Motor Testing Procedure

Purpose:

The Motor Testing Procedure allows you to measure performance parameters of the ESC, motor, and rotor, considered as a system. This document will serve as a guide to performing the required tests using the Motor Testing System.

Requirements:

- Quadcopter arm, motor, rotor, ESC, and associated accessories
- Completed torque and thrust testing stands with scale
- Completed Motor Test System with all required wiring
- A computer with a standard USB port and the Arduino IDE installed.

Safety:

THERE ARE MANY RISKS INHERENT TO ANY FORM OF MOTOR TESTING AND TO MULTICOPTERS IN GENERAL. NO GUARANTEE REGARDING THE SAFETY OF THIS EQUIPMENT OR PROCEDURE IS MADE OR IMPLIED, INCLUDING THE DESIGN AND CONSTRUCTION OF THE TEST EQUIPMENT. EXERCISE EXTREME CAUTION! **DO NOT BE FOOLED**, THE COMPONENTS USED ON RC MULIROTOR VEHICLES ARE NOT TOYS AND CAN HURT OR KILL YOU OR SOMEONE ELSE.

Switching away from red all caps for a moment to give your eyes a rest, I'll mention that we're being very serious. If you don't feel like you know what you're doing then you should stop, do some reading, post questions on some good message boards, and ask someone for help. However, if you understand the risks and feel confident that you've accounted for all realistic eventualities, we offer the following non-professional advice:

- Wear goggles. Seriously. You could lose an eye to a blade really easily if something unexpected happens.

- Stay as far away as possible from the rig while testing. Orient the rig so that you are not in the path-plane of the blades. I would also recommend not being directly in front of or behind the motor during testing. I prefer to test the motor with the rig in front of me and to the right or left, with no one else nearby during the test.
- Make sure the test equipment is well built and well secured to the testing table or bench. Be sure that the motor and arm are tightly attached to the correct mounting point.
- Do not leave the motor and ESC plugged in and unattended. Ever.
- Always be prepared to remove power from the ESC quickly. If this is difficult due to your wiring setup, CHANGE YOUR SETUP. I promise at some point you will have to quickly disconnect power to the motor, and you'll be glad you planned ahead.

A final disclaimer: we are not experts on any of this. RC aircrafts are notoriously dangerous, so if you're as new to them as we were when we began this project, you need to have a healthy fear. Obey all manufacture warnings and limitations, and again, exercise caution.

IF YOU PROCEED, IT IS AT YOUR OWN RISK.

Instructions:

Once the motor testing system is completed, the procedure for gathering data from the motor is fairly simple. Some details are left to the reader, and a bit of experimentation and careful consideration of the data will be required to achieve the best results. However, a brief overview of the process is given here to get you started. Expect that that this will be an iterative process and investigation as you gradually learn more about the performance of your vehicle.

1. **Setup**

See the associated documentation, specifically the Motor Testing System Setup guide, for details on wiring the circuit and getting everything properly connected.

Once all wiring connections are made (WITH THE BATTERY/POWER METER STILL DISCONNECTED FROM THE ESC'S), do the following:

a. Disconnect the battery from the ESC if is connected!
b. **If this is your first time using the test system, also remove the prop from the motor** (in case something unexpected happens, which is very possible, even likely).

c. Open the Arduino IDE. Connect the Uno to the computer using the USB cable. Upload the Motor Testing Program to the Uno if it is not already installed.
d. Open the serial window and select a **baud rate of 115,200**. You should see some instructions print on the screen, if not you may want to try resetting the board and make sure you have selected the correct COM port.
e. The Instructions that print look something like this:

```
Type desired motor percent throttle (must be integer between 0 and 100).
Alternatively for:
STEP RESPONSE MODE: Enter "-1" (See Instructions)
ESC THROTTLE CALIBRATION MODE: Enter "-2"
Seconds  Throttle   RPM
```

On each pass of a blade, the Arduino should print the time in seconds (time is measured from boot time of Uno) to the thousands decimal place, the current throttle setting, and the calculated RPM. The RPM value is only accurate during normal motor operation at non-zero throttle.

2. **Scale the ESC's (Normally only required once)**

NOTE: THE SCALING PROGRAM AND PROCEDURE GIVEN HERE WAS WRITTEN FOR A TURNIGY PLUSH ESC. Many ESCs have similar scaling procedures, but you **MUST** consult the manufacturer's instructions and look over the Motor Test Program code to check for compliance before proceeding. If there are differences, you will likely be able to modify the code using what we've provided as an example to meet your ESC's needs.

a. If the ESCs have not been scaled to the servo signal widths of the Arduino, enter "-2" at the top of the serial window and press enter. The following text should appear.

```
Starting Calibration Mode...
Type 1 to exit calibration mode:
FIRST, DISCONNECT ESC'S FROM POWER!! Then type -1 to continue with
calibration.
```

b. As stated**, disconnect the ESC's from battery power** (not from the Arduino signal connection). It is also best to remove the prop from the motors, or disconnect the 3 ESC motor leads from the motor in case something unexpected happens. Next type "-1". The following will appear:

```
ALL THROTTLES BEING SET TO 100%
CONNECT POWER TO ESC:
```

```
WHEN ESC EMITS TWO BEEPS, IMMEDIATELY ENTER ANYTHING (E.G. 0) TO SET
MOTORS TO ZERO
```

c. **This next part must be executed quickly.** Generally, it is wise to **type a character such as "0" into the serial window but do not send** (don't hit enter), and only ***then*** plug the power into the ESC. It will emit two beeps. **As soon as both beeps sound, immediately hit enter** to send the character to the Uno. When this is performed, the following text will appear:

```
Motor Speed set to zero
ESC's should now have throttle calibrated for Arduino signals. Returning
to regular operation...
```

d. If the steps were performed correctly, the ESC should now recognize the scaling of Arduino throttle commands correctly. If you are in doubt about the success of your attempt, simply repeat the scaling process till you are confident it has worked. This type of scaling is vital not just for testing, but for any other customized control system you intend to connect to your quadcopter. Note that each ESC must be scaled separately. When in doubt, its best to just scale all of them.

3. **Throttle Response Testing (Normal Mode, Applicable to both torque and thrust testing)**
   a. Create an Excel spreadsheet (recommended) or a table in a notebook to record test results. You can obviously record any information of interest to your project, but at a minimum you will generally want the following:
      i. <u>Thrust Testing:</u>
         1. "**h**": height from the pin joint to the center of the motor (in.)
         2. "**b**": length from the pin joint to the contact point on the scale (in.)
         3. **Scale measurement vs. throttle setting**: As the test proceeds, you will need to record the scale measurement for each throttle setting you are interested in. This <u>measurement should be made in grams</u>, or converted to grams if necessary.
         4. Voltage, Amperage, Power, etc. (optional): depending on your setup, having this power supply information saved with the data might prove useful for later analysis. This information is not needed for our

parameter fitting however. Monitor Battery voltage during the test to avoid ruining your battery!

ii. Torque Testing:

1. "**L**": distance from the test rig arm's axis of rotation (which should be coincident with the rotational axis of the motor), to the axis of the rod serving as the point of contact with the scale (see torque testing rig documentation for more detail) (in.).
2. **Scale measurement vs. throttle setting**: As the test proceeds, you will need to record the output from the scale for each throttle setting you are interested in. This measurement should be made in grams, or converted to grams if necessary.
3. Voltage, Amperage, Power, etc. (optional): depending on your setup, having this power supply information saved with the data might prove useful for later analysis. This information is not needed for our parameter fitting however. Monitor Battery voltage during the test to avoid ruining your battery!

b. **If this is your first time running through a test, you'll first want to test that the motor responds normally without the blades attached**. Make sure everything is plugged in, and enter a desired throttle % value of 1.

c. This will likely result in no movement! That's normal. Increase the throttle setting one integer at a time until the motor begins to spin. Record this value as your "***Minimum throttle***" or something similar, this will be used when you create the Simulation Model.

d. Try a higher percentage throttle or two (10% to 20%) and make sure the response seems normal. If everything appears safe, stop the motor and disconnect the ESC. SECURELY reinstall the rotor on the motor (if too loose they WILL come off during testing).

e. You are now ready to do a test! At this point you know very little about how your motor will behave. Thus the process should be fairly iterative. I'll leave it to you to determine the wisest course of action, but here's what we did:

i. Run through a few sample throttle values, in steps of 5% or so, collecting thrust data. Stop at a value that still seems safe (no crazy vibrations, overly high currents or sagging battery voltage, etc.) perhaps 40%.
ii. Save this data as described in the Importing Data to MATLAB document.
iii. Analyze the data as described in the Data Analysis Program Document. Draw some conclusions from the data; does everything seem reasonable to you? According to your calculated C_T, C_R, and b values, what throttle setting is necessary to obtain hover performance (hint, you'll need a little algebra with the C_T and C_R equations and a rough estimate of total vehicle weight). Assuming you get a sane result (e.g. perhaps 35-60% throttle, ours was ≈ 38%), this is probably the throttle setting you are most interesting in, since most flight maneuvering will occur at around hover power.

f. Continue testing with the aim of obtaining consistent results. Note that as the battery voltage decreases, the test results will strongly reflect this in a number of ways. We chose to always use test results taken from a freshly charged battery for finding our parameters, though choosing to use test results from some middle range of battery charge levels may be preferable to you. Gather some data and make your own decision. If you want to go deeper you could even try to find a functional relation between your coefficients and voltage (i.e. $C_T(V)$) but we skipped this as it has limited value for our application and this type of feature is not yet incorporated into the simulation.
g. What is the maximum safe throttle? Honestly, I don't know and it will depend on your motor, prop, and ESC. My advice is to be very careful. We once took our motor up to 78% throttle, and it suddenly started shaking so bad I thought it was going to break the rig or shatter the rotor. Not fun. Once you know the hover throttle setting, perhaps go a bit above this by 15% to 20% to feel out performance at that higher range, but be careful and don't be reckless. You could damage the motor, rotor, ESC, battery, or yourself!

4. Step Input Testing

a. Step input testing uses a specialized program accessed through the normal mode screen by typing "-1". **<u>Step input testing should not be performed until safe and</u>**

consistent operational limits have been established for normal motor throttle range.

b. Create an Excel spreadsheet (recommended) or a table in a notebook to record test results. You can obviously record any information of interest to your project, but at a minimum you will generally want the following:
    i. Step Input Testing:
        1. Voltage, Amperage, Power, etc. (optional): depending on your setup, having this power supply information saved with the data might prove useful for later analysis. This information is not needed for our parameter fitting, however. Monitor Battery voltage during the test to avoid ruining your expensive battery!

c. Upon entering Step Input Mode the following will be displayed:

```
Entering Step input mode...
Step mode is for measuring the step response of the motor.
Type the starting percent throttle:
```

d. Enter in a starting throttle (e.g. “25”%). Note: step test should always go from a lower number to a higher number since the analysis code does not yet support downward steps. Feel free to add support for this though if you think it is of value to your project.

e. You will then be asked for the percent throttle you wish to step to. Enter this value in (e.g. “35”%).

f. A confirmation dialog will display the following:

```
Initial Throttle: 25    Step Throttle: 35
Motor Speed will be stepped low for 5 seconds then stepped high for 5
seconds.
If these values are acceptable, type "1". To abort Step Input Mode,
type "-1"
```

g. If you want to change the duration of the low and high tests take a look in the Arduino code. It should be easy to find which variables to adjust.

h. If you entered a value incorrectly, type “-1”. Otherwise, type “1” when you are ready to begin the test. The following will be displayed if you continue:

```
Step Mode Beginning In 5 Seconds!
```

If you need to abort, pull the power from the motor, then from the Uno.

i. The test will begin as described in the text output above. ONCE THE TEST BEGINS, IT CANNOT BE ABORTED via a serial window command. If something goes wrong, YOU WILL NEED TO REMOVE THE POWER SUPPLY FROM THE ESC/MOTOR.

j. That said, as long as everything proceeded normally, you just sit back and watch as the test is performed. When the test is done the motor is set to 0% throttle and the following message is displayed:

```
Step input test complete. Returning to normal operation...
```

k. Save the data as described in the Importing Data to MATLAB document.

l. What throttle values should you use for your step tests? It's really up to you. We used tests from just below, to just above hover throttle, for example: 35% to 45% or 37 to 39% (our hover throttle is about 38%). These results were compared to tests representing a number of other ranges, and the results were relatively consistent between trials and throttle settings used. Note that the first order model we use is only a rough approximation of actual RPM behavior, so some error and variation between tests is expected. That said, the effect of this first order delay on control system performance and stability is not insignificant, so try to obtain reasonable data you think represents your motor.