

Homework 3

Problem 1: Forward and Inverse Kinematics, and Resolved-Rate Motion Control (RRMC)

Consider the PUMA 560 robot.

1.1. Obtain a photo of the robot. The DH parameters of the PUMA robot are available. Explain, using the DH modeling convention, how all the PUMA a , d , θ and α DH parameters are found.

1.2. Recall the task (that was discussed in class) in which the robot is moved at high speed, from its zero position P_0 to a place P_1 above the part that, in later steps, needs to be picked up. Pretend that the robot is a PUMA and also pretend that you (the operator who programs the robot) see the part lying on the conveyor at a repeatable position and orientation P_3 . Use the teach pendant to move the robot to the desired P_1 configuration, right above the part. Then use the “fkine” forward kinematics command to find the tool’s homogenous transformation matrix (with respect to the robot’s base.

1.3. Create, using the teach pendant and the “fkine” command, the homogeneous transformation matrix of the part’s pose P_3 , and also create the homogeneous transformation of an intermediate position P_2 between P_1 and P_3 . The PUMA is supposed to move slowly and carefully from P_1 to P_3 (via P_2). Use “ikine” (or “ikine6s”) inverse kinematics command to check if the inverse kinematics solutions agree with what was obtained during teaching-by-doing.

1.4. Let us assume a different scenario (than the one described in 1.2-1.3). Say that the homogeneous transformations for P_1 (start of the careful motion) and P_3 (end of the careful motion), that you have found earlier, were obtained using a camera and real-time machine vision (rather than by teaching-by-doing as done in 1.2-1.3). Plan, using Peter Corke’s Simulink model of RRMC, a RRMC controlled motion to descend slowly from P_1 to P_3 by a straight-line motion along the Z axis. Plot the robot tool’s cartesian motion, and also plot the PUMA joint variable angles (as function of t).

Problem 2: Mobile Robots Closed-Loop Controlled Motion

2.1. Demonstrate in Simulink how a differential-drive mobile robot moves, using closed-loop feedback control, from an initial pose $[x, y, \theta] = [7, 5, \pi/8]$ to a final position of $[x, y] = [20, 15]$, in which the vehicle's orientation is unimportant. Plot the vehicle's XY trajectory, and also plot $x(t)$, $y(t)$ and $\theta(t)$, $V_l(t)$ and $V_r(t)$. You may use Corke's Simulink model (but replace the bicycle by a differential-drive vehicle) or use your own Simulink model.

2.2. Demonstrate in Simulink how a synchro mobile robot (also known as unicycle) moves, using closed-loop feedback control, from an initial pose $[x, y, \theta] = [2, 3, 0]$ to follow a line of coefficients $[3, -3, 4]$. Plot the vehicle's XY trajectory, and also plot $x(t)$, $y(t)$, $\theta(t)$, and the two control signals. You may use Corke's Simulink model (but replace the bicycle by a differential-drive vehicle) or use your own Simulink model.

Problem 3: Joint Space Motion Planning

A 3DOF XYZ manipulator consists of three prismatic joints that move simultaneously for a variety of tool positioning tasks. The maximum velocity and acceleration of joint X are 0.7[m/s] and 3.1[m/s²], respectively. The maximum velocity and acceleration of joint Y are 0.4[m/s] and 2.1[m/s²], respectively. The maximum velocity and acceleration of joint Z are 1.0[m/s] and 1.0[m/s²]. Need to plan trapezoidal velocity profiles for X, Y and Z as the (X,Y,Z) tool coordinates change from (4 , 1, 1) to (-2 , 3.5, 1) [m] (each).

3.1. Decide what should be the synchronized motion time T, whether each velocity profile be trapezoidal or triangular and what should be the acceleration/deceleration times. Also provide information about the chosen velocities and accelerations.

3.2. Use MATLAB "jtraj" command (or another related command such as "mtraj") to implement the needed motion. See if the RTB command that you choose allows you to integrate all the velocity and acceleration limits, or if it would lead to a solution that is different than the one that is obtained in 3.1.

Problem 4: DC Motor Control

Use lectures notes "DC Motor Simulation" and "Simulation DC Motor PID Tuning" as references and use the posted Simulink models (in Canvas module "Fall 2021 Resource Material") so that

you don't have to recreate the Simulink diagram, just to slightly modify existing diagrams. No need to use the motor's simplified transfer function – the provided Simulink diagrams provide exact models of the motor.

Replace the smaller motor E-508A parameters by a larger motor E-510C parameters (based on the data sheets posted in the lecture).

In both assignments (below) **the goal is to advance the motor's angle θ from an initial angle $\theta(0) = 0.1[\text{rad}]$ to a final angle $\theta(t_f) = 0.5 [\text{rad}]$, in $t_f = 35\text{ms}$, or as fast as possible.** Assume that the total moment of inertia (motor plus load) is $J = 10J_m$, where J_m is listed in the data sheets. In each of the two tasks check that the motor's peak torque value and motor's maximum pulse current value are not exceeded. If needed slow the motor down a little. Also be sure that the output angle does not overshoot by more than 10%.

In each of the problems below, show the Simulink diagram, show all the relevant graphical results and add brief explanations.

4.1 Use the DC motor in open loop control. That is, leave the feedback of Ω via K_E , but do not create an additional unity feedback for Ω . No need to use any PID controller block. The task is to create a specific trapezoidal velocity profile input command signal, so that the motor's angle moves by the right amount by the specified timing.

4.2 Use the DC motor in closed loop velocity control. That is, use the given diagram in which you leave the feedback of Ω via K_E , and add unity feedback for Ω . Need to tune a PID controller block. Again, the task is to create a specific trapezoidal velocity profile input command signal, so that the motor's angle moves by the right amount by the specified timing.

Submission Deadline: Friday 11/18/2021 by 11:59 PM.