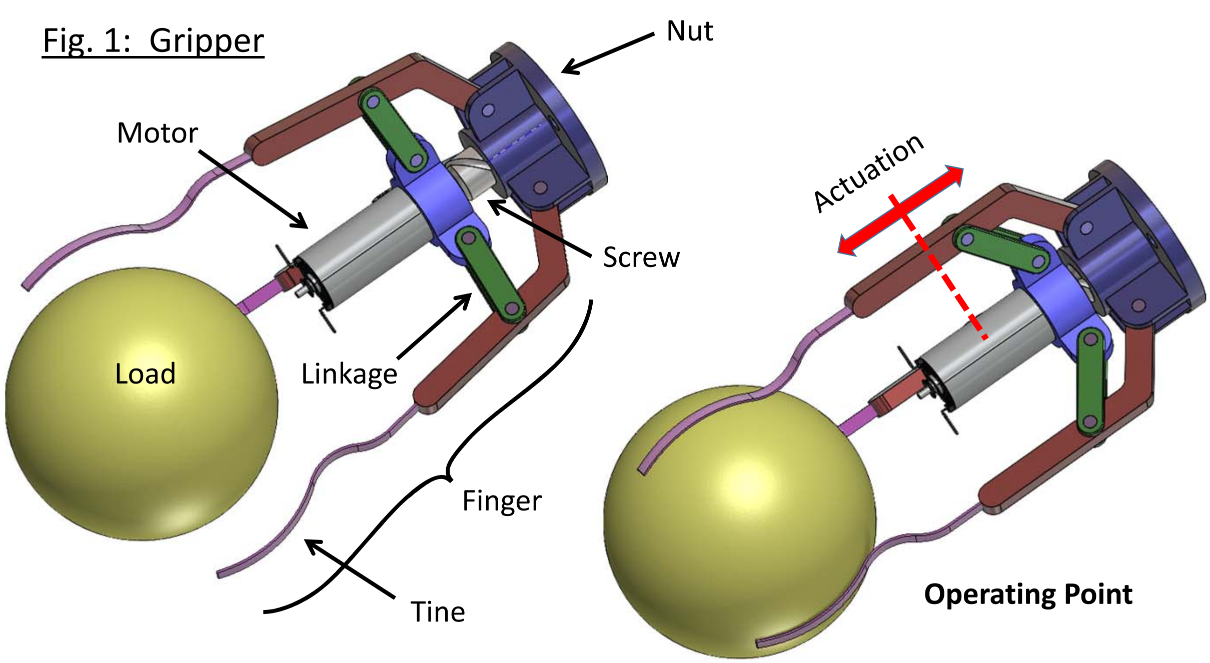
ELEC 341 – Electro-Mechanical Control Project

Project Part 1
System Identification

20 Marks

Fig. 1: Gripper



Develop a control system for a 3-finger gripper for grasping spherical loads. Each finger has a flexible tine to avoid damaging the load. The fingers move in unison so it is sufficient to model one finger assembly and multiply its mechanical impedance by 3.

The gripper uses a DC motor and lead-screw to convert rotation to linear motion (translation) so the entire actuator (motor & screw) acts like a linear actuator. A linkage connected to the motor changes the finger angle when the motor moves back and forth. Changes in finger angle cause the gripper to grasp and release objects.

A gripper is a 1-DOF robot which, like most robots, is not LTI. The relationship between motor speed and finger angle speed depends on finger angle, which changes over time. It's like a gear-box with a variable transmission ratio, like your bicycle. Except, not only do parts move at different speeds, but some translate while others rotate.

To develop an LTI model, transmission ratios are estimated at an operating point, and simplifying assumptions are made. The operating point should be a meaningful point in the workspace, since controller performance will be optimized at this point.

- *The operating point is when the tines first contact the load (see Figs 1-3).*
- *The Linkage mass & inertia is neglected.*
- *The friction Bf includes the friction at all joints "a", "b" & "c" at the operating point.*
- *The tine is assumed to remain connected to the load, even when the gripper opens.*

Non-linearities may be re-introduced and dealt with AFTER the linear model is optimized.

In Fig. 2, L1-L6 are fixed lengths, qr is the rotor angle, dh is the housing position, measured as the distance between the two joints, and qf is the finger angle, measured from the horizontal, at the operating point.

- *From geometry, dh > L2-L3*
- *A mechanical stop in the mechanism further reduces this constraint to avoid the singularity.*

The gripper consists of a motor, lead-screw, linkage, finger, and load. The motor consists of a housing and rotor. The lead-screw consists of a screw and nut which move relative to one another in a helical motion.

- *The nut is the ground reference. It is ultimately connected to the robot arm.*
- *The rotor and screw are connected and rotate in unison.*
- *The motor & screw move back and forth with respect to the nut.*
- *The finger is connected to the nut at joint "a".*
- *The linkage is connected to the housing at joint "b" and the linkage at joint "c".*
- *Most of the finger is stiff. Only the tine component is flexible.*
- *The tine first contacts the load at the operating point.*
- *Rotor angle qr=0, housing position dh=0, and finger angle qf=0, at the operating point.*

The gripper has the effective mechanical impedances shown in Fig 3.

- *The rotor and screw rotate in unison with inertia Jr, Js, and friction Br.*
- *The housing (includes rotor) & screw all translate in unison with mass Mh, Ms.*
- *The finger rotates at pin "a" with inertia Jf and friction Bf.*
- *The tine bends with stiffness Kt and friction Bt.*
- *The lead-screw & linkage introduce transmission ratios between coordinate frames.*

Fig. 2: Geometry

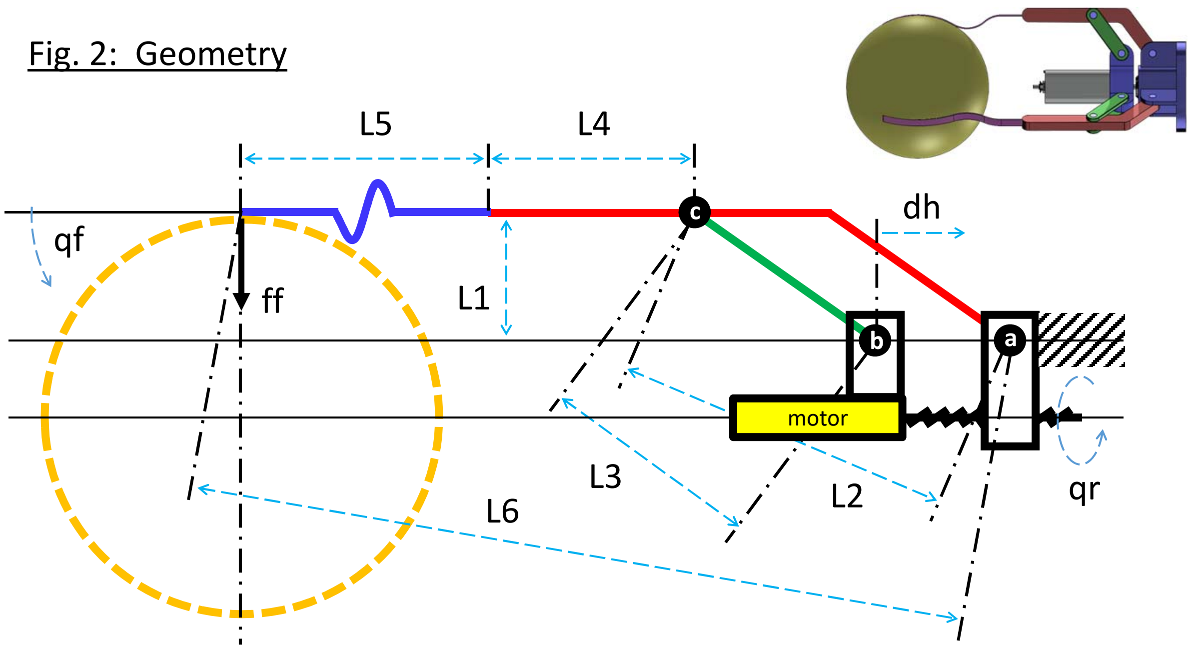
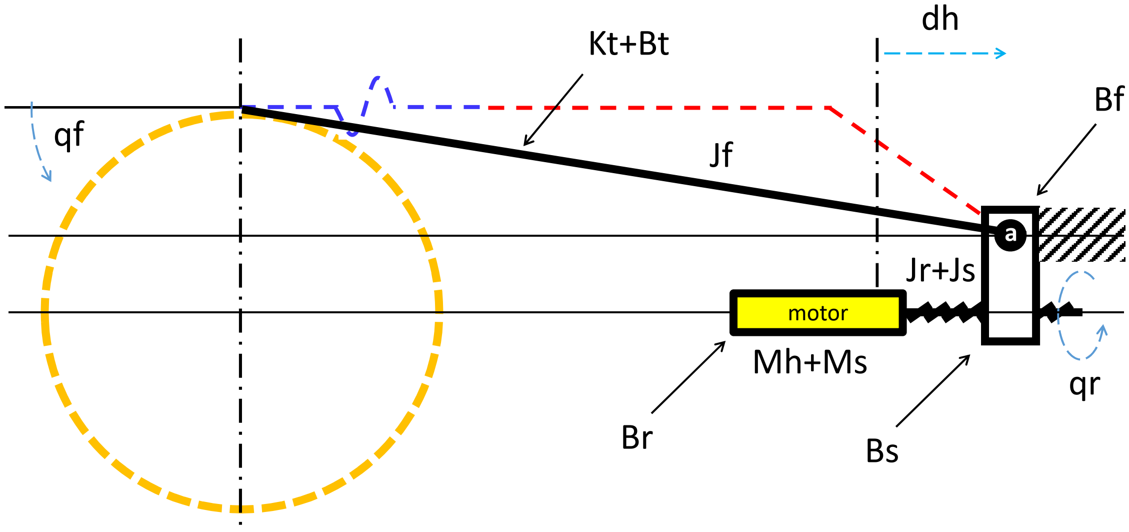


Fig. 3: LTI Model



The motor is supplied by an off-the-shelf (OTS) voltage amplifier.

The data-sheet provides an experimental step response.

*Run **p1DSPlot.p** to plot the step response.*

The data is discrete which isn't ideal, but the manufacturer had to measure it somehow so they used a DAQ with a fixed resolution.

Estimate the continuous curve measured by the DAQ, which you actually can't see.

Q1 1 mark(s) Black-Box Specs

Estimate Rise-Time Tr, Peak Time Tp, Settle Time Ts, and Percent Overshoot Pos.

- Q1.Tr (msec) Scalar
- Q1.Tp (msec) Scalar
- Q1.Ts (msec) Scalar
- Q1.Pos (%) Scalar

Q2 3 mark(s) 2nd Order Approx

Compute the 2nd order approximation that best approximates the raw data.
The approximation should provide a balance between rise and peak time.

- Q2.Ga (V/V) LTI

The motor angle is measured using an OTS position sensor with an internal processor for noise reduction.

The data-sheet provides an equivalent block diagram of the sensor & processor.

*Use block diagram manipulation to find the transfer function **Hs** in **Fig 4**.*

Q3 4 mark(s) Analog Position Sensor

Find Kdc: DC Gain of the sensor
Find H: Input = Angle (deg) Output = Voltage (mV)

- Q3.Kdc (mV/deg) Scalar
- Q3.Hs (mV/deg) LTI

*A pole-zero plot **pzmap()** is something that can be used to check if your transfer function is reasonable for a sensor. You can also plot the response and check the time constant.*

*Poles & zeros that **almost** overlap have a minimal effect on dynamics. Including a tolerance with **minreal()** cancels them out to simplify the math, but also affects **DC Gain**. In practice you can compensate for this but for now, **DO NOT use mineral() at all**. Your pole-zero plot may be a little messier, but your transfer function will be more accurate.*

*You can use **minreal()** just to produce a cleaner pole-zero plot, but you are better off using the original un-simplified transfer function for your calculations and simulations.*

Fig. 4: Sensor Block Diagram

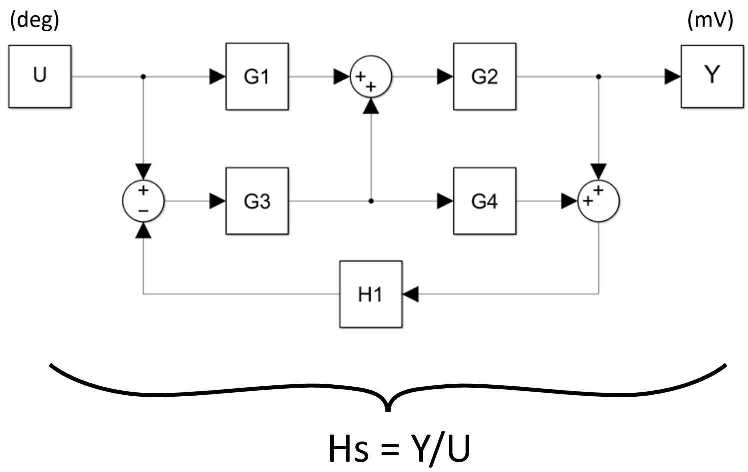
G1 = 100/(s + #Ax500)

G2 = 100/(s + #Bx600)

G3 = 10^5/(s + #Cx700)

G4 = 1500/(s + #Dx800)

H1 = 3/(s + #Fx5)



The system has 3 coordinate frames due to the 2 transmissions in "System Model":
Rotor coordinates (rad, rad/s)
Motor coordinates (m, m/s)
Gripper coordinates (rad, rad/s)

*Do all calculations (A & B matrices) in **Rotor Coordinates** since that is what you are controlling.*

To show results in a different coordinate frame, transform them after all calculations are done. This is what the C and D matrices do. To transform between coordinate frames, simply multiply by the transmission ratio.

*Neglect **Gravity** (grav). This non-linearity may be added later for grasping vertical objects.*

*A Simulink model is provided that represents the complete control system. Check it to see where Gj and Gt appear, and how they are used. Next time you have to develop a system mode, **DO THIS YOURSELF**. You are about to find out how useful this is for **COW**.*

Q4 6 mark(s) Joint & Task Space

Find the Joint-Space Gj & Task-Space Gt transfer functions.

Find Gj:	Input = Vin (V)	Output = qr (deg)	**Rotor** angle
Find Gt:	Input = Vin (V)	Output = ff (g)	**Finger** force (per finger)

- Q4.Gj (deg/V) LTI
- Q4.Gt (g/V) LTI

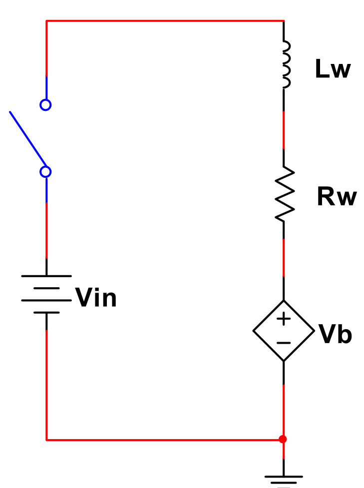
Motor & Lead-Screw

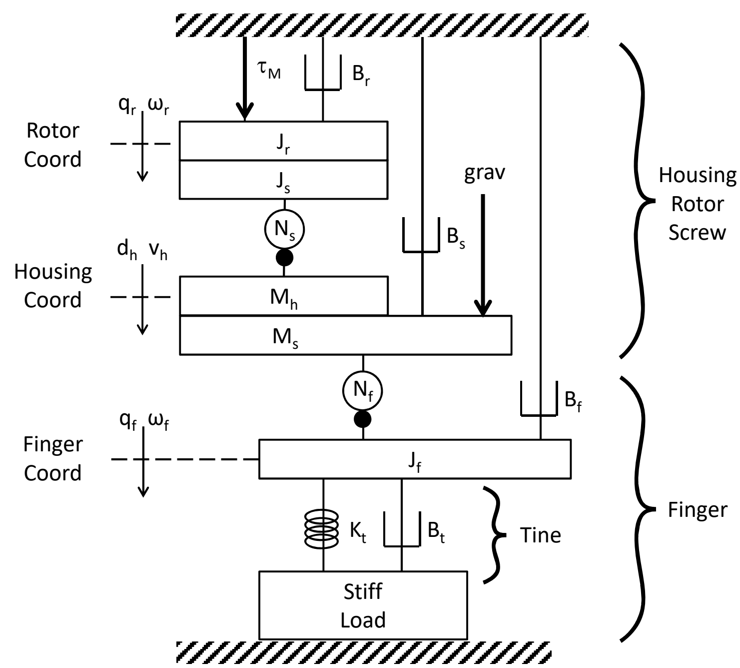
Rw =	#A / 2	(Ω)
Lw =	#B x 30	(µH)
Km =	#C	(mNm/A)
Mh =	#D + #E	(g)
Jr =	#F / 15	(g-cm^2)
Br =	#G / 30	(µNms)
Js =	#H / 5	(g-cm^2)
Ms =	#A / 4	(g)
Bs =	#B / 3	(Ns/m) ← Translational Damping Note physical units
Ns =	3	(cm/turn)

Mechanism & Controller

Jf =	#C / 3	(g-cm^2)	Each Finger
Bf =	#D / 50	(Nms)	Each Finger
Nf =	10	(deg/cm)	Each Finger
Bt =	#E	(mNms)	Each Finger
Kt =	#F x 30	(mNm)	Each Finger
L6 =	100	(mm)	
CF =	200	(Hz)	

Fig. 5: System Model





*The sensor voltage is read by a DAQ in units of (**V**) at the control frequency **CF** and is multiplied by a controller gain Kfb to convert the voltage into its equivalent angle.*

*Find the feedback sub-systems Du and Kfb, and the **loop** transfer function GH.*

Q5 2 mark(s) Open-Loop Function

Find Du: Micro-Controller DAQ Dynamics
Find Kfb: Feedback Gain applied by controller
Find GH: Input = Des Rotor Ang (deg) Output = Sensed Rotor Ang (deg)

- Q5.Du (V/V) LTI
- Q5.Kfb (deg/V) scalar
- Q5.GH (deg/deg) LTI

What is the DC gain of your entire feedback path ???

*Add a unity-gain proportional controller and compute the **closed-loop** transfer function.*

Q6 2 mark(s) Joint-Space Function

Find Xj: Input = Des Rotor Ang (deg) Output = Actual Rotor Ang (deg)

- Q6.Xj (deg/deg) LTI

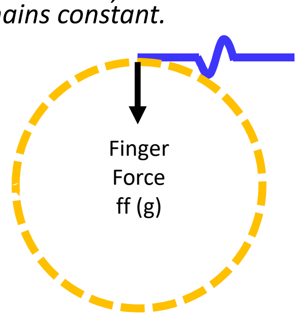
*The **task-space** function Xt shows how much force each finger applies to the load, as the system is controlled. Assume the tine doesn't bend very much so L6 remains constant.*

Continue to use the Unity-Gain Proportional Controller.

Q7 2 mark(s) Task-Space Function

Find Xt: Input = Des Rotor Ang (deg) Output = Finger Force (g)

- Q7.Xt (g/deg) LTI



A common physical unit when forces are small is grams (g).

Are the step responses of Xt and Xj similar ??? Should they be ???

During normal operation, the Finger rotates 10° to grasp an object.

In a good control system, the output tracks the input (desired output) closely. In other words, Tr, Tp, Ts and OS are all low, and DC gain ≈ 1.

- *Overshoot can damage the object*
- *Undershoot can cause the gripper to release (and drop) the object*

Are you satisfied with the performance of the Unity-Gain Proportional Controller ???

If not, you're ready for Part 2 of this project.

You don't need any transfer functions to figure out:

- *If you apply a volt to the motor, how much current will flow ???*
- *How much torque will result ???*
- *How much torque will be applied by each finger ???*
- *How much force will be applied to the object by each finger ???*

Should the DC gain of one of your transfer functions have the EXACT same value ???

You don't need any transfer functions to figure out:

- *How much must the finger move to apply that much force to the object ???*
- *How much must the motor move to make the finger move that much ???*

Should the DC gain of one of your transfer functions have the EXACT same value ???

You were handed a Simulink model on a silver platter that organizes all your sub-systems into a feedback control system. This doesn't give you the equation of the closed-loop transfer function, but it does plot the response.

It can also be used to plot intermediate values like motor voltage.

Are all the values reasonable for the application, or big enough to crush rocks ???

Was this useful for COW ??? How useful ???

Is there any reason you can't do the same for the sensor in Q3 ???