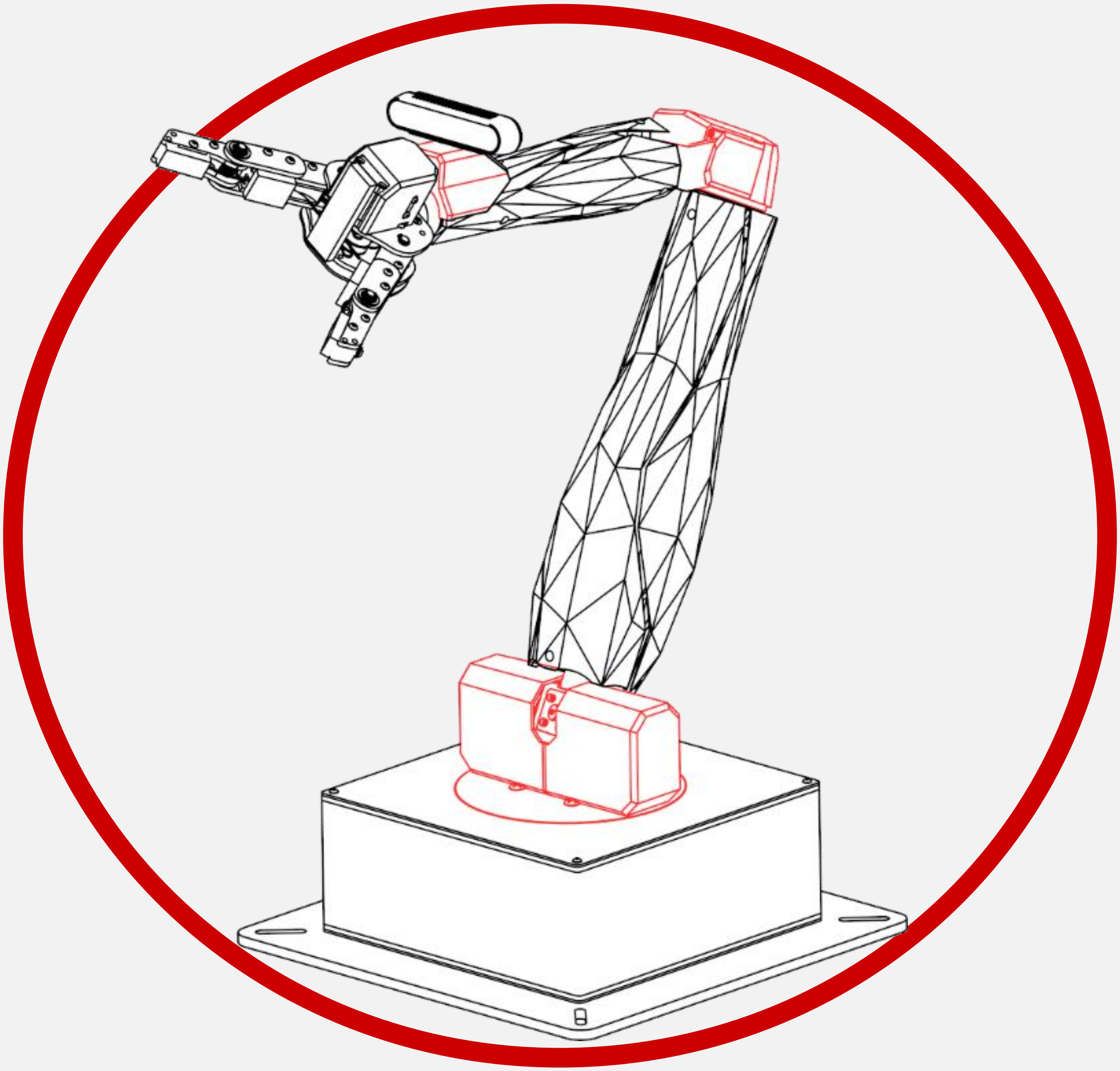

QArm

Visual Servoing

V1.0 – 13th February 2025

Quanser Consulting Inc.
119 Spy Court
Markham, Ontario
L3R 5H6, Canada

info@quanser.com
Phone : 19059403575
Fax : 19059403576
printed in Markham, Ontario.

Caution

This equipment is designed to be used for educational and research purposes and is not intended for use by the public. The user is responsible for ensuring that the equipment will be used by technically qualified personnel only. Users are responsible for certifying any modifications or additions they make to the default configuration.

QArm – Application Guide

Visual Servoing

Why explore visual servoing?

Control for a robotic manipulator can come from a pre-defined trajectory, forward kinematic joint tasks or a teach and repeat task where an operator teaches the arm a predefined task. Visual servoing utilizes image based information to control the movement of robotic arm. In image acquisition and object detection labs you learn how to extract features of an image for a specific application. In this lab you will expand this knowledge to create machine vision pipeline which will allow the QArm to track a color of your choice.

Background

The QArm content contains 3 labs that focus on visual servoing. The first one focuses on learning how to do image acquisition, the second one in object detection, and the last one focuses on visual servoing, which is moving the arm based on what the camera sees.

Prior to starting this lab, please review the following concept reviews (should be located in Documents/Quanser/4_concept_reviews/),

- Concept Review – Camera Model (Camera Intrinsics and Pinhole Camera Model).

QArm Camera Representation

For our application we will use the QArm's Intel RealSense D415 camera. Ideally the camera calibration process also yields the camera distortion parameters. For the purposes of this lab we will take the raw image from the RGB sensor and apply the following calculations:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K \begin{bmatrix} X_p \\ Y_p \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} K^{-1} = \begin{bmatrix} X_p \\ Y_p \\ 1 \end{bmatrix} \tag{2}$$

This assumes the camera has no distortion present. Vector $P_I = \left[X_p \; Y_p \, 1\right]^T$ represents the distance normalized physical coordinates of an object in space. To project this distance back into 3D coordinates we will need to scale vector these coordinates based on some depth value.

We will use the depth measurements from the Intel RealSense D415 to scale vector P_I into 3D coordinates. Before we can scale P_I we have to identify the coordinates for the centroid of the object we wish to track. In Lab 10 – Object Detection we looked at how to identify objects of a specific color.

We also need to be careful when we scale vector P_I. 3D cameras have a offset between the RGB an Depth measurements which can be problematic when sampling a depth measurements from coordinates obtained from an RGB camera. We need to align the two images first before a depth value can be used to scale X_p and Y_p.

Figure 1: RGB and Depth images from D415 with Depth unaligned.

Figure 2: RGB and Depth images from D415 with Depth aligned.

With the pixel coordinates of the centroid we can sample our depth aligned image to identify the distance to the object of interest. Given the pixel coordinates of the centroid and their corresponding depth measurement we can scale P_I using equations:

$$\begin{bmatrix} X_p \\ Y_p \end{bmatrix} = \begin{bmatrix} X/Z \\ X/Z \end{bmatrix}$$

$$\begin{bmatrix} X_p Z \\ Y_p Z \end{bmatrix} = \begin{bmatrix} X \\ X \end{bmatrix} \tag{3}$$

One consideration is vector $P_c = [X\ Y\ Z]^T$ is calculated with respect to the camera frame defined by the pinhole camera model. Given the QArm has an axis definition at the wrist we will need to make a change in the frame of reference for vector $P_c = [X\ Y\ Z]^T$ to ensure motion commands align with the correct axis definitions.

QArm Reference Frame Assignments

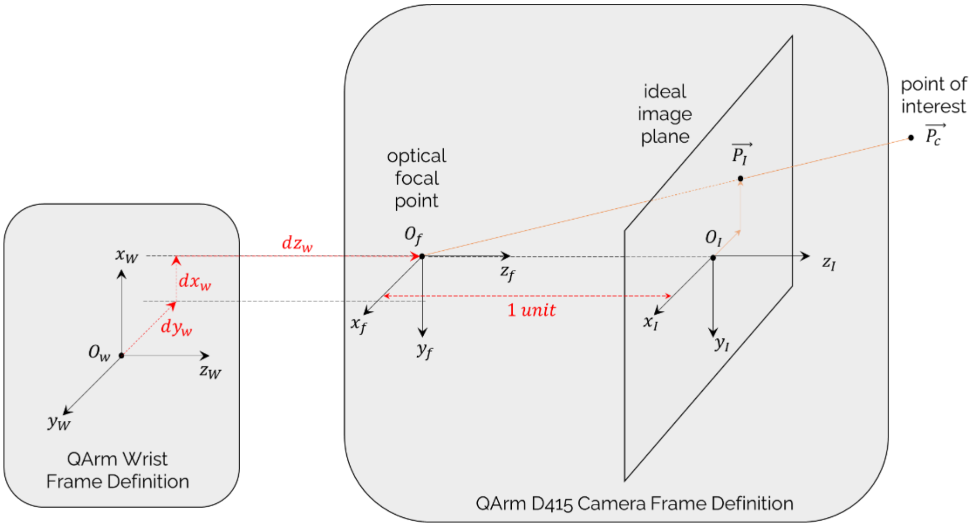

Figure 3: Reference Frames in QArm wrist

For our visual servoing application the point of interest P_C will need to be converted from the camera frame definition to the QArm wrist frame definition. The rotation matrix which aligns the camera frame to the QArm wrist can be calculated as:

$$R_C^W = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4}$$

A rotation of $\theta = 90°$ aligns the wrist with the camera frame. The final rotation matrix becomes:

$$R_C^W = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{5}$$

Since we are using the RGB camera of the QArm to track an object of interest we also need to include the offset between the RGB sensor and the center of the wrist on the QArm. In Figure 3 these offsets are defined as $t_c^w = [dx_w \ \ dy_w \ \ dz_w]$. This final offset allows us to fully describe P_c from the perspective of of the QArm Wrist.

We include the final offset into our homogeneous transformation matrix, which becomes:

$$T_C^W = \begin{bmatrix} 0 & -1 & 0 & dx_w \\ 1 & 0 & 0 & dy_w \\ 0 & 0 & 1 & dz_w \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{6}$$

An example of a point of interest seen by the camera $P^c = [X_c \, Y_c \, Z_c]$ but expressed in the reference frame of the wrist can be written as:

$$P^W = T_c^W P^c \tag{7}$$

We can use this point of interest to drive the motion of the QArm using the differencial inverse kinematics.

Visual Servoing Control

With a point of interest transformed to the frame of reference of the wrist the next step is to define a control strategy for controlling the motion of the QArm in a safe way. As seen in Lab 1 – Low Level Control, a PID controller is an intuitive way of ensuring a measured signal reaches a desired setpoint. For this application the desired setpoint will be the origin of the wrist with an offset along the z direction. The following control structure will be used:

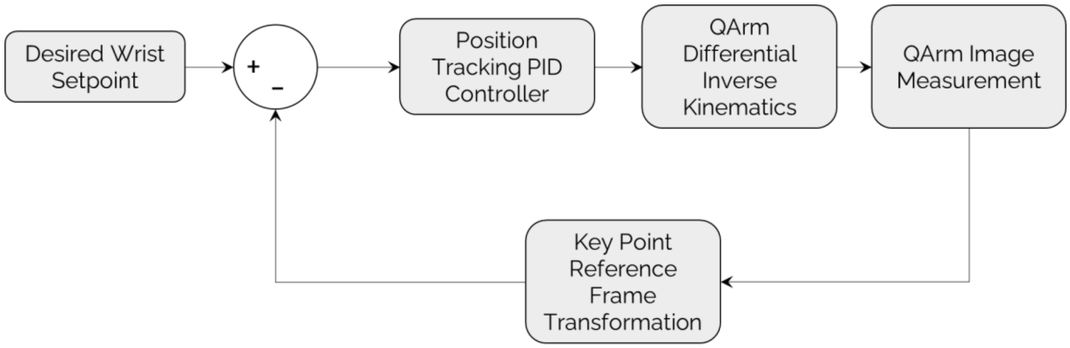

Figure 4: Flow diagram for visual servoing control architecture.

Safety

For safety the QArm includes the following statemachine to ensure tracking of an object is only valid when an object is detected:

State Number	Description
0	Idle
1	Tracking Ready
2	Motion
9	Error

Table 1: State machine nomenclature

Signal flow for the state machine looks as follows:

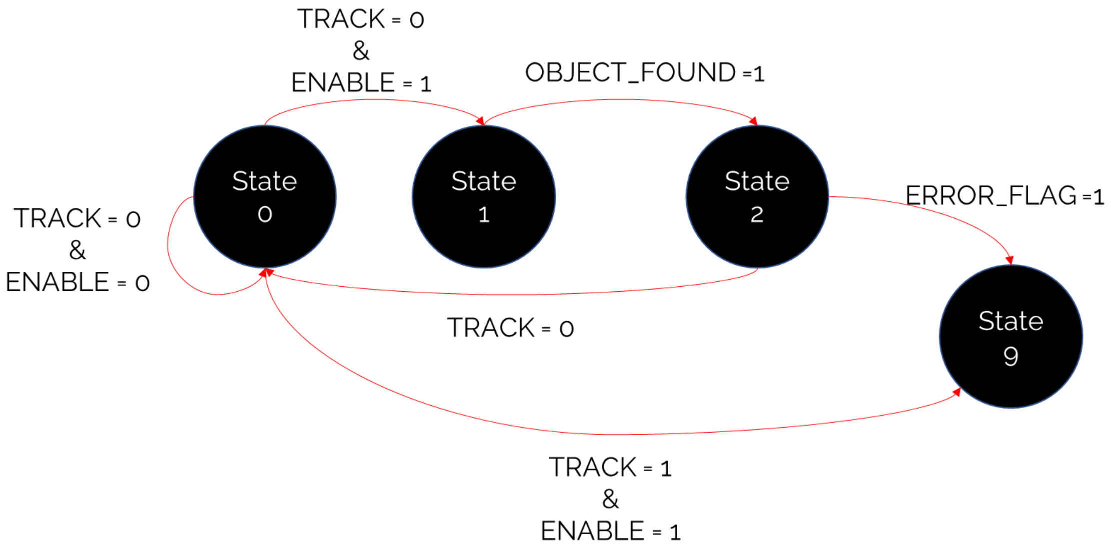

Figure 5: Visual servoing safety state machine

Getting started

The goal of this lab is to use the QArm to track an object using visual servoing. A Simulink model will be used to read the image, find the blob and start using kinematics to track the object using the QArm.

Ensure you have completed the following labs

- **Image Acquisition Lab**
- **Visual Servoing Lab**

Before you begin this lab, ensure that the following criteria are met.

- The QArm has been setup and tested. See the QArm Quick Start Guide for details on this step.
- You are familiar with the basics of Simulink. See the Simulink Onramp for more help with getting started with Simulink.