

Modelagem, Otimização do Espaço de Trabalho, Controle e Simulação de um Robô Delta via Torque Computado

Mathias Giordani Tilton¹ e Vítor Tumelero Valente²

TCC2 - Curso de Engenharia de Controle e Automação

Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul - Campus Farroupilha
Farroupilha, Brasil

mathiastilton@hotmail.com¹, vitor.valente@farroupilha.ifrs.edu.br²

Resumo—Os robôs de topologia Delta têm demonstrado excelente desempenho em aplicações que exigem alta precisão e velocidade, sendo muito utilizados na automação dos processos de embalagem, montagem e reposicionamento de pequenos itens. Este artigo faz parte da disciplina Trabalho de Conclusão de Curso II, do curso de Engenharia de Controle e Automação do IFRS - Campus Farroupilha e tem como objetivo geral realizar uma simulação do modelo dinâmico e do controle de posicionamento de um robô Delta aplicando o método do torque computado. Após o dimensionamento do robô via algoritmo genético e a implementação das equações e trajetórias no Simulink, validaram-se os modelos teóricos e o projeto do controlador, o qual obteve excelente desempenho de rastreamento, obtendo erro de posição inferior a 1 milímetro para a trajetória selecionada.

Palavras-chave — Modelagem cinemática e dinâmica de um robô Delta, algoritmo genético, controle de torque computado.

I. INTRODUÇÃO

Nos últimos anos, o mundo tem experimentado uma revolução tecnológica na informação, nos processos industriais e no gerenciamento de recursos, surgindo um novo conceito de automação: a Indústria 4.0, ou também conhecida como quarta revolução industrial [1]. Neste cenário, os robôs industriais são componentes indispensáveis, uma vez que proporcionam rapidez, precisão, robustez, confiabilidade e flexibilidade no desempenho das tarefas, além de possuírem a capacidade de substituir funções agressivas ao ser humano.

Impulsionados pela demanda gradativa de robôs industriais, os maiores fabricantes de robôs vêm investindo em novas tecnologias, a fim de manter a competitividade no mercado e fornecer a seus clientes produtos diferenciados. Um destes casos é o robô paralelo do tipo Delta, que, embora sua concepção seja do início dos anos 80 [2], tem ganhado popularidade recentemente, com o lançamento dos modelos MPP3S e MPP3H [3], linhas M-1*i*A, M-2*i*A, M-3*i*A [4] e IRB 360 FlexPicker [5], apresentado na Fig. 1.

As características construtivas do Delta conferem algumas vantagens em relação aos manipuladores seriais tradicionais, tais como: menor inércia em movimento, sendo que os atuadores ficam dispostos em uma base fixa; melhor distribuição do peso da carga nos elos, o que eleva a rigidez, a razão de carga

útil por peso e reduz o consumo de potência elétrica; maior precisão, visto que nos robôs de arquitetura serial a precisão final é afetada pelos erros cumulativos de todos os membros; maiores acelerações e velocidades. Contudo, as desvantagens residem na complexidade dos modelos e no espaço de trabalho reduzido [6].



Fig. 1: IRB 360 FlexPicker® [5].

Este robô é caracterizado mecanicamente, inicialmente, pela disposição de 120° entre os atuadores localizados na base fixa, remetendo ao símbolo grego Delta. Também observa-se o arranjo em paralelogramos dos braços de ligação que, em conjunto com as juntas esféricas proporcionam os movimentos no plano da base móvel, a qual é sempre paralela à base fixa durante os deslocamentos. Em relação à liberdade de movimentação no espaço, o Delta possui três graus de liberdade (GDL) translacionais (x, y, z) e um rotacional, caso exista um atuador adicional para rotacionar o efetuador.

Conforme [7], o robô Delta pode atingir acelerações em ambientes industriais de até 12 vezes a aceleração da gravi-

dade, sendo ideal o trabalho com cargas entre 10 gramas a 1 quilograma. Estas características fazem com que o Delta seja a melhor escolha para as operações de pegar e reposicionar embalagens de alimentos, componentes eletrônicos e pequenos objetos. Além da aplicabilidade na indústria, a topologia vem sendo utilizada em aplicações médicas [8] e em dispositivos sensíveis ao toque (*haptic*) [9].

O objetivo geral deste trabalho é simular o controle de posicionamento do Delta aplicando o método do torque computado no ambiente Simulink, como forma de continuação do Trabalho de Conclusão de Curso I. Assim, busca-se desenvolver uma metodologia de projeto sequencial que contemple desde os requisitos de espaço de trabalho até o dimensionamento dos elos e controle de posicionamento do robô.

Para atingir este objetivo, em um primeiro momento serão demonstrados os cálculos cinemáticos, os quais relacionam a posição, velocidades e acelerações nos espaços cartesiano e de juntas. Após será discorrido sobre a otimização do espaço de trabalho e a modelagem dinâmica, que concerne ao estudo das forças e torques nos atuadores. Explanadas estas etapas, parte-se para o projeto do controlador Proporcional-Derivativo (PD) em malha fechada, por meio do método de torque computado. Por último, serão apresentados os resultados e as considerações finais sobre o trabalho.

II. REVISÃO BIBLIOGRÁFICA

O trabalho publicado por Williams [10] detalha o desenvolvimento da solução analítica dos cálculos da cinemática inversa e direta, do jacobiano e das velocidades do robô Delta, conforme topologia original de Clavel.

Hsu et al. [11], demonstram um modo algébrico distinto para obter os cálculos cinemáticos. As equações do jacobiano também são derivadas e os autores apresentam índices de performance para caracterizar o comportamento do sistema como um todo, prevendo o espaço de trabalho resultante.

Laribi et al. [12], otimizaram os tamanhos dos *links* do robô para um espaço de trabalho desejado a partir da aplicação de um algoritmo genético. Gharahsofloo e Rahmani [13], destacam o alto custo computacional para calcular o espaço de trabalho e apresentam um algoritmo com métodos numéricos que visa maximizar a eficiência das operações. Kosinska et al. [14], abordam um algoritmo que busca selecionar os melhores parâmetros dos elos com base na análise do jacobiano, uma vez que, segundo os autores, é relevante para a implementação do controle na prática.

Zhang et al. [15] apresentam o dimensionamento de um robô Delta a partir de uma análise dinâmica, com base nas restrições dos ângulos de transmissão de forças. Os autores visam a maximização e extensão da performance para todo espaço de trabalho, a fim de determinar os valores das variáveis físicas de projeto. Também indicam ao longo do trabalho os cálculos da cinemática inversa e da dinâmica inversa, obtida a partir da aplicação do princípio do trabalho virtual.

Guglielmetti e Longchamp [16] demonstram o modelo completo da cinemática e dinâmica inversa na forma fechada (formato da equação geral do torque) e concluem que o

número de cálculos necessários para a implementação dos algoritmos não é superior quando comparados a um robô de arquitetura serial.

Brinker, Corves and Wahle [17], realizaram um estudo comparativo de três métodos para a análise dinâmica: princípio do trabalho virtual, formulação de Newton-Euler e equações de Lagrange, concluindo que o último é o mais eficiente do ponto de vista de computação. Um modelo simplificado para implementação em tempo real utilizando Lagrange é proposto por Park et al. [18], obtendo tempos para cômputo dos cálculos inferiores a 0.04 ms com processador de 2.16 GHz.

Codourey [19] desenvolveu um método eficiente para a aplicação do princípio do trabalho virtual nas equações dinâmicas, utilizando hipóteses de simplificação dos parâmetros dinâmicos. O método revelado possui a vantagem de fornecer explicitamente as matrizes de forças Coriolis, centrífugas e gravitacionais, além da matriz de massas do robô, a qual pode ser utilizada na estratégia de implementação de um controlador de antecipação (*FeedForward*), composto por um controlador PD e pelo modelo dinâmico que, segundo o autor, proporciona redução do erro de rastreamento de trajetórias em até 600%, quando comparado a um controlador PD simples. Em um trabalho adicional [20], o autor complementa e reescreve os cálculos anteriores em termos de energia cinética e conclui que as hipóteses de simplificação geram cálculos eficientes da dinâmica sem perda expressiva de precisão, podendo ser calculados em tempo real com um controle de torque computado.

Kuo e Huang [21] dissertam sobre o controle de posição do robô Delta, obtendo os torques aplicados nos motores a partir das equações de Lagrange. Voltado à aplicação prática, o trabalho apresenta o diagrama de controle no Simulink com controlador Proporcional-Integrativo-Derivativo (PID), bem como seus resultados experimentais e teóricos, os quais consideram as funções de transferência dos *drives* de acionamento e dos motores no modelo dinâmico. Os autores concluem que o controle integrado com o modelo dinâmico possui desempenho superior ao controle baseado em posição.

Olsson [22], em sua dissertação de mestrado, aborda profundamente o projeto de um robô Delta. São demonstradas as equações referentes à cinemática, dinâmica e trajetórias além do *software* e do esquema de controle realizado no Simulink.

III. FUNDAMENTAÇÃO TEÓRICA

Esta seção é subdividida nos seguintes tópicos: Cinemática, Espaço de Trabalho, Dinâmica e Controle e Simulação, os quais serão apresentados sequencialmente, visto que seus conceitos estão fortemente interligados.

A. Cinemática

A premissa inicial para a análise cinemática do manipulador, consiste em tratá-lo como um corpo rígido. Segundo Tsai [23] um corpo material é definido como um corpo rígido se a distância entre quaisquer dois pontos permanece constante. Na prática, um corpo rígido não existe, contudo, pode-se considerá-lo se a deformação do material sob estresse é muito pequena.

Os elos, ou *links* de um robô, são definidos como corpos rígidos individuais, não possuindo movimento relativo entre si. Já as juntas ou articulações são as conexões entre dois elos, permitindo movimento relativo entre os mesmos e adicionando restrições aos movimentos. Sendo assim, o conjunto de elos e juntas compõem a cadeia cinemática do manipulador, fundamental para descrever as relações de movimento e mapear o sistema mecânico para um modelo matemático.

Conforme Craig [24], cinemática é a ciência do movimento que descreve-os sem considerar as forças que os causam. Logo, o estudo da cinemática refere-se a todas as propriedades geométricas e da forma como se dá a evolução das variáveis de posição, velocidade e aceleração ao longo do tempo.

Romano [25], divide o estudo da cinemática em dois problemas: localização da posição do efetuator a partir dos ângulos das juntas (cinemática direta) e determinação dos ângulos a partir da posição do efetuator (cinemática inversa). Assim, define-se dois espaços de variáveis, o espaço das juntas e o espaço cartesiano. Nos próximos tópicos serão apresentados, inicialmente, os cálculos da cinemática inversa e direta e posteriormente, os cálculos das velocidades e acelerações.

1) *Cinemática Inversa*: O método para cômputo da cinemática inversa é demonstrado em Williams [10]. Todas as equações deste tópico são retiradas desta fonte, sendo que foi utilizada a mesma convenção dos eixos estabelecida pelo autor. Seja $p = [p_x, p_y, p_z]^T$ o vetor da posição final do efetuator em relação ao sistema de coordenadas da base fixa, informado em metros, conforme diagrama cinemático apresentado na Fig. 2.

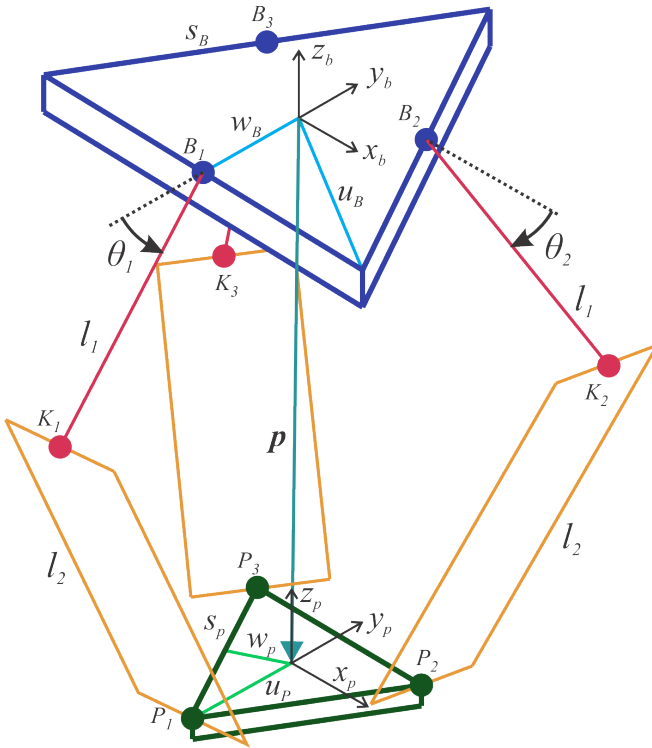


Fig. 2: Diagrama cinemático do robô Delta. Adaptado de [10].

No diagrama podem ser visualizados os seguintes parâmetros físicos, para $i = 1, 2$ e 3 , dados em metros:

- l_1 : Comprimento do elo de ligação entre o ponto de transmissão das forças geradas pelos atuadores (B_i) às juntas passivas (K_i).
- l_2 : Comprimento do elo de ligação entre as juntas passivas (K_i) e o vértice da base móvel (P_i);
- s_B : Comprimento da aresta do triângulo formado no plano da base fixa;
- u_B : Distância da origem do sistema de coordenadas da base fixa ao seu vértice, com valor $u_B = s_B\sqrt{3}/3$;
- w_B : Distância da origem do sistema de coordenadas da base fixa ao ponto central de uma aresta (B_i). É também o raio do círculo circunscrito no triângulo. Aplicando as relações trigonométricas, $w_B = s_B\sqrt{3}/6$;
- s_P : Comprimento da aresta do triângulo formado no plano da base móvel;
- u_P : Distância da origem do sistema de coordenadas da base móvel ao seu vértice. É também o raio do círculo inscrito no triângulo, dado por $u_P = s_P\sqrt{3}/3$;
- w_P : Distância da origem do sistema de coordenadas da base móvel ao ponto central de uma aresta. Pela trigonometria, $w_P = s_P\sqrt{3}/6$;

A seguir, é apresentado o método algébrico para obtenção dos valores de θ_i , em radianos.

$$E_i \cos(\theta_i) + F_i \sin(\theta_i) + G_i = 0 \quad (1)$$

Esta equação é comum na área da robótica e é solucionada realizando a substituição tangente de meio ângulo. Por conseguinte, resultam as equações (2), (3) e (4), onde A , B e C são calculadas em metros e as demais variáveis em metro ao quadrado, comprovando que t_i é adimensional. Logo, a unidade de θ_i é o radiano.

$$t_i = \frac{-F_i \pm \sqrt{E_i^2 + F_i^2 - G_i^2}}{G_i - E_i} \quad (2)$$

$$\theta_i = 2 \tan^{-1}(t_i) \quad (3)$$

$$\begin{aligned} A &= w_B - u_P = \frac{\sqrt{3}}{3} \left(\frac{s_B}{2} - s_P \right) \\ B &= \frac{s_P}{2} - \frac{\sqrt{3}}{2} w_B = -\frac{\sqrt{3}}{2} A \\ C &= w_P - \frac{1}{2} w_B = -\frac{A}{2} \\ E_1 &= 2l_1(p_y + A) \\ F_1 &= F_2 = F_3 = 2p_z l_1 \\ G_1 &= p_x^2 + p_y^2 + p_z^2 + A^2 + l_1^2 + 2p_y A - l_2^2 \\ E_2 &= -l_1[\sqrt{3}(p_x + B) + p_y + C] \\ G_2 &= p_x^2 + p_y^2 + p_z^2 + B^2 + C^2 + l_1^2 + 2(p_x B + p_y C) - l_2^2 \\ E_3 &= l_1[\sqrt{3}(p_x - B) - p_y - C] \\ G_3 &= p_x^2 + p_y^2 + p_z^2 + B^2 + C^2 + l_1^2 + 2(-p_x B + p_y C) - l_2^2 \end{aligned} \quad (4)$$

Analisando a equação (2), observa-se que existem duas soluções válidas para cada um dos ângulos de junta. A solução negativa resulta em um posicionamento do Joelho (K_i) apontado para fora do robô (ou Joelho), conforme Fig. 2. A solução positiva não é utilizada, pois o Joelho apontaria

para dentro e poderia provocar colisões dos elos na base fixa e entre si. Solucionadas as equações, define-se o vetor de posição no espaço de juntas $q = [\theta_1, \theta_2, \theta_3]^T$, informado em radianos. Os resultados obtidos foram validados mediante implementação das equações demonstradas em [11], [12], [15], [16], [17], [18] e [19].

2) *Cinemática Direta*: A solução da cinemática direta é baseada em Williams [10], que aplica o algoritmo das três esferas virtuais para o cálculo da cinemática direta, ou seja, para encontrar a posição do efetuador (p_x, p_y, p_z) dados os valores dos ângulos de junta. Os centros das esferas virtuais, em metros, são determinados por meio das equações (5) a (7), com as variáveis já expressas no sistema de coordenadas da base fixa.

$$C_1 = \begin{bmatrix} 0 \\ -w_b - l_1 \cos(\theta_1) + u_p \\ -l_1 \sin(\theta_1) \end{bmatrix} \quad (5)$$

$$C_2 = \begin{bmatrix} \frac{\sqrt{3}}{2}(w_b + l_1 \cos(\theta_2)) - \frac{s_p}{2} \\ \frac{1}{2}(w_b + l_1 \cos(\theta_2)) - w_p \\ -l_1 \sin(\theta_2) \end{bmatrix} \quad (6)$$

$$C_3 = \begin{bmatrix} -\frac{\sqrt{3}}{2}(w_b + l_1 \cos(\theta_3)) + \frac{s_p}{2} \\ \frac{1}{2}(w_b + l_1 \cos(\theta_3)) - w_p \\ -l_1 \sin(\theta_3) \end{bmatrix} \quad (7)$$

As esferas podem ser representadas por um centro e um raio, que neste caso é o elo l_2 , resultando em três equações linearmente independentes:

$$l_2^2 = (p_x - C_{1,x})^2 + (p_y - C_{1,y})^2 + (p_z - C_{1,z})^2 \quad (8)$$

$$l_2^2 = (p_x - C_{2,x})^2 + (p_y - C_{2,y})^2 + (p_z - C_{2,z})^2 \quad (9)$$

$$l_2^2 = (p_x - C_{3,x})^2 + (p_y - C_{3,y})^2 + (p_z - C_{3,z})^2 \quad (10)$$

As variáveis auxiliares para os cálculos são definidas conforme equação (11), sendo k_1 e k_2 expressas em metro ao quadrado, k_3 a k_8 , u_3 , u_5 e u_7 em metros, e as demais, adimensionais.

$$\begin{aligned} k_1 &= -C_{1,x}^2 - C_{1,y}^2 - C_{1,z}^2 + C_{3,x}^2 + C_{3,y}^2 + C_{3,z}^2 \\ k_2 &= -C_{2,x}^2 - C_{2,y}^2 - C_{2,z}^2 + C_{3,x}^2 + C_{3,y}^2 + C_{3,z}^2 \\ k_3 &= 2(C_{3,x} - C_{1,x}) & k_6 &= 2(C_{3,x} - C_{2,x}) \\ k_4 &= 2(C_{3,y} - C_{1,y}) & k_7 &= 2(C_{3,y} - C_{2,y}) \\ k_5 &= 2(C_{3,z} - C_{1,z}) & k_8 &= 2(C_{3,z} - C_{2,z}) \\ u_1 &= \frac{k_3}{k_5} - \frac{k_6}{k_8} & u_5 &= -\frac{u_3}{u_1} \\ u_2 &= \frac{k_4}{k_5} - \frac{k_7}{k_8} & u_6 &= \frac{-k_6 u_4 - k_7}{k_8} \\ u_3 &= \frac{k_2}{k_8} - \frac{k_1}{k_5} & u_7 &= \frac{k_2 - k_6 u_5}{k_8} \\ u_4 &= -\frac{u_2}{u_1} \end{aligned} \quad (11)$$

Através das equações (12) a (14), encontram-se as soluções para p_x , p_y e p_z , devendo ser escolhida a que resulta em um valor de p_z negativo após obter p_y , visto que o vetor de posição

aponta no sentido contrário do eixo Z convencional na Fig. 2.

$$p_{y\pm} = \frac{-E \pm \sqrt{E^2 - 4DF}}{2D} \quad (12)$$

$$p_{x\pm} = u_4 p_{y\pm} + u_5 \quad (13)$$

$$p_{z\pm} = u_6 p_{y\pm} + u_7 \quad (14)$$

Onde as incógnitas de $p_{y\pm}$ são definidas pela equação (15), sendo a unidade de D adimensional, E expressa em metros e F em metro ao quadrado, comprovando dimensionalmente a equação (12).

$$\begin{aligned} D &= u_4^2 + 1 + u_6^2 \\ E &= 2u_4(u_5 - C_{1,x}) - 2C_{y,1} + 2u_6(u_7 - C_{1,z}) \\ F &= u_5(u_5 - 2C_{1,x}) + u_7(u_7 - 2C_{1,z}) + C_{1,x}^2 + C_{1,y}^2 + C_{1,z}^2 \\ &\quad - l_2^2 \end{aligned} \quad (15)$$

Se p_y for igual a zero, a solução é única. Caso for retornado um número imaginário, não há solução válida para os ângulos informados. Existem também condições de singularidade (divisão por zero) quando $k_5 = 0$, $k_8 = 0$, $u_1 = 0$ ou $D = 0$. Os dois primeiros casos ocorrem quando os centros das esferas 1 e 3 ou 2 e 3 possuem as mesmas coordenadas no eixo Z (requer abordagem especial). O terceiro caso é gerado quando o centro das esferas é colinear no plano XZ, porém nunca ocorre devido aos valores dos parâmetros físicos serem positivos, assim como a quarta singularidade.

As condições que evitam as singularidades nos dois primeiros casos são contemplados pelas equações (16) a (18).

$$p_x = \frac{h_3 h_5 - h_2 h_6}{h_1 h_5 - h_2 h_4} \quad (16)$$

$$p_y = \frac{h_1 h_6 - h_3 h_4}{h_1 h_5 - h_2 h_4} \quad (17)$$

$$p_z = \frac{-E \pm \sqrt{E^2 - 4F}}{2} \quad (18)$$

Onde as variáveis h_3 , h_6 e F da equação (19) são dadas em metro ao quadrado e as demais em metros, validando dimensionalmente as equações anteriores.

$$\begin{aligned} h_1 &= 2(C_{3,x} - C_{1,x}) \\ h_2 &= 2(C_{3,y} - C_{1,y}) \\ h_3 &= -C_{1,x}^2 - C_{1,y}^2 + C_{3,x}^2 + C_{3,y}^2 \\ h_4 &= 2(C_{3,x} - C_{2,x}) \\ h_5 &= 2(C_{3,y} - C_{2,y}) \\ h_6 &= -C_{2,x}^2 - C_{2,y}^2 + C_{3,x}^2 + C_{3,y}^2 \\ z_n &= C_{1,z} = C_{2,z} = C_{3,z} \\ E &= -2z_n \\ F &= z_n^2 - l_2^2 + (p_x - C_{1,x})^2 + (p_y - C_{1,y})^2 \end{aligned} \quad (19)$$

Nesta solução particular, existem duas singularidades, quando $h_1 h_5 - h_2 h_4 = 0$ e $E^2 - 4F < 0$. A primeira é evitada pelo desenho do projeto e a segunda ocorrerá nos pontos localizados fora do espaço de trabalho.

3) *Equações de Velocidade*: As equações de velocidade foram baseadas em Williams [10], pois o autor demonstra os termos do jacobiano, derivados a partir da cinemática inversa. O jacobiano relaciona as velocidades no espaço cartesiano com as velocidades no espaço de juntas e normalmente é representado em formato de matriz.

Para Spong [26], a matriz jacobiana pode ser visualizada como uma representação vetorial da primeira derivada de uma função escalar. É muito utilizada para o planejamento e execução de movimentos e trajetórias suaves, determinação das singularidades, ou seja, dos pontos nos quais o mapeamento das velocidades entre os espaços não é invertível, na derivação das equações dinâmicas e na transformação das forças e torques requeridas pelo efetuador para as juntas do robô.

As equações (20) a (27) apresentam a solução analítica do jacobiano do robô Delta, sendo \dot{p} expresso em metros por segundo e \dot{q} , em radianos por segundo, onde $\dot{p} = [\dot{p}_x, \dot{p}_y, \dot{p}_z]^T$ e $\dot{q} = [\dot{\theta}_1, \dot{\theta}_2, \dot{\theta}_3]^T$ são, respectivamente, as velocidades no espaço cartesiano e no espaço de juntas. As matrizes J_x e J_q são de ordem 3×3 , sendo que esta última é diagonal.

$$J_x \dot{p} = J_q \dot{q} \quad (20)$$

$$J_x = \begin{bmatrix} j_{1x} & j_{1y} & j_{1z} \\ j_{2x} & j_{2y} & j_{2z} \\ j_{3x} & j_{3y} & j_{3z} \end{bmatrix} \quad (21)$$

$$J_q = l_1 \text{diag} \{J_{q_1}, J_{q_2}, J_{q_3}\} \quad (22)$$

$$j_{ix} = \begin{bmatrix} p_x \\ 2(p_x + B) - \sqrt{3}l_1 \cos(\theta_2) \\ 2(p_x - B) + \sqrt{3}l_1 \cos(\theta_3) \end{bmatrix} \quad (23)$$

$$j_{iy} = \begin{bmatrix} p_y + A + l_1 \cos(\theta_1) \\ 2(p_y + C) - l_1 \cos(\theta_2) \\ 2(p_y + C) - l_1 \cos(\theta_3) \end{bmatrix} \quad (24)$$

$$j_{iz} = \begin{bmatrix} p_z - l_1 \sin(\theta_1) \\ 2(p_z - l_1 \sin(\theta_2)) \\ 2(p_z - l_1 \sin(\theta_3)) \end{bmatrix} \quad (25)$$

$$J_{q_1} = [(p_y + A) \sin(\theta_1) + p_z \cos(\theta_1)] \quad (26)$$

$$J_{q_2} = -[(\sqrt{3}(p_x + B) + p_y + C) \sin(\theta_2) - 2p_z \cos(\theta_2)]$$

$$J_{q_3} = [(\sqrt{3}(p_x - B) - p_y - C) \sin(\theta_3) + 2p_z \cos(\theta_3)]$$

Deste modo, é possível encontrar o jacobiano direto ou inverso, conforme equações (27) e (28). A unidade de J_x e J_q é o metro.

$$J = J_x^{-1} J_q = J_{dir} \quad (27)$$

$$J^{-1} = J_q^{-1} J_x = J_{inv} \quad (28)$$

Assim, se for desejado obter o valor da velocidade no espaço de juntas, define-se uma velocidade a ser executada no espaço cartesiano e rearranja-se a equação (20) para $\dot{q} = J^{-1} \dot{p}$. Já para obter a velocidade no espaço cartesiano a partir das velocidades no espaço de juntas, basta utilizar $\dot{p} = J \dot{q}$. Em relação às singularidades do jacobiano, estas ocorrem nas seguintes situações:

- Quando $\det(J_q) = 0$. Isto ocorre quando o ângulo formado nas juntas passivas é 0 ou π , ou ângulo interno ao paralelogramo (formado pelos elos inferiores) é 0 ou π , para $i = 1, 2$, ou 3. Fisicamente ocorre quando os três pares de elos são paralelos.
- Quando $\det(J_x) = 0$. Situação ocorre quando a soma dos ângulos ativos de junta com os ângulos formados nas juntas passivas é igual a 0 ou π , ou quando o ângulo interno ao paralelogramo é 0 ou π , para $i = 1, 2$ e 3. Além disso, singularidades ocorrem quando $j_i = \pm j_k$ na matriz J_x para $i \neq k$. Fisicamente ocorre quando dois pares de elos são paralelos e nestes pontos a plataforma móvel possui apenas 1 GDL, pois move-se em uma única direção.
- Quando $\det(J_q) = 0$ e $\det(J_x) = 0$. Esta situação ocorre quando o ângulo interno ao paralelogramo é zero ou π para $i = 1, 2$ ou 3. Fisicamente, dois pares de elos estão no mesmo plano ou em planos paralelos. Neste caso, a plataforma móvel possui 1 GDL, rotacionando sobre o eixo horizontal.

4) *Equações de Aceleração*: Os cálculos das acelerações apresentados a seguir são de autoria própria, e complementam as equações já desenvolvidas por [10]. As acelerações da plataforma podem ser obtidas derivando a equação (20) utilizando a regra da cadeia, conforme demonstrado abaixo:

$$\dot{J}_x \dot{p} + J_x \ddot{p} = \dot{J}_q \dot{q} + J_q \ddot{q} \quad (29)$$

Os termos J_x e J_q já foram introduzidos anteriormente, e \dot{J}_x e \dot{J}_q são encontrados por meio das equações (30) a (33) e (34) a (37), respectivamente. A unidade destes termos é o metro ao quadrado por segundo.

$$\dot{J}_x = \begin{bmatrix} \dot{j}_x & \dot{j}_y & \dot{j}_z \end{bmatrix} \quad (30)$$

$$\dot{j}_x = \begin{bmatrix} \dot{p}_x \\ 2\dot{p}_x + \sqrt{3}l_1 \dot{q}_2 \sin(\theta_2) \\ 2\dot{p}_x - \sqrt{3}l_1 \dot{q}_3 \sin(\theta_3) \end{bmatrix} \quad (31)$$

$$\dot{j}_y = \begin{bmatrix} \dot{p}_y - l_1 \dot{q}_1 \sin(\theta_1) \\ 2\dot{p}_y + l_1 \dot{q}_2 \sin(\theta_2) \\ 2\dot{p}_y + l_1 \dot{q}_3 \sin(\theta_3) \end{bmatrix} \quad (32)$$

$$\dot{j}_z = \begin{bmatrix} \dot{p}_z + l_1 \dot{q}_1 \cos(\theta_1) \\ 2(\dot{p}_z + l_1 \dot{q}_2 \cos(\theta_2)) \\ 2(\dot{p}_z + l_1 \dot{q}_3 \cos(\theta_3)) \end{bmatrix} \quad (33)$$

$$\dot{J}_q = l_1 \text{diag} \{\dot{j}_{q_1}, \dot{j}_{q_2}, \dot{j}_{q_3}\} \quad (34)$$

$$\dot{j}_{q_1} = [(p_y + A) \dot{q}_1 \cos(\theta_1) + \dot{p}_y \sin(\theta_1) + p_z \dot{q}_1 \sin(\theta_1) - \dot{p}_z \cos(\theta_1)] \quad (35)$$

$$\dot{j}_{q_2} = -\sqrt{3}[(p_x + B) \dot{q}_2 \cos(\theta_2) + \dot{p}_x \sin(\theta_2)] - [(p_y + C) \dot{q}_2 \cos(\theta_2) + \dot{p}_y \sin(\theta_2)] + 2[p_z \dot{q}_2 \sin(\theta_2) - \dot{p}_z \cos(\theta_2)] \quad (36)$$

$$\dot{j}_{q_3} = \sqrt{3}[(p_x - B) \dot{q}_3 \cos(\theta_3) + \dot{p}_x \sin(\theta_3)] - [(p_y + C) \dot{q}_3 \cos(\theta_3) + \dot{p}_y \sin(\theta_3)] + 2[p_z \dot{q}_3 \sin(\theta_3) - \dot{p}_z \cos(\theta_3)] \quad (37)$$

Deste modo, o valor da aceleração no espaço de juntas, dado em radianos por segundo ao quadrado, pode ser obtido por meio da equação (38) e a aceleração no espaço cartesiano, dada em metros por segundo ao quadrado, a partir da equação (39).

$$\ddot{q} = J_q^{-1}(\dot{J}_x \dot{p} + J_x \ddot{p} - \dot{J}_q \dot{q}) \quad (38)$$

$$\ddot{p} = J_x^{-1}(\dot{J}_q \dot{q} + J_q \ddot{q} - \dot{J}_x \dot{p}) \quad (39)$$

B. Espaço de Trabalho

Para Spong [26], o espaço de trabalho de um manipulador é o volume total varrido pelo efetuador quando são executados todos os movimentos possíveis. É limitado pelas restrições mecânicas nas juntas e pelas características geométricas do manipulador.

A equação que define o espaço de trabalho do robô Delta é deduzida a partir da equação (2). É possível notar que apenas existirão soluções válidas se a equação (40) for satisfeita:

$$E_i^2 + F_i^2 - G_i^2 \geq 0 \quad (40)$$

Assim, define-se uma função que representa o espaço de trabalho do robô em função de seus parâmetros físicos, conforme equação (41), para os ângulos de junta $i = 1, 2, 3$:

$$h_i = G_i^2 - (E_i^2 + F_i^2) \leq 0 \quad (41)$$

Nota-se que a equação (41) é função de quatro parâmetros independentes: l_1 , l_2 , A e o vetor de posições p , tornando-a não-linear e multidimensional. Além disso, o valor mínimo global da mesma é desconhecido. Deste modo, o problema da otimização consiste em selecionar os parâmetros ótimos destas variáveis com base em um volume de trabalho predefinido.

Tendo em vista que os três primeiros parâmetros são constantes e positivos, existem três situações possíveis para o resultado desta função. Na primeira, se um ponto p estiver contido no espaço de trabalho, então $h_i < 0$. Na segunda, caso o ponto estiver na fronteira do mesmo, resultará em $h_i \leq 0$ para $i = 1, 2, 3$ e $h_i = 0$ para $i = 1$ ou $i = 2$ ou $i = 3$. A última possibilidade é contemplada quando o ponto p está fora do espaço alcançável e, portanto, $h_i > 0$ para $i = 1$ ou $i = 2$ ou $i = 3$.

Assim, delimitou-se como volume de trabalho um cubo cujas coordenadas dos pontos P_k , em metros, estão indicadas na Fig. 3, para $k = 1 \dots 9$.

O ponto $(0, 0, -H)$ foi adicionado no centro do plano superior do cubo devido ao fato da superfície de trabalho do robô ser convexa, garantindo com que o mesmo atinja todos os pontos deste plano. Já o parâmetro H é definido como a distância entre o sistema de referência da base fixa em relação à superfície do volume, medida sobre o eixo Z , e o parâmetro Q estabelecido pelo valor de metade da aresta do cubo, ambos informados em metros.

O conjunto de parâmetros ótimos é encontrado a partir da minimização do valor da função de aptidão (ou função objetivo), definida na equação (42). O termo F_{1Obj} desta equação verifica se os pontos da Fig. 3 estão contidos no espaço de trabalho. Já o termo F_{2Obj} realiza a soma do módulo de todos os vértices, dando um indicativo numérico

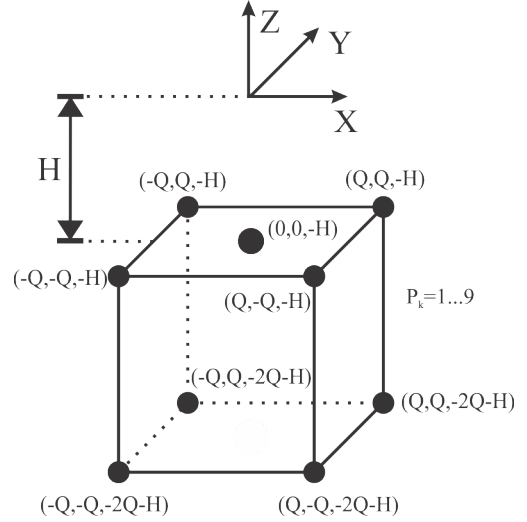


Fig. 3: Esquema do volume de trabalho estabelecido para o robô Delta. Adaptado de [12].

da qualidade da solução. Assim, um valor de $F_{1Obj} \neq 0$ significa que ao menos um dos vértices está fora do espaço de trabalho e deste modo F_{2Obj} receberá o valor zero, isto é, dá-se prioridade a encontrar um desenho de projeto que contenha os pontos dentro do volume desejado antes de minimizar a soma dos mesmos.

$$F_{Obj} = (F_{1Obj} + F_{2Obj}) \quad (42)$$

$$F_{1Obj} = \sum_{k=1}^9 \sum_{i=1}^3 \delta_i \quad (43)$$

$$F_{2Obj} = \sum_{k=1}^9 \sum_{i=1}^3 |h_i(l_1, l_2, A, P_k)| \quad (44)$$

Onde:

$$\delta_i = \begin{cases} 0, & \text{se } h_i(l_1, l_2, A, P_k) \leq 0 \\ cte, & \text{se } h_i(l_1, l_2, A, P_k) > 0 \end{cases} \quad (45)$$

$cte =$ constante positiva de alto valor

Para minimizar o valor da função de aptidão e solucionar o problema de otimização do espaço de trabalho, foi desenvolvido no TCC 1 um algoritmo genético (AG), cujas características, funcionamento e o fluxograma do código implementado no MATLAB foram explanados na ocasião. Os resultados de otimização obtidos serão recapitulados em seção posterior.

C. Dinâmica

Conforme Spong [26], a dinâmica preocupa-se em estabelecer as relações entre as forças e movimentos. Para isto, são levados em conta os atributos espaciais e temporais do caminho designado, as propriedades inerciais dos elos e da carga, atrito nas articulações, entre outros fatores, constituindo uma ferramenta muito útil para o projeto, simulação e controle de robôs.

Segundo Codourey [19], a maior dificuldade é encontrar uma solução que represente fielmente o sistema físico e que possa ser calculada em tempo real pelo controlador, uma vez que a arquitetura paralela do robô Delta e a existência de

juntas passivas dificultam a obtenção de soluções analíticas completas para a dinâmica. Assim, são adotadas as seguintes hipóteses de simplificação:

- As inércias rotacionais dos antebraços (l_2) são negligenciadas;
- Os efeitos da elasticidade do material e do atrito são negligenciados;
- O fato de que a inércia de uma haste rígida de comprimento L e massa m sobre uma das extremidades é dado por $I = (1/3)mL^2$, faz com que seja possível otimizar as massas dos antebraços em duas porções localizadas nas extremidades. Isto é, $2/3$ da massa concentra-se na extremidade superior (pontos K_i da Fig. 2) e $1/3$ na extremidade inferior (pontos P_i da Fig. 2), considerando como eixo o ponto de junção entre os elos l_1 e l_2 .

Com estas hipóteses, reduz-se o problema a quatro corpos rígidos: a plataforma móvel e os três elos superiores. A Fig. 4 representa o diagrama dinâmico do robô, podendo ser visualizados os seguintes parâmetros, informados em quilograma:

- m_{br} : Massa do elo superior (l_1);
- m_c : Massa do joelho, ou junta passiva;
- m_{ab} : Massa do elo inferior (l_2);
- m_n : Massa da base móvel;
- m_{carga} : Massa da carga útil transportada pelo robô;

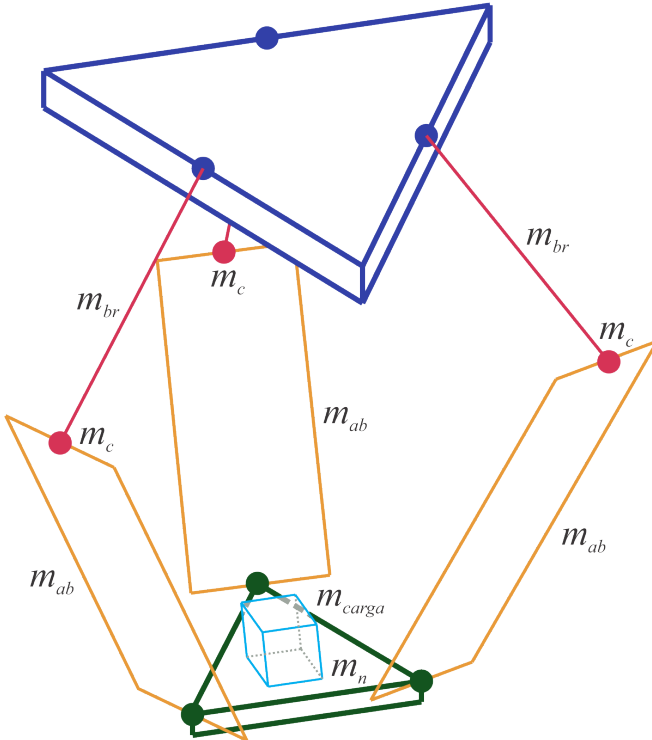


Fig. 4: Diagrama dinâmico do robô Delta.

A massa total referida à plataforma móvel é calculada por meio da equação (46), onde $r = 2/3$, que é a relação da massa localizada na articulação, conforme hipótese de simplificação. O valor obtido será útil para os cálculos da parcela de torque correspondente à aceleração da base móvel.

$$m_{nt} = m_n + m_{carga} + 3(1 - r)m_{ab} \quad (46)$$

No nível do braço, calcula-se a posição do centro de massa, em metros, por meio da equação (47):

$$r_{gb} = l_1 \left(\frac{m_{br} + m_c + rm_{ab}}{m_b} \right) \quad (47)$$

Onde:

$$m_b = m_{br} + m_c + rm_{ab} \quad (48)$$

A inércia do braço i , desconsiderando a inércia do freio do motor, é calculada pela equação (49), sendo I_m a inércia do motor e I_{br} a inércia do braço, cuja unidade é o quilograma vezes metro ao quadrado.

$$I_{bi} = I_m + I_{br} \quad (49)$$

Sendo a variável I_{br} definida por:

$$I_{br} = l_1^2 \left(\frac{m_{br}}{3} + m_c + rm_{ab} \right) \quad (50)$$

Com estas variáveis resolvidas, é possível estabelecer o modelo dinâmico do robô. Para isto, foi escolhido o método do princípio do trabalho virtual. De acordo com Olsson [27], o trabalho virtual em um sistema é o trabalho resultante de forças virtuais atuando através de um deslocamento real ou de forças reais atuando através de um deslocamento virtual, sendo que o termo deslocamento pode se referir a uma translação ou a uma rotação e o termo força a uma força ou a um momento.

É possível encontrar o princípio do trabalho virtual em soluções de diversos problemas relacionados a movimentação de corpos rígidos, fluidos ou sistemas mecânicos, uma vez que não é condicionado a parâmetros geométricos ou inerciais. Na robótica, em especial, estabelece a conexão entre a cinemática e a dinâmica.

Aplicando este princípio, é possível estabelecer a igualdade dos trabalhos virtuais associados aos espaços cartesianos e de juntas, dada pela equação (51):

$$\tau^T \cdot \delta\theta = \tau_i^T \cdot \delta p_i \quad (51)$$

Desta forma, τ é o vetor de torque relativo ao deslocamento virtual no espaço de juntas $\delta\theta$ e τ_i , o vetor de torque que atua na plataforma móvel, correspondente ao deslocamento δp_i . Juntando esta equação com o jacobiano, já encontrado a partir da equação (20), nota-se que $\tau^T = \tau_i^T \cdot J$ ou também $\tau = J^T \cdot \tau_i$. A unidade dos torques é o Newton-metro.

Em relação às forças que atuam na plataforma móvel, são duas: a gravitacional e a inercial, dadas respectivamente por (52) e (53), sendo que a unidade é o Newton. A constante da gravidade é denotada por $g \cong 9.81m/s^2$.

$$G_n = m_{nt}[0 \ 0 \ -g]^T \quad (52)$$

$$F_n = m_{nt}\ddot{p} \quad (53)$$

Reescrevendo no formato da equação (51), obtém-se os torques gravitacional e inercial:

$$\tau_{Gn} = J^T G_n \quad (54)$$

$$\tau_n = J^T F_n \quad (55)$$

Já nos elos superiores atuam dois torques sob cada elo, sendo o gravitacional representado por τ_{Gb} , cuja força atua

perpendicularmente no centro de massa r_{gb} e o inercial, cuja parcela de torque é indicada por τ_b .

$$\tau_{Gb} = m_b r_{gb} g [\cos(\theta_1) \quad \cos(\theta_2) \quad \cos(\theta_3)]^T \quad (56)$$

$$\tau_b = I_b \ddot{q} \quad (57)$$

Onde a matriz de inércia dos braços superiores é dada pela equação (58).

$$I_b = \begin{bmatrix} I_{b1} & 0 & 0 \\ 0 & I_{b2} & 0 \\ 0 & 0 & I_{b3} \end{bmatrix} \quad (58)$$

De acordo com o princípio do trabalho virtual, a contribuição de todas as forças inerciais deve ser igual a contribuição das forças não inerciais, levando à equação (59), onde Γ é o vetor de torques a ser aplicado nos motores.

$$\Gamma + \tau_{Gn} + \tau_{Gb} = \tau_b + \tau_n \quad (59)$$

Isolando o termo de torques aplicados e reescrevendo a equação anterior em termos do jacobiano e dos sistemas de referência, obtém-se as equações (60) e (61):

$$\Gamma = \tau_b + \tau_n - \tau_{Gn} - \tau_{Gb} \quad (60)$$

$$\Gamma = I_b \ddot{q} + J^T m_{nt} (J \ddot{q} + \dot{J} \dot{q}) - J^T G_n - \tau_{Gb} \quad (61)$$

Onde o termo \dot{J} é definido pela equação (62), e encontrado a partir de manipulação algébrica variáveis \ddot{p} e \dot{p} , presentes na parcela de torque τ_n . A unidade deste termo é o metro por segundo.

$$\dot{J} = J_x^{-1} (\dot{J}_q - \dot{J}_x J) \quad (62)$$

Agrupando os termos similares da equação (61), obtém-se a equação (63), solucionando a dinâmica inversa do robô.

$$\Gamma = (I_b + m_{nt} J^T J) \ddot{q} + (J^T m_{nt} \dot{J}) \dot{q} + (-\tau_{Gn} - \tau_{Gb}) \quad (63)$$

Este formato de solução é bastante conhecido na robótica, pois está na forma matricial das equações de Euler-Lagrange, conforme equação (64), onde M é a matriz de massas do robô de ordem 3×3 , C a matriz de contribuição das forças Coriolis e centrífuga, também de ordem 3×3 e G a matriz de contribuição gravitacional, de ordem 3×1 .

$$\Gamma = M(q) \ddot{q} + C(q, \dot{q}) \dot{q} + G(q) \quad (64)$$

Segundo Guglielmetti e Longchamp [16], a equação (64) é denominada de modelo dinâmico inverso nos dois espaços, uma vez que usa os estados do robô no espaço cartesiano e no espaço de juntas, embora não seja usual a representação nesta forma. A conversão entre os estados é dada por meio da cinemática, já abordada anteriormente.

D. Controle e Simulação

O controlador Proporcional-Derivativo de torque computado foi implementado nos trabalhos [8], [20] e [28], cujos autores observaram que o mesmo apresentou desempenho eficiente no rastreamento de trajetórias, com baixo erro de seguimento especialmente quando utilizado em conjunto com o modelo dinâmico.

Este controlador compensa o sinal do torque computado pela dinâmica direta por meio de um laço de controle externo, conforme demonstrado na Fig. 5. Este laço tem como sinais de referência uma trajetória no espaço de juntas \ddot{q}_d , \dot{q}_d e q_d , uma vez que na prática os sinais de realimentação para o controlador são oriundos deste espaço, como por exemplo, ângulo e velocidade dos atuadores. Contudo, elas normalmente são geradas no espaço cartesiano, que é um sistema de referência mais intuitivo para a localização espacial dos objetos, para posteriormente converter para o espaço de juntas.

No diagrama de blocos, estas trajetórias são comparadas com os ângulos e velocidades medidos do robô, respectivamente q e \dot{q} . Desta forma, obtém-se os sinais de erros e posteriormente multiplica-os pelos ganhos K_P e K_D . O sinal resultante de aceleração u , é transmitido para a matriz de massas M , a qual desacopla os três eixos do robô.

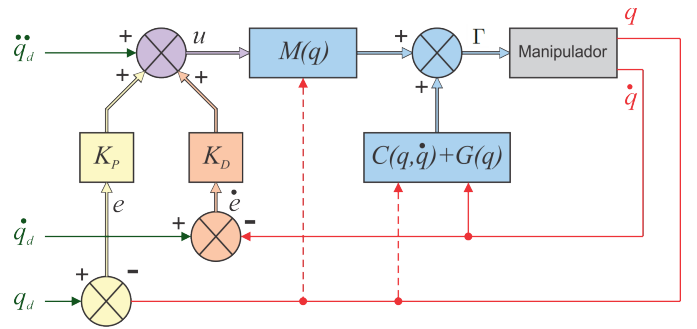


Fig. 5: Diagrama de Blocos do controlador PD de torque computado. Adaptado de [8].

Considerando o sistema com ausência de distúrbios externos, a lei de controle do torque é dada pela equação (65). Analogamente à equação (64), nota-se que $\ddot{q} = u$.

$$\Gamma = M(q)(u) + C(q, \ddot{q}) + G(q) \quad (65)$$

Por meio do diagrama de blocos, observa-se que o vetor u também pode ser expresso em termos dos erros, conforme equação (66):

$$u = \ddot{q}_d + K_D \dot{e} + K_P e \quad (66)$$

Onde K_D é uma matriz ganhos derivativos e K_P uma matriz de ganhos proporcionais, ambas diagonais, de ordem 3×3 , definidas e positivas. Os vetores de erros de posição e velocidade são dados respectivamente pelas equações (67) e (68).

$$e = q_d - q \quad (67)$$

$$\dot{e} = \dot{q}_d - \dot{q} \quad (68)$$

Logo,

$$u = \ddot{q}_d + K_D(\dot{q}_d - \dot{q}) + K_P(q_d - q) \quad (69)$$

A partir da equação anterior, nota-se que para o rastreamento do PD possuir erros extremamente baixos, o sinal de referência da trajetória deve ser da seguinte forma:

$$r = \ddot{q}_d + K_D \dot{q}_d + K_P q_d \quad (70)$$

A próxima etapa é definir os ganhos de realimentação do controlador. Para isto, reescreve-se a dinâmica do rastreamento de erros, conforme equação (71). Observe que $\ddot{e} = \ddot{q}_d$.

$$\ddot{e} + K_D \dot{e} + K_P e = 0 \quad (71)$$

A dinâmica dos erros trata-se de um sistema de segunda ordem, com frequência natural ω_n (rad/s) e taxa de amortecimento ζ (adimensional), dados pela equação (72). Escolhendo apropriadamente os ganhos de realimentação conforme as características do sistema, o erro tenderá para zero assintoticamente.

$$\omega_n = \sqrt{K_P}, \quad \zeta = \frac{K_D}{2\sqrt{K_P}} \quad (72)$$

O diagrama de blocos do controlador foi implementado no ambiente Simulink com o intuito de verificar o modelo dinâmico e simular o controle do robô, conforme Figs. 6, 7, 8 e 9. Nos blocos do Simulink, as variáveis p , \dot{p} e \ddot{p} , são representadas por 'X', 'dX' e 'ddX'; as variáveis q , \dot{q} e \ddot{q} por 'q', 'dq' e 'ddq' e os sinais de referência desejados possuem o subscrito 'des'. Além disso, as cores utilizadas nos sinais e nos blocos são correspondentes às cores apresentadas na Fig. 5.

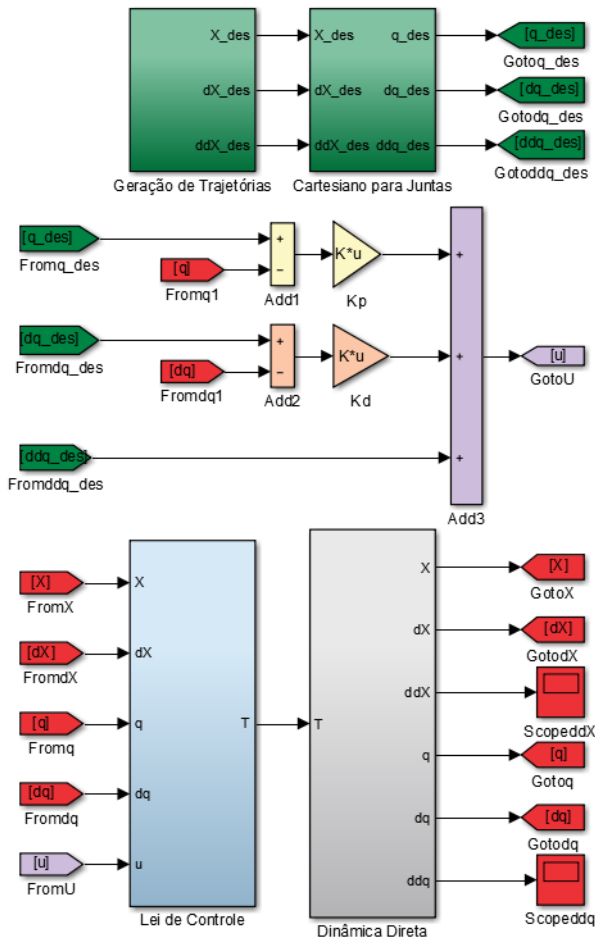


Fig. 6: Implementação do controlador PD de torque computado no Simulink.

Analisando a Fig. 6, observa-se que os sinais \ddot{q}_d , \dot{q}_d e q_d são obtidos através da conversão das funções de movimentação

definidas no espaço cartesiano (Bloco: Geração de Trajetórias) para o espaço de juntas (Bloco: Cartesiano para Juntas), por motivos já explanados anteriormente.

Além disso, é possível verificar que no controle de torque computado o manipulador é representado pelo bloco da Dinâmica Direta e que suas saídas são utilizadas no controle do *loop* externo, com exceção das variáveis 'ddX' e 'ddq'.

Os blocos que definem a lei de controle e que computam a dinâmica direta foi implementado conforme Fig. 7. Já o bloco responsável pela dinâmica direta foi dividido nas Figs. 8 e 9.

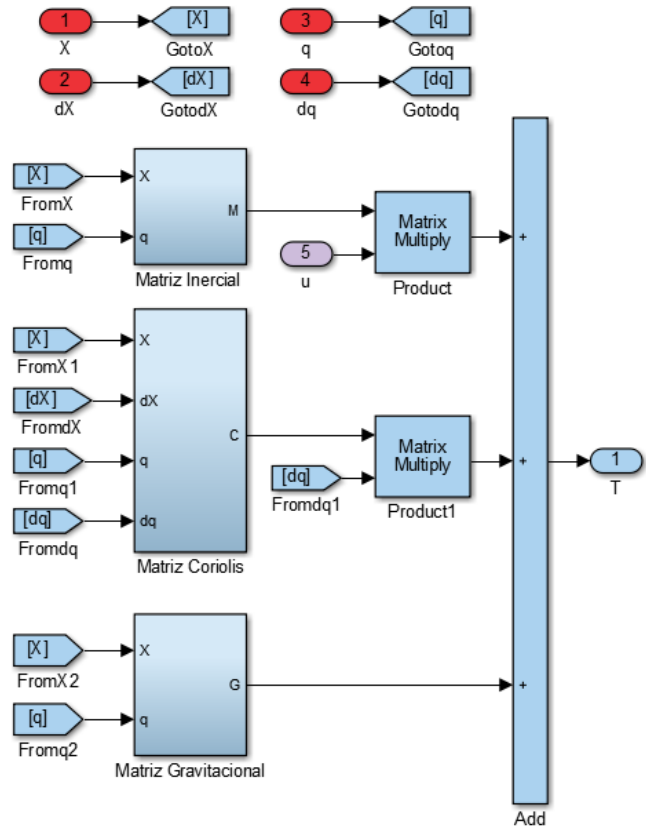


Fig. 7: Lei de controle do controlador PD de torque computado.

Em relação ao bloco da Fig. 7, este é desenvolvido a partir do modelo da dinâmica inversa, dado pela equação (64). Como se trata de um controle em malha fechada, neste caso o vetor de acelerações é substituído pelo sinal de controle u .

No que diz respeito ao bloco da dinâmica direta, tem-se como única entrada o sinal de torque e deseja-se obter na saída os valores de q , \dot{q} , \ddot{q} e suas respectivas conversões de espaços. Contudo, não é possível obter ao mesmo instante ambos valores, sendo necessário reescrever a equação (64) em termos de \ddot{q} , uma vez que dentre as variáveis citadas, é a obtida mais facilmente. Deste rearranjo, resulta:

$$\ddot{q} = M(q)^{-1}(\Gamma - C(q, \dot{q})\dot{q} - G(q)) \quad (73)$$

Após a obtenção de \ddot{q} é possível obter \dot{q} e q integrando numericamente as variáveis. No bloco de integração do Simulink, é possível definir as condições iniciais das mesmas, sendo que devem ser compatíveis com as condições iniciais da trajetória desejada. Este bloco deve sempre ser testado em

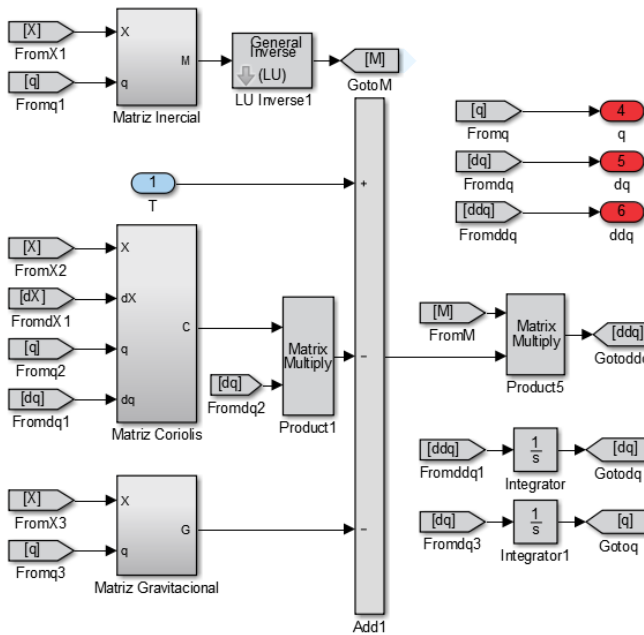


Fig. 8: Bloco da dinâmica direta do robô Delta (Parte I).

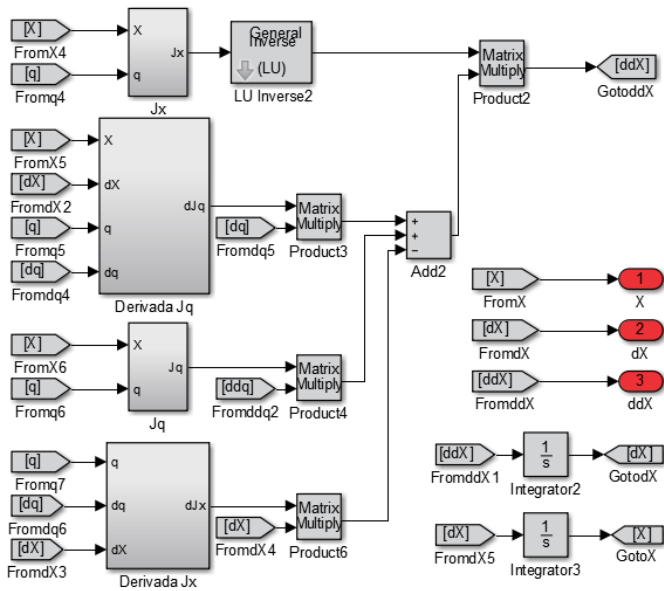


Fig. 9: Bloco da dinâmica direta do robô Delta (Parte II).

controle de malha fechada, caso contrário ocorrerá a saturação dos blocos de integração.

Por fim, obtidas as variáveis de saída no espaço de juntas, realiza-se a conversão para espaço cartesiano a partir da implementação dos cálculos demonstrados na equação (39), realimentando os laços de controle interno (do próprio bloco) e externo.

IV. PROPOSTA

O propósito geral deste trabalho é realizar a simulação do controle do posicionamento de um robô Delta via torque computado com um controlador Proporcional-Derivativo, continuando a pesquisa desenvolvida ao longo do TCC I. Após o controle e simulação, espera-se validar os modelos dinâmicos para uma trajetória de referência e obter erro de rastreamento

de posição inferior a 1 milímetro. As técnicas metodológicas tem como suporte os livros de robótica, artigos acadêmicos e dissertações de mestrado encontrados em bases científicas disponíveis *on-line*.

A. Estruturação do Trabalho

A organização sequencial das etapas do TCC 1 e TCC 2 é indicada na Fig. 10, posto que para avançar no desenvolvimento do trabalho, é necessário adquirir conhecimentos das etapas anteriores. Para isso, foi fundamental seguir a rigor o cronograma estabelecido no TCC 1, já elaborado com o intuito de prever possíveis dificuldades entre os estágios.

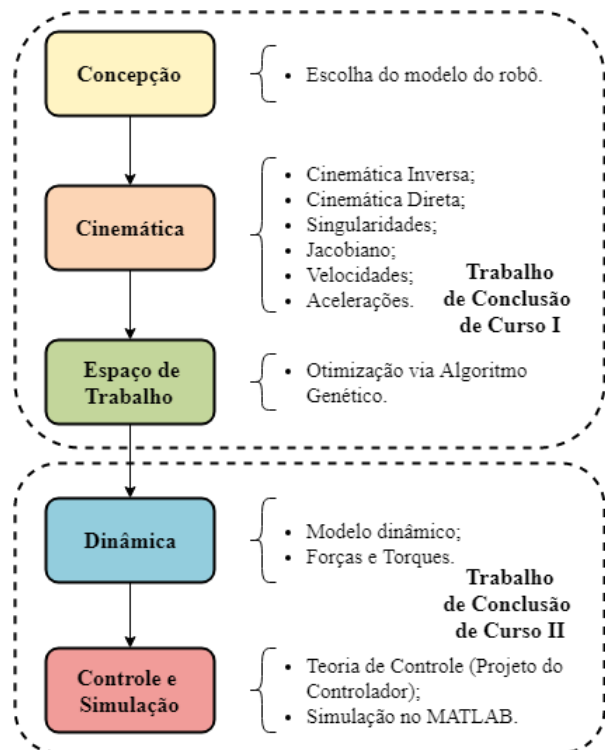


Fig. 10: Proposta para o Trabalho de Conclusão de Curso.

O Trabalho de Conclusão de Curso I foi dividido nos seguintes tópicos: Concepção, Cinemática e Espaço de Trabalho. Na concepção, foram apresentadas a motivação e as justificativas para a escolha do tema; os grupos dos principais manipuladores robóticos; as vantagens, aplicações e características construtivas do robô Delta e a comprovação de sua movimentação a partir do uso do Critério de Kutzbach [29].

Na segunda etapa foi realizada a análise cinemática do robô, compreendendo os cálculos de cinemática inversa e direta, singularidades, jacobiano e velocidades, onde foram comprovados a partir da comparação dos resultados de diferentes autores. O cálculo das acelerações, embora previstos no TCC I, não foram expostos naquele momento pois necessitavam de averiguação. Deste modo, encontram-se na Seção III, subseção A, item 4 deste trabalho.

Em relação ao terceiro tópico, otimizou-se o espaço de trabalho do robô a partir da implementação de um algoritmo genético, visto que é um fator limitante e complexo do robô

Delta, alvo de grande parte das pesquisas na área. O procedimento desenvolvido oportuniza determinar as dimensões dos elos de qualquer robô Delta, com base na escolha de um volume de trabalho.

O Trabalho de Conclusão de Curso II foi organizado nos tópicos Dinâmica e Controle e Simulação. Na dinâmica, são apresentadas as hipóteses de simplificação do modelo de inércias do robô, as equações de forças e os cálculos do torque a partir do princípio do trabalho virtual. Na etapa da Simulação e Controle, são demonstrados o esquema e a lei de controle, incluindo os blocos implementados para simulação no Simulink. Os resultados das simulações obtidas com o controlador são indicados na Seção V.

V. RESULTADOS

Aplicando a metodologia apresentada ao longo dos Trabalhos de Conclusão de Curso I e II, será demonstrado a seguir um exemplo de projeto e simulação de um robô Delta, de modo a possibilitar a avaliação numérica dos resultados.

Consideremos a seguinte aplicação: um robô Delta deve ser aplicado em uma linha de produção para automatizar o processo de pegar e reposicionar pequenas embalagens alimentícias. Após um estudo sobre os pontos que o robô deve atingir, especificou-se um espaço de trabalho cúbico de 8 m^3 . Conforme Fig. 3, o parâmetro Q corresponde ao valor de metade da aresta do cubo, sendo o volume deste calculado por $V = 8Q^3$. Logo, obtém-se $Q = 1 \text{ m}$.

Em virtude do dimensionamento construtivo, limitaram-se as variáveis físicas para os valores indicados na Tabela I. Já os parâmetros do AG foram escolhidos de modo a refinar a solução, conforme Tabela II.

Tabela I: Limites utilizados no AG.

Limites (m)	l_1	l_2	A	H
Inferior	0	0	0	1
Superior	3	3	0.5	2

Tabela II: Parâmetros utilizados no AG.

Tamanho da População	400
Número de Gerações	100
Taxa de <i>Crossover</i>	0.6
Taxa de Mutação	0.05
Número de Variáveis	4

Para este caso, o AG produziu os resultados indicados na Tabela III, com $F_{Obj} = 237.81$ e tempo de execução de aproximadamente 2.3 s, rodando em processador com *clock* de 2.40 GHz. Embora a execução do algoritmo seja rápida, ressalta-se que o tempo não é um fator crucial para esta aplicação, visto que não é utilizado em uma estratégia de controle em tempo real e que o projeto do robô pode ser feito executando-o uma única vez.

Tabela III: Resultados do processo de otimização.

	l_1	l_2	A	H
I^* (m)	1.236	2.364	0.215	1.110

Como o parâmetro A é função de s_B e s_P , conforme equação (4), existem infinitas soluções positivas para os valores físicos de s_B e s_P . Portanto, são também uma definição

de projeto, e devem ser dimensionados para comportar as dimensões dos atuadores e da ferramenta em suas respectivas bases. Para este exemplo, determinou-se que $s_B = 1.559 \text{ m}$ e $s_P = 0.407 \text{ m}$.

O espaço de trabalho obtido é apresentado na Fig. 11. Observa-se que o volume predefinido está contido nos limites de alcance do robô, ou seja, a solução encontrada está muito próxima à ideal. Ressalta-se que não foram incorporados na função objetivo limites de segurança para evitar pontos de singularidade e restrições angulares das juntas passivas.

Também é importante mencionar que a precisão da codificação do AG e que o arredondamento dos resultados do processo de otimização podem modificar o espaço de trabalho final, fazendo com que nem todos os pontos sejam atingidos, em especial aqueles localizados na parte central do plano superior do cubo.

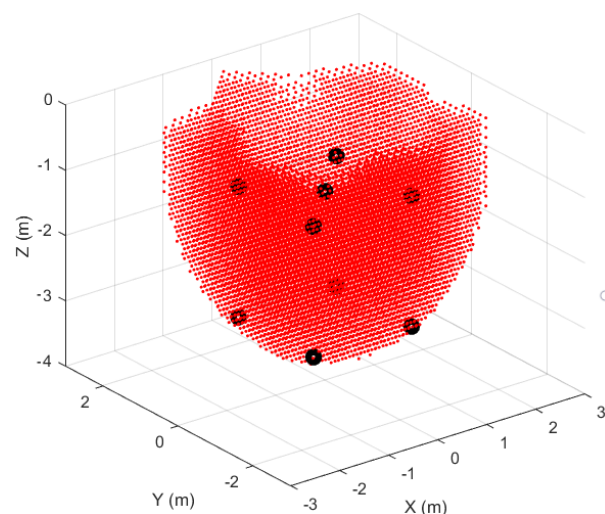


Fig. 11: Espaço de trabalho otimizado.

Para dar sequência aos cálculos, é necessário informar as massas dos elos, da base móvel, a carga útil e a inércia do motor, os quais são específicos para cada projeto, visto que variam conforme os materiais empregados. Para o exemplo, foram adotados os valores apresentados na Tabela IV.

Tabela IV: Valores dos parâmetros utilizados na simulação.

Parâmetro	Valor	Unidade
l_1	1.236	m
l_2	2.364	m
s_B	1.559	m
s_P	0.407	m
m_{br}	0.5	kg
m_c	0.3	kg
m_{ab}	1.0	kg
m_n	0.5	kg
m_{carga}	1.0	kg
I_m	0.1	kgm ²

A próxima etapa é definir os ganhos de realimentação para o controlador de torque computado, definindo assim a dinâmica do erro do sistema ao longo do tempo. Para suavizar as trajetórias, escolheu-se $\zeta = 1$, correspondente a um comportamento criticamente amortecido. Já a frequência natural está relacionada à rapidez com que o sistema seguirá

o sinal de referência. Ganhos proporcionais elevados aceleram a convergência, contudo dependem de maiores ações de controle.

Com base na equação (72), foi encontrado um ponto de operação interessante, resultando em $K_P = 100 \text{ diag}\{111\}$, $K_D = 20 \text{ diag}\{111\}$ e $\omega_n = 10 \text{ rad/s}$.

A trajetória escolhida para o teste realiza um movimento circular no plano XY e um movimento senoidal no plano XZ , conforme indicado nas Figs. 12 e 13. A função que rege o movimento é definida conforme equação (74), sendo que o termo $(1 - e^{-at})$ foi adicionado para tornar as condições iniciais de posição, velocidade e aceleração nulas.

$$\begin{aligned} p_x(t) &= A_x \cos(\omega_x t)(1 - e^{-at}) \\ p_y(t) &= A_y \sin(\omega_y t)(1 - e^{-at}) \\ p_z(t) &= C_z + A_z \sin(\omega_z t)(1 - e^{-at}) \end{aligned} \quad (74)$$

As amplitudes dos sinais foram definidas como $A_x = A_y = A_z = 0.5 \text{ m}$. As frequências, com valor $\omega_x = \omega_y = 1 \text{ rad/s}$ e $\omega_z = 2 \text{ rad/s}$ e por último, as constantes $a = 3$ e $C_z = -1.65 \text{ m}$, respeitando os limites estabelecidos para o espaço de trabalho. Para a simulação, foi estipulado um tempo de $t = 10 \text{ s}$.

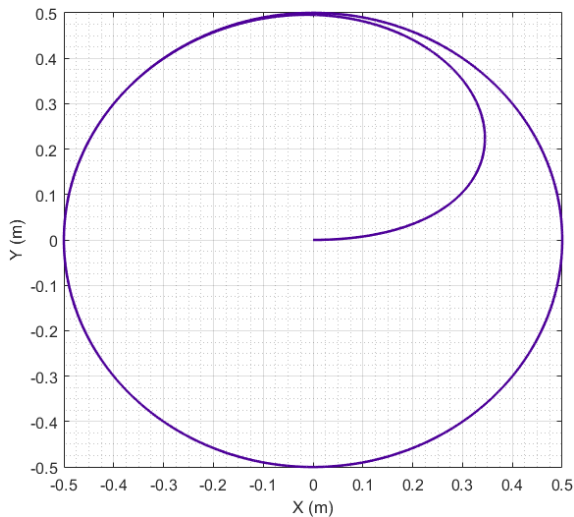


Fig. 12: Plano XY da trajetória de posição desejada.

As funções que representam as velocidades e acelerações desejadas no espaço cartesiano foram obtidas derivando a equação (74) em relação ao tempo e simplificando as parcelas transitórias que não possuíam condições iniciais nulas, resultando nas equações (75) e (76).

$$\begin{aligned} \dot{p}_x(t) &= -A_x \omega_x \sin(\omega_x t)(1 - e^{-at}) \\ \dot{p}_y(t) &= A_y \omega_y \cos(\omega_y t)(1 - e^{-at}) \\ \dot{p}_z(t) &= A_z \omega_z \cos(\omega_z t)(1 - e^{-at}) \end{aligned} \quad (75)$$

$$\begin{aligned} \ddot{p}_x(t) &= -A_x \omega_x^2 \cos(\omega_x t)(1 - e^{-at}) \\ \ddot{p}_y(t) &= -A_y \omega_y^2 \sin(\omega_y t)(1 - e^{-at}) \\ \ddot{p}_z(t) &= -A_z \omega_z^2 \sin(\omega_z t)(1 - e^{-at}) \end{aligned} \quad (76)$$

Os sinais de referência de posição, velocidade e aceleração no espaço cartesiano são demonstrados, respectivamente, nas

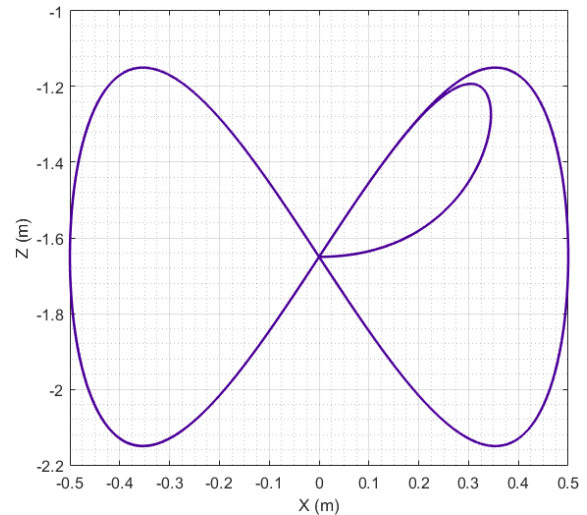


Fig. 13: Plano XZ da trajetória de posição desejada.

Figs. 14, 15 e 16. Os erros cartesianos obtidos após a inserção do controlador são demonstrados nas Figs. 17, 18 e 19.

De modo análogo, os sinais no espaço de juntas são apresentadas nas Figs. 20, 21 e 22, com seus erros exibidos nas Figs. 23, 24 e 25.

A partir destes sinais, encontra-se por meio dos cálculos da dinâmica inversa, a referência de torque, conforme Fig. 26.

Analisando os resultados obtidos, verifica-se que o esquema de torque computado e o modelo dinâmico direto foram implementados com sucesso, visto que os erros de posição, velocidade e aceleração tendem assintoticamente para zero na trajetória utilizada. Também é possível verificar que a resposta transitória apresentou um comportamento criticamente amortecido, conforme projetado. Nota-se também que a estabilização do sistema ocorre após aproximadamente 2 segundos, visto que os ganhos do controlador foram escolhidos para reduzir o torque inicial requerido pelos atuadores. Dependendo da aplicação, o tempo de estabilização pode ser reduzido a custo de maiores atuadores ou aumentando a relação das engrenagens de redução.

Em termos de posicionamento do robô, o erro de posição máximo no espaço cartesiano foi de 0.15 metros no eixo X, 0.03 metros no Eixo Y e 0.07 metros no eixo Z. Contudo, após 0.5 segundos, os erros reduzem para aproximadamente 0.02 metros em cada eixo, considerado relativamente baixo em uma trajetória com amplitude de 0.5 metros.

Com os resultados da simulação das variáveis de juntas e do torque, é possível realizar o dimensionamento dos atuadores com base nos valores máximos atingidos, sendo que as velocidades e acelerações encontradas podem ser facilmente atingidas em uma aplicação prática. Na Fig. 26, o sinal negativo de torque indica que o motor está aplicando uma força no sentido contrário à atribuição dos ângulos de junta representados na Fig. 2, ou seja, no sentido positivo do eixo Z. Já quando o torque está positivo, o atuador está acelerando a base móvel no mesmo sentido da gravidade.

Para complementar as informações já apontadas, os gráficos das variáveis de saída (atuais) podem ser obtidos subtraindo os gráficos dos sinais de referência com os sinais de erros das respectivas variáveis.

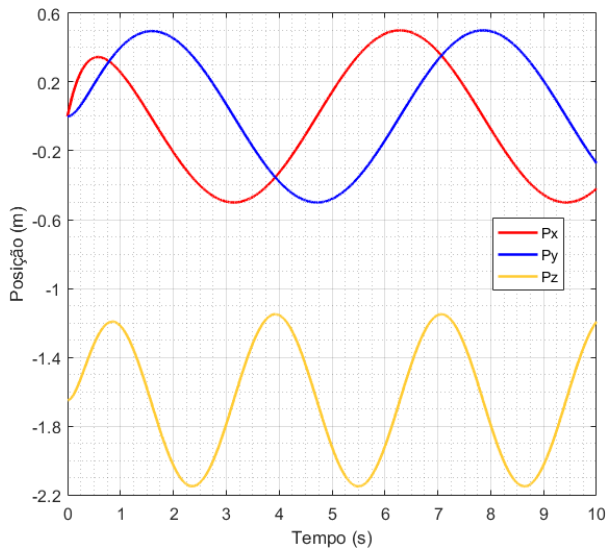


Fig. 14: Referência de posição no espaço cartesiano.

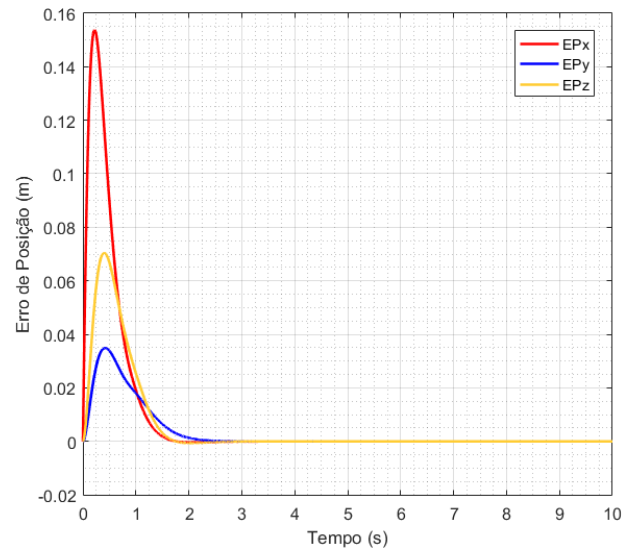


Fig. 17: Erro de posição no espaço cartesiano.

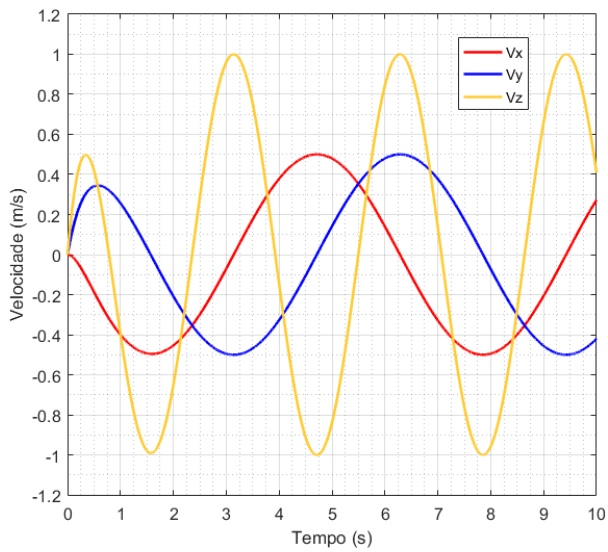


Fig. 15: Referência de velocidade no espaço cartesiano.

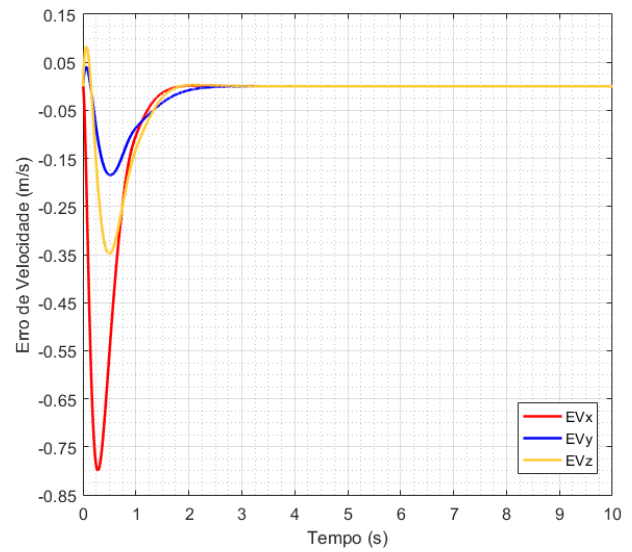


Fig. 18: Erro de velocidade no espaço cartesiano.

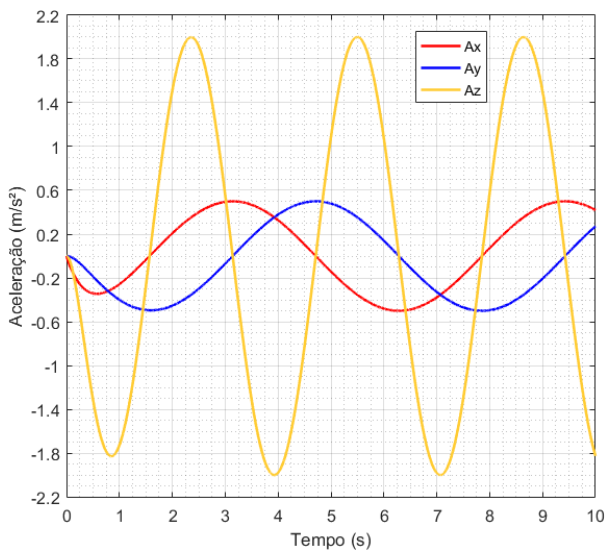


Fig. 16: Referência de aceleração no espaço cartesiano.

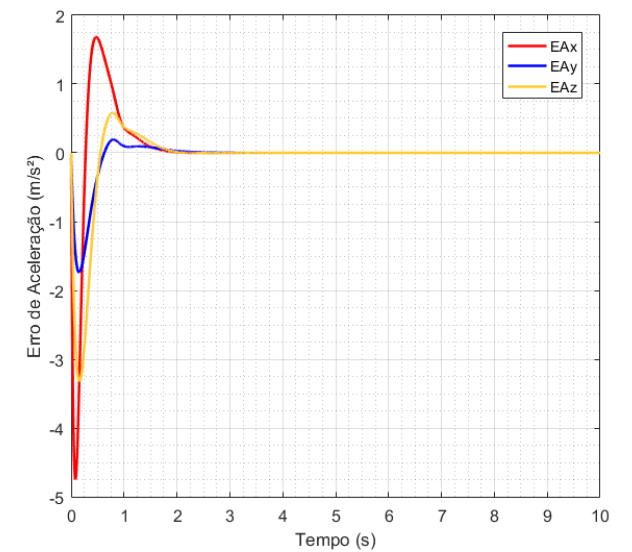


Fig. 19: Erro de aceleração no espaço cartesiano.

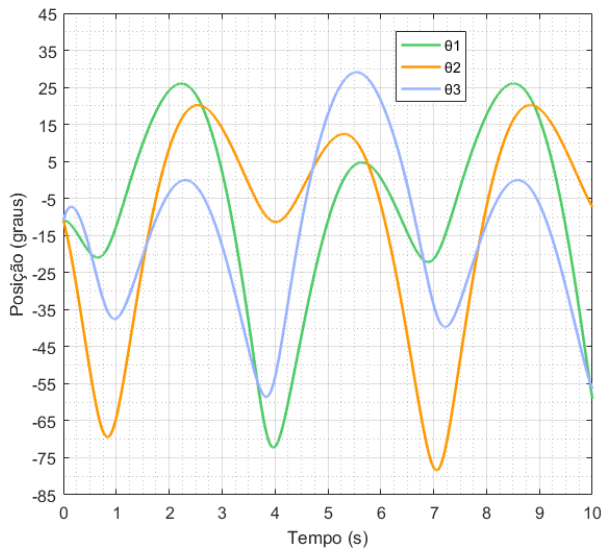


Fig. 20: Referência de posição no espaço de juntas.

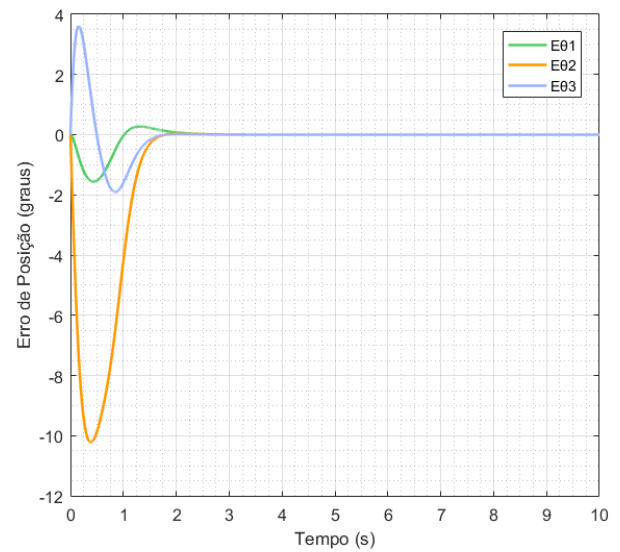


Fig. 23: Erro de posição no espaço de juntas.

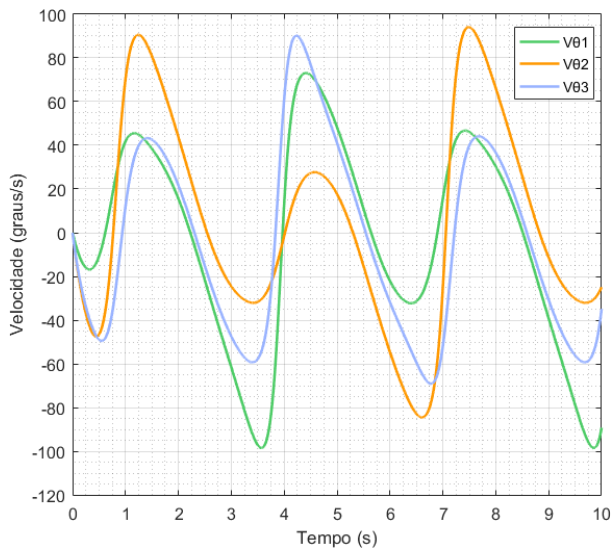


Fig. 21: Referência de velocidade no espaço de juntas.

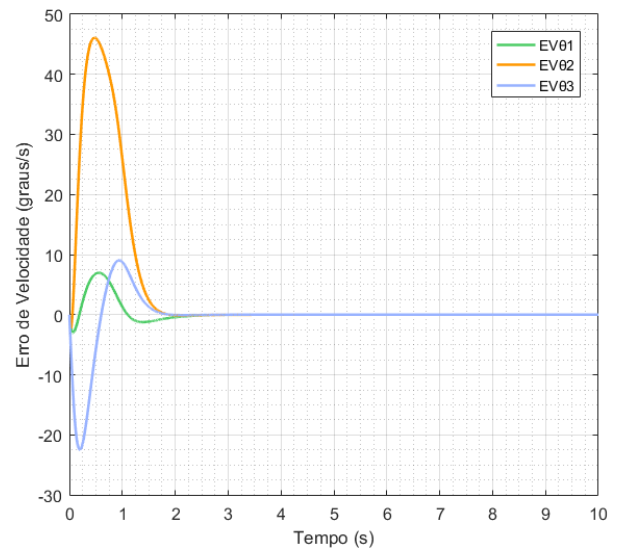


Fig. 24: Erro de velocidade no espaço de juntas.

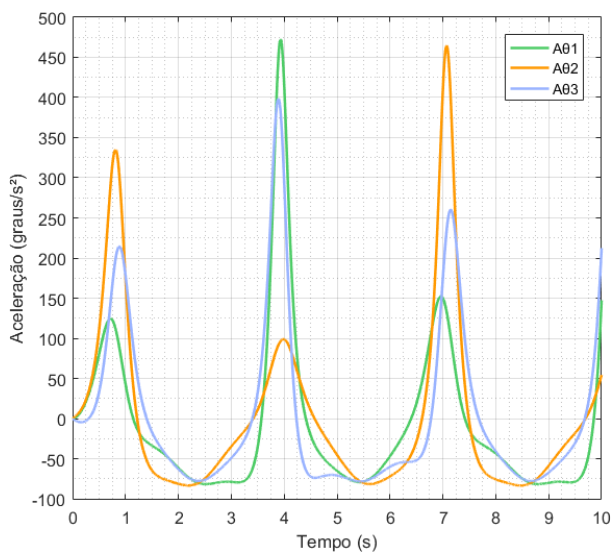


Fig. 22: Referência de aceleração no espaço de juntas.

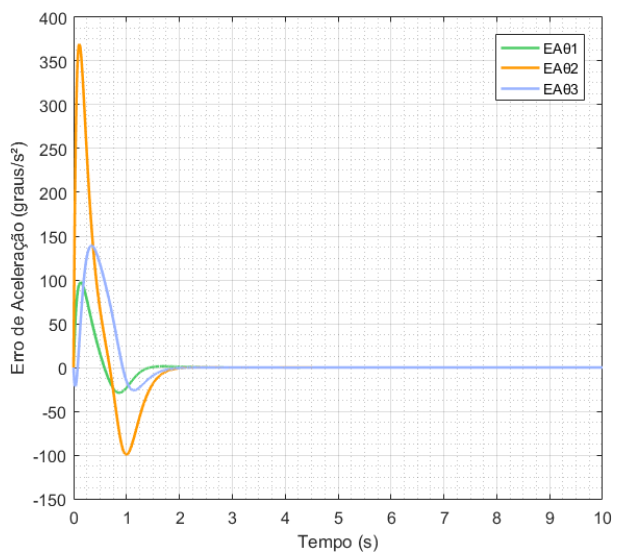


Fig. 25: Erro de aceleração no espaço de juntas.

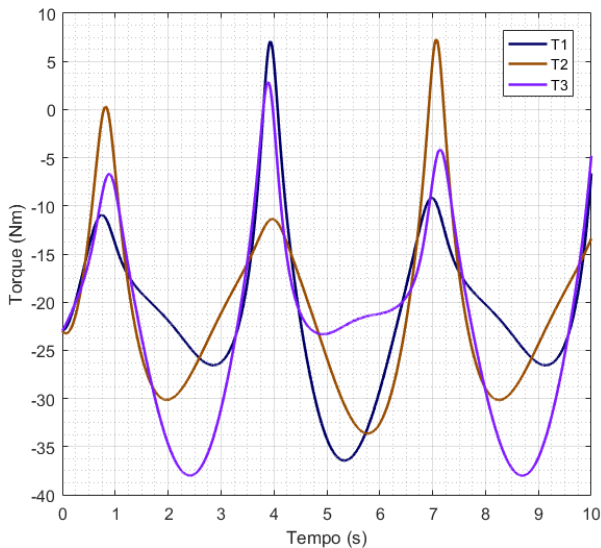


Fig. 26: Referência de Torque.

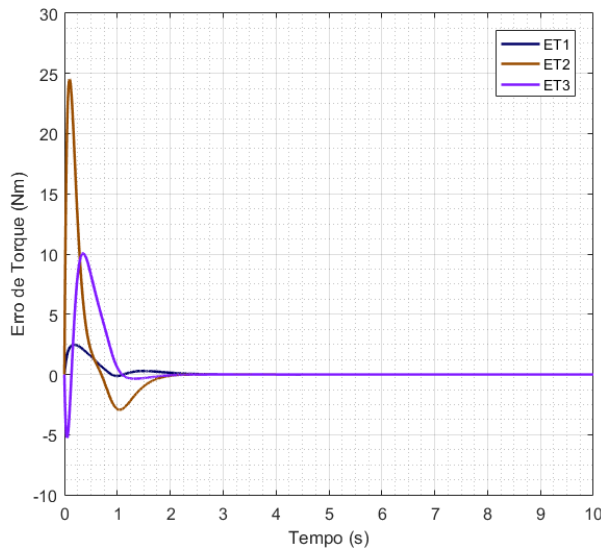


Fig. 27: Erro de Torque.

VI. CONCLUSÃO

Neste artigo foi apresentada uma estratégia de projeto de um robô Delta, compreendendo desde as etapas de especificação do espaço de trabalho até o projeto e simulação do controlador no ambiente Simulink.

Inicialmente, foram desenvolvidos os cálculos analíticos da cinemática inversa e direta, jacobiano, velocidades e acelerações nos espaços cartesianos e de juntas.

Com os métodos implementados, foram derivadas as equações que regem o espaço de trabalho, onde constatou-se a dificuldade de otimização dos parâmetros físicos por se tratar de um sistema não linear. Para solucionar este problema, foi elaborado um algoritmo genético, o qual obteve ótimo desempenho em relação à velocidade de execução do código e qualidade da solução.

Na sequência, foram estabelecidos os modelos dinâmico inverso e direto em formato de solução fechada, a partir da aplicação do princípio do trabalho virtual, obtendo as forças e torques requeridos pelos atuadores para uma dada trajetória.

Continuando as etapas, foi realizado o controle de posicionamento via esquema de torque computado proporcional derivativo, sendo verificado seu excelente desempenho nos testes realizados no Simulink, uma vez que os erros de rastreamento de torque e por consequência, de posição, são aproximadamente nulos.

Com base nos fatos mencionados, conclui-se que foram atingidos todos os objetivos previstos para o Trabalho de Conclusão de Curso II, respaldados pelos excelentes resultados indicados ao longo do mesmo. Assim, esta pesquisa contribui com a formação de novos conhecimentos relacionados à robótica, em especial sobre o robô Delta, e solidifica a teoria existente.

REFERÊNCIAS

- [1] G. Federal, "Indústria 4.0," 2018. Disponível em: <http://www.industria40.gov.br>. Acesso em: 20 de setembro de 2018.
- [2] R. Clavel, "Device for the movement and positioning of an element in space," 1990. US Patent 4,976,582.
- [3] Yaskawa, "Automação pick and place," 2018. Disponível em: <https://www.motoman.com/robotic-packagingrobots>. Acesso em: 21 de setembro de 2018.
- [4] FANUC, "Robôs industriais para manufatura," 2018. Disponível em: <https://www.fanucamerica.com/product/robots/>. Acesso em: 21 de setembro de 2018.
- [5] ABB, "IRB 360 FlexPicker," 2018. Disponível em: <http://new.abb.com>. Acesso em: 21 de setembro de 2018.
- [6] F. Pierrot, C. Reynaud, and A. Fournier, "Delta: a simple and efficient parallel robot," *Robotica*, vol. 8, no. 2, pp. 105–109, 1990.
- [7] I. Bonev, "Delta parallel robot—the story of success," *Newsletter*, no. 4, pp. 1–8, 2001.
- [8] Y. Li and Q. Xu, "Design and development of a medical parallel robot for cardiopulmonary resuscitation," *IEEE/ASME Transactions on Mechatronics*, vol. 12, no. 3, pp. 265–273, 2007.
- [9] S. Grange, F. Conti, P. Rouiller, P. Helmer, and C. Baur, "The delta haptic device," tech. rep., 2001.
- [10] R. L. W. II, "The delta parallel robot: Kinematics solutions," *Internet Publication*, 2016. Disponível em: www.ohio.edu/people/williar4/html/pdf/DeltaKin.pdf.
- [11] K. Hsu, M. Karkoub, M.-C. Tsai, and M. Her, "Modelling and index analysis of a delta-type mechanism," *Proceedings of the Institution of Mechanical Engineers, Part K: Journal of Multi-body Dynamics*, vol. 218, no. 3, pp. 121–132, 2004.
- [12] M. A. Laribi, L. Romdhane, and S. Zeghloul, "Analysis and dimensional synthesis of the delta robot for a prescribed workspace," vol. 42, pp. 859–870, 07 2007.
- [13] A. Gharahsofloo and A. Rahmani, "An efficient algorithm for workspace generation of delta robot," 2015.
- [14] A. Kosinska, M. Galicki, and K. Kedzior, "Designing and optimization of parameters of delta-4 parallel manipulator for a given workspace," *Journal of Field Robotics*, vol. 20, no. 9, pp. 539–548, 2003.
- [15] L. Zhang, J. Mei, X. Zhao, and T. Huang, "Dimensional synthesis of the delta robot using transmission angle constraints," *Robotica*, vol. 30, no. 3, pp. 343–349, 2012.
- [16] P. Guglielmetti and R. Longchamp, "A closed form inverse dynamics model of the delta parallel robot," *IFAC Proceedings Volumes*, vol. 27, no. 14, pp. 51–56, 1994.
- [17] J. Brinker, B. Corves, and M. Wahle, "A comparative study of inverse dynamics based on clavel's delta robot," in *Proceedings of the 14th World Congress in Mechanism and Machine Science. Taipei, Taiwan*, pp. 25–30, 2015.
- [18] S. B. Park, H. S. Kim, C. Song, and K. Kim, "Dynamics modeling of a delta-type parallel robot (isr 2013)," in *Robotics (ISR), 2013 44th International Symposium on*, pp. 1–5, IEEE, 2013.
- [19] A. Codourey, "Dynamic modelling and mass matrix evaluation of the delta parallel robot for axes decoupling control," in *Intelligent Robots and Systems' 96, IROS 96, Proceedings of the 1996 IEEE/RSJ International Conference on*, vol. 3, pp. 1211–1218, IEEE, 1996.
- [20] A. Codourey, "Dynamic modeling of parallel robots for computed-torque control implementation," *The International Journal of Robotics Research*, vol. 17, no. 12, pp. 1325–1336, 1998.

- [21] Y.-L. Kuo and P.-Y. Huang, "Experimental and simulation studies of motion control of a delta robot using a model-based approach," *International Journal of Advanced Robotic Systems*, vol. 14, no. 6, p. 1729881417738738, 2017.
- [22] A. Olsson, "Modeling and control of a delta-3 robot," *MSc Theses*, 2009.
- [23] L.-W. Tsai, *Mechanism design: enumeration of kinematic structures according to function*. CRC press, 2000.
- [24] J. J. Craig, *Introduction to robotics: mechanics and control*, vol. 3. Pearson Prentice Hall Upper Saddle River, 2005.
- [25] A. J. Álvares, A. Q. Bracarense, A. H. R. Costa, C. E. Pereira, D. J. d. Santos Filho, E. R. d. Pieri, H. Andreas, H. M. Caulliraux, I. Felizardo, J. M. Rosário, *et al.*, *Robótica industrial: aplicação na indústria de manufatura e de processos*. Editora Edgard Blücher, 2002.
- [26] M. W. Spong and M. Vidyasagar, *Robot dynamics and control*. John Wiley & Sons, 2008.
- [27] A. Olsson, "Modeling and control of a delta-3 robot," *MSc Theses*, 2009.
- [28] J. Hao, X. Xie, G. Bian, Z. Feng, Z. Gao, Z. Hou, H. Yu, and L. Vladareanu, "Dynamic modeling and control simulation of a modified delta manipulator," in *Information and Automation, 2015 IEEE International Conference on*, pp. 1573–1578, IEEE, 2015.
- [29] J. M. McCarthy and G. S. Soh, *Geometric design of linkages*, vol. 11. Springer Science & Business Media, 2010.