Homework 5: Input/Output Calculations for a Real Robot

MEAM 520, University of Pennsylvania
Katherine J. Kuchenbecker, Ph.D.

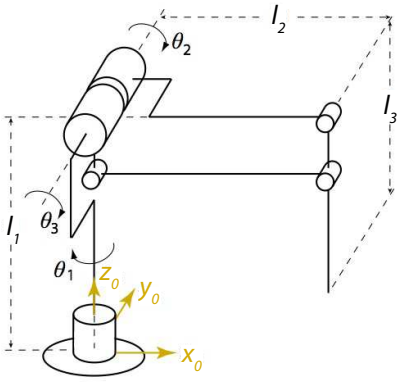
November 13, 2012

This assignment is due on **Tuesday, November 20**, by 5:00 p.m. sharp. You should aim to turn the paper part in during class that day. If you don't finish until later in the day, you can turn it in to Professor Kuchenbecker's office, Towne 224. Late submissions will be accepted until 5:00 p.m. on Wednesday, November 21, but they will be penalized by 25%. After that deadline, no further assignments may be submitted.

You may talk with other students about this assignment, ask the teaching team questions, use a calculator and other tools, and consult outside sources such as the Internet. To help you actually learn the material, what you write down should be your own work, not copied from a peer or a solution manual.

SensAble Phantom Premium 1.0 *(60 points)*

This entire assignment is focused on a particular robot – the SensAble Phantom Premium 1.0. As shown in the photo below left, the Phantom is an impedance-type haptic interface with three actuated rotational joints. Designed to be lightweight, stiff, smooth, and easily backdrivable, this type of haptic interface enables a human user to interact with a virtual environment or control the movement of a remote robot through the movement of their fingertip while simultaneously feeling force feedback.



A thimble is attached to the tip of the robot via a passive non-encoded three-axis gimbal to allow the user to move the robot around while freely rotating their fingertip. As shown in the diagram above right, the Phantom haptic device looks similar to the standard RRR articulated manipulator base, but it uses a unique four-bar mechanism to co-locate the shoulder and elbow joints while also keeping the upper arm and forearm in the plane that intersects the axis of the waist joint.

Each of the four questions below includes both a written explanation and the programming of a specific Matlab function. For the paper parts, write in pencil, show your work clearly, box your answers, and staple your pages together. For the programming, download the starter code from this assignment's page on the class wiki, change all function and file names to include your PennKey, comment your code, and follow the instructions at the end of this document to submit all of your Matlab files.

1. **From Encoder Counts to Joint Angles** ***(15 points)***

   Imagine your boss bought this robot on eBay and has asked you to get it working. It is available on the table next to the PUMA in B2 Towne for you to look at. Note that you do not need to use the PUMA computer for any part of this assignment. You should just be looking at the Phantom to understand how it works and measure any parameters you need. Measuring tools will be available by the robot.

   Each of the Phantom's motors includes a shaft-mounted HEDM-5500-B02 optical encoder. The data sheet for this family of encoders is available on this assignment's page on the class wiki. Drum-and-capstan cable drives are used to connect the motors to the respective joints.

   After figuring out the wiring, imagine you connected all three of the Phantom's incremental optical encoders to a quadrature encoder input card on your computer. You zeroed all of the encoders when the device was in the configuration shown in the schematic on the previous page, where the upper arm is horizontal, the lower arm is vertical, and the tip is located above the x-axis. This is the Phantom's zero configuration.

   As you played with the Phantom, you noticed that the encoder counts increase when each joint is rotated in the direction indicated on the schematic. You decide you will use these positive directions for your joint angles as well. You moved the robot through an interesting near-planar trajectory and recorded the triplets of encoder count values $[Q_1 \quad Q_2 \quad Q_3]^T$ that occurred at 206 points along the way. This list of values is available with the starter code and is automatically loaded into the `phantom_robot_yourpennkey.m` script.

   (a) Based on your inspection of the Phantom and the above information, write a careful explanation of how to convert its encoder counts, $[Q_1 \quad Q_2 \quad Q_3]^T$, to joint angles in radians, $[\theta_1 \quad \theta_2 \quad \theta_3]^T$, including step-wise calculations and any helpful diagrams.

   (b) Implement this function in Matlab as `phantom_counts_to_angles_yourpennkey.m` by modifying the provided starter code.

2. **From Joint Angles to Robot Position** ***(15 points)***

   Now it is time to express the position of the Phantom's key components in terms of its joint angles. Your boss specifically told you to use the coordinate frame indicated on the schematic above (even though it does not comply with DH conventions). Since the Phantom is not a serial manipulator, you realize that DH parameters are not a good method for analyzing its kinematics anyway; simple geometry is a better choice.

   (a) Based on your inspection of the Phantom and the above information, write a precise explanation of how to convert joint angles in radians to the position of all points needed to create a simple stick-figure animation of the robot. At a minimum, your animation should include the origin of frame 0, two other useful points along the robot, and the tip of the robot. Express all positions in frame 0 using millimeters, include the physical values of any constants, and include any diagrams you used to figure this out.

   (b) Implement this function in Matlab as `phantom_angles_to_positions_yourpennkey.m` by modifying the provided starter code. Use the graphing provided in the `phantom_robot_yourpennkey.m` script to check your calculations, and fix any errors you find in this step or the previous one. For example, you can check your work by making sure the joint angles stay within the robot's joint limits and ensuring that the span of the robot's motion is physically realizable. Furthermore, the trajectory of the tip should look like a near-planar motion created by a human user.

3. **From Force Vector to Joint Torques** ***(15 points)***

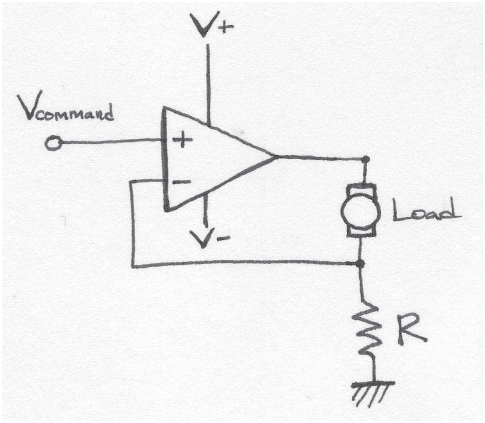
   Now imagine you want to output a force vector at the tip of the robot for the user to feel. The force vector will be specified using newtons in frame 0 as $[F_x \quad F_y \quad F_z]^T$. You remember that a Jacobian Transpose controller is good for situations like this. The robot's joint torques follow the same positive direction convention as the angles; a positive torque will make each joint move in the positive direction.

(a) Write a precise explanation of how to convert a desired three-dimensional force vector to the triplet of joint torques $[\tau_1 \;\; \tau_2 \;\; \tau_3]^T$ in newton·meters (not newton·millimeters) needed to produce that force vector at a given robot configuration $[\theta_1 \;\; \theta_2 \;\; \theta_3]^T$. Include step-by-step calculations.

(b) Implement this function in Matlab as `phantom_force_to_torques_yourpennkey.m` by modifying the provided starter code. You may set the desired force to any vector you want on lines 51 to 53 of the starter code. It should probably stay constant across this simulation to make it easier to debug. A scaled version of this force vector is graphed in magenta to help you visualize the torques needed to create it. The graphing provided in `phantom_robot_yourpennkey.m` should help you debug this function; you are welcome to update the graphing however you want, but please add comments to explain what you did. If you were actually using the Phantom for haptic rendering, you would calculate a force vector at each time step, based on the current tip position, the history of tip positions, and the geometry and dynamics of the virtual environment.

4. **From Joint Torques to DAC Voltages** ***(15 points)***

Lastly, you need to figure out how to create the torques needed on each joint of the robot. All three joints are driven by Maxon 118743 DC brushed motors. The data sheet for this family of motors is available on this assignment's page on the class wiki. Recall that drum-and-capstan cable drives are used to connect the motors to the respective joints.



You wire up each motor to a custom-built linear current amplifier. Each of these amplifiers uses one OPA544 high-voltage high-current operational amplifier (op-amp), configured in the circuit shown above. The motor is connected where the Load is drawn, and you specify the desired motor current by using a DAC card on your computer to apply a command voltage V_{command} to the op-amp's non-inverting input, relative to ground. The op-amp supply voltages are symmetric and within the ranges specified on the OPA544 data sheet. The resistor is a high-precision power resistor with a resistance of $R = 5.00\ \Omega$. When analyzing this circuit to determine the relationship between command voltage and load current, you should assume that the op-amp behaves ideally.

Your three linear current amplifiers cause three respective current to flow through the three motors of the Phantom, $[i_1 \;\; i_2 \;\; i_3]^T$. The current amplifiers are numbered the same as the joints, and you set it up so that positive current makes each joint want to move in its positive direction. You should never command more than the maximum steady-state current the motor can take.

(a) Write a precise explanation of how to convert a triplet of desired joint torques $[\tau_1 \;\; \tau_2 \;\; \tau_3]^T$ into the corresponding triplet of voltage commands $[V_1 \;\; V_2 \;\; V_3]^T$, making sure never to command more current than the motor can take in steady state. Include step-by-step calculations and the values of any physical constants needed.

(b) Implement this function in Matlab as `phantom_torques_to_voltages_yourpennkey.m`, and use the graphing provided in `phantom_robot_yourpennkey.m` to debug this function.

Submitting Your Code

Follow these instructions to submit your code:

1. Start an email to `meam520@seas.upenn.edu`
2. Make the subject *Homework 5: Your Name*, replacing *Your Name* with your name.
3. Attach your five correctly named MATLAB files to the email as individual attachments; please do not zip them together. The files should be the following:
   - `phantom_robot_yourpennkey.m`
   - `phantom_counts_to_angles_yourpennkey.m`
   - `phantom_angles_to_positions_yourpennkey.m`
   - `phantom_force_to_torques_yourpennkey.m`
   - `phantom_torques_to_voltages_yourpennkey.m`
4. Optionally include any comments you have about this assignment.
5. Send the email.

You are welcome to resubmit your code if you want to make corrections. To avoid confusion, please state in the new email that it is a resubmission, and include all of your MATLAB files, even if you have updated only some of them.