# Robótica Industrial

## Engenharia de Controle e Automação – 9º Período

PROF. LUCAS VAGO SANTANA
lucas@ifes.edu.br

# Aula 06 – Cinemática Inversa e de Velocidade em 2D

- Recordando: O que é cinemática inversa?
- Cinemática Inversa em 2D: Método Analítico da Trigonometria
- Cinemática inversa em 2D: Método Numérico
  - Cinemática de Velocidade em 2D;
  - O Jacobiano do Manipulador;
  - Resolved Rate of Motion;
  - Solução numérica pelo método de Inversão do Jacobiano.

# Referências Bibliográficas


- CORKE, Peter. **Robotics, Vision and Control: Fundamentals Algorithms in MATLAB**. 2. ed. Springer-Verlag Berlin Heidelberg, 2017.
- CORKE, Peter. **QUT Robot Academy: The open online robotics education resource**. Disponível em: <https://robotacademy.net.au/>. Acesso em 27 fev. 2020.
- NIKU, Saeed B. **Introduction to Robotics: Analysis, Control, Applications**. 2. ed. Springer-Verlag Berlin Heidelberg, 2011.
- SPONG, Mark W.; HUTCHINSON, Seth; VIDYASAGAR, M. **Robots Modeling and Control**. 1. ed. John Wiley & Sons, 2005.
- LYNCH, Kevin M.; PARK, Frank C. **Modern Robotics: Mechanics, Planning and Control**. 1. ed. Cambridge University Press, 2017.
- SICILIANO, Bruno; KHATIB, Oussama. **Springer Handbook of Robotics**. 2. ed. Springer-Verlag Berlin Heidelberg, 2016.
- SCIAVICCO, Lorenzo; SICILIANO, Bruno. **Modelling and Control of Robot Manipulators**. 2. ed. Springer-Verlag Berlin Heidelberg, 2000.

# Recordando:
# O que é a Cinemática Inversa?

# Cinemática Inversa

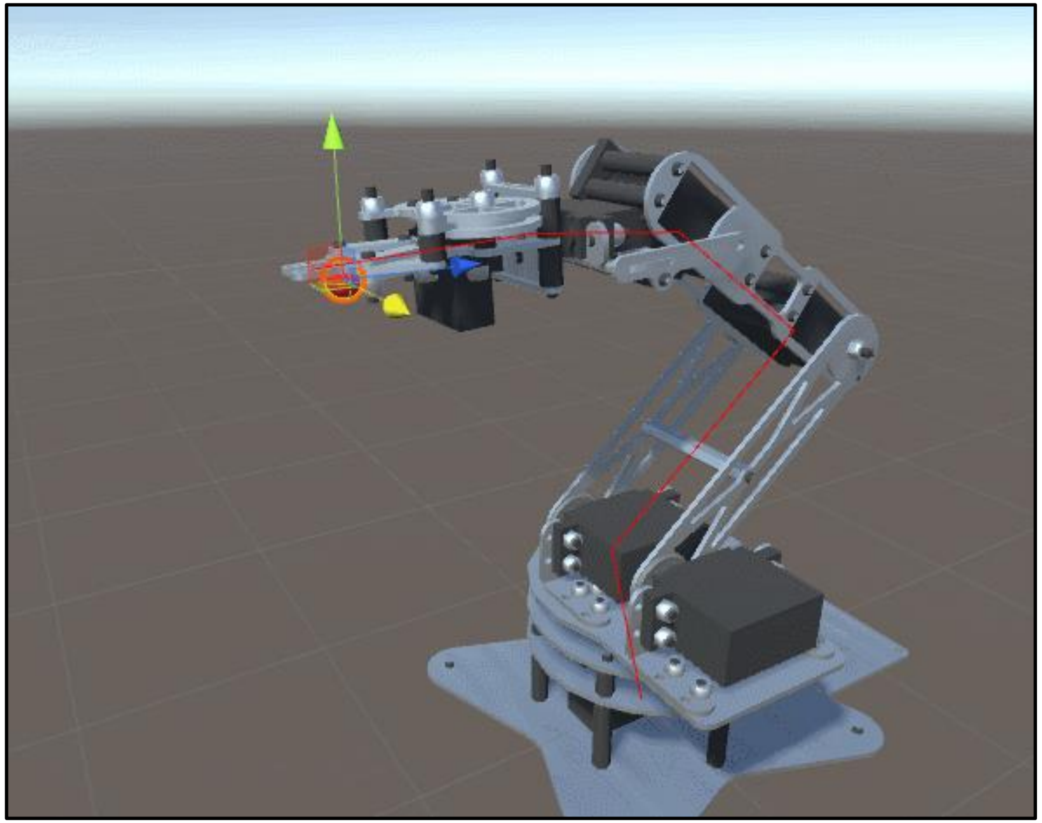
- Dada a posição do efetuador final, quais condições das juntas levam o robô até lá?

Cinemática Inversa é um problema muito complexo.

Ainda hoje há publicações de trabalhos inovando sobre este conceito.

Exemplo:

LIAO, Z. et al. A novel solution of inverse kinematic for 6R robot manipulator with offset joint based on screw theory. *International Journal of Advanced Robotic Systems*, Thousand Oaks, California, v. 17, n. 3, 2020. [LINK]



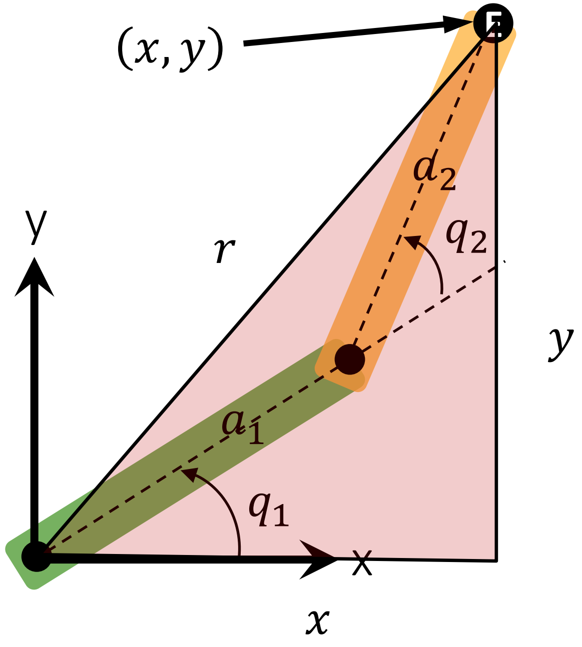
https://www.alanzucconi.com/2017/04/06/forward-kinematics/

# Cinemática Inversa via Método Analítico da Trigonometria

# Cinemática Inversa do Robô RR

- Dado o manipulador planar RR;
- Aplicar o teorema de Pitágoras:

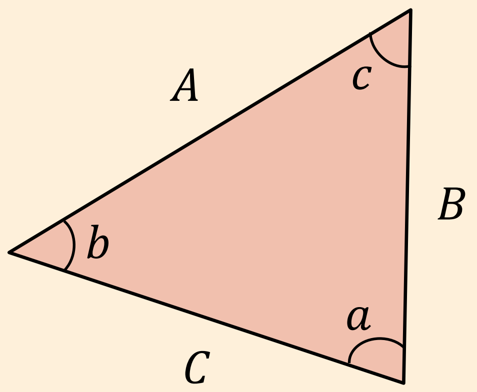
$$r^2 = x^2 + y^2$$



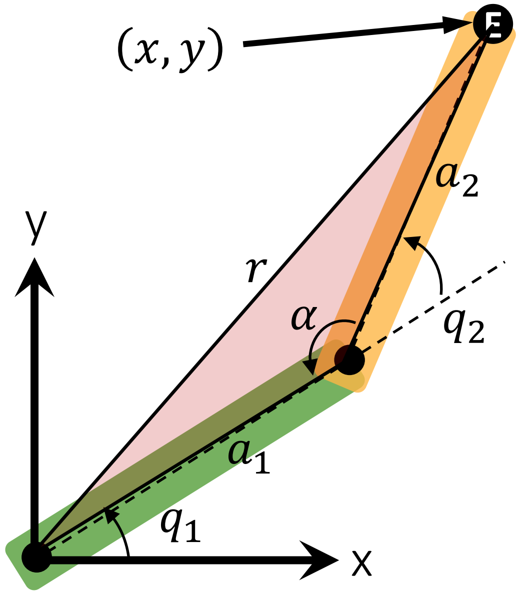
# Cinemática Inversa do Robô RR

- Aplicar a lei dos cossenos em função de $\alpha$ e dos elos do robô:

**Lei dos Cossenos:**




$$A^2 = B^2 + C^2 - 2BC\cos a$$




$$r^2 = a_1^2 + a_2^2 - 2a_1a_2\cos\alpha$$

# Cinemática Inversa do Robô RR

- Dada a expressão anterior, isolar $\cos\alpha$:

$$r^2 = a_1^2 + a_2^2 - 2a_1a_2\cos\alpha$$

$$\cos\alpha = \frac{a_1^2+a_2^2-r^2}{2a_1a_2}$$

- Ao substituir $r^2$, encontra-se:

$$r^2 = x^2 + y^2 \Rightarrow \cos\alpha = \frac{a_1^2+a_2^2-x^2-y^2}{2a_1a_2}$$

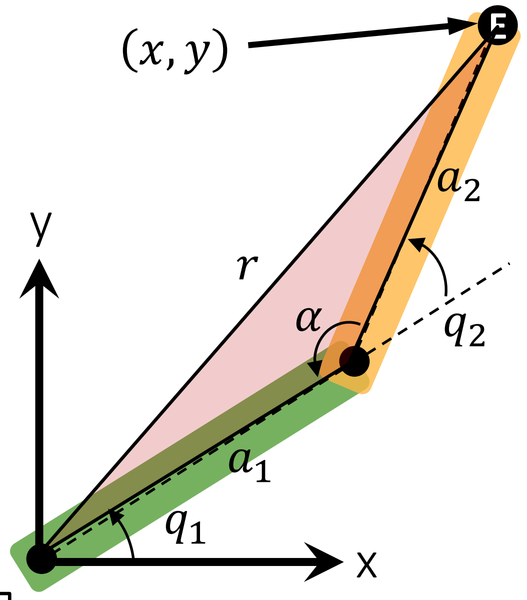
- Observe no desenho que:

$$q_2 = \pi - \alpha$$

- Aplicando cosseno em ambos os lados:

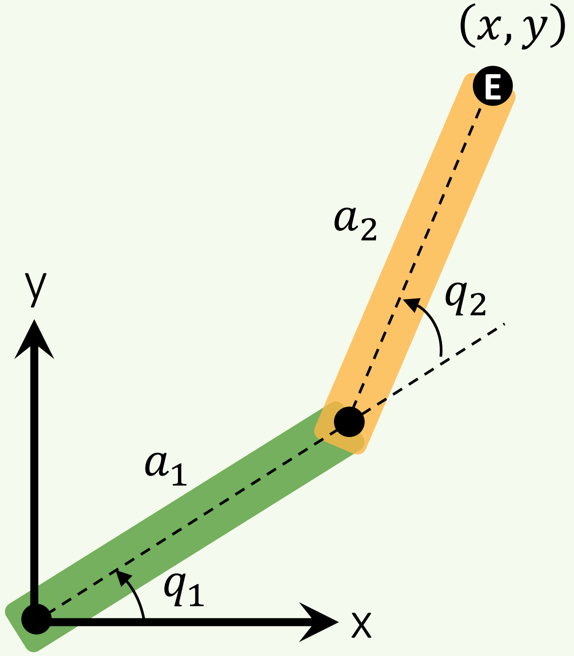
$$\cos q_2 = \cos(\pi - \alpha)$$

$$\cos q_2 = -\cos\alpha \xrightarrow{\textit{arco cosseno}} \boxed{q_2 = \operatorname{acos}\left(\frac{x^2+y^2-a_1^2-a_2^2}{2a_1a_2}\right)}$$



# Cinemática Inversa do Robô RR

- Observar que existem duas soluções neste problema:

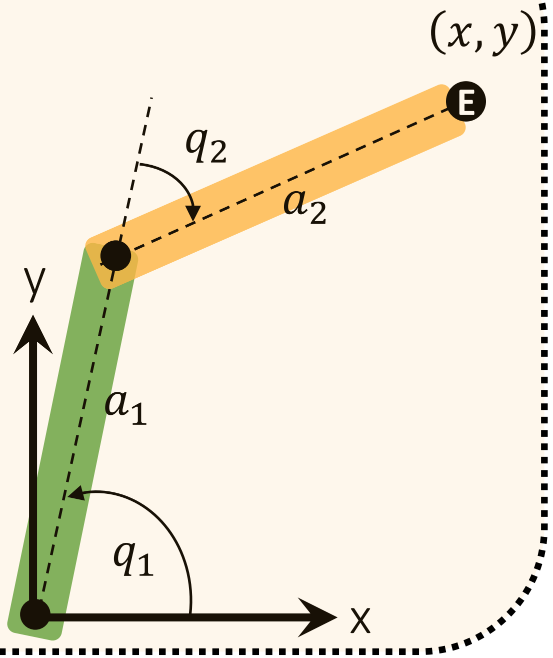



Se $q_2 \geq 0°$

$$q_2 = \operatorname{acos}\left(\frac{x^2+y^2-a_1^2-a_2^2}{2a_1a_2}\right)$$

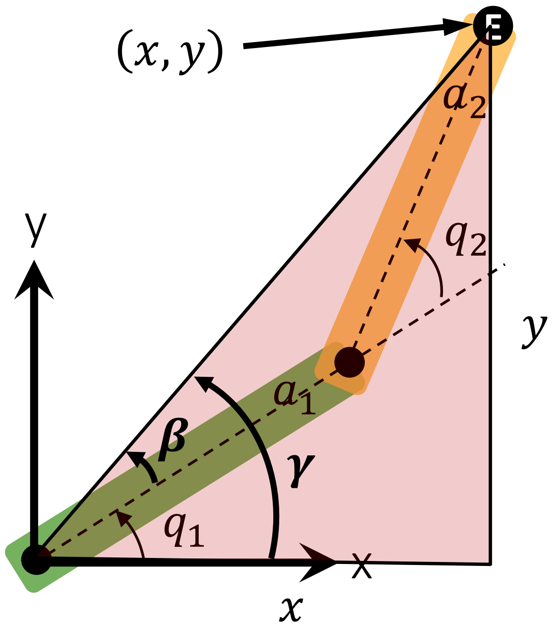
Se $q_2 < 0°$

$$q_2 = -\operatorname{acos}\left(\frac{x^2+y^2-a_1^2-a_2^2}{2a_1a_2}\right)$$



# Cinemática Inversa do Robô RR

- Reconsidere o triângulo observando os ângulos $q_1$, $\gamma$ e $\beta$:

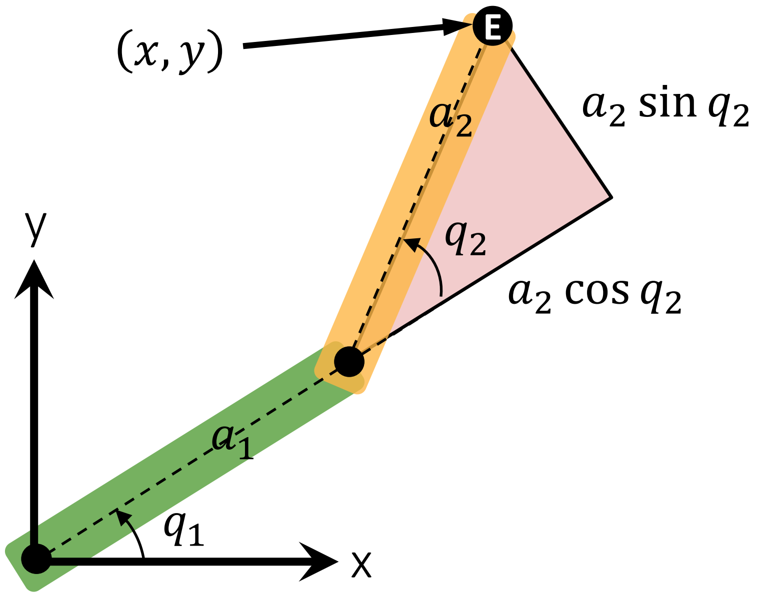



$$q_1 = \gamma - \beta$$

$$\gamma = \tan^{-1}\left(\frac{y}{x}\right)$$

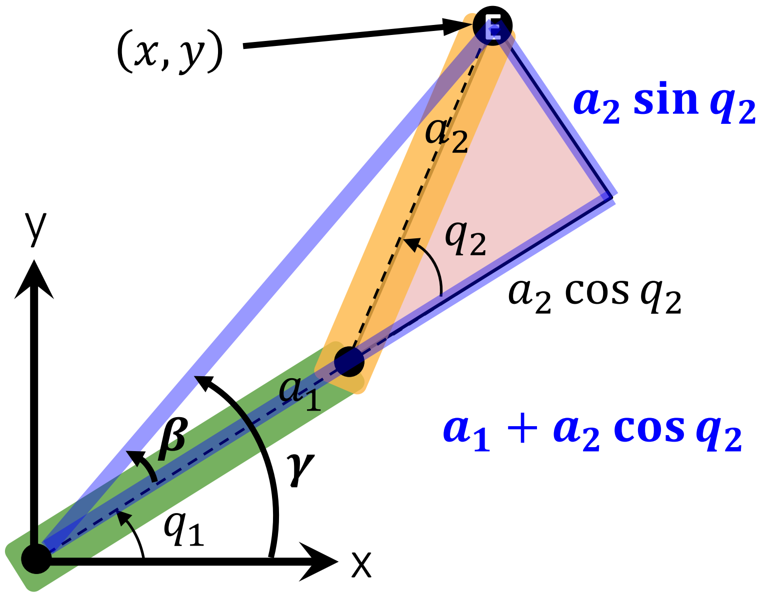
# Cinemática Inversa do Robô RR

- Agora, para encontrar $\beta$, considere as laterais do triângulo:



# Cinemática Inversa do Robô RR

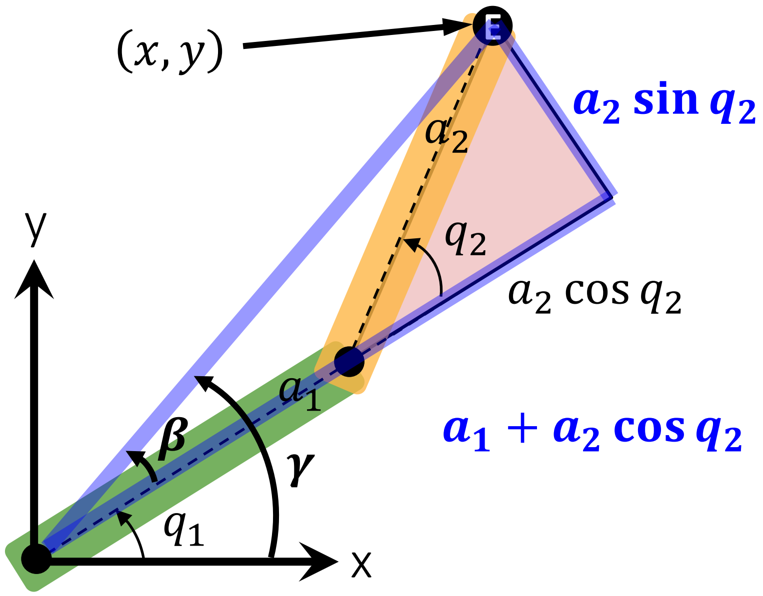
- Assim, as laterais para o novo triângulo ficam:




$$\beta = \tan^{-1}\left(\frac{a_2 \sin q_2}{a_1 + a_2 \cos q_2}\right)$$

# Cinemática Inversa do Robô RR

- Retomando as relações encontradas:




$$q_1 = \gamma - \beta$$

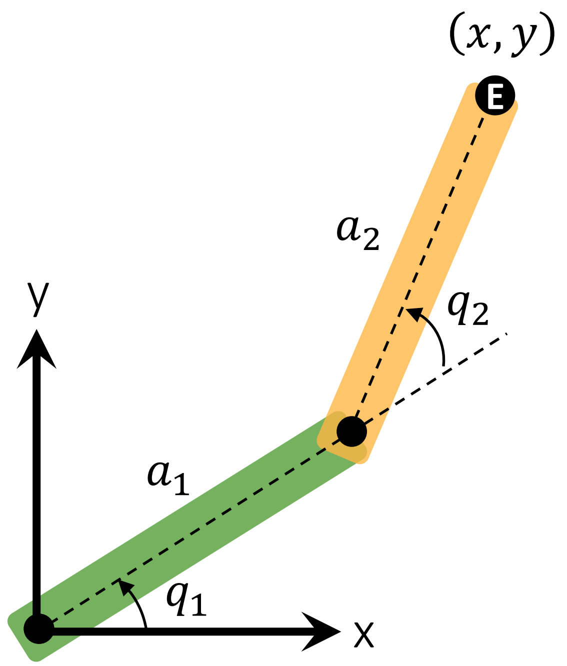
$$\gamma = \tan^{-1}\left(\frac{y}{x}\right)$$

$$\beta = \tan^{-1}\left(\frac{a_2 \sin q_2}{a_1 + a_2 \cos q_2}\right)$$

$$\boxed{q_1 = \tan^{-1}\left(\frac{y}{x}\right) - \tan^{-1}\left(\frac{a_2 \sin q_2}{a_1 + a_2 \cos q_2}\right)}$$

# Cinemática Inversa do Robô RR

- Solução existe se: $\sqrt{x^2+y^2} \in [a_1 - a_2, a_1 + a_2]$



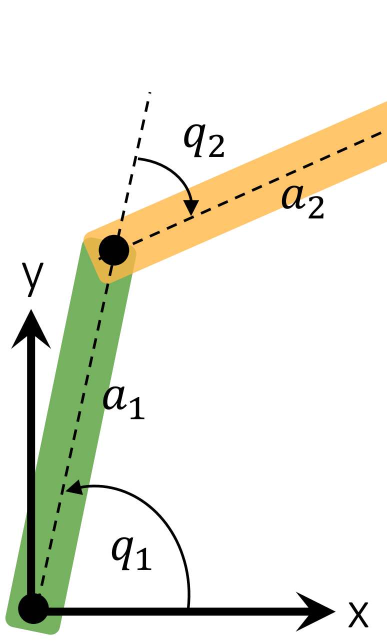

**Se $q_2 \geq 0°$**

$$q_2 = \operatorname{acos}\left(\frac{x^2 + y^2 - a_1^2 - a_2^2}{2a_1 a_2}\right)$$

$$q_1 = \tan^{-1}\left(\frac{y}{x}\right) - \tan^{-1}\left(\frac{a_2 \sin q_2}{a_1 + a_2 \cos q_2}\right)$$

# Cinemática Inversa do Robô RR

- Solução existe se: $\sqrt{x^2 + y^2} \in [a_1 - a_2, a_1 + a_2]$




**Se $q_2 < 0°$**

$$q_2 = -\operatorname{acos}\left(\frac{x^2 + y^2 - a_1^2 - a_2^2}{2a_1a_2}\right)$$

$$q_1 = \tan^{-1}\left(\frac{y}{x}\right) - \tan^{-1}\left(\frac{a_2 \sin q_2}{a_1 + a_2 \cos q_2}\right)$$

# Exercício de Programação

- Visualizando a cinemática inversa de um robô planar de duas juntas;
- Acessar o *Jupyter Notebook* disponível em:

https://colab.research.google.com/drive/1LJphDRc--4bV-BUcKDrGbjF08mZAL8aU

- Visualizar as instruções nos exemplos;
- Resolver o exercício de fixação proposto.

# Exercício de Simulação (CoppeliaSim)

- Apresentação do simulador CoppeliaSim;
- API Remota para linguagem Python;
- Visualizar o modelo de simulação do manipulador planar 2R;
- Incorporar a cinemática inversa no modelo de simulação;
- Realizar uma simulação e verificar o funcionamento;

INSTITUTO FEDERAL
ESPIRITO SANTO

# Cinemática Inversa via Método Numérico

# Cinemática Inversa: Método Numérico

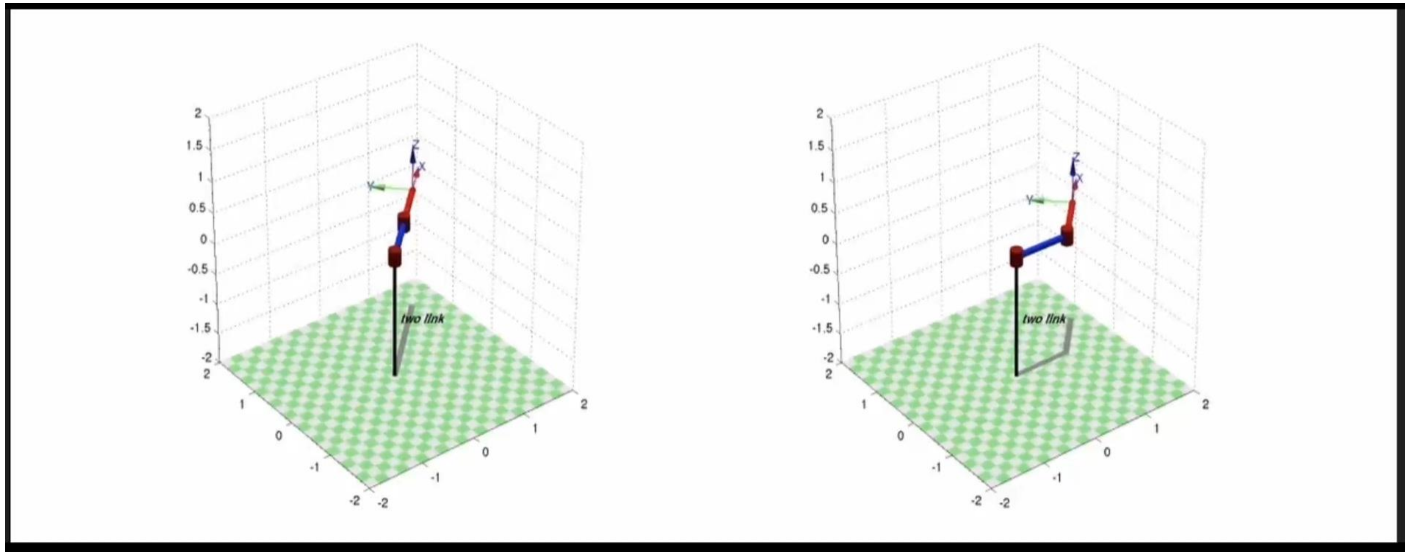
- Avaliando o método analítico aplicado ao robô planar 2R, é possível perceber sua **desvantagem**:
  - Quanto maior a complexidade do arranjo cinemático (3R, SCARA, Articulado 3D, multiarticulado)
  - Maior também é a dificuldade de encontrar a solução analítica para sua cinemática inversa;
- Porém, suas **vantagens** são:
  - Uma vez descoberta, a solução é computacionalmente barata e rápida;
  - Já existem soluções analíticas para vários arranjos cinemáticos na literatura;

# Cinemática Inversa: Método Numérico

- Entretanto, uma metodologia genérica se torna importante quando a solução analítica não é viável.
  - Exemplo: Robôs sobreatuados (7R planar, ou tipo Cobra em 3D);
- Há muitas possibilidades:
  - **Baseadas no Jacobiano do manipulador;**
  - Baseadas em métodos numéricos derivados da série de Taylor (Newton-Raphson, Gradiente Descendente)
  - Baseados em métodos heurísticos (FABRIK).
  - Baseados em volume de dados (Deep Learning, Métodos Estocásticos)
  - Outros

ARISTIDOU, A. et al. Inverse Kinematics Techniques in Computer Graphics: A Survey. *Computer Graphics Forum*, v. 37, n. 6, 2017. [LINK]

# Cinemática de Velocidades: O Jacobiano do Manipulador



Fonte:
https://robotacademy.net.au/masterclass/velocity-kinematics-in-2d/?lesson=321

Lucas Vago Santana

Campus Linhares

# O Jacobiano do Manipulador

- Cinemática Direta:

$\boldsymbol{q} \longrightarrow \boxed{\text{MANIPULADOR}} \longrightarrow \boldsymbol{x}$

- Cinemática Inversa:

$\boldsymbol{x} \longrightarrow \boxed{\text{MANIPULADOR}} \longrightarrow \boldsymbol{q}$

- Jacobiano: Relaciona velocidades

$\dot{\boldsymbol{q}} \longrightarrow \boxed{\text{MANIPULADOR}} \longrightarrow \dot{\boldsymbol{x}}$

$\dot{\boldsymbol{x}} \longrightarrow \boxed{\text{MANIPULADOR}} \longrightarrow \dot{\boldsymbol{q}}$

# Obtendo o Jacobiano: Método analítico sobre a cinemática direta

# O Jacobiano do Manipulador

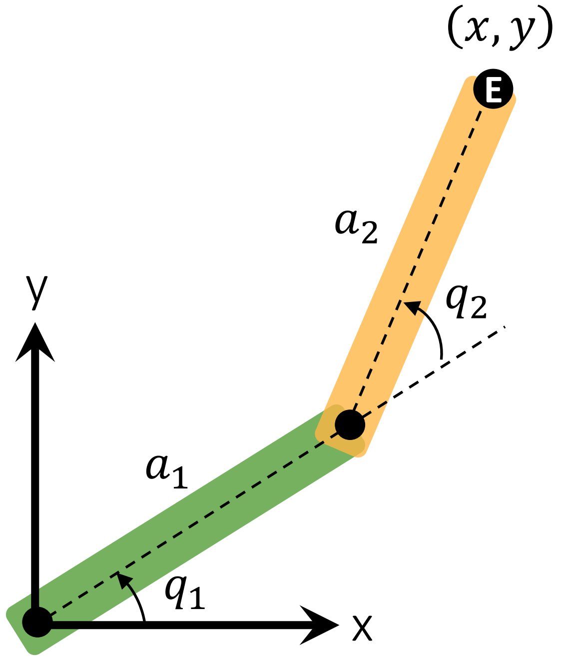
- A cinemática direta do manipulador planar 2R pode ser escrita como:

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} a_1 \cos q_1 + a_2 \cos(q_1 + q_2) \\ a_1 \sin q_1 + a_2 \sin(q_1 + q_2) \end{bmatrix}$$

- Considerando $q_1$ e $q_2$ funções no tempo;
- A derivada temporal, implica em:

*Aplicando-se a regra da cadeia*

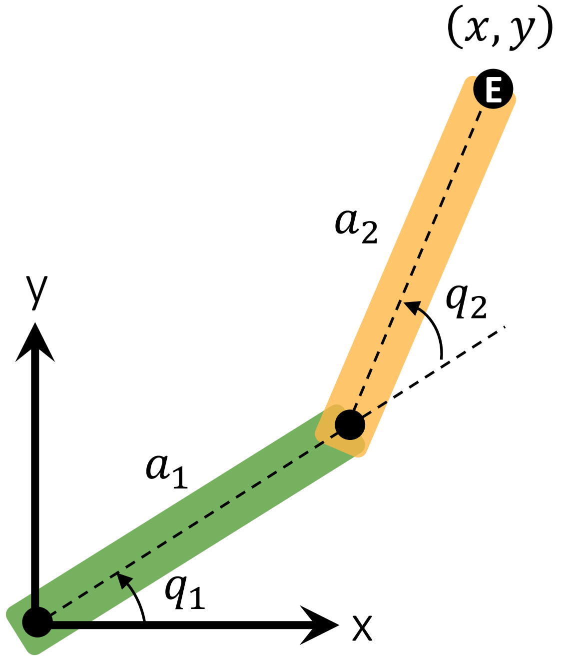
$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} -a_1 \sin q_1 \, \dot{q}_1 - a_2 \sin(q_1 + q_2)\,(\dot{q}_1 + \dot{q}_2) \\ a_1 \cos q_1 \, \dot{q}_1 + a_2 \cos(q_1 + q_2)\,(\dot{q}_1 + \dot{q}_2) \end{bmatrix}$$



# O Jacobiano do Manipulador

- Reorganizando em formato matricial, isolando $\dot{q}_1$ e $\dot{q}_2$ :

$$\begin{bmatrix}\dot{x}\\ \dot{y}\end{bmatrix} = \begin{bmatrix}-a_1 \sin q_1 - a_2 \sin(q_1 + q_2) & -a_2 \sin(q_1 + q_2)\\ a_1 \cos q_1 + a_2 \cos(q_1 + q_2) & a_2 \cos(q_1 + q_2)\end{bmatrix}\begin{bmatrix}\dot{q}_1\\ \dot{q}_2\end{bmatrix}$$




- Simplificando a forma de escrever:

$$\boxed{\boldsymbol{v} = \boldsymbol{J}(\boldsymbol{q})\,\dot{\boldsymbol{q}}}$$

Velocidade Cartesiana do Efetuador Final

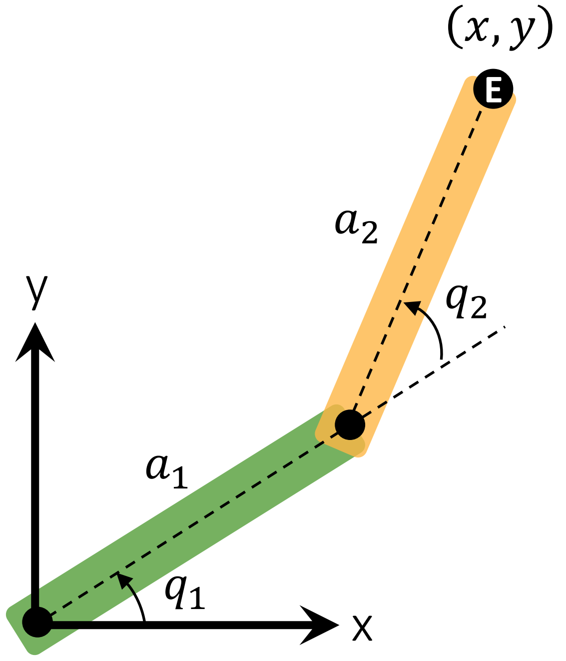
Matriz Jacobiana do Manipulador

Velocidade das Juntas

# Obtendo o Jacobiano: Método analítico generalizado

# O Jacobiano do Manipulador

- Do Cálculo Diferencial e Integral, o Jacobiano pode ser visto um equivalente matricial (multidimensional) da derivada de funções unidimensionais;



# O Jacobiano do Manipulador

- Exemplo unidimensional:
  - Dada a função $y = f(x)$, cujo argumento $x$ e o resultado $y$ são escalares;
  - A derivada de $f$ em relação a $x$ é $\frac{df}{dx} = \frac{dy}{dx}$;
  - Exemplo:

$$y = x^2$$

$$\frac{dy}{dx} = 2x$$

# O Jacobiano do Manipulador

- Exemplo matricial (Jacobiano):
  - Dada a função $\boldsymbol{y} = \boldsymbol{F}(\boldsymbol{x})$, cujo argumento $\boldsymbol{x} \in \mathbb{R}^n$ e o resultado $\boldsymbol{y} \in \mathbb{R}^m$ são vetores de dimensão $n$ e $m$, respectivamente;
  - O Jacobiano será a matriz $m \times n$:

$$\boldsymbol{J}(\boldsymbol{x}) = \frac{\partial \boldsymbol{F}}{\partial \boldsymbol{x}} = \begin{bmatrix} \frac{\partial F_1}{\partial x_1} & \cdots & \frac{\partial F_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial F_m}{\partial x_1} & \cdots & \frac{\partial F_m}{\partial x_n} \end{bmatrix}$$

# O Jacobiano do Manipulador

- Exemplo matricial (Jacobiano):
  - Na função:

$$\boldsymbol{y} = \boldsymbol{F}(\boldsymbol{x}) = \begin{bmatrix} a^2 + 2b + 3abc \\ 2a + 3b - 2c \end{bmatrix} \begin{matrix} \to F_1 \\ \to F_2 \end{matrix}$$

  - Percebe-se os argumentos $\boldsymbol{x} = \begin{bmatrix} a \\ b \\ c \end{bmatrix} \begin{matrix} \to x_1 \\ \to x_2 \\ \to x_3 \end{matrix}$

  - Aplicando o modelo implica em:

$$\boldsymbol{J}(\boldsymbol{x}) = \frac{\partial \boldsymbol{F}}{\partial \boldsymbol{x}} = \begin{bmatrix} \frac{\partial F_1}{\partial x_1} & \cdots & \frac{\partial F_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial F_m}{\partial x_1} & \cdots & \frac{\partial F_m}{\partial x_n} \end{bmatrix} = \begin{bmatrix} \frac{\partial F_1}{\partial x_1} & \frac{\partial F_1}{\partial x_2} & \frac{\partial F_1}{\partial x_3} \\ \frac{\partial F_2}{\partial x_1} & \frac{\partial F_2}{\partial x_2} & \frac{\partial F_2}{\partial x_3} \end{bmatrix}$$

$$\boldsymbol{J}(\boldsymbol{x}) = \begin{bmatrix} 2a + 3bc & 2 + 3ac & 3ab \\ 2 & 3 & -2 \end{bmatrix}$$

# O Jacobiano do Manipulador

- Na aula anterior, a cinemática direta foi apresentada na sua forma genérica como uma função matricial $\mathcal{K}$ de argumentos $\boldsymbol{q}$:

$$\xi = \mathcal{K}(\boldsymbol{q}) \qquad \boldsymbol{q} = \{q_j \ , j \in [1 \ \ \dots \ \ N]\} \in \mathcal{C}$$

- Quando comparado ao modelo genérico do Jacobiano, obtém-se:

$\boldsymbol{y} = \boldsymbol{F}(\boldsymbol{x})$: $\quad \boldsymbol{x} \in \mathbb{R}^n$ e $\boldsymbol{y} \in \mathbb{R}^m$

$$\boldsymbol{J}(\boldsymbol{x}) = \frac{\partial \boldsymbol{F}}{\partial \boldsymbol{x}} = \begin{bmatrix} \frac{\partial F_1}{\partial x_1} & \cdots & \frac{\partial F_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial F_m}{\partial x_1} & \cdots & \frac{\partial F_m}{\partial x_n} \end{bmatrix}$$

$\Rightarrow$

$\xi = \mathcal{K}(\boldsymbol{q})$: $\quad \boldsymbol{q} \in \mathbb{R}^N$ e $\xi \in \mathbb{R}^M$

$$\boldsymbol{J}(\boldsymbol{q}) = \frac{\partial \xi}{\partial \boldsymbol{q}} = \begin{bmatrix} \frac{\partial \xi_1}{\partial q_1} & \cdots & \frac{\partial \xi_1}{\partial q_N} \\ \vdots & \ddots & \vdots \\ \frac{\partial \xi_M}{\partial q_1} & \cdots & \frac{\partial \xi_M}{\partial q_N} \end{bmatrix}$$

# O Jacobiano do Manipulador

- Interpretação:
  - A partir da **cinemática direta** é possível obter o Jacobiano de qualquer arranjo de manipulador;
  - No Jacobiano, a dimensão $N$ representa:
    - O número de juntas do robô, pertencentes ao espaço de configuração;
    - O número de colunas da matriz Jacobiana;
  - Por sua vez, a dimensão $M$ representa:
    - O número de coordenadas espaciais mapeadas, isto é, o espaço de trabalho do manipulador;
    - O número de linhas da matriz Jacobiana

$$\xi = \mathcal{K}(\boldsymbol{q}): \quad \boldsymbol{q} \in \mathbb{R}^N \text{ e } \xi \in \mathbb{R}^M$$

$$\boldsymbol{J}(\boldsymbol{q}) = \frac{\partial \xi}{\partial \boldsymbol{q}} = \begin{bmatrix} \frac{\partial \xi_1}{\partial q_1} & \cdots & \frac{\partial \xi_1}{\partial q_N} \\ \vdots & \ddots & \vdots \\ \frac{\partial \xi_M}{\partial q_1} & \cdots & \frac{\partial \xi_M}{\partial q_N} \end{bmatrix}$$

# O Jacobiano do Manipulador

- **Exemplo:** Retomando o Manipulador Planar 2R.
  - A cinemática direta, na forma genérica é:

$$\xi = \mathcal{K}(\boldsymbol{q}) \qquad \boldsymbol{q} = \{q_j \ , j \in [1 \ \ \dots \ \ N]\} \in \mathcal{C}$$

  - A cinemática direta do Manipulador Planar 2R é:

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} a_1 \cos q_1 + a_2 \cos(q_1 + q_2) \\ a_1 \sin q_1 + a_2 \sin(q_1 + q_2) \end{bmatrix} \begin{cases} \xi = \begin{bmatrix} x \\ y \end{bmatrix} \in \mathbb{R}^2 & \textit{(espaço de trabalho)} \\ \boldsymbol{q} = \begin{bmatrix} q_1 \\ q_2 \end{bmatrix} \in \mathbb{R}^2 & \textit{(espaço de configurações)} \end{cases}$$

# O Jacobiano do Manipulador

- Sendo:

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} a_1 \cos q_1 + a_2 \cos(q_1 + q_2) \\ a_1 \sin q_1 + a_2 \sin(q_1 + q_2) \end{bmatrix} \quad \begin{matrix} \boxed{\xi_1(\boldsymbol{q})} \\ \boxed{\xi_2(\boldsymbol{q})} \end{matrix}$$

- O Jacobiano pode ser obtido como:

$$\boldsymbol{J}(\boldsymbol{q}) = \frac{\partial \xi}{\partial \boldsymbol{q}} = \begin{bmatrix} \dfrac{\partial \xi_1}{\partial q_1} & \dfrac{\partial \xi_1}{\partial q_2} \\ \dfrac{\partial \xi_2}{\partial q_1} & \dfrac{\partial \xi_2}{\partial q_2} \end{bmatrix} = \begin{bmatrix} -a_1 \sin q_1 - a_2 \sin(q_1 + q_2) & -a_2 \sin(q_1 + q_2) \\ a_1 \cos q_1 + a_2 \cos(q_1 + q_2) & a_2 \cos(q_1 + q_2) \end{bmatrix}$$

Atenção nas derivadas parciais!

# Obtendo o Jacobiano: Método simbólico usando Python

# Exercício de Programação

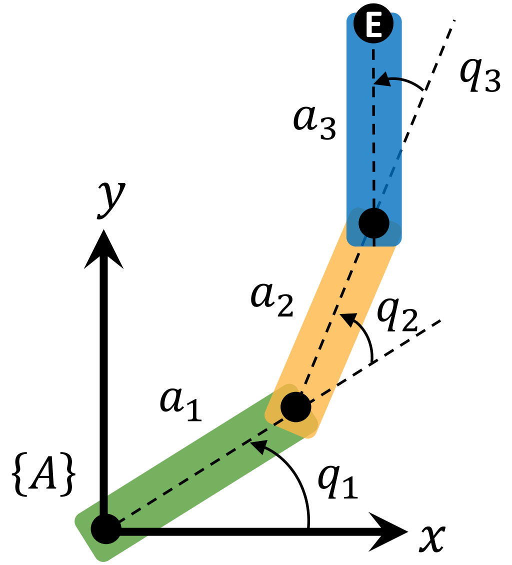
- Acessar o *Jupyter Notebook* disponível em:

https://colab.research.google.com/drive/1p4r8VLz1Rt0Y38CaTs8ZSGd_Burz4wC3

- Visualizar as instruções nos exemplos;
- Resolver o exercício de fixação proposto.

# Exercício

- Obter o Jacobiano do manipulador planar 3R:
  - Analiticamente pelo método generalizado;
  - Conferir pelo método simbólico;



# A Inversão do Jacobiano

# A Inversão do Jacobiano

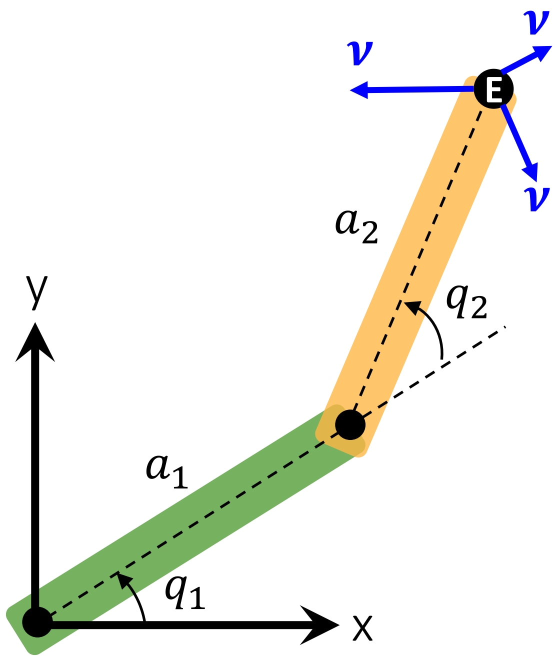
- Retomando a forma genérica:

$$\boldsymbol{\nu} = \boldsymbol{J}(\boldsymbol{q})\,\dot{\boldsymbol{q}}$$

- Supondo $\boldsymbol{J}(\boldsymbol{q})$ inversível, é possível escrever:

$$\dot{\boldsymbol{q}} = \boldsymbol{J}(\boldsymbol{q})^{-1}\,\boldsymbol{\nu}$$

- Que é uma **informação muito útil**, pois:
  - Diz como as velocidades angulares $(\dot{\boldsymbol{q}})$ das juntas refletem na velocidade cartesiana $(\boldsymbol{\nu})$do efetuador final;
  - Isso é útil no controle do robô, pois ao estabelecer a velocidade desejada $(\boldsymbol{\nu})$ do efetuador final, pode-se determinar quais velocidades das juntas irão controlá-la;



# A Inversão do Jacobiano

- **O problema:**
  - Retomando o caso 2R:

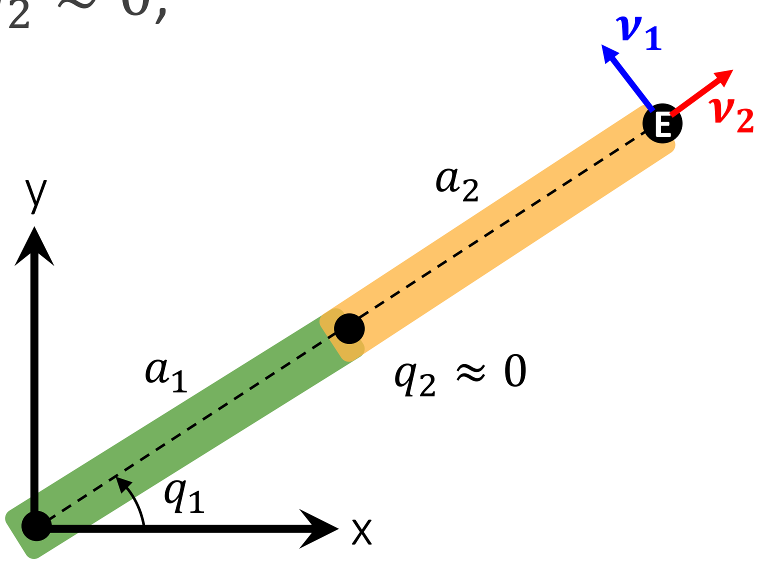
$$J(q) = \begin{bmatrix} -a_1 \sin q_1 - a_2 \sin(q_1 + q_2) & -a_2 \sin(q_1 + q_2) \\ a_1 \cos q_1 + a_2 \cos(q_1 + q_2) & a_2 \cos(q_1 + q_2) \end{bmatrix}$$

$$J^{-1}(q) = \frac{1}{a_1 a_2 \sin q_2} \begin{bmatrix} a_2 \cos(q_1 + q_2) & a_2 \sin(q_1 + q_2) \\ -a_1 \cos q_1 - a_2 \cos(q_1 + q_2) & -a_1 \sin q_1 - a_2 \sin(q_1 + q_2) \end{bmatrix}$$

Infinito quando $q_2 \approx 0$

# A Inversão do Jacobiano

- **Interpretação do problema:**
  - Estas situações são chamadas de **singularidades do manipulador**;
  - São condições nas quais haverá dificuldade de realizar determinado movimento;
  - No exemplo, o 2R perde um grau de liberdade quando $q_2 \approx 0$;
  - Fisicamente é o mesmo que:
    - $v_1$ é uma velocidade possível de se executar na singularidade;
    - $v_2$ é uma velocidade impossível de se executar na singularidade;
  - Matematicamente, identificam-se singularidades por:
    - $\det[\boldsymbol{J}(\boldsymbol{q})] \approx 0$;
    - E $\boldsymbol{J}(\boldsymbol{q})$ não pode ser invertida;



# A Inversão do Jacobiano

- **A solução computacional:**
  - A álgebra linear, a inversa de uma matriz pode ser definida apenas quando ela é **quadrada e não-singular**;
  - A alternativa é a **Pseudo-Inversa de Moore e Penrose**, definida para qualquer matriz $A_{m \times n}$, como $A^+$:

$$A^+ = [A^T A]^{-1} A^T$$

  - Que para funções no formato $y = Ax$, minimiza a norma $||y - Ax||$, o que significa:

$$A^+ y = A^+ A x \rightarrow A^+ y \approx x$$

# A Inversão do Jacobiano

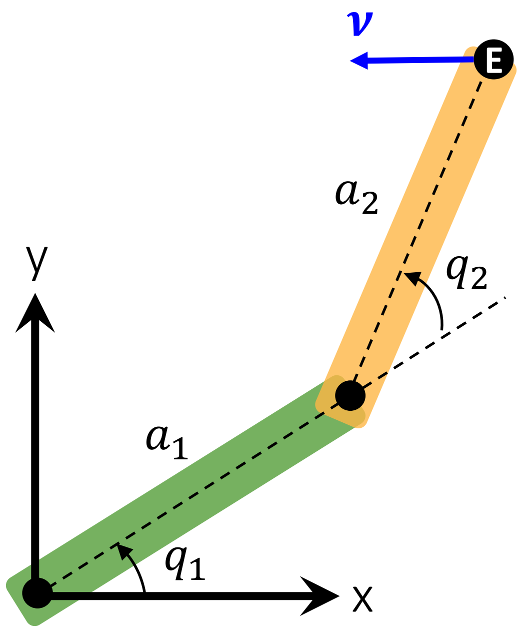
- **Problema da solução computacional (Pseudoinversa):**
  - Nas singularidades ela pode causar descontinuidade de sinais;
  - Na prática, o controle do mecanismo insere grandes ações de controle em circunstâncias que não deveria;
  - Cujo efeito prático é o *chattering* ou trepidação das juntas em singularidade;

INSTITUTO FEDERAL
ESPIRITO SANTO

# *Resolved Rate Motion Control*
# ou
# Controle Resolvido por Taxa de Movimento

# Controle Resolvido por Taxa de Movimento

- É um conceito que faz uso das velocidades cartesianas e das juntas do manipulador para controla-lo:
  1) Dada uma velocidade cartesiana $\boldsymbol{v}$;
  2) Determinar $\dot{\boldsymbol{q}} = \boldsymbol{J}(\boldsymbol{q})^{-1}\,\boldsymbol{v}$;
  3) Em seguida, comandar o robô para se mover com $\dot{\boldsymbol{q}}$;
  4) Após um intervalo de amostragem $\Delta t$, atualizar $\boldsymbol{J}(\boldsymbol{q})^{-1}$;
  5) E repetir todos os cálculos;



# Controle Resolvido por Taxa de Movimento

- **No contexto computacional**, aplicar o controle resolvido por taxa de movimento significa:
  - Calcular as velocidades das juntas para um determinado instante discreto $k$:

$$\dot{\boldsymbol{q}}_k = \boldsymbol{J}(\boldsymbol{q}_k)^{-1}\,\boldsymbol{v}_k$$

  - Calcular a posição das juntas, baseado em integração numérica:

$$\boldsymbol{q}_{k+1} = \boldsymbol{q}_k + \dot{\boldsymbol{q}}_k \Delta t$$

  - Mover o robô para posição $\boldsymbol{q}_{k+1}$;
  - Atualizar o jacobiano para $\boldsymbol{J}(\boldsymbol{q}_{k+1})$ e refazer todos os cálculos;

# Controle via cinemática inversa: Solução de linearização por retroalimentação do Jacobiano Invertido

# Controle via Cinemática Inversa

- Partindo-se da relação:

$$\boldsymbol{v} = \dot{\boldsymbol{x}} = \boldsymbol{J}(\boldsymbol{q})\,\dot{\boldsymbol{q}}$$

- Para fins de controle, uma **escolha inteligente** para um sinal de controle $\dot{\boldsymbol{q}}$ é:

$$\dot{\boldsymbol{q}} = \boldsymbol{J}(\boldsymbol{q})^{-1}\,(\dot{\boldsymbol{x}}_{\boldsymbol{d}} + \boldsymbol{K}\tilde{\boldsymbol{x}})$$

- Onde:
  - $\boldsymbol{x}$: vetor de estados corrente (atual);
  - $\boldsymbol{q}$: configuração do robô;
  - $\boldsymbol{x}_{\boldsymbol{d}}$: vetor de estados desejado;
  - $\tilde{\boldsymbol{x}} = \boldsymbol{x}_{\boldsymbol{d}} - \boldsymbol{x}$: vetor de erro de estados;
  - $\boldsymbol{K}$: matriz diagonal $n \times n$, de ganhos positivos;
  - $n$: dimensão do espaço de estados;
- Pois ...

# Controle via Cinemática Inversa

- Pois substituir a equação (2) na (1), resulta em:

$$\dot{\boldsymbol{x}} = \boldsymbol{J}(\boldsymbol{q})\,\dot{\boldsymbol{q}} \qquad (1)$$
$$\dot{\boldsymbol{q}} = \boldsymbol{J}(\boldsymbol{q})^{-1}\,(\dot{\boldsymbol{x}}_d + \boldsymbol{K}\widetilde{\boldsymbol{x}}) \qquad (2)$$
$$\dot{\boldsymbol{x}} = \dot{\boldsymbol{x}}_d + \boldsymbol{K}\widetilde{\boldsymbol{x}} \qquad \textit{sistema linearizado}$$
$$\dot{\widetilde{\boldsymbol{x}}} + \boldsymbol{K}\widetilde{\boldsymbol{x}} = \boldsymbol{0} \qquad (3)$$

- A Eq. (3) representa uma dinâmica linear, globalmente assintoticamente, estável para o sinal de erro $\widetilde{\boldsymbol{x}}$;

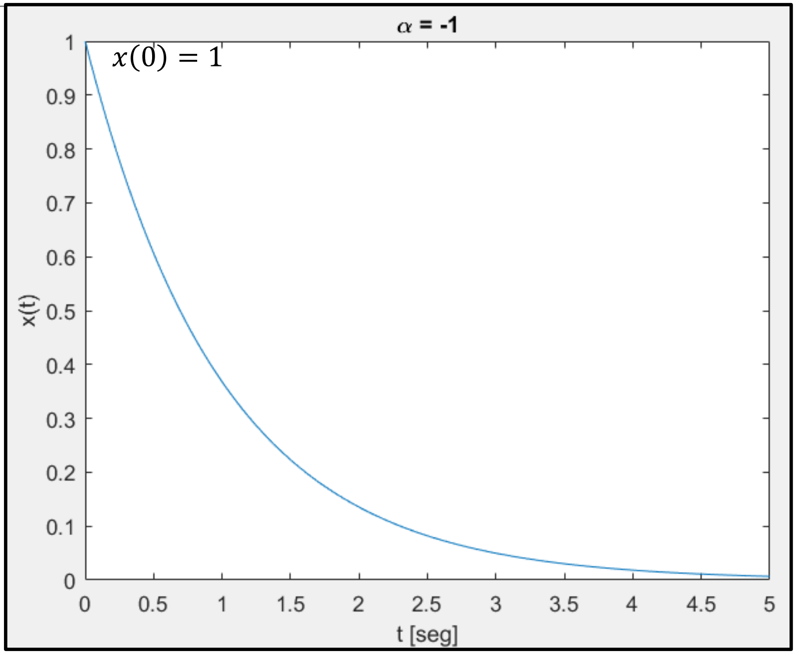
- Em outras palavras, este formato garante que para **qualquer condição inicial do erro**, o sinal de controle da Eq. (2) o conduz a $\widetilde{\boldsymbol{x}} \rightarrow \boldsymbol{0}$, para $t \rightarrow \infty$;

# Controle via Cinemática Inversa

- OK, mas de onde vem essa conclusão?
- **Resposta:**
  - Da análise de estabilidade de sistemas lineares invariantes no tempo (LTIs);
  - Isto é, sistemas dinâmicos, cuja resposta é sempre a mesma, dadas as mesmas condições iniciais;
  - Especificamente, neste caso, após a linearização (que é uma técnica de controle não-linear), a conclusão da estabilidade sai da análise da resposta do **Sistema Linear Autônomo Multivariável** representado na Eq. (3)**;**
  - Lembrando que um **Sistema Autônomo** é aquele cuja **dinâmica** não depende explicitamente da variável independente (exemplo o $t$);

# Controle via Cinemática Inversa

- Exemplo Unidimensional:
  - **Sistema Autônomo:** $\dot{x} = ax$
  - **Solução desta EDO:** $x(t) = x_0 e^{at}$
    - $x_0$ – condição inicial
    - $a$ – parâmetro da equação
  - **Conclusão:**
    - Se $\alpha < 0$,
    - $x(t) \to 0$ quando $t \to \infty$;




- Comparação com Eq. (3):
  - $\dot{\tilde{\boldsymbol{x}}} = -\boldsymbol{K}\tilde{\boldsymbol{x}}$
  - A análise do caso matricial não é idêntica, mas possui conclusão é similar;

# (Extra) Solução Sistema Linear Autônomo

- Dado $\dot{x} = ax$, mostrar que a solução é $x(t) = x_0 e^{at}$:
- Pré-multiplicando por $e^{-at}$:
  - $e^{-at}\dot{x} = e^{-at}ax$
  - $e^{-at}\dot{x} - e^{-at}ax = 0$
  - $(x\,.\,e^{-at})' = 0$

Regra do produto:
$(f\,.\;g)' = f'\,g + f\,g'$

- Integrando de 0 a $t$:
  - $x\,.\,e^{-a\tau}\,|_0^t = 0$
  - $x(t)\,.\,e^{-at} - x(0)\,.\,e^{-a0} = 0$
  - $x(t)\,.\,e^{-at} - x(0) = 0$
  - $x(t) = x_0 e^{at}$
- Logo se $\alpha < 0$, $x(t) \rightarrow 0$ quando $t \rightarrow \infty$;
- A análise do caso matricial não é idêntica, mas possui conclusão é similar;

# Exercício de Programação

- Acessar o *Jupyter Notebook* disponível em:

https://colab.research.google.com/drive/1uW8iso7pyKP7mB8KPABd3Ebw-3hv0aXR

- Visualizar as instruções nos exemplos;
- Resolver o exercício de fixação proposto.