HW2

December 2024

1 Joint-space Inverse dynamics(JSID)

For the joint-space control we had the following equation describing the motion of a robot

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) = u \tag{1}$$

By designing the control law as

$$u = M(q)a_q + C(q, \dot{q})\dot{q} + g(q) \tag{2}$$

we can achive the following

$$M(q)\ddot{q} = M(q)a_q \tag{3}$$

and since the $M(q)$ matrix is invertible we had

$$\ddot{q} = a_q$$

That means that we can change the system's behavior to a new input a_q as

$$a_q = -K_p q - K_d \dot{q} + r \tag{4}$$

where r is the control input based on the desired trajectory

$$r(t) = \ddot{q}_d(t) + K_p q_d(t) + K_d \dot{q}_d(t) \tag{5}$$

so the final control law $u(q, \dot{q}, t)$ that we used was
u = M(q)aq + C(q, q˙) ˙q + g(q)

$$u = M(q)(\ddot{q}_d(t) + K_p(q_d(t) - q) + K_d(\dot{q}_d(t) - \dot{q})) + C(q, \dot{q})\dot{q} + g(q) \tag{6}$$

2 Task Space Inverse Dynamics

Let's denote the position of the end-effector as p then we can describe its behavior using the jacobians and the joint space as

$$\dot{p} = J(q)\dot{q}\ddot{p} = J(q)\ddot{q} + \dot{J}(q)\dot{q} \tag{7}$$

from the second equation we can find the realtion of the task space to the joint space accelerations as

$$\ddot{q} = J^{-1}(q)\{\ddot{p} - \dot{J}(q)\dot{q}\} \tag{8}$$

this is actually our a_q from the JSID

Because the representation of the end effector conists of its position and oreintation we can split the p vector into 2 parts

$$p = \begin{bmatrix} p_x \\ p_\omega \end{bmatrix}$$

2.1 Position law

For the control law of $\ddot{p_x}$ everything is pretty straightforward - we simply use error dynamics as follows

$$a_x(t) = \ddot{x}_d(t) + K_p(x_d(t) - x) + K_d(\dot{x}_d(t) - \dot{x}) \tag{9}$$

But we have to adjust the jacobian to be alligned with axes where we want to specify the desired velocity. For my own case it's easier to define the trajectory in the world frame, thus I need to use the upper part of jacobians J and $\dot{J}$ in the local world aligned frame.

2.2 Orientation law

The orientation can be specified using a twist vector, however it's not that trivial to obtain the error. this problem can be resolve by using $SO(3)$ error which is defined as

$$\tilde{S} = log(R_d R^T) \tag{10}$$

Both matrices are orthonormal by definition, therefore in case of R_d being equal to R (the current state) we are getting an identity which after resolving the skew-symmetric matrix to a vector lead to a zero error vector which is what we need. There is a more complex explanation of this error within the group theory. However I did not study it and that's why for me it's enough of logical reasoning behind the formal for right now.

$$a_\omega(t) = \ddot{\omega}_d(t) + K_p\tilde{S}_\omega + K_d\tilde{S}_{\dot{\omega}} \tag{11}$$

3 Finalized solution

We define our desired behavior of the system in task-space as

$$\ddot{p} = \begin{bmatrix} a_x(t) \\ a_\omega(t) \end{bmatrix} \tag{12}$$

Which we substitute into the outer loop equation 8 and get the desired behaviour of the system in the joint space

$$\ddot{q} = J^{-1}(q)\{\begin{bmatrix} a_x(t) \\ a_\omega(t) \end{bmatrix} - \dot{J}(q)\dot{q}\} \tag{13}$$

and finally we substitute the outer loop into the inner loop resulting in the finalized control law as

$$u = M(q)J^{-1}(q)\{\begin{bmatrix} \ddot{x}_d(t) + K_p(x_d(t) - x) + K_d(\dot{x}_d(t) - \dot{x}) \\ \ddot{\omega}_d(t) + K_p\tilde{S}_\omega + K_d\tilde{S}_{\dot{\omega}} \end{bmatrix} - \dot{J}(q)\dot{q}\} + C(q, \dot{q})\dot{q} + g(q) \tag{14}$$