Física II - Grado en Robótica (EPSE-USC Lugo)

Mecánica de Robots: Manipulabilidad y Estática de Robots

Ángel Piñeiro

E-mail: Angel.Pineiro@usc.es
Depto de Física Aplicada
Universidade de Santiago de Compostela

8 de abril de 2024

Resumen

Introducción

Consideremos el caso en el que el elemento terminal está representado por un número de coordenadas mínimo $x \in \mathbb{R}^m$. La velocidad vendrá dada por $\dot{x} = dx/dt \in \mathbb{R}^m$. En este caso la cinemática directa puede escribirse como: $x(t) = f(\theta(t))$, donde $\theta \in \mathbb{R}^m$ son las coordenadas de las articulaciones.

$$\Rightarrow \dot{x} = \frac{\partial f(\theta)}{\partial \theta} \frac{d\theta(t)}{dt} = \frac{\partial f(\theta)}{\partial \theta} \dot{\theta} = J(\theta)\dot{\theta}$$

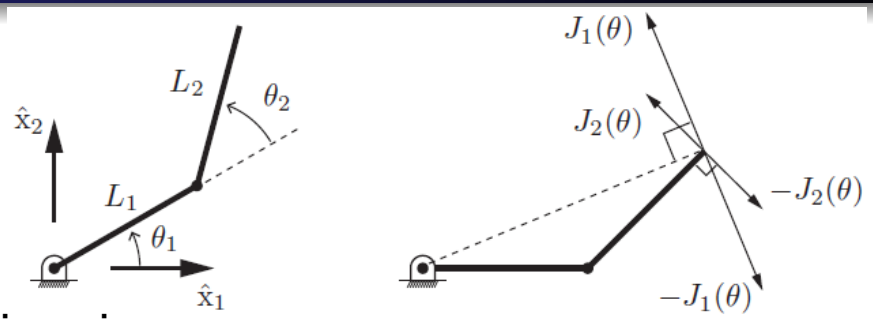
donde $J(\theta) \in \mathbb{R}^{m \times n}$ se llama **Matriz Jacobiana**.

La matriz Jacobiana representa la sensibilidad lineal de la velocidad del elemento terminal a la velocidad de las articulaciones $\dot{\theta}$ y es función de las coordenadas de las articulaciones.

Introducción

En representación explícita:

$$x_1 = L_1 \cos(\theta_1) + L_2 \cos(\theta_1 + \theta_2)$$
$$x_2 = L_1 \operatorname{sen}(\theta_1) + L_2 \operatorname{sen}(\theta_1 + \theta_2)$$



$$\dot{x}_1 = -L_1\dot{\theta}_1 \operatorname{sen}(\theta_1) - L_2(\dot{\theta}_1 + \dot{\theta}_2)\operatorname{sen}(\theta_1 + \theta_2)$$
$$\dot{x}_2 = L_1\dot{\theta}_1 \cos(\theta_1) + L_2(\dot{\theta}_1 + \dot{\theta}_2)\cos(\theta_1 + \theta_2)$$

Esta última ecuación puede escribirse en forma matricial:

$$\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\end{bmatrix} = \begin{bmatrix}-L_1 \operatorname{sen}(\theta_1) - L_2 \operatorname{sen}(\theta_1 + \theta_2) & -L_2 \operatorname{sen}(\theta_1 + \theta_2)\\ L_1 \cos(\theta_1) + L_2 \cos(\theta_1 + \theta_2) & L_2 \cos(\theta_1 + \theta_2)\end{bmatrix}\begin{bmatrix}\dot{\theta}_1\\ \dot{\theta}_2\end{bmatrix} \Rightarrow \boxed{\dot{x} = J(\theta)\dot{\theta}}$$

También podemos escribirlo en forma vectorial: $\boxed{v = J_1(\theta)\dot{\theta}_1 + J_2(\theta)\dot{\theta}_2}$

Si $J_1(\theta)$ y $J_2(\theta)$ no son paralelos se puede generar una velocidad del elemento terminal en cq dirección del plano, escogiendo valores apropiados de $\dot{\theta}_1$ y $\dot{\theta}_2$.

Si $\theta_2 = 0$ o $\theta_2 = \pi \Rightarrow J_1(\theta)$ y $J_2(\theta)$ serán paralelos independientemente del valor de $\theta_1 \Rightarrow$ la Matriz Jacobiana es Singular $\Rightarrow$ **Configuración Singular o Singularidad** $\Rightarrow$ el robot no puede generar velocidades en ciertas direcciones.

Configuración Singular

Demuestra que para un robot plano 2R: $\theta_2 = 0 \Rightarrow J_1(\theta) = \frac{L_1+L_2}{L_2} J_2(\theta)$

Introducción

Código para calcular las configuraciones singulares de un robot 2R plano:

```
import sympy as sp

t1=sp.var("t1")
t2=sp.var("t2")
L1=sp.var("L1")
L2=sp.var("L2")

a=-L1*sp.sin(t1)-L2*sp.sin(t1+t2)
b=L1*sp.cos(t1)+L2*sp.cos(t1+t2)
c=-L2*sp.sin(t1+t2)
d=L2*sp.cos(t1+t2)
```

```
m=sp.Matrix([[a,c],[b,d]])

sp.solve(m.det())
```

Resultado general:

```
[{L1: 0},
 {L2: 0},
 {t2: -
t1 - 2*atan(1/tan(t1/2))},
 {t2: -
t1 + 2*atan(tan(t1/2))}]
```

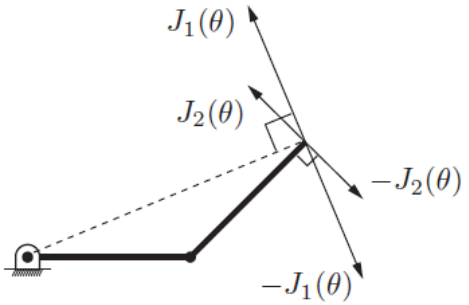
Ejemplo:

```
sp.solve(m.subs({t1:0, L1:1, L2:1}).det())
[0, pi]
```

Introducción

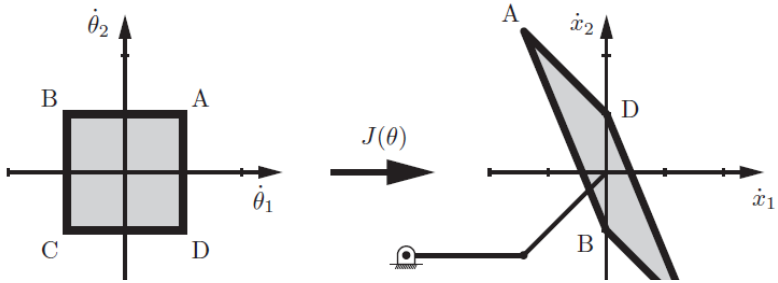
Escribimos la matriz Jacobiana para 2 configuraciones diferentes del robot (con $L_1 = L_2 = 1$):

$$J\begin{bmatrix}0\\ \pi/4\end{bmatrix} = \begin{bmatrix}-0.71 & -0.71\\ 1.71 & 0.71\end{bmatrix}; \quad J\begin{bmatrix}0\\ 3\pi/4\end{bmatrix} = \begin{bmatrix}-0.71 & -0.71\\ 0.29 & -0.71\end{bmatrix}$$

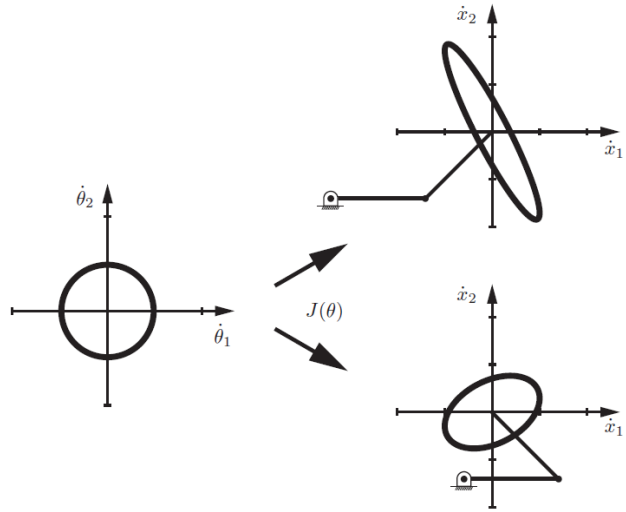


La columna i de la matriz Jacobiana representa la velocidad del elemento terminal para $\dot{\theta}_i = 1$ y $\dot{\theta}_j = 0 \ \forall j \neq i$.

La matriz Jacobiana puede utilizarse para mapear los límites en la velocidad rotacional de las articulaciones sobre las componentes de la velocidad del elemento terminal $\dot{\theta} \rightarrow \dot{x}$



Podemos mapear un círculo unitario de velocidades de las articulaciones. Este círculo se transforma, a través de la matriz Jacobiana, en una elipse en el espacio de las componentes de la velocidad del elemento terminal $\rightarrow$ **Elipsoide de Manipulabilidad**, que depende de la configuración del robot. Al acercarnos a una **singularidad** la elipse colapsa y se transforma en un segmento $\Rightarrow$ hay una dirección en la que el elemento terminal no puede moverse.



Introducción

El elipsoide de manipulabilidad nos sirve para saber qué tan cerca estamos de una singularidad: comparando las longitudes de los semiejes. Cuanto más cerca estamos de un círculo más fácilmente es mover el elemento terminal en cualquier dirección y más lejos estamos de una singularidad.

La matriz Jacobiana es también importante en análisis de estática: si aplicamos una fuerza externa al elemento terminal, ¿ qué fuerzas hay que aplicar en las articulaciones para compensar esa fuerza?

Trabajo realizado por el elemento terminal = producto de la fuerza por el desplazamiento: $W = f_n^T x$

Potencia = energía por unidad de tiempo, derivando $W \Rightarrow P = f_n^T \dot{x}$

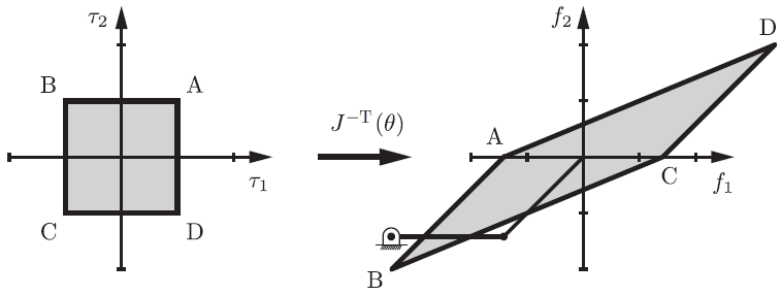
Si asumimos que la potencia aplicada sobre las articulaciones es igual a la potencia medida en el elemento terminal: $\Rightarrow \boxed{f_n^T v_n = \tau^T \dot{\theta}}$ ya que $v_n = J(\theta)\dot{\theta} \Rightarrow \tau = J^T(\theta) f_n$.

Así podemos calcular el torque en las articulaciones a partir de la fuerza medida en el elemento terminal y de la matriz Jacobiana.

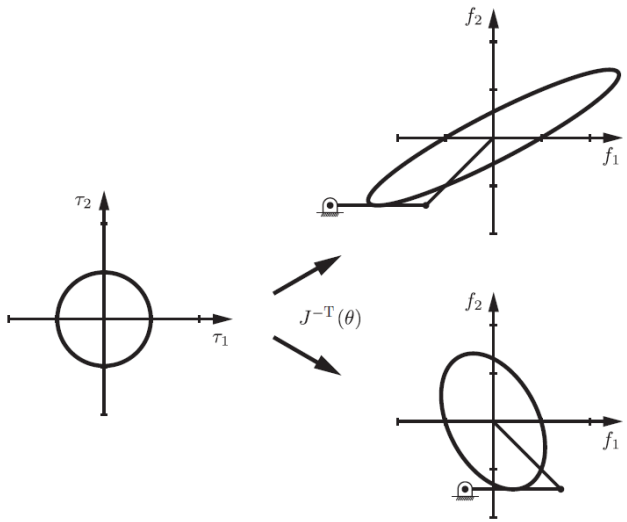
Si la matriz Jacobiana y su traspuesta son invertibles, podemos calcular qué torque es necesario aplicar para generar una determinada fuerza en el elemento terminal (los torques en las articulaciones necesarios para que el elemento terminal empuje contra una pared con una fuerza normal predeterminada): $f_n = J^{-T}(\theta)\tau$.

Introducción

Para una configuración dada del robot θ y unos torques en las articulaciones $\tau_1, \tau_2 \in [-1, 1]\, N\, m$, se generan unas fuerzas en el elemento terminal dadas por el paralelogramo de la figura.



Igual que hay una equivalencia entre velocidades de articulaciones y velocidad del elemento terminal ($\Rightarrow$ Elipsoide de Manipulabilidad), hay también una equivalencia entre torques en las articulaciones y fuerza en el elemento terminal $\Rightarrow$ **Elipsoide de Fuerza**.



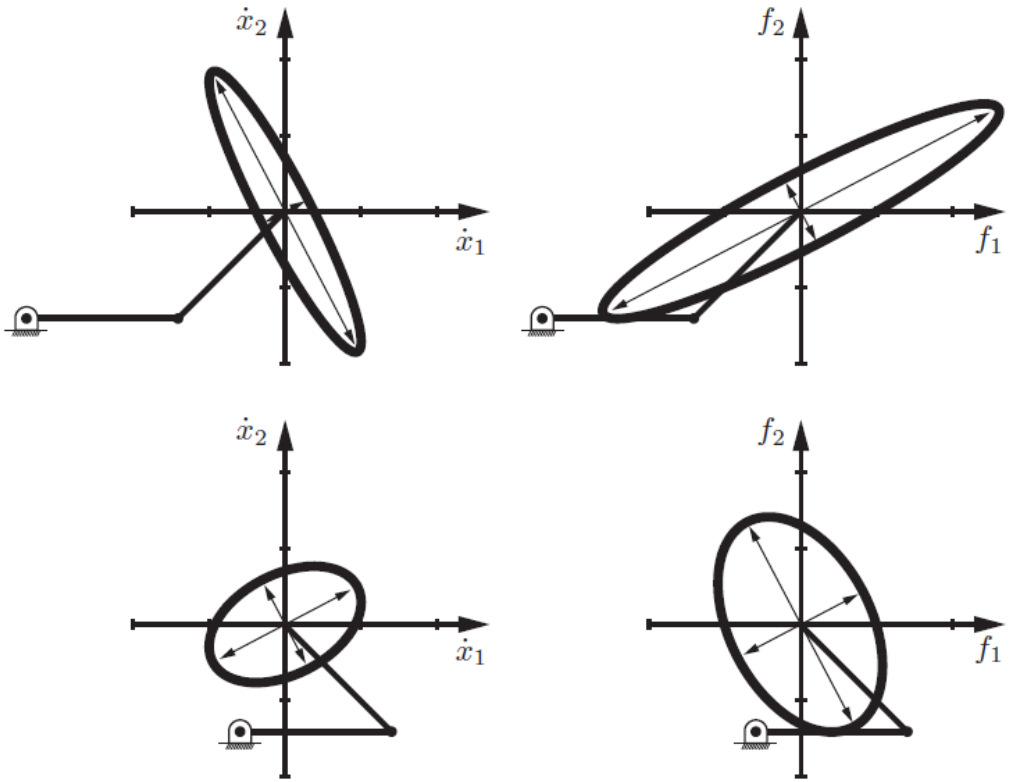
El elipsoide de fuerza indica la facilidad con la que el robot puede generar fuerzas en diferentes direcciones. Las direcciones en las que el elemento terminal se mueve rápido son aquellas en las que es difícil ejercer fuerza y viceversa.

Para una configuración dada, los elipsoides de fuerza y manipulabilidad están alineados y las longitudes de sus ejes principales tienen dimensiones recíprocas.

Introducción

En una singularidad, el elipsoide de manipulabilidad colapsa en un segmento lineal mientras que el elipsoide de fuerza es infinitamente largo en la dirección perpendicular.

En una singularidad se pueden soportar fuerzas virtualmente infinitas, es la estructura del robot el que soporta las fuerzas, no las articulaciones.



Ejercicios

- Desarrolla un código en Python que genere las gráficas correspondientes a los elipsoides de manipulabilidad y de fuerza para una configuración singular del robot 2R plano y para las configuraciones con coordenadas de articulaciones $(0, \pi/4)$ y $(0, 3\pi/4)$.

Matriz Jacobiana

En el ejemplo del robot 2R vimos que: $v_n = J_1(\theta)\dot{\theta}_1 + J_2(\theta)\dot{\theta}_2 \Rightarrow v_n = J(\theta)\dot{\theta}$.

En el caso más general, v_n es un vector *Giro* $\nu \in \mathbb{R}^6 \Rightarrow J_i(\theta)$ (componente i de la matriz $J(\theta)$) es el vector *Giro* correspondiente a $\dot{\theta}_i = 1$ y $\dot{\theta}_j = 0 \; \forall \; j \neq i$.

Matriz Jacobiana expresada en el SR fijo

La Matriz Jacobiana en coordenadas del SR fijo $\{s\}$ ($J_s(\theta) \in \mathbb{R}^{6\times n}$) relaciona las derivadas temporales de las coordenadas de las articulaciones ($\dot{\theta} \in \mathbb{R}^n$) con los vectores *Giro* expresados en el mismo SR: $\boxed{\nu_s = J_s(\theta)\dot{\theta}}$.

La columna i de la Matriz Jacobiana es: $J_{si}(\theta) = Ad_{T_{i-1}}(\mathcal{S}_i)$ ($\forall \; i = 2, \cdots, n$) donde $T_{i-1} = e^{[\mathcal{S}_1]\theta_1} \cdots e^{[\mathcal{S}_{i-1}]\theta_{i-1}}$. La primera columna de la Matriz Jacobiana $J_{s1} = \mathcal{S}_1$

$\mathcal{S}_i$ es el eje helicoidal que describe el eje de la articulación i con el robot en la posición cero.

La columna i de la Matriz Jacobiana $Ad_{T_{i-1}}(\mathcal{S}_i) = J_{si}(\theta)$ es el eje helicoidal que describe el eje de la articulación i después de un desplazamiento rígido correspondiente a la matriz de transformación $T_{i-1} \Rightarrow$ esto es equivalente a mover las primeras $i-1$ articulaciones desde su posición cero hasta la configuración dada por $\theta_1, \cdots, \theta_{i-1}$.

Los valores de $J_{si}(\theta)$ se obtienen de manera parecida a cómo calculábamos $\mathcal{S}_i \Rightarrow$ cada columna de la matriz Jacobiana es el vector helicoidal que describe el eje de la articulación i pero para valores de θ arbitrarios en lugar de para $\theta = 0$.

Protocolo para calcular la Matriz Jacobiana de un Robot

- Al igual que se hizo en cinemática directa, **obtén los ejes helicoidales en la posición cero del Robot**: calcula el eje de rotación ω_i, el vector de posición desde el sistema de referencia fijo hasta cada articulación q_{si}, el producto vectorial de estos 2 vectores con signo negativo ($v_i = -\omega_i \times q_{si}$) y concatena ω_i con el resultado de este producto vectorial: $\mathcal{S}_i = (\omega_i, v_i)$. En el caso de que se trate de una articulación prismática simplemente se coge el vector (0,0,0) como eje de rotación y se completa el eje helicoidal con un vector unitario en el sentido de la extensión: $\mathcal{S}_i = (0, 0, 0, v_i)$
- Transforma el eje helicoidal a su forma de matriz antisimétrica ($[\mathcal{S}_i]$) y calcula las matrices de transformación homogénea correspondientes a cada articulación a través de la matriz exponencial con las coordenadas θ_i elegidas: $e^{[\mathcal{S}_i]\theta_i}$
- A continuación calcula las matrices de transformación homogéneas T_i obtenidas como el producto de matrices exponenciales correspondientes a las articulaciones $(0, 1, \cdots, i)$: $T_i = e^{[\mathcal{S}_1]\theta_1} \cdots e^{[\mathcal{S}_i]\theta_i}$
- Utiliza las matrices adjuntas de T_i para transformar los ejes helicoidades a la conformación que corresponda. Estos ejes helicoidales transformados serán directamente las columnas de la matriz Jacobiana: $J_{si}(\theta) = Ad_{T_{i-1}}(\mathcal{S}_i)$ $(\forall\ i = 2, \cdots, n)$.

Fíjate que para transformar cada eje helicoidal utilizamos la matriz adjunta del producto de matrices exponenciales desde la primera articulación hasta la inmediatamente anterior por lo que la primera columna de la matriz jacobiana es directamente el eje helicoidal de la primera articulación en la posición cero del robot.

Cálculo de la Matriz Jacobiana de un Robot

Dado que la matriz Jacobiana depende de la configuración del robot, es conveniente tener una expresión genérica parametrizada en función de las coordenadas de las articulaciones. Por este motivo vamos a programar nuestro robot utilizando cálculo simbólico, de manera que si queremos la Jacobiana de una configuración dada basta con sustituir los valores de θ_i. Esto significa que necesitamos transformar nuestras funciones para calcular ejes helicoidales, matrices exponenciales, matrices adjuntas, etc en funciones parametrizadas con sympy.

```
import sympy as sp
import numpy as np

# calcula una matriz de transformación homogénea a partir de un eje helicoidal y un ángulo de rotación
def MatrixExp6sp(S,theta):
    w = sp.Matrix(S[0:3])
    v = sp.Matrix(S[3:6])
    omgmat = VecToso3sp(w)
    if (w.norm() < 10e-6):
        return sp.eye(3).row_join(v*theta).col_join(sp.transpose(sp.Matrix([0,0,0,1])))
    else:
        R = MatrixExp3sp(w,theta)
        G_theta=sp.eye(3) * theta + (1 - sp.cos(theta)) * omgmat
                + (theta - sp.sin(theta)) * (omgmat*omgmat)
        return R.row_join(G_theta*v).col_join(sp.transpose(sp.Matrix([0,0,0,1])))

# calcula una matriz de rotación a partir del eje y del ángulo de rotación
def MatrixExp3sp(w, theta):
    w = sp.Matrix(w).normalized()
    omgmat = VecToso3sp(w)
    return sp.eye(3) + (sp.sin(theta)*omgmat) + ((1-sp.cos(theta))*(omgmat*omgmat))
```

```
# transforma un vector de 3 componentes a matriz antisimétrica
def VecToso3sp(v):
    v = sp.Matrix(v)
    return sp.Matrix([[0, -v[2], v[1]], [v[2], 0, -v[0]],[-v[1], v[0], 0]])

# calcula la matriz adjunta de una matriz de transformación homogénea
def Adjunta(T):
    R, p = TransToRp(T)
    return R.row_join(sp.zeros(3)).col_join((VecToso3sp(p)*R).row_join(R))

# separa la matriz de rotación y el vector de traslación
# de una matriz de transformación homogénea
def TransToRp(T):
    return sp.Matrix(T[0:3,0:3]), sp.Matrix(T[0:3,3])

# Coordenadas de las articulaciones (suponiendo 6 dof)
t=sp.symbols('t0, t1, t2, t3, t4, t5')
```

Los ejes de rotación y vectores entre articulaciones constituyen la estructura del robot y deben definirse en una función específica para este fin. Es una buena práctica introducir esta información a través de un archivo

Ejemplo de archivo de estructura de un robot: robot.yaml

```
robot:
  name: simple_robot
  links:
    - id: 0
      length: 0.080
      type: revolute
      link_orientation: [0, 0, 1]
      joint_coords: [0, 0, 0.103]
      joint_axis: [0, 0, 1]
    - id: 1
      length: 0.210
      type: revolute
      link_orientation: [0, 0, 1]
      joint_coords: [0, 0, 0.080]
      joint_axis: [0, 1, 0]
    - id: 2
      length: 0.0415
      type: prismatic
      link_orientation: [0, 0, 1]
      joint_coords: [0, 0, 0.210]
      joint_axis: [1, 0, 0]
```

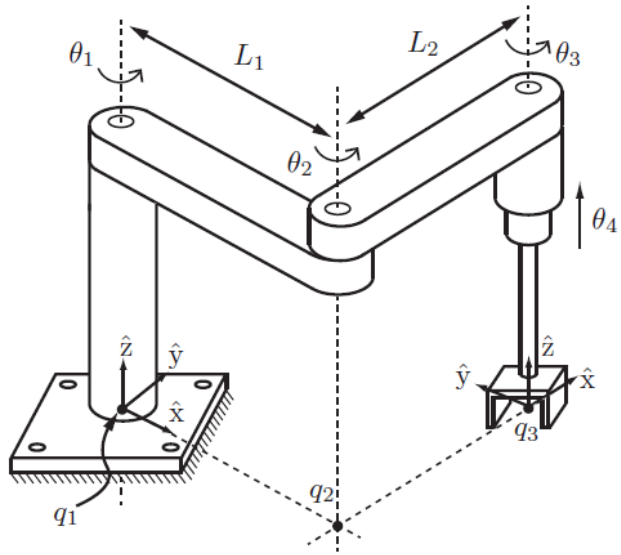
Para leer el archivo podemos usar el siguiente código:

```
import yaml

with open('robot.yaml', 'r') as file:
    robotdata = yaml.safe_load(file)

robotname = robotdata['robot']['name']
eslabones = robotdata['robot']['links']

for eslabon in eslabones:
    print("eslabón ID = ", eslabon['id'])
    print("tipo de eslabón = ", eslabon['type'])
    print("coordenadas articulación = ", eslabon['joint_coords'])
    print("Eje rotación (R) o expansión (P) = ", eslabon['axis'])
```

En general en robótica se utilizan archivos en formato URDF, que contienen mucha más información acerca del robot.

Ejemplo: Matriz Jacobiana para un robot 3RP

$$J_s(\theta) = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & L_1 s_1 & L_1 s_1 + L_2 s_{12} & 0 \\ 0 & -L_1 c_1 & -L_1 c_1 - L_2 c_{12} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



Matriz Jacobiana en el SR del Elemento Terminal

Relaciones entre las velocidades de las coordenadas de las articulaciones y el vector *Giro* expresado en coordenadas del SR localizado en el Elemento terminal ($\nu_b = T^{-1}\dot{T}$).

Matriz Jacobiana expresada en el SR fijo

La Matriz Jacobiana en coordenadas del SR del Elemento Terminal $\{b\}$ ($J_b(\theta) \in \mathbb{R}^{6\times n}$) relaciona las derivadas temporales de las coordenadas de las articulaciones ($\dot{\theta} \in \mathbb{R}^n$) con los vectores *Giro* en el mismo SR: $\boxed{\nu_b = J_b(\theta)\dot{\theta}}$.

La columna i de la Matriz Jacobiana es: $J_{bi}(\theta) = Ad_{e^{-[\mathcal{B}_n]\theta_n}\cdots e^{-[\mathcal{B}_{i+1}]\theta_{i+1}}}\mathcal{B}_i$ for $i = n-1, \cdots, 1$. La columna n de la Matriz Jacobiana $J_{bn} = \mathcal{B}_n$

Cada columna de la matriz Jacobiana $J_{bi}(\theta) = (\omega_{bi}(\theta), v_{bi}(\theta))$ es el vector helicoidal para el eje de la articulación i, expresado en las coordenadas del elemento terminal.

Las columnas de J_b se calculan de manera similar al proceso de derivar la cinemática directa en el producto de exponenciales $T = Me^{-[\mathcal{B}_n]\theta_n}\cdots e^{-[\mathcal{B}_{i+1}]\theta_{i+1}}$ pero ahora los ejes helicoidales se expresan para valores de θ arbitrarios en lugar de para $\theta = 0$.

Las matrices Jacobianas expresadas en los dos SR están relacionadas mediante las siguientes ecuaciones:

$$\boxed{J_b(\theta) = Ad_{T_{bs}}(J_s(\theta)) = [Ad_{T_{bs}}]J_s(\theta)} \quad \text{y} \quad \boxed{J_s(\theta) = Ad_{T_{sb}}(J_b(\theta)) = [Ad_{T_{sb}}]J_b(\theta)}$$

Estática

$$\text{Potencia aplicada a las articulaciones} = \text{Potencia para mover el Robot} + \\ + \text{Potencia disponible para el elemento terminal}$$

Si el Robot está en equilibrio estático (no se emplea potencia para moverlo) la potencia aplicada a las articulaciones coincide con la potencia en el elemento terminal:

$$\Rightarrow \tau^T \dot{\theta} = \mathcal{F}_b^T \nu_b \Rightarrow \tau = J_b^T(\theta)\mathcal{F}_b \quad (\text{utilizando la identidad: } \nu_b = J_b(\theta)\dot{\theta})$$

La misma ecuación se puede plantear en el SR fijo: $\tau = J_s^T(\theta)\mathcal{F}_s$ por lo que podemos prescindir del subíndice.

$\tau = J^T(\theta)\mathcal{F} \Rightarrow$ Si se aplica una *Llave* externa $-\mathcal{F}$ al elemento terminal cuando el Robot está en equlibrio con las coordenadas de las articulaciones en θ, esta ecuación calcula los torques τ que se necesitan para generar una llave de posición igual y de signo contrario $\mathcal{F}$, manteniendo el robot en equilibrio.

Análogamente, ¿ cuál es la *Llave* generada en el elemento terminal por un conjunto de torques dado? si J^T es una matriz 6×6 invertible $\Rightarrow \mathcal{F} = J^{-T}(\theta)\tau$. Si el número de articulaciones es distinto de 6 $\Rightarrow J^T$ no es invertible y la pregunta no está bien formulada

- Robots redundantes $\Rightarrow n > 6 \Rightarrow$ incluso para posiciones fijas del elemento terminal puede haber movimientos internos de las articulaciones y
- Robots con $n < 6 \Rightarrow$ para una posición fija del elemento terminal el Robot está inmobilizado y además no puede generar fuerzas en $6 - n$ direcciones. El robot puede resistir cq fuerza aplicada en esas direcciones sin moverse, sólo limitadas por su estructura.

Singularidades

Definición de Singularidad Cinemática

Es una configuración en la que el Robot no puede moverse de manera instantánea en una o más direcciones.
En una Singularidad la Matriz Jacobiana $J(\theta)$ tiene un rango inferior a 6.

$$\nu_b = \begin{bmatrix} J_{b1}(\theta) & \cdots & J_{bn-1}(\theta) & J_{bn}(\theta) \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \vdots \\ \dot{\theta}_{n-1} \\ \dot{\theta}_n \end{bmatrix} = J_{b1}(\theta)\dot{\theta}_1 + \cdots + J_{bn-1}(\theta)\dot{\theta}_{n-1} + J_{bn}(\theta)\dot{\theta}_n$$

Esto quiere decir que el elemento terminal puede alcanzar Giros que son combinaciones lineales de $J_{bi}(\theta)$. Para un robot de 6 o más articulaciones, el rango máximo de $J_b(\theta)$ es 6. Las configuraciones singulares son aquellas para las cuales el rango es inferior al máximo posible. En esa configuracion el elemento terminal no puede generar velocidades instantáneas en una o más dimensiones. Esta pérdida de mobilidad implica que el robot resiste Llaves de cualquier magnitud en las direcciones correspondientes a la pérdida de mobilidad.
El rango de la Matriz Jacobiana no depende del SR $\Rightarrow r(J_s(\theta)) = r(J_{s'}(\theta)) = r(J_b(\theta))$

Singularidades

Robot con 2 articulaciones de revolución colineales

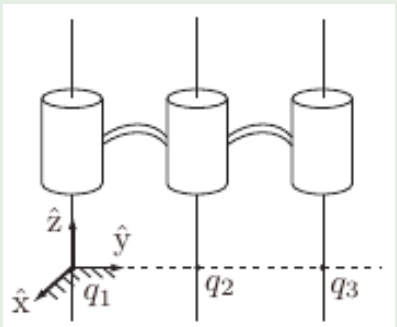
$$J_{s1}(\theta) = \begin{bmatrix} \omega_{s1} \\ -\omega_{s1} \times q_1 \end{bmatrix}; \; J_{s2}(\theta) = \begin{bmatrix} \omega_{s2} \\ -\omega_{s2} \times q_2 \end{bmatrix} \text{ donde } \begin{array}{c} \omega_{s1} = \pm\omega_{s2} \Rightarrow \\ \omega_{si} \times (q_1 - q_2) = 0 \end{array}$$

En este caso $J_{s1} = J_{s2}$ y los vectores $\{J_{s1}, J_{s2}, J_{s3}, J_{s4}, J_{s5}, J_{s6}\}$ no pueden ser L.I. con lo que el rango de la matriz Jacobiana es inferior a 6.

Robot con 3 articulaciones de revolución coplanares y paralelas no colineales

$$J_s(\theta) = \begin{bmatrix} \omega_{s1} & \omega_{s1} & \omega_{s1} & \cdots \\ 0 & -\omega_{s1} \times q_2 & -\omega_{s1} \times q_3 & \cdots \end{bmatrix}$$

Como q_2 y q_3 están en el mismo eje, las 3 columnas no pueden ser L.I.



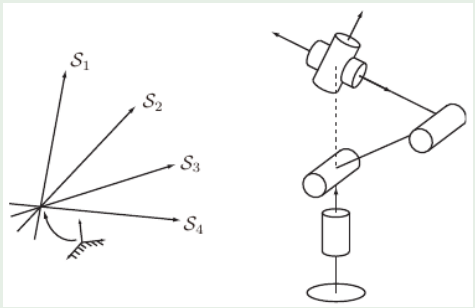
Singularidades

Robot con 4 articulaciones de revolución cuyos ejes se intersectan en un punto común

Escogiendo el SR fijo en el punto de unión de las 4 articulaciones:

$$J_s(\theta) = \begin{bmatrix} \omega_{s1} & \omega_{s2} & \omega_{s3} & \omega_{s3} & \cdots \\ 0 & 0 & 0 & 0 & \cdots \end{bmatrix}$$

Es evidente que las 4 columnas son L.D.



Robot con 4 articulaciones de revolución coplanares

$$\omega_{si} = \begin{bmatrix} \omega_{six} \\ \omega_{siy} \\ 0 \end{bmatrix} ; \; q_i = \begin{bmatrix} q_{ix} \\ q_{iy} \\ 0 \end{bmatrix} \Rightarrow v_{si} = -\omega_{si} \times q_i = \begin{bmatrix} 0 \\ 0 \\ \omega_{siy} q_{ix} - \omega_{six} q_{iy} \end{bmatrix} \Rightarrow$$

por lo que las 4 columnas correspondientes de la matriz Jacobiana no pueden ser L.I., ya que tienen sólo 3 componentes no nulas $\Rightarrow$ sólo pueden generar un espacio 3D en el que no puede haber 4 vectores L.I.:

$$J_{si}(\theta) = \begin{bmatrix} \omega_{six} \\ \omega_{siy} \\ 0 \\ 0 \\ 0 \\ \omega_{siy} q_{ix} - \omega_{six} q_{iy} \end{bmatrix}$$

Singularidades

Robot con 6 articulaciones de revolución que se intersectan en una línea común

Escogemos un SR fijo tal que la línea de intersección de los ejes de las 6 articulaciones es el eje z y seleccionamos la intersección entre esta línea común y el eje de la articulación i como punto de referencia para calcular la velocidad
$\Rightarrow q_i = (0, 0, q_{iz}) \Rightarrow v_{si} = -\omega_{si} \times q_i = (\omega_{siy} q_{iz}, -\omega_{six} q_{iz}, 0) \quad \forall\ i = 1, \cdots, 6.$
Las 6 componentes de la Matriz Jacobiana serán:

$$\begin{bmatrix} \omega_{s1x} & \omega_{s2x} & \omega_{s3x} & \omega_{s4x} & \omega_{s5x} & \omega_{s6x} \\ \omega_{s1y} & \omega_{s2y} & \omega_{s3y} & \omega_{s4y} & \omega_{s5y} & \omega_{s6y} \\ \omega_{s1z} & \omega_{s2z} & \omega_{s3z} & \omega_{s4z} & \omega_{s5z} & \omega_{s6z} \\ \omega_{s1y} q_{1z} & \omega_{s2y} q_{2z} & \omega_{s3y} q_{3z} & \omega_{s4y} q_{4z} & \omega_{s5y} q_{5z} & \omega_{s6y} q_{6z} \\ -\omega_{s1x} q_{1z} & -\omega_{s2x} q_{2z} & -\omega_{s3x} q_{3z} & -\omega_{s4x} q_{4z} & -\omega_{s5x} q_{5z} & -\omega_{s6x} q_{6z} \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

que es claramente singular por tener una fila nula.

Manipulabilidad

En una singularidad cinemática el elemento terminal no puede trasladarse o rotar en una o más direcciones. Podemos determinar cuándo nos acercamos a una singularidad o qué tan cerca estamos de ellas a partir del elipsoide de manipulabilidad.

Para un Robot genérico de lazo abierto y un espacio de tareas con coordenadas $q \in \mathbb{R}^m$ donde $m \leq n$, el elipsoide de manipulabilidad representa las velocidades del elemento terminal para velocidades de articulaciones dadas por $\dot{\theta}$ donde $||\dot{\theta}|| = 1 \Rightarrow$ una esfera unitaria en el espacio de las velocidades de las articulaciones $\Rightarrow \dot{\theta}^T \dot{\theta} = 1$.

Si la matriz Jacobiana se puede invertir
$\Rightarrow \dot{\theta}^T \dot{\theta} = (J^{-1}\dot{q})^T (J^{-1}\dot{q}) = \dot{q}^T J^{-T} J^{-1} \dot{q} = \dot{q}^T (JJ^T)^{-1} \dot{q} = \dot{q}^T A^{-1} \dot{q} = 1$

Si J es de rango completo $\Rightarrow r(J) = m$, la matriz $A = JJ^T \in \mathbb{R}^{m \times m}$ es cuadrada, simétrica y definida positiva, al igual que A^{-1}.

Para toda matriz simétrica y definida positiva $A^{-1} \in \mathbb{R}^{m \times m}$ el conjunto de vectores $\dot{q}$ tal que $\dot{q}^T A^{-1} \dot{q} = 1$ define un elipsoide en el espacio de de m dimensiones correspondiente cuyos semiejes van en la dirección de los autovectores v_i y tienen las dimensiones de las raíces cuadradas de los autovalores correspondientes $\sqrt{\lambda_i}$.

El volumen del elipsoide es proporcional al producto de los autovalores, que coincide con el determinante de la matriz $A \Rightarrow V \propto \sqrt{\lambda_1 \lambda_2 \cdots \lambda_m} = \sqrt{det(A)} = \sqrt{det(JJ^T)}$

Manipulabilidad

La matriz Jacobiana puede expresarse como: $J(\theta) = \begin{bmatrix} J_\omega(\theta) \\ J_v(\theta) \end{bmatrix}$

$J_\omega(\theta)$ es la matriz formada por las 3 primeras filas de $J(\theta)$ y $J_v(\theta)$ por las otras 3.

$\Rightarrow$ podemos tener 2 elipsoides de manipulabilidad, uno para las velocidades angulares y otra para las velocidades lineales, cada uno de ellos en 3D.

Estos elipsoides tienen los semiejes alineados con los autovectores de $(J_\omega J_\omega^T)$ o de $(J_v J_v^T)$, y sus dimensiones vienen dadas por las raíces cuadradas de los autovalores correspondientes.

A partir del elipsoide de manipulabilidad, podemos asignar una cantidad escalar para determinar la facilidad con la que se mueve el elemento terminal a una configuración dada:

$$\mu_1 = \sqrt{\frac{\lambda_{max}(A)}{\lambda_{min}(A)}} \geq 1 \qquad \mu_2 = \frac{\lambda_{max}(A)}{\lambda_{min}(A)} \geq 1 \qquad \mu_3 = \sqrt{\lambda_1 \lambda_2 \cdots} = \sqrt{det(A)}$$

- μ_1 es una medida de la isotropía del elipsoide, es deseable que sea esférico ($\Rightarrow \mu_1 = 1$). Al aproximarnos a una singularidad $\Rightarrow \mu_1 \to \infty$.
- $\mu_2 = \mu_1^2$ y se suele llamar Número de Condición de la matriz A.
- μ_3 es proporcional al volumen del elipsoide. Se busca el mayor valor posible.

Manipulabilidad

De manera análoga al elipsoide de manipulabilidad se puede dibujar un elipsoide de fuerza para representar la fuerza en el elemento terminal cuando los torques en las articulaciones τ se distribuyen de manera que $||\tau|| = 1$.

Partiendo de la expresión $\tau = J^T(\theta)\mathcal{F}$ llegamos a un resultado análogo al anteior pero para un elipsoide de fuerza: $\boxed{1 = f^T JJ^T f = f^T B^{-1} f}$, donde $B = (JJ^T)^{-1} = A^{-1}$.

Como $B = A^{-1}$, los ejes principales del elipsoide de fuerza están alineados con los del elipsoide de manipulabilidad. El elipsoide de fuerza se puede obtener a partir del elipsoide de manipulabilidad multiplicando su eje principal por un factor $1/\lambda_i$.

El producto de los volúmenes de los elipsoides de fuerza y manipulabilidad es una constante independiente de las coordenadas de las articulaciones. Una configuración que incremente $\mu_3(A)$ reducirá $\mu_3(B)$ y viceversa.