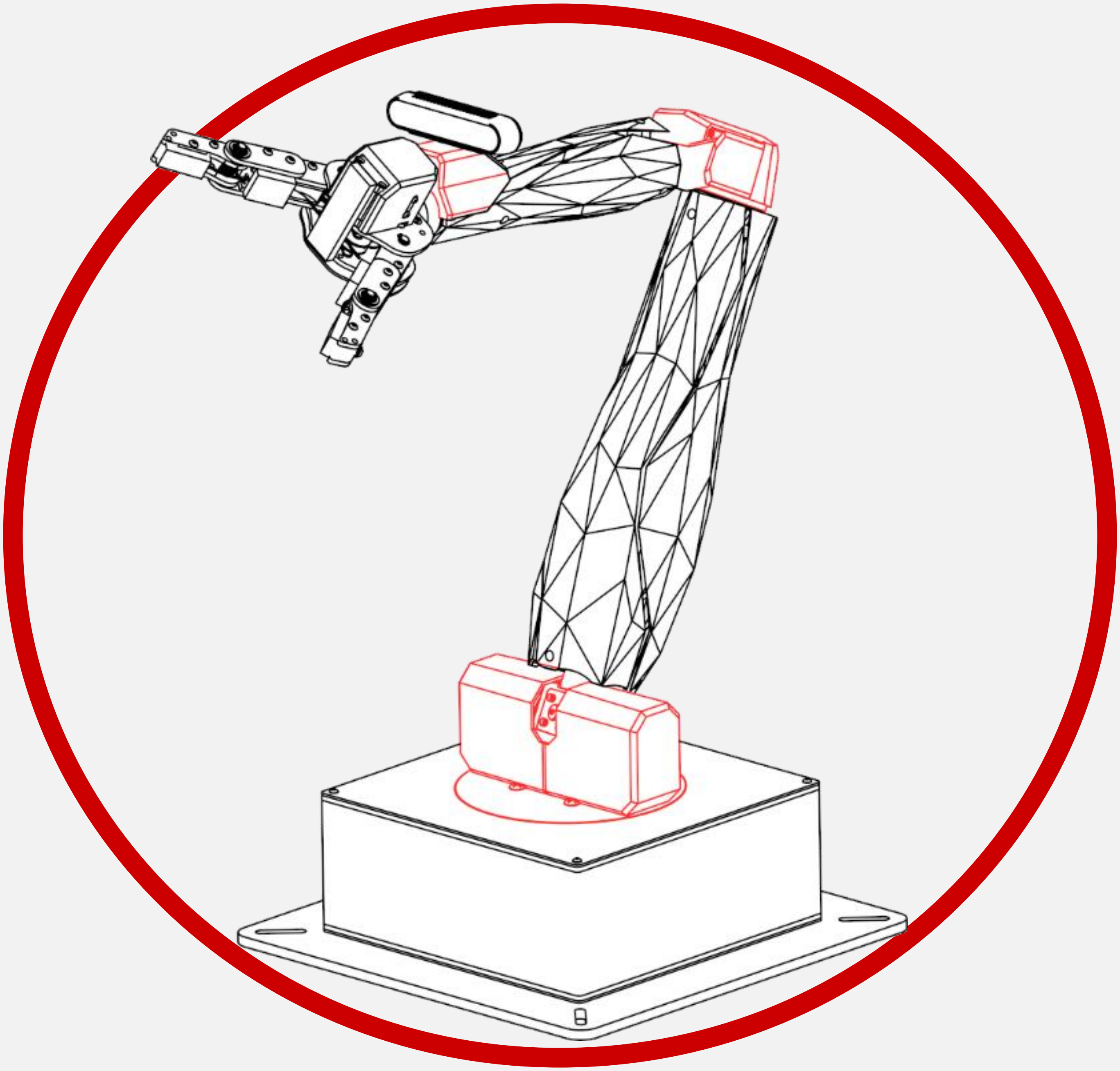


QArm

Tool Manipulation

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
Tool Manipulation

Why explore Tool Manipulation?

In the previous labs, we analyzed the forward and inverse kinematic formulations of a serial manipulator, which related the pose of the end-effector to its joint variables. The speed of the manipulator's joints or end-effector was not considered. For a task such as spray painting, not only do we care about precision in terms of 'where' the manipulator is painting, but also the uniformity with which it paints. To regulate a uniform speed for the end-effector, we must consider the relationship between the joint velocities of the manipulator and the task space velocities of the end-effector. This relationship, called the differential kinematics formulation, involves taking the time derivative of the forward kinematics formulation.

For the previous labs, the low-level joint controllers moved the manipulator to the desired joint-space position. Here, the controller regulates the arm based on a joint space error but does not monitor the task space performance. To regulate task space motion, a state machine is introduced that automates decision making. The manipulator will be commanded to move towards a desired setpoint at a constant rate until it gets close enough, regulating both position and speed.

Differential Kinematics

The forward kinematics formulation that related the joint angles to the end-effector position and orientation are outline below. (Recall that c_1 stands for $\cos\theta_1$, s_1 stands for $\sin\theta_2$, s_{23} stands for $\sin(\theta_2+\theta_3)$ etc.).

$$\begin{gathered} P = f_{FPK}(\Theta) \\ p_x = \lambda_2 c_1 c_2 - \lambda_3 c_1 s_{23} \\ p_y = \lambda_2 s_1 c_2 - \lambda_3 s_1 s_{23} \\ p_z = \lambda_1 - \lambda_2 s_2 - \lambda_3 c_{23} \\ \gamma = \theta_4 \end{gathered} \tag{1}$$

where Θ is the vector of the joint states,

$$\Theta = \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \end{bmatrix} \tag{2}$$

and p is the position of the end-effector {4} (expressed in base frame {0})

$$\mathrm{p} = \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} = {}^0p_4 \tag{3}$$

To formulate the differential kinematics, we differentiate the equation set 1 with respect to time

$$\begin{gathered} v_x = \frac{dp_x}{dt} = -\lambda_2 s_1 c_2 \dot{\theta}_1 - \lambda_2 c_1 s_2 \dot{\theta}_2 + \lambda_3 s_1 s_{23} \dot{\theta}_1 - \lambda_3 c_1 c_{23} \dot{\theta}_2 - \lambda_3 c_1 c_{23} \dot{\theta}_3 \\ v_y = \frac{dp_y}{dt} = \lambda_2 c_1 c_2 \dot{\theta}_1 - \lambda_2 s_1 s_2 \dot{\theta}_2 - \lambda_3 c_1 s_{23} \dot{\theta}_1 - \lambda_3 s_1 c_{23} \dot{\theta}_2 - \lambda_3 s_1 c_{23} \dot{\theta}_3 \\ v_z = \frac{dp_z}{dt} = -\lambda_2 c_2 \dot{\theta}_2 + \lambda_3 s_{23} \dot{\theta}_2 + \lambda_3 s_{23} \dot{\theta}_3 \\ \dot{\gamma} = \frac{d\gamma}{dt} = \dot{\theta}_4 \end{gathered} \tag{4}$$

Grouping the terms based on joint velocities and representing it in matrix form yields,

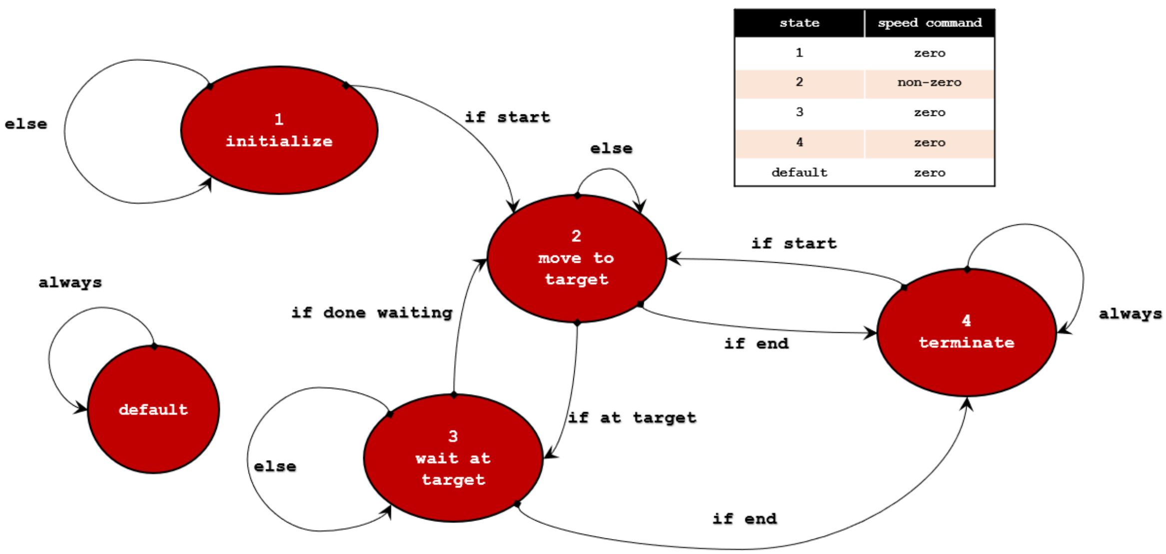
$$\begin{gathered} V = {}^{\mathbf{0}}\boldsymbol{J}\,\dot{\Theta} \\ \begin{bmatrix} v_x \\ v_y \\ v_z \\ \dot{\gamma} \end{bmatrix} = \begin{bmatrix} -\lambda_2 s_1 c_2 + \lambda_3 s_1 s_{23} & -\lambda_2 c_1 s_2 - \lambda_3 c_1 c_{23} & -\lambda_3 c_1 c_{23} & 0 \\ \lambda_2 c_1 c_2 - \lambda_3 c_1 s_{23} & -\lambda_2 s_1 s_2 - \lambda_3 s_1 c_{23} & -\lambda_3 s_1 c_{23} & 0 \\ 0 & -\lambda_2 c_2 + \lambda_3 s_{23} & \lambda_3 s_{23} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \\ \dot{\theta}_4 \end{bmatrix} \end{gathered} \tag{5}$$

where ${}^{\mathbf{0}}\boldsymbol{J}$ is the manipulator Jacobian expressed in base frame {0}.

State Machine

State Machines are abstract computational models that can be in exactly one of many states. When the number of possible states is finite, it is referred to as a Finite State Machine (FSM). In each state, a **state transition** logic determines the next state, and the **state action** logic determines the output. Both the state transition and state action logic vary from state to state and may also depend on the inputs to the state machine. Typically, there will be an initial state where the execution starts and a final state to handle graceful termination.

A state machine diagram can be used to outline all the states, and the general transition and action logic as pseudocode. For the one you will use in this lab, a state diagram is provided in Figure 1.



state	speed command
1	zero
2	non-zero
3	zero
4	zero
default	zero

Figure 1. State Machine diagram

State 1 is an initialization state, and state 4 is one for termination. In state 2, the manipulator moves towards the desired task space position at a uniform task space speed. Once it is close enough (that is, the task space position error is less than some threshold), the state machine moves to state 3 where it waits for a fixed duration. By varying parameters such as maximum speed, wait duration etc., you should be able to replicate tasks such as welding, assembly etc.

Note that state machines are laid out structurally as shown above, but users can choose their own implementation. You can use switch-cases, if-else statements, while loops, etc., so long as the state diagram logic is met.

Background

The QArm content contains 3 labs that focus on velocity manipulation. They focus on tool manipulation, singularity identification and singularity avoidance respectively. This lab focuses on understanding differential kinematics to get task space velocities of the end effector.

Getting started

The goal of this lab is to understand how to avoid singularity positions based on the singularity identification lab before.

Ensure you have completed the following labs

- **Kinematic Manipulation Labs**

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.