Daniel Diamont

EE 445L

Lab 10

A) Objectives

Requirements document

1. Overview

1.1. Objectives: Why are we doing this project? What is the purpose?

The objectives of this project are to design, build and test a brushed DC motor controller. The motor should spin at a constant speed and the operator can specify the desired set point. Educationally, students are learning how to interface a DC motor, how to measure speed using input capture, and how to implement a digital controller running in the background.

1.2. Process: How will the project be developed?

The project will be developed using the LaunchPad. There will be two switches that the operator will use to specify the desired speed of the motor. The system will be built on a solderless breadboard and run on the usual USB power. The system will use on-board switches. A hardware/software interface will be designed that allows software to control the DC motor. There will be at least five hardware/software modules: tachometer input, switch input, motor output, LCD output, and the motor controller. The process will be to design and test each module independently from the other modules. After each module is tested, the system will be built and tested.

1.3. Roles and Responsibilities: Who will do what? Who are the clients?

EE445L students are the engineers and the TA is the client. Daniel Diamont will build and test the sensor system, build the actuator and switch input, and design the motor controller.

1.4. Interactions with Existing Systems: How will it fit in?

The system will use the microcontroller board, a solderless breadboard, and the DC motor shown in Figure 10.1. The wiring connector for the DC motor is described in the PCB Artist file `Lab10E_Artist.sch`. It will be powered using the USB cable. **You must use a +5V power from the lab bench, but please do not power the motor with a voltage above +5V. Do not connect this bench supply to Vbus (LaunchPad +5V). However, you must have a common ground.**

1.5. Terminology: Define terms used in the document.

Integral Controller – A control system where the actuator output depends on linear combination of the current error and the integral of the error.

PWM – A technique to deliver a variable signal (voltage, power, energy) using an on/off signal with a variable percentage of time the signal is on (duty cycle)

Tachometer – a sensor that measures the revolutions per second of a rotating shaft.

Board support package – a set of software routines that abstract the I/O hardware such that the same high-level code can run on multiple computers.

Back EMF – Voltage that opposes the change in current which induced it.

Torque – Torque is the available force times distance the motor can provide at a certain speed.

Time Constant – the time to reach 63.2% of the final output after the input is instantaneously increased.

Hysteresis – A condition when the output of a system depends not only on the input, but also on the previous outputs, e.g. a transducer that follows a difference response curve when the input is increasing than when the input is decreasing.

1.6. Security: How will intellectual property be managed?

The system may include software from TivaWare and from the book. No software written for this project may be transmitted, viewed, or communicated with any other EE445L student past, present, or future (other than the lab partner of course). It is the responsibility of the team to keep its EE445L lab solutions secure.

2. Function Description

2.1. Functionality: What will the system do precisely?

If all buttons are released, then the motor should spin at a constant speed. If switch 1 is pressed and released, the desired speed should increase by 5 rps, up to a maximum of 40 rps. If switch 2 is pressed and released, the desired speed should decrease by 5 rps, down to a minimum of 0 rps.

Both the desired and actual speeds should be plotted on the color LCD as a function of time similar to Figure 10.4.

2.2. Scope: List the phases and what will be delivered in each phase.

Phase 1 is the preparation; phase 2 is the demonstration; and phase 3 is the lab report. Details can be found in the lab manual.

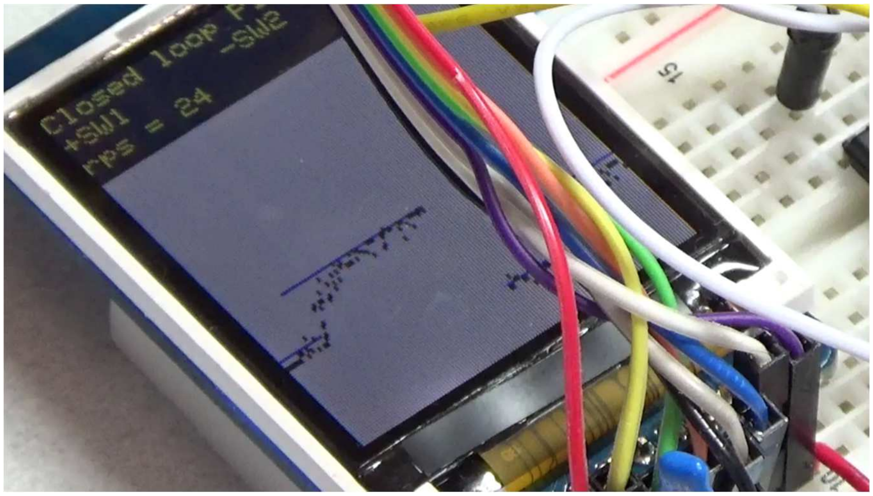


Figure 10.4. Measured and desired speeds are plotted versus time as the set point is increased from 20 to 25 rps.

2.3. Prototypes: How will intermediate progress be demonstrated?

A prototype system running on the LaunchPad and solderless breadboard will be demonstrated. Progress will be judged by the preparation, demonstration and lab report.

2.4. Performance: Define the measures and describe how they will be determined.

The system will be judged by three qualitative measures. First, the software modules must be easy to understand and well-organized. Second, the system must employ an integral controller running in the background. There should be a clear and obvious abstraction, separating the state estimator, user interface, the controller and the actuator output. Backward jumps in the ISR are not allowed. Third, all software will be judged according to style guidelines. Software must follow the style described in Section 3.3 of the book *(note to students: you may edit this sentence to define a different style format)*. There are three quantitative measures. First, the average speed error at a desired speed of 60 rps will be measured. The average error should be less than 5 rps. Second, the step response is the time it takes for the new speed to hit 60 rps after the set point is changed from 40 to 60 rps. Third, you will measure power supply current to run the system. There is no particular need to minimize controller error, step response, or system current in this system.

2.5. Usability: Describe the interfaces. Be quantitative if possible.

There will be two switch inputs. The tachometer will be used to measure motor speed. The DC motor will operate under no load conditions,

2.6. Safety: Explain any safety requirements and how they will be measured.

Figure 10.2 shows that under a no load condition, the motor current will be less than 100 mA. However, under heavy friction this current could be 5 to 10 times higher. Therefore, please run the motors unloaded. Connecting or disconnecting wires on the protoboard while power is applied will damage the microcontroller. Operating the circuit without a snubber diode will also damage the microcontroller.

3. Deliverables

3.1. Reports: How will the system be described?

A lab report described below is due by the due date listed in the syllabus. This report includes the final requirements document.

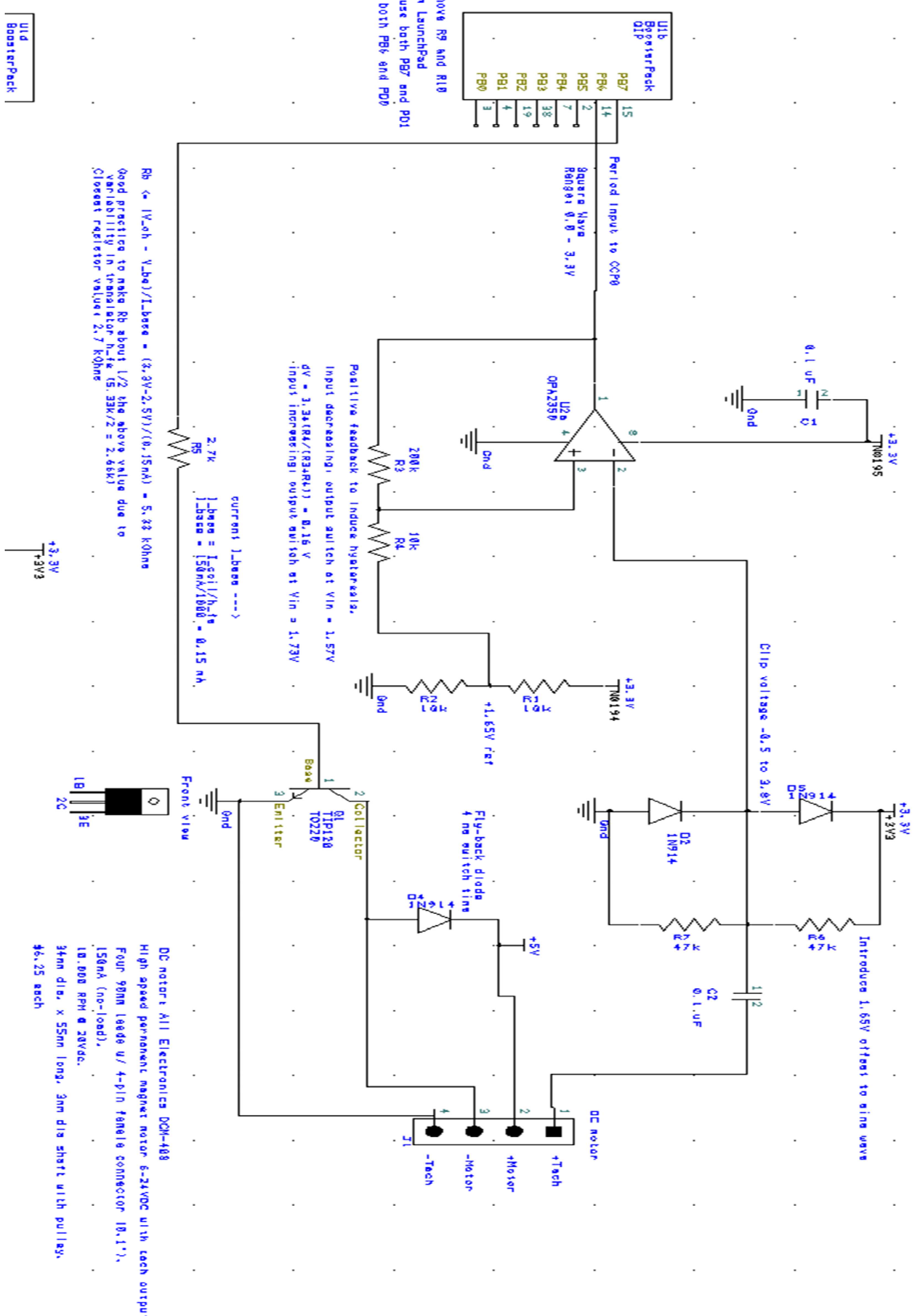
3.2. Audits: How will the clients evaluate progress?

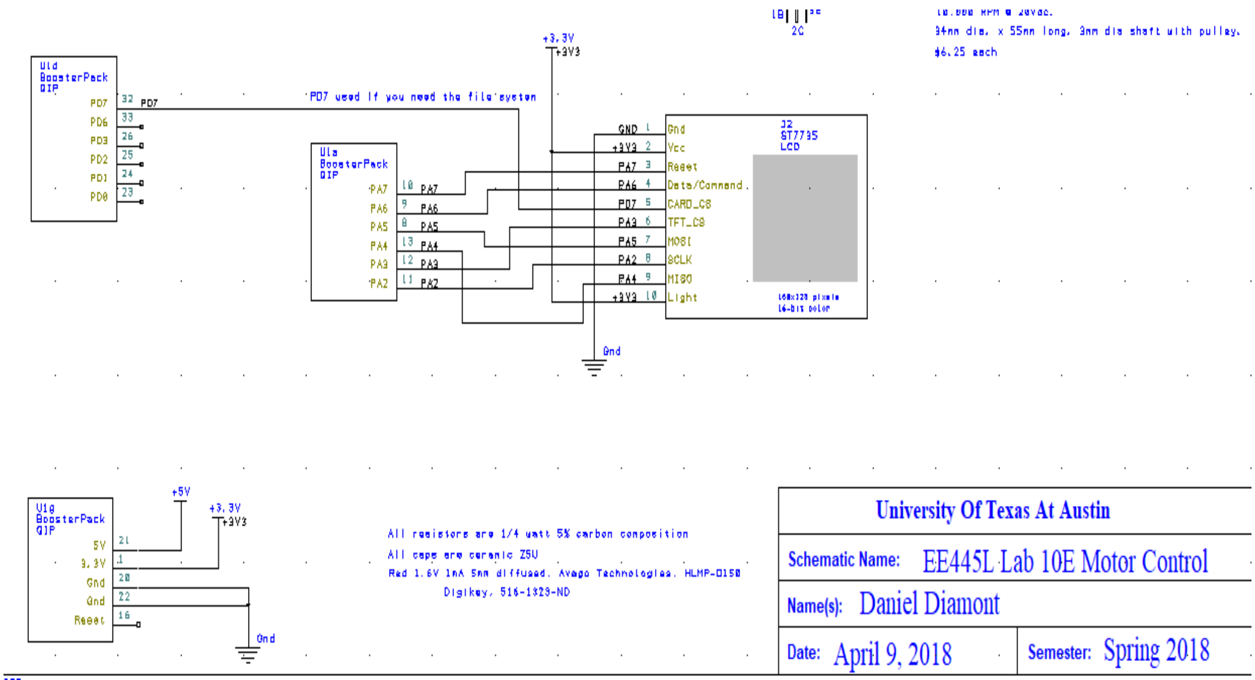
The preparation is due at the beginning of the lab period on the date listed in the syllabus.

3.3. Outcomes: What are the deliverables? How do we know when it is done?

There are three deliverables: preparation, demonstration, and report.

B) Hardware Design





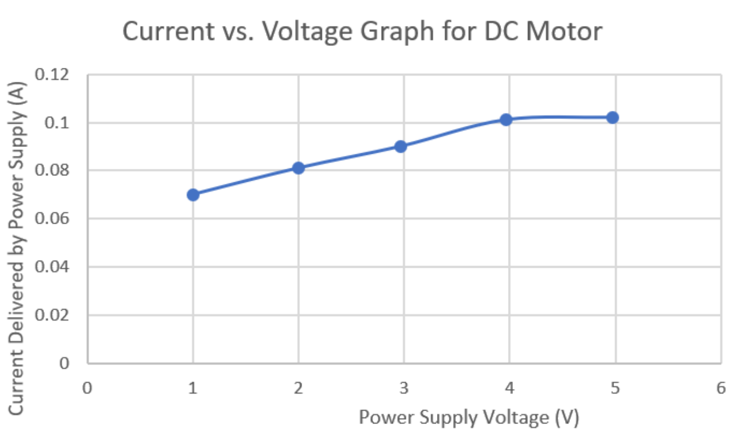
C) Software Design (see attached in .zip file)

D) Measurement Data

Procedure 1) Voltage, Current, and Resistance Measurements

Motor Resistance: 23.6 Ω

Supply Voltage (V)	Voltage (V)	Current (A)
1 V	0.998 V	0.070 A
2 V	1.998 V	0.081 A
3 V	2.969 V	0.090 A
4 V	3.961 V	0.101 A
5 V	4.968 V	0.102 A



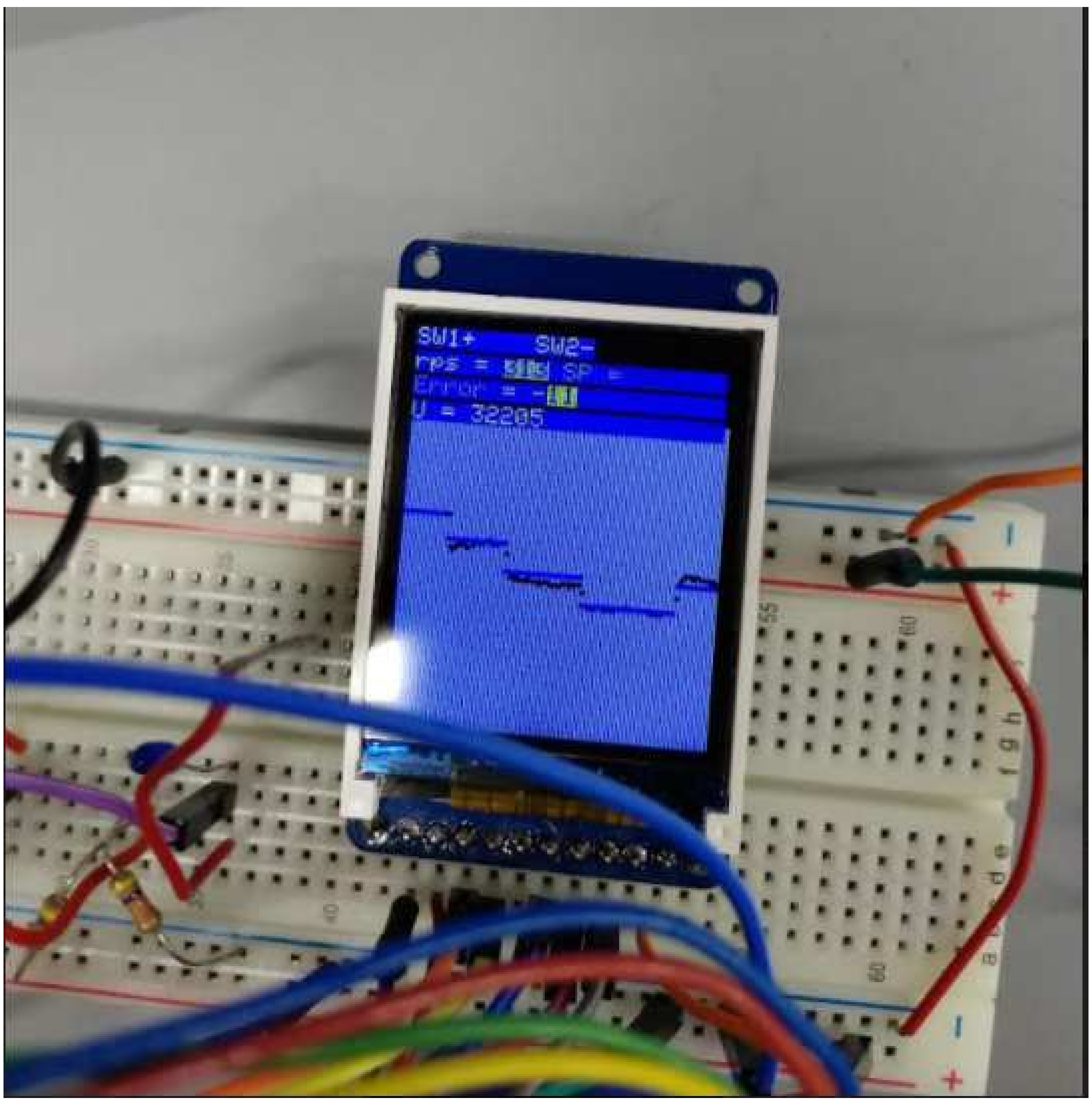
Procedure 2)

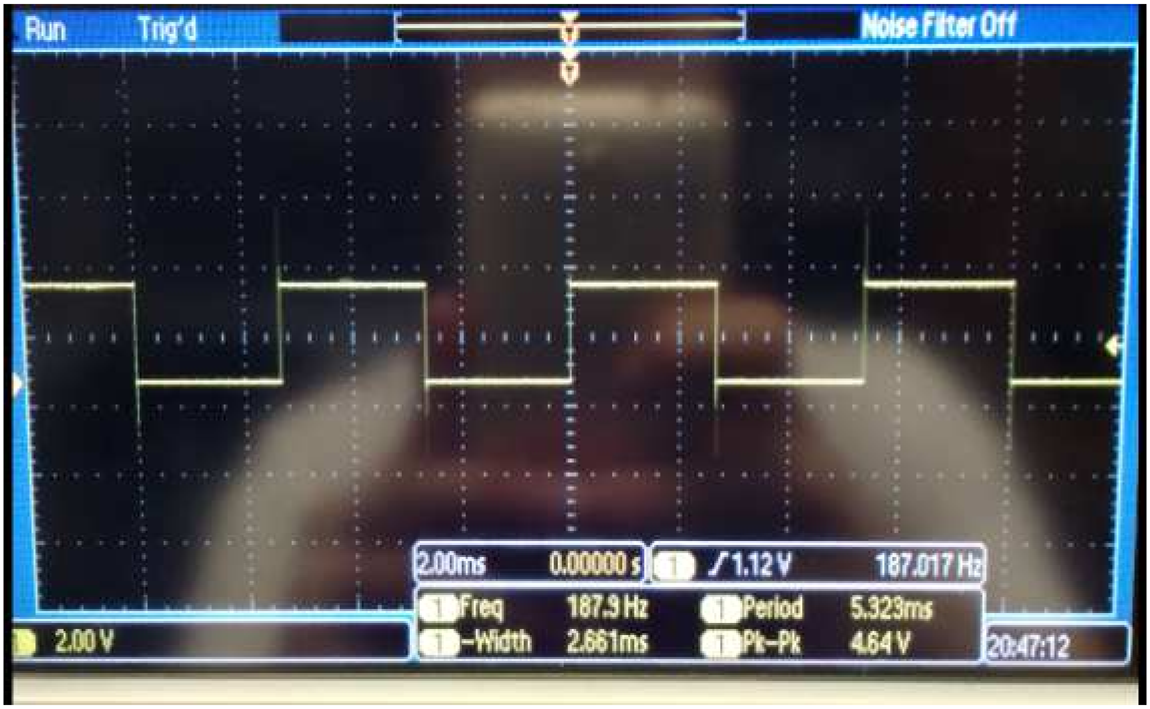
$I_{BE} = 0.253\ mA \qquad Expected\ I_{BE} = \frac{3.3V - 2.5V}{2.7k\Omega} = 0.296\ mA$

$I_{CE} = 93.2\ mA \qquad Expected\ I_{CE} = I_{BE} * h_{fe} = 0.296\ mA * 1000 = 296\ mA$

The value for I_{BE} is 14.5% off from the expected value. This can perhaps be attributed to the 5% resistor tolerance, which could allow I_{BE} to reach up to 0.312 mA. The value for I_{CE} is 68.5% off from the expected value. This may be indication that the transistor gain h_{fe} may be much less than 1000. Indeed, per my measurements, the gain would have to be ~368.379.

Procedure 3) Photos of Hardware in Operation (See attached photos folder for more)





Procedure 4) (See MeasurementData.xlsx for more details)

Maximum time to execute Input Capture ISR: 0.999 ms

Average Controller Error (Steady-State): 1.386 ms

Approximate Response Time: 168.021 ms

Procedure 5) Measurement of current required to run the system with and without the motor spinning.

$I_{running} = 0.181A$

$I_{stopped} = 0.080\ A$

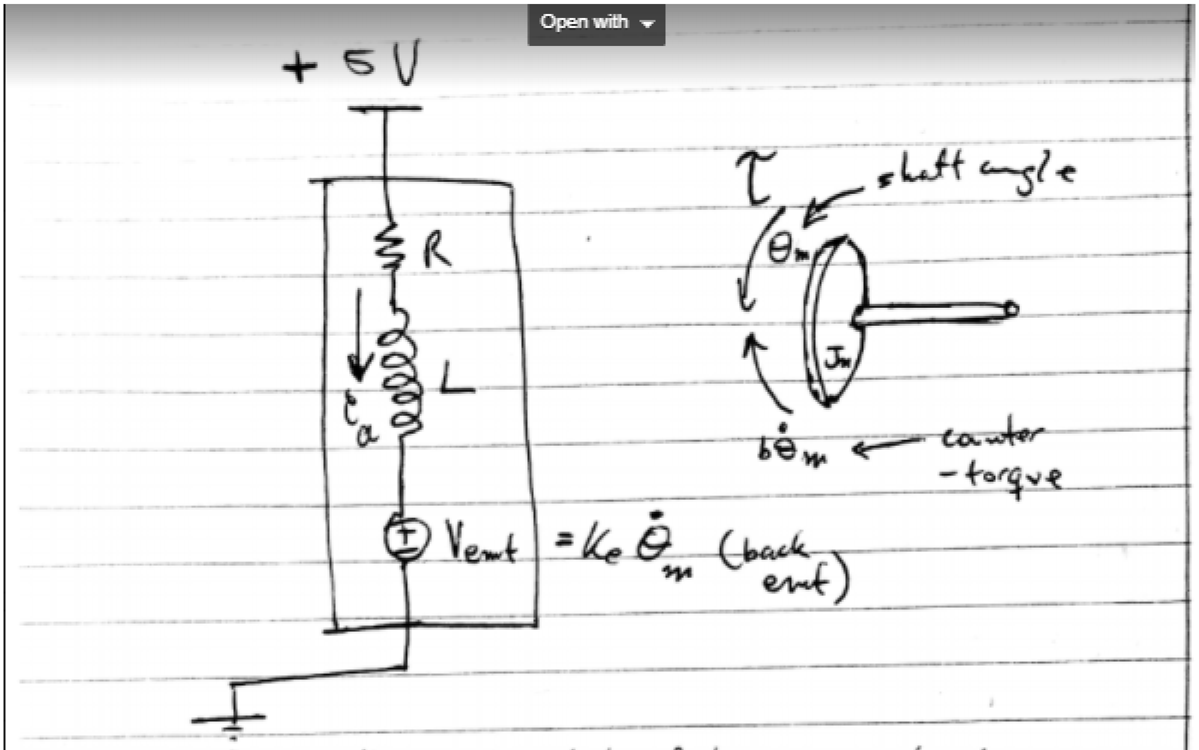
E) Analysis and Discussion

1) Torque – "a force that produces or tends to produce rotation or torsion; also : a measure of the effectiveness of such a force that consists of the product of the force and the perpendicular distance from the line of action of the force to the axis of rotation".

link to definition

defined as $T = F\ \times d = ||F|| * ||d|| * sin\theta$ with units of Newton meters (Nm)

2) Electrical Circuit Model for DC Motor



- K_e is the electrical torque constant
- $\dot{\Theta}_m$ is the rotational velocity of the shaft.
- $b\dot{\theta}_m$ is the counter-torque applied by friction to the shaft, where b is the coefficient of friction.

If we increase the coefficient of friction or the the load on the shaft, then the rotational velocity of the shaft decreases. Thus, the back-emf decreases causing the voltage difference ~~to the~~ across the coil to be greater, so the armature current i_a increases.

3) Parameters for choosing motor drive interface chip

We know from Dr. Valvano's experimental data that the DC motor under no load will draw about 150 mA. We want to drive our transistor into saturation without having to drive more than 8 mA from the TM4C output. The TIP120 will saturate at 2.5V and has a current gain factor of 1000. This means that we can choose a 2.7kΩ base resistor to drive 0.296 mA into the transistor base to drive it into saturation. The current through the coil will then be less than or equal to 296 mA, which is more than enough to supply the DC motor under no load.

4) Another choice for a controller

I could have implemented a PI controller. The proportional term would have provided a proportional but opposite controller reaction to the speed error. This would have increased the rise time of the step response of the system, thus decreasing the response time.

Another controller I could have implemented is a PID controller with a low-pass filter. This would have given me an error correction term proportional to the time derivative of the system's error which would decrease maximum overshoot and decrease the settling time of the system step response.

Lastly, I could have implemented a lag compensator, which would have similar action to a PI controller, except the PI controller is guaranteed to provide zero steady state error for the system step response at low frequencies, whereas a lag compensator cannot provide the same guarantee.

5)

Electrical Power – The product of the armature current multiplied by the power supply voltage. This is the power applied externally to the motor. $P_{IN} = I_a * V_{dc}$

Mechanical Power – Rotational mechanical power is defined as the cross product of the load torque on the motor and the angular velocity of the shaft. $P_{OUT} = T_{load} \times \omega$

In DC motors, electrical power is converted into mechanical power with some frictional losses.

$P_{IN} = P_{OUT} + P_{losses}$

Where $P_{losses} \approx (I_a)^2 * R_{armature} + P_{friction}$