



Programa Interdisciplinar de Pós-Graduação em
Computação Aplicada

Mestrado Acadêmico

Luiz Ricardo Bertoldi de Oliveira

Cinemática Inversa para Simulação de Figuras Articuladas

São Leopoldo
2018

Luiz Ricardo Bertoldi de Oliveira

CINEMÁTICA INVERSA PARA SIMULAÇÃO DE FIGURAS ARTICULADAS

Dissertação submetida à avaliação para a
obtenção do título de Mestre pelo Programa de
Pós-Graduação em Computação Aplicada da
Universidade do Vale do Rio dos Sinos —
UNISINOS

Orientador:
Profa. Dra. Marta Becker Villamil

São Leopoldo
2018

O48c Oliveira, Luiz Ricardo Bertoldi de.
Cinemática inversa para simulação de figuras articuladas /
Luiz Ricardo Bertoldi de Oliveira. – 2018.
45 f. : il ; 30 cm.

Dissertação (mestrado) – Universidade do Vale do Rio dos
Sinos, Programa de Pós-Graduação em Computação Aplicada,
2018.

“Orientador: Prof. Dra. Maria Becker Villamil”.

1. Cinemática. 2. Figuras articuladas. 3. Topologia inversa. 4.
Cinemática inversa. I. Título.

CDU 004

Dados Internacionais de Catalogação na Publicação (CIP)
(Silvana Teresinha Dornelles Studzinski – CRB 10/2524)

Luiz Ricardo Bertoldi de Oliveira

Cinemática Inversa para Simulação de Figuras Articuladas

Dissertação apresentada à Universidade do Vale do Rio dos Sinos - Unisinos, como requisito parcial para obtenção do título de Mestre em Computação Aplicada.

Aprovado em 19/07/2018

BANCA EXAMINADORA

Prof. Dr. Anderson Maciel - UFRGS

Prof. Dr. Maurício Roberto Veronez - UNISINOS

Prof^a. Dr^a. Marta Becker Villamil (Orientadora)

Visto e permitido a impressão
São Leopoldo, 19 de setembro de 2018.

Prof. Dr. Rodrigo da Rosa Righi
Coordenador do PPG em Computação Aplicada

*Dedico este trabalho para todos aqueles que apesar das minhas ausências,
continuaram incansavelmente me apoiando.*

AGRADECIMENTOS

Sempre achei que esta seria a pior parte da dissertação para escrever, talvez porque a vida não se coloca em análise de regressão e não é um valor de p de probabilidade que descobrimos a significância das pessoas em nossa trajetória.

Primeiro de tudo, gostaria de agradecer a Deus por me guiar, iluminar e me dar tranquilidade para seguir em frente com meus objetivos e não desanimar com as dificuldades.

Agradeço a minha família, amigos e minha namorada, que sempre me motivaram, entenderam as minhas faltas e momentos de afastamento e reclusão e me mostraram que por mais difícil que seja uma caminhada sempre teremos apoio daqueles que nos amam.

Os meus sinceros agradecimentos à minha orientadora profa. Dra. Marta Becker Villamil, primeiramente pela oportunidade, posteriormente pela paciência, incentivos e contribuições para o meu aprendizado e minha formação.

Ao PPGCA/UNISINOS, funcionários e professores, que contribuíram diretamente e indiretamente em minha formação profissional e pessoal.

Colegas de PPG, muito obrigado pela ajuda, discussões, apoio e *coffee breaks*. Em especial ao colega Vinicius Facchin Nonnenmacher pelos ensinamentos de programação e dedicação para me auxiliar no desenvolvimento do algoritmo dessa dissertação.

Agradeço à CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior) pela concessão da bolsa durante o período de realização deste mestrado.

Muito obrigado a todos!

*“Existe uma coisa que uma longa existência me ensinou:
toda a nossa ciência, comparada à realidade,
é primitiva e inocente; e, portanto,
é o que temos de **mais valioso**”.*
(Albert Einstein)

RESUMO

A simulação de figuras articuladas em computação gráfica que representam animais, pessoas ou robôs, normalmente, são compostas por uma cadeia de segmentos rígidos conectados por juntas dispostas de maneira hierárquica. Quando estas figuras possuem muitos segmentos, e precisam ser animadas por cinemática inversa, sua configuração de movimento pode apresentar múltiplas soluções ou soluções não-lineares além de singularidades de movimento. Este trabalho propõe uma nova topologia de juntas de hierarquia variável, generalista o suficiente para se aplicar a qualquer modelo articulado. Além disto, propõe um algoritmo de cinemática inversa que supra a não-linearidade dos modelos propostos até o momento. Um modelo como este, formaliza a animação de figuras articuladas que não precisam ter uma hierarquia de juntas rígida ou que possuam muitas juntas e segmentos. Para validação do modelo foi usada a Distância de Hausdorff para cálculo da similaridade do *end-effector* e um conjunto de pontos ideais com uma precisão 91.23%. O movimento foi suavizado em comparação a outros modelos por meio da minimização de 8,7% nos ângulos nas juntas.

Palavras-chave: cinemática inversa. figuras articuladas. topologia inversa.

ABSTRACT

The simulation of articulated figures in computer graphics that represents animals, people or robots is usually composed of a chain of rigid segments connected by joints arranged in a hierarchical way. When these figures have many segments and need to be animated by inverse kinematics, their motion configuration may present multiple solutions or nonlinear solutions and motion singularities. This work proposes a new topology of joints of a variable hierarchy, general enough to apply to any articulated model. In addition, it proposes an inverse kinematics algorithm to supply the non linearity of the models already proposed. A model like this generalizes the animation of articulated figures because it do not need to have a rigid joints topology and works to figures with many joints and segments. Hausdorff Distance was used to calculate the *end-effector* similarity with an ideal set of points. The accuracy achieved is 91,23% of similarity. Motion of all joints is smoother when compared to other models by minimizing 8,7% of the angles.

Keywords: inverse kinematics. articulated figures. inverse topology.

LISTA DE FIGURAS

Figura 1 – Árvore hierárquica para estrutura articulações de movimento	18
Figura 2 – Exemplificação de um modelo hierárquico e a estrutura da árvore	18
Figura 3 – Relações entre as cinemáticas diretas e inversa	19
Figura 4 – Exemplos de redundâncias no plano x, y, com dois graus de liberdade (esquerda) e três graus de liberdade (direita)	20
Figura 5 – Modelo aproximado da estrutura cinemática do manipulador e análise de convergência da resposta final para os três ângulo formadores do braço	24
Figura 6 – Planejamento do trajeto de todo o corpo e modelo de robô <i>hexapod</i> usando o SolidWorks 2015	25
Figura 7 – Um braço robótico de 4-DOF em 3-dimensões e seu volume de alcance dos <i>end-effectors</i>	26
Figura 8 – Modelo robótico 3D e curva de análise da trajetória teórica com relação curva de trajetória real	27
Figura 9 – Modelo robótico de 2 DOF e 4 DOF, construídos para realização dos resultados obtidos no artigo	27
Figura 10 – Cada articulação é composta por dofs.	29
Figura 11 – Movimento realizado por uma articulação em dois segmentos para direções diferentes	30
Figura 12 – Diferentes posições de rotação para a figura articulada em contida na maior circunferência (C_1).	30
Figura 13 – Etapas do processo atuando para posicionar a extremidade do braço no ponto E	31
Figura 14 – Circunferências interseccionadas sobre o ponto efetor E	32
Figura 15 – Articulado com diferentes tamanhos de elos e estrutura do algoritmo consistente	33
Figura 16 – Mudança da topologia da estrutura das juntas	34
Figura 17 – Representação da nomenclatura utilizada para o estudos dos ângulos gerados pela figura articulada até o ponto E	34
Figura 18 – Representação angular das juntas, seguindo a estrutura geométrica.	35
Figura 19 – Representação do modelo 3D seguindo a mesma estrutura de construção do modelo 2D	36
Figura 20 – Formação de planos de orientação para os cálculos angulares.	36
Figura 21 – Plano gerado pelos pontos J_i, J_{i+1} e E	37
Figura 22 – Amostra do conjunto de pontos utilizado em um dos métodos nos experimentos	38
Figura 23 – Comparação de configuração para alcançar o ponto objetivo, (a) Modelo proposto e (b) matriz Jacobiana.	40
Figura 24 – Representação de construção para o estudo comparativo de similaridade para ser aplicado à métrica de Hausdorff.	42

Figura 25 – Visualização dos dados de similaridade a partir de DH. (a) Rastro gerado pelo modelo proposto; (b) Conjunto de pontos objetivos a serem comparados; (c) Jacobiana e Rede Neural ; (d) D-H, *fsolve* e Jacobiana; (e) Jacobiana e Modelo do autor; (f) Inversa da Jacobiana; (g) Algoritmos Genéticos e Jacobiana. 43

LISTA DE TABELAS

Tabela 1 – Comparativo entre os trabalhos selecionados, relacionando (Autor; DOFs; Juntas; Topologia; Técnicas)	28
Tabela 2 – Valores aproximados dos métodos utilizados pelos autores para alcançar o ponto objetivo.	41
Tabela 3 – Valores aproximados de similaridade dos métodos utilizados pelos autores para alcançar o ponto objetivo.	43

LISTA DE SIGLAS

DOF	Grau de liberdade de movimento de uma junta
2D	Segunda dimensão
3D	Terceira dimensão

SUMÁRIO

1	INTRODUÇÃO	14
1.1	Objetivos	15
1.2	Contribuições	15
1.3	Organização da dissertação	15
2	FUNDAMENTOS TEÓRICOS	17
2.1	Hierarquia de movimento	17
2.2	Cinemática	19
2.2.1	Cinemática Inversa	19
3	REVISÃO DE BIBLIOGRÁFICA: TRABALHOS RELACIONADOS	23
4	MODELO GEOMÉTRICO GENERALIZADO PARA APLICAÇÕES EM CINEMÁTICA INVERSA	29
4.1	Modelo 2D	33
4.2	Modelo 3D	35
5	RESULTADOS DA ANÁLISE DO MODELO	38
5.1	Elementos para o desenvolvimento dos experimentos	38
5.1.1	Distância de Hausdorff	39
5.2	Resultados	39
5.2.1	Pontos objetivos unitários	40
5.2.2	Pontos objetivos múltiplos	42
6	CONCLUSÃO E TRABALHOS FUTUROS	44
	REFERÊNCIAS	45

1 INTRODUÇÃO

As principais abordagens desenvolvidas para animar personagens virtuais na área da computação gráfica, são: captura de movimento, métodos baseados em cinemática e dinâmica, ou abordagens conjuntas. Todos estes métodos utilizam modelos muito semelhantes para a animação de figuras articuladas: um conjunto de segmentos rígidos interligados por juntas com uma topologia hierárquica.

Os modelos hierárquicos comumente usados são organizados sob a forma de uma árvore, os nós correspondem as juntas que possuem como filhos outras juntas e os segmentos rígidos. Um método típico da aplicação da cinemática é representar um personagem como uma figura composta de palitos, que simulam a estrutura de um esqueleto. A figura é manipulada por um algoritmo de animação que reposiciona as juntas na estrutura esquelética da mesma maneira que se manipula uma marionete. Os pontos nos quais as cordas prendem aos palitos são as juntas em outras palavras as variáveis de animação (BROOKSHEAR, 2013).

Para a representação de figuras de humanos e animais a topologia de árvore das juntas e segmentos é suficiente. Entretanto complexas estruturas articuladas, de alta dimensão, com muitas possibilidades de posição final e movimento que pode ser gerado em qualquer articulação e em qualquer direção, são extremamente complexas para modelar. Este trabalho, se propõe a justamente modelar o movimento deste tipo de estruturas com foco na cinemática inversa.

A cinemática inversa possui uma aplicabilidade importante na animação por computador, simulação de figuras articuladas e robótica. Com o suporte de algoritmos de cinemática inversa, o animador só aponta a localização desejada de alguns pontos escolhidos e deixa o algoritmo automaticamente calcular o conjunto de ângulos para cada junta que fazem o *end-effector* atingir o ponto desejado. O termo *end-effector* é usual na robótica para identificar qualquer objeto anexado na extremidade do braço robótico.

Para colocar a extremidade de um braço robótico com n articulações e n graus de liberdade (DOFs¹) em um local estipulado utiliza-se a cinemática inversa. Diante disso, cria-se a necessidade de métodos e modelos para solucionar a tarefa de calcular o ângulo em cada junta. Um dos métodos matemáticos mais utilizados para calcular os ângulos das articulações em cinemática inversa é a Jacobiana que possui soluções não-lineares, e com o aumento da quantidade de juntas e DOF's torna a implementação quase impossível.

Este trabalho propõe um modelo computacional que desenvolva a cinemática inversa de forma mais simples, diminuindo restrições e limitações das técnicas propostas até o momento, aplicando também mudança em sua estrutura hierárquica para aperfeiçoamento do modelo proposto.

¹Degree Of Freedom. Graus de liberdade é o número de parâmetros independentes que são necessários para se definir a posição de um corpo no espaço em qualquer instante

1.1 Objetivos

O objetivo principal deste trabalho é construir um modelo de cinemática inversa para simulação de figuras articuladas com n juntas, sem topologia rígida de juntas.

Para a consecução do objetivo principal acima, objetivos parciais foram estabelecidos:

1. Definir um processo de modelagem acessível para aplicabilidade e construção de problemas acerca da cinemática inversa;
2. Identificar e estudar a não linearidade da técnica jacobiana;
3. Estabelecer um modelo com mudança em sua topologia (sem hierarquia de movimento rígida), ou seja, modelo se adaptará de forma a criar mais possibilidades de movimento.
4. Construir modelos 2D e 3D, ambos com o estudo de resultados com relação a sua funcionalidade.

1.2 Contribuições

A técnica de cinemática inversa não possuiu em seu estado da arte um modelo generalista o suficiente para várias aplicações. Os trabalhos relacionados a essa área de estudo funcionam para casos isolados. Como por exemplo, restrição na quantidades de juntas e hierarquia de movimento rígida. As principal contribuição deste trabalho é um modelo no qual seja viável a aplicação da cinemática inversa em **figuras articuladas com quantidade de juntas ilimitadas e mudança de hierarquia em suas juntas** (ou mudança na direção do movimento).

Na robótica, a aplicação de um modelo como este impede a colisão do braço/manipulador robótico com o ponto objetivo desejado para o seu *end-effector*, além da minimização e suavização nas articulações. Como contribuição adicional o modelo possui um custo computacional baixo, a precisão do *end-effector* para atingir o ponto desejado e a quantidade de iterações com relação ao tempo significativamente menores do que modelos já consolidados.

1.3 Organização da dissertação

Este trabalho está organizado de forma que o leitor tenha uma breve visão sobre os fundamentos teóricos da cinemática inversa e animação de figuras articuladas, no Capítulo 2. Sendo apresentados as técnicas utilizadas e teorias que auxiliam as aplicações e o entendimento de modelos criados para estudos de figuras articuladas e manipuladores robóticos.

No Capítulo 3 é apresentado o estado da arte relacionado ao desenvolvimento deste trabalho. São apresentadas as técnicas comumente usadas na robótica, já que aplicabilidade de cinemática inversa encontra-se parcialmente nessa área. São destacadas as questões pertinentes as articulações e topologia, em especial ao avanço de técnicas a solucionar a cinemática inversa.

No Capítulo 4 expõe o processo do modelo aqui proposto para solução de cinemática inversa, apresentando a técnica e o algoritmo para a estrutura 2D e 3D de forma detalhada.

No Capítulo 5 apresenta os resultados obtidos pela comparação dos métodos utilizados pelos autores no Capítulo 3, estudando a minimização das articulações, colisão com o ponto objetivo e custo computacional. No Capítulo final são tecidas algumas considerações, comentários dos resultados encontrados, assim como as expectativas de trabalhos futuros que podem ser desenvolvidos.

2 FUNDAMENTOS TEÓRICOS

Este capítulo apresenta a abordagem de hierarquia de juntas comumente usada para representação de corpos articulados. Em seguida são explanados os principais conceitos de cinemática para aplicações relacionadas a figuras articuladas e à robótica. A cinemática inversa, que é o foco desse trabalho, utiliza-se de métodos geométricos e algébricos.

2.1 Hierarquia de movimento

Figuras humanas e animais são convenientemente modelados como ligações hierárquicas. Tais ligações podem ser representadas por uma estrutura em árvore de nós conectados por arcos.

O nó de inicial da árvore é o nó da raiz, que corresponde ao objeto raiz da hierarquia cuja posição é conhecida no sistema de global. A posição de todos os outros nós da hierarquia será localizada em relação ao nó da raiz.

O mapeamento entre a hierarquia e a estrutura da árvore relaciona um nó da árvore com a transformação para se aplicar a todos os nós abaixo da hierarquia. Relacionar um arco de árvore com uma figura articular pode parecer contra-intuitivo, mas é conveniente porque um nó da árvore pode ter vários arcos emanando nele, assim como e a parte do objeto pode ter várias juntas anexadas a ele. Em uma discussão de um modelo hierárquico apresentado por uma estrutura de árvore específica, os termos nó, parte do objeto e link são usados na forma de ligação, uma vez que todos se referem à geometria a ser articulada. Os termos *joint* e *arc* são usados similarmente (PARENT, 2002).

Na estrutura da árvore, como na Figura 1, existe um arco de raiz que representa uma transformação global para se aplicar ao nó da raiz (e, portanto, indiretamente a todos os nós da árvore). Alterar essa transformação irá reposicionar rígida toda a estrutura no sistema de coordenadas global (MARTINS; GUENTHER, 2003) (PARENT, 2002).

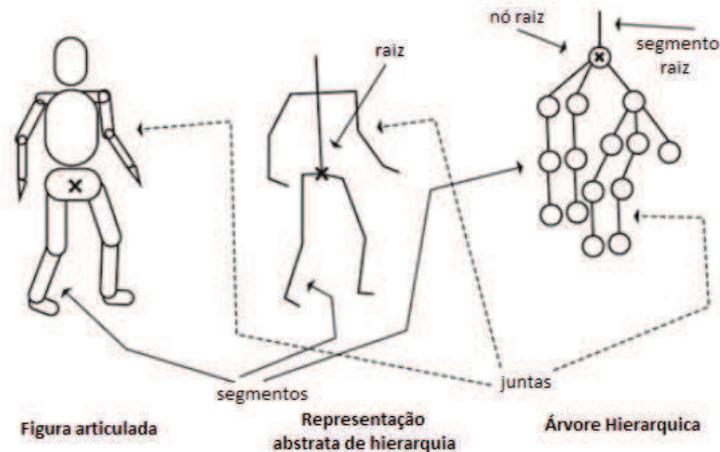
Um nó da estrutura da árvore contém a informação necessária para definir a parte do objeto em uma posição pronta para ser articulada. No caso das articulações rotativas, isso significa que o ponto de rotação na peça do objeto é feito para coincidir com a origem.

Os dados do objeto podem ser definidos em tal posição, ou pode haver uma matriz de transformação contida no nó que, quando aplicada aos dados do objeto, posiciona-o assim. Em ambos os casos, toda a informação necessária para preparar os dados do objeto para articulação está contida no nó. O nó representa a transformação dos dados do objeto em um link do modelo hierárquico (ESCANDE; MANSARD; WIEBER, 2010) (PARENT, 2002).

Dois tipos de transformações estão associados a um arco que leva a um nó. Uma transformação roda e traduz o objeto em sua posição de anexo em relação ao link de uma posição na hierarquia. Isso define a posição neutra do link em relação ao pai. A outra transformação é a informação variável responsável pela articulação atual.

A partir da Figura 2 é estudado a estrutura de exemplificação criado por Villamil (VILLA-

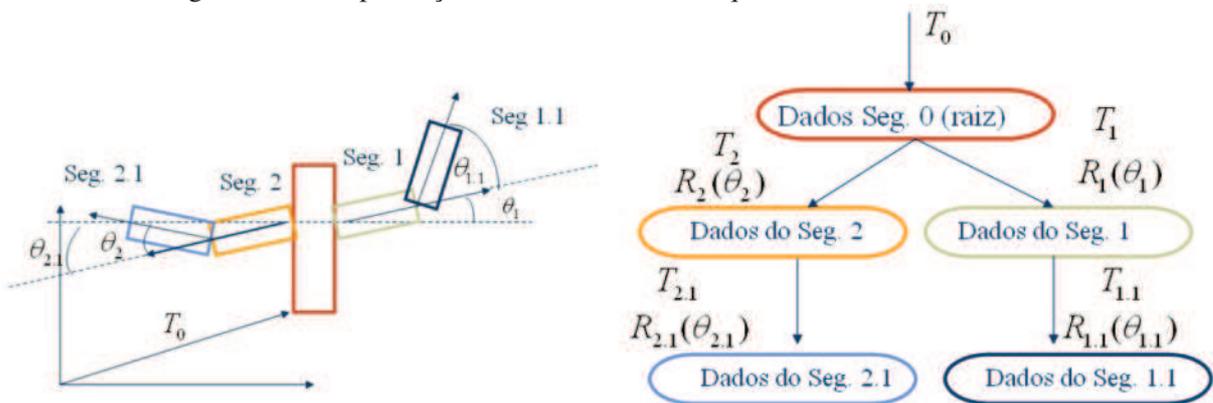
Figura 1 – Árvore hierárquica para estrutura articulações de movimento



Fonte: Adaptado de (PARENT, 2002).

MIL, 2009). O *Segmento 0* é a representação do objeto raiz, sendo transformado para a sua posição e orientação por T_0 . Como todas as outras partes da hierarquia estão relacionadas por esta parte, esta transformação afeta todas as outras partes. Dessa forma, transformando a posição e a orientação de toda a estrutura. Esta transformação pode sofrer modificações ao longo do tempo para animar a posição e orientação de toda a estrutura rígida. O *Segmento 1* é definido relativamente ao objeto raiz pela translação T_1 e rotação R_1 .

Figura 2 – Exemplificação de um modelo hierárquico e a estrutura da árvore



Fonte: Extraído de (VILLAMIL, 2009).

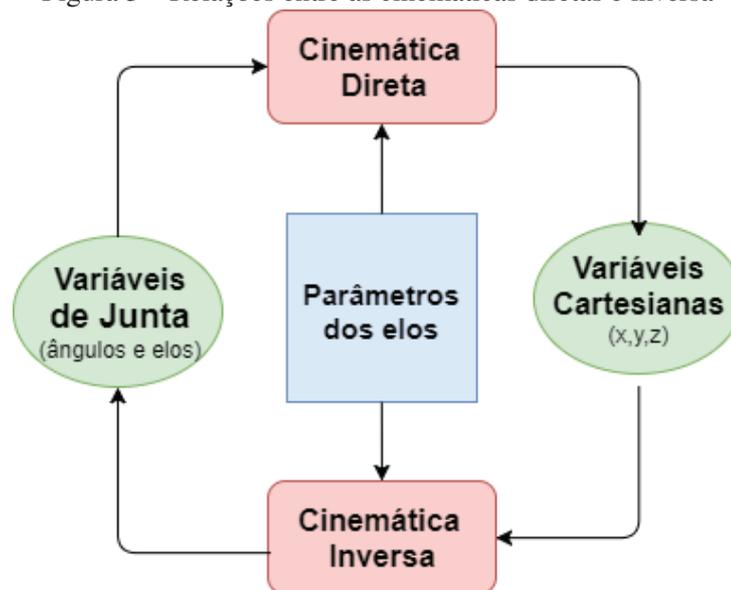
Da mesma forma, o *Segmento 1.1* é definido em relação ao *Segmento 1* pela translação $T_{1.1}$ e rotação $R_{1.1}$. Notações diferentes vêm sendo utilizadas para representação de hierarquia em corpos articulados. A notação Denavit-Hartenberg é uma das notações usuais na robótica para representar uma estrutura articulada (DENAVID J.; HARTENBERG, 1955). Contudo esta notação, mesmo havendo semelhanças na representação descrita anteriormente, possui ainda diferenças na maneira como se gera o movimento.

2.2 Cinemática

A cinemática é um dos ramos da mecânica que estuda o posicionamento e a movimentação de corpos, sem considerar as, forças e causas presentes (CRAIG, 2012). Na área da robótica é possível, a partir de estudos cinemáticos e de características geométricas do manipulador criar relações de sistema de referencial do manipulador com o espaço onde se encontra.

Pela cinemática direta é possível determinar a posição da ferramenta em um sistema de coordenadas cartesianas, sabendo as variáveis de cada junta e os parâmetros dos segmentos, encontrando solução única. Entretanto, na cinemática inversa, por meio do conhecimento da posição da figura articulada e os parâmetros dos segmentos, é possível determinar a configuração das variáveis das juntas, ou seja, os ângulos que satisfazem o correto posicionamento do manipulador, gerando n possíveis soluções. A representação da relação entre essas cinemáticas (Figura 4).

Figura 3 – Relações entre as cinemáticas diretas e inversa



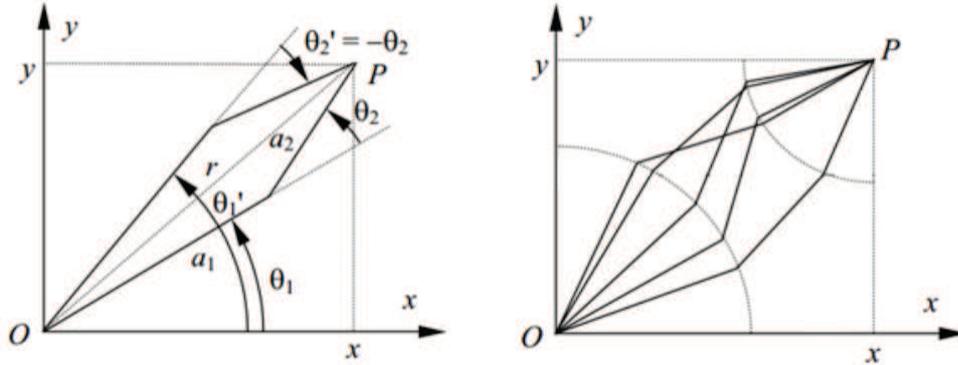
Fonte: Adaptado de (PARENT, 2002).

2.2.1 Cinemática Inversa

Diferentemente da cinemática direta que tem solução única, o problema da cinemática inversa é não-linear nas suas funções e as redundâncias que causam múltiplas soluções (NUNES, 2016). A cinemática inversa é utilizada quando se deseja determinar os ângulos das juntas de modo que satisfaça a posição final do terminal do manipulador (*end effector*) no plano de coordenadas tridimensionais ou cartesianas. A estrutura matemática é frequentemente requerida para programação de braços robóticos, uma vez que existe singularidades e infinitas possibilidades do *end effector* atingir determinada posição e orientação desejada (CRAIG, 2012).

A partir da posição inicial do manipulador, ou seja, o referencial de sua base representado por O (Figura 4) e a posição desejada no outro extremo do manipulador representada por P , existem muitas possibilidades de configuração dos ângulos das juntas. Portanto, existem redundâncias na solução da cinemática inversa, como representado na Figura 4 para manipuladores planares de dois e três graus de liberdade.

Figura 4 – Exemplos de redundâncias no plano x, y , com dois graus de liberdade (esquerda) e três graus de liberdade (direita)



Fonte: Extraído de (NUNES, 2016).

A partir da Figura 4 à esquerda, percebe-se que o valor do ângulo θ_1 depende se o ângulo de θ_2 é positivo ou negativo, isto é, se o manipulador está com a junta para cima ou para baixo. Na Figura 4 à direita, o primeiro segmento está com uma extremidade fixa na base e é possível posicionar a outra extremidade, o mesmo ocorrerá com o último segmento, que deve estar posicionado na posição desejada. Logo, podem existir infinitas soluções que satisfaçam o posicionamento da figura articulada de forma que alcance determinada posição.

A maioria dos mecanismos de interesse para a animação por computador são muito complexas para permitir uma solução analítica. Para isso, o movimento pode ser construído de forma incremental. Em cada passo de tempo, é realizada uma determinada apostas para mudar cada ângulo da articulação para direcionar a posição atual e a orientação da extremidade até a configuração desejada. O cálculo forma a matriz de derivadas parciais chamada Jacobiana.

Para explicar as funções da jacobiana de um ponto de vista estritamente matemático, considere as seis funções arbitrárias representadas pelas equações 2.1, 2.2, 2.3, 2.4, 2.5 e 2.6, cada uma das quais é uma função de seis variáveis independentes. Dado valores específicos para as variáveis de entrada, x_i , cada uma das variáveis de saída y_i , pode ser calculada pela respectiva função.

$$y_1 = f_1(x_1, x_2, x_3, x_4, x_5, x_6) \quad (2.1)$$

$$y_2 = f_2(x_1, x_2, x_3, x_4, x_5, x_6) \quad (2.2)$$

$$y_3 = f_3(x_1, x_2, x_3, x_4, x_5, x_6) \quad (2.3)$$

$$y_4 = f_4(x_1, x_2, x_3, x_4, x_5, x_6) \quad (2.4)$$

$$y_5 = f_5(x_1, x_2, x_3, x_4, x_5, x_6) \quad (2.5)$$

$$y_6 = f_6(x_1, x_2, x_3, x_4, x_5, x_6) \quad (2.6)$$

Essas equações também podem ser usadas para descrever a mudança nas variáveis de saída em relação à mudança nas variáveis de entrada. Os diferenciais de y_i podem ser escritos em termos dos derivadas de x_i usando a regra da cadeia, gerando a equação 2.7. A Equação 2.8 e a Equação 2.9 podem ser colocadas em notação vetorial, produzindo a Equação 2.8 e a Equação 2.9, respectivamente.

$$\delta y_i = \frac{\delta f_i}{\partial x_1} \cdot \delta x_1 + \frac{\delta f_i}{\partial x_2} \cdot \delta x_2 + \frac{\delta f_i}{\partial x_3} \cdot \delta x_3 + \frac{\delta f_i}{\partial x_4} \cdot \delta x_4 + \frac{\delta f_i}{\partial x_5} \cdot \delta x_5 + \frac{\delta f_i}{\partial x_6} \cdot \delta x_6 \quad (2.7)$$

$$Y = F(x) \quad (2.8)$$

$$\partial Y = \frac{\partial F}{\partial X} \cdot \partial X \quad (2.9)$$

A matriz 6x6 de derivados parciais, $\frac{\partial F}{\partial X}$, é chamado de *jacobiana* e é uma função do valor corrente de x_i . O jacobiano pode ser pensado como mapeando as velocidades de X para as velocidades de Y (Equação 2.10) em qualquer ponto no tempo, o jacobiano é uma função linear de x_i . No instante seguinte, X modifica sendo a transformação linear representa pelo Jacobiano.

$$Y' = J(X) \cdot X' \quad (2.10)$$

Quando se aplica a Jacobiana a um apêndice ligado, as variações de entrada x_i , tornam-se os ângulos das juntas e as variáveis de saída, y_i , tornam-se a posição e a orientação da extremidade. Neste caso, o jacobiano relaciona as velocidades dos ângulos das juntas com as velocidades da posição e orientação do *end effector* (Equação 2.11).

$$V = J(\theta) \cdot \theta' \quad (2.11)$$

V é o vetor de velocidades lineares e rotacionais e representa a mudança desejada no efector final. A mudança desejada será baseada na diferença entre sua posição/orientação atual para a configuração do ponto objetivo. Essas velocidades são vetores em três dimensões, então cada um tem componentes x, y e z (Equação 2.13). O θ' é um vetor de velocidades do ângulo articular

que são as incógnitas da equação (Equação 2.13). J , a jacobiana, é uma matriz que relaciona os dois e é uma função de representação corrente (Equação 2.14).

$$V = [v_x, v_y, v_z, \omega_x, \omega_y, \omega_z]^T \quad (2.12)$$

$$\theta' = [\theta'_1, \theta'_2, \theta'_3, \dots, \theta'_n] \quad (2.13)$$

$$J = \begin{bmatrix} \frac{\partial v_x}{\partial \theta_1} & \frac{\partial v_x}{\partial \theta_2} & \dots & \frac{\partial v_x}{\partial \theta_n} \\ \frac{\partial v_y}{\partial \theta_1} & \frac{\partial v_y}{\partial \theta_2} & \dots & \frac{\partial v_y}{\partial \theta_n} \\ \dots & \dots & \dots & \dots \\ \frac{\partial \omega_z}{\partial \theta_1} & \frac{\partial \omega_z}{\partial \theta_2} & \dots & \frac{\partial \omega_z}{\partial \theta_n} \end{bmatrix} \quad (2.14)$$

Cada termo da jacobiana relaciona a mudança de uma articulação específica com uma mudança específica do *end effector*. A mudança de rotação do *end effector*, ω , é apenas a velocidade do ângulo articular em relação ao eixo de revolução na articulação em consideração. A mudança linear do *end effector* é o produto cruzado do eixo de revolução e um vetor da junção para o *end effector*. A rotação na articulação induz uma orientação linear instantânea do movimento do *end effector*.

As velocidades angulares e lineares desejadas são calculadas encontrando a diferença entre a configuração atual do *end effector* e a configuração desejada. O problema é determinar a melhor combinação linear de velocidades induzidas pelas várias articulações que resultariam nas velocidades desejadas do *end effector*, ou seja, um problema não-linear. O jacobiano é utilizado por colocar o problema em forma matricial.

Quando monta-se a jacobiana, é importante certificar-se de que todos os valores das coordenadas estão no mesmo sistema de coordenadas. É frequente o caso de informações conjuntas específicas no sistema de coordenadas locais dessa articulação (PARENT, 2002). Ao formar a matriz jacobiana, essa informação deve ser convertida em algum sistema de coordenadas comum, como o sistema de coordenadas inerciais globais ou o sistema de coordenadas do *end effector*. Vários métodos foram desenvolvidos para calcular a Jacobiana com base na obtenção de eficiência computacional máxima, dada a informação necessária em sistemas de coordenada locais, mas todos os métodos produzem a matriz derivada em um sistema de coordenadas comum, vale salientar que sua implementação é não-trivial.

3 REVISÃO DE BIBLIOGRÁFICA: TRABALHOS RELACIONADOS

Alguns dos trabalhos descritos a seguir são advindos de áreas como Robótica, Biomecânica e Engenharias. O enfoque na criação de modelos computacionais do corpo humano e algoritmos robóticos variam bastante. Existem modelos que estão focados em sua maioria na área da indústria de manipuladores robóticos, que trabalhem em funções de linha de montagem, até a delicados robôs em áreas médicas.

A partir de dados da *International Federation of Robotics*, (IFR, 2014), no ano de 1959 um engenheiro e um inventor de uma empresa chamada *Unimation* desenvolveu o primeiro braço robótico, chamado de UNIMATE. Em 1961, esta empresa instala na linha de produção da General Motors, o seu primeiro robô industrial, que através de instruções, ordenava e empilhava peças quentes de metal forjado.

O braço de um robô industrial pode distinguir-se em tamanho e formas. Com o braço do robô, o ombro, o cotovelo, o pulso, ou seja, suas juntas se movem e rotacionam para posicionar a extremidade atuadora no lugar de objetivo de forma exata. Cada uma destas articulações dá ao braço robótico um grau de liberdade, DOF. Muitos braços robóticos industriais são robôs de 6-DOF ou de seis eixos, com três movimentos de rotação em cada junta (2 juntas).

A funcionalidade da indústria em linhas de montagem com a utilização de braços robóticos é um tema pertinente na área científica. Um dos principais temas acerca do desenvolvimento de manipuladores robóticos é a cinemática inversa, que buscam a melhoria da técnica e soluções que possam acelerar sua solução, utilizando métodos oriundos de inteligência artificial, algoritmos genéticos ou melhoria das ferramentas matemáticas utilizadas.

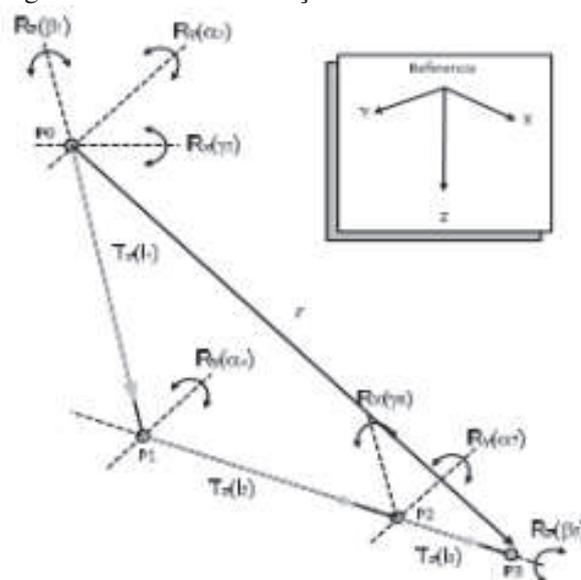
O trabalho de Veslin et.al (VESLIN et al., 2014) apresenta uma solução para a cinemática inversa em um modelo cinemático de braço humano de sete DOFs com aplicação de Algoritmos Genéticos na articulação do ombro e o método Piper em todo o braço para tornar a resposta do algoritmo mais rápida e precisa (MACHADO; AZEVEDO, 2009). Tal proposta possui uma abordagem vetorial para representação do braço e cálculos matemáticos (Figura 5).

O código do Algoritmo Genético explora um conjunto de soluções iniciais e o otimiza até encontrar uma resposta angular que satisfaça as restrições posicionais e rotacionais propostas ao modelo com um erro mínimo e uma velocidade considerável. A precisão e a confiabilidade do algoritmo aplicado são avaliadas aplicando uma análise estatística sobre um conjunto de respostas usando o SPSS Statistics.

A combinação de uma metodologia analítica com uma meta-heurística permite reduzir a complexidade e reduzir o tempo de cálculo, limitando a análise iterativa a um número reduzido de incógnitas (TOLANI; GOSWAMI; BADLER, 2000). Os testes realizados pelos autores em MatLab e em SPSS Statistics permitiu corroborar que o Algoritmo Genético tende a convergir em torno da resposta objetiva, apresentando um sistema que determina uma posição e orientação do braço nas proximidades das condições solicitadas.

Verificou-se que, ao reduzir o espaço de avaliação do manipulador para determinação dos

Figura 5 – Modelo aproximado da estrutura cinemática do manipulador e análise de convergência da resposta final para os três ângulo formadores do braço



Fonte: Extraído de (VESLIN et al., 2014)

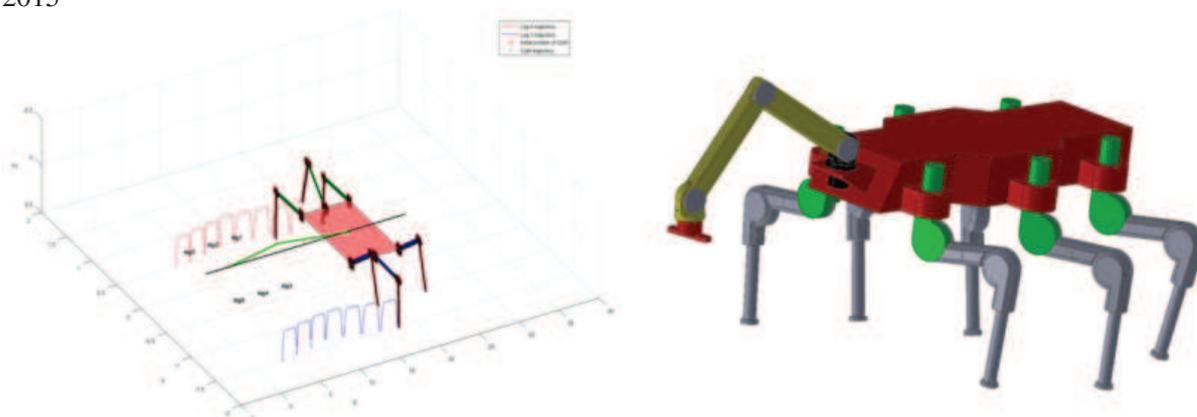
ângulos dos ombros, por sua vez, a avaliação e o tempo de processamento são reduzidos. Por outro lado, restringir os ângulos de posição através da penalidade e condicionar o peso do indivíduo em termos de rotação e posição do braço, permitiu estabelecer critérios de avaliação que fossem efetivos e eficientes para marcar o indivíduo. As restrições colocadas auxiliaram Veslin et.al a alcançar objetivos a sua pesquisa, porém em casos mais abrangentes sua funcionalidade possuirá falhas.

Khudher et.al (KHUDHER; POWELL, 2016) apresenta uma lei de controle de cinemática inversa (MISTRY et al., 2008) para controlar o movimento de um robô de seis patas destinado a ser usado em desminagem humanitária, utilizando para sua simulação este software. Foi explorado um método de programação quádrlica (QP) para resolver o problema de redundância cinemática restrita e incluímos restrições de desigualdade e igualdade, como a taxa de junção e limites de ângulo de junção.

Para solucionar o controle de velocidades, força e angulação de cada junta do robô em estudo, os autores utilizaram a técnica da função matriz jacobiana inversa. Porém não apresentam de forma descritiva os cálculos matemáticos utilizados, focando seu estudo na trajetória requerida por controle de parâmetros. Priorização de tarefas para o robô *hexapod* com cabeça de sensor é usado para resolver a questão do equilíbrio do robô, movimento de caminhada e manipulador (Figura 6).

O algoritmo foi testado em uma única perna de três graus de liberdade, dado que o robô é formado por seis patas com braço portador de cabeça de sensor. Apresentaram tal simulação

Figura 6 – Planejamento do trajeto de todo o corpo e modelo de robô *hexapod* usando o SolidWorks 2015



Fonte: Extraído de (KHUDHER; POWELL, 2016)

usando o MatLab² e modelo estrutural SolidWorks 2015. O autor visa como trabalho futuro se concentrar na integração deste controlador em um robô controlado por torque.

Um dos trabalhos que apresentam as soluções cinemáticas inversas baseadas em redes neurais é o de Srisuk et.al (SRISUK; KITJAIDURE, 2017). A rede neural aborda os dados de uso das posições finais que o manipulador pretender atingir como entrada e as juntas angulares como uma saída para treinar a rede neural e mapear a entrada para a saída.

No entanto, o seu método proposto cria as redes personalizadas de equações cinemáticas inversa. Esta estrutura especial torna a rede como um localizador de posição com a capacidade de ajustar automaticamente as junções angulares até que manipulador atinja a posição desejada por meio da técnica de rede neural BackPropagation, com o algoritmo de taxa de aprendizado variável (TEJOMURTULA; KAK, 1999). As soluções de ângulos podem ser encontradas a partir dos pesos finais e valores de polarização. Além disso, a rede proposta usa menos número de neurônios e a quantidade de espaço da solução não depende dos dados de treinamento.

Para avaliar o algoritmo de desempenho deste trabalho, Srisuk et.al, utilizou o programa MatLab para demonstrar o movimento do braço robótico de 4-DOF em 3 dimensões (Figura 7)

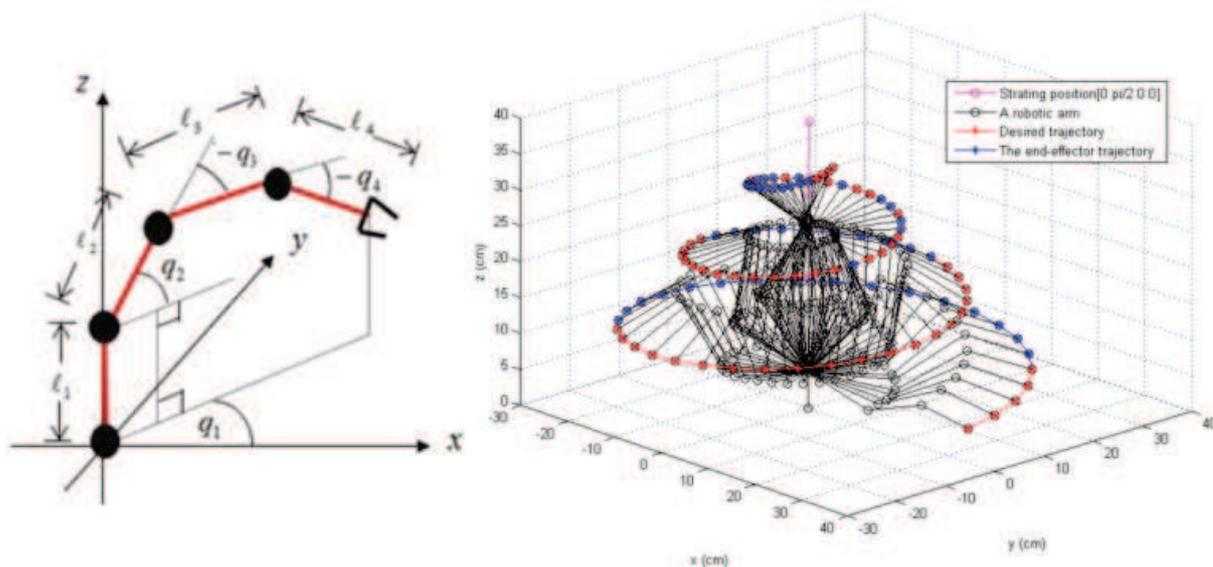
Como resultado, o algoritmo proposto pode ajudar um braço robótico a se mover para a posição desejada (alcance 3D) de forma rápida e correta.

Este artigo, porém pode resolver a cinemática inversa de um braço robótico restrito a 4-DOF em espaços tridimensionais usando a rede neural personalizada criada a partir das equações cinemáticas diretas com rapidez e precisão. Além disso, a estrutura de rede neural usa menos números de camadas ocultas e nós neural. Embora o método proposto não seja mais rápido do que os métodos já existentes, a velocidade da convergência pode ser aumentada.

Latombe (LATOMBE, 1991) define um robô como um dispositivo mecânico versátil que está equipado com atuadores e sensores, sob um controle computacional. Este robô opera num

²MATLAB - MATrix LABoratory é um software interativo de alta performance voltado para o cálculo numérico (CORKE, 2011) (<https://www.mathworks.com>).

Figura 7 – Um braço robótico de 4-DOF em 3-dimensões e seu volume de alcance dos *end-effectors*



Fonte: Extraído de (SRISUK; KITJAIIDURE, 2017)

espaço que pertence ao mundo real, que é sempre constituído por objetos físicos que estão sujeitos às leis da natureza e sofrem escolhas de processos para seu melhor funcionamento e prática.

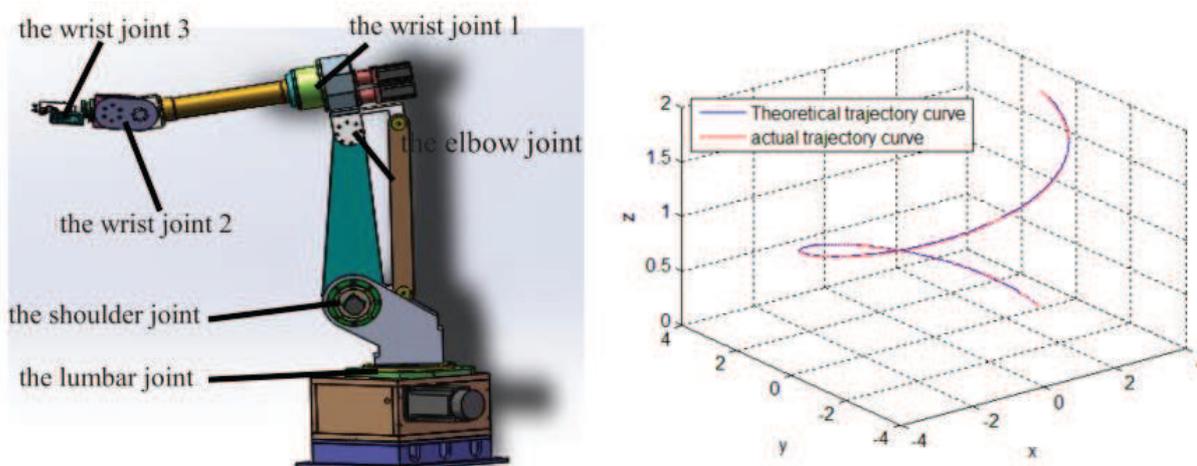
O problema de cinemática inversa de um manipulador de seis graus de liberdade para o planejamento da trajetória foi estudado por Yu et.al (YU; YOU; LIU, 2017). O modelo estudado de cinemática de um manipulador de 6 DOF é estabelecido pelo método de parâmetro D-H e método de espaço articular proposto por Wang et.al (WANG YONG WANG LEI, 2003), no qual é usado para realizar o planejamento da trajetória. O autor utilizou o *toolbox* de robótica do MatLab para resolver a cinemática inversa de cada curva e sua respectiva trajetória.

As curvas de ângulo de junção desenhadas no eixo de tempo existem diferenças locais e, para sua solução, adotam o método de cinemática inversa com base na função *fsolve* (SHI; LVZHONG, 2010). Diante disso, Yu et.al calcula as curvas de ângulo de junção resultantes, nas quais são contínuas e suaves, o que resolve o problema de diferença das curvas de trajetória.

Os autores utilizaram da função Fourier para ajustar as curvas de ângulo das articulações para gerar as funções, das quais a curva de trajetória real é obtida perto da curva de trajetória teórica e o grau de coincidência é alto.

Ao comparar os dois métodos de cinemática inversa Yu et.al, mostrou que a função *fsolve* do Matlab pode resolver o problema multi-solução do processo de cinemática inversa do manipulador, sob a condição de definir o valor inicial apropriado. A precisão do método é verificada pela curva e a simulação de movimento, Figura 8. O efeito de ajuste e o resultado da simulação são satisfatórios. O Software Matlab possui grande aplicabilidade e auxílio no ramo da robótica.

Figura 8 – Modelo robótico 3D e curva de análise da trajetória teórica com relação curva de trajetória real



Fonte: Extraído de (YU; YOU; LIU, 2017)

Perrusquia et.al (PERRUSQUÍA et al., 2017) aborda controladores quando usados principalmente para a interação humano-robô. Quando o controle é aplicado no espaço da tarefa, ele precisará da matriz jacobiana (SINGH; POPA, 2004). Quando o controlador está no espaço articular, requer cinemática inversa para a configuração do robô. O aplicativo é modificado para usar apenas as orientações do *end effector* para gerar os ângulos de junção desejados para evitar cinemática inversa. O controle clássico também é modificado com compensação adaptativa e compensação do modo deslizante. Experimentos em tempo real foram apresentados pelos autores com um robô de 2-DOF e um exoesqueleto de 4-DOF (Figura 9).

Figura 9 – Modelo robótico de 2 DOF e 4 DOF, construídos para realização dos resultados obtidos no artigo



Fonte: Extraído de (PERRUSQUÍA et al., 2017)

Os autores apresentaram novos controladores que trabalham no espaço articular para a solução da cinemática inversa. Esses controladores também usam compensações de modo adaptativo e deslizante para melhorar a precisão de rastreamento. A estabilidade dos controladores

é comprovada através da Análise *Lya-Punov*⁴ (MING; HUANG, 1995). Os controladores propostos são verificados usando um robô de inclinação de 2 DOF e um exoesqueleto de 4 DOF com sensor F/T.

As comparações entre controladores tradicionais e propostos são feitas, através de modelos de análise estatística de dados. Os controladores não utiliza os parâmetros jacobianos e usam os componentes de orientação gerados pelos autores, para realizar a tarefa de cinemática inversa. A abordagem proposta visou apresentar um modelo que não necessite da cinemática inversa usual, com a construção matricial da jacobiana ou sua inversa da matriz, mas apartir de seu modelo.

Abaixo na tabela 1 esta o comparativo entre os trabalhos selecionados do estado da arte atual. Os trabalhos acima descritos e analisados apresentaram em sua totalidade problemas aplicados em casos genéricos, ou seja, com restrições a casos particulares aplicados. No trabalho Perrusquia et.al (PERRUSQUÍA et al., 2017), mostra que a utilização de técnicas como a construção matricial da jacobiana para solução da cinemática inversa, podem haver outras soluções, assim Perrusquia et.al propôs seu modelo próprio que não é necessário a utilização dos modelos de cinemática inversa usuais, assim como o trabalho aqui proposto.

Tabela 1 – Comparativo entre os trabalhos selecionados, relacionando (Autor; DOFs; Juntas; Topologia; Técnicas)

Autor	DOFs	Juntas	Topologia	Técnicas
(SRISUK; KITJAIDURE, 2017)	4	4	hierarquia rígida	Jacobiana e Rede Neural
(YU; YOU; LIU, 2017)	6	3	hierarquia rígida	D-H, <i>fsolve</i> e Jacobiana
(VESLIN et al., 2014)	7	3	hierarquia rígida	Algoritmos Genéticos e Jacobiana
(KHUDHER; POWELL, 2016)	3	2	hierarquia rígida	Inversa da Jacobiana
(PERRUSQUÍA et al., 2017)	2 e 4	2 e 4	hierarquia rígida	Jacobiana/Modelo próprio
Modelo proposto	6	n	mudança de hierarquia	geométrica

A partir da Tabela 1 os trabalhos selecionados para o estudo em sua maioria utilizam a técnica da matriz jacobiana, pois é a técnica usual para problemas relacionados a robótica. Entretanto, a jacobiana possui limitações devido a sua solução não-linear. O modelo aqui proposto difere dos demais trabalhos por não possuir restrição na quantidade de juntas da figura articulada e a possibilidade de aplicação da inversão da direção do movimento.

⁴Utilizado para análise de estabilidade de sistemas de potência

4 MODELO GEOMÉTRICO GENERALIZADO PARA APLICAÇÕES EM CINEMÁTICA INVERSA

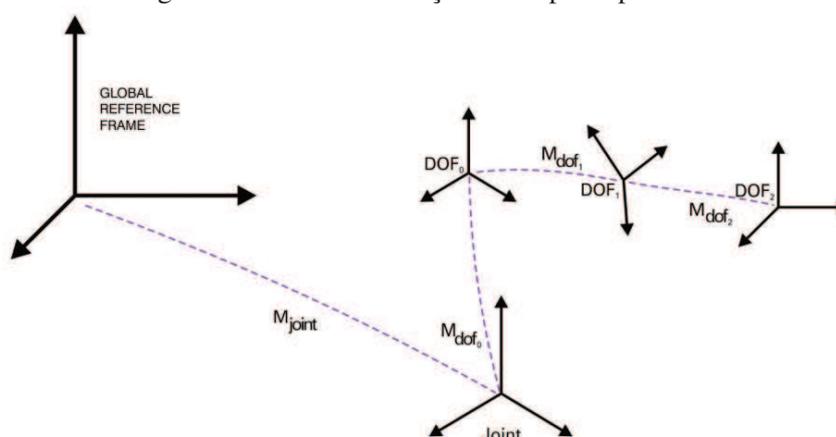
O modelo de uma figura articulada de n juntas com inversão de hierarquia de movimento consiste em apresentar uma solução simples para aplicações de cinemática inversa, dado que as soluções existentes possuem restrições, funcionam para alguns casos controlados e utilizam modelos matemáticos de solução não-linear.

Uma junta representada neste modelo é um conjunto de movimentos possíveis que são definidos de acordo com seus graus de liberdade (dofs em minúsculo). A presença ou não de um DOF em uma junta, define se a junta pode (ou não) permitir um movimento específico (MACIEL; NEDEL; FREITAS, 2002) (MACIEL, 2010).

Uma junta de 2 DOFs, por exemplo, determina dois tipos de movimento, flexão/extensão ou adução/abdução. Basicamente, para cada estrutura de Dof corresponde: sua posição de repouso, seus limites angulares máximo e mínimos, e o eixo onde o movimento será feito. Para flexionar uma junta, devemos enviar-lhe um novo parâmetro de flexão que é usado para determinar o novo ângulo para o Dof de flexão para a junta (Figura 10).

O sistema de referência global é representado por uma matriz que descreve o referencial posicionado e orientado (GIM). Mover uma junta é o mesmo que modificar sua matriz. Para fazer as mudanças desejáveis em uma matriz, usamos seus Dofs. Cada Dof também possui uma matriz, que descreve um quadro de referência local do Dof e é baseado no seu Dof pai. Para o primeiro Dof de uma articulação, que não possui um Dof principal, a matriz deste é filha da GIM. Assim, a matriz de uma junta é recalculada a partir de suas matrizes Dofs toda vez que ocorre movimento.

Figura 10 – Cada articulação é composta por dofs.



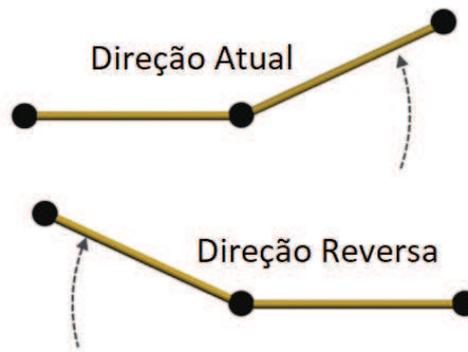
Fonte: (MACIEL; NEDEL; FREITAS, 2002).

A topologia aqui proposta precisa lidar com o padrão e direções de movimento reverso de cada junta da figura articulada. A Figura 11 apresenta a direção da origem do movimento de

uma figura articulada com 3 juntas, inicialmente da esquerda para a direita e depois sua direção reversa (esquerda para a direita).

A animação das figuras articuladas torna-se complexa quando estas possuem segmentos e inversão de direção da origem de movimento em suas juntas.

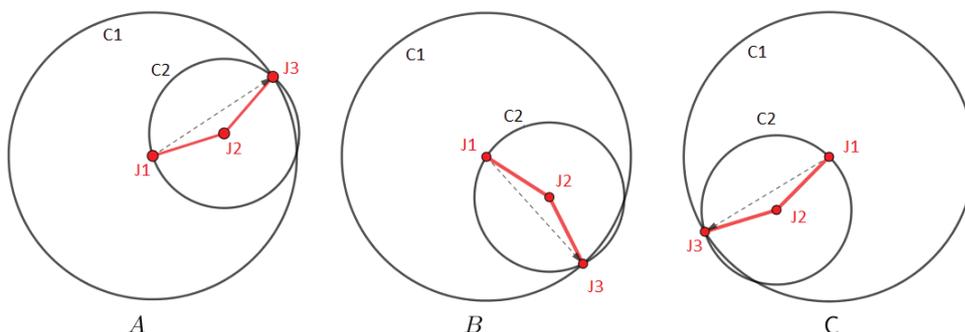
Figura 11 – Movimento realizado por uma articulação em dois segmentos para direções diferentes



Fonte: Criado pelo autor

A ideia do modelo associa uma circunferência com o centro de cada junta da figura articulada com exceção da última junta. Sendo assim, uma junta J_i , possui circunferência C_i onde o centro é a própria junta, e o raio é a distância da junta até a extremidade do braço articulado. A circunferência representa todos os pontos alcançáveis pelo braço através da rotação da junta J_i . Por exemplo, a Figura 12 apresenta-se diferentes valores de rotação aplicados a junta J_1 onde cada rotação posiciona a extremidade em um diferente ponto da circunferência C_1 . As rotações nas juntas subsequentes podem alterar o raio da circunferência C_1 .

Figura 12 – Diferentes posições de rotação para a figura articulada em contida na maior circunferência (C_1).



Fonte: Criado pelo autor

Em um braço articulado com n juntas, cada uma com sua circunferência o *end effector* é a intersecção entre todas as circunferências. Se o *end effector* for posicionado em um ponto E , esse mesmo ponto E deverá ser a intersecção de todas as circunferências (Figura 13).

Algoritmo 1: MÉTODO RECURSIVO PARA DETECÇÃO DO PONTO OBJETIVO

Entrada: $J_i, J_{i+1}, \dots, J_n; C_i, C_{i+1}, \dots, C_n; E$
Saída: Intersecção de todas as $C_i \dots C_n$ com os pontos E e J_n

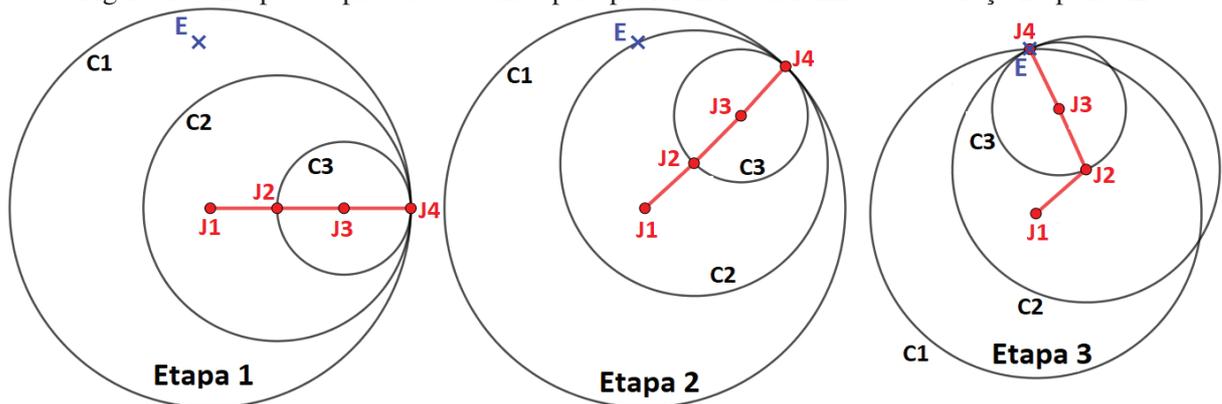
```

1 início
2    $C_i \leftarrow 0$ 
3    $J_i \leftarrow 0$ 
4   while  $E \neq J_n$  do                                     O ponto  $J_n$  é o end-effector
5        $C_i \leftarrow C_{i+1}$ 
6        $J_i \leftarrow J_{i+1}$ 
7       while  $E \subset C_{i+1}$  do                                $E$  é o ponto contido ou não no círculo corrente
8           Rotacionar  $J_i$ 
9       end
10  end
11 fim

```

O algoritmo (1) percorre cada junta e durante cada iteração, a junta J_i verifica se o ponto E está fora ou dentro da sua circunferência. Caso esteja dentro, a junta J_i verifica se o ponto E está fora da circunferência da junta J_{i+1} . Se sim, rotaciona todos os segmentos e juntas filhas de forma que ponto objetivo E seja pertencente à circunferência da junta seguinte (C_{i+1}). Caso o ponto E esteja dentro da circunferência da junta J_{i+1} significa que a junta J_i não necessita de rotação.

Figura 13 – Etapas do processo atuando para posicionar a extremidade do braço no ponto E



Fonte: Criado pelo autor

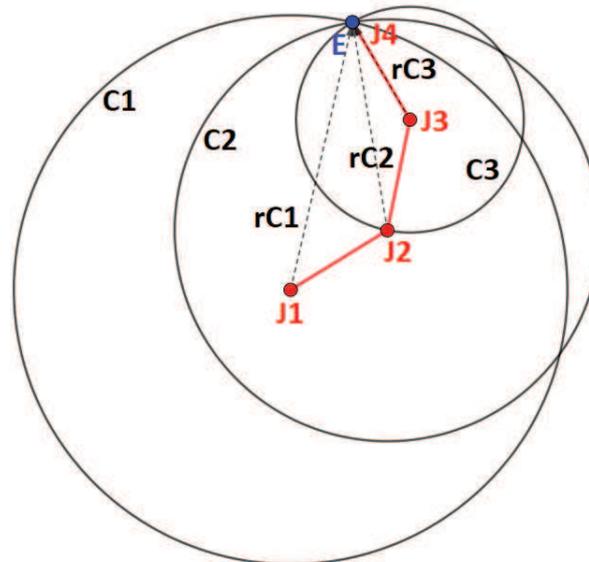
A partir disso, uma vez que a extremidade do braço é a intersecção de todas as circunferências, cada junta busca ajustar a circunferência da junta seguinte de forma que ela compreenda o ponto objetivo. Na Figura 13 é possível observar o algoritmo atuando para posicionar a extremidade do braço no ponto E . Na primeira etapa a junta J_1 verifica que o ponto E está dentro da circunferência C_1 e fora da circunferência C_2 , sendo assim, na Etapa 2 ele é rotacionado até

que a circunferência C_2 esteja sobre o ponto E . Após o mesmo procedimento para as juntas subsequentes, até atingir objetivo E , representada pela Etapa 3.

Vale notar, que uma vez que o ponto E esteja em uma circunferência C_i , seria necessário somente a rotação da junta J_{n-1} , para alcançar o objetivo, uma vez que a própria circunferência C_i representa todos os pontos alcançáveis através da rotação da junta J_i . Foi decidido realizar cada iteração da forma a distribuir os ângulos a serem utilizados para as juntas. Porém, essa questão deve ser melhor analisada para verificar as possíveis soluções geradas para cada caso.

Após o algoritmo percorrer todas as juntas com exceção da extremidade, as circunferências de todas as juntas estarão se interseccionando no ponto objetivo faltando apenas rotacionar a última de forma que a extremidade se encontre com o ponto E , mostrado na Figura 14.

Figura 14 – Circunferências interseccionadas sobre o ponto efetor E



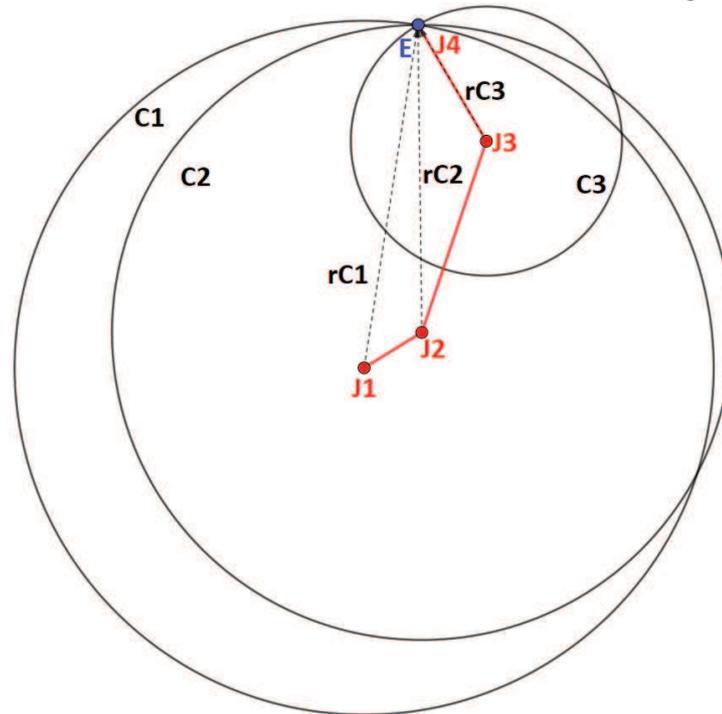
Fonte: Criado pelo autor

Vale notar, que durante a iteração, para que a junta posicione a circunferência da próxima junta de forma que o ponto E esteja sobre ela, haverá sempre duas opções, podendo a junta ser rotacionada positivamente, alcançando o ponto E com a parte de cima da circunferência, ou rotacionando negativamente, alcançando com a parte de baixo da circunferência. Tendo em vista realizar um número mínimo de rotações, a junta rotaciona o menor ângulo possível. A ideia de minimização de rotações está em teste para a confirmação deste resultado.

O modelo proposto também busca evitar colisões de partes do articulado com o ponto objetivo. Dado a possibilidade de ter seus elos com tamanhos distintos, como por exemplo na Figura 15. A abordagem proposta busca minimizar rotação em cada junta, mantendo o ponto E o mais distante possível do braço visto que a cada iteração busca-se posicionar a circunferência C_n sob o ponto E .

É possível perceber que o algoritmo pode percorrer as juntas também de trás pra frente, invertendo a hierarquia ou utilizando somente partes da hierarquia. Sendo assim, é possível

Figura 15 – Articulado com diferentes tamanhos de elos e estrutura do algoritmo consistente



Fonte: Criado pelo autor

que a extremidade de um braço seja dinâmica e aplicada à estruturas articuladas de diferentes complexidades. A partir disto e a direção da origem do movimento, cria-se a possibilidade de escolha da junta raiz dentre todas as juntas da figura articulada, nas quais todas as outras serão filhos da junta escolhida. Isto proporciona maiores possibilidades de movimentação das articulações como mostrado na Figura 16.

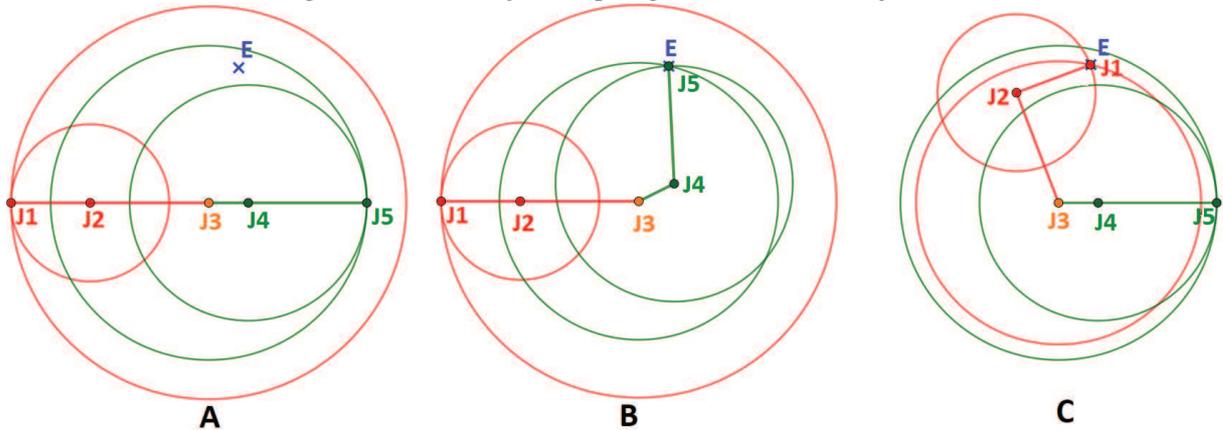
No exemplo da Figura 16, representada pela estrutura Observa-se que a junta $J3$ foi a escolha para ser a junta raiz como também a junta atual J_i . Dessa forma, o exemplo possuirá duas extremidades, ou seja, dois *end effectors*. A estrutura de circunferências seguirá da mesma forma representada anteriormente, possibilitando duas possíveis soluções. Note que a localização do ponto E (ponto objetivo) criou a possibilidade da solução B em verde, com menor distância até o ponto E , por outro lado, também a possibilidade da solução C em vermelho.

4.1 Modelo 2D

O modelo aqui proposto é uma nova abordagem para cinemática inversa, em outras palavras, uma técnica a ser explorada. Esta seção apresenta a construção matemática para calcular os ângulos das articulações em um espaço bidimensional.

O estudo dos ângulos segue o algoritmo desenvolvido pelas circunferências. Desta forma, é o estudo local da junta, iniciando-se pela junta de maior raio de circunferência e seguindo para as menores. Dado uma figura articulada com 3 juntas (J_1, J_2, J_3) e com estrutura do algoritmo

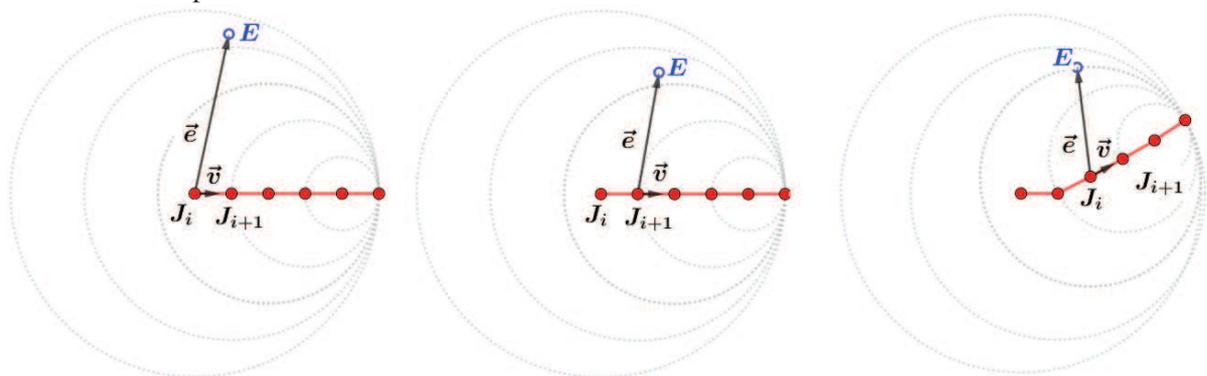
Figura 16 – Mudança da topologia da estrutura das juntas



Fonte: Criado pelo autor

proposto, o cálculo da junta inicia-se pela J_1 , ou seja, a junta atual J_i a sofrer angulação e J_2 a junta adjacente. Realizada a rotação na junta J_1 , a junta J_2 é a nova junta atual J_i e a junta J_3 a nova junta adjacente. Vale ressaltar que esse processo é recursivo para n juntas de uma figura articulada.

A partir da Figura 17, apresenta como notação: J_i a junta atual a sofrer inclinação; J_{i+1} a junta adjacente a J_i ; J_n a última junta, ou seja, a extremidade da figura articulada; E -end effector, ponto objetivo; \vec{e} o vetor criado partindo do J_i até o ponto E ; \vec{v} , o vetor auxiliar unitário para o cálculo do ângulo entre os vetores \vec{v} e \vec{e} . Dado que o vetor \vec{v} possui como sentido J_i para J_{i+1} e direção a configuração anterior da figura articulada antes de exercer a inclinação para o seu objetivo posterior.

Figura 17 – Representação da nomenclatura utilizada para o estudos dos ângulos gerados pela figura articulada até o ponto E 

Fonte: Criado pelo autor

Com base na Figura 18, são apresentados abaixo as fórmulas para obtenção dos ângulos α e φ , tendo como objetivo a subtração destes ângulos que será o ângulo exercido pela junta J_i . Dado a notação $d(J_i, J_{i+1})$, a distância entre a junta atual e a junta adjacente e teremos a mesma

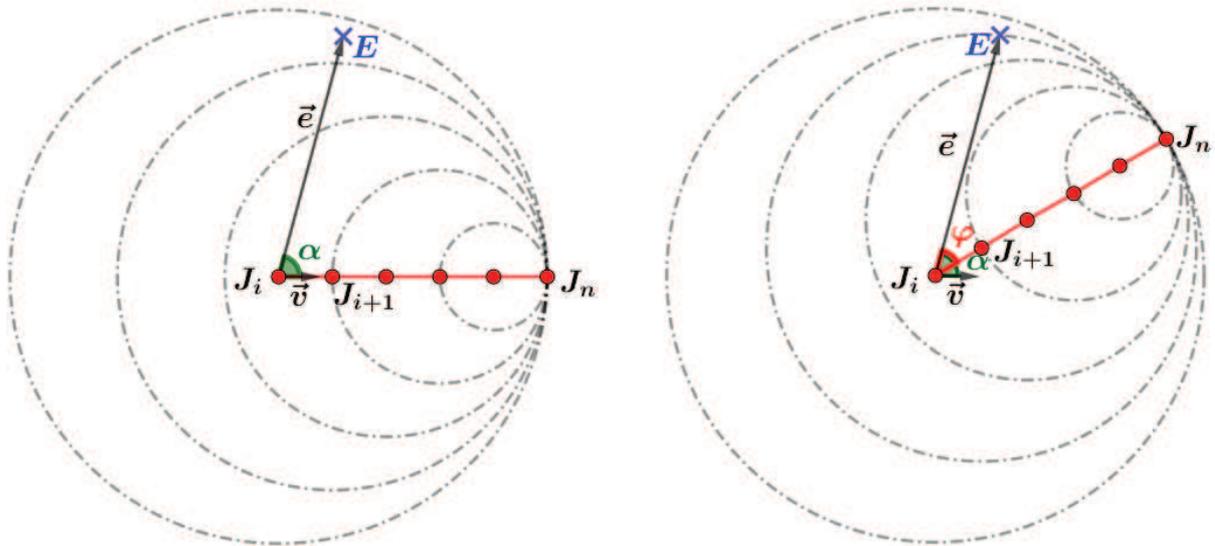
lógica para as representações subsequentes.

$$\varphi = \arccos \left(\frac{d(J_i, E)^2 + d(J_i, J_{i+1})^2 - d(J_i, J_n)^2}{2d(J_i, E)d(J_i, J_{i+1})} \right) \quad (4.1)$$

$$\alpha = \arccos \left(\frac{\langle e, v \rangle}{|e||v|} \right) = \arccos \left(\frac{x_e x_v + y_e y_v}{\sqrt{x_e^2 + y_e^2} \sqrt{x_v^2 + y_v^2}} \right) \quad (4.2)$$

Os cálculos dos ângulos foi separado entre o ângulo φ gerado pelo o elo de ligação das juntas J_i e J_{i+1} com o vetor \vec{e} , dada pela Equação 4.1, utilizando a lei dos cossenos, e pelo ângulo α gerado pelo angulação do vetor \vec{v} como o \vec{e} , dada pela Equação 4.2. Calculado estes dois ângulos, calcula-se o ângulo θ que resulta da diferença de α com φ ($\theta = \alpha - \varphi$). Os ângulos θ representam o quanto a figura articulada precisará se angular para atingir o *end effector* até o ponto E .

Figura 18 – Representação angular das juntas, seguindo a estrutura geométrica.



Fonte: Criado pelo autor

4.2 Modelo 3D

Esta seção apresenta a construção do modelo tridimensional, que por sua vez é a aplicação direta do modelo bidimensional e do algoritmo recursivo. No espaço tridimensional é perdido a ideia de orientação trivial, portanto cada junta terá que ser representada por um conjunto de dois ângulos. Os ângulos formados são constituídos pelo par de ângulos de cada junta. Para melhor entendimento e clareza das próximas notações os eixos coordenadas (xzy) , possuem orientações y como profundidade (formam 90° com os outros dois eixos) os eixos x e z são eixos nos quais todos os seus pontos são colineares, no caso, pertencem ao mesmo plano.

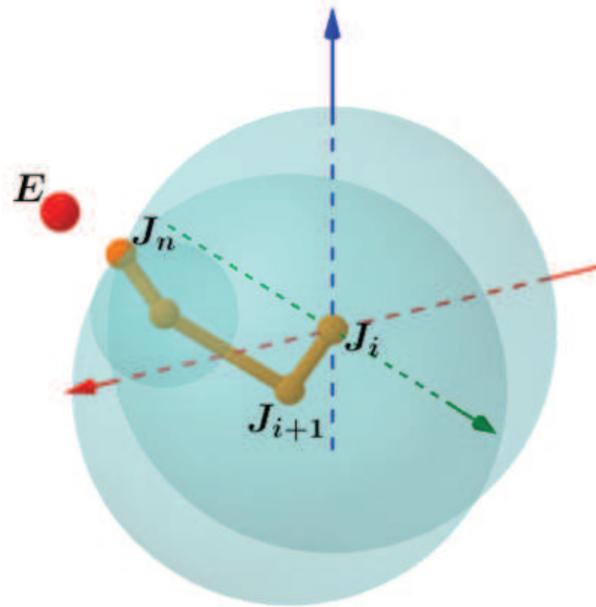


Figura 19 – Representação do modelo 3D seguindo a mesma estrutura de construção do modelo 2D

O modelo 3D segue a mesma estrutura de construção de juntas e hierarquia do modelo 2D, como representado pelas figura 19 e 20.

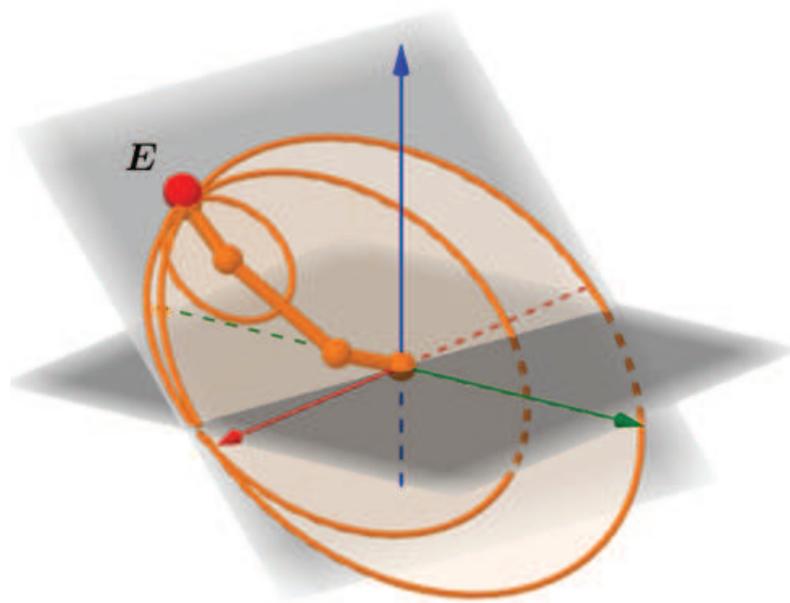


Figura 20 – Formação de planos de orientação para os cálculos angulares.

Para a formulação de um dos ângulos, é iniciada a construção do primeiro plano formado pelos pontos J_i , J_{i+1} e o ponto objetivo E , figura 21. O segundo plano é formado pela base da figura articulada, como por exemplo a primeira junta J_i pertencente ao plano xz , o ângulo formado por esse dois planos é justamente o primeiro par de ângulos com relação a rotação com ao eixo y .

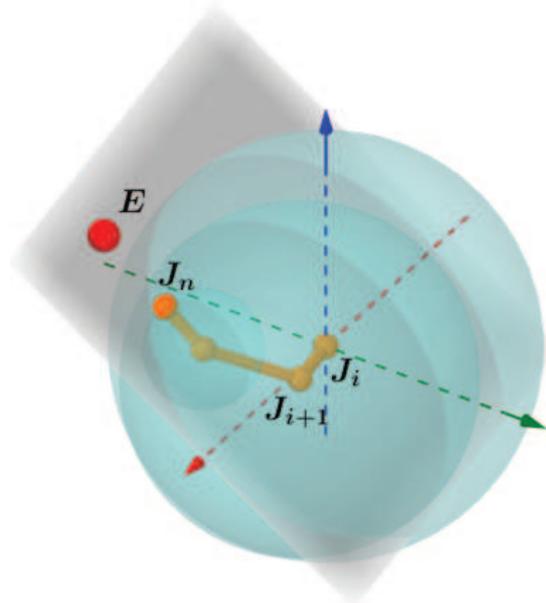


Figura 21 – Plano gerado pelos pontos J_i , J_{i+1} e E .

O cálculo do ângulo formado por esses dois planos é formado a partir da criação de dois vetores ortonormais (vetores que formam 90° com relação ao seu respectivo plano e unitário) em um dos pontos de intersecção entre os planos. Vamos tomar como nomenclatura o plano formado pelos 3 pontos de P_E , o plano xz de P_{xz} e os vetores ortonormais de N_{P_E} e $N_{P_{xz}}$. Diante disso, o ângulo formado pelos planos é definido pelos ângulos formados pelos vetores ortonormais que é dado pelo ângulo ψ .

$$\psi = \arccos \frac{|N_{P_E} \cdot N_{P_{xz}}|}{|N_{P_E}| \cdot |N_{P_{xz}}|} \quad (4.3)$$

O segundo ângulo para completar o par de ângulos da articulação em estudo é calculado no plano P_E seguindo o mesmo cálculo da estrutura e hierarquia do modelo 2D. Vale notar que tanto o modelo 2D quanto o 3D seguiram a estrutura do algoritmo 1, a diferença esta que a análise muda de círculos para esferas em causa da mudança espaço dimensional.

5 RESULTADOS DA ANÁLISE DO MODELO

Este capítulo apresenta, compara e discute os resultados dos métodos de cinemática inversa, conforme apresentados e utilizados no Capítulo 3 pelos autores, sob diferentes condições de análise. Cada metodologia é implementada em diversas situações de pontos destinos unitários e múltiplos, testados e analisados com o uso da Distância de Hausdorff e do custo computacional.

5.1 Elementos para o desenvolvimento dos experimentos

Para comparação/validação do modelo proposto neste trabalho, foi criado um conjunto de pontos objetivos. O conjunto de pontos consiste em alvos que se movem de forma contínua e suave no espaço. Os testes também consistem em uma análise percentual dos pontos alcançados por cada método. Os exemplos são demonstrados em cinco métodos para solução de cinemática inversa diferentes com 3 articulações e permitindo todos graus de liberdade. A Figura 22 mostra a dispersão dos pontos objetivos para os diferentes métodos alcançar o ponto objetivo.

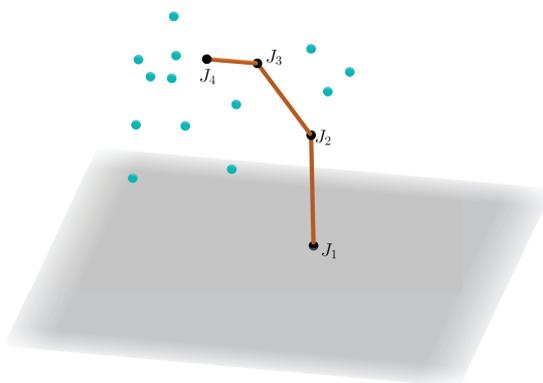


Figura 22 – Amostra do conjunto de pontos utilizado em um dos métodos nos experimentos

Os métodos utilizados pelos autores foram comparados principalmente com posições e orientações especificadas pelo animador e analisadas através do rastro do *end-effector* de sua posição inicial até alcançar os pontos objetivos. Os parâmetros de análise foram norteados pelo trabalho de Arsitidou et.al que em seu trabalho realiza o estudo comparativo de modelos de cinemática inversa (ARSITIDOU; LASENBY, 2009). Todos os experimentos foram executados e implementados usando MATLAB em um computador com um processador Intel Core i5 2.4 GHz. O sistema operacional usado foi o Microsoft Windows 10 Enterprise com 8 Gb de Memória Ram.

5.1.1 Distância de Hausdorff

A medição de similaridade entre dois conjuntos de dados em computação é uma ação fundamental para determinar desempenho e precisão computacional. O problema de medir o quanto dois conjuntos são similares é um desafio aos programadores, quando considerado fatores como eficiência na medida e custo computacional (GONÇALVES; MITISHITA, 2016).

A Distância de Hausdorff (DH) é definida para mensurar o quanto dois conjuntos de pontos em um espaço cartesiano são similares. As medidas de distâncias entre conjuntos podem ser aplicadas por diferentes formas, como a distância euclidiana das médias dos elementos de um conjunto. Porém, a DH apresenta maior exatidão na maioria dos casos de similaridade, dado que foi demonstrado pelo matemático Hunttenlocher (HUTTENLOCHER; KEDEM, 1990).

Dado dois conjuntos $A = a_1, a_2, \dots, a_n$ e $B = b_1, b_2, \dots, b_n$ com números finitos de pontos e d a distância Euclidiana a definição da distância de Hausdorff é dada como:

$$h(A, B) = \max_{a \in A} \{ \min_{b \in B} \{ d(a, b) \} \} \quad (5.1)$$

$$h(B, A) = \max_{b \in B} \{ \min_{a \in A} \{ d(b, a) \} \} \quad (5.2)$$

$$H(A, B) = \max \{ h(A, B), h(B, A) \} \quad (5.3)$$

O algoritmo para a métrica de DH tem formulação simples:

1. A partir de um dos elementos de A calcula as distâncias a todos elementos de B , ao fim registra o menor valor entre todos os calculado;
2. O passo anterior é repetido para todos os demais elementos de A ;
3. Determina-se o maior valor.

A equação 5.1 representa a maior das menores distâncias registradas que distanciam os pontos de A a qualquer outro ponto no conjunto de B . Podemos notar que a equação 5.2 possui o processo idêntico, porém considerando o sentido inverso para o cálculo das distâncias. Por fim, a equação 5.3 é a DH, e será a maior entre as duas $h(A, B)$ e $h(B, A)$.

A distância de Hausdorff também esta presente no reconhecimento de padrões e visão computacional, e é muito importante comparar modelos e técnicas, para dar um valor numérico indicando sua similaridade e precisão de métodos computacionais para problemas reais.

5.2 Resultados

Nesta seção, o modelo proposto foi avaliado em relação a alguns dos métodos usuais de cinemática inversa, como a matriz jacobiana, algoritmos genéticos, redes neurais e pseudo-

inversa Jacobiana.

Em alguns dos experimentos, foi implementado exemplos com distâncias significativas do *end-effector* para o ponto objetivo. Dessa forma, alguns métodos tendem exigir uma maior quantidade de iterações para atingir o ponto objetivo e a comparação da diferença entre os métodos é mais simples. Diversos testes e comparações foram implementados em relação ao seu custo computacional, tempo de processamento, número de iterações utilizadas para atingir o ponto objetivo e a precisão do movimento através da Distância de Hausdorff.

5.2.1 Pontos objetivos unitários

Nesta seção, os métodos foram testados com um único ponto objetivo e posições de destino fixa. Esses experimentos não incluíram nenhuma restrição nas articulações, mas todos os métodos poderiam ser aprimorados para criar limites e gerar uma análise de resultados restrita. Na figura 23, observa-se dois exemplos de configurações dos métodos para alcançar o ponto objetivo.

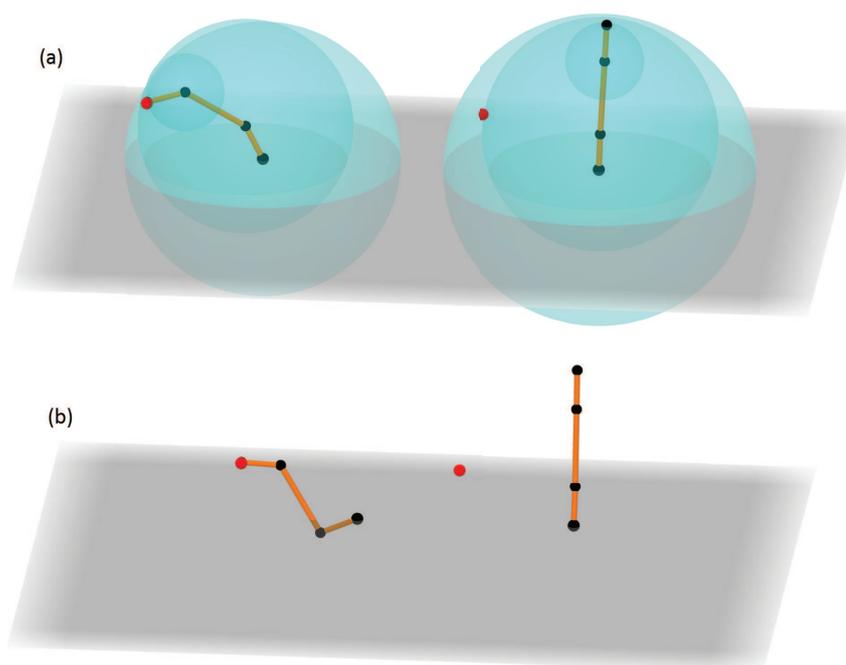


Figura 23 – Comparação de configuração para alcançar o ponto objetivo, (a) Modelo proposto e (b) matriz Jacobiana.

O modelo proposto apresenta resultados significativos de resposta comparados a outros métodos de cinemática inversa. É aproximadamente 500 vezes mais rápido do que os métodos que utilizam a Jacobiana. Além disso, o modelo apresentado neste trabalho possui menor custo computacional e produzindo visualmente uma movimentação mais suave. Em média, o modelo proposto necessita de 17,34 iterações e apenas 11,89ms para chegar ao ponto objetivo.

Os trabalhos do estado da arte apresentados no Capítulo 3 para solução de cinemática inversa em suma todos tiveram a aplicação da Jacobiana em seus métodos, como exemplo o trabalho de

Srisuk et.al (SRISUK; KITJADURE, 2017) que utilizou a matriz jacobiana para o treinamento de sua rede neural. Diante disso, analisaremos mais profundamente o método da Jacobiana.

Os métodos que utilizam da Jacobiana, ou apenas da matriz, retornam resultados razoáveis. Porém, a maior vantagem de se utilizar este método sobre outras técnicas é que eles podem tratar o Jacobiano para resolver problemas mais realistas e precisos. Algoritmos jacobianos podem ser aplicados sobre restrições, mas a forma que essas restrições são colocadas nos problemas não é trivial. Além disso, a Jacobiana possui o problema de singularidade em seus cálculos de inversão das matrizes.

Os métodos por matriz Jacobianas possuem alto custo computacional, logo não sendo adequado para problemas em tempo real. No estudo atual e as restrições colocadas, essa técnica necessita em média de 1145 iterações em 12s para atingir o ponto objetivo desejado. Esse método ainda geralmente converge lentamente para seus pontos objetivos, pois por ser um modelo não-linear, realiza aproximações lineares com pequenas passagens de aplicação.

A tabela 2 apresenta as quantidades de iterações necessárias para alcançar os pontos objetivos e tempos de execução para cada método apresentado pelo estado da arte no Capítulo 3. Os tempos para execução dos métodos estão na métrica segundos e foram mensurados através de um código MATLAB em um processador Core Intel i5 2,4 GHz. Uma iteração no modelo proposto possui menor custo computacional, dado que uma vez conhecidas a localização das juntas e os tamanhos que as ligam o cálculo é realizado sobre um plano no espaço cartesiano tridimensional utilizando cálculo de geometria bidimensional da formação de um triângulo através da geração de dois vetores e reduzindo o a dimensionalidade do espaço.

Tabela 2 – Valores aproximados dos métodos utilizados pelos autores para alcançar o ponto objetivo.

	Quantidade de Iterações	Tempo (s)
Jacobiana e Rede Neural	293,0921	2,38472
D-H, <i>fsolve</i> e Jacobiana	646,8272	5,2918
Algoritmos Genéticos e Jacobiana	872,1902	7,9812
Inversa da Jacobiana	1645,9387	14,2983
Jacobiana e Modelo do autor	172,918	1,0185
Modelo Proposto	7,9283	0,0183

Vale salientar que métodos que utilizam a matriz Jacobiana podem por sua vez produzir movimentos oscilantes e não contínuos. O tempo necessário para encontrar o ponto objetivo é uma das piores características, apenas sobre a utilização de bibliotecas específicas e programação utilizando C ou C++, tornam possíveis a aplicação em tempo real, mas isso não indica necessariamente em uma melhora significativa.

O modelo proposto em teste com diferentes pontos objetivos apresentou minimização nos ângulos das articulações e não possui colisões dos segmentos com o ponto objetivo. Com comparação aos outros métodos o modelo aqui estudado apresenta suavização para alcançar o ponto objetivo e uma redução na angulação de 8,7632%. Para verificar essa afirmação foi realizado diferentes processos de iterações da figura articulada para chegar ao ponto objetivo.

5.2.2 Pontos objetivos múltiplos

A grande parte de modelos reais e industriais, como um manipulador robótico possuem vários pontos objetivos a serem alcançados em espaço de tempo. Desta forma, existe a importância de testar os métodos apresentados pelos autores em comparação com o modelo aqui proposto.

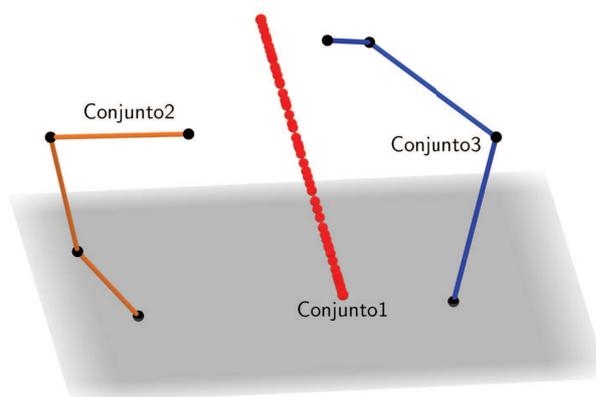


Figura 24 – Representação de construção para o estudo comparativo de similaridade para ser aplicado à métrica de Hausdorff.

Para comparação entre os trabalhos são escolhidos três conjuntos de pontos, como ilustrado na figura 24. O Conjunto 1 é um segmento de reta, ou seja, um conjunto de pontos objetivos finitos a serem alcançados pelos métodos de cinemática inversa. Por sua vez, o Conjunto 2 é o grupo de pontos formados pelo rastro do *end-effector* do método proposto neste trabalho de um ponto ao outro do segmento. O Conjunto 3 é formado igual ao segundo, porém o método utilizado é uma das técnicas a serem comparadas.

A métrica de Hausdorff foi desenvolvida e aplicada nesse capítulo com a ideia de mostrar o quanto um método pode ser eficaz, robusto e similar a um conjunto de pontos ideais. Para isso utilizaremos da visualização bidimensional para os três conjuntos de pontos e de uma porcentagem de similaridade diante dos resultados da DH.

Visualmente podemos observar que (c) , (f) e (g) ocorreu um distúrbio no fim de suas últimas iterações para alcançar os pontos objetivos, dado que os métodos geraram os rastros de baixo para cima. Também podemos visualizar que (f) e (g) possuem maior distúrbio nos pontos se comparados aos outros modelos. Entretanto, (a) e (e) modelos propostos pelos seus respectivos autores possuem maior robustez na geração do seu rastro quando comparado com (b) .

A partir do algoritmo da Distância de Hausdorff foi possível criar um ambiente de validade de similaridade e comparação de modelos. Nas tabelas 3 e 2 podemos observar que quando a Jacobiana esta realizando uma tarefa de cinemática inversa seus parâmetros de similaridade, iterações, taxa de erro e tempo de operação não são eficientes computacionalmente. Apesar da Jacobiana não apresentar grandes resultados quando utilizada sozinha, em soluções conjuntas

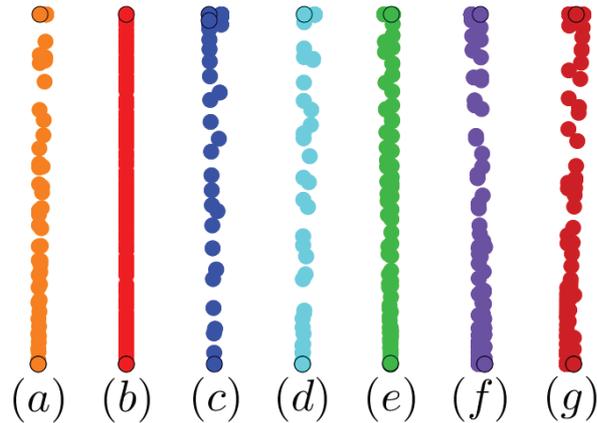


Figura 25 – Visualização dos dados de similaridade a partir de DH. (a) Rastro gerado pelo modelo proposto; (b) Conjunto de pontos objetivos a serem comparados; (c) Jacobiana e Rede Neural ; (d) D-H, *fsolve* e Jacobiana; (e) Jacobiana e Modelo do autor; (f) Inversa da Jacobiana; (g) Algoritmos Genéticos e Jacobiana.

Tabela 3 – Valores aproximados de similaridade dos métodos utilizados pelos autores para alcançar o ponto objetivo.

	Porcentagem de Similaridade	Taxa de erro
Jacobiana e Rede Neural	81,9128%	18,0872%
D-H, <i>fsolve</i> e Jacobiana	78,1739%	21,8261%
Algoritmos Genéticos e Jacobiana	79,6380%	20,3620%
Inversa da Jacobiana	76,2309%	23,7691%
Jacobiana e Modelo do autor	89,5693%	10,4307%
Modelo Proposto	91,2350%	8,7650%

com outras técnicas apresenta melhora nos resultados. Em contrapartida, o modelo proposto apresentou resultados de eficiência e aplicabilidade em problemas reais, também como de precisão em movimentos em tempo real.

O modelo proposto é um modelo inovador para solução de problemas aplicados a cinemática inversa. Além disso, é um modelo linear e sua eficiência computacional é notável. Nos testes aqui desenvolvidos o modelo apresentado, desenvolveu o menor de número de iterações para atingir o ponto objetivo. Da mesma forma que a sua taxa de similaridade atingiu 91,2350%, dado um conjunto ideal de pontos.

6 CONCLUSÃO E TRABALHOS FUTUROS

Os aplicação dos métodos de cinemática inversa estão concentrados em sua maioria nas áreas de Computação Gráfica e Robótica. No entanto, grande parte dos métodos de cinemática inversa sofrem com alto custo computacional e precisão. Além disso, foi verificado que a não-linearidade da técnica da jacobiana e o cálculo de sua inversa (abordagem mais usada para animação por cinemática inversa) causa impedimentos em sua aplicação.

Este trabalho apresenta uma proposta de modelo geométrico com hierarquia de movimento não rígida para solucionar a cinemática inversa, com baixo custo computacional e precisão desejadas para métodos que são aplicados acerca desse tema.

O modelo proposto apresentou um novo método de solução de cinemática inversa ainda não abordado. O algoritmo usa um cálculo geométrico simples, rápido, com baixo custo computacional e com uma taxa de precisão superior aos trabalhos relacionados. Os métodos já existentes, quando comparados, apresentaram uma melhora na utilização da Jacobiana quando utilizada em conjunto com outras técnicas. Logo, o método de utilização unicamente pela Jacobiana não se mostrou eficaz, nem no custo computacional, nem na precisão do *end-effector* para alcançar os pontos objetivos. Com relação a precisão desse método, pode-se analisar que a singularidade para alcançar o ponto objetivo, que é causada pela inversão da matriz jacobiana, causa danos no algoritmo a cada iteração.

Como trabalhos futuros, o objetivo é seguir no estudo de resultados de comparação com as demais técnicas de cinemática inversa. Apesar do modelo possuir hierarquia não rígida de movimento, não foi estudado nenhum tipo de aplicação. O modelo apresentou diversas melhorias em comparação com outros trabalhos da temática de cinemática inversa, porém ainda existe a necessidade da realização de uma aplicação real em um modelo de simulação robótica ou industrial para estudar a mudança de hierarquia e os ganhos que essa estrutura pode ocasionar nos resultados finais.

Também como complemento para trabalhos futuros a aplicação de uma meta-heurística no estudo do *end-effector*, com a proposta de evolução do modelo proposto para determinar as variações angulares a serem aplicadas nas articulações das figuras articuladas de forma a posicionar corretamente no espaço.

REFERÊNCIAS

- ARSITIDOU, A.; LASENBY, J. Inverse Kinematics a review of existing techniques and introduction of a new fast iterative solver. **University of Cambridge**, [S.l.], 2009.
- BROOKSHEAR, J. G. **Computer science: a comprehensive view**. São Paulo: Bookman, 2013.
- CORKE, P. **Robotics, vision and control: fundamental algorithms in matlab**. [S.l.: s.n.], 2011. v. 73.
- CRAIG, J. J. **Robótica**. [S.l.: s.n.], 2012. v. 3.
- DENAVID J.; HARTENBERG, R. S. A kinematic notation for lower-pair mechanisms based on matrices. **Trans ASME J. Appl. Mech**, [S.l.], p. 215—221, 1955.
- ESCANDE, A.; MANSARD, N.; WIEBER, P.-B. Fast resolution of hierarchized inverse kinematics with inequality constraints. **IEEE International Conference on Robotics and Automation**, [S.l.], p. 3733–3738, 2010.
- GONÇALVES, G. A.; MITISHITA, E. A. O USO DA DISTÂNCIA DE HAUSDORFF COMO MEDIDA DE SIMILARIDADE EM SISTEMAS AUTOMÁTICOS DE ATUALIZAÇÃO CARTOGRÁFICA. **Boletim de Ciências Geodésicas**, [S.l.], v. 22, n. 4, p. 719–735, dec 2016.
- HUTTENLOCHER, D. P.; KEDEM, K. Computing the minimum Hausdorff distance for point sets under translation. **Proceedings of the sixth annual symposium on Computational geometry - SCG '90**, [S.l.], p. 340–349, 1990.
- IFR. **Brochure of history of industrial robots – from the first installation until today**. [S.l.]: Bookman, 2014.
- KHUDHER, D.; POWELL, R. Quadratic programming for inverse kinematics control of a hexapod robot with inequality constraints. In: INTERNATIONAL CONFERENCE ON ROBOTICS: CURRENT TRENDS AND FUTURE CHALLENGES (RCTFC), 2016., 2016. **Anais...** [S.l.: s.n.], 2016. p. 1–5.
- LATOMBE, J. **Robot motion planning. the springer international series in engineering and computer science**. [S.l.: s.n.], 1991. v. 124.
- MACHADO, M. M.; AZEVEDO, J. Trajectory planning of redundant manipulators using genetic algorithms. **Commun Nonlinear Sci**, [S.l.], p. 2858–2869, 2009.
- MACIEL, A. **Biomechanics-based Articulation Models for Medical Applications** . [S.l.: s.n.], 2010. v. 1.
- MACIEL, A.; NEDEL, L.; FREITAS, C. Anatomy based joint models for virtual humans skeletons. **Proceedings of Computer Animation**, [S.l.], p. 110—116, 2002.
- MARTINS, D.; GUENTHER, R. Hierarchical kinematic analysis of robots. **Mechanism and Machine Theory**, [S.l.], p. 497–518, 2003.

MING, C.; HUANG, A. An analysis of some fundamental problems in adaptive control of force and impedance behavior: theory and experiments. **IEEE Transactions on Robotics and Automation**, [S.l.], p. 912—921, 1995.

MISTRY, M. et al. Inverse kinematics with floating base and constraints for full body humanoid robot control. **IEEE-RAS International Conference on Humanoid Robots**, [S.l.], p. 22–27, 2008.

NUNES, R. F. **Mapeamento da cinemática inversa de um manipulador robótico utilizando redes neurais artificiais configuradas em paralelo**. [S.l.: s.n.], 2016.

PARENT, R. **Computer animation: algorithms and techniques**. [S.l.: s.n.], 2002.

PERRUSQUÍA, A. et al. Stable admittance control without inverse kinematics. In: IFAC-PAPERSONLINE, 2017, Elsevier B.V. **Anais...** [S.l.: s.n.], 2017. p. 15835–15840.

SHI, Q.; LVZHONG, M. **Feature simulation of obstacle avoidance problem of six-degree-of-freedom robot**. [S.l.: s.n.], 2010. 89–90 p. v. 9.

SINGH, S.; POPA, D. Adaptive impedance control of robot manipulators based on function approximation technique. **Cambridge University Press**, [S.l.], v. 22, p. 395–403, 2004.

SRISUK, P.; KITJAIIDURE, Y. Forward kinematic-like neural network for solving the 3d reaching inverse kinematics problems. In: IEEE COMPUTER, 2017, Elsevier B.V. **Anais...** [S.l.: s.n.], 2017. p. 214–217.

TEJOMURTULA, S.; KAK, S. Inverse kinematics in robotics using neural networks. **Information Sciences**, [S.l.], v. 116, p. 147–164, 1999.

TOLANI, D.; GOSWAMI, A.; BADLER, N. Real-time inverse kinematics techniques for anthropomorphic limbs. **Graphic Models**, [S.l.], p. 353–388, 2000.

VESLIN, E. Y. et al. A hybrid solution for the inverse kinematic on a seven dof robotic manipulator. **IEEE Latin America Transactions**, [S.l.], v. 12, n. 2, p. 212–218, mar 2014.

VILLAMIL, M. B. **Modelagem e simulação da articulação temporomandibular**. [S.l.: s.n.], 2009.

WANG YONG WANG LEI, X. Y. **Kinematic inverse solution of redundant free-degree robot**. [S.l.: s.n.], 2003. 249–252 p. v. 3.

YU, J.; YOU, D.; LIU, J. Analysis of inverse kinematics method for six degrees of freedom manipulator based on matlab. **IEEE Computer**, [S.l.], p. 211–215, 2017.