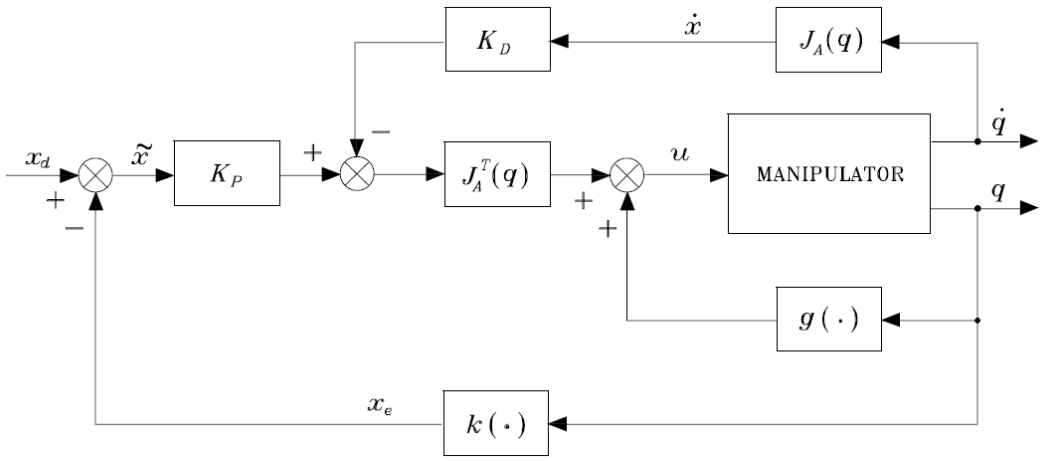
MAE C163C / C263C Homework #5

*(Due via Gradescope by **11:59pm on Friday, 5/16**)*

1. Operational space PD control with gravity compensation



The given block diagram describes operational space PD control with gravity compensation.

Consider a two-link planar arm with the following parameters:

$$a_1 = a_2 = 1\,\text{m} \quad \ell_1 = \ell_2 = 0.5\,\text{m} \quad m_{\ell_1} = m_{\ell_2} = 50\,\text{kg} \quad I_{\ell_1} = I_{\ell_2} = 10\,\text{kg}\cdot\text{m}^2$$

$$k_{r1} = k_{r2} = 100 \qquad m_{m1} = m_{m2} = 5\,\text{kg} \qquad I_{m1} = I_{m2} = 0.01\,\text{kg}\cdot\text{m}^2$$

The arm is assumed to be driven by two equal actuators with the following parameters:

$$F_{m1} = F_{m2} = 0.01\,\text{N}\cdot\text{m}\cdot\text{s/rad} \qquad R_{a1} = R_{a2} = 10\,\text{ohm}$$

$$k_{t1} = k_{t2} = 2\,\text{N}\cdot\text{m/A} \qquad k_{v1} = k_{v2} = 2\,\text{V}\cdot\text{s/rad}$$

a) Express the direct kinematics equation using link lengths and joint angles. Derive the analytical Jacobian.

b) Consider the addition of a point mass load of 10 kg. Design a <u>single</u> closed-loop stable PD controller with gravity compensation that works for both of the following postures: $\boldsymbol{p} = [0.6 \quad -0.2]^T$ and $\boldsymbol{p} = [0.5 \quad 0.5]^T$.
For each of the two postures, initialize the x and y coordinates of the end effector position at t = 0 with values that are perturbed from the desired posture by $[-0.1 \quad -0.1]^T$.
Design the controller in the provided `HW5_P1.py` file. Implement the control in discrete-time with a sampling time of 1 ms and simulation duration of 2.5 s.

1. Complete the `calc_analytical_jacobian` and `calc_direct_kinematics` Python functions in the `HW5_P1.py` file using the results from part (a).
2. Connect the controller block diagram in the `OperationalSpacePDControllerWithGravityCompensation` class of `HW5_P1.py` based on the block diagram provided above.

3. Connect the simulation block diagram inside the `run_simulation` function of `HW5_P1.py`.
4. Fill in the appropriate values in `HW5_P1.py` in the section labeled "`TODO: Problem 1 - Part (b)`".
5. Run simulations of your controller for both cases of desired postures.
6. Plot the controller and simulation block diagrams.

c) Report your K_P and K_D gain matrices.

d) Simulate the closed-loop response of the system. For each desired posture, plot the time history of the x and y coordinates of the end effector position in separate subfigures for a time horizon of 2.5 sec. Indicate the desired coordinate value in each subfigure by drawing a dashed horizontal line at the desired value and show that your controller drives each end effector position coordinate to the desired value within the allotted time.

e) State at least one reason why you cannot make your K_P gain values too large.

Summary of deliverables:

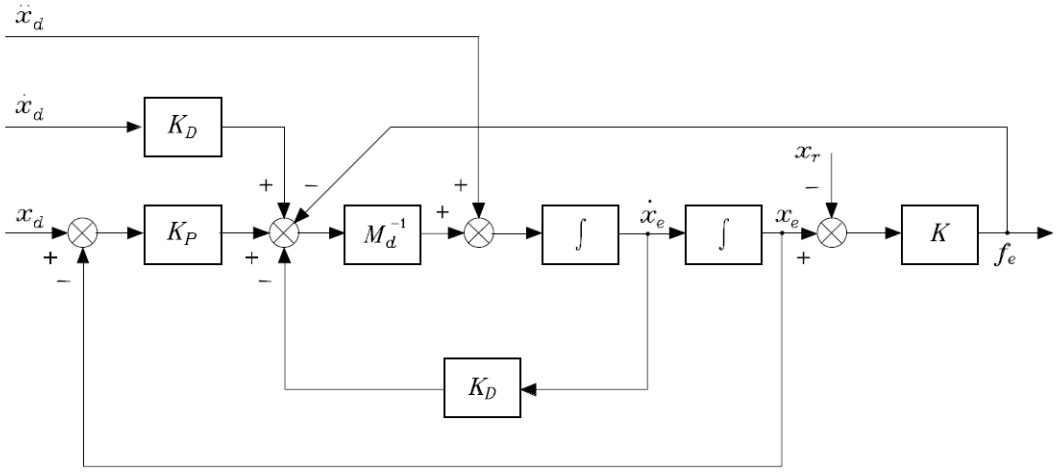
Your submission should include:

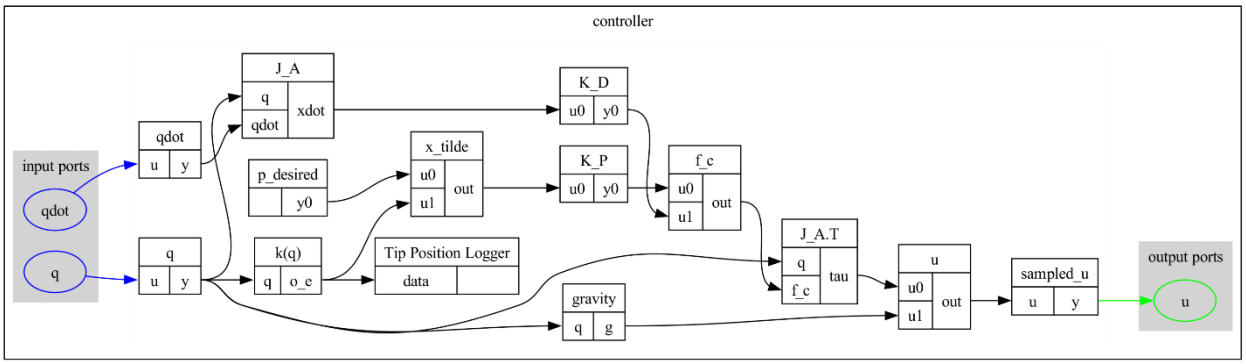
- Direct kinematics and analytical Jacobian
- Your K_P and K_D gain matrices
- Labeled time history plots
- Comments requested in part (e)
- A plot of your final Drake controller block diagram
- A plot of your final Drake simulation block diagram
- Your **completed** `HW5_P1.py` file converted to a PDF (To facilitate grading, see the relevant PyCharm help page for how to print a *.py* file to a PDF).

NOTE: Each student must submit their own independent work. **For full credit, you must submit to Gradescope all custom Python code and requested plots with labels.** You may save this content to PDF or take screenshots for electronic submission via Gradescope. Files of the *.py* and *.toml* format cannot be directly uploaded to Gradescope and should not be e-mailed to instructors for grading. The more intermediate results and comments you provide, the greater the opportunity for partial credit.

2. Impedance control for interaction with an elastic environment

The block diagrams that follow describe impedance control for a manipulator interacting with an elastically compliant environment.





Consider a two-link planar arm with the following parameters:

$$a_1 = a_2 = 1\,\text{m} \quad \ell_1 = \ell_2 = 0.5\,\text{m} \quad m_{\ell_1} = m_{\ell_2} = 50\,\text{kg} \quad I_{\ell_1} = I_{\ell_2} = 10\,\text{kg}\cdot\text{m}^2$$

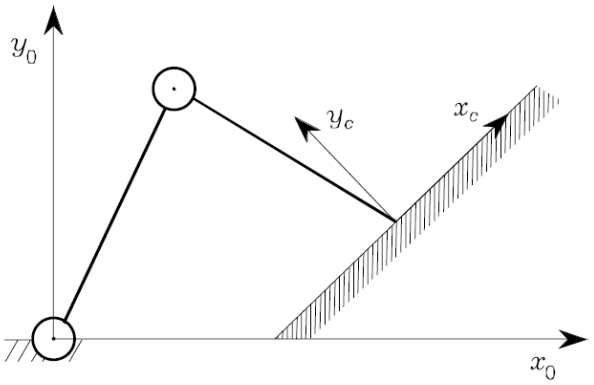
$$k_{r1} = k_{r2} = 100 \qquad m_{m1} = m_{m2} = 5\,\text{kg} \qquad I_{m1} = I_{m2} = 0.01\,\text{kg}\cdot\text{m}^2$$

The arm is assumed to be driven by two equal actuators with the following parameters:

$$F_{m1} = F_{m2} = 0.01\,\text{N}\cdot\text{m}\cdot\text{s/rad} \qquad R_{a1} = R_{a2} = 10\,\text{ohm}$$

$$k_{t1} = k_{t2} = 2\,\text{N}\cdot\text{m/A} \qquad k_{v1} = k_{v2} = 2\,\text{V}\cdot\text{s/rad}$$

Consider the planar arm to be in contact with a frictionless, elastically compliant plane that forms an angle of $\pi/4$ with base frame axis x_0 and whose undeformed position intersects axis x_0 at base frame coordinates of [1, 0]. The environment stiffness along axis y_c is 5000 N/m.



a) Design a closed-loop stable impedance controller that interacts with the elastically compliant environment and moves the manipulator along a rectilinear path from an initial end-effector position $\underline{p}_i = [1 + 0.1\sqrt{2};\ 0]$ (in m) to a final end-effector position $\underline{p}_f = [1.2 + 0.1\sqrt{2};\ 0.2]$ (in m) with a trapezoidal velocity profile and a trajectory duration of 1 sec. Design the controller in Drake using the provided `HW5_P2.py` file.

Implement the control in discrete-time with a sampling time of 1 ms. Use a time horizon of 2.5 sec. The response should stabilize by t = 2 sec or earlier. While there are infinite possible solutions, it is possible to select diagonal M_d, K_D and K_P matrices in the constraint frame such that the system stabilizes within 1-1.5 sec if you design a closed-loop system that is nearly critically damped. Report your M_d, K_D and K_P matrices designed in the constraint frame.

1. Complete the `calc_analytical_jacobian` and `calc_direct_kinematics` Python functions in the `HW5_P2.py` file using the results from Problem 1, Part (a).
2. Connect the controller block diagram in the `OperationalSpaceImpedanceController` class of `HW5_P2.py` based on the block diagrams provided above.
3. Connect the simulation block diagram inside the `run_simulation` function of `HW5_P2.py`.

4. Fill in the appropriate values in `HW5_P2.py` in the section labeled "`TODO: Problem 2 - Part (a)`".

   Hint: Any matrix X^c in the rotated x_c-y_c constraint frame can be transformed into the x_0-y_0 base frame using $X = R_c X^c R_c^T$ where R_c is the rotation matrix that rotates the x_0-y_0 base frame into the x_c-y_c constraint frame.

b) Report the damping ratio ζ and undamped natural frequency ω_n for each of the x_c and y_c directions.

   Hint: See Example 9.2 in the Siciliano *et al.* course textbook.

c) In a single figure, plot the x- and y-coordinate end-effector position errors in m as a function of time, as expressed in the x_0-y_0 base frame. In another figure, plot the x-and y-coordinate end-effector contact forces in N as a function of time, as expressed in the x_0-y_0 base frame.

d) Repeat *part c* above, but with end-effector position errors and contact forces expressed in the rotated x_c-y_c constraint frame. Note that, when expressed in the rotated constraint frame, the desired end-effector location would be indented past the neutral, rest plane of the compliant environment ($y_{c,r}$) by 0.1 m along the y_c direction.

e) Describe the two plots expressed in the rotated x_c-y_c constraint frame. Comment on why the transient behavior makes sense according to your controller design. Comment on why the steady-state values for end-effector position errors and contact forces make numerical sense for the given environment stiffness.

Summary of deliverables:

Your submission should include:

- Your M_d, K_D and K_P gain matrices designed in the constraint frame
- Your damping ratio ζ and undamped natural frequency ω_n for each of the x_c and y_c directions
- Labeled time history plots
- Comments requested in part (e)
- A plot of your final Drake controller block diagram
- A plot of your final Drake simulation block diagram
- Your **completed** `HW5_P2.py` file converted to a PDF (To facilitate grading, see the relevant PyCharm help page for how to print a *.py* file to a PDF).

NOTE: Each student must submit their own independent work. **For full credit, you must submit to Gradescope all custom Python code and requested plots with labels.** You may save this content to PDF or take screenshots for electronic submission via Gradescope. Files of the *.py* and *.toml* format cannot be directly uploaded to Gradescope and should not be e-mailed to instructors for grading. The more intermediate results and comments you provide, the greater the opportunity for partial credit.