UNIVERSIDADE FEDERAL DE SANTA CATARINA
CAMPUS FLORIANÓPOLIS
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA MECÂNICA

RENAN ZUBA PARRELA

TÍTULO: PLANEJAMENTO DE MOVIMENTOS DE ROBÔS MANIPULADORES

FLORIANÓPOLIS
2021

Renan Zuba Parrela

TÍTULO:

PLANEJAMENTO DE MOVIMENTOS DE ROBÔS MANIPULADORES

Dissertação submetida ao Programa de Engenharia Mecânica da Universidade Federal de Santa Catarina para a obtenção do título de mestre em Engenharia Mecânica
Orientador: Prof. Dr. Edson Roberto De Pieri

Florianópolis
2021

Ficha de identificação da obra

Parrela, Renan Zuba
Planejamento de Movimentos de Robôs manipuladores / Renan Zuba Parrela; orientador, Edson Roberto de Pieri, 2020. 155 p.

Dissertação (mestrado) - Universidade Federal de Santa Catarina, Centro Tecnológico, Programa de Pós-Graduação em Engenharia Mecânica, Florianópolis, 2020.

Inclui referências.

1. Engenharia Mecânica. 2. Planejamento de Movimentos. 3. Otimização Energética. 4. Algoritmos Genéticos. I. de Pieri, Edson Roberto. II. Universidade Federal de Santa Catarina. Programa de Pós-Graduação em Engenharia Mecânica. III. Título.

Renan Zuba Parrela

Título: Planejamento de Movimentos de Robôs Manipuladores

O presente trabalho em nível de mestrado foi avaliado e aprovado por banca examinadora composta pelos seguintes membros:

Prof. Antônio Carlos Valdiero, Dr.
Universidade Federal de Santa Catarina

Prof. Ebrahim Samer El Youssef, Dr.
Universidade Federal de Santa Catarina

Prof. Edson Roberto De Pieri, Dr.
Universidade Federal de Santa Catarina

Certificamos que esta é a **versão original e final** do trabalho de conclusão que foi julgado adequado para obtenção do título de mestre em Engenharia Mecânica

Documento assinado digitalmente
Paulo de Tarso Rocha de Mendonca
Data: 16/03/2021 17:01:58-0300
CPF: 149.822.691-49
Verifique as assinaturas em https://v.ufsc.br

Prof. Dr. Paulo de Tarso Rocha de Mendonça
Coordenador do Programa

Documento assinado digitalmente
Edson Roberto de Pieri
Data: 12/03/2021 11:15:43-0300
CPF: 002.042.498-13
Verifique as assinaturas em https://v.ufsc.br

Prof. Dr. Edson Roberto De Pieri
Orientador

Florianópolis, 29 de Janeiro de 2021.

À minha família.

AGRADECIMENTOS

Gostaria de agradecer as seguintes pessoas:

Ao meu orientador, Prof. Dr. Edson Roberto De Pieri, pela orientação, confiança e paciência depositada durante realização desta pesquisa.

Aos professores do POSMEC, os quais tive a oportunidade de ser aluno e extrair o máximo de seus conhecimentos e experiências que pude.

À minha família, pelo suporte e apoio tanto financeiro quanto moral que me deram, especialmente a minha mãe, Roziane Pereira Zuba, por ser uma mulher incrível e não medir esforços para a concretização de meus sonhos.

À família Mendes, pelo carinho, apoio e acolhimento de sempre, em especial, a minha namorada Maria Eduarda, por ser companheira, amorosa e fazer meus dias mais felizes.

Aos colegas e amigos que fiz ao decorrer do Mestrado, principalmente Gabriel, Kleverton e Alisson, pelas conversas e incentivos prestados.

A todos os outros que, de certa forma, são coadjuvantes em meu trajeto e contribuíram direta ou indiretamente para a realização deste trabalho.

Obrigado.

A vida não dá e nem empresta, não se comove e nem se apieda. Tudo que ela faz é retribuir e transferir aquilo que nós lhe oferecemos.

(Albert Einstein)

RESUMO

Este trabalho se trata de um estudo sobre os métodos utilizados no planejamento de movimentos de robôs manipuladores, principalmente daqueles que buscam otimização energética. Foram construídos dois modelos de simulação, ambos utilizando o Braço Antropomórfico com Punho Esférico. O primeiro modelo objetivava o planejamento de movimentos livres de obstáculos, que inclui o projeto do sistema de controle do manipulador, o planejamento de trajetórias e de tarefas. Foi necessária a solução dos problemas de Cinemática Inversa e Direta. Construiu-se um modelo geométrico do robô no Simscape™, para que ele fosse tratado como um sistema de dinâmica desconhecida. Logo, realizou-se a identificação da dinâmica do manipulador através da estimação de funções de transferências, utilizando o método dos mínimos quadrados. O projeto do controlador foi realizado por meio de um Algoritmo Genético, que atuava selecionando os parâmetros do controlador PD com filtro que remetem a valores mínimos do índice ITAE sob restrições de sobressinal. As trajetórias do robô foram confeccionadas considerando desvio de obstáculos, elas eram geradas através da *B-Spline* cúbica no espaço cartesiano e os obstáculos eram modelados através de elipsoides. Foram confeccionadas três tarefas para validação dos métodos, nas quais o robô desviava de obstáculos, agarrava objetos e desenhava sobre uma mesa. No segundo modelo de simulação, o planejamento de trajetórias é visto como um problema de otimização energética. A trajetória é gerada através de polinômios no espaço de configuração, nos quais os coeficientes dos polinômios são escolhidos de maneira a minimizar uma função de esforço mecânico e evitar restrições. O cálculo da função depende da solução da Dinâmica Inversa do manipulador. Para isso, implementou-se o algoritmo recursivo de Newton-Euler, devido ao robô apresentar muitos graus de liberdade. O método de otimização utilizado também foi o Algoritmo Genético, em virtude da dinâmica complexa e não-linear dos robôs. Constatou-se que a quantidade de esforço mecânico evitado depende das condições iniciais e finais impostas e que existe um *trade-off* entre a diminuição do esforço mecânico e a adição de restrições nos polinômios. Ambos os métodos foram efetivos em suas demandas. A vantagem do primeiro é sua simples e intuitiva implementação, enquanto no segundo, as trajetórias possuem esforços mecânicos reduzidos ao mesmo tempo que são livres de obstáculos e suaves nas extremidades.

Palavras-chave: Planejamento de Trajetória. Otimização Energética. Algoritmos Genéticos.

ABSTRACT

This work is a study on planning manipulator robot trajectories, especially those that aim energy optimization. Two simulation models were built, both using the Anthropomorphic Arm with Spherical Wrist. The first model considers the planning of obstacle-free movements, including the design of the manipulator control system, the planning of trajectories and tasks. It was necessary to solve the problems of Inverse and Direct Kinematics. A geometric robot model was built in Simscape™ so that it would be treated as a system of unknown dynamics. Therefore, the manipulator dynamics were identified through the estimation of transfer functions using the least-squares method. The controller project was carried out using a Genetic Algorithm, which worked by selecting the PD controller parameters with a low pass filter that refers to the ITAE index's minimum values under overshoot constraints. The robot's trajectories were made considering obstacle deviation, they were generated through the cubic B-Spline in Cartesian space, and the obstacles were modeled using ellipsoids. Three tasks were carried out to validate the methods, in which the robot avoided obstacles, grabbed objects, and drew on a table. In the second simulation model, trajectory planning is seen as an energy optimization problem. The trajectory is generated through polynomials in the configuration space, in which the coefficients of the polynomials are chosen in order to minimize a mechanical effort function and avoid restrictions. The function calculation depends on the Inverse Dynamics solution of the manipulator. For this, the Newton-Euler recursive algorithm was implemented, due to the robot having many degrees of freedom. The optimization method was also the Genetic Algorithm due to the robot's complex and nonlinear dynamics. It was found that the amount of mechanical effort avoided depends on the initial and final conditions imposed and that there is a trade-off between the reduction of mechanical stress and the addition of restrictions in the polynomials. Both methods were effective in their demands. The advantage of the first is its intuitive and straightforward implementation. In contrast, the trajectories have reduced mechanical efforts for the second while being free of obstacles and smooth at the ends.

Keywords: Trajectory Planning, Energy Optimization, Genetic Algorithm.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

CAD	*Computer-Aided Design*
CCD	*Cyclic Coordinate Descente*
DLS	*Damped Least-Squares*
FIK	*Feedback Inverse Kinematics*
PRM	*Probabilistic Roadmap Method*
RRT	*Rapidly exploring random trees*
CHOMP	*Covariant Hamiltonian Optimization for Motion Planning*
ITOMP	*Incremental Trajectory Optimization Algorithm*
STOMP	*Stochastic Trajectory Optimization for Motion Planning*
DC	*Direct Current*
SQP	*Sequential Quadratic Programming*
DDP	*Discrete Dynamic Programming*
IDP	*Interative Dynamic Programming*
PD	Ações Proporcional e Derivativa
PID	Ações Proporcional, Integrativa e derivativa
AG	Algoritmo Genético
SISO	*Single Input Single Output*
ITAE	*Integral Time Absolute Error*
ISE	*Integrative Square Error*
IAE	*Integral Absolute Error*
IE	*Integral Error*
ME	Mínimo Esforço Mecânico
C-*Space*	*Configuration Space*
SISO	*Single Input Single Output*

LISTA DE SÍMBOLOS

a_i	Coeficiente independente do Polinômio relacionado a junta i.
$B_{i,n}$	Polinômios de Bernstein de grau n.
τ_i	Torque da junta i
C	Matriz de Forças Centrífugas
m	Massa
L_i	Comprimento do elo i
G	Vetor de força gravitacional
q_i	Deslocamento de uma variável generalizada da junta i
$\dot{q}_i$	Velocidade de uma variável generalizada da junta i
$\ddot{q}_i$	Aceleração de uma variável generalizada da junta i
θ_i	Ângulo de rotação de uma junta i
$\dot{\theta}_i$	Velocidade angular de uma junta i
$\ddot{\theta}_i$	Aceleração angular de uma junta i
R_i^o	Matriz de rotação de um ponto em i em relação a um referencial o.
P	Posição do Efetuador Final
r_i^o	Vetor de posição de ponto em i visto sobre a perspectiva do referencial o.
$r_{i-1,i}^o$	Vetor de posição de ponto em i em relação a um ponto em $i-1$ visto sobre a perspectiva do referencial o.
$\dot{p}_i^o$	Velocidade linear de um ponto p em i visto sobre a perspectiva do referencial o.
$\ddot{p}_i^o$	Aceleração linear de um ponto p em i visto sobre a perspectiva do referencial o.
ω_i^o	Velocidade angular de ponto em i em relação a um referencial o.
$\dot{\omega}_i^o$	Aceleração angular de ponto em i em relação a um referencial o.
J	Matriz Jacobiana
A_i^o	Matriz de transformação de um ponto i em relação a um referencial o.
ϕ	Orientação do Efetuador Final.
M	Matriz de inércia do manipulador
$\bar{I}^o$	Tensor de inércia em um referencial o.
I_{XY}	Produtos de inércia relativos aos eixos “x” e “y”.
G_P	Função de Transferência do Processo
G_C	Função de Transferência do Controle
G_{MF}	Função de Transferência em Malha Fechada
K_C	Vetor de constantes de controle
K_p	Constante de ganho proporcional
K_I	Constante de ganho integral
K_D	Constante de ganho derivativo
τ_F	Constante de tempo do filtro
$N_{i,k}$	Funções de base B-Spline.

SUMÁRIO

CAPÍTULO 1: INTRODUÇÃO

Devido às características de versatilidade e flexibilidade proporcionadas pela arquitetura e controle, os robôs manipuladores substituíram a automação tradicional em determinados âmbitos quando foram introduzidos na segunda metade do século XX (Lozano-Pérez, 1983). Hoje, com os avanços tecnológicos, essas máquinas estão cada vez mais eficientes e precisas, trabalhando de maneira eficiente e multifuncional (Gamero, 2018).

Um robô manipulador é formado por diferentes segmentos, ou elos, que são unidos através de articulações (ou juntas). O número dessas articulações é determinado de acordo com a mobilidade desejada no espaço de trabalho (Simas, 2008). Para que um robô realize tarefas, é necessário que o efetuador final dele percorra um caminho específico com determinadas velocidades e acelerações das juntas, o que constituí um problema de planejamento de trajetórias. Ademais, quando o ambiente do robô possui obstáculos, o movimento deve ser planejado de maneira a evitar colisões entre eles e o manipulador.

Muitas estratégias podem ser encontradas na literatura para o planejamento de trajetórias livres de obstáculos. As trajetórias e os obstáculos podem ser modelados tanto no espaço das juntas quanto no cartesiano. Elas podem ser planejadas de uma maneira *on-line* e *off-line*. Podem ser considerados ambientes dinâmicos ou estáticos. Nos ambientes dinâmicos, o algoritmo é implementado para funcionar de maneira online, permitindo que o robô reaja a movimentos externos (Khatib, 1985, Bosscher & Hedman, 2009, Polverini et al. 2014). Em ambientes estáticos há um conhecimento mais minucioso do ambiente, permitindo a confecção de trajetórias com melhores desempenhos. Cada estratégia possui vantagens intrínsecas, cabendo ao projetista escolher a melhor baseando-se em seus requisitos.

Nas últimas décadas, o aumento do preço da energia e a preocupação com o meio ambiente fez com que os cientistas e engenheiros buscassem novas soluções para redução de gastos energéticos em manufaturas (Carabin *et al*., 2017). Na indústria de robôs, entre as formas utilizadas para redução de gastos, estão inclusas a adoção de sistemas mecatrônicos com *designs* mais leves e a minimização da energia demandada pelos atuadores durante a execução dos movimentos (Carabin *et al*., 2017). Em vista disto, o planejamento de movimentos, quando alicerçado na necessidade de economia energética, pode ser visto como um problema de otimização.

1.1. Objetivo Geral

O objetivo desta dissertação é planejar movimentos de robôs manipuladores quando são dadas duas posições, uma inicial e outra final. Os movimentos devem ser livres de obstáculos e otimizados de maneira que demandem menos energia dos atuadores.

1.2. Objetivos Específicos

- Pesquisar sobre as principais estratégias utilizadas para o planejamento de trajetórias de manipuladores com otimização energética;
- Escolher um robô manipulador do tipo serial e resolver seus problemas de cinemática e dinâmica inversas, aplicar a solução desses problemas em um modelo geométrico utilizando algum *software*;
- Propor um sistema de controle para o manipulador;
- Desenvolver um Algoritmo de Planejamento de trajetórias livres de colisões;
- Redefinir o planejamento de trajetórias livres de colisões como um problema de otimização energética.

1.3. Contribuições

O estudo realizado na dissertação apresenta dois métodos que são utilizados para o planejamento de movimentos de manipuladores. Os métodos podem ser aplicados em robôs industriais reais para desviar de obstáculos ou diminuir o consumo energético. Além disso, os resultados obtidos nesta pesquisa podem servir como fonte de referência a futuros trabalhos na área de planejamento de movimentos de robôs.

1.4. Estrutura do Trabalho

Além desta Introdução, o trabalho está estruturado em mais cinco Capítulos, Referências e cinco Apêndices que abordam os seguintes temas:

O Capítulo 2 apresenta o estado da arte do planejamento de trajetórias de robôs manipuladores. São expostos os métodos clássicos, métodos de otimização e, especificamente, os métodos de otimização quando se utiliza a energia como critério.

O Capítulo 3 apresenta a teoria associada à modelagem e ao controle de robôs manipuladores. Inicialmente, são expostos os modelos matemáticos utilizados para abstrair as

características essenciais dos robôs. Ele é analisado sob duas perspectivas: Cinemática e Dinâmica. Em seguida, é apresentado um método para o controle de movimentos desses manipuladores.

O Capítulo 4 trata acerca do planejamento de trajetórias de manipuladores. São apresentadas as curvas contínuas, que são mais empregadas, a modelagem matemática de obstáculos, os quais podem estar presentes no ambiente de trabalho do robô, e um algoritmo para o planejamento de trajetórias livres de obstáculos.

O Capítulo 5 apresenta o planejamento de movimentos livres de obstáculos de um manipulador, que integra o problema de controle e os planejamentos de trajetórias e tarefas. Nesse capítulo os métodos apresentados nos capítulos anteriores e nos apêndices são unificados para a realização de simulações.

O Capítulo 6 redefine o planejamento de trajetórias como um problema de otimização energética. Foram realizados dois experimentos nesse capítulo. Um em que as trajetórias são planejadas na ausência de obstáculos e outro para serem livres de colisões quando são considerados obstáculos no espaço de trabalho.

O Capítulo 7 apresenta as considerações finais acerca desta dissertação, bem como uma lista de sugestões para trabalhos futuros.

O Apêndice A fornece os dados a respeito do manipulador Braço Antropomórfico com Punho Esférico empregado nas simulações.

O Apêndice B apresenta a solução do problema de Cinemática Inversa e as singularidades do manipulador.

O Apêndice C concede dados das curvas utilizadas para geração dos gráficos no Modelo de Simulação “Otimização Energética no Planejamento de Trajetórias”.

O Apêndice D apresenta as equações dos polinômios de 3ª até 8ª ordem utilizados no Modelo de Simulação “Otimização Energética no Planejamento de Trajetórias”.

O Apêndice E apresenta o método para identificação de sistemas empregado, o estimador “Mínimos Quadrados não recursivo”.

O Apêndice F apresenta o Algoritmo Genético, principal método de otimização utilizado no trabalho.

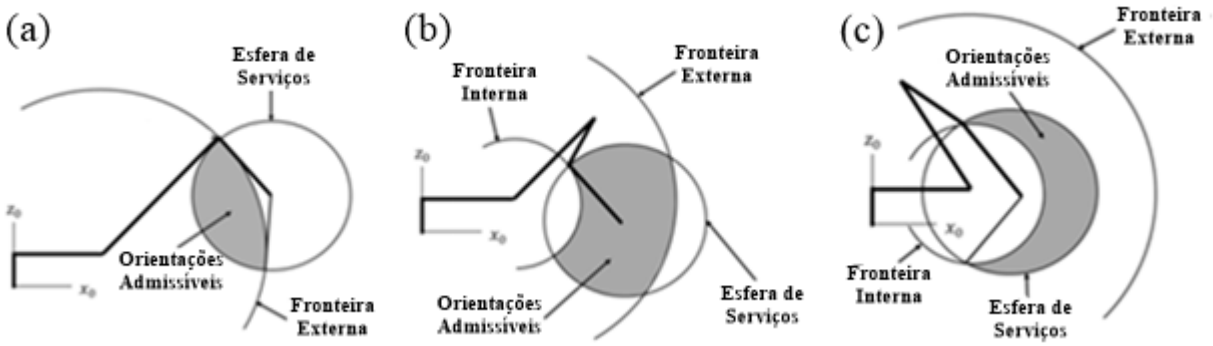
CAPÍTULO 2: ESTADO DA ARTE

Em processos industriais de montagem, embalagem e inspeção, grande repetibilidade é requerida. Neles, os robôs realizam tarefas do tipo *pick-and-place*: se movem entre determinadas posições apanhando e transportando objetos. Quando se trata do planejamento de trajetórias entre duas posições, podem ser geradas infinitas trajetórias e, por isso, existem inúmeros métodos de planejamento. De acordo com Petrovic (2018), os métodos utilizam dois paradigmas na maior parte das vezes: o primeiro subdivide o planejamento de trajetória em dois outros problemas, o planejamento de caminho e sua execução temporal, e o segundo determina a trajetória considerando restrições dinâmicas, nesse caso, o caminho e a lei de tempo são calculados simultaneamente. Além disso, ainda que uma trajetória possa ser consideravelmente complexa, ela pode ser subdividida em segmentos, transformando-se em um problema de cálculo entre dois pontos e de continuidade entre os segmentos.

A estratégia mais intuitiva na solução da trajetória entre dois pontos é determinar a reta que os interpola no espaço cartesiano e especificar as restrições de tempo em suas coordenadas. Caso exista obstáculos ao longo da reta, realizam-se desvios de maneira a evitar colisões com os obstáculos. Essa estrategia foi utilizada por Mondragon (2013), que propos um algoritmo que gera um caminho retilíneo entre as duas posições e os obstáculos são modelados matematicamente com formatos de elipsoide. Quando é detectado que as coordenadas da linha estão dentro da região do elipsoide, o que nesse caso seria uma rota de colisão entre o robô e o obstáculo, as coordenadas da linha não são incluídas no caminho desejado, mas sim, as coordenadas do elipsoide e, dessa forma, o caminho da trajetória é modificado e a colisão é evitada.

Mondragon (2013) apresenta também o conceito de análise de destreza, que consiste em verificar todas as possibilidades de orientações que o efetuador final poderia possuir em localizações particulares. As possibilidades para uma determinada posição possuem um formato esférico que é chamado de Esfera de Serviços. As orientações que são admissíveis são calculadas verificando três regiões: da Esfera de Serviços e das fronteiras dos espaços de trabalho interno e externo do robô. A Fig. 1. ilustra três situações distintas em que as orientações admissíveis foram calculadas, exibidas em uma coloração cinza. Na Fig. 1 (a), o manipulador está próximo da fronteira externa, na Fig. 1 (b), em uma região intermediária e, na Fig. 1 (c), próximo à fronteira interna.

Figura 1: Cálculo das orientações admissíveis.



Fonte: Adaptado de Mondragon (2013).

Pode-se ainda variar a velocidade e a aceleração do efetuador final como um problema de otimização energética e de tempo, conforme realizado por Bailón *et al.* (2010). Todavia, planejar trajetórias no espaço cartesiano não assegura facilmente que os motores produzam movimentos suaves. Além disso, existe a necessidade de resolver problemas de singularidade, devido o cálculo da Cinemática Inversa em tempo real e, eventualmente, de redundância, em consequência do robô ser construído com um número maior de juntas para lhe conferir maior destreza para a realização de tarefas complexas.

Existem técnicas desenvolvidas para evitar singularidades. Segundo Serapião (2017), a maioria dessas técnicas é baseada em métodos numéricos e de otimização. Destaque para os métodos: CCD (*Cyclic Coordinate Descente*), baseado em programação não-linear, DLS (*Damped Least-Squares*), baseado no cálculo da Pseudo-Inversa do Jacobiano com fator de amortecimento, e FIK (*Feedback Inverse Kinematics*), que emprega um laço de realimentação para minimizar a diferença entre velocidade atual e desejada no espaço operacional.

A seguir são apresentados os principais métodos de planejamento de trajetórias, que foram subdivididos em Métodos Clássicos, Seção 2.1., Métodos de Otimização, Seção 2.2., e os Métodos que levam em conta critérios baseado em energia, Seção 2.3..

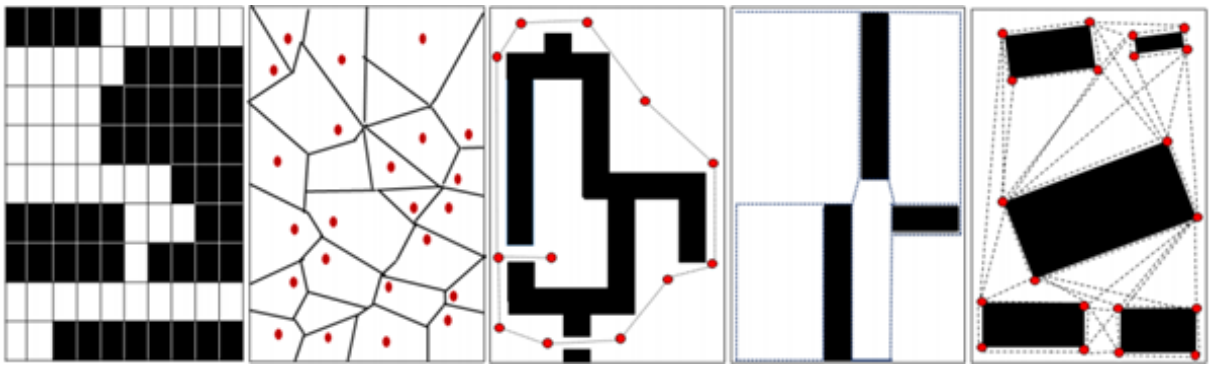
2.1. Métodos Clássicos

Outras técnicas realizam o planejamento de trajetórias diretamente no espaço de configuração, para evitar ambos os problemas, tanto de singularidades quanto de redundância.

Nesse espaço, já existem métodos que são considerados clássicos e que funcionam computando o espaço de configuração do robô (também conhecido como *C-space*). Segundo Hentout *et al.* (2008), tais métodos podem ser divididos em três categorias: globais, locais e mistos.

Os métodos globais são baseados no conhecimento completo do espaço de configuração, são chamados de g*rid-based approaches (*Petrovic, 2018) (métodos baseados em grades). Nesses métodos, determina-se o espaço livre no *C-space* do robô e o subdivide em grades (que são modeladas por grafos) que capturam a topologia do mapa. Alguns exemplos desses métodos podem ser observados na Fig. 2. Da esquerda para direita, os métodos são denominados, respectivamente: *Regular grids, Voronoi Diagram*, *Waypoints Graph*, *Navigation Meshes* e *Visibility Graph*. Uma vez que os mapas de estrada são construídos, a busca pelo caminho mais curto é realizada por técnicas padrões de busca em grafos tais como os algoritmos Dijkstra ou A* (Llopis-Abert *et al*. 2018).

Figura 2: Alguns métodos baseados em grades.



Fonte: Majeed & Lee (2018).

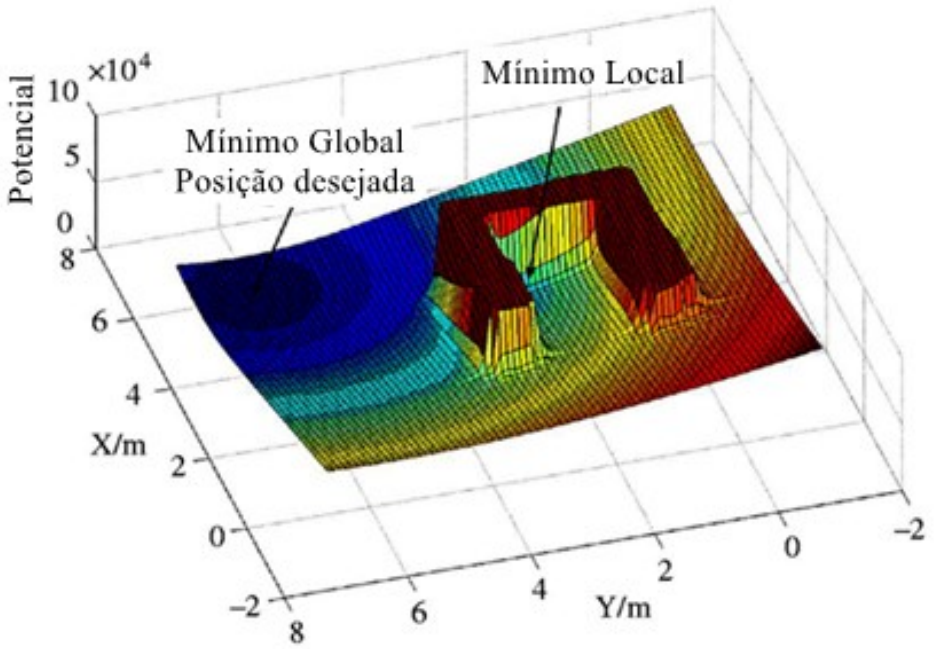
Os métodos locais precisam apenas de um conhecimento parcial do espaço de trabalho do robô. Eles modificam a trajetória à medida que o robô avança. Em cada estágio, o sistema de controle checa a existência de possíveis obstáculos. Uma das vantagens desses métodos é de possibilitar a capacidade de reação do robô, uma vez que as trajetórias são planejadas de maneira *online*. O método do campo potencial artificial é o exemplo mais conhecido entre tais métodos (Khatib, 1985). Por ser calculado online, esse método permite o desvio de obstáculos

dinâmicos, mas o custo computacional e a necessidade de respostas rápidas para viabilizar o cumprimento de tarefas, são suas principais desvantagens.

Os métodos mistos buscam combinar as principais vantagens das abordagens global e local. Geram uma trajetória inicial *offline* e depois, de maneira análoga aos métodos locais, adaptam-na progressivamente à medida que o robô avança (Hentout, 2008).

De acordo com Llopis *et al.* (2018), as abordagens clássicas possuem certas desvantagens que as fazem ser ineficientes na prática. Elas tendem a ficar presas em mínimos locais (Fig. 3), levam a problemas difíceis de tempo polinomial não-determinísticos (*NP-hard*) na presença de muitos obstáculos e apresentam soluções complicadas quando a dinâmica do ambiente é complexa. Os métodos baseados em grades, por exemplo, podem resultar em um grande custo computacional para determinar uma trajetória livre de colisões: o número de grades aumenta exponencialmente quando se aumenta os graus de liberdade do robô, fazendo com que mesmo os métodos mais eficientes sejam inadequados para tratar problemas com essa complexidade.

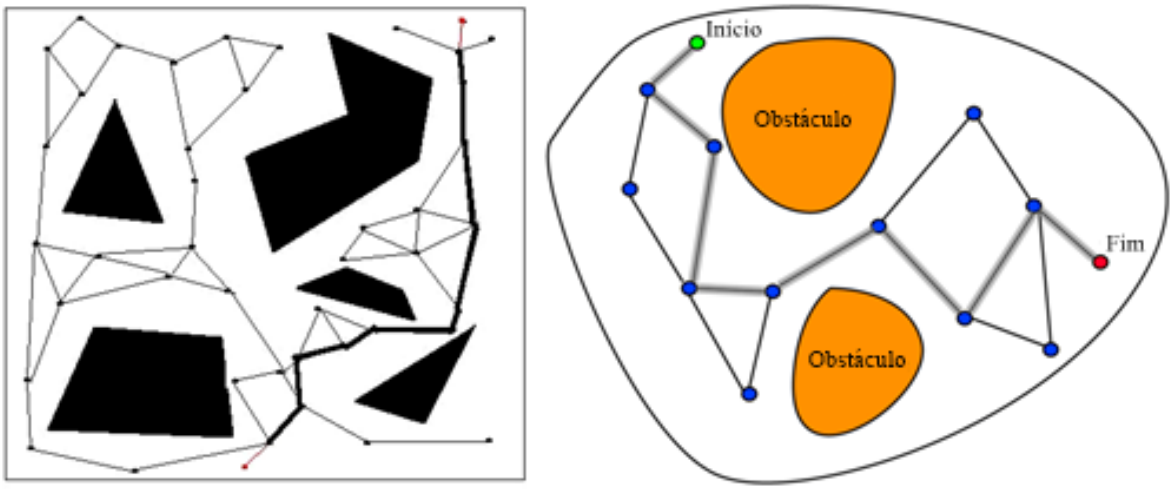
Figura 3: Campo potencial artificial com mínimo global e local.



Fonte: Adaptado de Palmieri, L. (2015).

Uma abordagem que não se encaixa nas três apresentadas anteriormente diz respeito aos métodos baseados em amostragem (*sampling-based methods)*. Eles basicamente mapeiam o espaço configuracional com pontos. Dessa forma, o caminho é definido conectando os pontos da amostragem entre os pontos inicial e final de acordo com algum critério (Fig. 4).

Figura 4: Exemplo de funcionamento de método baseado em amostragem.



Fonte: Adaptado de Girdhar (2005) e Masehian & Sedighizadeh (2010).

Os métodos mais comumente utilizados dessa categoria são *Probabilistic Roadmap Method* (PRM) e *Rapidly exploring random trees* (RRT). Apesar de serem completos probabilisticamente e eficientes em problemas práticos, não asseguram trajetórias ótimas, necessitando um pós-processamento. Devido a isso, de maneira análoga aos métodos baseados em grades, podem ser ineficientes em espaços configuracionais com muitos graus de liberdade (Petrovic, 2018).

2.2. Métodos de Otimização

Dentro da perspectiva de busca de eficiência e de soluções ótimas, vários pesquisadores vêm tratando o planejamento de trajetórias como um problema de otimização. Na otimização são incluídas restrições que asseguram que robô não realize movimentos indesejados que são divididas em dois tipos: restrições de sistema e de tarefa. As restrições de sistema são impostas pela dinâmica do manipulador, estabelece quais valores de posições, velocidades, acelerações

e arranques são permitidos. As restrições de tarefa são intrínsecas à tarefa a ser realizada e podem ser, por exemplo, obstáculos que estão presentes no ambiente (Ata, 2007).

Os critérios mais utilizados na otimização, aqueles que apresentam melhores resultados em termos de performance das trajetórias, possuem significados diferentes para os pesquisadores (Ata, 2007). Alguns consideram minimizar o tempo de ciclo ou gasto energético, outros acreditam que minimizar o arranque (*jerk*), a terceira derivada da posição em relação ao tempo, é mais vantajoso.

Segundo Ratiu (2017), o critério de otimização mais comumente utilizado na robótica é o tempo de ciclo, pois possui uma conexão direta com a redução do tempo de manufatura, que resulta em um aumento de produtividade. Um exemplo de método para redução do tempo de ciclo é o algoritmo desenvolvido por Kim *et al*. (2007), que funciona de maneira *online* e considera em sua estrutura a dinâmica do manipulador. Além disso, os valores de torques permitidos nas juntas são limitados. Segundo os autores, quando a dinâmica do manipulador é considerada, existe uma redução considerável na eficiência global de capacidade do manipulador em relação aos métodos tradicionais.

Utilizar o arranque como critério também é interessante, pois ele indica o quão rápido as juntas irão variar suas forças. Segundo Ratiu (2017), reduzí-lo, além de garantir continuidade para as trajetórias, garante que o rastreio delas tenha mais acurácia, proporcionando redução nos limites de vibração do robô que, por conseguinte, reduz o desgaste dos motores e aumenta o ciclo de vida desses dispositivos.

Para obter trajetórias com melhores performances, alguns autores vêm realizando otimizações com multicritérios. Por exemplo, Milkjovic e Petrovic (2017) consideraram o tempo de ciclo e o custo de produção como critérios. Feifei & Fei (2016), consideram o tempo mínimo e arranque contínuo. Petrovic (2018) cita o algoritmo "*Covariant Hamiltonian Optimization for Motion Planning*" (CHOMP), que se baseia em duas funções: uma que atua na suavização da curva e outra que captura os requisitos para o desvio de obstáculos. A função de suavização é definida em termos de métrica no espaço das trajetórias. A função do obstáculo é desenvolvida como uma integral de linha de um campo com custo escalar, um campo de distância pré-computado, definido de maneira a ser invariante no tempo. As duas funções possuem papeis complementares: a função do obstáculo governa o formato do caminho e a função de suavização é responsável pelas restrições temporais ao longo do caminho.

Petrovic (2018) apresenta também as diversas melhorias no algoritmo CHOMP que foram propostas nos últimos anos: Multigrid CHOMP com suavidade local, que promove o tempo de execução do CHOMP sob restrições, sem reduzir significativamente a otimização, T-CHOMP, um algoritmo de gradiente funcional que incorpora restrições e funções dependentes do tempo, *Incremental Trajectory Optimization Algorithm* (ITOMP), que realiza o planejamento em tempo real, para ambientes dinâmicos e *Stochastic Trajectory Optimization for Motion Planning* (STOMP), uma variação do CHOMP que utiliza de amostras de uma série de trajetórias ruidosas para explorar o espaço.

2.3. Otimização Energética

De acordo com Chiddarwar & Babu (2012), quando se usa a energia como critério no planejamento de trajetórias, de maneira análoga ao critério de arranque, resultam em trajetórias com comportamento suave, reduzindo o estresse nos atuadores e na estrutura robótica. Ademais, minimizar o consumo de energia pode ser desejável em diversas aplicações em que a fonte energética possui uma capacidade limitada, por exemplo, em robôs para explorações espaciais, submarinas, entre outros.

Para robôs manipuladores, segundo Carabin *et al*. (2017), são utilizadas duas abordagens na otimização energética: inversa e direta. Ambas empregam modelos dinâmicos do sistema mecatrônico. Na abordagem inversa, o sistema é formulado através de modelos simplificados. São consideradas as propriedades: comprimento, massa e inércia dos elos, relações de transmissão, coeficientes de atrito e parâmetros elétricos do sistema (torque do motor, resistência de enrolamento do motor etc.). A trajetória é planejada levando em conta o conteúdo energético e resulta como solução de um problema de otimização. Na abordagem direta, o modelo do sistema mecatrônico é mais completo, leva em consideração perdas secundárias (perdas do motor, auxiliares etc.) e não-linearidades em geral. O modelo é explorado para avaliar o impacto dos diferentes parâmetros no consumo de energia através de uma análise de sensibilidade. A abordagem direta é mais empregada em casos onde são utilizados robôs com poucos graus de liberdade por envolver um grande esforço computacional.

Para computar as equações dinâmicas são utilizados majoritariamente os métodos de Newton-Euler e de Lagrange. Uma vez que o modelo dinâmico é desenvolvido, o segundo passo é formular o tipo de trajetória e os parâmetros a serem otimizados, ou seja, as variáveis de projeto. É comum o uso de funções contínuas na formulação das trajetórias, uma vez que elas

são eficientes e práticas: permitem a geração de trajetórias suaves através da manipulação de um número reduzido de variáveis. São úteis também na filtragem de dados obtidos através de outros métodos de planejamento de trajetórias (aqueles baseados em amostragem, por exemplo) (Petrovic, 2018).

Para robôs seriais, as trajetórias são comumente formuladas através de polinômios de ordem elevada ou curvas *B-spline* (Carabin *et al.*, 2017). A ordem das curvas deve ser determinada de maneira a satisfazer as condições iniciais e finais, que são geralmente definidas em termos de posições, velocidades, acelerações e arranques, variáveis essas que asseguram movimentos suaves nas bordas da trajetória. Quando se utiliza polinômios em problemas de otimização, por exemplo, a ordem do polinômio deve ser maior do que o número de restrições, o sistema deve apresentar grau de liberdade maior ou igual a 1. Entretanto, como mostrado por Fung (2014), aumentar demasiadamente o número de graus de liberdade do sistema não apresenta mudanças significativas no que diz respeito ao consumo de energia.

Outro fator a ser considerado, é a função custo para minimização energética. Segundo Carabin et al. (2017) existem várias formulações de propostas na literatura, as mais utilizadas são:

- Mínimo Esforço, cujo modelo de esforço está relacionado com o torque, conforme apresentado na Eq. (1).

$$f(\tau_i) = \sum_{i=1}^{n} \int_0^{t_f} \tau_i^2 dt \tag{1}$$

- Mínima taxa de torque, que utiliza a taxa de variação infinitesimal dos torques das juntas, conforme a Eq (2).

$$f(\tau_i) = \sum_{i=1}^{n} \int_0^{t_f} \frac{d\tau_i}{dt}^2 dt \tag{2}$$

- Mínima energia elétrica, a qual relaciona a tensão, $e(t)$, e corrente elétrica, $i(t)$, conforme apresentado na Eq (3).

$$f(e,i) = \int_o^{t_f} e(t)\, i(t) dt \tag{3}$$

- Mínima energia mecânica, que utiliza a integral do produto das velocidades de cada junta i, $\dot{q}_i$, com o torque demandado por elas, dado pela Eq. (4).

$$f(q_i, \tau_i) = \sum_{i=1}^{n} \int_{o}^{t_f} \dot{q}_i(t)\, \tau_i(t) dt \tag{4}$$

Quando se trata de algoritmo de otimização, de acordo com Carabin *et al*. (2017), os principais são: algoritmos de otimização baseados em gradiente (mais especificamente programação quadrática sequencial, SQP), Algoritmos Genéticos (AG), Algoritmos Meta-Heurísticos, Programação Dinâmica Discreta (DDP) ou Interativa (IDP). Porém, os mais utilizados entre esses são os métodos SQP e AG (Carabin *et al*., 2017).

Os métodos que empregam uma busca direta, AG por exemplo, baseiam as buscas nas funções objetivo e nas restrições do processo. Apresentam como desvantagens o tempo de resposta e o grande número de iterações para convergir. Os métodos baseados em gradiente utilizam derivadas de primeira e segunda ordem das funções objetivos e das restrições como guias como busca da solução ótima. Eles apresentam a vantagem ter um menor tempo de convergência e soluções próximas ao ótimo global (Pervier *et al*., 2011). Porém, como qualquer outro método convencional de otimização não-linear, são atraídos por mínimos locais (Saravan *et al*., 2007).

Chiddarwar & Babu (2012), realizaram uma comparação entre os métodos SQP e AG na otimização do robô KUKA-Kr. Os dois métodos conseguiram encontrar resultados satisfatórios, os quais foram analisados e comparados. Os autores concluíram que o AG forneceu melhores resultados do que o SQP para o problema: um menor tempo de convergência e uma quantidade menor de energia consumida durante a performance da trajetória. Eles acreditam que a razão disso é a dependência do SQP da solução inicial, diferentemente do AG, e também da necessidade de computar o Gradiente e a Hessiana da função objetivo e restrições, o que implica na necessidade de que as derivadas de segunda ordem sejam continuas.

Devido à complexidade das funções dinâmicas que modelam os robôs manipuladores, os Algoritmos Genéticos são muito utilizados na otimização da trajetória desses robôs. Alguns exemplos são os trabalhos de Ata (2007), Chiddarwar & Babu (2012), Mulik (2015), Saravan *et al*. (2007) e Zhang (2018).

Em geral, encontrar a trajetória ótima é um problema difícil devido à complexidade da dinâmica dos robôs, o que impossibilita o uso dos métodos analíticos. Os métodos de otimização não-lineares resolvem o planejamento de movimento com altas dimensões de maneira rápida, porém, ficam presos a soluções localmente ótimas e podem apresentar problemas em espaços de busca discretos. Os métodos de busca direta ficam sobrecarregados devido ao tamanho do espaço e também apresentam soluções ótimas locais (Abu-Dakka, 2011). Os desafios futuros incluem encontrar a função custo adequada para otimização e resolver o problema de mínimos locais, seja empregando outro tipo de método de otimização ou encontrando melhores inicializações de trajetória (Petrovic, 2018).

CAPÍTULO 3: MODELAGEM E CONTROLE DE ROBÔS MANIPULADORES

Uma das grandes ambições do homem é dar vida aos seus artefatos (Siciliano *et al.*, 2008). Mecanismos com figura humana são datados desde os antigos tempos na Grécia e o conceito de androide é visto em trabalhos de ficção desde o começo do século XIX (Bellis, 2019). Certamente, isso inspirou diversas pessoas em novos desenvolvimentos na robótica a partir da década de 50. Por exemplo, o George Devol, que inventou o primeiro robô manipulador programável operado digitalmente, o qual era capaz de operar de maneira autônoma para executar diferentes tarefas (Malone, 2011).

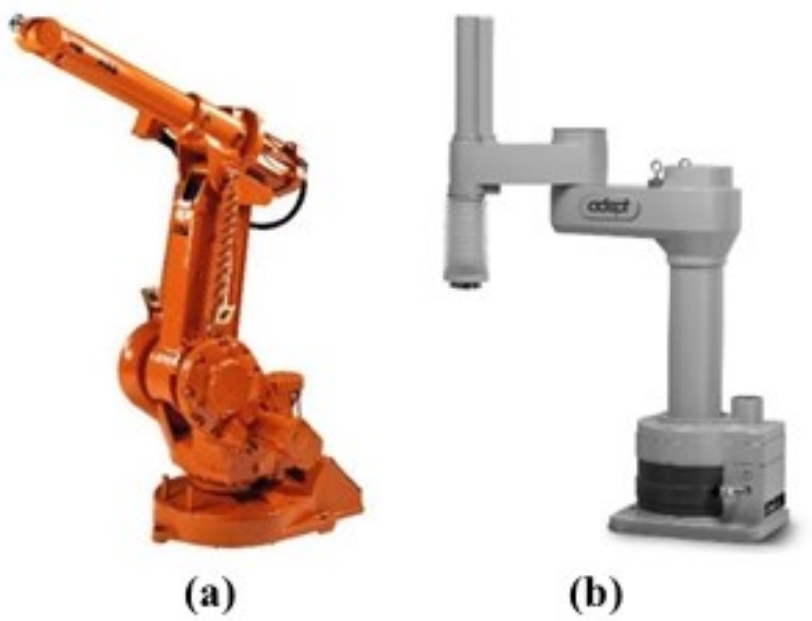
Existem diversas definições para robôs. *Robotics Institute of America apud* Tsai (1999) define um robô como um "manipulador multifuncional e reprogramável designado a mover materiais, partes, equipamentos ou dispositivos especializados que, através de movimentos programados, conseguem realizar tarefas". A definição de Corke (2017) é mais generalizada, para ele um robô é uma máquina orientada a resolver um problema, sendo essa capaz de sentir, planejar e agir.

Entre os tipos de robôs estão inclusos os manipuladores. As estruturas desses robôs são comumente representadas de maneira esquemática como uma cadeia cinemática de corpos rígidos, elos, que são conectados em série através de juntas (Siciliano *et al.*, 2008). A escolha do manipulador adequado depende da tarefa a ser realizada, tornando conveniente classificar os robôs conforme aspectos construtivos e operacionais:

- **Tecnologia / tipos de atuadores**: hidráulica, pneumática ou elétrica.
- **Tipos de juntas**: prismáticas, de revolução ou uma combinação dessas.
- **Geometria**: Articulada (RRR), esférica (RRP), SCARA (RRP), cilíndrica (RPP) ou cartesiana (PPP). Observação: a geometria esféria e SCARA possuem mesma sequência de juntas, mas com diferente arranjo entre os elos.
- **Cadeia cinemática**: paralela ou serial.
- **Trajetória**: robôs que movimentam ponto-a-ponto ou com caminho contínuo.

Em âmbito industrial, a arquitetura de manipulador mais utilizada é a do tipo serial (exemplo na Fig. 5) e nesse tipo de robôs, um dos corpos rígidos está conectado à base, enquanto o efetuador final está na outra extremidade.

Figura 5: Robôs seriais, (a) ABB IRB1400 e (b) AdeptOne XL.



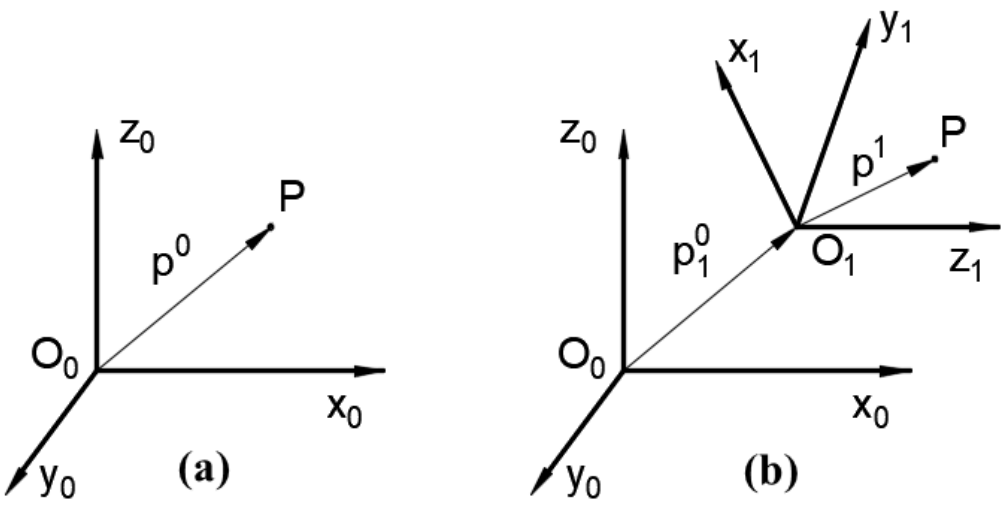
Fonte: (a) Spong *et al*. (2004) e (b) Siciliano *et al*. (2008).

O movimento resultante da estrutura é obtido pela composição de movimentos elementares de cada elo em relação ao elo anterior (Siciliano *et al*., 2008). Por isso, as próximas seções irão abordar as Modelagens Cinemática, Seção 3.1., e Dinâmica, Seção 3.2., de robôs manipuladores. A Seção 3.3. visa apresentar a Teoria de controle.

3.1. Modelagem Cinemática

Para localizar um ponto P no espaço é necessário considerar a existência de um sistema de coordenadas. Um exemplo é o sistema de coordenadas cartesiano, no qual um conjunto de eixos ortogonais se interceptam em um ponto conhecido como origem. Para o caso de três dimensões, tomemos a origem como $O_0x_0y_0z_0$. O deslocamento do ponto em relação a esse referencial expressa a sua posição, a qual pode ser descrita por um vetor de coordenadas, Fig. 6 (a), p^0. Neste trabalho, convencionou-se que os vetores são escritos com letras sobrescritas indicando o sistema de coordenadas aos quais se referem. Por exemplo, p^0 tem um valor de distância $[p_x \quad p_y \quad p_z]^T$ em relação ao referencial $O_0x_0y_0z_0$.

Figura 6: Descrevendo pontos através de vetores de coordenadas.



Fonte: autor.

É comum adotar um referencial $O_1x_1y_1z_1$ para corpos rígidos, pois, dessa forma, cada ponto que compõe o corpo terá uma posição constante em relação a esse referencial. Assim sendo, torna-se desnecessária a indicação das coordenadas dos pontos do objeto de uma maneira individual, uma vez que os corpos rígidos podem ser descritos no espaço de uma maneira mais prática: utilizando posições e orientações relativas entre o seu referencial, $O_1x_1y_1z_1$, e o referencial global, $O_0x_0y_0z_0$. Logo, as coordenadas de um ponto P, contido no corpo rígido, Fig. 6 (b), em relação à origem, $O_0x_0y_0z_0$, pode ser calculada conhecendo a distância em relação ao referencial do corpo, p^1, a distância entre os dois referenciais, p_1^0, e a orientação do corpo em relação à origem, a qual pode ser descrita através de uma matriz de rotação, R_1^0, conforme a Eq. (5).

$$p^0 = p_1^0 + R_1^0 p^1 \tag{5}$$

Ademais, as informações acerca de um ponto do corpo rígido em relação à origem podem ser escritas de uma maneira compacta, por meio de matrizes de transformação, $A_1^0\ (4x4)$, conforme a Eq. (6). Nessa situação, os pontos devem ser escritos de forma aumentada, $\hat{p}^T = [p^T \quad 1]$, e um vetor de zeros, $0^T\ (1x3)$, deve ser empregado.

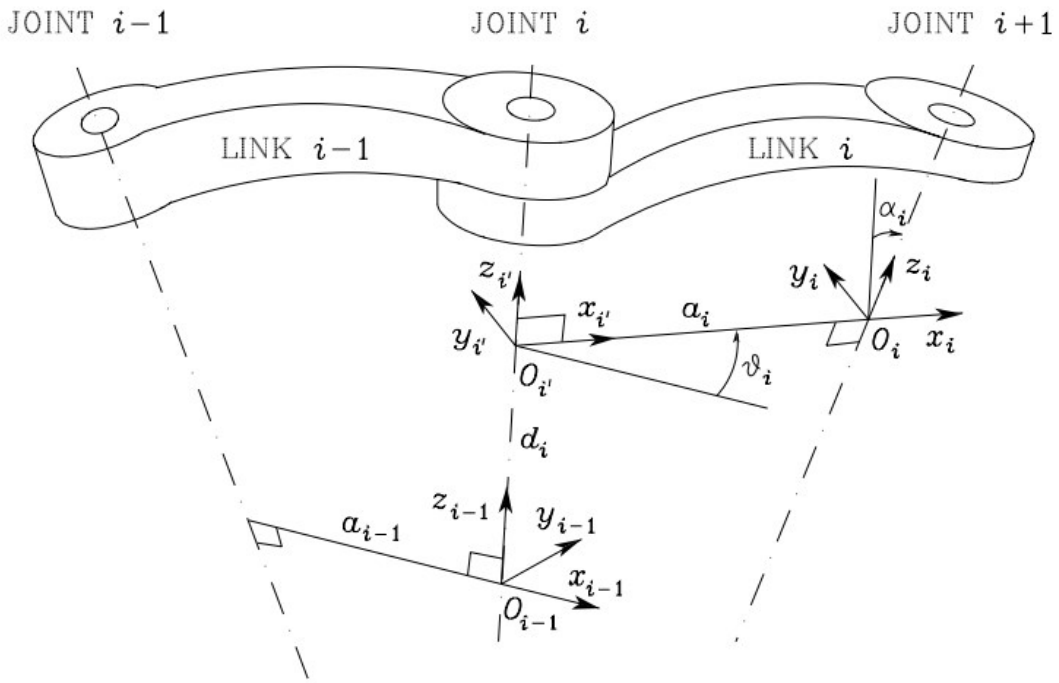
$$\hat{p}^0 = A_1^0 \hat{p}^1 = \begin{bmatrix} R_1^0 & p_1^0 \\ 0^T & 1 \end{bmatrix} \hat{p}^1 \tag{6}$$

No projeto de robôs manipuladores é necessário que cada junta possua seu próprio referencial e que seja fácil relacioná-los. Para isso, existem métodos que visam sistematizar a obtenção das matrizes de transformação. Portanto, a Seção 3.1.1. apresenta o método de Denavit-Hartenberg, a Seção 3.1.2. exibe o método dos Helicoides Sucessivos e a Seção 3.1.3. é dedicada a obtenção das velocidades dos elos e juntas de robôs.

3.1.1. Método de Denavit-Hartenberg

O método de Denavit-Hartenberg objetiva sistematizar a obtenção das matrizes de transformação espacial entre dois elos consecutivos. Para isso, primeiramente, determina-se os referenciais anexados aos elos. A Fig. 7 representa um par de elos adjacentes de um robô manipulador (i e $i-1$) e as respectivas juntas dele ($i-1$, i e $i+1$).

Figura 7: Parâmetros Cinemáticos de Denavit-Hartenberg.



Fonte: Siciliano *et al.* (2008).

A convenção de Denavit-Hartenberg para o sistema de coordenadas do elo i, conforme Siciliano *et al.* (2008), é:

- Escolher um eixo z_i ao longo do eixo da junta $i+1$
- Localizar a origem O_i na intersecção do eixo z_i com a normal comum dos eixo z_{i-1} e z_i, bem como localizar $O_{i'}$ na interesecção da normal comum com o eixo z_{i-1}.
- Escolher o eixo x_i ao longo da normal comum dos eixos z_{i-1} e z_i com direção da junta i até a junta $i+1$.
- Escolher o eixo y_i de maneira a completar a regra da mão direita.

Para que a posição relativa entre os sistemas de coordenadas seja devidamente computada, a convenção de Denavit-Hartenberg considera os seguintes parâmetros:

- a_i, a distância entre os eixos z_{i-1} e z_i, medida ao longo do eixo x_i (distância entre O_i e $O_{i'}$).
- α_i, o ângulo entre os eixos z_{i-1} e z_i, medida ao longo do eixo x_i e seguindo a regra da mão direita: ângulo de rotação de x_i para que z_{i-1} fique paralelo a z_i.
- d_i, a distância entre os eixos x_{i-1} e x_i, medida ao longo do eixo z_{i-1} (coordenada de $O_{i'}$ ao longo de z_{i-1}).
- θ_i, o ângulo entre os eixos x_{i-1} e x_i, medida ao longo do eixo z_i e seguindo a regra da mão direita: ângulo de rotação de z_i para que x_{i-1} fique paralelo a x_i.

Dois desses parâmetros são sempre constantes, a_i e α_i . Os restantes vão depender do tipo de junta que sustenta o elo i. Se for uma junta de revolução, θ_i é uma variável. Para juntas prismáticas, a variável é d_i. Utilizando esses parâmetros e adotando a convenção apresentada em Siciliano *et al.* (2008), determina-se a posição e orientação do referencial i em relação a $i-1$ através da seguinte matriz:

$$A_i^{i-1} = \begin{bmatrix} \cos\theta_i & -\sin\theta_i \cos\alpha_i & \sin\theta_i \sin\alpha_i & a_i \cos\theta_i \\ \sin\theta_i & \cos\theta_i \cos\alpha_i & -\cos\theta_i \sin\alpha_i & a_i \sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{7}$$

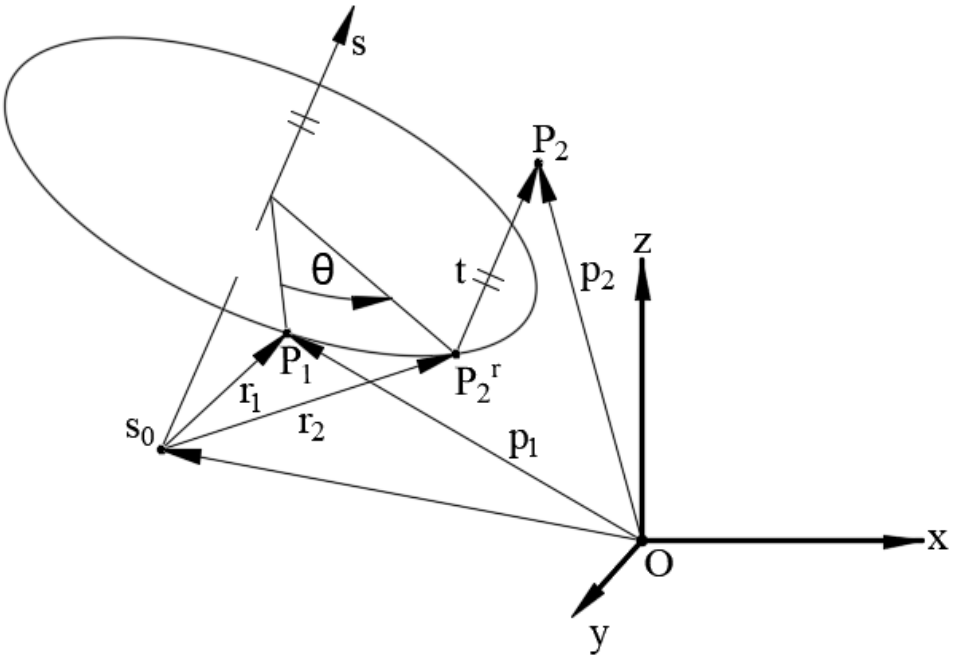
A matriz de transformação A_i^{i-1} é uma função apenas da variável de junta q_i que, como mencionado, pode ser θ_i ou d_i. Caso se deseje obter a posição do efetuador final em relação à base, deve-se obter as matrizes de transformação de cada junta e multiplicá-las, conforme a Eq. (8).

$$A_n^0(q) = A_1^0(q_1)A_2^1(q_2) \dots A_n^{n-1}(q_n) \tag{8}$$

3.1.2. Método dos Helicoides Sucessivos

A Teoria dos Helicóides é baseada no teorema de Mozzi-Chasles. Ela parte do princípio de que o movimento de um corpo rígido pode ser representado por uma rotação seguida de uma translação ao longo do mesmo eixo, conforme a Fig. 8. O eixo é chamado de eixo helicoidal e é representado pelo vetor unitário $s = [s_x \quad s_y \quad s_z]^T$. A posição do eixo helicóidal é denotada por $s_0 = [s_{0x} \quad s_{0y} \quad s_{0z}]^T$ (Tsai, 1999). Como exposto na Fig. 8, inicialmente um ponto $P \in \mathbb{R}^3$ realiza um movimento circular de ângulo θ, move-se da posição $P_1 \in \mathbb{R}^3$ até a posição $P_2^r \in \mathbb{R}^3$, seguido de uma translação da posição P_2^r até a coordenada do ponto $P_2 \in \mathbb{R}^3$.

Figura 8: Diagrama vetorial do teorema de Mozzi-Chasles.



Fonte: autor, adaptado de Tsai (1999).

O ângulo θ, devido à rotação, e à distância t, devido a translação, são chamados de parâmetros helicoidais. Para obter uma representação compacta entre as posições dos pontos, p_2 e p_1, de maneira análoga conforme mostrado na representação da Orientação e Posição, eles podem ser escritos em uma forma aumentada, $\hat{p}_1^T = [p_1^T \quad 1]$ e $\hat{p}_2^T = [p_2^T \quad 1]$, estabelecendo a relação entre ambos a partir da fórmula de Rodrigues, Eq. (9), a qual é escrita através de matrizes de transformação homogêneas, $A(\theta, t)$ $(4x4)$.

$$\hat{p}_2 = A(\theta, t)\hat{p}_1$$
$$A(\theta, t) = \begin{bmatrix} R(\theta) & d(t) \\ 0 & 1 \end{bmatrix} \tag{9}$$

A matriz de rotação, $R(\theta)$ e o vetor de translação, $d(t)$, são calculados através da Eq. (10).

$$R(\theta) = \begin{bmatrix} (s_x^2 - 1)(1 - \cos\theta) + 1 & s_x s_y(1 - \cos\theta) - s_z \sin\theta & s_x s_z(1 - \cos\theta) - s_y \sin\theta \\ s_y s_x(1 - \cos\theta) - s_z \sin\theta & (s_y^2 - 1)(1 - \cos\theta) + 1 & s_y s_z(1 - \cos\theta) - s_x \sin\theta \\ s_z s_x(1 - \cos\theta) - s_y \sin\theta & s_z s_y(1 - \cos\theta) - s_x \sin\theta & (s_z^2 - 1)(1 - \cos\theta) + 1 \end{bmatrix} \tag{10}$$

$$d(t) = ts + [I - R(\theta)]s_0$$

No âmbito da robótica, a fórmula de Rodrigues, em sua forma matricial, pode ser utilizada de maneira sucessiva para o cálculo do deslocamento absoluto entre um elo i, por exemplo, até outro, n. A matriz de transformação resultante, A_r, é obtida através da Eq. (11), em que A_i, A_{i+1}, A_{n-1} e A_n representam as matrizes de transformação entre os referenciais dos elos i, $i + 1$, $n - 1$ e n em relação a um sistema de coordenadas de referência.

$$A_r = A_i A_{i+1} \dots A_{n-1} A_n \tag{11}$$

Percebe-se que os métodos de Denavit-Hartenberg e dos Helicoides Sucessivos possuem similaridades. Por exemplo, em ambos o deslocamento resultante é obtido através da multiplicação das matrizes de transformação de cada junta. Porém, as difereças decorrem da forma de como essas matrizes de transformação são obtidas. Por exemplo, no método dos Helicoides Sucessivos não é possível a obtenção de deslocamentos relativos representados pelas

variáveis de juntas conforme o método de Denavit-Hartenberg. Entretanto, o método dos Helicoides Sucessivos possui uma maior flexibilidade, uma vez que o referencial do sistema pode ser escolhido de maneira arbitrária. Mais informações em Tsai (1999) e Rocha *et al.* (2011).

3.1.3. Cinemática Diferencial

As velocidades linear e angular de uma junta depende do tipo de junta que é empregada. Para as do tipo Prismática, deve-se considerar o deslocamento linear dessa junta, $\dot{d}_i$, conforme a Eq. (12). A velocidade angular, ω_i, é dada pela Eq. (13).

$$\dot{p}_i = \dot{p}_{i-1} + \dot{d}_i z_{i-1} + \omega_i \times r_{i-1,i} \tag{12}$$

$$\omega_i = \omega_{i-1} \tag{13}$$

Na Eq. (12), $r_{i-1,i}$ representa o vetor que liga o elo $i-1$ ao elo i com referencial na base, $\dot{p}_{i-1}$ a velocidade linear da junta $i-1$ em relação à base, z_{i-1} o eixo z da junta $i-1$.

Para uma junta de Revolução as velocidades linear e angular são apresentadas pelas Eq. (14) e (15), respectivamente.

$$\dot{p}_i = \dot{p}_{i-1} + \omega_i \times r_{i-1,i} \tag{14}$$

$$\omega_i = \omega_{i-1} + \dot{\theta}_i z_{i-1} \tag{15}$$

A Velocidade do Centro de Massa de um Elo i que é conectado a um elo $i-1$, $\dot{p}_{C_i}$, é dada pela Eq. (16). O cálculo da Velocidade Angular do Rotor i, ω_{m_i}, inclui efeitos de relações de transmissões, conforme a Eq. (17).

$$\dot{p}_{C_i} = \dot{p}_i + \omega_i \times r_{i,C_i} \tag{16}$$

$$\omega_{m_i} = \omega_{i-1} + k_{ri}\dot{q}_i z_{m_i} \tag{17}$$

3.2. Modelagem Dinâmica

A modelagem dinâmica é desenvolvida no espaço de configuração e busca relacionar as variáveis de juntas com as respectivas causas do movimento. A equação dinâmica geral é dada, em uma forma matricial, pela Eq. (18).

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + F(\dot{q}) + G(q) = \tau + J_C(q)^T F_C \tag{18}$$

Nessa equação, q, $\dot{q}$ e $\ddot{q}$ são, respectivamente, o vetor de coordenadas generalizadas (θ para juntas rotacionais e d para juntas prismáticas), o vetor contendo as velocidades das juntas e o vetor contendo as acelerações das juntas. $M(q)$ é a matriz de inércia do manipulador, $C(q,\dot{q})$ é a matriz de forças centrifuga e de Coriolis, $F(\dot{q})$ é o vetor de forças de atrito, $G(q)$ é o vetor da forças gravitacionais, $\tau(t)$ é o vetor de torques das juntas, F_C são forças externas cartesianas (de contato, por exemplo) e $J_C(q)$ o Jacobiano Geométrico relativo às forças externas.

A dinâmica de manipuladores possui basicamente dois problemas. O primeiro é quando a trajetória do manipulador, q, $\dot{q}$ e $\ddot{q}$, é conhecida e deseja-se saber o vetor de torque nas juntas, τ, caracterizando um problema de Dinâmica Inversa. O segundo problema é descobrir como o mecanismo se move através da aplicação de torque nas juntas: o vetor τ é conhecido e deseja-se saber o movimento resultante, q, $\dot{q}$ e $\ddot{q}$, abordagem conhecida como Dinâmica Direta. As próximas seções dedicam-se a apresentar soluções para esses problemas. A Seção 3.2.1. emprega a formulação de Newton-Euler aplicada a solução da Dinâmica Inversa e a Seção 3.2.2. apresenta a solução da Dinâmica Direta através do Simscape™.

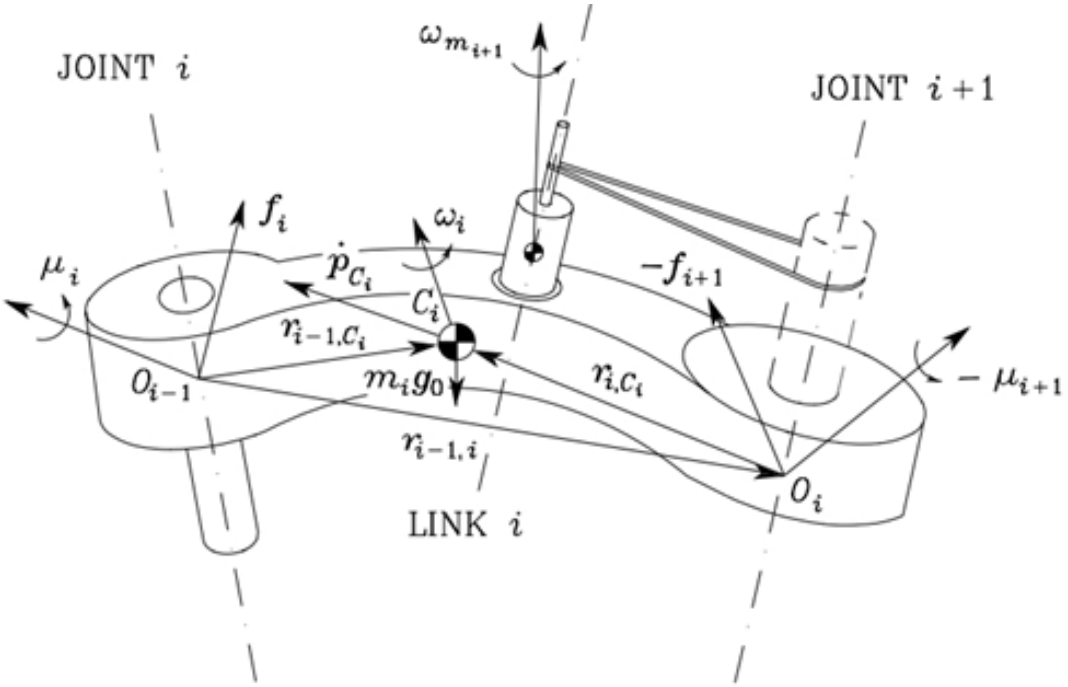
3.2.1. Dinâmica Inversa através das Equações de Newton-Euler

Para o Planejamento de Trajetórias Energeticamente Ótimas é necessário resolver o problema de Dinâmica Inversa. Pois, como visto na Seção 2.3., as funções de energia estão subordinadas aos torques em cada junta do robô manipulador. Então, faz-se necessário obter os valores desses torques conhecendo as variáveis da trajetória: posições, velocidades e acelerações, no que se traduz em um problema de Dinâmica Inversa.

Para solução desse problema, Siciliano *et al.* (2008) apresenta um algoritmo que se baseia nas equações de Newton-Euler. A estratégia consiste na "propagação" das velocidades e acelerações dos elos inferiores aos superiores, em adição da "propagação inversa" das forças e

torques que agem sobre os elos superiores aos inferiores. Para que um elo i permaneça em uma dada posição, o elo $i-1$ deve sustentá-lo com uma força normal e torque. Cada elo está sujeito ao seu próprio peso, bem como a forças de reações e torques de elos que estão suportando (Corke, 2017). A Fig. 9 exibe as variáveis que estão associadas no cálculo das equações dinâmica associadas a um elo i.

Figura 9: Variáveis associadas ao cálculo da Dinâmica de uma junta e elo i.



Fonte: Siciliano *et al*. (2009).

As acelerações linear e angular de uma Junta Prismática i são obtidas conforme as Eq. (19) e (20), que são derivadas das Eq. de velocidades da Seção 3.1.3.

$$\ddot{p}_i^i = R_{i-1}^i(\ddot{p}_{i-1}^{i-1} + \ddot{d}_i z_0) + 2\dot{d}_i^i \omega_i^i \times R_{i-1}^i z_0 + \dot{\omega}_i^i \times r_{i-1,i}^i + \omega_i^i \times \left(\omega_i^i \times r_{i-1,i}^i\right) \qquad (19)$$

$$\dot{\omega}_i^i = R_{i-1}^i \dot{\omega}_{i-1}^{i-1} \qquad (20)$$

Onde, $\ddot{p}_{i-1}^{i-1}$ é a aceleração linear da junta $i-1$ vista sob o referencial $i-1$, $\ddot{d}_i$ corresponde a aceleração do deslocamento linear da junta e $r_{i-1,i}^i$ é o vetor que liga o referencial

$i-1$ ao i visto sob a perspectiva de i. $\dot{\omega}_{i-1}^{i-1}$ é a velocidade angular da junta $i-1$ vista sob a perspectiva de $i-1$.

Para Juntas Rotativas i, as acelerações são dadas pelas Eq. (21) e (22). Onde $\ddot{\theta}_i$ é a aceleração do deslocamento angular da junta.

$$\ddot{p}_i^i = R_{i-1}^i \ddot{p}_{i-1}^{i-1} + \dot{\omega}_i^i \times r_{i-1,i}^i + \omega_i^i \times \left(\omega_i^i \times r_{i-1,i}^i\right) \tag{21}$$

$$\dot{\omega}_i^i = R_{i-1}^i \left(\dot{\omega}_{i-1}^{i-1} + \ddot{\theta}_i z_0 + \dot{\theta}_i \omega_{i-1}^{i-1} \times z_0\right) \tag{22}$$

A obtenção da aceleração linear do centro de massa do elo acontece conforme a Eq. (23), na qual r_{i,C_i}^i é o vetor que conecta o referencial i à posição do centro de massa do elo i visto sob o referencial i.

$$\ddot{p}_{C_i}^i = \ddot{p}_i^i + \dot{\omega}_i^i \times r_{i,C_i}^i + \omega_i^i \times (\omega_i^i \times r_{i,C_i}^i) \tag{23}$$

A aceleração angular do rotor é calculada através da Eq. (24), na qual $k_{r,i}$ é uma constante de relação de transmissão e $z_{m_i}^{i-1}$ é o eixo z do rotor visto sob o referencial de $i-1$.

$$\dot{\omega}_{m_i}^{i-1} = \dot{\omega}_{i-1}^{i-1} + k_{r,i} \ddot{q}_i z_{m_i}^{i-1} + k_{r,i} \dot{q}_i \omega_{i-1}^{i-1} \times z_{m_i}^{i-1} \tag{24}$$

Aplicando a segunda Lei de Newton para o movimento translacional do centro de massa de elo aumentado, que considera o elo i junto ao motor da junta $i+1$, obtém-se a Eq. (25). Em tal equação, m_i é a massa do elo aumentado i, o termo $R_{i+1}^i f_{i+1}^{i+1}$ corresponde a força exercida pelo elo $i+1$ ao elo i no referencial de i e f_i^i é a força exercida pelo elo $i-1$ ao elo i.

$$f_i^i = R_{i+1}^i f_{i+1}^{i+1} + m_i \ddot{p}_{C_i}^i \tag{25}$$

A equação de Euler para movimentos rotacionais aplicada a um elo aumentado i, referindo ao centro de massa dele, pode ser escrita na forma da Eq. (26).

$$\mu_i^i = -f_i^i \times \left(r_{i-1,C_i}^i\right) + R_{i+1}^i \mu_{i+1}^{i+1} + R_{i+1}^i f_{i+1}^{i+1} \times r_{i,C_i}^i + \bar{I}_i^i \dot{\omega}_i^i + \omega_i^i \times \left(\bar{I}_i^i \omega_i^i\right) + \omega_i^i \times \left(\bar{I}_i^i \omega_i^i\right) + k_{r,i+1} \ddot{q}_{i+1} J_{m_{i+1}} z_{m_{i+1}}^i + k_{r,i+1} \dot{q}_{i+1} J_{m_{i+1}} \omega_i^i \times z_{m_{i+1}}^i \quad (26)$$

Nessa equação, μ_i^i é o momento exercido do elo $i-1$ ao elo i em relação à origem do referencial i, o termo $R_{i+1}^i \mu_{i+1}^{i+1}$ é o momento exercido do elo $i+1$ ao elo i em relação à origem do referencial i, $\bar{I}_i^i$ é o tensor de inércia do elo aumentado i e $J_{m_{i+1}}$ é o momento de inércia do rotor $i+1$.

O cálculo do Torque de uma Junta i depende de como a massa dos elos que ela sustenta está distribuída (Corke, 2017) e também do tipo de junta. Para juntas prismáticas, o cálculo do torque é dado pela Eq. (27). Para juntas de revolução, a Eq. (28) é utilizada.

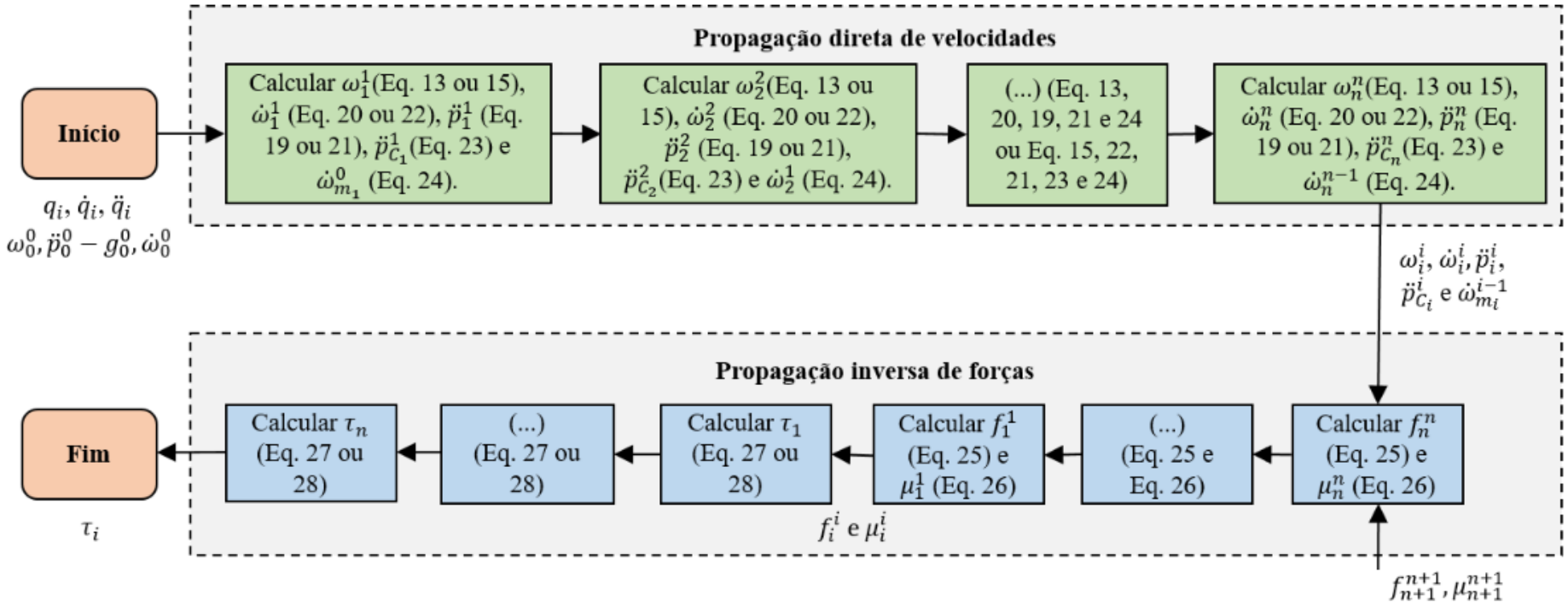
$$\tau_i = \left(f_i^i\right)^T \left(R_i^{i-1}\right)^T z_0 + k_{ri} J_{m_i} \left(\dot{\omega}_{m_i}^{i-1}\right)^T z_{m_i}^{i-1} + B\dot{d}_i + \mu\, sgn(\dot{d}_i) \quad (27)$$

$$\tau_i = \left(\mu_i^i\right)^T \left(R_i^{i-1}\right)^T z_0 + k_{ri} J_{m_i} \left(\dot{\omega}_{m_i}^{i-1}\right)^T z_{m_i}^{i-1} + B\dot{\theta}_i + \mu\, sgn(\dot{\theta}_i) \quad (28)$$

Nas Eq. (27) e (28), B e μ são constantes de forças dissipativas, atrito viscoso e de Coulomb respectivamente e $z_0 = [0 \quad 0 \quad 1]^T$.

O Algoritmo, apresentado por Siciliano *et al.* (2008), considera que as velocidades e acelerações da base em que o robô está inserido, $\dot{p}_0$, ω_0, $\ddot{p}_0$, $\dot{\omega}_0$ são conhecidas, além das variáveis das juntas q, $\dot{q}$ e $\ddot{q}$. Portanto, as variáveis ω_i, $\dot{\omega}_i$, $\ddot{p}_i$, $\ddot{p}_{C_i}$ e $\dot{\omega}_{mi}$, das i juntas e elos, $i \in \{1,2,\ldots,n\}$, podem ser calculadas devido ao efeito de propagação de velocidades e acelerações. Conhecendo todas as velocidades e acelerações, calcula-se, posteriormente, as forças e os momentos, iniciando com a força e o momento que age no efetuador final. A Fig. 10 resume a sequência de cálculos.

Figura 10: Algoritmo para cálculo da Dinâmica Inversa em robô com n juntas.

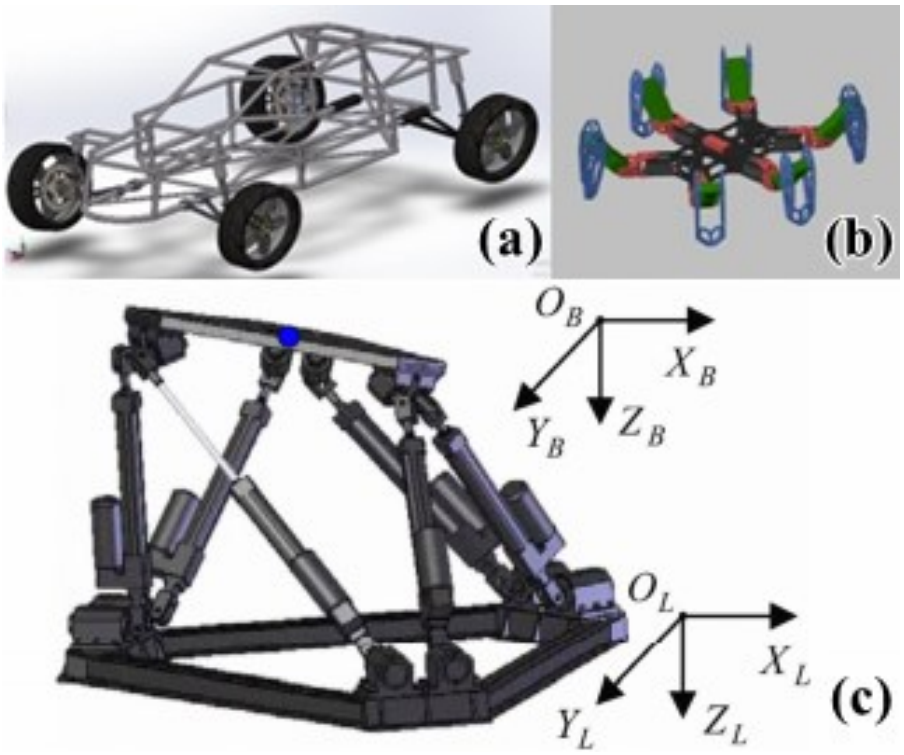


Fonte: autor.

3.2.2. Dinâmica Direta através do Simscape™

O Simscape™, que está inserido no *software* Simulink®, permite a construção de componentes utilizando conexões físicas de suas partes que se integram diretamente através de diagrama de blocos. Sistemas mecânicos como motores, robôs, suspensão de veículos e atuadores hidráulicos são representados montando seus componentes elementares em um esquema. Os componentes podem ser cubos, esferas, cilindros etc. e são representados por um bloco de elemento sólido. Nesses blocos é possível impor valores de variáveis como a geometria e a inércia dos elementos. Ademais, existem também os blocos de transformações espaciais, que possibilitam estabelecer valores de posições e orientações de uma estrutura em relação à outra, e de juntas, os quais aceitam como entrada e/ou saída variáveis como, por exemplo, posição, velocidade, aceleração e torque. Logo, o sistema resultante pode ser modelado por uma "rede" de blocos, o que torna fácil a simulação de seu comportamento dinâmico. Na Fig. 11, pode-se ver alguns modelos geométricos desenvolvidos no Simscape™.

Figura 11: (a) veículo, (b) robô hexápode e (c) robô paralelo com 6 graus de liberdade desenvolvidos no Simscape™.



Fonte: (a) Szántó & Hajdu (2019), (b) Beaber (2018) e (c) Yang *et al*. (2010).

A resposta dinâmica dos sistemas mecânicos feitos pelo Simscape™ em um manipulador equivale à solução de sua Dinâmica Direta. Logo, diversos trabalhos empregam essa ferramenta na modelagem de sistemas. Por exemplo, Szántó & Hajdu (2019) realizou a modelagem simulação de um veículo, Fig. 11 (a). Beaber (2018), realizou a modelagem, simulação e controle de um robô Hexápode, Fig. 11 (b). Yang *et al.* (2010) realizou a modelagem e simulação de um robô paralelo com 6 DOF, Fig. 11 (c). Apesar de ser possível construir o modelo geométrico diretamente no Simscape™, nesses trabalhos o modelo foi importado de um software CAD, pois permitem mais detalhes aos modelos e são mais frequentemente utilizados. Mais detalhes e tutoriais sobre esse recurso estão disponíveis no site MathWorks.

3.3. Teoria de Controle

Em âmbito de robótica, o sistema de controle é imprescindível para garantir que o robô realize os movimentos, ele é definido por Nise (2002) como um subsistema construído com o objetivo de obter uma saída desejada em um processo (ou planta) com um desempenho desejado dada uma entrada específica. A construção dos sistemas de controle depende da dinâmica do processo a ser controlado, a qual, segundo Coelho (2016), pode ser modelada matematicamente de duas maneiras:

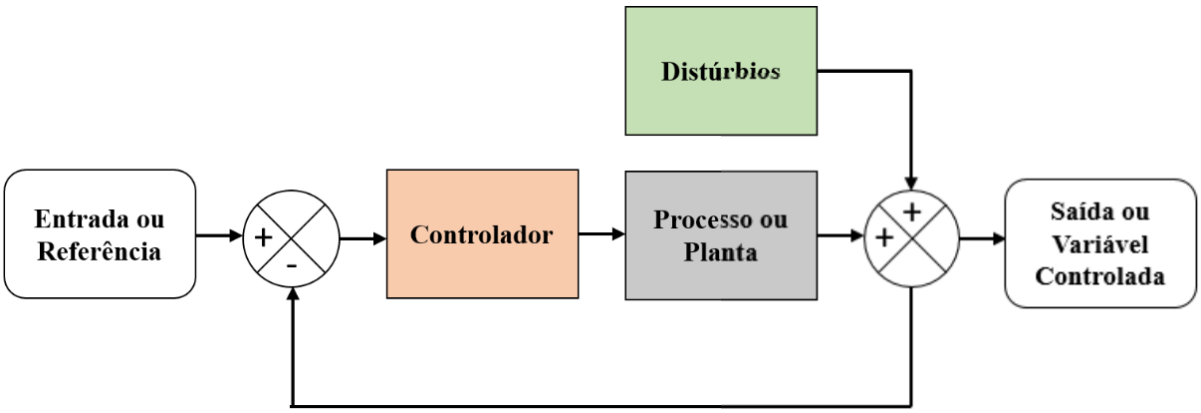
- **Análises físico-matemáticas**: leis da física que caracterizam o sistema (conservação de massa, energia e momento).
- **Análises experimentais**: medidas e observações do sistema.

Para manipuladores robóticos, as análises físico-matemáticas podem ser os modelos Dinâmico ou Cinemático. A Dinâmica Direta é um exemplo de modelo que é empregado para a simulação de manipuladores. No Apêndice E é apresentado um método para modelagem da dinâmica de sistemas via análises experimentais, o Estimador dos Mínimos Quadrados. Uma vez descrito matematicamente o modelo da planta/sistema, projeta-se o sistema de controle.

As duas configurações de controle mais conhecidas são em malha aberta e em malha fechada. Quando em malha aberta, o controlador age no processo de maneira direta, não há a possibilidade de realizar compensações caso aconteça perturbações adicionais no sistema. Quando em malha fechada, Fig. 12, são utilizados dispositivos de medição do sinal de saída.

Os sinais são comparados com as referências, gerando um sinal de erro, que é enviado ao controlador que realiza a ação de correção da saída.

Figura 12: Sistema de controle em malha fechada.



Fonte: autor.

A seguir são apresentados os Controladores do tipo PID, Seção 3.3.1., e um método para Projeto de Sistema de Controle, Seção 3.3.2..

3.3.1. Controladores do tipo PID

Os controladores PID estão no grupo dos controladores clássicos e são os mais utilizados na literatura. Eles funcionam em malha fechada e utilizam três ações: Proporcional, Integral e Derivativa.

No efeito Proporcional, a ação do controle, $u(t)$, é proporcional ao erro, $e(t)$, segundo a Eq. (29). O erro é calculado através da diferença da variável de saída do processo e a referência. Onde K_p é uma constante adimensional, usualmente chamada de ganho proporcional do controlador.

$$u(t) = K_p e(t) \tag{29}$$

Um ganho elevado proporciona ações elevadas do controlador e, de maneira antagônica, ganhos baixos promovem pequenas ações. Um controle puramente proporcional possui como

desvantagem a presença de *offset*, ele não elimina totalmente o erro estacionário após mudanças no referencial ou distúrbios no sistema (Seborg, 2004).

Na ação Integral, a saída do controle depende da integral do erro, conforme a Eq. (30). Onde K_i é uma constante chamada de ganho integral.

$$u(t) = K_i \int_0^t e(t)dt \tag{30}$$

A principal vantagem de utilização dessa ação, quando combinada com a ação proporcional (Controle PI), é a eliminação do *offset*, do erro estacionário. A desvantagem é a produção de respostas oscilatórias na variável de saída, que estão associadas ao aumento do tempo de acomodação (Seborg, 2004).

A ação Derivativa depende da derivada do erro, de acordo com a Eq. (31). Onde K_D é uma constante chamada de ganho derivativo. Essa ação tem a função de antecipar o desempenho do controlador, diminuindo o tempo de acomodação.

$$u(t) = K_d \frac{de(t)}{dt} \tag{31}$$

Na prática, um controle puramente derivativo não é implementável. Logo, essa ação é combinada ou com a proporcional, concebendo o controle PD, ou com as ações proporcional-integral, formando o controle PID. As equações que representam o controle PID básico nos domínios do tempo e de Laplace são apresentadas em (32).

$$\begin{aligned} u(t) &= K_p e(t) + K_i \int_0^t e(t)dt + K_d \frac{de(t)}{dt} \\ u(s) &= \left(K_P + \frac{K_I}{s} + K_D s\right) e(s) \end{aligned} \tag{32}$$

A parte derivativa do controle pode causar ganhos elevados quando ocorrem mudanças bruscas no sinal de referência (também chamado de *kick derivativo*). Isso pode ser evitado aplicando a ação derivativa apenas na saída do processo, $y(s)$, ou adicionando um filtro,

conforme a Eq. (33), o qual realizaria também a filtragem de ruídos provenientes dos sensores de medida da saída.

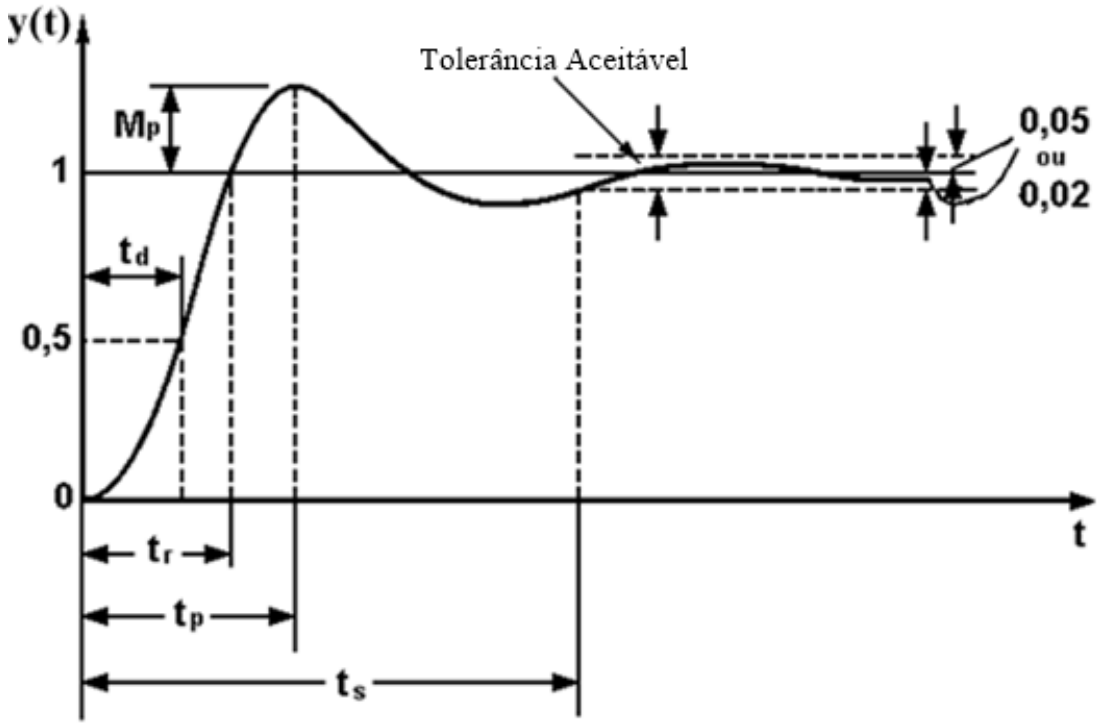
$$u(s) = \left(K_P + \frac{K_I}{s} + \frac{K_D \tau_F s}{1 + \tau_F s}\right) e(s) \tag{33}$$

Nessa equação, τ_F representa a constante de tempo do filtro.

3.3.2. Projeto de Sistemas de Controle

Projetar um sistema de controle é garantir que ele seja estável e possua um desempenho particular. O desempenho de controladores é especificado em relação às características da resposta do processo (Ogata, 2003). Para exemplificar, a Fig. 13 representa a resposta transitória de um sistema dada uma entrada do tipo degrau, onde alguns dos critérios de desempenho são exibidos.

Figura 13: Curva de resposta a uma entrada em degrau unitário.



Fonte: Ogata (2003).

Os critérios de desempenho mais utilizados são:

- O **tempo de acomodação,** t_s, que é o tempo que a resposta leva para se acomodar em uma tolerância aceitável de sinal.
- O **tempo de atraso**, que se trata do tempo necessário para a resposta alcançar metade do valor final pela primeira vez.
- O **erro em regime permanente**, que é a diferença entre o valor do sinal e o valor de referência em regime permanente.
- O **sobressinal máximo**, M_p, que caracteriza o valor máximo que o sinal atingiu em relação ao sinal de referência.
- O **tempo de subida,** t_r, que é o tempo necessário para a resposta alcançar pela primeira vez o valor de regime permanente.
- O **tempo de pico**, t_p, que é o tempo gasto para atingir o valor de sobressinal máximo.

Outra forma de avaliar o desempenho do controlador, é em termos de alguns índices propostos. Entre eles, estão o IAE, ISE e o ITAE. O IAE é definido como a integral do erro absoluto, conforme a Eq. (34).

$$J_{IAE} = \int_0^\infty |e(t)|dt \tag{34}$$

O ISE é estabelecido como a integral do quadrado do erro, de acordo com a Eq. (35).

$$J_{ISE} = \int_0^\infty e(t)^2 dt \tag{35}$$

O ITAE é definido como a integral do produto do tempo com o erro absoluto, em conformidade com a Eq. (36).

$$J_{ITAE} = \int_0^\infty t|e(t)|dt \tag{36}$$

3.3.3. Algoritmo para Projeto de Controlador

Segundo Tomei (1991), um sistema de controle do tipo PD, além de possuir simples implementação, é suficiente para estabilizar juntas tanto rígidas quanto elásticas de robôs para uma dada posição de referência. Então, para estabilizar uma junta de um manipulador com dinâmica conhecida, o controlador pode ser definido como sendo do tipo PD com filtro, com parâmetros $K_C = [K_P \quad K_D \quad \tau_F]$, os quais podem ser vistos como as variáveis de projeto em um problema de otimização. Na otimização, a função objetivo poderia ser um dos índices de desempenho propostos anteriormente ou uma combinação deles. Dessa forma, defindo o índice como sendo o ITAE, Eq. (36), o qual dedica-se em minimizar o erro em regime permanente, e utilizando o sobressinal máximo, M_S, como critério de restrição, o projeto do sistema de controle pode ser visto como um problema de otimização:

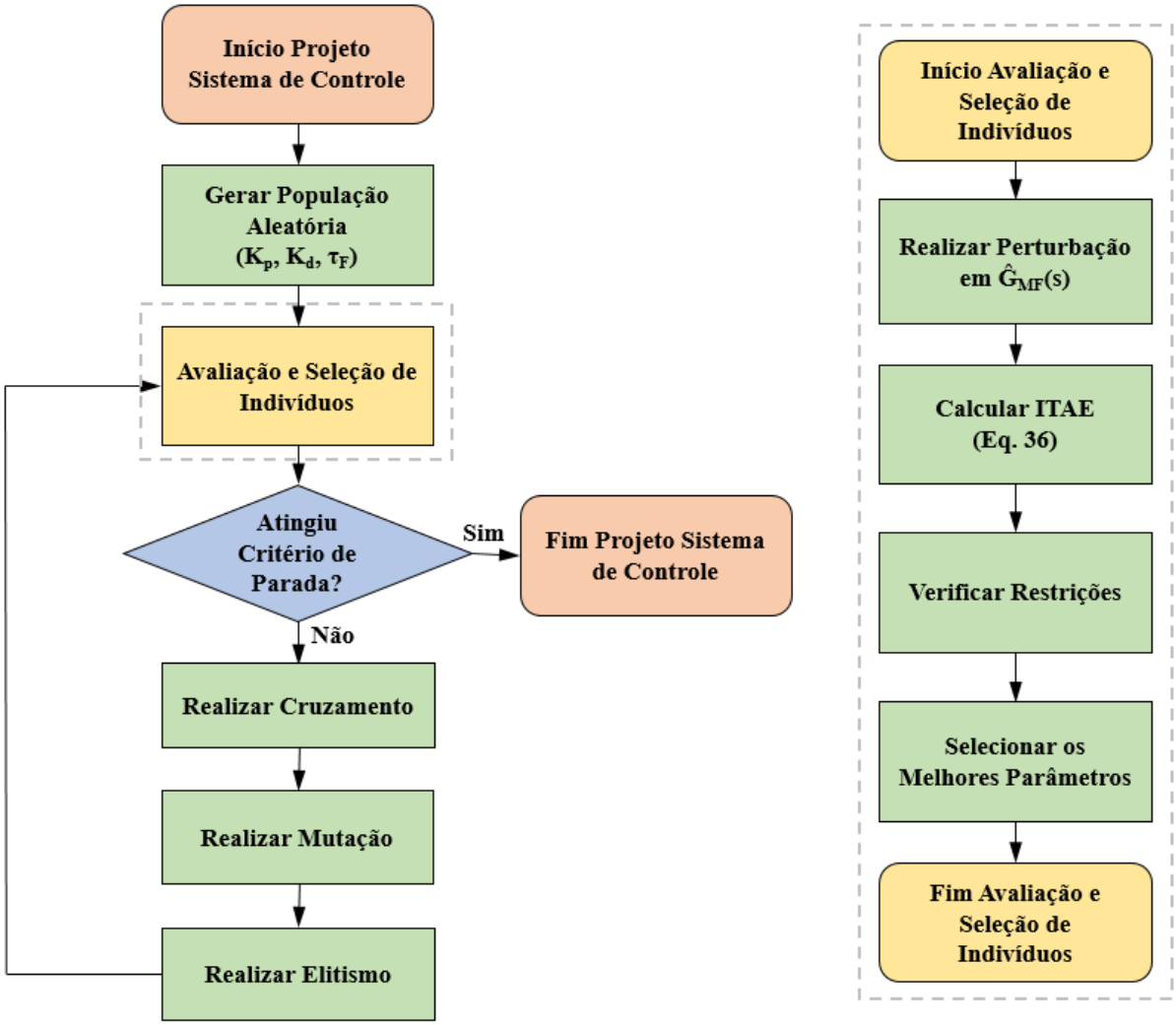
$$K_C^* = \arg\min_{K_C} J_{ITAE}(K_C)$$

$$\text{Sujeito a } M_{s_i} \leq \overline{M}_{s_i}$$

Este problema pode ser resolvido utilizando Algoritmos Genéticos, denotado no Apêndice F, conforme os passos a seguir, com fluxograma apresentado na Fig. 14.

- Primeiramente, uma população inicial com valores aleatórios dos parâmetros de controle é produzida.
- Em seguida, durante a etapa de avaliação e seleção, cada indivíduo da população gera uma função de transferência do sistema em malha fechada, $\hat{G}_{MF}(s)$, na qual realiza-se uma perturbação utilizando uma entrada do tipo degrau, $u(t)$.
- Posteriormente, calcula-se o ITAE com os dados da resposta obtida e verifica-se a restrição de sobressinal máximo, M_S;
- Por último, seleciona os melhores parâmetros do controle, K_C^*.
- Caso tenha atingido os critérios de parada, o algoritmo finaliza. Senão, são realizados os processos de cruzamento, mutação e elitismo, para que eles retornem à etapa de avaliação para repetição do processo.

Figura 14: Fluxograma do Algoritmo para Projeto de Sistema de Controle.



Fonte: autor.

CAPÍTULO 4: PLANEJAMENTO DE TRAJETÓRIAS

Trajetórias são definidas como caminhos com restrições temporais. Elas são as referências de movimento para o sistema de controle. De acordo com Zhao (2015), o planejamento de trajetórias de robôs é importante pois elas garantem:

- **Segurança e Conforto**: por possibilitar limitar as velocidades e acelerações das juntas;
- **Otimização**: ao integrar geometria, tempo e eficiência energética;
- **Flexibilidade**: por serem facilmente adaptadas e/ou transformadas;
- **Viabilidade**: ao proporcionar a verificação de possíveis movimentos em relação às restrições dos controladores dos níveis inferiores.

Existem diversos critérios para classificar trajetórias. Siciliano *et al*. (2008) considera que os principais critérios são:

- **O espaço de definição**: espaço cartesiano ou de junta.
- **Tipo de tarefa**: ponto-a-ponto, multíplos pontos, contínua ou concatenada.
- **Dimensão**: uma ou várias dimensões.
- **Geometria do caminho**: retilínea, curvilínea, circular, etc.
- **Lei de tempo**: velocidade trapezoidal, polinomial, etc.

Além desses critérios, elas podem ser classificadas de acordo com a complexidade das restrições a partir das quais foram geradas e do modo de funcionamento, podendo ser *online* ou *off-line* (Zhao, 2015). No primeiro caso, tem-se situações onde o ambiente é dinâmico, sendo assim, implementa-se uma rotina de programação que será executada em tempo real, atribuindo ao manipulador uma capacidade de reação naquele meio. No segundo caso, no planejamento *off-line*, o ambiente é conhecido, as trajetórias são planejadas considerando que não haverá mudanças nesse ambiente, ou seja, que ele será estático. Por isso, algoritmos de planejamento de trajetórias *off-line* possuem custos computacionais relativamente maiores, uma vez que pode levar em consideração mais detalhes acerca do ambiente e utilizar critérios com complexidades maiores.

É importante diferenciar também as formas que as trajetórias são programadas, o que também se divide em *on-line* e *off-line*. A programação *on-line* é realizada por meio de aprendizagem: o robô é guiado através das posições desejadas e isso é atribuído em sua rotina. Na programação *off-line*, a tarefa é indicada mediante o uso de um *software* com linguagem de programação de alto nível: o *software* processa os dados fornecidos pelo usuário e aciona os motores do manipulador, o qual executa os movimentos conforme foram planejados.

As características da trajetória devem ser escolhidas de acordo com a tarefa especificada. Por exemplo, em uma tarefa de soldagem, a trajetória é mais facilmente construída no espaço cartesiano, pois a tarefa demanda um caminho contínuo nesse espaço de operação além de restrições de velocidade. Já em tarefas do tipo *pick-and-place*, onde existem mais possibilidades de caminhos entre as posições especificadas, o espaço das juntas também pode ser considerado. Outrossim, outro fator de decisão é o número de graus de liberdade do robô: manipuladores com baixos graus de liberdade permitem a utilização de métodos mais complexos no planejamento, uma vez que eles apresentam modelos matemáticos mais simplificados e com menores custos computacionais.

As curvas contínuas estão entre as formas mais utilizadas para modelagem de trajetórias. Por isso, a Seção 4.1. trata-se sobre a teoria acerca de curvas aplicadas no planejamento de trajetórias, a Seção 4.2. aborda a modelagem matemática de obstáculos, esses que restringem o espaço de operação de manipuladores, e a Seção 4.3. apresenta um algoritmo para o planejamento de trajetórias livres de colisões.

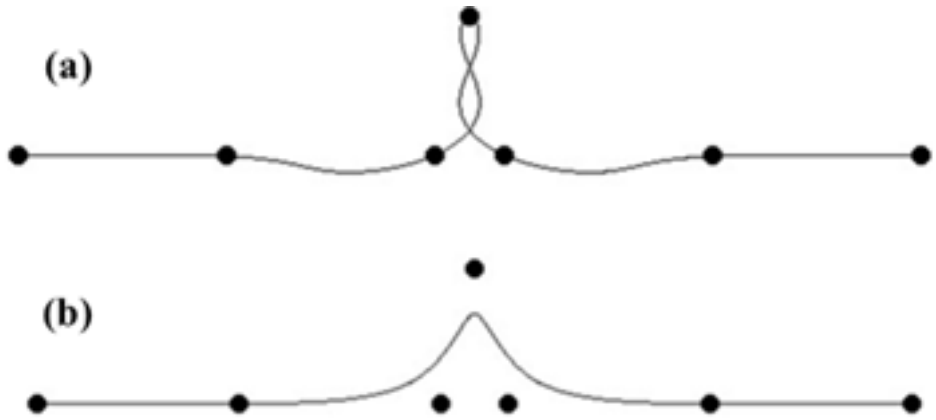
4.1. Curvas Contínuas

A modelagem de curvas começou com construtores de barcos, com o propósito de mecanizar operações. Posteriormente, durante a década de 60, as manufaturas de carros e aviões investiram em computadores na produção para automatizar o design de produtos. Diversas foram as contribuições para a modelagem de curvas e superfícies. Entre os autores que se destacaram nos avanços estão: Pierre Bézier da Renault, Paul de Casteljau da Citroen, Steve Coons da Ford Motors, Willian Gordon e Riesenfeld da General Motors etc. (Salomon, 2005).

Curvas são definidas, para a generalidade, como linhas arqueadas não necessariamente retas. Mortenson (1997) as define como o *locus* de um ponto que se move com um grau de liberdade. Quanto a classificação, podem ser dividas em não-paramétricas e paramétricas. As primeiras possuem um uso mais limitado, pois são funções matemáticas implícitas e explícitas

as quais são dependentes do sistema de coordenadas, problema que é contornado pelas curvas paramétricas, em que as variáveis do sistema de coordenadas são representadas através de parâmetros independentes, permitindo uma maior flexibilidade em suas confecções.

Figura 15: Curva de (a) interpolação e de (b) aproximação.



Fonte: Adaptado de Durand and Cutler.

Se porventura uma curva passar pelas coordenadas de um conjunto de pontos, tem-se um caso de interpolação. Caso ela apenas se aproxime das coordenadas dos pontos, fazendo uma espécie de média, tem-se um problema de aproximação. Segundo Durand and Cutler, é mais difundido o uso de curvas de aproximação, uma vez que as curvas de interpolação podem apresentar oscilações indesejadas em certas situações, como pode ser visto na Fig. 15 (a).

As próximas seções visam apresentar as formulações matemáticas de algumas curvas aplicadas no Planejamento de Trajetória de Manipuladores: Seguimento Linear, Seção 4.1.1., de Bézier, Seção 4.1.2., B-Spline, Seção 4.1.3., e Polinomiais, Seção 4.1.4..

4.1.1. Segmento Linear

Seja $x(u) \in \mathbb{R}$ o segmento entre as coordenadas de dois pontos x_f e $x_i \in \mathbb{R}$, sua forma matemática é dada pela Eq. (37), onde u é um parâmetro que varia entre $0 \leq u \leq 1$.

$$x(u) = x_i + u(x_f - x_i) \tag{37}$$

Tal equação pode ser utilizada, por exemplo, para modelagem do caminho e orientação que serão seguidos pelo efetuador final do robô manipulador.

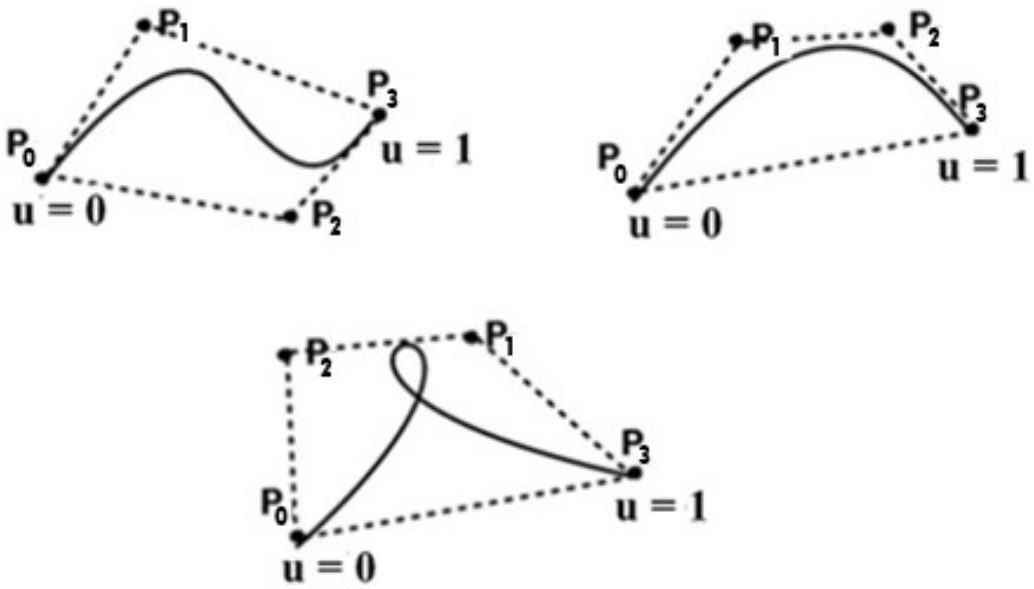
4.1.2. Curvas de Bézier

A curva de Bézier de grau n é formulada pela Eq. (38), onde os coeficientes geométricos, P_i, são denominados de pontos de controle, u é um parâmetro que assume valores entre $0 \leq u \leq 1$ e $B_{i,n}(t)$ são polinômios de Bernstein de grau n.

$$r(u) = \sum_{i=0}^{n} B_{i,n}(u) P_i$$
$$B_{i,n}(u) = \frac{n!}{i!\,(n-i)!} u^i (1-u)^{n-i} \tag{38}$$

A curva foi desenvolvida para interpolar os pontos da extremidade e aproximar dos pontos restantes, conforme a Fig. 16. O número de pontos de controle da curva altera-se quando sua ordem é modificada.

Figura 16: Exemplos de Curvas de Bézier Cúbicas.



Fonte: Adaptado de Durand and Cutler.

Por exemplo, para uma Bézier cúbica, onde $n = 3$, são utilizados 4 pontos de controle em sua confecção. Ela é obtida algebricamente e matricialmente conforme as Eq. em (39).

$$r(u) = (1-u)^3 P_0 + 3u(1-u)^2 P_1 + 3u^2(1-u)P_2 + u^3 P_3$$

$$r(u) = [u^3 \quad u^2 \quad u \quad 1] \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} P_0 \\ P_1 \\ P_2 \\ P_3 \end{bmatrix} \tag{39}$$

A Fig. 16 apresenta exemplos de curvas de Bézier cúbicas, que passam pelas coordenadas dos pontos P_0 e P_3 e aproximam dos pontos P_1 e P_2.

4.1.3. Curvas B-Spline Uniformes

Diferentemente da curva de Bézier, a curva B-Spline não é restringida a interpolar nenhum ponto de controle, ela somente se aproxima deles. Além disso, a ordem da curva é indepentende do número de pontos de controle (Mortenson, 1997). Ela é comumente formulada utilizando a forma recursiva de Cox DeBoor, que a descreve como uma combinação linear entre os pontos de controle, P_i, e as Funções de Base, $N_{i,k}\,(u)$, conforme a Eq. (40).

$$r(t) = \sum_{i=0}^{n} N_{i,k}(t) P_i$$

$$N_{i,1}(t) = \begin{Bmatrix} 1\ para\ t_i \le t \le t_{i+1} \\ 0\ nos\ demais\ intervalos \end{Bmatrix} \tag{40}$$

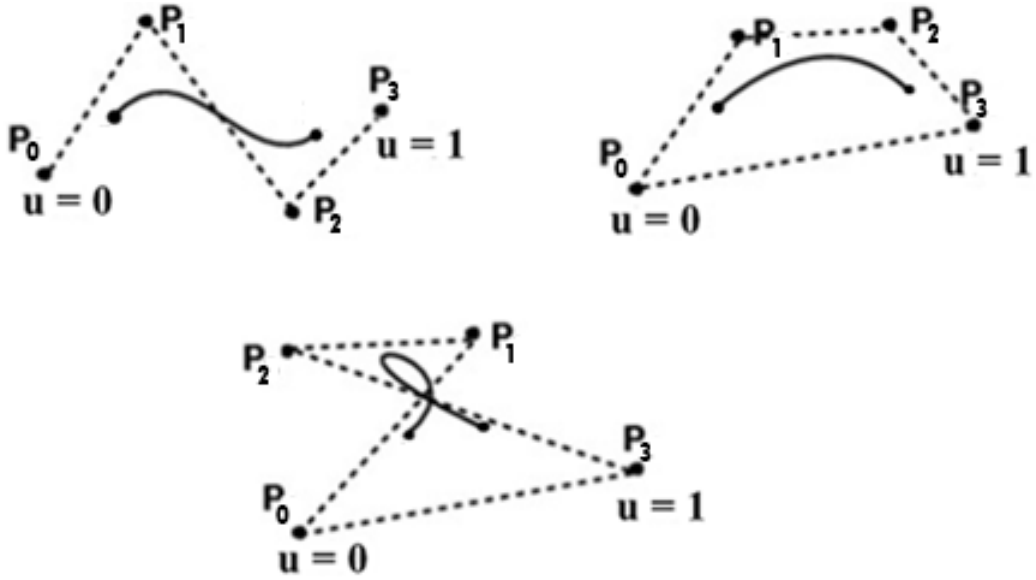
$$N_{i,k}(t) = \frac{t - t_i}{t_{i+k-1} - t_i} N_{i,k-1}(t) + \frac{t_{i+k} - t}{t_{i+k} - t_{i+1}} N_{i+1,k-1}(t)$$

Nessa equação, t_i é ponto da curva que representa um nó (a quantidade de nós é dada por $n + k + 1$, por isso, o índice dos nós varia entre 0 e $n + k$), k é o grau da curva e $n + 1$ é a quantidade de pontos de controle.

Se uma curva B-Spline possui os nós igualmente espaçados, $t_{i+1} - t_i = cte$, a curva é chamada de Uniforme, caso contrário, ela é denominada Não-Uniforme. Um exemplo de curva Não-Uniforme é a NURBS ("*Non-Uniform Rational B-Splines*") a qual também se encontra em

uma forma racional. Entretanto, devido maior simplicidade, a curva B-Spline Cúbica Uniforme é mais utilizada no Planejamento de Trajetórias, que é calculada através de 4 pontos de controle e ilustrada através da Fig. 17.

Figura 17: Exemplos de Segmentos de Curvas B-Spline.



Fonte: Adaptado de Durand and Cutler.

De acordo com Mortenson (1997), o cálculo de um segmento parte da Eq (40) e pode ser realizado como uma função de um parâmetro u, onde $0 \leq u \leq 1$. A obtenção algébrica e matricial do segmento $r(u)$ é dada pela Eq. (41).

$$r(u) = \frac{(1-u)^3}{6} P_0 + \frac{3u^3 - 6u^2 + 4}{6} P_1 + \frac{-3u^3 + 3u^2 + 3u + 1}{6} P_2 + \frac{u^3}{6} P_3$$

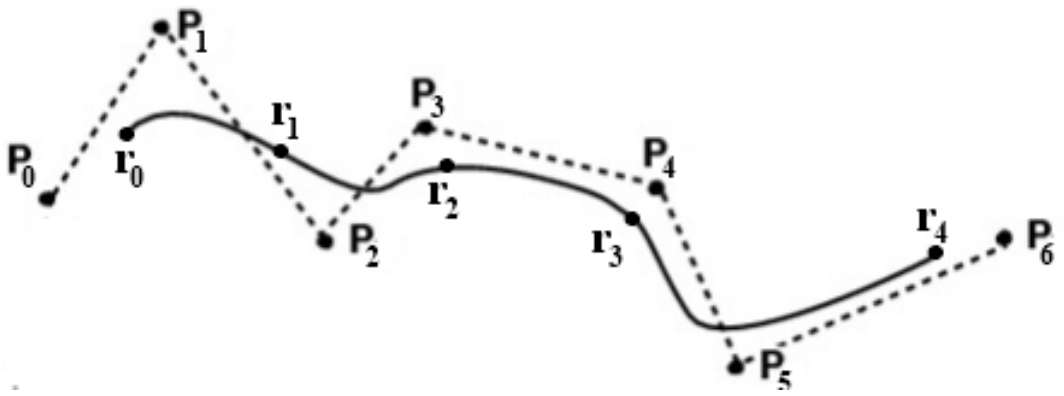
$$r(u) = \frac{\begin{bmatrix} 1 & u & u^2 & u^3 \end{bmatrix}}{6} \begin{bmatrix} 1 & 4 & 1 & 0 \\ -3 & 0 & 3 & 0 \\ 3 & -6 & 3 & 0 \\ -1 & 3 & -3 & 1 \end{bmatrix} \begin{bmatrix} P_0 \\ P_1 \\ P_2 \\ P_3 \end{bmatrix} \tag{41}$$

As coordenadas das extremidades de cada segmento podem ser calculadas como uma função dos pontos de controle utilizando $u = 0$ e $u = 1$, conforme as Eq. em (42).

$$r(0) = \frac{1}{6}(P_0 + 4P_1 + P_2)$$
$$r(1) = \frac{1}{6}(P_1 + 4P_2 + P_3) \quad (42)$$

A ideia-chave do algoritmo recursivo de Cox DeBoor é utilizar vários segmentos de curva para a formação de uma curva maior, conforme a Fig. 18.

Figura 18: Curvas B-Spline.



Fonte: Adaptado de Durand and Cutler.

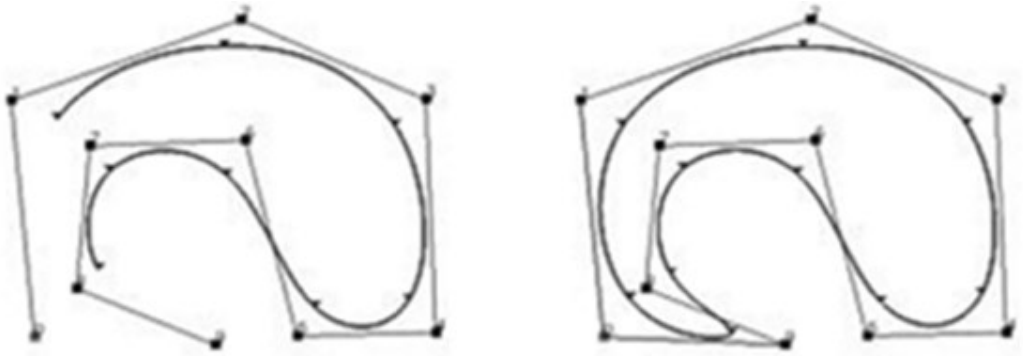
Para manter a continuidade entre dois segmentos, a seguinte condição deve ser respeitada: $r_i(1) = r_{i+1}(0)$, onde i é um segmento e $i + 1$ o segmento sucessor. Pois, isso irá garantir que os pontos de extremidade final de uma curva sejam coincidentes ao ponto de extremidade inicial da curva seguinte. A fins de exemplo, tem-se a Fig. 18, em que o primeiro segmento é influenciado pelos pontos P_0, P_1, P_2 e P_3 e possui extremidades inicial e final iguais a r_0 e r_1, respectivamente, enquanto o segundo segmento é influenciado pelos pontos P_1, P_2, P_3 e P_4 e possui extremidades inicial e final iguais a r_1 e r_2.

Percebe-se que para $n + 1$ pontos de controle é possível obter somente $n - 1$ pontos de extremidade: a Eq. (42) pode ser reproduzida somente $n - 1$ vezes. Logo, para calcular os $n + 1$ pontos de controle é necessário adicionar duas novas equações ao sistema. A solução para isso é utilizar equações de restrições. Por exemplo, pode-se impor que as derivadas de primeira ordem nas extremidades da curva "global" sejam iguais a zero, conforme as Eq. em (43).

$$\dot{r}_0 = \frac{1}{2}(P_2 - P_0) = 0$$
$$\dot{r}_{n-1} = \frac{1}{2}(P_{n+1} - P_{n-1}) = 0 \tag{43}$$

Essa restrição é utilizada no Planejamento de Trajetórias pois garante que a curva apresente suavidade nas extremidades. Todavia, pode-se utilizar outros tipos de restrições, dependendo da especificidade do problema a ser resolvido. Por exemplo, pode-se estabelecer uma curva do tipo fechada, onde as duas extremidades se conectam e possuem uma mesma inclinação, como pode ser visto na Fig. 19 (b).

Figura 19: Curvas B-Spline com diferentes restrições.



(a) Curva Aberta **(b) Curva Fechada**

Fonte: Adaptado de Barbarini, L. H. M. (2007).

Então, adicionando as restrições de derivadas de primeira ordem nulas nas extremidades, passando para forma matricial e isolando o vetor de pontos de controle, o cálculo desses pontos é realizado conforme a Eq. (44).

$$\begin{bmatrix} P_0 \\ P_1 \\ P_2 \\ P_3 \\ \vdots \\ P_{n-1} \\ P_n \end{bmatrix} = \left(\frac{1}{6} \begin{bmatrix} 1 & 4 & 1 & 0 & \cdots & 0 & 0 & 0 & 0 \\ 0 & 1 & 4 & 1 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 0 & 1 & 4 & 1 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ -1 & 0 & 1 & 0 & \cdots & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 & -1 & 0 & 1 \end{bmatrix} \right)^{-1} \begin{bmatrix} r_0 \\ r_1 \\ r_2 \\ \vdots \\ r_{n-2} \\ 0 \\ 0 \end{bmatrix} \tag{44}$$

Logo, a curva $r(u)$ é uma função indireta dos pontos de extremidade e é obtida resolvendo as Eq. (44) e (41), respectivamente.

Diferentemente da curva de Bézier, a curva B-Spline, conforme foi formulada, permite alterar os pontos de controle sem comprometer sua integridade, pois sua modificação acontece apenas localmente. Essa propriedade proporciona um uso difundido dessas curvas.

4.1.4. Curvas Polinomiais

Um polinômio de ordem n, $r(t)$, é representado matematicamente conforme a Eq. (45). Onde a_i são os coeficientes do polinômio, n é a sua ordem e i o número do coeficiente, para $i \in \{0,1,2,\dots,n\}$.

$$r(t) = \mathrm{a}_0 + \mathrm{a}_1\mathrm{t} + \mathrm{a}_2 t^2 + \mathrm{a}_3 t^3 + \cdots + \mathrm{a}_\mathrm{n} t^\mathrm{n} \tag{45}$$

Os valores de posições, velocidades, acelerações e arranques nas extremidades de uma curva devem ser fornecidos conforme estabelecido na Tab. 1. Assim sendo, é assegurado que a trajetória calculada cumpra tais restrições.

Tabela 1: Restrições para trajetórias suaves nas extremidades.

Tempo, t	t_i	t_f
Posição, $r(t)$	r_i	r_f
Velocidade, $\dot{r}(t)$	v_i	v_f
Aceleração, $\ddot{r}(t)$	a_i	a_f
Arranque, $\dddot{r}(t)$	j_i	j_f

Fonte: autor.

A ordem do polinômio é determinada de acordo com as restrições que se deseja adicionar. Por exemplo, para satisfazer 4 restrições da Tab. 1, isto é, garantir que pelo menos as velocidades sejam nulas nas extremidades, pode-se utilizar Polinômios cúbicos, pois eles possuem 4 coeficientes na formulação. A quantidade de critérios que um polinômio pode

satisfazer é definida como o valor de sua ordem mais 1, $n+1$. Por exemplo, para satisfazer todos os critérios da Tab. 1, seria necessário, no mínimo, um polinômio de Sétima Ordem, devido a eles possuírem 8 coeficientes.

Para problemas de otimização, utiliza-se polinômios com ordens elevadas para que haja mais variáveis do que necessário para encontrar uma solução. Uma vez que esses polinômios introduzem graus de liberdade na solução do problema, possibilitam otimizar o formato da curva com os critérios desejados. Por exemplo, um polinômio de quarta ordem, além de garantir cumpridas de um polinômio de terceira ordem, admitiria que uma variável permaneça livre para ser modificada. No Apêndice D estão explicitadas as equações para o cálculo dos coeficientes dos polinômios de 3ª à 8ª ordem.

4.2. Modelagem de Obstáculos

Os obstáculos e as fronteiras dos espaços de trabalho de um robô manipulador podem ser encapsulados através de Elipsoides. O modelo matemático implícito de um Elipsoide é denotado pela Eq. (46).

$$f_{el}(x,y,z) = \frac{(x-x_c)^2}{x_r^2} + \frac{(y-y_c)^2}{y_r^2} + \frac{(z-z_c)^2}{z_r^2} - 1 = 0 \tag{46}$$

No caso em que os parâmetros radiais são equivalentes, a Eq. (46) torna-se a equação implícita de uma esfera, que é dada pela Eq. (47), na qual R é o raio da esfera.

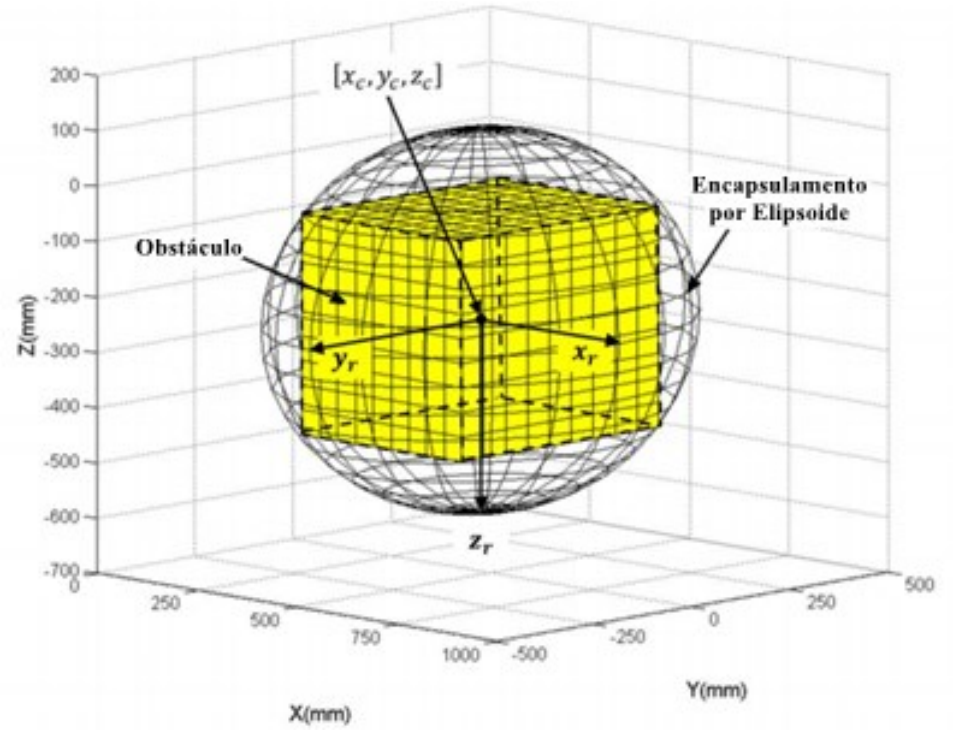
$$f_{es}(x,y,z) = \frac{(x-x_c)^2}{R^2} + \frac{(y-y_c)^2}{R^2} + \frac{(z-z_c)^2}{R^2} - 1 = 0 \tag{47}$$

Para calcular os parâmetros dos elipsoides, pode-se considerar que o vetor de posição do elipsoide, $r_C = [x_c \;\; y_c \;\; z_c]^T$, coincida com o vetor de posição do obstáculo. Os parâmetros radiais, $r_R = [x_r \;\; y_r \;\; z_r]^T$, podem ser funções das dimensões dos obstáculos (com formato de prisma retangular): altura (h), largura (w) e comprimento (d), conforme as Eq. em (48).

$$x_r = k_x w; \qquad y_r = k_y d; \qquad z_r = k_z \mathrm{h} \tag{48}$$

Na Fig. 20 tem-se um Elipsoide encapsulando um obstáculo e são mostrados seus parâmetros radiais e vetor de posição.

Figura 20: Um elipsoide encapsulando um obstáculo com formato de prisma retangular.



Fonte: Adaptado de Mondragon (2013).

Outra forma de calcular os parâmetros do elipsoide, é através de um problema de otimização, conforme Martínez-Alfaro (1993). Considerando as posições dos centros do elipsoide do objeto coincidentes, as variáveis da otimização podem ser os parâmetros radiais do elipsoide e de sua orientação, escrita através dos Ângulos de Euler, $\phi = [\alpha \ \beta \ \gamma]^T$, como denotado pelo vetor τ_e em (49).

$$\tau_e = [x_r \ y_r \ z_r \ \alpha \ \beta \ \gamma]^T \tag{49}$$

O perímetro do obstáculo pode ser estimado utilizando uma quantidade de amostras dele, N_S, com posições $\hat{P}_S = (\hat{x}_i, \hat{y}_i, \hat{z}_i)$, para $i \in \{1, \dots, N_s\}$ (Martínez-Alfaro, 1993). O objeto,

o qual se encontra inicialmente em um sistema de coordenadas $o_0\hat{x}\hat{y}\hat{z}$, deve ter sua orientação ajustada para coordenadas o_0xyz. Isso é feito utilizando a matriz de rotação, $R(\phi)$, por meio da Eq. (50), onde $P_S = (x_i, y_i, z_i)$.

$$P_S = R(\phi)^{-1}\hat{P}_S \tag{50}$$

A função *fitness* do problema de otimização pode ser escrita como a diferença entre o volume do elipsoide e o volume do obstáculo, V_0, conforme a Eq. (51).

$$f(x_r, y_r, z_r, V_0) = \frac{4}{3}\pi x_r y_r z_r - V_0 \tag{51}$$

O conjunto de equações em (52) denotam as restrições do problema.

$$\begin{gathered} h_i(\tau_e, P_S) = (y_r z_r x_i)^2 + (x_r z_r y_i)^2 + (x_r y_r z_i)^2 - (x_r y_r z_r)^2 \leq 0 \\ c^2\alpha + s^2\alpha - 1 = 0 \\ c^2\beta + s^2\beta - 1 = 0 \\ c^2\gamma + s^2\gamma - 1 = 0 \end{gathered} \tag{52}$$

Logo, o problema do elipsoide que melhor encapsula um obstáculo, sujeito as restrições em (52), apresenta-se da seguinte forma:

$$\tau_e^* = \min_{\tau_e} f(\tau_e)$$

4.3. Planejamento de Trajetórias Livres de Colisões

Mondragon (2013), em seu trabalho, apresenta um algoritmo de planejamento de trajetórias no espaço cartesiano que são livres de colisões. As trajetórias são formuladas matematicamente utilizando segmentos lineares e a lógica para o desvio de obstáculos é bastante simples: os obstáculos são encapsulados por elipsoides; caso haja elipsoides interseccionando os segmentos lineares, a trajetória é modificada para não passar pelas coordenadas deles, evitando, por conseguinte, colisões com os obstáculos.

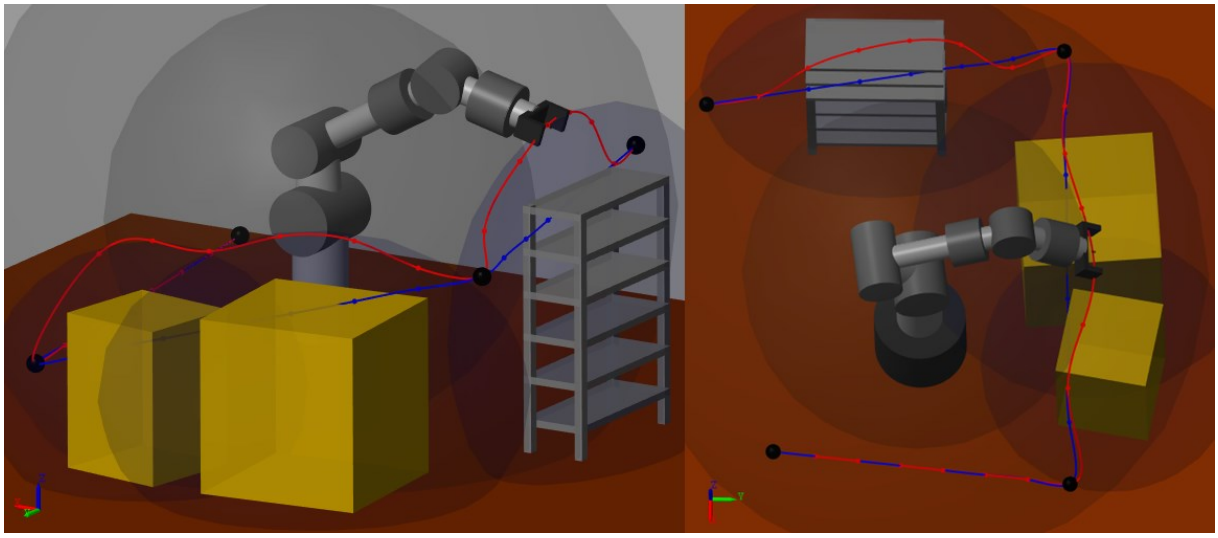
Baseando-se nessa ideia, desenvolveu-se um algoritmo para o Planejamento de Trajetórias Livres de Colisões. As entradas do algoritmo são dois pontos de posição, P_i e P_f, e dois de orientação, ϕ_i e ϕ_f. Analogamente ao trabalho de Mondragon (2013), os obstáculos são envolvidos por elipsoides, cujos parâmetros são calculados através da Eq. (48) ou resolvendo o problema de otimização da seção anterior como funções das dimensões dos obstáculos. Ademais, os limites do espaço de trabalho do robô são modelados utilizando duas esferas: uma externa e outra interna, as quais representam, respectivamente, os alcances máximos e as restrições mecânicas do manipulador.

Existem duas diferenças entre a metodologia utilizada e a do trabalho de Mondragon (2013). A primeira é em relação às restrições temporais da trajetória: no presente trabalho foram utilizadas curvas B-Spline para esse fim, enquanto no trabalho de Mondragon (2013), foi realizado, apenas, o planejamento dos caminhos livres de colisões. A segunda se refere ao cálculo das orientações: no presente trabalho elas foram calculadas utilizando segmentos lineares entre as orientações inicial e final dadas, enquanto no trabalho de Mondragon (2013) isso foi realizado por meio de uma análise de destreza. O algoritmo desenvolvido baseia-se nas seguintes etapas:

- Primeiramente, são gerados n pontos que estão compreendidos no segmento linear entre as posições, P_i e P_f, utilizando a Eq. (37), os quais são chamados de Pontos de Posição. Em seguida, são gerados n pontos inseridos no segmento linear entre as orientações, ϕ_i e ϕ_f, que são chamados de Pontos de Orientação.
- Para certificar que os Pontos de Posição estejam em posições factíveis pelo manipulador, as coordenadas deles são substituidas na equação implícita do elipsoide e da esfera, Eq. (46) e (47), respectivamente, de três maneiras distintas.
- A primeira maneira consiste em verificar se os pontos estão em rota de colisão com os obstáculos, dentro de seus elispoides, nesse caso, $f_{el}(P_X, P_Y, P_Z) \leq 0$. Se sim, a coordenada z do ponto muda para a coordenada z do elipsoide, conforme a Fig. 21 (a), na qual a trajetória inicial está em azul e a alterada em vermelho.
- A segunda baseia-se em checar se os pontos estão fora dos limites de trabalho do robô, fora da esfera externa, nessa ocasião, $f_{es}(P_X, P_Y, P_Z) > 0$. Dessa forma, as coordenadas xy do ponto mudam para as coordenadas xy da esfera, caso contrário, nada acontece.

- A terceira consiste em verificar se as coordenadas estão em posições de condições mecanicamente restritas, em outras palavras, dentro da esfera interna, nessa situação, $f_{es}(P_X, P_Y, P_Z) \leq 0$. Assim sendo, as coordenadas xy do ponto mudam para as coordenadas xy da esfera, senão, nada acontece.

Figura 21: Trajetórias sendo modificadas devido (a) obstáculo e (b) esfera interna.

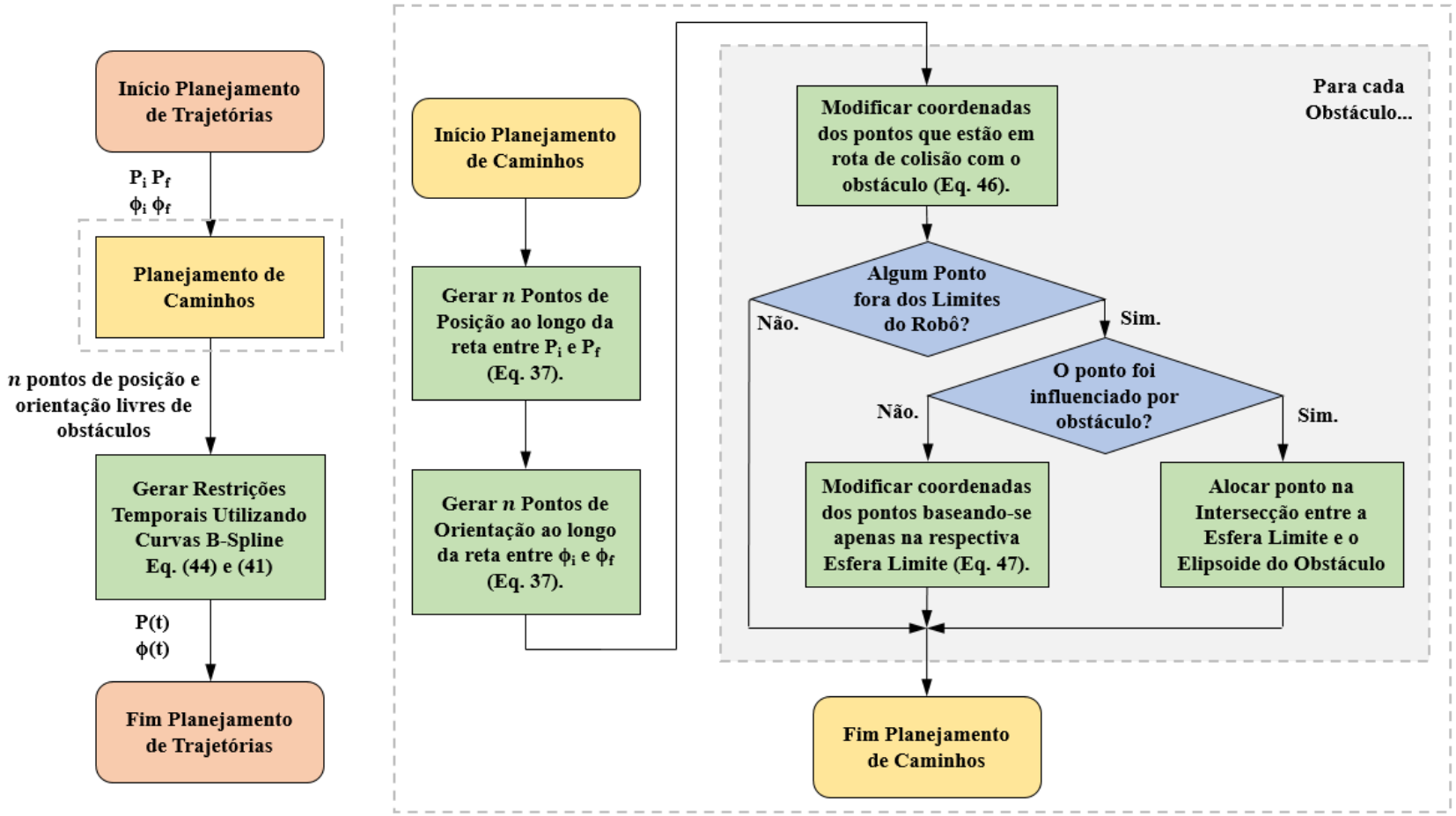


Fonte: autor.

- Os pontos que estão tanto fora dos limites do espaço de trabalho quanto em rota de colisão são realocados na intersecção do elipsoide do obstáculo e da respectiva esfera de trabalho, conforme acontece no exemplo da Fig. 21 (b), na qual a trajetória inicial está em azul e a alterada em vermelho.
- Após todas as verificações, os pontos estão em condições factíveis pelo manipulador, isto é, o caminho livre de colisões foi gerado. Dessarte, as restrições temporais são incluidas ao utilizar os pontos de posição e orientação como pontos de junção da *B-Spline* cúbica, apresentada na Seção 4.1.3., e a trajetória é criada. Ademais, o parâmetro u deve ser ajustado como uma função do tempo para possibilitar a obtenção de curvas $P(t)$ e $\phi(t)$, que devem ser geradas para cada coordenada cartesiana.

O fluxograma do Algoritmo para o Planejamento de Trajetórias Livres de Colisões é exposto na Fig. 22.

Fig 22: Fluxograma do Algoritmo para Planejamento de Trajetórias Livres de Colisões. Modificar equações



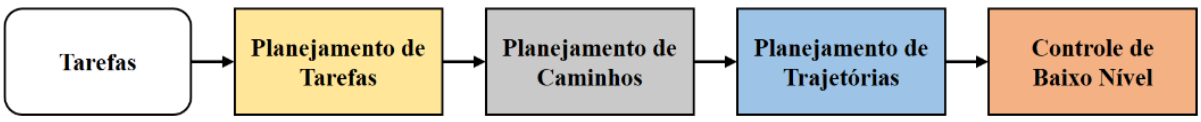
Fonte: autor.

CAPÍTULO 5: PLANEJAMENTO DE MOVIMENTOS

Segundo Sicilliano (2008), os métodos de Planejamento de Trajetórias operam assumindo que o espaço de trabalho do manipulador esteja vazio e, ao incluir restrições ao ambiente como, por exemplo, a presença de obstáculos, deve-se utilizar métodos de Planejamento de Movimentos. Assim sendo, para o autor, o Planejamento de Movimentos é visto como sendo uma extensão do Planejamento de Trajetórias, o qual inclui dados acerca do ambiente. Todavia, em algumas situações, o Planejamento de Movimentos é tratado de maneira análoga ao que é chamado de Planejamento de Caminhos, no qual o objetivo principal é apenas encontrar o caminho livre de colisões entre duas posições, por isso, os métodos baseados em grades e amostragem, apresentados no Capítulo 2, os quais são métodos de Planejamento de Caminhos, também são denominados como métodos de Planejamento de Movimentos (Sicilliano, 2008), o que gera confusão entre as terminologias.

Outros autores como, por exemplo, Sebastian (2019), definem o Planejamento de Movimentos como sendo uma hierarquia, a qual engloba todos os outros problemas: os Planejamentos de Tarefas, Caminhos e Trajetórias, e o Controle de Baixo Nível, conforme a Fig. 23.

Figura 23: Hierarquia do Planejamento de Movimento.



Fonte: autor.

Essa definição de Planejamento de Movimentos é que foi considerada no presente trabalho. Aqui, o Planejamento de Tarefas consiste em definir uma sequência de atividades para a realização de uma tarefa robotizada. Por exemplo, em uma tarefa do tipo *Pick-and-place*, a sequência poderia ser: pegar uma caixa de um local e transportá-la até uma esteira. O Planejamento de Caminhos consiste em transcrever tal sequência em uma forma concebível pela máquina: como um conjunto de pontos no Espaço Cartesiano que o efetuador final do robô

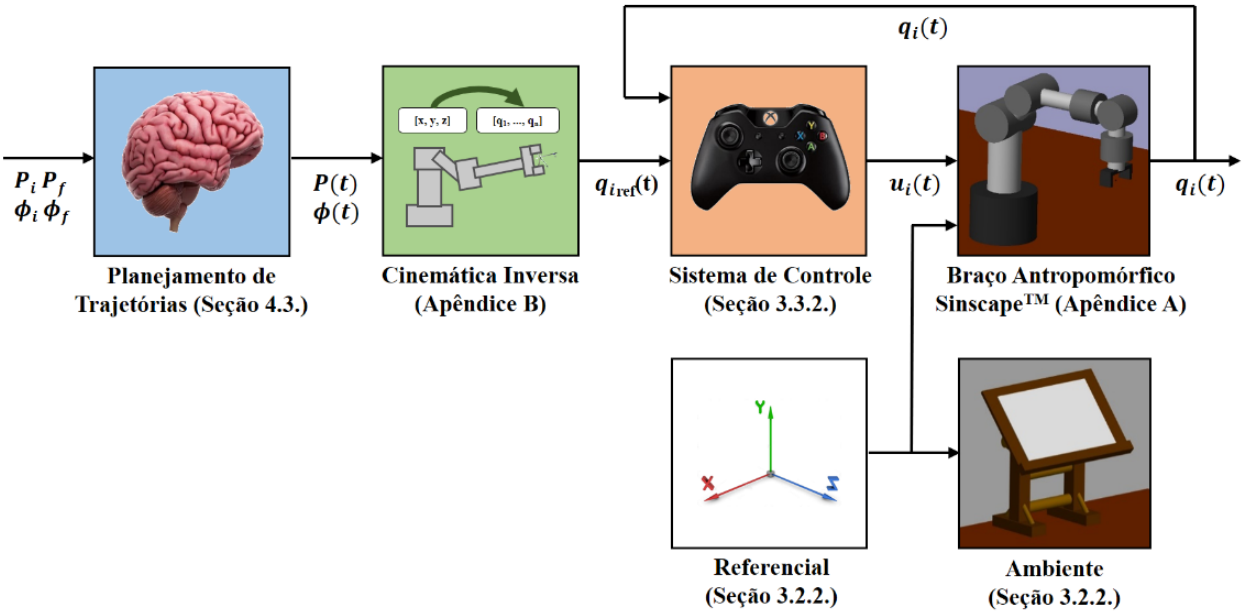
deva passar. O conjunto de pontos representa o caminho para a execução da tarefa. Quando são acrescentadas restrições temporais, ou seja, instantes de tempo associados aos pontos, temos o que é definido como trajetória. Para que o efetuador final possa rastrear a trajetória no espaço cartesiano é necessário que os controladores dos motores das juntas gerem as posições e orientações necessárias para esse rastreamento.

Este capítulo possui como objetivo tratar esses problemas e implementar um Planejamento de Movimentos para um robô manipulador. Por isso, ele é subdividido em duas partes: Metodologia utilizada, Seção 5.1., e os Resultados obtidos e as Discussões realizadas, Seção 5.2..

5.1. Metodologia

A Fig. 24 ilustra a estratégia utilizada nas simulações para a obtenção dos movimentos das juntas do manipulador, $q_i(t)$, onde $i \in \{1,2,\dots,6\}$, na qual cada bloco é um subproblema a ser resolvido.

Figura 24: Esquema utilizado no Simulink® para geração de movimentos do Robô.



Fonte: autor.

As entradas do algoritmo são as condições iniciais e finais em termos de posições, P_i e P_f, e orientações, ϕ_i e ϕ_f, bem como os parâmetros do ambiente e do robô. Primeiramente, as trajetórias são planejadas no espaço cartesiano, $P(t)$ e $\phi(t)$, de maneira *off-line*, de acordo com o Algoritmo para Planejamento de Trajetórias Livres de Colisões (Seção 4.3.). Em seguida, elas são convertidas em posições de referência das juntas, $q_{i_{ref}}$, através do cálculo da Cinemática Inversa (Apêndice B), utilizando o método dos Helicoides Sucessivos para obtenção das matrizes de transformação. Posteriormente, os sinais provindos dos controladores (Seção 3.3.2.), $u_i(t)$, baseando-se no erro entre as posições, assegura que o efetuador final do manipulador rastreie as posições de referência das juntas, gerando o movimento necessário para o rastreamento da trajetória do efetuador final. É importante lembrar que o modelo construído no Simscape™ (Seção 3.2.2.) é equivalente ao modelo de Dinâmica Direta, representado Eq. (20). Dessa forma, sob a perspectiva de Teoria de Controle, as posições angulares das juntas, $q_i(t)$, são as variáveis controladas e os torques das juntas são as variáveis manipuladas. Para fins de simplificação, a dinâmica do motor foi considerada como sendo desprezível em relação a do robô e, por isso, ela não foi incluída ao modelo.

O manipulador considerado no Planejamento de Movimentos foi o Braço Antropomórfico com Punho Esférico, cujos parâmetros geométricos e dinâmicos utilizados são expostos no Apêndice A. Os referenciais adotados para o robô são apresentados como um resultado dos modelos de Cinemática Direta, na Seção 5.2.. Optou-se por escolher esse manipulador por ele possuir uma arquitetura completa para operar no espaço cartesiano, pois contêm 6 graus de liberdade. Em razão disso, o efetuador final pode ser posicionado e orientado livremente nas três dimensões desde que seja respeitado o espaço de trabalho do manipulador. Além disso, ele é bem conhecido na literatura e as soluções dos problemas de Cinemática e Dinâmica estão facilmente disponíveis.

Para realização das simulações, foi utilizado um notebook IdeaPad 320 com processador do tipo Intel® Core™ i5-7200U CPU @ 2.50 GHz (4 CPUs), ~2.7GHz. O *software* adotado para a implementação dos métodos desenvolvidos nesse trabalho foi o Matlab®, o qual contém o Simulink® e o Simscape™. Nele, utilizou-se o *solver* DAESSC (*solver* DAE para Simscape™) com um passo do tipo variável para a realização das simulações numéricas. As próximas seções apresentam mais detalhes sobre a implementação do sistema de controle das juntas e a respectiva identificação de suas dinâmicas, Seção 5.1.1., e sobre o Planejamento de Trajetórias e Tarefas, Seção 5.1.2..

5.1.1. Controle e Identificação dos Parâmetros das Juntas

Com intuito de simular o projeto de um sistema de controle de um manipulador real, desenvolveu-se um modelo do robô no Simscape™. O modelo foi tratado como um sistema na forma de uma relação entrada-saída, como sendo uma "caixa-preta". No projeto do sistema de controle, considerou-se a dinâmica do manipulador como sendo descentralizada, isto é, cada junta i, onde $i \in \{1, 2, \dots, 6\}$, possui uma dinâmica independente das outras, ou seja, um conjunto de sistemas SISO (*Single Input, Single Output*).

O projeto dos controladores começa com a identificação da dinâmica de cada junta como uma função de transferência. Para isso, empregou-se o Algoritmo de Identificação, que é detalhado no Apêndice E. Esse algoritmo é dependente de um sistema em malha fechada estável para a realização das estimativas. Por isso, utilizou-se o recurso de sintonia de controle PID do Matlab®, para projetar controladores "*a priori*". A Tab. 2 apresenta os valores dos parâmetros dos controladores e dos degraus para a obtenção das respostas dinâmicas.

Tabela 2: Parâmetros do sistema de controle utilizados para estimar Funções de Transferência e perturbações do tipo degrau aplicadas nas juntas do manipulador.

Junta	**Parâmetros Controle**			**Degrau 1**			**Degrau 2**		
	K_P	K_D	τ_F	**Tempo Degrau**	**Valor Inicial**	**Valor Final**	**Tempo Degrau**	**Valor Final**	**Valor Final**
1	800	80	50	3,0	0,0	$\pi/3$	1,5	0,0	$\pi/3$
2	10000	200	50	6,0	0,0	$\pi/3$	4,5	0,0	$\pi/3$
3	10000	200	50	9,0	0,0	$\pi/3$	7,5	0,0	$\pi/3$
4	500	100	10	12,0	0,0	$\pi/3$	10,5	0,0	$\pi/3$
5	500	100	10	15,0	0,0	$\pi/3$	13,5	0,0	$\pi/3$
6	500	80	10	18,0	0,0	$\pi/3$	16,5	0,0	$\pi/3$

Fonte: autor.

As funções de transferência das juntas obtidas são indicadas em (53).

$$G_{P1}(s) = \frac{2{,}425 \cdot 10^{-4}s^4 + 0{,}04249s^3 + 12{,}65s^2 + 830{,}8s + 1{,}371 \cdot 10^4}{s^5 + 125{,}8s^4 + 6028s^3 + 1{,}055 \cdot 10^5\, s^2 + 6{,}03 \cdot 10^5 s - 151{,}7}$$

$$G_{P2}(s) = \frac{2{,}307 \cdot 10^{-4}s^4 + 0{,}03418s^3 + 1{,}55s^2 + 76{,}41s + 3240}{s^5 + 73{,}53s^4 + 2899s^3 + 1{,}053 \cdot 10^5\, s^2 + 9{,}577 \cdot 10^5 s - 8{,}219 \cdot 10^4}$$

$$G_{P3}(s) = \frac{2{,}261 \cdot 10^{-4}s^4 + 0{,}2862s^3 + 32{,}06s^2 + 1231s + 6{,}617 \cdot 10^4}{s^5 + 227{,}4s^4 + 1{,}593 \cdot 10^4 s^3 + 6{,}601 \cdot 10^5\, s^2 + 1{,}536 \cdot 10^7 s - 6{,}363 \cdot 10^5} \quad (53)$$

$$G_{P4}(s) \frac{0{,}01165s^4 + 0{,}01462s^3 + 2699s^2 + 1{,}669 \cdot 10^5 + 1{,}525 \cdot 10^6}{s^5 + 372{,}1s^4 + 1{,}15 \cdot 10^5 s^3 + 5{,}528 \cdot 10^6\, s^2 + 3{,}829 \cdot 10^7 s - 2{,}864 \cdot 10^4}$$

$$G_{P5}(s) \frac{0{,}00758s^4 - 0{,}6374s^3 + 419{,}4s^2 + 3{,}264 \cdot 10^4 s + 4{,}564 \cdot 10^5}{s^5 + 180{,}1s^4 + 4{,}517 \cdot 10^4 s^3 + 1{,}97 \cdot 10^6\, s^2 + 1{,}949 \cdot 10^7 s + 1{,}026 \cdot 10^6}$$

$$G_{P6}(s) = \frac{0{,}003982s^4 + 0{,}8005s^3 + 1778s^2 + 1{,}886 + 1{,}475 \cdot 10^6}{s^5 + 562{,}5s^4 + 2{,}257 \cdot 10^5 s^3 + 1{,}81 \cdot 10^7\, s^2 + 8{,}112 \cdot 10^7 s - 1{,}58 \cdot 10^4}$$

Utilizando essas funções de transferência, os Sistemas de Controle foram projetados conforme o Algoritmo para Projeto de Controladores, Seção 3.3.2.. A Tab. 3 contém os parâmetros utilizados no Algoritmo Genético. Ademais, os limites dos parâmetros possíveis de serem atingidos no AG foram fixados em $0 \leq K_p \leq 50000$, $0 \leq K_D \leq 500$ e $0 \leq \tau_F \leq 500$.

Tabela 3: Parâmetros utilizados no Algortimo Genético para sintonia de Controle.

População	**Genótipo**	**Gerações**	**Taxa de Cruzamento**	**Taxa de Mutação**
200	5	100	0,5	0,5

Fonte: autor.

Em relação ao Algoritmo Genético, empregou-se os seguintes operadores: números reais como método de codificação, método da roleta como método de seleção e o *flat crossover* como o tipo de cruzamento. Além desses operadores, utilizou-se o elitismo, para manter o melhor indivíduo de cada geração.

5.1.2. Planejamento de Trajetórias e Tarefas

Foram elaboradas três tarefas para validação dos algoritmos: na primeira, o robô realiza uma tarefa do tipo *pick-and-place*; na segunda, evita colisões com obstáculos; na terceira, evita obstáculos e desenha sobre uma mesa. Os detalhes das tarefas são apresentados a seguir.

- **Tarefa 1: *Pick-And-Place***

O planejamento dessa tarefa foi realizado objetivando simular situações cotidianas que acontecem em âmbito industrial, nas quais robôs manipuladores apanham e transportam objetos de diferentes tipos. Nessa tarefa, o manipulador deve pegar uma caixa sobre uma mesa e colocá-la sobre uma esteira transportadora. Para isso, foram geradas 7 trajetórias nas quais o efetuador final percorre pontos de posição, P, e orientação, ϕ, com as coordenadas indicadas na Tab. 4.

Tabela 4: Coordenadas dos pontos de posição e orientação da Tarefa 1.

Numeração	**Pontos de Posição** $[x \quad y \quad z]^T$ (m)	**Pontos de Orientação** $[\alpha \quad \beta \quad \gamma]^T$ (rad)
1	$[0{,}475 \quad 0{,}000 \quad 0{,}220]^T$	$[0{,}00 \quad \pi/2 \quad 0{,}00]^T$
2	$[0{,}200 \quad -0{,}450 \quad 0{,}200]^T$	$[\pi/2 \quad \pi/2 \quad 0{,}00]^T$
3	$[0{,}200 \quad -0{,}450 \quad 0{,}120]^T$	$[\pi/2 \quad \pi/2 \quad 0{,}00]^T$
4	$[0{,}200 \quad -0{,}450 \quad 0{,}200]^T$	$[\pi/2 \quad \pi/2 \quad 0{,}00]^T$
5	$[0{,}150 \quad -0{,}0125 \quad 0{,}220]^T$	$[\pi/2 \quad \pi/2 \quad 0{,}00]^T$
6	$[0{,}100 \quad 0{,}425 \quad 0{,}330]^T$	$[\pi/2 \quad \pi/2 \quad 0{,}00]^T$
7	$[0{,}100 \quad 0{,}425 \quad 0{,}250]^T$	$[\pi/2 \quad \pi/2 \quad 0{,}00]^T$
8	$[0{,}100 \quad 0{,}425 \quad 0{,}330]^T$	$[\pi/2 \quad \pi/2 \quad 0{,}00]^T$

Fonte: autor.

As trajetórias foram concebidas utilizando o Algoritmo de Planejamento de Trajetórias, da Seção 4.3., o qual é alimentado com pares de pontos de posição e orientação, sendo denominados como condições iniciais e finais da trajetória. A Tab. 5 exibe as condições inciais e finais utilizadas na geração de cada trajetória.

Tabela 5: Condições iniciais e finais de posição e orientação de cada trajetória da Tarefa 1.

Trajetória	**1**	**2**	**3**	**4**	**5**	**6**	**7**
Condições Iniciais	P_1, ϕ_1	P_2, ϕ_2	P_3, ϕ_3	P_4, ϕ_4	P_5, ϕ_5	P_6, ϕ_6	P_7, ϕ_7
Condições Finais	P_2, ϕ_2	P_3, ϕ_3	P_4, ϕ_4	P_5, ϕ_5	P_6, ϕ_6	P_7, ϕ_7	P_8, ϕ_8

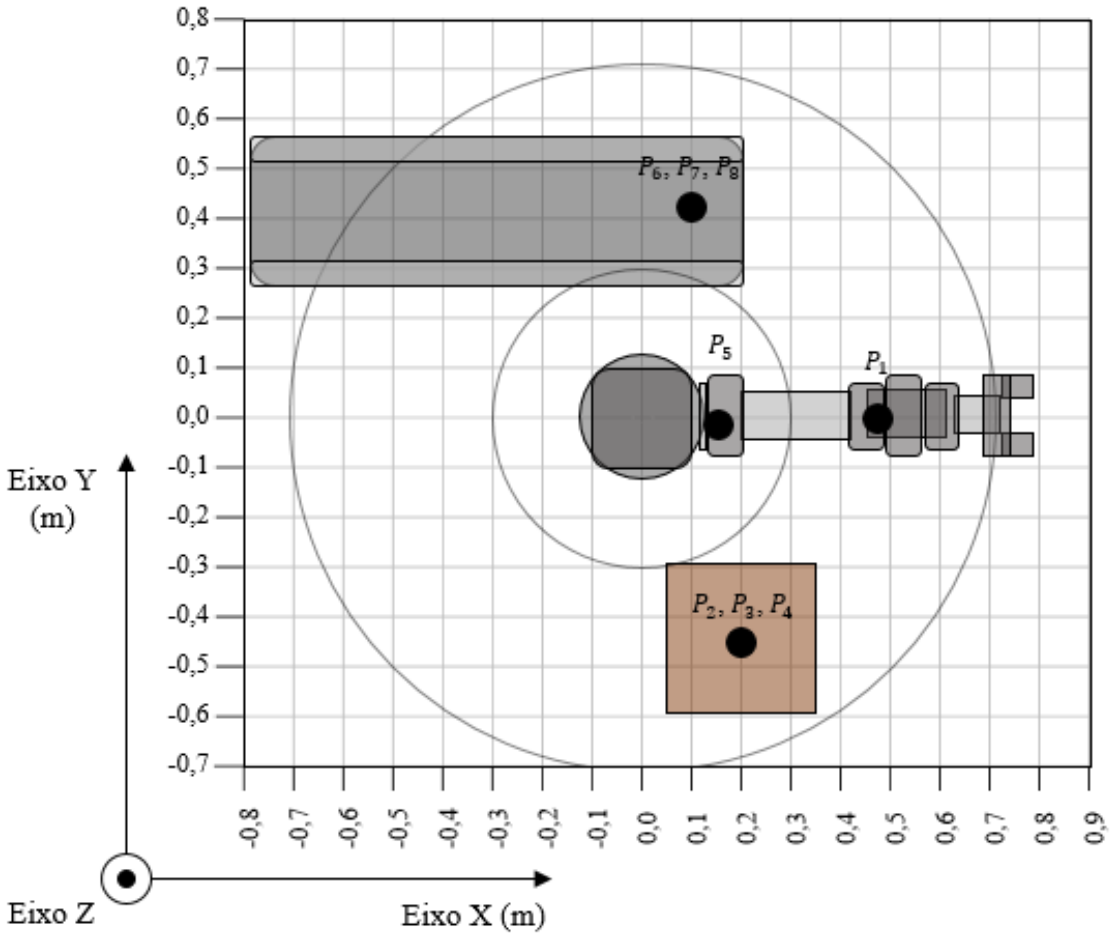
Fonte: autor.

As leis de tempo das trajetórias foram geradas por meio da curva *B-Spline*, na qual o parâmetro u foi utilizado como sendo uma função do tempo, dessa forma, foi estabelecido um intervalo de tempo de 2 segundos para cada uma delas. Ademais, os pontos de orientação foram escolhidos de maneira a manter o efetuador final com uma orientação constante e perpendicular às superfícies da mesa e da esteira.

Na Fig. 25 temos uma vista superior do ambiente dessa tarefa, nela, os pontos P_i com $i = \{1, \dots, 8\}$ denotam as posições iniciais e finais que constroem as trajetórias, apresentadas na Tab. 4, e o eixo Z está na direção normal do plano da página. O manipulador inicia o movimento do ponto P_1, se desloca até o ponto P_2, depois até o ponto P_3 e assim sucessivamente. As coordenadas dos pontos foram escolhidas de maneira que o manipulador não ficasse em condições de singularidade, segundo o Apêndice B.

Ao observar a Fig. 25, nota-se que os pontos P_2, P_3 e P_4 estão em uma mesma posição nos eixos X e Y, porém, como pode ser visto na Tab. 4, eles se diferenciam no eixo Z. As trajetórias geradas por esses pontos representam o ato de captura da caixa. O mesmo acontece com os pontos P_6, P_7 e P_8, que estão associados às trajetórias responsáveis pela liberação da caixa na esteira. Ademais, o ponto P_5 foi posto em uma região restrita do manipulador, dentro da esfera interna, com o intuito de verificar o funcionamento do algoritmo: a trajetória construída não deve passar por esse ponto, sendo modificada nos eixos X e Y.

Figura 25: Vista superior do ambiente na Tarefa 1.



Fonte: autor.

Em relação aos limites do espaço de trabalho do robô, os raios das esferas interna e externa utilizados nessa tarefa foram $0{,}30\ m$ e $0{,}71\ m$, respectivamente. É importante lembrar que as esferas possuem o objetivo de limitar o espaço de trabalho do manipulador e os raios foram determinados empiricamente.

- **Tarefa 2: Desvio de Obstáculos**

Essa tarefa visa, sobretudo, validar o algoritmo para desvio de colisões, apresentado na Seção 4.3. Dessa forma, o efetuador deve desviar de 3 obstáculos: duas caixas e um armário e,

para isso, foram formuladas 3 trajetórias, nas quais o robô passa pelos pontos de posição, P, e orientação, ϕ, cujas coordenadas são exibidas na Tab. 6.

Tabela 6: Coordenadas dos pontos de posição e orientação da Tarefa 2.

Numeração	**Pontos de Posição** $[x \quad y \quad z]^T\ (m)$	**Pontos de Orientação** $[\alpha \quad \beta \quad \gamma]^T\ (rad)$
1	$[0{,}426 \quad -0{,}372 \quad 0{,}165]^T$	$[-0{,}694 \quad 1{,}203 \quad 0{,}119]^T$
2	$[0{,}462 \quad 0{,}402 \quad 0{,}0639]^T$	$[0{,}710 \quad 0{,}897 \quad 0{,}0326]^T$
3	$[-0{,}625 \quad 0{,}369 \quad 0{,}398]^T$	$[2{,}613 \quad 0{,}249 \quad 0{,}0125]^T$
4	$[-0{,}442 \quad -0{,}560 \quad 0{,}439]^T$	$[4{,}083 \quad 0{,}341 \quad 0{,}0224]^T$

Fonte: autor.

De maneira análoga à Tarefa 1, as condições iniciais e finais alimentam o Algoritmo de Planejamento de Trajetórias (Seção 4.3.) em pares para geração das curvas de posição, $P(t)$, e orientação, $\phi(t)$. A Tab. 7 indica as condições iniciais e finais de cada trajetória.

Tabela 7: Condições iniciais e finais de posição e orientação de cada trajetória da Tarefa 1.

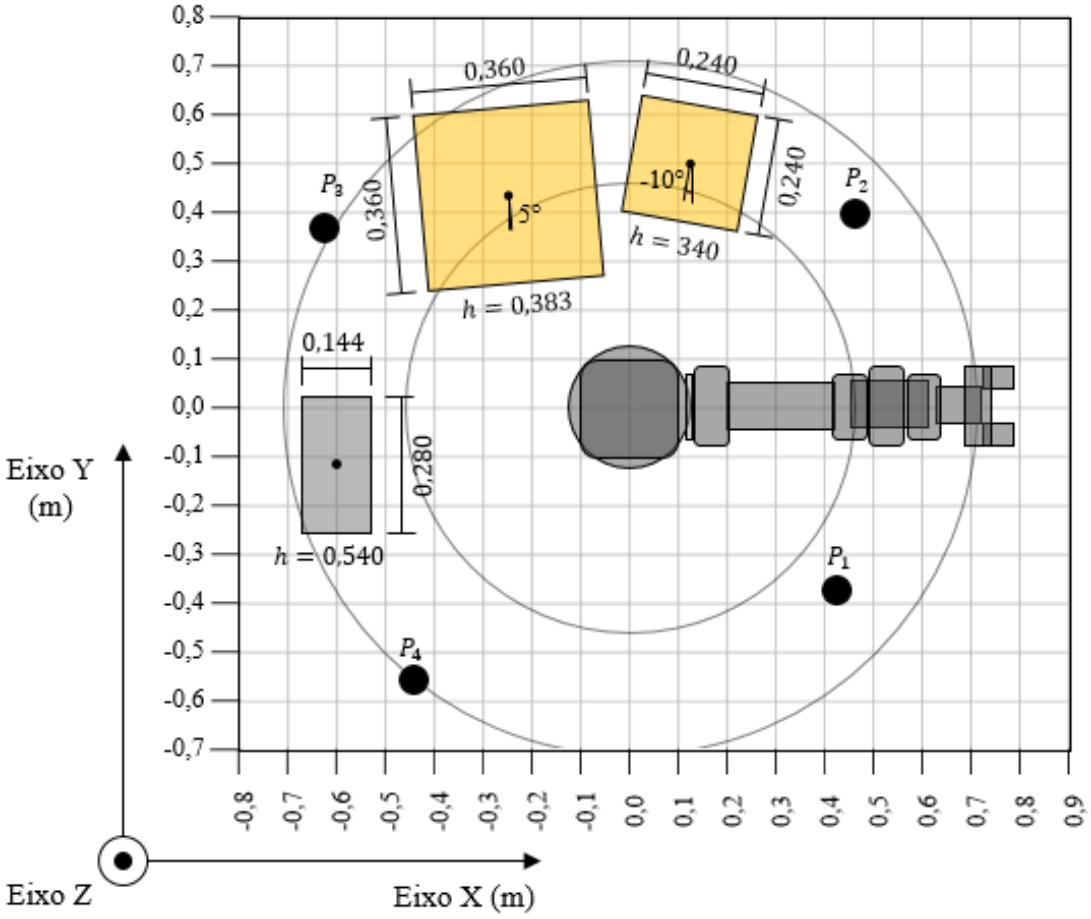
Trajetória	**1**	**2**	**3**
Condições Iniciais	P_1, ϕ_1	P_2, ϕ_2	P_3, ϕ_3
Condições Finais	P_2, ϕ_2	P_3, ϕ_3	P_4, ϕ_4

Fonte: autor.

A Fig. 26 é a vista superior do ambiente da tarefa e exibe as dimensões e posições dos obstáculos bem como os pontos que o efetuador final deve percorrer. Nela, os pontos P_i com $i = \{1, \dots, 4\}$ representam as posições iniciais e finais que constroem as trajetórias, apresentadas na Tab. 6, e o eixo z está na direção normal do plano da página. O manipulador inicia o movimento do ponto P_1, se desloca até o ponto P_2, depois até o ponto P_3 e assim sucessivamente. As coordenadas dos pontos foram escolhidas de maneira que o manipulador

não ficasse em condições de singularidade, conforme Apêndice B. Analogamente à primeira tarefa, a lei de tempo foi concebida pela curva *B-Spline* e o intervalo de tempo estabelecido para cada trajetória foi de 2 segundos.

Figura 26: Vista superior do ambiente da Tarefa 2.



Fonte: autor.

Como pode ser visto na Fig. 26, durante a segunda trajetória, entre os pontos P_2 e P_3, o manipulador deve desviar de 2 prismas retangulares retos por meio de movimentos realizados pelo efetuador final no eixo Z. Além disso, ele deve evitar invadir os limites internos do manipulador, modificando a trajetória nos eixos X e Y. Na terceira trajetória, entre os pontos P_3 e P_4, também há um obstáculo que deve ser evitado, o que também é feito por movimentos do

efetuador final no eixo Z. Ademais, o manipulador não deve passar pelo ponto P_3, o qual está fora de seus limites externos, a trajetória deve ser modificada nos eixos X e Y.

Em relação aos limites do espaço trabalho do robô, as esferas externa e interna, nessa tarefa, possuem raios de $0{,}71\ m$ e $0{,}46\ m$, respectivamente. Ademais, os raios dos elipsoides dos obstáculos foram calculados como funções das dimensões utilizando as equações em (48), onde as constantes dessas equações foram determinadas empiricamente e possuiam os seguintes valores: $k_x = 2{,}0$, $k_y = 2{,}0$ e $k_z = 1{,}414$.

- **Tarefa 3: Desenho sobre mesa**

A última tarefa foi planejada com o intuito de simular a execução de tarefas relacionadas à escrita e à pintura, bem como validar o algoritmo de desvio de obstáculos da Seção 4.3.. Ela foi subdividida em 5 trajetórias e, nela, o manipulador desvia de dois obstáculos e desenha sobre uma mesa inclinada. As coordenadas dos pontos de posição e orientação utilizados são expressadas na Tab. 8, as quais foram escolhidas de maneira que o manipulador não ficasse em condições de singularidade, conforme o Apêndice B.

Tabela 8: Coordenadas dos pontos de posição e orientação da Tarefa 3.

Numeração	**Pontos de Posição** $[x \quad y \quad z]^T\ (m)$	**Pontos de Orientação** $[\alpha \quad \beta \quad \gamma]^T\ (rad)$
1	$[-0{,}548 \quad -0{,}460 \quad 0{,}482]^T$	$[0{,}360 \quad 0{,}0637 \quad 0{,}0726]^T$
2	$[-0{,}359 \quad 0{,}484 \quad 0{,}458]^T$	$[2{,}280 \quad 0{,}235 \quad 0{,}330]^T$
3	$[-0{,}246 \quad 0{,}556 \quad 0{,}389]^T$	$[1{,}571 \quad 0{,}524 \quad 0{,}000]^T$
4	$[0{,}282 \quad 0{,}493 \quad 0{,}279]^T$	$[1{,}571 \quad 0{,}524 \quad 0{,}000]^T$
5	$[0{,}429 \quad 0{,}447 \quad 0{,}294]^T$	$[0{,}758 \quad 0{,}857 \quad -0{,}142]^T$
6	$[0{,}342 \quad -0{,}486 \quad 0{,}240]^T$	$[-0{,}939 \quad 0{,}911 \quad 0{,}148]^T$

Fonte: autor.

Empregou-se diferentes métodos na formulação das trajetórias com o intuito de explorar as vantagens de cada método para as situações disponíveis nessa tarefa. As condições iniciais e finais utilizadas nas trajetórias são indicadas na Tab. 9.

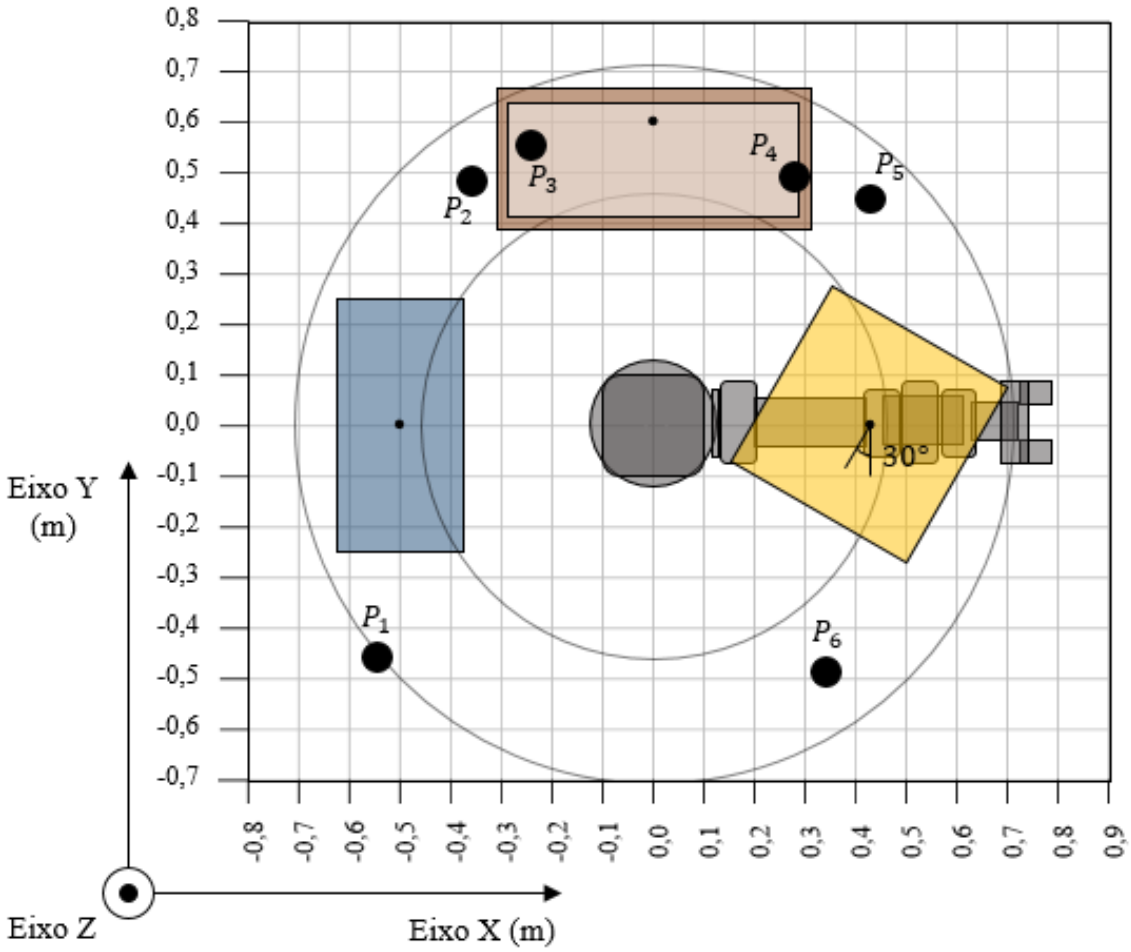
Tabela 9: Condições iniciais e finais de posição e orientação das trajetórias 1 e 5 da Tarefa 3.

Trajetória	**1**	**2**	**3**	**4**	**5**
Condições Iniciais	P_1, ϕ_1	P_2, ϕ_2	P_3, ϕ_3	P_4, ϕ_4	P_5, ϕ_5
Condições Finais	P_2, ϕ_2	P_3, ϕ_3	P_4, ϕ_4	P_5, ϕ_5	P_6, ϕ_6

Fonte: autor.

A vista superior do ambiente da Tarefa 3 é apresentada na Fig. 27.

Figura 27: Vista superior do ambiente da Tarefa 3.



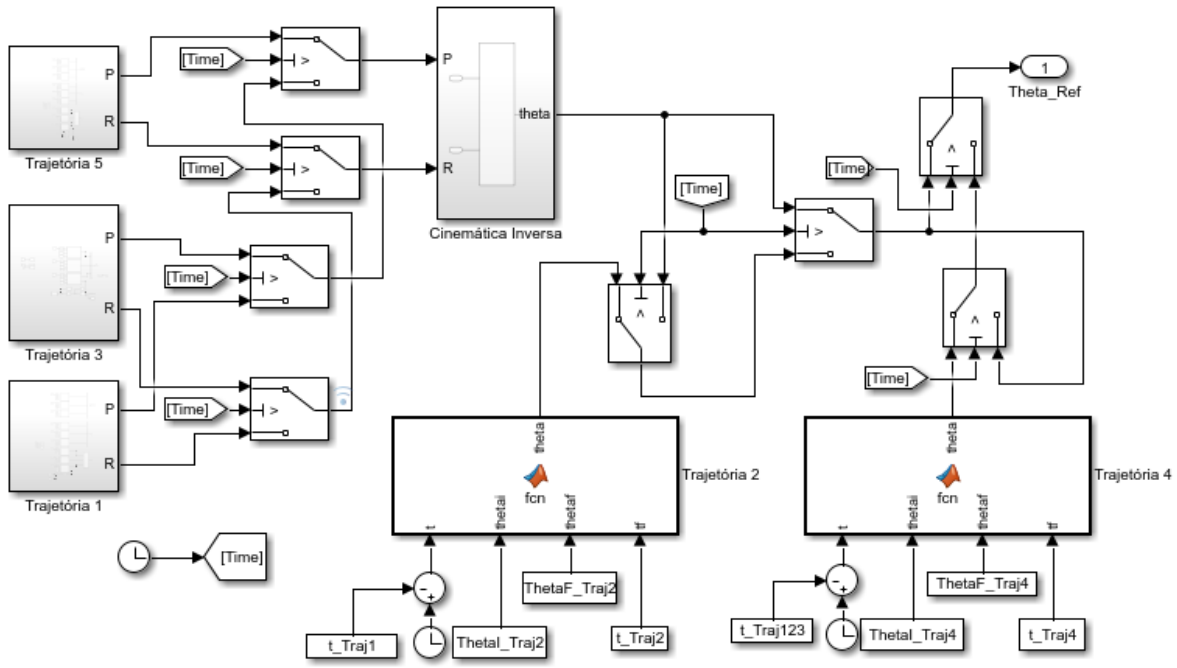
Fonte: autor.

Na Fig. 27, os pontos P_i com $i = \{1, \dots, 6\}$ denotam as posições iniciais e finais que constroem as trajetórias, apresentadas na Tab. 8, e o eixo Z está na direção normal do plano da página. Cada trajetória é construída entre dois pontos consecutivos, por exemplo, os pontos P_1 e P_2 constroem a trajetória 1, os pontos P_2 e P_3 constroem a trajetória 2 e assim por diante. O manipulador inicia seu movimento do ponto P_1: o efetuador final passa sobre o obstáculo em azul, realizando modificações na coordenada Z, e se desloca em direção à mesa. Nela, realiza o processo de escrita e, após isso, desloca-se em direção ao ponto P_5. Em seguida, desvia do obstáculo em amarelo, também realizando alterações no eixo Z, e finaliza o trajeto no ponto P_6.

A Fig. 28 indica também as dimensões dos obstáculos bem como as posições e as orientações de cada objeto presente na tarefa, incluindo a mesa. Os elipsoides dos obstáculos foram estimados com os mesmos valores das constantes da segunda tarefa, utilizando as equações em (48).

Para o acionamento das trajetórias nos momentos corretos, implementou-se uma lógica no Simulink® em forma de diagrama de blocos, exibido na Fig. 28.

Figura 28: Diagrama de blocos da Lógica de Acionamento das Trajetórias da Tarefa 3.



Fonte: autor.

Conforme a Fig. 28, as trajetórias 1, 3 e 5, foram formuladas no espaço cartesiano e passam por um bloco de Cinemática Inversa, onde são convertidas em posições angulares das juntas, enquanto as trajetórias 2 e 4 não passam, uma vez que foram planejadas no espaço de configuração do robô.

Na formulação das Trajetórias 1 e 5, utilizou-se o algoritmo de planejamento de trajetórias (Seção 4.3.), posto que o manipulador, ao executar essas trajetórias, deve desviar de obstáculos. O intervalo de tempo considerado para a execução dessas trajetórias foi de 2 segundos.

As Trajetórias 2 e 4 foram planejadas no espaço das juntas para evitar problemas que iriam ocorrer durante o cálculo da Cinemática Inversa caso essas trajetórias fossem planejadas em espaço cartesiano. Assim sendo, as condições iniciais e finais dessas trajetórias foram convertidas em ângulos das juntas. O intervalo de tempo considerado nessas trajetórias foi de 1 segundo. Além disso, foram utilizados polinômios de 7ª ordem na implementação delas, uma vez que polinômios com essa ordem possuem coeficientes suficientes para assegurar velocidades, acelerações e arranques nulos nas extremidades das trajetórias. As equações matemáticas relacionadas aos polinômios são apresentadas no Apêndice D.

A Trajetória 3 contempla a escrita na mesa, inclinada em $\pi/3\ rad$. Nela, a orientação do robô foi mantida constante enquanto ele desenha, é perpendicular a mesa e foi calculada com base em matrizes de rotação relacionadas ao seu ângulo. Um desenho esquemático da mesa é apresentado na seção de resultados, Seção 5.2.. O processo de escrita foi feito em termos de pontos em um plano bidimensional, os quais foram transladados utilizando matrizes de rotação relacionadas à inclinação da mesa. As posições e a orientação do efetuador final consideradas na escrita são indicadas em (54).

$$\phi = [1{,}571 \quad 0{,}524 \quad 0{,}000]^T$$

$$\begin{aligned} P_x = [&-0{,}2459;\ -0{,}2459;\ -0{,}2459;\ -0{,}2622;\ -0{,}2773;\ -0{,}2564;\ -0{,}2320;\\ &-0{,}2076;\ -0{,}1878;\ -0{,}1681;\ -0{,}1499;-0{,}1263;\ -0{,}1263;\ -0{,}1495;\\ &-0{,}1797;\ -0{,}2297;\ -0{,}2548;\ -0{,}2781;\ -0{,}2835;\ -0{,}2617;\ -0{,}2204;\\ &-0{,}1783;\ -0{,}1264;\ -0{,}1077;\ -0{,}1042;\ -0{,}0880;\ -0{,}0539;\ -0{,}0264;\\ &-0{,}0252;\ -0{,}0183;\ 0{,}0116;\ 0{,}0456;\ 0{,}0607;\ 0{,}0781;\ 0{,}0723;\ 0{,}0526;\\ &0{,}0444;\ 0{,}0514;\ 0{,}0886;\ 0{,}1287;\ 0{,}1385;\ 0{,}1188;\ 0{,}0981;\ 0{,}0829;\\ &0{,}0941;\ 0{,}1118;\ 0{,}1747;\ 0{,}1641;\ 0{,}1658;\ 0{,}1906;\ 0{,}2401;\ 0{,}2641;\\ &0{,}2674;\ 0{,}2430;\ 0{,}2000;\ 0{,}1795;\ 0{,}2210;\ 0{,}2561;\ 0{,}2697;\ 0{,}2769;\\ &0{,}2822;\ 0{,}2822;\ 0{,}2822] \end{aligned} \tag{54}$$

$$P_y = [0{,}5564;\ 0{,}5564;\ 0{,}5564;\ 0{,}5616;\ 0{,}5675;\ 0{,}5750;\ 0{,}5738;\ 0{,}5709;\ 0{,}5646;\ 0{,}5599;\ 0{,}5545;\ 0{,}5611;\ 0{,}5715;\ 0{,}5715;\ 0{,}5465;\ 0{,}5175;\ 0{,}4944;\ 0{,}4985;\ 0{,}5095;\ 0{,}5183;\ 0{,}4960;\ 0{,}4917;\ 0{,}4956;\ 0{,}5164;\ 0{,}5303;\ 0{,}4960;\ 0{,}4944;\ 0{,}5082;\ 0{,}5344;\ 0{,}4965;\ 0{,}4908;\ 0{,}4908;\ 0{,}5088;\ 0{,}5599;\ 0{,}5692;\ 0{,}5721;\ 0{,}5611;\ 0{,}5280;\ 0{,}4926;\ 0{,}4936;\ 0{,}5222;\ 0{,}5355;\ 0{,}5344;\ 0{,}5280;\ 0{,}5228;\ 0{,}5222;\ 0{,}5312;\ 0{,}5204;\ 0{,}5050;\ 0{,}4922;\ 0{,}4922;\ 0{,}5037;\ 0{,}5268;\ 0{,}5373;\ 0{,}5390;\ 0{,}5328;\ 0{,}5392;\ 0{,}5324;\ 0{,}5160;\ 0{,}5021;\ 0{,}4933;\ 0{,}4933;\ 0{,}4933]$$

$$P_z = [0{,}3885;\ 0{,}3885;\ 0{,}3885;\ 0{,}3976;\ 0{,}4077;\ 0{,}4208;\ 0{,}4187;\ 0{,}4137;\ 0{,}4027;\ 0{,}3946;\ 0{,}3853;\ 0{,}3966;\ 0{,}4147;\ 0{,}4147;\ 0{,}3714;\ 0{,}3212;\ 0{,}2812;\ 0{,}2883;\ 0{,}3072;\ 0{,}3226;\ 0{,}2839;\ 0{,}2764;\ 0{,}2832;\ 0{,}3192;\ 0{,}3433;\ 0{,}2839;\ 0{,}2812;\ 0{,}3051;\ 0{,}3504;\ 0{,}2847;\ 0{,}2749;\ 0{,}2749;\ 0{,}3061;\ 0{,}3946;\ 0{,}4106;\ 0{,}4157;\ 0{,}3966;\ 0{,}3393;\ 0{,}2780;\ 0{,}2797;\ 0{,}3292;\ 0{,}3523;\ 0{,}3504;\ 0{,}3393;\ 0{,}3303;\ 0{,}3292;\ 0{,}3448;\ 0{,}3262;\ 0{,}2995;\ 0{,}2773;\ 0{,}2773;\ 0{,}2972;\ 0{,}3373;\ 0{,}3554;\ 0{,}3583;\ 0{,}3476;\ 0{,}3587;\ 0{,}3469;\ 0{,}3186;\ 0{,}2944;\ 0{,}2792;\ 0{,}2792;\ 0{,}2792]$$

As restrições temporais dessa trajetória foram determinadas utilizando a curva *B-Spline* cúbica. Primeiramente, os pontos de controle que modelam as curvas foram calculado utilizando a Eq. (44) e, em seguida, tomando-os 4 a 4, determinou-se curvas "locais" por meio da Eq. (41). As curvas locais, quando unidas, constroem a Trajetória 3. O intervalo de tempo dessa trajetória foi de 10 segundos.

5.2. Resultados e Discussões

Primeiramente, são apresentados os parâmetros helicoidais utilizados para resolver o problema Cinemática Direta, por meio do método dos helicoides sucessivos, expostos na Tab. 10.

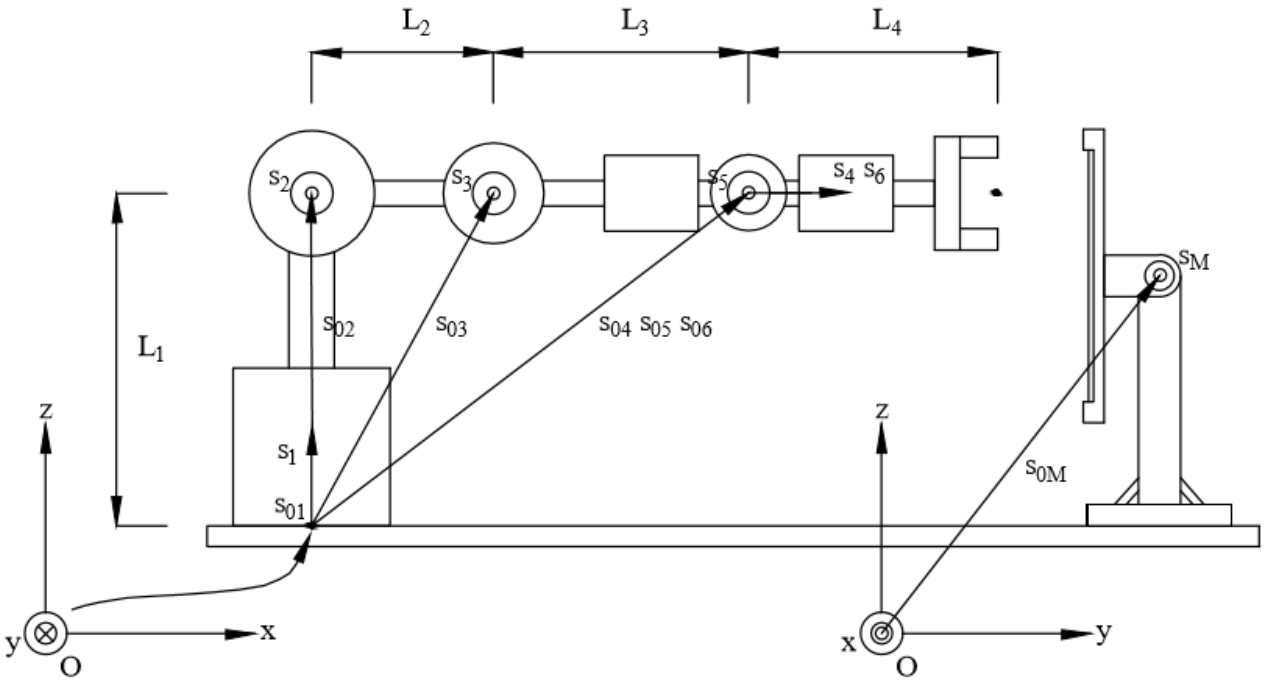
Tabela 10: Parâmetros helicoidais.

Junta	$\boldsymbol{s}$	$\boldsymbol{s_0}$	$\boldsymbol{t}$	$\boldsymbol{\theta}$
1	$[0 \quad 0 \quad 1]^T$	$[0 \quad 0 \quad 0]^T$	0	θ_1
2	$[0 \quad -1 \quad 0]^T$	$[0 \quad 0 \quad L_1]^T$	0	θ_2
3	$[0 \quad -1 \quad 0]^T$	$[L_2 \quad 0 \quad L_1]^T$	0	θ_3
4	$[1 \quad 0 \quad 0]^T$	$[L_2 + L_3 \quad 0 \quad L_1]^T$	0	θ_4
5	$[0 \quad -1 \quad 0]^T$	$[L_2 + L_3 \quad 0 \quad L_1]^T$	0	θ_5
6	$[1 \quad 0 \quad 0]^T$	$[L_2 + L_3 \quad 0 \quad L_1]^T$	0	θ_6
Mesa	$[1 \quad 0 \quad 0]^T$	$[L_{Mx} \quad L_{My} \quad L_{Mz}]^T$	0	θ_M

Fonte: autor.

Esses parâmetros são utilizados na obtenção das matrizes de transformação para cada uma das juntas e para a mesa de escrita. A Fig. 29 ilustra os eixos helicoidais considerados no manipulador.

Figura 29: Eixos helicoidais adotados no manipulador.



Fonte: autor.

É importante salientar que os referenciais da mesa e do manipulador se coincidem, mesmo eles estando ilustrados de uma maneira distinta na Fig. 29. O deslocamento do efetuador final do manipulador em relação ao referencial é resultante dos deslocamentos helicoidais sucessivos dos eixos das juntas. O primeiro eixo de junta, s_1, aponta verticalmente na direção positiva do eixo Z; o segundo, o terceiro e o quinto eixos de junta, s_2, s_3 e s_5, apontam no plano da página, o que seria na direção negativa do eixo Y; o quarto e o sexto eixos de juntas, s_4 e s_6, apontam horizontalmente na direção positiva do eixo X, assim como o eixo da mesa, s_M.

Utilizando a fórmula de Rodrigues, as matrizes de transformação foram calculadas utilizando os parâmetros helicoidais. Dessa forma, o problema de Cinemátiva Inversa foi resolvido conforme o Algoritmo do Apêndice B e o modelo geométrico do manipulador foi construído no Simscape™. A seguir serão explicitados outros resultados e discussões, que são

divididos em Identificação e Controle do robô, Seção 5.2.1., e Planejamento de Trajetórias e Tarefas, Seção 5.2.2..

5.2.1. Identificação e Controle dos Parâmetros das Juntas

Na Seção 5.1.1. foram apresentados métodos para identificação e controle dos parâmetros das juntas. Essa seção tem como objetivo apresentar os resultados obtidos e discutí-los. A Fig. 30 exibe os sinais de saída (ângulos das juntas) após as perturbações realizadas no modelo geométrico (Simscape^{TM}) e nas funções de transferência estimadas.

Conforme pode ser observado nessa figura, as aproximações realizadas foram bem fidedignas em relação ao modelo do Simscape^{TM}. Somente as dinâmicas das juntas 2 e 3 exibiram diferenças em relação ao modelo do Simscape^{TM}, mas nada tão significativo. Em relação ao perfil das respostas, as juntas 1, 2 e 3 apresentaram oscilação e sobrepassagem, porém, em aproximadamente 0,5 segundos, a variação das respostas torna-se pouco significativa. Em geral, as funções de transferência foram bem estimadas e, dessa forma, podem ser utilizadas no projeto do sistema de controle.

Os parâmetros do sistema de controle obtidos após sintonia utilizando o Algoritmo Genético são indicados na Tab. 11.

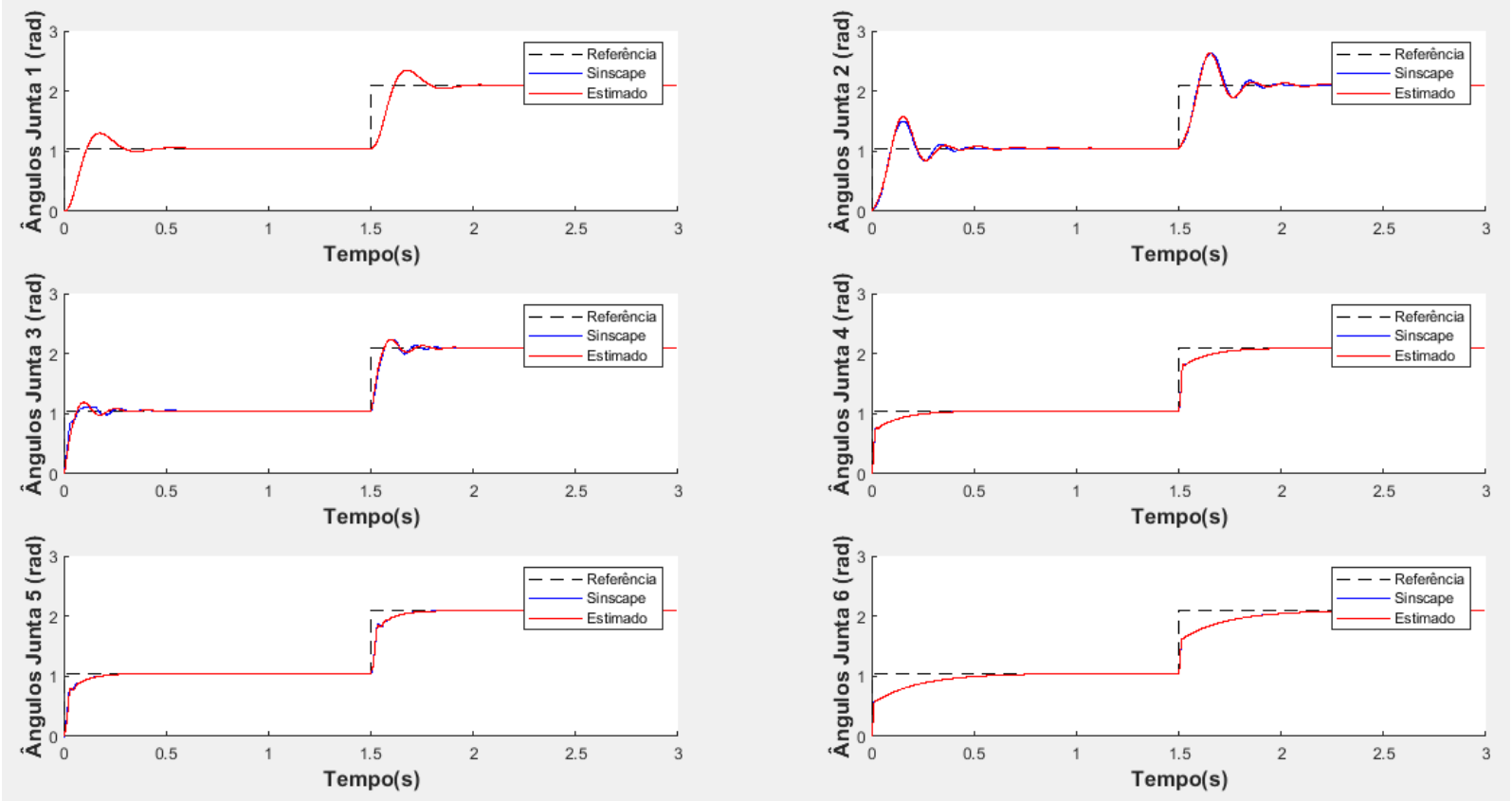
Tabela 11: Parâmetros dos sistemas de controle PD com filtro de cada junta obtidos por meio de sintonia realizada em Algoritmo Genético.

Junta	**1**	**2**	**3**	**4**	**5**	**6**
K_P	1056,2	10062	49987	2488,2	4977,8	2226,8
K_D	97,7	288,0	480,0	66,7	17,6	202,3
τ_F	96,1	399,0	288,0	9,8	12,0	8,8

Fonte: autor.

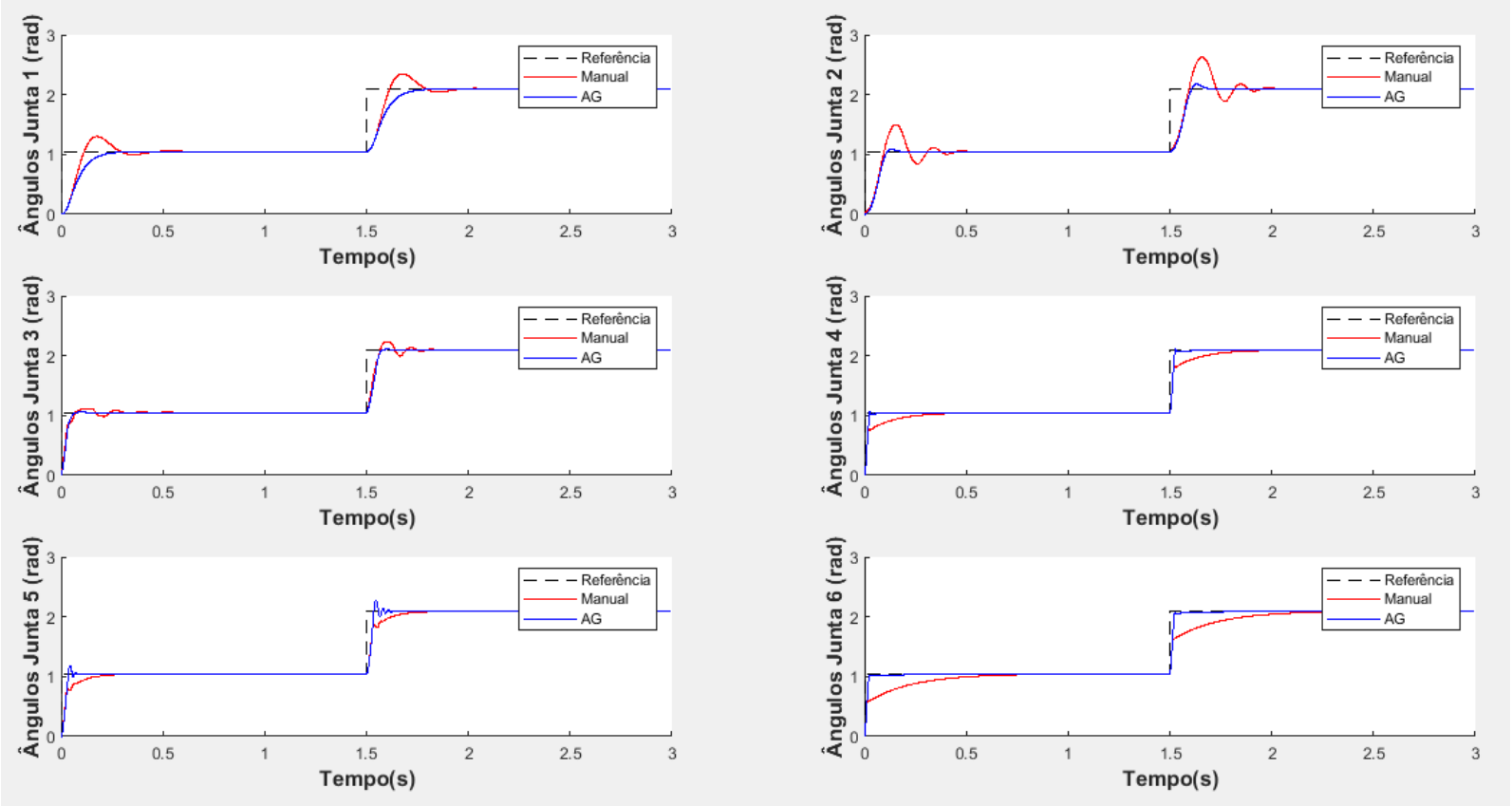
Os gráficos das respostas dos sistemas de controle, para cada uma das juntas, são indicados na Fig. 31.

Figura 30: Respostas das Funções de Transferência estimadas em comparação à resposta do modelo do robô no Simscape™.



Fonte: autor.

Figura 31: Comparação dos desempenhos dos controladores inicial e sintonizado via AG.



Fonte: autor.

Para fins de comparação de desempenho dos sistemas de controle projetados, o inicial e o sintonizado pelo AG, calculou-se os valores dos índices ITAE após aplicações de perturbações do tipo degrau em todas as juntas. É importante lembrar que o índice ITAE quantifica o produto da integral do erro com o tempo. Isso significa que quanto menor é o valor desse indice, melhor é o desempenho do controlador, pois, por conseguinte, os erros em regime permanente, a sobrepassagem do sinal saída e o tempo de resposta são minimizados. Os valores obtidos do indice ITAE após perturbações são indicados na Tab. 12.

Tabela 12: Índices ITAE obtidos após perturbações do tipo degrau em cada uma das juntas.

Junta	**1**	**2**	**3**	**4**	**5**	**6**
Controle Inicial	0,569	2,654	1,359	1,676	2,312	4,024
Controle Sintonizado (AG)	0,496	1,826	0,642	0,477	0,954	0,688

Fonte: autor.

Quando se compara o desempenho dos controladores em termos do índice, o segundo não apresenta melhoras consideráveis para a primeira junta, pois os valores obtidos foram próximos, 0,569 e 0,496. Entretanto, quando se avalia a resposta graficamente, por meio da Fig. 31, nota-se que o sinal de saída dessa junta para o sistema de controle inicial apresenta sobressinal, o que não ocorre no controle sintonizado. Isso mostra que apesar do índice ITAE apresentar valores próximos para respostas distintas, não significa que os perfis serão parecidos.

Entretanto, no restante das juntas, as diferenças entre os indices ITAE são maiores e, por isso, percebe-se uma melhora considerável em suas respostas. Um exemplo é a junta 6, na qual o ITAE diminuiu de 4,024 para 0,688, remetendo em grande melhora em relação à sua velocidade de resposta.

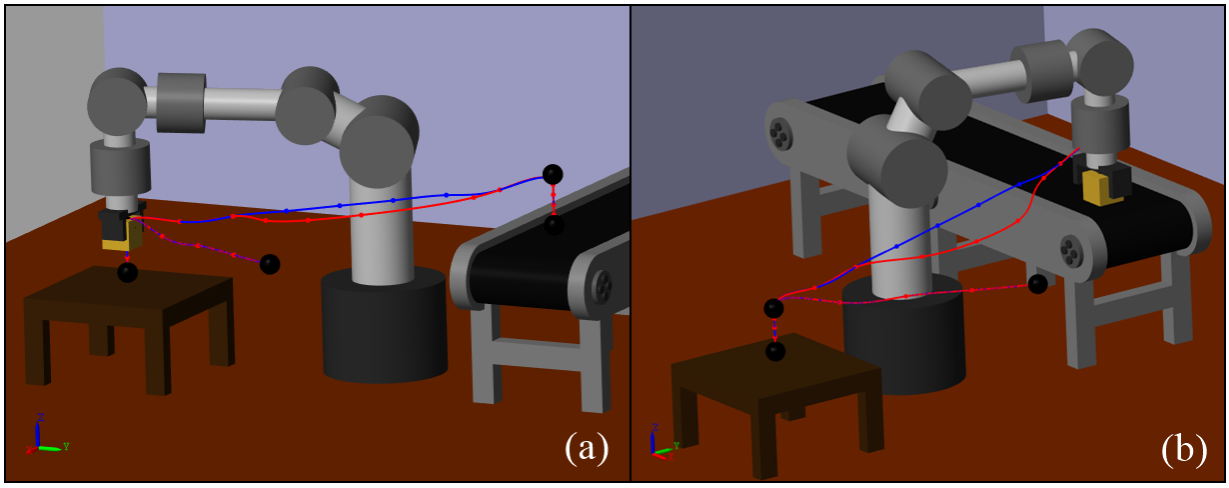
Em geral, os desempenhos dos sistemas de controle projetados foram bons devido à estratégia empregada, PD com filtro, o qual funciona como um compensador avanço de fase, que apesar de ficar sujeito a ruídos de alta frequência, é útil para melhorar a margem de estabilidade, a velocidade de resposta e diminuir o sobressinal. Os bons desempenhos são ratificados pelos baixos valores do índice ITAE obtidos e em termos de critérios de desempenho, posto que as saídas possuem baixos tempos de resposta e sobressinal.

Todavia, é possível projetar sistemas de controle com melhores desempenhos. Um exemplo seria adotar uma estratégia de controle do tipo MIMO, uma vez que os robôs são sistemas não-lineares, onde as dinâmicas das juntas são dependentes umas das outras. Isso é um problema pois quando se estima uma função de transferência de uma junta, assim que o manipulador se configura de uma maneira distinta, a função de transferência inicial já não representa de maneira fidedigna a dinâmica daquela junta devido à interconexão entre elas e o acoplamento dinâmico. Dessa forma, um controle centralizado agiria nas juntas considerando uma informação completa do estado do sistema, permitindo considerar o problema globalmente. Isso, por outro lado, ocasiona complexidade muito maior na síntese do controlador.

5.2.2. Planejamento de Trajetórias e Tarefas

Essa seção tem como objetivo apresentar os resultados obtidos da Seção 5.1.2. e discutí-los. A primeira tarefa planejada é a mais simples de todas, pois não há a presença de obstáculos. A trajetória retilínea é modificada somente quando há invasão do efetuador final na esfera interna do manipulador, que limita seu espaço de trabalho. Por isso, na Fig. 32, existem duas trajetórias: uma que o robô passaria se não houvessem as restrições, trajetória em azul, e outra em que ele sofre desvios, trajetória em vermelho.

Figura 32: Tarefa 1, (a) Manipulador pegando caixa sobre a mesa e (b) colocando-a sobre esteira transportadora.



Fonte: autor.

A diferença entre elas acontece na região em que elas estão mais próximas do manipulador: a trajetória que sofreu modificações possui um formato mais curvo, devido as modificações da esfera interna, enquanto a não sofreu alterações, tem o perfil retilíneo.

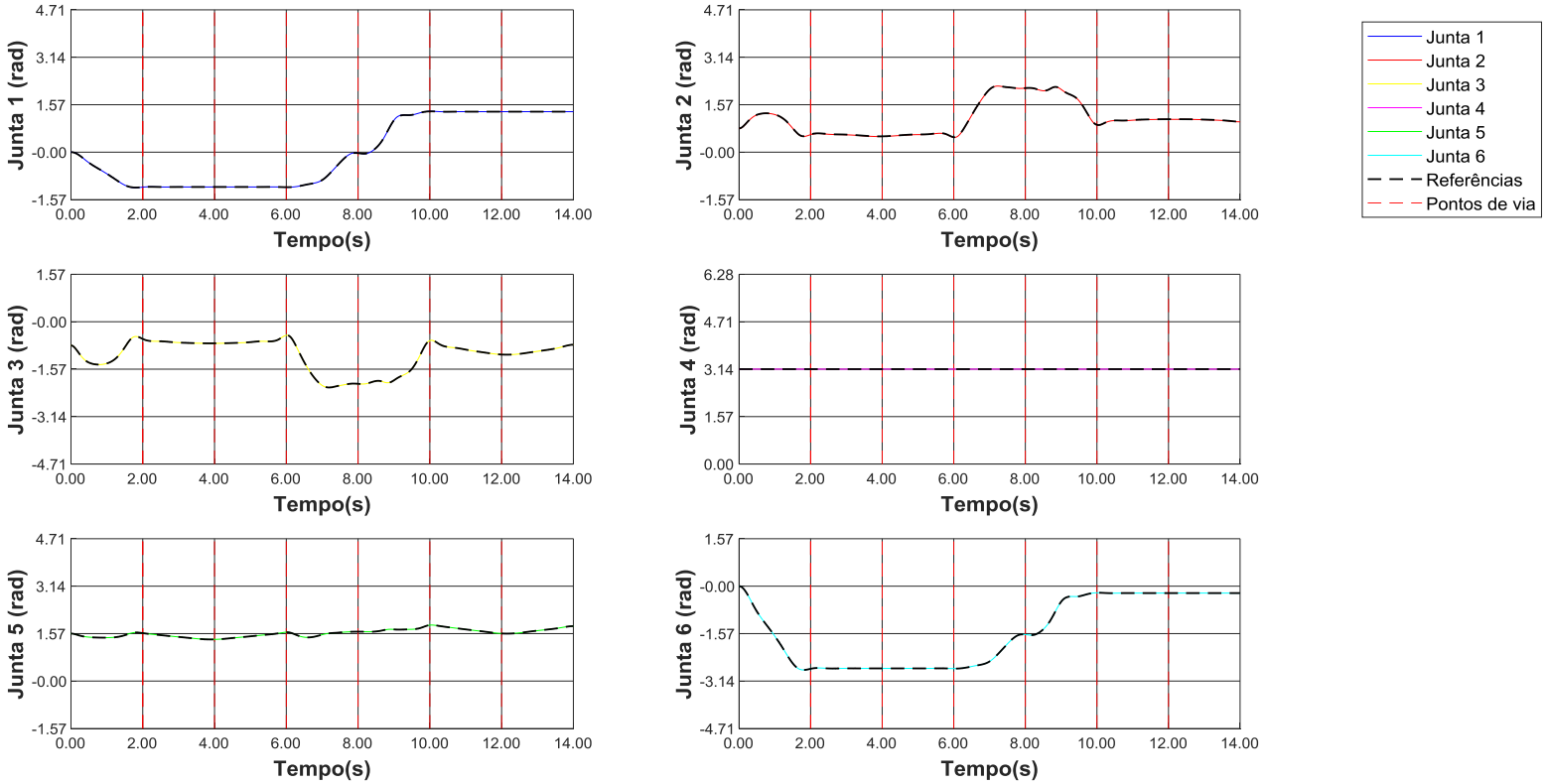
Na Fig. 33 pode-se ver as variações das posições das juntas ao longo da execução da tarefa e na Fig. 34 as posições do efetuador final em coordenadas cartesianas. O momento em que o manipulador evita a esfera interna do robô acontece entre 7 a 9 segundos, como pode ser visto na Fig. 33. Nessas situações, o caminho é ajustado modificando as coordenadas X e Y, conforme pode ser visto na Fig. 34.

Constatou-se que a modelagem das restrições mecânicas utilizando uma esfera não é a maneira mais adequada, visto que existem espaços da esfera que ainda poderiam ser alcançados pelo efetuador sem causar impactos entre os membros. Por exemplo, se o manipulador se posicionar de uma maneira comprimida, ele consegue atingir regiões que estão dentro da esfera. Dessa forma, recomenda-se, para melhor aproveitamento dos espaços, um estudo mais minucioso das restrições mecânicas do manipulador e também a realização da análise de destreza (Mondragon, 2013), que visa identificar orientações executáveis pelo efetuador final. Com ela, o robô poderia rearranjar-se de mais formas e, assim, atingir mais posições, aumentando o seu limite de trabalho.

Outro problema foi a falta de liberdade nas escolhas das posições e orientações iniciais e finais para o planejamento das trajetórias. Elas eram realizadas baseando-se nos problemas de singularidade do manipulador (Apêndice B). Para contornar esse problema, evidencia-se uma necessidade de utilização de métodos complementares para evitar as singularidades. Alguns exemplos são a utilização do método dos Mínimos Quadrados Amortecidos ou planejar as trajetórias diretamente no espaço das juntas.

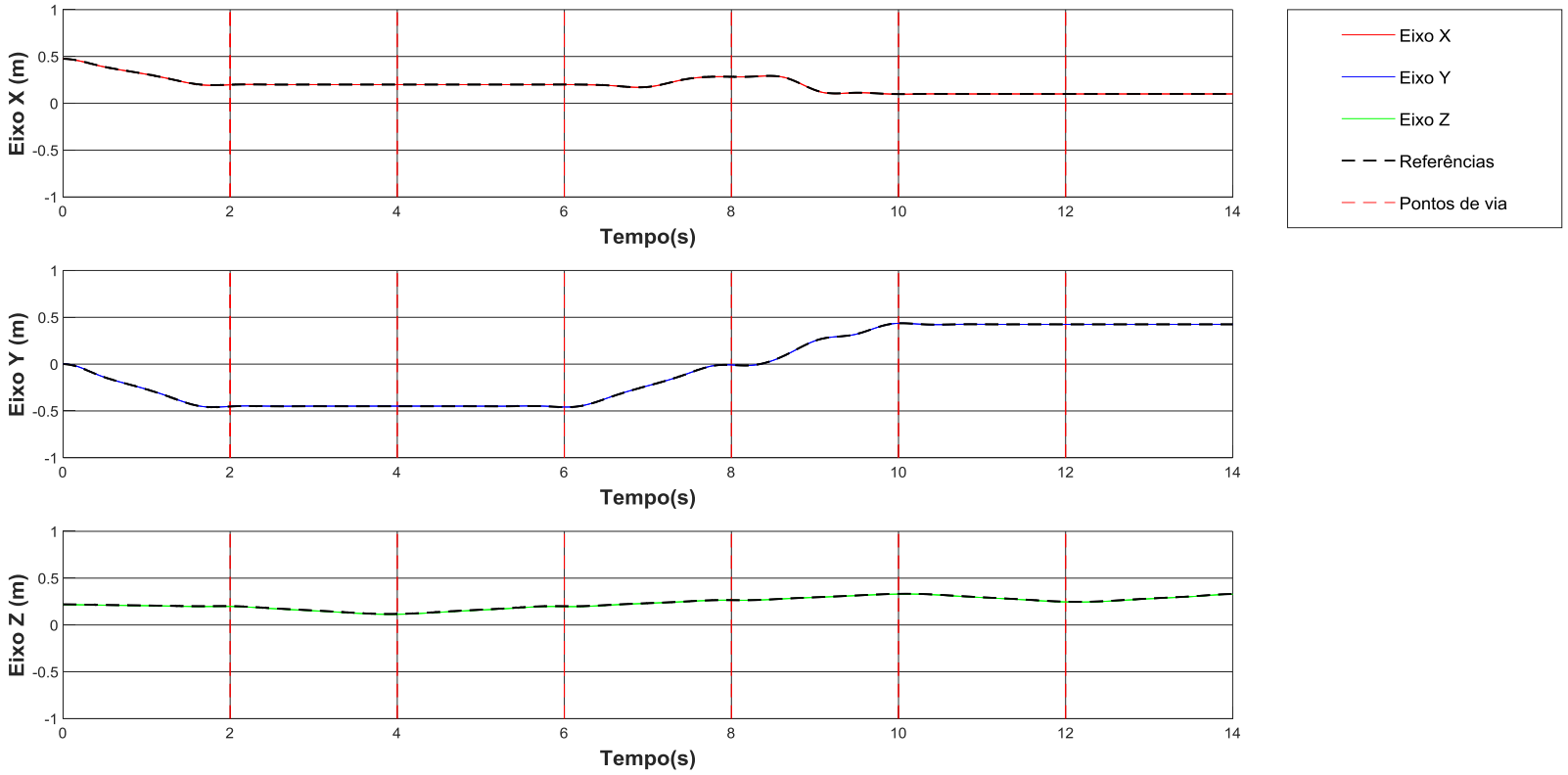
Entretanto, notou-se uma boa sinergia entre o uso da curva *B-Spline* e a lógica utilizada no Planejamento de Caminhos para o contorno da esfera interna e de obstáculos (próximas tarefas). Quando condições restritas acontece no caminho inicial gerado, os pontos que estão nessa rota são alterados. Esse fato possui harmonia com uso da *B-Spline*, uma vez que essa curva, da maneira em que foi formulada, possibilita adaptações locais. Dessa forma, a trajetória não é modificada em sua íntegra, como aconteceria utilizando outras curvas contínuas tais como de Bézier, de Hermite ou Polinomiais.

Figura 33: Ângulos das juntas do robô ao longo da Tarefa 1.



Fonte: autor.

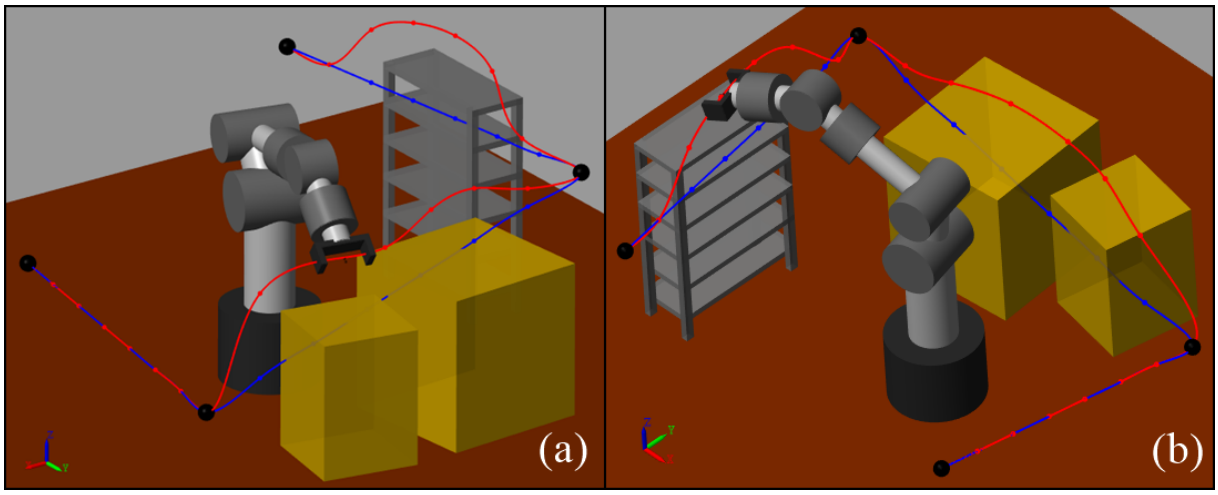
Figura 34: Posições do efetuador final ao longo da Tarefa 1.



Fonte: autor

Na Fig. 35 temos a representação da Tarefa 2, na qual o robô desvia de obstáculos composto por duas caixas e um armário. Na figura, existe a presença de duas trajetórias, uma que é a trajetória com segmento retilíneo que o manipulador realizaria caso não houvesse a presença dos obstáculos, representada em azul, e outra com um formato curvilíneo que considera os obstáculos encapsulados pelos elipsoides na rotina do algoritmo, representada em vermelho.

Figura 35: Tarefa 2, (a) Robô desviando das caixas e (b) da prateleira.

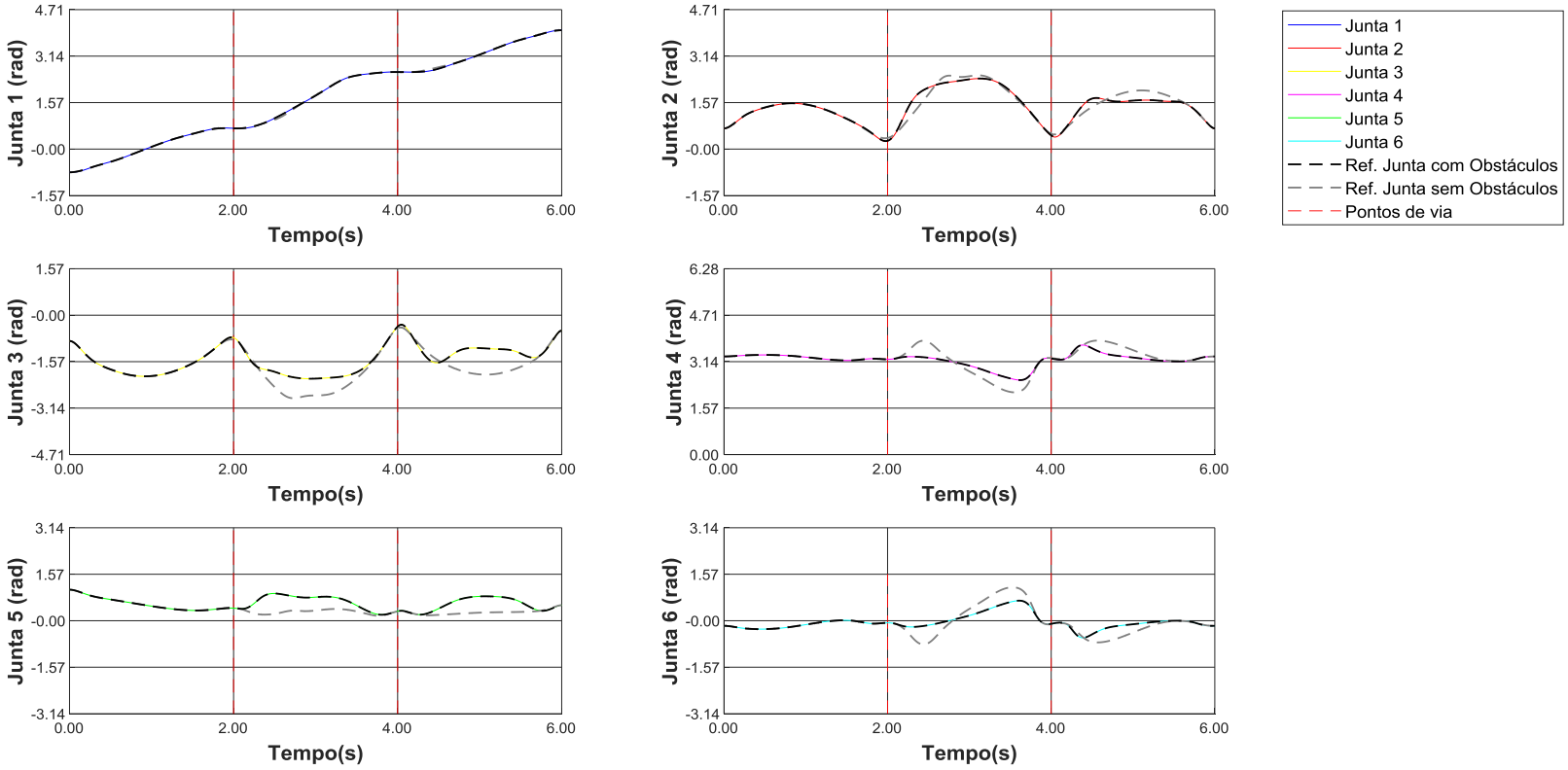


Fonte: autor.

Na Fig. 36, pode-se ver como os ângulos das juntas variaram durante a execução da tarefa, enquanto a Fig. 37 apresenta as posições do efetuador final e a Fig. 38 exibe as distâncias entre o efetuador e os obstáculos nesses mesmos instantes. Os momentos em que o robô começa a passar pelos obstáculos são aqueles em que os valores da posição do efetuador final no eixo Z modificam-se, Fig. 37. Os instantes em que o robô está mais próximo dos obstáculos são os pontos de mínimo das curvas de distância entre o efetuador e os obstáculos, Fig. 38. Nesses momentos, o robô passa sobre as arestas dos prismas retangulares.

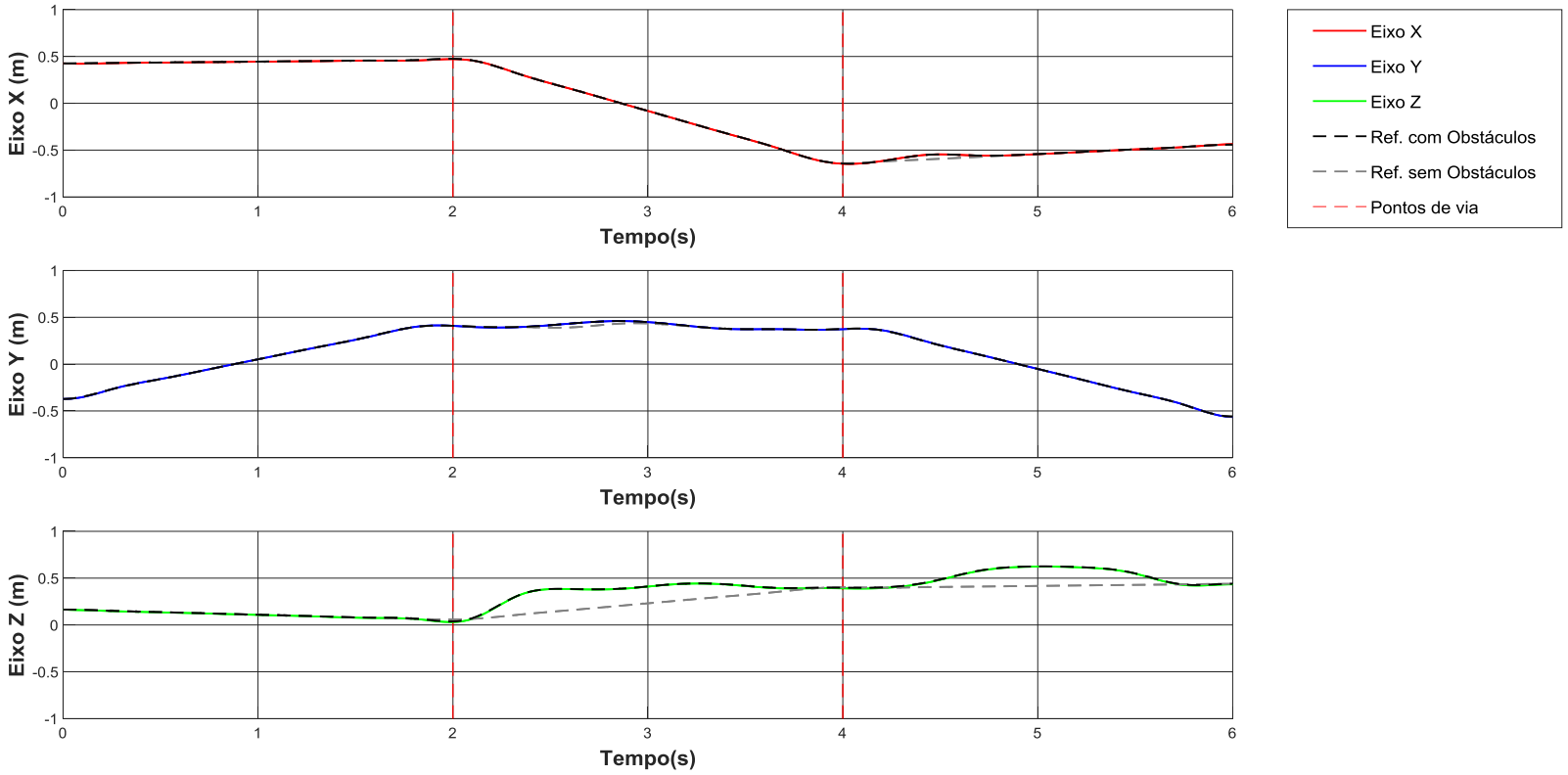
Percebeu-se que, analogamente ao problema da esfera interna, dependendo do formato do obstáculo, o encapsulamento por elipsoide pode não ser o mais adequado para sua modelagem, pois ele pode incluir em suas coordenadas espaços livres que não estão sendo necessariamente preenchidos por obstáculos e que poderiam ser aproveitados pelo manipulador nas trajetórias. Por exemplo, no prisma retangular reto, que foi utilizado para representar os obstáculos, o encapsulamento utilizando o elipsoide pode, no máximo, tangenciar as arestas do

Figura 36: Ângulos das juntas do robô ao longo da Tarefa 2.



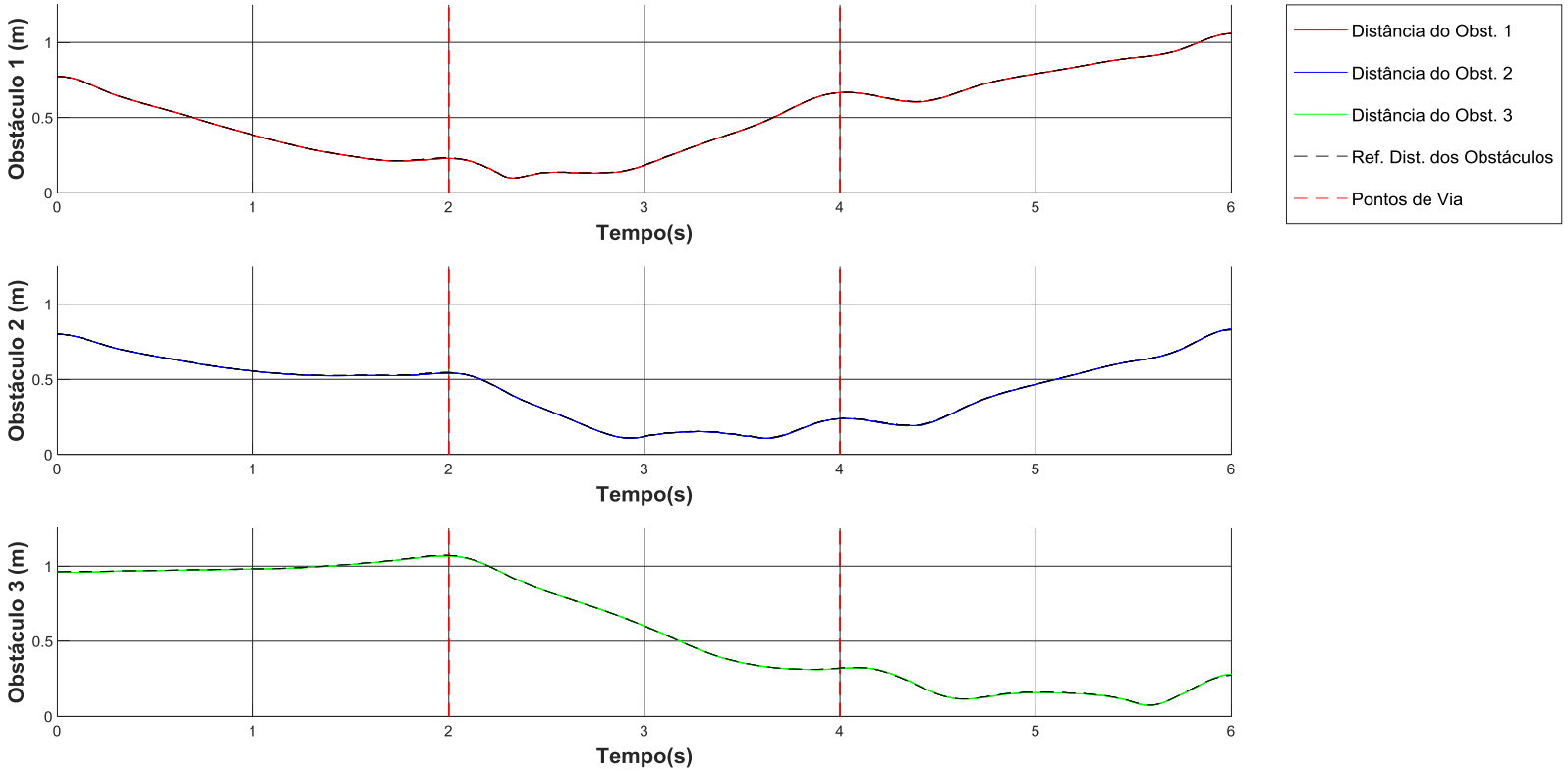
Fonte: autor.

Figura 37: Posição do efetuador final do robô ao longo da Tarefa 2.



Fonte: autor.

Figura 38: Distâncias entre o efetuador final e cada obstáculo.

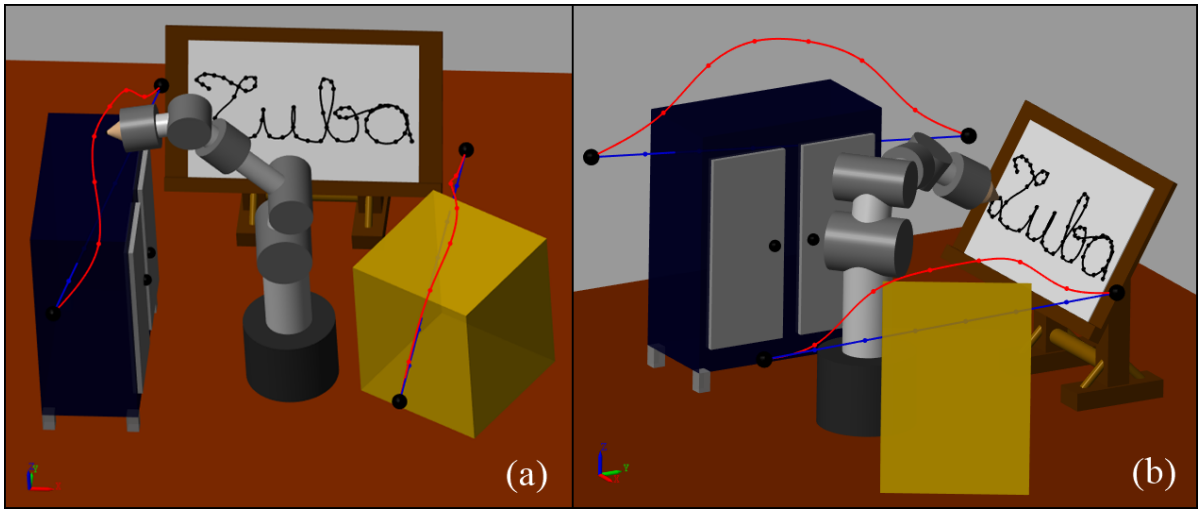


Fonte: autor.

prisma, entretanto, o efetuador final é incapaz de atingir as regiões que estão entre os centros das faces do prisma e as fronteiras dos elipsoides, as quais são factíveis. Em ambientes fechados e na presença de muitos obstáculos, o robô poderia ficar incapacitado de mover-se devido à grande diminuição de seu espaço de trabalho, mesmo tendo espaços possíveis.

Na Fig. 39 temos o manipulador executando a Tarefa 3, em momentos em que o robô escreve sobre uma mesa e desvia dos obstáculos. Nela, é visível a presença da trajetória retilínea, em azul, e da trajetória modificada para evitar colisões e restrições, em vermelho.

Figura 39: Tarefa 3, (a) Manipulador evitando colisão com armário industrial e (b) desenhando sobre uma mesa de desenho.

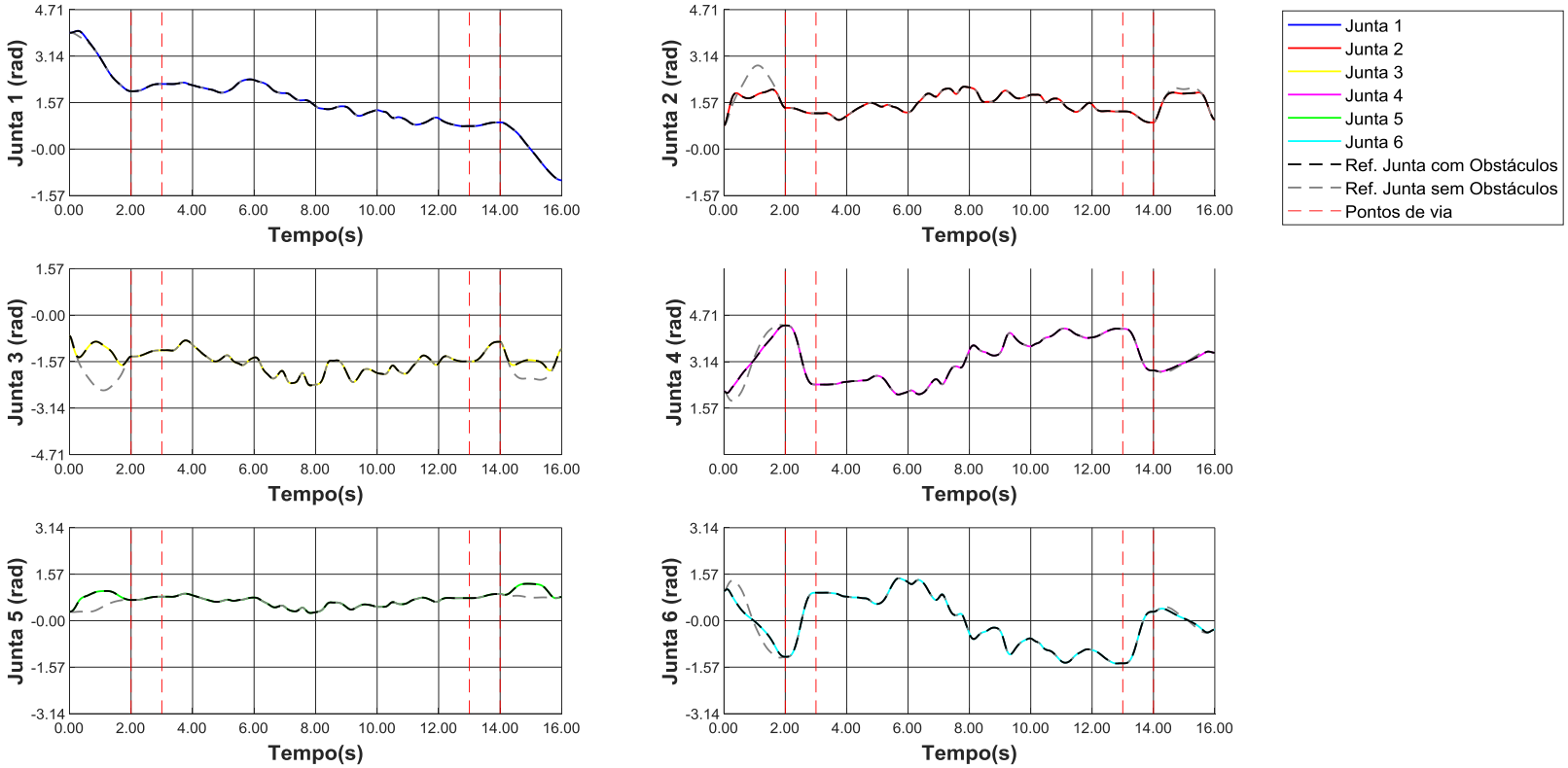


Fonte: autor.

A Fig. 40 apresenta as variações dos ângulos das juntas, enquanto a Fig. 41 a posição do efetuador final nos eixos cartesianos e a Fig. 42 a distância entre o efetuador e os obstáculos. Durante o processo de escrita a variação dos ângulos das juntas é mais turbulenta, conforme a Fig. 40. Os dois primeiros e últimos segundos são os instantes em que acontecem o desvio de obstáculos, também marcados pelas modificações das posições no eixo Z, Fig. 41. Ademais, neles, as distâncias entre o efetuador e os objetos são menores, Fig. 42.

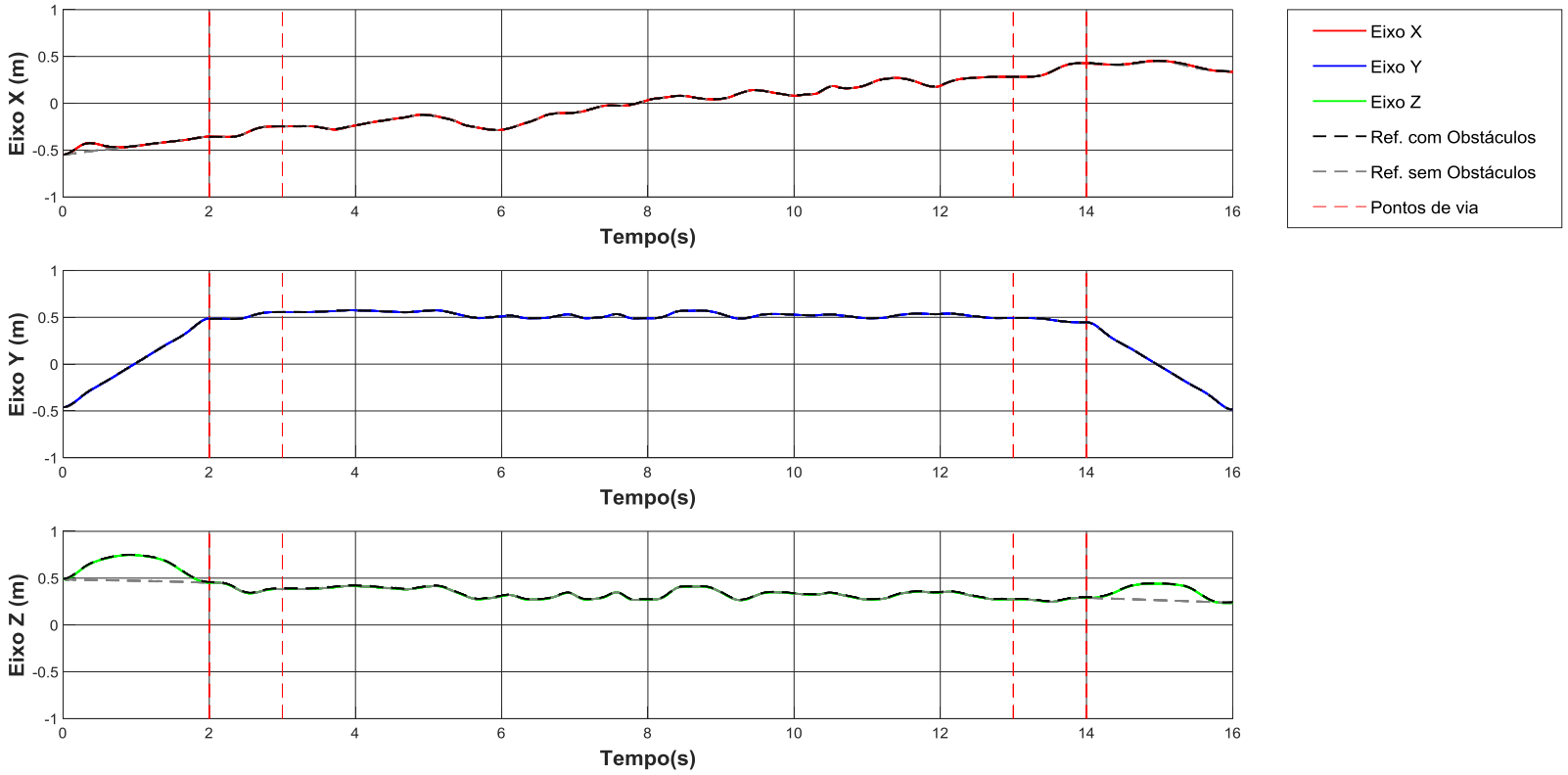
Em geral, o algoritmo desenvolvido no planejamento das trajetórias mostrou-se eficaz na execução das tarefas estabelecidas. O robô conseguiu mover-se entre posições pré-determinadas, evitar obstáculos em seu caminho, mover caixas até a esteira transportadora e desenhar sobre uma mesa utilizando um método de implementação simples e intuitivo.

Figura 40: Ângulos das juntas do robô ao longo da execução da Tarefa 3.



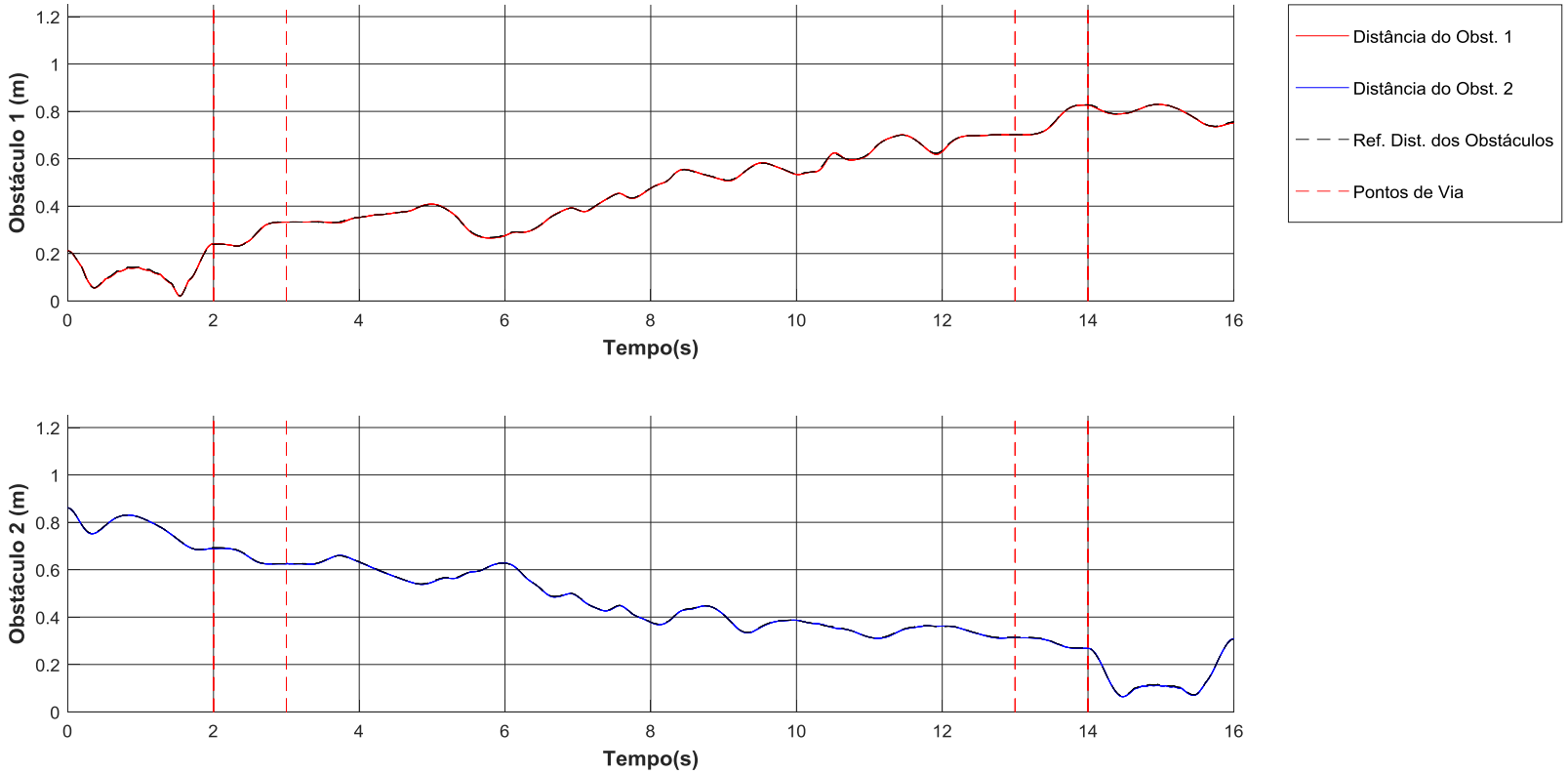
Fonte: autor.

Figura 41: Posições do efetuador final ao longo da Tarefa 3.



Fonte: autor.

Figura 42: Distâncias entre o efetuador final e os obstáculos ao longo da Tarefa 3.



Fonte: autor.

CAPÍTULO 6: PLANEJAMENTO DE TRAJETÓRIAS COM OTIMIZAÇÃO ENERGÉTICA

Este capítulo trata do planejamento de trajetórias visando minimizar o consumo de energia para a realização de uma dada tarefa. Trata-se de um problema de interesse prático pois um robô manipulador, ao ser introduzido em uma linha de produção, será responsável por um número elevado de tarefas a serem realizadas de forma interativa e a redução no consumo de energia para sua realização, pelo fator de escala, sempre pode representar uma redução não negligenciável no custo daquela etapa do processo.

Primeiramente, faz-se necessário comparar a estratégias de Planejamento de Trajetórias entre esse capítulo e o anterior. No anterior, a construção da trajetória é feita no Espaço Cartesiano e é subdividida em duas etapas. Primeiramente, o caminho é modelado através de pontos inseridos em segmentos retilíneos que podem ser modificados por elipsoides caso haja obstáculos ou saia dos limites do espaço de trabalho. Em seguida, as restrições temporais são impostas utilizando curvas B-Splines, que interpolam os pontos. Neste Capítulo, a trajetória é modelada em sua íntegra: o caminho e as restrições temporais são gerados simultaneamente. Além disso, ela é construída no Espaço das Juntas utilizando polinômios e também de maneira *off-line*. Os coeficientes dos polinômios, a_k, são modificados conforme o Algoritmo de Otimização Energética, Seção 6.1., para a obtenção de curvas com o menor custo energético possível. Para a validação do método, realizou-se dois experimentos: um sem considerar a presença de obstáculos, Seção 6.2., e outro considerando, Seção 6.3..

6.1. Algoritmo de Otimização Energética

Em um problema de otimização, a definição da função objetivo e as restrições relacionadas ao problema são determinantes para a busca de soluções. Dependendo do problema ou da tarefa a ser considerada, diferentes funções e restrições podem ser geradas. Neste trabalho, optou-se pela função "Mínimo Esforço Mecânico", dada pela Eq. (55).

$$f(a_k) = \sum_{i=1}^{n} \int_0^{t_f} \tau_i(a_i)^2 dt \tag{55}$$

Onde a_i é o coeficiente independente do polinômio associado à i junta do manipulador, com $i \in \{0,1,2,\dots,n\}$. Cada junta realiza um movimento diferente uma das outras e, por isso, está associada com um polinômio distinto. A trajetória polinomial é construída, conforme o Apêndice D, como uma função de uma variável independente. Ademais, a função Mínimo Esforço foi escolhida devido a sua relação com os torques, τ_i, que se relacionam diretamente com a energia mecânica do manipulador. Ela é utilizada em diversos trabalhos no planejamento de trajetórias energeticamente eficientes (Ayten *et al*., 2016, Wilson *et al*., 2004 etc.).

A obtenção dos torques das juntas dependem das variáveis cinemáticas, posições, velocidades e acelerações, obtidas por meio das equações polinomiais, e isso é que denota o problema de Dinâmica Inversa. Ademais, o algoritmo apresentado por Siciliano (2008) demanda a utilização de matrizes de transformação obtidas utilizando o método de Denavit-Hartenberg, dessa forma, não foi possível aproveitar os resultados do modelo de simulação anterior, o qual se baseou no método dos Helicoides Sucessivos. A figura esquemática do manipulador, bem como os referenciais adotados em cada junta, são expostos como um resultado na Seção 6.2.2..

Para gerar trajetórias que são livres de colisões com obstáculos e que sejam realizáveis pelo robô é indispensável a adição de restrições ao problema de otimização. Elas são divididas em dois tipos: Restrições de Sistema e Restrições de Ambiente. As Restrições de Sistema estão relacionadas às variáveis intrínsecas do manipulador: posições, velocidades, acelerações, arranques e torques das juntas, conforme as Eq. em (56).

Torques máximos	$\lvert\tau_i(a_i)\rvert \leq \tau_i^{max}$	
Arranques máximos	$\lvert\dddot{q}_i(a_i)\rvert \leq \dddot{q}_i^{\max}$	
Acelerações máximas	$\lvert\ddot{q}_i(a_i)\rvert \leq \ddot{q}_i^{max}$	(56)
Velocidades máximas	$\lvert\dot{q}_i(a_i)\rvert \leq \dot{q}_i^{max}$	
Valores factíveis de posições	$q_i^{min} \leq q_i(a_i) \leq q_i^{max}$	

Os limites das juntas devem ser escolhidos de forma que não aconteça sobreposições entre os elos do manipulador. Isso, fisicamente, acarretaria em uma colisão. Ademais, as outras variáveis devem ser limitadas para que estejam em valores possiveis fisicamente. Por exemplo, os torques máximos devem ser escolhidos conforme as limitações dos motores utilizados.

No que concerne às Restrições do Ambiente, foram estabelecidas restrições para três posições do manipulador: do efetuador final, do punho esférico, e do seu terceiro elo. As restrições referentes às posições do Efetuador Final, $P(a_i) = [P_X(a_i) \quad P_Y(a_i) \quad P_Z(a_i)]^T$, são dadas pelas Eq. em (57).

$$\begin{gathered} \frac{P_X(a_i)^2 + P_Y(a_i)^2}{(r_{RA})^2} \leq 1 \; e \; P_Z(a_i) \leq h_{RA} \\ \frac{(P_X(a_i) - x_c)^2}{x_r^2} + \frac{(P_Y(a_i) - y_c)^2}{y_r^2} + \frac{(P_Z(a_i) - z_c)^2}{z_r^2} \leq 1 \\ P_Z(a_i) \leq h_C \end{gathered} \tag{57}$$

Elas estabelecem que o efetuador final não esteja posicionado nas coordenadas de um cilindro de altura h_{RA} e raio r_{RA}, que representa as estruturas mais inferiores do robô, de elipsoides, que modelam os obstáculos (Seção 4.2.), e nem abaixo de uma altura mínima h_C, para assegurar que não aja colisão com o solo. As restrições relacionadas às posições do punho esférico, $P_W(a_i) = [P_{W_X}(a_i) \quad P_{W_Y}(a_i) \quad P_{W_Z}(a_i)]^T$, são indicadas nas Eq. em (58).

$$\begin{gathered} \frac{P_{W_X}(a_i)^2 + P_{W_Y}(a_i)^2}{(r_{RA})^2} \leq 1 \; e \; P_{W_Z}(a_i) \leq \; h_{RA} \\ \frac{\left(P_{W_X}(a_i) - x_c\right)^2}{x_r^2} + \frac{\left(P_{W_Y}(a_i) - y_c\right)^2}{y_r^2} + \frac{\left(P_{W_Z}(a_i) - z_c\right)^2}{z_r^2} \leq 1 \\ P_{W_Z}(a_i) \leq h_C \end{gathered} \tag{58}$$

De maneira análoga às restrições em (58), tais restrições se referem, respectivamente, ao cilindro das estruturas inferiores, aos elipsoides dos obstáculos e à altura mínima. Finalmente, tem-se a Eq. em (59), que representa a restrição das posições do Elo 3 do Manipulador, $P_3(a_i) = [P_{3_X}(a_i) \quad P_{3_Y}(a_i) \quad P_{3_Z}(a_i)]^T$, relacionada apenas ao cilindro que assegura que ele não entre em rota de colisão com os membros inferiores.

$$\frac{P_{3_X}(a_i)^2 + P_{3_Y}(a_i)^2}{(r_{RA})^2} \leq 1 \; e \; P_{3_Z}(a_i) \leq h_{RA} \tag{59}$$

Os parâmetros do cilindro, altura do solo e elipsoides são determinados empiricamente. Não foram consideradas restrições para os Elos 4, 5 e 6 pois as restrições do punho esférico e do efetuador final cumprem as restrições desses elos indiretamente. Sobre o Elo 2, as limitações impostas das posições da segunda junta já asseguram que esse elo não fique sobreposto ao primeiro, devido a isso não é necessário adicionar restrições de ambiente a ele.

Para garantir que todas as restrições sejam obedecidas, utilizou-se uma Função de Penalizações (*Penalty function*), que age modificando a Função Objetivo com um valor especificado. Logo, os coeficientes dos polinômios relacionados à Função Objetivo modificada não serão incluídos aos ótimos e, em consequência, serão descartados. A Função Objetivo Aumentada, que inclui a Função de Penalizações, é dada pela Eq. (60), onde $\tilde{F}$ denota um espaço de busca factível e $f_P(a_i)$ é a função de penalização.

$$F(a_{ni}) = \begin{cases} f(a_i), & se\ a_i \in \tilde{F} \\ f(a_i) + f_P(a_i), & caso\ contrário \end{cases} \tag{60}$$

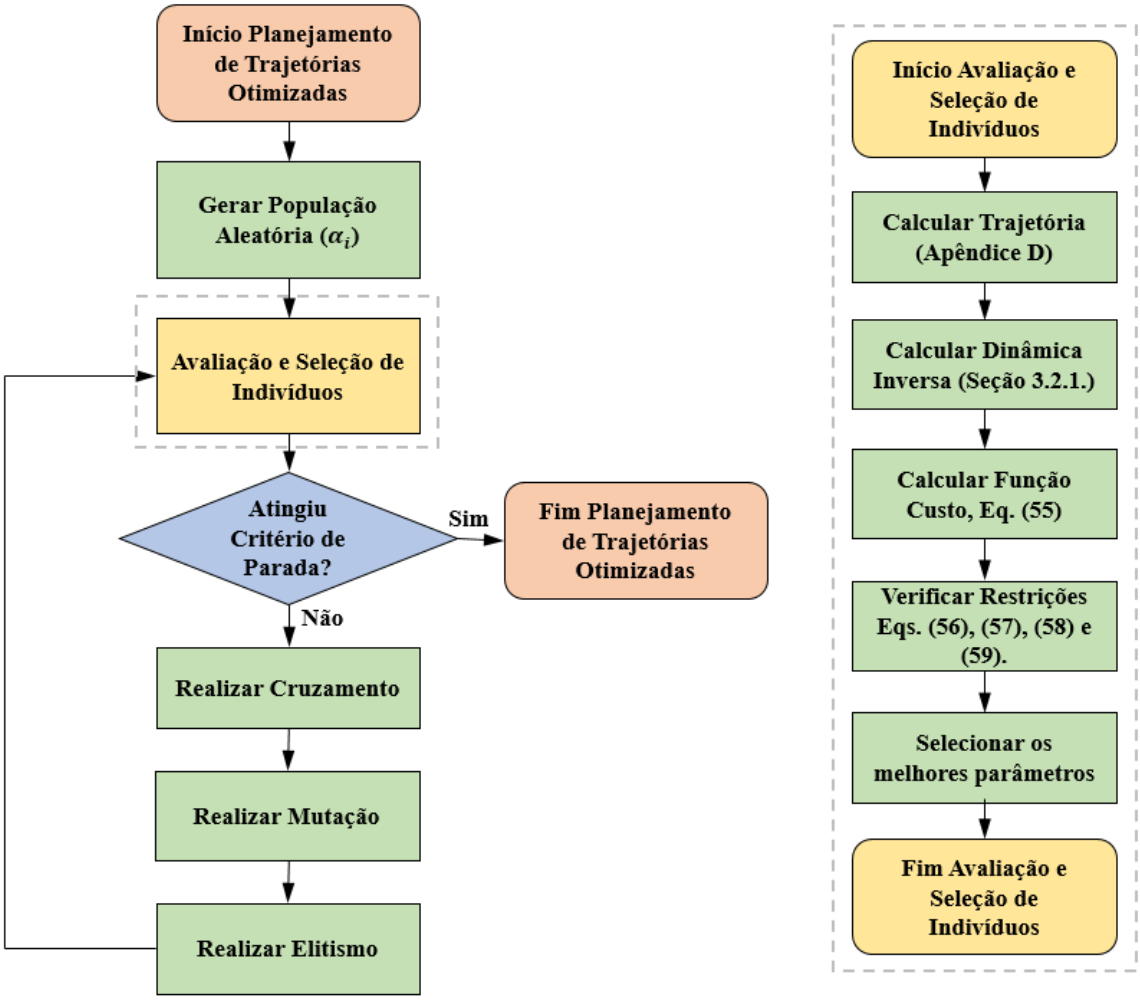
Logo, o problema de otimização é descrito como:

$$a_i^* = \arg\min_{a_i} F(a_i)$$

Para sua solução, utilizou-se os Algoritmos Genéticos (AG). Esse método foi empregado pela sua simplicidade de implementação e por possuir buscas baseadas em métodos heurísticos e probabilísticos, que contornam as não-linearidades apresentadas na Função Objetivo, devido ao cálculo da Dinâmica Inversa, por exemplo.

O algoritmo implementado é análogo ao modelo padrão de AG, apresentado no Apêndice F, cujo esquema, adaptado ao Planejamento de Trajetórias, é representado pela Fig. 43.

Figura 43: Esquema do Algoritmo Genético desenvolvido para planejamento de trajetórias.



Fonte: autor.

O Algoritmo inicia-se gerando uma população com valores aleatórios dos coeficientes dos polinômios, a_i. Em seguida, acontece a fase de avaliação, que é subdividida em etapas. Nela, primeramente, as trajetórias são calculadas através da formulação matemática do polinômio e, em seguida, a partir da Dinâmica Inversa, os torques nas juntas são calculados. Com os valores de torque determinados, a avaliação dos indivíduos é efetivada por meio da Função *fitness*. Posteriormente, constata-se se os indivíduos atingem todos os critérios de restrições com o intuito de selecionar os melhores. Após isso, verifica-se se o critério de parada foi atingido. Se sim, o algoritmo se encerra. Senão, as etapas Cruzamento, Mutação e Elitismo

são executadas e o algoritmo retorna à etapa de Avaliação. Isso é repetido até atingir o critério de parada.

6.2. Experimento 1: Otimização Energética de Trajetórias Sem Obstáculos

Esse experimento visa a obtenção de trajetórias entre duas posições com consumo energético reduzido. Não são considerados obstáculos entre as posições. Ele é subdividido nas seguintes etapas: 6.2.1. Metodologia e 6.2.2. Resultados.

6.2.1. Metodologia

As trajetórias foram avaliadas e comparadas em termos da Função Objetivo para as juntas 2, 3 e 5 do robô devido a essas juntas suportarem cargas mais significativas dada a arquitetura e os parâmetros inerciais considerados. Para uma implementação mais simplificada, não foram considerados efeitos de atrito viscoso, de Coulomb, tampouco de redução de engrenagens. Todavia, cabe ressaltar que esses efeitos podem ter influencia no desempenho do sistema. Por exemplo, em experimentos realizados com o Robô Puma por Corke (2017), os efeitos de fricção exigiram entre 10 a 47% do torque máximo nas três primeiras juntas.

A implementação do algoritmo de otimização começa com o desenvolvimento dos modelos do manipulador. Primeiramente, resolveu-se o problema de Cinemática Direta pelo método de Denavit-Hartenberg para utilizá-lo na solução da Dinâmica Inversa empregando o método de Newton-Euler. Com tais resultados, implementou-se o Algoritmo de Otimização conforme a Seção 6.1.. Nessa etapa, definiu-se, empiricamente, os parâmetros de restrições utilizados. Em relação às Restrições de Sistema, utilizou-se os valores expostos na Tab. 13.

Tabela 13: Restrições do robô antropomórfico adotadas para as juntas 2, 3 e 5.

Junta	**2**	**3**	**5**
$\boldsymbol{q_{min}\ (rad)}$	$-\pi/2$	$-3\pi/2$	$-\pi$
$\boldsymbol{q_{max}\ (rad)}$	$3\pi/2$	$\pi/2$	π
$\boldsymbol{\lvert\dot{q}\rvert_{max}(rad/s)}$	10,0	10,0	10,0
$\boldsymbol{\lvert\ddot{q}\rvert_{max}(rad/s^2)}$	20,0	20,0	20,0
$\boldsymbol{\lvert\dddot{q}\rvert_{max}(rad/s^3)}$	100,0	100,0	100,0
$\boldsymbol{\lvert\tau\rvert_{max}\ (N \cdot m)}$	100,0	100,0	100,0

Fonte: Autor.

No que concerne àqueles referentes às Restrições de Ambiente, os valores foram: $h_{RA} = 0{,}48$, $r_{RA} = 0{,}144$ e $h_C = 0{,}05$.

Os operadores empregados no Algoritmo Genético foram os seguintes: codificação com números reais, seleção utilizando o método da roleta e um cruzamento do tipo *flat*. Adotou-se, também, o operador elitismo, para garantir que não haja perda de informações. A Tab. 14 mostra os valores dos parâmetros do algoritmo genético utilizados.

Tabela 14: Parâmetros do algoritmo genético utilizados.

N° População	N° Genótipo	N° Geração	Taxa de Cruzamento	Taxa de Mutação
100	1	100	0,5	0,5

Fonte: autor.

Com o Algoritmo de Otimização implementado, realizou-se sua sintonia por meio de 14 ensaios que variam parâmetros do AG como População, Genótipos, Gerações, Taxas de Cruzamento e Mutação, os quais são expostos na Tab. 15.

Tabela 15: Parâmetros do algoritmo genético associados aos ensaios.

Ensaio	N° População	N° Genótipo	N° Geração	Taxa de Cruzamento	Taxa de Mutação
1	100	1	100	0,8	0,2
2	100	10	100	0,8	0,2
3	10	100	100	0,8	0,2
4	10	10	100	0,8	0,2
5	10	10	50	0,8	0,2
6	10	10	10	0,8	0,2
7	10	10	5	0,8	0,2
8	10	10	10	0,5	0,2
9	10	10	10	0,7	0,2
10	10	10	10	0,9	0,2
11	10	10	10	0,8	0,1
12	10	10	10	0,8	0,3
13	10	10	10	0,8	0,5
14	10	10	10	0,8	0,7

Fonte: autor.

Os ensaios foram subdivididos de maneira a variar alguns parâmetros e fixar outros. Por exemplo, os ensaios 1, 2, 3 e 4 variam o tamanho da população e da quantidade de genótipos e mantêm fixos o restante, os ensaios 4, 5, 6 e 7 modificam o número de gerações enquanto os outros são constantes etc. Cada ensaio foi simulado 30 vezes de forma que os resultados gerados correspondessem a uma média e um intervalo de confiança em torno dele. Isto foi feito pois o AG nem sempre retorna os mesmos resultados ao ser executado.

- **Modelo de Simulação**

O objetivo desse experimento é comparar o desempenho das diferentes trajetórias, tanto das que foram otimizadas energeticamente quanto das que não foram, em termos de Mínimo Esforço e na ausência de obstáculos. As trajetórias polinomiais de 4ª, 6ª e 8ª ordens são as trajetórias que foram otimizadas, pois elas, devido à maneira que seus modelos matemáticos foram confeccionados (Apêndice D), possuem variáveis independentes. As trajetórias polinomiais de 3ª, 5ª e 7ª ordens, bem como a do Capítulo 4, são as trajetórias que não foram otimizadas. As condições iniciais e finais de posição e orientação utilizadas nesse experimento são indicadas na Tab. 16.

Tabela 16: Condições iniciais e finais de posição e orientação utilizadas no experimento sem obstáculos.

Condições		**Posição** $[x \quad y \quad z]^T\ (m)$	**Orientação** $[\alpha \quad \beta \quad \gamma]^T\ (rad)$
1	Iniciais	$[0{,}426 \quad -0{,}372 \quad 0{,}165]^T$	$[-0{,}374 \quad 2{,}756 \quad 0{,}298]^T$
	Finais	$[0{,}462 \quad 0{,}402 \quad 0{,}064]^T$	$[0{,}762 \quad 2{,}48 \quad 0{,}0409]^T$
2	Iniciais	$[0{,}462 \quad 0{,}402 \quad 0{,}064]^T$	$[0{,}762 \quad 2{,}48 \quad 0{,}0409]^T$
	Finais	$[-0{,}0495 \quad 0{,}587 \quad 0{,}152]^T$	$[1{,}71 \quad 2{,}33 \quad 0{,}148]^T$
3	Iniciais	$[-0{,}0495 \quad 0{,}587 \quad 0{,}152]^T$	$[1{,}71 \quad 2{,}33 \quad 0{,}148]^T$
	Finais	$[-0{,}625 \quad 0{,}369 \quad 0{,}398]^T$	$[2.626 \quad 1.820 \quad 0{,}0032]^T$

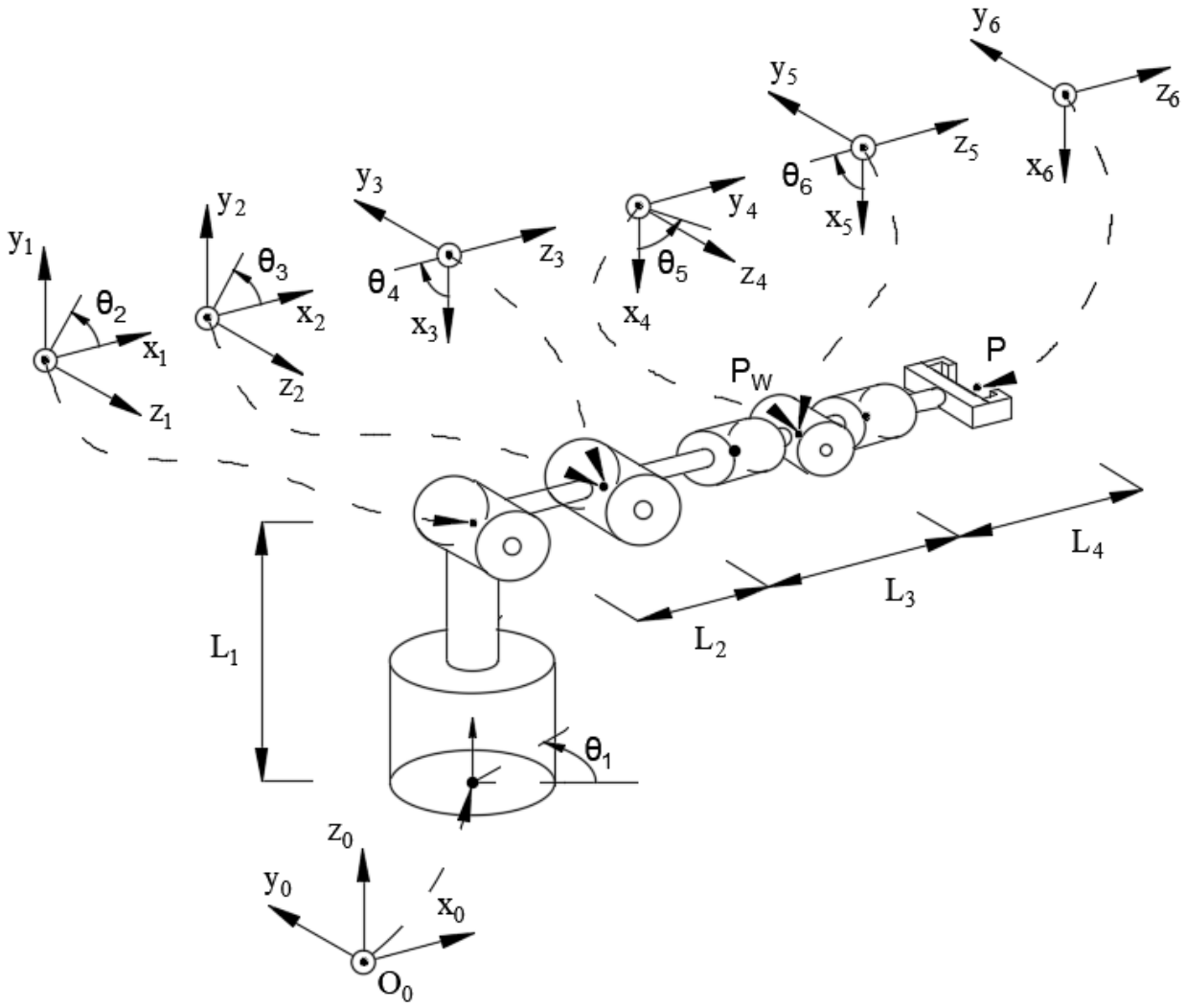
Fonte: Autor.

As trajetórias, que empregavam diferentes métodos, foram geradas para cada uma das condições iniciais e finais e comparadas. Além disso, para as trajetórias polinomiais foi necessário converter essas condições para o espaço das juntas, por meio da Cinemática Inversa.

6.2.2. Resultados e Discussões

A Fig. 44 mostra os referenciais considerados para a definição dos parâmetros de Denavit-Hartenberg para a solução da Dinâmica Direta.

Figura 44: Determinação dos referenciais de cada junta do robô através do método de Denavit-Hartenberg



Fonte: autor.

Como pode ser visto na figura, o eixo Z dos referenciais 1, 2 e 4 está na mesma direção mas no sentido oposto do eixo Y do referencial global; o eixo Z dos referenciais 3, 5 e 6 está

na mesma direção e sentido do eixo X do refencial global. A Tab. 17 exibe os valores dos parâmetros de Denavit-Hartenberg utilizados para obtenção dos referenciais desejados.

Tabela 17: Parâmetros do método de Denavit-Hartenberg utilizados.

Junta	$\boldsymbol{d}$	$\boldsymbol{\theta}$	$\boldsymbol{a}$	$\boldsymbol{\alpha}$
1	L_1	θ_1	0	$\pi/2$
2	0	θ_2	L_2	0
3	0	$\theta_3 - \pi/2$	0	$-\pi/2$
4	L_3	θ_4	0	$\pi/2$
5	0	θ_5	0	$-\pi/2$
6	L_4	θ_6	0	0

Fonte: autor.

A seguir são apresentados os resultados e discussões dos tópicos: Sintonia do Algoritmo de Otimização e Modelos de Simulação.

- **Sintonia do Algoritmo de Otimização**

A Tab. 18 apresenta os valores de Mínimo Esforço (ME) do robô e o tempo de execução em cada ensaio.

Tabela 18: Mínimo Esforço (ME) e tempo de execução do algoritmo em cada ensaio.

Ensaio	**ME**	**Tempo (s)**	**Ensaio**	**ME**	**Tempo (s)**
1	$623{,}90 \pm 3{,}94$	$108{,}03 \pm 0{,}97$	8	$667{,}34 \pm 6{,}36$	$11{,}07 \pm 0{,}20$
2	$622{,}37 \pm 2{,}45$	$1027{,}80 \pm 4{,}01$	9	$674{,}09 \pm 5{,}37$	$10{,}74 \pm 0{,}35$
3	$627{,}60 \pm 4{,}25$	$1170{,}30 \pm 74{,}5$	10	$670{,}80 \pm 5{,}33$	$12{,}05 \pm 0{,}40$
4	$648{,}55 \pm 8{,}55$	$106{,}11 \pm 0{,}43$	11	$674{,}49 \pm 5{,}39$	$10{,}83 \pm 0{,}25$
5	$653{,}39 \pm 9{,}68$	$55{,}16 \pm 1{,}69$	12	$669{,}83 \pm 5{,}84$	$9{,}92 \pm 0{,}15$
6	$665{,}37 \pm 12{,}15$	$10{,}32 \pm 0{,}56$	13	$669{,}17 \pm 4{,}86$	$11{,}89 \pm 0{,}41$
7	$680{,}01 \pm 10{,}84$	$4{,}98 \pm 023$	14	$660{,}36 \pm 5{,}09$	$12{,}61 \pm 0{,}11$

Fonte: autor.

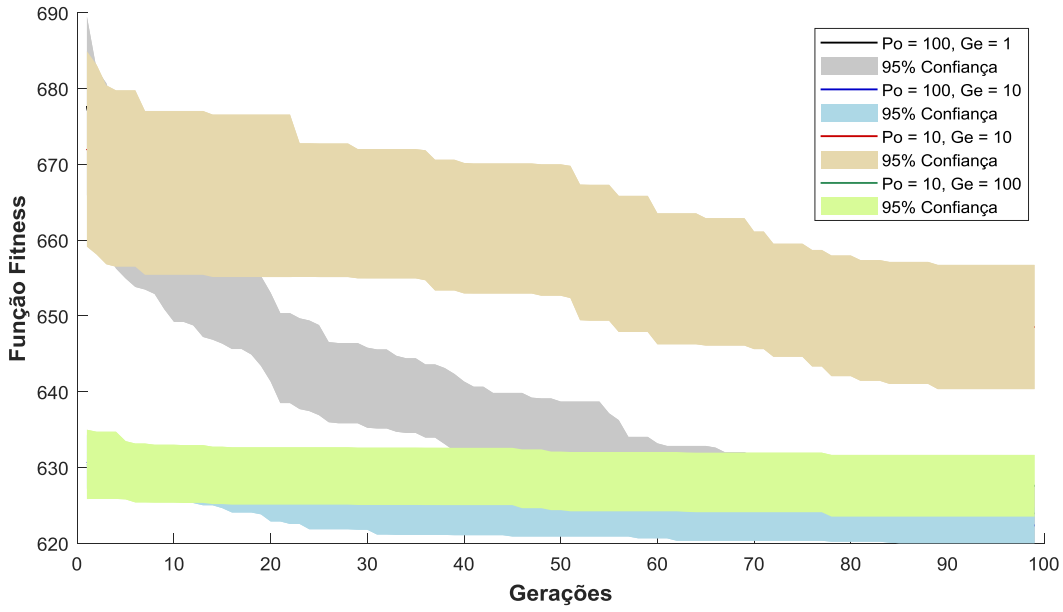
Analisando-a, juntamente com a Tab. 15, nota-se que o Esforço Mecânico é menor nos ensaios 1, 2 e 3, que apresentam maiores números de geração, população e genótipo. Um

resultado esperado, uma vez que quanto mais indivíduos ou quanto mais esses indivíduos se reproduzem, maiores são as chances de seus descendentes apresentarem boas características.

Todavia, apesar do aumento das variáveis supracitadas gerarem melhores resultados, percebe-se que o tempo de processamento também aumenta. Sendo assim, esses parâmetros do algoritmo devem ser escolhidos com cautela pelo projetista. Por exemplo, em movimentos pré-programados, aqueles em que a execução do algoritmo ocorre de maneira *offline*, o tempo de processamento não é limitante, uma vez que é necessário executar o algoritmo uma única vez, pois os movimentos desejados serão sempre repetidos. Já em movimentos *online*, o tempo de simulação também deve ser levado em conta, ele deve ser minimizado a ponto de possuir valores ínfimos, uma vez que esse tempo estará incluso ao tempo de resposta do manipulador. Nesses casos, devido aos resultados obtidos, recomenda-se a sintonia do algoritmo com valores menores de gerações, genótipos e população. Pode-se ainda utilizar um computador de melhor processamento e trocar informações na nuvem (como um sistema IoT).

Percebe-se ainda uma melhoria nos valores da Função *Fitness* ao aumentar a população em relação ao aumento do número de gerações e genótipos, como pode ser visto na Fig. 45 para o segundo caso.

Figura 45: Convergência do AG ao variar a População (Po) e a quantidade de Genótipos (Ge).



Fonte: autor.

Analisando a figura, em uma população igual a 100 indivíduos, com cada um possuindo 10 genes, o valor *fitness* médio após 100 gerações foi de 623,9 (linha em azul claro), enquanto para uma população de 10 indivíduos, com cada um possuindo 100 genótipos, o valor médio da função *fitness* após as 100 gerações foi de 627,6 (linha em verde). Possivelmente isso acontece pois uma maior população inicial promove uma grande variabilidade genética, o que promove resultados melhores.

Além disso, observa-se também que maiores taxas de mutação também provocam melhorias, pois, conforme o AG foi implementado, taxas maiores equilibram o *Exploitation e o Exploration*, o que é fundamental para bom desempenho do AG. *Exploitation* consiste em aproveitar as informações das soluções conhecidas da melhor maneira possível, como um refinamento, o que é feito por métodos de seleção considerados fortes e pela utilização de operadores complementares. *Exploration* se baseia na investigação de áreas desconhecidas no espaço de busca, o que é realizado pelo operador de mutação. Dessa forma, devido ao uso do método da Roleta e do operador Elitismo, que aumentam o *Exploitation*, é necessário aumentar a taxa de mutação, a qual aumenta o *Exploration*, para promover equilíbrio.

O equilíbrio explica o sucesso dos Algoritmos Genéticos em problemas com grandes complexidades, em detrimento dos Métodos Clássicos. Os métodos baseados em gradiente, por exemplo, não são capazes de explorar todo o espaço de busca com muito rigor nesse tipo de problema, pois, devido à existência de múltiplos mínimos e máximos locais, os métodos podem ficar presos nessas regiões e não encontrar a solução ótima global. Assim sendo, os Algoritmos Genéticos, por apresentarem uma capacidade multidirecional de busca, são mais adequados para tratar esse tipo de problema.

- **Modelos de Simulação**

Os valores de esforço mínimo do robô obtidos das trajetórias são indicados na Tab. 19. Nessa tabela, nota-se que as trajetórias polinomiais que não foram otimizadas, com 3ª, 5ª ou 7ª ordens, exibiram valores mais altos de esforço do que as demais trajetórias. A explicação para esse fato é que esses polinômios geram curvas que se aproximam dos limites externos do espaço de trabalho do manipulador e isso promove esforços mecânicos maiores, uma vez que ele deve ficar extendido.

Tabela 19: Mínimo Esforço das trajetórias no Experimento 1.

Trajetória	Condições Iniciais e Finais		
	1	2	3
3ª ordem	964,71	870,83	971,54
5ª ordem	961,22	868,64	967,70
7ª ordem	958,90	866,89	964,98
4ª ordem	$585{,}57 \pm 3{,}23$	$543{,}82 \pm 2{,}21$	$543{,}99 \pm 3{,}15$
6ª ordem	$644{,}33 \pm 3{,}58$	$590{,}63 \pm 4{,}24$	$604{,}55 \pm 3{,}18$
8ª ordem	$684{,}32 \pm 3{,}57$	$628{,}26 \pm 3{,}54$	$659{,}92 \pm 3{,}97$
Algoritmo Cap. 4	662,93	756,17	800,87

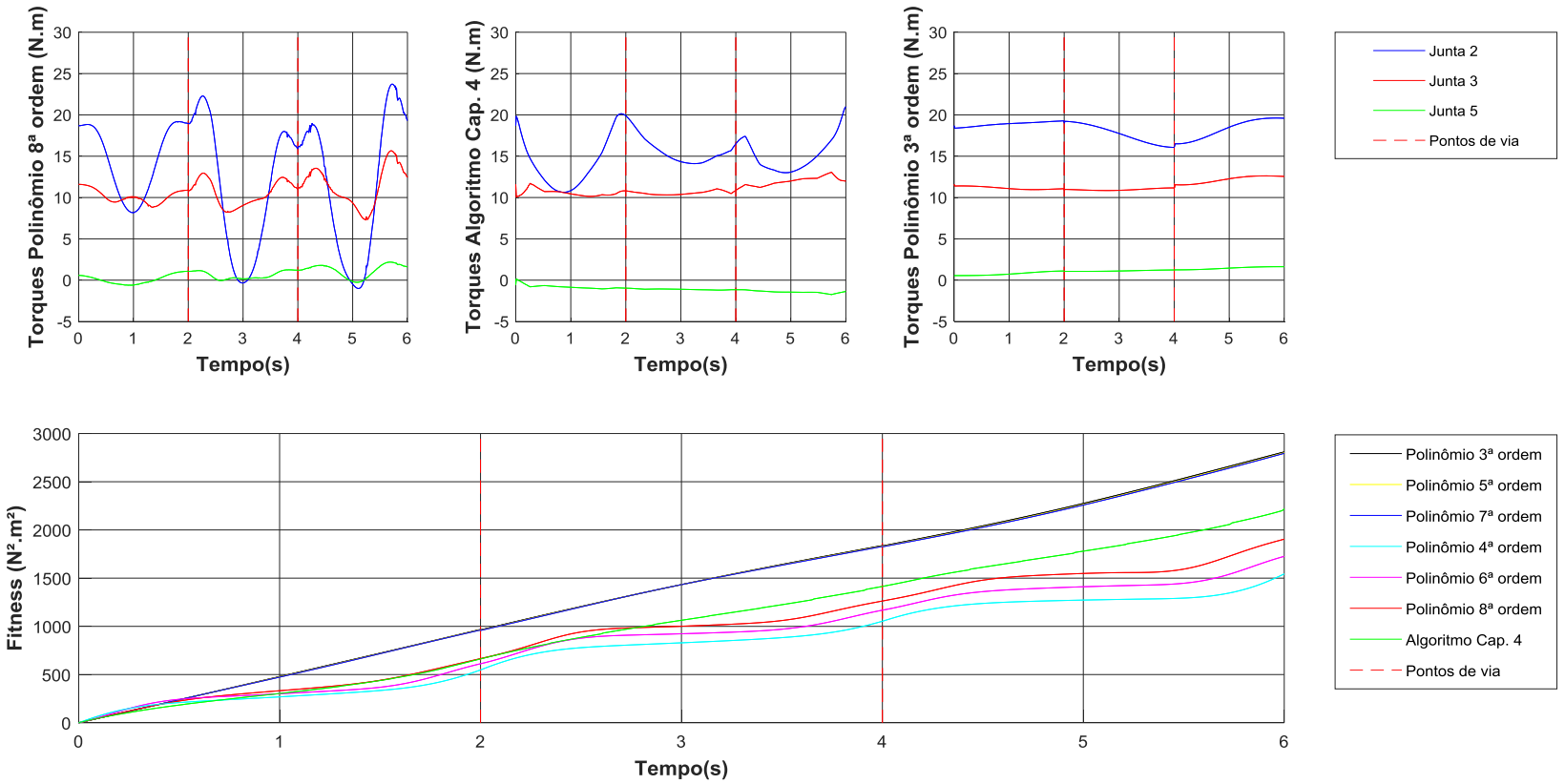
Fonte: autor.

Analisando novamente a Tab. 19, nota-se que as trajetórias polinomiais otimizadas, de 4ª, 6ª e 8ª ordens, apresentaram os menores valores de da função *Fitness*. Ademais, cosntata-se que quando a ordem do polinômio decresce, esses valores diminuem, por exemplo, na Condição Inicial 1, os valores são 585,57 quando se utiliza polinômio de 4ª ordem, 644,33 para o polinômio de 6ª ordem e 684,32 no polinômio de oitava ordem. Possivelmente, isso acontece devido aos polinômios de ordens mais altas apresentarem mais restrições em suas formulações matemáticas, o que diminui o número de possibilidades na geração das trajetórias.

Verifica-se também através da Tab. 19 que as trajetórias construídas pelo Algoritmo Cap. 4, que não foram otimizadas, apresentaram baixos valores de Esforço, quando comparadas, por exemplo, com as trajetórias polinomiais não otimizadas. Inclusive uma delas apresentou um esforço menor do que uma trajetória que foi otimizada. Isso aconteceu devido essas trajetórias possuírem caminhos retilíneos que, de certa forma, demandam que o manipulador fique em posições que são favoráveis energeticamente.

Para caracterizar o que vem a ser uma posição favorável energeticamente, tem-se a Fig. 46, que exibe os valores de torque das juntas de algumas trajetórias. Quando o efetuador percorre uma trajetória retilínea, o robô, de certa forma, se "comprime", ficando em uma posição que diminui os efeitos gravitacionais e explica os baixos valores de esforços mecânicos. Esse mesmo ato de se encolher é observado em maior escala nas trajetórias otimizadas, principalmente nos instantes intermediários de tempo entre as duas posições. Nessas situações supracitadas, o torque nas juntas possui menores valores em relação as posições em que o robô

Figura 46: Toques nas juntas 2, 3 e 5, e função *fitness* como funções do tempo utilizando os diferentes métodos de planejamento de trajetórias.



Fonte: autor.

está mais distendido, o que acontece com as trajetórias polinomiais não otimizadas (3ª, 5ª e 7ª ordens). Portanto, infere-se que a distribuição de massa do manipulador é um dos principais fatores que remetem à diminuição de consumo energético.

A Tab. 20 apresenta as porcentagens de esforço mecânico poupado ao utilizar as trajetórias otimizadas em relação às não otimizadas. Os valores são apresentados em intervalos devido as porcentagens variarem em cada condição inicial e final imposta.

Tabela 20: Porcentagens de esforço mecânico poupado das trajetórias otimizadas, 4ª, 6ª e 8ª ordens, em relação às não otimizadas, 3ª, 5ª, 7ª ordens e a obtida através do Algoritmo Elipsoide.

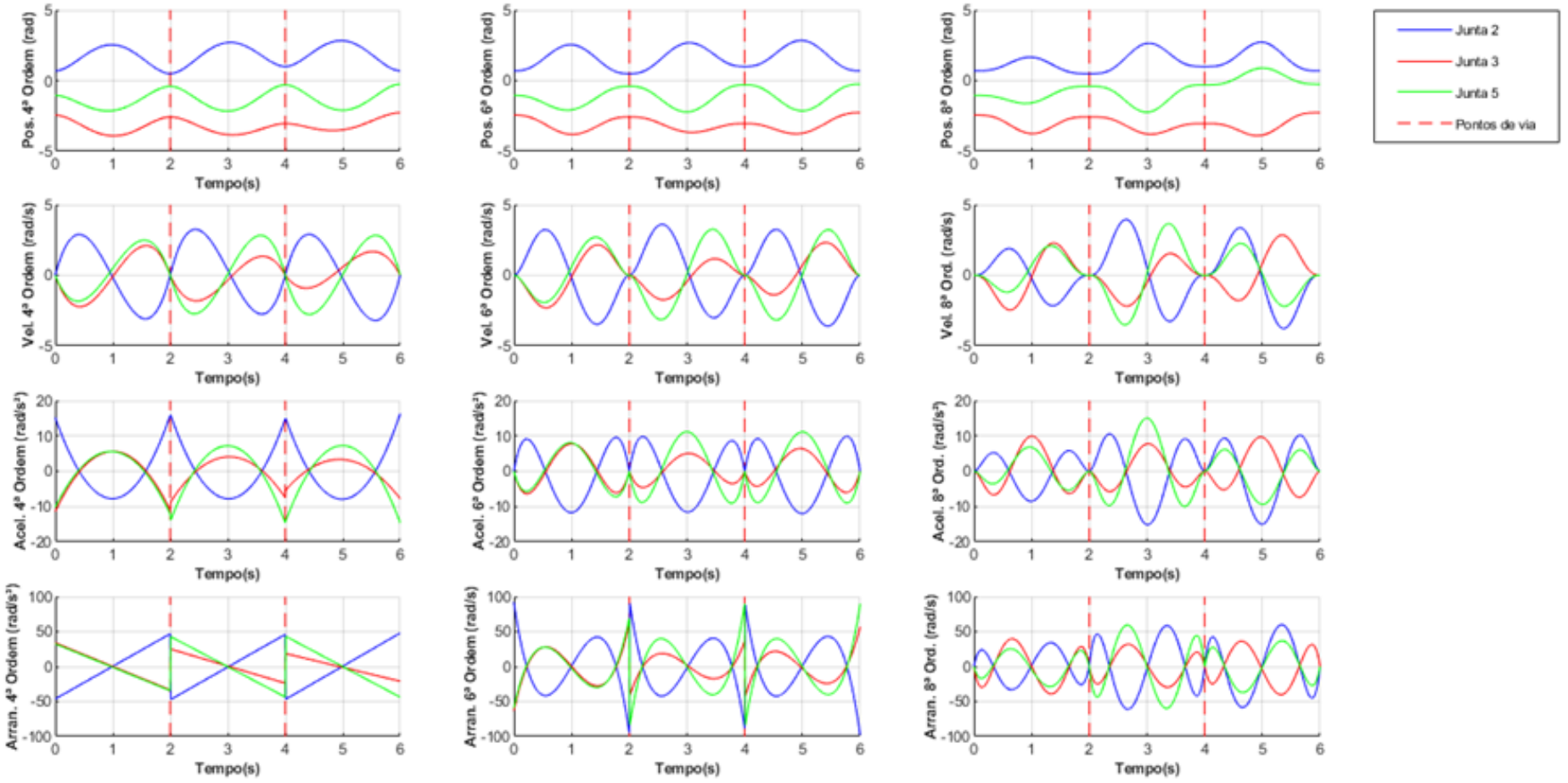
Trajetórias	**4ª ordem**	**6ª ordem**	**8ª ordem**
3ª ordem	Entre 37 e 44%	Entre 32 e 38%	Entre 27 e 33%
5ª ordem	Entre 37 e 44%	Entre 32 e 38%	Entre 27 e 33%
7ª ordem	Entre 37 e 44%	Entre 31 e 38%	Entre 27 e 33%
Algoritmo Elipsoide	Entre 11 e 32%	Entre 3 e 25%	Entre -3 e 18%

Fonte: autor.

As condições iniciais e finais afetam diretamente na quantidade de energia que será consumida, pois elas indicam as posições das juntas que impreterivelmente estarão inseridas nas trajetórias. Logo, quando se emprega, por exemplo, posições inicias e finais que são favoráveis energeticamente e a trajetória é planejada com um caminho retilíneo, o manipulador apresentará um baixo consumo energético, uma vez que o manipulador irá ficar contraído durante todo o percurso e, dessa forma, estará em posições de menor consumo energético. Isso explica o porquê da trajetória gerada através do Algoritmo Cap. 4 apresentou esforços mecânicos menores do que a polinômial de 8ª ordem na primeira condição inicial e final.

Comparando agora as trajetórias sob outras perspectivas, em termos de velocidades, acelerações e arranques, percebe-se certas diferenças entre elas, Fig. 47. As trajetórias construídas através de polinômio de 4ª e 6ª ordem apresentaram descontinuidades de arranques nas extremidades, o que não aconteceu nas curvas polinomiais de 8ª, devido a forma que esses polinômios foram formulados. É importante ressaltar que a continuidade e a suavidade da velocidade, da aceleração e do arranque são importantes pois diminuem desgastes dos motores. Portanto, nota-se a existência de um *trade-off* entre o esforço mecânico do robô e a suavidade das curvas: trajetórias polinomiais de quarta ordem possuem esforços mecânicos menores mas

Figura 47: Comportamento das variáveis de junta como funções do tempo empregando os métodos de otimização.



Fonte: autor.

não asseguram aceleração e arranque suaves na extremidade, enquanto trajetórias polinomiais de oitava ordem possuem esforços maiores mas asseguram continuidade e suavidade. Manifesta-se a dúvida se os esforços mecânicos que seriam poupados utilizando trajetórias de ordens inferiores compensam os possíveis desgastes que serão causados nos motores.

Ademais, constata-se uma vantagem clara das trajetórias otimizadas em relação as trajetórias do Algoritmo Cap. 4. Em relação aos esforços mecânicos, apesar das trajetórias retilíneas fornecerem esforços baixos, eles são ocasionais, pois são dependentes das condições iniciais e finais impostas. Em relação a suavidade das extremidades do movimento dos motores, essas trajetórias não a asseguram, uma vez que são formuladas no espaço cartesiano.

6.3. Experimento 2: Otimização Energética de Trajetórias Com Obstáculos

No segundo experimento, considerando agora os obstáculos, as trajetórias foram formuladas, Para cada uma das condições inicial e final, através de polinômios com ordens que possibilitassem otimização energética, isto é, polinômios com 4ª, 6ª e 8ª ordens. Para fins comparativos, também foram confeccionadas trajetórias utilizando o Algoritmo Cap. 4. Essa Seção é dividida nas seguintes partes: 6.3.1. Metodologia e 6.3.2. Resultados e Discussões.

6.3.1. Metodologia

As diferenças entre esse experimento e o anterior é basicamente a inclusão dos obstáculos durante a execução da trajetória. Logo, foi possível aproveitar os resultados obtidos anteriormente referentes à Cinemática/Dinâmica do manipulador e à Sintonia do Algoritmo Genético.

Foram considerados dois obstáculos entre as Condições Iniciais e Finais 1 e um obstáculo entre as Condições Inicias e Finais 2 e 3. As Condições Iniciais e Finais 1 e 2 são as mesmas utilizadas na Tarefa 2 do Cap. 5, nos instantes em que o manipulador desvia de duas caixas e de uma prateleira, respectivamente. Já as Condições Iniciais e Finais 3 são as mesmas utilizadas na Tarefa 3, no momento em que o manipulador desvia de armário industrial.

A Tab. 21 mostra os valores das condições iniciais e finais de posição e orientação utilizadas no experimento.

Tabela 21: Condições iniciais e finais de posição e orientação utilizadas no experimento com a presença de obstáculos.

Condições		**Posição** $[x \quad y \quad z]^T\ (m)$	**Orientação** $[\alpha \quad \beta \quad \gamma]^T\ (rad)$
1	Iniciais	$[0{,}462 \quad 0{,}402 \quad 0{,}0639]^T$	$[0{,}762 \quad 2{,}468 \quad 0{,}0409]^T$
	Finais	$[-0{,}625 \quad 0{,}369 \quad 0{,}398]^T$	$[2{,}626 \quad 1{,}820 \quad 0{,}0032]^T$
2	Iniciais	$[-0{,}625 \quad 0{,}369 \quad 0{,}398]^T$	$[2{,}626 \quad 1{,}820 \quad 0{,}0032]^T$
	Finais	$[-0{,}442 \quad -0{,}56 \quad 0{,}439]^T$	$[2{,}758 \quad 1{,}817 \quad 0{,}0360]^T$
3	Iniciais	$[-0{,}548 \quad -0{,}460 \quad 0{,}482]^T$	$[3{,}675 \quad 1{,}634 \quad -3{,}137]^T$
	Finais	$[-0{,}359 \quad 0{,}484 \quad 0{,}458]^T$	$[2{,}622 \quad 1{,}792 \quad -3{,}064]^T$

Fonte: Autor.

A Tab. 22 apresenta os vetores de posição e as dimensões dos obstáculos empregadas associados a cada condição inicial e final da Tab. 21.

Tabela 22: Dimensões e Vetor de posição dos obstáculos utilizados no Experimento 2 associados às Condições Iniciais e Finais.

Condições	**Obstáculos**	**Dimensões**			**Vetor de Posição r_c**		
		w	d	h	x_c	y_c	z_c
1	1	0,240	0,240	0,339	0,125	0,500	0,0797
	2	0,360	0,360	0,382	-0,250	0,438	0,101
2	1	0,144	0,360	0,540	-0,608	-0,125	0,196
3	1	0,252	0,502	0,496	−0,500	0,000	0,260

Fonte: Autor.

Analogamente ao Cap. 5, considerou-se que os obstáculos tinham formatos de prismas retangulares, modelados por elipsoides utilizando as equações em (48). Os valores das constantes dessas equações considerados foram: $k_x = 2{,}0$, $k_y = 2{,}0$ e $k_z = 1{,}414$.

6.3.2. Resultados e Discussões

A Tab. 23 apresenta os Esforços Mecânicos obtidos nas trajetórias com desvio de obstáculos. Como pode ser visto, os melhores resultados em termos de Esforços, para todas as condições iniciais estabelecidas, advieram das trajetórias de polinômios de 4ª ordem.

Tabela 23: Mínimos Esforços Mecânicos obtidos em trajetórias do Experimento 2.

Trajetória	**Condições Iniciais e Finais**		
	1	**2**	**3**
4ª ordem	$663{,}44 \pm 7{,}63$	$506{,}87 \pm 3{,}93$	$511{,}38 \pm 5{,}16$
6ª ordem	$735{,}22 \pm 6{,}57$	$580{,}75 \pm 7{,}27$	$591{,}98 \pm 6{,}46$
8ª ordem	$821{,}04 \pm 6{,}98$	$628{,}55 \pm 4{,}01$	$640{,}70 \pm 5{,}20$
Algoritmo Cap. 4	669,20	817,02	537,31

Fonte: autor.

Todavia, é importante destacar os resultados obtidos pelas trajetórias geradas pelo Algoritmo Cap. 4, por exemplo, como pode ser observado nas Tab. 23, apresentaram valores menores de esforços nas Condições 1 e 3 do que as polinomiais de 6ª e 8ª ordens. Entre os motivos encontrados para explicar isso estão os seguintes:

1) Na existência de obstáculos, o número de possibilidades das curvas polinomiais diminui dependendo da forma como os obstáculos estão distribuídos espacialmente. Por exemplo, na Condição Inicial 1, a posição inicial imposta era próxima das coordenadas do elipsoide, o que limita o número de caminhos concebíveis entre as posições.
2) Os polinômios de 8ª ordem incluem em seus modelos restrições de acelerações e arranques nas extremidades das trajetórias, o que não ocorre na B-Spline cúbica, que assegura somente velocidades nulas. Isso diminui o número de trajetórias admissíveis.
3) Conforme discutido no experimento anterior, as condições iniciais e finais podem favorecer, sob perspectivas energéticas, trajetórias formuladas pelo Algoritmo Cap. 4.

É importante ressaltar que esses bons resultados obtidos das trajetórias do Algoritmo Cap. 4 são ocasionais, uma vez que são dependentes das condições impostas. Como discutido no experimento anterior, essas trajetórias apresentam desvantagens por não assegurarem velocidades, acelerações e arranques suaves nas juntas, o que pode causar desgastes prematuros

nos motores. Por exemplo, na Fig. 48, percebe-se que os torques nas juntas utilizando esse método variam abruptamente, o que não é visivel quando se utiliza métodos de otimização.

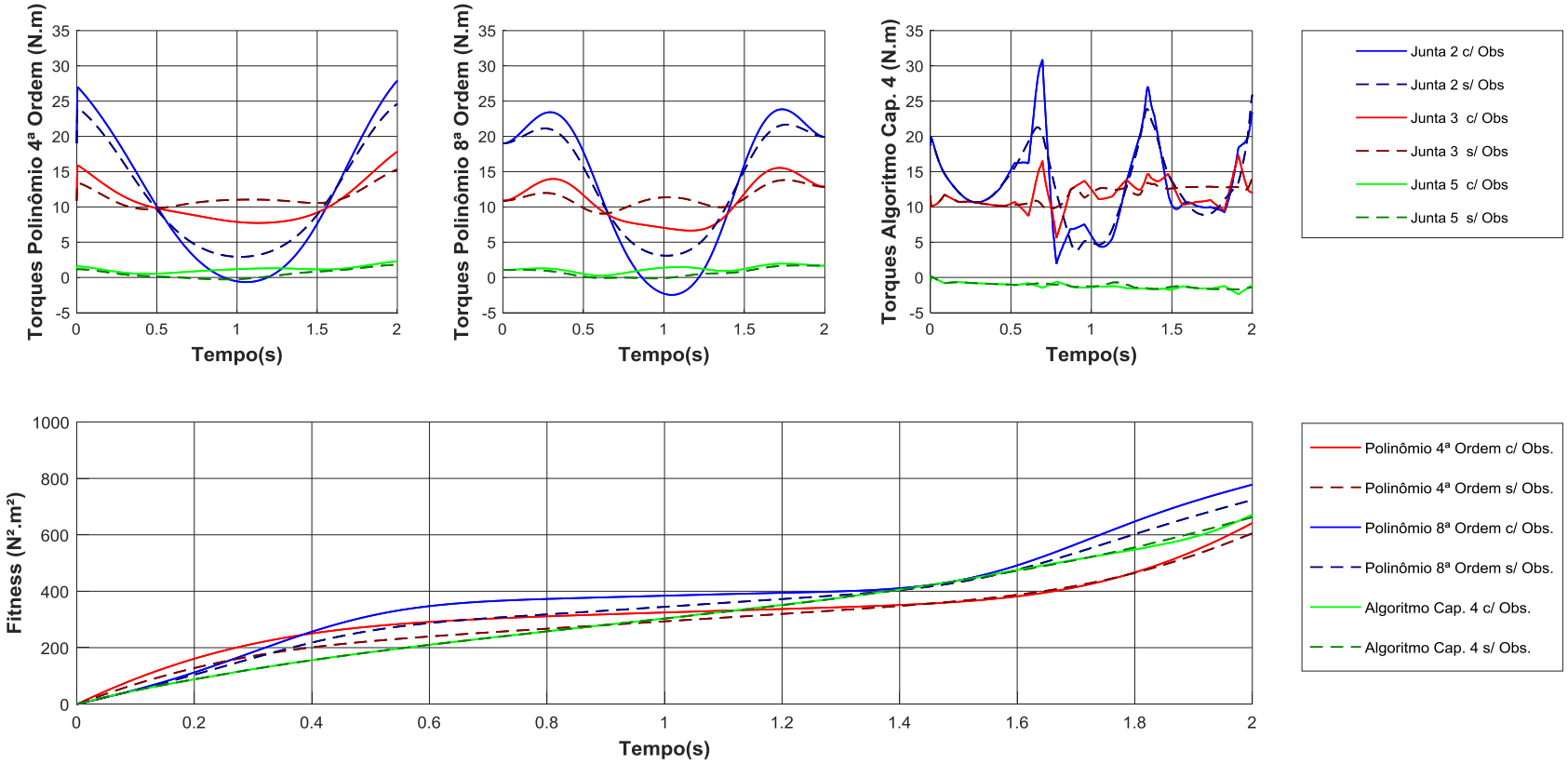
Comparando as trajetórias que foram otimizadas na presença e na ausência de obstáculos, aquelas construídas na ausência deles apresentaram valores menores da função *fitness* após a execução da trajetória, como pode ser visto na Fig. 48. Esse resultado é esperado, pois quando não há obstáculos, existe uma quantidade maior de caminhos possíveis entre as posições, uma vez que o obstáculo é uma restrição espacial. Dessa forma, o caminho ótimo, na ausência de obstáculos, pode incluir posições espaciais em que o obstáculo se encontraria.

Um outro fator que aumentaria a quantidade de caminhos concebíveis é tornar o tempo de execução das trajetórias uma variável de projeto. Nos experimentos realizados esse parâmetro foi mantido constante, mas sendo uma variável poderia ser estabelecido um problema de otimização onde a ordem do polinômio interpolador é fixa.

Também pode ser constatado na Fig. 48 que a segunda junta é a junta com os maiores valores de torque. Isso pois ela é a responsável pela sustentação de todos os elos superiores. Diferentemente, a junta 5 sustenta apenas o útimo elo e o efetuador final, os quais possuem pesos relativamente baixos, por isso os valores de torque nessa junta são mais baixos.

Além disso, pode-se observar que os picos de torque foram diferentes nos métodos utilizados. Na trajetória com obstáculo utilizando o Algoritmo Cap. 4 para a segunda junta, ele acontece em torno de 0,75 segundos, enquanto nas trajetórias otimizadas em torno de 1,1 segundos. Isso aconteceu, pois, as posições do manipulador quando se utiliza o Algoritmo Cap. 4 deve tangenciar o elipsoide do obstáculo. Diferentemente, as posições do manipulador quando se emprega o Algoritmo de Otimização são dependentes dos perfis ocasionados por polinômios, que apresentam concavidades.

Figura 48: Torques nas juntas 2, 3 e 5, e função *fitness* das trajetórias planejadas utilizando Métodos de Otimização e Algoritmo do Capítulo 4.



Fonte: autor.

CAPÍTULO 7: CONSIDERAÇÕES FINAIS

Neste trabalho, realizou-se uma revisão bibliográfica sobre os principais métodos de otimização energética no planejamento de trajetórias de robôs manipuladores. Além disso, dois modelos de simulação foram realizados, os quais empregaram um manipulador de 6 graus de liberdade, o Braço Antropomórfico com Punho Esférico. O primeiro objetivava o planejamento de movimentos do robô, que inclui o projeto de seu sistema de controle e o planejamento de suas trajetórias e tarefas. O segundo tratava o planejamento de trajetórias como um problema de otimização energética.

No primeiro Modelo de Simulação foram resolvidos os problemas de Cinemática Direta e Inversa do Robô. Construiu-se um modelo geométrico e dinâmico através do Simscape™, no qual a resposta foi identificada através do método de mínimos quadrados. Em seguida, utilizando uma estratégia descentralizada, projetou-se o sistema de controle do robô, utilizando as funções de transferência estimadas. O sistema era do tipo PD com filtro, no qual os parâmetros foram sintonizados de maneira a diminuir o índice ITAE, integral do produto do erro absoluto com o tempo. Os controladores projetados apresentaram boas performances e mostraram a possibilidade de melhoria do desempenho do sistema, buscando de forma eficiente e metodológica o cálculo dos diferentes parâmetros do sistema usando as ferramentas apresentadas. Além disso, foram planejadas tarefas que deveriam ser realizadas pelo robô, que eram formadas por um conjunto de trajetórias e incluíam os atos de desviar de obstáculos, escrever e apanhar objetos. Para a execução delas, desenvolveu-se um algoritmo para o planejamento de trajetórias livres de colisões, que possui implementação simples e intuitiva. No algoritmo, os obstáculos são cercados por elipsoides e esferas virtuais e, caso a trajetória esteja em rota de colisão, o efetuador final do robô passa sobre as coordenadas do elipsoide. Utilizou-se as curvas *B-Spline* cúbica para a modelagem matemática da trajetória. Por fim, os métodos foram integrados no Simulink® e validados, o modelo robótico construído no Simscape™ executou as tarefas com eficácia.

O método de planejamento de trajetórias desenvolvido apresentou algumas adversidades. Em ambientes fechados, com muitos obstáculos, os elipsoides podem diminuir demasiadamente o espaço de trabalho livre do manipulador, impossibilitando a geração de curvas. Além disso, dependendo do obstáculo, os elipsoides podem não ser o melhor formato para sua modelagem superficial. Todavia, o método se mostrou efetivo na realização de tarefas em ambientes mais simples e sua implementação é fácil e intuitiva.

No segundo modelo de simulação foi realizado dois experimentos: um considerando a presença de obstáculos e outro não considerando. O planejamento de trajetórias foi visto como um problema de otimização, no qual a função objetivo é o esforço mecânico do manipulador. As trajetórias foram geradas através de polinômios no espaço de configuração. Aplicou-se a otimização nas juntas 2, 3 e 5 do robô. Devido às não-linearidades do problema, empregou-se Algoritmos Genéticos como método de otimização. Notou-se que os valores de esforços mecânicos obtidos dependem do tempo de processamento do AG: quanto maior é o tempo, melhores são as trajetórias obtidas. Constatou-se a existência de um *trade-off* entre a redução do esforço mecânico e do quanto a trajetória pode ser suave, tendo em vista que o esforço mecânico é maior em trajetórias retilíneas e em trajetórias polinomiais que permitem impor restrições nas acelerações e arranques. Além disso, notou-se que os obstáculos diminuem o número de possibilidades de trajetórias concebíveis. Em geral, o método se mostrou eficaz, uma vez que as trajetórias geradas são suaves, concebem esforços mínimos do manipulador e são livres de obstáculos.

Além das aplicações exploradas no trabalho: manipulação de objetos, desvio de obstáculos e pintura, os métodos de planejamento de trajetórias desenvolvidos podem ser aplicados em diversas outras áreas. Algumas aplicações possíveis são: a soldagem e o corte a lazer de peças, a montagem de equipamentos e de circuitos elétricos etc. Inclusive, os métodos podem ser empregados no planejamento de trajetórias de drones, possibilitando-os realizar movimentos autônomos nas áreas de agricultura e pecuária, de construção civil, segurança pública, entre outras.

Ademais, o método utilizado ainda possui diversas alternativas que possibilitam a obtenção de trajetórias melhores. Por exemplo, pode-se modificar a ordem dos polinômios a serem otimizados, incluir o tempo de execução das trajetórias como um parâmetro de projeto, bem como utilizar algoritmos de otimização mais eficientes para tratar os problemas de otimização obtidos. Poderia ser calculado um tempo mínimo para sua execução. Isso permitiria estabelecer o tempo mínimo necessário para executar uma trajetória livre de obstáculos e suave, evitando arranques e descontinuidades que colocam em risco a vida útil dos motores.

7.1 Sugestões para trabalhos futuros

A seguir, são apresentadas algumas sugestões para trabalhos futuros:

- Realizar um Planejamento de Trajetórias que leve em consideração a otimização de múltiplos critérios como, por exemplo, energia, tempo de ciclo e arranque. Esses

critérios devem ser abstraídos matematicamente de forma a retornar o menor custo de produção;

- Investigar a adição do tempo de ciclo como uma variável de projeto, ao invés de utilizá-lo como uma contribuição na função custo, pois essa variável influencia nos perfils de posições, velocidades, acelerações e arranques que serão obtidos, os quais afetam, diretamente, a suavidade e a energia que será consumida;
- Incluir os efeitos de forças atrito no Planejamento de Trajetórias com Eficiência Energética. Considerar as seguintes características de atrito: estático, de Coulomb, viscoso, de arraste, além do atrito de *Stribeck*. Provavelmente, a inclusão dessas características iriam modificar o perfil das trajetórias obtidas. Um exemplo seria nos movimentos com baixas velocidades, nos quais as forças de atrito são mais intensas, devido ao atrito estático. As trajetórias que seriam obtidas, por meio da otimização, ficariam, provavelmente, em menores intervalos de tempo com essas velocidades, nas quais os torques são mais elevados.
- Realizar um planejamento de trajetórias online: capacidade de reação do robô frente a obstáculos dinâmicos. Considerar tarefas onde o robô interage com o ambiente externo: uso de técnica de visão computacional em ambientes reais, modelagem de forças de contato entre o robô manipulador e objetos da tarefa em simulações. Essa sugestão aumentaria o controle e a segurança no ambiente de fábrica;
- Modificar os métodos de otimização e curvas contínuas empregados: utilizar, por exemplo, o SQP na otimização onde as variáveis a serem otimizadas são os parâmetros de uma curva *B-Spline*. Pesquisar sobre os métodos de otimização que estão no estado da arte. Comparar os resultados com os métodos utilizados no presente trabalho;
- Sintonizar o Sistema de Controle utilizando também os critérios energia, tempo de ciclo e arranque do manipulador associado.
- Realizar o Planejamento de Movimento com Eficiência Energética de outros tipos de robôs, por exemplo, de drones, do tipo paralelo, e em robôs que realizam tarefas em cooperação. Pesquisar sobre as técnicas que otimizam o planejamento das tarefas: definir o que seria a sequência ótima para a realização de uma determinada tarefa.

REFERÊNCIAS

[1] ABBU-DAKKA, Fares. 2011. **Trajectory Planning for Industrial Robot Using Genetic Algorithms**. Dissertation (Doctor of Philosophy in Mechanical Engineering) - Universidad Politècnica de València, Valencia, 2011.

[2] ABDOUN, Otman; ABOUCHABAKA, Jaafar. 2011. **A Comparative Study of Adaptive Crossover Operators for Genetic Algorithms to Resolve the Traveling Salesman Problem. 2011.** International Journal of Computer Applications (0975 - 8887) Volume 31 - No.11, October 2011.

[3] ANTÔNIO, Thiago Brada de Almeida. 2014. **Método de controle e detecção de obstáculos para robôs manipuladores aplicado à interação humano-robô**. Dissertação (Mestrado em Engenharia Elétrica) - Programa de Pós-graduação em Engenharia Elétrica, COPPE, UFRJ, Rio de Janeiro, 2014.

[4] ATA, Atef. 2007. **Optimal Trajectory Planning Of Manipulators: A Review**. Journal of Engineering Science and Technology. 2(1), 35-54.

[5] AYTEN, K.; Sahinkaya, M.; Dumlu, A. **Optimum trajectory generation for redundant/hyper-redundant manipulators.** IFAC Proc. Vol. 2016, 49, 493–500.

[6] BAILON, Waldemar; CARDIEL, Edmundo; JUAREZ-CAMPOS, Ignacio; RAMOS-PAZ, Antonio. 2010. **Mechanical energy optimization in trajectory planning for six DOF robot manipulators based on eighth-degree polynomial functions and a genetic algorithm**. IEEE Internet Computing - INTERNET. 446-451. 10.1109/ICEEE.2010.5608583.

[7] BARBARINI, Luiz Henrique Maiorino. **Síntese de Casos de Embarcações Através dos Métodos de Otimização Aplicados a Curvas *B-Spline***. 2007. 142 f. Dissertação de Mestrado em Engenharia Mecatrônica - Escola Politécnica da Universidade de São Paulo, São Paulo, São Paulo.

[8] BEABER, Sameh; ZAGHLOUL, Abdelrahman; KAMEL, Mohamed; HUSSEIN, Wessam. 2018. **Dynamic Modeling and Control of the Hexapod Robot Using Matlab SimMechanics**. 2018. V04AT06A036. 10.1115/IMECE2018-88226.

[9] BELLIS, Mary. 2020. **Who Pioneered Robotics?** ThoughtCo, January, 2020. Disponível em: https://www.thoughtco.com/timeline-of-robots-1992363. Acesso: 07/02/2020.

[10] BOSSCHER, Paul; HEDMAN, Daniel. 2009. **Real-time collision avoidance algorithm for robotic manipulators**. In Proceedings of the IEEE International Conference on Technologies for Practical Robot Applications, pages 113–121, 2009.

[11] CAMPOS, Paulo R. B. **Compensadores**. Universidade Tecnológica Federal do Paraná. Disponível em: http://paginapessoal.utfpr.edu.br/brero/controle_1/material-didatico/5_Compensadores.pdf/view. Acesso: 23/07/2020.

[12] CARABIN, Giovanni; WEHRLE, Erich; VIDONI, Renato. 2017. **A Review on Energy-Saving Optimization Methods for Robotic and Automatic Systems**. Robotics 2017, 6, 39.

[13] CHIDDARWAR, Shital; BABU, N. Ramesh. 2012. **Optimal Trajectory Planning for Industrial Robot Along a Specified Path with Payload Constraint using Trigonometric Splines**. Int. J. of Automation and Control. 6. 39 - 65. 10.1504/IJAAC.2012.045439.

[14] COELHO, Antônio A. R.; COELHO, Leandro dos Santos. 2016. **Identificação De Sistemas Dinâmicos Lineares**. 2. Ed. Ver. - Florianópolis: Ed. da UFSC, 2016.

[15] CORKE, Peter. 2017. **Robotics, Vision and Control: Fundamental Algorithms In MATLAB.** Second Edition (2nd. ed.). Springer Publishing Company, Incorporated.

[16] CRAIG, John J. 2005. **Introduction to robotics: mechanics and control.** Upper Saddle River, NJ: Pearson, Prentice Hall.

[17] DA SILVA, Amanda Maria. 2014. **A Representação das Matrizes de Rotações Com o Uso dos Quatérnios: Aplicações à Fotogrametria**. Dissertação (Mestrado em Ciências Geodésicas e Tecnologias da Geoinformação) - Centro de Tecnologia e Geociências, UFPE, Recife, 2014.

[18] DURAND, Frédo; CUTLER, Barb. **Curves & Surfaces**. Computer Graphics - Lecture Slides, Slides for Computer Science. Disponível em: https://slideplayer.com/slide/4635393/. Data de acesso: 17/09/2019.

[19] FEIFEI, Liu; FEI, Lin. 2016. **Time-Jerk Optimal Planning of Industrial Robot Trajectories**. International Journal of Robotics and Automation. 31. 10.2316/Journal.206.2016.1.206-4055.

[20] FU, King-Su.; GONZALEZ; Rafael C.; LEE, C.. S. G. 1987. **Robotics: control, sensing, vision, and intelligence.** New York: McGraw-Hill.

[21] FUNG, Rong-Fong; CHENG, Yi-Hsin. 2011. **A Minimum-Energy-Based High-Degree Polynomial Trajectory Planning and Tracking Control for an LCD Glass-handling Robot**. International Journal of Intelligent Engineering and Systems. 4. 10.1109/ICINIS.2011.28.

[22] GAMERO, Isis. 2018. **Robôs industriais: tudo que você precisa saber!** Pollux Blog. Disponível em: https://www.pollux.com.br/blog/robos-industriais-tudo-o-que-voce-precisa-saber/. Acesso: 07/02/2020.

[23] GIRDHAR, Yogesh. 2005. **Efficient Sampling of Protein Folding Funnels Using Hmmstr and Pathway Generation Using Probabilistic Roadmaps**. Thesis (Master of Science in Computer Science) - Rensselaer Polytechnic Institute, Troy, New York, April, 2005.

[24] HENTOUT, Abdelfetah; BOUZOUIA, Brahim; TOUKAL, Zakaria. 2008. **Trajectories Planning in Presence of Obstacles for Manipulator Robots**. 421 - 426. 10.1109/AMS.2008.64.

[25] HOLLAND, John H. **Genetic Algorithms**. 1992. Scientific American 267, no. 1 (1992): 66-73. Disponível em: https://www.jstor.org/stable/24939139?seq=1. Acesso: 07/02/2020.

[26] HOLLAND, J.H. 1975. **Adaptation in Natural and Artificial Systems**. University of Michigan Press, Ann Arbor. (2nd Edition, MIT Press, 1992).

[27] JEBARI, Khalid. 2013. **Parent Selection Operators for Genetic Algorithms**. International Journal of Engineering Research & Technology. 12. 1141-1145.

[28] KHATIB, Okashiro. 1985. **Real-time obstacle avoidance for manipulators and mobile robots.** IEEE International Conference on Robotics and Automation, St. Louis, MO, USA, 1985, pp. 500-505.

[29] KIM, Joonyoung; KIM, Dong-Hyeok; KIM, Sung-Rak. 2007. **On-line minimum-time trajectory planning for industrial manipulators.** ICCAS 2007 - International Conference on Control, Automation and Systems. 36 - 40. 10.1109/ICCAS.2007.4406875.

[30] LLOPIS-ALBERT, Carlos; RUBIO, Francisco; VALELO, Francisco. 2018. **Optimization approaches for robot trajectory planning.** Multidisciplinary Journal for Education, Social and Technological Sciences. 5. 1. 10.4995/muse.2018.9867.

[31] LOZANO-PÉREZ, Tomás. 1983. **Robot Programming**. Proceedings of the IEEE, Vol 71, No. 7, 821 – 841, July 1983.

[32] MAJEED, Abdul; LEE, Sungchang. 2018. **A Fast-Global Flight Path Planning Algorithm Based on Space Circumscription and Sparse Visibility Graph for Unmanned Aerial Vehicle**. Electronics. 7. 375. 10.3390/electronics7120375.

[33] MALONE, Bob. 2011. **George Devol: A Life Devoted to Invention, and Robots**. IEEE Spectrum. Disponivel em: https://spectrum.ieee.org/automaton/robotics/industrial-robots/george-devol-a-life-devoted-to-invention-and-robots. Acesso: 01/10/2020.

[34] MARTINEZ-ALFARO, Horacio. 1993. **Collision-free path planning for robots using B-splines and simulated annealing.** Dissertation (Doctor of Philosophy in Mechanical Engineering) - Iowa State University, Ames, Iowa, 1993.

[35] MASEHIAN, Ellips; SEDIGHIZADEH, D. 2010. **Multi-objective robot motion planning using a particle swarm optimization model.** Journal of Zhejiang University - Science C. 11. 607-619. 10.1631/jzus.C0910525.

[36] MATHWORKS®. **Simscape Multibody: Model and simulate multibody mechanical systems.** Disponível em: https://www.mathworks.com/products/simmechanics.html. Acesso: 15/02/2021.

[37] MENDES, Jorge. 2013. **A comparative study of crossover operators for genetic algorithms to solve the job shop scheduling problem**. WSEAS Transactions on Computers. 12. 164-173.

[38] MICHALEWICZ, Zbigniew. 1996. **Genetic algorithms + data structures = evolution programs** (3rd ed.). Springer-Verlag, Berlin, Heidelberg.

[39] MILJKOVIC, Zoran; PETROVIC, Milica. 2017. **Application of modified multi-objective particle swarm optimisation algorithm for flexible process planning problem**. Int. J. Computer Integrated Manufacturing, 30, 271-291.

[40] MONDRAGON, Carlos Arturo Sanchez. 2012. **Dynamics and Motion of a Six Degree of Freedom Robot Manipulator.** Thesis (Master of Science in Mechanical Engineering) - University of Saskatchewan, Saskatoon, December, 2012.

[41] MORTENSON, Michael E. 1997. **Geometric modeling** (2nd ed.). John Wiley & Sons, Inc., USA.

[42] MULIK, P. B. 2015. **Optimal trajectory planning of industrial robot with evolutionary algorithm.** International Conference on Computation of Power, Energy, Information and Communication (ICCPEIC) (2015): 0256-0263.

[43] NISE, Norman. S. 2002. **Engenharia de Sistemas de Controle**, 3ª Edição, LTC, 2002.

[44] OGATA, Katsuhiko. 2001. **Modern Control Engineering** (4th. ed.). Prentice Hall PTR, USA.

[45] PALMIERI, Luigi. 2015. **A Behavioural Approach to Obstacle Avoidance for Mobile Manipulators Based on Distributed Sensing**. Publicado no ArXiv 2015. Disponível em: https://arxiv.org/abs/1502.02465. Acesso: 07/02/2020.

[46] PERVIER, Hugo; NALIANDA, Devaiah; ESPI, Ramon; SETHI, Vishal; PILIDIS, Pericles; ZAMMIT-MANGION, David; ROGERO, Jean-Michel; ENTZ, Ricardo. 2011. **Application of Genetic Algorithm for Preliminary Trajectory Optimization**. SAE International Journal of Aerospace. 4. 973-987. 10.4271/2011-01-2594.

[47] PETROVIC, Luka. 2018. **Motion Planning in High-Dimensional Spaces**. Publicado no ArXiv 2018. Disponível em: https://arxiv.org/abs/1806.07457. Acesso: 07/02/2020.

[48] POLVERINI, Matteo; ZANCHETTIN, Andrea Maria; ROCCO, Paolo. 2014. **Real-Time Collision Avoidance in Human-Robot Interaction Based on Kinetostatic Safety Field**. IEEE/RSJ International Conference on Intelligent Robots and Systems, Chicago, IL, 2014, pp. 4136-4141.

[49] RATIU, Mariana; PRICHICI, Adriana. 2017. **Industrial Robot Trajectory Optimization - A Review.** MATEC Web Conf. 126 02005.

[50] ROCHA, Carlos; TONETTO, Cristiane; DIAS, Altamir. 2011. **A comparison between the Denavit-Hartenberg and the screw-based methods used in kinematic modeling of robot manipulators.** Robotics and Computer-integrated Manufacturing - Robot Comput-Integr Manuf. 27. 723-728. 10.1016/j.rcim.2010.12.009.

[51] SAINI, N. 2017. **Review of Selection Methods in Genetic Algorithms. International Journal of Engineering and Computer Science**, 6(12), 22261-22263. Disponível em: http://www.ijecs.in/index.php/ijecs/article/view/2562. Data de Acesso: 07/02/2020.

[52] SALOMON, David. 2005. **Curves and Surfaces for Computer Graphics**. Springer-Verlag, Berlin, Heidelberg.

[53] SARAVANAN, R.; RAMABALAN, S.; BALAMURUGAN, C. 2008. **Evolutionary optimal trajectory planning for industrial robot with payload constraints**. Int J Adv Manuf Technol (2008) 38: 1213. https://doi.org/10.1007/s00170-007-1169-7.

[54] SCIENCELEARN. 2011. **DNA, chromosomes and cells**. Disponível em: https://www.sciencelearn.org.nz/images/198-dna-chromosomes-and-cells. Data de Acesso: 07/02/2020.

[55] SEBASTIAN, Castro. 2019. Trajectory Planning for Robot Manipulators. MathWorks®. Disponível em: https://www.mathworks.com/videos/trajectory-planning-for-robot-manipulators-1556705635398.html. Acesso: 19/09/2020.

[56] SEBORG, D. E.; Edgar, T F.; Mellichamp, D. A. 2004. **Process Dynamics and Control**, 2ª Edition, John Wiley Andamp; Sons, New York, 2004.

[57] SERAPIÃO, André de Freitas. 2018. **Controle de Robôs em Configurações Singulares**. Relatório anual submetido ao XXVI Seminário de Iniciação Científica e Tecnológica da PUC-RIO.

[58] SICILIANO, Bruno; SCIAVICCO Lorenzo; VILLANI, Luigi; ORIOLO, Giuseppe. 2008. **Robotics: Modelling, Planning and Control** (1st. ed.). Springer Publishing Company, Incorporated.

[59] SIMAS, Henrique. 2008. **Planejamento de trajetórias e evitamento de colisão em tarefas de manipuladores redundantes operando em ambientes confinados.** 2008. Tese (Doutorado em Engenharia Mecânica) - Programa de Pós-graduação em Engenharia Mecânica, PosMEC, UFSC, Florianópolis, 2008.

[60] SPONG, Mark W., Hutchinson, Seth, and Vidyasagar, M. 2004. **Robot Dynamics and Control**. Second Edition (2nd. ed.).

[61] SU, Bu-qing; LIU, Ding-zhe. 1989. **Computational geometry: curve and surface modeling.** Academic Press Professional, Inc., USA.

[62] SZÁNTÓ, András; HAJDU, Sándor. 2019. **Vehicle Modelling and Simulation in Simulink. International Journal of Engineering and Management Sciences**. 2019. 4. 260-265. 10.21791/IJEMS.2019.1.33.

[63] TOMEI, Patrizio. 1991. **A Simple PD Controller for Robots with Elastic Joints**. IEEE Transactions on Automatic Control. Vol. 36. No. 10, October 1991.

[64] TSAI, Lung-Wen. 1999. **Robot Analysis and Design: The Mechanics of Serial and Parallel Manipulators.** (1st. ed.). John Wiley & Sons, Inc., USA.

[65] YANG, Chifu; YE, Zhengmao; OGBOBE, Peter; HAN, Junwei. 2010. **Modeling and simulation of spatial 6-DOF parallel robots using Simulink and SimMechanics.** Proceedings - 2010 3rd IEEE International Conference on Computer Science and Information Technology, ICCSIT 2010. 4. 444 - 448. 10.1109/ICCSIT.2010.5563824.

[66] WIKIPEDIA COMMONS. 2016. **Meiose Overview.** Disponível em: https://commons.wikimedia.org/wiki/File:Meiosis_Overview_new.svg. Acesso 28/07/2020.

[67] WILSON, David G.; ROBINETT, Rush R., III; EILSER, G. Richard. **Discrete dynamic programming for optimized path planning of flexible robots**. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Sendai, Japan, 28 September–2 October 2004; Volume 3, pp. 2918–2923.

[68] ZHANG, Jiayan; MENG, Qingxi; FENG, Xugang; SHEN, Hao. 2018. **A 6-DOF robot-time optimal trajectory planning based on an improved genetic algorithm**. Robotics and Biomimetics. 5. 10.1186/s40638-018-0085-7.

[69] ZHAO, R., 2015. **Trajectory planning and control for robot manipulations**. Tese de Doutorado. Robotics [cs.RO]. Université Paul Sabatier - Toulouse III. English. <NNT : 2015TOU30240>.

APÊNDICE A: PARÂMETROS UTILIZADOS NOS MODELOS CINEMÁTICO E DINÂMICO DO BRAÇO ANTROPOMÓRFICO COM PUNHO ESFÉRICO

Neste Apêndice, são expostos os parâmetros utilizados nos modelos cinemático e dinâmico do Robô Antropomórfico com Punho Esférico. No modelo dinâmico, considerou-se um valor de gravidade equivalente a 9,81 m/s^2. A Tab. 24 contém os Parâmetros da geometria do manipulador. As Tab. 25, 26 e 27 contém os Parâmetros utilizados no Modelo Dinâmico.

Tabela 24: Parâmetros Geométricos do Manipulador.

L_1	L_2	L_3	L_4
0,360	0,155	0,365	0,250

Fonte: autor.

Tabela 25: Massas e coeficientes do tensor de inércia dos elos 1, 2 e 3 do manipulador.

Elo	**1**	**2**	**3**
$\boldsymbol{m\ (Kg)}$	11,00	3,03	1,95
$\boldsymbol{I_{XX}\ (kg \cdot m^2)}$	$1{,}30 \cdot 10^{-1}$	$7{,}81 \cdot 10^{-3}$	$1{,}01 \cdot 10^{-2}$
$\boldsymbol{I_{YY}\ (kg \cdot m^2)}$	$1{,}30 \cdot 10^{-1}$	$7{,}81 \cdot 10^{-3}$	$1{,}01 \cdot 10^{-2}$
$\boldsymbol{I_{ZZ}\ (kg \cdot m^2)}$	$2{,}00 \cdot 10^{-2}$	$3{,}49 \cdot 10^{-3}$	$9{,}19 \cdot 10^{-4}$
$\boldsymbol{I_{XY}\ (kg \cdot m^2)}$	0,00	0,00	0,00
$\boldsymbol{I_{ZX}\ (kg \cdot m^2)}$	0,00	0,00	0,00
$\boldsymbol{I_{YZ}\ (kg \cdot m^2)}$	0,00	0,00	0,00

Fonte: autor.

Tabela 26: Massas e coeficientes do tensor de inércia dos elos 4, 5 e 6 do manipulador.

Elo	**4**	**5**	**6**
$\boldsymbol{m\ (Kg)}$	0,97	0,97	0,68
$\boldsymbol{I_{XX}\ (kg \cdot m^2)}$	$1{,}43 \cdot 10^{-3}$	$1{,}43 \cdot 10^{-3}$	$5{,}79 \cdot 10^{-4}$
$\boldsymbol{I_{YY}\ (kg \cdot m^2)}$	$1{,}43 \cdot 10^{-3}$	$1{,}43 \cdot 10^{-3}$	$5{,}79 \cdot 10^{-4}$
$\boldsymbol{I_{ZZ}\ (kg \cdot m^2)}$	$4{,}60 \cdot 10^{-4}$	$4{,}60 \cdot 10^{-4}$	$3{,}23 \cdot 10^{-4}$
$\boldsymbol{I_{XY}\ (kg \cdot m^2)}$	0,00	0,00	0,00
$\boldsymbol{I_{ZX}\ (kg \cdot m^2)}$	0,00	0,00	0,00
$\boldsymbol{I_{YZ}\ (kg \cdot m^2)}$	0,00	0,00	0,00

Fonte: autor.

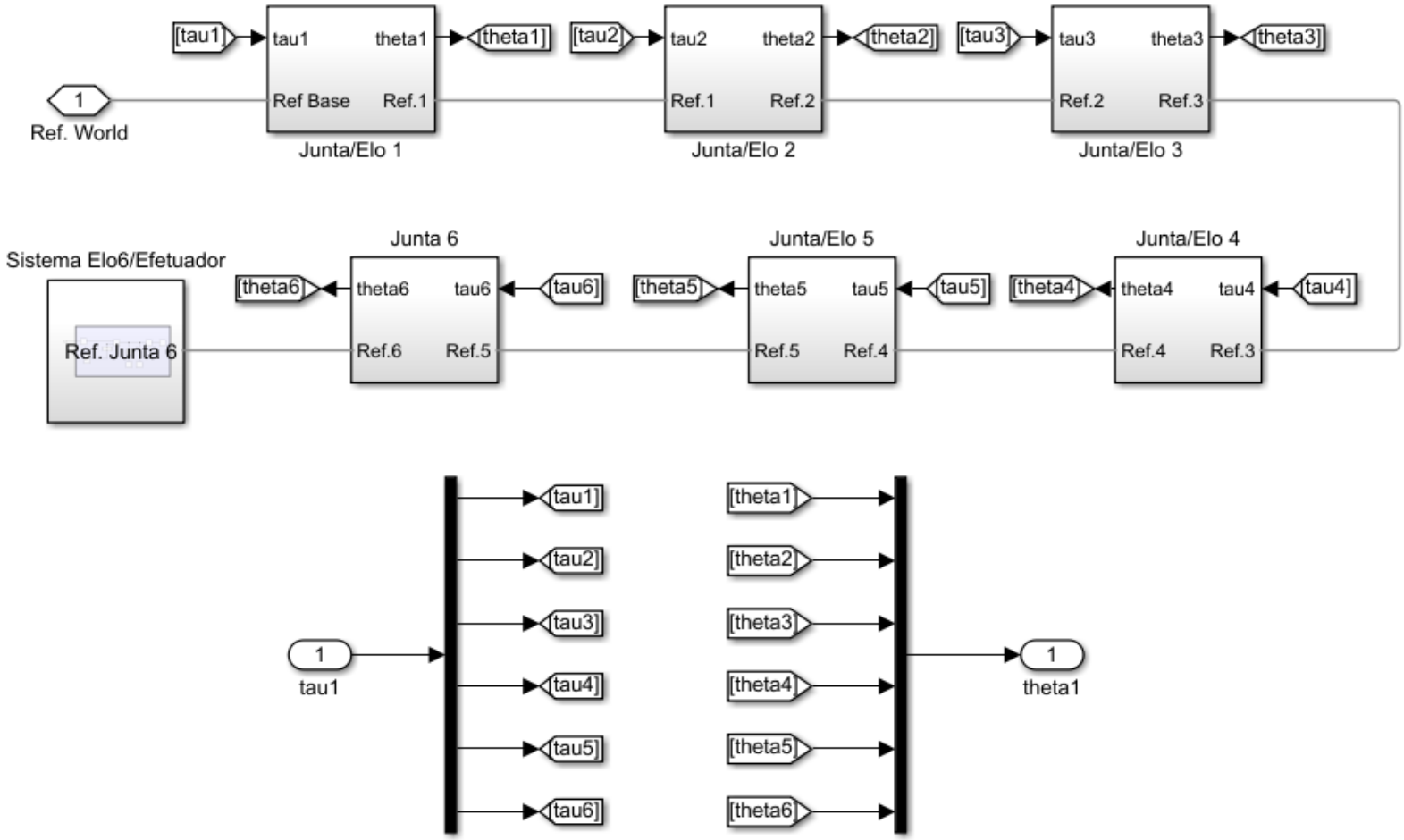
Tabela 27: Vetores que representam as posições do centro de massa dos elos e das juntas.

$\boldsymbol{r_{0,1}^{1}}$	$\boldsymbol{r_{1,2}^{2}}$	$\boldsymbol{r_{2,3}^{3}}$	$\boldsymbol{r_{3,4}^{4}}$
$[0 \quad L_1 \quad 0]^T$	$[L_2 \quad 0 \quad 0]^T$	$[0 \quad 0 \quad 0]^T$	$[0 \quad L_3 \quad 0]^T$
$\boldsymbol{r_{4,5}^{5}}$	$\boldsymbol{r_{5,6}^{6}}$	$\boldsymbol{r_{1,c_1}^{1}}$	$\boldsymbol{r_{2,c_2}^{2}}$
$[0 \quad 0 \quad 0]^T$	$[0 \quad 0 \quad L_4]^T$	$[0 \quad -L_1/2 \quad 0]^T$	$[-L_2/2 \quad 0 \quad 0]^T$
$\boldsymbol{r_{3,c_3}^{3}}$	$\boldsymbol{r_{4,c_4}^{4}}$	$\boldsymbol{r_{5,c_5}^{5}}$	$\boldsymbol{r_{6,c_6}^{6}}$
$[0 \quad 0 \quad L_3/3]^T$	$[0 \quad -L_3/6 \quad 0]^T$	$[0 \quad 0 \quad L_3/6]^T$	$[0 \quad (L_4 - 4L_3/3)/3 \quad 0]^T$

Fonte: autor.

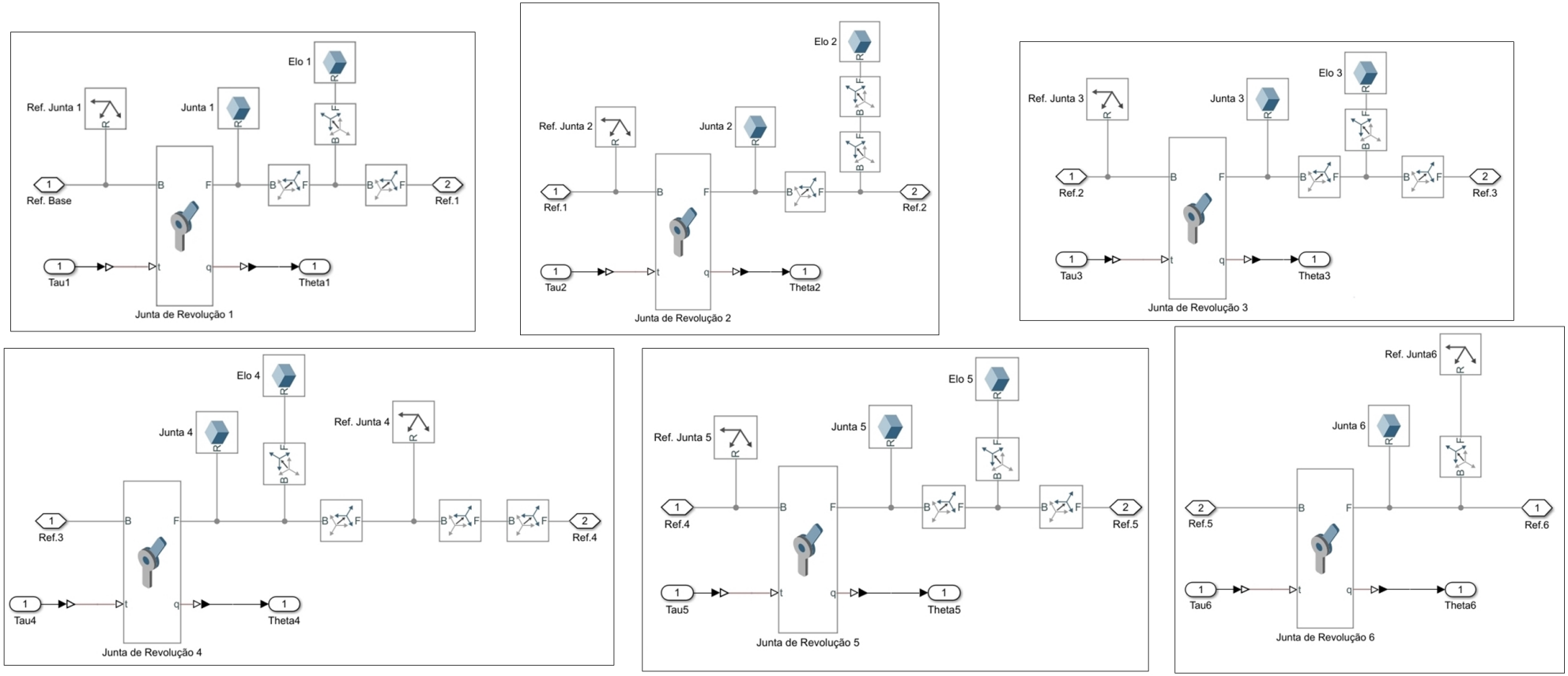
As Figuras em sequência são os modelos geométricos implementados no Sincape™/Simulink®. A Fig. 49 é o modelo geométrico do robô superficialmente, que só mostra a relação entre elos dele. Enquanto a Fig. 50 revela os detalhes de construção de cada bloco do modelo da Fig. 49: as relações entre juntas e os elos, e os blocos que constroem os objetos geometricamente.

Figura 49: Modelo Geométrico do Braço Antropomórfico com Punho Esférico implementado no Simulink®.



Fonte: autor.

Figura 50: Diagrama de blocos de cada Junta do Braço Antropomórfico com Punho Esférico.



Fonte: autor.

APÊNDICE B: ORIENTAÇÃO DE CORPOS RÍGIDOS, CINEMÁTICA INVERSA E SINGULARIDADES DO BRAÇO ANTROPOMORFICO COM PUNHO ESFÉRICO

- **Orientação de Corpos Rígidos**

As maneiras mais empregadas para descrição matemática da orientação de corpos rígudos são através de matrizes de rotação (por exemplo, utilizando Ângulos de Euler) e Quatérnions. Nos ângulos de Euler, a orientação de um referencial $O_1x_1y_1z_1$ em relação ao referencial $O_0x_0y_0z_0$ pode ser especificada através de três ângulos, $\phi = [\alpha \quad \beta \quad \gamma]^T$, que representam os ângulos de uma sequência de três rotações elementares sucessivas. Tomando como exemplo as rotações nos eixos ZYZ: 1ª rotação no referencial $O_1x_1y_1z_1$ por um ângulo α sobre o eixo z; 2ª rotação no referencial atual por um ângulo β sobre o eixo y'; 3ª rotação no referencial atual por um ângulo γ sobre o eixo z''. A orientação do referencial resultante é obtida pela multiplicação sucessiva das matrizes de rotação elementares. A matriz resultante é expressa conforme a Eq. (61) (Siciliano *et al*., 2008).

$$R(\phi) = \begin{bmatrix} c\alpha c\beta c\gamma - s\alpha s\gamma & -c\alpha c\beta s\gamma - s\alpha c\gamma & c\alpha s\beta \\ s\alpha c\beta c\gamma + c\alpha s\gamma & -s\alpha c\beta s\gamma + c\alpha c\gamma & s\alpha s\beta \\ -s\beta c\gamma & s\beta s\gamma & c\beta \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \qquad (61)$$

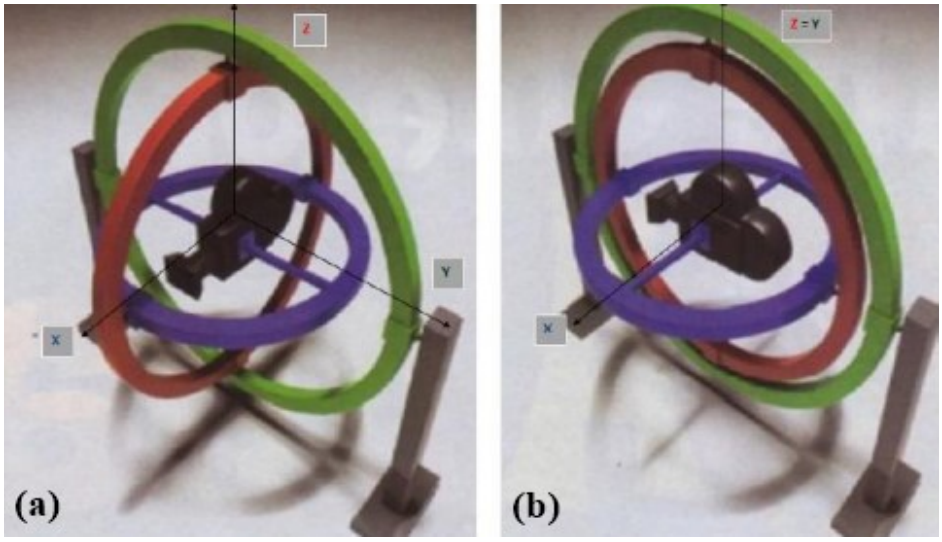
Onde $c\alpha$, $c\beta$, $c\gamma$ e $s\alpha$, $s\beta$, $s\gamma$ são as funções cosseno e seno dos ângulos α, β e γ, respectivamente, e r_{ij} são os elementos da matriz de rotação. Logo, os ângulos podem ser calculados através da Eq. (62), na qual $atan2$ é usada para encontrar o ângulo nos quatro quadrantes.

$$\alpha = atan2(r_{23}, r_{13}); \quad \beta = atan2\left(\sqrt{r_{32}^2 + r_{31}^2}, r_{33}\right); \quad \gamma = atan2(r_{32}, -r_{31}) \qquad (62)$$

A utilização de Ângulos de Euler apresenta duas desvantagens: problema do *Gimbal Lock* e ambiguidade nos ângulos. O *Gimbal Lock* acontece quando duas das rotações são realizadas sobre um eixo paralelo. Neste caso, o sistema perde um grau de liberdade (Fig. 51). No caso das ambiguidades, pode acontecer de dois valores de ângulos satisfazerem uma mesma

solução. O segundo problema pode ser superado estudando os sinais da rotação intermediária e utilizando outras equações na obtenção dos ângulos (Da Silva, 2014).

Figura 51: Perda de grau de liberade ao orientar uma câmera. (a) objeto com três graus de liberdade, (b) objeto com dois graus de liberdade.



Fonte: Da Silva (2014).

Da Silva (2014) lista as vantagens e desvantagens do uso de Quatérnions em relação aos Ângulos de Euler na representação da orientação. As vantagens incluem: não acontece o problema de instabilidade das soluções, podem ser compostos ou multiplicados de uma maneira simples para acumular os efeitos de rotação, possui métodos de interpolação mais simples e não acarretam o *Gimbal lock*. Todavia, as desvantagens são: não apresentam boas representações de translação e necessitam de um aprendizado maior para a sua aplicação.

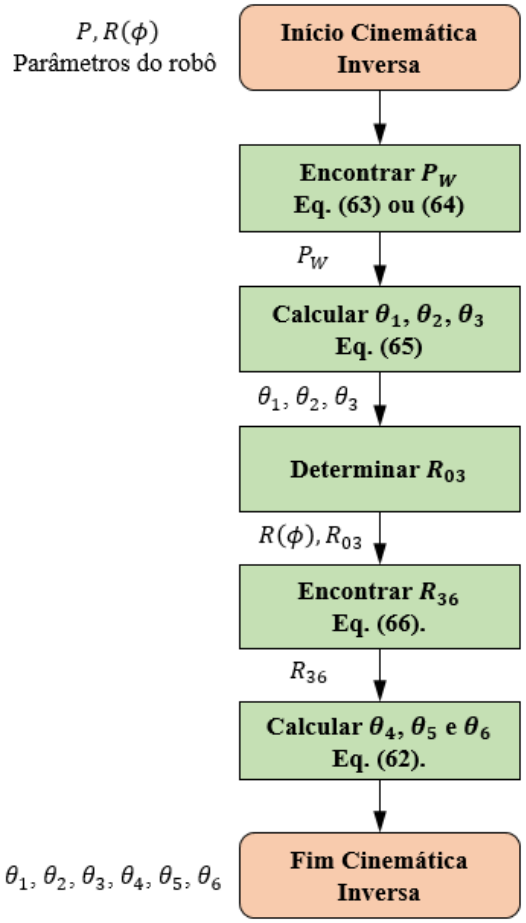
- **Cinemática Inversa**

A Cinemática Inversa se baseia na obtenção das variáveis de junta quando são conhecidas as variáveis do espaço de trabalho bem como os parâmetros do robô. O manipulador utilizado nos Modelos de Simulação deste trabalho foi o Braço Antropomórfico com Punho Esférico. Neste robô, a solução da Cinemática Inversa pode ser construída a partir da sua Cinemática Direta. Dessarte, é necessário considerar algum método para sistematizar as

transformações espaciais. Para isso, pode-se empregar tanto a Teoria de Helicoides quanto a Notação de Denavit-Hartenberg.

A solução da Cinemática Inversa acontece de uma forma “desacoplada”, conforme a Fig. 52. O problema é dividido em duas etapas: obtenção dos ângulos do cotovelo, através da Cinemática Inversa de posição, e do punho, empregando a Cinemática Inversa de orientação.

Figura 52: Algoritmo de solução da Cinemática Inversa.



Fonte: autor.

Primeiramente, resolve-se o problema de Cinemática Direta, obtêm-se as matrizes de rotação através do método desejado. Em seguida, calcula-se as coordenadas do centro do punho esférico, $P_W = [P_{Wx} \quad P_{Wy} \quad P_{Wz}]^T$. O cálculo depende do método de cálculo das transformações espaciais. Quando se utiliza a Teoria de Helicoides, a obtenção das coordenadas

ocorre conforme a Eq. (63). Onde $R(\phi)$ é a matriz de orientação, P o vetor de posição do efetuador final desejado e L_4 é o comprimento entre o punho e o efetuador final.

$$\mathrm{P_W} = P - L_4 R(\phi)[1 \quad 0 \quad 0]^T \tag{63}$$

Quando se emprega o Método de Denavit Hartenberg a obtenção dessas coordenadas ocorre conforme a Eq. (64).

$$\mathrm{P_W} = P - L_4 R(\phi)[0 \quad 0 \quad 1]^T \tag{64}$$

Posteriormente, calcula-se os ângulos θ_1, θ_2 e θ_3 partindo da análise geométrica no cotovelo ou resolvendo as equações obtidas após da solução da Cinemática Direta. As Equações decorrentes após tais análises/cálculos são representadas em (65) para os ângulos θ_1, θ_3 e θ_2, respectivamente.

$$\theta_1 = atan2\left(\pm\frac{P_{Wy}}{P_{Wx}}\right)$$

$$\theta_3 = \pm\cos\left(\frac{P_{Wx}^2 + P_{Wy}^2 + (P_{Wz} - L_1)^2 - L_2^2 - L_3^2}{2L_2L_3}\right)$$

$$\sin\theta_2 = \pm\frac{(L_2 + L_3\cos\theta_3)(P_{Wz} - L_1) - L_3\sin\theta_3\sqrt{P_{Wx}^2 + P_{Wy}^2}}{P_{Wx}^2 + P_{Wy}^2 + (P_{Wz} - L_1)^2} \tag{65}$$

$$\cos\theta_2 = \pm\frac{(L_2 + L_3\cos\theta_3)\sqrt{P_{Wx}^2 + P_{Wy}^2} + L_3\sin\theta_3\,(P_{Wz} - L_1)}{P_{Wx}^2 + P_{Wy}^2 + (P_{Wz} - L_1)^2}$$

$$\theta_2 = atan2\left(\frac{sin\theta_2}{cos\theta_2}\right)$$

Com esses ângulos é possível encontrar a matriz de rotação $R_3^0 = A_1A_2A_3$ utilizando o algum dos métodos de transformação espacial e, com isso, calcular R_6^3 através da Eq. (66).

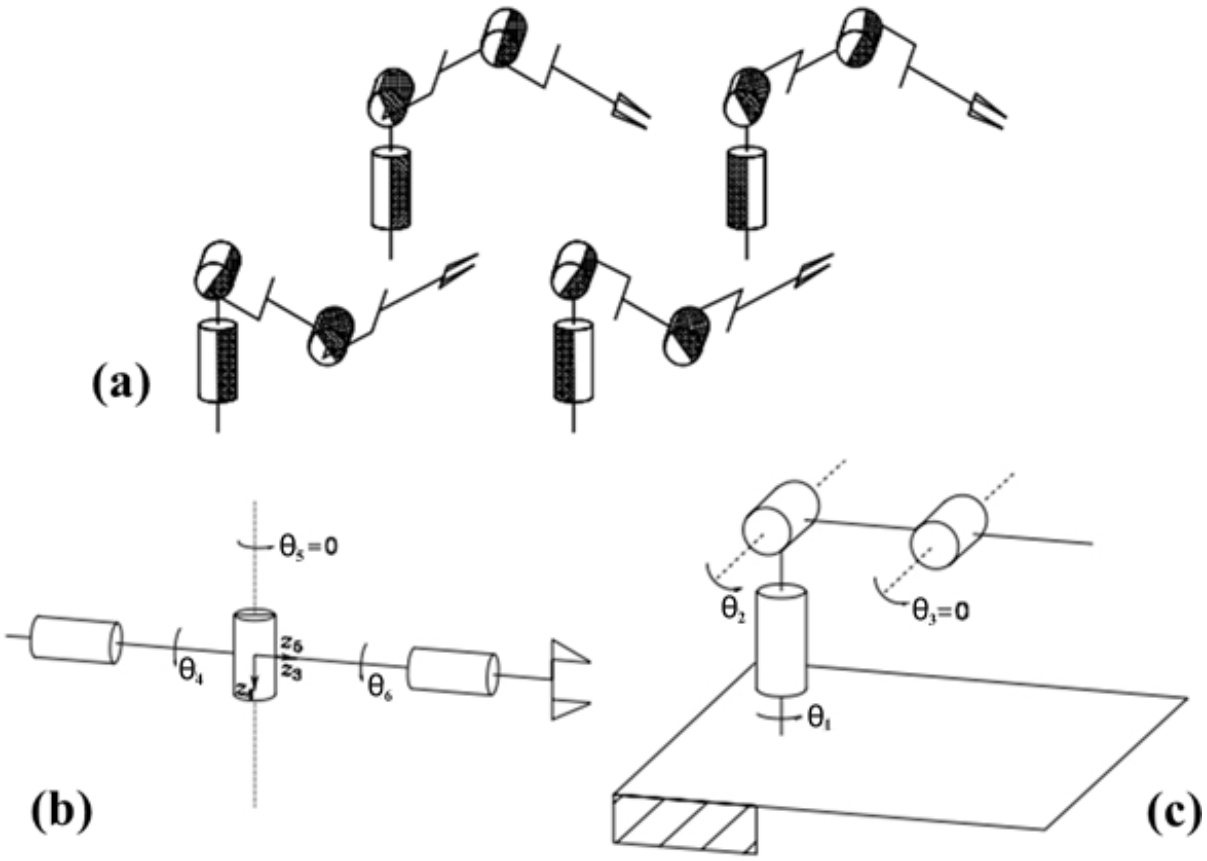
$$R_6^3 = (R_3^0)^{-1}R \tag{66}$$

O último passo é calcular os ângulos θ_4, θ_5 e θ_6 através da matriz R_6^3, que são os ângulos do punho esférico. Isso pode ser feito utilizando as equações para cálculo dos ângulos de Euler em (62), pois o punho representa a orientação do robô manipulador.

- **Singularidades**

Como mostrado por Siciliano *et al*. (2008), existem diversas soluções para a Cinemática Inversa do cotovelo, devido ao manipulador possuir diversas configurações, como pode ser visto na Fig 53 (a). Isso acontece devido ao fato dos valores dos ângulos θ_1, θ_2 e θ_3 dependerem dos sinais, que podem ser positivos ou negativos.

Figura 53: (a) As quatro possíveis configurações para o braço antropomórfico, (b) exemplo de Singularidade no Punho Esférico, (c) exemplo de Singularidade no Cotovelo.



Fonte: Adaptado de Siciliano *et al*. (2009).

Além disso, as soluções só podem ser encontradas caso $P_{Wx} \neq 0$ ou $P_{Wy} \neq 0$, porque nessas condições ocorrem Configurações Singulares, que são aquelas em que a mobilidade da estrutura é reduzida: um ponto no espaço de trabalho do robô que a matriz jacobiana perde *rank*. Tais Singularidades Cinemáticas são indesejadas no Planejamento de Trajetórias, porque impossibilitam o efetuador de se mover em certas direções e acontece quando duas das juntas do robô se alinham.

As singularidades do manipulador elucidado podem ser divididas em dois tipos: Singularidades do Punho (*Wrist Singularities*) e Singularidades do Cotovelo (*Elbow Singularities*). As Singularidades do Punho ocorrem quando $\theta_5 = 0$ (os eixos das juntas z_3 e z_5 são colineares), conforme a Fig 53 (b). As Singularidades do Cotovelo acontecem quando $\theta_3 = 0$, Fig 53 (c), $\theta_3 = \pi$ ou para qualquer $L_2 \cos\theta_2 + L_3 \cos\theta_2 + \theta_3 = 0$.

APÊNDICE C: DADOS UTILIZADOS PARA GERAR VISUALIZAÇÃO GRÁFICA DAS TRAJETÓRIAS DO MODELO DE SIMULAÇÃO "OTIMIZAÇÃO ENERGÉTICA NO PLANEJAMENTO DE TRAJETÓRIAS"

As Tab. 28, 29 e 30 apresentam os valores dos coeficientes utilizados em polinômios de 4ª, 6ª e 8ª ordens, respectivamente, para geração dos gráficos apresentados nos resultados da otimização energética **sem** o desvio de obstáculos.

Tabela 28: Valores dos coeficientes utilizados na simulação da trajetória do polinômio de 4ª ordem Experimento 1.

Condições	Junta 2	Junta 3	Junta 5
1	1,3309	−1,8809	0,9317
2	0,7417	−1,4425	1,0747
3	0,9379	−1,7097	0,8920

Fonte: autor.

Tabela 29: Valores dos coeficientes utilizados na simulação da trajetória do polinômio de 6ª ordem Experimento 1.

Condições	Junta 2	Junta 3	Junta 5
1	−1,9498	1,8571	0,7925
2	−1,7319	1,6882	0,8743
3	−1,3483	1,9647	−1,3968

Fonte: autor.

Tabela 30: Valores dos coeficientes utilizados na simulação da trajetória do polinômio de 8ª ordem Experimento 1.

Condições	Junta 2	Junta 3	Junta 5
1	1,8045	−1,9186	−0,2718
2	1,1448	−1,6081	1,1696
3	1,8520	−1,8515	−1,1374

Fonte: autor.

As Tab. 31, 32 e 33 apresentam os valores dos coeficientes utilizados em polinômios de 4ª, 6ª e 8ª ordens, respectivamente, para geração dos gráficos **com** o desvio de obstáculos.

Tabela 31: Valores dos coeficientes utilizados na simulação da trajetória do polinômio de 4ª ordem Experimento 2.

Condições	Junta 2	Junta 3	Junta 5
1	1,3309	−1,8809	0,9317
2	0,7417	−1,4425	1,0747
3	0,9379	−1,7097	0,8920

Fonte: autor.

Tabela 32: Valores dos coeficientes utilizados na simulação da trajetória do polinômio de 6ª ordem Experimento 2.

Condições	Junta 2	Junta 3	Junta 5
1	−1,9498	1,8571	0,7925
2	−1,7319	1,6882	0,8743
3	−1,3483	1,9647	−1,3968

Fonte: autor.

Tabela 33: Valores dos coeficientes utilizados na simulação da trajetória do polinômio de 8ª ordem Experimento 2.

Condições	Junta 2	Junta 3	Junta 5
1	1,8045	−1,9186	−0,2718
2	1,1448	−1,6081	1,1696
3	1,8520	−1,8515	−1,1374

Fonte: autor.

Esses coeficientes foram obtidos através do algoritmo de otimização energética.

APÊNDICE D: CÁLCULO DOS COEFICIENTES DE POLINÔMIOS DE 3ª À 8ª ORDEM

Neste Apêndice serão apresentadas as equações para obtenção dos parâmetros dos Polinômios de 3ª a 8ª ordem.

- **Polinômios de Terceira Ordem**

Conforme mencionado na Seção 4.1.4, Polinômios de Terceira Ordem satisfazem 4 condições inicias, pois eles possuírem 4 coeficientes na formulação: a_k com $k \in \{0, 1, \dots, 3\}$. As Eq. em (67) descrevem um polinômio cúbico e sua derivada de primeira ordem.

$$\begin{aligned} r(t) &= a_0 + a_1 t + a_2 t^2 + a_3 t^3 \\ \dot{r}(t) &= a_1 + 2a_2 t + 3a_3 t^2 \end{aligned} \tag{67}$$

Utilizando os valores das restrições de velocidades nulas nas extremidades, valores r_i para a posição inicial e r_f para a posição final, e substituindo nas Eq. em (67), obtêm-se os coeficientes do polinômio de terceira ordem, explicitados nas Eq. em (68).

$$a_0 = r_i; \qquad a_1 = 0; \qquad a_2 = \frac{3}{t_f^2}(r_f - r_i); \qquad a_3 = \frac{2}{t_f^3}(r_f - r_i) \tag{68}$$

- **Polinômios de Quarta Ordem**

Polinômios de quarta ordem, Eq. em (69), satisfazem as mesmas restrições de um de terceira e sobra uma variável independente, pois possuem 5 coeficientes: a_k com $k \in \{0, 1, \dots, 4\}$.

$$\begin{aligned} r(t) &= a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4 \\ \dot{r}(t) &= a_1 + 2a_2 t + 3a_3 t^2 + 4a_4 t^3 \end{aligned} \tag{69}$$

Os coeficientes desse polinômio podem ser expressados como uma função do coeficiente a_4, conforme as Eq. em (70).

$$a_0 = r_i; \qquad a_1 = 0; \qquad a_2 = \frac{3(r_f - r_i) + a_4 t_f^4}{t_f^2}; \quad a_3 = \frac{2(r_f - r_i) + a_4 t_f^4}{t_f^3} \tag{70}$$

- **Polinômios de Quinta Ordem**

Polinômios de Quinta Ordem satisfazem 6 restrições por possuírem 6 coeficientes: a_k com $k \in \{0, 1, \dots, 5\}$. As Eq. em (71) representa esses polinômios, bem como suas derivadas de primeira e segunda ordem.

$$\begin{aligned} r(t) &= a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4 + a_5 t^5 \\ \dot{r}(t) &= a_1 + 2a_2 t + 3a_3 t^2 + 4a_4 t^3 + 5a_5 t^4 \\ \ddot{r}(t) &= 2a_2 + 6a_3 t + 12a_4 t^2 + 20a_5 t^3 \end{aligned} \tag{71}$$

Os coeficientes desses polinômios, para velocidades e acelerações nulas, são obtidos através da Eq. em (72).

$$\begin{aligned} a_0 &= r_i; & a_1 &= 0; & a_2 &= 0; \\ a_3 &= \frac{10(r_f - r_i)}{t_f^3}; & a_4 &= -\frac{15(r_f - r_i)}{t_f^4}; & a_5 &= \frac{6(r_f - r_i)}{t_f^5}; \end{aligned} \tag{72}$$

- **Polinômios de Sexta Ordem**

O Polinômio de sexta ordem, cujo é representado através da Eq. em (73), satisfaz todos critérios de um Polinômio de quinta ordem como uma função do coeficiente a_6, pois possuem 7 coeficientes: a_k com $k \in \{0, 1, \dots, 6\}$

$$\begin{aligned} r(t) &= a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4 + a_5 t^5 + a_6 t^6 \\ \dot{r}(t) &= a_1 + 2a_2 t + 3a_3 t^2 + 4a_4 t^3 + 5a_5 t^4 + 6a_6 t^5 \end{aligned} \tag{73}$$

$$\ddot{r}(t) = 2a_2 + 6a_3t + 12a_4t^2 + 20a_5t^3 + 30a_6t^4$$

O cálculo dos coeficientes é realizado através das Eq. em (74).

$$a_0 = r_i; \qquad a_1 = 0; \qquad a_2 = 0; \qquad a_3 = \frac{10(r_f - r_i) - a_6 t_f^6}{t_f^3};$$
$$a_4 = -\frac{15(r_f - r_i) + 3a_6 t_f^6}{t_f^4}; \qquad a_5 = \frac{6(r_f - r_i) - 3a_6 t_f^6}{t_f^5} \tag{74}$$

- **Polinômios de Sétima Ordem**

Polinômios de Sétima Ordem satisfazem 8 restrições, pois possuem 8 coeficientes: a_k com $k \in \{0, 1, \dots, 7\}$. As Eq. em (75) definem esse polinômio e suas derivadas de primeira, segunda e terceira ordem.

$$\begin{aligned} r(t) &= a_0 + a_1t + a_2t^2 + a_3t^3 + a_4t^4 + a_5t^5 + a_6t^6 + a_7t^7 \\ \dot{r}(t) &= a_1 + 2a_2t + 3a_3t^2 + 4a_4t^3 + 5a_5t^4 + 6a_6t^5 + 7a_7t^6 \\ \ddot{r}(t) &= 2a_2 + 6a_3t + 12a_4t^2 + 20a_5t^3 + 30a_6t^4 + 42a_7t^5 \\ \dddot{r}(t) &= 6a_3 + 24a_4t + 60a_5t^2 + 120a_6t^3 + 210a_7t^4 \end{aligned} \tag{75}$$

Após substituir as restrições, velocidades, acelerações e arranques nulos nas extremidades, os coeficientes desses polinômios são calculados através das Eq. em (76).

$$a_0 = r_i \qquad a_1 = 0 \qquad a_2 = 0 \qquad a_3 = 0$$
$$a_4 = \frac{35(r_f - r_i)}{t_f^4} \qquad a_5 = -\frac{84(r_f - r_i)}{t_f^5} \qquad a_6 = \frac{70(r_f - r_i)}{t_f^6} \qquad a_7 = -\frac{20(r_f - r_i)}{t_f^7} \tag{76}$$

- **Polinômios de Oitava Ordem**

Polinômios de Oitava Ordem, satisfaz todos os critérios de um polinômio de 7ª ordem e permite que um dos coeficientes fique como uma variável independente, possibilitando a

otimização da curva, pois possuem 9 coeficientes: a_k com $k \in \{0, 1, \dots, 8\}$. As Eq. em (77) definem esses polinômios e suas derivadas de primeira, segunda e terceira ordem.

$$\begin{aligned} r(t) &= \mathrm{a_0} + \mathrm{a_1 t} + \mathrm{a_2 t^2} + \mathrm{a_3 t^3} + \mathrm{a_4 t^4} + \mathrm{a_5 t^5} + \mathrm{a_6 t^6} + \mathrm{a_7 t^7} + \mathrm{a_8 t^8} \\ \dot{r}(t) &= \mathrm{a_1} + 2\mathrm{a_2 t} + 3\mathrm{a_3 t^2} + 4\mathrm{a_4 t^3} + 5\mathrm{a_5 t^4} + 6\mathrm{a_6 t^5} + 7\mathrm{a_7 t^6} + 8\mathrm{a_8 t^7} \\ \ddot{r}(t) &= 2\mathrm{a_2} + 6\mathrm{a_3 t} + 12\mathrm{a_4 t^2} + 20\mathrm{a_5 t^3} + 30\mathrm{a_6 t^4} + 42\mathrm{a_7 t^5} + 56\mathrm{a_8 t^6} \\ \dddot{r}(t) &= 6\mathrm{a_3} + 24\mathrm{a_4 t} + 60\mathrm{a_5 t^2} + 120\mathrm{a_6 t^3} + 210\mathrm{a_7 t^4} + 336\mathrm{a_8 t^5} \end{aligned} \tag{77}$$

Após substituir as restrições, os coeficientes desses polinômios podem ser calculados como uma função do coeficiente a_8 através das Eq. em (78).

$$\begin{aligned} a_0 &= r_i & a_1 &= 0 & a_2 &= 0 & a_3 &= 0 \\ a_4 &= \frac{35(r_f - r_i) + a_8 t_f^8}{t_f^4} & a_5 &= -\left(\frac{84(r_f - r_i) + 4a_8 t_f^8}{t_f^5}\right) \\ a_6 &= \frac{70(r_f - r_i) + 6a_8 t_f^8}{t_f^6} & a_7 &= -\left(\frac{20(r_f - r_i) + 12a_8 t_f^8}{t_f^7}\right) \end{aligned} \tag{78}$$

APÊNDICE E: ESTIMADOR MÍNIMOS QUADRADOS NÃO RECURSIVO PARA IDENTIFICAÇÃO DE SISTEMAS

Seja um processo físico caracterizado por uma entrada, $u(t)$, uma saída, $y(t)$, e distúrbios, $d(t)$. A função de transferência discreta linear desse processo é representada pela Eq. em (79). Onde m é o atraso máximo do polinômio $B(z^{-1})$, com b_i, $i = 1,\dots,m$, sendo os coeficientes; n, é o atraso máximo do polinômio $A(z^{-1})$, com a_i, $i = 1,\dots,n$, sendo os coeficientes.

$$\begin{aligned} A(z^{-1})y(k) &= z^{-d}B(z^{-1})u(k) + d(k) \\ A(z^{-1}) &= 1 + a_1 z^{-1} + \cdots + a_n z^{-n} \\ B(z^{-1}) &= b_0 + b_1 z^{-1} + \cdots + b_m z^{-m} \end{aligned} \tag{79}$$

Representando a Eq. em (79) por uma equação a diferenças, tem-se a Eq. (80).

$$\begin{aligned} y(k) = -a_1 y(k-1) - a_2 y(k-2) - \cdots - a_n y(k-n) + b_0 u(k) + b_1 u(k-1) \\ + \cdots + b_m u(k-m) + d(k) \end{aligned} \tag{80}$$

A Eq. (80) pode ser escrita em forma matricial, Eq. (81), denominada modelo de regressão linear (Coelho, 2016). Na qual $\varphi(k)$ o vetor de medidas e $\theta(k)$ é definido como vetor de parâmetros.

$$\begin{aligned} y(k) &= \varphi^T(k)\theta(k) + d(k) \\ \varphi(k) &= [y(k-1) \;\; y(k-2) \;\; \dots \;\; y(k-n) \;\; u(k) \;\; u(k-1) \;\; \dots \;\; u(k-m)]^T \\ \theta(k) &= [-a_1 \;\; -a_2 \;\; \dots \;\; -a_n \;\; b_0 \;\; b_1 \;\; \dots \;\; b_m]^T \end{aligned} \tag{81}$$

Realizando N medidas, suficientes para determinar os parâmetros a_i e b_j, a Eq. (81) fica na forma da Eq. (82).

$$\begin{aligned} \begin{bmatrix} y(0) \\ y(1) \\ \dots \\ y(N-1) \end{bmatrix} &= \begin{bmatrix} \varphi^T(0) & \varphi^T(-1) & \dots \\ \varphi^T(1) & \varphi^T(0) & \\ \dots & \dots & \\ \varphi^T(N-1) & \varphi^T(N-2) & \end{bmatrix} \theta + \begin{bmatrix} d(0) \\ d(1) \\ \dots \\ d(N-1) \end{bmatrix} \\ Y &= \phi\theta + D \end{aligned} \tag{82}$$

O vetor de parâmetros, θ, pode ser estimado, $\hat{\theta}$, utilizando o método dos mínimos quadrados não recursivo, Eq. (84), que é um caso particular do estimador dos mínimos quadrados ponderado, Eq. (83), quando $W = \sigma^2 I_{NxX}$.

$$\hat{\theta} = [\phi^T W \phi]^{-1} \phi^T W Y \tag{83}$$

$$\hat{\theta} = [\phi^T \phi]^{-1} \phi^T Y \tag{84}$$

O estimador dos mínimos quadrados, Eq. (84), é uma transformação linear sobre Y (medidas) e, por isso, é denominado estimador linear (Coelho, 2016). Uma vez estimada a função de transferência discreta, a determinação da função de transferência contínua correspondente pode ser obtida através de uma das relações em (85).

$$\begin{aligned} z &= 1 + T_s s && \text{Aproximação retangular } \textit{backward} \\ z &= 1/(1 - T_s s) && \text{Aproximação retangular } \textit{forward} \\ z &= (1 + T_s s/2)/(1 - T_s s/2) && \text{Aproximação trapezoidal} \end{aligned} \tag{85}$$

- **Algoritmo para Identificação**

A identificação do modelo, $G_{Pi}(s)$, de cada junta i, para $i \in \{1,2,\dots,n\}$, começa assumindo a existência de um controle, $G_{Ci}(s)$, para cada uma delas. Nesses controles, os parâmetros podem ser selecionados através de uma sintonia manual, de forma que os sistemas em malha fechada apresentem estabilidade. Com os dados de entrada $u(k)$ e saída $y(k)$ desses sistemas, estima-se, através dos Mínimos Quadrados, as funções de transferência em malha fechada $\hat{G}_{MFi}(s)$ de cada junta com os valores de resposta obtidos. Sabe-se que, em malha fechada, a relação entre $\hat{G}_{MF}(s)$, $G_C(s)$ e $G_P(s)$ é denotada através da Eq. (86).

$$\hat{G}_{MF}(s) = \frac{G_P(s) G_C(s)}{1 + G_P(s) G_C(s)} \tag{86}$$

Sabe-se ainda que tais funções podem ser decompostas conforme as Eq. em (87).

$$\hat{G}_{MF}(s) = \frac{N_{\hat{G}_{MF}(s)}}{D_{\hat{G}_{MF}(s)}}; \qquad G_C(s) = \frac{N_{G_C(s)}}{D_{G_C(s)}}; \qquad G_P(s) = \frac{N_{G_P(s)}}{D_{G_P(s)}} \tag{87}$$

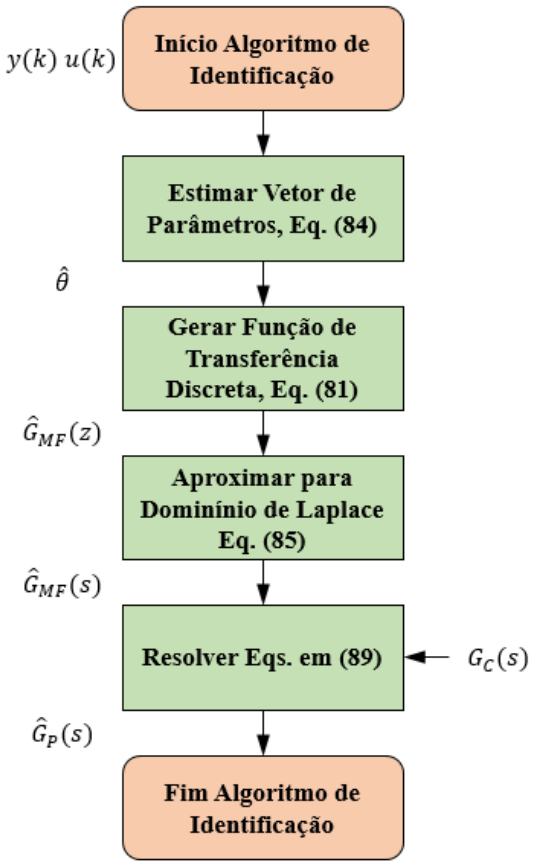
Então $\hat{G}_{MF}(s)$ pode ser reescrita conforme a Eq. (88) e, por conseguinte, seu numerador e denominador, conforme as Eq. em (89).

$$\hat{G}_{MF}(s) = \frac{N_{G_P(s)} N_{G_C(s)}}{D_{G_P(s)} D_{G_C(s)} + N_{G_P(s)} N_{G_C(s)}} \tag{88}$$

$$N_{\hat{G}_{MF}(s)} = N_{G_P(s)} N_{G_C(s)}; \qquad D_{\hat{G}_{MF}(s)} = D_{G_P(s)} D_{G_C(s)} + N_{G_P(s)} N_{G_C(s)} \tag{89}$$

Portanto, com $G_{Ci}(s)$ conhecidos, a dinâmica de cada junta, $G_{Pi}(s)$, pode ser obtida através da solução das Eq. em (89) como um Sistema de Equações Lineares. O Algoritmo de identificação pode ser resumido conforme a Fig. 54.

Figura 54: Algoritmo para identificação de $G_P(s)$.

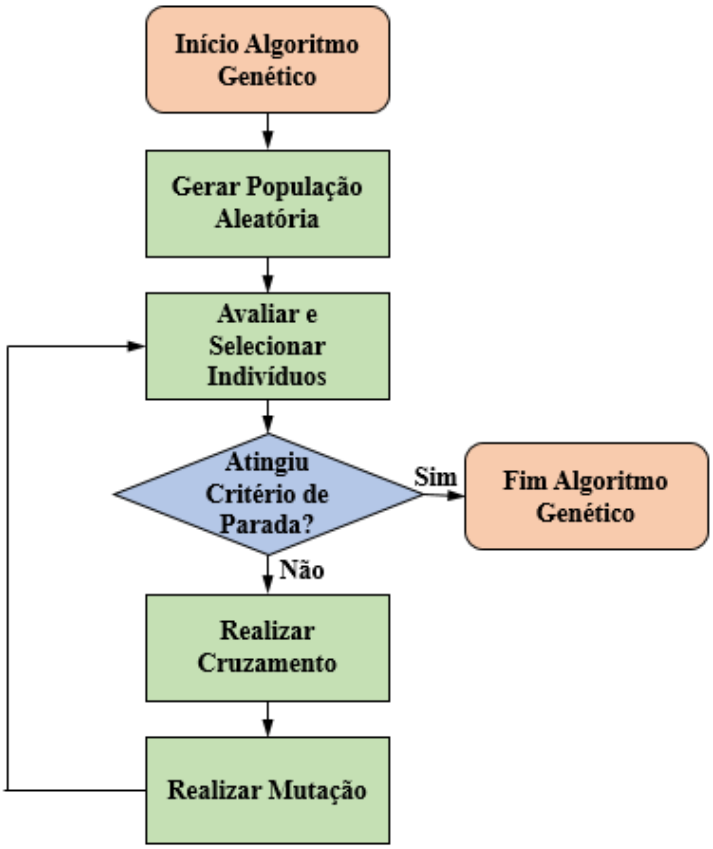


Fonte: autor.

APÊNDICE F: ALGORITMOS GENÉTICOS

Algoritmos genéticos (AG) são métodos de otimização meta-heurísticos inspirados na seleção natural. Eles foram introduzidos por John Holland, baseado na teoria de da evolução de Charles Darwin de 1859. David E. Goldberg, aluno de Holland, estendeu os AG em 1989. Segundo a teoria de Darwin, apenas aqueles organismos mais aptos conseguem se destacar e, em conseguinte, transmitir características às próximas gerações. Os algoritmos genéticos partem desse pressuposto para obtenção de resultados otimizados.

Figura 55: Algoritmo Genético em sua generalidade.



Fonte: autor.

Ademais, por ser um método heurístico, o algoritmo genético nem sempre vai encontrar o máximo global da função a ser otimizada. As chances de obter resultados melhores estão relacionadas ao tempo investido na busca dos resultados, em adição aos métodos de seleção

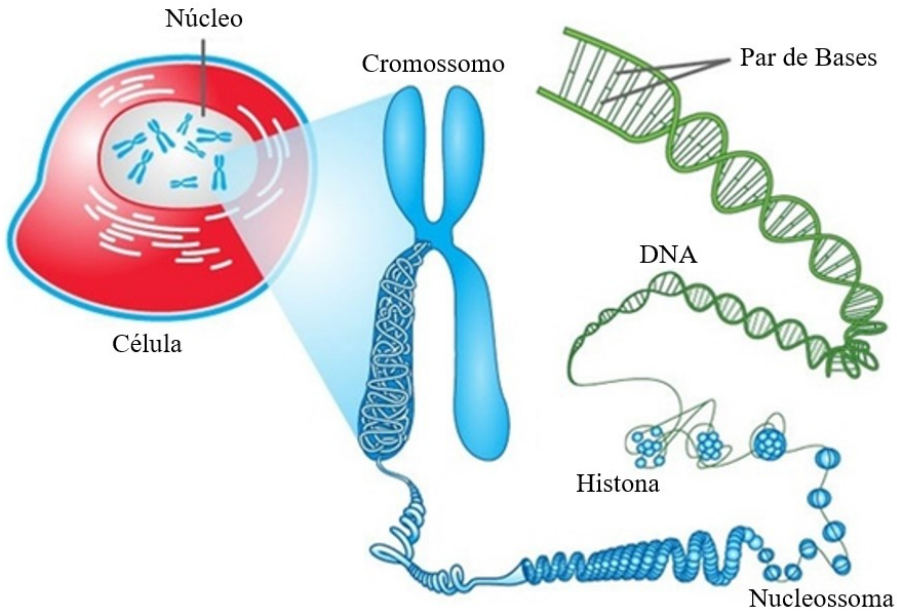
empregados (Saini, 2017). A estrutura desses algoritmos usualmente abrange as iterações apresentadas no fluxograma indicado na Fig. 55.

A seguir, serão apresentados os fundamentos biológicos utilizados nos algoritmos genéticos e como eles são abstraídos em linguagem computacional.

- **Fundamentos de Citologia**

A seleção natural é uma realidade. Os indivíduos mais capacitados interagem melhor com o ambiente em que estão contidos e sobressaem em relação aos demais. Essa interação com o meio externo depende do conteúdo contido nos organismos, ou seja, em como o desoxirribonucleico (DNA) é constituído. Quando o DNA, formado por dois filamentos longos que estão unidos em forma de espiral, modelo de dupla hélice, está unido a um grupo de proteínas específico, as histonas, formam o que é chamado de cromossomo. O cromossomo é a estrutura que transporta a informação que as células precisam para o crescimento, desenvolvimento e reprodução. Nos humanos, as células possuem 46 cromossomos, exceto as germinativas (gametas), que possuem metade desse número. A Fig. 56 exemplifica os componentes mencionados e as características a eles relacionadas.

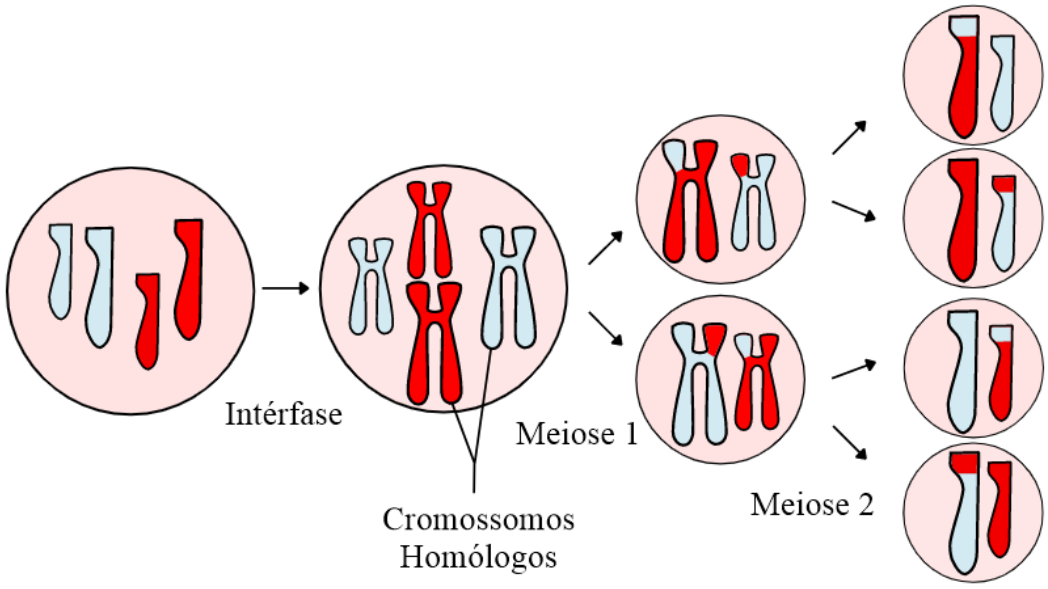
Figura 56: A célula, o cromossomo e o DNA.



Fonte: Sciencelearn (2011), direitos autorais de: University of Waikato.

As células humanas podem se dividir através de dois fenômenos biológicos: a mitose e a meiose. Porém, é a meiose que se responsabiliza pela formação dos gametas, que geram os novos organismos. Nesse processo, uma célula diploide (que possui $2n$ cromossomos) forma 4 células haploides (n cromossomos cada). Ela possui duas fases: Meiose 1 (Reducional) e Meiose 2 (Equacional). Na primeira, os pares de cromossomos da célula diploide realizam o cruzamento: trocam "pedaços" entre si. Em seguida, os pares se separam: originam-se duas novas células com a metade do número de cromossomos da primeira, as células haploides. Na fase Equacional, cada célula gerada se duplica, restando em totalidade 4 células haploides.

Figura 57: A Meiose resumidamente.



Fonte: Adaptado de Wikipedia Commons (2016).

O processo de permutação ou cruzamento, *crossing over*, ocorrido durante a fase reducional, é importante porque possibilita a variação dos genes dos gametas e consequente diversidade entre os indivíduos. Ademais, existe outro fenômeno que pode ser de origens química ou física que contribui para essa variabilidade genética: a mutação, mudança no gene que ocorre ao acaso pela interferência de fatores externos, agentes mutagênicos.

No entanto, em ponto de vista evolutivo, apenas a mutação que ocorre em gametas, também chamada de mutação germinativa, é imprescindível. A Fig. 57 mostra o processo de meiose resumidamente.

- **Operadores Genéticos**

Os algoritmos genéticos objetivam minimizar uma $f(x)$, $x \in X$. Onde f é definida como a função *fitness*, x representa o conjunto inicial aleatório de organismos, uma população de cromossomos, que otimiza f, e X define as restrições do problema.

Os tópicos seguintes apresentarão maneiras de abstração dos fenômenos biológicos no ambiente computacional na codificação dos cromossomos, na seleção natural e nos operadores mais importantes utilizados no algoritmo: cruzamento, mutação, elitismo, etc.

- **Representações dos Cromossomos**

A população inicial de cromossomos, x, pode ser codificada de duas maneiras: binária e real. O método binário é o método clássico proposto por Holland (1975). A variável $x \in (a, b)$ é codificada utilizando cadeias binárias de comprimento l, que resultará em uma precisão numérica de $(a - b)/(2^l - 1)$. Em outras palavras, cada indivíduo é considerado um cromossomo e codificado através de um vetor de caracteres de tamanho fixo formado pela concatenação das variáveis do sistema binário (0 ou 1). Segundo Holland (1992), a vantagem de se utilizar esse método é a maximização do paralelismo implícito de AG. Na Tab. 34 segue um exemplo do funcionamento da representação.

Tabela 34: Codificação binária e real.

Genótipo, Cod. Binária	Fenótipo, x	Fitness, f(x)
[1 0 0 0]	8	7,8278
[1 0 0 1]	9	6,5401
[0 0 1 0]	2	3,3735
[0 1 0 0]	4	2,2482

Fonte: autor.

Entretanto, em aplicações práticas, a abordagem via binários não apresenta bons desempenhos. Segundo Michalewicz (2007), em problemas numéricos de grande dimensão, em que alta precisão é necessária, empregar cadeias binárias é infactível: são necessários muitos bits, um maior esforço computacional. Michalewicz (2007) mostra que a representação por pontos flutuantes, onde os elementos do vetor cromossomo são números reais, produz melhores resultados quando os parâmetros são contínuos.

- **Seleção Natural**

Quais características um indivíduo deve possuir para sobressair perante aos demais em um ambiente? Em âmbito computacional é a função *fitness*, *f(x)*, que possui a atribuição de avaliar o quão adaptado está um candidato genérico, x_i, no ambiente que ele interage. A definição dessa função depende do problema que se deseja otimizar.

Existem diversos métodos para determinar numericamente quais são os indivíduos ótimos. Os mais potentes apresentam rápida convergência, porém a solução final tem maiores chances de ser um mínimo local, pois eles possuem uma baixa capacidade de diversificação. Algoritmos fracos, em contrapartida, possuem uma evolução demasiadamente lenta, mas tendem a convergir para a solução ótima global. Entre os métodos mais comuns estão: o Método da Roleta, as Seleções por Classificação, por Torneio e de Boltzmann.

O Método da Roleta é clássico e o mais empregado nos trabalhos científicos. O método atribui a cada um dos cromossomos uma probabilidade, para passar para a próxima geração, proporcional a função *fitness*. Sendo f a função *fitness*, a probabilidade Pb que um cromossomo x_i seja escolhido em população de tamanho n (Jebari, 2013) é determinada através da Eq. (90).

$$\mathrm{Pb}(\mathrm{x_i}) = \frac{f(x_i)}{\sum_{j=1}^{n} f(x_j)} \tag{90}$$

Os indivíduos que apresentarem maiores probabilidades terão mais chances de serem selecionados. O método é simples, mas sofre com convergência prematura.

Na Seleção por Classificação, os indivíduos são classificados com base no cálculo do *fitness* de cada um. A probabilidade de seleção é determinada através da Eq. (91) (Jebari, 2013).

$$\mathrm{Pb}(\mathrm{x_i}) = \frac{rank(x_i)}{n(n-1)} \tag{91}$$

De acordo com Saini (2017), a seleção por classificação possui uma pressão de seleção consistente e estratégia de seleção robusta. Quando comparado ao da roleta, a velocidade de convergência é mais lenta e a participação dos indivíduos na reprodução é quase idêntica.

A Seleção por Torneio é um dos métodos mais sofisticados. Assume-se que os indivíduos batalham entre si e aqueles que apresentam um maior número de vitórias em q competições são selecionados. Os torneios permitem uma chance igual de competição, preservando a diversidade. Entretanto, possui uma velocidade de convergência demorada. A probabilidade de um indivíduo ser selecionado é dada pela Eq. (92) (Jebari, 2013).

$$\begin{aligned} \mathrm{Pb}(\mathrm{x_i}) &= \frac{C_{n-1}^{k-1}}{C_n^k} & x_i &\in (1, n-k-1) \\ Pb(x_i) &= 0 & x_i &\in (n-k, n) \end{aligned} \tag{92}$$

Na Seleção de Boltzmann, a separação dos indivíduos ótimos é controlada por uma variação de temperatura contínua. Inicialmente, a temperatura é alta e a pressão de seleção é inversamente proporcional a ela. Enquanto a temperatura decresce ao longo do tempo, a pressão de seleção aumenta. Como resultado, a diversidade da população é mantida. A probabilidade de um indivíduo da população ser selecionado é dada pela Eq. (93) (Saini, 2017).

$$\mathrm{Pb}(\mathrm{x_i}) = -\exp\left[\frac{fmax - f(x_i)}{T}\right] \tag{93}$$

Na qual $T = T_0(1-\alpha)^k$, $k = 1 + 100i/G$, T_0 é a temperatura inicial, α é uma constante, i representa a geração atual e G é o valor de geração máximo. Saini (2017) afirma que o método de Boltzmann apresenta bons resultados, entretanto, não converge rapidamente.

- **Cruzamento**

Conforme apresentado, biologicamente, o cruzamento (*crossing over*) acontece durante a meiose, na etapa reducional. Nos algoritmos genéticos, esse fenômeno acontece após a etapa

de seleção dos indivíduos. De maneira análoga a teoria biológica, os indivíduos são selecionados aos pares e em seguida eles trocam informações do genótipo. Existem diversos métodos para realização do Cruzamento, serão apresentados os seguintes: Cruzamento Simples, Cruzamento Duplo, Cruzamento Plano e Cruzamento Discreto (Mendes, 2013).

No Cruzamento Simples, é selecionado aleatoriamente um número inteiro k, $1 \leq k \leq l$, onde l é o tamanho do vetor cromossomo. Então, dois novos cromossomos são criados trocando os genes que estão entre $k-1$ e l dos cromossomos pais, conforme mostrado na Tab. 35.

Tabela 35: Cruzamento Simples.

Geração i	**1**	**2**	**(…)**	**k**	**$k+1$**	**(...)**	**l**
Crom. Asc. 1	0,64	0,23	0,91	0,59	0,73	0,96	0,41
Crom. Asc. 2	0,88	0,09	0,87	0,09	0,51	0,19	0,89
Geração $i+1$	**1**	**2**	**(…)**	**k**	**$k+1$**	**(...)**	**l**
Crom. Desc. 1	0,64	0,23	0,91	0,59	0,51	0,19	0,89
Crom. Desc. 2	0,88	0,09	0,87	0,09	0,73	0,96	0,41

Fonte: autor.

O Cruzamento Duplo é bem parecido com o Simples, difere-se apenas no fato que são duas variáveis inteiras geradas aleatoriamente ao invés de apenas uma. O cruzamento acontece nos espaços entre os dois inteiros, como mostrado na Tab. 36.

Tabela 36: Cruzamento Duplo.

Geração i	**1**	**(…)**	**k**	**$k+1$**	**(...)**	**s**	**l**
Crom. Asc. 1	0,17	0,41	0,89	0,73	0,71	0,61	0,58
Crom. Asc. 2	0,58	0,15	0,61	0,44	0,07	0,44	0,56
Geração $i+1$	**1**	**(…)**	**k**	**$k+1$**	**(...)**	**s**	**l**
Crom. Desc. 1	0,17	0,41	0,89	0,44	0,07	0,44	0,58
Crom. Desc. 2	0,58	0,15	0,61	0,73	0,71	0,61	0,56

Fonte: autor.

O Cruzamento Plano utiliza vetores (r^1 e r^2) com valores aleatórios (entre 0 e 1) que ponderam o material genético fornecido pelos cromossomos ascendentes. A Eq. (94) denota obtenção matemática dos genótipos dos descendentes 1, x_i^{d1}, e 2, x_i^{d2}.

$$\begin{aligned} x_i^{d1} &= \mathrm{r_i^1 x_i^{a1}} + (1 - \mathrm{r_i^1}) \mathrm{x_i^{a2}} \\ x_i^{d2} &= \mathrm{r_i^2 x_i^{a1}} + (1 - \mathrm{r_i^2}) \mathrm{x_i^{a2}} \end{aligned} \tag{94}$$

Na qual x_i^{a1} e x_i^{a2} representam os cromossomos ascendentes 1 e 2, respectivamente. A Tab. 37 exemplifica o cálculo dos indivíduos descendentes. Uma variação muito utilizada do Cruzamento Plano acontece quando os valores dos vetores são fixados em 0,5, em outras palavras, realiza-se uma média entre o material genético dos pais.

Tabela 37: Cruzamento Plano.

Geração i	**1**	**2**	**3**	**4**	**5**	(...)	l
Crom. Asc. 1	0,19	0,17	0,20	0,79	0,57	0,76	0,46
Crom. Asc. 2	0,86	0,25	0,21	0,89	0,48	0,31	0,01
Vetor Aleatório 1	0,90	0,70	0,006	0,51	0,28	0,13	0,81
Vetor Aleatório 2	0,45	0,54	0,48	0,87	0,21	0,28	0,40
Geração $i+1$	**1**	**2**	**3**	**4**	**5**	(...)	l
Crom. Desc. 1	0,26	0,19	0,21	0,84	0,51	0,37	0,37
Crom. Desc. 2	0,56	0,20	0,20	0,80	0,50	0,44	0,19

Fonte: autor.

O Cruzamento Discreto utiliza uma probabilidade Pb na geração dos genes. O descendente 1 tem uma probabilidade Pb de herdar os genes do ascendente 1, enquanto o 2 possui uma probabilidade Pb de herdar os genes do ascendente 2. Nesse método, é comum utilizar um valor de probabilidade equivalente a 0.7. A Tab. 38 exemplifica esse tipo cruzamento para essa probabilidade.

Tabela 38: Cruzamento Discreto.

Geração i	**1**	**2**	**3**	**4**	**5**	(...)	**l**
Crom. Asc. 1	0,60	0,84	053	0,50	0,046	0,89	0,17
Crom. Asc. 2	0,24	0,89	0,97	0,34	0,23	0,34	0,82
Vetor Aleatório	0,74	0,56	0,65	0,82	0,40	0,20	0,38
Geração $i+1$	**1**	**2**	**3**	**4**	**5**	(...)	**l**
Crom. Desc. 1	0,24	0,84	053	0,34	0,046	0,89	0,17
Crom. Desc. 2	0,60	0,89	0,97	0,50	0,23	0,34	0,82

Fonte: autor.

- **Mutação**

De maneira análoga ao cruzamento, a mutação também possibilita a diversificação dos organismos. Por ser um operador genético secundário, os valores de probabilidade escolhidos na mutação são geralmente baixos. Sua ação é fundamental para evitar o problema de mínimos locais, pois modifica levemente a direção de busca. Se o cruzamento é responsável por encontrar os melhores indivíduos da seleção atual, a mutação é a responsável por explorar o espaço de busca (Abdoun *et al*., 2012).

Ela age basicamente trocando os genes de um cromossomo de posição. Por exemplo, seja um cromossomo que já passou pela etapa de seleção e $crossover$, suponha uma taxa de mutação 0,2%, a cada 1000 posições de genes, apenas 2 dessas sofrerão mutação e se modificarão (Mendes, 2013). Um individuo x_i é escolhido para sofrer mutação cujos genes estão, inicialmente, na ordem $x_i = [gene_1 \;\; gene_2 \;\; gene_3 \;\; ... \;\; gene_n]$. Após a mutação, ele fica, por exemplo, na ordem $x_i = [gene_2 \;\; gene_1 \;\; gene_3 \;\; ... \;\; gene_n]$. Nessa situação, o gene que estava na posição 1 trocou de lugar com o gene da posição 2 e, assim, modificou completamente o fenótipo obtido da cadeia binária.

Entretanto, quando se emprega uma codificação real, essa troca de posições entre o genótipo não irá modificar os valores obtidos do cromossomo, pois uma codificação real já é a representação do fenótipo em si. Nesse tipo de codificação, é comum substituir os valores de genótipos por valores aleatórios garantindo, assim, a leve modificação na direção de busca.

- **Outros operadores**

Além dos operadores citados, existem outros que quando são incluídos aumentam o desempenho de Algoritmos genéticos. Os que merecem ser destacados são o Elitismo e a Reinicialização.

O Elitismo consiste em aproveitar os melhores indivíduos da geração atual para a próxima geração, fazendo com que a solução seja alcançada mais rapidamente. Esse método possui boa sinergia com alguns dos métodos de seleção apresentados. Por exemplo, segundo Saini (2017), o elitismo pode contornar o problema de perda de informações da seleção de Boltzmann que ocorre durante a fase de mutação.

A Reinicialização é empregada para evitar que o algoritmo fique estagnado. Os melhores indivíduos são armazenados, acontece a reinicialização e uma nova população aleatória é gerada. Então, os melhores indivíduos são inseridos nessa nova população e o algoritmo continua o funcionamento. Essa reinicialização é periódica e acelera a convergência do algoritmo.