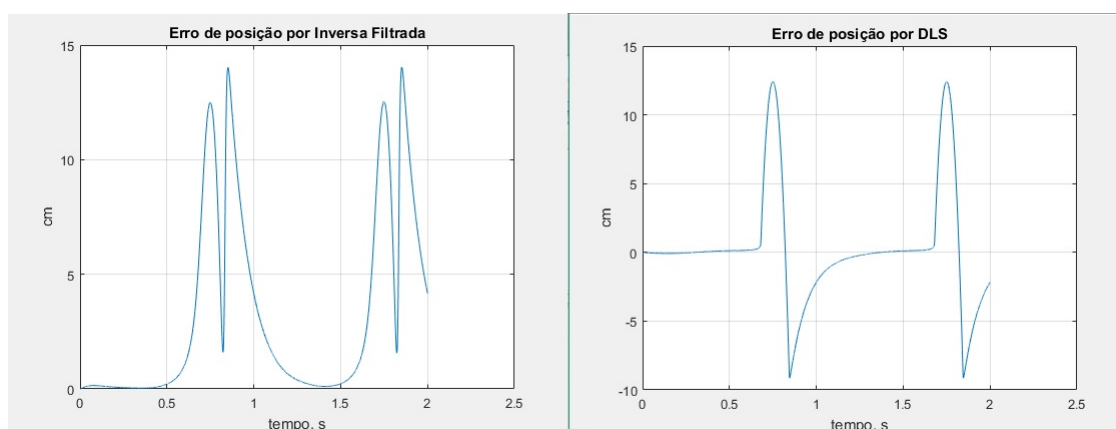


Figura 19: Comparativo do erro de trajetória em controle DLS e Inversa Filtrada

Nota-se que a análise destes gráficos não é trivial, eles representam o erro de posição no tempo, e uma vez que a trajetória é unidimensional e extrapola ao alcance da área de trabalho a presença do erro e sua composição é de fundamental importância para o estudo da eficiência dos algoritmos de controle.

O trajeto que pertence ao espaço de trabalho é observado quando o erro está próximo de zero. O erro é calculado pela diferença entre o cálculo de posição por cinemática direta (usando as soluções do controle para as configurações das juntas) e a trajetória desejada. Observe que o erro chega a ser negativo em controle por DLS, o que ocorre quando o efetuador passa pelo ponto de singularidade como se fosse uma função contínua, enquanto que em controle por inversa filtrada o erro se mantém sempre positivo, o que indica que o manipulador apenas se aproxima suavemente do ponto de singularidade como um limiar, mas não o alcança e permanece confinado na região aonde a manipulabilidade não é nula.

Como o manipulador Slider Crank é configurado por uma junta ativa e uma junta passiva, a modelagem exigia um controle de caráter escalar.

As dimensões dos links na simulação foram  $r = 212.13mm$  e  $L = 200mm$  de modo que resultasse em uma configuração de singularidade cinemática conforme detalhado anteriormente nesta seção.

De acordo com a modelagem de controle para manipuladores paralelos apresentada, a matriz jacobiana de restrição passiva é um elemento necessário e que deve manter-se invertível para evitar algumas configurações de singularidade. O Índice de manipulabilidade nos permite verificar o quão próximo seu determinante se aproxima de zero. O eixo abscisso representa a condição limite onde a perda da manipulabilidade é causada pela não invertibilidade da matriz



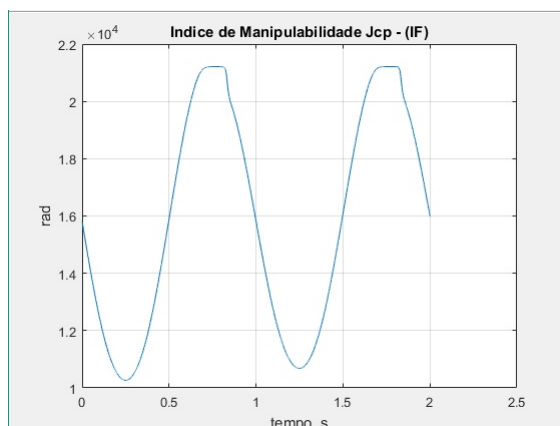


Figura 20: Índice de manipulabilidade da matriz jacobiana de restrição das juntas passivas (IF).

jacobiana sendo  $\det(J_{cp}) = 0$ .

A Figura 20 mostra que utilizando o algoritmo da Inversa Filtrada na simulação o jacobiano de restrição das juntas passivas se manteve estável durante a realização da trajetória uma vez que não toca o eixo abissal.

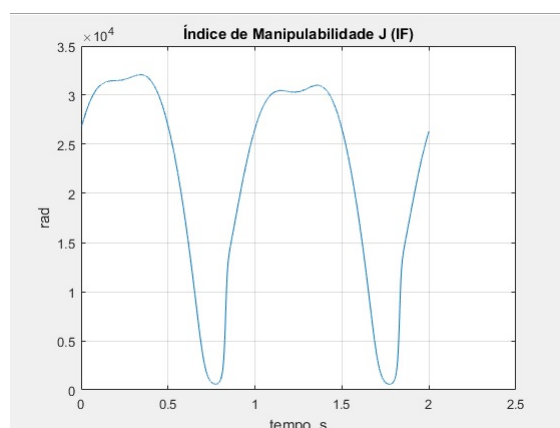


Figura 21: Índice de manipulabilidade da matriz jacobiana do manipulador (IF).

Paralelamente o Jacobiano do manipulador, que determina a relação de transformação dos espaços da juntas ao espaço do efetuador é fundamentalmente o que determina a habilidade de controle do mecanismo propriamente dito, e sua inversa deve necessariamente existir para que haja controle.

Observa-se pela Figura 21 que semelhantemente ao Jacobiano de restrição, a existência da sua inversa é garantida uma vez que  $\det(J) \neq 0$ .

A figura 22 mostra que do mesmo modo que utilizando o algoritmo da Inversa Filtrada foi possível garantir o índice de manipulabilidade não nulo, o algoritmo DLS obteve resultado equiparável.

É possível observar pela Figura 23 que em alguns trechos da trajetória a matriz jacobiana do manipulador possui índice de manipulabilidade nula o que resulta na perda de controle e consequentemente singularidade cinemática.

Nesta etapa fora realizado um estudo comparativo quanto a robustez dos algoritmos de controle quanto a capacidade de evitar configurações de singularidade (singularities avoidance).

Os testes de desempenho mostraram que o algoritmo DLS consumiu 2.9 segundos de processamento enquanto o algoritmo da Inversa Filtrada consumiu 4.2 segundos. Isto implica que nestas simulações o algoritmo da Inversa Filtrada exigiu aproximadamente 45% a mais de esforço computacional em comparação ao algoritmo DLS.

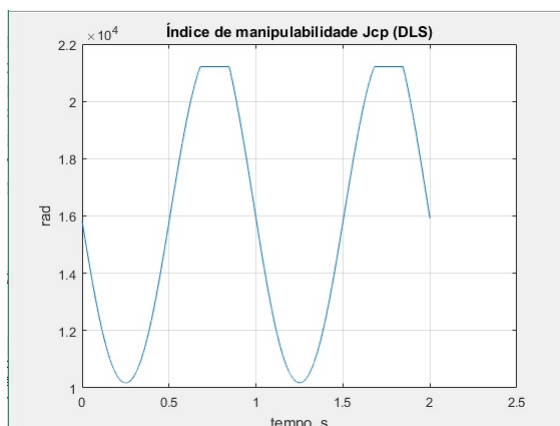


Figura 22: Índice de manipulabilidade da matriz jacobiana de restrição das juntas passivas (DLS).

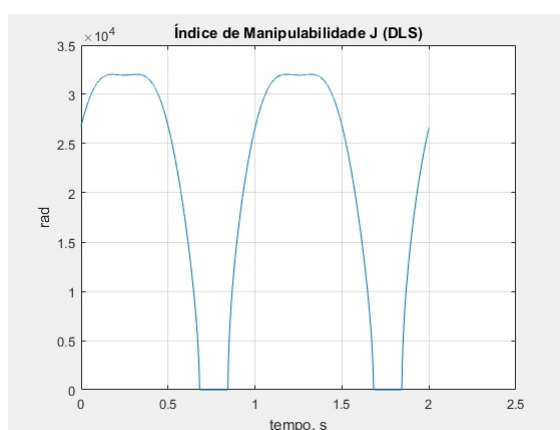


Figura 23: Índice de manipulabilidade da matriz jacobiana do manipulador (DLS).

## b Simulações - Slider Crank em Simulink

Simulink foi desenvolvido pela companhia MathWorks e é uma ferramenta para modelagem, simulação e análise de sistemas dinâmicos. Sua interface primária é uma ferramenta de diagramação gráfica por blocos e bibliotecas customizáveis de blocos. O software oferece alta integração com o resto do ambiente MATLAB. Simulink é amplamente usado em teoria de controle e processamento digital de sinais para projeto e simulação multi-domínios.

O toolbox de simulink utilizado para realizar simulações do comportamento estrutural do mecanismo paralelo slider crank foi o Simscape Multibody.

Simscape Multibody <sup>TM</sup> (anteriormente SimMechanics <sup>TM</sup>) fornece um ambiente de simulação multi-corpo para sistemas mecânicos 3D, como robôs, suspensões de veículos, equipamentos de construção e trem de pouso de aeronaves. Por ele é possível modelar sistemas de múltiplos corpos usando blocos que representam corpos, juntas, restrições, elementos de força e sensores. O Simscape Multibody formula e resolve as equações de movimento para o sistema mecânico completo.

Este pacote é capaz de importar montagens de CAD completas, incluindo todas as massas, inércias, juntas, restrições e geometria 3D. Uma animação 3D gerada automaticamente permite visualizar a dinâmica do sistema.

O pacote Simscape Multibody permite desenvolver sistemas de controle e testar o desempenho do nível do sistema. Parametrizar modelos usando MATLAB, variáveis e expressões, e sistemas de controle de design para o sistema de vários corpos em Simulink.

Por ele é possível integrar sistemas hidráulicos, elétricos, pneumáticos e outros sistemas físicos no modelo usando componentes do pacote Simscape.

Utilizando o pacote Simscape foi possível modelar a estrutura paralela Slider Crank, capturar o sinal de controle produzido pela simulação numérica e prever o movimento resultante da estrutura robótica. Deste modo, visualmente foi determinado o quão próximo o mecanismo estaria fisicamente das configurações de singularidades cinemáticas em cada caso simulado e qualquer algoritmo de controle, uma vez que a entrada de dados dos blocos do Simscape depende única e exclusivamente do sinal de controle no tempo.

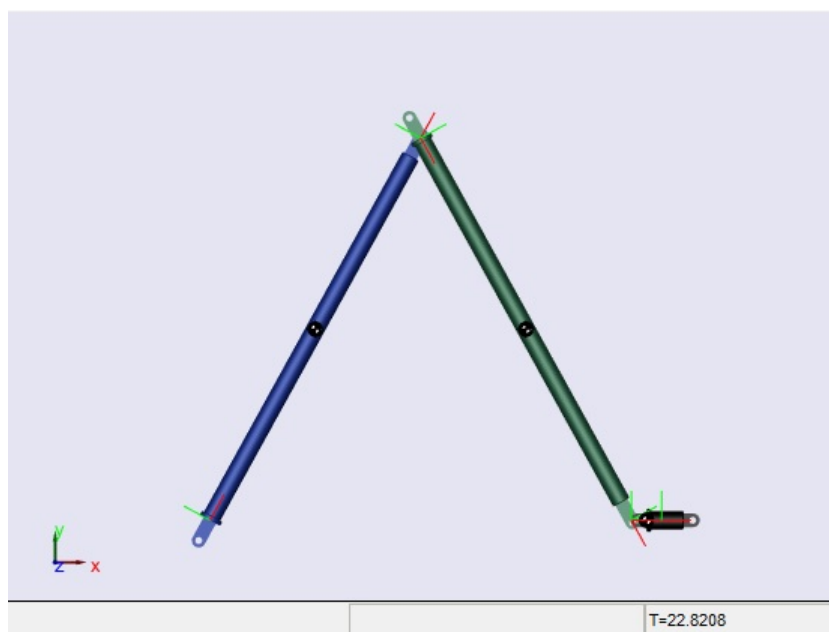


Figura 24: Simulação do mecanismo paralelo Slider Crank

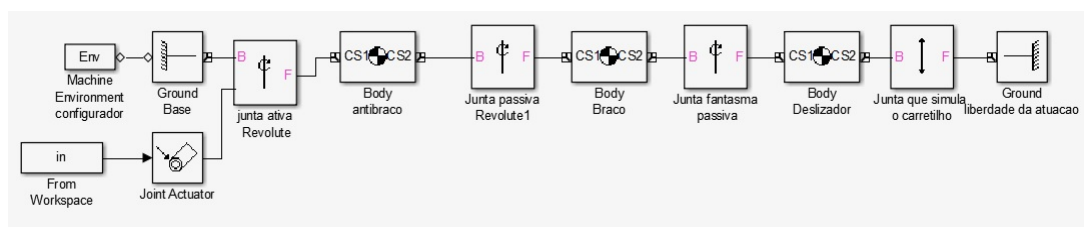


Figura 25: Modelagem de Blocos - Slider Crank em Simulink

## c Manipulador paralelo FourBar Linkage

Avançando no estudo de casos, fora modelado o manipulador paralelo conhecido como fourbar linkage composto por uma junta ativa e três juntas passivas, duas cadeias cinemáticas e susceptível à singularidade estrutural onde  $J_{cp}$  perde posto e simultaneamente pode existir movimento na junta passiva quando a junta ativa está em repouso. Pela cinemática direta é possível determinar as trajetórias possíveis:

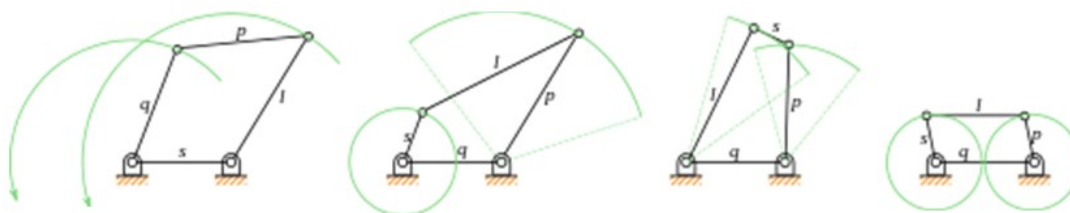


Figura 26: Modelos 26.a 26.b 26.c e 26.d para FourBar Linkage

Considerando o modelo 26.c observa-se que podem existir limites no alcance da trajetória que ocorrem quando o link  $I$  está alinhado ao link  $S$ . No caso do modelo 26.c a configuração limite com o alinhamento de  $I$  com  $S$  exige de todas as juntas, ativa e passivas, mudança de sentido, mas devido à perda de posto em  $J_{cp}$  a junta passiva entre  $I$  e  $S$  pode permanecer parada no caso da junta ativa estar entre  $p$  e  $q$ , e isso travaria a estrutura. Se a junta entre  $I$  e  $S$  simplesmente continuar no mesmo sentido devido a inércia, causaria singularidade estrutural e colapso. Equivalentemente, o mesmo efeito se daria no modelo 26.c se tivéssemos  $S$  alinhado à  $P$  com junta ativa entre  $q$  e  $I$  revertendo para sentido horário.

O objetivo neste estudo de caso é modelar com sucesso um manipulador four bar linkage semelhante ao modelo 26.c, propor uma trajetória que exceda o espaço de trabalho passando por configurações singulares e efetuar o controle cinemático por inversa filtrada, aplicando o enfoque matricial de maneira que se limite o movimento do efetuador à uma região tão longe dos pontos de singularidade quanto se determine pelo parâmetro de atualização dinâmica do algoritmo de controle estudado.

As dimensões dos links no fourbar 26.c simulado foram respectivamente  $I = 10\text{cm}$ ,  $S = 2\text{cm}$ ,  $p = 8\text{cm}$  e  $q = 5\text{cm}$  com posição inicial do vetor das juntas:

$$q_0 = [0 \quad 0 \quad 0 \quad 0] \quad , \quad (117)$$

Observe que esta posição inicial para a configuração das juntas é completamente insolúvel para a cinemática direta dada a estrutura do fourbar que forma uma cadeia fechada, é impossível fisicamente que todas as juntas estejam em posição angular 0 simultaneamente.

O parâmetro único do algoritmo da inversa filtrada que representa o ganho atualização dinâmica agora é duplo, como visto anteriormente, um para  $J_{cp}$  e outro para  $J_m$  que nesta simulação ambos foram escolhidos como  $10^{-2}$ , mas que poderiam ser ajustados independentemente. A frequência angular  $w = 1.2$  e a trajetória proposta será uma elipse  $x_d = w * 8 * \sin(w * t)$  e  $y_d = 2 * \cos(w * t)$  o vetor tempo limita-se  $0 < t < 20$  ao passo  $10^{-3}$  de modo que a matriz homogênea será dada por:

$$J = \begin{bmatrix} 0 & I * \cos(q(1)) & 0 & 0 \\ q & q + I * \sin(q(1)) & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & S & S + p * \cos(q(3)) \\ 0 & 0 & -q & -q + p * \sin(q(3)) \\ 0 & 0 & 1 & 1 \end{bmatrix} \quad , \quad (118)$$

$J$  está na forma de uma matriz empilhada por duas matrizes 3x2.

Observe que o modelo fourbar 26.c não é capaz de executar uma trajetória elíptica completa, mas apenas um pequeno segmento aproximado desta e também não é fisicamente capaz de partir da posição inicial estabelecida como parâmetro. Neste enfoque as piores condições possíveis estão propostas para experimentação da robustez do algoritmo de controle da inversa filtrada que deverá confinar pelo sinal de controle o efetuador a região pertencente a sua área de trabalho, de modo que este comportamento esteja explicitado como erro de posição que denota a diferença entre a posição desejada (trajetória) e realizada (simulada pela cinemática direta).

Como a trajetória proposta exige do efetuador uma posição completamente fora do seu espaço de trabalho, o manipulador já encontra-se em configuração de singularidade no instante de partida onde não existem soluções possíveis para as juntas que satisfaçam à cinemática inversa. O erro tem caráter oscilatório pois no ensaio a trajetória definida e a executada possuem naquele momento direções ortogonais entre si a uma distância de 10cm, fazendo com que o efetuador oscile entre dois pontos da trajetória alternativa que tenham a menor distância da trajetória desejada. O erro é apresentado em centímetros por iterações de um período e a trajetória em centímetros contendo três períodos e meio.

É possível observar pelo comportamento do erro que uma vez que a trajetória desejada pertença ao espaço de trabalho do efetuador o erro converge para zero, ainda verificam-se perturbações no acompanhamento da trajetória que se sucedem pelo fato de que a trajetória admissível em um four bar linkage é extremamente limitada geometricamente, mas ainda assim é feito um esforço por parte do algoritmo de controle na compensação do erro pelo parâmetro de ganho em atualização dinâmica com objetivo de acompanhar a trajetória proposta ou uma trajetória alternativa, como observado

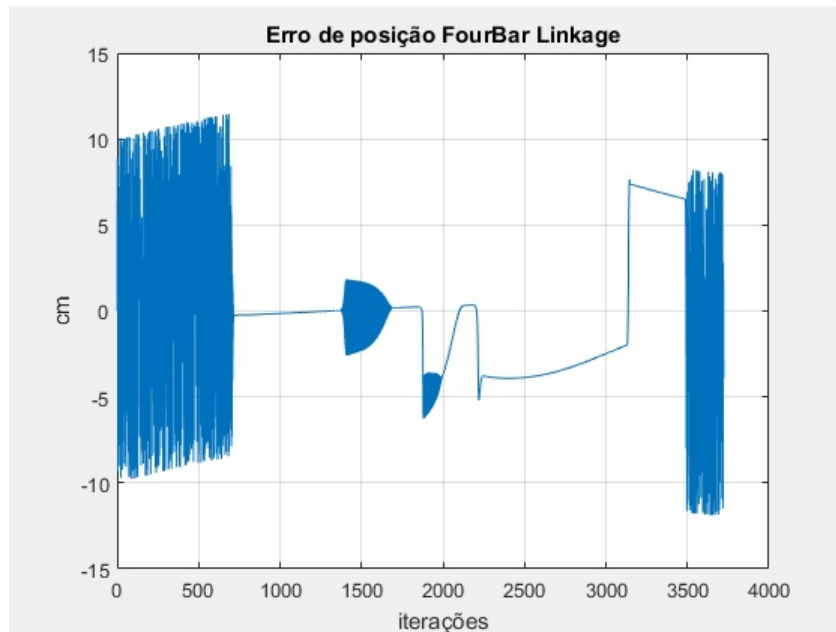


Figura 27: Erro de trajetória - FourBar Linkage

no trajeto onde o erro é constante (2250-3250) que representa deslocamento do efetuador em paralelo à trajetória.

O fato do erro permanecer confinado em 10cm que equivale a distancia máxima da trajetória elíptica ao efetuador e mostra que ao efetuador não foi solicitado nenhuma posição errática como visto nos métodos de controle alternativos. Este comportamento do erro de posição é desejado pois representa a distancia do efetuador à posição inalcançável na trajetória.

## 7 Conclusão

Grande esforço é realizado na intenção de propor trajetórias que se desviem de configurações singulares pertencentes a área de trabalho do efetuador afim de proteger a integridade robótica. Entretanto, baseado em algoritmos de controle robustos às singularidades é possível minimizar este esforço, bem como minimizar as consequências de erros no planejamento de trajetórias. O algoritmo estudado e detalhado neste projeto pode oferecer uma alternativa viável tanto do ponto de vista matemático quanto físico, exigindo menor esforço computacional e maior robustez aos erros de planejamento de trajetórias.

O estudo de casos na simulação do algoritmo da Inversa Filtrada em manipuladores seriais com diferentes configurações permitiu em um primeiro momento validar expectativas iniciais da capacidade do método em limitar a velocidade das juntas na vizinhança das singularidades, manter o índice de manipulabilidade elevado para cada instante de tempo, bem como evitar regiões de singularidade através de trajetórias alternativas do efetuador.

Em contraste com os métodos clássicos, a implementação do algoritmo de controle requer apenas um parâmetro de configuração: o ganho de atualização que limita o quão distante o manipulador deve admitir-se a uma configuração singular. Este parâmetro deve ser estabelecido levando em consideração características próprias de cada manipulador. Em um espaço multivariável de juntas em manipuladores paralelos é necessário atualizar dois ganhos que podem ser ajustados separadamente e também são valores relativos à precisão e robustez de cada manipulador. Em um segundo momento, observamos que a otimização destes parâmetros constitui-se ainda um objeto de pesquisa bastante relevante para o tema de pesquisa proposto.

Neste projeto, foi possível simular com sucesso o sinal de controle que o algoritmo da inversa filtrada aplicaria a um manipulador de estrutura paralela composto por três juntas passivas que reflete a contribuição que o algoritmo de controle estudado representa para evitar as singularidades cinemáticas, minimizando o esforço computacional e otimizando objetivos de controle secundários.

A pesquisa, ainda possui como desafios a universalidade aplicacional, desvio de obstáculos variantes no tempo e auto calibração paramétrica baseado em percentual de segurança.

A pesquisa deve estender-se para o estudo de casos em manipuladores compostos por juntas prismáticas, esféricas e mistas e estender-se do ambiente de simulações numéricas para simulações mecânicas experimentais afim de concretizar resultados esperados.

## Referências

- [1] P. Ráanky 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. Hannel,  
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. Tchon 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. Tcho ´n 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. Tschón 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 realtime 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 endeffector 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 Systems, 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. Tcho'n. 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'Universit'a 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.
- [68] A. C. Leite, and R. R. Costa, Singularity and Joint Limits Avoidance for Parallel Mechanisms using the Filtered Inverse Method, Proceedings of the 2016 American Control Conference, pp.1197-1202, Boston, MA, USA, Jul. 2016.
- [69] L. V. Vargas, A. C. Leite, and R. R. Costa, Overcoming Kinematic Singularities with the Filtered Inverse Approach, Proceedings of the 19th IFAC World Congress, p. 8496-8502, Cape Town, South Africa, Aug 2014
- [70] P.I. Corke; Robotics, Vision and Control - Fundamental Algorithms in Matlab; Springer; 2011;
- [71] P.I. Corke, MATLAB toolboxes: robotics and vision for students and teachers, IEEE Robotics and Automation Magazine, Volume 14(4), Dezembro 2007,