

Interfaz multimodal para KUKA LBR iiwa 7 R800

Generado por Doxygen 1.9.8



|   |          |
|---|----------|
| <b>1 Índice de clases</b>   | <b>1</b> |
| 1.1 Lista de clases . . . . .   | 1        |
| <b>2 Índice de archivos</b>   | <b>3</b> |
| 2.1 Lista de archivos . . . . .   | 3        |
| <b>3 Documentación de clases</b>  | <b>5</b> |
| 3.1 Referencia de la clase <code>iiwa_surgery</code> . . . . .            | 5        |
| 3.1.1 Descripción detallada . . . . .                                     | 6        |
| 3.1.2 Documentación de constructores y destructores . . . . .             | 6        |
| 3.1.2.1 <code>__init__()</code> . . . . .                                 | 6        |
| 3.1.3 Documentación de funciones miembro . . . . .                        | 7        |
| 3.1.3.1 <code>move_cartesian()</code> . . . . .                           | 7        |
| 3.1.3.2 <code>move_cartesian_fulcrum()</code> . . . . .                   | 7        |
| 3.1.3.3 <code>move_joint()</code> . . . . .                               | 7        |
| 3.1.3.4 <code>set_fulcrum_fi()</code> . . . . .                           | 8        |
| 3.1.3.5 <code>set_robot_ip()</code> . . . . .                             | 8        |
| 3.1.3.6 <code>set_tool_data()</code> . . . . .                            | 8        |
| 3.1.3.7 <code>set_work_mode()</code> . . . . .                            | 8        |
| 3.1.4 Documentación de datos miembro . . . . .                            | 9        |
| 3.1.4.1 <code>cartesian_pub</code> . . . . .                              | 9        |
| 3.1.4.2 <code>fulcrum_fi</code> . . . . .                                 | 9        |
| 3.1.4.3 <code>joint_pub</code> . . . . .                                  | 9        |
| 3.1.4.4 <code>Pf</code> . . . . .   | 9        |
| 3.1.4.5 <code>Ptcp</code> . . . . .                                       | 9        |
| 3.1.4.6 <code>robot_ip</code> . . . . .                                   | 9        |
| 3.1.4.7 <code>simulation_mode</code> . . . . .                            | 10       |
| 3.1.4.8 <code>tool_length</code> . . . . .                                | 10       |
| 3.1.4.9 <code>tool_orientation</code> . . . . .                           | 10       |
| 3.1.4.10 <code>work_mode</code> . . . . .                                 | 10       |
| 3.2 Referencia de la clase <code>iiwa_surgery_node_class</code> . . . . . | 10       |
| 3.2.1 Descripción detallada . . . . .                                     | 11       |
| 3.2.2 Documentación de constructores y destructores . . . . .             | 11       |
| 3.2.2.1 <code>__init__()</code> . . . . .                                 | 11       |
| 3.2.3 Documentación de funciones miembro . . . . .                        | 12       |
| 3.2.3.1 <code>cartesian_pose_callback()</code> . . . . .                  | 12       |
| 3.2.3.2 <code>joint_command_callback()</code> . . . . .                   | 12       |
| 3.2.3.3 <code>joint_position_callback()</code> . . . . .                  | 12       |
| 3.2.3.4 <code>joint_state_callback()</code> . . . . .                     | 12       |
| 3.2.3.5 <code>pose_command_callback()</code> . . . . .                    | 14       |
| 3.2.3.6 <code>run()</code> . . . . .                                      | 14       |
| 3.2.4 Documentación de datos miembro . . . . .                            | 14       |
| 3.2.4.1 <code>cartesian_pose_sub</code> . . . . .                         | 14       |

---

|  |           |
|--|-----------|
| 3.2.4.2 ef_pose_pub . . . . .  | 14        |
| 3.2.4.3 first_position . . . . .   | 14        |
| 3.2.4.4 fulcrum_fi . . . . .   | 15        |
| 3.2.4.5 iiwa . . . . .   | 15        |
| 3.2.4.6 joint_position_sub . . . . .   | 15        |
| 3.2.4.7 joint_pub . . . . .  | 15        |
| 3.2.4.8 joint_state_sub . . . . .  | 15        |
| 3.2.4.9 joint_sub . . . . .  | 15        |
| 3.2.4.10 pose_sub . . . . .  | 15        |
| 3.2.4.11 robot_ip . . . . .  | 16        |
| 3.2.4.12 simulation_mode . . . . .   | 16        |
| 3.2.4.13 tcp_pose_pub . . . . .  | 16        |
| 3.2.4.14 tool_length . . . . .   | 16        |
| 3.2.4.15 tool_orientation . . . . .  | 16        |
| 3.2.4.16 work_mode . . . . .   | 16        |
| <b>4 Documentación de archivos</b>   | <b>17</b> |
| 4.1 Referencia del archivo TFM_JavierLara/Archivos comentados para Doxygen/iiwa_control_class.py . . | 17        |
| 4.2 Referencia del archivo TFM_JavierLara/Archivos comentados para Doxygen/iiwa_control_node.py . .  | 17        |
| <b>Índice alfabético</b>   | <b>19</b> |

# Capítulo 1

## Índice de clases

### 1.1. Lista de clases

Lista de clases, estructuras, uniones e interfaces con breves descripciones:

|   |  |    |
|---|--|----|
| <a href="#">iiwa_surgery</a>            | Clase para controlar un robot KUKA LBR iiwa con fines quirúrgicos en un entorno de ROS . .   | 5  |
| <a href="#">iiwa_surgery_node_class</a> | Clase que define el nodo ROS para el control del robot KUKA LBR iiwa con fines quirúrgicos . | 10 |



## Capítulo 2

# Índice de archivos

### 2.1. Lista de archivos

Lista de todos los archivos con breves descripciones:

|  |    |
|--|----|
| TFM_JavierLara/Archivos comentados para Doxygen/ <a href="#">iiwa_control_class.py</a> . . . . . | 17 |
| TFM_JavierLara/Archivos comentados para Doxygen/ <a href="#">iiwa_control_node.py</a> . . . . .  | 17 |





## Capítulo 3

# Documentación de clases

### 3.1. Referencia de la clase `iiwa_surgery`

Clase para controlar un robot KUKA LBR iiwa con fines quirúrgicos en un entorno de ROS.

#### Métodos protegidos

- `__init__` (self, `simulation_mode`=True)  
*Inicializa una instancia de la clase `iiwa_surgery`.*
- `move_cartesian` (self, pose)  
*Mueve el robot en el espacio cartesiano.*
- `move_cartesian_fulcrum` (self, pose, `increment_vector`, j)  
*Mueve el robot en el espacio cartesiano alrededor de un punto de fulcro.*
- `move_joint` (self, `joint_config`)  
*Mueve el robot en el espacio de articulación.*
- `set_fulcrum_fi` (self, `fulcrum_fi`)  
*Configura el punto de fulcro.*
- `set_robot_ip` (self, `robot_ip`)  
*Configura la dirección IP del robot.*
- `set_tool_data` (self, `tool_length`, `tool_orientation`)  
*Configura los datos de la herramienta.*
- `set_work_mode` (self, `work_mode`)  
*Configura el modo de trabajo del robot.*

#### Atributos protegidos

- `cartesian_pub`  
*Publicador ROS utilizado para enviar mensajes de posición cartesiana al robot.*
- `fulcrum_fi`  
*Valor que indica en que punto de la longitud de la herramienta se encuentra el punto de fulcro.*
- `joint_pub`  
*Publicador ROS utilizado para enviar mensajes de posición articular al robot.*
- `Pf`  
*Vector que almacena la posición del punto de fulcro en el espacio cartesiano.*

- **Ptcp**  
*Vector que almacena la posición del TCP (Tool Center Point) en el espacio cartesiano.*
- **robot\_ip**  
*Cadena que representa la dirección IP del robot.*
- **simulation\_mode**  
*Booleano que indica si el sistema está en modo de simulación.*
- **tool\_length**  
*Valor que representa la longitud de la herramienta.*
- **tool\_orientation**  
*Lista que almacena la orientación de la herramienta en radianes con el en formato [roll, pitch, yaw].*
- **work\_mode**  
*Cadena que indica el modo de trabajo del robot ('free' o 'pivot').*

### 3.1.1. Descripción detallada

Clase para controlar un robot KUKA LBR iiwa con fines quirúrgicos en un entorno de ROS.

Esta clase permite configurar el robot, establecer datos de la herramienta y realizar movimientos articulares y movimientos en el espacio cartesiano.

Configuración del robot:

- Esta clase se utiliza para controlar el robot KUKA LBR iiwa en un entorno de simulación con Gazebo o en un robot real.
- Es posible configurar la dirección IP del robot utilizando el método `set_robot_ip`.
- El modo de trabajo del robot se puede configurar como 'free' o 'pivot' utilizando el método `set_work_mode`.

Configuración de la herramienta:

- Utilizar el método `set_tool_data` para definir la longitud de la herramienta y su orientación. La herramienta a utilizar siempre se encuentra alineada con el eje Z del efector final (EF).

Movimiento articular preciso:

- El método `move_joint` te permite mover el robot en el espacio articular proporcionando una configuración de las articulaciones del robot.

Movimiento libre en el espacio cartesiano:

- El método `move_cartesian` permite mover el robot en el espacio cartesiano proporcionando una posición y orientación deseada del TCP.

Movimiento en el espacio cartesiano alrededor de un punto de fulcro:

- El método `move_cartesian_fulcrum` permite mover el robot alrededor de un punto de fulcro. Proporcionar una posición y orientación deseada del TCP, un vector de incrementos y un indicador de posición.

### 3.1.2. Documentación de constructores y destructores

#### 3.1.2.1. `__init__()`

```
__init__ (
    self,
    simulation_mode = True ) [protected]
```

Inicializa una instancia de la clase `iiwa_surgery`.

## Parámetros

|                        |   |
|------------------------|---|
| <i>simulation_mode</i> | (bool, opcional): Indica si se está ejecutando en modo simulación. El valor predeterminado es True. |
|------------------------|---|

**3.1.3. Documentación de funciones miembro****3.1.3.1. `move_cartesian()`**

```
move_cartesian (
    self,
    pose ) [protected]
```

Mueve el robot en el espacio cartesiano.

## Parámetros

|             |   |
|-------------|---|
| <i>pose</i> | ( <code>geometry_msgs.msg.PoseStamped</code> ): Posición y orientación deseada del Tool Center Point (TCP) en espacio cartesiano. |
|-------------|---|

**3.1.3.2. `move_cartesian_fulcrum()`**

```
move_cartesian_fulcrum (
    self,
    pose,
    increment_vector,
    j ) [protected]
```

Mueve el robot en el espacio cartesiano alrededor de un punto de fulcro.

## Parámetros

|                         |   |
|-------------------------|---|
| <i>pose</i>             | ( <code>geometry_msgs.msg.PoseStamped</code> ): Posición y orientación deseada del Tool Center Point (TCP) en espacio cartesiano. |
| <i>increment_vector</i> | (list): Vector de incrementos [Ph1, Ph2, Ph3] para mover la herramienta.  |
| <i>j</i>                | (int): Indicador de posición, 0 para la posición inicial, 1 para las posiciones sucesivas.  |

**3.1.3.3. `move_joint()`**

```
move_joint (
    self,
    joint_config ) [protected]
```

Mueve el robot en el espacio de articulación.

## Parámetros

|                     |  |
|---------------------|--|
| <i>joint_config</i> | ( <code>iiwa_msgs.msg.JointPosition</code> ): Configuración de las articulaciones del robot. |
|---------------------|--|

**3.1.3.4. set\_fulcrum\_fi()**

```
set_fulcrum_fi (
    self,
    fulcrum_fi ) [protected]
```

Configura el punto de fulcro.

**Parámetros**

|                   |  |
|-------------------|--|
| <i>fulcrum_fi</i> | (float): Posición del punto de fulcro, un valor entre 0 y 1. |
|-------------------|--|

**3.1.3.5. set\_robot\_ip()**

```
set_robot_ip (
    self,
    robot_ip ) [protected]
```

Configura la dirección IP del robot.

**Parámetros**

|                 |                                |
|-----------------|--------------------------------|
| <i>robot_ip</i> | (str): Dirección IP del robot. |
|-----------------|--------------------------------|

**3.1.3.6. set\_tool\_data()**

```
set_tool_data (
    self,
    tool_length,
    tool_orientation ) [protected]
```

Configura los datos de la herramienta.

**Parámetros**

|                         |  |
|-------------------------|--|
| <i>tool_length</i>      | (float): Longitud de la herramienta.                                 |
| <i>tool_orientation</i> | (list): Orientación de la herramienta en formato [roll, pitch, yaw]. |

**3.1.3.7. set\_work\_mode()**

```
set_work_mode (
    self,
    work_mode ) [protected]
```

Configura el modo de trabajo del robot.

## Parámetros

|                        |  |
|------------------------|--|
| <code>work_mode</code> | (str): Modo de trabajo del robot ('free' o 'pivot'). |
|------------------------|--|

### 3.1.4. Documentación de datos miembro

#### 3.1.4.1. cartesian\_pub

```
cartesian_pub [protected]
```

Publicador ROS utilizado para enviar mensajes de posición cartesiana al robot.

#### 3.1.4.2. fulcrum\_fi

```
fulcrum_fi [protected]
```

Valor que indica en que punto de la longitud de la herramienta se encuentra el punto de fulcro.

Debe estar entre 0 y 1, siendo 0 el EF y el 1 el TCP.

#### 3.1.4.3. joint\_pub

```
joint_pub [protected]
```

Publicador ROS utilizado para enviar mensajes de posición articular al robot.

#### 3.1.4.4. Pf

```
Pf [protected]
```

Vector que almacena la posición del punto de fulcro en el espacio cartesiano.

#### 3.1.4.5. Ptcp

```
Ptcp [protected]
```

Vector que almacena la posición del TCP (Tool Center Point) en el espacio cartesiano.

#### 3.1.4.6. robot\_ip

```
robot_ip [protected]
```

Cadena que representa la dirección IP del robot.

Ej: "192.228.17.57"

### 3.1.4.7. `simulation_mode`

`simulation_mode` [protected]

Booleano que indica si el sistema está en modo de simulación.

### 3.1.4.8. `tool_length`

`tool_length` [protected]

Valor que representa la longitud de la herramienta.

### 3.1.4.9. `tool_orientation`

`tool_orientation` [protected]

Lista que almacena la orientación de la herramienta en radianes con el en formato [roll, pitch, yaw].

### 3.1.4.10. `work_mode`

`work_mode` [protected]

Cadena que indica el modo de trabajo del robot ('free' o 'pivot').

La documentación de esta clase está generada del siguiente archivo:

- TFM\_JavierLara/Archivos comentados para Doxygen/[iiwa\\_control\\_class.py](#)

## 3.2. Referencia de la clase `iiwa_surgery_node_class`

Clase que define el nodo ROS para el control del robot KUKA LBR iiwa con fines quirúrgicos.

### Métodos protegidos

- [\\_\\_init\\_\\_](#) (self)  
*Constructor de la clase.*
- [cartesian\\_pose\\_callback](#) (self, msg)  
*Callback para recibir estados de posición cartesiana y posteriormente publicarlos en el topic `iiwa_surgery/output/ef←_pose` y tras transformación en `iiwa_surgery/output/tcp_pose`.*
- [joint\\_command\\_callback](#) (self, msg)  
*Callback para recibir comandos de posición articular y posteriormente enviarlos al método `move_joint` de la clase `iiwa_surgery`.*
- [joint\\_position\\_callback](#) (self, msg)  
*Callback para recibir estados de posición articular y posteriormente publicarlos en el topic `iiwa_surgery/output/joints`.*
- [joint\\_state\\_callback](#) (self, msg)  
*Callback para recibir estados de posición articular y posteriormente publicarlos en el topic `iiwa_surgery/output/joints`.*
- [pose\\_command\\_callback](#) (self, msg)  
*Callback para recibir comandos de posición cartesiana y posteriormente enviarlos a los métodos `move_cartesian` o `move_cartesian_fulcrum` de la clase `iiwa_surgery` dependiendo si el movimiento es libre o de pivoteo.*
- [run](#) (self)  
*Ejecuta el bucle principal mientras el nodo está en funcionamiento.*

**Atributos protegidos**

- `cartesian_pose_sub`  
*Suscriptor ROS utilizado para recibir estados de posición cartesiana.*
- `ef_pose_pub`  
*Publicador ROS utilizado para enviar estados de posición final del efector (EF) del robot.*
- `first_position`  
*Variable que se utiliza para almacenar la primera posición recibida en el callback `pose_command_callback`.*
- `fulcrum_fi`  
*Valor que indica en que punto de la longitud de la herramienta se encuentra el punto de fulcro.*
- `iiwa`  
*Es una instancia de la clase `iiwa_surgery` que se utiliza para controlar el robot `iiwa`.*
- `joint_position_sub`  
*Suscriptor ROS utilizado para recibir estados de posición articular.*
- `joint_pub`  
*Publicador ROS utilizado para enviar mensajes de posición articular.*
- `joint_state_sub`  
*Suscriptor ROS utilizado para recibir estados de posición articular.*
- `joint_sub`  
*Suscriptor ROS utilizado para recibir comandos de posición articular.*
- `pose_sub`  
*Suscriptor ROS utilizado para recibir comandos de posición cartesiana.*
- `robot_ip`  
*Almacena la dirección IP del robot `iiwa`.*
- `simulation_mode`  
*Indica si el robot está en modo de simulación o no, siendo `True` para simulación y `False` para el robot real.*
- `tcp_pose_pub`  
*Publicador ROS utilizado para enviar estados de posición del Tool Center Point (TCP).*
- `tool_length`  
*Almacena la longitud de la herramienta utilizada en el robot.*
- `tool_orientation`  
*Lista que almacena la orientación de la herramienta en radianes con el en formato `[roll, pitch, yaw]`.*
- `work_mode`  
*Indica el modo de trabajo predeterminado del robot ("`free`" para movimiento libre y "`pivot`" para movimiento alrededor de punto de fulcro).*

**3.2.1. Descripción detallada**

Clase que define el nodo ROS para el control del robot KUKA LBR `iiwa` con fines quirúrgicos.

**3.2.2. Documentación de constructores y destructores****3.2.2.1. `__init__()`**

```
__init__ (
    self ) [protected]
```

Constructor de la clase.

### 3.2.3. Documentación de funciones miembro

#### 3.2.3.1. cartesian\_pose\_callback()

```
cartesian_pose_callback (
    self,
    msg ) [protected]
```

Callback para recibir estados de posición cartesiana y posteriormente publicarlos en el topic `iiwa_↔surgery/output/ef_pose` y tras transformación en `iiwa_surgery/output/tcp_pose`.

##### Parámetros

|            |   |
|------------|---|
| <i>msg</i> | Mensaje de estado de posición cartesiana (PoseStamped). |
|------------|---|

#### 3.2.3.2. joint\_command\_callback()

```
joint_command_callback (
    self,
    msg ) [protected]
```

Callback para recibir comandos de posición articular y posteriormente enviarlos al método `move_joint` de la clase `iiwa_surgery`.

##### Parámetros

|            |  |
|------------|--|
| <i>msg</i> | Mensaje de posición articular (JointPosition). |
|------------|--|

#### 3.2.3.3. joint\_position\_callback()

```
joint_position_callback (
    self,
    msg ) [protected]
```

Callback para recibir estados de posición articular y posteriormente publicarlos en el topic `iiwa_↔surgery/output/joints`.

##### Parámetros

|            |  |
|------------|--|
| <i>msg</i> | Mensaje de estado de posición articular (JointPosition). |
|------------|--|

#### 3.2.3.4. joint\_state\_callback()

```
joint_state_callback (
    self,
    msg ) [protected]
```



Callback para recibir estados de posición articular y posteriormente publicarlos en el topic `iiwa_↔surgery/output/joints`.

**Parámetros**

|            |   |
|------------|---|
| <i>msg</i> | Mensaje de estado de posición articular (JointState). |
|------------|---|

**3.2.3.5. pose\_command\_callback()**

```
pose_command_callback (
    self,
    msg ) [protected]
```

Callback para recibir comandos de posición cartesiana y posteriormente enviarlos a los métodos `move_cartesian` o `move_cartesian_fulcrum` de la clase `iiwa_surgey` dependiendo si el movimiento es libre o de pivoteo.

**Parámetros**

|            |   |
|------------|---|
| <i>msg</i> | Mensaje de posición cartesiana (PoseStamped). |
|------------|---|

**3.2.3.6. run()**

```
run (
    self ) [protected]
```

Ejecuta el bucle principal mientras el nodo está en funcionamiento.

**3.2.4. Documentación de datos miembro****3.2.4.1. cartesian\_pose\_sub**

```
cartesian_pose_sub [protected]
```

Suscriptor ROS utilizado para recibir estados de posición cartesiana.

**3.2.4.2. ef\_pose\_pub**

```
ef_pose_pub [protected]
```

Publicador ROS utilizado para enviar estados de posición final del efector (EF) del robot.

**3.2.4.3. first\_position**

```
first_position [protected]
```

Variable que se utiliza para almacenar la primera posición recibida en el callback `pose_command_callback`.

**3.2.4.4. fulcrum\_fi**

```
fulcrum_fi [protected]
```

Valor que indica en que punto de la longitud de la herramienta se encuentra el punto de fulcro.

Debe estar entre 0 y 1, siendo 0 el EF y el 1 el TCP.

**3.2.4.5. iiwa**

```
iiwa [protected]
```

Es una instancia de la clase iiwa\_surgery que se utiliza para controlar el robot iiwa.

**3.2.4.6. joint\_position\_sub**

```
joint_position_sub [protected]
```

Suscriptor ROS utilizado para recibir estados de posición articular.

**3.2.4.7. joint\_pub**

```
joint_pub [protected]
```

Publicador ROS utilizado para enviar mensajes de posición articular.

**3.2.4.8. joint\_state\_sub**

```
joint_state_sub [protected]
```

Suscriptor ROS utilizado para recibir estados de posición articular.

**3.2.4.9. joint\_sub**

```
joint_sub [protected]
```

Suscriptor ROS utilizado para recibir comandos de posición articular.

**3.2.4.10. pose\_sub**

```
pose_sub [protected]
```

Suscriptor ROS utilizado para recibir comandos de posición cartesiana.

#### 3.2.4.11. robot\_ip

```
robot_ip [protected]
```

Almacena la dirección IP del robot iiwa.

#### 3.2.4.12. simulation\_mode

```
simulation_mode [protected]
```

Indica si el robot está en modo de simulación o no, siendo True para simulación y False para el robot real.

#### 3.2.4.13. tcp\_pose\_pub

```
tcp_pose_pub [protected]
```

Publicador ROS utilizado para enviar estados de posición del Tool Center Point (TCP).

#### 3.2.4.14. tool\_length

```
tool_length [protected]
```

Almacena la longitud de la herramienta utilizada en el robot.

#### 3.2.4.15. tool\_orientation

```
tool_orientation [protected]
```

Lista que almacena la orientación de la herramienta en radianes con el en formato [roll, pitch, yaw].

#### 3.2.4.16. work\_mode

```
work_mode [protected]
```

Indica el modo de trabajo predeterminado del robot ("free" para movimiento libre y "pivot" para movimiento alrededor de punto de fulcro).

La documentación de esta clase está generada del siguiente archivo:

- TFM\_JavierLara/Archivos comentados para Doxygen/[iiwa\\_control\\_node.py](#)

## Capítulo 4

# Documentación de archivos

### 4.1. Referencia del archivo TFM\_JavierLara/Archivos comentados para Doxygen/iiwa\_control\_class.py

#### Clases

- class [iiwa\\_surgery](#)

*Clase para controlar un robot KUKA LBR iiwa con fines quirúrgicos en un entorno de ROS.*

#### Espacios de nombres

- namespace [iiwa\\_control\\_class](#)

### 4.2. Referencia del archivo TFM\_JavierLara/Archivos comentados para Doxygen/iiwa\_control\_node.py

#### Clases

- class [iiwa\\_surgery\\_node\\_class](#)

*Clase que define el nodo ROS para el control del robot KUKA LBR iiwa con fines quirúrgicos.*

#### Espacios de nombres

- namespace [iiwa\\_control\\_node](#)

#### Variables

- [node](#) = [iiwa\\_surgery\\_node\\_class](#)()



# Índice alfabético

- `__init__`
  - `iiwa_surgery`, 6
  - `iiwa_surgery_node_class`, 11
- C:/Users/Javier/Desktop/TFM/Repositorio/Archivos comentados para Doxygen/iiwa\_control\_class.py, 17
- C:/Users/Javier/Desktop/TFM/Repositorio/Archivos comentados para Doxygen/iiwa\_control\_node.py, 17
- `cartesian_pose_callback`
  - `iiwa_surgery_node_class`, 12
- `cartesian_pose_sub`
  - `iiwa_surgery_node_class`, 14
- `cartesian_pub`
  - `iiwa_surgery`, 9
- `ef_pose_pub`
  - `iiwa_surgery_node_class`, 14
- `first_position`
  - `iiwa_surgery_node_class`, 14
- `fulcrum_fi`
  - `iiwa_surgery`, 9
  - `iiwa_surgery_node_class`, 14
- `iiwa`
  - `iiwa_surgery_node_class`, 15
- `iiwa_surgery`, 5
  - `__init__`, 6
  - `cartesian_pub`, 9
  - `fulcrum_fi`, 9
  - `joint_pub`, 9
  - `move_cartesian`, 7
  - `move_cartesian_fulcrum`, 7
  - `move_joint`, 7
  - `Pf`, 9
  - `Ptcp`, 9
  - `robot_ip`, 9
  - `set_fulcrum_fi`, 8
  - `set_robot_ip`, 8
  - `set_tool_data`, 8
  - `set_work_mode`, 8
  - `simulation_mode`, 9
  - `tool_length`, 10
  - `tool_orientation`, 10
  - `work_mode`, 10
- `iiwa_surgery_node_class`, 10
  - `__init__`, 11
  - `cartesian_pose_callback`, 12
  - `cartesian_pose_sub`, 14
  - `ef_pose_pub`, 14
  - `first_position`, 14
  - `fulcrum_fi`, 14
  - `iiwa`, 15
  - `joint_command_callback`, 12
  - `joint_position_callback`, 12
  - `joint_position_sub`, 15
  - `joint_pub`, 15
  - `joint_state_callback`, 12
  - `joint_state_sub`, 15
  - `joint_sub`, 15
  - `pose_command_callback`, 14
  - `pose_sub`, 15
  - `robot_ip`, 15
  - `run`, 14
  - `simulation_mode`, 16
  - `tcp_pose_pub`, 16
  - `tool_length`, 16
  - `tool_orientation`, 16
  - `work_mode`, 16
- `joint_command_callback`
  - `iiwa_surgery_node_class`, 12
- `joint_position_callback`
  - `iiwa_surgery_node_class`, 12
- `joint_position_sub`
  - `iiwa_surgery_node_class`, 15
- `joint_pub`
  - `iiwa_surgery`, 9
  - `iiwa_surgery_node_class`, 15
- `joint_state_callback`
  - `iiwa_surgery_node_class`, 12
- `joint_state_sub`
  - `iiwa_surgery_node_class`, 15
- `joint_sub`
  - `iiwa_surgery_node_class`, 15
- `move_cartesian`
  - `iiwa_surgery`, 7
- `move_cartesian_fulcrum`
  - `iiwa_surgery`, 7
- `move_joint`
  - `iiwa_surgery`, 7
- `Pf`
  - `iiwa_surgery`, 9
- `pose_command_callback`
  - `iiwa_surgery_node_class`, 14
- `pose_sub`

- iiwa\_surgery\_node\_class, [15](#)
- Ptcp
  - iiwa\_surgery, [9](#)
- robot\_ip
  - iiwa\_surgery, [9](#)
  - iiwa\_surgery\_node\_class, [15](#)
- run
  - iiwa\_surgery\_node\_class, [14](#)
- set\_fulcrum\_fi
  - iiwa\_surgery, [8](#)
- set\_robot\_ip
  - iiwa\_surgery, [8](#)
- set\_tool\_data
  - iiwa\_surgery, [8](#)
- set\_work\_mode
  - iiwa\_surgery, [8](#)
- simulation\_mode
  - iiwa\_surgery, [9](#)
  - iiwa\_surgery\_node\_class, [16](#)
- tcp\_pose\_pub
  - iiwa\_surgery\_node\_class, [16](#)
- tool\_length
  - iiwa\_surgery, [10](#)
  - iiwa\_surgery\_node\_class, [16](#)
- tool\_orientation
  - iiwa\_surgery, [10](#)
  - iiwa\_surgery\_node\_class, [16](#)
- work\_mode
  - iiwa\_surgery, [10](#)
  - iiwa\_surgery\_node\_class, [16](#)