

Curso: Robótica IELE-3338
Semestre: 2022-10
Profesor: Juan José García
Asistentes: Susana Marcela Chavez
Monitores: Vilma Tirado
Entrega: No entregable



Actividad en clase Cinemática directa e inversa



Figure 1: Manipulador Panda, by Emika Franka, más información en: <https://www.franka.de/>

Enunciado

1. **Forward and Inverse Kinematics:** Para ejecutar la cinemática inversa y directa del manipulador debe ejecutar lo siguiente:

```
roslaunch moveit_tutorials robot_model_and_robot_state_tutorial.launch
```

La salida al ejecutar esto debe ser:

```
ros.moveit_tutorials: Model frame: /panda_link0  
ros.moveit_tutorials: Joint panda_joint1: 0.000000  
ros.moveit_tutorials: Joint panda_joint2: 0.000000  
ros.moveit_tutorials: Joint panda_joint3: 0.000000  
ros.moveit_tutorials: Joint panda_joint4: 0.000000
```

```

ros.moveit_tutorials: Joint panda_joint5: 0.000000
ros.moveit_tutorials: Joint panda_joint6: 0.000000
ros.moveit_tutorials: Joint panda_joint7: 0.000000
ros.moveit_tutorials: Current state is not valid
ros.moveit_tutorials: Current state is valid
ros.moveit_tutorials: Translation:
-0.541498
-0.592805
0.400443

ros.moveit_tutorials: Rotation:
-0.395039 0.600666 -0.695086
0.299981 -0.630807 -0.715607
-0.868306 -0.491205 0.0690048

ros.moveit_tutorials: Joint panda_joint1: -2.407308
ros.moveit_tutorials: Joint panda_joint2: 1.555370
ros.moveit_tutorials: Joint panda_joint3: -2.102171
ros.moveit_tutorials: Joint panda_joint4: -0.011156
ros.moveit_tutorials: Joint panda_joint5: 1.100545
ros.moveit_tutorials: Joint panda_joint6: 3.230793
ros.moveit_tutorials: Joint panda_joint7: -2.651568
ros.moveit_tutorials: Jacobian:
0.592805 -0.0500638 -0.036041 0.366761 -0.0334361 0.128712 -4.33681e-18
-0.541498 -0.0451907 0.0417049 -0.231187 0.0403683 0.00288573 3.46945e-18
0 -0.799172 0.0772022 -0.247151 0.0818336 0.0511662 0
0 0.670056 -0.742222 0.349402 -0.748556 -0.344057 -0.695086
0 -0.74231 -0.669976 -0.367232 -0.662737 0.415389 -0.715607
1 4.89669e-12 0.0154256 0.862009 0.021077 0.842067 0.0690048

```

La primera parte muestra el estado de cada uno de los joints que componen el manipulador. Seguido de esto se muestra la matriz de translación y rotación que debe tener la posición del end-effector para ser evaluada en términos de la posición global del manipulador (Cinemática directa). Finalmente se muestra el estado de cada uno de los joints para llegar a una posición final del end-effector en cuestión.

- Calcule la posición en el marco de coordenadas global del actuador (end-effector) del manipulador a partir de las matrices de traslación y rotación calculadas por el nodo.
- Estudie y modifique el código del archivo *robot_model_and_robot_state_tutorial.cpp* de forma que a partir de ciertos goals personalizados puedan ejecutar la cinemática inversa y se pueda observar y analizar el estado de cada joint. El objetivo es que explique de forma detallada como funciona este nodo de ROS y lo ponga a prueba con posiciones finales deseadas (Mínimo dos goals de prueba). Siga el siguiente ejemplo

Ejemplo: En el archivo *robot_model_and_robot_state_tutorial.cpp* se encuentran programadas en C++ las funcionalidades del nodo previamente descritas. A partir de la línea 124 se comienza a calcular la cinemática inversa (Se cuenta los comentarios). Como se puede observar en las líneas siguientes a la función que calcula la cinemática inversa del manipulador (*kinematic_state->setFromIK(joint_model_group, end_effector_state, timeout);*) se le pasa como argumento una posición del end effector que ya se tiene calculada por la cinemática directa. Para llegar a cambiar esta posición se deben agregar las siguientes líneas de código de manera previa:

```

geometry_msgs::Pose desirePose;
desirePose.position.x = 0.5;
desirePose.position.y = 0.5;

```

```
desiredPose.position.z = 0.5;  
desiredPose.orientation.x = 0.0;  
desiredPose.orientation.y = 0.0;  
desiredPose.orientation.z = 0.0;  
desiredPose.orientation.w = 1.0;
```

Adicionalmente se debe cambiar el argumento de entrada de dicha función, tal que su resultado final sea el siguiente:

```
kinematic_state->setFromIK(joint_model_group, desiredPose, timeout);
```