

Data Analysis Program

Purpose:

The simulation requires us to estimate several quadcopter performance parameters that are acquired through experimental testing of the motors of the quadcopter vehicle. This GUI was made to help analyze this motor test data. The parameters this GUI is able to calculate are called: CR and b (**Throttle to RPM conversion**), CT (**Thrust coefficient**), CQ (**Torque coefficient**), and TC (**motor time constant**). By loading test data into the workspace the GUI can automatically calculate parameters and plot the data. The Data Analysis GUI can be seen below in Figure 1.

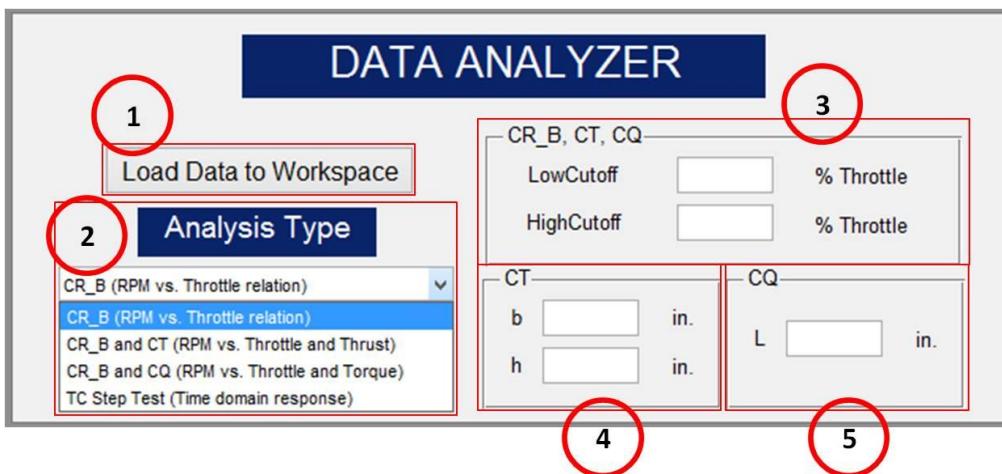


Figure 1. Data Analysis Program

Description:

1. Load button: this button loads in structures of data that were previously saved. The data loaded should **correspond** to the analysis to be performed (i.e. don't try to use the TC Step Test analysis on Throttle Response Data!)
2. Analysis Type – there are 4 options accessible through the **drop down box**:
 - o **CR_B (RPM vs. Throttle relation)**

Required User Input: Seconds, Throttle, and RPM data
Optional User Input: Low Cutoff, High Cutoff

- i. This type of analysis requires experimental data input of **Seconds**, **Throttle**, and **RPM**. It is the most general type of analysis this

interface can perform. No raw scale readings of thrust or torque are needed. Once the “Run” button is selected, the program will analyze the **RPM** vs. **Throttle** data to obtain a fit that calculates the C_R and b parameters (linear relation with non-zero intercept between throttle percentage and RPM).

- ii. The **LowCutoff** and **HighCutoff** for throttle percentage can also be included in this program. These values are entered as shown on figure 1 above. For example, if a test is run between throttle limits of 0 and 60%, you may still only want the program to analyze the data between values of 20 and 40% throttle. This is an important concept since having the fit be accurate around the anticipated normal operating range is often more important than having the fit be based on the full possible range of RPM values. Therefore, we recommend some iteration involving finding the approximate throttle required for hover and then performing additional testing and parameter fits around this range (+/- 5-10% throttle for example). This will produce more accurate simulation results, and presumably better control design, at near hover conditions, which are usually the conditions of primary interest. If these values aren't entered, the program will automatically find the lowest (non-zero) and highest throttle values from the test data and use those instead.
- iii. See the function “calculate_CR_B.m” for more details. It is strongly recommended that users spend some time understanding the function since the approach utilized might not be ideal for all users.

- **CR_B and CT (RPM vs. Throttle and Thrust)**

Required User Input: Seconds, Throttle, and RPM data
gramsMeas (raw scale), b, h

Optional User Input: Low Cutoff, High Cutoff

- i. This type of analysis requires data inputs of **Seconds**, **Throttle**, **RPM**, and grams measured (**gramsMeas**: raw scale readings from thrust test). The analysis also requires inputs for the **dimensions** of the thrust test rig, namely: **b** and **h**. “b” is the distance from the **pin** joint on the **rig** to the point of contact with the scale. “h” is the height of the center motor **axis** above **the pin joint on the rig**.
 - ii. NOTE: If the “Load Data to Workspace” button is used but the only data saved is **Seconds**, **Throttle**, and **RPM**, the **gramsMeas** (raw scale Thrust values) parameter must be added by another means, such as direct **vector** entry at the MATLAB command window or copy paste and import from **a spreadsheet**.
 - iii. As with the CR_b analysis, this type of analysis will also accept user inputs of low and high **cutoff** for Throttle %. See discussion above.
 - iv. Once the data is loaded and the user inputs filled in, this program once again calculates the C_R and b parameters along with the C_T **coefficient** (Thrust coefficient). C_T is a relation between Thrust and RPM².
 - v. See the function “calculate_CT.m” for more details. It is strongly recommended that users spend some time understanding the function since the approach utilized might not be ideal for all users.
- **CR_B and C₀ (RPM vs. Throttle and Torque)**

Required User Input: **Seconds**, **Throttle**, and **RPM data**
gramsMeas (raw scale), **L**

Optional User Input: **Low Cutoff**, **High Cutoff**

- i. This type of analysis requires data inputs of **Seconds**, **Throttle**, **RPM**, and grams measured (**gramsMeas**: raw scale readings from torque test). The analysis also requires an input for a **Torque rig dimension**, **L**. “L” is the **perpendicular** distance between the axis of the motor and the axis of the arm that is in contact with the scale.

- ii. NOTE: If the “Load Data to Workspace” button is used but the only data saved is **Seconds**, **Throttle**, and **RPM**, the **gramsMeas** (raw scale force values in grams) parameter must be added by another means, such as **direct vector entry** at the MATLAB command window.
- iii. As with the C_R and b analysis, this type of analysis will also accept user inputs of low and high cutoffs for Throttle %.
- iv. Once the data is loaded and the user input filled in, this program once again calculates the C_R and b parameters along with the C_Q coefficient (Torque coefficient). C_Q is a relation between Torque and RPM².
- v. See the function “calculate_CQ.m” for more details. It is strongly recommended that users spend some time understanding the function since the approach utilized might not be ideal for all users.

- **TC Step Test (Time domain response)**

Required User Input: **Seconds, Throttle, and RPM data**

Optional User Input: **N/A**

- i. This type of analysis requires data inputs of **Seconds**, **Throttle**, and **RPM**. The program calculates the time constant (TC) of the motor from a **step change** in throttle input.
- ii. The program does not utilize the low and high cutoffs for Throttle %, because it automatically recognizes the two throttle settings that the **step command** is operating between.
- iii. See the function “calculate_TC.m” for more details. It is strongly recommended that users spend some time understanding the function since the approach utilized might not be ideal for all users.