Visual Servoing Control

Robotics instructional lab #5

Spring 2022

In this lab, we will implement a simple controller for the robot gripper to track an object. A camera on the robot would track a marker on the object and maintain a desired relative pose.

1 Background

Let $\mathcal{O}_{cam}$ be a coordinate frame of a camera fixed on the gripper. Furthermore, a fiducial[1] marker as seen in Figure 1 is fixed on the object such that a calibrated camera can observe its position and orientation. Hence, the camera provides the position $\mathbf{p} \in \mathbb{R}^3$ and orientation $R \in SO(3)$ of an object represented with respect to $\mathcal{O}_{cam}$. The gripper is required to maintain a relative position $\mathbf{p}_{target}$ and orientation R_{target}.

Given the current pose, $\mathbf{p}_{cur}$ and R_{cur}, we compute a rigid-object Jacobian for correction the pose of the gripper. The vector of pose error is defined as

$$\mathbf{e} = (\mathbf{p}_{cur} - \mathbf{p}_{target}, \theta\mathbf{u})^T \tag{1}$$

where $\mathbf{u}$ and θ are the axis and angle of rotation, respectively, extracted from the rotation error matrix $R_{target}R_{cur}^T$.

Define a skew-symmetric representation of a vector $\mathbf{p}$ as

$$S(\mathbf{p}) = \begin{bmatrix} 0 & -p_3 & p_2 \\ p_3 & 0 & -p_1 \\ -p_2 & p_1 & 0 \end{bmatrix}. \tag{2}$$

The Jacobian J of the gripper is defined by

$$J = \begin{bmatrix} -I_{3\times3} & S(\mathbf{p}_{cur}) \\ \mathbf{0} & J_{\theta\mathbf{u}} \end{bmatrix}, \tag{3}$$

such that

$$J_{\theta\mathbf{u}} = I_{3\times3} - \frac{\theta}{2}S(\mathbf{u}) + \left(1 - \frac{\text{sinc}\theta}{\text{sinc}^2\frac{\theta}{2}}S^2(\mathbf{u})\right). \tag{4}$$

[1] https://docs.opencv.org/4.x/d5/dae/tutorial_aruco_detection.html

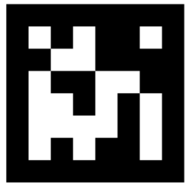

Figure 1: A fiducial marker of type ArUco.

Function sinc is the sinus cardinal implemented using `numpy.sinc()`. The pose change control for the gripper is then given by

$$\mathbf{t} = -\lambda J^{-1}\mathbf{e} \tag{5}$$

such that $\mathbf{t} = (\mathbf{v}, \mathbf{w})^T$ is the required gripper position $\mathbf{v}$ and orientation $\mathbf{w}$ change. Then, pose change $\mathbf{t}$ is applied to the gripper. Control (5) is repeatedly applied for some number of iterations while updating J and $\mathbf{e}$ until reaching $\|\mathbf{e}\| < \epsilon$ for some small value ϵ.

2 Prerequisites

In file **lab07_student.py**, fill:

- Function `calculate_error`($\mathbf{p}_{cur}, R_{cur}, \mathbf{p}_{target}, R_{target}$) to compute pose error in (1).
- Function `feature_jacobian`($\mathbf{p}_{cur}, R_{cur}, R_{target}$) to calculate the Jacobian matrix of the gripper in (3).
- Function `control`($\mathbf{J}_{gripper}, \mathbf{e}, \lambda$) to compute the required gripper change.

Prerequisites are a mandatory in order to carry out the lab .

3 Lab instructions

You are required to evaluate the accuracy of the control. Do the following:

1. Position the object at a random location such that the marker is observable.
2. Manually move the robot to a desired target pose and run file `Lab_07.py` to log-in the pose.
3. Repeat the following for 5 values of λ in the range $[0.1, 0.8]$:
   - Repeat 5 times:
     - Manually move the robot to an arbitrary pose where the marker is visible to the camera.

- Store the tracking file.

Test the ability of the control to track a moving object:

1. Position the object in front of the gripper at a desired target pose and run file `Lab_07.py` to log-in the pose and track the object.
2. Move the object for approximately 10 sec .
3. Repeat the test 5 times with different values of λ and observe performance.

4 Report requirements

The lab report should include the following:

1. Describe the process of the lab.
2. Plot the Root-Mean-Square-Error (RMSE) of with regards to the number of iterations. In the same plot, include the 5 curves for the various values of λ. Discuss your results and the effect of λ.
3. How does λ affect the ability of the controller to track a moving object?
4. Provide a summary for your results.