

**UNIVERSIDADE FEDERAL DE VIÇOSA
CENTRO DE CIÊNCIAS EXATAS E TECNOLÓGICAS
DEPARTAMENTO DE ENGENHARIA ELÉTRICA**

JORGE AUGUSTO SALVADOR DOS SANTOS

**CONTROLE DE POSIÇÃO DE UM BRAÇO
ROBÓTICO REDUNDANTE DE QUATRO GRAUS DE
LIBERDADE UTILIZANDO ALGORITMOS
GENÉTICOS**

**VIÇOSA
2010**

JORGE AUGUSTO SALVADOR DOS SANTOS

**CONTROLE DE POSIÇÃO DE UM BRAÇO
ROBÓTICO REDUNDANTE DE QUATRO GRAUS DE
LIBERDADE UTILIZANDO ALGORITMOS
GENÉTICOS**

Monografia apresentada ao Departamento de Engenharia Elétrica do Centro de Ciências Exatas e Tecnológicas da Universidade Federal de Viçosa, para a obtenção dos créditos da disciplina ELT 490 - Monografia e Seminário e cumprimento do requisito parcial para obtenção do grau de Bacharel em Engenharia Elétrica.

Orientador: Prof. M.Sc. Alexandre Santos Brandão.

VIÇOSA
2010

JORGE AUGUSTO SALVADOR DOS SANTOS

**CONTROLE DE POSIÇÃO DE UM BRAÇO ROBÓTICO
REDUNDANTE DE QUATRO GRAUS DE LIBERDADE
UTILIZANDO ALGORITMOS GENÉTICOS**

Monografia apresentada ao Departamento de Engenharia Elétrica do Centro de Ciências Exatas e Tecnológicas da Universidade Federal de Viçosa, para a obtenção dos créditos da disciplina ELT 490 - Monografia e Seminário e cumprimento do requisito parcial para obtenção do grau de Bacharel em Engenharia Elétrica.

Aprovada em 02 de dezembro de 2010.

COMISSÃO EXAMINADORA

Prof. M.Sc. Alexandre Santos Brandão - Orientador
Universidade Federal de Viçosa

Prof. Dr. Tarcísio de Assunção Pizziolo - Membro
Universidade Federal de Viçosa

Prof. Dr. Leonardo Bonato Felix - Membro
Universidade Federal de Viçosa

À minha família, pela oportunidade única, paciência e amor incondicional.

Agradecimentos

Ao Prof. M.Sc. Alexandre Santos Brandão (Timótiu), meu orientador, minha profunda gratidão por inúmeras vitórias alcançadas. Por me preparar para o futuro com desafios, injeções de ânimo e algumas retroalimentações negativas. A oportunidade de participar das JAR e conhecer pessoas fantásticas e aprender algumas lições valiosas (“*Números, eu quero números*”). Nunca esquecerei de uma conversa pela internet em que após uma falha minha disse: “*Perdão se lhe faltei com respeito, professor.*” e então como resposta, “*Que isso, não se preocupe, quero ser mais que seu professor, quero ser seu amigo*”. Desta brincadeira surgiu uma amizade que aprecio muito e pode deixar, chefia, que eu confio. Hoje sou o desenho que se formou de um comando.

A República Entrometeu, por transformar uma casa em um lar. Nesses 5 anos de convívio agradeço muito aos almoços do Elbinho, as cervejas com tira gosto do D2, aos drinks inovadores do Magro e a cara de desespero do Talles.

Aos colegas da primeira geração NERO por sempre fazerem uma grande festa naquela pequena sala.

Aos sobreviventes de 2005, pela força, coragem e determinação mostrado no decorrer dos últimos anos.

A todos os companheiros de 2006, principalmente ao Dudu e a Priscilla. Dudu por ser sempre disponível a ajudar e ser ajudado. A Priscilla que nos últimos 4 anos se mostrou uma ótima colega de classe e uma companheira maravilhosa, dengosa e muito nervosa.

Aos caras de 2007, que na sua simplicidade e nerdisse mostraram seu companheirismo.

Aos meus pais, Luiz Carlos e Rosângela, pela paciência, compreensão, apoio e por sempre acreditarem na minha capacidade. As minhas irmãs, Giovana e Cybelle, por sempre cuidarem de mim e de nossos pais. Ao meu irmão Ronaldo, por me fazer sorrir e compartilhar momentos inesquecíveis em suas visitas. Amo vocês.

“Escolhe um trabalho de que gostes, e não terás que trabalhar nem um dia na tua vida.”

Confúcio

Resumo

A palavra robô sempre desperta a curiosidade e o interesse de pessoas que se enquadram em qualquer faixa etária, afinal é desejo irrefutável ter um dispositivo reprogramável que seja capaz de realizar tarefas perigosas e monótonas do cotidiano. O aspecto interdisciplinar da robótica propicia a aplicação dos estudos específicos de engenharia em situações práticas, como, por exemplo, visão computacional, inteligência artificial, sistema de controle, eletrônica, dentre outras. Neste contexto, o desenvolvimento de projetos na área de robótica é uma ferramenta de fixação e desenvolvimento do conhecimento na engenharia. O presente projeto de monografia apresenta o desenvolvimento de um simulador de um manipulador robótico redundante de quatro graus de liberdade (DOF - Degrees of Freedom). O simulador propicia uma visualização tridimensional do movimento do braço, recebendo os ângulos das articulações como parâmetros de entrada. Através da utilização dos sistemas de controle, a movimentação instantânea do manipulador simulado comporta como um manipulador real com todas as suas limitações físicas que definem seu espaço de trabalho. Quanto maior a fidelidade entre o modelo simulador e o real, melhores serão os resultados das tarefas a serem executadas, tais como manipulação de objetos em meios industriais, residenciais e, até mesmo, militares. Para isto, será obtido o modelo do manipulador, contemplando as equações da cinemática direta e inversa, para a primeira será utilizada a notação de Denavit-Hartenberg que contempla a rotação e a translação juntas com a posição facilitando a movimentação do simulador e para a inversa será utilizado Algoritmo Genético devido o fato de o braço robótico ser redundante (isto é, possui infinitas soluções). O simulador a ser desenvolvido é classificado como online, pois é possível modificar a posição dos objetos na cena de navegação, alterar os parâmetros do sistema, enquanto a simulação transcorre, além de possibilitar um controle por periféricos.

Abstract

The word robot always arouses curiosity and interest of people who fall into any age group, because it is compelling desire to have a reprogrammable device which is capable to perform dangerous and monotonous daily tasks. The interdisciplinary aspect of robotics provides specific engineering studies in practical situations, for example, computer vision, artificial intelligence, control systems, electronics, among others. In this context, the development of projects in robotics is a tool for securing and development of knowledge in engineering. This manuscript presents the development of a simulator of a four degrees of freedom redundant robotic manipulator. The simulator provides a three-dimensional view of arm movement, after receiving the joint angles as input parameters. When control systems are used, a simulated manipulator behaves like a real manipulator, considering its physical limitations which define its workspace. The higher the fidelity between the model simulation and the real, the better the results of the tasks to be performed, such as manipulation of objects in industrial, residential and even military. In such context, this work proposes the development of robotic simulator, considering the equations of forward and inverse kinematics using the Denavit-Hartenberg notation. Genetic algorithm is used to determine the inverse kinematic of the manipulator due to its redundant characteristics (i.e., infinite solutions). The simulator developed is classified as online one, due to the possibility to modify the position of objects in the scene of navigation, to change the system parameters and to provide a control for peripherals.

Sumário

Lista de Figuras

Lista de Tabelas

1	Introdução	12
1.1	Robótica	12
1.2	Manipuladores Robóticos	13
1.2.1	Manipulador Articulado (RRR)	15
1.2.2	Manipulador Esférico (RRP)	15
1.2.3	Manipulador SCARA (RRP)	15
1.2.4	Manipulador Cilíndrico (RPP)	16
1.2.5	Manipulador Cartesiano (PPP)	16
1.2.6	Manipulador Paralelo	17
1.3	Cinemática Direta	17
1.4	Cinemática Inversa	22
1.5	Algoritmo Genético	23
1.5.1	Representação	25
1.5.2	Inicialização da População	25
1.5.3	Validação	26
1.5.4	Seleção	26
1.5.5	Recombinação	27
1.5.6	Mutação	28

1.5.7	Parâmetros dos AGs	29
1.6	Objetivos	29
2	O Simulador	31
2.1	Criação da Interface	32
2.2	Rotação e Translação	34
2.3	Funções do Algoritmo Genético	36
2.3.1	ga_principal	36
2.3.2	ga_geracao(populacao,taxa_mutacao)	37
2.3.3	ga_cruzamento(populacao)	38
2.3.4	ga_mutacao(populacao_filhos)	38
2.3.5	aptidao_manipulador(populacao,pos_des)	38
2.3.6	ga_elitismo(populacao,aptidao,taxa_elitismo)	38
2.3.7	ga_morte (populacao_comum,taxa_mortalidade)	39
2.4	Função de Movimento	39
3	Resultados e Discussões	40
3.1	Simulador	40
3.2	Algoritmo Genético	42
4	Conclusões	47
	Referências	48

Lista de Figuras

1	Representação simbólica das juntas de um robô. A junta rotacional (Revolute) produz rotação sobre os eixos adjacentes. A junta prismática (Prismatic) produz um movimento linear ou telescópico sobre os eixos adjacentes.	14
2	Manipulador Articulado.	15
3	Manipulador Esférico.	16
4	Manipulador SCARA.	16
5	Manipulador Cilíndrico.	17
6	Manipulador Cartesiano.	17
7	Estrutura do Manipulador Paralelo.	18
8	Eixo de coordenadas de um manipulador planar.	18
9	Condição para a utilização de DH.	21
10	Múltiplas soluções para cinemática inversa.	22
11	Método de Seleção por Amostragem Universal Estocástica.	27
12	Cruzamento em um ponto.	28
13	OWI 535 Robotic Arm Edge (vide Apêndice A).	32
14	Padrão de cores RGB.	33
15	Imagem do simulador desenvolvido.	34
16	Diagrama de Corpo Livre.	34
17	Simulação valores aleatórios no parâmetro <code>ArmPose</code> .	41
18	Aplicações.	41
19	Espaço de Trabalho.	42
20	Simulação Pais1.	44

21	Simulação Pais2.	45
22	Simulador na posição desejada com os pontos calculados pelo AG.	46

Lista de Tabelas

1	Notação de Denavit-Hartenberg.	21
2	Definições em Algoritmos Genéticos.	24
3	Principais Representações de AG.	25
4	Mutação AG.	28
5	Sistemas RGB e cores básicas do MATLAB.	33
6	Notação de DH para o simulador.	35
7	Primeira população inicial (Pais1).	43
8	Melhores parâmetros para Pais1.	43
9	Dados da Simulação 1.	44
10	Melhores parâmetros para Pais 2.	44
11	Dados da Simulação 2	45

1 *Introdução*

A palavra robô sempre desperta a curiosidade e o interesse de pessoas que se enquadram em qualquer faixa etária, afinal é desejo irrefutável ter um dispositivo reprogramável que seja capaz de realizar tarefas perigosas e monótonas do cotidiano.

O aspecto interdisciplinar da robótica propicia a aplicação dos estudos específicos de engenharia em situações práticas, como, por exemplo, visão computacional, inteligência artificial, sistema de controle, eletrônica, dentre outras. Neste contexto, o desenvolvimento de projetos na área de robótica é uma ferramenta de fixação e desenvolvimento do conhecimento na engenharia.

No presente projeto foi utilizado conceitos de robótica e inteligência artificial para a construção de uma plataforma de simulação.

1.1 *Robótica*

A palavra “robô” tem origem na palavra theca “robota” que significa “trabalho forçado”. Esta expressão foi utilizada pela primeira vez em uma peça teatral de Karel Capek chamada “R.U.R” (*Rossum’s Universal Robots*), encenada em 1921. Desde então os dispositivos que operam com algum grau de autonomia, sobre um controle computacional, é chamado de Robô [1].

Existem várias definições para a palavra robô. Uma das principais definições oficiais vem do *Robotics Industries Association* (RIA):

“Um robô é um manipulador multifuncional reprogramável projetado para mover materiais, peças, ferramentas, ou dispositivos especiais através de variáveis de movimento programadas para a realização de uma variedade de tarefas”. [2]

O elemento principal sobre a definição de um robô é a sua capacidade de ser reprogramável dando características de adaptabilidade e utilidade. O termo robótica foi us-

ado para designar o campo científico em que se aborda os robôs e foi usado pela primeira vez pelo escritor Isaac Asimov. Na obra de ficção “Liar!” (1941), o autor introduziu a ideia das Três Leis da Robótica que mais tarde foi concretizada na obra “Eu, robô” [3], publicada em 1950.

A robótica está fundamentada na junção de duas tecnologias: Teleoperação e Controle Numérico Computadorizado (CNC). A Teleoperação, ou sistema mestre-escravo, foi desenvolvida durante a Segunda Guerra Mundial para a manipulação de materiais radioativos. O Controle Numérico Computadorizado foi desenvolvido devido à necessidade de uma alta precisão nos processos de fabricação e de manufatura [1].

Os primeiros robôs tinham aplicações simples, como descarregar, transferir, empilhar, entre outras. No decorrer dos anos, as aplicações foram se tornando mais complexas, tais como, soldagem, esmerilhamento, montagem, o que gerou a necessidade de sensoriamento externo (visão, força de detecção entre outros) em virtude da interação do robô com o ambiente.

Os robôs podem ser classificados em dois grandes grupos que são móveis e articulares. Um veículo é definido robô móvel, quando possui a capacidade de movimentação autônoma, é equipado com efetadores controlados por um computador embarcado. Eles são classificados pelo ambiente em que navegam (terra, ar, água ou espaço).

A próxima sessão é destinada aos Manipuladores Robóticos, que é o enfoque desse trabalho.

1.2 Manipuladores Robóticos

Manipuladores robóticos são comumente dispositivos compostos por membros conectados por juntas em uma cadeia cinemática aberta. As juntas são responsáveis pela conexão entre dois membros e podem ser do tipo rotacional ou prismática. A junta rotacional é como uma dobradiça que permite o movimento angular entre dois membros e é representado por R. A junta prismática permite a variação linear no comprimento entre dois membros e é representado por P [2, 4, 5]. A Figura 1 ilustra tais juntas.

A movimentação do braço robótico no espaço bidimensional ou tridimensional é determinado pelos graus de liberdade (GL ou *DOF* - *Degree Of Freedom*) [6]. O número de graus de liberdade de um robô é igual ao número de juntas articuladas. Quanto maior a quantidade de graus de liberdade, mais complicadas serão a cinemática, a dinâmica e o

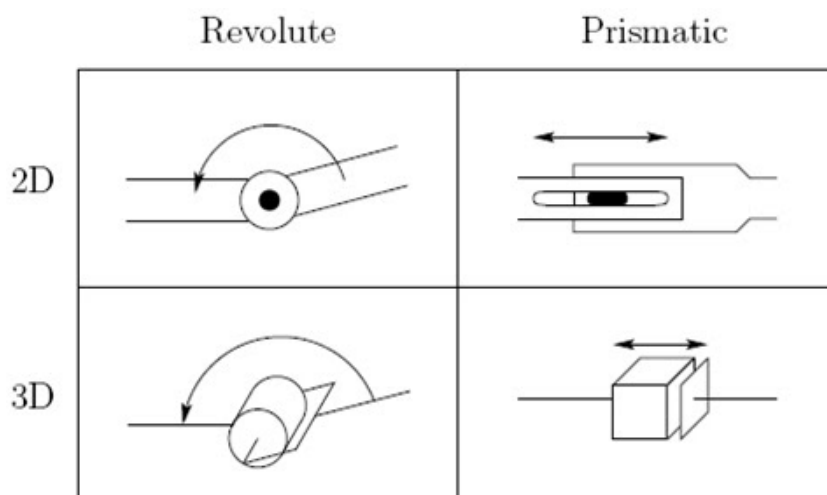


Figura 1: Representação simbólica das juntas de um robô. A junta rotacional (Revolute) produz rotação sobre os eixos adjacentes. A junta prismática (Prismatic) produz um movimento linear ou telescópico sobre os eixos adjacentes.

controle do manipulador. O número de graus de liberdade de um manipulador está associado ao número de variáveis posicionais independentes que permitem definir a posição de todas as partes de forma unívoca.

Um objeto rígido no espaço tridimensional possui seis DOF: três para o posicionamento e três para a orientação. Portanto, um manipulador deve possuir pelo menos seis DOF independentes, para descrever a movimentação e orientação o efetuador [2].

Os manipuladores são classificados pela sua geometria. Partindo de análises de diagrama de corpo livre e do espaço de trabalho, a maioria dos manipuladores são classificados em: articulados (RRR), esférico (PRR), SCARA (PRR), cilíndrico (RPP) e cartesiano (PPP) [5]. As Seções 1.2.1 a 1.2.6 descrevem tais manipuladores.

A escolha de um manipulador depende de sua aplicação. Além da geometria, são levados em consideração a precisão e a repetibilidade. A precisão de um manipulador é uma medida de quão perto o manipulador pode chegar a um determinado ponto dentro de sua área de trabalho. A repetibilidade é uma medida de quão próximo um manipulador pode retornar a um ponto previamente ensinados [2].

O diagrama de corpo livre é uma representação gráfica simplificada de um modelo real de manipulador, facilitando a compreensão e visualização das juntas prismáticas e rotacionais. O espaço de trabalho é definido pela configuração geométrica do manipulador pelas restrições físicas das juntas, indicando os limites alcançáveis pelo efetuador.

Elemento final do manipulador, o efetuador determina as funções que o manipulador

poderá exercer. Vários manipuladores tem a versatilidade de acoplar outro efetuator aumentando sua funcionalidade. Os efetutores podem ser do tipo garra, ou ferramentas especializadas, como furadeiras, polidoras, pontas de solda, entre outros. Vale comentar que efetuator não adiciona grau de liberdade ao manipulador.

1.2.1 Manipulador Articulado (RRR)

Constituído por três juntas rotacionais também é conhecido como manipulador antropomórfico. Estrutura e terminologia associada ao manipulador e sua área de trabalho podem ser vistas, respectivamente, nas Figuras 2(a) e 2(b).

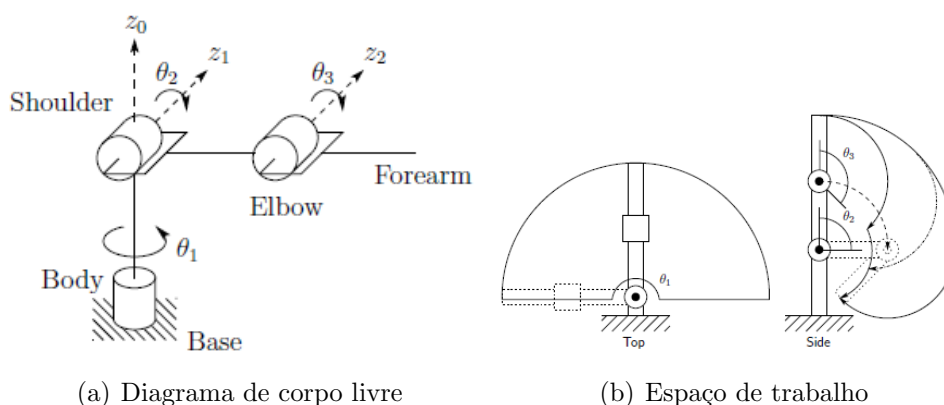


Figura 2: Manipulador Articulado.

Possui uma grande liberdade de movimentos para espaços relativamente pequenos, sendo assim um manipulador atraente e popular.

1.2.2 Manipulador esférico (RRP)

Substituindo a última junta rotacional do manipulador articulado por uma prismática, tem-se o manipulador esférico. A denominação esférica deriva do fato das coordenadas esféricas definirem a posição final do efetuator, como pode ser visto na Figura 3.

1.2.3 Manipulador SCARA (RRP)

O manipulador SCARA (*Selective Compliant Articulated Robot Arm*) é um manipulador muito popular, como o próprio nome sugere (Braço Robótico Articulado Seletivo e Compatível) é voltado para operações de montagem. Embora o SCARA tenha uma estrutura igual ao esférico (RRP), eles são muito diferentes em relação as suas aplicações. A Figura 4 ilustra tal manipulador.

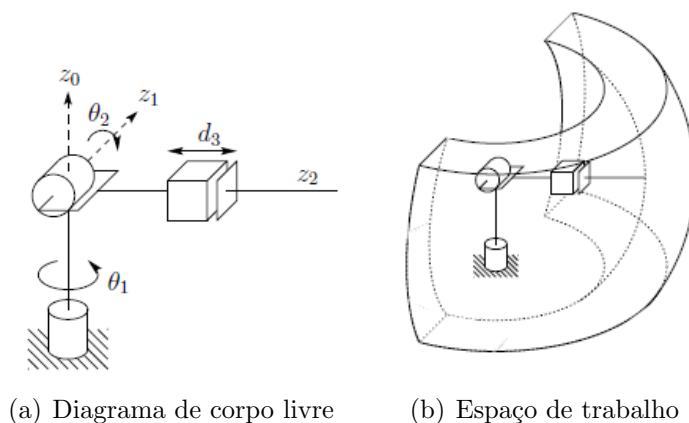


Figura 3: Manipulador Esférico.

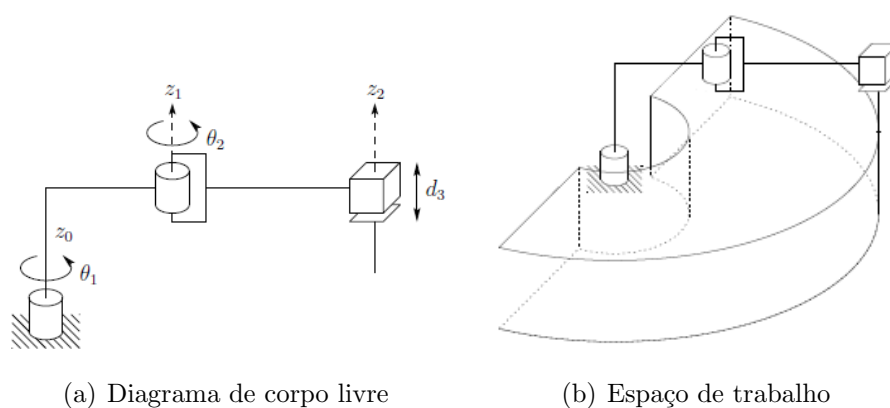


Figura 4: Manipulador SCARA.

1.2.4 Manipulador Cilíndrico (RPP)

O manipulador cilíndrico tem um eixo de rotação em torno da base e duas articulações prismáticas. O nome é devido as coordenadas cilíndricas aplicadas no efetuador. A Figura 5 ilustra sua configuração articular e seu espaço de trabalho.

1.2.5 Manipulador Cartesiano (PPP)

Constituído de três juntas prismáticas, o manipulador cartesiano (Figura 6) possui a cinemática mais simples dentre os manipuladores, já que a variação das juntas são as variações das coordenadas cartesianas do efetuador. Este manipulador é muito utilizado em portos marítimos para transferência de material ou cargas.

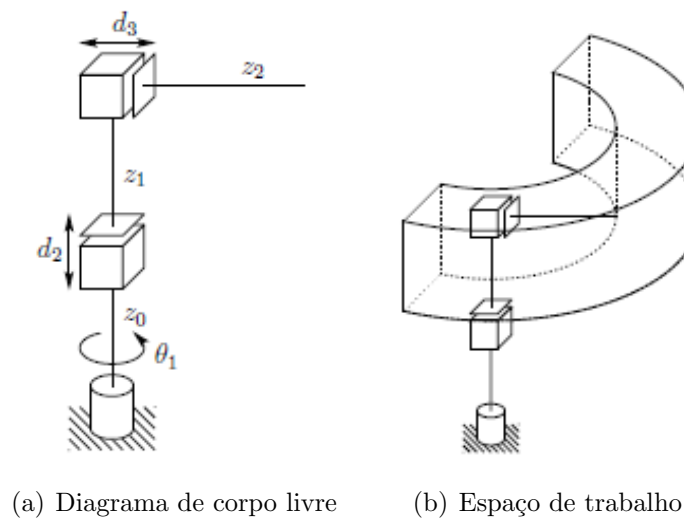


Figura 5: Manipulador Cilíndrico.

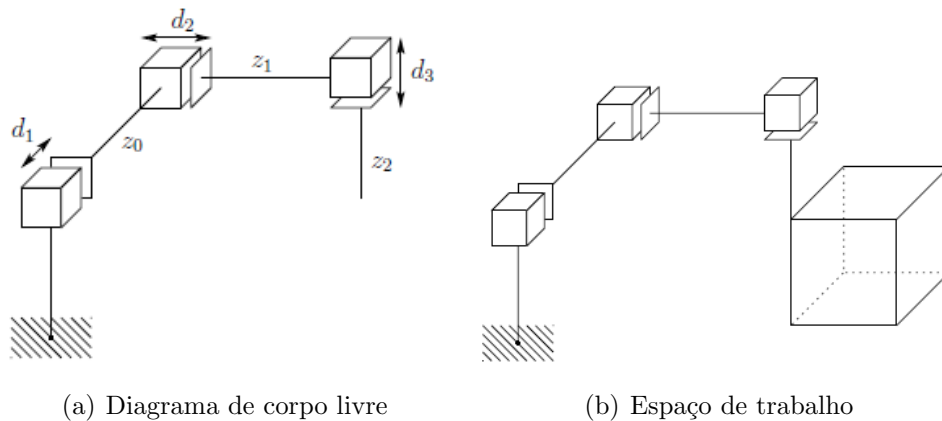


Figura 6: Manipulador Cartesiano.

1.2.6 Manipulador Paralelo

Um manipulador paralelo (Figura 7) é um manipulador formado por cadeia fechada. As cadeias cinemáticas fechadas de robôs paralelos podem resultar em uma maior rigidez estrutural e, portanto, maior precisão, do que robôs de cadeia aberta. A descrição da cinemática dos robôs paralelos é um pouco diferente da dos anteriores, requerendo métodos diferente de análise.

1.3 Cinemática Direta

Cinemática Direta é a relação entre as variáveis das juntas (ângulo entre juntas e/ou extensão de uma junta) e a posição e orientação do atuador. Sendo assim, pode ser definido o **Problema da Cinemática Direta**: “Dado um vetor de variáveis de juntas

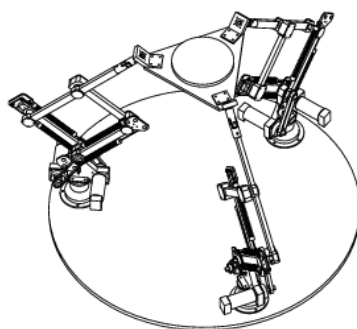


Figura 7: Estrutura do Manipulador Paralelo.

de um manipulador robótico, determinar a posição e orientação da ferramenta em relação à base do robô” [2, 4, 5].

As equações provenientes da cinemática direta variam de acordo com a configuração do robô. Assim, torna-se necessário o conhecimento dos parâmetros cinemáticos envolvidos em cada elemento do mesmo. As juntas podem ser rotacionais ou prismáticas formando, os manipuladores descritos nas seções 1.2.1 a 1.2.6.

Para a definição cinemática direta do manipulador planar (Figura 8). Inicialmente, estabelece-se um sistema de coordenadas fixo. Neste caso, a base das coordenadas é a mesma da base do manipulador que é chamada de $o_0x_0y_0$. Em cada eixo ocorre uma mudança nas coordenadas, que são chamadas de $o_1x_1y_1$ no eixo 1 e $o_2x_2y_2$ no eixo 2.

As coordenadas (x, y) são expressas da forma:

$$x = x_2 = \alpha_1 \cos(\theta_1) + \alpha_2 \cos(\theta_1 + \theta_2) \quad (1.1)$$

$$y = y_2 = \alpha_1 \sin(\theta_1) + \alpha_2 \sin(\theta_1 + \theta_2) \quad (1.2)$$

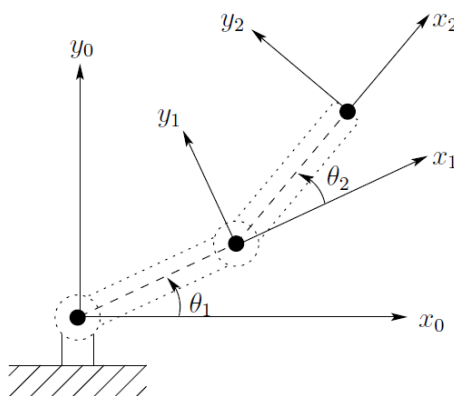


Figura 8: Eixo de coordenadas de um manipulador planar.

Sendo que o comprimento das partes constituintes do manipulador são α_1 e α_2 , respectivamente, e os ângulos das articulações são θ_1 e θ_2 . A orientação das novas coordenadas descritas por x_2y_2 relacionadas ao referencial inicial x_0y_0 , na forma matricial tem-se:

$$\begin{bmatrix} x_2.x_0 & y_2.x_0 \\ x_2.y_0 & y_2.y_0 \end{bmatrix} = \begin{bmatrix} \cos(\theta_1 + \theta_2) & -\text{sen}(\theta_1 + \theta_2) \\ \text{sen}(\theta_1 + \theta_2) & \cos(\theta_1 + \theta_2) \end{bmatrix} \quad (1.3)$$

As Equações 1.1, 1.2 e 1.3 são chamadas de Equações da Cinemática Direta de um manipulador. Quanto maior forem os graus de liberdade, maior será a complexidade das equações.

Um método utilizado para a simplificações das três equações descritas anteriormente é a notação de Denavit-Hartenberg (DH). Antes de descrever este importante método, torna-se necessária a compreensão de Transformação Homogênea.

Transformação homogênea é uma representação matricial de um movimento rígido, usando notação tridimensional para representar tanto o conjunto de movimentos referente as rotações quanto os referente as translações, formando uma matriz 4 x 4 (H).

$$H = \begin{bmatrix} R & d \\ 0 & 1 \end{bmatrix}, \quad (1.4)$$

onde R é a matriz de rotação e d é o ponto final da transformação, que representa a cinemática direta. Abaixo, pode-se observar as transformações homogêneas de base, dados pela rotação e translação em torno dos eixos x, y e z .

$$Trans_{x,a} = \begin{bmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}; Rot_{x,\alpha} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c_\alpha & -s_\alpha & 0 \\ 0 & s_\alpha & c_\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1.5)$$

$$Trans_{y,b} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & b \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}; Rot_{y,\beta} = \begin{bmatrix} c_\beta & 0 & s_\beta & 0 \\ 0 & 1 & 0 & 0 \\ -s_\beta & 0 & c_\beta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1.6)$$

$$Trans_{z,c} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & c \\ 0 & 0 & 0 & 1 \end{bmatrix}; Rot_{z,\gamma} = \begin{bmatrix} c_\gamma & -s_\gamma & 0 & 0 \\ s_\gamma & c_\gamma & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1.7)$$

A notação DH segue uma sequência de translações e rotações no eixos correntes definidas por 1.8. Em outras palavras, a determinação da postura da i -ésima articulação de um braço robótico relativa à articulação i_0 deste mesmo braço é dada pela multiplicação de cada uma das matrizes de transformação homogênea necessárias para alcançarem a i -ésima articulação [5].

Na notação de DH, cada transformação homogênea A_i é representada por um produto de quatro das transformações básicas descritas anteriormente, onde cada um dos parâmetros a_i , d_i , α_i e θ_i representam o comprimento, a excentricidade, a torção e o ângulo de rotação da junta, respectivamente, definidos por:

$$A_i = Rot_{z,\theta_i} Trans_{z,d_i} Trans_{x,a_i} Rot_{a,\alpha_i}, \quad (1.8)$$

$$A_i = \begin{bmatrix} c_{\theta_i} & -s_{\theta_i} & 0 & 0 \\ s_{\theta_i} & c_{\theta_i} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c_{\alpha_i} & -s_{\alpha_i} & 0 \\ 0 & s_{\alpha_i} & c_{\alpha_i} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (1.9)$$

$$A_i = \begin{bmatrix} c_{\theta_i} & -s_{\theta_i}c_{\alpha_i} & s_{\theta_i}s_{\alpha_i} & a_i c_{\theta_i} \\ s_{\theta_i} & c_{\theta_i}c_{\alpha_i} & -c_{\theta_i}s_{\alpha_i} & a_i s_{\theta_i} \\ 0 & s_{\alpha_i} & c_{\alpha_i} & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (1.10)$$

A aplicação da notação de DH depende de duas condições:

- 1^a) eixo x_1 perpendicular ao eixo z_0 ,
- 2^a) eixo x_1 intercepte o eixo z_0 .

Essas condições possibilitam a representação de uma transformação homogênea entre dois sistemas de coordenadas (Figura 9).

Tomando a Figura 8 como exemplo, é dada a tabela a seguir, onde os termos com * são os ângulos ou distâncias variantes.

Aplicando os dados da Tabela 1 em 1.10, são geradas duas matrizes do transformação

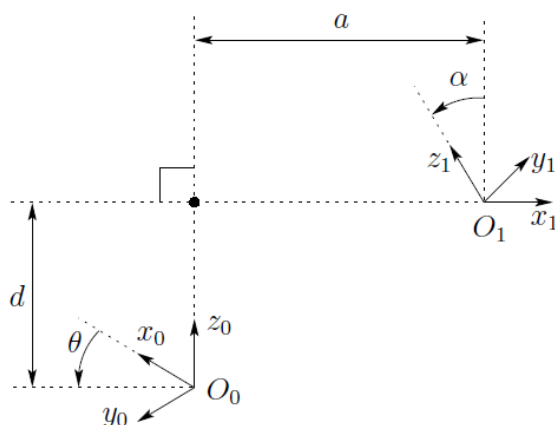


Figura 9: Condição para a utilização de DH.

Tabela 1: Notação de Denavit-Hartenberg.

Link	a_i	α_i	d_i	θ_i
1	a_1	0	0	θ_1^*
2	a_2	0	0	θ_2^*

homogênea que satisfazem as condições de DH.

$$A_1 = \begin{bmatrix} c_1 & -s_1 & 0 & a_1 c_1 \\ s_1 & c_1 & 0 & a_1 s_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1.11)$$

$$A_2 = \begin{bmatrix} c_2 & -s_2 & 0 & a_2 c_2 \\ s_2 & c_2 & 0 & a_2 s_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1.12)$$

$$T_1^0 = A_1 = \begin{bmatrix} c_1 & -s_1 & 0 & a_1 c_1 \\ s_1 & c_1 & 0 & a_1 s_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1.13)$$

$$T_2^0 = A_1 A_2 = \begin{bmatrix} c_{12} & -s_{12} & 0 & a_1 c_1 + a_2 c_2 \\ s_{12} & c_{12} & 0 & a_1 s_1 + a_2 s_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1.14)$$

Onde T_1^0 é chamado de matriz de transformação do referencial $o_0x_0y_0z_0$ para o referencial $o_1x_1y_1z_1$, analogamente, T_2^0 é a transformação do referencial $o_0x_0y_0z_0$ para o referencial $o_2x_2y_2z_2$.

1.4 Cinemática Inversa

Dadas a posição e orientação do efetuador do manipulador robótico, a cinemática inversa possibilita calcular todas as possíveis configurações de juntas que permitem o manipulador alcançar esta posição e orientação [7]. Como as equações cinemáticas de posição e orientação do manipulador são não-lineares, a solução do problema da cinemática inversa para um manipulador não é tão simples quanto à aplicação da cinemática direta para o mesmo.

A Figura 10 representa as possibilidades do manipulador da Figura 8 chegar na mesma posição (neste caso independente da orientação do efetuador). A configuração, com articulação virada para cima é chamada de cotovelo para cima (*elbow up*), e com articulação virada para baixo é chamada cotovelo para baixo (*elbow down*).

Um manipulador redundante é aquele que possui mais graus de liberdade controlados pelos efetuador no espaço de juntas do que aqueles necessários para realizar uma tarefa especificada no espaço cartesiano. Estes graus de liberdade excedentes garantem uma melhor manipulabilidade, permitindo, por exemplo, contornar singularidades, obstáculos ou limitações estruturais [1].

A redundância também permite executar tarefas adicionais de acordo com algum critério: baixo consumo de energia, balanceamento de velocidades de junta, balanceamento de esforços de juntas, entre outros. Os graus de liberdade adicionais tornam o manipulador redundante mais confiável e tolerante a falhas.

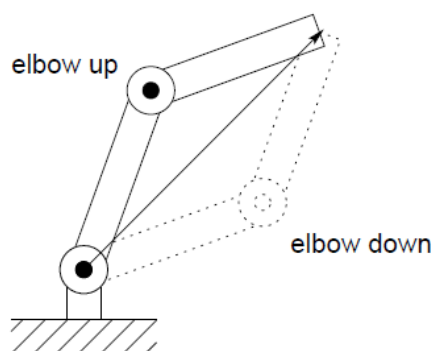


Figura 10: Múltiplas soluções para cinemática inversa.

Em algumas situações, o robô pode executar certas tarefas mesmo com a falha de uma ou mais juntas. Por outro lado, de modo geral, os manipuladores redundantes apresentam uma maior complexidade, são mais pesados e requerem algoritmos de controle mais complicados do que os manipuladores não redundantes, tais como sistemas nebulosos (*fuzzy*), redes neurais artificiais, redes neurais nebulosas, computação evolutiva, dentre outros.

1.5 Algoritmo Genético

A busca por uma solução em uma coleção de soluções candidatas para um determinado tipo de problema é muito comum em problemas computacionais. A utilização de técnicas de paralelismo acelera o processo, pois várias possibilidades são analisadas simultaneamente. Outra ferramenta bastante interessante são os algoritmos adaptativos, que permitem que problemas com variações constantes no ambiente sejam solucionados [8].

Os métodos clássicos de otimização utilizam derivadas para direcionar a busca do próximo candidato. Este método pode ficar preso a um extremo local, fazendo assim, que algoritmo nem sempre convirja para a melhor solução.

Os Algoritmos Genéticos representam uma classe de ferramentas muito versátil e robusta a ser empregada na solução de problemas de otimização, pois incorporam várias características importantes (busca de solução, paralelismo, adaptação) [9]. São diferenciados dos métodos clássicos por:

- Utiliza codificação das possíveis soluções (genótipos) e não com as soluções propriamente ditas (fenótipos).
- Não necessitam do cálculo do gradiente nem derivadas, operam unicamente com o valor da função a ser otimizada.
- Realizam a busca por todo o espaço de uma só vez,
- Não se prendem tão facilmente a extremos locais.

O Algoritmo Genético (AG) teve sua criação e desenvolvimento entre as décadas de 60 e 70 por Jonh Holland. Inspirado no princípio Darwiniano de seleção natural e reprodução genética tinha como objetivo implementação algoritmos computacionais de busca estocástica e métodos de otimização, capaz de solucionar problemas adversos [10].

Tabela 2: Definições em Algoritmos Genéticos.

Nome	Definição
Cromossomo	Cadeia de caracteres representando informações relativa às variáveis do problema.
Gene	Unidade básica do cromossomo. Cada cromossomo possui um certo número de genes, cada um descrevendo uma certa variável do problema.
Alelo	Valores que o gene pode possuir.
Indivíduo	Possível solução do problema.
População	Conjunto de cromossomos.
Espaço de Busca	Espaço que compreende as possíveis ou viáveis soluções do problema a ser otimizado. Deve ser caracterizado pelas funções de restrição, que definem as soluções viáveis do problema.
Geração	Número de interações do Algoritmo Genético

Partindo dos conceitos da Tabela 2 [9], o operador de seleção escolhe os cromossomos na população que irão se reproduzir, por meio de um algoritmo de seleção, e descarta aleatoriamente alguns cromossomos por meio de um algoritmo de mortalidade. Os cromossomos selecionados irão gerar dois novos cromossomos que irão herdar características dos pais e podem sofrer alguma modificação provinda do algoritmo de mutação, originando a nova geração composta pelos indivíduos antigos e novos.

Existem variações de implementação até conceituais referentes aos algoritmos genéticos, porém, para quaisquer variações, o algoritmo segue os componentes [9]:

1. Adaptação do problema para os genes constituintes dos cromossomos,
2. Criação de uma população inicial,
3. Uma função para calcular o *fitness* (aptidão),
4. Operadores genéticos responsáveis pelo cruzamento e mutação,
5. Escolha dos parâmetros referentes ao tamanho da população, número de gerações, probabilidades, dentre outros.

O AG tradicional [11] realiza as seguintes funções:

1. Gera a população inicial;
2. Avalia os indivíduos da população;

3. Enquanto o critério de parada não for satisfeito faça:
 - (a) Seleciona os indivíduos mais aptos de acordo com a avaliação;
 - (b) Cria novos indivíduos aplicando *crossover* e mutação;
 - (c) Atualiza a população;
 - (d) Avalia cada cromossomo da nova população

Os conceitos de um Algoritmo Genético serão descritos nos próximos tópicos.

1.5.1 Representação

A representação das possíveis soluções do espaço de busca de um problema define a estrutura do cromossomo a ser manipulado pelo algoritmo. A representação do cromossomo depende do tipo de problema. Cada possível solução no espaço de busca é representada por uma sequência de símbolos X gerados a partir de um alfabeto (binário ou real). Cada sequência corresponde a um cromossomo e cada elemento de X é equivalente a um gene. Nenhuma forma de codificação funcionaria igualmente bem em todas as situações. Para cada caso, deve-se fazer uma escolha cuidadosa do tipo de codificação a ser utilizada, pois uma codificação ruim pode não levar ao resultado esperado [12]. Os principais tipos de representação [13] são:

Tabela 3: Principais Representações de AG.

Representação	Problemas
Binária	Numéricos, Inteiros
Números Reais	Numéricos
Permutação de Símbolos	Baseados em Ordem
Símbolos Repetidos	Grupamento

Alguns pesquisadores indicam que a representação real (ponto flutuante) apresenta melhor desempenho em relação a representação binária para aplicações que necessitem de tratamento de valores contínuos, pois apresenta mais compatibilidade, precisão e rapidez de execução. A representação binária é mais apropriada em aplicações que requeiram o tratamento de valores discretos [10].

1.5.2 Inicialização da População

A inicialização básica de um algoritmo genético clássico se resume à síntese de uma população inicial. Comumente, são utilizadas funções aleatórias para gerar tal população,

na qual visa fornecer maior “biodiversidade”, uma característica fundamental para garantir uma boa abrangência do espaço de pesquisa [11]. Para acelerar o processo de busca, pode ser inserido alguma informação prévia que indique regiões promissoras no espaço de busca. O tamanho da população inicial é diretamente proporcional ao tamanho do cromossomo, ou seja, quanto maior o cromossomo maior deve ser a população para manter uma boa diversidade[8].

1.5.3 Validação

Validação é uma função que tem o objetivo de avaliar os indivíduos, análoga ao papel do meio ambiente, ou seja, os indivíduos mais bem adaptados a um determinado meio terão valores de aptidão melhores do que os indivíduos menos adaptados.

1.5.4 Seleção

A maioria dos métodos de seleção são projetados para determinar os indivíduos da população que vão participar da reprodução, sendo eles os indivíduos com maiores ou menores valores de aptidão, perpetuando assim, os melhores genes.

O método de Seleção da Roleta é um método bastante utilizado, onde cada indivíduo da população é representado na roleta proporcionalmente ao seu índice de aptidão. Isso faz com que os indivíduos com alta aptidão tenham uma parcela maior na roleta, enquanto os de baixa, uma menor parcela. A roleta é então girada um número determinado de vezes formando uma população diferenciada, os indivíduos desta população irão obrigatoriamente participar da nova geração [12].

Um método semelhante ao anterior é denominado Seleção por Amostragem Universal Estocástica. Este método considera um círculo dividido em i regiões (tamanho da população), onde a área de cada região é proporcional à aptidão do indivíduo conforme a Figura 11. É então colocada uma “roleta” sobre o círculo com n cursores, igualmente espaçados. Com apenas um giro da roleta, n indivíduos serão selecionados. Este método é mais rápido que o anterior, porém a seleção pode conter cópias de um mesmo indivíduo enquanto outros podem desaparecer.

Outro método conhecido é método de Seleção por Torneio em que são selecionado dois indivíduos aleatoriamente, e elege o mais adaptado a continuar na população, em seguida os dois cromossomos voltam para a população original e podem ser novamente selecionados. Este processo se repete até que se forme uma nova população com um

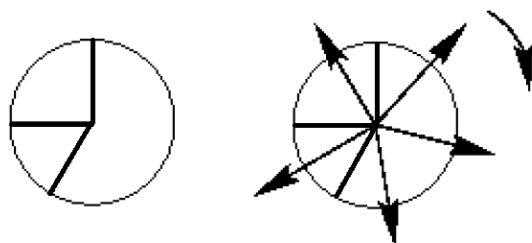


Figura 11: Método de Seleção por Amostragem Universal Estocástica.

tamanho pre determinado.

1.5.5 Recombinação

Considerado um dos principais operadores do AG, a recombinação é aplicada a uma parcela da população relacionada à taxa de reprodução. Se muito alta, maior parte da população será substituída e poderá perder indivíduos de alta aptidão, por outro lado, se muito baixa o algoritmo se torna muito lento.

Após a seleção, é efetuada a reprodução, na qual os melhores indivíduos adaptados (melhor aptidão) têm maiores chances de participar da reprodução. O operador da reprodução é o modelo artificial da seleção natural. E a aptidão ou fitness é a habilidade que o indivíduo possui para sobreviver.

A maioria dos operador utilizam números pares de indivíduos para efetuar o cruzamento [9]. Os pares de indivíduos são escolhidos aleatoriamente variando de acordo com a taxa de cruzamento. Existem vários tipos de *crossover* (cruzamento entre cromossomos), dentre eles podem ser citados os cruzamentos de um ponto ou de múltiplos pontos.

O cruzamento de um ponto (Figura 12) é o método de cruzamento mais simples, consiste na escolha aleatória de um ponto para a divisão da cadeia do cromossomo. Copia-se então a primeira parte para um filho, e a segunda para o outro filho. O mesmo é feito para o outro pai, gerando assim dois filhos que contém parte dos cromossomos dos dois pais.

O cruzamento de múltiplos pontos é menos utilizado que o anterior e consistem em dividir o cromossomo em vários pontos e os recombina com outro cromossomo para formar os filhos, possuindo a vantagem de assegurar uma variedade genética maior [12].

Na utilização da representação real, os operadores descritos anteriormente não são adequados, sendo os operadores aritméticos os mais indicados. Dentre eles pode-se citar:

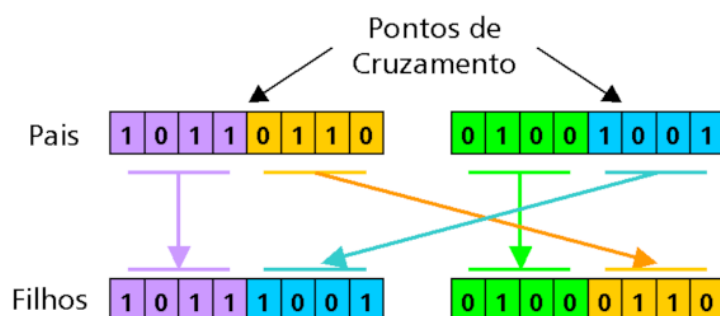


Figura 12: Cruzamento em um ponto.

média aritmética, média geométrica, BLX- α e heurístico

1.5.6 Mutação

A mutação visa evitar que alguns elementos desapareçam definitivamente da população [9]. O processo de mutação é aplicável para todas as formas de representação adotada.

O processo de mutação para a base binária é simples: de acordo com a probabilidade de mutação, cada bit de cada um dos genes um cromossomo escolhido aleatoriamente tem a probabilidade aleatória de ser mutado. Esta mutação é simplesmente a mudança deste valor em um determinado gene do cromossomo escolhido. Também pode ser usada a técnica de troca, na qual o valor de 2 *bits* são trocados entre si.

Tabela 4: Mutação AG.

Condição	Cromossomo
Antes	1001 11100 001
Depois	1001 10100 001

A Tabela 4 mostra um cromossomo constituído de 3 genes, cada um dos seus genes possuiu um número próprio de *bits*, a mutação ocorreu somente no segundo *bit* do segundo gene do cromossomo mutado.

A maioria das literaturas adotam uma taxa de mutação abaixo de 2%. Em alguns casos da natureza, organismos só conseguem mater sua sobrevivência se possuírem um alta taxa de mutação, que é o caso dos vírus. O vírus é um organismo unicelular com reprodução assexuada consegue ter uma alta capacidade de adaptação devido a sua taxa de mutação, tornando-os uma motivação para os AGs. Segundo [12] se apenas possuir uma taxa de mutação bastante elevada, a população poderá ser capaz de comportar-se

como os vírus, mudando sempre para se adaptar ao seu meio ambiente e reproduzindo-se de forma assexuada.

1.5.7 Parâmetros dos AGs

Um dos aspectos mais importantes da estratégia do AG é a configuração dos parâmetros de controle. A eficiência e bom funcionamento de uma AG depende da escolha dos seus parâmetros [9], sendo os principais:

- **Tamanho da População**

O tamanho da população determina o número de pontos do espaço de busca. Uma pequena população pode levar a uma convergência errada, por outro lado uma grande população prejudicará o desempenho computacional;

- **Taxa de Cruzamento**

Indica a taxa de indivíduos selecionados que iram participar do cruzamento. Se muito alta, indivíduos com alta aptidão podem ser perdidos; se baixa, pode tornar a convergência do algoritmo muito baixa;

- **Taxa de Mutação**

Indica a taxa de indivíduos que sofreram a mutação. É utilizada para fornecer novas informações à população de forma aleatória evitando a saturação;

- **Critério de Parada**

Indica o instante de parada que o algoritmo é interrompido. O mais simples é a parada por número de gerações, podendo ser mesclado por uma solução ótima ou uma convergência de grande parte da população.

1.6 Objetivos

Este trabalho tem como objetivo geral o desenvolvimento de um simulador tridimensional utilizando o Software MATLAB, dotando-o de uma inteligência computacional por ter características redundantes.

Como objetivos específicos, este trabalho visa a construção de um simulador de um braço robótico antropomórfico redundante de quatro DOF. Na sequência, aplicar técnicas

de inteligência artificial especificamente, algoritmos genéticos para a solução do problema da cinemática inversa nestes tipos de manipuladores.

2 *O Simulador*

Sabe-se que um simulador de um manipulador robótico facilita a compreensão do seu princípio de funcionamento e a sua mobilidade no espaço de trabalho. Sendo assim, ferramentas de simulação são úteis e importantes para reduzir o tempo de desenvolvimento, evitar danos ao equipamento devido a falhas nos algoritmos de controle e facilitar a procura de erros nas estratégias apresentadas, com foco também, em questões de inteligência e controle.

Além disso, em um ambiente de simulação, alguns comportamentos podem ser perfeitamente controlados, por exemplo, a incidência de ruído, a veracidade das informações sensoriais, o desempenho do efetuador, os efeitos dinâmicos do robô, e assim por diante. Adicionalmente, o acesso da navegação através de simulação de dados é mais fácil do que em uma situação experimental, embora tal situação seja essencial para verificar os resultados de forma eficaz.

O simulador aqui apresentado propicia uma visualização tridimensional do movimento do braço, recebendo os ângulos das articulações como parâmetros de entrada. Através da utilização dos sistemas de controle, a movimentação instantânea do manipulador simulado comporta como um manipulador real com todas as suas limitações físicas que definem seu espaço de trabalho.

Quanto maior a fidelidade entre o modelo simulado e o real, melhores serão os resultados das tarefas a serem executadas, tais como manipulação de objetos em meios industriais, residenciais e, até mesmo, militares.

O simulador é originado de um manipulador robótico Figura 13 disponível no Núcleo Especializado em Robótica (NERO) do Departamento de Engenharia Elétrica da Universidade Federal de Viçosa (DEL / UFV).

O manipulador em questão possui 4 DOF, como nesse trabalho não considera a orientação do efetuador, o manipulador para essa aplicação é redundante, e como discutido na Sessão 1.4, sua cinemática inversa não é resolvida facilmente. No entanto, foi adotada



Figura 13: OWI 535 Robotic Arm Edge (vide Apêndice A).

a utilização de Algoritmo Genético para a resolução deste problema.

2.1 Criação da Interface

A visualização tridimensional foi adquirida utilizando a função *patch* do software MATLAB. Esta função tem como parâmetros de entrada, os vetores x, y, z , a cor da face e a cor do contorno. Juntos, os vetores formam um conjunto de pontos que ao serem ligados (pela função) completam uma face de um polígono. Tais vetores de posição formam uma matriz na ordem $3 \times N$ (três linhas por N colunas, sendo N o número de pontos que constituem as faces). Utilizando um conjunto de faces, pode-se construir sólidos para a visualização tridimensional.

A cor da face pode ser dada na base Red Green Blue (RGB) ou em um sistema básico de cores determinadas do software. O RGB (Figura 14) é representada por um vetor que tem a intensidade das três cores utilizadas como base (vermelho, verde e azul) para gerar todas as cores. Sistema de cores RGB inserido no código do programa por `COLOR = [Red Green Blue]`, onde a intensidade de cada canal varia de 0 a 100% e, desta forma, pode-se definir os diferentes níveis de cores. A Tabela 5 apresenta algumas cores padrão.

A vantagem apresentada pela notação em RGB são as possíveis variações das cores base, como um vermelho escuro $[0.8, 0, 0]$ ou um vermelho claro $[1, 0.2, 0.2]$.

Foram então criadas as funções *ArmParameters* e *ArmManipulator_Plot*, a primeira é constituída pelas matrizes que representam as faces do simulador, quando utilizada carrega todos os dados referentes as faces que ilustram o manipulador. A outra é responsável

Tabela 5: Sistemas RGB e cores básicas do MATLAB.

Cor	Código da Cor	RGB
Vermelho	[1 0 0]	'r'
Verde	[0 1 0]	'g'
Azul	[0 0 1]	'b'
Branco	[0 0 0]	'w'
Preto	[1 1 1]	'k'

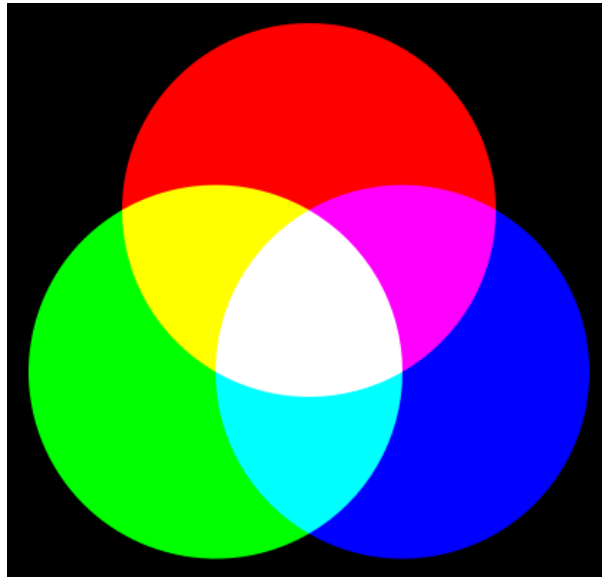


Figura 14: Padrão de cores RGB.

pela criação da visualização das faces no local determinado. Esta criação é feita imprimindo face por face no display. Todos os sólidos foram criados na etapa de desenvolvimento do simulador com seus centros de rotação na origem devido à notação de DH (Sessão 1.3).

A função *ArmParameters* não possui parâmetros de entrada, quando utilizada a função cria uma estrutura que contém todas as faces. O sistema de estrutura foi utilizado com intuito de organização, facilitando assim a função *ArmManipulator_Plot*, que tem como parâmetros de entrada os ângulos das juntas (*ArmVariables*) e a posição (*ArmPose*) do simulador em um espaço tridimensional, como pode ser visto abaixo.

```
ArmManipulator_Graph = ArmManipulator_Plot(ArmPose,ArmVariables)
```

Uma preocupação existe quando se trabalha com simulações em robótica é o tempo gasto para serem efetuados os cálculos e a movimentação do simulador. Para este trabalho foi adotado um tempo máximo de 500 milissegundos (ms), desde a entrada da posição desejada até a visualização final (de cada frame) na tela do usuário. O simulador desenvolvido pode ser visto na Figura 15 e possuiu 266 faces.

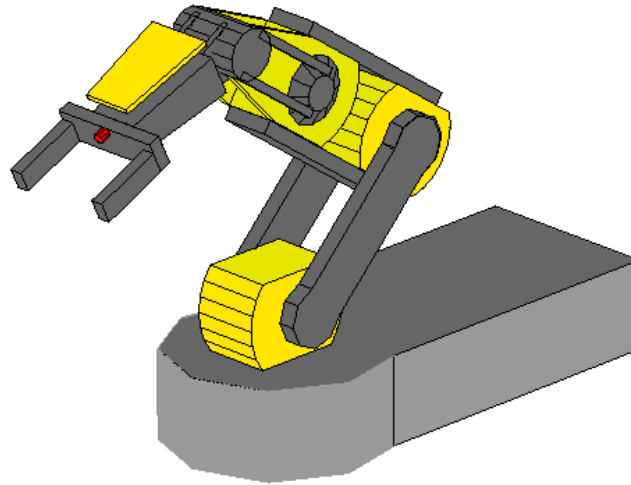


Figura 15: Imagem do simulador desenvolvido.

2.2 Rotação e Translação

A notação de DH é utilizada para efetuar a movimentação dos sólidos. Para cada um dos DOF foi realizado o estudo, para encontrar a matriz de DH que, ao ser multiplicada pela matriz das faces, efetuar a rotação e translação do sólido levando-o à localização final. Para isso se faz necessária a análise de movimentação no diagrama de corpo livre identificando os ângulos de rotação fixos e variáveis.

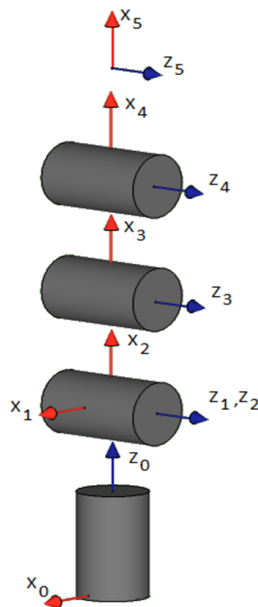


Figura 16: Diagrama de Corpo Livre.

Pela observação do diagrama de corpo livre pode-se montar a Tabela 6 que contém os

dados necessários para a construção das matrizes de DH. A partir dessa tabela são criadas as matrizes que descrevem a rotação e a translação dos sólidos referentes a cada um dos graus de liberdade com o intuito de imprimir a posição final do simulador.

Tabela 6: Notação de DH para o simulador.

<i>Link</i>	a_i	α_i	d_i	θ_i
1	0	-90°	d_1	θ_1^*
2	0	0	0	-90°
3	a_2	0	0	θ_2^*
4	a_3	0	0	θ_3^*
5	a_4	0	0	θ_4^*

$$T_1^0 = A_1 = \begin{bmatrix} c_1 & 0 & -s_1 & 0 \\ s_1 & 0 & c_1 & 0 \\ 0 & -1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.1)$$

$$T_2^0 = A_1 A_2 = \begin{bmatrix} 0 & c_1 & -s_1 & 0 \\ 0 & s_1 & c_1 & 0 \\ 1 & 0 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.2)$$

$$T_3^0 = A_1 A_2 A_3 = \begin{bmatrix} c_1 s_2 & c_1 c_2 & -s_1 & a_2 c_1 s_2 \\ s_1 s_2 & s_1 c_2 & c_1 & a_2 s_1 s_2 \\ c_2 & -s_2 & 0 & d_1 + a_2 c_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.3)$$

$$T_4^0 = A_1 A_2 A_3 A_4 = \begin{bmatrix} c_1 s_{23} & c_1 c_{23} & -s_1 & c_1(a_2 s_2 + a_3 s_{23}) \\ s_1 s_{23} & s_1 c_{23} & c_1 & s_1(a_2 s_2 + a_3 s_{23}) \\ c_{23} & -s_{23} & 0 & d_1 + a_2 c_2 + a_3 c_{23} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.4)$$

$$T_5^0 = A_1 A_2 A_3 A_4 A_5 = \begin{bmatrix} c_1 s_{234} & c_1 c_{234} & -s_1 & c_1(a_2 s_2 + a_3 s_{23} + a_4 s_{234}) \\ s_1 s_{234} & s_1 c_{234} & c_1 & s_1(a_2 s_2 + a_3 s_{23}) + a_4 s_{234} \\ c_{234} & -s_{234} & 0 & d_1 + a_2 c_2 + a_3 c_{23} + a_4 c_{234} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.5)$$

Para efetuar a rotação e translação, basta multiplicar a matriz de DH pelas matrizes das faces que completam os sólidos. Como visto na Sessão 1.3, as matrizes de DH são 4x4 (quatro linhas por quatro colunas) e como visto na Sessão 2.1 as matrizes de face são

3xN.

Pela propriedade da multiplicação de matrizes, a multiplicação existe apenas se o número de colunas da matriz da esquerda (matriz de DH) for igual ao número de linhas da matriz da direita (matriz das faces). Sendo assim, é necessário incrementar uma nova linha na matriz das faces para que a multiplicação seja efetuada. Neste sentido, foi acrescentada uma linha com 1's na matriz das faces, isto é,

$$Faces = \begin{bmatrix} x_1 & x_2 & x_3 & \dots & x_n \\ y_1 & y_2 & y_3 & \dots & y_n \\ z_1 & z_2 & z_3 & \dots & z_n \\ 1 & 1 & 1 & \dots & 1 \end{bmatrix}, \quad (2.6)$$

onde n é o número de pontos que constituem as faces.

2.3 Funções do Algoritmo Genético

Uma grande vantagem do algoritmo genético é a de possuir funções genéricas, ou seja, pode-se utilizar o mesmo código para vários tipos de problemas com poucas modificações. As modificações necessárias são referentes à escolha dos pais e as funções responsáveis pelo cálculo e escolha dos indivíduos mais aptos a serem a solução do problema.

Para se adequar o algoritmo genético ao problema proposto foram adotados que cada gene representa um ângulo, fazendo com que cada cromossomo possua quatro genes. A aptidão será determinada pela otimização entre a menor energia consumida para chegar ao ponto e a melhor aproximação do ponto encontrado pelo desejado.

2.3.1 ga_principal

É a função principal do algoritmo genético, nela se encontra todas as funções e parâmetros referentes ao AG. Os principais parâmetros são

- *taxa_mutacao*

Constante responsável pelo percentual de novos indivíduos que serão mutados na função *ga_mutacao*.

- *taxa_elitismo*

Constante responsável pela porcentagem da população que será selecionada por

serem mais aptos pela função `ga_elitismo`.

- *taxa_mortalidade*

Constante responsável pelo percentual de indivíduos que serão descartados da população comum na função `ga_morte`.

- *n_geracao_max*

Constante responsável pelo número de gerações máxima que o algoritmo irá percorrer antes que interrompa o ciclo de geração elegendo o melhor indivíduo da população.

- *n_bits_th1, n_bits_th2, n_bits_th3 e n_bits_th4*

Conjunto de constantes responsável pela conversão do ângulo em decimal para binário. O número de bits não necessariamente tem que ser o mesmo para cada gene.

As constantes adotadas são importantes para a otimização do tempo pelo resultado que o AG irá gerar. A escolha dos pais também influencia diretamente na otimização do problema fazendo com que a população seja maior, gastando mais tempo para executar as rotinas. Por outro lado, tem-se uma maior chance de encontrar melhores indivíduos. Foram necessários vários testes com diversos valores dos parâmetros e a escolha de diversos pais para encontrar um modelo que se enquadrasse nas condições de tempo e aproximação do ponto desejado.

Em seguida, a função entra em um *looping* que consiste em gerar novos indivíduos, selecionar os melhores, e descartar parte dos piores, afim de encontrar o melhor indivíduo no decorrer das gerações.

2.3.2 `ga_geracao(populacao, taxa_mutacao)`

Cria rotina para realizar o cruzamento com uma varredura para certificar-se que não existe nenhum indivíduo igual ao gerado, se existir, o indivíduo gerado é então descartado e a função continua a rotina. Após o cruzamento de todos os pais, ocorre a mutação nos novos indivíduos, adicionando-os, em seguida, à população total.

2.3.3 `ga_cruzamento(populacao)`

Realiza o cruzamento entre pais gerando dois novos indivíduos que irão compor a geração. Seleciona-se aleatoriamente o número de genes a serem combinados em cada cromossomo e define a máscara de cruzamento, a qual é responsável pelo direcionamento do gene dos pais para cada filho, assim cada filho pega o gene de um dos pais de forma aleatória até formar os cromossomos completos dos dois novos indivíduos (filhos).

2.3.4 `ga_mutacao(populacao_filhos)`

Realiza a mutação de um gene (ou vários) do indivíduo de entrada. Encontra o número de cromossomos e o número de genes de cada cromossomo. Em seguida, seleciona aleatoriamente o cromossomo a ser modificado e o gene a ser mutado. Finalmente, ocorre a troca do gene específico, se o gene for ‘1’ troca para ‘0’ e se for ‘0’ troca-se para ‘1’.

2.3.5 `aptidao_manipulador(populacao, pos_des)`

Recebe a população e a posição desejada, efetua a soma da diferença (equação 2.8) dos ângulos iniciais aos ângulos finais (mínima energia) e calcula a distância Euclidiana (equação 2.7) entre o valor da posição atingida e desejada (melhor aproximação). Retornando a aptidão de cada indivíduo para atingir a nova posição e a soma das diferenças dos ângulos. Aptidão é uma matriz que contém os valores dados a cada indivíduo da melhor aproximação e menor energia.

$$aptidao(1, 1) = norm(pos_cal(1, :) - pos_des) \quad (2.7)$$

$$aptidao(1, 2) = 40abs(th2 - th02) + 20abs(th3 - th03) + 2abs(th4 - th04) \quad (2.8)$$

2.3.6 `ga_elitismo(populacao, aptidao, taxa_elitismo)`

Recebe os dados de aptidão referentes aos indivíduos e baseado na taxa de elitismo, seleciona a porção da população mais apta que irá sobreviver “obrigatoriamente” na próxima geração. Retorna a população elite e a população comum.

2.3.7 ga_morte (populacao_comum,taxa_mortalidade)

De forma aleatória são selecionados os indivíduos da população comum que serão descartados. O número de indivíduos descartados é proporcional à constante taxa_mortalidade.

2.4 Função de Movimento

É a função responsável pela criação de uma noção de movimento do simulador. Inicialmente são configurados alguns parâmetros visuais com o objetivo de melhorar a interface com o usuário entre eles estão a escolha do ângulo de visualização, o tamanho fixo dos eixos, as proporções da imagem durante a movimentação e a nomeação dos eixos.

Na rotina de movimentação ocorre a soma de uma constante ao ângulo inicial até que chegue ao ângulo encontrado pela AG, a cada novo ângulo a função *ArmManipulator_Plot* irá criar a imagem do manipulador na posição atual, seguindo de um *pause(0.1)*. Esta função faz com que o programa fique parado um tempo de 100 ms, tempo suficiente para que o usuário consiga ver a imagem, para que esta possa ser apagada, para o aparecimento da próxima imagem, formando assim um *looping* que termina quando o ângulo que vem sendo incrementado linearmente ate alcançar o ângulo desejado. O algoritmo em questão é descrito a seguir:

Faz Posicionamento por cinemática direta

Enquanto “Há posições articulares a atingir”

- . Exibe graficamente o Manipulador
- . Pausa o tempo 1 ms
- . Apaga a imagem

Fim do Laço.

3 *Resultados e Discussões*

A representação no AG veio do problema da cinemática inversa do manipulador simulado. A criação da interface gráfica interligada com o simulador mostra uma aplicação direta do algoritmo genético, já que, após calculado o ponto, o simulador cria uma noção de movimentação ate o ponto encontrado.

AGs são muito utilizados em manipuladores redundantes, porém existem outros métodos também são utilizados para calcular a cinemática inversa, tais como, redes neurais, sistemas numéricos, dentre outros. Com esse simulador, pode ser adotado qualquer método para os cálculos, devido à utilização de funções genéricas na constituição desse trabalho.

3.1 Simulador

O simulador projetado possui duas funções básicas, descritas na seção 2.1, responsáveis por carregar os dados das faces e por imprimir a face. Como parâmetros de entrada, a função recebe os ângulos das articulações e a posição em que o manipulador robótico se encontra.

```
ArmManipulator_Graph = ArmManipulator_Plot(ArmPose, ArmVariables)
```

Uma das preocupações no desenvolvimento do simulador foi criá-lo de forma generalizada, para que apenas com os parâmetros de entrada pudessem ser utilizado em várias aplicações. Sendo assim, o vetor `ArmPose` descreve a posição no espaço tridimensional e a angulação que o manipulador se encontra (Figura 17).

Esta característica faz com que o simulador possa ser acoplado a outros simuladores criados no MATLAB ou até mesmo simular diferentes aplicações. A Figura 18 mostra dois exemplos de aplicação. A Figura 18(a) representação a simulação de uma esteira industrial em que o manipulador pega somente os blocos azuis, na Figura 18(b) o manipulador foi acoplado ao um robô móvel (1.1), podendo servir como simulador de um robô anti-bomba.

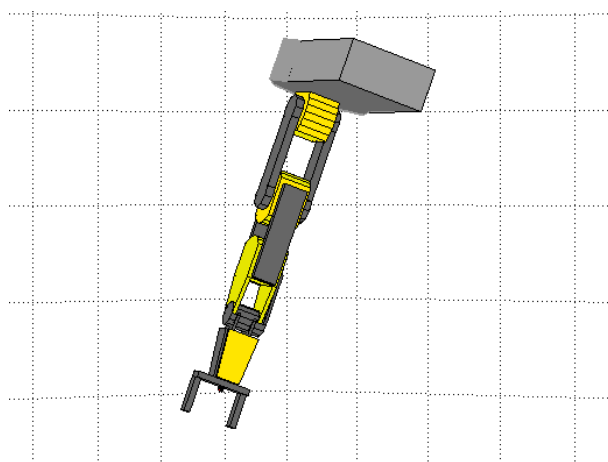
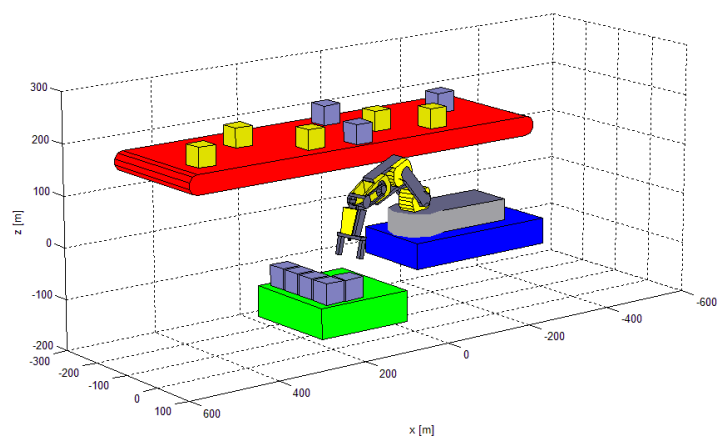
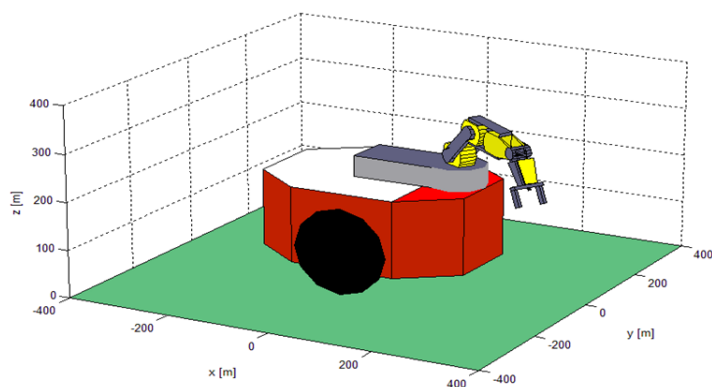


Figura 17: Simulação valores aleatórios no parâmetro ArmPose.



(a)



(b)

Figura 18: Aplicações.

A simulação dá uma ótima noção tridimensional do espaço de trabalho (Figura 19) facilitando a escolha de suas aplicações.

A construção da interface tridimensional do manipulador foi feita utilizando o manual do mesmo para adquirir suas proporções reais.

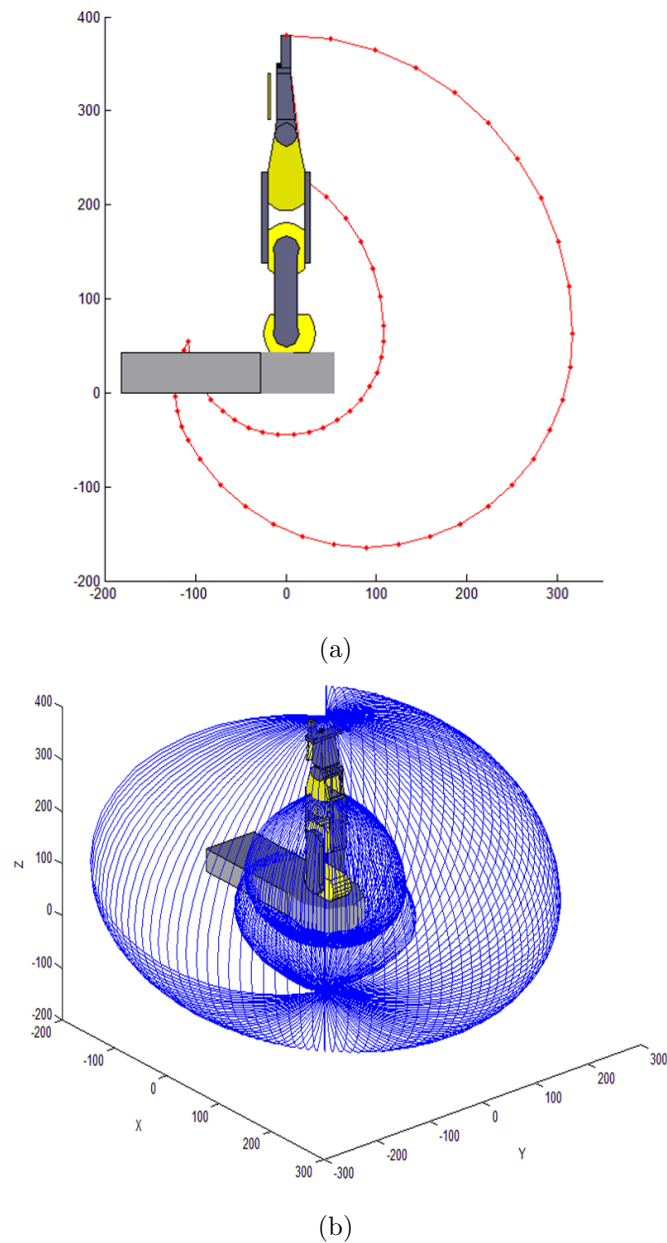


Figura 19: Espaço de Trabalho.

3.2 Algoritmo Genético

Como mencionado na Seção 1.5.7, um dos aspectos mais importantes do AG é a determinação de seus parâmetros. O tempo da simulação final é diretamente proporcional ao AG, já que, o tempo gasto com a criação da interface gráfica é sempre constante. Para este simulador foi adotado um tempo máximo de 600 ms, ou seja, teve que adequar o AG para achar uma boa solução obedecendo esta condição. Contando com os 100 ms gastos para a exibição do protótipo simulado.

O algoritmo adotado não possui taxa de cruzamento, ou seja, todos os indivíduos da

população participam do cruzamento e, para o controle do número da população, existe um operador que efetua o papel de eliminar, de forma aleatória os indivíduos que não fazem parte da elite da população.

Os parâmetros do AG foram descritos na seção 2.3.1. Por questões práticas, foi adotada a base binária. A inicialização da população foi um dos primeiros problemas encontrados, como variáveis angulares são cíclicas e o manipulador possui um espaço de trabalho restrito, a inicialização da população não pode ser feita de forma aleatória. Então, restringiu-se à duas situações de escolha de pais, em que foram utilizados os valores máximos e mínimos de cada articulação.

A primeira população foi escolhida com poucos geradores, a condição de contorno foi estabelecida para variações de mínimo e máximo de apenas um dos ângulos por pai, totalizando em um número de sete pais (Tabela 7).

Tabela 7: Primeira população inicial (Pais1).

Pais1		
0	0	0
0	0	<i>min_th4</i>
0	0	<i>max_th4</i>
0	<i>min_th3</i>	0
0	<i>max_th3</i>	0
<i>min_th2</i>	0	0
<i>max_th2</i>	0	0

Pela análise do manipulador estudado, pode-se notar que o valor do *th_1* é encontrado facilmente (única rotação no eixo z é feita pela base), cabe então ao algoritmo encontrar os outros três ângulos restantes. Após uma série de testes com os Pais1, foram alcançados tais parâmetros.

Tabela 8: Melhores parâmetros para Pais1.

Parâmetro	Taxa
<i>mutacao</i>	0.10
<i>elitismo</i>	0.20
<i>mortalidade</i>	0.20
<i>n_geracao_max</i>	10

Como o AG é um método probabilístico, o seu resultado tem muitas variações, fazendo-se necessária uma grande repetição da mesma operação analisar o desempenho da função, como mostrado na Tabela 9. A Figura 20 representa a posição final encontrada por cada um dos dados da tabela.

Tabela 9: Dados da Simulação 1.

Repetição	Aproximação (mm)	tempo(s)
1	31.4754	0.416638
2	10.6035	0.571074
3	13.8075	0.459263
4	16.5188	0.505531

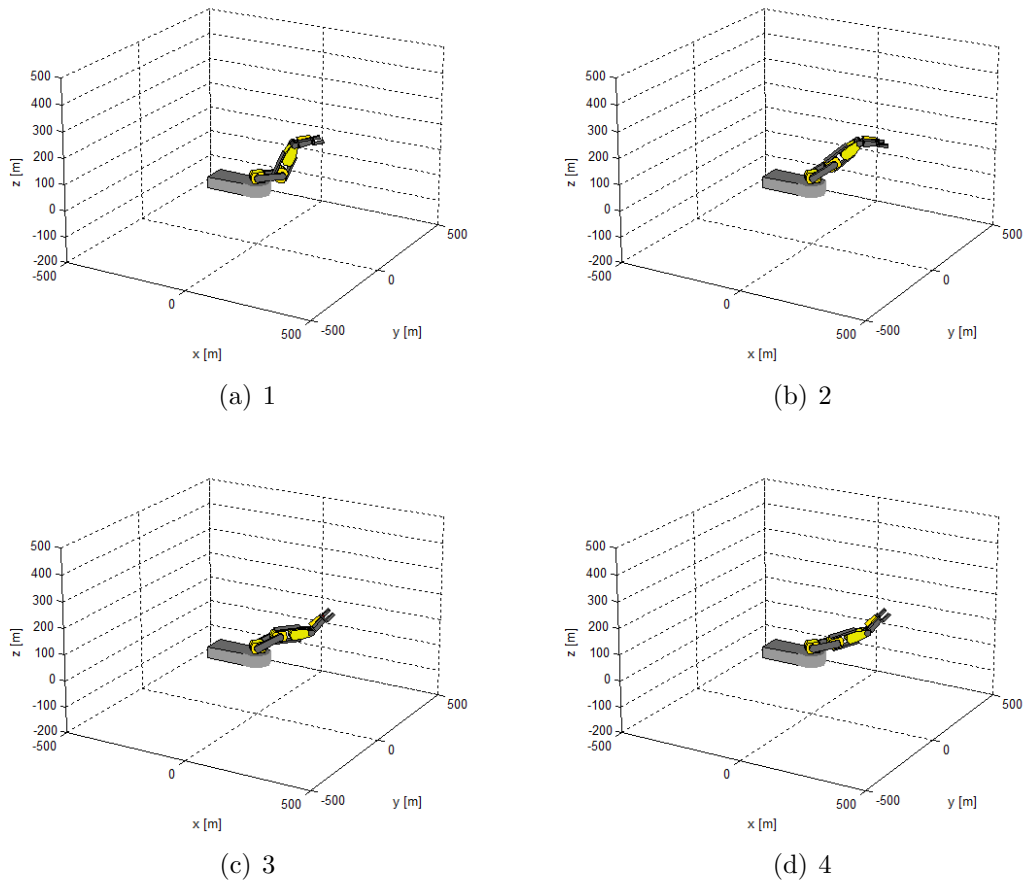


Figura 20: Simulação Pais1.

O segundo grupo de pais testados é uma combinação entre os ângulos máximos e mínimos, formando 27 indivíduos que constituem a matriz Pais2. Da mesma forma anterior, depois de vários testes, foi encontrada a Tabela 10. A Figura 21 representa a posição final encontrada por cada um dos dados da tabela.

Tabela 10: Melhores parâmetros para Pais 2.

Parâmetro	Taxa
<i>mutacao</i>	0.10
<i>elitismo</i>	0.05
<i>mortalidade</i>	0.35
<i>n_geracao_max</i>	8

Tabela 11: Dados da Simulação 2

Repetição	Aproximação (mm)	tempo(s)
1	6.6026	0.297208
2	0.5166	0.314112
3	6.6026	0.308644
4	13.7122	0.334570

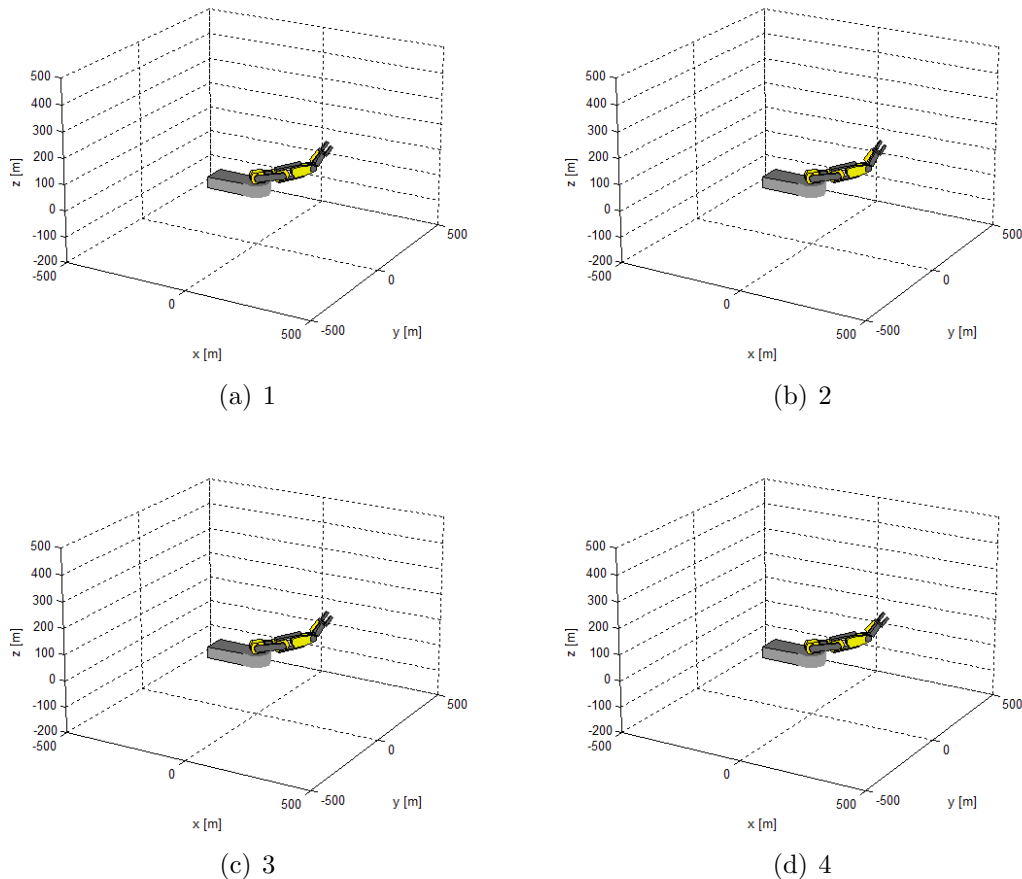


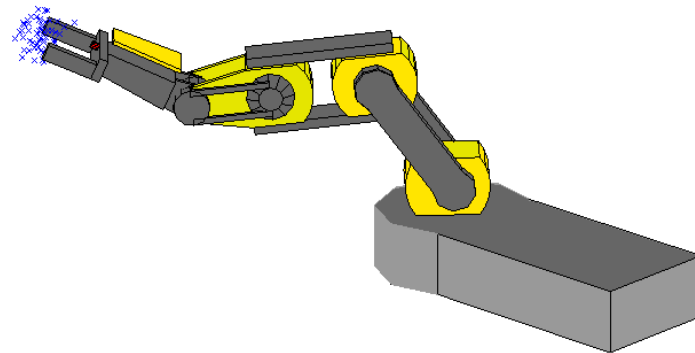
Figura 21: Simulação Pais2.

Um fator que influencia muito na posição final é a função de energia mínima, porém para uma melhor aproximação, o função de aptidão do manipulador tem uma preferência maior na aproximação do ponto. A mínima energia faz com que o manipulador se aproxime do ponto em um menor esforço, considerando que a posição *home* ocorre quando os ângulos das articulações são nulos, ou seja, o manipulador fica na horizontal.

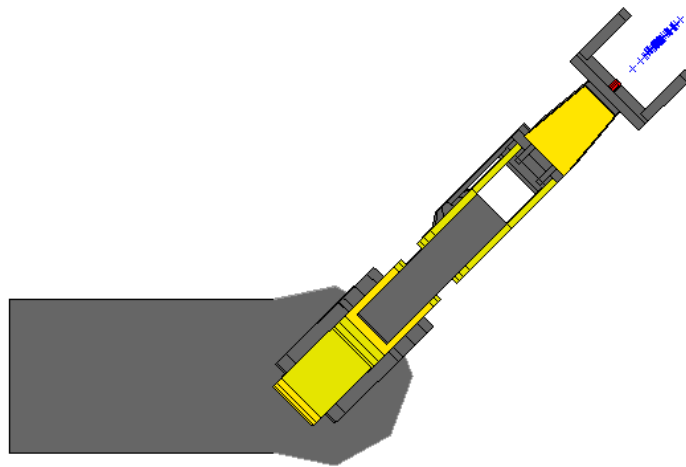
O grupo Pais2, por ser inicialmente maior e possuir uma maior diversidade, precisou de um número menor de gerações para convergir. Os valores das aproximações foram melhores e próximos entre si e o tempo em que a função gastou se manteve constante. As simulações foram feitas para uma variação simples de ângulo (45° na cintura, 36° no

ombro, 60° no cotovelo e -30° no punho), porém o simulador funciona igualmente para diversas configurações.

Foi realizada a mesma simulação do AG para o mesmo ponto 500 vezes, na Tabela 11 foram mostrados apenas os quatro primeiros dados. A média das aproximações foram de $6.8688mm$. A Figura 22 exemplifica a área de variação dos resultados do AG. Mostrando que para a utilização do AG faz com que o simulador tenha um boa precisão e repetibilidade.



(a) Vista Lateral



(b) Vista Superior

Figura 22: Simulador na posição desejada com os pontos calculados pelo AG.

Analisando a Figura 22, observa-se a necessidade de utilizar um controle de posição para "inserir" a área de soluções no interior do espaço de trabalho do efetuador aumentaria a eficiência das soluções para o manipulador.

A rotina do algoritmo genético é utilizado somente para encontrar o ponto desejado, em seguida é utilizado um controle linear de incremento angular, para que o simulador alcance a posição desejada, gerando assim uma noção da movimentação.

4 *Conclusões*

O modelo final do simulador apresentou-se generalizado, fazendo-se uma ótima ferramenta de aplicações em outros trabalhos. Estudo do controle de posição e trajetória pode ser um dos focos para o simulador nos trabalhos subsequentes.

O algoritmo comportou-se muito bem nas variações dos parâmetros, obteve aproximações com um erro de máximo de $21.8304mm$ e mínimo de $0.5166mm$. Nas simulações, a média das distâncias ao ponto desejado foi de $6.8688mm$. Como o manipulador tem uma altura de $381mm$ o mesmo possui um erro máximo de 5.7298% na posição final. Por ser um manipulador pequeno e sem precisão física, a variação de no máximo $22mm$ é um resultado satisfatório.

Como trabalhos futuros, pode-se exportar o simulador do software, para que o mesmo possa ser utilizado em uma janela executável com botões de parâmetros angulares. Faz-se interessante também a criação da interface entre o manipulador real e o simulador, assim após programado o manipulador real iria efetuar os movimentos similares ou idênticos aos do simulador.

Referências

- [1] ZWIRTES, R. A. *Cinemática Inversa para Controle da Abordagem de Órgãos Terminais de Robôs Manipuladores*. [S.l.], 2004.
- [2] SPONG, S. H. M. W.; VIDYASAGAR, M. *Robot Modeling and Control*. [S.l.]: John Wiley & Sons, 2005.
- [3] ASIMOV, I. I. *Robot*. [S.l.]: Ed. Bantam, 2004.
- [4] ASADA, H.; SLOTINE, J. J. E. *Robot Analysis and Control*. [S.l.]: Cambridge: MIT Press, 1985.
- [5] CRAIG, J. J. *Introduction to Robotics: Mechanics and Control*. [S.l.]: AddisonWesley Publishing Company, Inc, 1986.
- [6] DIAS, A. M. *Controle de Manipulador Redundante Utilizando Realimentação Visual*. Dissertação (Mestrado) — UFRN, Natal, 2002.
- [7] SANTOS, H. S. L. e. M. G. J. A. F. Cinemática inversa de trajetória de manipuladores robóticos redundantes utilizando algoritmos genéticos com redução progressiva do espaço de busca. *II IEEE LARS*, Setembro 2005.
- [8] SILVA, I. N. Avanços em tecnologias de sistemas inteligentes e aplicações. In: *XVIII Congresso Brasileiro de Automática*. Ilha Solteira: [s.n.], 2010. v. 1, p. 75–81.
- [9] NUNES, L. E. N. P. *Avaliação de Diferentes Técnicas para Reconhecimento de Fala*. Tese (Doutorado) — USP, 2007.
- [10] COELHO, L. S. C. e A. A. R. Algoritmos evolutivos em identificação e controle de processos: uma visão integrada e perspectivas. *SBA Controle & Automação*, Jan., Fev., Mar., Abril 1999.
- [11] HOSOGOSHI, A. C. C. R. e M. C. *Estudos dos Algoritmos Genéticos Usando Técnicas Computacionais*. [S.l.], 2005.
- [12] CATARINA, A. S. *Um Algoritmo Genético Híbrido com Representação Explícita de Relacionamentos Espaciais para Análise de Dados Geoespaciais*. Tese (Doutorado) — INPE, 2009.
- [13] PACHECO, M. A. C. *Algoritmos genéticos: princípios e aplicações*. [S.l.]: PUC/RIO, 1999.