**FUNCIONALIDADES DEL INTERFAZ**

Como primer paso, debes crear un paquete llamado *iiwa\_surgery* en el que debes crear una clase (no un nodo) que cumpla los siguientes requisitos:

Variables miembro de la clase

* 1. Variable miembro de la clase en la que se configure si se trabaja en modo simulación (Gazebo) o con el robot real.

Nota: los topic en simulación y en real no son exactamente los mismo. Con esta variable se deberían utilizar unos u otros dependiendo de su configuración (te indicaremos cuales son).

* 1. Datos de la herramienta:
     1. Variables miembro (estructura) que indique la longitud y orientación de la herramienta respecto del EF.

Con esta información se calcula el Tool center point (TCP), que es la posición y orientación de la punta de la herramienta respecto del EF.

* 1. Punto de fulcro:
     1. Variable miembro de la posición del punto de fulcro. Se da como un parámetro decimal entre 0 y 1 que dice en que punto de la longitud de la herramienta se encuentra el punto de fulcro.
  2. Configuración del robot:
     1. Variable miembro con el valor de la IP del robot
     2. Si hiciera falta algún dato más asociado al robot habría que incluirlo como otra variable miembro. (Realmente no sabemos si hace falta algún dato más de configuración del robot).

Funciones de la clase

1. Movimiento libre articular: control del movimiento del robot sin restricción de punto de fulcro. Como parámetro de entrada se recibirían la configuración articular deseada. Sería mapear una de las funciones que ya tenga implementado el iiwa\_stack.
2. Movimiento libre cartesiano: control del movimiento del robot sin restricción del punto de fulcro. Como parámetro de entrada se recibiría la posición del TCP (x,y,z,rx,ry,rz). Sería mapear una de las funciones que ya tenga implementado el iiwa\_stack.
3. Movimiento cartesiano alrededor del punto de fulcro: movimiento de la punta de la herramienta teniendo en cuenta las restricciones impuestas por el punto de fulcro (TFG Pablo). Como parámetro de entrada se recibiría la posición del TCP (x,y,z,rx,ry,rz).

El paso siguiente sería montar un nodo que implemente la clase que se ha creado previamente y que cumpla los siguientes requisitos:

Suscriptores

1. iiwa\_surgery/command/joints
2. iiwa\_surgery/command/pose

Publicadores

1. iiwa\_surgery/output/joints
2. iiwa\_surgery/output/ef\_pose
3. iiwa\_surgery/output/tcp\_pose

Configuración (yaml)

Todas las variables miembro de la clase inicial deben estar mapeadas como un parámetro del fichero .yaml. Además debe aparecer otro parámetro que indique si se está trabajando en modo libre o modo pivotaje alrededor del punto de fulcro.

Configuración (launch)

En este fichero se configura cómo se lanza el nodo y también se puede configurar el mapeo de los topic dependiendo de si se trabaja en simulación o con el robot real.

**DOCUMENTACIÓN**

La implementación de la clase y del nodo tienen que estar documentados de manera que se pueda crear un doxygen de los mismos.

NOTA: nombre de variables, funciones, etc. en inglés

**¿QUÉ FALTARÍA?**

Para concluir el TFM habría que añadir a lo anterior:

* Probarlo en el robot real.
* Añadir al Gazebo una herramienta quirúrgica. Hay que buscar el modelo de la herramienta (en GitHub), incluirlo en Gazebo y adaptarlo a la orientación requerida.
* Memoria del trabajo. La memoria de los TFM tiene que estar más elaborada que la de los TFGs por lo que habrá que incluir un capítulo de Introducción, Estado del Arte con lo que tendrás que buscar bibliografía con la que poder redactar esta parte.

La fecha límite para solicitar la defensa en septiembre es el día 7, teniendo en cuenta que en el mes de agosto la escuela cierra y nosotras no vamos a estar disponibles, todo tendría que estar hecho y revisado para el 28 de julio. En el estado en el que está el trabajo a día de hoy, creemos que habría que presentarlo en diciembre.