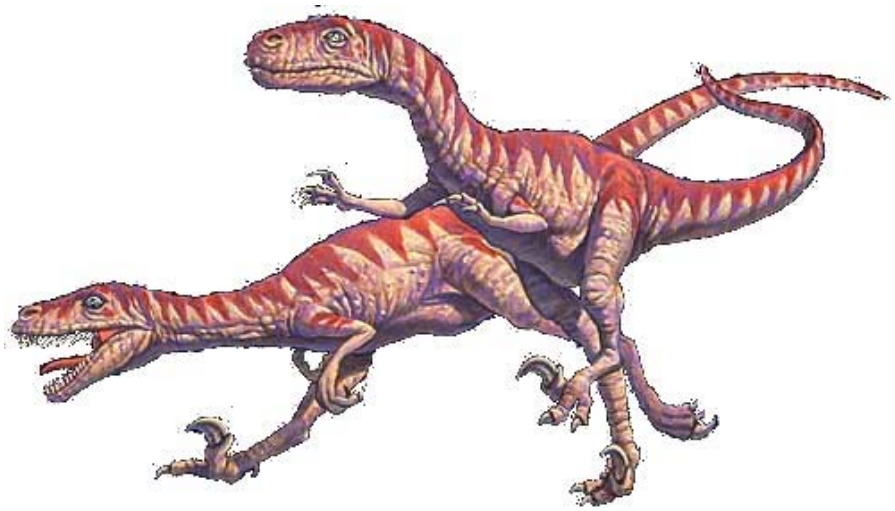
Deinonychus Desktop Dynamometer

ENGI 9548 – Mechatronics Design Project



Designed By:

Greg Browne
Dennis Fifield
Taufiq Rahman

Table of Contents

Introduction and Overview

The purpose of any dynamometer (dyno) is to measure the output characteristics of any rotating machine – power and torque (Power Dynamometers). Furthermore, a well designed dyno will provide further information, such as a graphical representation of the outputs, efficiency, and characteristic equations.

Motor Characteristics

Torque may typically be graphed against rotational speed, measured in Rotations Per Minute (RPM), rather than the SI metric, rad/s. One can expect a linear decrease in torque from its stall value (max torque), τ_s, as rotational speed increases to the system's no load speed (max speed), ω_n,(Center for Innovation in Product Development, 1999). Power, also typically graphed against RPM, can be expected to have a parabolic form centred about the midway speed between stall torque and no load speed. A representation of the relationship of torque, speed, and power can be seen in Figure 1 (Center for Innovation in Product Development, 1999).

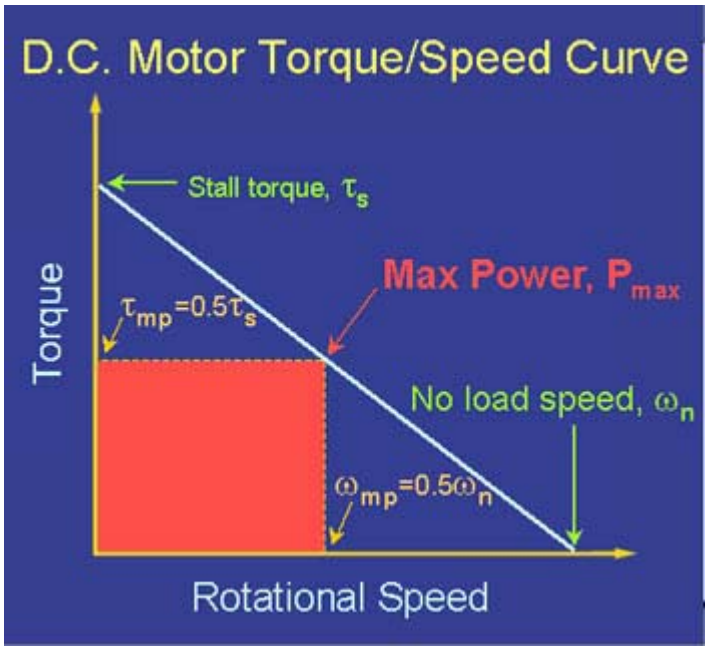


Figure 1 D.C. Motor Torque/Speed Curve

For a DC motor, the input power, P_I, (for a single phase) is related to the input current, I (A), and input voltage, V (V), as given by Eq (1).

$$P_I = VI \tag{1}$$

The output power, P_O is related to the torque, τ (N·m), and rotational speed, ω (rad/s), as given by Eq (2).

$$P_O = \tau\omega \tag{2}$$

Therefore, the percentage efficiency, η, of the motor can be illustrated as a ratio of its output power to input power, as shown in Eq (3).

$$\eta = 100\% \times \frac{P_O}{P_I} = 100\% \times \frac{\tau\omega}{VI} \tag{3}$$

Brushless DC Motors

A Brushless DC Motor (BLDC) is a synchronous AC motor powered by DC energy and commutated electronically instead of via conventional mechanical means, i.e. brushes. It takes in electrical energy and converts it to rotational energy. Like typical dynamic systems, BLDC characterization means understanding the relationship between input and output and quantifying the input and output and the desired performance. Motor characterization is particularly of help when it is needed to maximize the output and design around the motor. Input to the BLDC is electrical energy which is characterized by voltage, V and current, I. Output from the BLDC is rotational energy which is characterized by torque, τ and rotational speed, ω. A block diagram of a BLDC is illustrated in Figure 2.

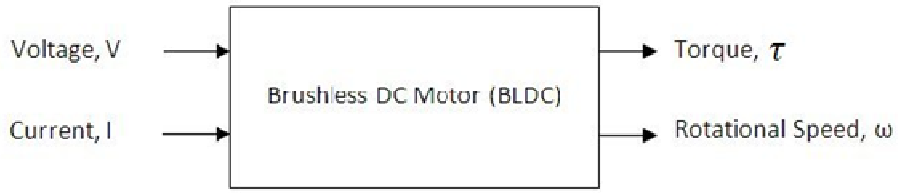


Figure 2 Block Diagram of a Brushless DC Motor

The Deinonychus Approach

The *Deinonychus Desktop Dynamometer* is named after the Deinonychus, which lived 98.9 to 121 million years ago (Wikipedia, 2008) and was considered to be a "small dino". Measuring 20.3 cm × 20.8 cm × 18.1 cm, the Deinonychus dyno described in this document would also be considered small compared to most commercially available dynos.

A DC brushless motor can be mounted to the dyno for characterization. An eddy current brake (ECB) is utilized to break the rotation of the motor while characteristics such as RPM, input voltage, input current, and output torque are monitored. The data acquisition and algorithm are implemented on a Phytec phyCORE®-MPC555 single-board computer and the processed data is transmitted via RS-232 to a GUI hosted on a computer for display and interpretation.

The system can be broken down into the following main modules:

1. Eddy Current Brake
2. Mechanical Rig
3. Interface Circuitry
4. Model (Firmware)
5. Graphical User Interface (GUI)

These modules are discussed in the following sections.

Eddy Current Brake Description

The Eddy Current Brake (ECB) is the main mechanism by which the primary output (torque) is exercised for measurement.

The principles behind the ECB are based on Faraday's Law (of Induction) and the Biot-Savart Law. Faraday's Law states that an emf (electromotive force, i.e. voltage) is produced across a conductor which is subject to a changing magnetic flux (Tankersley & Mosca, 2005). This voltage would cause the flow of eddy currents about the surface of the conductor. The Biot-Savart Law states that flowing electrons (current) will generate a magnetic field (Spencer, 1997).

An aluminum disc is attached to the DC motor to be characterized such that its rotation is directly coupled with that of the shaft of the motor. This disc is separated by a gap distance, *d*, from a magdisk – a disk that contains permanent magnets arranged along its circumference at a radius, *R*, from its centre with alternating poles.

As the motor rotates, the eddy currents, *I*, are induced onto the aluminum disc by its rotation through the magnetic field, *B*, of the magdisk (Faraday's Law) as given by Eq (4). These eddy currents, in turn, generate an opposing magnetic field to resist the magnetic field of the magdisk.

$$I = \frac{4\pi B r^3}{2\pi \mu_0 R^2}$$, where *r* is the distance from the magnet centres to the aluminum disc centre, given by Eq (5) and μ_0 is the permeability of free space ($4\pi \times 10^{-7}$ T·m/A). (4)

$$r = \sqrt{d^2 + R^2} \quad (5)$$

The torque, τ, produced by this opposing (θ=90°) magnetic field is the "dipole moment", which is the energy required to align the magnetic poles, given by Eq (6).

$$\vec{\tau} = \vec{\mu} \times \vec{B} \xrightarrow{yields} \tau = \mu B \sin\theta = \mu B \quad (6)$$

The magnetic moment, μ, is a vector quantity perpendicular to the loop and is given by Eq (7) (Nave, 2004).

$$\mu = IA$$, where *A* is the area of the loop through the centres of the magnets. (7)

By substituting (7), (4), and (5) into (6), one can see the torque provided by the ECB as a function of the magnets chosen (constant *B*), their position (constant *R*), and the varying gap distance, *d*, used to brake the motor, as shown in Eq (8).

$$\tau = IAB = \left(\frac{4\pi B r^3}{2\pi\mu_0 R^2}\right)(\pi R^2)B = 5 \times 10^6 B^2 (d^2 + R^2)^{3/2} \quad (8)$$

However, simply observing (8), one would think that torque would decrease as the gap distance decreases. Not so – once the eddy currents are flowing across the surface of the aluminum disk, revisiting the Biot-Savart Law shows that, as the gap distance decreases, the magnetic field strength increases. The opposing magnetic fields cause the kinetic energy of the rotating disc to start converting to heat. This loss of energy causes the disc rotation to slow, which in turn causes the torque to increase.

This is the braking process – the rotational speed decreases to a minimum, at which point the torque is maximal. Torque values throughout the braking process produce a family of curves that characterize the motor.

Mechanical Rig Design Description

This section deals with the mechanical design of the eddy current brake. Please refer to the included CD for the SolidWorks models of the various components used in the design and assembly of the rig. The basic idea of this design is to mount the brushless DC motor on a linearly moving platform, driven by a stepper motor lead screw drive. To eliminate the friction of this drive a set of linear bearings with aluminum rails were used. On the other end of the rails, the brake was mounted on a stationary platform. By moving the brushless DC motor platform towards the magnetic brake provided braking on the motor. Construction of the mechanical system of the brake can be divided into five major parts, discussed in the following sections.

Base frame

The base frame of the brake is designed using Aluminum Fractional T-Slotted Framing System extrusions, which are made of 6105-T5 Aluminum. This particular material was chosen for this purpose for its strength, while being light in weight. One such extrusion bar is shown in Figure 3. This extrusion is the building block of the braking system.

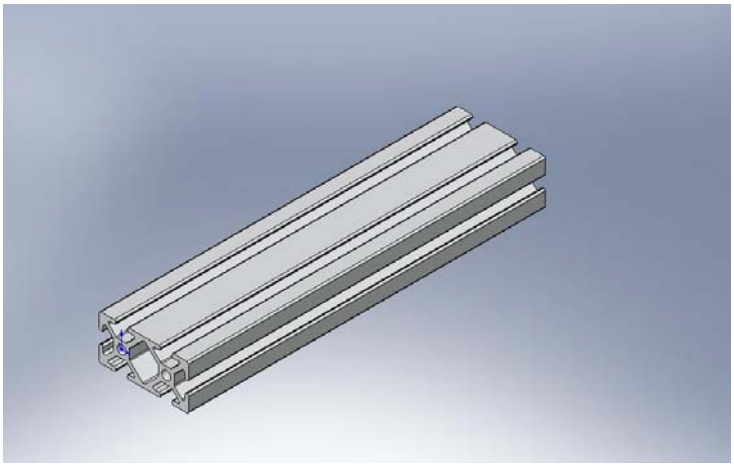


Figure 3 Aluminum Fractional T-Slotted Framing System Extrusion

To provide a guide way for the stepper motor lead screw drive of the motor platform, a pair of aluminum rails were used, as shown in Figure 4.

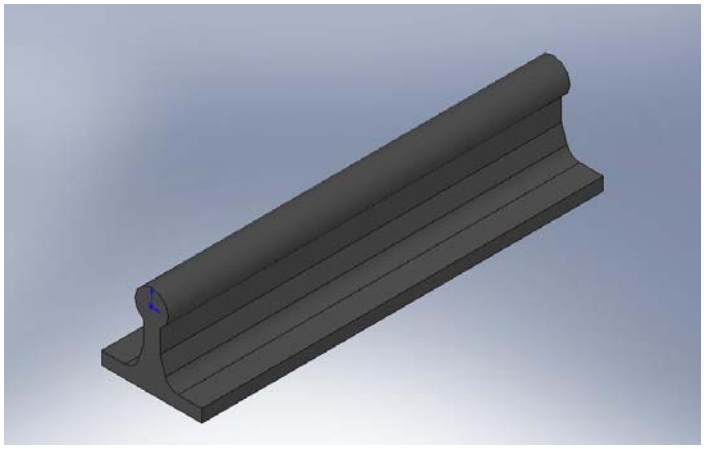


Figure 4 Rail to Provide Guide Way for the Motor Platform Movement

Assembly of the base frame is given in Figure 5 in exploded view. A pair of aluminum extrusions (2 inches in height) was used as mechanical stops at one end of the rails.

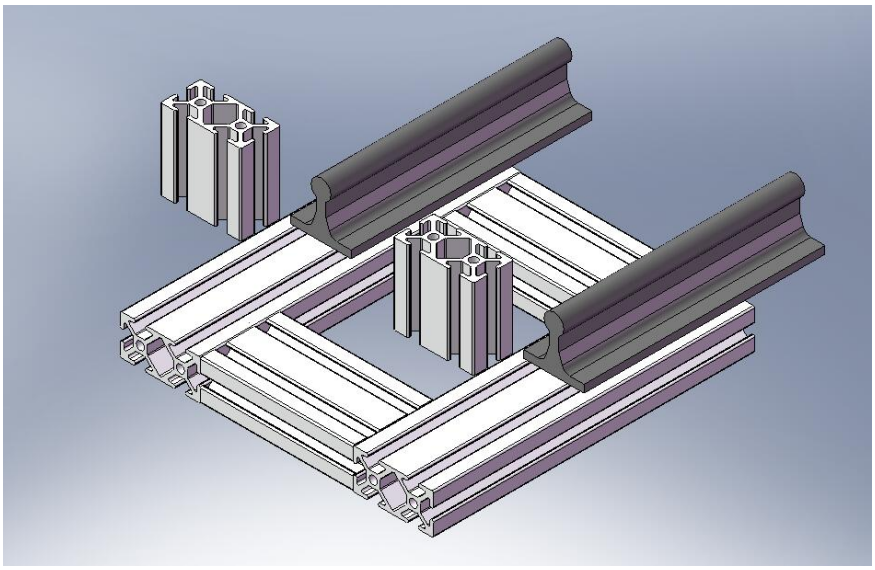


Figure 5 Exploded View of Base Frame Assembly

Motor Platform

The motor to be characterized is mounted on an 8-inch long aluminum extrusion. At two ends of the extrusion, the linear bearings are mounted. On the platform, the mount for the motor is bolted to the T-Slot of the extrusion. Under the platform there is an attachment, called the underhanger, which attaches the stepper motor lead screw to the platform. A SolidWorks rendering of the motor platform is given in Figure 6.

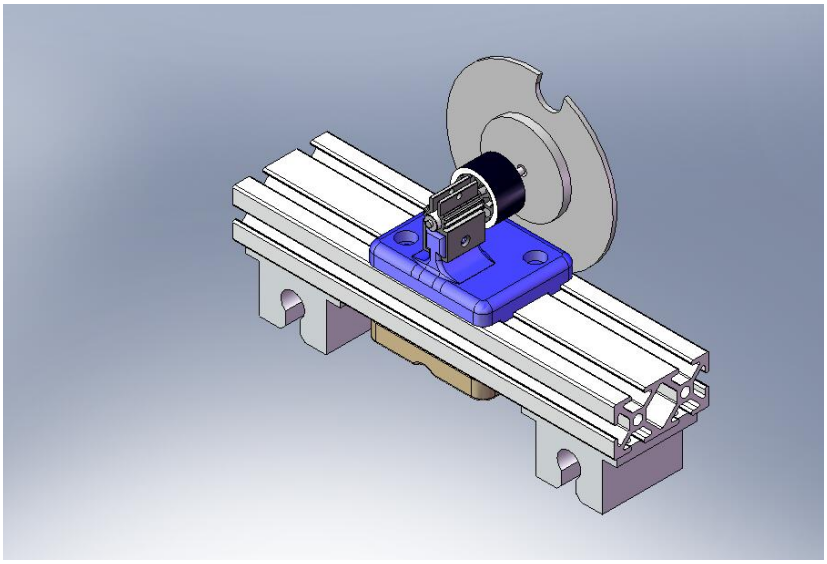


Figure 6 Motor Platform Assembly

An exploded view of the motor mount and the encoder disk is given in Figure 7.

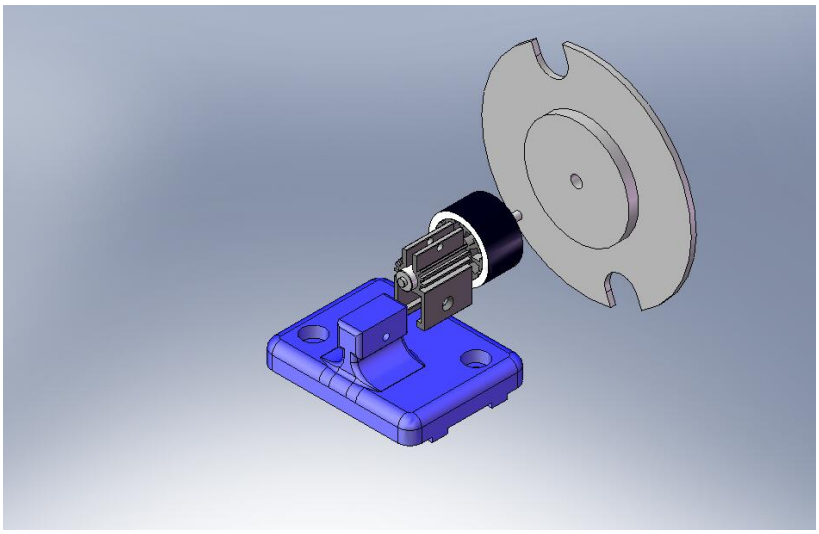


Figure 7 Exploded View of the Motor Mount

An exploded view of the underhanger assembly is given in Figure 8. A 2-56 nut is used between the lead screw and the hex standoff, which is counter-tightened against the standoff to strengthen the junction.

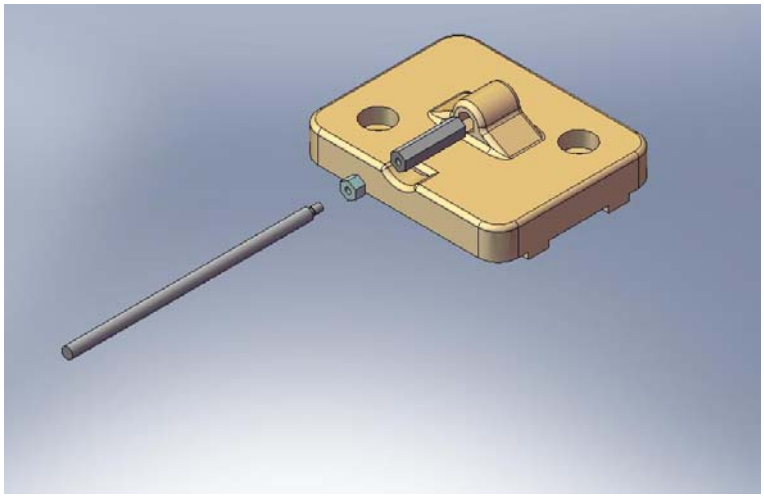


Figure 8 Exploded View of the Underhanger Assembly

Stepper Motor Lead Screw Drive

A stepper motor with a spring-loaded lead screw is used for driving the motor platform. An exploded view of the assembly is presented in Figure 9.

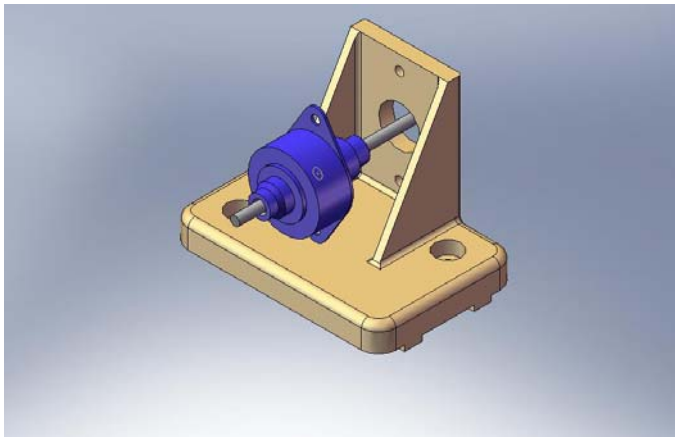


Figure 9 Stepper Motor Assembly

Brake platform

The (stationary) brake platform houses the eddy current brake. It is supported on the rails by two rapid-prototyped linear bearing replicas. The material of these replicas is ABS and they are snugly fitted on the rails. This snug fit allows enough traction to hold the brake base under operating condition.

Moreover, they can slide on the rails to control the distance between the motor platform and the brake platform. This gives additional control over the distance besides the stepper motor drive.

Replicated linear bearings have extended shafts on one end which serve as mechanical stops (besides the mechanical stops on the base frame). These mechanical stops ensure safe operation of the rig. A SolidWorks rendering of the brake platform is given in Figure 10.

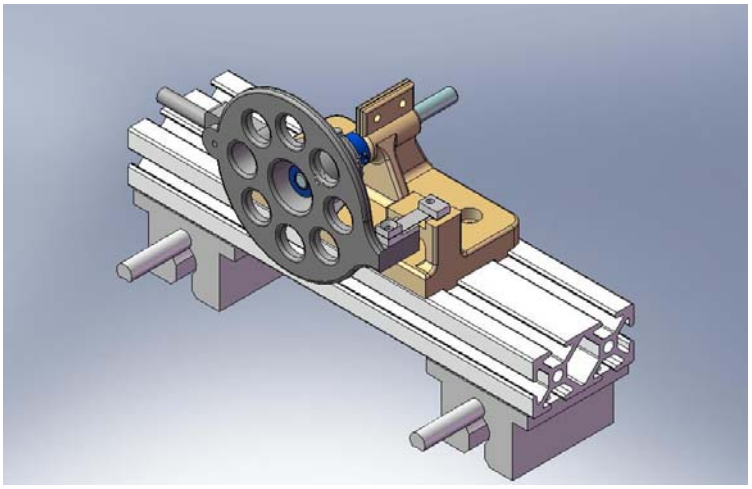


Figure 10 Brake Platform Assembly

Among all the different components used to construct the rig, design of the brake is comparatively crucial and complicated and therefore warrants for special care and attention. Because the brake is attached to the load cell to measure the torque, there are some requirements to be met in order to measure the torque properly. These requirements are:

1. The brake houses the magnets. As the brake has to withstand the torque produced by the motor to be characterized, and at the same time the torque is measured, it should be designed to take the radial loading frictionless.
2. There must not be any axial movement of the brake for safety reasons.
3. The weight distribution of the brake should be such that it provides preloading to the load cells inherently. Moreover, the amount of preloading should also be controllable to within a reasonable tolerance in order to enable the rig to characterize a variety of motors.
4. It should be also considered that the load cell needs to be calibrated before operation. Therefore the brake should also facilitate this task.

To meet these requirements, the brake is designed to be a disk with through holes. These through holes match the diameter of the magnets to be used in the brake. The brake also has a hub to house the bearings. A precisely machined shaft is used to support the brake. A horizontal shaft is attached to the brake to provide preloading on the load cell. However, using some metal rings and a suitable shaft collar

on the shaft enables us to control the preloading. Two shaft collars are used to mount the brake on the shaft and they are counter-pressed against each other to eliminate any lateral movement of the brake. A retaining ring is used between each set of bearing and shaft collar to prevent rubbing of the bearing rings on the shaft collars. It is also noteworthy that if the shaft collars are counter-pressed too much, it can provide excessive axial loading to hamper the bearings' frictionless operation. An exploded view of the brake is presented in Figure 11.

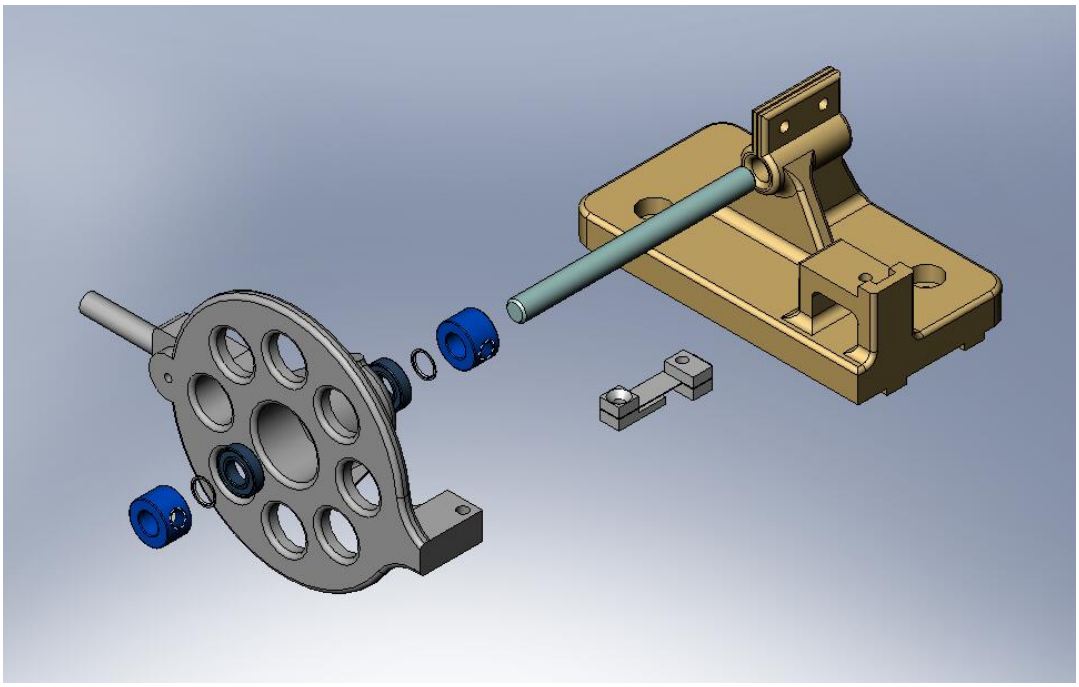


Figure 11 Exploded View of the Eddy Current Brake

Mounted Peripherals

There are three peripherals mounted on the base frame – two limit switches and one optical encoder. The function of the limit switches is to limit the drive length of the motor platform. The optical encoder is mounted on the side of the motor platform under the encoder disk. The limit switch on the brake side of the rail is adjustable. It should be noted that, it is of prime importance to adjust this limit switch in a way, such that the rotating encoder disk does not hit the brake.

Overall Assembly

A SolidWorks rendered view of the overall assembly of the rig is shown in Figure 12.

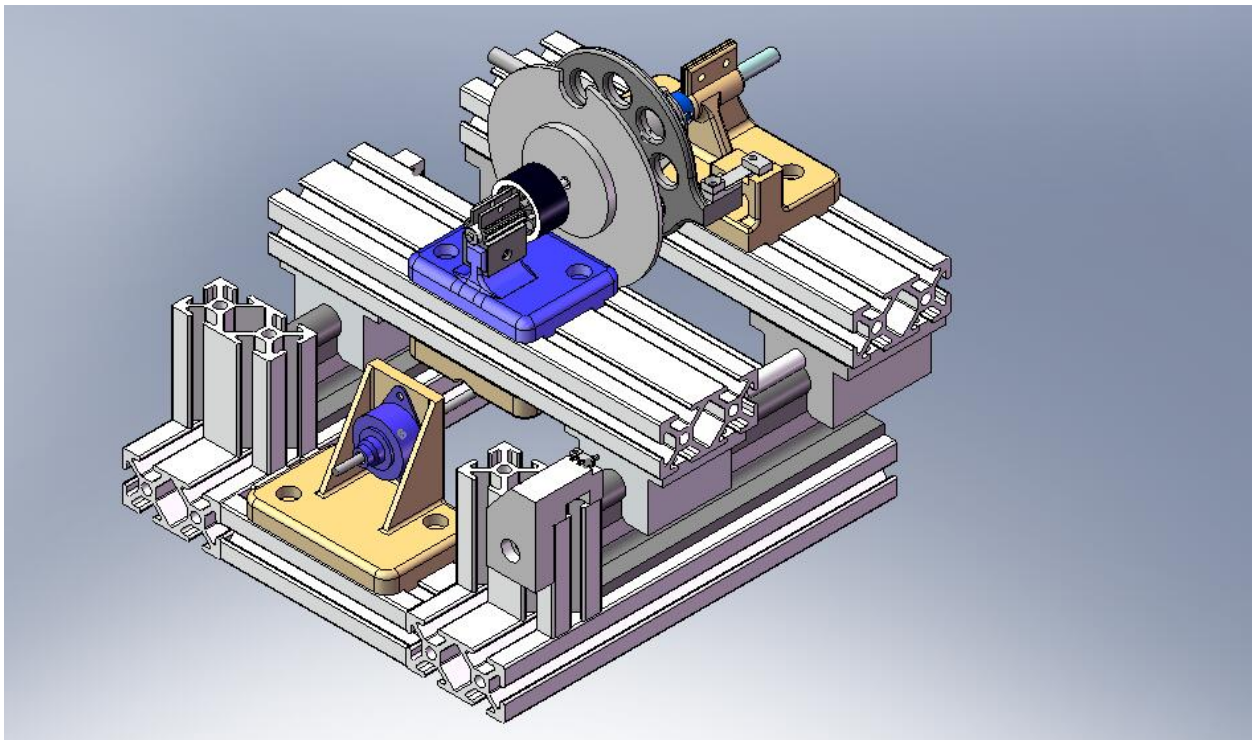


Figure 12 Overall Assembly of the Rig

In Figure 13, a side view is shown of the overall assembly features the mounting of the limit switches.

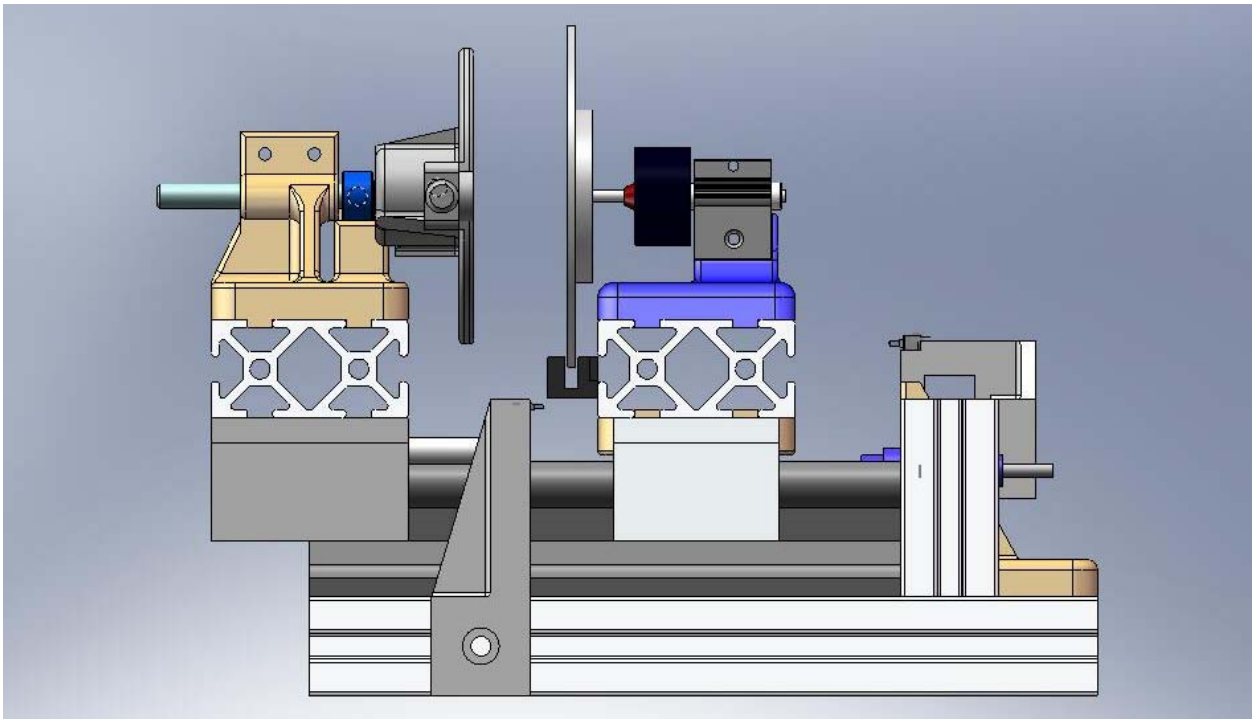


Figure 13 Featuring the Position and Mountings of the Limit Switch

Recommendations

Mechanically, the rig meets the design requirements in terms of strength, load withstanding, induced vibration, flexibility and safety. However, a thorough review of the design calls for some minor adjustments, which have the potential to enhance the mechanical performance of the rig.

- The rig uses a pair of 6 inch long rails which provide a guide way for the moving motor platform. To measure the unloaded RPM of the motor, one would want the ECB to be away from the encoder disk, such that it does not influence the RPM. Experimentally, it was found that when the motor platform is fully retracted towards the stepper motor, there is no influence of the magnets on the motor RPM. But, it would be desirable to have 9 inch long rails, so that the rig can provide enough spacing between the motor platform and the brake platform to measure the unloaded RPM without the influence of the magnets.

- Design requirements demand for a light, yet rigid, encoder disk. It must be dynamically balanced to reduce vibration. But at higher RPMs, the disk flexes slightly. The slot of the used optical encoder is just wide enough to let the encoder rotate freely. Making the disk thinner would worsen the problem as it would flex more. A solution to this problem is to cut a step near the circumference of the disk, so that the part of the disk which passes through the slot is thinner than the rest of the disk. It would also allow us to have a bigger disk thickness near the center to minimize flexing. CNC machining is recommended to manufacture the disk to make it more balanced dynamically.

- The test motors came with a mount having a rectangular inside slot. The slot was mounted on a matching rectangular base which was rapid prototyped. Rapid prototyping could not give us enough accuracy to ensure a snug fit between the motor mount and the base. To eliminate the loose fit, the mount was glued onto the base. However, an alternative mount can be sourced or manufactured which will give rigid mounting of the motor onto the base.

Interface Circuitry Description

The following sections describe the main electronic modules used for signal conditioning and interfacing to the Phytec phyCORE®-MPC555 single-board computer.

Supply Voltages

The phyCORE interface contains five distinct supply voltages, shown in Figure 14, that serve various functions related to power and referencing.

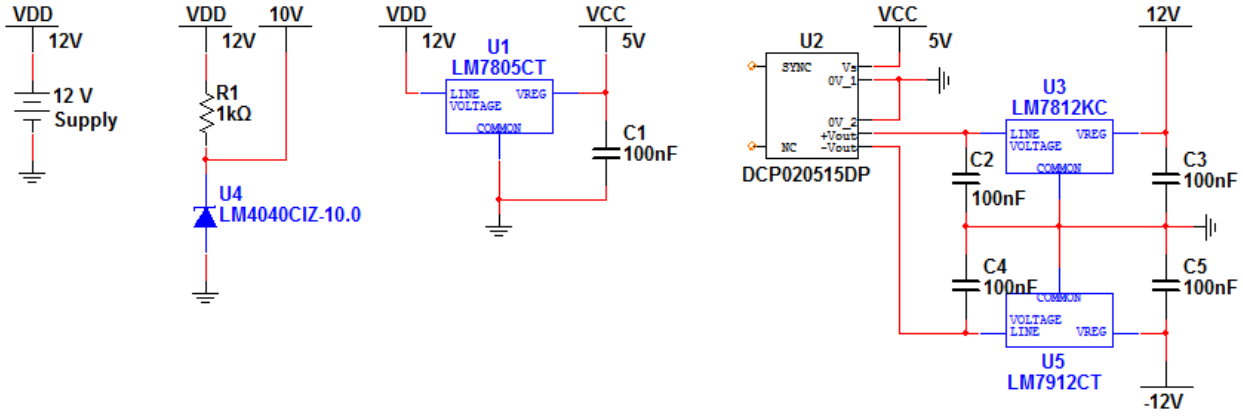


Figure 14 Interface Supply Voltages

V_{DD} is the (approximate) 12V supply provided by the power supply (or 12V battery) that is used to power the motor controller and stepper motor, as well as provide a source of regulation for the other supplies.

The 10V supply is provided by U_4, a zener voltage reference – this provides a precision excitation supply for the load cell to ensure the stability of any measurements.

V_{CC} is the 5V supply provided by U_1, a voltage regulator, which is used as the main source for most of the circuitry of the interface. This regulator was used instead of the onboard 5V supply as a precaution, due to current sourcing limitations of the phyCORE.

±12V supplies are regulated (via U_3 and U_5 voltage regulators) from U_2, an unregulated switching DC/DC converter that converters a 5V input into ±15V outputs. The ±12V supplies power to some of the ICs used for current and voltage measurements in the circuitry.

Motor Controller

The motor controller, U_6, is a Thunderbird-9 (shown in Figure 15), typically used for RC applications.

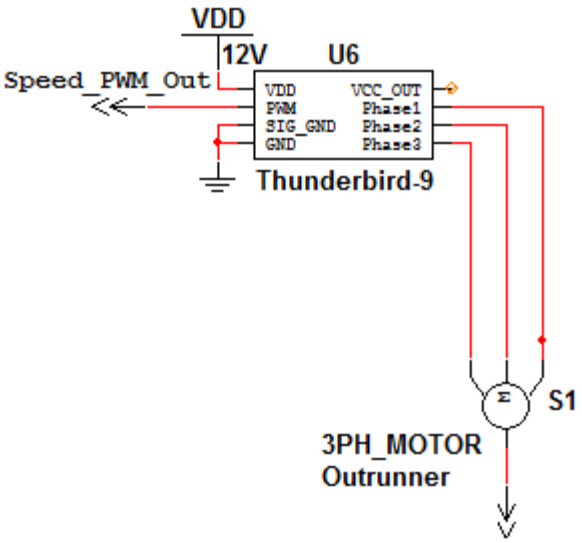


Figure 15 Thunderbird-9 Motor Controller

The Thunderbird-9 accepts a PWM signal (*Speed_PWM_Out*, in this case from the phyCORE) with a period of 20ms, of which 1ms of on-time represents a duty cycle of 0 and 2ms of on-time represents a duty cycle of 1.

Phases 1-3 are switched by the motor controller and are connected to the 3 phases of the outrunner DC, brushless motor, S_1, to be characterized.

RPM Feedback

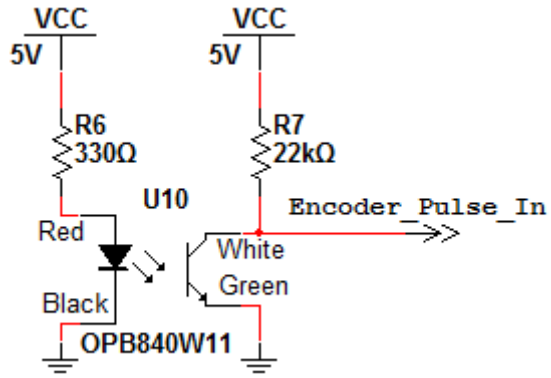


Figure 16 RPM Feedback

An optical switch, U_{10}, shown in Figure 16, is mounted such that the encoder disk (spun by the rotation of the outrunner motor) passes through its slot. The encoder disk contains two notches which cause the optical switch to have an output that is idle-high with a duty proportional to the speed of rotation (RPM).

This relationship is defined by Eq (9).

$$RPM = \frac{\frac{1}{2}Rotation}{T(s)} \times \frac{60s}{1min} = \frac{30}{T(s)} \quad \text{, where T(s) is the period (in seconds) output from the optical switch.} \tag{9}$$

T(s) is measured using the MIOS waveform period measurement module of the phyCORE. This is the actual RPM achieved based on the desired RPM given to the motor controller (described previously).

Voltage Measurement

One phase from the motor controller is arbitrarily chosen (in this case, Phase 1) for measurement. The RMS voltage is measured, by means of an RMS-to-DC converter, U_{11}, as shown in Figure 17. The DC output equivalent of the RMS input, *RMS_Voltage_In*, is read by the phyCORE QADC for processing.

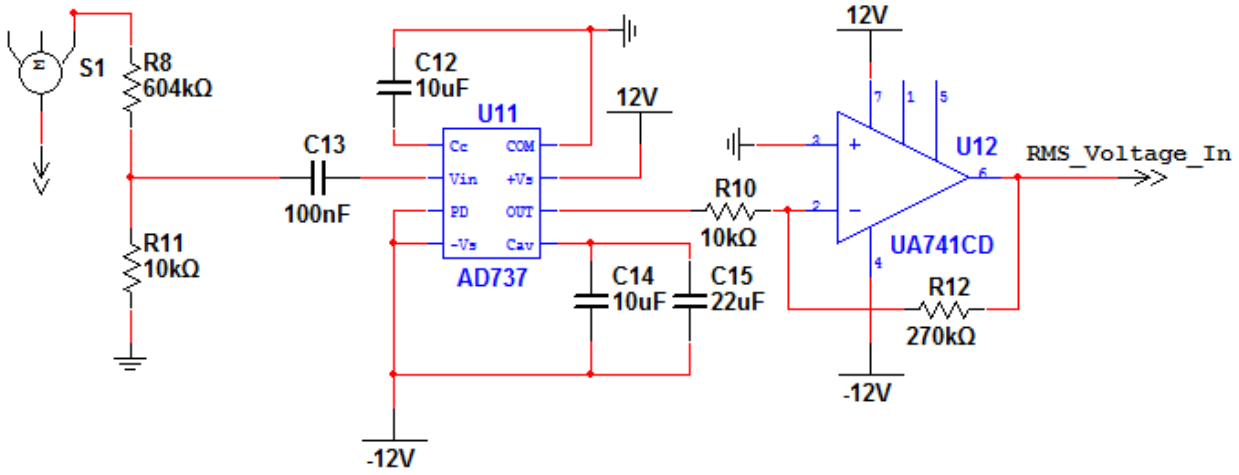


Figure 17 RMS Voltage Measurement Module

Due to the fact that the RMS-to-DC converter can only accept an input as high as 200mV, the phase voltage is divided by 60 via R_8 and R_{11} and decoupled by C_{13}. Finally, the opamp U_{12} introduces a gain of 27 to make the signal useable by the phyCORE QADC.

Current Measurement

The same phase that is measured for voltage is also measured for current. The phase output from the motor controller is passed through the hole in the Hall Effect sensor, U_7, before supplying the motor phase, as shown in Figure 18.

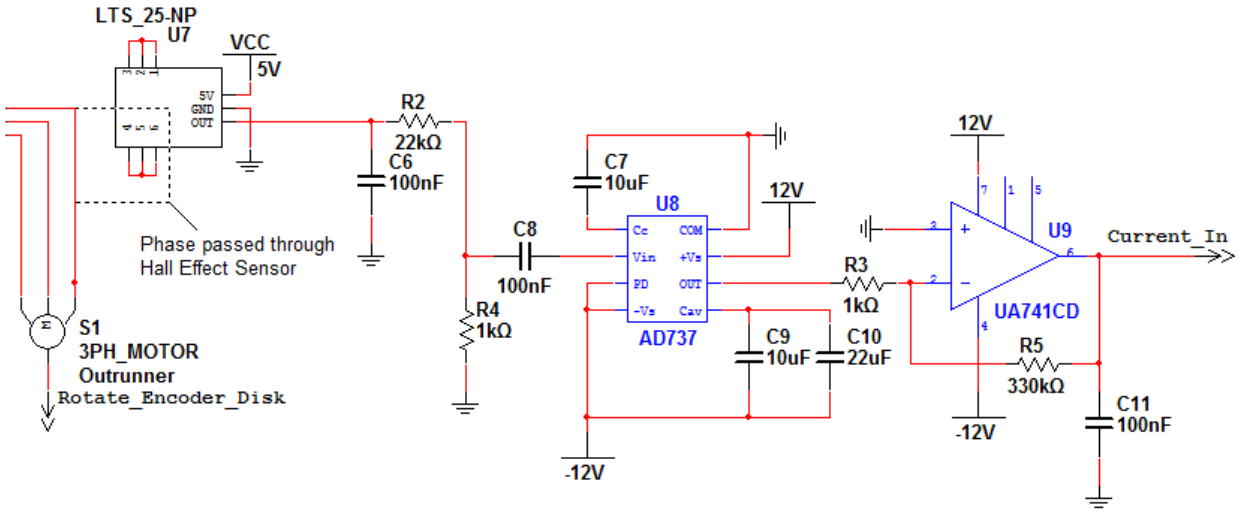


Figure 18 Current Measurement Module

The output of the Hall Effect sensor is a noisy AC signal (with a DC offset). Capacitors C_6 and C_{11} were placed to reduce the spikes in this signal. Furthermore, to convert the AC signal into DC, a circuit similar to the voltage-measurement circuit is used (with a gain of 330 instead of 27). This value (*Current_In*) is read by the phyCORE QADC for processing.

Eddy Current Brake Control

The Eddy Current Brake position is controlled by a lead-screw stepper motor, S_2, shown in Figure 19.

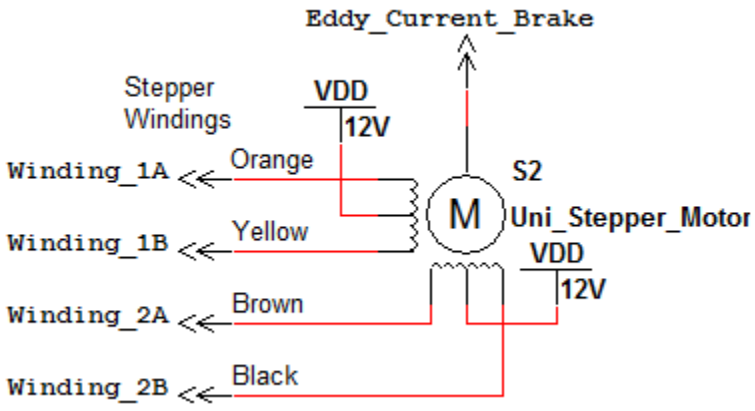


Figure 19 Stepper Motor

The stepper motor moves the Eddy Current Brake forward and reverse (with respect to the magdisk) based on the switching pattern provided from the digital outputs of the phyCORE.

The minimum and maximum positions allowed for the Eddy Current Brake are enforced by limit switches J_2 and J_3, as shown in Figure 20 (matching configuration used for the front limit switch).

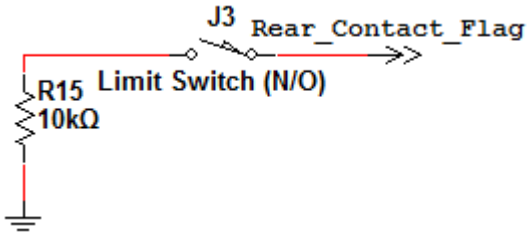


Figure 20 Limit Switch

The limit switches pull down the idle-high digital inputs when triggered. The front limit switch is triggered when the Eddy Current Brake has reached its closest allowable distance from the magdisk, while the rear limit switch is triggered to set the brake to its minimum-load position (maximum distance).

Torque Measurement

The motor torque is measured using a load cell, U_{13}, as shown in Figure 21.

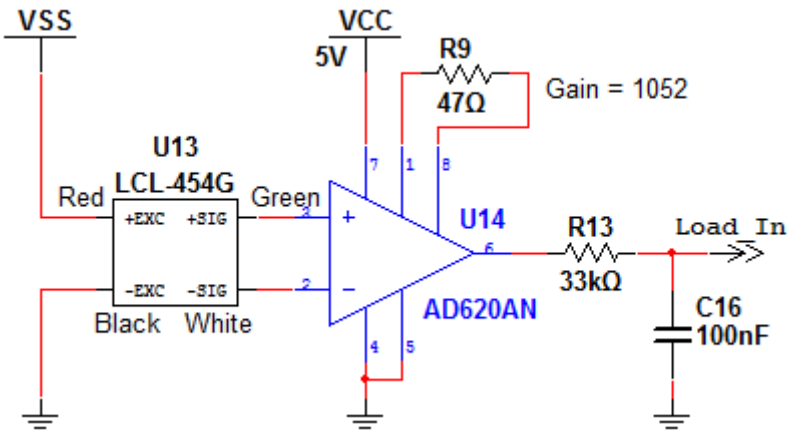


Figure 21 Torque Measurement Module

Due to the fact that the load cell output is typically on the order of millivolts, an instrumentation amplifier, U_{14}, is used with a gain of 1052 (set by R_9) to make the measurement useful. Furthermore, the output undergoes simple 1st order low pass filtering to remove any unwanted fluctuations in the measurement. The resulting output, *Load_In*, is read by the phyCORE QADC for processing.

Moreover, the load cell was calibrated using precisely measured weights of relatively even distribution, with the resulting data being fit to an appropriate curve.

The complete circuit diagram is given in Appendix A.

Recommendations

The basics for data acquisition and signal conditioning are implemented in the interface circuitry described above. However, the following recommendations could be considered to improve the motor characterization and circuit performance.

- The Thunderbird-9 motor speed controller used could not switch the motor to rotate any slower than 2500 rpm (unloaded). It would be desirable to characterize the motor at slower unloaded speeds for a more complete data set. Therefore, a better motor controller could be sourced to improve the results.

- A single phase of the motor is measured for voltage and current with the assumption that it is indicative of each phase. These modules could be tripled to measure all 3 phases and provide more information to the operator. This could reveal information such as the degree to which the phases are balanced as well as average input power.

- The QADC of the phyCORE is referenced to its internal default value of 5V. The resolution of the QADC could be more efficiently used if the solder jumper was removed and an external voltage reference applied. The analog signals measured by the phyCORE do not encompass the 5V envelope which leads to wasted accuracy.

- The motors could be further distanced from the circuitry to reduce interference. For example, the Hall Effect sensor measures the magnetic field of the current passed through it – which is undoubtedly subject to the interference of the noisy motors.

- The entire interface should be implemented on a PCB to make it more physically robust and to ensure that signal propagation is maximally efficient.

Model (Firmware) Description

The following sections describe the model deployed to the Phytec phyCORE®-MPC555 single-board computer which executes the algorithm for characterizing the motor in question.

Algorithm Initialization

When the phyCORE and interface circuitry are initially powered up, or when an iteration of characterization is completed, the system waits for a value for motor speed (kRPM) to be received via the serial port. Moreover, the motor speed controller (Thunderbird-9) requires programming upon power-up (max and min duty); therefore the phyCORE sends the max duty (2 ms on, with a period of 20 ms) with initialization.

In order to start the motor under minimum load from the eddy current brake (ECB), the stepper motor is reversed until the rear limit switch is triggered, as shown in Figure 22.

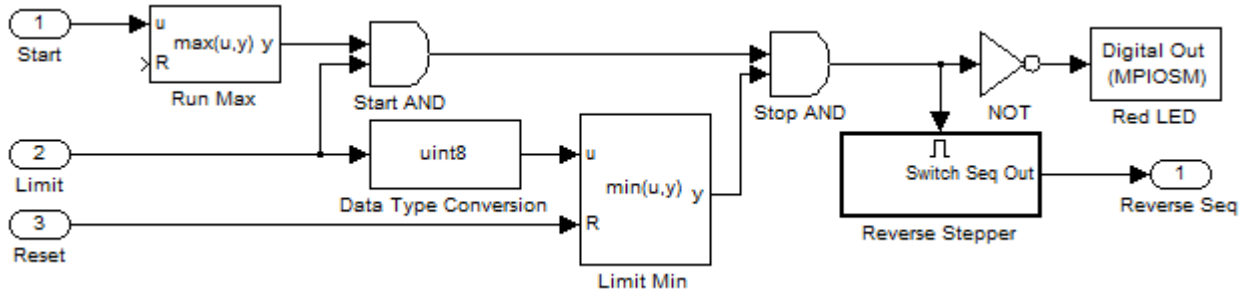


Figure 22 Reverse Stepper until Rear Limit Switch is Triggered

Where *Start* triggers the stepper motor to start reversing, *Limit* terminates the reversing, and *Reset* resets the memory to allow the stepper to be reversed again (in a subsequent iteration).

The switching sequence (*Switch Seq Out*) for the stepper motor is generated as shown in Figure 23.

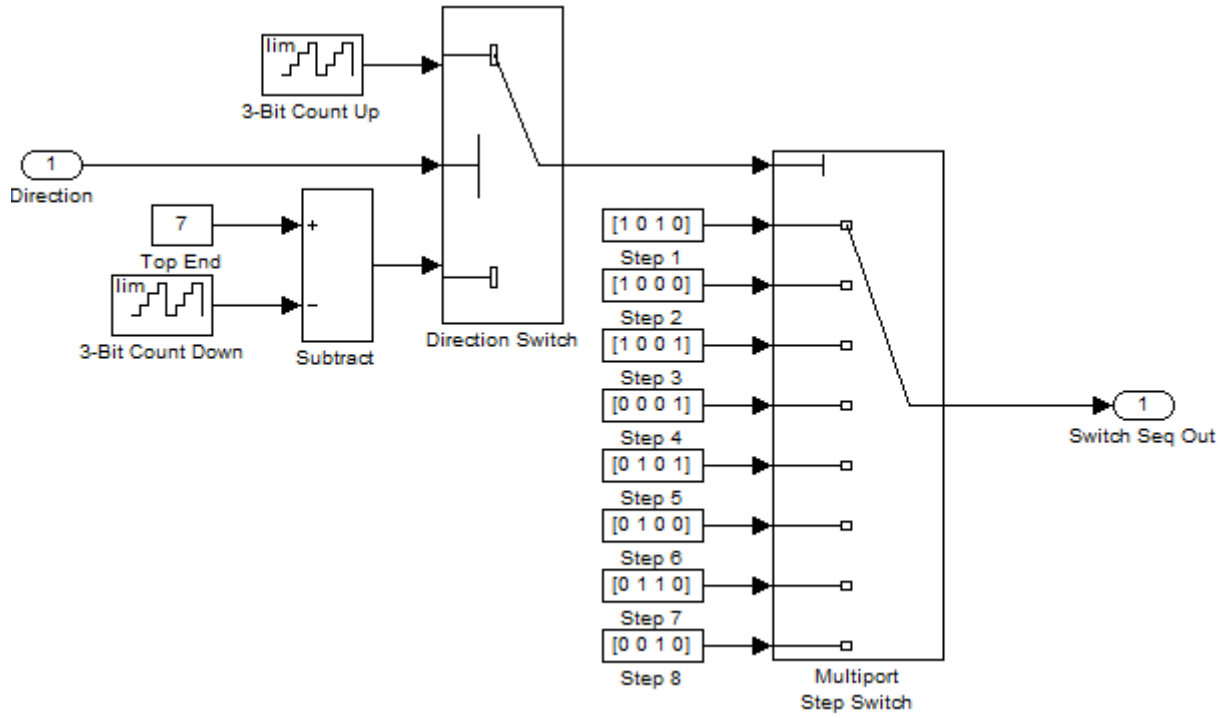


Figure 23 Stepper Motor Switching Sequence Generator

Where *Direction* executes forward motion if true or reverse motion if false.

When the rear limit switch is triggered, the stepper motor stops and the motor speed controller is sent the min duty (1 ms on, with a period of 20 ms) to complete the configuration.

Motor Speed Control

After the system has been initialized and configured (or an iteration is complete), the motor speed at which to start characterization is set, as shown in Figure 24.

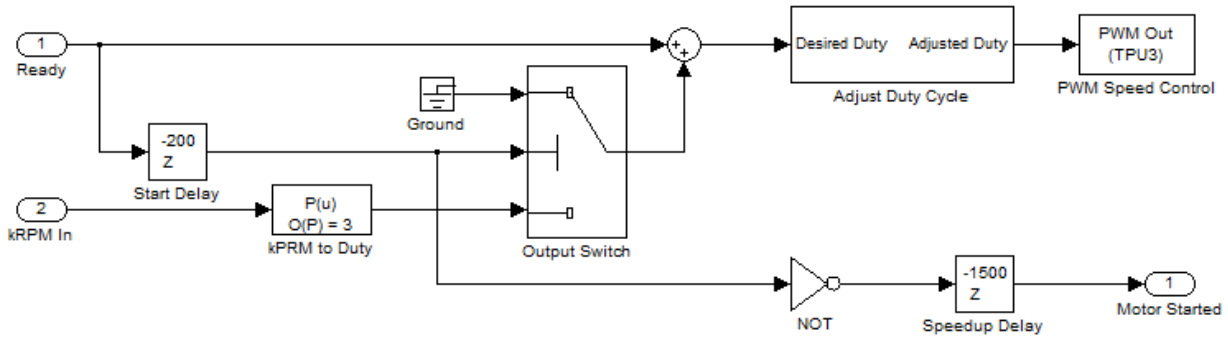


Figure 24 Motor Speed Control

Where *Ready* is a flag that states whether the motor control is programmed and *kRPM In* is the thousands of RPM desired to be set by the subsystem. The conversion from the duty waveform required by the Thunderbird-9 (*Desired Duty*) to a generic PWM is calculated in *Adjust Duty Cycle*, as shown in Figure 25. The *Speedup Delay* (1500 samples) allows the motor to reach the set speed before the algorithm progresses.

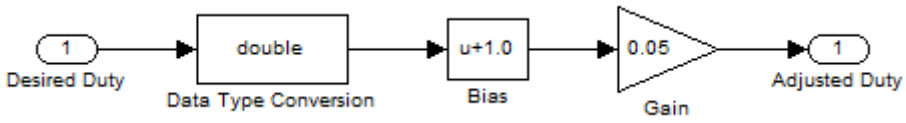


Figure 25 Duty Cycle-PWM Conversion

Eddy Current Braking

Once the motor is rotating at the desired speed, the stepper motor is set to move forward using a similar method as that to reverse the stepper.

The current induced by the encoder disk attached to the motor increases as it moves toward the magdisk (disk with two rare-earth magnets with opposite poles aligned). This causes the aligning torque to increase which, ultimately, slows the motor to a few hundred RPM. The stepper motor stops once the front limit switch is triggered.

Data Acquisition

In order to create a family of curves for each motor control duty cycle, data is acquired for each full cycle of the stepper motor switching sequence as it moves forward (shown in Figure 26).

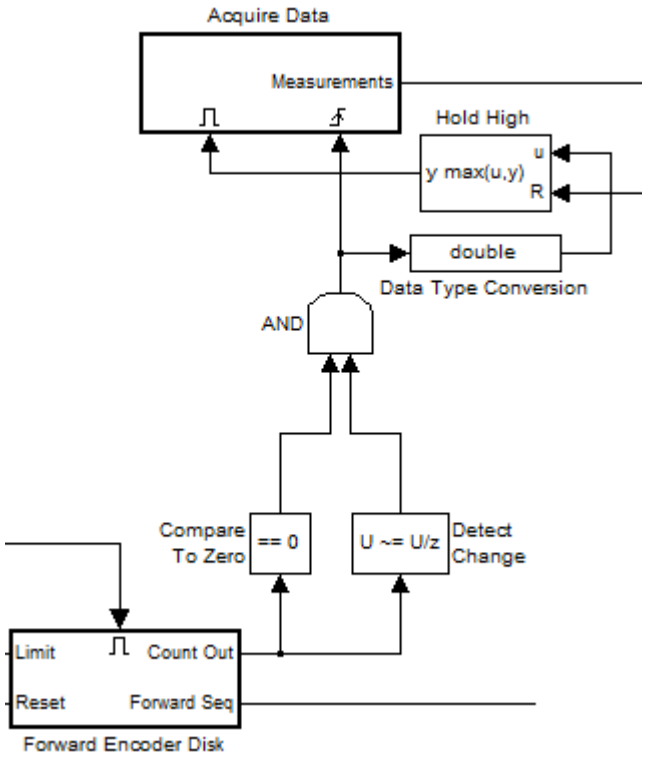


Figure 26 Data Acquisition

The data acquisition is triggered whenever the stepper switching sequence rolls over from 7 to 0 and also when it has changed from the previous value (to avoid uncontrolled acquisition if the stepper happens to stop while its sequence is 0). This provides data which is evenly distributed across the time between when the motor is under minimum load until it reaches maximum load.

As shown in Figure 27, the RPM feedback is acquired via the MPC555 MIOS Pulse Width Measurement module and calculated as discussed in the circuitry section.

The analog signals from the voltage measurement circuit, the current measurement circuit and the load cell circuit are acquired via the MPC555 QADC module. These values are resolved from the 10-bit ADC back to voltage and then converted to real-world values based on calibrated (polynomial) transfer functions.

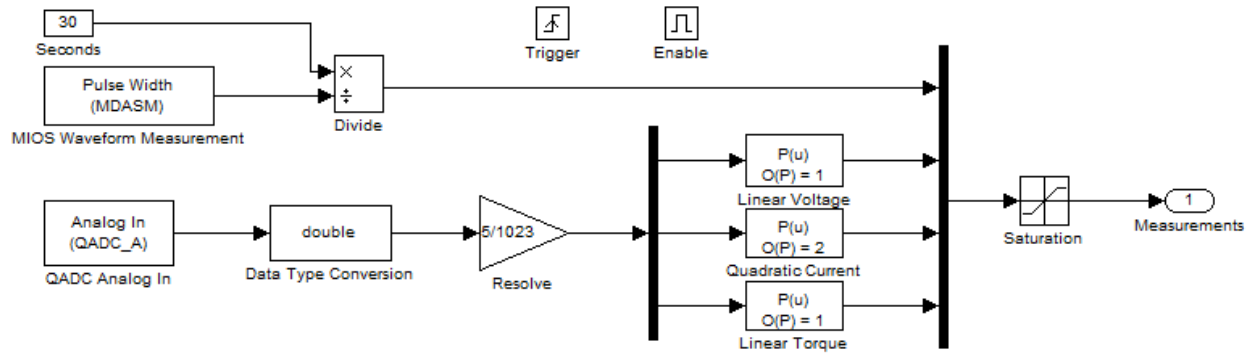


Figure 27 Data Processing

Data Transmission

Each time data is acquired, it is transmitted over the serial port via the MPC555 SCI1 module, as shown in Figure 28.

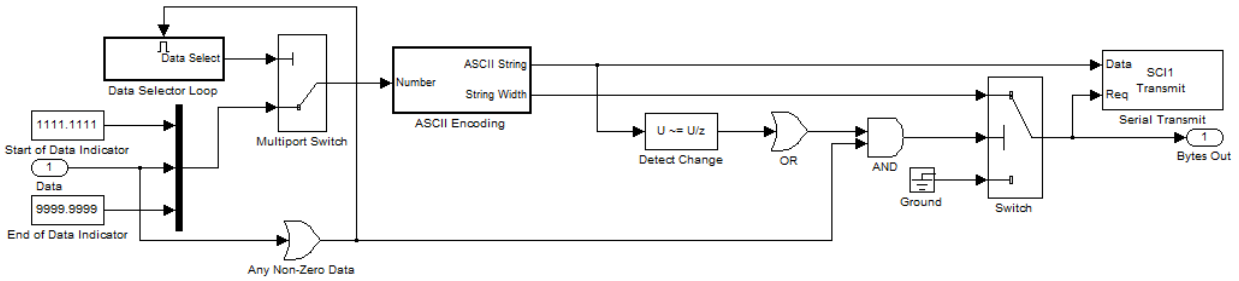


Figure 28 Data Transmission

The acquired data, *Data*, contains the four elements (RPM, voltage, current, and torque) that each get encoded as ASCII, as shown in Figure 29, and transmitted via serial. Start-of-data and end-of-data indicators are used to facilitate flow control. Furthermore, repeated values are not transmitted to avoid congestion across the serial line.

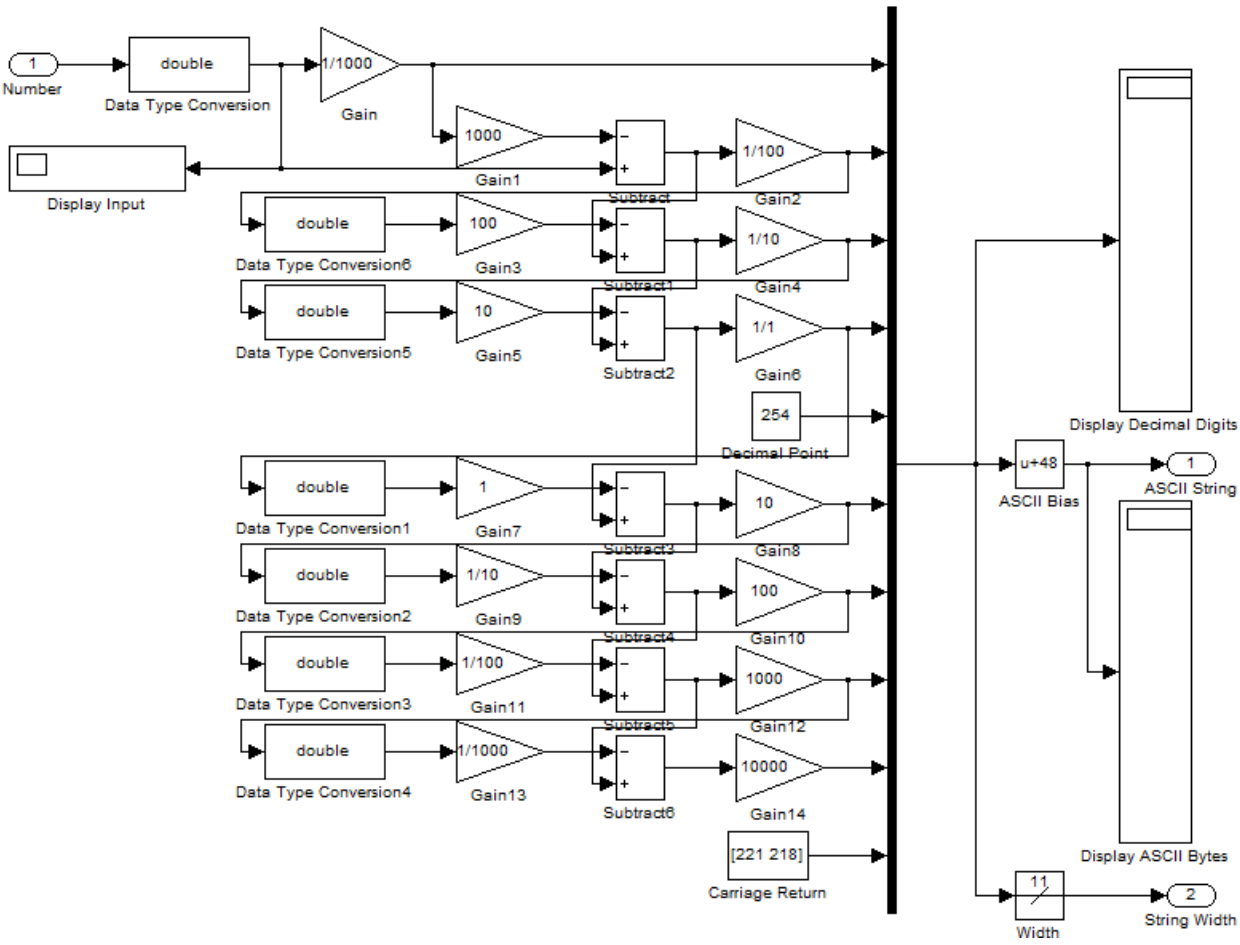


Figure 29 ASCII Encoding

Number is a number between 0 and 9999.9999 to be encoded as the ASCII format compatible with the serial communication of the phyCORE.

The characterization routine will repeat every time a value is sent via the serial port with the exception of the *Program Motor Speed Controller*, which is only needed upon power-up to configure the Thunderbird-9.

The high-level model is included as Appendix B.

A flowchart describing the model is included as Appendix CAppendix .

Recommendations

A working model for utilizing the acquired signals to characterize the motor is described above. However, the following recommendations could be considered to improve the implementation of the given model.

- The Thunderbird-9 motor speed controller used requires programming of max and min duty cycles upon every power-up. If a motor controller were sourced that either did not require programming or could store its programmed configuration in memory, the model complexity could be significantly reduced.

- Due to the fact that the characterization points are ultimately fit to a curve, the motor speed control is open-loop. If exact data points are desired, the model could be adjusted to use the RPM feedback from the MIOS module in conjunction with a PI(D) control algorithm to bring the rotating speed of the motor to within a certain threshold.

- A design challenge was undertaken to implement the entire model using common Simulink™ blocks (as well as the Embedded Target for Motorola® MPC555). Therefore, the model complexity could be significantly reduced if any or all of the subsystems were replaced with a more conventional programming language, such as C.

Graphical User Interface Description

The Phytec phyCORE®-MPC555 acquires motor characterization data, namely, RPM, voltage, current, and torque in conjunction with the interface circuitry. This data is sent to the host computer via serial port communication. After sending a command to the phyCORE, the host computer receives the acquired data, which is logged and graphically presented. In order to achieve this, a dedicated computer application is needed. To design this computer application, Matlab Graphical User Interface Development Environment (GUIDE) was used. Design considerations for this particular choice were:

- Matlab has built-in serial communication functionality, which can be implemented easily.
- Matlab has versatile capabilities for graphical representation of data.
- Matlab also offers flexibility to modify the source code. Engineering professionals are usually well versed in Matlab, which was also of interest, as this dynamometer is expected to be used by engineering students afterwards.

Design requirements of the GUI were:

- The ability to send commands to the phyCORE via a serial port in order to run the motor at specified RPMs.
- Implementation of a robust protocol for serial communication with the phyCORE.
- Graphical representation of the data in real time, as it is received via the serial port.
- Logging of the data for subsequent analysis and storage.

The following sections deal with the design approaches undertaken to meet these requirements.

Host-Controller Serial Communication

The phyCORE has serial ports available to connect to the host computer. As serial ports are becoming obsolete, a Serial-to-USB converter was used to connect to the embedded data acquisition board. This converter emulates a serial port over USB so that the computer may recognize serial communication. This was advantageous in the development effort as recent desktop and laptop computer do not have an on-board serial port. Using built-in Matlab serial functions eliminated much of the low-level design effort.

A custom communication protocol was implemented in the firmware to achieve the desired data resolution and facilitate flow control. Incoming data points were formatted into 11-byte strings with the following format: 4 numeric characters, one decimal point, another 4 numeric characters, followed by

the line feed and carriage return character. A data packet was comprised of 7 such strings. The leading string was a predefined start-of-data identifier, *1111.1111*, which indicates the beginning of a data packet. The following four strings contained consecutively RPM, Voltage, Current, and Torque data. The terminating string in a data packet was the predefined end-of-data identifier, *9999.9999*, which indicated termination of the reception of one single data packet. Validation of data packets was implemented by the algorithm presented in Figure 30.

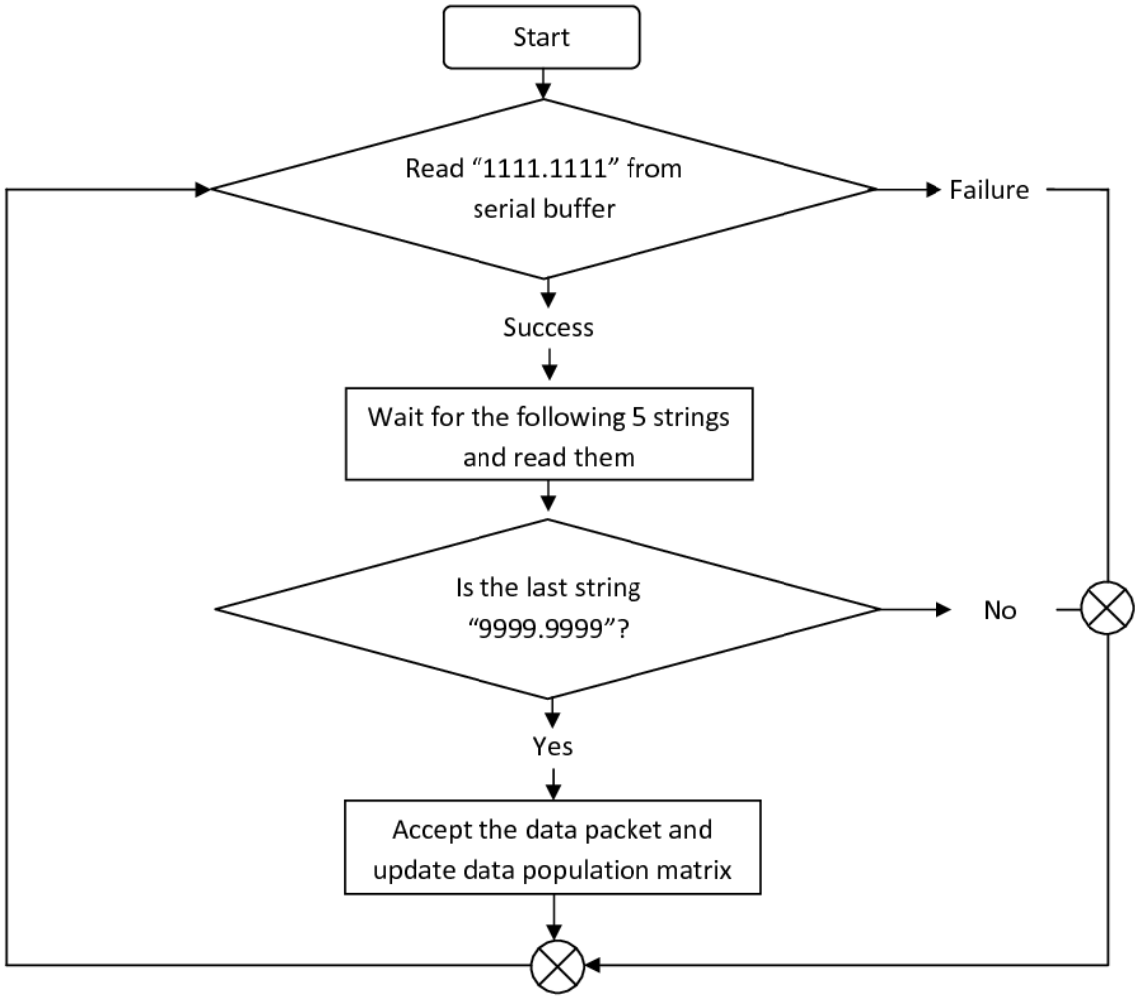


Figure 30 Communication Protocol Algorithm

GUI Control Organization

A snapshot of the GUI is presented in Figure 31.

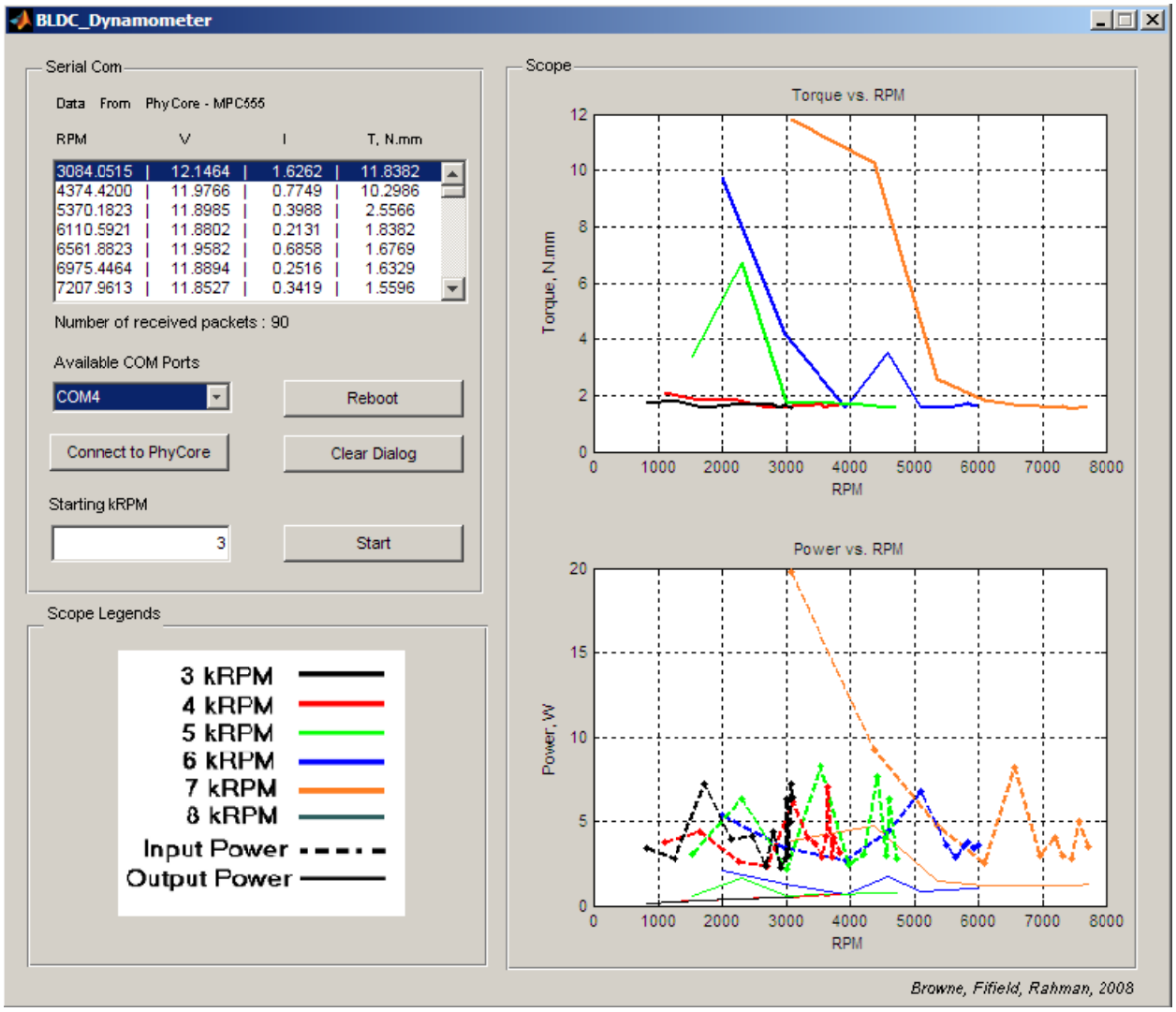


Figure 31 GUI Snapshot

The GUI has three frames where various controls were organized. These frames are named as:

- Serial Com – this frame has the controls for serial communication with the phyCORE. A list box was used to show the data as it is received in the serial buffer. A popup menu was used to allow the user to choose from the available serial COM ports to connect. A push button, *Connect to PhyCore*, initiates the serial communication. A push button, *Reboot*, closes all open serial ports and another push button, *Clear Dialog*, clears the list data box dialog. In this frame, there is a

text box for the user to specify the initial RPM with which the data acquisition and representation algorithm starts. A push button, *Start*, initiates this algorithm.

- Scope – this frame contains two 2-D graphs. One is *Torque vs. RPM* where the torque and the respective RPMs are plotted. The other graph is *Power vs. RPM* where input power and output power are plotted against respective RPM.
- Scope Legends – this frame has one image showing the color legends for the graphs.

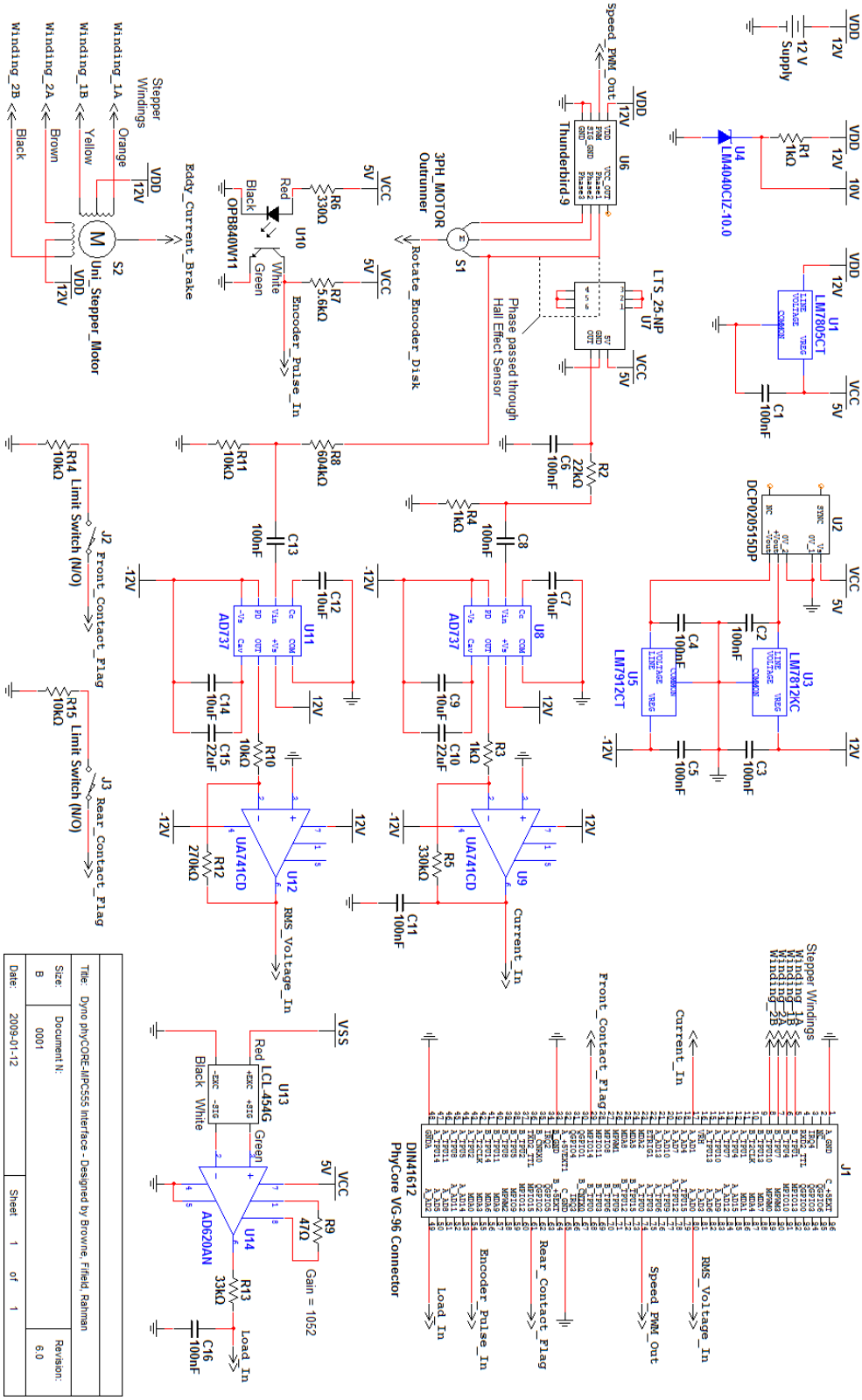
Data Acquisition and Control Algorithm

This algorithm starts with the user specified initial thousands of RPM (kRPM). The motor attains this RPM under minimum load and then, as the stepper motor advances towards the brake, it is loaded. The motor controller tries to maintain this RPM and works against the braking force, drawing more electrical power, as required.

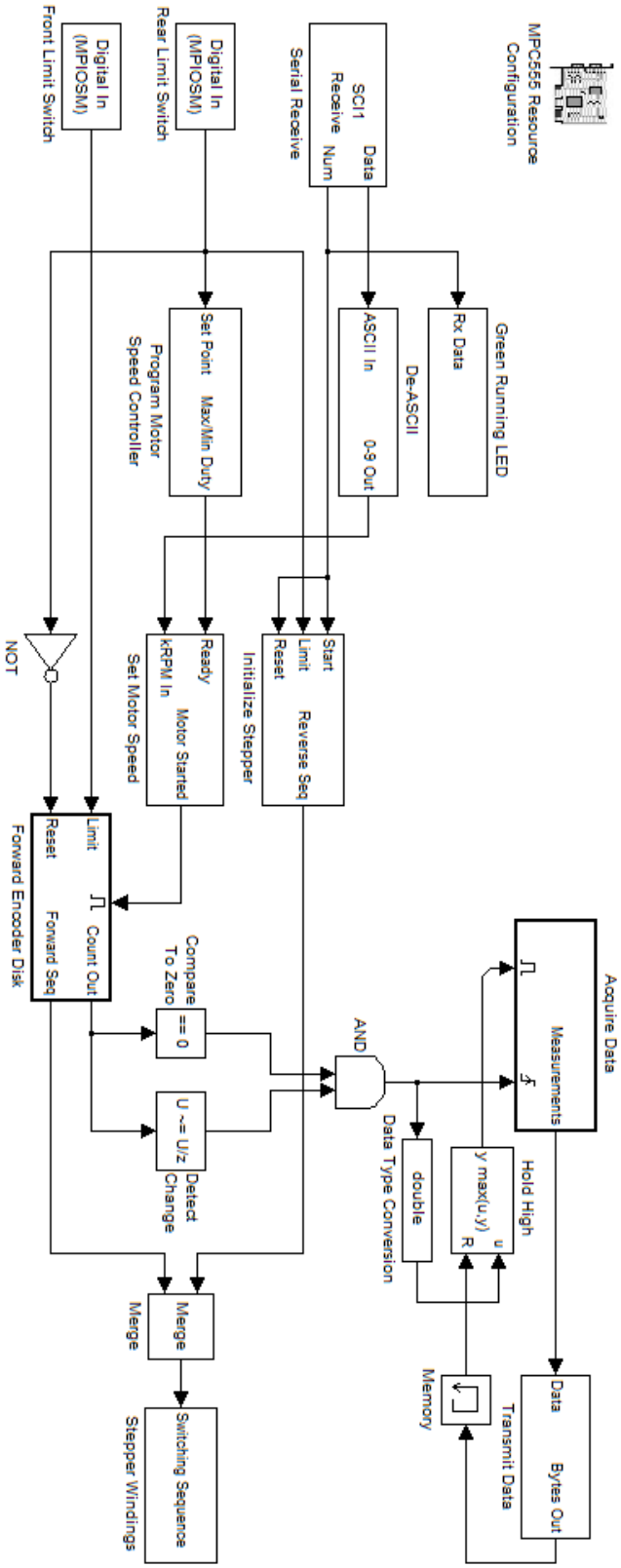
Data is sent over the serial port to be logged, graphically presented, and saved to a log file. This data gives us one set of characteristic curves for that particular RPM which the motor controller is trying to maintain. When the moving platform hits the front limit switch, data transfer from the phyCORE is suspended, the motor reverses, the kRPM is incremented, and the algorithm iterates.

After logging the data for 7 kRPM, the GUI sends a command to stop the motor. Subsequently, the log file is saved and closed and the serial communication is terminated. For a detailed and comprehensive description of the algorithm, please refer to Appendix D.

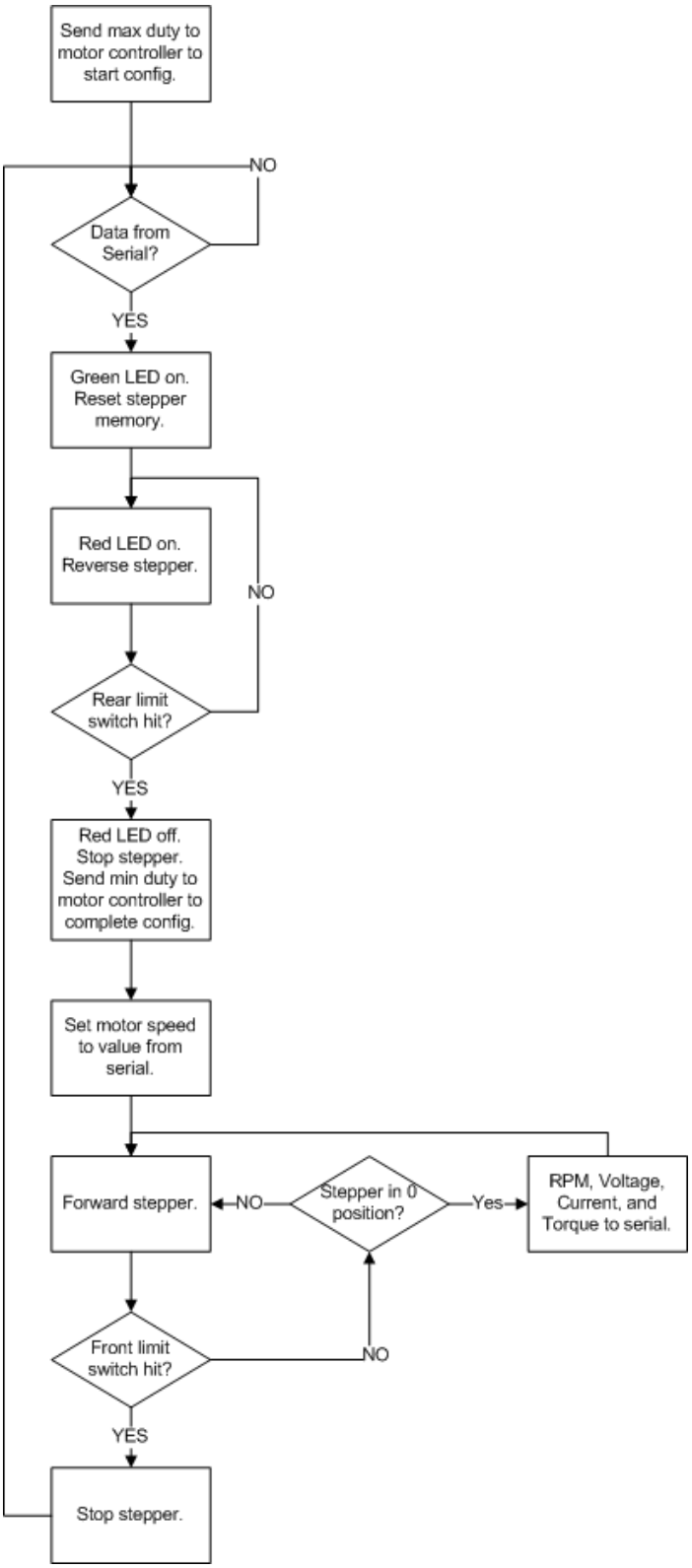
Appendix A


Title: Dyno phyCORE-MPC555 Interface - Designed by Browne, Fifield, Rahman
Size: B
Document N: 0001
Revision: 6.0
Date: 2009-01-12
Sheet 1 of 1
Thunderbird-9
LTS_25-NP
DCP020515DP
LM7805CT
LM7812KC
LM7912CT
LM4040CIZ-10.0
AD737
UA741CD
AD620AN
LCL-454G
DIN41612
PhyCore VG-96 Connector
3PH_MOTOR Outrunner
Uni_Stepper_Motor
OPB840W11
Phase passed through Hall Effect Sensor
Rotate_Encoder_Disk
Encoder_Pulse_In
Speed_PWM_Out
Eddy_Current_Brake
Front_Contact_Flag
Rear_Contact_Flag
RMS_Voltage_In
Current_In
Load_In
Gain = 1052

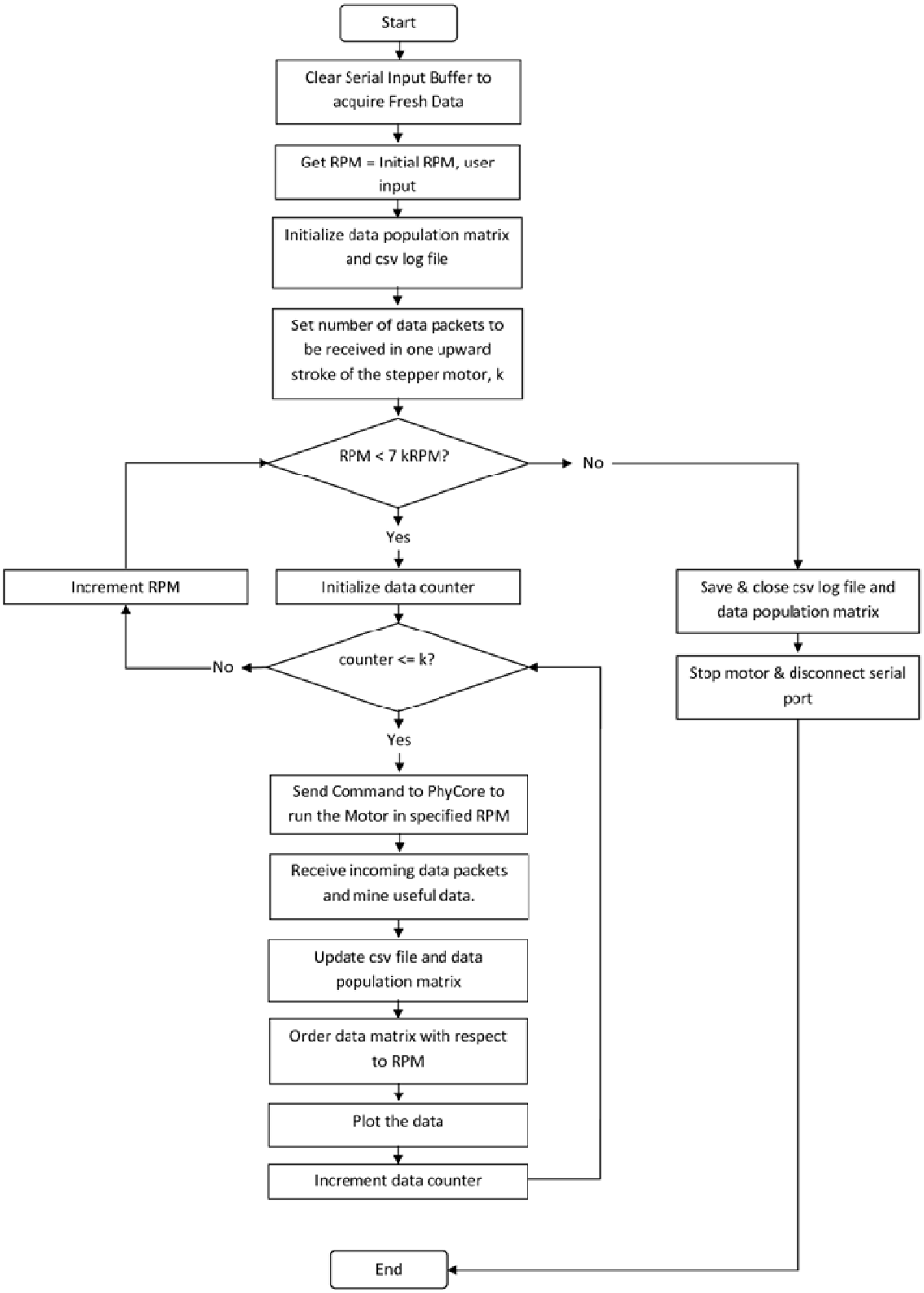
Appendix B


MPC555 Resource Configuration
Digital In (MPIOSM)
Front Limit Switch
Digital In (MPIOSM)
Rear Limit Switch
SCI1 Receive Num Data
Serial Receive
Set Point Max/Min Duty
Program Motor Speed Controller
ASCII In 0-9 Out
De-ASCII
Rx Data
Green Running LED
NOT
Ready kRPM In Motor Started
Set Motor Speed
Start Limit Reset Reverse Seq
Initialize Stepper
Limit Reset Count Out Forward Seq
Forward Encoder Disk
Compare To Zero == 0
U ~= U/z Detect Change
AND
Acquire Data
Measurements
Data Type Conversion
double
y max(u,y) u R
Hold High
Merge
Merge
Memory
Data Bytes Out
Transmit Data
Switching Sequence
Stepper Windings

Appendix C


Send max duty to motor controller to start config.
NO
Data from Serial?
YES
Green LED on. Reset stepper memory.
Red LED on. Reverse stepper.
NO
Rear limit switch hit?
YES
Red LED off. Stop stepper. Send min duty to motor controller to complete config.
Set motor speed to value from serial.
Forward stepper.
NO
Stepper in 0 position?
Yes
RPM, Voltage, Current, and Torque to serial.
Front limit switch hit?
NO
YES
Stop stepper.

Appendix D


Start
Clear Serial Input Buffer to acquire Fresh Data
Get RPM = Initial RPM, user input
Initialize data population matrix and csv log file
Set number of data packets to be received in one upward stroke of the stepper motor, k
RPM < 7 kRPM?
No
Yes
Increment RPM
Initialize data counter
Save & close csv log file and data population matrix
counter <= k?
No
Stop motor & disconnect serial port
Yes
Send Command to PhyCore to run the Motor in specified RPM
Receive incoming data packets and mine useful data.
Update csv file and data population matrix
Order data matrix with respect to RPM
Plot the data
Increment data counter
End

Works Cited

Center for Innovation in Product Development. (1999, January 20). *Understanding D.C. Motor Characteristics*. Retrieved December 2, 2008, from Designing with D.C. Motors: http://lancet.mit.edu/motors/motors3.html

Nave, R. (2004, February 16). *Magnetic Dipole Moment*. Retrieved December 6, 2008, from HyperPhysics: http://hyperphysics.phy-astr.gsu.edu/Hbase/magnetic/magmom.html

Power Dynamometers. (n.d.). *The Basics of Dynamometer*. Retrieved November 29, 2008, from Power Dynamometers: http://www.powerdynamometers.com/

Spencer, R. (1997, April 8). *Biot-Savart Law*. Retrieved December 20, 2008, from Brigham Young University: http://maxwell.byu.edu/~spencerr/websumm122/node70.html

Tankersley, L., & Mosca, E. P. (2005). *Introducing Faraday's Law.* United States Naval Academy, Physics Department, Annapolis.

Wikipedia. (2008, Novemebr 28). *Deinonychus*. Retrieved November 30, 2008, from Wikipedia, The Free Encyclopedia: http://en.wikipedia.org/wiki/Deinonychus