Sistemas Robóticos Autônomos

Geração de Trajetórias

Geração de Trajetórias

- Controladores de trajetória requerem que uma trajetória de referência seja especificada.

- Trajetória: caminho associado a restrições temporais. Em muitos casos, a trajetória de referência deve possuir primeiras derivadas contínuas (velocidade e aceleração).

- Para robôs não holonômicos, a trajetória gerada deve respeitar as restrições não holonômicas.

- Outras possíveis restrições: velocidades e/ou acelerações máximas admissíveis ao robô, comprimento mínimo, raio de giro mínimo, etc.

Geração por Polinômios Cúbicos

- Vantagens:
  - Trajetória gerada por fórmulas simples e de cálculo extremamente rápido
  - Gera movimentos sem chaveamentos ou descontinuidades na velocidade
  - Adequado para veículos de raio de giro nulo (acionamento diferencial) e para situações onde o objetivo final é móvel

Geração por Polinômios Cúbicos

- Desvantagens:
  - Não garante um raio de giro mínimo, embora a otimização tenda a gerar trajetórias com rotações suaves.
  - O teste de colisão é um pouco mais complexo
  - Definição da velocidade requer cálculo de relação $\lambda(t)$ (λ parâmetro da curva, t tempo).

• Geração por Polinômios Cúbicos

- As variáveis de configuração x e y são dadas por:

  $x(\lambda) = a_0 + a_1.\lambda + a_2.\lambda^2 + a_3.\lambda^3$
  $y(\lambda) = b_0 + b_1.\lambda + b_2.\lambda^2 + b_3.\lambda^3$

- a_0, a_1, a_2, a_3, b_0, b_1, b_2 e b_3 são parâmetros a determinar.
  $\lambda = 0$ quando $(x(0), y(0)) = (x_i, y_i)$,
  $\lambda = 1$ quando $(x(1), y(1)) = (x_f, y_f)$.

- Orientação θ deve satisfazer as restrições não holonômicas:

  $\theta(\lambda) = \tan^{-1}(dy/dx) = \tan^{-1}((dy/d\lambda)/(dx/d\lambda))$

- Definindo $\alpha(\lambda) = (b_1 + 2.b_2.\lambda + 3.b_3.\lambda^2)/(a_1 + 2.a_2.\lambda + 3.a_3.\lambda^2)$

  $\Rightarrow \quad \theta(\lambda) = \tan^{-1}(\alpha(\lambda)) \qquad \Rightarrow \quad \alpha(\lambda) = \tan(\theta(\lambda))$

Geração por Polinômios Cúbicos

- Há 6 (seis) condições de contorno e 8 (oito) coeficientes a determinar.

- Possibilidade de otimização: *x* e/ou *y* monotonicamente crescentes:
    - Gera trajetórias mais “inteligentes”: robô não se afasta do alvo (andando para trás).
    - Confina a trajetória ao retângulo (x_i,y_i) – (x_f,y_f) se garantida nas duas dimensões.

- Matematicamente: soluções para as quais o polinômio interpolador não apresente máximos em x e/ou y.

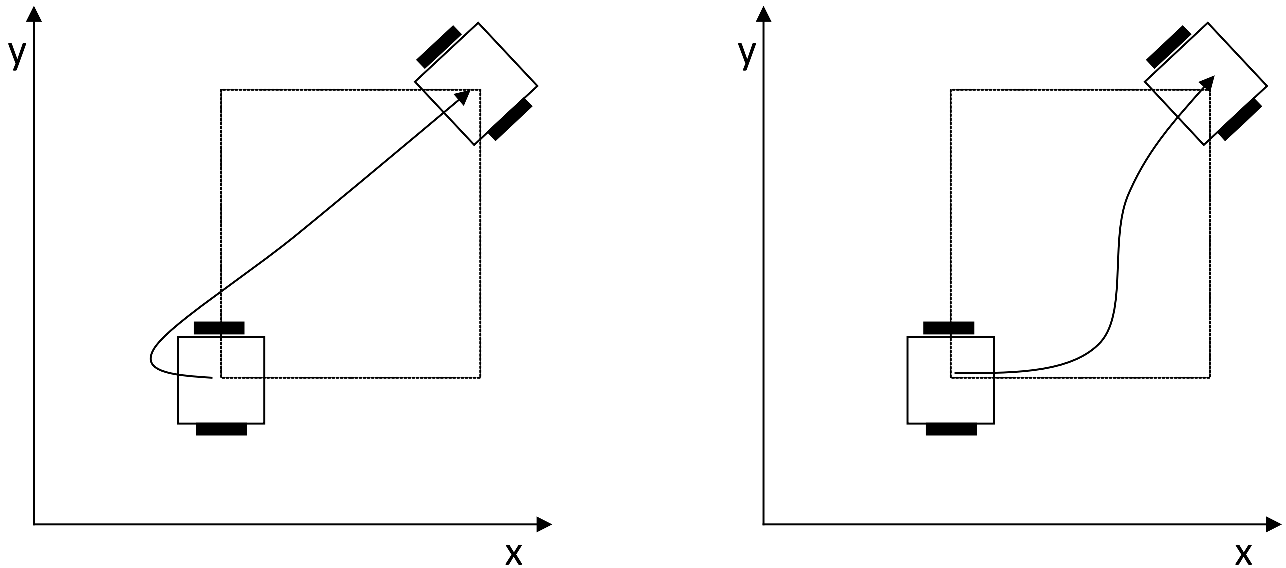
Geração por Polinômios Cúbicos

- Aplicando as seis condições de contorno de modo a impor que a curva inicie em $q_i = (x_i, y_i, \theta_i)$ e termine em $q_f = (x_f, y_f, \theta_f)$, obtemos um sistema de seis equações e oito incógnitas (a_0, a_1, a_2, a_3, b_0, b_1, b_2, b_3).

- Dois parâmetros são livres: podem ser usados para escolher solução que otimize algum critério específico.

- Trajetórias “mais inteligentes” podem ser geradas.

Geração por Polinômios Cúbicos



a) Sem otimização.

b) Com otimização.

Algoritmo de Geração por Polinômios cúbicos

- Dependendo das condições de contorno, para evitar divisões por zero na solução, torna-se necessário adotar diferentes parâmetros livres.

- Definindo:

$\Delta x = x_f - x_i$ $\Delta y = y_f - y_i$

δ = pequeno intervalo angular em torno da singularidade

$\Rightarrow$ Para efeitos práticos: $\delta \cong 1^o$

Algoritmo de Geração por Polinômios cúbicos

i. Se $\theta_i \in [(\pi/2 - \delta), (\pi/2 + \delta)]$ **e** $\theta_f \in [(\pi/2 - \delta), (\pi/2 + \delta)]$

$b_1 = \Delta y$ (coeficiente livre)

$b_2 = 0$ (coeficiente livre)

$a_0 = x_i$

$a_1 = 0$

$a_2 = 3.\Delta x$

$a_3 = -2.\Delta x$

$b_0 = y_i$

$b_3 = \Delta y - b_1 - b_2$

Algoritmo de Geração por Polinômios cúbicos

ii. Senão, se $\theta_i \in [(\pi/2 - \delta), \ (\pi/2 + \delta)]$

$a_3 = -\Delta x/2$ (coeficiente livre)

$b_3 =$ qualquer valor (coeficiente livre)

$a_0 = x_i$

$a_1 = 0$

$a_2 = \Delta x - a_3$

$b_0 = y_i$

$b_1 = 2(\Delta y - \alpha_f.\Delta x) - \alpha_f.a_3 + b_3$

$b_3 = (2.\alpha_f.\Delta x - \Delta y) + \alpha_f.a_3 - 2.b_3$

Algoritmo de Geração por Polinômios cúbicos

iii. Senão, se $\theta f \in [(\pi/2 - \delta), (\pi/2 + \delta)]$

$a1 = 3.\Delta x/2$ (coeficiente livre)

$b2 =$ qualquer valor (coeficiente livre)

$a0 = xi$

$a2 = 3.\Delta x – 2.a1$

$a3 = a1 – 2.\Delta x$

$b0 = yi$

$b1 = \alpha i.a1$

$b3 = \Delta y - \alpha i.a1 – b2$

Algoritmo de Geração por Polinômios cúbicos

iv. Senão

$a1 = \Delta x$ (coeficiente livre)

$a2 = 0$ (coeficiente livre)

$a0 = xi$

$a3 = \Delta x - a1 - a2$

$b0 = yi$

$b1 = \alpha i.a1$

$b2 = 3(\Delta y - \alpha f.\Delta x) + 2.(\alpha f - \alpha i).a1 + \alpha f.a2$

$b3 = 3.\alpha f.\Delta x - 2.\Delta y - (2.\alpha f - \alpha i).a1 - \alpha f.a2$

Algoritmo de Geração por Polinômios cúbicos

v. Fazer λ variar de 0 a 1

vi. Calcular

$$x(\lambda) = a_0 + a_1.\lambda + a_2.\lambda^2 + a_3.\lambda^3$$

$$y(\lambda) = b_0 + b_1.\lambda + b_2.\lambda^2 + b_3.\lambda^3$$

$$\theta(\lambda) = \tan^{-1}((b_1 + 2.b_2.\lambda + 3.b_3.\lambda^2)/(a_1 + 2.a_2.\lambda + 3.a_3.\lambda^2))$$

Algoritmo de Geração por Polinômios cúbicos

Observações:

- Os passos i, ii e iii correspondem a situações singulares (θ_i e/ou θ_f iguais a $\pm 90^o$).
- O passo iv corresponde ao caso geral (caso não singular).
- Os passos v e vi correspondem à geração do caminho ponto a ponto propriamente dito.

Reparametrização de Trajetória
Exemplo: Polinômios cúbicos

- O parâmetro λ é adimensional, $(0 \leq \lambda \leq 1)$.

- A velocidade com que o caminho é percorrido dado incrementos em λ depende de como a curva varia em função de λ.

- para garantir que a trajetória seja percorrida a uma determinada velocidade, torna-se necessário reparametrizar a trajetória.

- É necessário associar λ com o tempo, de acordo com algum perfil de velocidade especificado para a curva geométrica $(x(\lambda),y(\lambda))$.

Reparametrização de Trajetória
Exemplo: Polinômios cúbicos

- Dado o caminho:

$x(\lambda) = a_0 + a_1.\lambda + a_2.\lambda^2 + a_3.\lambda^3$

$y(\lambda) = b_0 + b_1.\lambda + b_2.\lambda^2 + b_3.\lambda^3$

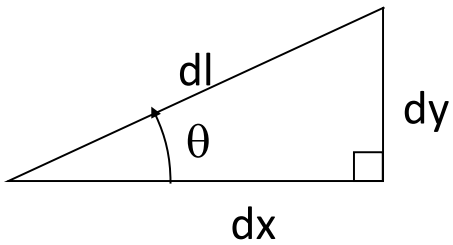
- Definindo o operador $D(.) = d(.)/d\lambda$, temos:

$Dx = dx(\lambda)/d\lambda = a_1 + 2.a_2.\lambda + 3.a_3.\lambda^2$

$Dy = dy(\lambda)/d\lambda = b_1 + 2.b_2.\lambda + 3.b_3.\lambda^2$

Reparametrização de Trajetória
Exemplo: Polinômios cúbicos

- Para um deslocamento infinitesimal, dl, do robô:



$dl = [(dx)^2 + (dy)^2]^{1/2} = [(Dx)^2 + (Dy)^2]^{1/2}.d\lambda$

$\Rightarrow d\lambda/dl = 1/[(Dx)^2 + (Dy)^2]^{1/2}$

$\Rightarrow \cos(\theta) = (dx/dl) = (dx/d\lambda).(d\lambda/dl) = Dx/[(Dx)^2 + (Dy)^2]^{1/2}$

$\Rightarrow \text{sen}(\theta) = (dy/dl) = (dy/d\lambda).(d\lambda/dl) = Dy/[(Dx)^2 + (Dy)^2]^{1/2}$

$\Rightarrow \tan(\theta) = Dy/Dx$

Reparametrização de Trajetória
Exemplo: Polinômios cúbicos

- A velocidade linear é dada por:

  $v(t) = dl/dt = (dl/d\lambda).(d\lambda/dt) = [(Dx)^2 + (Dy)^2]^{1/2}.d\lambda/dt$

  $\Rightarrow d\lambda/dt = v(t)/[(Dx)^2 + (Dy)^2]^{1/2}$

- Assim, dado o perfil de velocidade v(t), a expressão acima pode ser integrada de modo a reparametrizar a trajetória, (dado t e o perfil v(t), obter o valor correspondente de λ).

Reparametrização de Trajetória
Exemplo: Polinômios cúbicos

- O Comprimento percorrido, l, é:

  $l = \int v(t).dt = \int (dl/dt).dt = \int [(Dx)^2 + (Dy)^2]^{1/2}.d\lambda$

  O intervalo de integração para t é $[0, t_{max}]$, que corresponde ao intervalo [0, 1] de variação de λ, sendo t_{max} a duração especificada para a trajetória.

Reparametrização de Trajetória
Exemplo: Polinômios cúbicos

- A velocidade angular para um dado λ, pode ser obtida a partir da velocidade linear, através :

  $d(\tan(\theta))/dt = (1/\cos^2(\theta)).d\theta/dt = = (1/\cos^2(\theta)).\omega$

  $\Rightarrow \omega = \cos^2(\theta).[d(\tan(\theta))/d\lambda].d\lambda/dt$

  mas, $\cos^2(\theta) = (Dx)^2/[(Dx)^2 + (Dy)^2]$, $\tan(\theta) = Dy/Dx$. Assim:

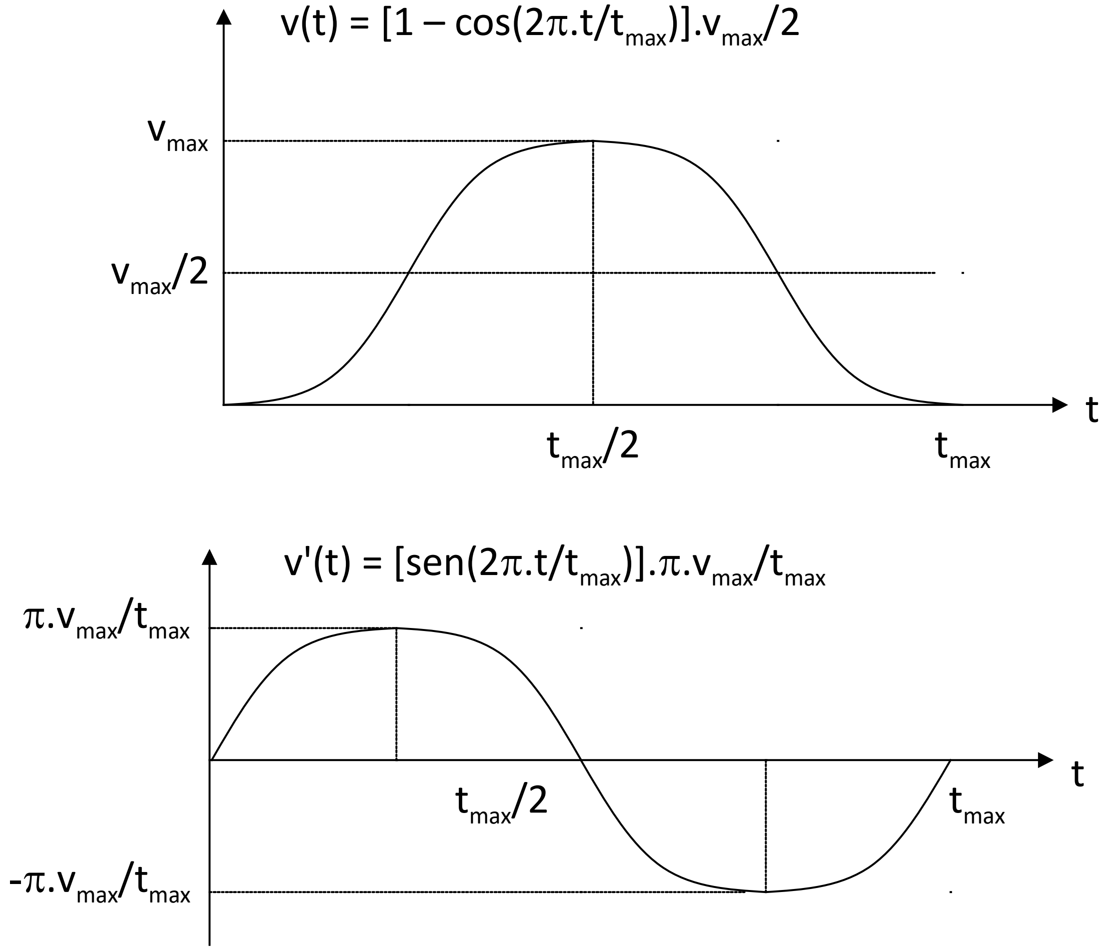
  $\omega(t) = v(t).[D^2y.Dx - D^2x.Dy]/[(Dx)^2 + (Dy)^2]^{3/2}$

  onde: $D^2x = D(Dx) = 2.a_2 + 6.a_3.\lambda$ $D^2y = D(Dy) = 2.b_2 + 6.b_3.\lambda$

- Então, para um dado λ, o raio de giro $r(t) = v(t)/\omega(t)$ é dado por:

  $r(\lambda) = [(Dx)^2 + (Dy)^2]^{3/2}/[D^2y.Dx - D^2x.Dy]$

Reparametrização de Trajetória
Exemplo: Perfil de Velocidade Cosenoidal



Reparametrização de Trajetória
Exemplo: Perfil de Velocidade Cosenoidal

- O valor máximo da velocidade, v_{max}, pode ser obtido em função do comprimento percorrido, l:

  $l = \int v(t).dt = \int [1 - \cos(2\pi.t/t_{max})].v_{max}/2.dt$

- Integrando no intervalo $[0, t_{max}]$, obtemos:

  $l = v_{max}.t_{max}/2$

- Assim:

  $v_{max} = 2.l/t_{max}$

Algoritmo de Reparametrização da Trajetória

i. Calcular o comprimento do caminho, integrando dl no intervalo $\lambda \in [0, 1]$:

$$l = \int [(Dx)^2 + (Dy)^2]^{1/2}.d\lambda$$

ii. Dado um tempo máximo para execução da trajetória, t_{max} , calcular a velocidade máxima v_{max} (ou vice-versa):

$$v_{max} = 2.l/t_{max} \quad \text{ou} \quad t_{max} = 2.l/v_{max}$$

Algoritmo de Reparametrização da Trajetória

iii. Incrementar o tempo t de um intervalo dt, calcular a velocidade desejada no instante t, de acordo com o perfil especificado:

$$v(t) = [1 – \cos(2\pi.t/t_{max})].v_{max}/2$$

iv. Calcular o parâmetro λ correspondente ao instante t, integrando $d\lambda$ de zero a t:

$$d\lambda = v(t)/[(Dx)^2 + (Dy)^2]^{1/2}.dt$$

$$\lambda(t) = \int d\lambda = \int v(t)/[(Dx)^2 + (Dy)^2]^{1/2}.dt$$

Algoritmo de Reparametrização da Trajetória

v. Conhecendo o λ no instante t, calcular as referências correspondentes:

$$x(\lambda) = a_0 + a_1.\lambda + a_2.\lambda^2 + a_3.\lambda^3$$

$$y(\lambda) = b_0 + b_1.\lambda + b_2.\lambda^2 + b_3.\lambda^3$$

$$Dx = dx(\lambda)/d\lambda = a_1 + 2.a_2.\lambda + 3.a_3.\lambda^2$$

$$Dy = dy(\lambda)/d\lambda = b_1 + 2.b_2.\lambda + 3.b_3.\lambda^2$$

$$\theta = \tan^{-1}(Dy/Dx)$$

vi. Enquanto $t < t_{max}$ Voltar ao passo iii

Sistemas Robóticos Autônomos

Geração de Trajetórias