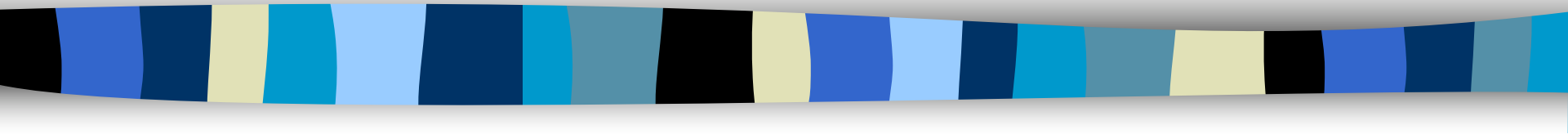
Robótica

Prof. Reinaldo Bianchi
Centro Universitário FEI
2016

4ª Aula



Pós Graduação IECAT

Objetivos desta aula

- Modelo cinemático inverso:
  - Métodos analíticos (ou soluções fechadas):
    - Geométrico (por Trigonometria).
    - Algébrico.
- Matlab.

Bibliografia

- Capítulos 4 do Craig.
- Robot Manipulators: Mathematics, Programming, and Control
  - Paul, R. P. - 1982 - MIT Press.
- Robot Analysis: The Mechanics of Serial and Parallel Manipulators
  - Lung-Wen TSAI - 1999 - John Wiley.

Cinemática Inversa no Matlab

Robotics Toolbox para o Matlab

- Toolbox de livre distribuição (9ª edição):
  - http://petercorke.com/Robotics_Toolbox.html
- Possui modelo de alguns manipuladores prontos:
  - PUMA560
  - Stanford Arm
- Permite criar seu próprio modelo.

Uso

- Copie o toolbox de :

  w:\eng\ele\bianchi\robotica\rvctools
- para o diretório c:\alunos
- Mude de diretorio no matlab:

  cd c:\alunos\rvctools
- O comando abaixo debe ser executado antes de iniciar o uso do toolbox:

  startup_rvc

Criando um robô no Matlab

Ângulos em radianos

- Criando os links:
  - `L = Link([`θ_i d_i a_i α_i`], opção)`
  - onde opção (que é opcional) =
    - ‘standard’ – parametros padrão.
    - ‘r’ para Rotacional (default) e
    - ‘p’ para Prismática
- Criando o robô:
  - `r = SerialLink([Link1 Link2 ...])`

Criando um manipulador 3R

- Queremos criar o seguinte manipulador:
  - 3 juntas rotacionais no eixo z
  - links de 1 metro cada.
- Parâmetros D-H:

Link	a_i	α_i	d_i	θ_i
1	1	0	0	θ_1
2	1	0	0	θ_2
3	1	0	0	θ_3

Comandos para criar um 3R

- Criando os links:

```
L1 = Link([0 0 1 0])
L2 = Link([0 0 1 0])
L3 = Link([0 0 1 0])
```

- Criando o robô:

```
r = SerialLink([L1 L2 L3])
```

Computando a cinemática inversa

- A função **`ikine`** é usada para computar a cinemática inversa:
  - **`q = robot.ikine(T,Q,M)`**

  onde:
  - **`robot`** = o robô.
  - T = a matriz de transformação
  - Q = situação atual do robô
  - M = matriz máscara
  - **`RETORNA q`** = vetor da posição das juntas.

A função ikine

- A solução é encontrada iterativamente utilizando a pseudo-inversa do Jacobiano:
  - $\dot{\underline{q}} = \mathbf{J}^{+}(\underline{q})\Delta\left(\mathrm{F}\,(\underline{q}) - T\right)$
  - A solução é geral (válida para qualquer robô).
  - Mais lenta que soluções específicas.

Usando ikine mais cuidadosamente

- r.ikine(T, Q, M), onde:
  - R = robô;
  - T = transformada com a posição desejada
  - Q = situação atual do robô
  - M = matriz máscara:
  - If the manipulator has fewer than 6 DOF then this method of solution will fail, since the solution space has more dimensions than can be spanned by the manipulator joint coordinates. In such a case it is necessary to provide a mask matrix, M, which specifies the Cartesian DOF (in the wrist coordinate frame) that will be ignored in reaching a solution. The mask matrix has six elements that correspond to translation in X, Y and Z, and rotation about X, Y and Z respectively. The value should be 0 (for ignore) or 1. The number of non-zero elements should equal the number of manipulator DOF.

r.ikine(T, Q, M)

- Definindo T:

  T = [1 0 0 1; 0 1 0 1; 0 0 1 0; 0 0 0 1]
- Definindo Q:

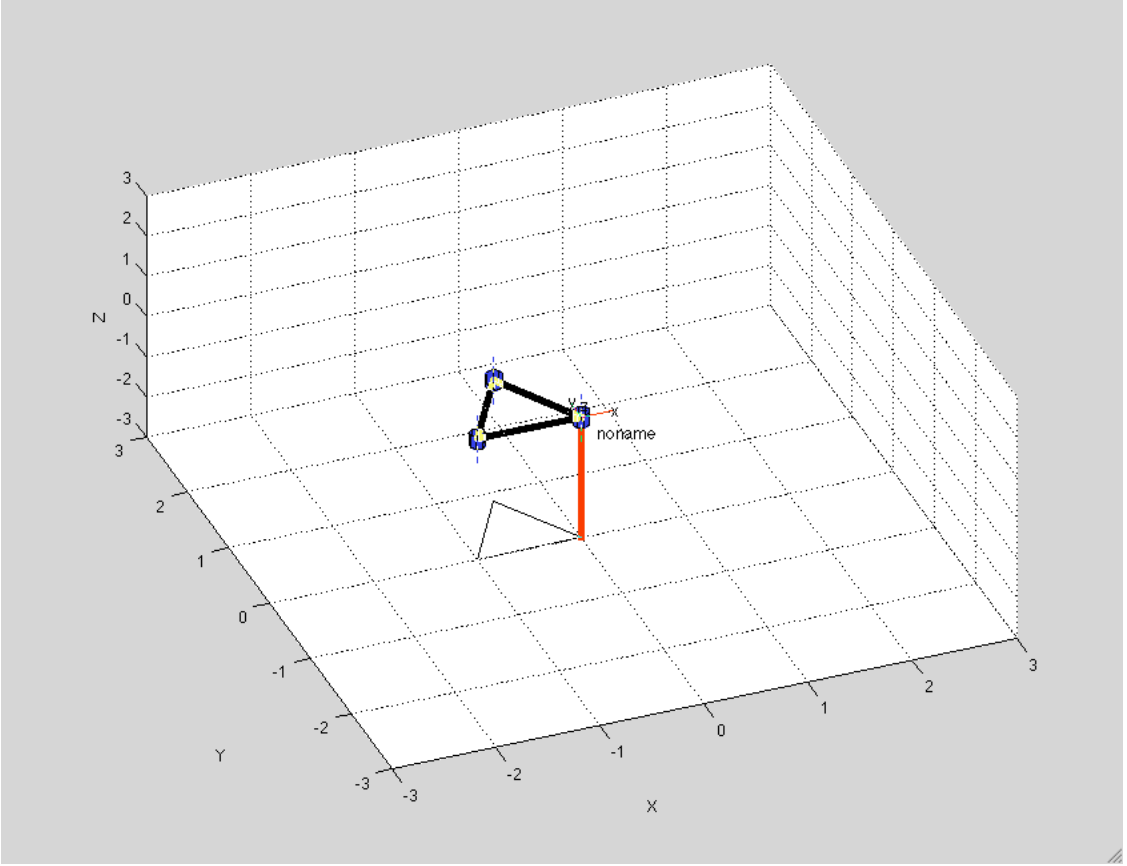
  Q = [2* pi/3 2*pi/3 2*pi/3]
- Definindo M:

  M = [1 1 0 0 0 1]
- Usando ikine():

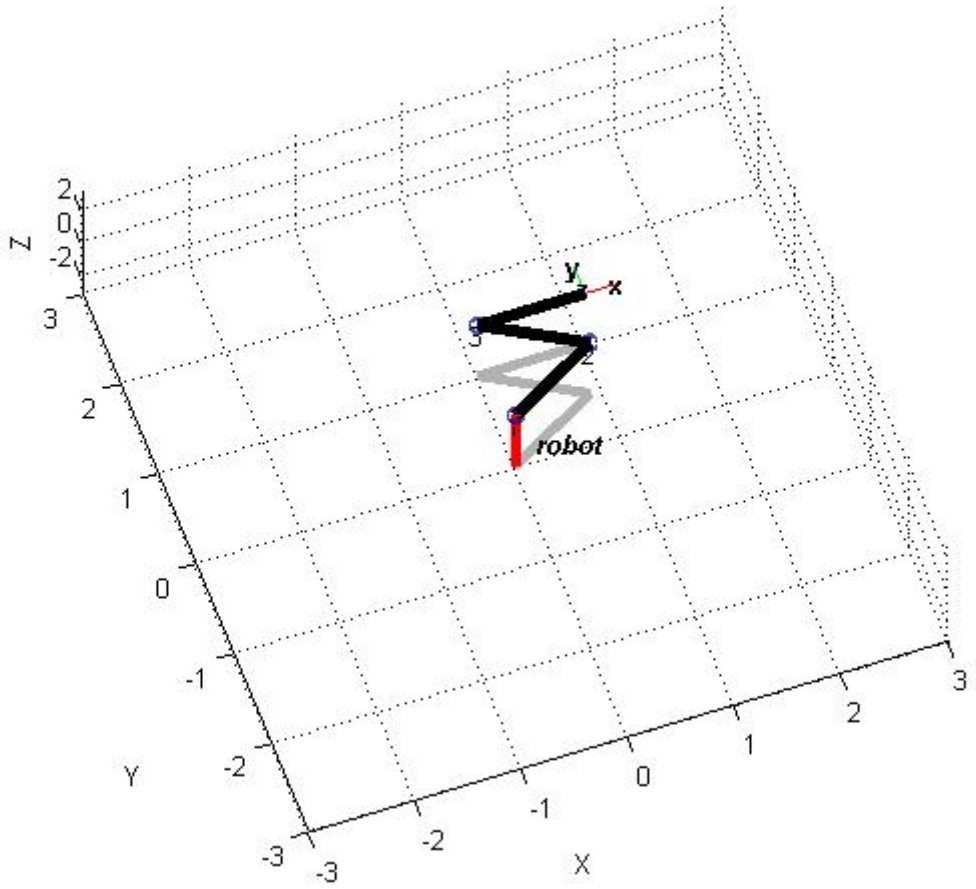
  r.ikine (T, Q, M)

  ans = 1.8235 2.6362 1.8235

Posicão [2π/3 2π/3 2π/3]



Resultado



Cinemática inversa do Puma

- O Toolkit ja possui implementada a cinemática inversa de robôs como o Puma 560

  p560.ikine6s(Matriz)

  – Abra o arquivo ikine6s e veja o que tem.

- Teste com a matriz T:

  mdl_puma560

  p560.ikine6s(T)

  p560.plot(ans)

Metodo numérico versus solução fechada

- ikine () calcula a cinemática inversa de qualquer robô, usando um método numérico.
- ikine6s() calcula a cinemática inversa apenas para robos como Puma 560, usando as equações de cinemática inversa.
- Verificar em casa que usar ikine6s é mais rápido que usar ikine.

Exercício 1:

- Calcular a posição das juntas de um manipulador 3R para os pontos:

X	Y
0	0
0	0,5
0	1
0,5	1
1	1

- E faca a animação do robô nessas posições, usando o r.plot.

Exercício 2:

- Implementar, a partir das equações mostradas na aula teórica, a cinemática inversa para o manipulador 3R

$$\theta_1 = \operatorname{atan} 2(y, x) \pm \frac{x^2 + y^2 + {l_1}^2 - {l_2}^2}{2 l_1 \sqrt{x^2 + y^2}}$$

$$\theta_2 = \arccos\left(\frac{x^2 + y^2 - {l_1}^2 - {l_2}^2}{2 l_1 l_2} \right)$$

$$\theta_3 = \varphi - (\theta_1 + \theta_2)$$

Lista de Exercícios

- Podem começar a fazer a lista de exercícios número 1.

Matlab help

- who: mostra as variáveis
- clear: limpa a memória.
- clc: limpa a tela de comando.
- cd: muda de diretório.
- help general: o básico
- workspace: mostra graficamente os objetos existentes.