Funciones RVCTOOLS MATLAB

Link

Función para generar una articulación. Modo de uso:

```
Links1 = Link('d', d, 'a', a, 'alpha', alpha, 'modified', 'm',
m, 'r', cg,'B', B,'G',1, 'qlim', linklim);
```

La función nos permite cargar los parámetros de vinculo DH de la articulación. Por la nomenclatura que se utilizo durante el curso, es importante agregar “modified” a la función, ya que esto indica que los parámetros cargados corresponden a nomenclatura DH modificado. Otro parámetro que puede especificarse es el límite de los actuadores “qlim”, el cual no está definido para articulaciones prismaticas, pero si para articulaciones giratorias (0 – 360º). Utilizar el help link para revisar todos los parámetros que pueden cargarse.

Transl

Función que genera la matriz de transformación homogénea correspondiente a una traslación tridimensional. Útil para generar la matriz T de la herramienta entre otras cosas. Ejemplo de uso:

```
Tool = transl([x2, y2, z2])
```

Seriallink

Función que genera un manipulador, uniendo los links sucesivos de un vector, y colocando luego del último la herramienta. Modo de uso:

```
Robot = SerialLink( [links{:}], 'tool', Tool);
```

Sobre el robot que hemos creado, podremos aplicar varias funciones del toolbox, estas funciones pueden encontrarse en help seriallink, y se aplican conjugando la variable que contiene a nuestro robot (en este ejemplo “Robot”) con el nombre de la función a aplicar. Por ejemplo Robot.plot.

Seriallink.Fkine

Función que genera la transformada de base a herramienta, para un vector “q” (posiciones de actuadores) dado. Ejemplo de uso:

```
t1=Robot.fkine([-pi/2 pi/2])
```

Ctraj

Función que genera una sucesión de n transformadas homogenes, conectando de manera lineal una matriz Ti y una matriz Tf. Ejemplo de uso:

```
traylin=ctraj(t1,t2,nPuntos)
```

Seriallink.Ikine

Función que genera el vector de posiciones de actuadores para una dada matriz T de la herramienta. Es importante agregar en esta función el vector "q" estimado, y la máscara de joints en casos donde existan menos de 6 grados de libertad. Ejemplo de uso en un robot con dos grados de libertad:

```
q1 = Robot.ikine(t1,[-pi/2 pi/2],[1 1 0 0 0 0])
```

Seriallink.Perturb

Función que genera un nuevo manipulador robótico, basado en un manipulador de base, con sus parámetros dinámicos (masas y momentos de inercia) modificados aleatoriamente en un rango definido (m x (p-1) < m* < m x (p+1)). Modo de uso:

```
Robotmodif = Robot.perturb(p)
```

Ejercitación recomendada

1. Generar una matriz de posiciones de actuadores que deba seguir el manipulador, para lograr que el efector final recorra una trayectoria dada. Para esto, genere una posición inicial y final para el actuador utilizado la función seriallink.fkine, luego, genere la trayectoria entre esas posiciones con la función ctraj, y por último, aplique la función seriallink.ikine sobre la trayectoria generada.
2. Revise las funciones presentadas en este resumen en el "help" de MATLAB. Con el diagrama de control que propone el libro de Craig en mente, busque funciones relacionadas a "Seriallink" que puedan ser de utilidad para definir parámetros del modelo de control.
3. Dado que las entradas al sistema de control de mi actuador son las posiciones cartesianas, las velocidades y las aceleraciones en los tres ejes de mi efector final, genere las matrices de X, Xd, Xdd sucesivos necesarios para seguir la trayectoria del punto 1. Para esto, se recomienda aislar las posiciones de las matrices de transformación homogénea generadas con ctraj, y luego, aplicar la función "diff" sobre esta matriz para calcular Xd, y sobre Xd para calcular Xdd.