Memo

To: Xiaoli Zhang and Mollie Brombaugh

From: Ben Pacheco and Tessa Haws

Team #: N-424

Date: September 12th

Re: Lab #1: Motor Characterization

Problem Statement

This lab aims to analyze and quantify the performance characteristics of DC motors by investigating their torque, power, and efficiency across different operating conditions. A cup with a known weight was lifted using a motor connected through a gear train to a pulley. The data collected will be used to plot performance graphs to characterize the motors. This process also allows us to gain experience with the Arduino programming environment, enabling precise motor control and data acquisition. The focus is on understanding the relationships between motor speed, torque, power, and efficiency and the effects of varying pulse-width modulation (PWM) settings on motor performance.

Methods

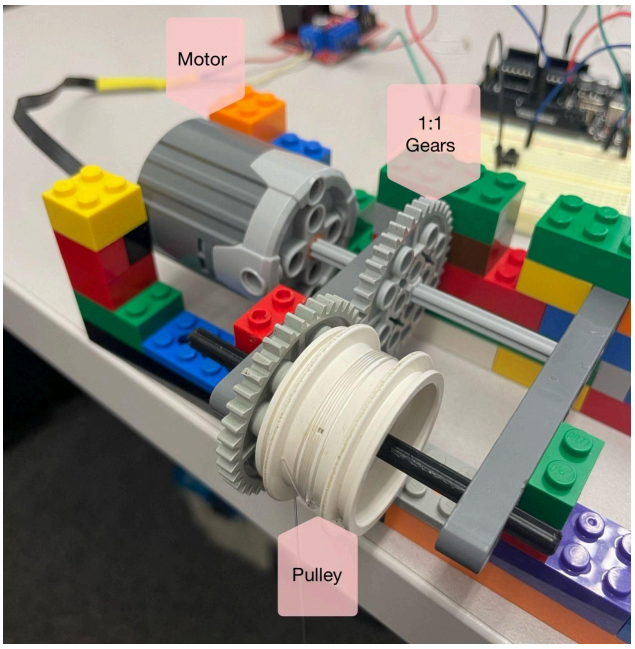


Figure 1. Labeled image of test setup

a.

The motor speed at the input and output remains identical due to using a 1:1 gear ratio. Consequently, the torque at the output load is equivalent to the torque generated by the input motor. The radius of the pulley is measured and used with a known weight of washers to calculate torque. To determine the motor's linear speed, we measure the distance traveled by the cup carrying the washers and divide this distance by the time required for the cup to reach its destination.

b.

The initial experiment involved determining the maximum number of washers the motor could lift from a predetermined height. This process requires identifying the motors' stall torque and the corresponding load that induced the stall, then systematically reducing the load by one for further testing. During this phase, we recorded both motors' travel time, current, and voltage with the software using the Arduino's built-in timer for the travel time and calculations from the registers A0 and A1.

Next, we conducted a Pulse Width Modulation (PWM) test on both motors, varying the PWM values in increments of 20 percent, starting from 20 percent to 100 percent. Throughout this experiment, we measured the travel time using the Arduino's millis function data.

The final experiment assessed the timing error by conducting ten trials with an empty cup. For each trial, we measured the travel time using Arduino's millis function, enabling us to evaluate the precision and consistency of the timing measurements.

c.

In this lab, we primarily used the example code, making essential modifications to suit our specific requirements. Based on button presses, we implemented logic to control the motor's start and stop actions. Additionally, we integrated the subtraction of stop time and start time to calculate the delta time, representing the duration the motor was active. Through the use of the millis() function, we measured the motor's runtime, with the results being printed to the serial monitor for straightforward analysis.

Results

a.

Table 1. Data recorded from lifting tests for motor A.

Num Washers	Travel Time (s)	Current (mA)	Voltage (V)
0	3.49	93	4.06
1	3.88	92	4.05
2	4.18	108	3.90

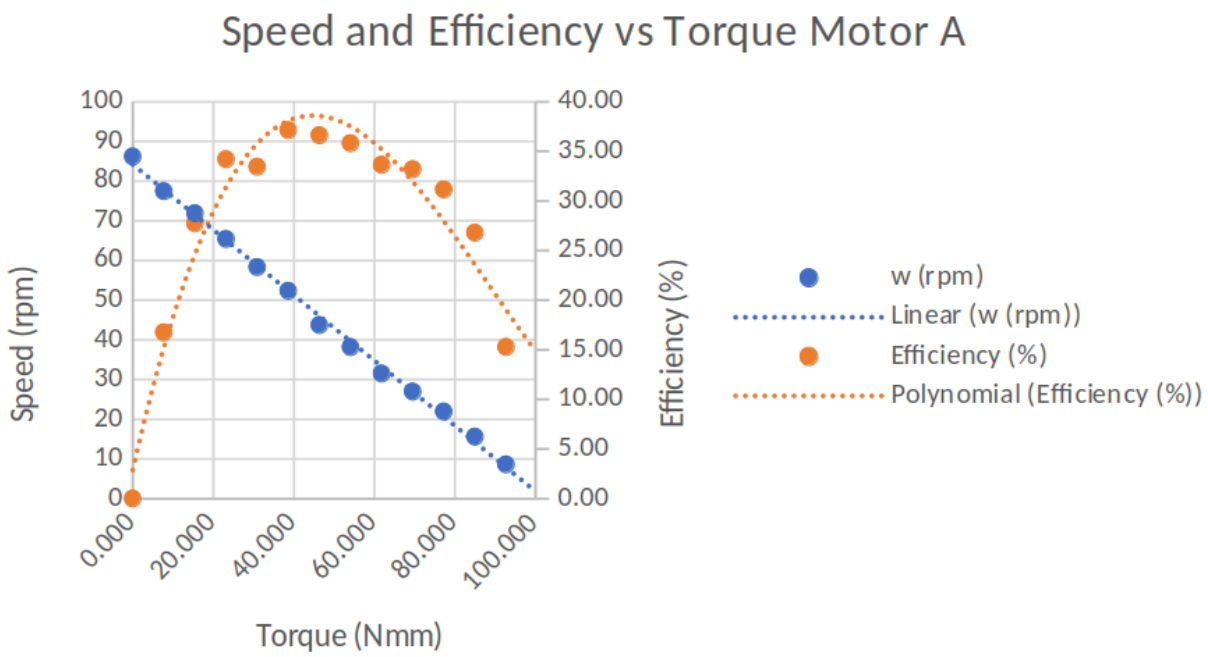
3	4.59	125	3.71
4	5.15	177	3.19
5	5.74	183	3.12
6	6.87	193	3.01
7	7.87	222	2.72
8	9.54	240	2.52
9	11.15	282	2.10
10	13.70	305	1.87
11	19.25	339	1.53
12	34.88	318	1.72
13	STALL	489	0.00

Table 2. Data recorded from lifting tests for Motor B.

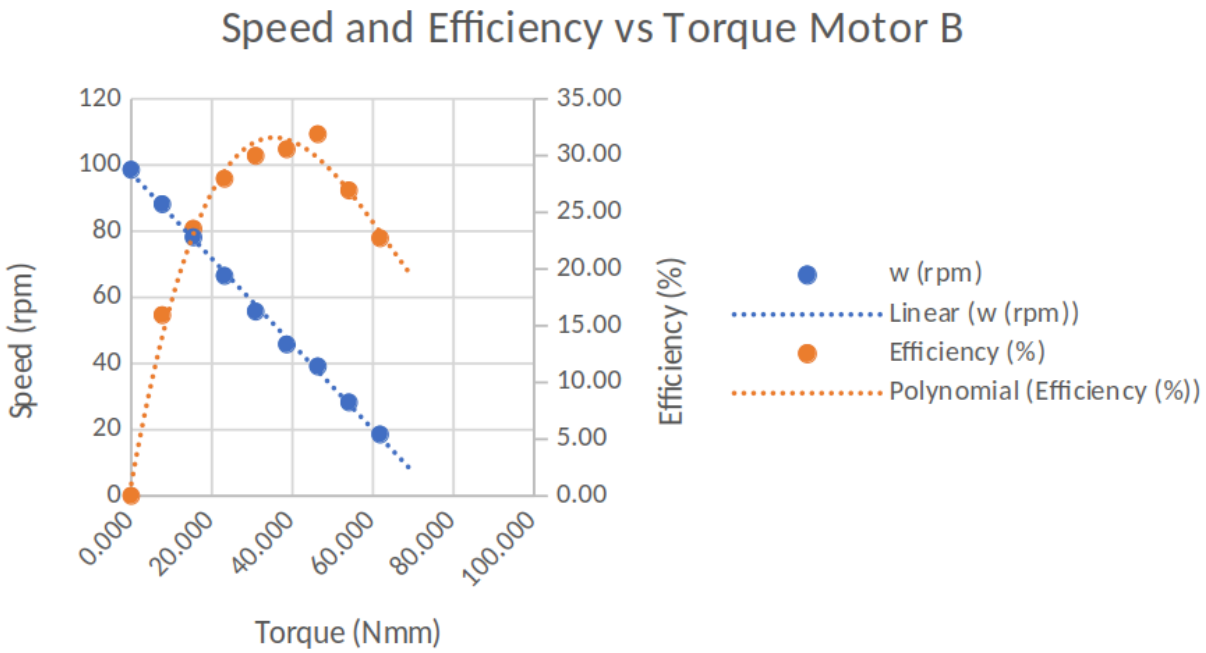
Num Washers	Travel Time (s)	Current (mA)	Voltage (V)
0	3.05	126	3.72
1	3.41	118	3.80
2	3.85	160	3.36
3	4.52	189	3.06
4	5.39	214	2.81
5	6.57	243	2.49
6	7.68	273	2.18
7	10.64	282	2.11
8	16.20	332	1.59
9	STALL	488	0.00

b.

Graph 1. Speed and Efficiency against Torque for Motor A.

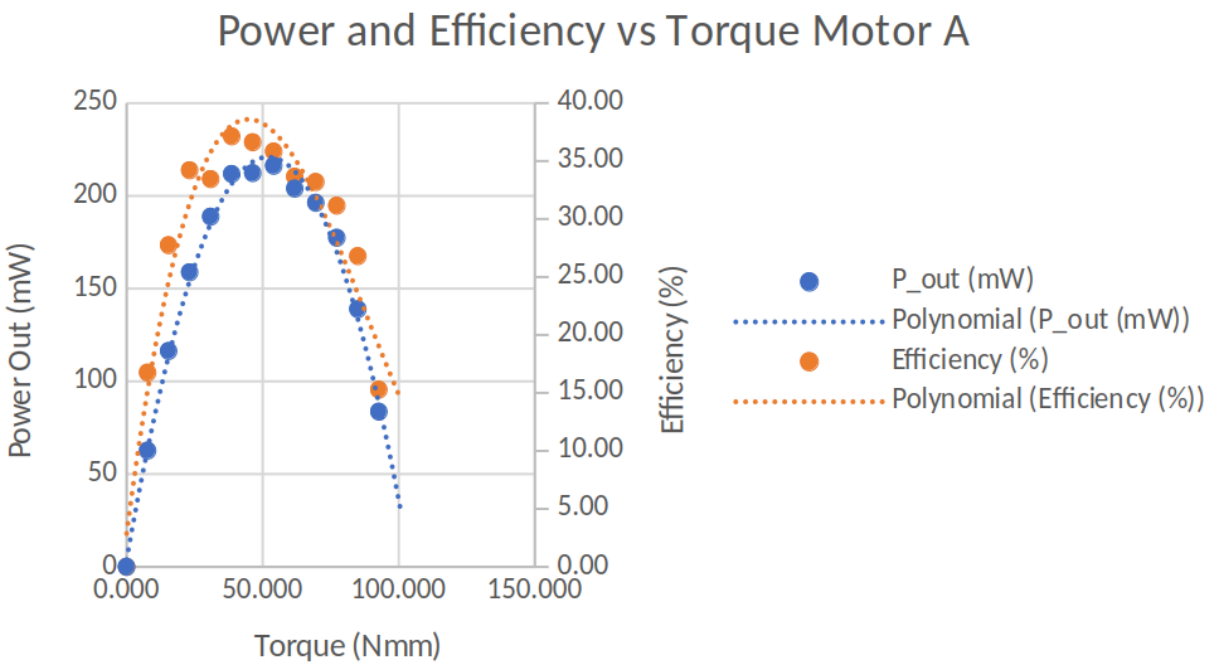


Graph 2. Speed and Efficiency against Torque for Motor B.

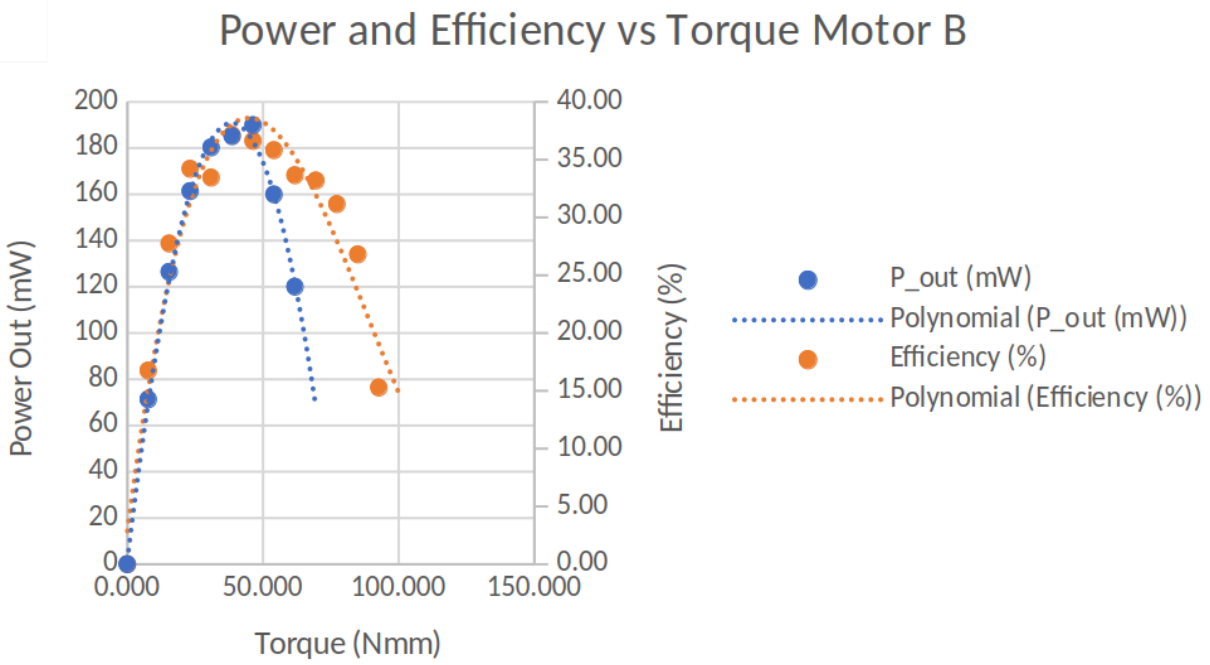


c.

Graph 3. Power and Efficiency against Torque for Motor A.

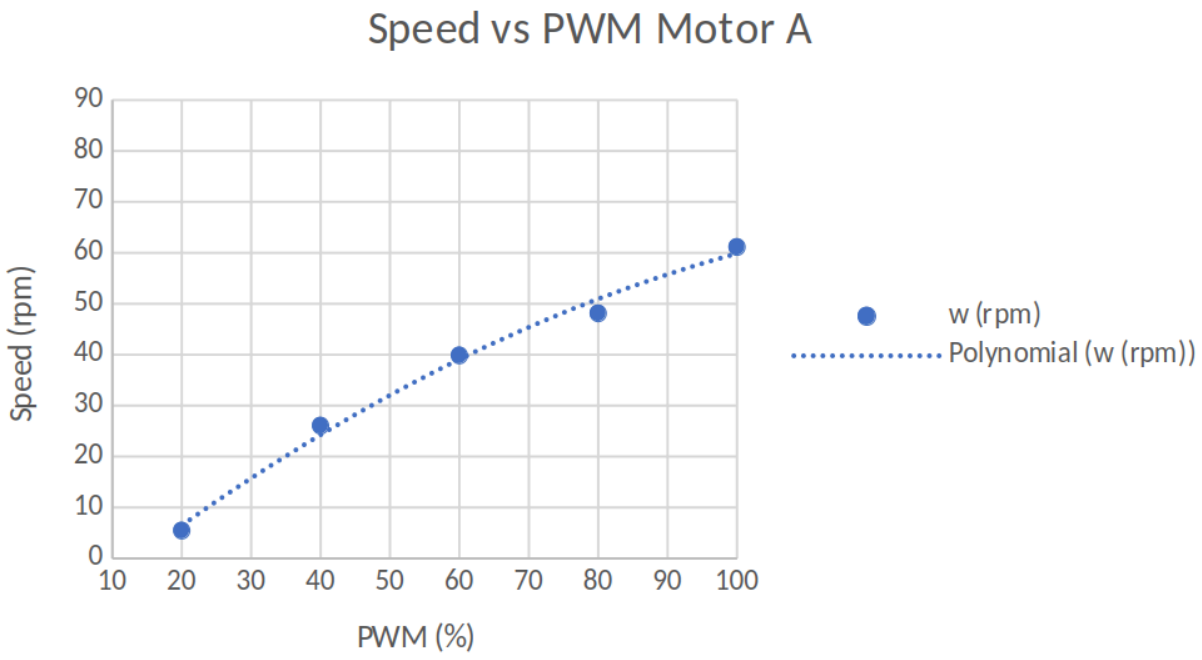


Graph 4. Power and Efficiency against Torque for Motor B.

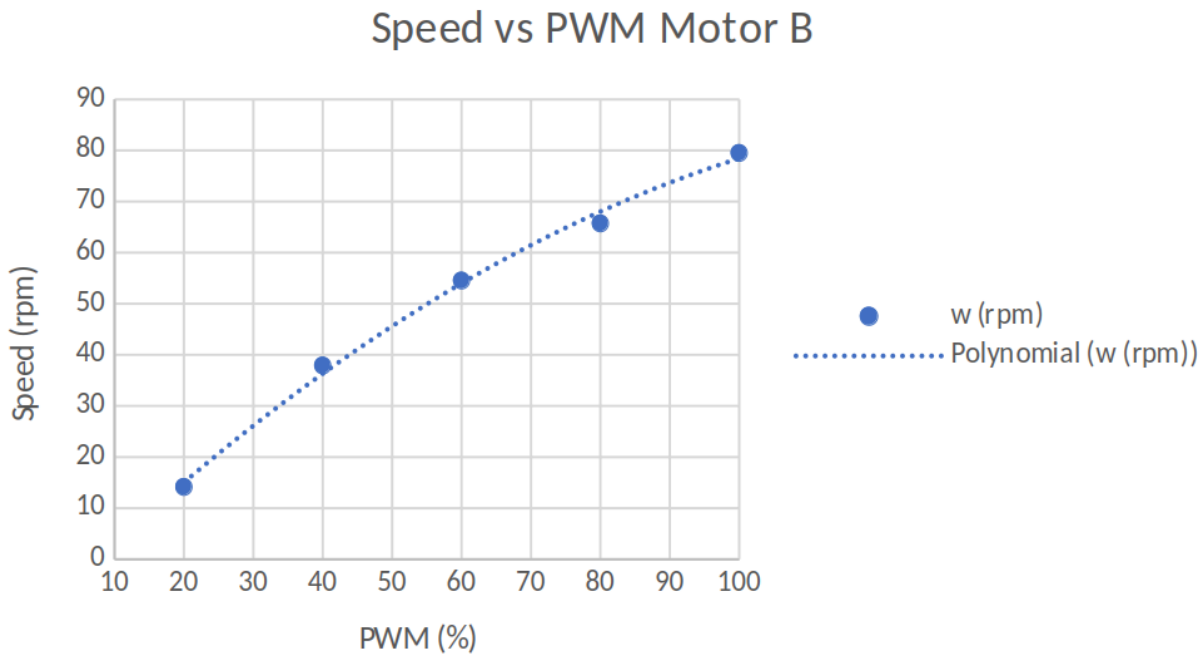


d.

Graph 5. Motor A's Speed at Different PWM Duty Cycles.



Graph 6. Motor B's Speed at Different PWM Duty Cycles.



e.

Motor A's standard deviation was 6.7361, and the mean was 0.1134 seconds for 5 washers.

Motor B’s standard deviation was 5.7668, and the mean was 0.1311 seconds for 5 washers.

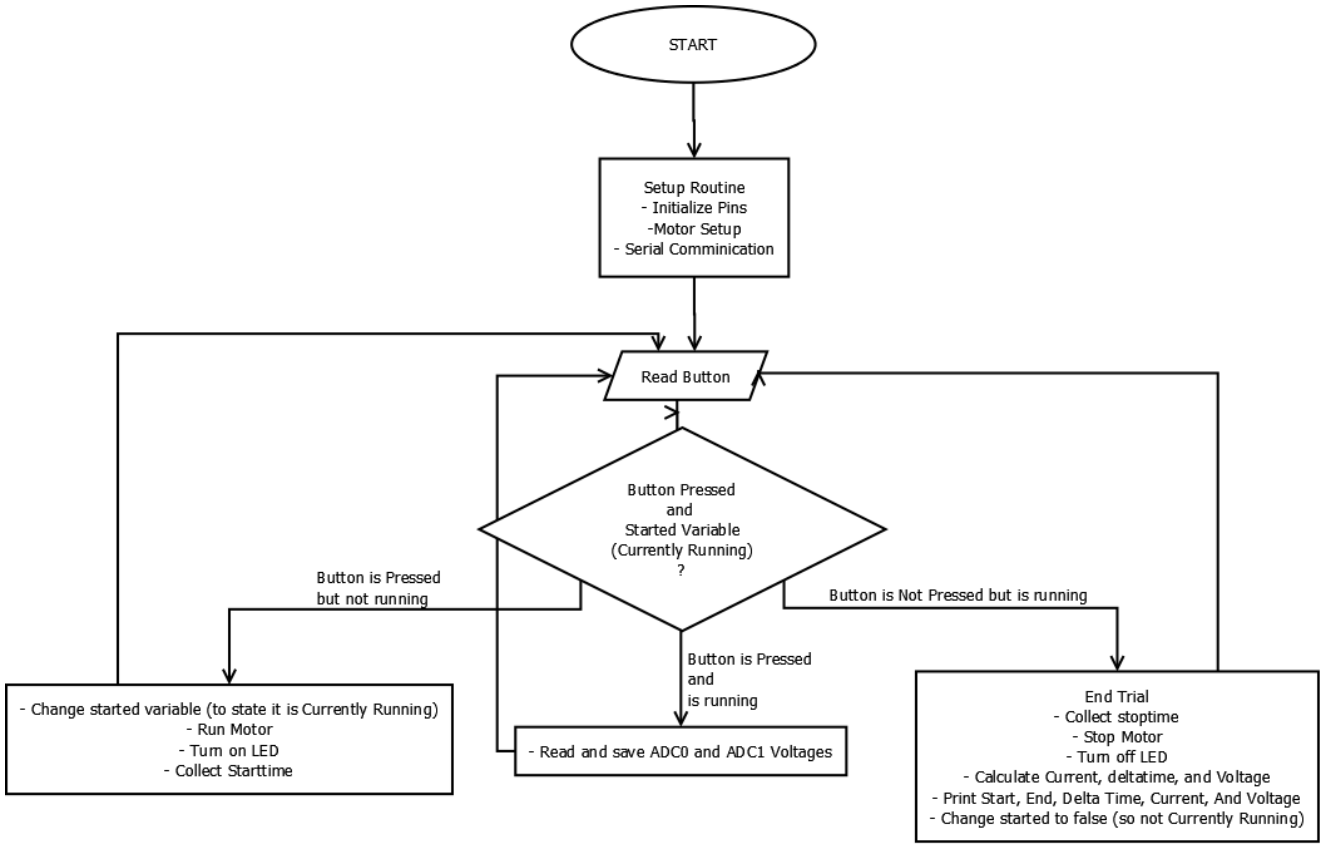
Conclusion

The speed and power vs. torque curves represent the motors' performance, crucial for optimizing motor operation during robot construction. Understanding the ideal regions to operate the motors allows for better design decisions that enhance overall motor performance. Stall torque, the maximum torque a motor can generate when its output rotational speed is zero, is a critical parameter in this analysis. Like the speed and power, the stall torque for motor A was higher than expected, while motor B's stall torque was closer to the average. For both motors, the maximum power is approximately half of the stall torque and maximum RPM, with some deviations due to experimental data errors. The minimum power for both motors occurs at zero torque, as no work is being done in this state. Motor B performed as expected, but motor A exceeded expectations, highlighting the need for careful gear train design to ensure optimal operating conditions for both motors.

Adjusting the power settings in the code did not produce a linear response in speed output. Motor A's response curve was slightly lower than motor B's, indicating that slight adjustments in PWM settings will be necessary to match their speeds. Additionally, some measurement errors were introduced due to the manual nature of the recordings. A test to quantify this error involved repeating the same experiment without changing any variables, resulting in a standard deviation of 0.11s in one experiment and 0.13s in the other. While this error level is acceptable, more precise measurements could be achieved using a break beam sensor.

References

Appendices



A.

```
/* MEGN 441 Lab 1
*  0842020 mshapiro
*  Updated 01122024 nave
*
* TODO inserted wherever edits are needed.
*
*/

#define pushButton 2 // install a button with its output into Pin 2
#define adc0 A0 // measure analog DC (ADC) voltage in Pin A0
#define adc1 A1 // measure analog DC (ADC) voltage in Pin A1
#define ledPin 13 // Builtin LED is connected to Pin 13

// H-Bridge initialization
```

```
#define SHIELD false // If you have a kit with the moto shield, set SHIELD to true
// If you have the Dual H-Bridge controller w/o the shield, set SHIELD to false

#define A 0 // Use letters instead of numbers for motors.ino
#define B 1

// SHIELD Pin varables - cannot be changed
// Can be deleted if using Dual H-Bridge
#define motorApwm 3
#define motorAdir 12
#define motorBpwm 11
#define motorBdir 13
// Driver Pin variables - any 4 pwm pins (marked with ~ on your board)
// Can be deleted if using SHIELD
#define IN1 9
#define IN2 10
#define IN3 5
#define IN4 6

unsigned long starttime, stoptime, deltatime;
bool started = false; // State variable to track system start/stop
int R = 10; //TODO: Input shunt resister value (in Ohms)
float ADC0, ADC1;

void setup() { // the setup routine runs once when you press reset:

  // Initialize I/O pins
```

```
  pinMode(pushButton, INPUT_PULLUP);  // set the pushbutton's pin as a
pull-up input:
  pinMode(ledPin, OUTPUT);  //set the LED pin as an output
  pinMode(adc0, INPUT); // set both ADC pins to input
  pinMode(adc1, INPUT);

  //use pre-built functions for motor setup (found in motors.ino)
  motor_setup();

  // initialize serial communication at 9600 bits per second:
  Serial.begin(9600);
}

void loop() { // the loop routine runs over and over again forever (or
until reset button/loss of power):

  // read the pushButton pin:
  bool buttonState = digitalRead(pushButton);
  // TODO: 1. Set up logic to run trial
    //Useful to debug button hardware:
    //Serial.println(buttonState);

  if (!buttonState & !started){
    // On trial start:
    started = true; //Keep track of trial start
    // 2. Run motor (turn on LED if you wish)
    run_motor(A, 255); // motor A is on at full speed (pwm = 0-255)
    digitalWrite(ledPin, HIGH); // LED to indicate motor running
    // 3. Measure start time
```

```
    starttime = millis();
    Serial.println("start \t stop \t dt \t I \t V");
    Serial.println(starttime);
  }else if (!buttonState & started){
    // 4. Read ADC voltages
    ADC0 = analogRead(adc0) * 5.0/1023.0; // Convert 10-bit value (0-1023) to voltage (0-5 V)
    ADC1 = analogRead(adc1) * 5.0/1023.0;

  } else if (buttonState & started){
    // On trial finish:
    // 3. Read timers
    stoptime = millis(); //stop timer
    // 5. Stop motor
    run_motor(A, 0); //turn motor off
    digitalWrite(ledPin, LOW); // turn LED off
    // 6. Calculate current, motor voltage, deltatime
    float current = (ADC0-ADC1)/R; //TODO (Use V=IR across small resistor, convert to mA)
    float motorVoltage = ADC1; //TODO
    deltatime = (stoptime-starttime); //TODO
    // 7. Print results to serial monitor
    Serial.print(" \t ");
    Serial.print(stoptime); //print stop time
    Serial.print("\t ");
    Serial.print(deltatime);
    Serial.print("ms\t ");
    //Serial.print(current*1000);
    //Serial.print("mA\t ");
```

```
    Serial.print(motorVoltage);
    Serial.println("V");
    started = false; // Reset for next trial
  }

}
```

B.

C. Test Results

Constants	
Radius of Pulley Wheel (m):	0.019
Distance Travelled (m):	0.590
Mass per Washer (kg):	0.042

Motor A Lifting Tests

Num Washers	Travel Time (s)	Current (mA)	Voltage (V)	Speed (m/s)	w (rad/s)	w (rpm)	Mass (kg)	Force (N)	Torque (Nmm)	P_in (mW)	P_out (mW)	Efficiency (%)
0	3.49	93	4.060	0.169	9.027	86	0.000	0.000	0.000	377	0	0.00
1	3.88	92	4.050	0.152	8.112	77	0.042	0.412	7.725	374	63	16.75
2	4.18	108	3.900	0.141	7.530	72	0.084	0.824	15.451	419	116	27.74
3	4.59	125	3.710	0.128	6.851	65	0.126	1.236	23.176	464	159	34.21
4	5.15	177	3.190	0.115	6.109	58	0.168	1.648	30.902	564	189	33.45
5	5.74	183	3.120	0.103	5.483	52	0.210	2.060	38.627	570	212	37.13
6	6.87	193	3.010	0.086	4.580	44	0.252	2.472	46.352	580	212	36.62
7	7.87	222	2.720	0.075	3.998	38	0.294	2.884	54.078	604	216	35.82
8	9.54	240	2.520	0.062	3.299	32	0.336	3.296	61.803	606	204	33.65
9	11.15	282	2.100	0.053	2.823	27	0.378	3.708	69.528	591	196	33.20
10	13.70	305	1.870	0.043	2.296	22	0.420	4.120	77.254	569	177	31.15
11	19.25	339	1.530	0.031	1.634	16	0.462	4.532	84.979	518	139	26.80
12	34.88	318	1.720	0.017	0.902	9	0.504	4.944	92.705	547	84	15.28
13	N/A	489	0.000	#VALUE!	#VALUE!	#VALUE!	0.546	5.356	100.430	0	#VALUE!	#VALUE!

dark grey

Motor B Lifting Tests

Num Washers	Travel Time (s)	Current (mA)	Voltage (V)	Speed (m/s)	w (rad/s)	w (rpm)	Mass (kg)	Force (N)	Torque (Nmm)	P_in (mW)	P_out (mW)	Efficiency (%)
0	3.05	126	3.720	0.194	10.320	99	0.000	0.000	0.000	467	0	0.00
1	3.41	118	3.800	0.173	9.230	88	0.042	0.412	7.725	448	71	15.93
2	3.85	160	3.360	0.153	8.184	78	0.084	0.824	15.451	537	126	23.55
3	4.52	189	3.060	0.131	6.963	66	0.126	1.236	23.176	577	161	27.95
4	5.39	214	2.810	0.109	5.835	56	0.168	1.648	30.902	602	180	29.97
5	6.57	243	2.490	0.090	4.793	46	0.210	2.060	38.627	606	185	30.55
6	7.68	273	2.180	0.077	4.097	39	0.252	2.472	46.352	596	190	31.89
7	10.64	282	2.110	0.055	2.958	28	0.294	2.884	54.078	594	160	26.93
8	16.20	332	1.590	0.036	1.942	19	0.336	3.296	61.803	528	120	22.71
9	N/A	488	0.000	#VALUE!	#VALUE!	#VALUE!	0.378	3.708	69.528	0	#VALUE!	#VALUE!

Motor A PWM tests		
PWM setting (%)	Travel Time(s)	w (rpm)
51	54.96	5.468
102	11.53	26.059
153	7.54	39.852
204	6.25	48.116
255	4.92	61.136

Motor A Timing Error	
Trial	Time (5 washers)
1	6.763
2	6.810
3	6.850
4	6.666
5	6.876
6	6.643
7	6.876
8	6.579
9	6.694
10	6.604

*Using 255 PWM

Motor B PWM tests		
PWM setting (%)	Travel Time(s)	w (rpm)
51	21.24	14.149
102	7.93	37.909
153	5.51	54.535
204	4.57	65.721
255	3.78	79.508

Motor B Timing Error	
Trial	Time (5 washers)
1	5.989
2	5.663
3	5.770
4	5.823
5	5.952
6	5.735
7	5.661
8	5.570
9	5.807
10	5.698