

Planificación

- Introducción a la Inteligencia Artificial.
- Inteligencia Artificial (IA)
- Introducción a la Robótica.
- **Herramientas Matemáticas para la localización espacial y Simulación en Matlab.**
- Dinámica del robot.
- Control Cinemático.
- Control Dinámico.
- Robótica Embedded , Compiladores y Lenguajes de programación.
- Diseño y Aplicaciones de Robots.
- Cobots.
- Impresoras 3D y 4D

Herramientas Matemáticas para la localización espacial

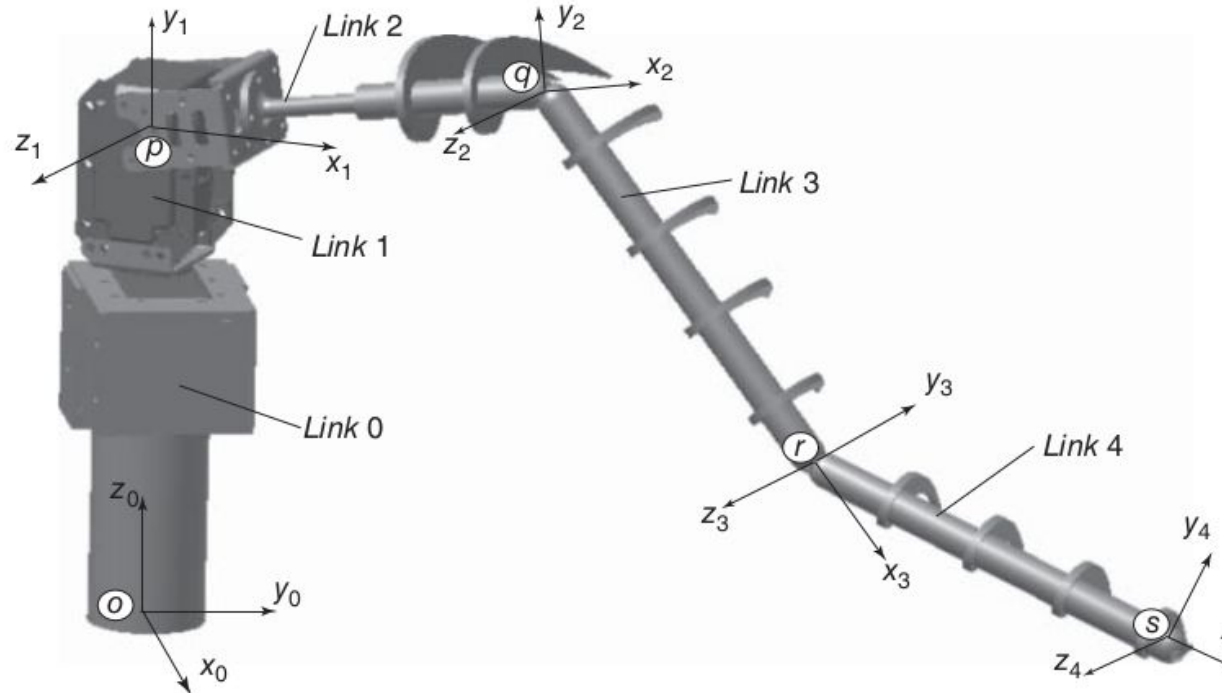
- **Representación de la Posición: Sistema cartesiano y de coordenadas.**
- Representación de la orientación: Matriz de rotación, Ángulos de Euler, Par de rotación, Cuaternios. Matrices de transformación homogéneas: Coordenadas y Matrices homogéneas.
- Simulación en Matlab: Robotic Toolbox, Simulink, SimMechanics y SimPowerSystems.

Representación de la Posición

Sistema de Coordenadas:

Supongamos tener un brazo robótico como el que se muestra a continuación, del cual queremos representar la posición de su extremo o, en caso de tener una pinza, queremos mover a esta. **¿Cómo lo podemos hacer?**

Representación de la Posición



[13]

Representación de la Posición

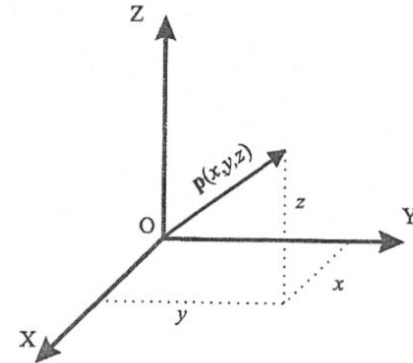
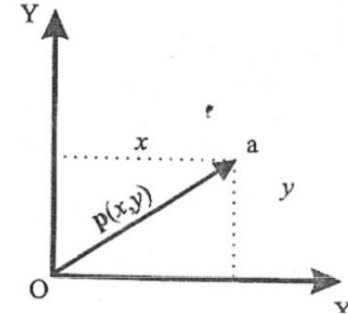
Sistema coordenadas Cartesianas:

- En un sistema de dos dimensiones (x,y)

La pinza se representa mediante el punto **p**,
de coordenadas **p(x,y)**

- En un sistema de tres dimensiones (x,y,z)

La pinza se representa mediante el punto **p**,
de coordenadas **p(x,y,z)**

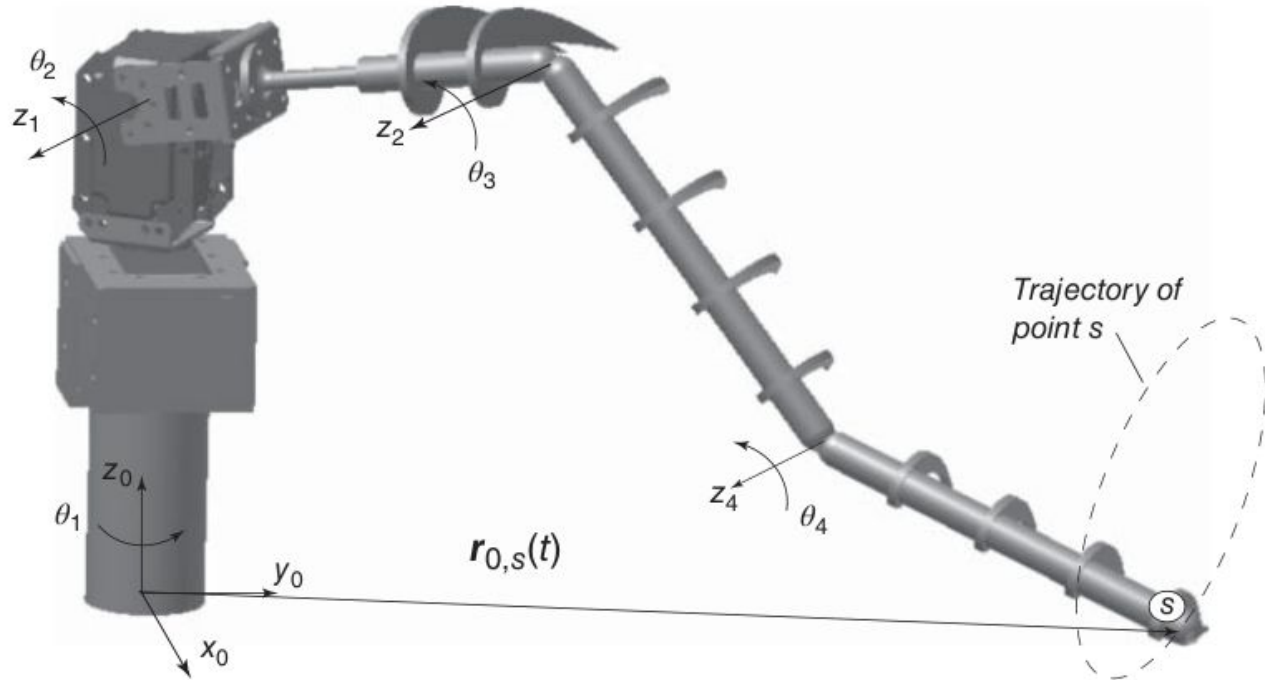


Representación de la Posición

Muchas veces, en vez de usar **coordenadas cartesianas**, resulta útil usar **coordenadas cilíndricas y/o esféricas**.

Veamos el siguiente ejemplo:

Representación de la Posición



[13]

Representación de la Posición

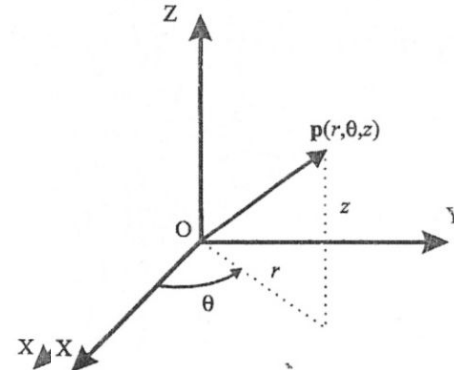
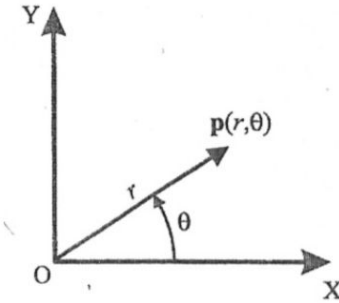
Sistema coordenadas Cilíndricas:

- En un sistema de dos dimensiones (x,y)

La pinza se representa mediante el punto **p**,
de coordenadas **$p(r,\theta)$**

- En un sistema de tres dimensiones (x,y,z)

La pinza se representa mediante el punto **p**,
de coordenadas **$p(r,\theta,z)$**



[2] B

Representación de la Posición

Sistema Cilíndricas:

- Estudiando al punto **p**, se tiene:
- **r** : es la distancia del origen O al extremo del vector p
- **θ** : es el ángulo que forma el vector p con el eje OX.
- **z**: representa la proyección sobre el eje OZ

Al punto **p** se lo puede escribir como **un vector**, de la siguiente forma: $\mathbf{p} = \begin{pmatrix} x \\ y \end{pmatrix}$

Por ejemplo, si el punto p estuviese en la coordenada $x=1, y=1$, sería $\mathbf{p} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$

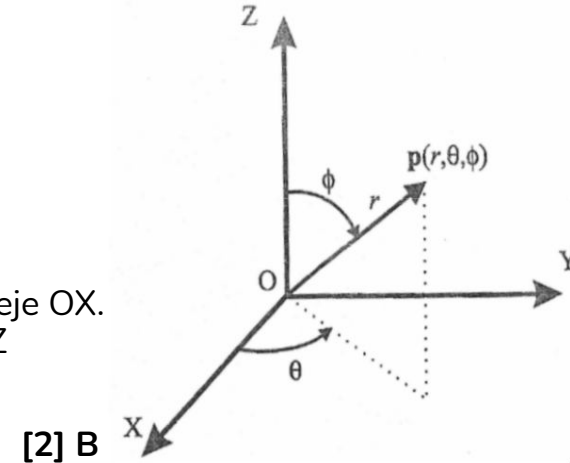
Representación de la Posición

Sistema coordenadas Esféricas:

- En tres dimensiones (x,y,z)

La pinza se representa mediante el punto **p**,
de coordenadas **$p(r, \Theta, \varphi)$** .

- Estudiando al punto **p**, se tiene:
- **r** : es la distancia del origen O al extremo del vector. p
- **Θ** : es el ángulo que forma la proyección del vector p con el eje OX.
- **φ** : representa el ángulo formado entre el vector p y el eje OZ



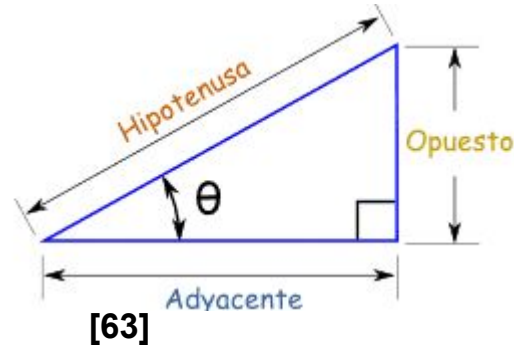
Representación de la Posición

En base a lo comentado anteriormente, podemos repasar algunos aspectos trigonométricos para entender lo que sigue en el curso:

$$\text{sen } \theta = \frac{\text{CAT. OP}}{\text{HIP}}$$

$$\text{cos } \theta = \frac{\text{CAT. ADY}}{\text{HIP}}$$

$$\text{tg } \theta = \frac{\text{CAT. OP}}{\text{CAT. ADY}}$$



Representación de la Posición

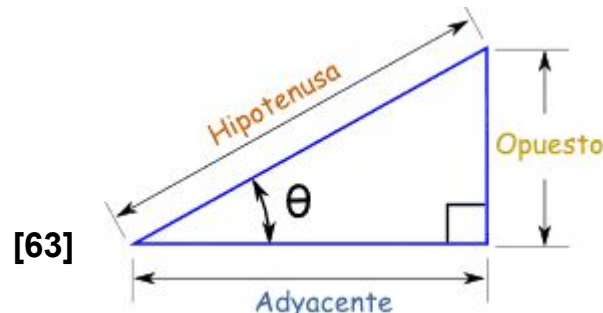
Ejemplo: Si la **Hipotenusa** vale 1m y el **ángulo** vale 45°, ¿cuánto vale el **Cateto Opuesto**?

De las expresiones anteriores, podemos usar que:

$$\text{sen } \theta = \frac{\text{CAT. OP}}{\text{HIP}}$$

Si despejamos el CAT.OP, queda:

$$\text{CAT. OP} = \text{HIP} \cdot \text{sen}(\theta) = 1 \text{ m} \cdot \text{sen}(45^\circ) = 1 \text{ m} \cdot 0,707 = 0,707 \text{ m}$$



Representación de la Posición

Ejemplo: Si la **Hipotenusa** vale 1m y el **ángulo** vale 45°, ¿cuánto vale el **Cateto Adyacente**?

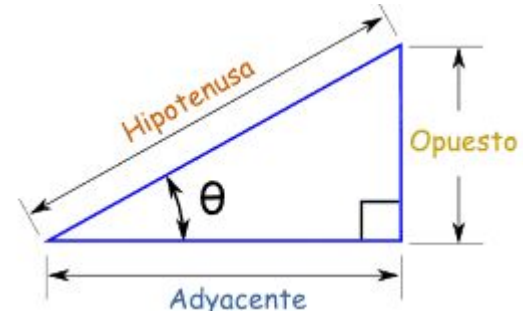
De las expresiones anteriores, podemos usar que:

$$\cos \theta = \frac{\text{CAT. ADY}}{\text{HIP}}$$

Si despejamos el CAT.ADY, queda:

[63]

$$\text{CAT. ADY} = \text{HIP} \cdot \cos(\theta) = 1\text{m} \cdot \cos(45^\circ) = 1\text{m} \cdot 0,707 = 0,707\text{m}$$



Representación de la Posición

Ejemplo: Si la **Hipotenusa** vale 2m y el **CAT.ADY** 1m, ¿cuánto vale el **ángulo**?

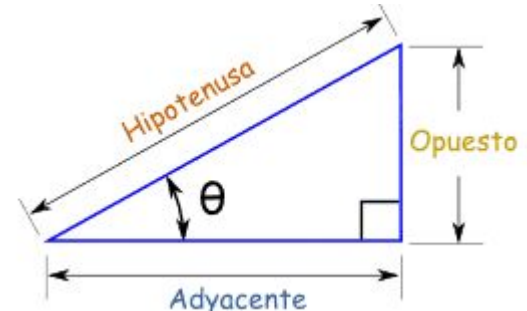
De las expresiones anteriores, podemos usar que:

$$\cos \theta = \frac{\text{CAT. ADY}}{\text{HIP}}$$

Si despejamos el ángulo, queda:

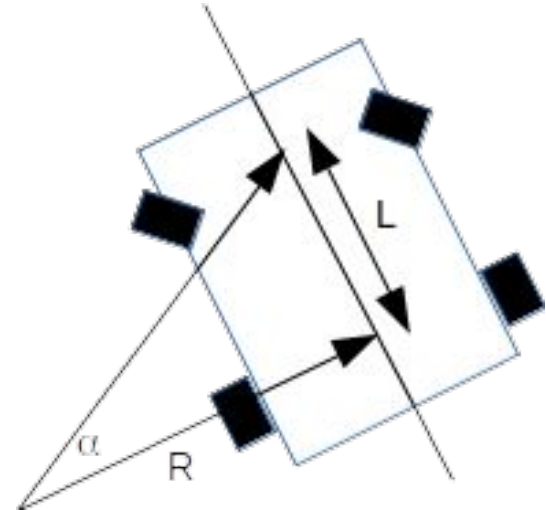
$$\theta = \arccos\left(\frac{1\text{m}}{2\text{m}}\right) = \arccos(0.5) = 30^\circ$$

[63]



Actividad

Se tiene un móvil como el que se muestra, el cual se encuentra girando mientras avanza. Sabiendo que el mismo gira con un **ángulo constante** de $\alpha = 45^\circ$ y que $L = 10\text{cm}$, calcular el radio de curvatura **R**.



Actividad

Aplicando las consideraciones vistas, tenemos que:

$$\operatorname{tg} \theta = \frac{\operatorname{CAT} . \operatorname{OP}}{\operatorname{CAT} . \operatorname{ADY}}$$

$$\frac{L}{R} = \tan \alpha$$

Con lo cual, si queremos despejar el radio de curvatura **R**, tenemos:

$$R = \frac{L}{\tan \alpha}$$

Actividad

Entonces, si:

$$R = \frac{L}{\tan \alpha}$$

y $L = 10$ cm y $\alpha = 45^\circ$, nos queda que el radio de curvatura vale:

$$R = 10 \text{ cm}$$

Herramientas Matemáticas para la localización espacial

- Representación de la Posición: Sistema cartesiano y de coordenadas.
- **Representación de la orientación: Matriz de rotación, Ángulos de Euler, Par de rotación, Cuaternios. Matrices de transformación homogéneas: Coordenadas y Matrices homogéneas.**
- Simulación en Matlab: Robotic Toolbox, Simulink, SimMechanic y SimPowerSystems.

Herramientas Matemáticas para la localización espacial

Repaso de Matrices:

Podemos llamar Matriz a un conjunto de números dispuestos en forma de filas y de columnas, como por ejemplo:

$$A = \begin{pmatrix} 4 & -14 & 6 \\ 2 & 0,3 & 2 \\ 9 & 38 & -7 \end{pmatrix}$$

El **número de filas y columnas de una matriz se denomina dimensión** de una matriz y a cada uno de los elementos de una matriz se lo llama **elemento**

Herramientas Matemáticas para la localización espacial

Operaciones con Matrices:

Si bien las operaciones resultan ser las básicas, nos concentramos en la del **producto**.

Producto de Matriz por vector

$$A = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}, \quad B = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} \quad A \times B = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \times \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} = \begin{pmatrix} a_{11}b_1 + a_{12}b_2 \\ a_{21}b_1 + a_{22}b_2 \end{pmatrix}$$

Herramientas Matemáticas para la localización espacial

Ejemplo:

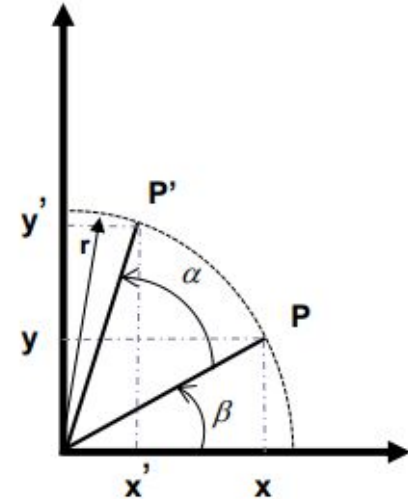
Tomamos una matriz de 2x2 y la multiplicamos por un vector de 2x1. Obtenemos otro vector:

$$v = A \cdot w = \begin{pmatrix} 2 & 3 \\ 4 & 1 \end{pmatrix} \cdot \begin{pmatrix} 5 \\ 1 \end{pmatrix} = \begin{pmatrix} 2.5 + 3.1 \\ 4.5 + 1.1 \end{pmatrix} = \begin{pmatrix} 13 \\ 21 \end{pmatrix} \quad w = \begin{pmatrix} 13 \\ 21 \end{pmatrix}$$

Representación de la orientación

Matriz de rotación:

Supongamos que queremos **rotar** el punto **P** hacia el punto **P'**. Es decir, si el punto **P** tiene un ángulo β , queremos llevar el mismo vector a una posición cuyo ángulo sea α más β (con respecto al eje **x**)

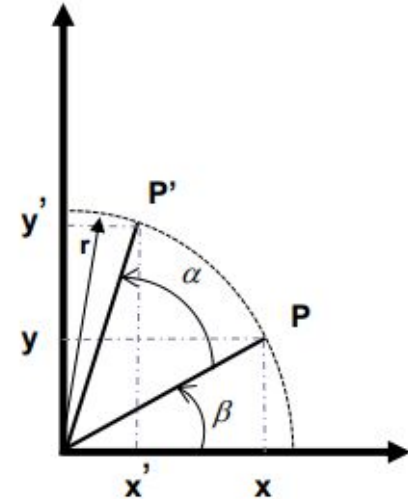


Representación de la orientación

La situación anterior la podemos describir mediante

la matriz de rotación, la cual viene dada por:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos \alpha & -\sin \alpha \\ \sin \alpha & \cos \alpha \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$



Representación de la orientación

En tres dimensiones la matriz de rotación queda:

$$R(z, \theta) = \begin{bmatrix} \cos(\theta) & -\text{sen}(\theta) & 0 \\ \text{sen}(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

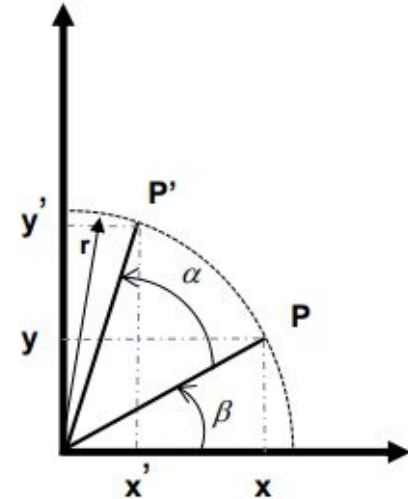
Representación de la orientación

Ejemplo:

Si la pinza se ubica en la coordenada $p(x=1, y=1)$

y se la rota en un ángulo de 90° , calcular la coordenada

donde se ubicará.

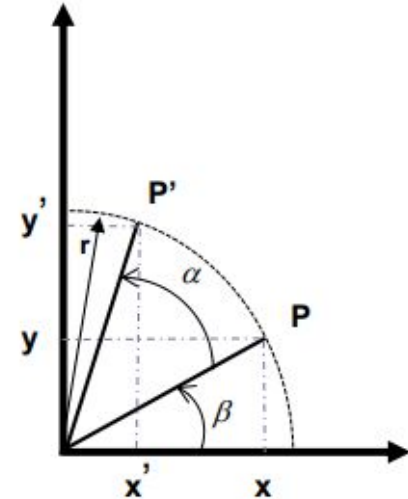


Representación de la orientación

Solución:

Si el ángulo de rotación es de 90° , la matriz queda:

$$R = \begin{pmatrix} \cos 90^\circ & -\sin 90^\circ \\ \sin 90^\circ & \cos 90^\circ \end{pmatrix} = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$$



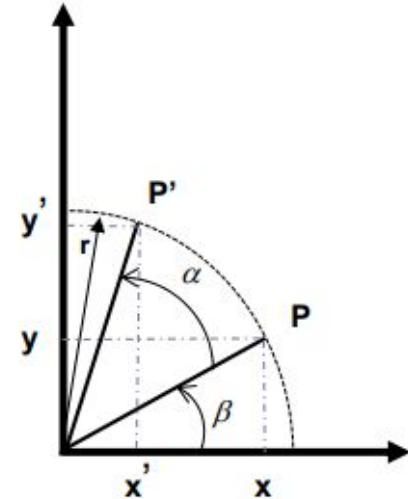
Representación de la orientación

Solución:

Luego, la nueva posición de la pinza queda en:

$$\begin{pmatrix} -1 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

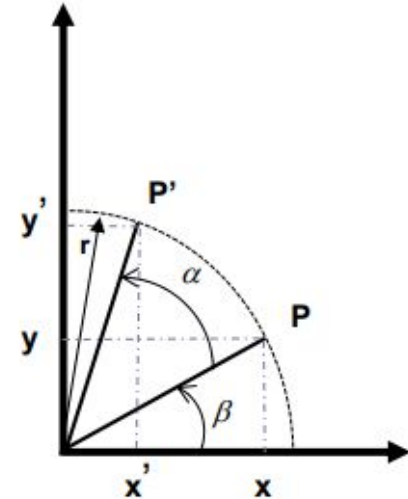
Es decir, la nueva posición será en $v = -1, 1$



Actividad

Problema:

Si la pinza se ubica en la coordenada $p(x=0, y=1)$
y se la rota en un ángulo de 180° , calcular la coordenada
donde se ubicará.



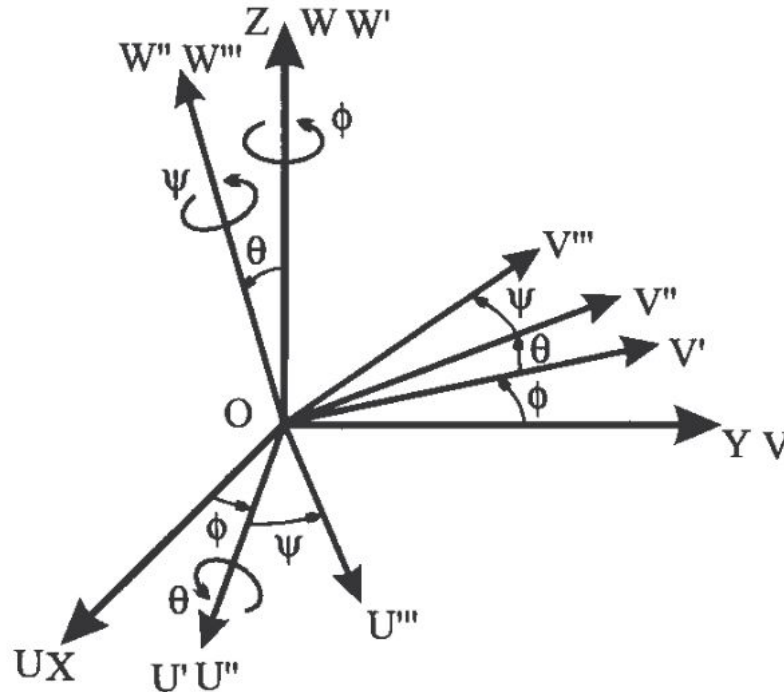
Representación de la orientación

Ángulos de Euler:

Si bien la matriz de Rotación presenta ventajas para la *manipulación matemática*, existen otros métodos matemáticos para la *orientación*, los cuales hacen uso de tres componentes. Estos son los llamados, “*Ángulos de Euler*”. [1] B

Estos ángulos suelen representarse mediante: Φ (Fi), θ (Theta), ψ (Psi)

Representación de la orientación



Representación de la orientación

Si se parte de los sistemas **OXYZ** y **OUVW**, *inicialmente coincidentes*, se puede colocar al sistema **OUVW** en *cualquier orientación*, mediante los siguientes pasos:

1) Girar el sistema **OUVW** un ángulo Φ

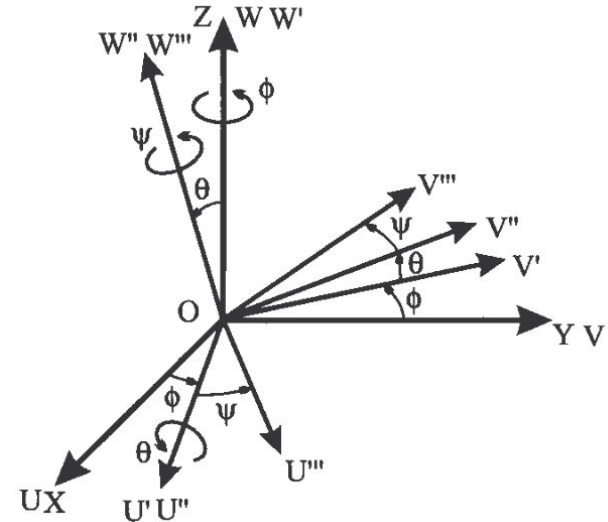
con respecto al **eje OZ**, convirtiéndose así en el **OU'V'W'**

2) Girar el sistema **OU'V'W'** un ángulo θ

con respecto al **eje OU'**, convirtiéndose así en el **OU''V''W''**

3) Girar el sistema **OU''V''W''** un ángulo ψ

con respecto al **eje OW''**, convirtiéndose así en el **OU'''V'''W'''**



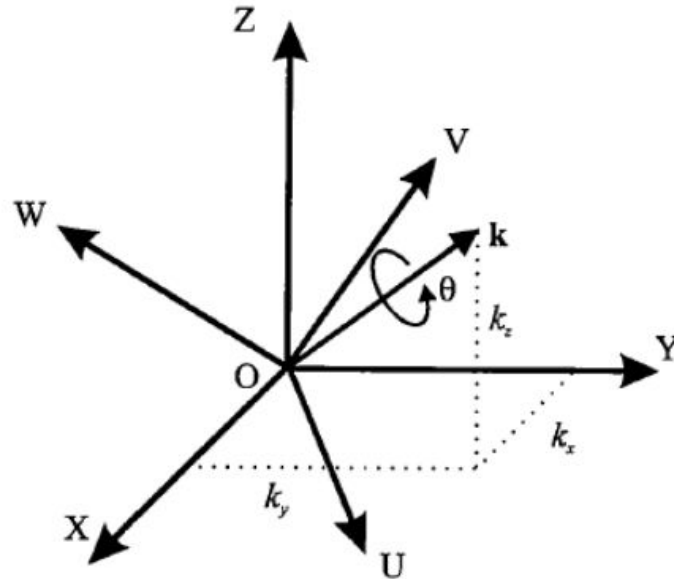
Representación de la orientación

Par de rotación:

La representación de la orientación de un sistema **OUVW** con respecto al sistema de referencia **OXYZ** también puede realizarse mediante la definición de un vector **k** (k_x , k_y , k_z) y un ángulo de giro **θ** , tal que el sistema OUVW corresponde al sistema OXYZ **girado un ángulo θ sobre el eje k**. El eje **k** ha de pasar por el origen O de ambos sistemas. **Al par (k , θ) se le denomina par de rotación.** [3] B

Representación de la orientación

Par de rotación:



Representación de la orientación

Cuaternios:

Los **cuaternios**, se utilizan como herramienta matemática de gran versatilidad computacional para trabajar con giros y orientaciones.

Un cuaternio **Q** está constituido por cuatro componentes (**q0, q1, q2, q3**) que representan las coordenadas en una base {e, i, j, k}. Es frecuente denominar **parte escalar** del cuaternio a la componente en e: **q0**, y **parte vectorial** al resto de componentes. De modo que un cuaternio se puede expresar como:

$$Q = [q_0, q_1, q_2, q_3] = [s, \mathbf{v}]$$

Donde **s** representa la **parte escalar** y **v** la **parte vectorial**. [3]

Representación de la orientación

Podemos decir que un **escalar** es un número real o compleja que puede describir un fenómeno físico.

En el caso del Álgebra de vectores, si multiplicamos al vector **v** por un escalar **$\lambda = 2$**

$$v = \begin{pmatrix} 5 \\ 1 \end{pmatrix} \quad \lambda \cdot v = 2 \cdot \begin{pmatrix} 5 \\ 1 \end{pmatrix} = \begin{pmatrix} 10 \\ 2 \end{pmatrix}$$

Representación de la orientación

Matrices de transformación homogéneas:

Hasta ahora vimos distintos métodos de representar la posición y/o la orientación de la pinza (sólido), en el espacio. Pero no tenemos una herramienta que nos permita representar, de manera conjunta, la posición y la orientación. Las **matrices de transformación homogénea** permiten esta representación conjunta. [3]

Representación de la orientación

Matrices de transformación homogéneas:

Se define una matriz **T** de dimensión **4x4** que representa la transformación de un vector de coordenadas homogéneas de un sistema de coordenadas, a otro. [3] B

$$\mathbf{T} = \begin{bmatrix} \mathbf{R}_{3 \times 3} & \mathbf{p}_{3 \times 1} \\ \mathbf{f}_{1 \times 3} & \mathbf{w}_{1 \times 1} \end{bmatrix} = \begin{bmatrix} \text{Rotación} & \text{Traslación} \\ \text{Perspectiva} & \text{Escalado} \end{bmatrix}$$

Vector bidimensional en coordenadas homogéneas: $(\lambda x, \lambda y, \lambda)$

Ej: si $\lambda=1$, $x=2$, $y=3$, entonces $(2,3,1)$

Representación de la orientación

Matrices de transformación homogéneas:

Como se puede ver, la matriz homogénea se compone de cuatro submatrices: **una submatriz R 3×3 que corresponde a una matriz de rotación; una submatriz p 3×1 que corresponde al vector de traslación; una submatriz f 1×3 que representa una transformación de perspectiva, y una submatriz w 1×1 que representa un escalado global.** [3] B

En robótica generalmente sólo interesa conocer el valor de la **matriz de rotación y del vector de traslación**, considerándose la transformación de perspectiva nula y el escalado global unitario

$$\mathbf{T} = \begin{bmatrix} \mathbf{R}_{3 \times 3} & \mathbf{p}_{3 \times 1} \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} \text{Rotación} & \text{Traslación} \\ 0 & 1 \end{bmatrix}$$

Herramientas Matemáticas para la localización espacial

- Representación de la Posición: Sistema cartesiano y de coordenadas.
- Representación de la orientación: Matriz de rotación, Ángulos de Euler, Par de rotación, Cuaternios. Matrices de transformación homogéneas: Coordenadas y Matrices homogéneas.
- **Simulación en Matlab: Robotic Toolbox, Simulink, SimMechanic y SimPowerSystems.**

Simulación en Matlab

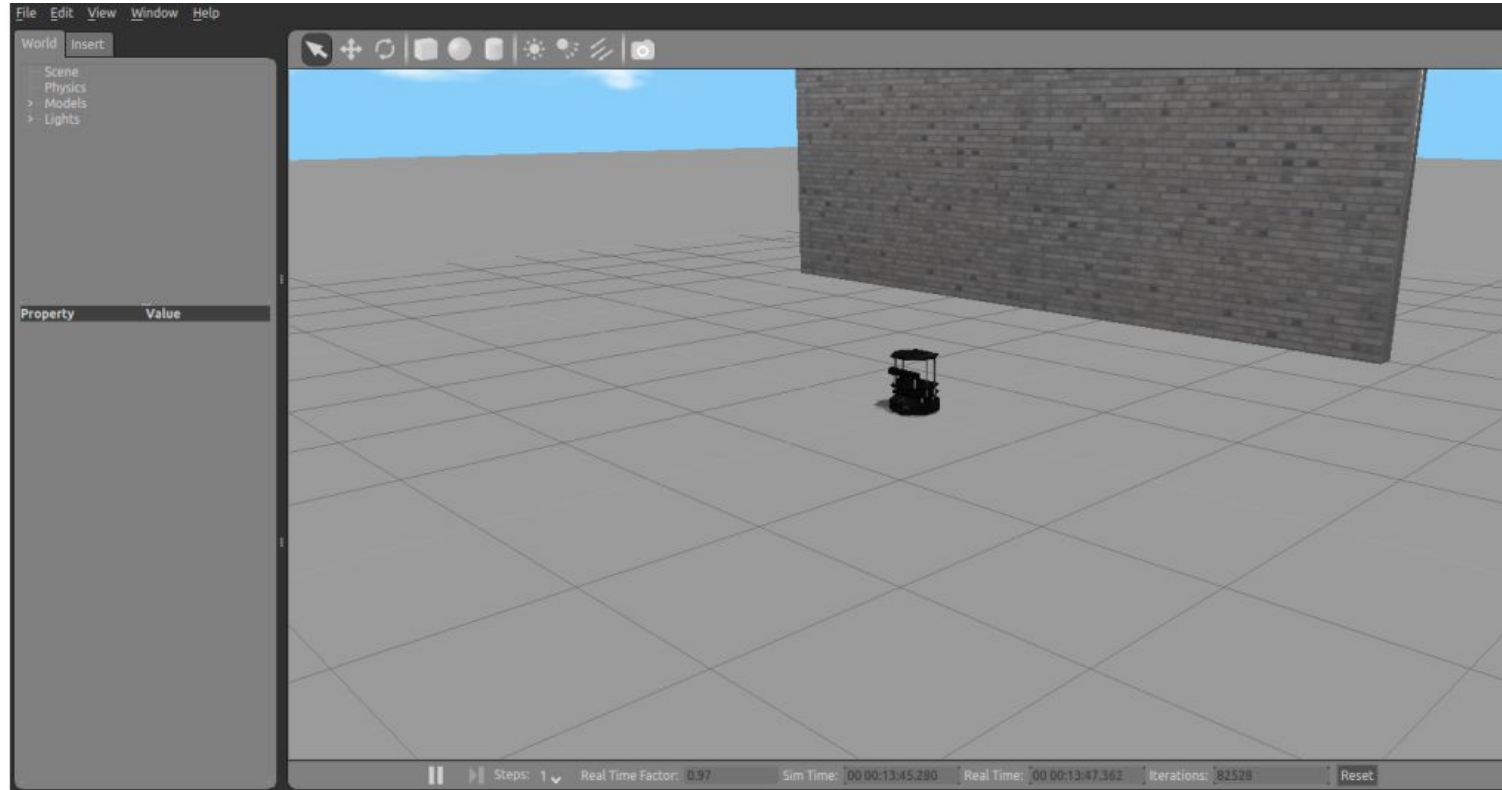
Robotic Toolbox:

Robotics System Toolbox™ proporciona *herramientas y algoritmos para diseñar, simular y probar manipuladores, robots móviles y robots humanoides*.

Para los *manipuladores y los robots humanoides*, esta toolbox incluye *algoritmos para comprobación de colisiones, generación de trayectorias, cinemática de avance e inversa*.

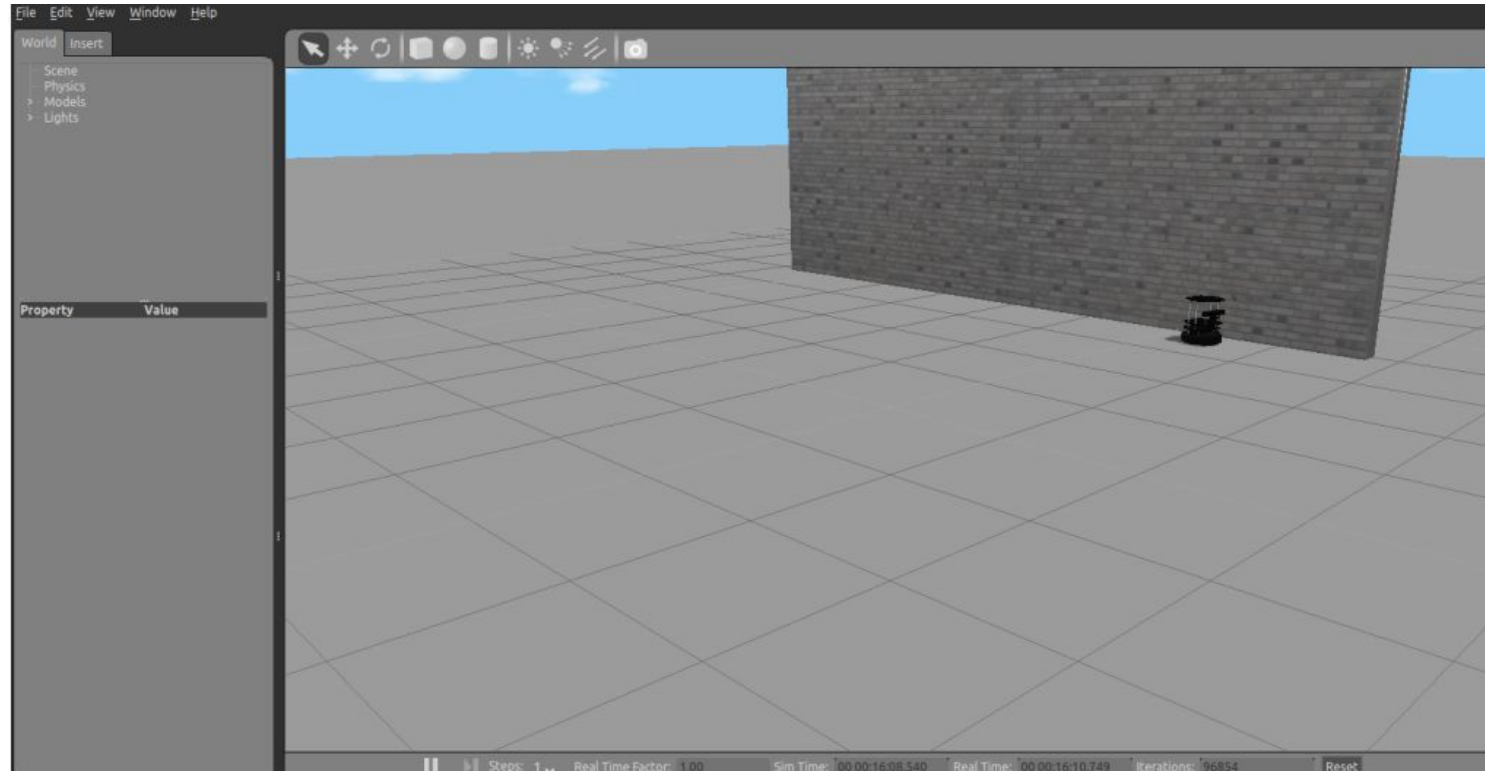
En el caso de los *robots móviles*, incluye *algoritmos para mapeo, localización, planificación de trayectorias, seguimiento de trayectorias y control de movimiento*. La toolbox ofrece ejemplos de referencia de aplicaciones robóticas industriales habituales. [35]

Simulación en Matlab

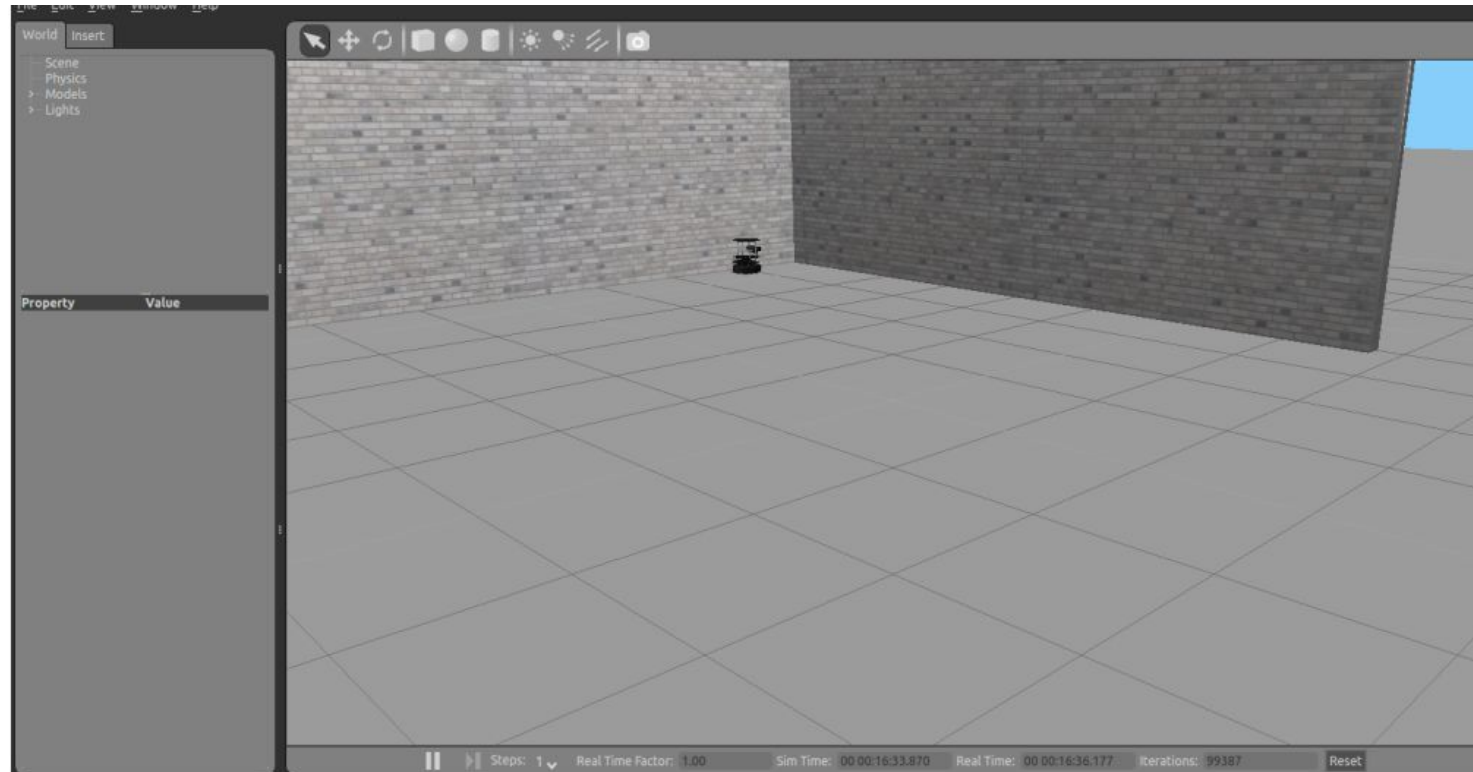


[36]

Simulación en Matlab



Simulación en Matlab



[36]

Simulación en Matlab

SimMechanic:

Simscape Multibody™ (*anteriormente SimMechanics™*) *proporciona un entorno de simulación multicuerpo para sistemas mecánicos 3D*, como robots, suspensiones de vehículos, maquinaria de construcción y trenes de aterrizaje de aeronaves.

Puede modelar sistemas multicuerpo utilizando bloques que representan cuerpos, articulaciones, restricciones, elementos de fuerza y sensores. Simscape

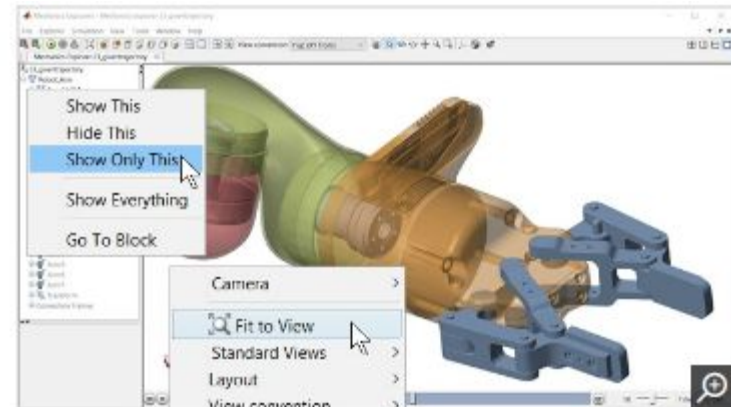
Multibody formula y resuelve las ecuaciones de movimiento de todo el sistema mecánico. [37]

Simulación en Matlab

Explore su mecanismo en una interfaz 3D y navegue hasta la vista esquemática para verificar la estructura del modelo y examinar los resultados representados. Defina puntos de vista estáticos o en movimiento para ver los resultados de la simulación desde un marco de referencia personalizado.

Exploración del comportamiento del mecanismo, la definición del montaje y los resultados de la simulación.

[37]

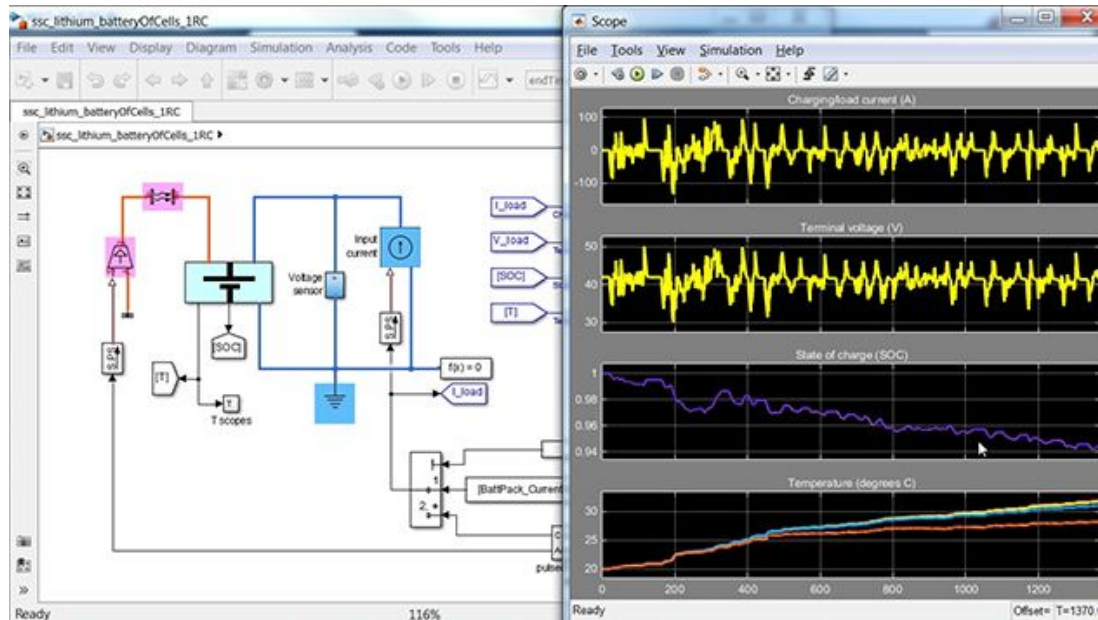


Simulación en Matlab

Simulink:

Los algoritmos de navegación son críticos para las aplicaciones de robots móviles, tales como la cobertura máxima del espacio en robots de limpieza en el hogar, entrega de artículos en robots logísticos de almacenes y hospitales, por nombrar dos de las más comunes. [38]

Simulación en Matlab



[38]

Simulación en Matlab

SimPowerSystems:

Simscape Electrical TM (anteriormente SimPowerSystems TM y SimElectronics®) *proporciona bibliotecas de componentes para modelar y simular sistemas electrónicos, mecatrónicos y de energía eléctrica.*

Incluye modelos de semiconductores, motores y componentes para aplicaciones como accionamiento electromecánico, redes inteligentes y sistemas de energía renovable.

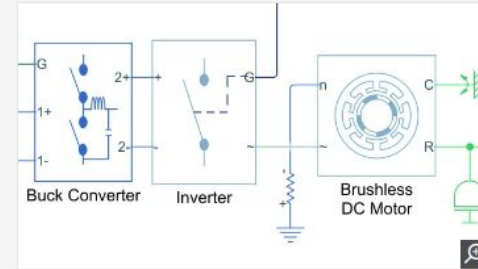
Puede usar estos componentes para evaluar arquitecturas de circuitos analógicos, desarrollar sistemas mecatrónicos con accionamientos eléctricos y analizar la generación, conversión, transmisión y consumo de energía eléctrica a nivel de red. **[39]**

Simulación en Matlab

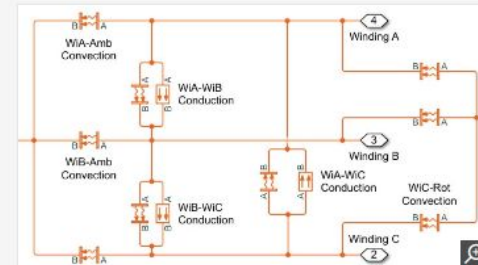
Seleccione modelos simples para que coincidan con el comportamiento en estado estable y logren velocidades de simulación más rápidas.

Especifique cómo cambia el comportamiento del actuador con la temperatura.

[39]



BLDC speed control.



Linear voltage regulator with thermal effects.

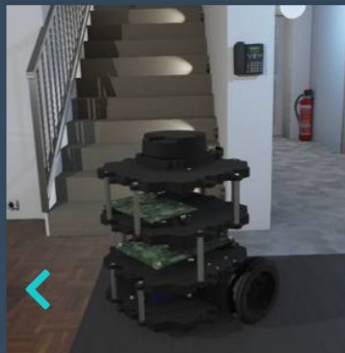
Simulación en Webots

Webots: **Webots** es una aplicación de **código abierto** y **multiplataforma** que se utiliza para **simular robots**. Proporciona un **entorno de desarrollo** completo para **modelar, programar y simular robots**.

Ha sido diseñado para un uso profesional y se usa ampliamente en la industria, la educación y la investigación. [1U]

Página web: <https://cyberbotics.com/>

Simulación en Webots



Interface simulated robots with your favorite robotics framework such as ROS.

Robotis TurtleBot3 Burger



Simulate assembly lines to prototype your production plant.

Universal Robots UR10e



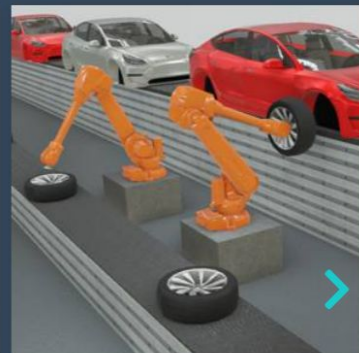
Simulate complex physics such as mechanism wheels or tracks simply from a high level of abstraction.

KUKA youBot



Simulate legged robots and train walking gaits.

ROBOTIS OP3

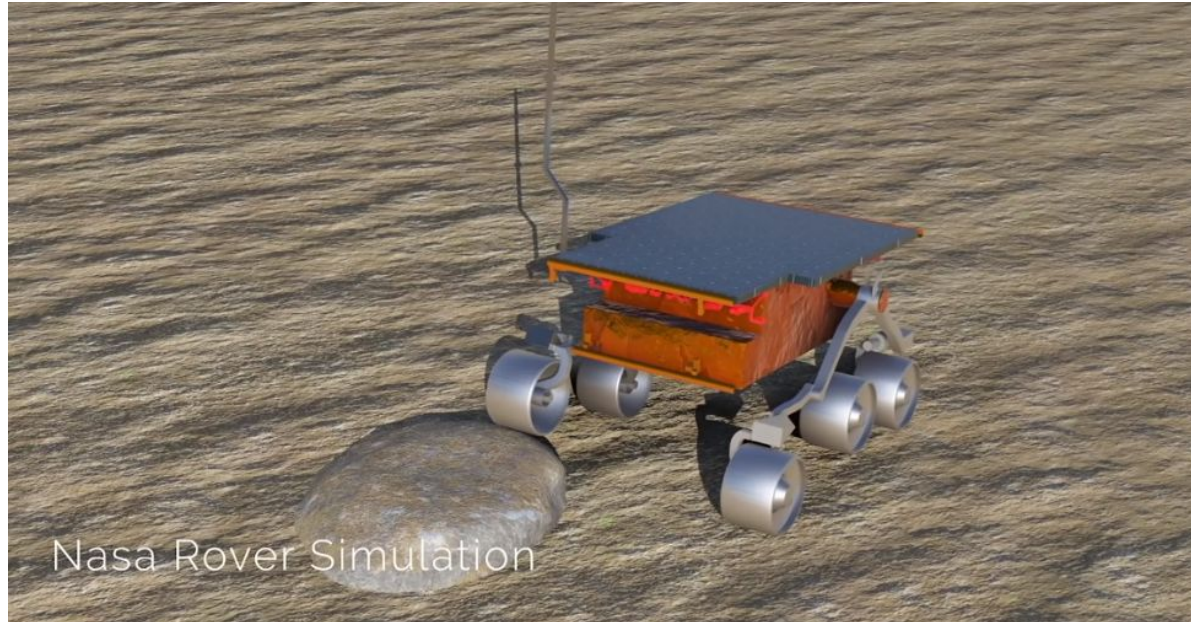


Simulate industrial arms, grippers and tools in factory environments.

ABB IRB 4600/40

<https://cyberbotics.com/>

Simulación en Webots



Simulación en Webots

Algunas características:

- ❖ **Prototipado rápido:** Simulaciones usando robots, sensores, actuadores
- ❖ **Aplicaciones:** Simulaciones con robots industriales, brazos robóticos, piernas robóticas, etc.
- ❖ **Corre en distintos Sistemas Operativos:** Linux, Mac y Windows
- ❖ **Se puede programar en:** C, C++, Python, Java, Matlab y ROS.

Simulación en Webots

¿Quiénes usan Webots?

They use Webots:

Industry



HONDA



HUAWEI



JOHN DEERE



Academia



ETH zürich



HARVARD
UNIVERSITY



Imperial College
London

<https://cyberbotics.com/>

Planificación

- Introducción a la Inteligencia Artificial.
- Inteligencia Artificial (IA)
- Introducción a la Robótica.
- Herramientas Matemáticas para la localización espacial y Simulación en Matlab.
- ***Dinámica del robot.***
- Control Cinemático.
- Control Dinámico.
- Robótica Embedded , Compiladores y Lenguajes de programación.
- Diseño y Aplicaciones de Robots.
- Cobots.
- Impresoras 3D y 4D

Dinámica del robot

- **Modelo dinámico de la estructura mecánica de un robot rígido. Modelo dinámico usando Newton-Euler.**
- Modelo dinámico usando Lagrange-Euler.
- Modelo dinámico usando variables de estado.

Modelo dinámico de la estructura mecánica de un robot rígido

La **dinámica** se ocupa de la **relación entre las fuerzas que actúan en un cuerpo y el movimiento** que las origina. Por esta razón, el **modelo dinámico** de un robot tiene por objetivo **conocer el movimiento y las fuerzas** implicadas en el mismo. [1]

La obtención del modelo dinámico de un mecanismo se basa en el planteo del equilibrio de fuerzas establecido en la **segunda ley de Newton**, es decir:

$$\sum F = m \cdot a$$

$$\sum T = I \cdot \gamma$$

Modelo dinámico de la estructura mecánica de un robot rígido

Veamos un **Ejemplo** de cómo se aplican las leyes de Newton. Supongamos que una grúa (puede ser un **Brazo Robótico**) tiene que levantar una **container**, de la siguiente forma:



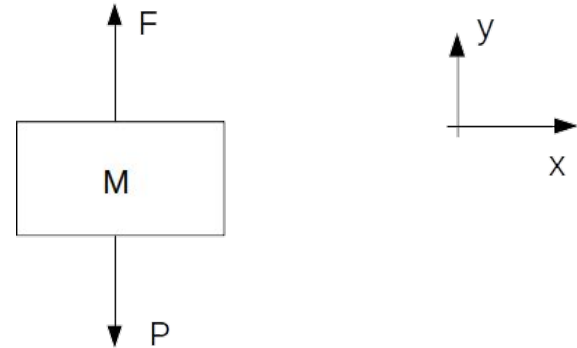
Modelo dinámico de la estructura mecánica de un robot rígido

Lo que se hace habitualmente es plantear el **diagrama de cuerpo libre**. En este caso, sería el del **container**.

F: Fuerza que hace la grúa/brazo robótico.

M: Masa del container

P: Peso del container



Estudiando el cuerpo, tendríamos que se mueve en el eje **"y" (vertical)** ya que puede subir o bajar mediante la grúa. En el eje **"x" (horizontal)**, no tendría movimiento

Modelo dinámico de la estructura mecánica de un robot rígido

Planteando las ecuaciones, podríamos pensar lo siguiente:

$$F - P = M \cdot a$$

F: Fuerza que hace la grúa/brazo robótico.

M: Masa del container

P: Peso del container ($P = M \cdot g$)

a: Aceleración

Modelo dinámico de la estructura mecánica de un robot rígido

Antes de continuar, tengamos en cuenta que:

$x(t)$ = trayectoria

$v(t)$ = velocidad

$$v(t) = \frac{dx(t)}{dt}$$

$a(t)$ = aceleración

$$a = \frac{d^2 x(t)}{dt^2}$$

Modelo dinámico de la estructura mecánica de un robot rígido

¿Qué es una *derivada*?



:

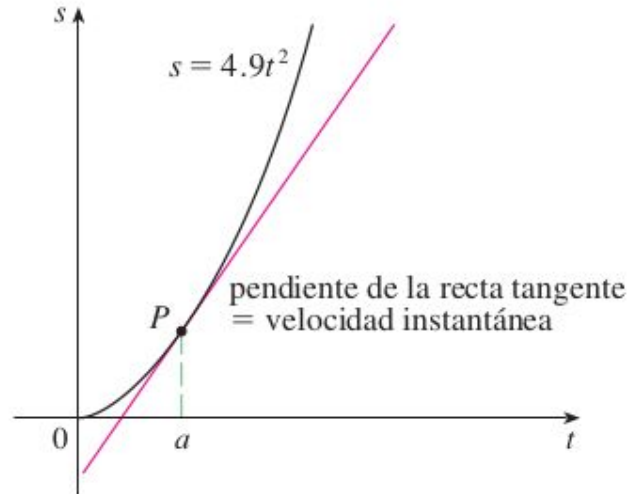
Modelo dinámico de la estructura mecánica de un robot rígido

Para esto, veamos el siguiente ejemplo:

Supongamos que vamos en auto y mientras viajamos, vamos mirando el velocímetro. Al ver esto durante todo el trayecto, se ve que **la velocidad va cambiando** de acuerdo al tráfico que exista, las disminuciones y aumentos de velocidad. Es decir, **la velocidad no es constante** o de otra forma, tenemos una **velocidad determinada en cada instante**, lo que se puede llamar, una “velocidad instantánea”.

Modelo dinámico de la estructura mecánica de un robot rígido

En base a esto, trazamos la **trayectoria recorrida** (curva color negro) y la recta **tangente** a esta curva en el punto P



Modelo dinámico de la estructura mecánica de un robot rígido

La noción de derivada se refiere a la *pendiente de la recta tangente en el punto P*. Este punto pasa por el *instante $t = a$* , con lo cual, con la *derivada*, podremos saber la *velocidad instantánea* (en ese momento).

$$v(t) = \frac{dx(t)}{dt}$$

A partir de la expresión anterior, si volvemos a derivar, podemos obtener la aceleración:

$$a = \frac{d^2 x(t)}{dt^2}$$

Modelo dinámico de la estructura mecánica de un robot rígido

Dado que el container, cuando sea movido, va a recorrer una **trayectoria hacia arriba o hacia abajo**, a esta la podemos denominar **$x(t)$** . Por esto, la aceleración vendrá dada por:

$$a = \frac{d^2 x(t)}{dt^2}$$

Entonces, en base a la notación anterior, a la ecuación obtenida a partir del diagrama de cuerpo libre la podemos escribir como:

$$F - P = M \cdot a \qquad F - P = M \cdot \frac{d^2 x(t)}{dt^2}$$

Modelo dinámico de la estructura mecánica de un robot rígido

La ecuación:

$$F - P = M \cdot \frac{d^2 x(t)}{dt^2}$$

Se denomina: **Ecuación Diferencial de segundo orden.**

Modelo dinámico de la estructura mecánica de un robot rígido

¿Qué es una **Ecuación Diferencial**?



Modelo dinámico de la estructura mecánica de un robot rígido

Podemos decir que una **ecuación diferencial** es una **ecuación** que **contiene derivadas** respecto de una variable independiente. En nuestro caso, **la variable independiente** es el **tiempo “t”**.

Las **ecuaciones diferenciales** tienen una gran aplicación dado que nos permite **modelar a muchos sistemas físicos**, por ejemplo, **la propagación de una enfermedad**, **el crecimiento de una población** y, en nuestro caso particular de la **Robótica**, nos puede ayudar a **describir cómo se mueve un brazo robótico**.

Modelo dinámico de la estructura mecánica de un robot rígido

Para el caso planteado

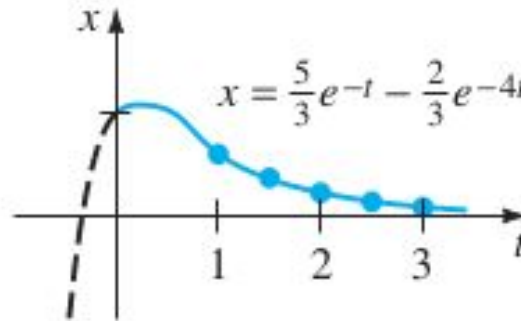
$$F - P = M \cdot \frac{d^2 x(t)}{dt^2}$$

El **objetivo** es encontrar “cuánto vale $x(t)$ ”, que **no será un valor sino una función** que describe cómo se mueve el container cuando se lo sube o baja. A muy grandes rasgos, sería algo similar a despejar la “ x ”, en un caso algebraico.

Para resolver esta ecuación, hay varios métodos de Resolución de Ecuaciones Diferenciales.

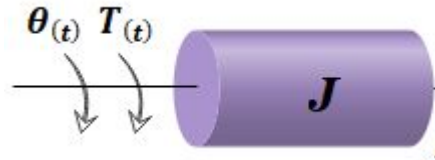
Modelo dinámico de la estructura mecánica de un robot rígido

Al aplicar estos métodos de resolución, se llega a la expresión de la función $x(t)$ y la podemos graficar. **Un posible ejemplo** de lo que puede dar $x(t)$ es lo siguiente:



Modelo dinámico de la estructura mecánica de un robot rígido

Lo comentado hasta aquí se refiere a **sistemas traslacionales** pero, esto mismo se puede aplicar a **sistemas rotacionales**, es decir, sistemas donde podemos ver movimientos que rotan sobre un eje. Por ejemplo, vemos el siguiente caso:



[62U]

En vez de **x(t)** tenemos **θ(t)** , luego en vez de la **F** tenemos a **T** (torque) y en vez de **M** tenemos a **J**. Esta última se llama Momento de inercia.

Modelo dinámico de la estructura mecánica de un robot rígido

Con esto entonces podemos plantear la ecuación

$$\sum T = J \cdot \gamma \qquad \sum T = J \cdot \frac{d^2 \theta(t)}{dt^2}$$

Siendo:

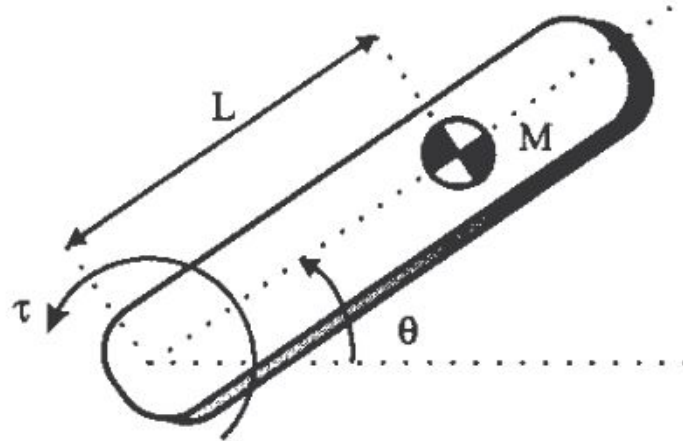
T = Torque/s

J = Momento de Inercia.

γ = Aceleración angular

Modelo dinámico de la estructura mecánica de un robot rígido

En base a todo esto, estudiemos el siguiente brazo robótico:



Modelo dinámico de la estructura mecánica de un robot rígido

La ecuación que modela a la figura anterior, teniendo en cuenta la segunda ley de Newton, la misma queda:

$$T - M \cdot g \cdot L \cdot \cos(\theta) = J \cdot \frac{d^2 \theta(t)}{dt^2}$$

Si despejamos el torque, nos queda:

$$T = J \cdot \frac{d^2 \theta(t)}{dt^2} + M \cdot g \cdot L \cdot \cos(\theta)$$

Modelo dinámico de la estructura mecánica de un robot rígido

El término J lo podemos reemplazar por $M.L^2$ dada la forma del brazo.

$$T = M.L^2 \cdot \frac{d^2 \theta(t)}{dt^2} + M.g.L.\cos(\theta)$$

$$T = M.L^2 \cdot \ddot{\theta} + M.g.L.\cos(\theta)$$

De esta manera, conociendo el torque ejercido (τ), la masa (M) y la longitud del brazo (L), se puede conocer $\theta(t)$, su primera y segunda derivada (velocidad y aceleración).

De forma inversa, si se quisiera que $\theta(t)$ evolucione según una determinada forma (función), podría obtenerse el valor del torque necesario (τ)

Modelo dinámico de la estructura mecánica de un robot rígido

De esta forma podemos definir lo siguiente:

- a) **Modelo dinámico directo:** expresa la *evolución temporal de las coordenadas articulares del robot, en función de las fuerzas y pares* que intervienen.

- b) **Modelo dinámico inverso:** expresa *las fuerzas y pares que intervienen en función de la evolución de las coordenadas articulares* y sus derivadas.

Modelo dinámico de la estructura mecánica de un robot rígido

En el caso de plantear los modelos descriptos para *robots que tienen una mayor cantidad de grados de libertad* (5 o 6), *los mismos comienzan a tener ciertas complicaciones* dado que aparecen *fuerzas de Coriolis* debidas al movimiento relativo existente entre los diversos elementos.

Por esta razón se pueden utilizar la *formulación Lagrangiana*, la cual resulta *sistemática y facilita la formulación de un modelo más complejo*.

Dinámica del robot

- Modelo dinámico de la estructura mecánica de un robot rígido. Modelo dinámico usando Newton-Euler.
- **Modelo dinámico usando Lagrange-Euler.**
- Modelo dinámico usando variables de estado.

Modelo dinámico usando Lagrange-Euler

La **formulación Lagrangiana** tiene en cuenta **las energías cinéticas y potenciales** puestas en juego en el brazo robótico.

$$\mathcal{L}(q, \dot{q}) = \mathcal{K}(q, \dot{q}) - \mathcal{P}(q).$$

Se tiene en cuenta la **Energía Cinética** de todo el sistema (K) y se le resta la **Energía Potencial** (P).

Modelo dinámico usando Lagrange-Euler

¿Qué son *las energías cinéticas y potenciales*?



:

Modelo dinámico usando Lagrange-Euler

La **energía cinética** está relacionada con el **movimiento de un cuerpo**. En el caso de que este sea del tipo traslacional, la energía se puede calcular como:

$$\mathcal{K}(x, \dot{x}) = \frac{1}{2} m \dot{x}^2$$

Siendo:

m: masa del cuerpo

x': velocidad del cuerpo

Modelo dinámico usando Lagrange-Euler

La **energía potencial** está relacionada con la altura a la que se encuentra un cuerpo. Se puede calcular como:

$$\mathcal{P}(x) = mgx.$$

Siendo:

m: masa del cuerpo

g: gravedad

x: velocidad del cuerpo.

La energía potencial también está relacionada con **movimientos elásticos**.

Modelo dinámico usando Lagrange-Euler

Con esto, el **Lagrangiano** queda:

$$\mathcal{L}(x, \dot{x}) = \mathcal{K}(x, \dot{x}) - \mathcal{P}(x) = \frac{1}{2}m\dot{x}^2 - mgx.$$

Modelo dinámico usando Lagrange-Euler

En base a esta formulación **Lagrangiana**, se puede entonces obtener las ecuaciones de movimiento mediante:

[14]

$$f = \frac{d}{dt} \frac{\partial \mathcal{L}}{\partial \dot{q}} - \frac{\partial \mathcal{L}}{\partial q},$$

Resolviendo estas ecuaciones, se llega al mismo resultado que define el **movimiento del Robot**.

$$ML^2\ddot{\theta} + MgL\cos\theta = \tau$$

Modelo dinámico usando Lagrange-Euler

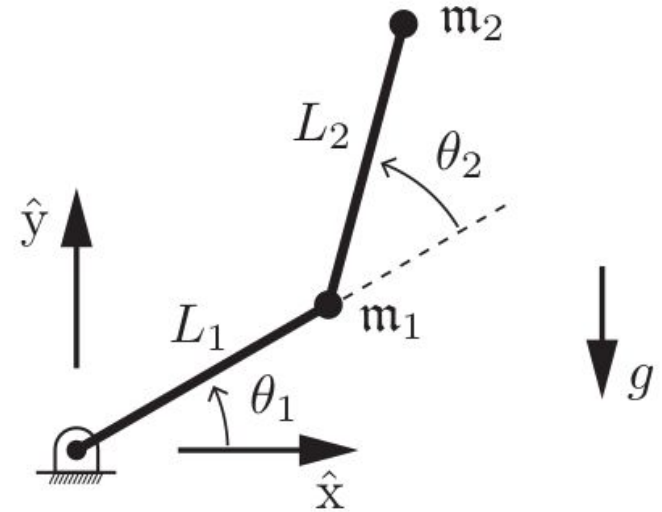
Ejemplo:

Tenemos el siguiente brazo robótico, el cual

está compuesto por masas: m_1 y m_2

Si queremos plantear la **formulación Lagrangiana**,

lo podemos hacer de la siguiente forma:

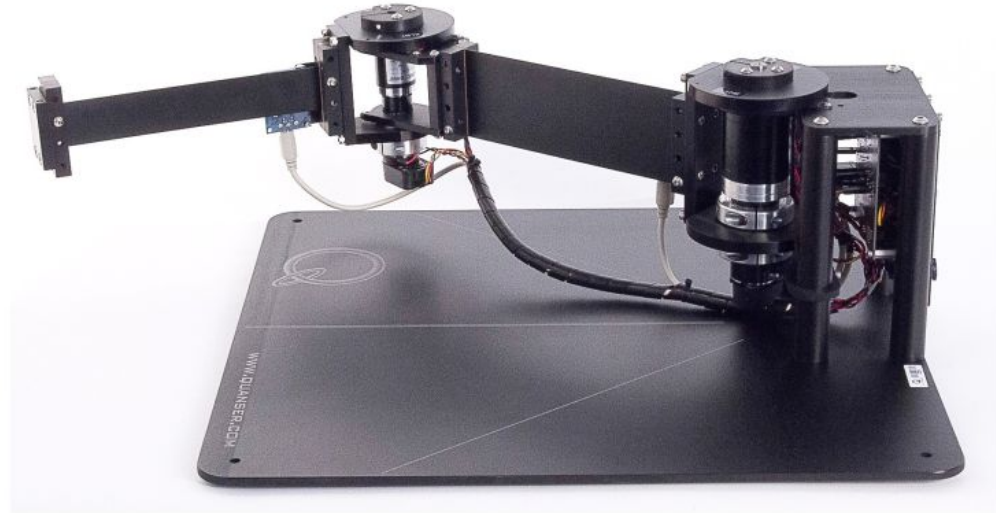


Control cinemático

Aclaración:

El Robot bajo estudio los podemos
asimilar al Robot comercial analizado, pero,
en desde una **vista superior**.

2-DOF Serial Flexible Link Robot



<https://www.quanser.com/products/2-dof-serial-flexible-link/>

Modelo dinámico usando Lagrange-Euler

Estudiamos las **energías cinéticas** puestas en juego

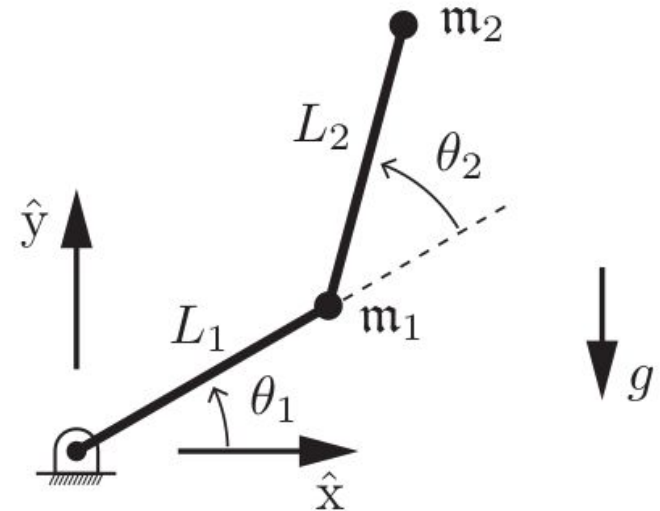
$$\mathcal{K}_1 = \frac{1}{2} m_1 (\dot{x}_1^2 + \dot{y}_1^2)$$

$$\mathcal{K}_2 = \frac{1}{2} m_2 (\dot{x}_2^2 + \dot{y}_2^2)$$

tener en cuenta que m_1 y m_2 tienen coordenadas

(x_1, y_1) y (x_2, y_2) , respectivamente.

[14]



Modelo dinámico usando Lagrange-Euler

Luego estudiamos las **energías potenciales**

puestas en juego

$$\mathcal{P}_1 = m_1 g y_1 = m_1 g L_1 \sin \theta_1,$$

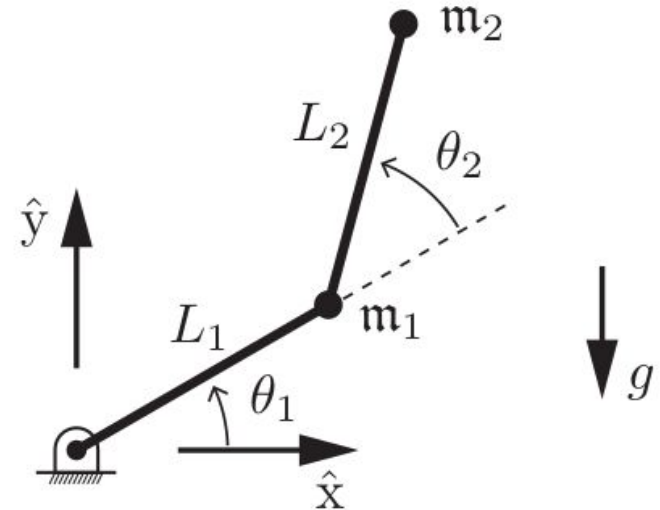
$$\mathcal{P}_2 = m_2 g y_2 = m_2 g (L_1 \sin \theta_1 + L_2 \sin(\theta_1 + \theta_2)).$$

Con esto entonces podemos definir el

Lagrangiano

$$\mathcal{L}(q, \dot{q}) = \mathcal{K}(q, \dot{q}) - \mathcal{P}(q).$$

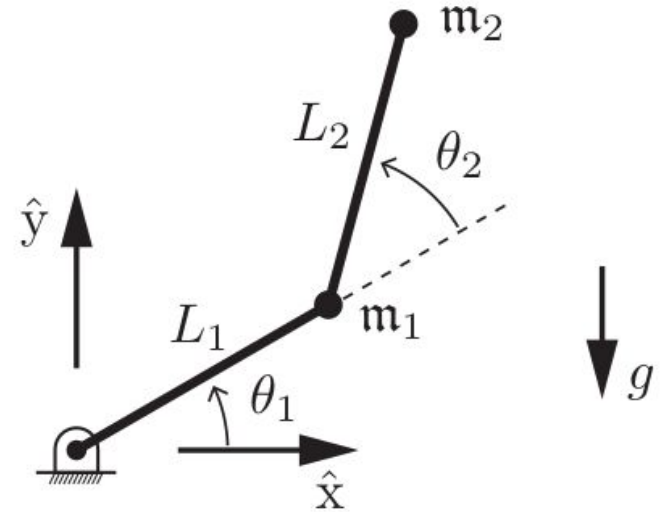
[14]



Modelo dinámico usando Lagrange-Euler

En base a esto, la ecuación de movimiento:

$$\tau_i = \frac{d}{dt} \frac{\partial \mathcal{L}}{\partial \dot{\theta}_i} - \frac{\partial \mathcal{L}}{\partial \theta_i}, \quad i = 1, 2.$$



Dinámica del robot

- Modelo dinámico de la estructura mecánica de un robot rígido. Modelo dinámico usando Newton-Euler.
- Modelo dinámico usando Lagrange-Euler.
- **Modelo dinámico usando variables de estado.**

Modelo dinámico usando variables de estado

El modelo dinámico se puede definir mediante la ecuación:

$$\tau = D\ddot{q} + H + C$$

Siendo:

D: Matriz de inercias **C:** Matriz columna de gravedad

H: Matriz columna de fuerzas de Coriolis y centrípetas

τ : torques existentes en la interacción, por ejemplo, torque de un motor, rozamiento viscoso, etc

Control Dinámico

Muchas veces se suele utilizar esta otra representación:

$$\tau = M(\theta)\ddot{\theta} + h(\theta, \dot{\theta}),$$

[14]

Llamada **Ecuación de Movimiento**.

Siendo:

τ : Vector que contiene fuerzas y torques articulares.

θ : Vector que contiene las variables articulares

$M(\theta)$: Matriz de Masa.

$h(\theta, \dot{\theta})$: Contiene a las fuerzas centrípetas y de Coriolis, términos de gravedad y fricción.

Control Dinámico

De esta forma, asociando con lo visto sobre Dinámica Directa y Dinámica Inversa, podemos ver que la ecuación de movimiento se la puede expresar como:

Dinámica Directa: $\ddot{\theta} = M^{-1}(\theta) \left(\tau - h(\theta, \dot{\theta}) \right)$

Dinámica Inversa: $\tau = M(\theta)\ddot{\theta} + h(\theta, \dot{\theta}),$

Planificación

- Introducción a la Inteligencia Artificial.
- Inteligencia Artificial (IA)
- Introducción a la Robótica.
- Herramientas Matemáticas para la localización espacial y Simulación en Matlab.
- Dinámica del robot.
- **Control Cinemático.**
- Control Dinámico.
- Robótica Embedded, Compiladores y Lenguajes de programación.
- Diseño y Aplicaciones de Robots.
- Cobots.
- Impresoras 3D y 4D

Control cinemático

- **Tipos de trayectoria. Generación de camino continuo.**
- Generación de trayectorias.
- Interpolación de trayectorias: Lineal, cúbica y por tramos.
- Muestreo de trayectorias cartesianas.

Control cinemático

El **control cinemático** establece **cuáles son las trayectorias** que debe seguir **cada articulación** del robot para **lograr objetivos fijados por el usuario** [3] B

Esto se puede pensar de dos formas:

- ★ **Cinemática Directa**
- ★ **Cinemática Inversa**

Control cinemático

Previamente a desarrollar los conceptos, veamos la siguiente animación:

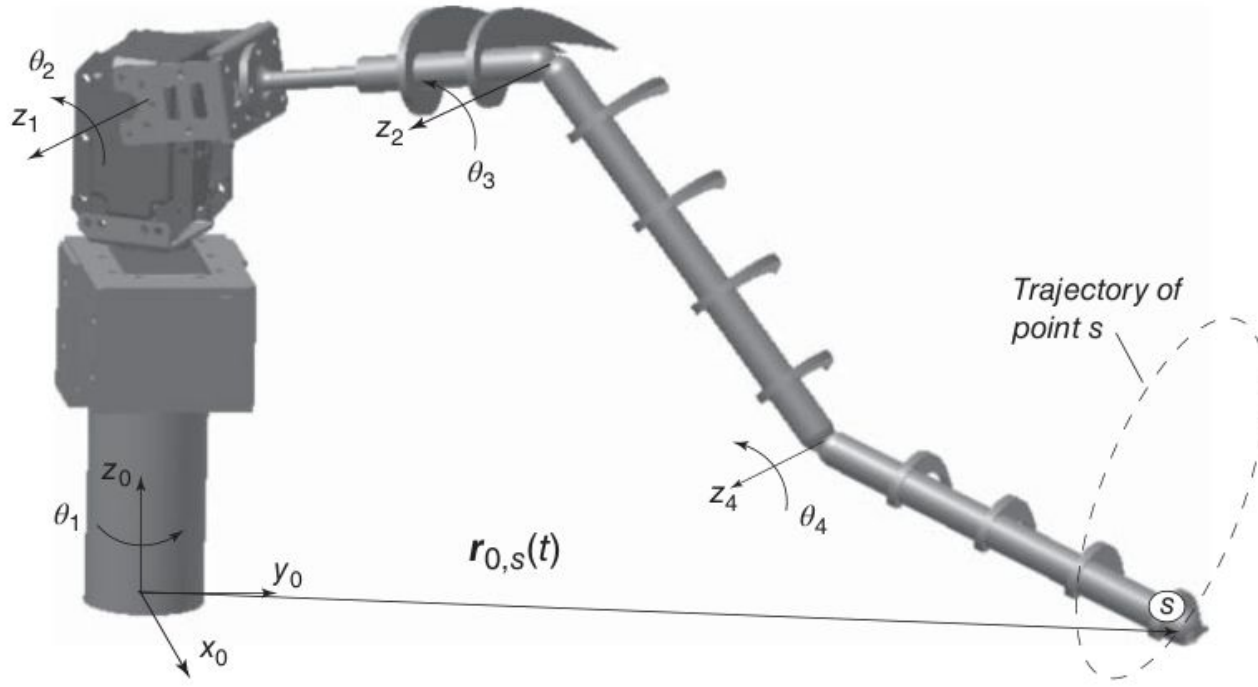
<https://www.youtube.com/watch?v=tXA9O9Wtz64>

Control cinemático

- ★ **Cinemática Directa:** Es una transformación de *ángulos a posición*. Generalmente se *indica la longitud y el ángulo de cada articulación* para *conocer la posición en cualquier punto*. [12]

Veamos el siguiente Ejemplo:

Control cinemático



Control cinemático

Para este caso, podemos tomar como **variables** a los **ángulos de articulación**

$$\theta_1, \theta_2, \theta_3 \text{ y } \theta_4$$

Quienes **determinan las rotaciones** relativas de los **cuerpos**.

Si el robot bajo análisis representa al aleteo de un ave, tendríamos que estudiar a estos **cuatro ángulos** para ver cómo se mueve el punto **S**

Control cinemático

En general, a un **conjunto de variables articulares** se las puede escribir como:

$$q_i(t) \text{ para } i = 1, \dots, N$$

donde **N** es el **número de grados de libertad** y **t** es la variable temporal o tiempo.

Por ejemplo, si estudiamos el **ángulo θ_1** , nos referimos como **$q_1(t)$**

Control cinemático

La **Cinemática Directa** estudia **cómo cambia la configuración del Robot** de acuerdo a **cómo varían las variables articulares ($q_i(t)$)**.

Siguiendo con el mismo análisis, si quisiéramos que el brazo robótico, asemeje sus movimientos al **aleteo de un ave**, la **Cinemática Directa** se va a encargar de estudiar la **posición**, la **velocidad** y la **aceleración** del punto **S**, teniendo en cuenta a las **variables articulares** $q_i(t)$, para $i = 1, \dots, N$. Es decir:

Queremos determinar las siguientes variables de **S**:

$r_{o,s}(t)$ = Distancia de **S** relativa a **O** (marco **O**)

$v_{o,s}(t)$ = Velocidad de **S** relativa a **O** (marco **O**)

$a_{o,s}(t)$ = Aceleración de **S** relativa a **O** (marco **O**)

Control cinemático

Por lo tanto, conociendo la **cinemática de las $q_i(t)$** , podemos obtener la **cinemática del punto S**, esto es:

$$\mathbf{q}(t), \dot{\mathbf{q}}(t), \ddot{\mathbf{q}}(t) \mapsto \mathbf{r}_{0,s}(t), \mathbf{v}_{0,s}(t), \mathbf{a}_{0,s}(t).$$

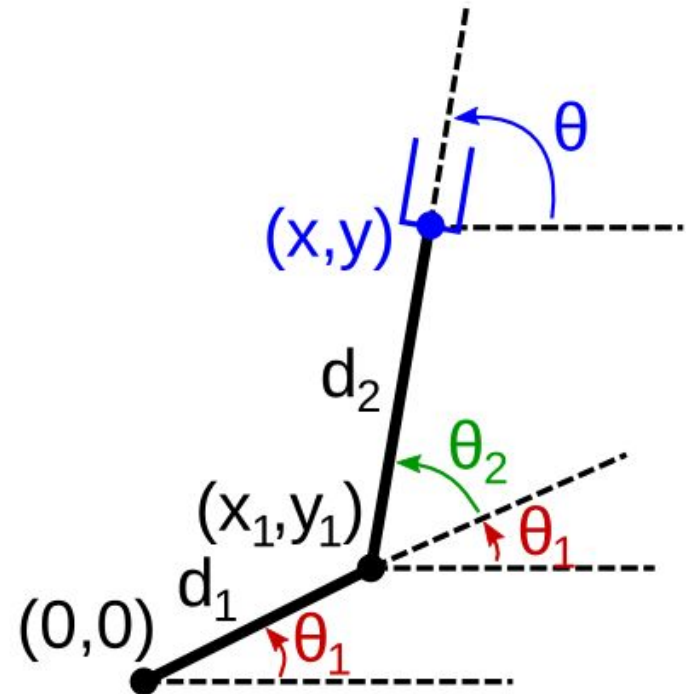
Control cinemático

Ejemplo: Consideremos un manipulador con dos articulaciones giratorias. Los variables de articulación se indican mediante $\mathbf{q} = (\theta_1, \theta_2)$.

Se desea saber la ubicación de la pinza (color azul)

la cual viene dada por $\mathbf{p} = (x, y, \theta)$.

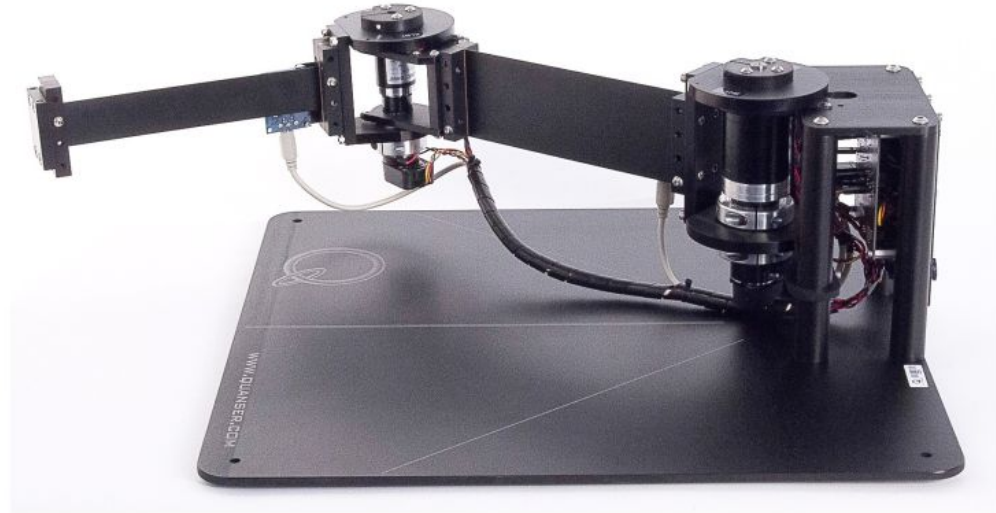
Con esto, lo que buscamos es: $\mathbf{q} \rightarrow \mathbf{p}$



Control cinemático

El Robot bajo estudio los podemos
asimilar al Robot comercial analizado, pero,
en desde una **vista superior**.

2-DOF Serial Flexible Link Robot



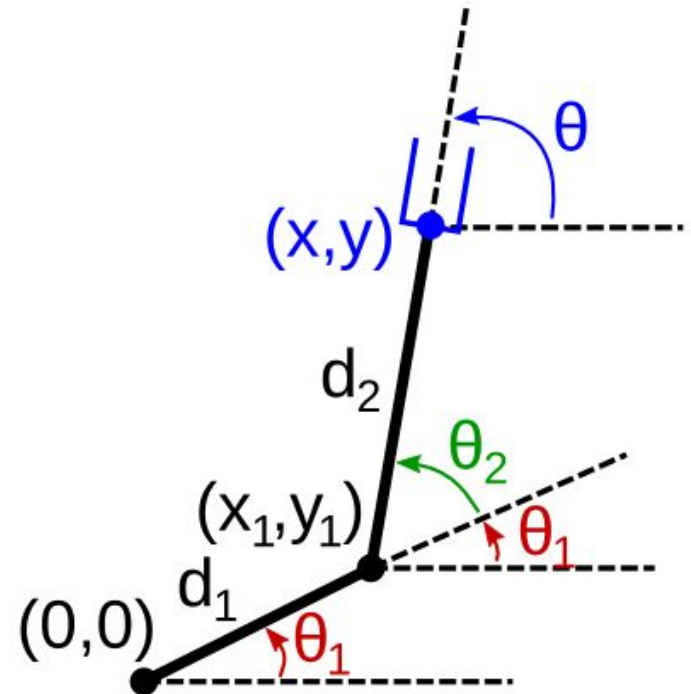
<https://www.quanser.com/products/2-dof-serial-flexible-link/>

Control cinemático

Para encontrar el punto **P**, que representa a la pinza, vamos a aplicar la matemática correspondiente. Empezamos por la coordenada dada por (x_1, y_1) :

$$\begin{cases} x_1 = d_1 \cos(\theta_1) \\ y_1 = d_1 \sin(\theta_1) \end{cases}$$

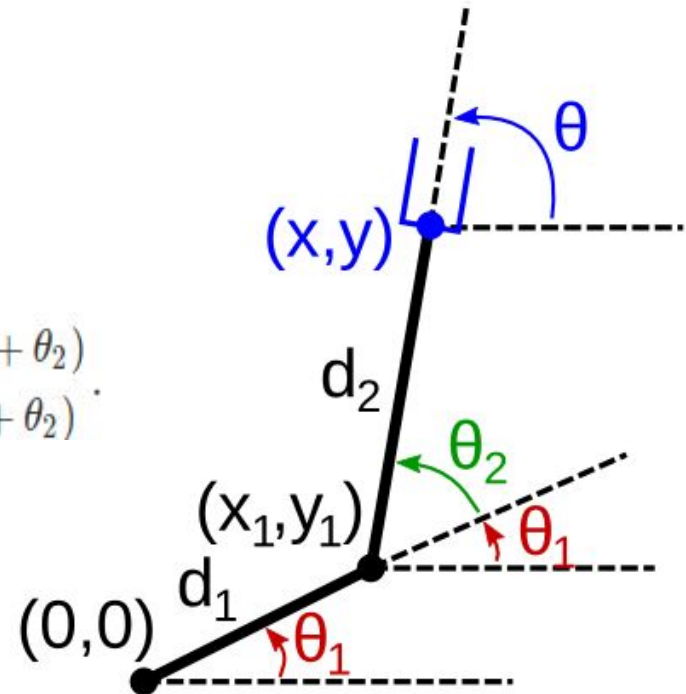
En la coordenada x_1, y_1 , se puede encontrar un motor **M1** que mueva al segundo brazo



Control cinemático

Luego, el punto $P = (x,y)$ lo podemos obtener como:

$$\begin{cases} x = x_1 + d_2 \cos(\theta_1 + \theta_2) = d_1 \cos(\theta_1) + d_2 \cos(\theta_1 + \theta_2) \\ y = y_1 + d_2 \sin(\theta_1 + \theta_2) = d_1 \sin(\theta_1) + d_2 \sin(\theta_1 + \theta_2) \end{cases}$$



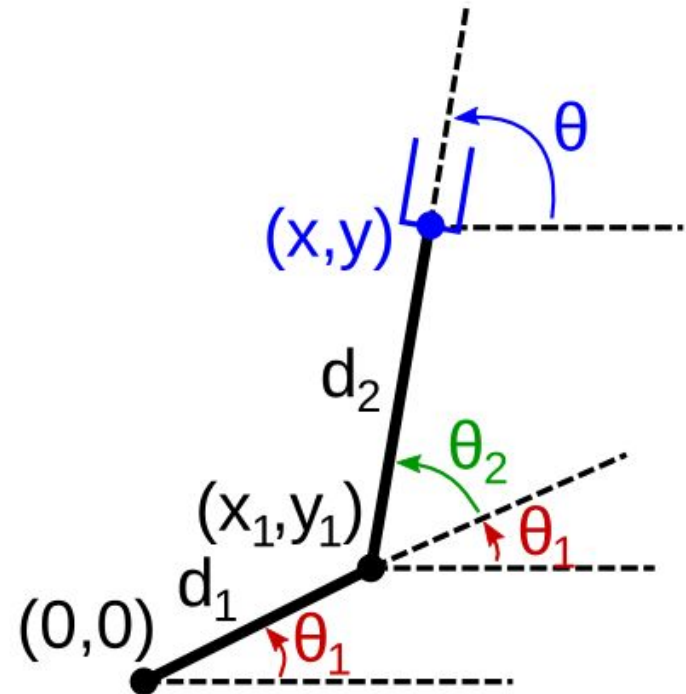
Control cinemático

Por último, podemos ver que: $\theta = \theta_1 + \theta_2$

Es decir, **a partir de los ángulos θ_1 y θ_2 podemos**

obtener la posición de la pinza la cual viene

dada por **P** = (x,y)



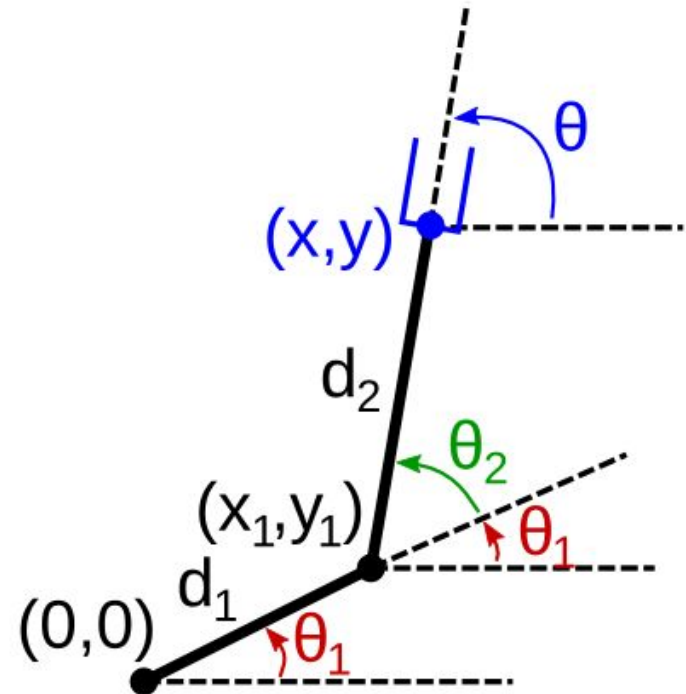
http://www.osrobotics.org/osr/kinematics/forward_kinematics.html

Control cinemático

Ejemplo 1: Supongamos que el robot lo estuvieron usando por un tiempo y lo apagaron. En base a esto, el brazo quedó en los ángulos: $\theta_1 = 45^\circ$ y $\theta_2 = 45^\circ$. Los brazos miden 1m cada uno.

Se pide: Encontrar la posición del motor 1

$M_1 = (x_1, y_1)$ y la de la pinza $P = (x, y)$.



http://www.osrobotics.org/osr/kinematics/forward_kinematics.html

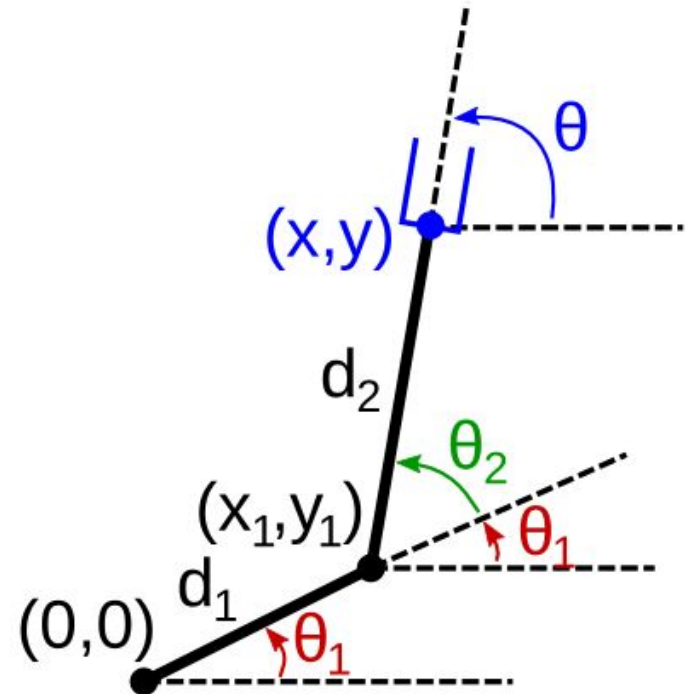
Control cinemático

La posición del motor M_1 la podemos encontrar como:

$$x_1 = d_1 \cdot \cos(\theta_1) = 1 \text{ m} \cdot \cos(45^\circ) = 0,707 \text{ m}$$

$$y_1 = d_1 \cdot \sin(\theta_1) = 1 \text{ m} \cdot \sin(45^\circ) = 0,707 \text{ m}$$

$$M_1 = (x_1, y_1) = (0.707, 0.707)$$



http://www.osrobotics.org/osr/kinematics/forward_kinematics.html

Control cinemático

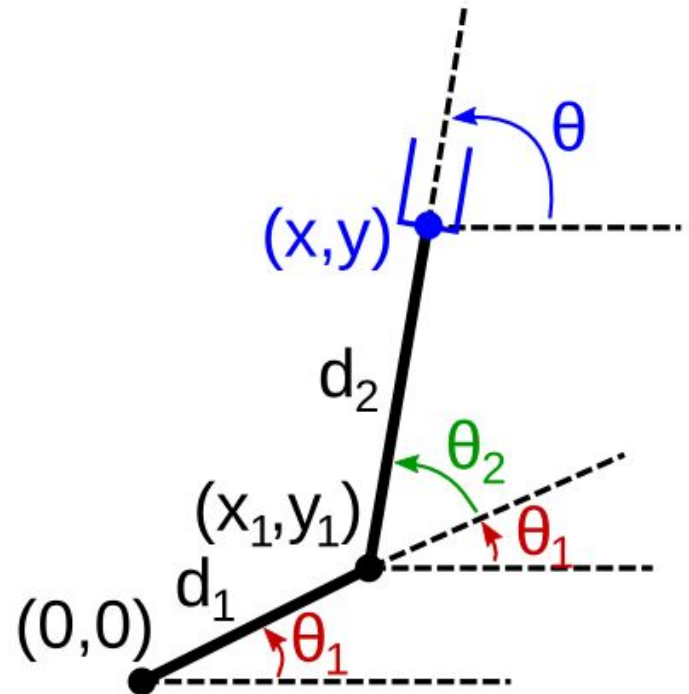
Luego la posición de la pinza la podemos calcular:

$$x = x_1 + d_2 \cdot \cos(\theta_1 + \theta_2)$$

$$x = 0,707\text{ m} + 1\text{ m} \cdot \cos(45^\circ + 45^\circ)$$

$$x = 0,707\text{ m} + 1\text{ m} \cdot \cos(90^\circ) = 0,707$$

http://www.osrobotics.org/osr/kinematics/forward_kin



Control cinemático

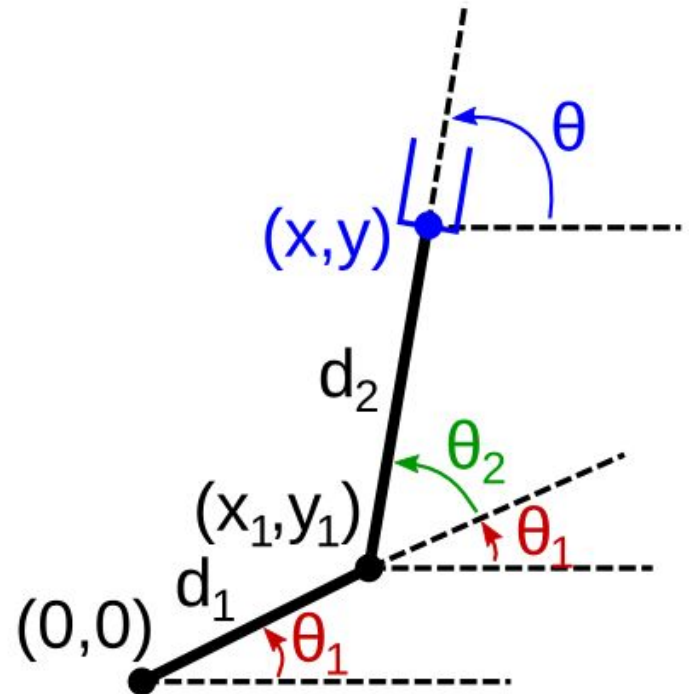
Luego la posición de la pinza la podemos calcular:

$$y = y_1 + d_2 \cdot \sin(\theta_1 + \theta_2)$$

$$y = 0,707 \text{ m} + 1 \text{ m} \cdot \sin(45^\circ + 45^\circ)$$

$$y = 0,707 \text{ m} + 1 \text{ m} \cdot \sin(90^\circ) = 1,707$$

http://www.osrobotics.org/osr/kinematics/forward_kin



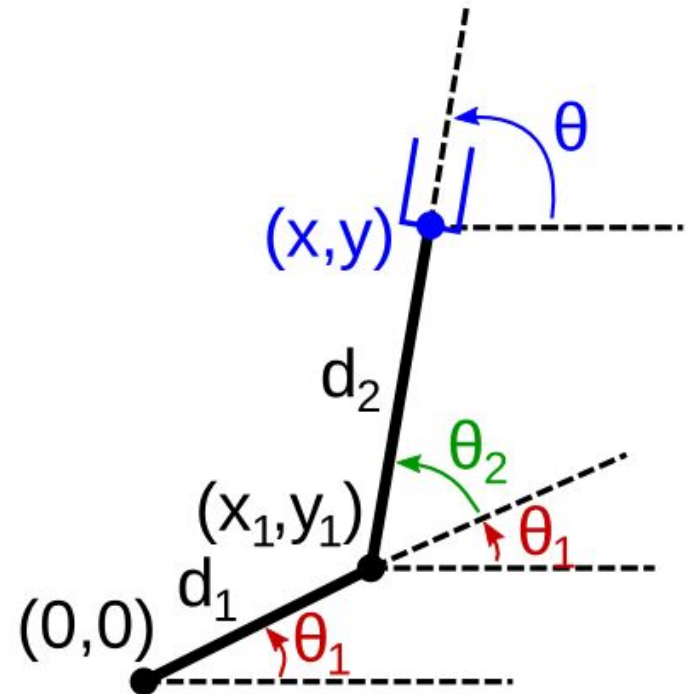
Control cinemático

Es decir, la pinza queda en la posición:

$$\mathbf{P} = (x, y) = 0,707; 1,707$$

y el motor M_1

$$\mathbf{M}_1 = (x_1, y_1) = 0,707; 0,707$$



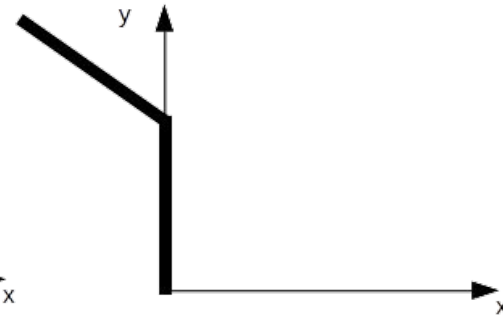
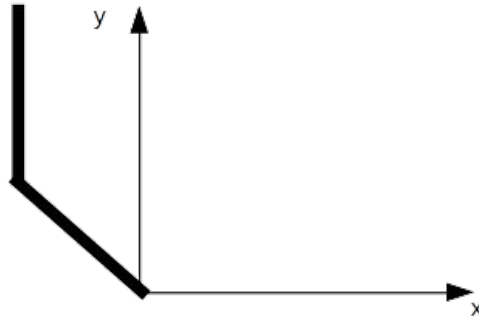
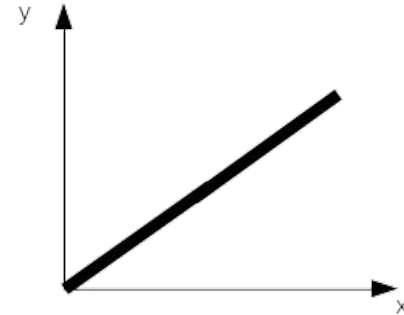
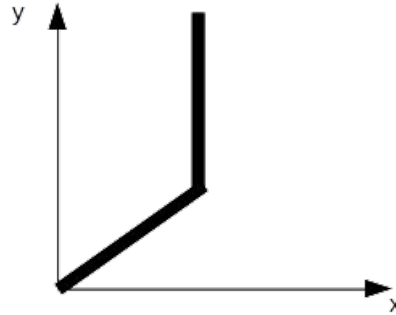
Control cinemático

Si tuviéramos que dibujar las posiciones del brazo, ¿Cómo quedarían?



Control cinemático

Algunas opciones:

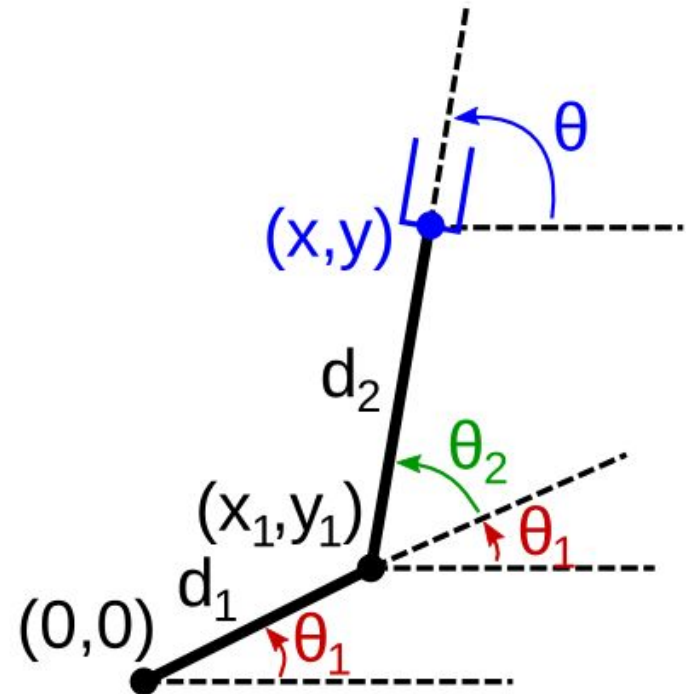


Control cinemático

Problema 1: Supongamos que el robot lo estuvieron usando por un tiempo y lo apagaron. En base a esto, el brazo quedó en los ángulos: $\theta_1 = 90^\circ$ y $\theta_2 = 30^\circ$. Los brazos miden 1m cada uno.

Se pide: Encontrar la posición del motor 1

$M_1 = (x_1, y_1)$ y la de la pinza $P = (x, y)$.

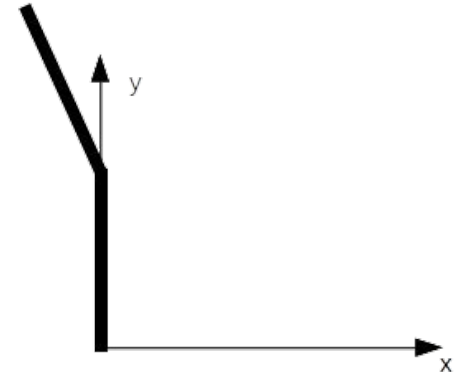
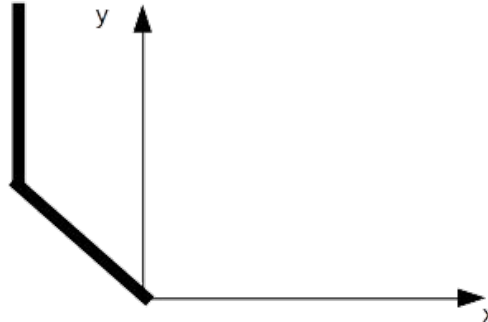
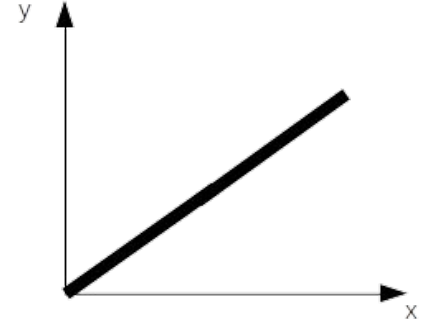
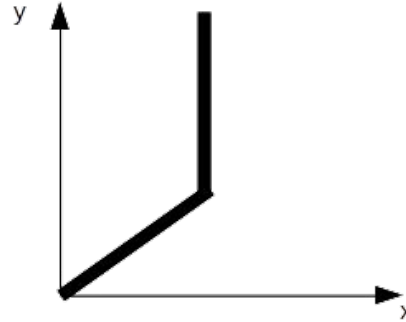


http://www.osrobotics.org/osr/kinematics/forward_kinematics.html

Control cinemático

De forma aproximada:

¿a cuál de las siguientes
opciones se acerca la
respuesta?



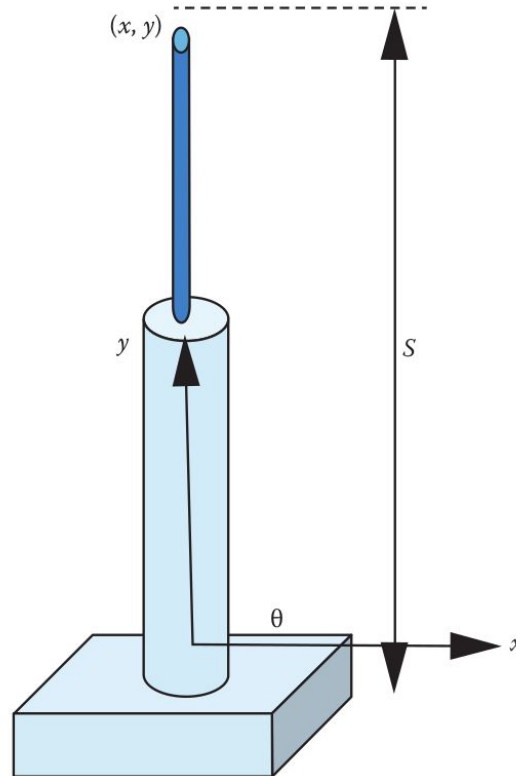
Control cinemático

- ★ **Cinemática Inversa:** En este caso *la posición es dato* y, a partir de de esta, se obtiene el/los *ángulos de cada articulación*.

Veamos el siguiente ejemplo:

Control cinemático

A partir de la **posición** (x,y) ,
podemos conocer los
ángulos.



$$\theta = \arctan\left(\frac{y}{x}\right)$$

$$S = \sqrt{x^2 + y^2}$$

Control cinemático

Siguiendo con el ejemplo del Robot que imita el **aleteo de un ave**, supongamos que luego de haber estudiado estos movimientos a partir de aves reales, se puede saber **la trayectoria que tiene recorrer el punto S**, es decir, podemos conocer la **distancia**, **velocidad** y **aceleración** del punto **S**, en función del tiempo.

La **Cinemática Inversa** estudia, a partir de la **Cinemática del punto S**, la **Cinemática** de las **variables articulares**, es decir, la **posición**, **velocidad** y **aceleración** de cada una de ellas.

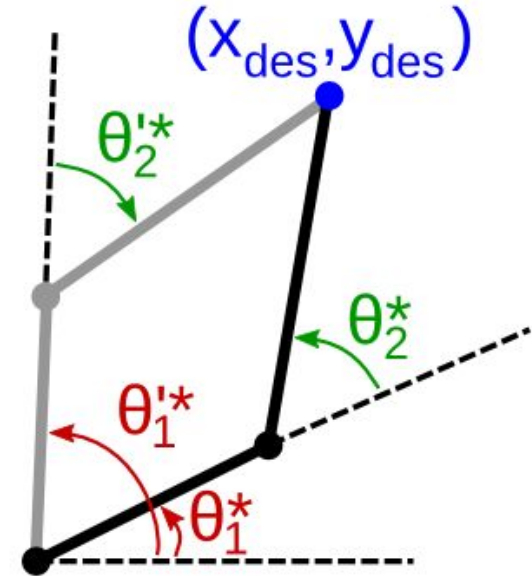
$$\mathbf{r}_{0,s}(t), \mathbf{v}_{0,s}(t), \mathbf{a}_{0,s}(t) \mapsto \mathbf{q}(t), \dot{\mathbf{q}}(t), \ddot{\mathbf{q}}(t).$$

Control cinemático

Ejemplo: Consideremos un manipulador con
dos articulaciones giratorias.

Supongamos que queremos colocar la pinza en la
posición deseada $\mathbf{p}_{des} = (x_{des}, y_{des})$, para lo cual,
tendremos que calcular las **variables de
articulación** $\mathbf{q} = (\theta_1^*, \theta_2^*)$.

Con esto, lo que buscamos es: $\mathbf{p} \rightarrow \mathbf{q}$

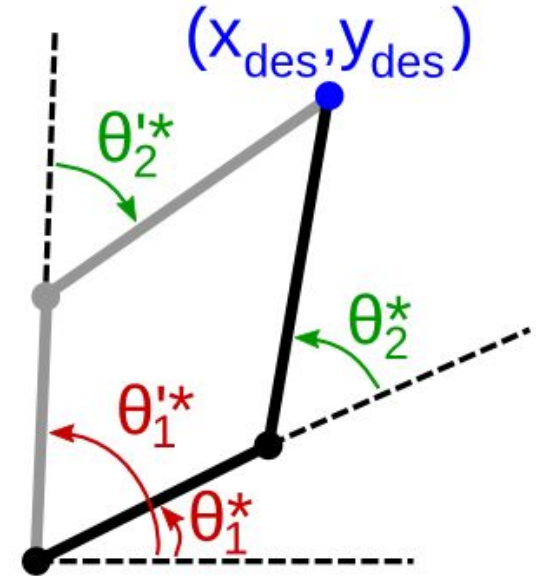


Control cinemático

Aplicando la matemática correspondiente, temos:

$$\begin{cases} x_{des} = d_1 \cos(\theta_1^*) + d_2 \cos(\theta_1^* + \theta_2^*) \\ y_{des} = d_1 \sin(\theta_1^*) + d_2 \sin(\theta_1^* + \theta_2^*) \end{cases} .$$

Recordemos que tenemos que despejar los ángulos

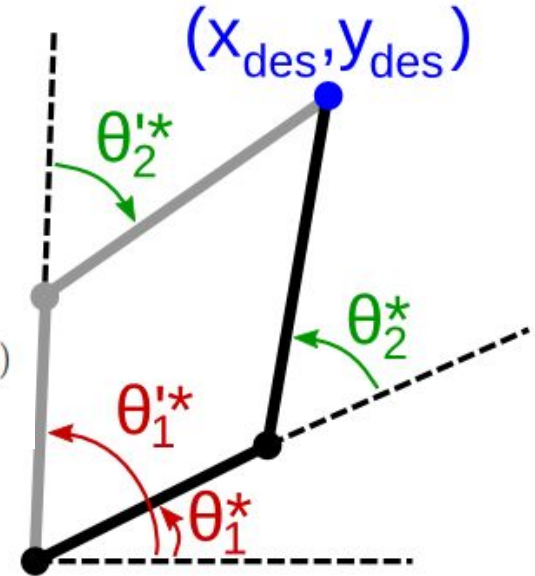


Control cinemático

Dado que tenemos que despejar los ángulos,
podemos elevar al cuadrado a x_{des} e y_{des} y sumarlos.

$$\begin{aligned}x_{\text{des}}^2 + y_{\text{des}}^2 &= d_1^2 + d_2^2 + 2d_1 d_2 (\cos(\theta_1^*) \cos(\theta_1^* + \theta_2^*) + \sin(\theta_1^*) \sin(\theta_1^* + \theta_2^*)) \\&= d_1^2 + d_2^2 + 2d_1 d_2 \cos(\theta_2^*).\end{aligned}$$

http://www.osrobotics.org/osr/kinematics/forward_kinematic



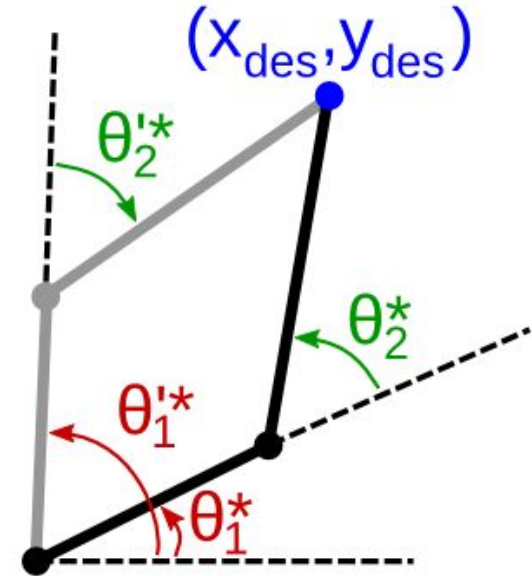
Control cinemático

Con lo anterior, podemos despejar a θ_2^*

$$\cos(\theta_2^*) = \frac{x_{\text{des}}^2 + y_{\text{des}}^2 - d_1^2 - d_2^2}{2d_1 d_2}.$$

$$\theta_2^* = \pm \arccos\left(\frac{x_{\text{des}}^2 + y_{\text{des}}^2 - d_1^2 - d_2^2}{2d_1 d_2}\right)$$

Luego, en base a θ_2^* podemos obtener θ_1^*



Control cinemático

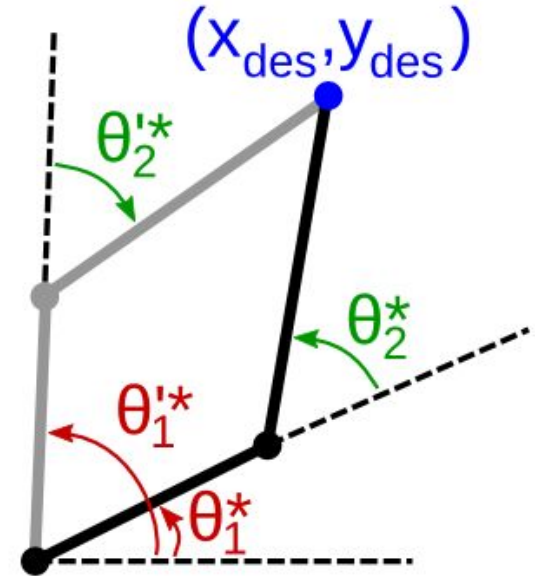
Ejemplo 2: Supongamos que el robot lo estuvieron usando por un tiempo y lo apagaron. En base a esto, el brazo quedó en las posiciones:

$P = (x, y) = 0,707; 1,707$ y el motor M_1

$M_1 = (x_1, y_1) = 0,707; 0,707$. Los brazos miden 1m cada uno.

Se pide: Encontrar el ángulo θ_2^* .

http://www.osrobotics.org/osr/kinematics/forward_kinematics.



Control cinemático

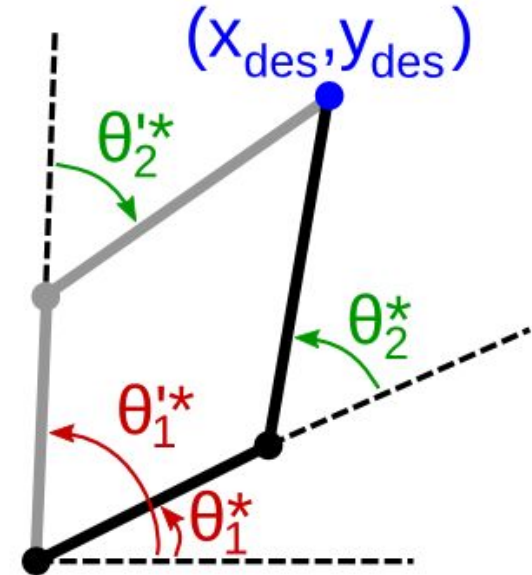
Recordando la geometría, tenemos que:

$$\begin{cases} x_{\text{des}} &= d_1 \cos(\theta_1^*) + d_2 \cos(\theta_1^* + \theta_2^*) \\ y_{\text{des}} &= d_1 \sin(\theta_1^*) + d_2 \sin(\theta_1^* + \theta_2^*) \end{cases}$$

Haciendo manipulaciones, llegamos a que:

$$\theta_2^* = \pm \arccos\left(\frac{x_{\text{des}}^2 + y_{\text{des}}^2 - d_1^2 - d_2^2}{2d_1d_2}\right)$$

http://www.osrobotics.org/osr/kinematics/forward_kinematics.

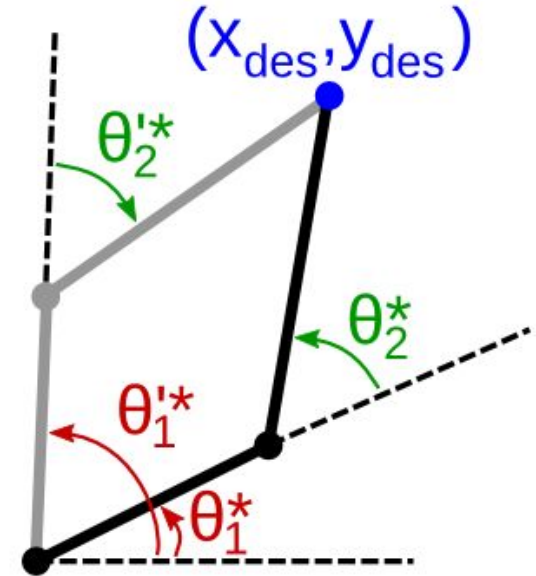


Control cinemático

Para calcular θ_2^* lo que conviene es trabajar primero con el **argumento** del arccos para luego, mediante calculadora, **calcular el arccos**

$$\theta_2^* = \pm \arccos\left(\frac{x_{\text{des}}^2 + y_{\text{des}}^2 - d_1^2 - d_2^2}{2d_1 d_2}\right)$$

http://www.osrobotics.org/osr/kinematics/forward_kinematics.



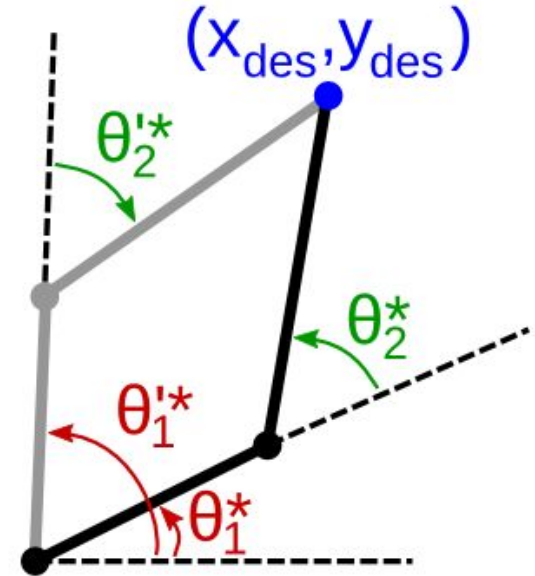
Control cinemático

Empezamos calculando el argumento:

$$\frac{x^2 + y^2 - d_1^2 - d_2^2}{2 \cdot d_1 \cdot d_2}$$

$$\frac{0,707^2 + 1,707^2 - 1^2 - 1^2}{2 \cdot 1 \cdot 1} = 0,707$$

http://www.osrobotics.org/osr/kinematics/forward_kinematics.



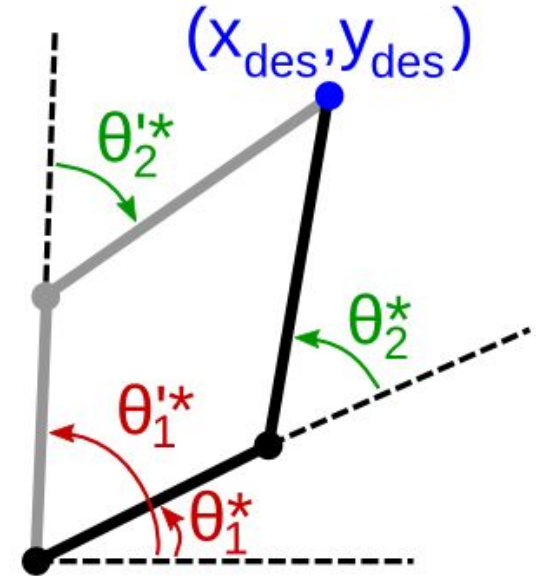
Control cinemático

Luego aplicamos el **arccos** (shift + cos, en la calculadora)
(cos + x⁻¹, en la calculadora del S.O de la PC)

$$\theta_2 = \arccos(0,707) = 45^\circ$$

Nos devuelve el mismo ángulo del **Ejemplo 1**

http://www.osrobotics.org/osr/kinematics/forward_kinematics.



Control cinemático

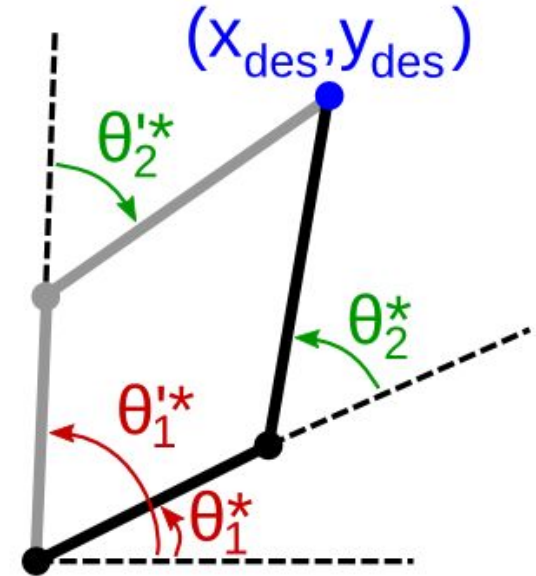
Problema 2: Supongamos que el robot lo estuvieron usando por un tiempo y lo apagaron. En base a esto, el brazo quedó en las posiciones:

$P = (x, y) = -0,5; 1,86$ y el motor

$M_1 = (x_1, y_1) = 0; 1$. Los brazos miden 1m cada uno.

Se pide: Encontrar el ángulo θ_2^* .

http://www.osrobotics.org/osr/kinematics/forward_kinematics.



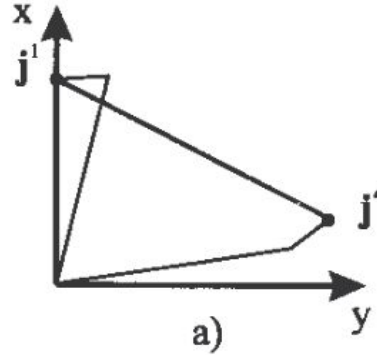
Control cinemático

El ángulo encontrado, ¿es el mismo que el del problema 1?



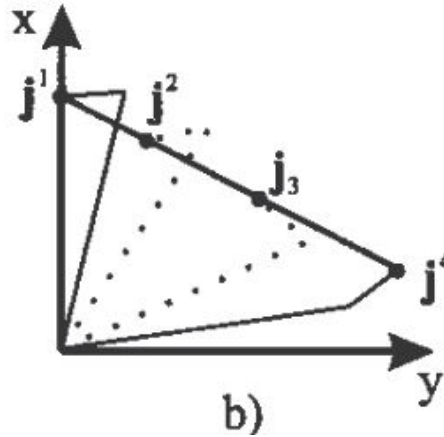
Control cinemático

Supongamos que se quiere mover un robot de 2 grados de libertad se mueva en línea recta desde j^1 hasta j^4 en un tiempo T



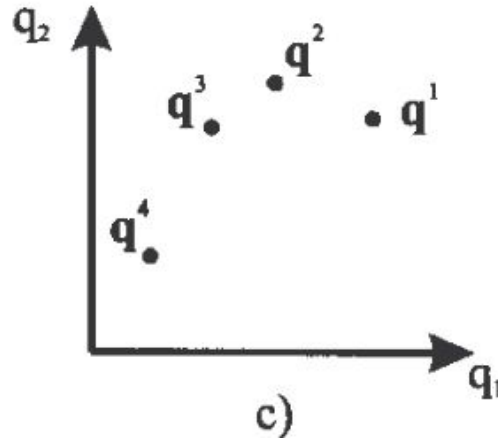
Control cinemático

El **control cinemático** selecciona **cuatro puntos de esta trayectoria**, por ejemplo, j^1 , j^2 , j^3 , j^4



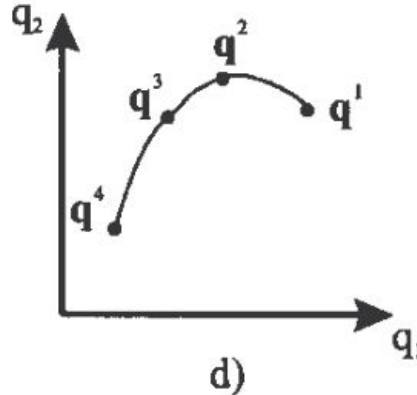
Control cinemático

Luego, mediante la *transformación homogénea inversa* se obtiene los *vectores articulares* q^1 , q^2 , q^3 , q^4 .



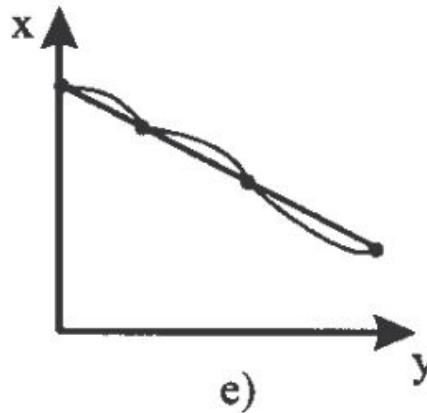
Control cinemático

A continuación se trata de unir estos 4 puntos con algún tipo de función tal que, pasando por todos ellos, **garantice la suavidad** y **no supere velocidades y aceleraciones máximas permisibles para cada accionador** [3] B



Control cinemático

El resultado final del movimiento del robot, es una **trayectoria** que se **aproxima a la línea recta** deseada [3] B



Tipos de Trayectorias

Por lo visto hasta aquí, para realizar una determinada tarea, *el robot debe moverse desde un punto inicial hasta otro final*. Este movimiento puede ser realizado según *infinitas trayectorias espaciales*. De todas ellas hay algunas que, bien por su sencillez en el control o por su utilidad, *son las que se implementan en la práctica en robots comerciales*. De esta forma entonces encontramos que existen las siguientes trayectorias.

- Trayectorias punto a punto
- Trayectorias coordinadas
- Trayectorias continuas

Tipos de Trayectorias

Trayectorias punto a punto:

En este tipo de trayectorias, *cada articulación evoluciona desde su posición inicial a la final sin realizar consideración alguna* sobre el *estado* o evolución *de las demás articulaciones*. Normalmente, cada actuador trata de llevar su articulación al punto destino en el menor tiempo posible.

Tipos de Trayectorias

Trayectorias coordinadas:

Para evitar que algunos actuadores *trabajen forzando sus velocidades y aceleraciones*, teniendo que esperar después *la conclusión del movimiento de la articulación más lenta*, puede hacerse un *cálculo previo*, averiguando *cuál es la articulación y qué tiempo invertirá*.

De esta forma *todas las articulaciones se coordinan comenzando y acabando su movimiento a la vez*, adaptándose a la más lenta.

El tiempo total invertido en el movimiento es el menor posible y no se piden aceleraciones y velocidades elevadas a los actuadores de manera inútil.

Tipos de Trayectorias

Trayectorias continuas:

Cuando se pretende que la *trayectoria que sigue el extremo del robot sea conocida por el usuario*, es preciso calcular de manera continua las trayectorias articulares.

Generalmente, las trayectorias que el usuario pretende que el robot describa, son trayectorias en *línea recta o en arco de círculo*. El resultado será que *cada articulación sigue un movimiento aparentemente caótico* con posibles cambios de dirección y velocidad y sin coordinación con el resto de las articulaciones. Sin embargo, el resultado conjunto será que el extremo del robot, seguirá la trayectoria deseada.

Control cinemático

- Tipos de trayectoria. Generación de camino continuo.
- **Generación de trayectorias.**
- Interpolación de trayectorias: Linear, cúbica y por tramos.
- Muestreo de trayectorias cartesianas.

Tipos de Trayectorias

Generación de trayectorias:

Normalmente *el usuario del robot indica el movimiento que éste debe realizar*, especificando las *localizaciones espaciales* por las que debe pasar el extremo. Por ejemplo, es frecuente *especificar que el robot debe ir de un punto a otro final*, siguiendo una *línea recta a velocidad constante*.

Dado que esos *puntos están excesivamente separados*, *resulta necesario seleccionar puntos intermedios lo suficientemente cercanos* como para que el control del robot *consiga ajustar la trayectoria seguida*.

Para esto, es *necesario establecer un interpolador entre las localizaciones* expresadas en el espacio, que dará como resultado una *expresión analítica de la evolución de cada coordenada*.

Tipos de Trayectorias

La *interpolación más frecuente es la lineal*, en la que *cada coordenada evoluciona a velocidad constante desde su valor inicial, hasta su valor final*.

$$j(t) = (j^f - j^i) \frac{t - t_i}{t_f - t_i} + j^i$$

Siendo t_i y t_f los instantes de tiempo donde inicia y donde finaliza la localización

Control cinemático

- Tipos de trayectoria. Generación de camino continuo.
- Generación de trayectorias.
- **Interpolación de trayectorias: Lineal, cúbica y por tramos.**
- Muestreo de trayectorias cartesianas.

Interpolación de trayectorias

Una de las *funciones* del *control cinemático* es la de unir una *sucesión de puntos en el espacio articular* por los que *se quiere que pasen las articulaciones* del robot en un instante determinado.

Además, junto con las *condiciones de posición-tiempo*, es conveniente *añadir restricciones de velocidad y aceleración* de paso por los puntos, de manera que se asegure *la suavidad de la trayectoria* y se *limiten las velocidades y aceleraciones máximas*.

Para esto, *se debe seleccionar algún tipo de función* (polinómica, por ejemplo), *cuyos coeficientes se ajustarán al imponer las condiciones de contorno*: posiciones, velocidades y aceleraciones.

Interpolación de trayectorias

Interpolador lineal:

Suponiendo que cada una de las **articulaciones q del robot**, pasa sucesivamente por los valores q^i en los instantes t^i , la trayectoria viene dada por:

$$q(t) = (q^i - q^{i-1}) \frac{t - t^{i-1}}{T} + q^{i-1} \quad t^{i-1} < t < t^i$$
$$T = t^i - t^{i-1}$$

Esta trayectoria **asegura la continuidad de la posición**, pero **origina saltos bruscos de la velocidad** de $\dot{q}$ de la articulación y de su **aceleración $\ddot{q}$** . Con lo cual, **se debe seleccionar los instantes de paso t^i por los pasos q^i**

Interpolación de trayectorias

Interpolador cúbico

Para asegurar que la **trayectoria que une los puntos** por los que tiene que pasar la articulación **presente continuidad en velocidad**, se puede recurrir a la **utilización de un polinomio de grado 3** que une cada pareja de puntos adyacentes. Esta expresión viene dada por:

$$q(t) = a + b(t - t^{i-1}) + c(t - t^{i-1})^2 + d(t - t^{i-1})^3 \quad t^{i-1} < t < t^i$$

q^{i-1} , $\dot{q}^i$: dos puntos adyacentes

$$a = q^{i-1}$$

$$b = \dot{q}^{i-1}$$

$$c = \frac{3}{T^2}(q^i - q^{i-1}) - \frac{2}{T^2}\dot{q}^{i-1} - \frac{1}{T^2}\dot{q}^i$$

$$d = -\frac{2}{T^3}(q^i - q^{i-1}) + \frac{1}{T^2}(\dot{q}^{i-1} + \dot{q}^i)$$

$$T = t^i - t^{i-1}$$

Interpolación de trayectorias

Interpolador a tramos

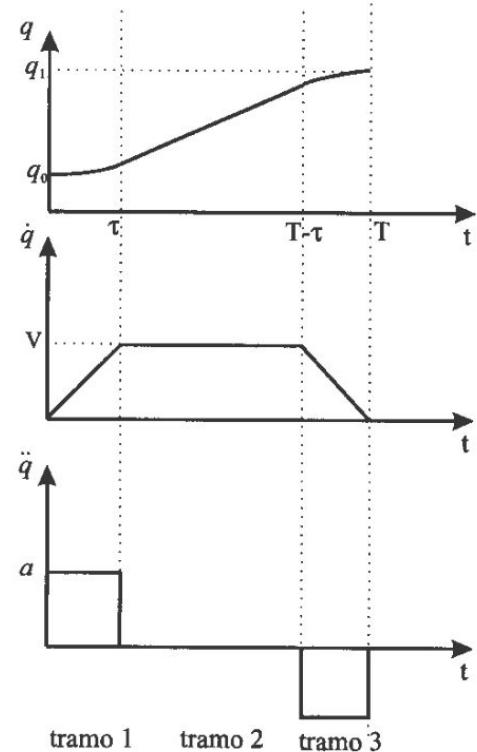
Los *interpoladores tratados hasta ahora*, resultan ser un *polinomio de grado 1 o 3*, para unir dos puntos de una trayectoria. *Existe otra alternativa*, que consiste *en descomponer en tres tramos consecutivos la trayectoria que une dos puntos, q^0 , q^1* .

En el *tramo central*, se utiliza un *interpolador lineal*, y por lo tanto la *velocidad se mantiene constante*, no siendo preciso imprimir la aceleración alguna al actuador.

En los *tramos iniciales y finales*, se utiliza un *polinomio de segundo grado*, de modo que en el tramo 1 la velocidad varíe linealmente desde la velocidad de la trayectoria anterior a la de la presente, y en el tramo 3 varíe desde la velocidad de la trayectoria presente hasta la siguiente. Se tiene entonces que en los *tramos iniciales y finales, la aceleración toma valores constantes distintos de cero*, mientras que en el *tramo central, la aceleración es nula*.

Interpolación de trayectorias

En el caso simple de una trayectoria con dos
únicos puntos de velocidad inicial y final nula,
se tiene:



Control cinemático

- Tipos de trayectoria. Generación de camino continuo.
- Generación de trayectorias.
- Interpolación de trayectorias: Linear, cúbica y por tramos.
- **Muestreo de trayectorias cartesianas.**

Muestreo de trayectorias cartesianas

Habiendo analizado las diferentes alternativas *para interpolar los puntos seleccionados* de la trayectoria que garanticen la obtención de curvas realizables por los actuadores, queda por establecer *criterios adecuados para la selección óptima de los puntos de la trayectoria.*

Sin embargo, es importante considerar que *cada uno de los puntos debe ser transformado a sus coordenadas articulares correspondientes* para luego utilizar alguno de los *interpoladores descriptos*. Esto entonces describe un *costo computacional importante*, pues se necesita hacer la *transformación continuamente* lo cual *entorpece a la respuesta en tiempo real del sistema de control del robot*. Por esta razón, *no es aconsejable tomar demasiados puntos.*

Muestreo de trayectorias cartesianas

Además, es importante tener en cuenta el *error* cometido entre la *trayectoria resultante del robot y la deseada*, la cual puede variar en gran medida. Por lo tanto, *resulta conveniente no sólo no tomar muchos puntos sino los que son realmente importantes para la trayectoria*.

En la práctica suele ser suficiente con seleccionar puntos equidistantes tan cercanos como sea posible.