

CIMAT

# Proyecto Final

# Métodos Numéricos 2019

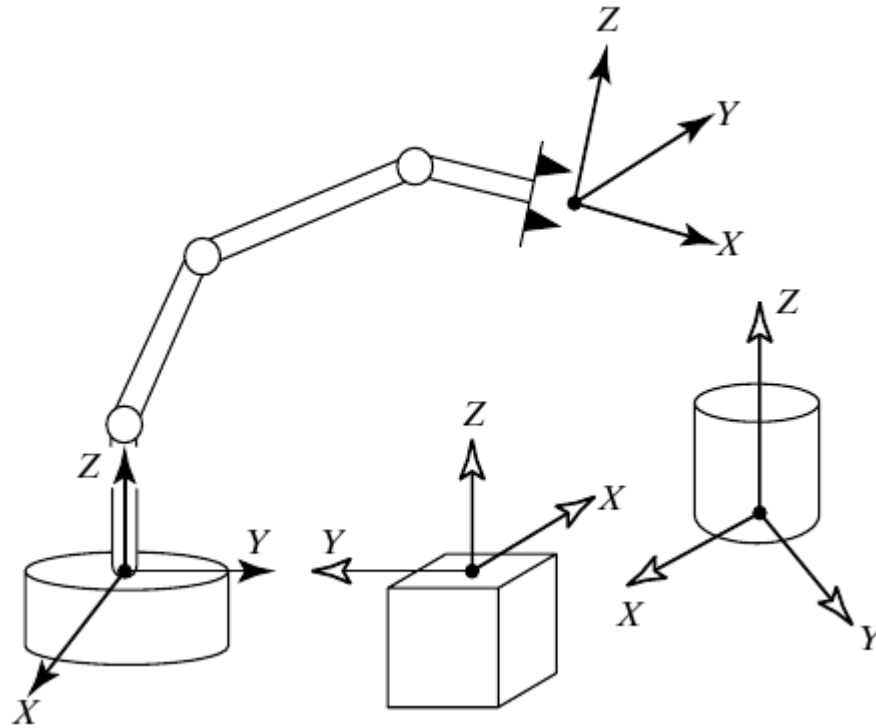
Solución Numérica al Problema de la Cinemática Inversa

Marco Antonio Esquivel Basaldua

# Descripción de Posición y Orientación

- Se expresa mediante la aplicación de matrices de transformación homogénea.

$$T = \begin{bmatrix} R & P \\ 0 & 1 \end{bmatrix}$$



# Cinemática Directa e Inversa

- En cinemática directa se conoce el valor de las articulaciones y se desea conocer la posición del efector final.
- En cinemática inversa se conoce la posición del efector final y se desean conocer los valores de las articulaciones.

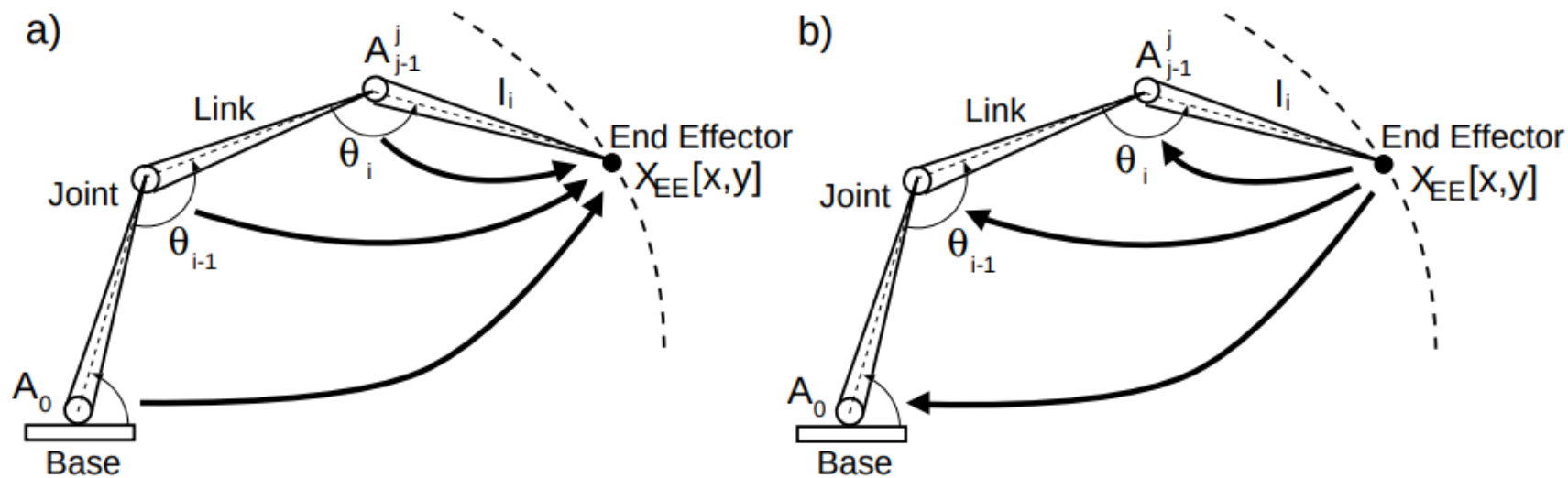
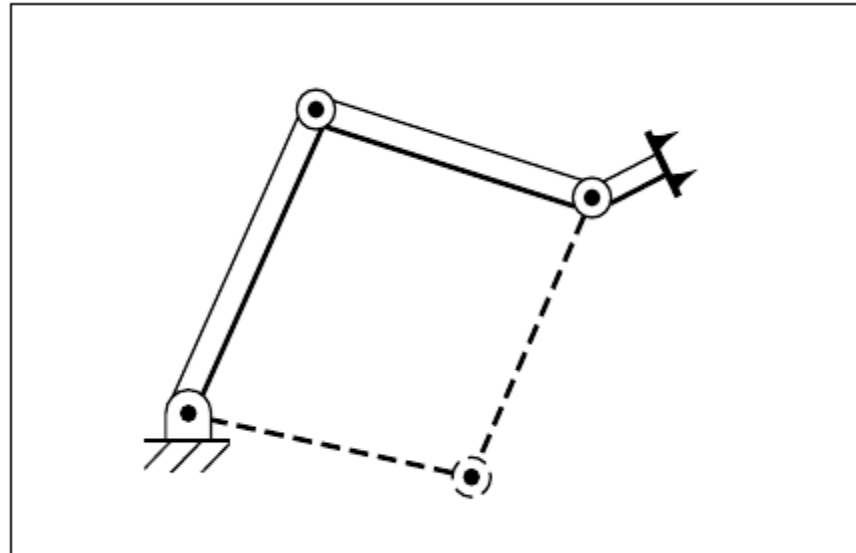


Figure 1: Process of manipulation of the structure through a) *forward* b) *inverse* kinematics.

# El Problema de la Cinemática Inversa

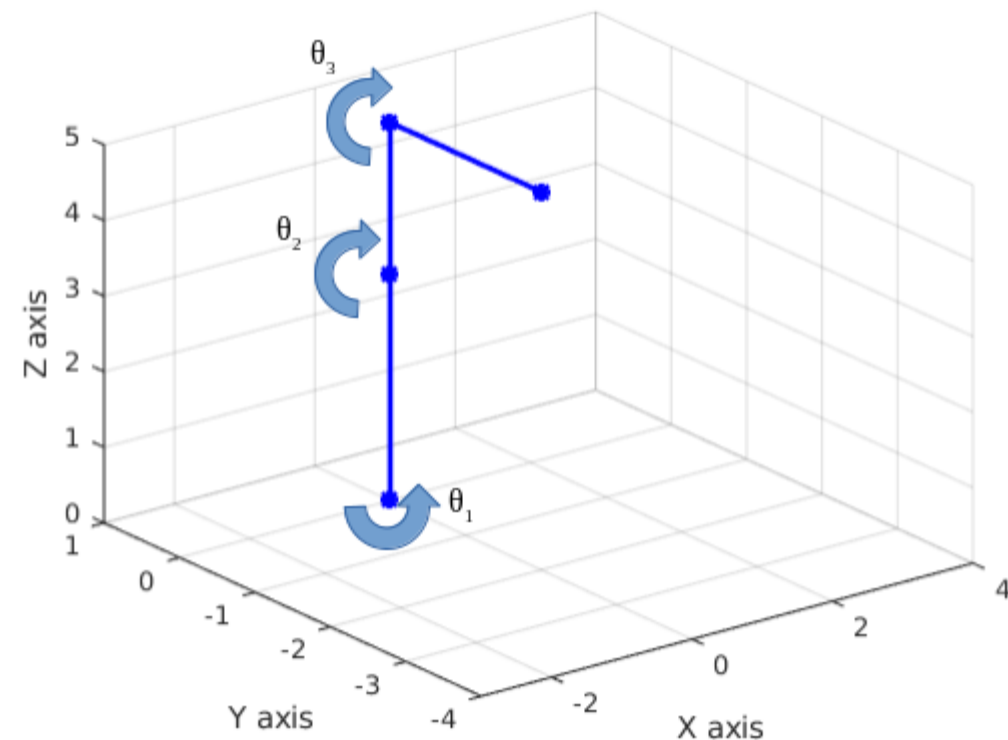
- Las ecuaciones cinemáticas son no lineales
- Puede no haber solución
- Puede haber más de una solución



# Definición del Problema

- Se tiene un brazo articulado de 3 grados de libertad, cuyos eslabones tienen longitud.

$$l_1, l_2, l_3$$



$$P = \begin{bmatrix} l_3 C\theta_1 C\theta_2 C\theta_3 - l_3 C\theta_1 S\theta_2 S\theta_3 + l_2 C\theta_1 C\theta_2 \\ l_3 S\theta_1 C\theta_2 C\theta_3 - l_3 S\theta_1 S\theta_2 S\theta_3 + l_2 S\theta_1 C\theta_2 \\ l_1 + l_2 S\theta_2 + l_3 C\theta_2 S\theta_3 + l_3 S\theta_2 C\theta_3 \end{bmatrix}$$

Posición inicial del manipulador:

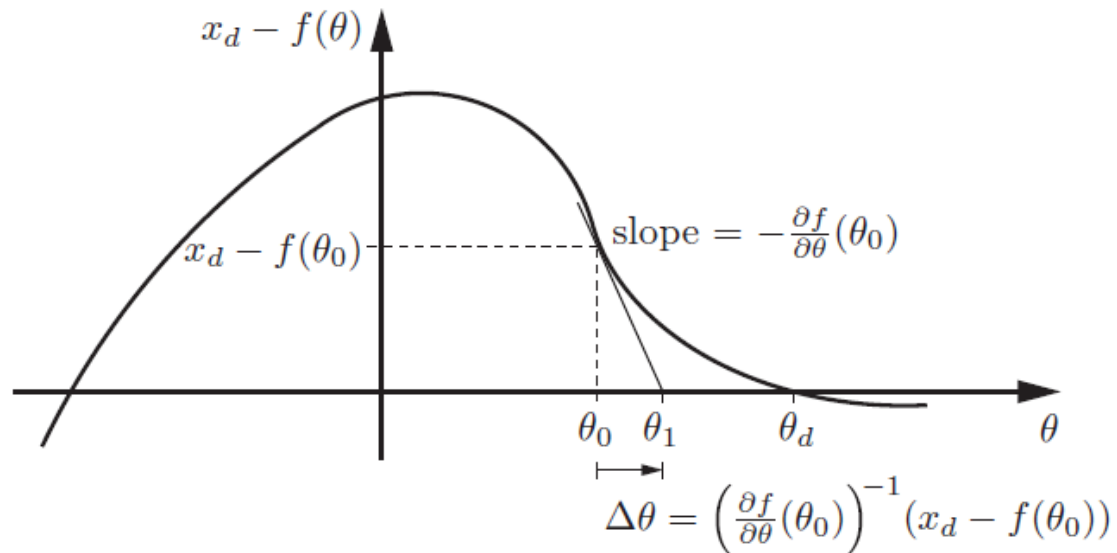
$$P_0 = \begin{bmatrix} 0 \\ -l_3 \\ l_1 + l_2 \end{bmatrix}$$

# Solución numérica

- Las coordenadas del efector final se expresan por  $P$
- La cinemática directa del efector final es  $P = f(\theta)$
- La posición deseada del efector final es  $x_d$
- Sea  $g(\theta_d) = x_d - f(\theta_d)$ , el objetivo es usar el método de Newton-Raphson para encontrar  $\theta_d$  tal que:

$$g(\theta_d) = x_d - f(\theta_d) = 0.$$

# Solución numérica



Se obtiene el set:  $\{\theta_0, \theta_1, \theta_2, \dots\}$ , hasta converger a  $\theta_d$

$$x_d = f(\theta_d) = f(\theta_0) + \frac{\partial f}{\partial \theta} \Big|_{\theta_0} (\theta_d - \theta_0) + h.o.t.$$

Truncando la expansión de Taylor:

$$J(\theta_0)\Delta\theta = x_d - f(\theta_0)$$

$$\Delta\theta = J^{-1}(\theta_0)(x_d - f(\theta_0))$$

$$(\theta_1 - \theta_0) = J^{-1}(\theta_0)(x_d - f(\theta_0))$$

$$\theta_1 = \theta_0 + J^{-1}(\theta_0)(x_d - f(\theta_0))$$

$$\theta_{i+1} = \theta_i + J^{-1}(\theta_i)(x_d - f(\theta_i))$$

Para el calculo de la inversa se utiliza la factorización LU por el método de Doolittle y la solución de sistemas de ecuaciones lineales.

# Solución numérica

El algoritmo iterativo de Newton-Raphson para encontrar  $\theta_d$  es:

---

1. Dado  $x_d$  y la estimación inicial  $\theta_0$ , inicializar  $i = 0$
  2. Definir  $e = x_d - f(\theta_i)$ . Mientras  $\|e\| > \epsilon$ , para un valor pequeño de  $\epsilon$ 
    - Definir  $\theta_{i+1} = \theta_i + J^{-1}(\theta_i)e$
    - Incrementar  $i$
-





CIMAT

# Resultados

Para un ejemplo en el que:

$$l_1 = 3, l_2 = 2, l_3 = 2$$

solicitando que se pase por los puntos:

$$P_1 = \begin{bmatrix} 2 \\ -3 \\ 3 \end{bmatrix}, P_2 = \begin{bmatrix} 0 \\ -2 \\ 2 \end{bmatrix}, P_3 = \begin{bmatrix} 0 \\ -2 \\ 5 \end{bmatrix},$$

Con los valores de las longitudes dadas se tiene un valor  $P_0$ :

$$P_0 = \begin{bmatrix} 0 \\ -2 \\ 5 \end{bmatrix},$$

```
→ CODIGO ./a.out
Visited points:
-2.44929e-16 -2 5
1.99318 -2.9777 2.98835
0.208905 -2.08772 2.10641
-2.07174e-16 -1.97703 4.40638
```

Para un valor  $\epsilon = 0.1$

angles.txt			
1	1.5708	1.5708	1.5708
2	1.6708	1.6208	1.6208
3	1.75614	1.69991	1.63712
4	1.83087	1.79347	1.63065
5	1.89688	1.89866	1.60178
6	1.95508	2.01276	1.55047
7	2.00648	2.13363	1.47668
8	2.05198	2.26014	1.37988
9	2.09239	2.39243	1.25838
10	2.12842	2.53232	1.10829
11	2.16067	2.6843	0.921095
12	2.12793	2.61251	1.1143
13	2.09184	2.56526	1.27329
14	2.05141	2.53194	1.41224
15	2.00586	2.51134	1.53404
16	1.9544	2.50263	1.64088
17	1.89611	2.5055	1.7337
18	1.83002	2.51971	1.81277
19	1.75513	2.54484	1.87781
20	1.67053	2.58009	1.92827
21	1.57562	2.62401	1.96342
22	1.57077	2.46655	2.03598
23	1.5708	2.30154	2.08431
24	1.5708	2.14008	2.10636
25	1.5708	1.98921	2.10189
26	1.5708	1.85616	2.07116
27	1.5708	1.74639	2.01559
28	1.5708	1.6624	1.93728
29	1.5708	1.60421	1.83815
30	1.5708	1.57049	1.71934
31	1.5708	1.55971	1.58079

# Resultados

Para un valor  $\epsilon = 1 \times 10^{-12}$ ,  
se obtiene un set de 122 tríos de ángulos

```
→ CODIGO ./a.out  
Visited points:  
-2.44929e-16 -2 5  
2 -3 3  
0.2 -2.1 2.1  
2.01609e-16 -2 4.96
```

# Resultados

MATLAB R2019b - online course use

HOME PLOTS APPS EDITOR PUBLISH VIEW

New Open Save Find Files Compare Go To Find Insert Comment Indent Breakpoints Run Run and Advance Run Section Advance Run and Time

FILE NAVIGATE EDIT BREAKPOINTS RUN

Search Documentation Sign In

C:\Users\Marco\Documents\Proyecto MN\PROYECTO\_M2019\_MCMESQ\CODIGO

Current Folder

- .vscode
- a.out
- angles.txt
- compute\_dh\_matrix.m
- equations.m
- lengths.txt
- Numerical\_IK
- Numerical\_IK.cpp
- Path.txt
- Path2.txt
- plot3d.m

Details

Workspace

Name	Value
a	[0,2,2]
A1	4x4 double
A2	4x4 double
A3	4x4 double
alpha	[1.5708,0,0]
d	[3,0,0]
i	31
l_axis	7
lengths	[3;2;2]

Editor - C:\Users\Marco\Documents\Proyecto MN\PROYECTO\_M2019\_MCMESQ\CODIGO\plot3d.m

```

1 - lengths = readmatrix('lengths.txt');
2 - path = readmatrix('Path.txt');
3 - a = [0,lengths(2),lengths(3)];
4 - alpha = [pi/2,0,0];
5 - d = [3;0;0];
6 -
7 - % Compute DH parameters
8 - % DH parameters are stored in the matrix 'a'
9 - % DH parameters are stored in the matrix 'alpha'
10 - % DH parameters are stored in the matrix 'd'
11 - % DH parameters are stored in the matrix 'i'
12 -
13 - l_axis = sum(a) + sum(d);
14 -
15 - figure(1)
16 - Plot = plot3(path(:,1),path(:,2),path(:,3),'r','LineWidth',3);
17 - axis([-2 4 -4 1 0 6])

```

Command Window

```

>> plot3d
>> plot3d
>> plot3d
fx >>

```

script Ln 7 Col 34

03:04 17/12/2019

# Conclusiones

- En la actualidad prácticamente todos los manipuladores cuentan con un algoritmo de cinemática inversa.
- Se aplica para tener una solución más generalizada.
- Con este método iterativo se obtiene una transición suave entre los puntos.
- El programa presenta los siguientes puntos de mejora:
  - Blindar el programa para puntos fuera del espacio de trabajo.
  - Generalizar para  $n$  eslabones.
  - Mejorar la visualización de resultados

# Fuentes

- Craig, John J. "*Robótica*". PEARSON EDUCATION, México, 2006.
- Kevin M. Lynch and Frank C. Park. "*Modern Robotics: Mechanics, Planning and Control*". 2017
- Lukas Barinka, Roman Berka. "*Inverse kinematics - Basic Methods*". Dept. of Computer science and Engineering, Czech Technical University.