



CINEMÁTICA INVERSA DE ROBÔS REDUNDANTES VIA TEORIA DE HELICÓIDES

Henrique Simas

EMC-CTC-UFSC – Universidade Federal de Santa Catarina – Campus Universitário – Trindade –
Caixa Postal 476 CEP – 88090-400 – e-mail, simas@lcmi.ufsc.br

Daniel Martins

EMC-CTC-UFSC – Universidade Federal de Santa Catarina – Campus Universitário – Trindade –
Caixa Postal 476 CEP – 88090-400 – e-mail, dmartins@lcmi.ufsc.br

Raul Guenther

EMC-CTC-UFSC – Universidade Federal de Santa Catarina – Campus Universitário – Trindade –
Caixa Postal 476 CEP – 88090-400 – e-mail, guenther@emc.ufsc.br

Resumo. *O presente trabalho apresenta duas propostas para cálculo do Jacobiano em algoritmos de cinemática inversa para um robô redundante. Estes algoritmos são baseados em duas formas de representação do Jacobiano do robô: o Jacobiano clássico baseado em Denavit–Hartenberg e baseado em helicóides. São apresentadas as teorias envolvidas e os aspectos numéricos da implementação destes Jacobianos via computador. Resultados comparativos são apresentados e discussões são realizadas na conclusão.*

Palavras chave: *robôs redundantes, cinemática diferencial, cinemática inversa, teoria de helicóides.*

1. INTRODUÇÃO

Um mecanismo pode ser analiticamente considerado como um mapeamento entre dois conjuntos de variáveis, ou mais precisamente, um mapeamento entre os espaços que estas variáveis geram. Um robô é um exemplo de um mecanismo e pode analiticamente ser considerado como um mapeamento $K(q)$ entre o espaço de juntas $q=[q_1, q_2, \dots, q_n]^T$, onde q_i é a atuação sobre um motor, para o espaço Cartesiano $x=[x_1, x_2, \dots, x_n]^T$, onde x_i são as coordenadas da tarefa. Este mapeamento, ou cinemática direta, é altamente não-linear, incluído, sobretudo funções trigonométricas.

$$K : q \rightarrow x \Rightarrow x = K(q) \quad (1)$$

A relação diferencial entre as juntas e o espaço Cartesiano é chamada cinemática diferencial. A cinemática diferencial relaciona a velocidade no espaço de coordenadas das juntas \dot{q} para o espaço Cartesiano (vetor de velocidades \vec{v}) através da Eq. (2).

$$\vec{v} = J\dot{q} \quad (2)$$

Onde J é chamado de matriz Jacobiana ou simplesmente Jacobiano e descreve a contribuição de cada velocidade no espaço de juntas no vetor velocidade do efetuador final.

O mapeamento inverso, ou cinemática inversa (mapeamento do espaço cartesiano para o espaço de juntas), pode ser resolvido por duas metodologias distintas, uma analítica e outra numérica.

No método analítico procura-se resolver diretamente a Eq.(1) encontrando uma função inversa para $K(q)$. Quando ocorrem casos em que um robô possui maior número de juntas que a necessária a realização da tarefa um mapeamento analítico $K^{-1}(q)$ é impossível, o robô então é dito redundante, assim o método mais adequado de obtenção da cinemática inversa é o método numérico. Martins et al(2000).

No método numérico usa-se o Jacobiano e sua respectiva matriz inversa. A metodologia consiste em resolver a Eq.(2) por integração numérica, para prever aproximadamente a configuração das juntas nas proximidades do último ponto computado, em função da taxa de variação do vetor velocidade no espaço Cartesiano e do vetor posição das juntas no espaço de juntas. No caso de robôs redundantes, o Jacobiano não é uma matriz quadrada, e para resolver a inversão da Eq.(2) tem-se como alternativa a utilização do método de inversão de uma matriz pseudo-inversa, (Tsai ,1999,Waldron et all ,1985).

Uma concepção cinemática redundante tem por objetivo proporcionar uma mobilidade adicional ao robô de acordo com especificações da tarefa a qual é empregado.

Na UFSC, o projeto ROBOTURB compreende de uma parceria entre laboratórios na UFSC, Fundação CERTI e LACTEC e tem por objetivo desenvolver um robô capaz de realizar a manutenção em superfícies de pás de turbinas hidráulicas deteioradas pelo processo de cavitação. A Fig. (1) apresenta o protótipo construído.



Figura 1. Protótipo do robô do projeto ROBOTURB

Este robô terá como tarefas, a medição das erosões e a deposição de material pelo processo de soldagem. O volume de trabalho restrito e complexo determinou uma concepção cinemática aberta e redundante para o robô, Guenther (2000), composta de uma junta prismática, caracterizada por um trilho onde o robô é posicionado por um carro e seis juntas rotativas, perfazendo sete graus de liberdade, que devem funcionar conjuntamente, Fig.(2).

O presente artigo apresenta uma comparação entre dois métodos de obtenção do Jacobiano e seu respectivo inverso para um robô redundante. A primeira proposta é utilizar um método clássico baseado na notação de Denavit-Hartenberg, Denavit et al (1955), a segunda proposta é baseada na teoria de helicóides Hunt (1987). Uma abordagem teórica de cada método é discutida e os respectivos Jacobianos são construídos.

Estes algoritmos numéricos para cálculo dos Jacobianos foram construídos considerando como dados de entrada, os parâmetros geométricos e cinemáticos de um robô, no caso, foi considerado o modelo mecânico do robô redundante do Projeto ROBOTURB. Foi realizada uma análise da esparsidade dos Jacobianos calculados, cujo objetivo foi comparar o esforço computacional para o

cálculo da pseudo-inversa e uma análise numérica, considerando aspectos de precisão e complexidade do algoritmo de cinemática inversa, mostrando qual metodologia e respectivo Jacobiano é o mais adaptado aos algoritmos de geração de trajetória no espaço de trabalho a serem construídos brevemente no projeto ROBOTURB.

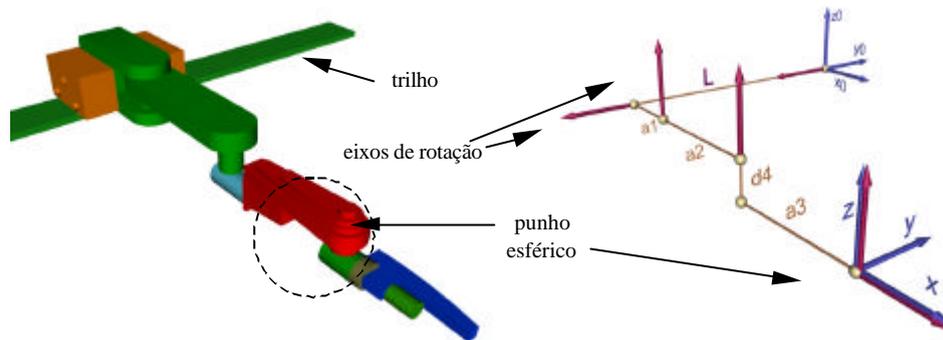


Figura 1. Modelo cinemático do robô do ROBOTURB e dimensões significativas.

2. Cinemática inversa de robôs redundantes

Para obter a cinemática inversa de robôs, pode-se proceder de duas maneiras, a primeira resolvendo analiticamente a Eq.(1) apresentada na seção 1, a segunda é inverter o Jacobiano, resolvendo a Eq.(3), ou seja:

$$\dot{q} = J^{-1}v \quad (3)$$

O Jacobiano de um robô redundante não possui posto completo, ou seja, o espaço de juntas possui maior número de variáveis que o espaço da tarefa, portanto na resolução da Eq.(1) obtém-se infinitas soluções. A solução é resolver a diferencial da Eq.(2) determinando a pseudo-inversa do Jacobiano. Devido as facilidades computacionais deste segundo método, Sciavicco et all(1996) este foi adaptado numericamente para calcular o Jacobiano.

No processo de geração de trajetórias um algoritmo numérico deve ser implementado e procedimentos devem numericamente, calcular a pseudo-inversa do Jacobiano e resolver a Eq.(3).

2.1 A pseudo-inversa do Jacobiano

Conhecido o Jacobiano J (para uma dada configuração de juntas), deve-se encontrar um conjunto de soluções que satisfaçam Eq.(3) de forma otimizada. Este problema pode ser resolvido utilizando o método dos multiplicadores de Lagrange Sciavicco (1996). Consideremos uma função custo quadrática g .

$$g(\dot{q}, \lambda) = \frac{1}{2} \dot{q}^T W \dot{q} + \lambda^T (v - J\dot{q}) \quad (4)$$

onde W ($n \times n$) é uma matriz simétrica positiva definida, n é o número de graus de liberdade do robô, λ é um vetor ($r \times 1$) de multiplicadores desconhecidos que minimizam Eq.(4) e r é a dimensão do espaço de trabalho a qual o robô realiza a tarefa, Sciavicco et all (1996).

Derivando-se a Eq.(4) em relação a \dot{q} e igualando-se a zero para obter o mínimo da função custo,

$$\dot{q} = W^{-1} J^T \lambda \quad (5)$$

Substituindo Eq.(5) em Eq.(2):

$$v = JW^{-1}J^T\lambda \quad (6)$$

Resolvendo ã em Eq.(6) tem-se

$$\lambda = (JW^{-1}J^T)^{-1}v \quad (7)$$

Substituindo Eq.(7) em Eq.(5) tem-se a soluão otimizada.

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

Simplificando a soluão para

$$\dot{q} = J^\#v \quad (9)$$

onde a matriz,

$$J^\# \triangleq W^{-1}J^T(JW^{-1}J^T)^{-1} \quad (10)$$

 a pseudo-inversa a direita de J, que minimiza a norma das velocidades das juntas.

2.2 Algoritmo de cinemtica inversa para robs redundantes

A implementaão numrica da cinemtica inversa pode ser realizada resolvendo numericamente a equaão diferencial (9) pelo mtodo de Euler definido em Ruggiero (1996). Em determinado instante t_{k+1} , $q(t_{k+1})$ pode ser calculado como segue.

$$q(t_{k+1}) = q(t_k) + J^\#(q(t_k))v(t_k)\Delta t \quad (11)$$

Este equacionamento  simplificado e de fcil implementaão, entretanto, no garante a reconstruão de $q(t)$, devido ao fenmeno chamado de *drift* na soluão, e conseqentemente a posião do efetuador final no corresponder a desejada.

Conhecendo-se o erro de posião e velocidade no espao Cartesiano, uma soluão para minimizar o *drift* da Eq. (11)  implementar o esquema de cinemtica inversa apresentada na Fig.(2). Sciavicco et all (1996).

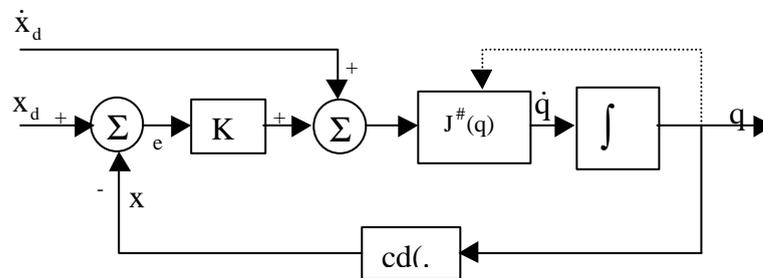


Figura 2. Esquema de algoritmo de cinemtica inversa para robs redundantes.

Tem-se como soluão:

$$q(t_{k+1}) = q(t_k) + J^\#(\dot{x}_d(t_k) + Ke(t_k))\Delta t_k \quad (12)$$

Onde $\dot{x}_d(t_k)$ é o vetor velocidade desejada no espaço cartesiano num instante t_k , $J^\#$ é a pseudo-inversa num dado instante t_k , $e(t_k)$ é o erro de posição no espaço cartesiano, K é uma matriz positiva definida, normalmente diagonal e $cd(\cdot)$ um algoritmo de cinemática direta.

Computacionalmente a implementação da Eq.(12) tem como o elemento mais complexo de ser determinado numericamente a pseudo-inversa do Jacobiano ($J^\#$). Conhecido o Jacobiano, utiliza-se a Eq.(10) para calcular a pseudo-inversa. A complexidade do algoritmo depende diretamente da complexidade das operações para obtenção do Jacobiano.

3. Jacobiano clássico

O Jacobiano baseado nas regras de Denavit-Hartenberg é aqui denominado clássico, pois é o mais estudado em robótica. Encontrado na maioria das bibliografias e é o mais difundido.

O Jacobiano clássico pode ser implementado baseado nas matrizes homogêneas entre os elos descritas pelos parâmetros de Denavit-Hartenberg.

Então para um robô de n graus de liberdade tem-se os seguintes passos para o cálculo do Jacobiano. Tsai (1999)

$$v = \begin{bmatrix} v_n \\ \omega_n \end{bmatrix} = J\dot{q} \quad (13)$$

Onde $v_n = [v_x \ v_y \ v_z]^T$ é o vetor velocidade linear no espaço cartesiano e $\omega_n = [\omega_x \ \omega_y \ \omega_z]^T$ o vetor velocidade angular de num sistema de coordenadas n representados sobre um sistema de coordenadas fixo. A matriz J é definida por:

$$J = \begin{bmatrix} J_{p_1} & \cdots & J_{p_n} \\ \vdots & \ddots & \vdots \\ J_{o_1} & \cdots & J_{o_n} \end{bmatrix} \quad (14)$$

onde,

J_{p_i} Contribuição da junta i na velocidade linear e $J_{p_i} = z_{i-1}$ para juntas prismáticas;
 $J_{p_i} = z_{i-1} \times^{i-1} p_n^*$ para juntas rotativas;

J_{o_i} Contribuição da junta i na velocidade angular e $J_{o_i} = 0$ para juntas prismáticas;
 $J_{o_i} = z_{i-1}$ para juntas rotativas;

Onde z_{i-1} corresponde a orientação do eixo z da junta i em relação ao sistema de coordenadas fixo. Pode ser calculado conhecendo-se a matriz de rotação entre o sistema 0 e $i-1$ (${}^0R_{i-1}$ que pode ser obtida através de Denavit-Hartenberg) como segue:

$$z_{i-1} = {}^0R_{i-1} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad (15)$$

o vetor ${}^{i-1}p_n^*$ corresponde a distância entre os sistemas de coordenadas $i-1$ e n expresso nas coordenadas do sistema fixo. É calculado como segue;

$${}^{i-1}P_n^* = {}^0R_{i-1} {}^{i-1}r_i + {}^iP_n^* \quad (16)$$

onde, ${}^{i-1}r_i$ é dado por:

$${}^{i-1}r_i = \begin{bmatrix} a_i \cos(\theta_i) \\ a_i \sin(\theta_i) \\ d_i \end{bmatrix} \quad (17)$$

e a_i , θ_i e d_i são parâmetros de Denavit-Hartenberg do elo i .

Esta metodologia de obtenção do Jacobiano mostra-se factível, conhecendo-se os parâmetros de Denavit-Hartenberg e as matrizes homogêneas associadas.

4. Jacobiano baseado na teoria de helicóides

O teorema de Mozzi, Ball(1900) afirma deslocamento espacial de um corpo rígido pode ser descrito por uma translação e uma rotação. A combinação desta translação e rotação define um eixo chamado de helicóide (*screw* em língua inglesa), o qual provoca um deslocamento helicoidal no corpo rígido.

O eixo helicoidal S possui uma orientação definida por um vetor unitário s e uma posição s_0 , também definida em relação a um sistema de coordenadas de referência.

Define-se uma posição inicial qualquer do robô e mais conveniente à tarefa que será realizada. Posicionam-se os eixos helicóides em cada eixo das juntas entre elos consecutivos.

Definem-se então os eixos s_i e $s_{0,i}$ em relação a um sistema de coordenadas j , que necessariamente não precisa ser a base do robô Tsai (1999). O primeiro eixo é fixo no sistema de coordenadas da base ($s_1, s_{0,1}$).

A Fig.(3) mostra uma seqüência de cinco juntas rotativas de um robô identificando a posição e orientação dos eixos helicóides.

Pela notação dos parâmetros dos eixos helicóides, é possível o levantamento de matrizes homogêneas entre os sistemas de coordenadas fixo nos elos de um robô Tsai (1999). Com isto é possível, não somente uma configuração inicial, mas sim qualquer uma que seja mais adequada à realização da tarefa a que se destina o robô. Associada a esta possibilidade pode-se escolher uma posição inicial tal que o Jacobiano obtido seja mais complexo, portanto deve-se ponderar entre um posicionamento inicial mais adequado e a obtenção de um Jacobiano mais simplificado.

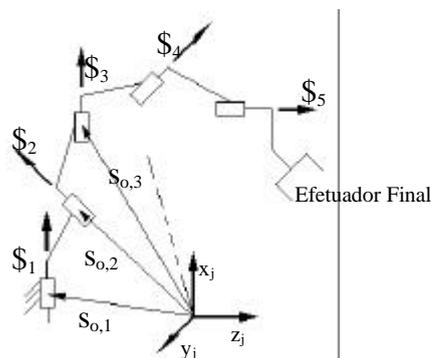


Figura 3. Posições dos eixos helicóides.

O Jacobiano calculado por helicóides é determinado em relação a um sistema de coordenadas de referência j . O Jacobiano é então calculado de acordo com a equação a seguir, Tsai (1999)

$$v = \begin{bmatrix} {}^j\omega_n \\ {}^jv_o \end{bmatrix} = {}^jJ\dot{q} \quad (18)$$

onde,

${}^j\omega_n$ é o vetor velocidade angular do efetuador final em relação ao sistema de coordenadas de referência j .

jv_o corresponde ao vetor velocidade linear de um ponto no efetuador final de um robô que é instantaneamente coincidente com a origem do sistema de coordenadas fixo.

O Jacobiano é obtido da seguinte forma:

$${}^jJ = [{}^j\hat{s}_1, {}^j\hat{s}_2, \dots, {}^j\hat{s}_n] \quad (19)$$

onde,

$$\hat{s}_i = \begin{bmatrix} s_i \\ s_{0,i} \times s_i + \lambda s_i \end{bmatrix}$$

λ é chamado de passo (*pitch*) e é a razão entre a translação (d) e a rotação (θ) do eixo helicoidal

$$\lambda = \frac{\dot{d}}{\dot{\theta}} = \frac{d}{\theta} \quad (20)$$

Para juntas rotativas $\lambda = 0$ então:

$$\hat{s}_i = \begin{bmatrix} s_i \\ s_{0,i} \times s_i \end{bmatrix} \quad (21)$$

Para juntas prismáticas $\lambda = \infty$ então:

$$\hat{s}_i = \begin{bmatrix} 0 \\ s_i \end{bmatrix} \quad (22)$$

Para calcular o Jacobiano iJ em relação à um sistema de coordenada j (que neste caso pode ser o sistema de coordenadas da base $i=1$) é necessário realizar uma transformação de coordenadas dos eixos helicóides.

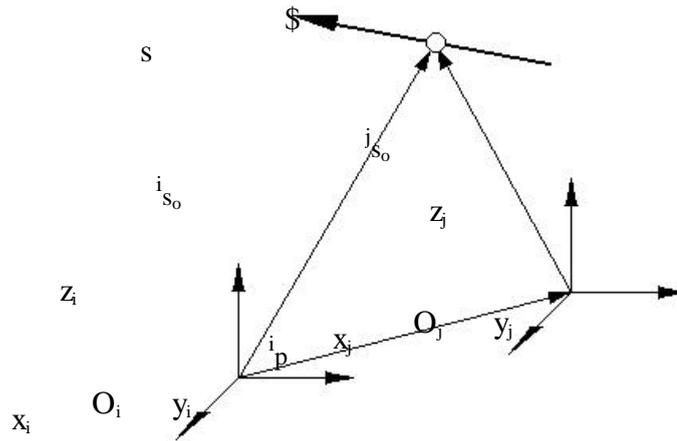


Figura 5. Transformação de coordenadas helicóides de um sistema j para um sistema i

Para obter-se as coordenadas de ${}^i s$ e ${}^i s_o$ em relação ao sistema de coordenadas j calcula-se da seguinte forma.

$${}^i s = {}^i R_j {}^j s \quad (23)$$

$${}^i s_o = {}^i p + {}^i R_j {}^j s_o \quad (24)$$

Onde ${}^i R_j$ é a matriz de rotação entre os sistemas de coordenadas i e j e ${}^i p$ é o vetor posição da origem do sistema de coordenadas j nas coordenadas do sistema de coordenadas i

Obtém-se o produto vetorial de ${}^i s$ e ${}^i s_o$ calculando o produto vetorial e substituindo as equações (23) e (24) tem-se:

$${}^i s_o \times {}^i s = ({}^i p + {}^i R_j {}^j s_o) \times ({}^i R_j {}^j s) = {}^i p \times ({}^i R_j {}^j s) + {}^i R_j ({}^j s_o \times {}^j s) \quad (25)$$

Então o eixo helicoidal representado no sistema i é dado por

$${}^i \$ = \begin{bmatrix} {}^i R_j {}^j s \\ {}^i R_j ({}^j s_o \times {}^j s) + {}^i p \times ({}^i R_j {}^j s) + \lambda {}^i R_j {}^j s \end{bmatrix} {}^j \$ \quad (26)$$

ou seja,

$${}^i \$ = {}^i \tilde{T}_j {}^j \$$$

Manipulando vetorialmente (26) obtém-se a seguinte representação de ${}^i \tilde{T}_j$:

$${}^i \tilde{T}_j = \begin{bmatrix} {}^i R_j & O \\ {}^i W_j {}^i R_j & {}^i R_j \end{bmatrix} \quad (27)$$

a matriz ${}^i W_j$ é uma matriz anti-simétrica dada por:

$${}^iW_j = \begin{bmatrix} 0 & -p_z & p_y \\ p_z & 0 & -p_x \\ -p_y & p_x & 0 \end{bmatrix} \quad (28)$$

[p_x, p_y, p_z] representam o vetor posição do sistema de coordenadas fixo da base do robô e o sistema de coordenadas de referência onde se quer representar o helicóide.

A transformação inversa, ou seja, representar um helicóide definido em relação a um sistema de coordenadas j em outro sistema de coordenadas i é dada pela seguinte matriz de transformação.

$${}^j\tilde{T}_i = \begin{bmatrix} {}^iR_j^T & O \\ {}^iR_j^T {}^iW_j^T & {}^iR_j^T \end{bmatrix} \quad (29)$$

Então as velocidades de junta, o vetor \dot{q} , é expresso por:

$$\dot{q} = {}^jJ^{\#} {}^j\tilde{T}_0 v \quad (30)$$

Onde ${}^jJ^{\#}$ é a pseudo-inversa expressa no sistema de coordenadas de referência j , ${}^j\tilde{T}_0$ é a matriz de transformação de coordenadas entre o sistema j e a base do robô onde são expressas as velocidades e orientações do efetuador final desejadas e $v(t)$ é o vetor de velocidade linear e velocidade angular desejada do efetuador final expressa nas coordenadas da base.

5. Implementação dos Jacobianos

Rotinas foram implementadas no pacote MATHEMATICA 3.0, para a obtenção dos Jacobianos. Assim, foi possível automatizar o processo de cálculo dos Jacobianos. Utilizou-se dos equacionamentos apresentados e foram definidos algoritmos que implementam o Jacobiano clássico e o Jacobiano baseado em helicóides, Simas et al (2001).

5.1 Implementação do Jacobiano clássico

Para implementação do Jacobiano clássico foram definidas matrizes homogêneas de acordo com os parâmetros segundo as regras de Denavit Hartenberg, mostrados na Tab. (1)

Tabela 1. Parâmetros cinemáticos do modelo obtidos pela regra de Denavit-Hartenberg

Junta	α_i	a_i	θ_i	d_i	Tipo de junta
1	0	0	0	L	Prismática
2	$-\bullet / 2$	$a_1 = 150\text{mm}$	θ_2	0	Rotacional
3	0	$a_2 = 300\text{mm}$	θ_3	0	Rotacional
4	$\bullet / 2$	0	θ_4	$d_4 = -89\text{mm}$	Rotacional
5	$-\bullet / 2$	0	θ_5	$a_3 = 300\text{mm}$	Rotacional
6	$\bullet / 2$	0	θ_6	0	Rotacional
7	0	0	θ_7	0	Rotacional

Utilizando-se da rotina implementada no MATHEMATICA 3.0 para cálculo do Jacobiano clássico, obteve-se um Jacobiano de ordem 6X7 com elementos muito extensos contendo termos de senos e cossenos para ângulos entre θ_2 e θ_7 .

Por exemplo, para o elemento da linha 4 e coluna 7 tem-se:

$$J(4,7)=0.125*(-2s_{2-3-4-6}+2s_{234-6}+2s_{2-5-6}-s_{2-3-4-5-6}-s_{234-5-6}-2s_{25-6}-s_{2-3-45-6}-s_{2345-6}-2s_{2-3-46}+2s_{2346}-2s_{26-5}+s_{2-3-4-56}+s_{234-56}+2s_{256}+s_{2-3-456}+s_{23456})$$

onde,

$$\begin{aligned} s_{2-3-4-6} &= \text{sen}(q_2-q_3-q_4-q_6); & s_{234-6} &= \text{sen}(q_2+q_3+q_4-q_6); & s_{2-5-6} &= \text{sen}(q_2-q_5-q_6); \\ s_{2-3-4-5-6} &= \text{sen}(q_2-q_3-q_4-q_5-q_6); & s_{234-5-6} &= \text{sen}(q_2+q_3+q_4-q_5-q_6); & s_{25-6} &= \text{sen}(q_2+q_5-q_6); \\ s_{2-3-45-6} &= \text{sen}(q_2-q_3-q_4+q_5-q_6); & s_{2345-6} &= \text{sen}(q_2+q_3+q_4+q_5-q_6); & s_{2-3-46} &= \text{sen}(q_2-q_3-q_4+q_6); \\ s_{2346} &= \text{sen}(q_2+q_3+q_4+q_6); & s_{26-5} &= \text{sen}(q_2-q_5+q_6); & s_{2-3-4-56} &= \text{sen}(q_2-q_3-q_4- \\ & & & & & q_5+q_6); \\ s_{234-56} &= \text{sen}(q_2+q_3+q_4-q_5+q_6); & s_{256} &= \text{sen}(q_2+q_5+q_6); & s_{2-3-456} &= \text{sen}(q_2-q_3- \\ & & & & & q_4+q_5+q_6); \\ s_{23456} &= \text{sen}(q_2+q_3+q_4+q_5+q_6); & & & & \end{aligned}$$

A complexidade se estende aos demais elementos de J, o que torna difícil a apresentação e utilização deste Jacobiano num algoritmo de cinemática inversa, baseando-se na Eq.(10). A solução é implementar numericamente um algoritmo que calcule passo a passo o Jacobiano clássico a partir dos parâmetros de Denavit-Hartenberg. O resultado foi um algoritmo complexo com uma série de etapas intermediárias e, portanto, numericamente suscetível a erros de arredondamentos e truncamento.

5.2 Implementação do Jacobiano helicoidal

Para a implementação do Jacobiano baseado em helicóides foram definidos os eixos de acordo com a posição inicial mostrada na Fig.(1), e com o sistema de referência colocado no centro do punho esférico $j=4$. Um algoritmo próprio de automatização do processo de cálculo do Jacobiano e a respectiva matriz de transformação de coordenadas foi implementado, Simas, Martins (2001). Este algoritmo proporcionou a flexibilidade necessária a definir qual posição inicial e localização dos eixos helicóides resultaria num modelo mais simplificado para o Jacobiano.

A Tab. (2) mostra os valores dos parâmetros dos eixos helicóides (s_i e $s_{0,i}$)

O resultado deste processo foi um Jacobiano como mostra a matriz 4J na Eq.(31) e a matriz de transformação de coordenadas, do sistema 4 para 0 como mostra a Eq.(32).

Tabela 2. Parâmetros determinados pelo método das helicóides

Junta	s_i	$s_{0,i}$
1	(0,-1,0)	$-(a_1+a_2+a_3),L,d_4$
2	(0,-1,0)	$-(a_1+a_2+a_3),0,d_4$
3	(0,0,1)	$-(a_2+a_3),0,d_4$
4	(0,0,1)	$(-a_3,0,d_4)$
5	(1,0,0)	(0,0,0)
6	(0,0,1)	(0,0,0)
7	(1,0,0)	(0,0,0)

$${}^4J = \begin{bmatrix} 0 & -s_{34} & 0 & 0 & 1 & 0 & c_6 \\ 0 & -c_{34} & 0 & 0 & 0 & -s_5 & c_5 s_6 \\ 0 & 0 & 1 & 1 & 0 & c_5 & s_5 s_6 \\ -s_{34} & d_4 c_{34} & a_2 s_4 & 0 & 0 & 0 & 0 \\ -c_{34} & -d_4 s_{34} & a_3 + a_2 c_4 & a_3 & 0 & 0 & 0 \\ 0 & a_1 + a_2 c_3 + a_3 c_{34} & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (31)$$

$${}^4T_0 = \begin{bmatrix} c_2c_{34} & s_{34} & c_{34}s_2 & 0 & 0 & 0 \\ -c_2c_{34} & c_{34} & -s_2s_{34} & 0 & 0 & 0 \\ -s_2 & 0 & c_2 & 0 & 0 & 0 \\ d_4c_2s_{34} - s_2p_1 & -d_4c_{34} & d_4s_2s_{34} + c_2p_1 & c_2c_{34} & s_{34} & c_{34}s_2 \\ d_4c_2c_{34} - s_2p_3 & d_4s_{34} & d_4s_2c_{34} + c_2p_3 & -c_2c_{34} & c_{34} & -s_2s_{34} \\ c_2p_2 & -a_1 - a_2c_3 - a_3c_{34} & s_2p_2 & -s_2 & 0 & c_2 \end{bmatrix} \quad (32)$$

onde,

$$p_1 = Lc_{34} + a_2s_4 + a_1s_{34}$$

$$p_2 = -L + a_2s_3 + a_3s_{34}$$

$$p_3 = a_3 + a_2c_4 + a_1c_{34} - Ls_{34}$$

$$s_2 = \text{sen}(q_2); c_2 = \text{cos}(q_2);$$

$$s_4 = \text{sen}(q_4); c_4 = \text{cos}(q_4);$$

$$s_5 = \text{sen}(q_5); c_5 = \text{cos}(q_5);$$

$$s_6 = \text{sen}(q_6); c_6 = \text{cos}(q_6);$$

$$s_{34} = \text{sen}(q_3+q_4); c_{34} = \text{cos}(q_3+q_4).$$

6. Análise comparativa da estrutura dos Jacobianos

A análise da esparsidade do Jacobiano permite pré avaliar a dificuldade de se obter a sua inversa.

Comparando a esparsidade do Jacobiano clássico obtido na seção 5.1

$$J|_{\text{clássico}} = \begin{bmatrix} 0 & x & x & x & 0 & 0 & 0 \\ 0 & x & x & x & 0 & 0 & 0 \\ x & 0 & x & x & 0 & 0 & 0 \\ 0 & 0 & x & x & x & x & x \\ 0 & 0 & x & x & x & x & x \\ 0 & x & 0 & 0 & x & x & x \end{bmatrix} \quad (33)$$

com a esparsidade do Jacobiano baseado em helicóides obtido na seção 5.2

$$J|_{\text{helicóides}} = \begin{bmatrix} 0 & x & 0 & 0 & x & 0 & x \\ 0 & x & 0 & 0 & 0 & x & x \\ 0 & 0 & x & x & 0 & x & x \\ x & x & x & 0 & 0 & 0 & 0 \\ x & x & x & x & 0 & 0 & 0 \\ 0 & x & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (34)$$

nota-se que o Jacobiano baseado em helicóides apresenta mais elementos estruturalmente nulos, o que o torna mais propício a inversão

Nota-se que o Jacobiano baseado em helicóides é mais simples de ser implementado podendo ser inserido diretamente num algoritmo de cinemática inversa, na Eq.(12), sem a necessidade de se implementar todo o algoritmo de cálculo do Jacobiano, como no Jacobiano clássico.

A esparsidade verificada na Eq.(34) demonstra que para o cálculo da pseudo-inversa deste Jacobiano, um número menor de operações será necessário.

Vale ressaltar que as regras de Denavit-Hartenberg determinam um posicionamento único do manipulador, enquanto a teoria de helicóides proporciona a liberdade de escolha da posição inicial e da junta de referência para determinação do Jacobiano. A desvantagem é a procura do melhor

posicionamento inicial do robô e de referência para o Jacobiano. Esta tarefa pode ser árdua, justificando sua automatização. Isto porque não há um método que determine as posições e referências na teoria de helicóides, tais que simplifiquem os resultados obtidos.

Numericamente o Jacobiano baseado na teoria de helicóides mostra-se como a alternativa mais viável para o algoritmo de cinemática inversa. A perspectiva é um estudo de um método de integração mais adequado na solução da equação diferencial Eq.(9).

7. Conclusão

Este artigo discutiu e apresentou duas propostas de implementação para o Jacobiano de robôs manipuladores redundantes, mais precisamente aplicado ao robô do projeto ROBOTURB.

A primeira proposta baseada num algoritmo clássico de acordo com as regras e parâmetros de Denavit-Hartenberg. A segunda alternativa apresentada foi o Jacobiano baseado em helicóides.

O objetivo foi comparar estes dois métodos e avaliar qual se adapta melhor a implementação numérica da cinemática inversa de robôs redundantes. Para este fim foi apresentada uma proposta de algoritmo de cinemática inversa levantando seus aspectos mais relevantes de sua implementação numérica, especialmente caracterizando o problema do cálculo da pseudo-inversa do Jacobiano.

Foram apresentados os aspectos teóricos de cada metodologia de cálculo dos Jacobianos.

Um robô redundante de sete graus de liberdade foi utilizado para validar as comparações. Para determinar os Jacobianos foram implementados no pacote MATHEMATICA, rotinas que calculam Jacobianos simbolicamente a partir dos parâmetros de cada método. Estas rotinas foram eficazes e aceleraram de forma expressiva a obtenção dos resultados finais.

O resultado final foi que o Jacobiano baseado na teoria de helicóides mostra-se o mais eficaz na implementação numérica de um algoritmo de cinemática inversa. O Jacobiano clássico não se mostrou eficaz numericamente, pois para calcular este Jacobiano é necessário um algoritmo que calcule todos os passos de construção, isto acarreta maior número de operações aritméticas de ponto flutuante.

Como perspectivas pretende-se aperfeiçoar os algoritmos implementados no MATHEMATICA automatizando o processo de construção das matrizes homogêneas e do Jacobiano baseado em helicóides.

Referências bibliográficas

Ball, R. S. , 1900, “ A Treatise on the Theory of Screw” Cambridge University Press, Cambridge.

Denavit, J. and Hartenberg, R. S. , 1955, “ A kinematic notation for lower-pair mechanisms based on matrices” Trans. ASME E, Journal of Applied Mechanics” 22:215-221.

Guenther, R.; Simas, H. , 2000, “Concepção cinemática de um robô redundante– Projeto ROBOTURB” Relatório Interno, Universidade Federal de Santa Catarina, Florianópolis, SC.

Simas, H., Martins, D., 2001 “Automatização do processo de cálculo de Jacobianos – Projeto ROBOTURB”, Relatório Interno, Universidade Federal de Santa Catarina, Florianópolis, SC

Ruggiero, M. A. G.; Lopes, V. L. R. , 1999, “ Cálculo Numérico – aspectos teóricos e computacionais” Makron Books, São Paulo, SP.

Hunt, K.H., 1987, Robot kinematics – a compact analytic inverse solution for velocities” Trans. ASME, journal of Mechanism, Transmission and Automation in Design, 109:42-49.

Waldron, K.J.;Wang, S.;Bolin, S.J. “ A study of the Jacobian matrix of serial manipulator” Transactions of the ASME, Journal of Applied Mechanics, 107:230-238.

Sciavicco, L. Siciliano, B. ;1996, ” Modeling and control of robot manipulator”, Mc-Graw-Hill, NewYork.

Tsai, L. W. ,1999; “Robot analysis - the mechanics of serial and parallel manipulators”, John Wiley & Sons, New York.

INVERSE KINEMATIC OF A REDUNDANT ROBOT VIA SCREW THEORY

Henrique Simas

EMC-CTC-UFSC – Universidade Federal de Santa Catarina – Campus Universitário – Trindade – Caixa Postal 476 CEP – 88090-400 – e-mail, simas@lcmi.ufsc.br

Daniel Martins

EMC-CTC-UFSC – Universidade Federal de Santa Catarina – Campus Universitário – Trindade – Caixa Postal 476 CEP – 88090-400 – e-mail, dmartins@lcmi.ufsc.br

Raul Guenther

EMC-CTC-UFSC – Universidade Federal de Santa Catarina – Campus Universitário – Trindade – Caixa Postal 476 CEP – 88090-400 – e-mail, guenther@emc.ufsc.br

***Abstract.** This work presents two proposals of how to obtain the Jacobian matrix for use in inverse kinematics algorithms of a redundant robot. These algorithms are based on two representations of manipulator Jacobians: the Denavit-Hartenberg approach and the screw theory approach. Both theories are presented and their numerical aspects are compared both theoretically as well as on computer simulations. Finally , we compare and discusses the obtained results.*

***Keywords.** redundant robots, differential kinematics, inverse kinematics, screw theory*