Task priority assignment with collision avoidance

Stefano De Filippis & Marco Menchetti

Sapienza - University of Rome

Why priority?

- Decomposition of problems in many tasks.
- Most problems can't be solved by just one task.
- Error is kept on the tasks that can't be executed **EXACTLY**
- More natural and smoother behavior.

Collision avoidance. How?

1. Control points

We push constantly the control point away from the obstacle so as to be sure they will never get in contact!

How do we handle priority?

- When the control point is too near the priority lowers (i.e. becomes more important)
- As soon as the distance exit the *dangerous* region the priority rise again

TODO: Figure of the KUKA and its control points

Collision avoidance. How?

2. How do we push?

We change approach whether the control point is on the e-e or on the structure:

- For the end-effector we add a cartesian velocity pointing away from the obstacle
- For the structure we add a velocity on the rotation plane of the joint **TODO:**fix

TODO:figure of the repulsive velocity

Tasks

We know why to prioritize Tasks, but which are the ones we are going to use?

- A cartesian positioning task (i.e. we want our e-e to behave in a certain way)
  - **3 DOFs**
- An orientation task used to simulate any kind of auxiliary task
  - **2 DOFs**
- Two collision avoidance task, each one on 1 DOF
  - **2 DOFs**

In the end we saturated all the 7 DOFs of the manipulator.

Tasks: Cartesian positioning

Cartesian positioning means we want the end effector to execute a given trajectory {**TODO:**check timing law } in $\mathbb{R}^3$.
In our case we admitted 2 different kind of paths:

- A linear path
- A circular path

We could have also used different paths but for the purpose of this project they were enough, and its associated jacobian is the analytical jacobian of the direct kinematics.

Tasks: Link orientation

The orientation task tries to keep constant the orientation of the **TODO:**link axis and it can be defined as follows **TODO:**check:

$$\sum_{i=2}^{4} q_i = \frac{\pi}{2} \longrightarrow \sum_{i=2}^{4} q_i - \frac{\pi}{2} = 0$$

Hence its associated jacobian is a row made by all zeroes except for the components of q which appear in the task.

Tasks: Collision avoidance control points

Code

Results