dem
universidade de aveiro
departamento de engenharia mecânica

Estudo do Planeamento de Trajetórias no Robô Delta

Gonçalo Rosa nº 102988
Gonçalo Aguiar nº 104058

8 de Maio de 2025

Conteúdo

Introdução

O planeamento de trajetórias em robótica é crucial para a execução de tarefas de forma eficiente e precisa. Esse processo consiste em gerar uma função que descreva o movimento das juntas do robô Delta ao longo do tempo, levando em consideração as suas posições, velocidades e acelerações, ou as posições e orientações do end-effector. Dependendo da tarefa, o planeamento pode ser realizado em dois espaços: o espaço das juntas, que descreve o movimento em termos das variáveis articulares, e o espaço cartesiano, que descreve o movimento em termos das posições e orientações do end-effector.

Embora o planeamento no espaço cartesiano seja intuitivo e adequado para descrever tarefas como a movimentação de um objeto de um ponto a outro, ou o seguimento de uma curva, o controlo do robô é feito no espaço das juntas. Para isso, é necessário utilizar a cinemática inversa para converter as trajetórias do espaço cartesiano em sequências de movimentos articulares.

O planeamento no espaço cartesiano também facilita a inclusão de restrições, como obstáculos ou regiões proibidas. No entanto, em situações de configurações singulares, onde o Jacobiano se torna singular e não é invertível, o planeamento no espaço cartesiano não é possível. Nesses casos, o planeamento no espaço das juntas é preferido.

Este relatório foca na avaliação de trajetórias no robô Delta, explorando as abordagens de planeamento no espaço das juntas e no espaço cartesiano, considerando suas aplicações, vantagens e limitações em diferentes cenários operacionais.

Fundamentos do Planeamento de Movimento

2.1 Tipos de Planeamento

2.1.1 Trajetória Reta

A trajetória representa a forma mais simples de descrever um movimento. Ela consiste em definir uma reta entre o ponto inicial e o ponto final do movimento.

O movimento é efetuado com uma velocidade constante, não havendo uma aceleração inicial e final que permitam criar uma suavidade no movimento. O facto de não haver pontos intermédios na trajetória pode levar a diversos problemas no mundo real.

A reta representativa deste movimento é dada pela equação (2.31):

$$\theta(t) = a_0 + a_1 t \tag{2.1}$$

Nesta equação:

- $\theta(t)$ é o ângulo da junta no instante t,
- a_0 representa o ângulo inicial da junta, isto é, $\theta(t_0)$,
- a_1 é o declive da reta, ou seja, a velocidade angular constante.

O valor do declive a_1 é obtido a partir da diferença entre os ângulos final e inicial, dividida pelo intervalo de tempo total, conforme a equação (2.2):

$$a_1 = \frac{\theta(t_f) - \theta(t_0)}{t_f - t_0} \tag{2.2}$$

Onde:

- $\theta(t_0)$ é o ângulo inicial no tempo t_0,
- $\theta(t_f)$ é o ângulo final no tempo t_f,
- $t_f - t_0$ é a duração do movimento.

2.1.2 Trajetória com Polinómio Cúbico

A trajetória baseada num polinómio cúbico representa uma forma mais realista de descrever o movimento de uma junta robótica entre dois instantes de tempo. Ao contrário da trajetória anterior, que assume uma velocidade constante e ignora as condições iniciais e finais de velocidade, o polinómio cúbico permite controlar não só as posições inicial e final, mas também as velocidades correspondentes. Esta característica permite criar uma suavidade no movimento.

A expressão geral da trajetória cúbica é dada por um polinómio de grau três:

$$\theta(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 \tag{2.3}$$

onde $\theta(t)$ representa o ângulo da junta no instante de tempo t, e os coeficientes a_0, a_1, a_2 e a_3 são determinados com base nas condições de contorno impostas.

Para garantir um movimento suave, impõem-se quatro condições: as posições e velocidades nos instantes inicial (t_0) e final (t_f). Essas condições são expressas por:

$$\begin{cases} \theta(t_0) = \theta_0 \\ \theta(t_f) = \theta_f \\ \dot{\theta}(t_0) = \dot{\theta}_0 \\ \dot{\theta}(t_f) = \dot{\theta}_f \end{cases} \tag{2.4}$$

A derivada da equação (2.3) em relação ao tempo fornece a expressão da velocidade angular:

$$\dot{\theta}(t) = a_1 + 2a_2 t + 3a_3 t^2 \tag{2.5}$$

enquanto a segunda derivada dá a aceleração angular:

$$\ddot{\theta}(t) = 2a_2 + 6a_3 t \tag{2.6}$$

A substituição das condições de contorno (2.4) nas equações (2.3) e (2.5) permite formar um sistema linear com quatro equações, o qual pode ser resolvido para obter os coeficientes do polinómio. A solução analítica deste sistema é:

$$a_0 = \theta_0 \tag{2.7}$$

$$a_1 = \dot{\theta}_0 \tag{2.8}$$

$$a_2 = \frac{3(\theta_f - \theta_0)}{(t_f - t_0)^2} - \frac{2\dot{\theta}_0 + \dot{\theta}_f}{t_f - t_0} \tag{2.9}$$

$$a_3 = \frac{2(\theta_0 - \theta_f)}{(t_f - t_0)^3} + \frac{\dot{\theta}_0 + \dot{\theta}_f}{(t_f - t_0)^2} \tag{2.10}$$

Estes coeficientes garantem que a trajetória cumpre as posições e velocidades impostas nos extremos, resultando num movimento suave e fisicamente plausível. Note-se, no entanto, com esta tipo de trajetória ainda não é possível controlar as diversas acelerações entre os diferentes pontos. No entanto é possível criar mais segmentos de trajetória, para que seja possível definir mais velocidades ao longo do mesmo movimento.

2.2 Polinómio Quíntuplo

Em sistemas robóticos, especialmente em manipuladores como o robô Delta, é fundamental garantir que os movimentos sejam suaves não só em termos de posição, mas também em velocidade e aceleração.

Embora os polinómios cúbicos permitam controlar posição e velocidade nos extremos de uma trajetória, não oferecem controlo sobre a aceleração. Por esse motivo, em aplicações onde a suavidade total do movimento é crucial, recorre-se ao polinómio de 5ª ordem, também conhecido como polinómio quíntuplo.

O polinómio quíntuplo é definido pela equação 2.11:

$$\theta(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4 + a_5 t^5 \tag{2.11}$$

As suas derivadas de 1ª e 2ª ordem — correspondentes à velocidade e aceleração — são enunciadas nas equações 2.12 e 2.13:

$$\dot{\theta}(t) = a_1 + 2a_2 t + 3a_3 t^2 + 4a_4 t^3 + 5a_5 t^4 \tag{2.12}$$

$$\ddot{\theta}(t) = 2a_2 + 6a_3 t + 12a_4 t^2 + 20a_5 t^3 \tag{2.13}$$

Este tipo de polinómio permite impor seis condições de contorno, três no instante inicial $t = 0$ e três no instante final $t = t_f$ (2.14):

$$\begin{aligned} \theta(0) &= \theta_0, & \theta(t_f) &= \theta_f \\ \dot{\theta}(0) &= \dot{\theta}_0, & \dot{\theta}(t_f) &= \dot{\theta}_f \\ \ddot{\theta}(0) &= \ddot{\theta}_0, & \ddot{\theta}(t_f) &= \ddot{\theta}_f \end{aligned} \tag{2.14}$$

Estas condições geram um sistema de seis equações lineares $Ax = b$, representado na forma matricial (2.15 e 2.16):

com:

$$A = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & t_f & t_f^2 & t_f^3 & t_f^4 & t_f^5 \\ 0 & 1 & 2t_f & 3t_f^2 & 4t_f^3 & 5t_f^4 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 6t_f & 12t_f^2 & 20t_f^3 \\ 0 & 0 & 2 & 0 & 0 & 0 \end{bmatrix} \tag{2.15}$$

$$x = \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \end{bmatrix}, \quad b = \begin{bmatrix} \theta_0 \\ \theta_f \\ \dot{\theta}_f \\ \dot{\theta}_0 \\ \ddot{\theta}_f \\ \ddot{\theta}_0 \end{bmatrix} \tag{2.16}$$

A solução explícita dos coeficientes(equações 2.17 a 2.22) é dada por:

$$a_0 = \theta_0 \tag{2.17}$$

$$a_1 = \dot{\theta}_0 \tag{2.18}$$

$$a_2 = \frac{1}{2}\ddot{\theta}_0 \tag{2.19}$$

$$a_3 = \frac{20(\theta_f - \theta_0) - (8\dot{\theta}_f + 12\dot{\theta}_0)t_f - (3\ddot{\theta}_0 - \ddot{\theta}_f)t_f^2}{2t_f^3} \tag{2.20}$$

$$a_4 = \frac{30(\theta_0 - \theta_f) + (14\dot{\theta}_f + 16\dot{\theta}_0)t_f + (3\ddot{\theta}_0 - 2\ddot{\theta}_f)t_f^2}{2t_f^4} \tag{2.21}$$

$$a_5 = \frac{12(\theta_f - \theta_0) - 6(\dot{\theta}_f + \dot{\theta}_0)t_f - (\ddot{\theta}_0 - \ddot{\theta}_f)t_f^2}{2t_f^5} \tag{2.22}$$

Apesar de requerer maior esforço computacional que polinómios de menor ordem, o polinómio quintuplo é essencial sempre que se deseja máxima suavidade.

2.3 Trajetória com Pontos Iniciais e Finais Dados θ_0 e θ_f

Em muitas aplicações práticas de planeamento de movimento, pretende-se uma trajetória que inicie e termine em repouso (velocidade e aceleração nulas), passando por uma fase de velocidade constante. Uma abordagem eficaz para este caso baseia-se na utilização de polinómios de 4ª ordem nas fases iniciais e finais, e uma reta na fase intermédia.

A trajetória é composta por três segmentos:

- Um segmento inicial (aceleração), entre $t = 0$ e $t = 2\epsilon$, gerado por um polinómio de 4ª ordem.
- Um segmento central (velocidade constante), entre $t = 2\epsilon$ e $t = t_f - 2\epsilon$, representado por uma equação linear.
- Um segmento final (desaceleração), entre $t = t_f - 2\epsilon$ e $t = t_f$, também gerado por um polinómio de 4ª ordem.

O parâmetro ϵ representa metade da duração das fases de aceleração e desaceleração.

Para definir a trajetória, consideram-se os seguintes pontos auxiliares:

- $\theta_{01} = \theta_0$ no instante $t = \epsilon$
- $\theta_{f2} = \theta_f$ no instante $t = t_f - \epsilon$

Os pontos θ_{02} e θ_{f1}, usados para definir os limites da fase de velocidade constante, são obtidos através da reta que liga θ_{01} a θ_{f2} :

$$\theta_{02} = \text{posição da reta em } t = 2\epsilon \tag{2.23}$$

$$\theta_{f1} = \text{posição da reta em } t = t_f - 2\epsilon \tag{2.24}$$

A trajetória completa é definida pelas seguintes expressões:

Segmento 1 — Polinómio de 4ª ordem (aceleração):

$$\theta(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4, \quad 0 \leq t < 2\epsilon \tag{2.25}$$

Este polinómio é construído de modo a satisfazer:

- $\theta(0) = \theta_0$
- $\dot{\theta}(0) = 0$
- $\ddot{\theta}(0) = 0$
- $\theta(2\epsilon) = \theta_{02}$
- $\dot{\theta}(2\epsilon) = v_{blend}$ (velocidade da reta)

Segmento 2 — Movimento linear (velocidade constante):

$$\theta(t) = v_{blend} \cdot t + c, \quad 2\epsilon \leq t \leq t_f - 2\epsilon \tag{2.26}$$

Segmento 3 — Polinómio de 4ª ordem (desaceleração):

$$\theta(t) = b_0 + b_1(t - t_f) + b_2(t - t_f)^2 + b_3(t - t_f)^3 + b_4(t - t_f)^4, \quad t_f - 2\epsilon < t \leq t_f \tag{2.27}$$

Este polinómio é construído para satisfazer:

- $\theta(t_f) = \theta_f$
- $\dot{\theta}(t_f) = 0$
- $\ddot{\theta}(t_f) = 0$

- $\theta(t_f - 2\epsilon) = \theta_{f1}$
- $\dot{\theta}(t_f - 2\epsilon) = v_{blend}$

Este método permite garantir uma transição suave e fisicamente consistente entre dois pontos, mesmo quando apenas as posições inicial e final são conhecidas. Ao usar polinómios de 4ª ordem nos extremos e um segmento linear central, assegura-se continuidade até à aceleração, o que é importante para aplicações com restrições dinâmicas em atuadores ou quando se pretende limitar vibrações mecânicas.

A escolha do parâmetro ϵ influencia a duração da aceleração/desaceleração, bem como o comprimento da fase de velocidade constante. A sua seleção deve considerar as capacidades do sistema físico e os requisitos temporais da tarefa.

2.3.1 Trajetória com Pontos Intermédios

Em muitas aplicações robóticas, não basta deslocar o robô apenas de um ponto inicial para um ponto final. Em vez disso, é necessário que a trajetória passe por um ou mais pontos intermédios definidos, garantindo um movimento suave e controlado ao longo de todo o caminho.

Esta abordagem consiste na combinação de segmentos de reta com polinómios de 4ª ordem, de forma a assegurar:

- Continuidade da posição e velocidade ao longo de toda a trajetória,
- Aceleração nula nos pontos de transição entre os diferentes segmentos.

Estrutura da Trajetória

Seja uma trajetória que passa pelos seguintes pontos:

$$\theta_0 \rightarrow \theta_1 \rightarrow \theta_2 \rightarrow \theta_f$$

Divide-se o tempo total em vários subintervalos: $[t_0, t_1], [t_1, t_2], [t_2, t_f]$.
A construção da trajetória é feita da seguinte forma:

1. Entre t_0 e um tempo auxiliar t_{02}, utiliza-se um polinómio de 4ª ordem.
2. Entre t_{02} e t_{11}, utiliza-se um segmento de reta (velocidade constante).
3. Entre t_{11} e t_1, outro polinómio de 4ª ordem, e assim por diante.
4. Os pontos auxiliares são posicionados de forma que a transição entre os polinómios e os segmentos lineares seja suave.

Polinómio de 4ª Ordem

A equação usada nos segmentos com aceleração (início e fim, por exemplo) é:

$$\theta(t) = a_0 + a_1(t - t_i) + a_2(t - t_i)^2 + a_3(t - t_i)^3 + a_4(t - t_i)^4 \tag{2.28}$$

Com as seguintes condições impostas:

- $\theta(t_i) = \theta_i$ (posição inicial),
- $\dot{\theta}(t_i) = 0$ (partida/paragem suave),
- $\ddot{\theta}(t_i) = 0$ (aceleração nula),

- $\theta(t_f) = \theta_f$ (posição final),
- $\dot{\theta}(t_f) = v$ (casamento com a reta),

A derivada de (2.28) fornece a velocidade:

$$\dot{\theta}(t) = a_1 + 2a_2(t - t_i) + 3a_3(t - t_i)^2 + 4a_4(t - t_i)^3 \tag{2.29}$$

E a segunda derivada, a aceleração:

$$\ddot{\theta}(t) = 2a_2 + 6a_3(t - t_i) + 12a_4(t - t_i)^2 \tag{2.30}$$

Segmento Linear

Nos segmentos centrais da trajetória, é usada uma equação linear:

$$\theta(t) = \theta_i + v(t - t_i) \tag{2.31}$$

onde v é a velocidade constante, igual à derivada dos polinómios nas fronteiras do segmento.

Exemplo Prático

Considere um movimento de uma junta que vai de $90°$ a $45°$ em $3\,s$, passando por um segmento linear entre os tempos $t = 0.6\,s$ e $t = 2.4\,s$, e com $\Delta t = 0.3\,s$ para aceleração e desaceleração.

Os pontos auxiliares são:

$$\begin{aligned}
\theta_0 = 90°, \quad \theta_{02} &= \frac{1875}{36}t^3 - \frac{9375}{216}t^4 \quad \text{(entre } 0 \le t \le 0.6\text{)} \\
\theta_{02\to f1} &= \frac{675}{8}t - \frac{150}{8} \quad \text{(entre } 0.6 \le t \le 2.4\text{)} \\
\theta_{f1\to f} &= \theta(t) = a_0 + a_1(t - 2.4) + a_3(t - 2.4)^3 + a_4(t - 2.4)^4 \quad \text{(entre } 2.4 \le t \le 3.0\text{)}
\end{aligned}$$

Com coeficientes calculados para garantir:

- Velocidade contínua nos pontos $0.6\,s$ e $2.4\,s$,
- Aceleração nula nas fronteiras.

Vantagens

- Permite transições suaves mesmo com múltiplos pontos de passagem.
- Evita acelerações abruptas.
- Evita o uso de polinómios de ordem elevada (que podem gerar oscilações).
- Controla a velocidade nos segmentos retos de forma clara.

Este método é ideal em aplicações industriais com trajetórias complexas, como soldadura, pintura automatizada ou manipulação de objetos com obstáculos.

2.4 Segmentos Lineares com Transições Parabólicas (LSPB)

Um dos métodos mais utilizados em aplicações industriais de planeamento de movimento é o perfil de velocidade trapezoidal, também conhecido como LSPB – Linear Segments with Parabolic Blend. Este método oferece uma transição suave entre pontos de partida e chegada, com controlo direto sobre a aceleração e velocidade máximas.

A trajetória é dividida em três partes: (1) aceleração constante, onde a posição segue um perfil parabólico; (2) velocidade constante, com posição linear; e (3) desaceleração constante, novamente com perfil parabólico. Assume-se que o tempo de aceleração e desaceleração é igual, o que resulta numa trajetória simétrica em torno do ponto médio de $[0, t_f]$.

Na fase de aceleração ($0 \leq t < t_c$), a posição evolui segundo um polinómio de 2ª ordem (equação (2.32)):

$$\theta_1(t) = a_{01} + a_{11}t + a_{21}t^2 \tag{2.32}$$

As derivadas da posição são (equações (2.33) e (2.34)):

$$\dot{\theta}_1(t) = a_{11} + 2a_{21}t \tag{2.33}$$

$$\ddot{\theta}_1(t) = 2a_{21} \tag{2.34}$$

Impondo as condições $\theta_1(0) = \theta_0$, $\dot{\theta}_1(0) = 0$ e $\dot{\theta}_1(t_c) = \dot{\theta}_c$, obtêm-se:

$$a_{01} = \theta_0 \tag{2.35}$$

$$a_{11} = 0 \tag{2.36}$$

$$a_{21} = \frac{\ddot{\theta}_c}{2} \tag{2.37}$$

A equação final para esta fase fica então (equação (2.38)):

$$\theta_1(t) = \theta_0 + \frac{\ddot{\theta}_c}{2}t^2 \tag{2.38}$$

Durante a fase de velocidade constante ($t_c \leq t \leq t_f - t_c$), a posição segue uma equação linear (equação (2.39)):

$$\theta_2(t) = a_{02} + a_{12}t \tag{2.39}$$

Com as condições de continuidade $\theta_2(t_c) = \theta_1(t_c)$ e $\dot{\theta}_2(t_c) = \dot{\theta}_c$, deduzem-se:

$$a_{12} = \dot{\theta}_c \tag{2.40}$$

$$a_{02} = \theta_0 - \frac{\dot{\theta}_c t_c}{2} \tag{2.41}$$

Logo, a equação da posição nesta fase é (equação (2.42)):

$$\theta_2(t) = \theta_0 + \dot{\theta}_c \left(t - \frac{t_c}{2}\right) \tag{2.42}$$

Na fase de desaceleração ($t_f - t_c < t \leq t_f$), a posição é novamente descrita por um polinómio de 2ª ordem (equação (2.43)):

$$\theta_3(t) = a_{03} + a_{13}t + a_{23}t^2 \tag{2.43}$$

Com derivadas (equações (2.44) e (2.45)):

$$\dot{\theta}_3(t) = a_{13} + 2a_{23}t \tag{2.44}$$

$$\ddot{\theta}_3(t) = 2a_{23} \tag{2.45}$$

Impondo $\theta_3(t_f) = \theta_f$, $\dot{\theta}_3(t_f) = 0$ e $\dot{\theta}_3(t_f - t_c) = \dot{\theta}_c$, obtêm-se:

$$a_{03} = \theta_f - \frac{\ddot{\theta}_c}{2}(t_f - t)^2 \tag{2.46}$$

$$a_{13} = \ddot{\theta}_c t_f \tag{2.47}$$

$$a_{23} = -\frac{\ddot{\theta}_c}{2} \tag{2.48}$$

Assim, a equação final desta fase é (equação (2.49)):

$$\theta_3(t) = \theta_f - \frac{\ddot{\theta}_c}{2}(t_f - t)^2 \tag{2.49}$$

A velocidade constante $\dot{\theta}_c$ necessária para garantir continuidade da posição ao longo de toda a trajetória é (equação (2.50)):

$$\dot{\theta}_c = \frac{\theta_f - \theta_0}{t_f - t_0 - t_c} \tag{2.50}$$

Finalmente, a expressão completa da trajetória é (equação (2.51)):

$$\theta(t) = \begin{cases} \theta_0 + \frac{\ddot{\theta}_c}{2}t^2, & 0 \leq t < t_c \\ \theta_0 + \dot{\theta}_c(t - \frac{t_c}{2}), & t_c \leq t \leq t_f - t_c \\ \theta_f - \frac{\ddot{\theta}_c}{2}(t_f - t)^2, & t_f - t_c < t \leq t_f \end{cases} \tag{2.51}$$

Este perfil é amplamente utilizado devido à sua simplicidade e à sua capacidade de limitar o esforço dinâmico imposto ao sistema robótico, garantindo transições suaves entre as fases da trajetória.

2.5 Trajetória Cicloidal

Uma alternativa ao uso de polinómios de ordem elevada, como os quíntuplos, que podem induzir picos excessivos na parte intermédia da trajetória, é a chamada trajetória cicloidal. Este tipo de perfil permite garantir que a velocidade e aceleração sejam nulas nos instantes inicial e final, oferecendo uma transição suave sem necessidade de polinómios complexos.

A equação da trajetória cicloidal é dada por (Equação 2.52):

$$\theta(t) = \theta_0 + (\theta_f - \theta_0)\left(\frac{t}{T} - \frac{1}{2\pi}\sin\left(\frac{2\pi t}{T}\right)\right) \tag{2.52}$$

onde:

- θ_0 e θ_f são os ângulos inicial e final da junta;
- $T = t_f - t_0$ é o tempo total de movimento;
- t é o tempo corrente dentro do intervalo $[0, T]$.

As derivadas temporais desta expressão são dadas pelas Equações 2.53 e 2.54:

$$\dot{\theta}(t) = (\theta_f - \theta_0)\left(\frac{1}{T} - \frac{1}{T}\cos\left(\frac{2\pi t}{T}\right)\right) \tag{2.53}$$

$$\ddot{\theta}(t) = (\theta_f - \theta_0)\left(\frac{2\pi}{T^2}\sin\left(\frac{2\pi t}{T}\right)\right) \tag{2.54}$$

Esta forma de trajetória apresenta uma evolução suave da posição, com crescimento progressivo da velocidade até um valor máximo e subsequente redução, mantendo continuidade em todas as derivadas até à aceleração. Em particular, verifica-se que:

- $\dot{\theta}(0) = \dot{\theta}(T) = 0$
- $\ddot{\theta}(0) = \ddot{\theta}(T) = 0$

A suavidade destas transições torna a trajetória cicloidal particularmente atrativa para movimentos repetitivos e contínuos em robótica. Além disso, é fácil de implementar computacionalmente, e o seu comportamento regular evita descontinuidades ou esforços abruptos nos atuadores.

2.6 Trajetória Spline

Ao planear uma trajetória que passa por N pontos intermédios, uma solução direta seria ajustar um único polinómio de grau $N-1$. No entanto, essa abordagem tende a gerar oscilações indesejadas, especialmente quando N é grande. Para evitar tal comportamento, recorre-se ao uso de splines cúbicas, isto é, segmentos polinomiais de grau três ajustados entre pares sucessivos de pontos.

Cada par consecutivo de pontos (t_k, θ_k) e (t_{k+1}, θ_{k+1}) é ligado por um polinómio cúbico $\theta_k(t)$ da forma dada pela Equação (2.55):

$$\theta_k(t) = a_{0k} + a_{1k}(t - t_k) + a_{2k}(t - t_k)^2 + a_{3k}(t - t_k)^3 \tag{2.55}$$

A derivada de primeira ordem da posição (velocidade) é expressa pela Equação (2.56):

$$\dot{\theta}_k(t) = a_{1k} + 2a_{2k}(t - t_k) + 3a_{3k}(t - t_k)^2 \tag{2.56}$$

A derivada de segunda ordem da posição (aceleração) é dada pela Equação (2.57):

$$\ddot{\theta}_k(t) = 2a_{2k} + 6a_{3k}(t - t_k) \tag{2.57}$$

Ao impor as condições no instante inicial $t = t_k$ de cada segmento, obtêm-se diretamente três dos quatro coeficientes do polinómio, como mostrado na Equação (2.58):

$$a_{0k} = \theta_k, \quad a_{1k} = \dot{\theta}_k, \quad a_{2k} = \frac{1}{2}\ddot{\theta}_k \tag{2.58}$$

Para assegurar continuidade em posição, velocidade e aceleração entre dois segmentos consecutivos nos nós, impõem-se as condições da Equação (2.59):

$$\theta_k(t_{k+1}) = \theta_{k+1}, \quad \dot{\theta}_k(t_{k+1}) = \dot{\theta}_{k+1}, \quad \ddot{\theta}_k(t_{k+1}) = \ddot{\theta}_{k+1} \tag{2.59}$$

A partir da Equação (2.57), e impondo continuidade na aceleração em $t = t_{k+1}$, pode-se isolar o coeficiente a_{3k} como função das acelerações nos extremos do intervalo, resultando na Equação (2.60):

$$6a_{3k}h_k = \ddot{\theta}_{k+1} - \ddot{\theta}_k \tag{2.60}$$

onde $h_k = t_{k+1} - t_k$.

Em seguida, para garantir continuidade da posição, escreve-se a Equação (2.61), que impõe a igualdade da posição no final do segmento com o valor conhecido:

$$\theta_{k+1} = \theta_k + \dot{\theta}_k h_k + \frac{1}{2}\ddot{\theta}_k h_k^2 + a_{3k} h_k^3 \tag{2.61}$$

Reformulando, é possível expressar a_{1k} (se necessário) em termos de posições e acelerações.

Para assegurar continuidade da velocidade entre segmentos consecutivos no nó t_k, impõe-se:

$$\dot{\theta}_k(t_k) = \dot{\theta}_{k-1}(t_k) \tag{2.62}$$

A substituição das expressões anteriores conduz à formulação de um sistema linear que relaciona todas as acelerações nos nós. Este sistema pode ser expresso na forma matricial da Equação (2.63):

$$A \cdot \ddot{q} = 6B \cdot q \tag{2.63}$$

em que A e B são matrizes $(N-2) \times N$, e q representa o vetor de posições nos nós.

Como o sistema é subdeterminado (menos equações do que incógnitas), é necessário especificar duas acelerações adicionais. Uma prática comum é definir $\ddot{\theta}_1 = \ddot{\theta}_N = 0$, gerando uma spline natural. Contudo, esta abordagem não oferece controlo sobre as velocidades nos extremos.

Caso se pretenda controlar também as velocidades e acelerações iniciais e finais, pode-se recorrer à introdução de pontos virtuais ou imposição de mais condições de continuidade, garantindo que o sistema (2.63) se torne bem determinado.

São amplamente utilizadas em robótica para o planeamento de trajetórias que passam por múltiplos pontos intermédios com elevada suavidade e controlo dinâmico.

Planeamento no Espaço Cartesiano

O planeamento de movimento no espaço cartesiano consiste em especificar o movimento desejado de um robô diretamente em termos da posição e orientação do seu efetor final (por exemplo, a ferramenta na extremidade do braço robótico), em vez de controlar diretamente os ângulos das suas juntas. Esta abordagem é altamente vantajosa quando se pretende que a trajetória da extremidade do robô siga uma linha, um arco, ou execute tarefas que exigem precisão espacial, como soldadura, pintura, inspeção ou montagem.

Definição da Trajetória Cartesiana

A posição do efetor final é representada por um vetor função do tempo:

$$\mathbf{p}(t) = \begin{bmatrix} x(t) \\ y(t) \\ z(t) \end{bmatrix} \tag{3.1}$$

onde $x(t)$, $y(t)$ e $z(t)$ são as coordenadas espaciais da extremidade do robô ao longo do tempo.
A orientação pode ser representada de diversas formas, tais como:

- Matrizes de rotação (t), que definem a orientação em relação a um referencial fixo;
- Ângulos de Euler $\phi(t)$, $\theta(t)$ e $\psi(t)$, que descrevem rotações sucessivas;
- Quaterniόns, uma representação compacta e livre de singularidades para rotações tridimensionais.

Métodos de Interpolação

Para gerar uma trajetória contínua e suave, tanto a posição como a orientação podem ser interpoladas ao longo do tempo. Entre os métodos mais utilizados destacam-se:

- Polinómios cúbicos ou quinticos, que garantem continuidade de posição, velocidade e eventualmente aceleração;
- Segmentos lineares com blends parabólicos (LSPB), que oferecem perfis de velocidade trapezoidal e transições suaves;
- Trajetórias cicloidais, úteis quando se deseja limitar a aceleração;
- Splines cúbicos, recomendadas quando há múltiplos pontos intermédios a percorrer.

Na orientação, é necessário ter especial cuidado com descontinuidades. Por exemplo, o uso de ângulos de Euler pode levar ao gimbal lock, uma situação em que se perde um grau de liberdade. Por este motivo, os quaterniόns são muitas vezes preferidos, especialmente quando se interpola entre orientações arbitrárias.

Conversão para o Espaço das Juntas

Embora o planeamento seja feito no espaço cartesiano, o robô é controlado através das suas juntas (ângulos ou translações internas). Assim, é necessário converter a trajetória desejada no espaço cartesiano para o espaço das juntas usando cinemática inversa.

Seja (t) a posição desejada e (t) a orientação desejada, o vetor de variáveis articulares (t) é obtido por:

$$\mathbf{q}(t) = IK(\mathbf{p}(t), \mathbf{R}(t)), \tag{3.2}$$

onde $IK(\cdot)$ representa o operador de cinemática inversa. Esta operação pode ter múltiplas soluções, nenhuma solução, ou ser sensível a pequenas variações em $\mathbf{p}(t)$, especialmente em regiões próximas de singularidades.

2.2.4 Problemas e Singularidades

Uma das maiores dificuldades no planeamento cartesiano é a ocorrência de singularidades, ou seja, configurações onde o Jacobiano do robô perde posto e, consequentemente, o robô perde graus de liberdade efetivos. Nestas situações:

- O movimento desejado no espaço cartesiano pode não ter solução no espaço das juntas;
- Pequenas variações na posição desejada causam grandes variações nas juntas;
- Pode ser impossível manter orientação constante enquanto se segue a trajetória.

Além disso, a resolução da cinemática inversa pode ser computacionalmente intensiva, especialmente para robôs com muitos graus de liberdade ou com estrutura complexa.

Vantagens do Planeamento Cartesiano

Apesar das dificuldades, o planeamento no espaço cartesiano oferece importantes vantagens:

- Permite especificar o movimento de forma mais intuitiva e direta;
- Facilita a integração com ferramentas de CAD ou interfaces gráficas;
- É ideal para tarefas onde o caminho da ferramenta (e não apenas os pontos inicial e final) é crítico;
- Torna mais fácil a aplicação de restrições geométricas, como evitar obstáculos no espaço tridimensional.

Por estas razões, é uma técnica amplamente utilizada em contextos industriais e académicos, embora deva ser sempre acompanhada de um módulo robusto de resolução de cinemática inversa e de mecanismos de deteção de singularidades.

3.0.1 Movimento ponto-a-ponto vs. movimento com caminho contínuo

No planeamento de movimentos para manipuladores robóticos, é essencial distinguir entre dois modos operacionais fundamentais: o movimento ponto-a-ponto e o movimento com caminho contínuo. Esta distinção prende-se com a forma como a trajetória do robô é especificada e controlada ao longo do tempo, refletindo diferentes exigências consoante a aplicação em causa.

O movimento ponto-a-ponto, também conhecido como point-to-point (PTP), consiste em deslocar o manipulador de uma configuração inicial para uma configuração final, sem qualquer preocupação com o caminho percorrido entre os dois instantes. Neste modo, apenas os estados inicial e final, normalmente expressos em termos das variáveis articulares, são especificados. O movimento é então gerado de forma a ligar esses dois pontos, respeitando eventuais restrições de velocidade e aceleração. Cada junta pode ser controlada de forma independente, por exemplo, utilizando uma interpolação polinomial de terceira ordem do tipo

$$\theta_i(t) = a_{0i} + a_{1i}t + a_{2i}t^2 + a_{3i}t^3, \tag{3.3}$$

cujos coeficientes são calculados a partir das condições de contorno impostas no início e no fim do movimento. Neste contexto, a posição do efetor final no espaço cartesiano não é diretamente controlada durante o trajeto, o que significa que o robô pode descrever trajetórias não previsíveis entre os pontos extremos. Apesar desta limitação, o movimento ponto-a-ponto é amplamente utilizado em tarefas como transporte de objetos, alimentação de máquinas ou operações de pick-and-place, onde apenas interessa atingir os pontos alvo, e não o percurso exato entre eles.

Em contraste, o movimento com caminho contínuo, também referido como continuous path (CP), exige que o robô siga uma trajetória pré-definida no espaço cartesiano, tanto em termos de posição como, quando aplicável, de orientação. Este tipo de movimento é essencial em aplicações que envolvem tarefas como soldadura, pintura, corte ou inspeção, onde o percurso da ferramenta deve ser cuidadosamente controlado. Neste caso, a trajetória é especificada ao longo do tempo através de funções contínuas da posição cartesiana, como

$$\mathbf{p}(t) = \begin{bmatrix} x(t) \\ y(t) \\ z(t) \end{bmatrix}, \tag{3.4}$$

e eventualmente da orientação, representada, por exemplo, por uma matriz de rotação $\mathbf{R}(t)$. A construção da trajetória pode recorrer a métodos de interpolação como polinómios cúbicos ou quinticos, segmentos lineares com blends parabólicos, funções cicloidais ou curvas spline, dependendo do grau de suavidade pretendido.

Uma vez definida a trajetória no espaço cartesiano, torna-se necessário convertê-la para o espaço das juntas através da resolução da cinemática inversa. Esta operação deve ser realizada em todos os instantes da trajetória, o que aumenta significativamente a complexidade computacional do planeamento. Além disso, a presença de singularidades pode dificultar ou mesmo impedir a obtenção de soluções viáveis para certas configurações desejadas. Ainda assim, o movimento com caminho contínuo oferece um controlo muito mais preciso sobre o comportamento do robô no espaço de trabalho, permitindo, por exemplo, evitar obstáculos e garantir conformidade com geometrias complexas.

Comparando os dois modos, observa-se que o movimento ponto-a-ponto apresenta menor complexidade e maior eficiência computacional, sendo indicado para tarefas em que a trajetória intermédia não é relevante. Já o movimento com caminho contínuo, embora mais exigente do ponto de vista do planeamento e controlo, é indispensável sempre que se pretende garantir a fidelidade da trajetória seguida pelo robô, tanto para efeitos funcionais como de segurança.

Aplicação ao Robô Delta

Este capítulo descreve a aplicação dos métodos de planeamento de movimento, apresentados nos capítulos anteriores, ao controlo de um robô Delta. Tendo em conta as suas características cinemáticas e a elevada rapidez com que executa movimentos, é essencial garantir trajetórias suaves e bem comportadas, tanto no espaço das juntas como no espaço cartesiano.

4.1 Escolha do método de planeamento

Durante o estudo dos diversos tipos de trajetórias, incluindo trajetórias retas, polinómios cúbicos e quíntuplos, trajetórias com pontos intermédios, LSPB, cicloidais e splines — avaliou-se o seu impacto no desempenho e estabilidade do robô. Para efeito de demonstração, foi selecionado o método de planeamento baseado em polinómios cúbicos, cuja estrutura matemática e propriedades de continuidade o tornam apropriado para robôs com movimentos rápidos e precisos como o Delta.

A escolha deste método justifica-se pela sua capacidade de garantir transições suaves entre os estados inicial e final, minimizando descontinuidades na velocidade e aceleração.

Neste método, podemos controlar desde as velocidades finais e inicias como também acelerações permitindo ter um controlo mais detalhado do movimento, como referido anteriormente.

4.2 Implementação em `MATLAB`

A trajetória de movimento do atuador foi definida utilizando a técnica de polinómios quíntuplos, uma abordagem clássica para o planeamento de movimento suave em sistemas robóticos. Este método permite especificar seis condições de contorno: posição, velocidade e aceleração, tanto no início como no fim do movimento.

Para a implementação em MATLAB, foi criada uma função `trajetoria_quintupla` que recebe os parâmetros de entrada em graus, a posição inicial e final, velocidades e acelerações nas extremidades e o tempo total de movimento. Internamente, todos os cálculos são efetuados em radianos, garantindo consistência matemática, sendo a conversão de e para graus realizada apenas na `main`, conforme ilustrado na Figura 4.1.

```
%% Calculo da trajetoria

p0 = 0;     % posição inicial (graus)
pf = 90;    % posição final (graus)
v0 = 0;     % velocidade inicial (graus/s)
vf = 0;     % velocidade final (graus/s)
a0 = 0;     % aceleração inicial (graus/s²)
af = -50;     % aceleração final (graus/s²)
tf = 4;     % tempo total (segundos)

[P, v, a] = trajetoria_quintupla(p0, pf, v0, vf, a0, af, tf);
```

Figura 4.1: Parâmetros da trajetória e chamada da função `trajetoria_quintupla`.

4.2.1 Definição do polinómio quíntuplo

A trajetória é definida por um polinómio de quinta ordem da forma:

$$p(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4 + a_5 t^5$$

Os coeficientes $a_0, \ldots, a_5$ são calculados resolvendo um sistema linear de seis equações, que impõem as condições de posição, velocidade e aceleração nos instantes $t = 0$ e $t = t_f$. A Figura 4.2 apresenta a estrutura do sistema em código MATLAB.

```
% --- Sistema linear para polinómio quíntuplo ---
A = [1 0 0 0 0 0;
     0 1 0 0 0 0;
     0 0 2 0 0 0;
     1 tf tf^2 tf^3 tf^4 tf^5;
     0 1 2*tf 3*tf^2 4*tf^3 5*tf^4;
     0 0 2 6*tf 12*tf^2 20*tf^3];

b = [p0; v0; a0; pf; vf; af];
a = A \ b;

% Polinómios
p_rad = @(t) a(1) + a(2)*t + a(3)*t.^2 + a(4)*t.^3 + a(5)*t.^4 + a(6)*t.^5;
v_rad = @(t)        a(2) + 2*a(3)*t + 3*a(4)*t.^2 + 4*a(5)*t.^3 + 5*a(6)*t.^4;
acc_rad = @(t)               2*a(3) + 6*a(4)*t + 12*a(5)*t.^2 + 20*a(6)*t.^3;
```

Figura 4.2: Sistema linear para o cálculo dos coeficientes do polinómio quíntuplo.

4.2.2 Construção das funções da trajetória

Após o cálculo dos coeficientes, são definidas três funções anónimas em MATLAB que representam, respetivamente, a posição $p(t)$, a velocidade $v(t)$ e a aceleração $a(t)$. Estas funções operam internamente em radianos e são convertidas para graus apenas no final. Além disso, as funções estão limitadas ao intervalo temporal $[0, t_f]$, assegurando que a posição inicial se mantém antes do início e a posição final após o fim do movimento.

4.2.3 Geração e visualização da trajetória

A avaliação da trajetória é feita num vetor de tempo definido no intervalo estendido $[-1, t_f + 1]$, para permitir observar o comportamento antes e depois do intervalo útil. A Figura 4.3 apresenta o código utilizado para gerar os gráficos.

```
% Graficos
t_vals = linspace(-1, tf + 1, 1000);
figure;
subplot(3,1,1); plot(t_vals, p(t_vals), 'LineWidth', 2); ylabel('Posição (°)'); title('Trajetória Quíntupla'); grid on;
subplot(3,1,2); plot(t_vals, v(t_vals), 'LineWidth', 2); ylabel('Velocidade (°/s)'); grid on;
subplot(3,1,3); plot(t_vals, a(t_vals), 'LineWidth', 2); ylabel('Aceleração (°/s²)'); xlabel('Tempo (s)'); grid on;
```

Figura 4.3: Geração dos gráficos de posição, velocidade e aceleração em função do tempo.

Como resultado, obtêm-se três curvas suaves e contínuas, representando o comportamento do atuador ao longo do tempo. A continuidade da posição, velocidade e aceleração assegura que o movimento é fisicamente realista e adequado para sistemas mecânicos reais.

4.3 Integração com Simscape Multibody

Para integrar a trajetória gerada em `MATLAB` com o `Simscape Multibody`, a função `trajetoria_quintupla` foi projetada para produzir funções contínuas $P(t)$, $v(t)$ e $a(t)$, que podem ser utilizadas como entradas de controlo do modelo físico.

Existem duas abordagens principais para esta integração:

a) Utilização do bloco `MATLAB Function`

A função $P(t)$ pode ser inserida diretamente num bloco `MATLAB Function` no `Simulink`. Esta função retorna a posição desejada do atuador em cada instante de simulação. O sinal de saída é ligado à entrada de movimento de uma junta (por exemplo, `Revolute Joint`), configurada para aceitar movimento imposto (`Motion > Provided by Input`). O sistema calcula automaticamente o torque necessário para seguir a trajetória definida.

b) Utilização do bloco `From Workspace`

Alternativamente, a trajetória pode ser exportada como um vetor de tempo e posição no formato $[t, P(t)]$, e importada no modelo através do bloco `From Workspace`. Esta abordagem é especialmente útil quando se pretende utilizar dados previamente calculados ou quando se trabalha com controladores externos. Neste caso, a junta segue diretamente os valores fornecidos, desde que esteja configurada para receber a posição como entrada.

Conclusão

O presente relatório abordou de forma estruturada o estudo do planeamento de trajetórias no robô Delta, com ênfase na implementação de trajetórias suaves e fisicamente realistas através do uso de polinómios quíntuplos. Esta abordagem permitiu controlar não apenas a posição, mas também a velocidade e a aceleração da junta robótica nos instantes inicial e final, assegurando uma transição contínua e livre de descontinuidades dinâmicas.

Através do desenvolvimento de uma função dedicada em `MATLAB`, foi possível automatizar a geração de trajetórias em função de parâmetros definidos pelo utilizador, com visualização direta dos perfis de movimento e respetiva exportação para utilização em modelos físicos no `Simscape Multibody`. A integração no ambiente Simulink revelou-se eficiente e versátil, podendo ser realizada através de blocos como `MATLAB Function` ou `From Workspace`, permitindo validar o comportamento do atuador de forma realista.

Para além da implementação do polinómio quíntuplo, foram também estudadas outras abordagens relevantes, como trajetórias com segmentos lineares e transições parabólicas (LSPB), bem como trajetórias com pontos intermédios. Estas alternativas permitem uma maior flexibilidade na construção de movimentos adaptados a tarefas específicas ou restrições físicas do sistema.

O estudo demonstrou a importância de selecionar métodos de planeamento de movimento adequados às exigências da aplicação, considerando fatores como suavidade, controlo dinâmico e facilidade de implementação. Conclui-se que o uso de polinómios quíntuplos é particularmente indicado para aplicações que exigem elevada continuidade e suavidade no movimento, como é o caso do robô Delta, permitindo uma execução eficiente e segura das trajetórias planeadas.