<u>TÉCNICAS DE CONTROLE DE ROBÔS MÓVEIS</u>

- **<u>O Problema de Controle de Robôs Móveis</u>:**

  - Robôs Móveis são sistemas MIMO, Não lineares e, às vezes, subatuados (Ex: Robô com acionamento diferencial, quadricóptero, etc.).
  - Alguns robôs móveis possuem restrições não holonômicas adicionais (Ex: robôs com acionamento diferencial, VANT´s de asa fixa, etc.)
  - Alguns robôs móveis são naturalmente instáveis (Ex: helicópteros autônomos.

  $\Rightarrow$ O problema de controle de robôs móveis não é trivial.

- **<u>Tipos de Controladores</u>:**

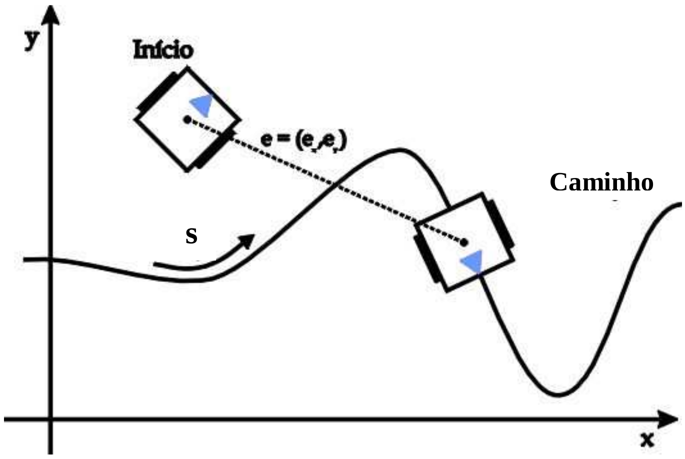
  - De acordo com o modelo utilizado:
    - Controladores Cinemáticos.
    - Controladores Dinâmicos.

    - <u>Controladores Cinemáticos</u>: o comportamento dinâmico do robô é desprezado. Baseados apenas no modelo cinemático do robô (que descreve a "dinâmica" do sistema). Pressupõem que a velocidade do robô pode ser imposta instantaneamente pelos atuadores. Aplicados a robôs com rodas que se movimentam em baixas velocidades e acelerações.

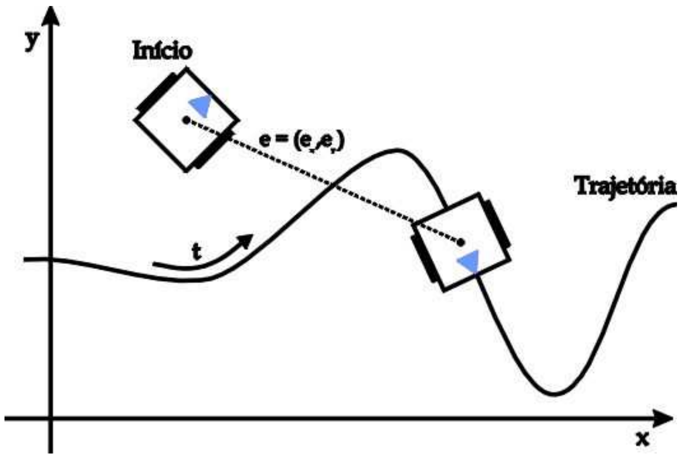
    - <u>Controladores Dinâmicos</u>: levam em conta o modelo dinâmico completo do robô.

  - De acordo com o objetivo de controle:
    - Seguidores de Caminho.
    - Seguidores de Trajetória.
    - Controladores Estabilizantes

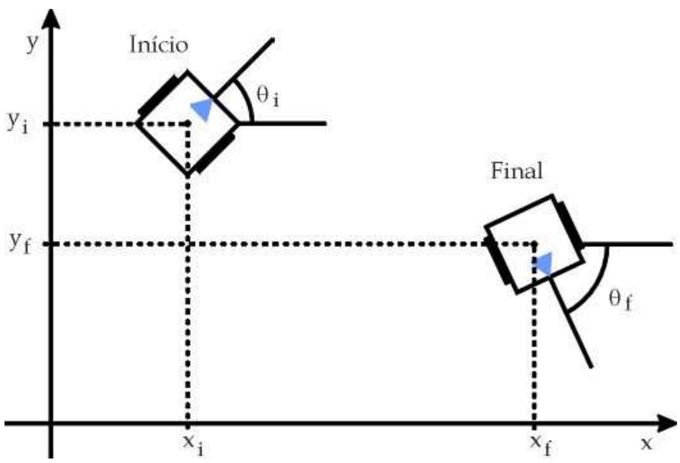
- Seguidores de Caminho: o objetivo de controle é seguir de forma monotônica um caminho (seguir uma curva geométrica no espaço), de um ponto inicial a um ponto final, sem levar em conta restrições temporais. Apenas restrições sobre a faixa de velocidades permissíveis são especificadas.
- Seguidores de Trajetória: o objetivo de controle é definido por uma trajetória contínua $\Rightarrow$ o objetivo é definido como a posição do robô, variando continuamente, em função do tempo, de uma posição inicial a uma posição final (suas derivadas de baixa ordem (velocidade e aceleração desejada podem ser necessárias). Para robôs não holonômicos, geralmente são mais simples que os controladores estabilizantes, já que a orientação do robô é naturalmente converge para a tangente da trajetória.
- Controladores Estabilizantes: o objetivo de controle é uma posição final, ou uma pose final (posição + orientação), a serem atingidas com erro zero quando o tempo tende a infinito. Para robôs não holonômicos, estes controladores são instáveis no objetivo final. De acordo com o *Teorema de Brockett*, (R. W. Brockett, R. S. Millman, and H. J. Sussmann. Asymptotic stability and feedback__stabilization.__Diferential Geometric Control Theory, Birkhäuser, Boston, USA, pp. 181–191, 1983), uma condição necessária para a estabilização suave de sistemas regulares sem drift (sistemas que permanecem em um ponto de equilíbrio quando as entradas de controle são zeradas), é que o número de entradas seja igual ao número de estados do sistema, o que geralmente não acontece em robôs não holonômicos com rodas. Assim, para obter um controlador estabilizante estável é necessário adotar leis de controle não contínuas ou variantes no tempo.



Seguidor de Caminho.



Seguidor de Trajetória.



Controlador Estabilizante.

- **Controladores Cinemáticos:**

  - Baseados apenas no modelo cinemático. A dinâmica a ser controlada é simplesmente o sistema não linear:

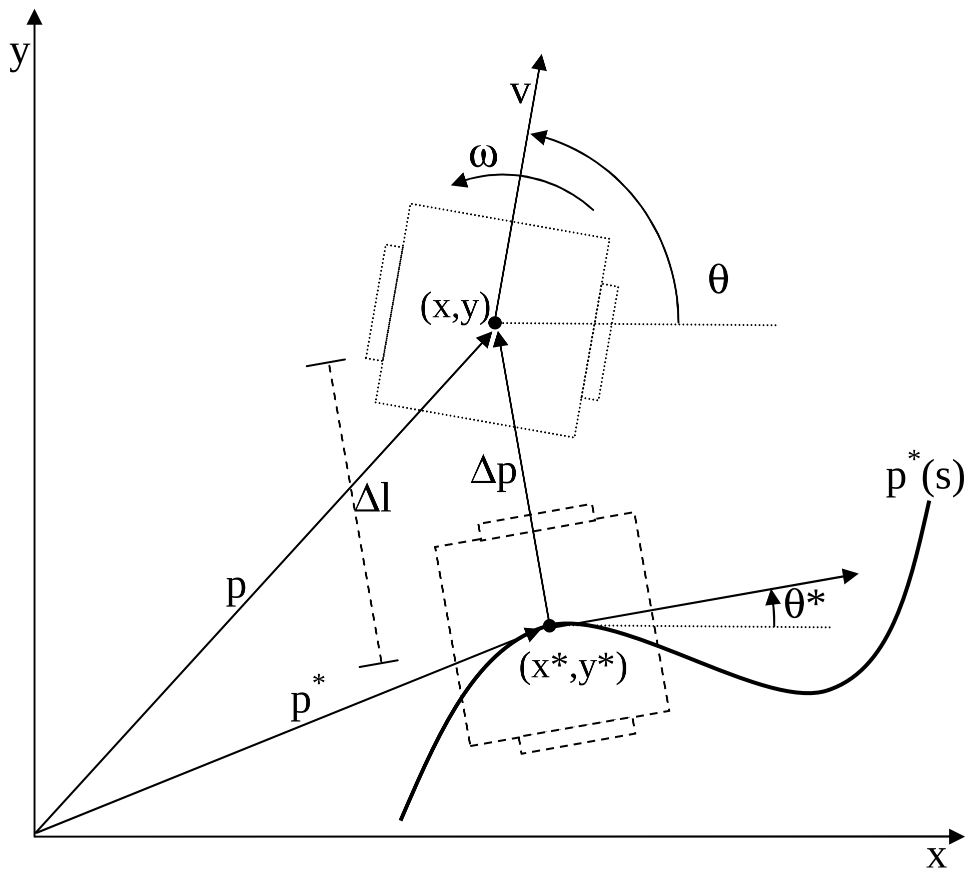
    $\mathbf{q'} = {}^{\mathbf{q}}\mathbf{T}_{\mathbf{V}}.\mathbf{V}$ ou, $\mathbf{q'} = {}^{\mathbf{q}}\mathbf{T}_{\mathbf{V}}.{}^{\mathbf{V}}\mathbf{T}_{\mathbf{W}}.\mathbf{W}$

  - O modelo dinâmico (incluindo as equações do movimento de Newton e Euler, bem como o modelo dinâmico dos atuadores) é desprezado.
  - Pressupõem que as velocidades das rodas ω_D ω_E (ou, equivalentemente, v e ω) podem ser impostas instantaneamente.
  - Válido para aplicações em que os efeitos dinâmicos podem ser desprezados. Exemplo: baixas velocidades e acelerações, baixa inércia.
  - Três categorias: Seguidores de Caminho, Seguidores de Trajetória e Controladores Estabilizantes.

- **<u>Controlador Seguidor de Caminho</u>:**

Descrição do Robô em relação ao Caminho Desejado

No controlador apresentado a seguir, ([Samson, C.: Control of chained systems—application to path following and time-varying point-stabilization of mobile robots. IEEE Trans. Autom. Control 40(1), 64–77 (1995)], ver também [Joanna Płaskonka, Different Kinematic Path Following Controllers for a Wheeled Mobile Robot of (2,0) Type. Journal of Intelligent & Robotic Systems. September, 2013]), a descrição da posição do robô em relação ao caminho desejado, é baseada em um referencial Serret-Frenet, (vetor tangente, vetor normal e vetor binormal a uma curva), movendo-se ao longo do caminho. No controlador aqui descrito, este referencial é localizado no ponto da projeção ortonormal do robô sobre o caminho.



O caminho desejado é uma curva parametrizada $p^*(s)$, onde s é o comprimento do mesmo medido a partir do seu início. Ou seja, a posição p do robô deve convergir para o alvo virtual $p^*(s)$, que se move com o referencial Serret-Frenet. Se a distância entre o robô e o alvo virtual é Δl, o erro de posição, expresso no referencial Serret-Frenet é dado por:

$$\Delta p = [0 \quad \Delta l \quad 0]^T$$

O caminho é caracterizado por uma curvatura $\kappa(s)$. Assim, a orientação desejada θ^* do robô satisfaz:

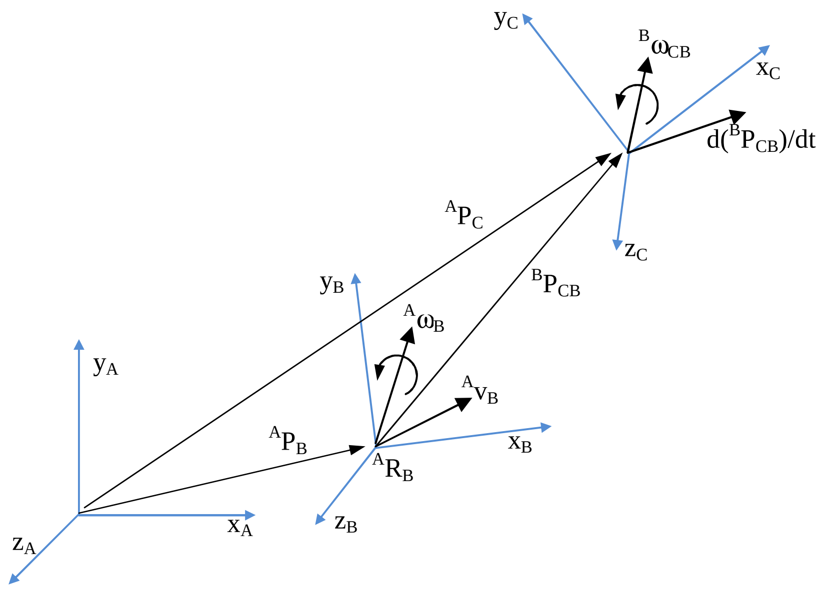
$$d(\theta^*)/dt = \pm\kappa(s).d(s)/dt$$

Na expressão acima, o sinal é positivo quando o referencial Serret-Frenet executa curvas no sentido anti-horário. O alvo virtual existe se as seguintes condições são satisfeitas:

i) A curvatura $\kappa(s)$ possui um limite superior (o caminho desejado não pode possuir curvas muito pronunciadas).
ii) O robô não pode se movimentar muito longe do caminho (uma vez que a parametrização Serret-Frenet é local).

Lembrando a expressão que descreve a relação entre as velocidades de três referenciais {A}, {B} e {C}, onde {A} é fixo, A orientação de {B} em relação a {A} é dada pela matriz de rotação ${}^{A}R_{B}$, {B} se movimenta em relação a {A} com velocidade linear ${}^{A}v_{B}$ e velocidade angular ${}^{A}\omega_{B}$ e a posição ${}^{B}P_{CB}$ de {C} em relação a {B} varia com velocidade $d({}^{B}P_{CB})/dt$:

$${}^{A}v_{C} = {}^{A}v_{B} + d({}^{A}R_{B})/dt.{}^{B}P_{CB} + {}^{A}R_{B}.d({}^{B}P_{CB})/dt$$



$$^{A}P_{C} = {}^{A}P_{B} + {}^{A}R_{B}.{}^{B}P_{CB}$$
$$^{A}v_{C} = {}^{A}v_{B} + {}^{A}\omega_{B}\times({}^{A}R_{B}.{}^{B}P_{CB}) + {}^{A}R_{B}.d({}^{B}P_{CB})/dt$$

No presente caso, {A} é o referencial fixo, {B} representa o referencial Serret-Frenet (alvo virtual) e {C} é o referencial fixo no robô. Desta forma, temos:

$$^{A}v_{C} = d(p)/dt = [d(x)/dt \;\; d(y)/dt \;\; 0]^{T}$$

$$^{A}v_{B} = d(p^{*})/dt$$

$$^{B}P_{CB} = \Delta p = [0 \;\; \Delta l \;\; 0]^{T} \Rightarrow d({}^{B}P_{CB})/dt = d(\Delta p)dt = [0 \;\; d(\Delta l)/dt \;\; 0]^{T}$$

$$^{A}R_{B} = R(z,\theta) = R = \begin{bmatrix} \cos(\theta^{*}) & -\mathrm{sen}(\theta^{*}) & 0 \\ \mathrm{sen}(\theta^{*}) & \cos(\theta^{*}) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$d({}^{A}R_{B})/dt = d(R)/dt = \begin{bmatrix} -\mathrm{sen}(\theta^{*}) & -\cos(\theta^{*}) & 0 \\ \cos(\theta^{*}) & -\mathrm{sen}(\theta^{*}) & 0 \\ 0 & 0 & 0 \end{bmatrix}.d(\theta^{*})/dt$$

Desta forma, a relação entre as velocidades dos três referenciais é expressa por:

$$d(p)/dt = d(p^*)/dt + d(R)/dt.\Delta p + R.d(\Delta p)dt$$

Que pode ser rearranjada como:

$$d(\Delta p)/dt = R^T.d(p)dt - R^T.d(p^*)/dt - R^T.d(R)/dt.\Delta p$$

O termo $R^T.d(p^*)/dt$ representa a velocidade do referencial Serret-Frenet em coordenadas do próprio referencial Serret-Frenet, que, por definição, se movimenta tangente ao caminho, com velocidade $d(s)/dt$ nesta direção, de tal forma que:

$$R^T.d(p^*)/dt = [d(s)/dt \;\; 0 \;\; 0]^T$$

Levando em conta que:

$$R^T.d(R)/dt = \begin{bmatrix} 0 & -\,d(\theta^*)/dt & 0 \\ d(\theta^*)/dt & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

A partir da relação entre as velocidades dos referenciais e lembrando que $d(\theta^*)/dt = \pm\kappa(s).d(s)/dt$, podemos obter as seguintes expressões:

$$d(\Delta l)/dt = -sen(\theta^*)dx/dt + cos(\theta^*)dy/dt$$

$$d(s)/dt = (cos(\theta^*)dx/dt + sen(\theta^*)dy/dt)/(1\mp\kappa(s)\Delta l)$$

Para evitar a singularidade $\Delta l = \pm 1/\kappa(s)$ deve-se assegurar que, durante o controle, $|\kappa(s)\Delta l| < 1$, ou seja, como a parametrização do referencial Serret-Frenet é local, a distância Δl do robô ao alvo virtual deve permanecer sempre menor do que o raio de giro corrente do caminho.

Definindo o erro de orientação $\Delta\theta = \theta - \theta^*$, sua derivada é dada por:

$d(\Delta\theta)/dt = d\theta/dt - d\theta^*/dt = \omega \mp \kappa(s).d(s)/dt$

Como para o robô com acionamento diferencial as velocidades no referencial fixo são dadas por:

$dx/dt = v.\cos(\theta)$ $\qquad\qquad$ $dy/dt = v.\mathrm{sen}(\theta)$

usando as expressões trigonométricas do cosseno da diferença de dois ângulos e do seno da diferença de dois ângulos, chegamos ao modelo que descreve a cinemática do sistema em relação ao referencial Serret-Frenet:

$d(\Delta l)/dt = v.\mathrm{sen}(\Delta\theta)$

$d(s)/dt = v.\cos(\Delta\theta)/(1\mp\kappa(s)\Delta l)$

$d(\Delta\theta)/dt = \omega \mp \kappa(s).v.\cos(\Delta\theta)/(1\mp\kappa(s)\Delta l)$

De acordo com o modelo acima, o problema de seguimento de caminho para o robô com acionamento diferencial, pode ser estabelecido da seguinte forma:

- Projetar uma lei de controle tal que o robô com acionamento diferencial, cuja dinâmica é completamente conhecida, siga um caminho desejado e os erros de rastreio do caminho convirjam para zero.
- O caminho desejado deve ser admissível, ou seja, o mesmo pode ser executado sem derrapagem das rodas do robô. Assume-se que o caminho desejado é uma curva suave e pelo menos as suas duas primeiras derivadas em relação ao parâmetro s são suaves e limitadas.

Sem perda de generalidade, considerando que o caminho seja uma curva fechada, nesta abordagem assume-se que a direção do movimento do robô dá-se no sentido anti-horário, de tal forma que:

$$d(\Delta\theta)/dt = \kappa(s).d(s)/dt$$

de modo que o modelo que descreve a cinemática do sistema em relação ao referencial Serret-Frenet é simplificado como:

$$d(\Delta l)/dt = v.sen(\Delta\theta)$$

$$d(s)/dt = v.cos(\Delta\theta)/(1-\kappa(s)\Delta l)$$

$$d(\Delta\theta)/dt = \omega - \kappa(s).v.cos(\Delta\theta)/(1-\kappa(s)\Delta l)$$

A lei de controle de Samson expressa os erros de seguimento de caminho da seguinte forma:

$$d(\Delta l)/dt = v.sen(\Delta\theta)$$

$$d(\Delta\theta)/dt = u$$

onde $u = \omega - \kappa(s).v.cos(\Delta\theta)/(1-\kappa(s)\Delta l)$ é uma nova entrada de controle para o erro angular. A lei de controle cinemático de Samson para seguimento de caminho é dada por:

$$v = \text{constante}$$

$$u = -(k_\theta.\Delta\theta + k_l.\Delta l.v.sen(\Delta\theta)/\Delta\theta)$$

onde k_θ e k_l são ganhos constantes positivos. Pode se provar que o sistema realimentado com esta lei de controle é assintoticamente estável, usando a função de Lyapunov abaixo e o Lema de Barbalat.

$$V(\Delta l,\Delta\theta) = (k_l.\Delta l^2 + \Delta\theta^2)/2$$

- **<u>Controle de Trajetória</u>:**

- Problema mais simples do que o problema de estabilização.
- A referência a ser seguida é apenas a posição $(x^*(t)\ ,\ y^*(t))$.
- A orientação desejada é definida implicitamente pela trajetória seguida e pelas restrições não holonômicas.

Trajetória de Referência

Dada uma trajetória de referência $(x^*(t)\ ,\ y^*(t))$, a orientação desejada $\theta^*(t)$ para um robô móvel com acionamento diferencial é restrita à direção tangente à curva desejada.

$$\tan(\theta^*) = y^{*\prime}/x^{*\prime}$$

$$\Rightarrow\ d(\tan(\theta^*))/dt = \theta^{*\prime}/\cos^2(\theta^*)$$

$$\Rightarrow\ \theta^{*\prime}/\cos^2(\theta^*) = [x^{*\prime}.y^{*\prime\prime}-x^{*\prime\prime}.y^{*\prime}]/(x^{*\prime})^2$$

$$\Rightarrow\ \theta^{*\prime} = [x^{*\prime}.y^{*\prime\prime}-x^{*\prime\prime}.y^{*\prime}]/(x^{*\prime}/\cos(\theta^*))^2$$

$$\Rightarrow\ \theta^{*\prime} = [x^{*\prime}.y^{*\prime\prime}-x^{*\prime\prime}.y^{*\prime}]/\ [(x^{*\prime})^2+(y^{*\prime})^2]$$

- As referências em espaço de configuração podem ser mapeadas para referências equivalentes em termos de velocidade linear e angular:

$$v^* = \pm[(x^{*\prime})^2+(y^{*\prime})^2]^{1/2} \qquad \omega^*=[x^{*\prime}.y^{*\prime\prime}-x^{*\prime\prime}.y^{*\prime}]/[(x^{*\prime})^2+(y^{*\prime})^2]$$

Controle por Linearização da Realimentação Dinâmica

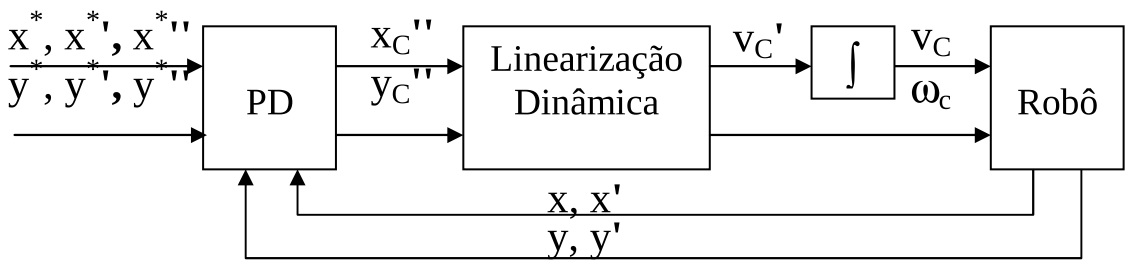
[B. d'Andrea Novel, G. Bastin, and G. Campion. Control of nonholonomic wheeled móbile robots by state feedback linearization. Int. Journal of Robotics Research, vol. 14, no. 6, pp. 543–559, 1995].

- O Controle por Linearização da Realimentação Dinâmica (DFL - *Dynamic Feedback Linearization*) envolve Realimentação PD e Compensação do Modelo Não Linear.
- Um estado adicional é acrescentado ao sistema, derivando o modelo cinemático $\mathbf{q'} = {}^{\mathbf{q}}\mathbf{T}_{\mathbf{V}}.\mathbf{V}$:

$$x' = v.\cos\theta \qquad \Rightarrow x'' = \cos\theta.v' - \text{sen}\theta.v.\omega$$
$$y' = v.\text{sen}\theta \qquad \Rightarrow y'' = \text{sen}\theta.v' + \cos\theta.v.\omega$$

$$\Rightarrow \begin{bmatrix} x'' \\ y'' \end{bmatrix} = \begin{bmatrix} \cos\theta & -v.\text{sen}\theta \\ \text{sen}\theta & v.\cos\theta \end{bmatrix} . \begin{bmatrix} v' \\ \omega \end{bmatrix}$$

- Trajetória desejada: $(x^*(t), y^*(t))$ e suas 1ª e 2ª derivadas.
- Requer medição da configuração $[x \;\; y \;\; \theta]^T$ e das velocidades x', y', bem como a determinação de v.
- Lei de Controle:



- Realimentação Proporcional Derivativa:

$$x_c'' = x^{*}'' + K_{dx}(x^{*}' - x') + K_{px}(x^{*} - x)$$
$$y_c'' = y^{*}'' + K_{dy}(y^{*}' - y') + K_{py}(y^{*} - y)$$

onde x_c'' e y_c'' são as acelerações de comando em espaço de configuração, K_{px} e K_{py} são ganhos proporcionais e K_{dx} e K_{dy} são ganhos derivativos positivos.

- Compensação do Modelo Não Linear:

$$\Rightarrow \begin{bmatrix} v_c' \\ \omega_c \end{bmatrix} = \begin{bmatrix} \cos\theta & \mathrm{sen}\theta \\ -\mathrm{sen}\theta/v & \cos\theta/v \end{bmatrix} . \begin{bmatrix} x_c'' \\ y_c'' \end{bmatrix}$$

- Integração - velocidade de comando: $v_c = \int v_c' \, dt$

- Resposta em malha fechada:

$$\begin{bmatrix} x'' \\ y'' \end{bmatrix} = \begin{bmatrix} \cos\theta & -v.\mathrm{sen}\theta \\ \mathrm{sen}\theta & v.\cos\theta \end{bmatrix} . \begin{bmatrix} v_c' \\ \omega_c \end{bmatrix}$$

$$\Rightarrow \begin{bmatrix} x'' \\ y'' \end{bmatrix} = \begin{bmatrix} \cos\theta & -v.\mathrm{sen}\theta \\ \mathrm{sen}\theta & v.\cos\theta \end{bmatrix} \begin{bmatrix} \cos\theta & \mathrm{sen}\theta \\ -\mathrm{sen}\theta/v & \cos\theta/v \end{bmatrix} . \begin{bmatrix} x_c'' \\ y_c'' \end{bmatrix}$$

$$\Rightarrow \begin{bmatrix} x'' \\ y'' \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} . \begin{bmatrix} x_c'' \\ y_c'' \end{bmatrix} = \begin{bmatrix} x_c'' \\ y_c'' \end{bmatrix}$$

$$\Rightarrow \quad x_c'' - x'' = 0$$
$$y_c'' - y'' = 0$$

Assim, a dinâmica do erro $[\Delta x \ \Delta y]^T = [(x^{*}-x) \quad (y^{*}-y)]^T$ é:

$$s^2.\Delta x(s) + K_{dx}.s.\Delta x(s) + K_{px}.\Delta x(s) = 0$$
$$s^2.\Delta y(s) + K_{dy}.s.\Delta y(s) + K_{py}.\Delta y(s) = 0$$

- Observações:

  - A trajetória desejada deve ser duplamente derivável.
  - O estado inicial do integrador deve ser $v_c \neq 0$.
  - A trajetória desejada deve ser persistente, de modo a evitar singularidades.
  - Uma possível modificação para contornar a divisão por zero quando v tende a zero é substituir v por $(v + \delta)$, onde δ é um coeficiente positivo e muito menor do que a velocidade nominal.

- **<u>Controle Estabilizante</u>:**

- Problema mais complexo do que o problema de controle de trajetória.
- A referência a ser alcançada é uma configuração (x^*, y^*, θ^*).
- O sistema é subatuado (apenas duas entradas, v e ω ou ω_D e ω_E, para controlar três saídas: x', y' e θ').

<u>Controle por Coordenadas Polares (*Polar Coordinate Control*)</u>

- Sistema descrito pelo modelo não linear: $\mathbf{q}' = {}^{\mathbf{q}}\mathbf{T}_{\mathbf{V}}.\mathbf{V}$
- Baseado na transformação para coordenadas polares da configuração do robô, a qual é singular na origem:

  $R = [x^2 + y^2]^{1/2}$
  $\gamma = \tan^{-1}(y/x)$
  $\delta = \gamma + \theta$

- Configuração final desejada: $q^* = [x^*,\ y^* \quad \theta^*]^T$. Sem perda de generalidade, assumiremos $q^* = [0 \quad 0 \quad 0]^T$.
- Medição apenas da configuração $[x \quad y \quad \theta]^T$.
- Regulador estável no sentido de Lyapunov, definido pela lei de controle:

  $v = K_R.R.\cos\gamma$
  $\omega = K_\gamma.\gamma + K_R.(\gamma + K_\delta.\delta).(\text{sen}\gamma.\ \cos\gamma)/\gamma$

  onde K_R, K_γ e K_δ são ganhos constantes e positivos.

- Observação: a lei de controle deve ser desligada ou modificada na vizinhança da origem, de modo a evitar a singularidade introduzida por $\tan^{-1}(y/x)$.

- **<u>Controladores Dinâmicos</u>:**

  - Baseados no modelo completo, incluindo as relações cinemáticas, as equações de movimento de Newton e Euler e a dinâmica de atuadores.
  - As entradas de controle são fornecidas diretamente pelos atuadores (tensões de armadura dos motores direito e esquerdo e_D e_E, ou entradas equivalentes).
  - As velocidades das rodas ω_D ω_E (ou, equivalentemente, as velocidades do robô v e ω) não são impostas instantaneamente, mas apresentam resposta de acordo com a dinâmica do sistema.
  - Válido para aplicações em que os efeitos dinâmicos não podem ser desprezados. Exemplo: altas velocidades e acelerações, alta inércia.
  - Duas categorias: Controle de Trajetória e Estabilização.

  - **<u>Controle de Trajetória</u>:**

  - Problema mais simples do que o problema de estabilização.
  - A referência a ser seguida é apenas a posição $(x^*(t) , y^*(t))$.
  - A orientação desejada é definida implicitamente pela trajetória seguida e pelas restrições não holonômicas.

$$\tan(\theta) = y'/x' \Rightarrow d(\tan(\theta))/dt = \theta'/\cos^2(\theta) = [x'.y''-x''.y']/(x')^2$$

$$\Rightarrow \theta' = [x'.y''-x''.y']/(x'/\cos(\theta))^2 = [x'.y''-x''.y']/ [(x')^2+(y')^2]$$

  - As referências em espaço de configuração podem ser mapeadas para comandos de velocidade linear e angular:

$$v^* = \pm[(x^{*\prime})^2+(y^{*\prime})^2]^{1/2} \qquad \omega^*=[x^{*\prime}.y^{*\prime\prime}-x^{*\prime\prime}.y^{*\prime}]/[(x^{*\prime})^2+(y^{*\prime})^2]$$

Controle de Velocidade – Compensação Não Linear + PI

- Sistema descrito pelo modelo não linear:

$$\tau_V = \mathbf{M_V}.\mathbf{V'} + \mathbf{B_V}.\mathbf{V}$$

- Trajetória desejada: $(x^*(t),\ y^*(t))$ e suas 1ª e 2ª derivadas. $\theta^*(t)$ e suas derivadas são calculados de modo a respeitar as restrições não holonômicas.
- Mapeamento das velocidades e acelerações desejadas em espaço de configuração para velocidades e acelerações no referencial do robô, $\mathbf{V}^* = ({}^q\mathbf{T_V})^T.\mathbf{q}^{*}{}'$ e $\mathbf{V}^{*}{}' = ({}^q\mathbf{T_V})^T.\mathbf{q}^{*}{}''$.
- Medição da configuração $q = [x \quad y \quad \theta]^T$ e velocidade em espaço de configuração: $q'' = [x'' \quad y'' \quad \theta'']^T$.
- Mapeamento das velocidades medidas em espaço de configuração para o referencial do robô: $\mathbf{V} = ({}^q\mathbf{T_V})^T.\mathbf{q}'$
- Lei de Controle incluindo Realimentação PI e Compensação do Modelo Não Linear:

  - Realimentação Proporcional Integral:

$$\mathbf{V_c}' = \mathbf{V}^{*}{}' + \mathbf{K_p}.\Delta\mathbf{V} + \mathbf{K_i}.\int\Delta\mathbf{V}dt$$

onde $\Delta\mathbf{V} = (\mathbf{V}^* - \mathbf{V})$, $\mathbf{V_c}'$ é a aceleração de comando, $\mathbf{K_p}$ é uma matriz diagonal de ganhos proporcionais positivos e $\mathbf{K_i}$ é uma matriz diagonal de ganhos integrais positivos.

  - Compensação do Modelo Não Linear:

$$\tau_V = \mathbf{M_V}^*.\mathbf{V_c}' + \mathbf{B_V}^*.\mathbf{V}$$

onde $\mathbf{M_V}^*$ e $\mathbf{B_V}^*$ são as matrizes de parâmetros dinâmicos disponíveis, que constituem o modelo nominal. São uma estimativa dos parâmetros reais $\mathbf{M_V}$ e $\mathbf{B_V}$.

- Em malha fechada:

$$\mathbf{M_V}^*.(\mathbf{V}^{*\prime} + \mathbf{K_p}.\ \Delta\mathbf{V} + \mathbf{K_i}.\int \Delta\mathbf{V}dt) + \mathbf{B_V}^*.\mathbf{V} = \mathbf{M_V}.\mathbf{V}' + \mathbf{B_V}.\mathbf{V}$$

$$\Rightarrow \quad \mathbf{M_V}^*.(\mathbf{V}^{*\prime} + \mathbf{K_p}.\ \Delta\mathbf{V} + \mathbf{K_i}.\int \Delta\mathbf{V}dt) = \mathbf{M_V}.\mathbf{V}' + \mathbf{B_V}.\mathbf{V} - \mathbf{B_V}^*.\mathbf{V}$$

$$\Rightarrow \quad \mathbf{M_V}^*.(\mathbf{V}^{*\prime} + \mathbf{K_p}.\ \Delta\mathbf{V} + \mathbf{K_i}.\int \Delta\mathbf{V}dt) = \mathbf{M_V}.\mathbf{V}' + (\mathbf{B_V} - \mathbf{B_V}^*).\mathbf{V}$$

$$\Rightarrow \quad \mathbf{M_V}^*.(\mathbf{V}^{*\prime} + \mathbf{K_p}.\ \Delta\mathbf{V} + \mathbf{K_i}.\int \Delta\mathbf{V}dt) = \mathbf{M_V}.\mathbf{V}' + \Delta_{\mathbf{BV}}.\mathbf{V}$$

onde $\Delta_{\mathbf{BV}} = (\mathbf{B_V} - \mathbf{B_V}^*)$ é o erro de modelagem da matriz $\mathbf{B_V}^*$.

Subtraindo $\mathbf{M_V}^*.\mathbf{V}'$ de ambos lados da equação acima:

$$\Rightarrow \mathbf{M_V}^*.(\mathbf{V}^{*\prime}+\mathbf{K_p}.\Delta\mathbf{V}+\mathbf{K_i}.\int\Delta\mathbf{V}dt)-\mathbf{M_V}^*.\mathbf{V}' = \mathbf{M_V}.\mathbf{V}'- \mathbf{M_V}^*.\mathbf{V}'+\Delta_{\mathbf{BV}}.\mathbf{V}$$

$$\Rightarrow \quad \mathbf{M_V}^*.(\mathbf{V}^{*\prime}- \mathbf{V}'+\mathbf{K_p}.\Delta\mathbf{V}+\mathbf{K_i}.\int\Delta\mathbf{V}dt) = (\mathbf{M_V}- \mathbf{M_V}^*).\mathbf{V}'+\Delta_{\mathbf{BV}}.\mathbf{V}$$

$$\Rightarrow \quad \mathbf{M_V}^*.(\Delta\mathbf{V}'+\mathbf{K_p}.\Delta\mathbf{V}+\mathbf{K_i}.\int \Delta\mathbf{V}dt) = \Delta_{\mathbf{MV}}.\mathbf{V}'+\Delta_{\mathbf{BV}}.\mathbf{V}$$

onde $\Delta_{\mathbf{MV}} = (\mathbf{M_V} - \mathbf{M_V}^*)$ é o erro de modelagem da matriz $\mathbf{M_V}^*$.

$$\Rightarrow (\Delta\mathbf{V}'+\mathbf{K_p}.\Delta\mathbf{V}+\mathbf{K_i}.\int \Delta\mathbf{V}dt) = (\mathbf{M_V}^*)^{-1}.(\Delta_{\mathbf{MV}}.\mathbf{V}'+\Delta_{\mathbf{BV}}.\mathbf{V})$$

Verifica-se que, se o modelo nominal disponível descrever exatamente o modelo real do robô, (ou seja, $\Delta_{\mathbf{MV}} = (\mathbf{M_V} - \mathbf{M_V}^*) = \mathbf{0}$ e $\Delta_{\mathbf{BV}} = (\mathbf{B_V} - \mathbf{B_V}^*) = \mathbf{0}$), então:

$$\Rightarrow (\Delta\mathbf{V}'+\mathbf{K_p}.\ \Delta\mathbf{V}+\mathbf{K_i}.\int\Delta\mathbf{V}dt) = \mathbf{0}$$

Ou seja, o sistema em malha fechada seria linear e desacoplado.

- **<u>Controle Estabilizante</u>:**

<u>Controle por Referência Variável</u>

- Configuração final desejada: $\mathbf{q^*} = [x^* \;\; y^* \;\; \theta^*]^T$.
- Requer apenas medição da configuração $\mathbf{q} = [x \;\; y \;\; \theta]^T$.
- Controlador baseado na realimentação de erro nas variáveis:

$$\mathbf{L}(t) = [l(t) \quad \theta(t)]^T = \int \mathbf{V}(t).dt$$

onde $l(t) = \int v(t).dt$ é o comprimento percorrido pelo robô no intervalo de tempo de integração e $\theta(t) = \int \omega(t).dt$ é o ângulo de orientação do robô. Tomando a transformada de Laplace, temos:

$$s.\mathbf{L}(s) = \mathbf{V}(s)$$

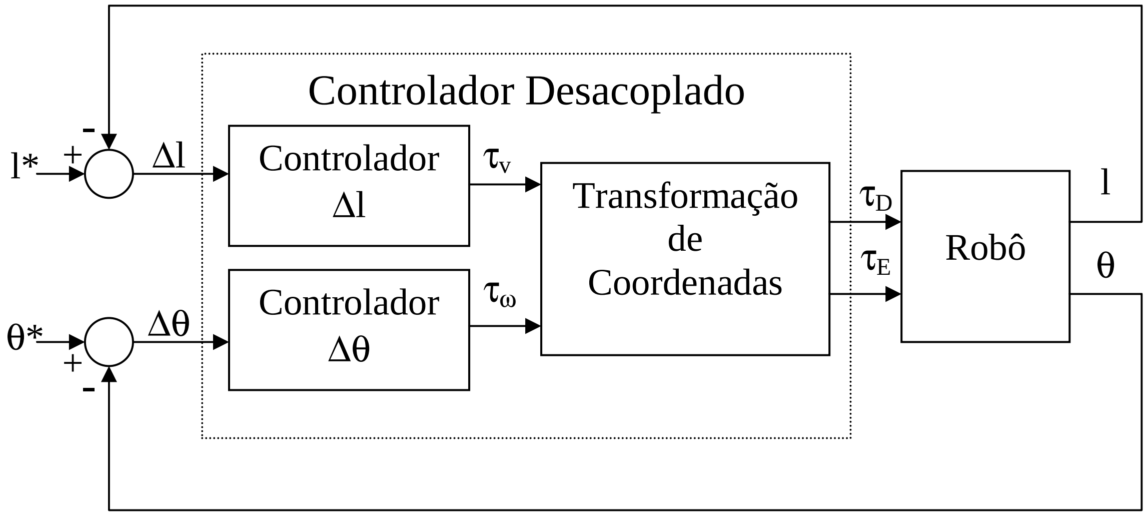
- Referências variáveis de posição e ângulo permitem desacoplar o controle de $l(t)$ e $\theta(t)$.
- Referências variáveis de posição e ângulo permitem atingir a posição e orientação final, mesmo sendo o sistema subatuado.
- Controladores lineares desacoplados em $l(t)$ e $\theta(t)$ atuam em paralelo de modo a garantir a posição e orientação final.
- Controladores lineares desacoplados em $l(t)$ e $\theta(t)$ (para deslocamento e orientação) são projetados a partir do modelo linear obtido aplicando a transformada de Laplace ao modelo dinâmico em **V**:

$$\boldsymbol{\tau}_\mathbf{V} = \mathbf{M}_\mathbf{V}.\mathbf{V}' + \mathbf{B}_\mathbf{V}.\mathbf{V} \quad \Rightarrow \quad \boldsymbol{\tau}_\mathbf{V}(s) = s.\mathbf{M}_\mathbf{V}.\mathbf{V}(s) + \mathbf{B}_\mathbf{V}.\mathbf{V}(s)$$

ou, substituindo $\mathbf{V}(s)$ por $s.\mathbf{L}(s)$:

$$\boldsymbol{\tau}_\mathbf{V}(s) = s.\mathbf{M}_\mathbf{V}.s.\mathbf{L}(s) + \mathbf{B}_\mathbf{V}.s.\mathbf{L}(s) = s.[s.\mathbf{M}_\mathbf{V} + \mathbf{B}_\mathbf{V}].\mathbf{L}(s)$$

- Controladores Desacoplados para l(t) e θ(t):



τ_v = entrada virtual, que controla l(t), (ou v(t)).
τ_ω = entrada virtual, que controla θ(t), (ou ω(t)).

- A transformação de coordenadas transforma as entradas de controle virtuais em entradas de controle em referencial de atuador (τ_D e τ_E ou, diretamente, e_D e e_E).
- Para obter esta transformação é necessário diagonalizar a matriz de transferência s.[s.$\mathbf{M_V}$ + $\mathbf{B_V}$], de modo a desacoplar o controle de l(t) e θ(t).

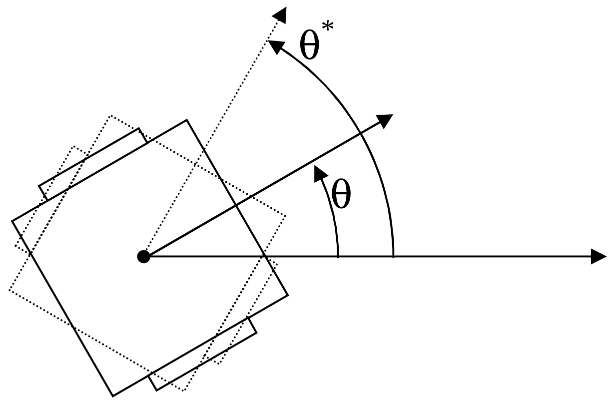
- <u>Problema</u>: como formular o erro Δl(t), uma vez que o mesmo não é mensurável.

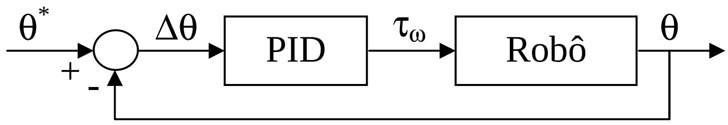
- <u>Solução</u>: utilizar referência variável, de modo a reduzir o problema às situações triviais.

a) Quando na posição desejada, girar (sem sair da posição atual) para atingir a orientação desejada.
b) Quando na orientação desejada, andar em frente para atingir a posição desejada na direção da orientação atual.

- Caso trivial a). Erro de orientação a ser realimentado para o controlador de orientação quando o robô já está na posição final desejada:

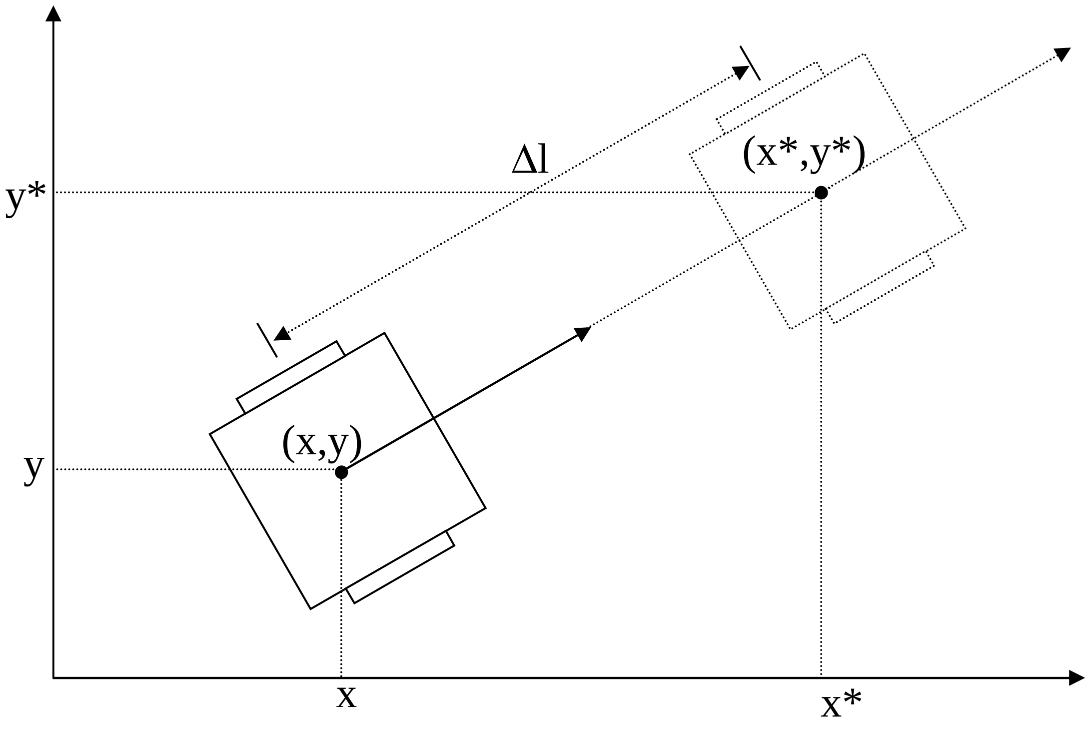
$\Delta\theta = \theta^{*} - \theta$

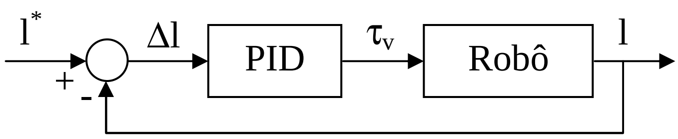




- Caso trivial b). Erro de posição a ser realimentado para o controlador de deslocamento quando o robô já está orientado em direção à posição final desejada:

$\Delta l = [(\Delta x)^2 + (\Delta y)^2]^{1/2}$ onde: $\Delta x = (x^{*} - x)$ $\Delta y = (y^{*} - y)$





- Referência Variável para controle de Posição:

  - Deseja-se atingir a posição final (x^*,y^*), não importando com que orientação se chegue à mesma (θ^* é escolhida variável, de modo a apontar sempre para o alvo).
  - Define-se a referência variável de posição (x_r^*,y_r^*), localizada na projeção da posição final (x^*,y^*) sobre a reta sobre a qual o robô está orientado.
  - Os erros de orientação $\Delta\theta$ e de deslocamento Δl a serem fornecidos a os controladores são obtidos a partir de:

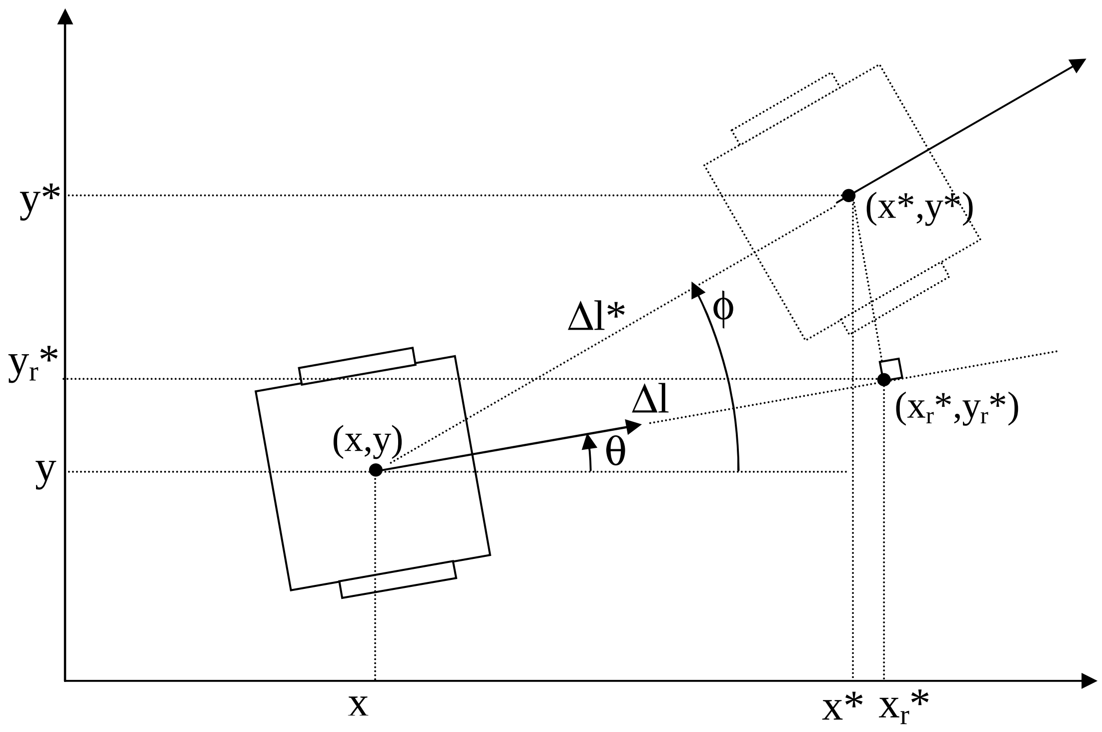
$\Delta x = x^* - x$ $\qquad\qquad$ $\Delta y = y^* - y$

$$\theta^* = \phi = \tan^{-1}(\Delta y/\Delta x)$$

$$\Delta l^* = [(\Delta x)^2 + (\Delta y)^2]^{1/2}$$

$$\Delta\theta = \theta^* - \theta$$

$$\Delta l = \Delta l^*.\cos(\Delta\theta)$$



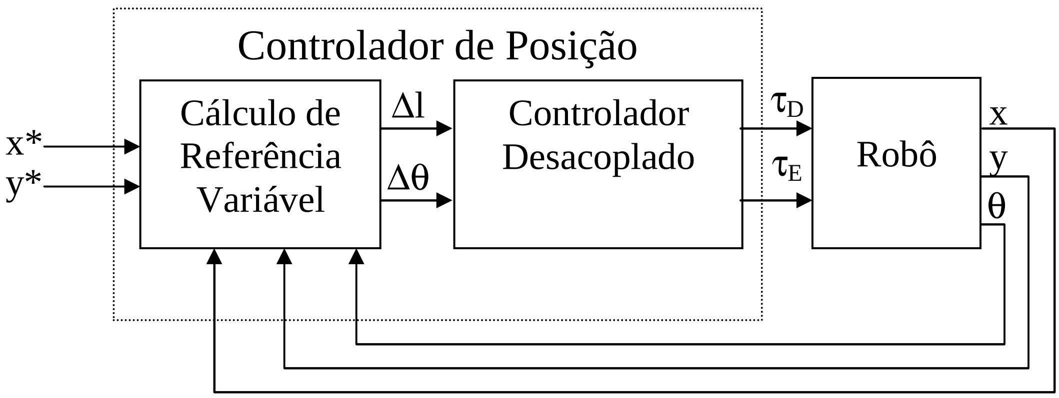
- O erro de realimentação Δl fornecido ao controlador de deslocamento é calculado em relação à posição de referência variável (x_r^*,y_r^*) mais próxima da posição desejada (x^*,y^*) e localizada sobre a reta que o robô percorreria com a orientação atual θ. A posição (x_r^*,y_r^*) varia de acordo com a mudança de orientação do robô.

- Como (x_r^*,y_r^*) está na frente do robô, neste (e somente neste caso), pode-se saber exatamente qual é o erro de deslocamento Δl: a distância euclidiana entre (x,y) e (x_r^*,y_r^*).

- O controlador de deslocamento guia o robô ao ponto (x_r^*,y_r^*), que é o mais próximo ao alvo na direção da orientação atual do robô.

- O controlador de orientação orienta o robô em direção ao alvo. É responsável por diminuir o erro de realimentação $\Delta\theta$ de modo a alinhar o robô com a reta que aponta para a posição final desejada (x^*, y^*). A orientação desejada θ^* é variável e sempre alinhada com esta reta.

- Se o controlador de orientação for mais rápido do que o controlador de deslocamento e for capaz de garantir que $\Delta\theta \to 0$ e se o controlador de deslocamento garantir que $\Delta l \to 0$, então, $\cos(\Delta\theta) \to +1$ e, conseqüentemente, devido a que $\Delta l^* = \Delta l/\cos(\Delta\theta)$, temos que $\Delta l^* \to 0$, ou seja, o robô atinge a posição final.



- Referência Móvel para controle de Posição e Orientação:

  - Deseja-se atingir a posição final (x^*,y^*), com uma orientação θ^* especificada.

  - Solução: usar o controlador de posição anterior usando como referência uma posição móvel (x_r^*,y_r^*).

  - Esta referência móvel deve fazer com que, inicialmente, o robô seja atraído para a reta de aproximação, (reta que passa pelo alvo (x^*,y^*) com a orientação θ^* final desejada.

  - À medida em que o robô chegar perto da reta de aproximação, a referência (x_r^*,y_r^*) deve ser progressivamente movida em direção ao alvo (x^*,y^*).

  - Para alcançar este objetivo, o erro angular a ser minimizado pelo controlador é definido como:

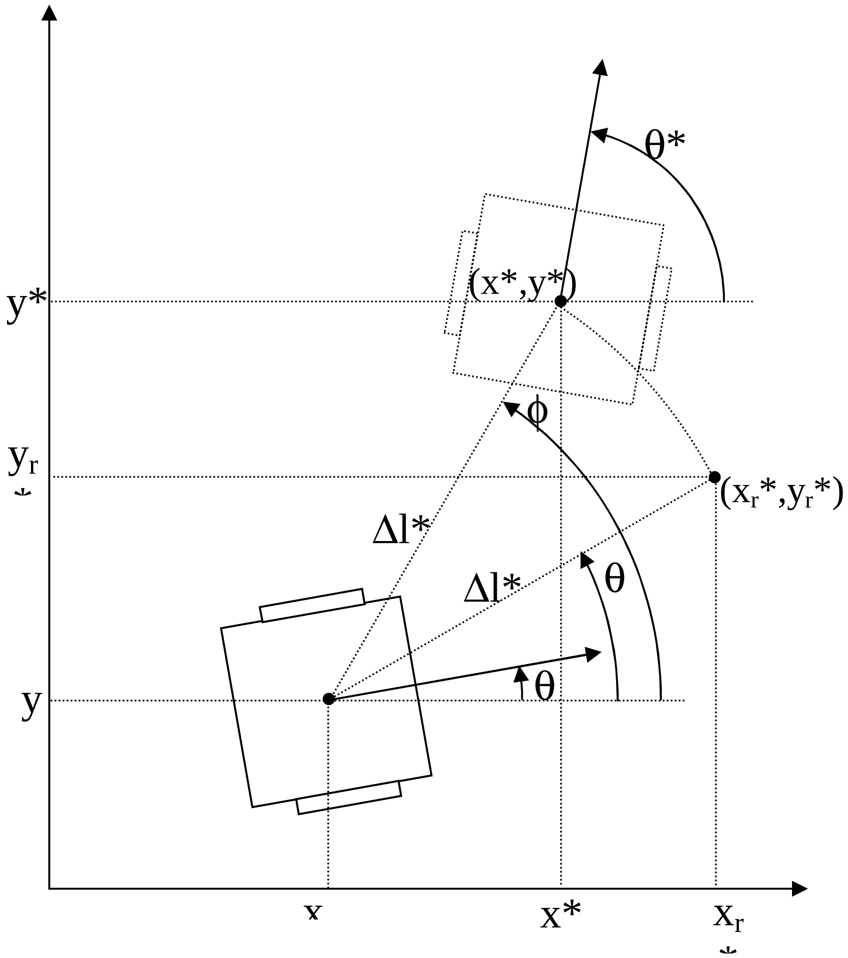
    $$\Delta\theta = \theta_r - \theta = (\phi - \theta^*) + (\phi - \theta)$$

    onde $\phi = \tan^{-1}(\Delta y/\Delta x) = \tan^{-1}((y^* - y)/(x^* - x))$. O ângulo de referência móvel é dado por:

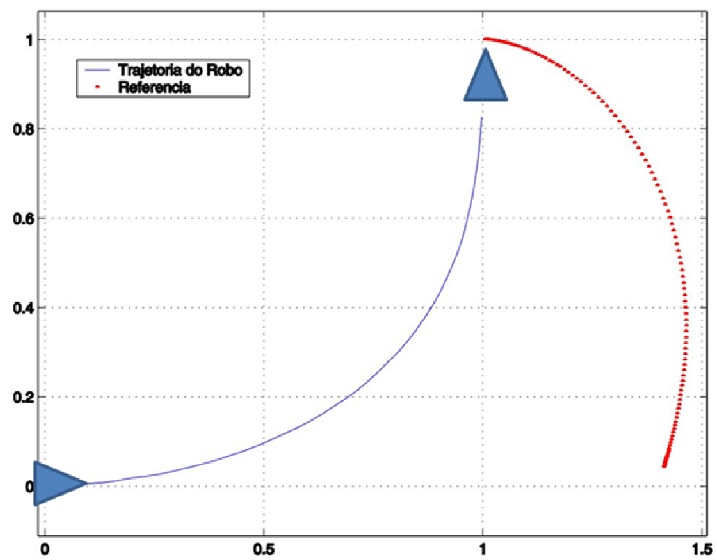
    $$\theta_r = 2.\phi - \theta^*$$

  - O erro $\Delta\theta$ possui duas parcelas:

    a) Erro angular $(\phi - \theta^*)$ entre a reta que liga o robô ao alvo e a reta de aproximação. Quando o robô alcança a reta de aproximação, este erro é nulo e o problema recai no caso anterior de controle de posição.
    b) Erro $(\phi - \theta)$, que guia o robô para apontar para o alvo.



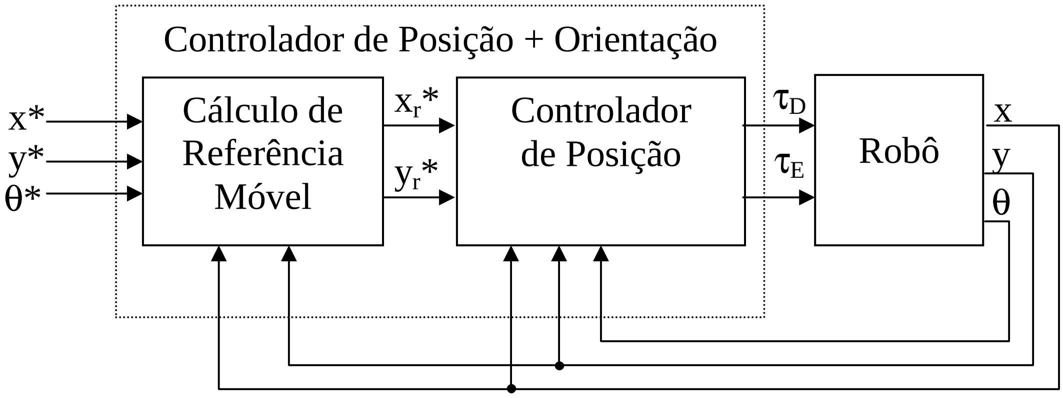
- A referência móvel a ser fornecida ao controlador de posição pode ser vista como uma rotação do alvo em torno da posição atual do robô. O valor desta rotação é de $(\theta_r - \phi) = (\phi - \theta^*)$ radianos.



- Assim, a referência móvel é dada por:

$x_r^* = x + \Delta l^*.\cos(\theta_r)$ $\qquad$ $y_r^* = y + \Delta l^*.\text{sen}(\theta_r)$

onde $\Delta l^* = [(\Delta x)^2 + (\Delta y)^2]^{1/2}$



<u>Observações</u>:

- O projeto dos controladores deve ser tal que o controlador de orientação seja mais rápido que o de deslocamento, uma vez que erros de orientação rapidamente amplificam erros de deslocamento.
- Quando a entrada é saturada, (tensão $||\mathbf{E}|| \leq ||\mathbf{E}||_{Max}$, ou torque $||\tau|| \leq ||\tau||_{Max}$), pelo mesmo motivo anterior, deve-se priorizar a entrada responsável pelo controle de orientação, τ_ω, em relação à entrada responsável pelo controle de deslocamento, τ_v.
- Assim, deve-se primeiro verificar se o valor de τ_ω requerido pode ser fornecido pela fonte de energia.

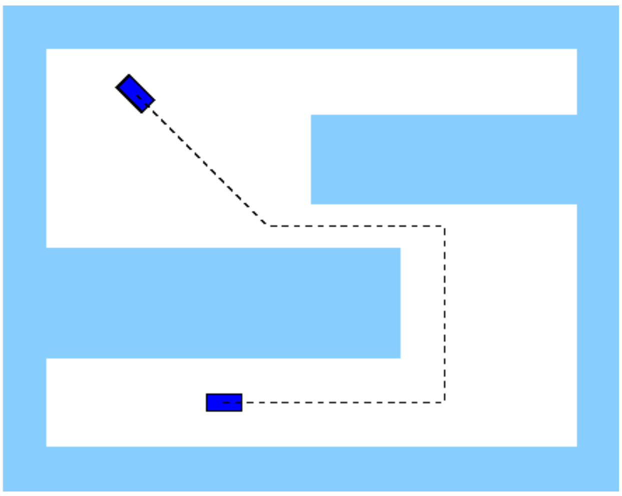
- Em caso negativo (não há energia suficiente), satura-se τ_ω no valor máximo disponível e impõe-se τ_v nulo.
- Em caso afirmativo (há energia suficiente para τ_ω), τ_ω assume o valor calculado e verifica-se se o que resta de capacidade disponível da fonte é suficiente para alimentar τ_v.
  - Em caso negativo (não há energia suficiente para τ_v), satura-se τ_v no valor do resto de entrada disponível.
  - Em caso afirmativo (há energia suficiente também para τ_v), τ_v assume o valor calculado.

- **<u>Geração de Trajetória</u>:**

  - Controladores de trajetória requerem que uma trajetória de referência seja especificada.
  - A trajetória pode ser caracterizada como uma curva geométrica contínua entre a configuração inicial e final, (denominada caminho), à qual se associam restrições temporais.
  - Em muitos casos, a trajetória de referência deve ser derivável ou duplamente derivável.
  - Para robôs não holonômicos, a trajetória gerada deve respeitar as restrições não holonômicas.
  - Outras restrições podem ter que ser levadas em conta ao especificar uma trajetória, por exemplo: velocidades e/ou acelerações máximas admissíveis ao robô, comprimento mínimo, etc.

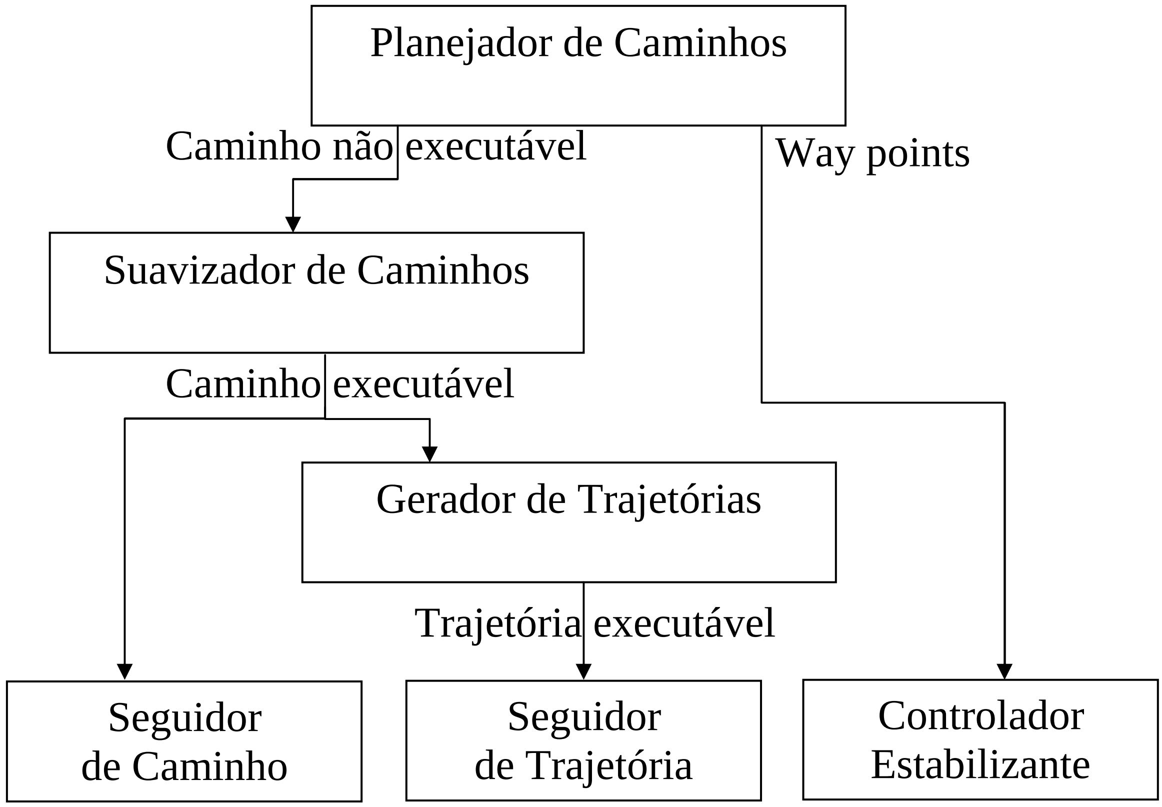
<u>Problema</u>:

- Planejadores de caminhos geralmente não produzem caminhos que satisfaçam restrições do robô e que possam ser executados pelo mesmo.



<u>Possíveis soluções</u>:

- Planejar levando em conta as restrições do robô (cinemáticas, mecânicas, etc.). Aumenta a complexidade do planejamento.
- Adaptar o caminho, suavizando-o para atender às restrições. Neste caso, existe a possibilidade de não encontrar uma trajetória executável. Quando obtida, esta pode não ser ótima.

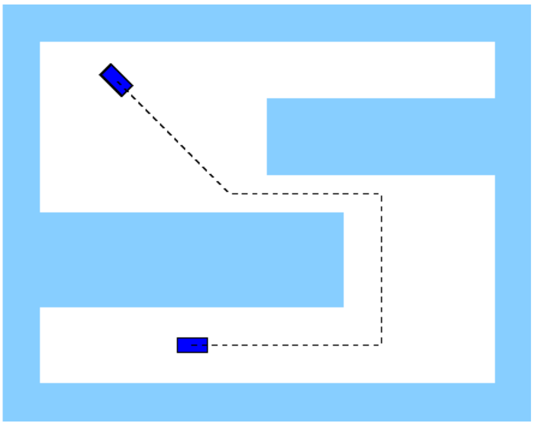


Adaptação do Caminho Geométrico:

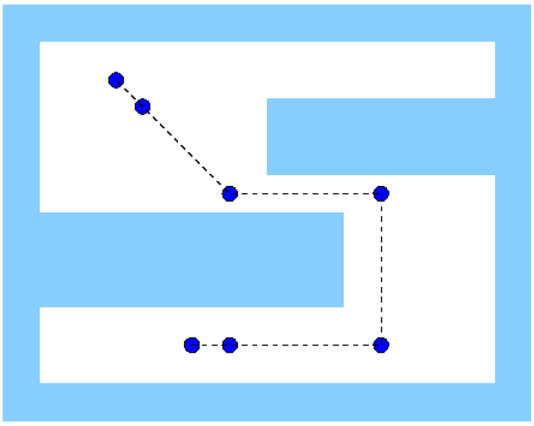
O procedimento de adaptação de um caminho geométrico para atender a restrições do robô pode ser feito de acordo com as seguintes etapas:

i. Subdivisão em segmentos delimitados por arestas
ii. Cálculo de caminhos executáveis entre arestas (ponto a ponto)
iii. Verificação de colisões
iv. Se necessário, subdivisão em maior número de segmentos e reinício da etapa i.

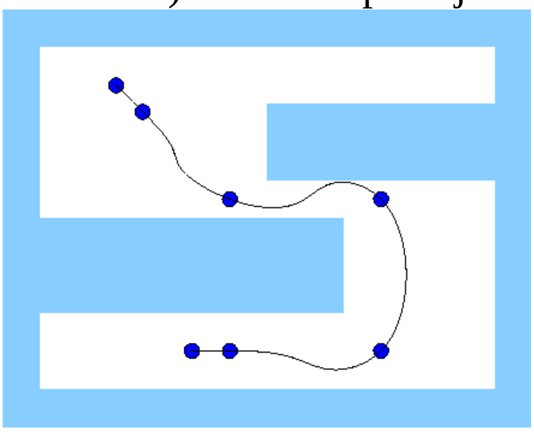
Exemplo de adaptação:



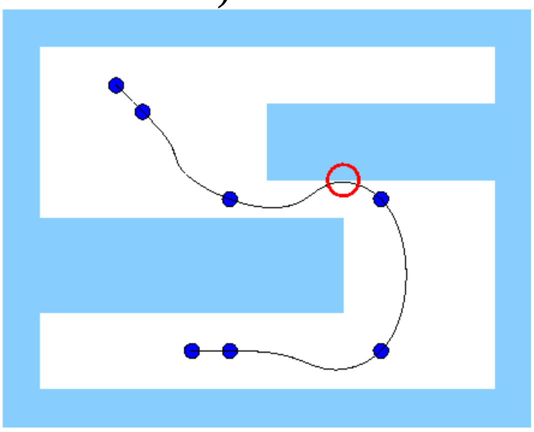
a) Caminho planejado.



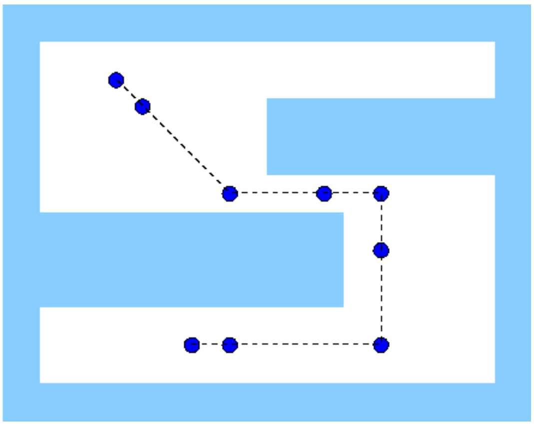
b)Subdivisão.



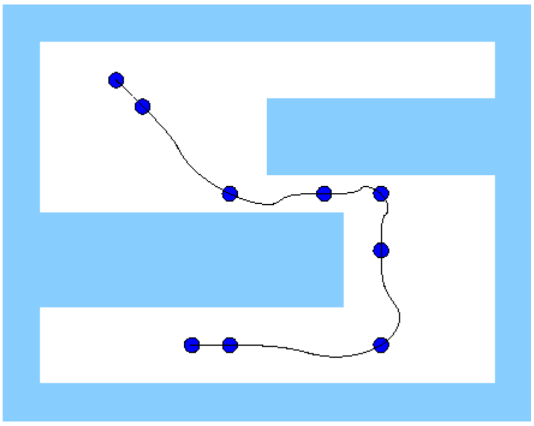
c) Caminhos ponto a ponto.



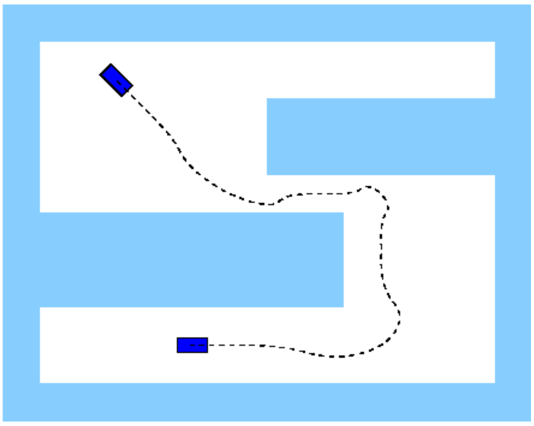
d) Teste de colisão.



e) Nova subdivisão



f) Caminhos ponto a ponto.



g) Não há colisão – Caminho executável.

<u>Métodos de Subdivisão</u>:

- Bisseção
  - Maior tempo médio
  - Garantia de convergência em tempo finito
- Aleatório
  - Menor tempo médio
  - Risco de não convergência
- Heurístico
  - Dependente do tipo de trajetória e dos obstáculos

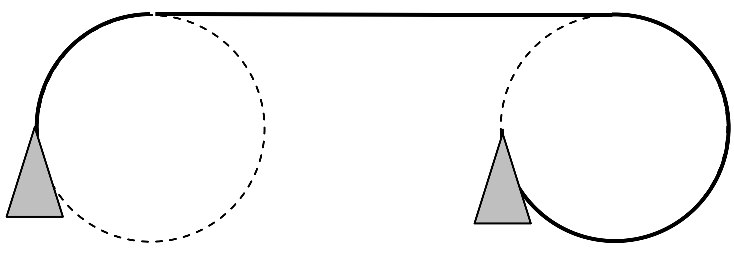
Geração de Caminhos Ponto a Ponto:

- Principais restrições cinemáticas:
  - Restrições não-holonômicas
  - Raio mínimo de giro
- Métodos de determinação:
  - Caminhos Dubins
  - Caminhos de Reeds & Shepp
  - Clotóides
  - Polinômios cúbicos

Caminhos de Dubins

[L. Dubins, On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents, American Journal of Mathematics (79) (1957) 497–516]

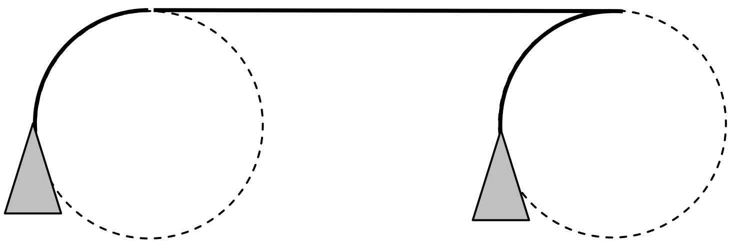
- Trabalho pioneiro no planejamento de caminhos ponto a ponto para veículos com restrições no movimento.
- Caminhos consistem em concatenações de arcos de circunferência de raio mínimo e segmentos de reta.
- Aplicado a veículos com restrições no raio de curvatura, como, por exemplo, um carro.
- Considera-se que o veículo se movimenta sempre para frente, como no caso de submarinos, aviões e veículos que não podem se deslocar para trás ou para os quais isto não seja vantajoso.
- Um total de seis caminhos possíveis podem ser encontrados entre duas configurações distintas. Geralmente, escolhe-se o de menor comprimento.



Caminhos de Reeds & Shepp

[J. Reeds, L. Shepp, Optimal paths for a car that goes both forward and backwards, Pacific Journal of Mathematics 2 (145) (1990) 367–393]

- Da mesma forma que os caminhos de Dubins, os caminhos de Reeds & Shepp consistem em concatenações de arcos de circunferência de raio mínimo e segmentos de reta.
- Aplicado a veículos com restrições no raio de curvatura, como, por exemplo, um carro.
- Movimentos de ré são permitidos.
- Um total de quarenta e oito caminhos possíveis podem ser encontrados entre duas configurações distintas. Geralmente, escolhe-se o de menor comprimento.

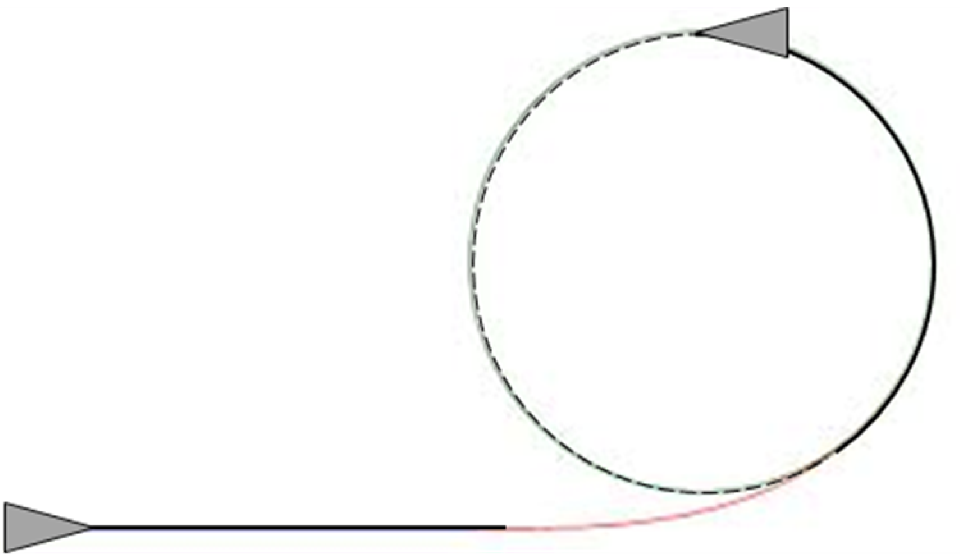


- Vantagens:
  - Caminho de comprimento mínimo;
  - Incorpora facilmente restrições quanto ao raio de giração;
  - Trajetória composta por encadeamento de segmentos padronizados, o que facilita o cálculo de velocidade e o teste de colisões.
- Desvantagens:
  - Exige o teste de 48 combinações;
  - As descontinuidades obrigam o robô a ter velocidade nula em ao menos uma das rodas ao mudar de tipo de movimento;
  - Se o raio mínimo de rotação é nulo, sempre degenera em trajetórias do tipo “gire – ande – gire”.

Clotóides

Ver, por exemplo: [T. Fraichard, A. Scheuer, From Reeds and Shepp’s to continuous-curvature paths, IEEE Transactions on Robotics 20 (6) (2004) 1025–1035]

- Caminhos de Dubins ou de Reeds & Shepp apresentam o problema da descontinuidade na velocidade angular na concatenação de segmento de reta e arco de circunferência.
- Clotóide é uma curvatura cujo raio de curvatura varia linearmente com o seu comprimento.
- Possibilitam concatenar segmentos de reta e arcos de circunferência através de uma transição suave, sem descontinuidade na curvatura ou na velocidade angular.



Interpolação por Polinômios Cúbicos

Dado o parâmetro λ, com $0 \leq \lambda \leq 1$, a trajetória do robô pode ser especificada através de polinômios cúbicos nas componentes x e y da posição do robô, expressos em função do parâmetro λ:

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

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

onde a_0, a_1, a_2, a_3, b_0, b_1, b_2 e b_3 são parâmetros a determinar. Considera-se que $\lambda = 0$ quando o robô está na sua posição inicial $(x(0), y(0)) = (x_i, y_i)$, assim como $\lambda = 1$ quando o robô está na sua posição final $(x(1), y(1)) = (x_f, y_f)$.

O ângulo de orientação θ do robô deve satisfazer as restrições não holonômicas ao longo de toda a trajetória:

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

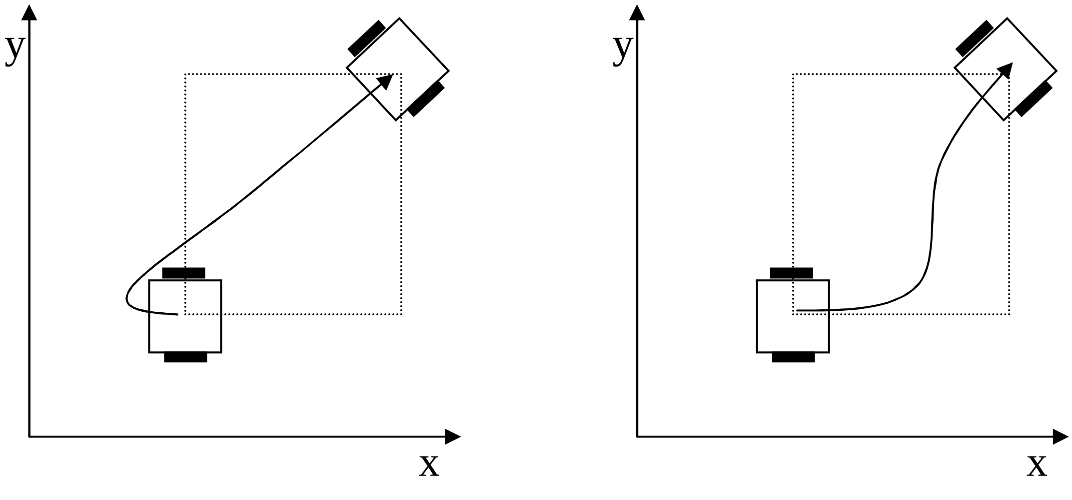
Definindo $\alpha(\lambda) = (b_1 + 2.b_2.\lambda + 3.b_3.\lambda^2)/(a_1 + 2.a_2.\lambda + 3.a_3.\lambda^2)$

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

Aplicando as seis condições de contorno de modo a impor que a trajetória inicie na configuração $q_i = (x_i,y_i,\theta_i)$ e termine na configuração $q_f = (x_i,y_i,\theta_i)$, obtemos um sistema de seis equações e oito incógnitas (os coeficientes a_0, a_1, a_2, a_3, b_0, b_1, b_2, b_3).

Como o número de coeficientes é maior do que o número de equações, dois coeficientes livres podem ser escolhidos de modo a obter soluções mais aprimoradas e que intuitivamente sejam "mais inteligentes".

O critério utilizado para aprimorar a solução é fazer com que, na medida do possível, o robô não execute movimentos afastando-se do alvo (para trás). Matematicamente, isto é alcançado através de soluções para as quais o polinômio interpolador não apresente máximos nas direções x e/ou y.



a) Sem aprimoramento. b) Caminho aprimorado.

Dependendo das condições de contorno, para evitar divisões por zero na solução, torna-se necessário adotar diferentes parâmetros livres. Seja $\Delta x = x_f - x_i$ e $\Delta y = y_f - y_i$, definindo δ como um pequeno intervalo angular, (para efeitos práticos, $\delta \cong 1^o$), temos o seguinte procedimento para geração da trajetória:

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)
$a_0 = x_i$
$a_1 = 0$
$a_2 = 3.\Delta x$
$a_3 = -2.\Delta x$
$b_0 = y_i$
$b_3 = \Delta y - b_1 - b_2$

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)
$a_0 = x_i$
$a_1 = 0$
$a_2 = \Delta x - a_3$
$b_0 = y_i$
$b_1 = 2(\Delta y - \alpha_f.\Delta x) - \alpha_f.a_3 + b_3$
$b_3 = (2.\alpha_f.\Delta x - \Delta y) + \alpha_f.a_3 - 2.b_3$

iii. **Senão, se** $\theta_f \in [(\pi/2 - \delta), (\pi/2 + \delta)]$
$a_1 = 3.\Delta x/2$ (coeficiente livre)
$b_2 =$ qualquer valor (coeficiente livre)
$a_0 = x_i$
$a_2 = 3.\Delta x - 2.a_1$
$a_3 = a_1 - 2.\Delta x$
$b_0 = y_i$
$b_1 = \alpha_i.a_1$
$b_3 = \Delta y - \alpha_i.a_1 - b_2$

iv. **Senão**

$a_1 = \Delta x$ (coeficiente livre)

$a_2 = 0$ (coeficiente livre)

$a_0 = x_i$

$a_3 = \Delta x - a_1 - a_2$

$b_0 = y_i$

$b_1 = \alpha_i.a_1$

$b_2 = 3(\Delta y - \alpha_f.\Delta x) + 2.(\alpha_f - \alpha_i).a_1 + \alpha_f.a_2$

$b_3 = 3.\alpha_f.\Delta x - 2.\Delta y - (2.\alpha_f - \alpha_i).a_1 - \alpha_f.a_2$

v. **Fazer λ variar de** 0 **a** 1

vi. **Calcular**

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

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

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

Observação:

- 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 da Trajetória

Como parâmetro λ é adimensional, $(0 \leq \lambda \leq 1)$, torna-se necessário reparametrizar a trajetória, de modo a associar o parâmetro λ com o tempo, de acordo com algum perfil de velocidade especificado para a curva geométrica $(x(\lambda),y(\lambda))$. Seja:

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

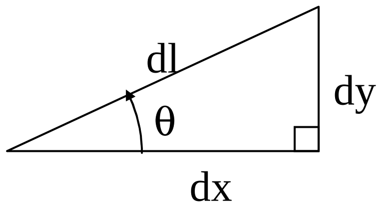
$$y(\lambda) = b_0 + b_1.\lambda + b_2.\lambda^2 + b_3.\lambda^3$$

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

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

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

Assim, para um deslocamento infinitesimal, dl, do robô, temos:



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

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

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

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

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

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). O Comprimento percorrido, l, é dado por:

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

Onde 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.

A velocidade angular do robô para um dado λ, pode ser obtida a partir da sua velocidade linear, através da derivada:

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

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

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

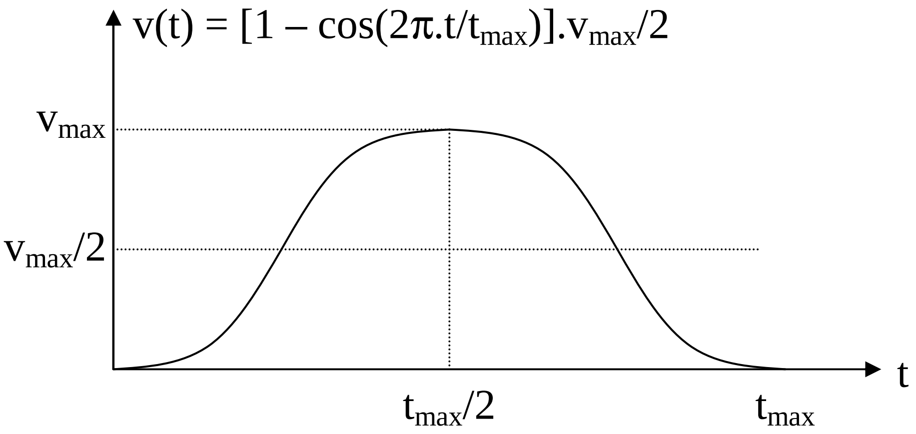
$$\omega(t) = v(t).[D^2y.Dx - D^2x.Dy]/[(Dx)^2 + (Dy)^2]^{3/2}$$

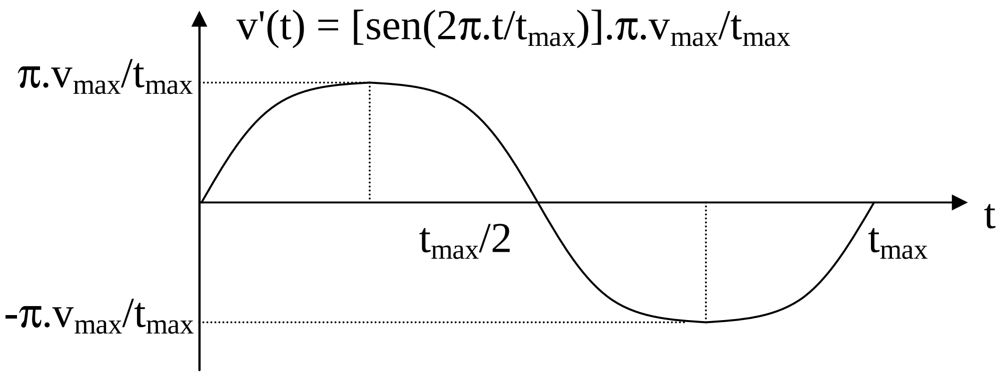
onde: $D^2x = D(Dx) = 2.a_2 + 6.a_3.\lambda$
$D^2y = D(Dy) = 2.b_2 + 6.b_3.\lambda$

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

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

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:

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

Assim, o algoritmo para reparametrizar a trajetória consiste em:

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, calcular a velocidade máxima (ou vice-versa):

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

iii. Incrementando o tempo t em intervalos 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:

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

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

v. Conhecendo o valor do parâmetro 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.