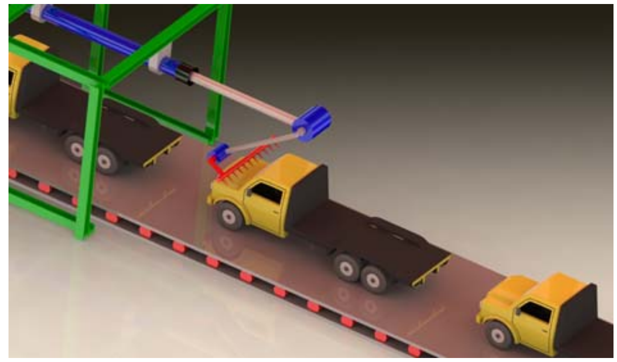
ELEC 341 Project 2016 - Automated Paint Booth

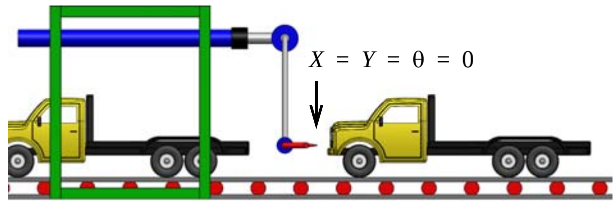
The University of British Columbia

An assembly line paints flat-deck trucks using a robotic Prismatic / Revolute / Revolute (PRR) painting arm. The position and angle of the paint gun follows a prescribed trajectory that corresponds to the trucks' geometry as they move by at constant speed on a conveyor belt.



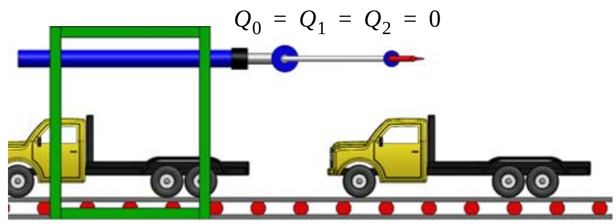
Your task is to:

- Model the inverse kinematics of the robot arm.
- Model the dynamics of the robot arm.
- Design a controller for the robot arm that:
  - Keeps the tip of the paint gun a maximum of **10cm** from its desired position so that paint is applied uniformly.
  - Paints the maximum number of trucks per hour.



Robot Specifications

The prismatic joint (Q0) is a simple hydraulic cylinder that is actuated by an electronic pump.

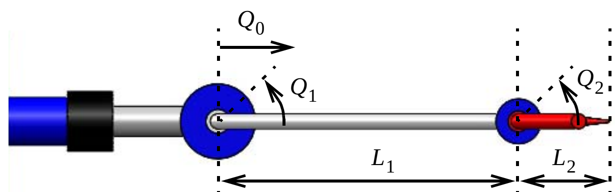


Pump

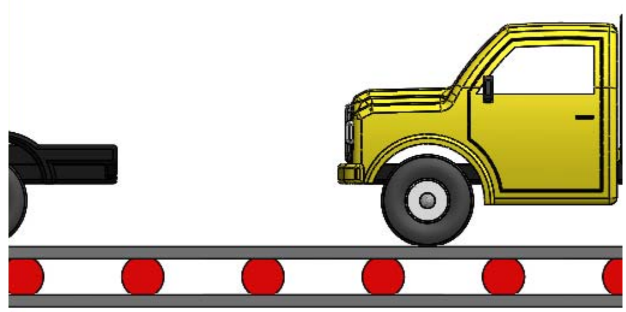
- Gain PK_0 = 600 PSI/V
- Time constant PT_0 = 10 ms
- Maximum output pressure PM_0 = ±150 PSI

Cylinder : Joint Q0

- Motion range Q_0 = -0.5 m to +4.5 m
- Cylinder Inner Diameter ID_0 = 25 cm
- Mass and Friction are unknown but the impulse response was measured experimentally.



The first and second motors are identical and driven by identical amplifier circuits, as shown. However, each motor drives a different gearbox.

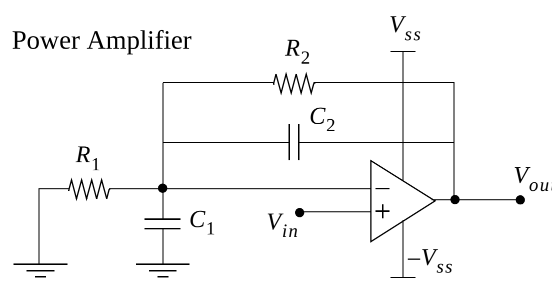


Amplifier

- Resistance R_1 = 1 KΩ
- Resistance R_2 = 4 KΩ
- Capacitance C_1 = 5 uF
- Capacitance C_2 = 2 uF
- Source Voltage V_{SS} = 480 V

Motor

- Motor mass MM_1 = 25 Kg
- Motor rotor inertia MJ_1= 160,000 gcm^2
- Motor friction MB_1 = 650 gcm^2/s
- Motor internal inductance ML_1 = 1.2 mH
- Motor internal resistance MR_1 = 75 mΩ
- Motor torque constant MT_1 = 5,000 mNm/A
- Motor back EMF constant MB_1 = 25 RPM/V



The first joint carries the weight of the first link and the second motor and gear and must also overcome gravity which is why a gear with a high ratio is used.

Joint Q1

- Link Length L_1 = 2.5 m
- Link mass M_1 = 780 g
- Motion range Q_1 = -180° to +90°
- Gear mass GM_1 = 17,000 g
- Gear inertia GJ_1= 98 gm^2 (at input)
- Gear friction GB_1 = 2 gm^2/s
- Gear ratio GR_1 = 50:1

The second joint only carries the weight of the paint gun (shown in red) which is negligible so a gear with a lower ratio is used.

Joint Q2

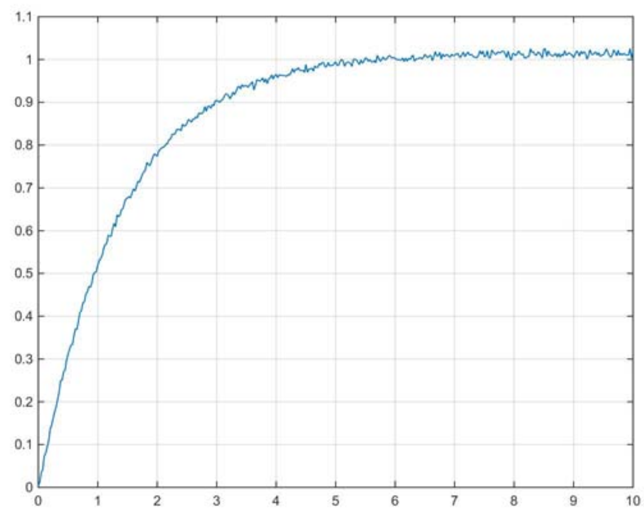
- Link Length L_2 = 0.75 m
- Motion Range Q_2 = -144° to +144°
- Gear mass GM_2 = 7,000 g
- Gear inertia GJ_2= 3.3 gm^2 (at input)
- Gear friction GB_2 = 0.65 gm^2/s
- Gear ratio GR_2 = 5:1

The impulse response of the joint Q0 is shown. The amplitude of the input pulse was 5uV. The curve was measured experimentally so some noise is present.

The data from this curve was saved in the file 'ExpData.mat'. Load this file and plot the two variables ImpResp vs TimeVec to re-produce this curve. Use this curve to check your estimated mass and friction values.

Make sure that your values seem reasonable.

Hint: hydraulic cylinders have large sliding surfaces and tight seals whereas motors use a pair of rolling bearings.



General Information:

- The model was created using Matlab Version **R2014b**.
- You may use any Matlab functions to complete this project. Include all Matlab scripts and output in your report. The control toolbox (help control) is useful for solving transfer functions, drawing a root locus or nyquist plot, etc.
- You are provided with the following files:
  - factory.mdl & FACTORY.WRL (Simulink model files of paint factory)
  - robot.mdl & ROBOT.WRL (Simulink model files of robot only - no trucks)
  - CONSTANTS.m contains all the numeric specifications listed above
  - DEFAULT.m contains default values for all system models
  - TRAJECTORY.m assigns the desired robot and conveyor belt trajectory

- Factory.m sets calls the Matlab functions to set up the environment to run the simulation. It is automatically run when you load or run either Simulink model (factory.mdl or robot.mdl).
- Controller.m sets the PID controller gains, the time spent (speed) painting and re-setting the robot position, and the initial position of the truck when painting starts. It over-writes the default values set in DEFAULT.m.
- Robot.m computes the system model. It over-writes the default values set in DEFAULT.m.
- ExpData.mat contains experimental impulse response data for the cylinder Q0.

How to Complete this Project:

System Model

- Modify the values in Robot.m.
  - Use the constants defined in Constants.m.
  - Start by converting all values into the correct physical units. For example, voltages should be specified in Volts, not miliVolts. Consult the physical units specified in the Controller (yellow) block.
  - Compute the values for the numerators, denominators and gains, as specified in each of the individual components in the Controller block. You should over-write the default values set in Default.m. Include detailed comments in Robot.m to explain your work.
  - A scope has been included in the Controller block. Use this to probe internal signals during debugging.
  - Evaluate the systems (transfer function, order, poles/zeros, time constant, etc.). Use Matlab functions to do this. You don’t need to compute any of this by hand.

Inverse Kinematics

- Modify the Inverse Kinematics (green) block.
  - Initially, X, Y and Θ are connected to Q0, Q1 and Q2, respectively.
  - Replace these connections with Simulink blocks that calculate the joint values from the cartesian coordinates.
  - When you have done this correctly, the XY Graph and the Computed XY Graph will produce the same plot.
  - Note that the Forward Kinematics block converts joint values into cartesian coordinates (the opposite of what you need to do). Consult this block for an example of how to do this sort of thing.

Controller

- Determine PID controller gains for each robot axis.
  - Compute the open-loop gain of each joint. Replace any non-linear elements with linear approximations. These are non-linear constraints that you cannot model and must compensate with some fine-tuning.
  - Use the techniques you learned in class to choose values for your controllers.
  - Put back all non-linear elements and adjust your controller gains by hand to optimize their response.
  - Use the PID “Tune” button in the PID controller box to produce automatically generated PID values. Compare these to your values.
  - Increase the conveyor speed by decreasing **PaintTime** and **ResetTime** as much as possible without allowing the position error to ever exceed its maximum allowable value. Check the “Maximum Position Error” displayed in the Accuracy block to verify that you satisfy this criteria.

Hints

- The goal of this project is to minimize the values assigned to **PaintTime** and **ResetTime** without exceeding the maximum **PositionError**. Improving your controller gains will allow you to reduce these values, and paint more trucks each hour.
- Change the **StartX** value in Controller.m if joint Q0 starts running into its limit.
- Create a copy of “factory.mdl” and name it something else like “fast.mdl”. Erase the ‘3D Graphic’ block in the top level of the model. This model will run faster which will be useful when optimizing (tuning) your system.
- Connect a Scope block to any signal in the model to see what its value is when you run the model. Use a multiplexor to see more than 1 signal in the same scope window. This is useful for general debugging.
- Double-click any of the elements in the Controller block to see how the values specified in Robot.m are linked to it.
- A throughput of 300 trucks/hour is easily possible and over 700 trucks/hour with careful tuning.
- Any file with its name in CAPITAL LETTERS is one that you should not modify.

Part 1: Informal Report - Paper Copy

1. Hand written calculations must be VERY neat. Your report should include:
   - Calculations for all of your transfer functions and gain values.
   - Printout of your Robot.m file. Comments should include the results obtained above.
   - Printout of your InverseKinematics block and the equations it is solving.
   - Printout of any matlab functions you wrote to do your evaluations.
   - Printout of any Root Loci or other evaluations that you did.
   - Estimated Mass and Friction of Q0, including a comparative impulse response WRT the experimental plot provided.
   - Printout of the step response of each joint using the automatically generated PID gain values (Tune button).

Part 2: Informal Report - Paper Copy + Email

1. Your report should include:
   - **Title page** specifying:
     - The names of the student or pair of students who did the work.
     - The throughput of your system in units of Trucks/hour.
   - **Description** of the steps you took to design your PID controllers:
     - Strategy
     - Starting Point
     - Tuning Process - show a few intermediate results
   - **Printout** of Controller.m
2. Email the **Project TA** (NOT the instructor) by the project due date:
   - Controller.m
3. Your system must meet the following criteria for the "Highest Throughput" competition:
   - Simulation / Configuration Parameters / Solver = **ode45 (Dormand-Prince).**
   - PID blocks: Derivative divisor (N) = **100.** Refer to the project web page for a description of this parameter.
   - PID values may not contain more than 3 significant figures (trailing zeros are ok)
   - Maximum Position Error not greater than 100cm after painting at least 3 trucks.

Grading:

- This project is worth 30% of your final grade.
- All printed reports AND emails must be received by the due date or the report will be considered late.
- LATE reports will be marked PASS/FAIL (PASS = 50%, FAIL = 0%).
- If you do not get a passing grade on this project, ALL OF YOUR QUIZ MARKS WILL COUNT.
- You may work in teams of 1 or 2. Submit 1 report per team. Both team members receive the same grade.
- The team that completes the task with the highest throughput will receive FULL MARKS (15%) for Part 2.

15% Part 1:

- 25% Inverse Kinematics
- 25% System Model Values (including physical units)
- 25% System Model Calculations
- 25% System Description & Evaluations

15% Part 2:

- 50% Starting Point & Tuning Process
- 40% Clarity of Explanation (including plots)
- 10% Resulting Throughput