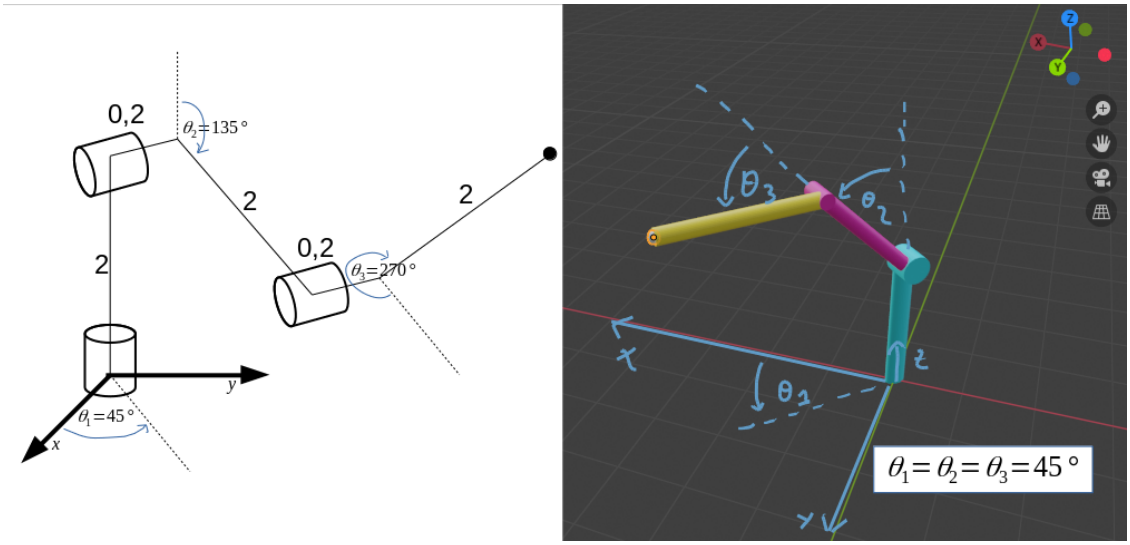
Laboratorio I
Introducción a Robótica

Iván Ferrari*

Junio, 2021



*Ingeniero Químico, FCAI-UNCuyo. Profesor Adjunto, Laboratorio I, departamento de Ingeniería Mecánica

Contenido

1 Introducción a Robótica

Definición de Robot

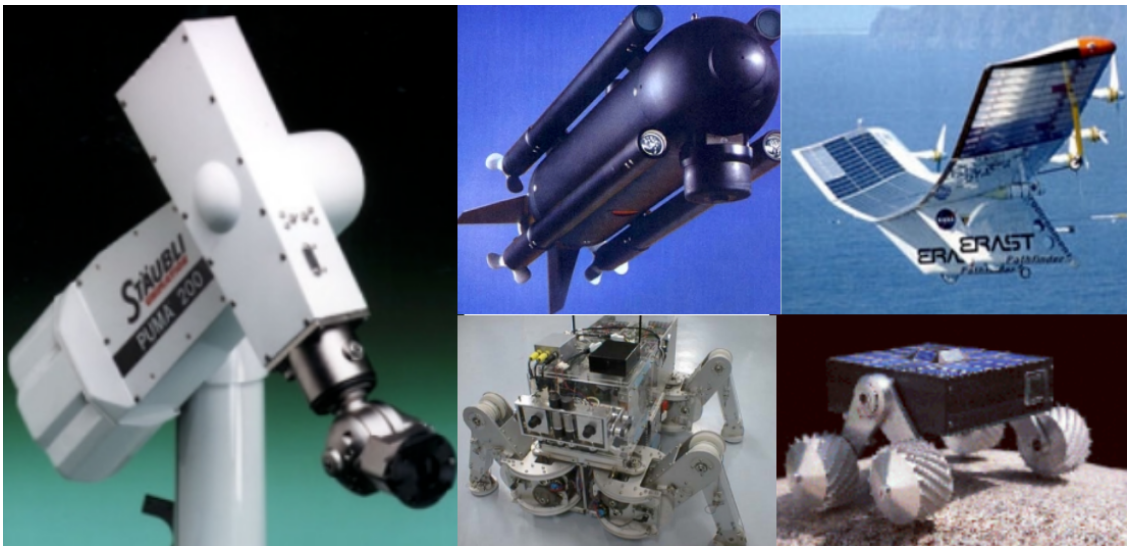
Es un manipulador multifuncional reprogramable diseñado para mover materiales, partes, herramientas o dispositivos especializados a través de movimientos variables programados para la realización de una variedad de tareas. (Robot Institute of America, 1979)

Es una máquina, especialmente una máquina programable, capaz de llevar a cabo una serie de acciones complejas de forma automática (Diccionario Oxford - 2016)

Un robot es una entidad virtual o mecánica artificial. En la práctica, esto es por lo general un sistema electromecánico que, por su apariencia o sus movimientos, ofrece la sensación de tener un propósito propio. La independencia creada en sus movimientos hace que sus acciones sean la razón de un estudio razonable y profundo en el área de la ciencia y tecnología. La palabra robot puede referirse tanto a mecanismos físicos como a sistemas virtuales de software, aunque suele aludirse a los segundos con el término de bots. (Alliance for Telecommunications Solutions 2001)

1.1 Tipos de robots

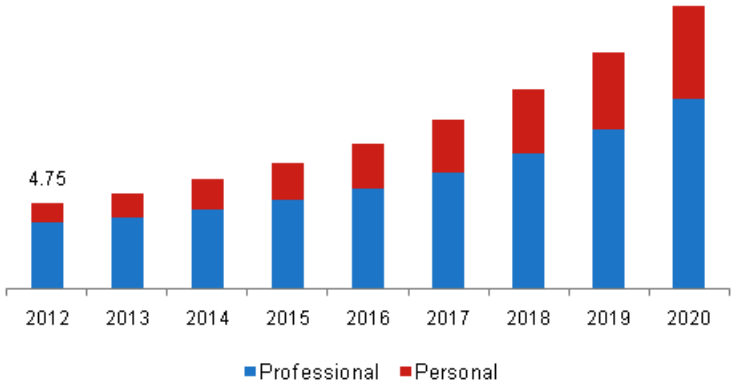
- Manipuladores
- vehículos autónomos, en tierra, aire y agua
- robots humanoides, con piernas, y otros

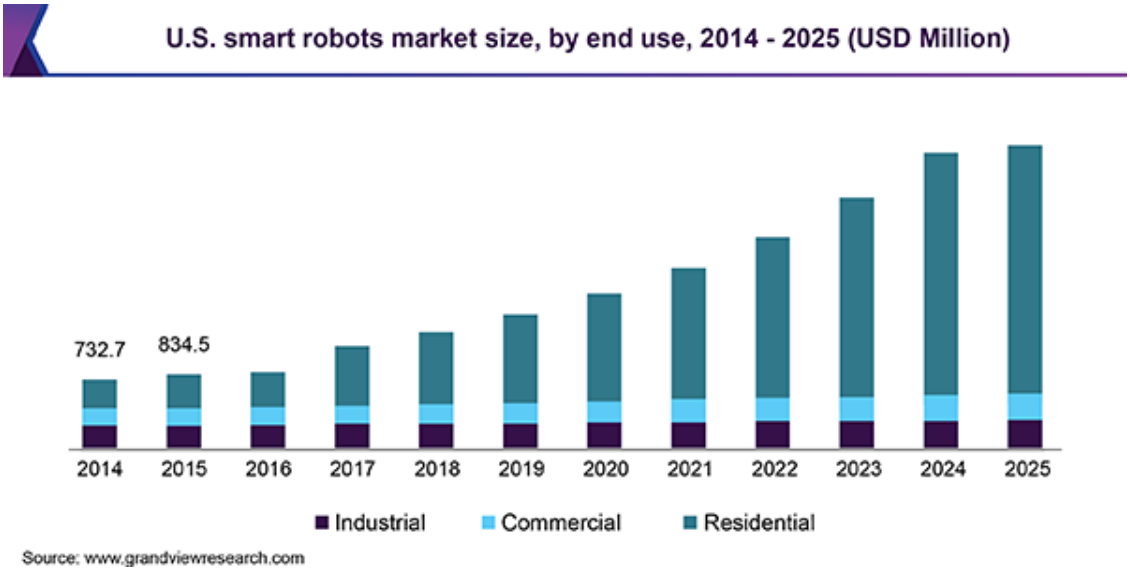


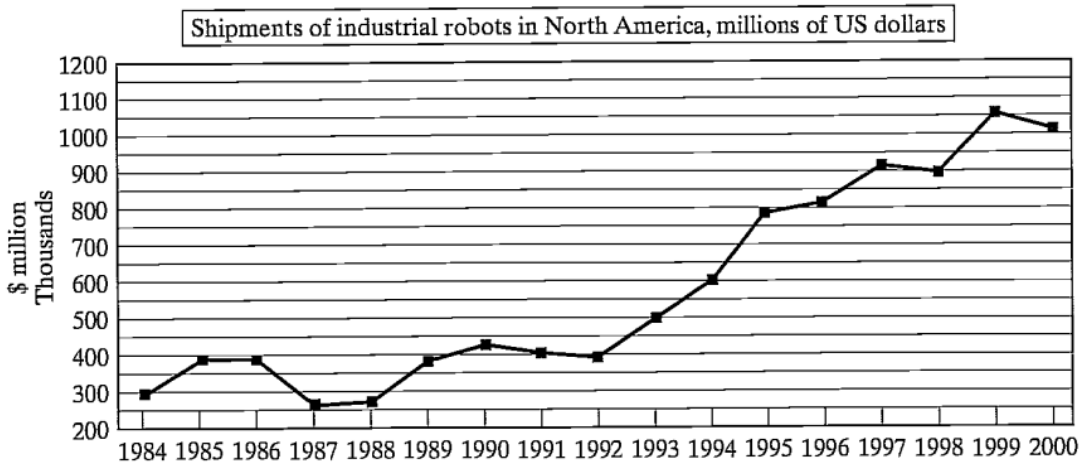
1.2 Importancia usos y aplicaciones

aplicaciones * industria: * tareas peligrosas * tareas repetitivas * tareas de gran potencia * tareas de presición * servicio, domésticos * investigación * industria espacial * area militar rescate

1.3 Mercado


4.75
2012
2013
2014
2015
2016
2017
2018
2019
2020
Professional
Personal


U.S. smart robots market size, by end use, 2014 - 2025 (USD Million)
732.7
834.5
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
Industrial
Commercial
Residential
Source: www.grandviewresearch.com


Shipments of industrial robots in North America, millions of US dollars
$ million
Thousands
1200
1100
1000
900
800
700
600
500
400
300
200
1984 1985 1986 1987 1988 1989 1990 1991 1992 1993 1994 1995 1996 1997 1998 1999 2000

¿es cierto que la automatización y la robótica generan desempleo? brazos robóticos para uso industrial

1.4 Estructura mecánica:

- eslabones
- articulaciones o uniones
  - esferica
  - plana
  - tornillo
  - prismatica
  - rotación
  - cilíndrica

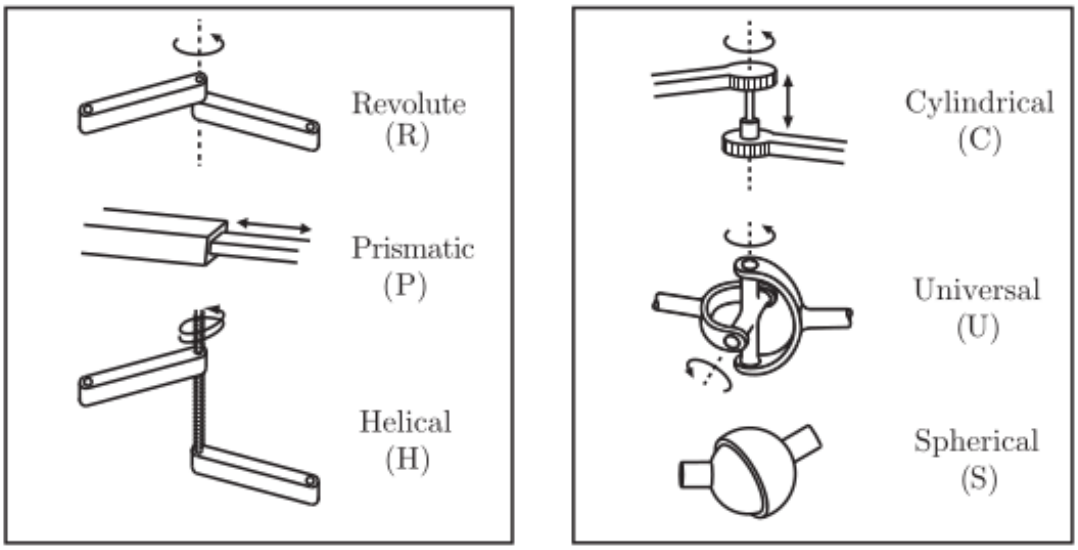


Figure 2.3: Typical robot joints.

Los más usados son Prismatíca y Rotación, combiando estos podemos llegar a cualquier tipo de posición o articulación equivalente

nomenclatura con articulaciones Prismáticas y Rotación robot scara RRRP (contadas desde la base)

scara son rapidos por no cargar su peso

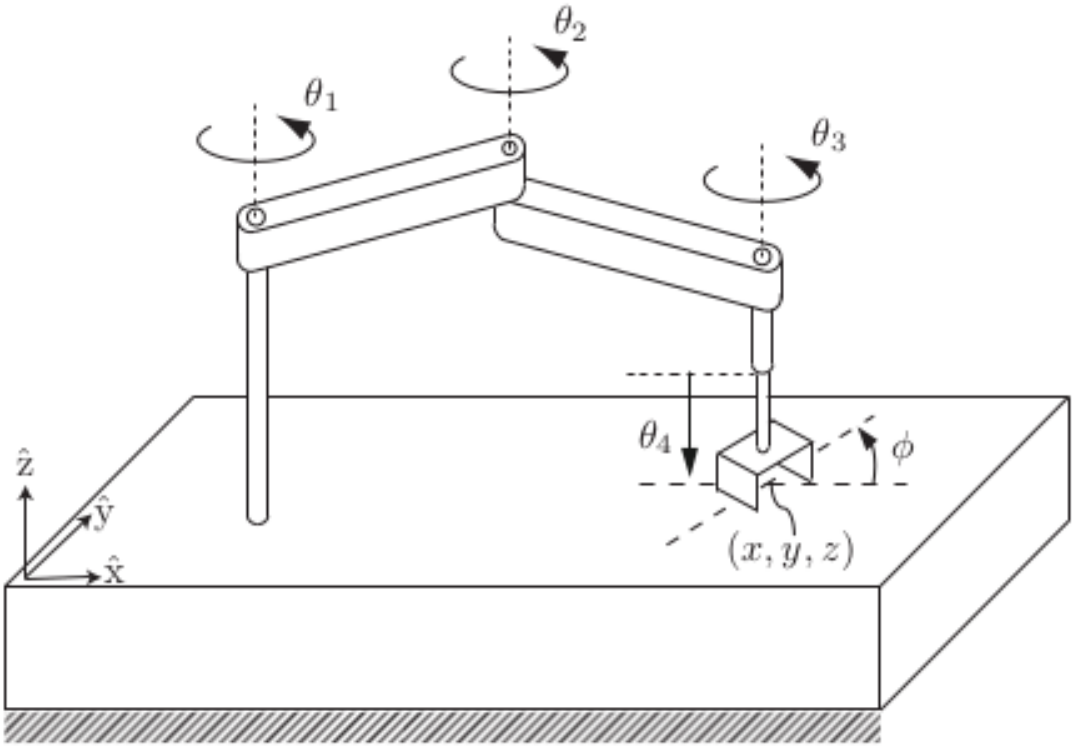


Figure 2.13: SCARA robot.

1.5 Transmisiones y reductores

- transmisión desde el actuador/motor hasta la articulación vs actuación directa (donde el cargador de la base debe cargar el peso de los motores de la punta, analogía cuerpo humano)
- Transmisión circular a circular, engranajes, cadenas, correas, cables
- circular a linear, tornillo sin fin
- lineal a circular

Reductores, adaptan velocidades y torques del actuador en las articulaciones

ver engranajes planetarios problema Backlash hueco entre engranajes q tarda en transmitir

transmisión directa motor montado sobre la articulación sin reductor, tiene gran presición , el motor debe estar hecho a medida para la articulación (ver motor sin escobilla mini cheetha del mit) usan encoders

1.6 Diseño mecánico

1.6.1 transmisión

- Engranajes otro

- correas otro correa con posible defecto de tensión
- cables
- neumático e hidráulico

1.7 Actuadores

Actuadores, generan el movimiento, conviertn un tipo de energía en movimiento

neumático (rapidos y limpios, bajo torq porq con gran carga tenes compresión del aire) * cilindro neumático * motor neumático https://www.youtube.com/watch?v=53EMnoYky0U&t=9s

eléctrico * DC por inducción o exitación * AC, sincrónici asincrónico * paso a paso

hidráulico (más lentos, soportan grandes torques) * cilindro hidráulico * motor hidráulico https://www.youtube.com/watch?v=_sBBaNYex3E

otros hidraulic artificial muscle https://www.youtube.com/watch?v=a6mRhuR_g-E&t=50s

1.8 Sensores

para medir el exterior o para medir variables internas

presencia * inductivo capacitivo * efecto hall * optico * ultrasonido * contacto

posición: analogicos: potenciometros, resolvert digitales, encoders

velocidad tacometros posición

1.9 Elementos terminales, punta de la herramienta

cadena cinemática abierta, punta de la herramienta al final es el que realiza la tarea particular

1.9.1 Elementos de sujección

- agarre magnético
- agarre tipo pinza
- suaves, soft robotics
- otros agarres

1.9.2 Espacio de trabajo y espacio de tarea

El espacio de tarea es el espacio en el que se puede usar de forma efectiva la punta de la herramienta

Es espacio de trabajo incluye además cualquier espacio que puede ser alcanzado por una parte del

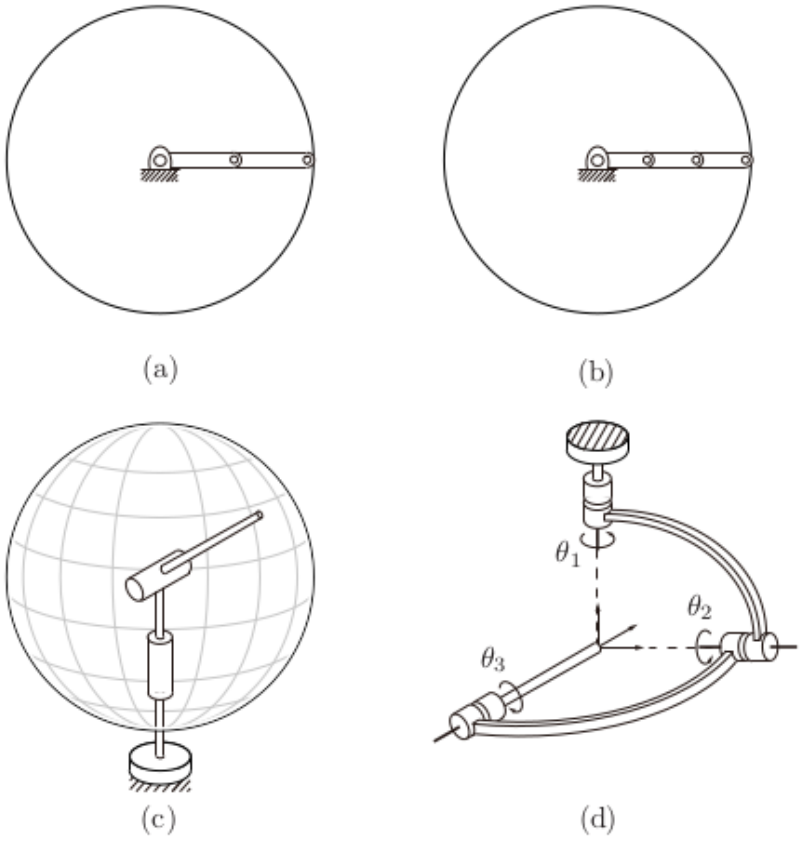


Figure 2.12: Examples of workspaces for various robots: (a) a planar 2R open chain; (b) a planar 3R open chain; (c) a spherical 2R open chain; (d) a 3R orienting mechanism.

manipulador.

La consideración del espacio de trabajo es particularmente importante para delimitar un área de seguridad alrededor del robot.

choque robot

falla y choque

seguridad choque

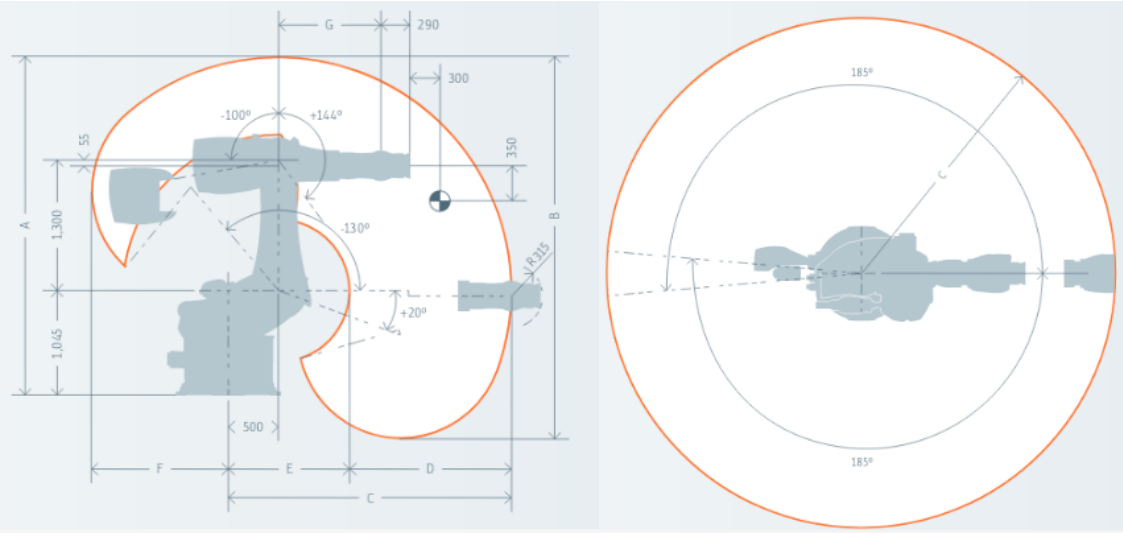
https://www.youtube.com/watch?v=GtNKX4kpC18&t=28s

barreras de protección para el espacio de trabajo

1.10 Características principales para la selección

- número de articulaciones
- carga máxima de trabajo
- Repetitividad

KR FORTEC	KR 360 R2830	KR 280 R3080	KR 240 R3330
Max. reach	2,826 mm	3,076 mm	3,326 mm
Rated payload	360 kg	280 kg	240 kg
Rated suppl. load, arm/link arm/rot. col.	50 kg / – / –	50 kg / – / –	50 kg / – / –
Rated total load	860 kg	780 kg	740 kg
Pose repeatability	±0.08 mm	±0.08 mm	±0.08 mm
Number of axes	6	6	6
Mounting position	Floor, ceiling [1]	Floor, ceiling	Floor, ceiling [2]
Variant	F	F	F
Robot footprint	1,050 mm x 1,050 mm	1,050 mm x 1,050 mm	1,050 mm x 1,050 mm
Weight (excluding controller), approx.	2,385 kg	2,415 kg	2,421 kg



datasheet kuka

demostración de presición

1.11 Cinemática directa e inversa

Directa: * Conociendo la configuración del robot y el ángulo de cada articulación, procedemos a calcular la ubicación de la punta de la herramienta

Inversa: * Para una posición objetivo de la punta de la herramienta, calculamos el ángulo necesario en cada articulación

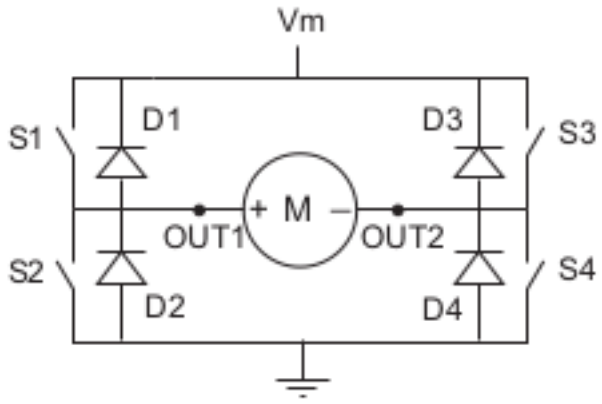
Ejemplos a desarrollar en el curso:

- Cinemática 2D con 2 grados de libertad + 1 actuador: permite, por ejemplo escribir en un plano
- Cinemática 3D con 3 grados de libertad: permite ubicar la punta de la herramienta en cualquier punto del espacio de trabajo
- Cinemática 3D con 6 grados de libertad: además de ubicar la posición también permite definir los ángulos de inclinación para la punta de la herramienta

termografía del fucionoamiento de un brazo robótico

1.12 Control de motores para robots

1.12.1 Control de un motor CC



Closed Switches	Voltage Across Moto
S1, S4	Positive (forward rota
S2, S3	Negative (reverse rota
S1, S3	Zero (short-circuit bra
S2, S4	Zero (short-circuit bra
None or one	Open circuit (coasti

MODE	PHASE	ENABLE	OUT1	OUT2	Function
1	x	0	L	L	Short-circuit braking (S2, S4 closed)
1	0	1	H	L	Forward (S1, S4 closed)
1	1	1	L	H	Reverse (S2, S3 closed)

Práctica de laboratorio: * acción con transitor + diodo anti-paralelo * pwm + transistor para controlar velocidades https://www.youtube.com/watch?v=m5JYkgCRbBI * puente H https://www.youtube.com/watch?v=fVgnUWIWzZ8 * práctica con un motor paso a paso

[ ]:

[ ]:

1.13 Algunos proyectos de código abierto y otros

contrucción de un robot de 7 ejes https://www.youtube.com/playlist?list=PL4njCTv7IRbyf3XfhUcp_jnkRrAu1U

impresion 3D de reductores https://www.youtube.com/watch?v=McfVmBNaNuA

1.13.1 Robot 3D, con tres grados de libertad

Cinemática directa:

Cinemática inversa

Para una posición [x,y,z]

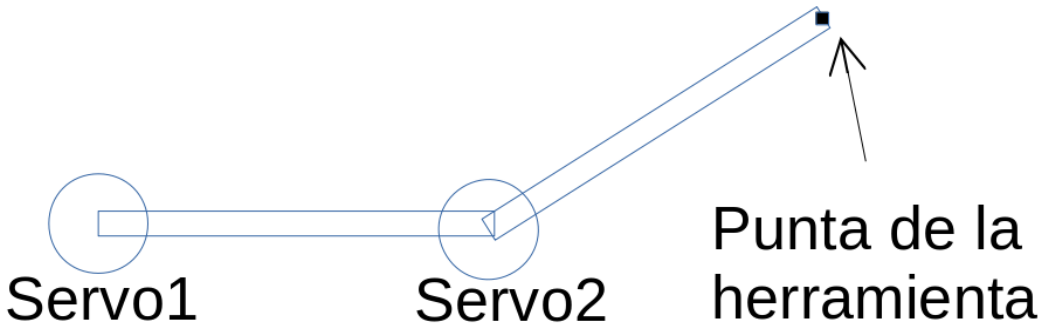
Primero, con la posición [x,Y], defino el ángulo del primer servo, es decir el de la base. Luego el problema se reduce al caso 2D donde quedan las otras dos articulaciones para llegar a la posición objetivo.

2 Control de dispositivos Robóticos

Veamos un ejemplo simple de un brazo robótico simple, compuesto por 2 servomotores sg-90 que pueden tomar posiciones de 0 a 180°, variando de a 1 grado a la vez.

Estos se unirán mediante piezas plásticas de 10 cm y en la "punta de la herramienta" se coloca un lápiz.

Como se observa en el diagrama. Ambos ejes de rotación son perpendiculares a la mesa donde se coloca el dispositivo. El servo1 está fijado a la mesa y mueve una pieza adherida al servo2, mientras que este último mueve a otra pieza que lleva la "punta de la herramienta"



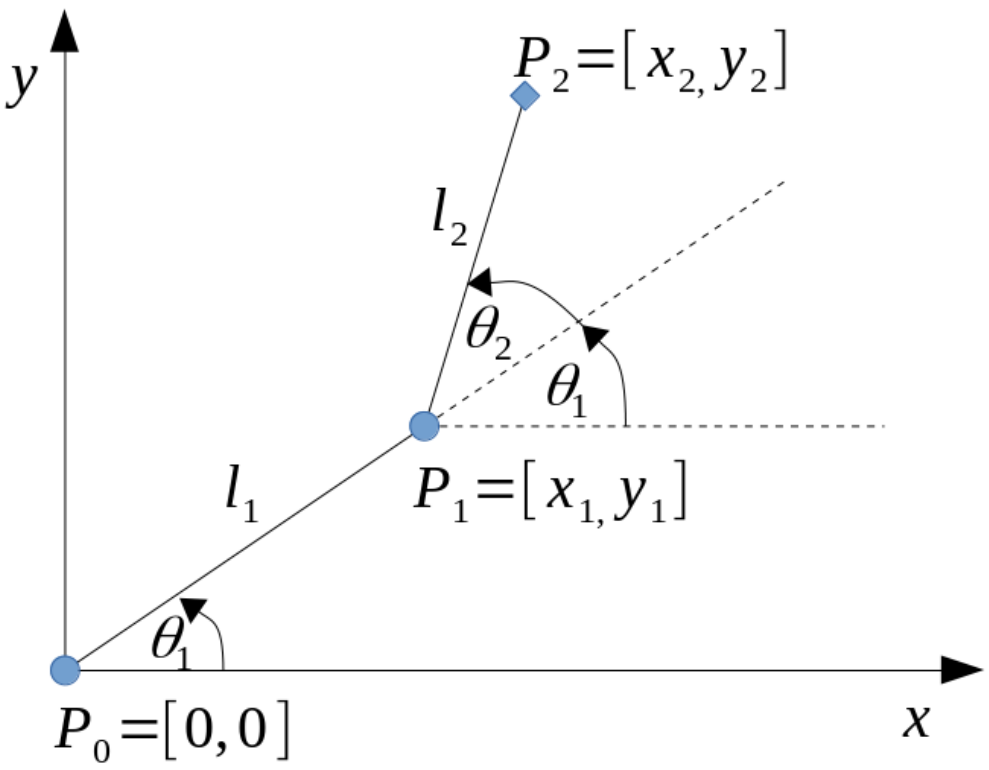
2.1 Cinemática directa 2D

A la hora de controlar este dispositivo podemos variar los ángulos θ_1, θ_2 de los servos. Nos interesa saber en qué posición queda la punta de la herramienta ante estos cambios. El cálculo de esa posición final en función de esas coordenadas generalizada variables se conoce como **cinemática directa**

Comenzaremos viendo el caso más simple para un desplazamiento en 2D con solo 2 articulaciones rotatorias, y en próximas secciones veremos un método que se puede generalizar para problemas más complejos en 3D con cualquier cantidad de articulariones

En el diagrama se muestan:

- servo 1: círculo en $P_0 = [0, 0]$ adherido a un brazo de longitud l_1 y varia su ángulo θ_1
- servo 2: círculo en $P_1 = [x_1, x_2]$ adherido a un brazo de longitud l_2 y varia su ángulo θ_2
- punta de la herramienta: en $P_2 = [x_2, y_2]$



Vemos que el ángulo θ_1 es el mismo que forma el eje x con la línea de trazos que representa la posición de brazo 2 cuando $\theta_2 = 0$, por lo que el ángulo entre el brazo 2 y el eje x es igual a $\theta 1 + \theta 2$

Con estos datos podemos determinar las posiciones P_1 y P_2, en función de los ángulos que indiquemos a los servomotores θ_1 y θ_2

Para el segundo servo, P_1:

$$x_1 = l_1.\cos(\theta_1) \qquad y_1 = l_1.\sin(\theta_1)$$

Para la punta de la herramienta P_2:

$$x_2 = l_1.\cos(\theta_1) + l_2.\cos(\theta_1 + \theta_2) \qquad y_2 = l_1.\sin(\theta_1) + l_2.\sin(\theta_1 + \theta_2)$$

2.2 Espacio de las configuraciones posibles

Nos interesa es conocer el área efectiva en la cual se puede posicionar la punta de la herramienta

En este caso los servomotores pueden tener valores de giro en grados que son discretos, es decir valores enteros. Por los que esta área estará definida por una colección de puntos en el espacio que constituyen las configuraciones posibles del dispositivo.

Entonces para graficar el espacio de las configuraciones posibles tenemos que obtener los puntos de las distintas configuraciones posibles, con $0 \leq \theta_1 \leq 180$, y $0 \leq \theta_2 \leq 180$

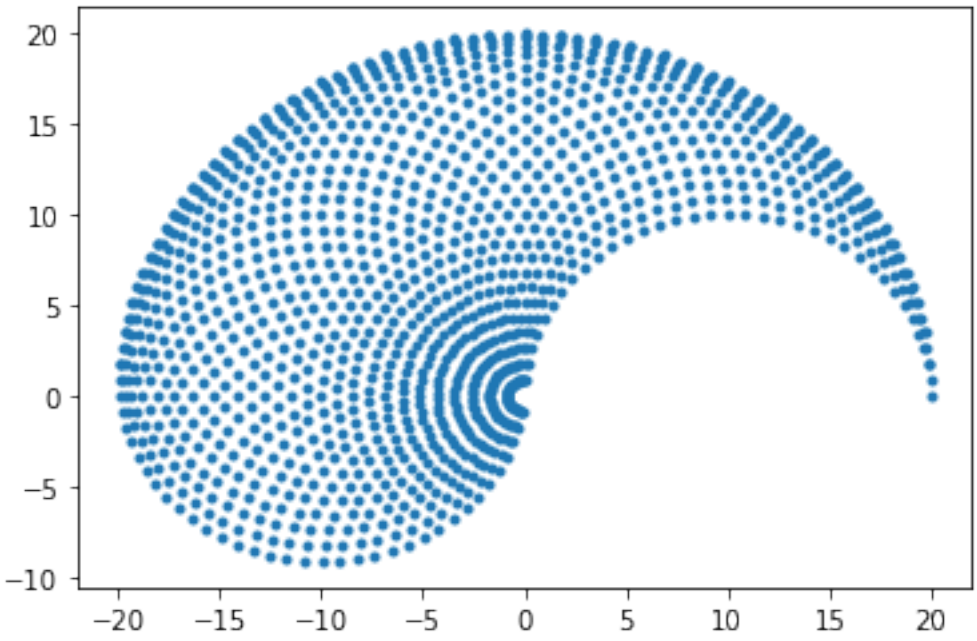
En el caso de querer tener todas las posiciones para los servos descriptos, que varían de a 1° tendríamos $180^2 = 32400$ puntos en total. En el siguiente ejemplo variaremos de a 5° de modo que cada servo tendra 180°/5°=36 posiciones con un total de 1296 puntos obtenidos.

[1]:
```
from numpy import *

# para un par de valores de ángulos devuelve la posición de la punta [x,y]
def X(th_1, th_2 , l1=10, l2=10):
    th1=th_1*pi/180                        # convierto a radianes
    th2=th_2*pi/180

    x2 = l1*cos(th1) + l2* cos(th1+th2)
    y2 = l1*sin(th1) + l2* sin(th1+th2)
    return(x2,y2)

X(90,90)
```

[1]:
```
(-10.0, 10.000000000000002)
```

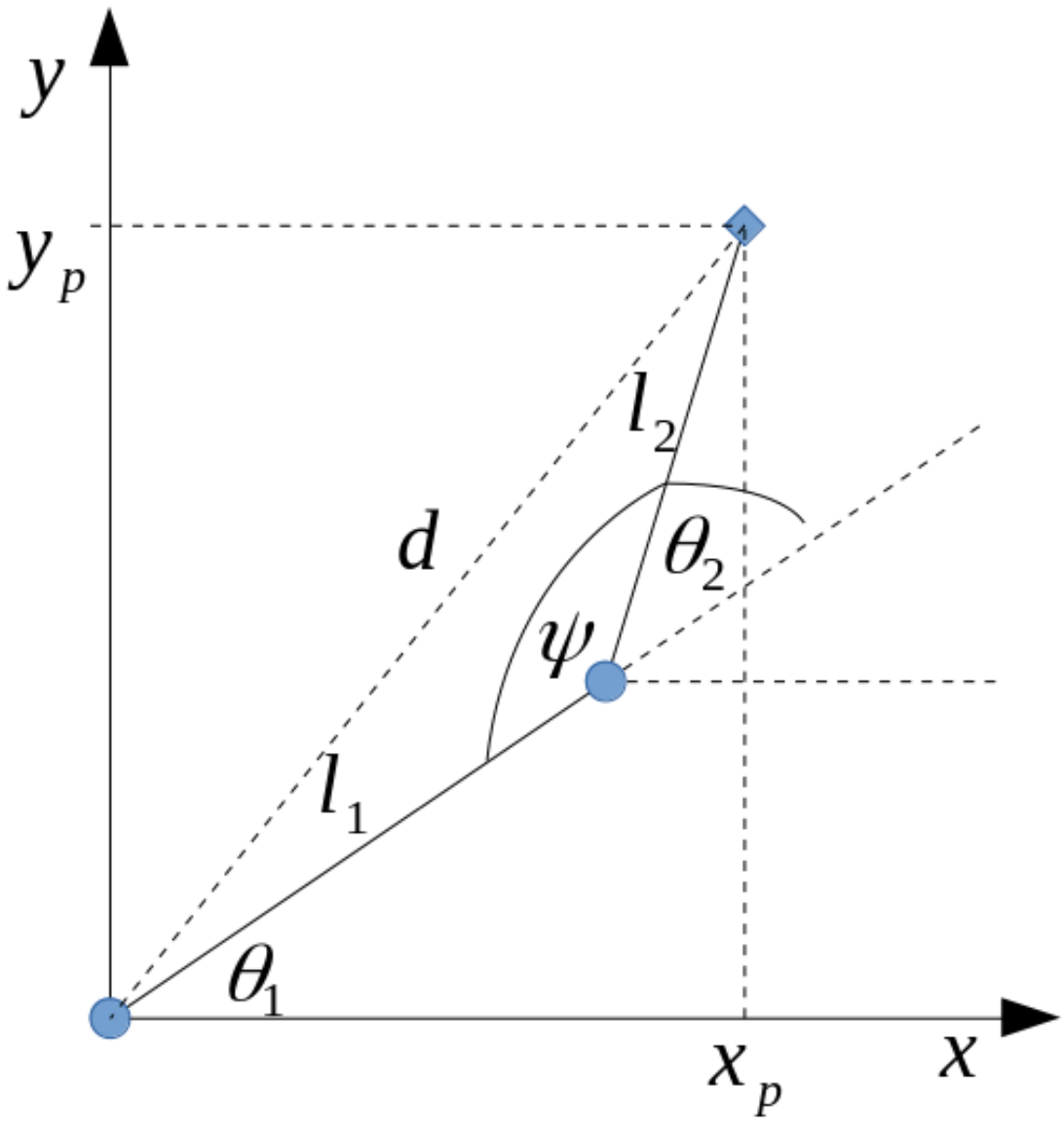
[2]:
```
from matplotlib import pyplot as plt
%matplotlib inline


#matriz para los puntos calculados, cantidad de puntos: (180/5)*(180/5)=1296
a=zeros([1296,2])

num=0
for th_1 in range(0,180,5):      # para todas las posiciones del ángulo theta_1
    for th_2 in range(0,180,5): # para todas las posiciones del ángulo theta_2
        x2, y2 = X(th_1, th_2)

        a[num]=[x2,y2]
        num=num+1

fig=plt.figure()
plt.plot(a[:,0], a[:,1],'.')        # ejes x= a[:,0] (1°col) ,y=a[:,1] (2°col)
fig.show()
```

```
/home/user/anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:18:
UserWarning: Matplotlib is currently using
module://ipykernel.pylab.backend_inline, which is a non-GUI backend, so cannot
show the figure.
```



2.3 Cinemática inversa 2D

A la hora de controlar un brazo robótico, nos interesa saber qué valores debe girar cada servomotor para llevar la punta de la herramienta a un ubicación objetivo. Esto se conoce como **cinemática inversa** y es justamente lo que permite controlar el movimiento del brazo para seguir una trayectoria de desplazamiento determinada.



Del diagrama observamos

$$\psi = 180 - \theta_2 \tag{1}$$

Por el teorema del coseno tenemos

$$d^2 = l_1^2 + l_2^2 - 2.l_1.l_2.cos(\psi) \tag{2}$$

Considerando los valores de la punta de la herramienta x_p, y_p y usando el teoréma de Pitagoras, tenemos:

$$d^2 = x_p^2 + y_p^2 \tag{3}$$

Si reemplazamos (2) en (3) y despejamos $\cos(\psi)$ tenemos:

$$\cos(\psi) = \frac{x_p^2 + y_p^2 - l_1^2 - l_2^2}{-2.l_1.l_2} \tag{4}$$

Para que la expresión sea más clara, creamos una variable que reemplace la expresión anterios:

$$C = \frac{x_p^2 + y_p^2 - l_1^2 - l_2^2}{-2.l_1^2.l_2^2} \qquad \Rightarrow \qquad \cos(\psi) = C \tag{5}$$

Usaremos esta conocida relación trigonométrica, y despejamos $\sin(\psi)$

$$1 = \cos^2(\psi) + \sin^2(\psi) \qquad \Rightarrow \qquad \sin(\psi) = \pm\sqrt{1 - \cos^2(\psi)} \tag{6}$$

reemplazando (5) en (6)

$$\sin(\psi) = \pm\sqrt{1 - C^2} \tag{7}$$

Usaremos la definición de la función tangente:

$$\tan(\psi) = \frac{\sin(\psi)}{\cos(\psi)} \qquad \Rightarrow \qquad \psi = atan\left(\frac{\sin(\psi)}{\cos(\psi)}\right) \tag{8}$$

Atención: dados valores de x, y la función arcotangente, no reconoce la diferencia entre un ángulo del primer cuadrante como $\pi/4$ y uno del tercer cuadrante como $-3\pi/4$

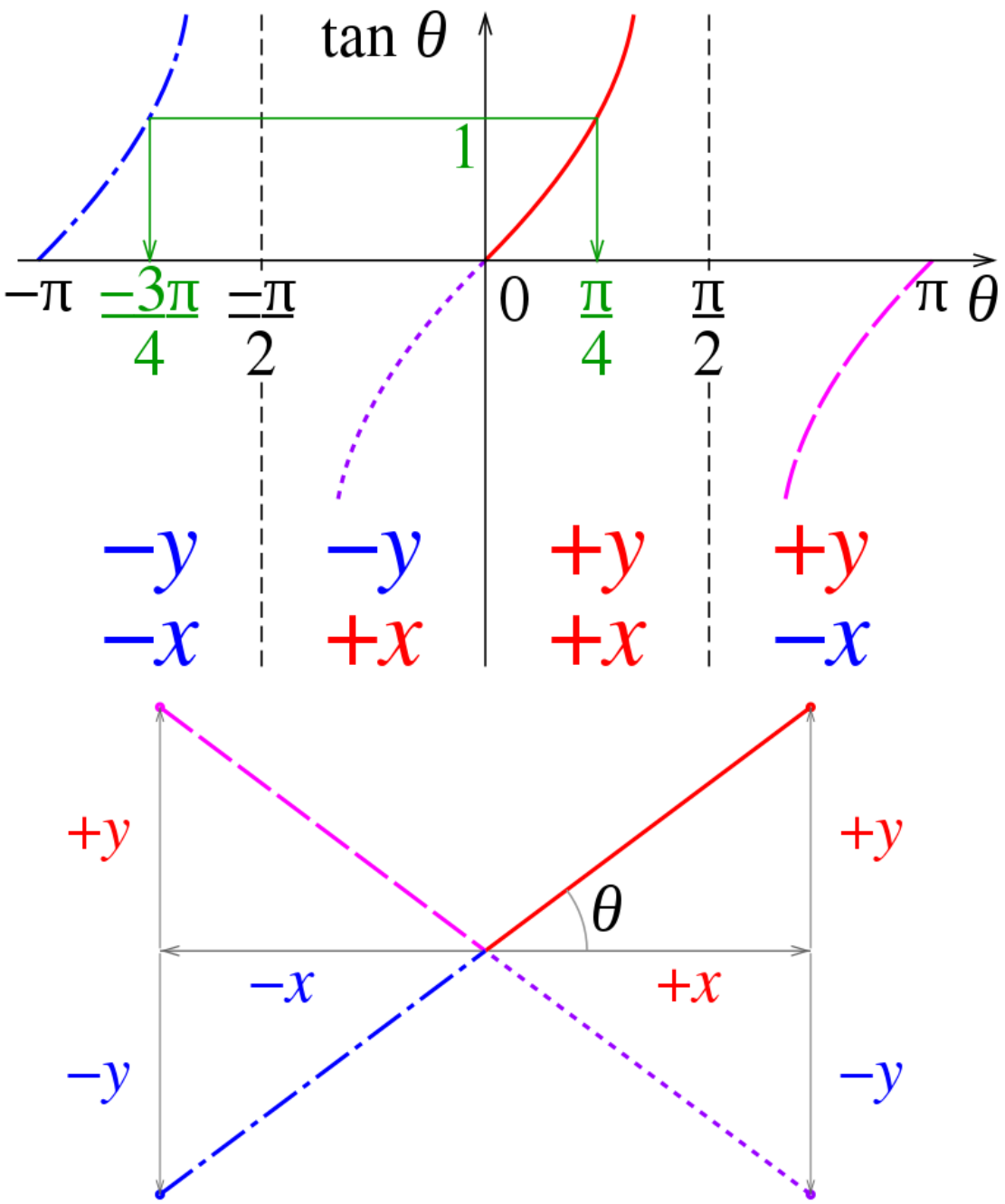


Gráfico de la funciónn tangente By Cmglee - Own work, CC BY-SA 4.0, https://commons.wikimedia.org/w/index.php?curid=51647820

En lugar de $arctan(y/x)$ usaremos la función $atan2(y, x)$ que está disponible en la mayoría de los lenguajes de programación modernos y permite determinar el ángulo real, ademas tambíen sirve si el ángulo es $\pi/2$.

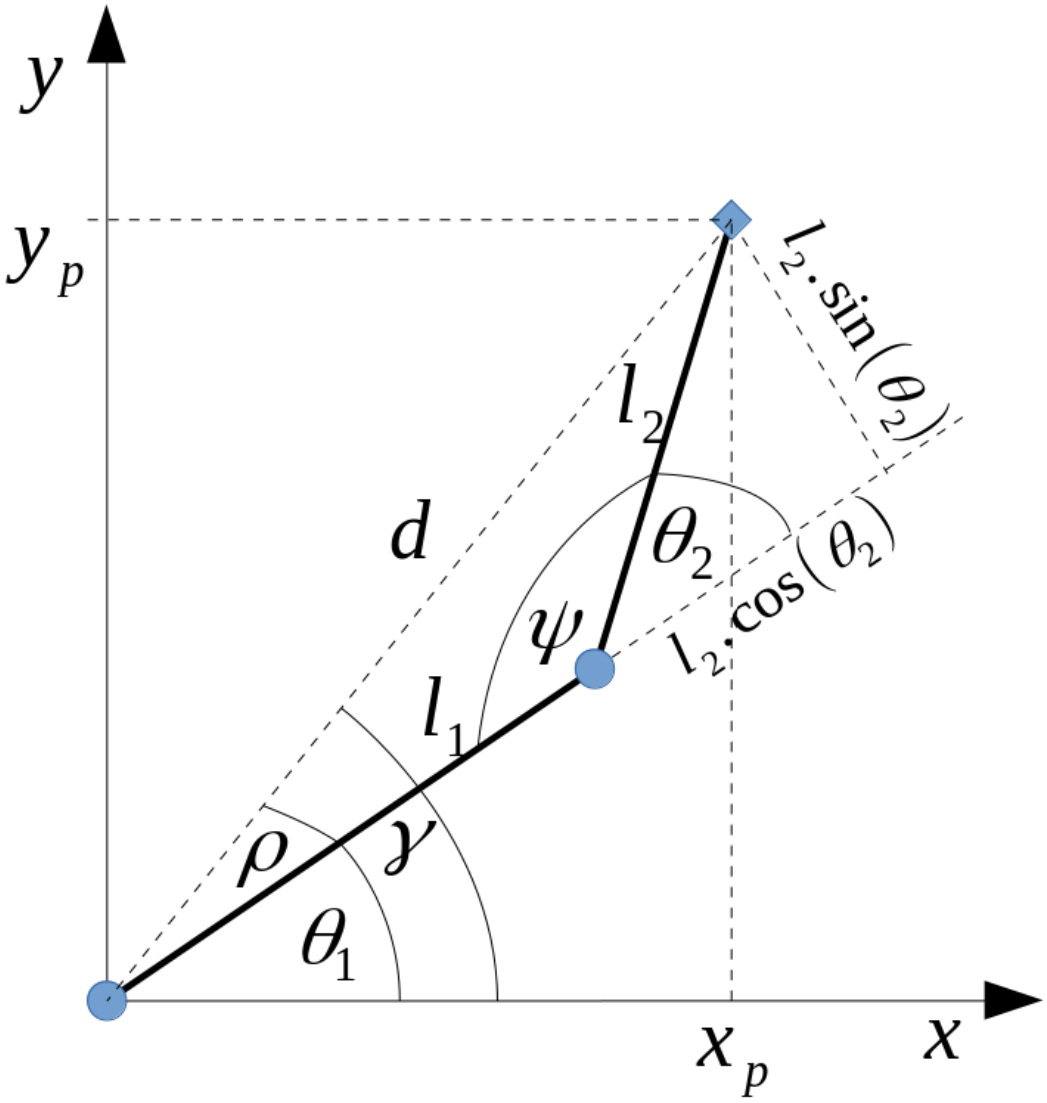
Cuidado: en algunos lenguajes está definida como $atan2(x, y)$, primero el valor en x, luego en y, por lo que siempre es conveniente revisar la documentación de la función.

Por lo tanto usando $atan2$ y reemplazando (7) y (5)

$$\psi = atan2(\sin(\psi), \cos(\psi)) \quad = \quad atan2(\pm\sqrt{1-C^2}, C) \tag{9}$$

reemplazando (9) en (1) tenemos :

$$\theta_2 = 180 - atan2(\pm\sqrt{1-C^2}, C) \tag{10}$$



Si, además, consideramos los ángulos ρ, γ, que se muestran en el diagrama, tenemos:

$$\theta_1 = \gamma - \rho \tag{11}$$

$$\gamma = atan2(x_p, y_p) \tag{12}$$

$$\rho = atan2(\ l_1 + l_2 cos(\theta_2) \quad , \quad l_2 \sin(\theta_2)) \tag{13}$$

Reemplazando (12) y (13) en (11) finalmente tenemos:

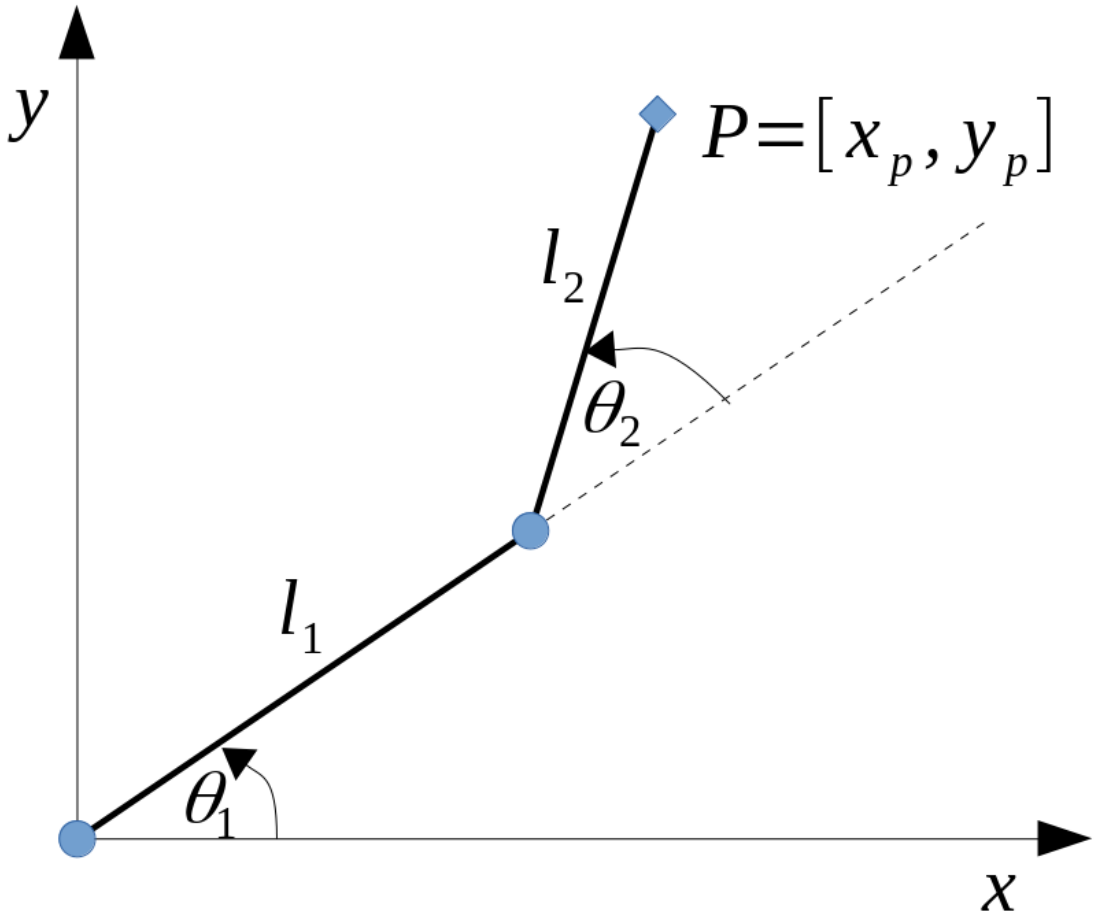
$$\theta_1 = atan2(\ y_p, x_p) \ - \ atan2(\ l_2 \sin(\theta_2) \quad , \quad l_1 + l_2 cos(\theta_2)) \tag{14}$$

2.3.1 Resumen de ecuaciones de cinemática inversa 2D

$$C = \frac{x_p^2 + y_p^2 - l_1^2 - l_2^2}{-2.l_1.l_2} \qquad \text{(ver (5))}$$

$$\theta_2 = 180 - atan2(\pm\sqrt{1 - C^2}, C) \qquad \text{(ver (10))}$$

$$\theta_1 = atan2(\ p_p, x_p) \ - \ atan2(\ l_1 + l_2 cos(\theta_2) \quad , \quad l_2 \sin(\theta_2)) \qquad \text{(ver (14))}$$



- Puede llegar al punto con dos configuraciones por el signo $\pm$ en (10)
- θ_1 no solo depende de $[x_p, y_p]$ sino también del valor de θ_2

2.4 Definir una trayectoria

Podemos definir cualquier trayectoria para la punta de la herramienta dentro del espacio de las configuraciones posibles.

Para esto usaremos las ecuaciones (5), (10) (14), para crear una función que, a partir de una lista de valores $[x_p, y_p]$, que pertenecen al espacio de las configuraciones posibles, nos devuelva una lista de valores $[\theta_1, \theta_2]$

$$[\theta_1, \theta_2] = f(x_p, y_p)$$

[21]:
```
from numpy import *

def inversa2D( xp, yp, l1=10, l2=10 ):
    C = (xp**2 + yp**2 - l1**2 -l2**2 ) / (-2* l1 * l2)

    th2 = pi - arctan2(sqrt(1-C**2),C )
    th1 = arctan2(yp, xp) - arctan2(  l2*sigimn(th2), l1+l2*cos(th2))

    # a cada ángulo le sumo 2pi y dejo el resto de dividir por 2pi
    th1 = (th1+ 2 * pi) % (2 * pi)  # convierte ángulos negativos
    th2 = (th2+ 2 * pi) % (2 * pi)  # ej -5/4*pi -> 3/4* pi

    th1 = th1*180/pi
    th2 = th2*180/pi

    return ( th1, th2)

inversa2D(-10,0)  #ejemplo
```

[21]:
```
(119.99999999999997, 120.00000000000006)
```

Esta función está pensada para usar de forma optimizada tomando como argumentos de entrada vectores con los valores de x, y. En el siguiente ejemplo se da como ingreso los puntos para recorrer una trayectoria recta y se obtiene como salida los vectores con los valores de θ_1, θ_2 necesarios para esto.

[23]:
```
x=linspace(-19,-1,10)
y=linspace(0,0,10)

th1, th2 = inversa2D(x,y)
th1, th2
```

[23]:
```
(array([161.80512766, 148.21166938, 138.59037789, 130.54160187,
        123.36701297, 116.74368395, 110.48731511, 104.47751219,
         98.62692656,  92.86598398]),
 array([ 36.38974468,  63.57666123,  82.81924422,  98.91679625,
        113.26597406, 126.5126321 , 139.02536977, 151.04497563,
        162.74614688, 174.26803203]))
```

Para el caso descripto inicialmente, donde los servomotores varían de a 1°, necesitamos valores enteros de θ_1, θ_2. Podemos convertirlos como se muestra a continuación.

[24]:
```
th1, th2 = th1.astype(int), th2.astype(int)
th1, th2
```

[24]:
```
(array([161, 148, 138, 130, 123, 116, 110, 104,  98,  92]),
 array([ 36,  63,  82,  98, 113, 126, 139, 151, 162, 174]))
```

En muchos casos nos sirve guardar esta información en un archivo tipo .csv para luego pasársela a los servomotores como en ejemplos de unidades anteriores.

En el siguiente código se muestra una forma simple de generar este tipo de archivos

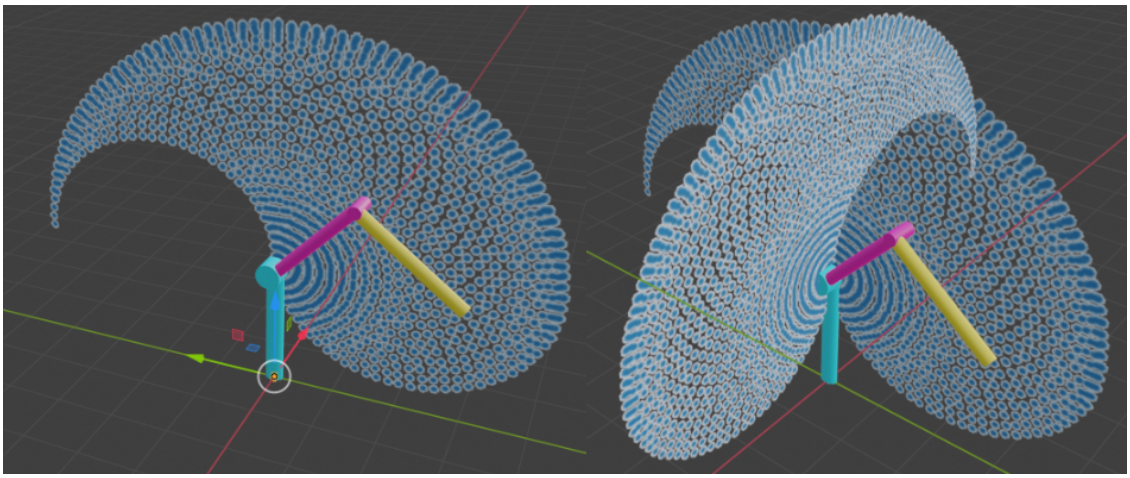
[25]:
```
angulos = array([th1,th2],dtype=int).T
savetxt("ejemplo.csv", angulos, delimiter=",",fmt='%0.0f')
```

[26]:
```
!cat ejemplo.csv
```

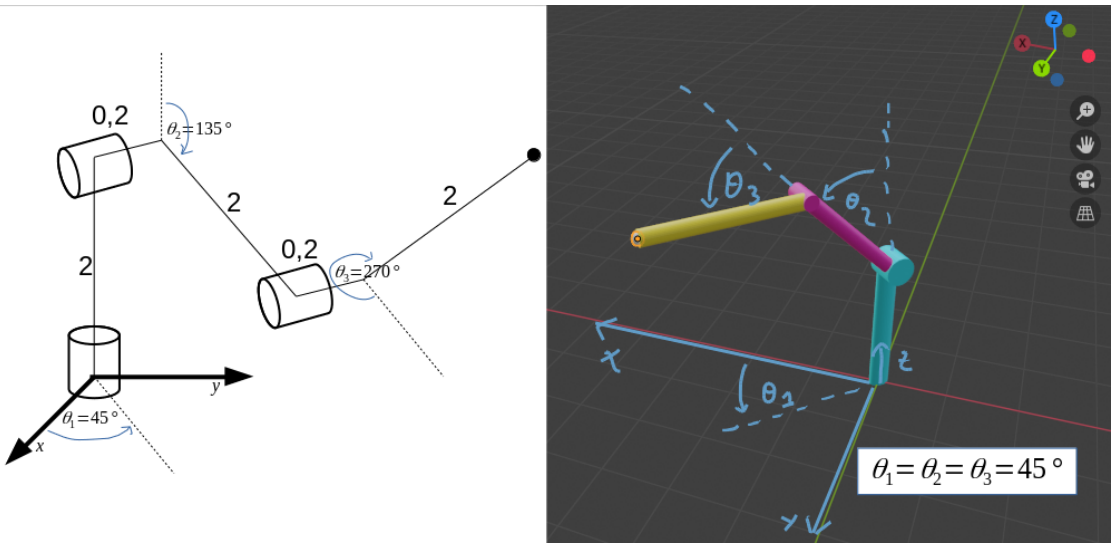
```
161,36
148,63
138,82
130,98
123,113
116,126
110,139
104,151
98,162
92,174
```

2.4.1 Cinemática inversa para un brazo RRR 3D

Observe que a la configuración que tenemos en este caso es equivalente a tener el mismo caso para anterior con un plano 2D, pero ahora es perpendicular al piso y es rotado por la acción del servomotor 1.



En el Brazo de la figura, cada una de las articulaciones rotatorias puede tomar una posición entre 0 y 360°, (cambiando de a 1°), por lo que el espacio de trabajo se puede simplificar considerando una esfera. Determine la ecuación para conocer todos los puntos dentro de este espacio.

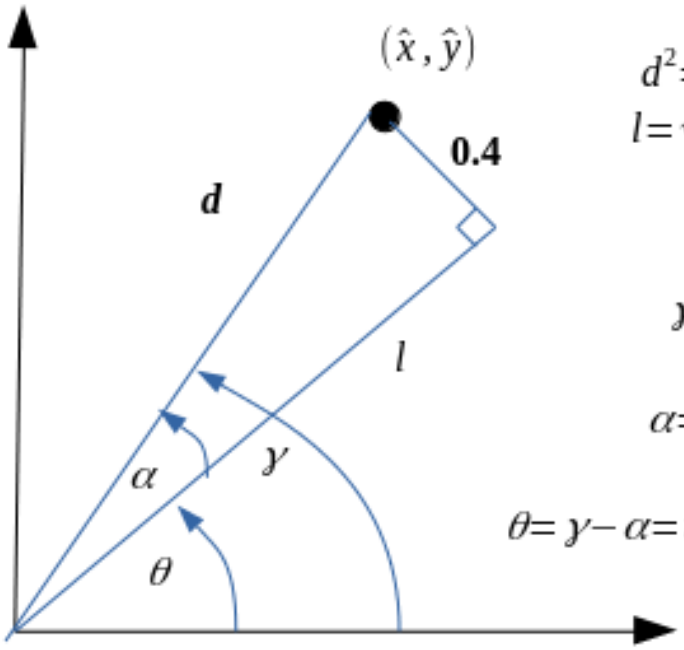


Procedimiento para el cálculo de $(\theta_1, \theta_2, \theta_3)$ a partir de valores dados para (x, y, z): * Determinamos la rotación del servo de la base θ_1, para llegar a una posición (x, y), entonces $\theta_1 = atan2(x, y)$ * redefinimos $(y, z) \rightarrow (\hat{x}\ ,\ \hat{y})$, luego el problema se reduce un plano donde se debe encontrar los valores (θ_2, θ_3) para llegar a $(\hat{x}, \hat{y})$.

es necesario tener en cuenta que el nuevo plano está centrado en la articulación 2 por lo que $\hat{y} = z - 2$

Para obtener $\hat{x}$ uno puede verse tentado a plantear simplemente la proyección del brazo sobre el plano xy : $d = \sqrt{y^2 + x^2} = \hat{x}$

pero eso no tiene en cuenta que el brazo está ligeramente desplazado hacia un lado por el espacio de las articulaciones. Debo ajustar ese desplazamiento para obtener los verdaderos valores de θ, l.



$$d^2 = 0.4^2 + l^2$$
$$l = \sqrt{d^2 - 0.4^2}$$
$$\gamma = tg^{-1}\left(\frac{y}{x}\right)$$
$$\alpha = \sin^{-1}\left(\frac{0.4}{d}\right)$$
$$\theta = \gamma - \alpha = tg^{-1}\left(\frac{y}{x}\right) - \sin^{-1}\left(\frac{0.4}{d}\right)$$

Note que para re utilizar los cálculos del modelo 2D, nuestro plano pasa por l apuntando con un ángulo θ. A pesar de no apuntar directamente al punto objetivo se llega por efecto del desplazamiento de las articulaciones.

Se desea realizar una trayectoria que una

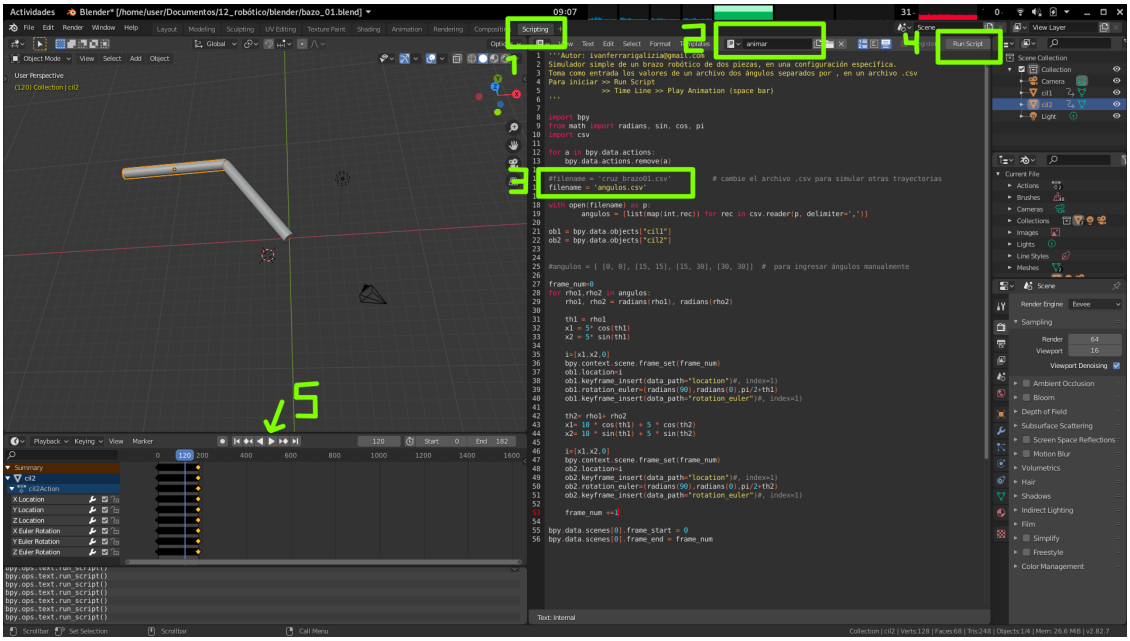
2.4.2 Simular cinética de brazo robótico

Para simular la trayectoria del brazo que acabos de calcular existen varios simuladores. En este caso vamos a usar Blender que es un programa para animaciones ditales. Lo primero es descargar Blender en https://www.blender.org/download/, una vez instalado pueden descargar y abrir el archivo *brazo_01.blend* que pueden descargar en https://github.com/ieferrari/Lab1/archive/master.zip

Una vez abierto el archivo brazo_01.blend:

- seleccionar la pestaña srcripting
- seleccionar el script "animar"
- indicar el nombre del archivo .csv , si está en una carpeta distinta a la que ejecuta el programa deben dar el path completo. En la carpeta se incluyen angulos.csv y cruz_brazo1.csv que pueden usar.
- clic en Run Script, lee el archivo .csv y por cada par [x,y] genera un cuadro en la animación
- clic en play animation

Es importante que el archivo .csv no tenga líneas en blanco. Con este simulador sencillo pueden verificar si los la trayectoria calculada es correcta.



Bibliografía

[1] F. C. P. K. M. Lynch, *Modern Robotics: Mechanics, Planning, and Control*. Cambridge University Press, 2017.

[2] R. Nagarajan, *Introduction to Industrial Robotics*. Pearson Education India, 2016.

[3] J. J.Craig, *Introduction to Robotics Mechanics and Control 3rd edition*. Pearson Education, Inc., 3 ed., 2005.

[4] J. J. Craig, *Introduction to robotics : mechanics and control*. Pearson Higher Education, 3rd international ed. ed., 2014.