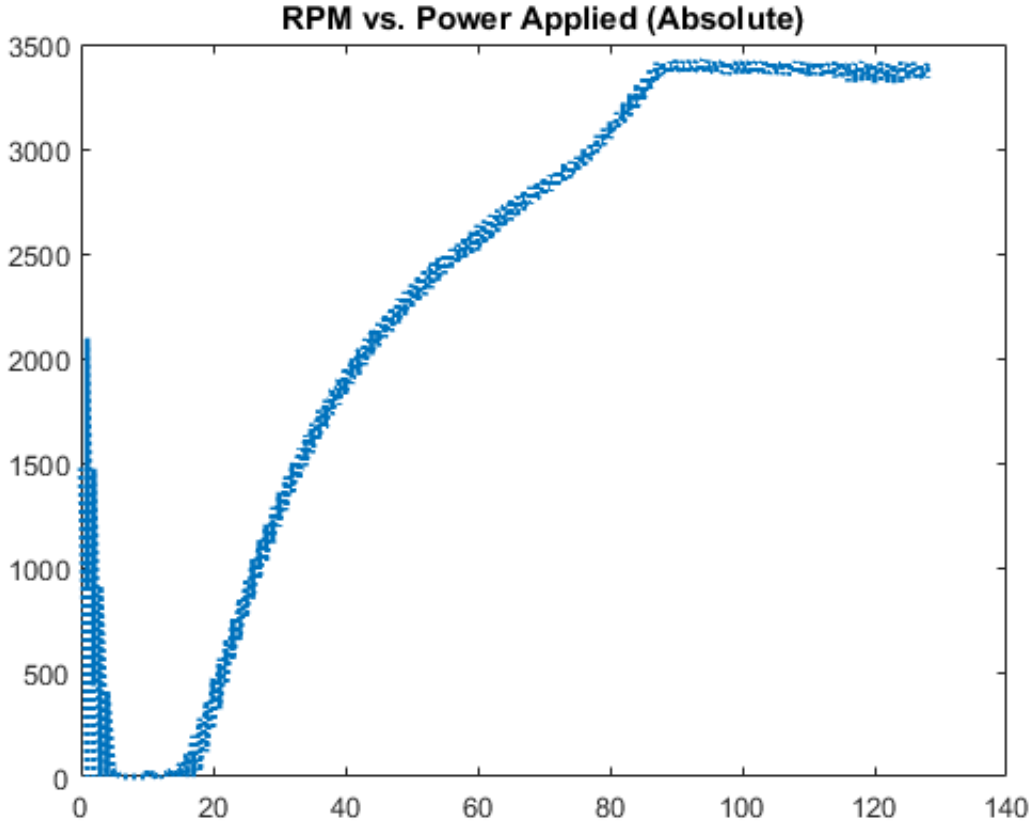
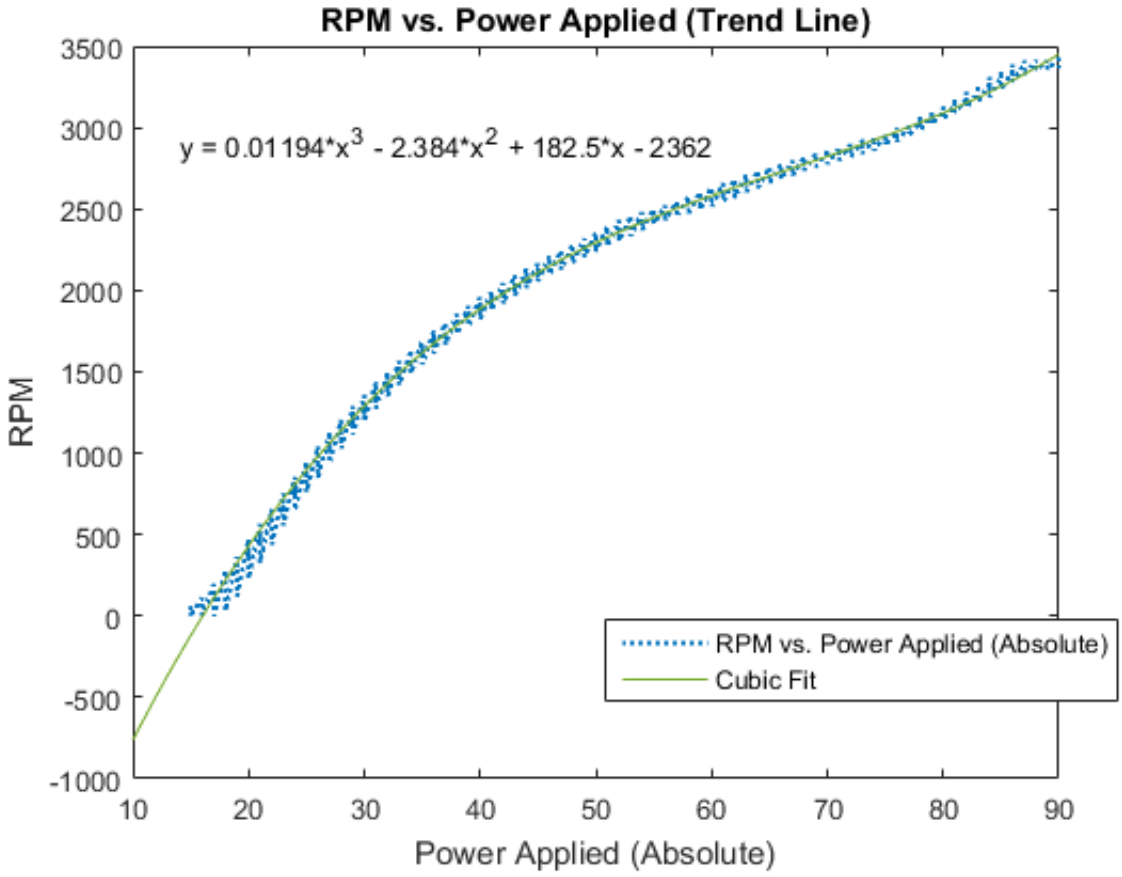
Single Flywheel Encoder Data Analysis

```
% Housekeeping
format shortg;

% Sort the data by PowerApplied and then delegate it to sVariables. Also,
% convert the values to absolute value so the data is universally
% meaningful and included in the equation.
data = sortrows([abs(PowerApplied), abs(EncoderCount), abs(RPM)], 1);
sRPM = data(:,3);
sEncoderCount = data(:,2);
sPowerApplied = data(:,1);
sdata = [sPowerApplied, sEncoderCount, sRPM];

% Plot the intermediate results with all of the data. The spotty data
% around 0 power applied is due to the residual wheel spin during testing.
% Ignore it.
PlotAbsoluteData(sPowerApplied, sRPM);

% Filter out the values outside the analytical range (~15 to ~90).
% Everything outside this range is either zero or maximum speed so do not
% include them in the calculations.
fdataindexes = sdata(sdata(:,1)>14 & sdata(:,1)<91,1);
fdatavalues = sdata(sdata(:,1)>14 & sdata(:,1)<91,3);
fdata = [fdataindexes, fdatavalues];

% Then, output the filtered plot within the analytical range. Develop a
% trendline using the data.
FilteredPlot(fdata(:,1), fdata(:,2));
```


RPM vs. Power Applied (Absolute)
3500
3000
2500
2000
1500
1000
500
0
0
20
40
60
80
100
120
140


RPM vs. Power Applied (Trend Line)
y = 0.01194*x^3 - 2.384*x^2 + 182.5*x - 2362
RPM
Power Applied (Absolute)
RPM vs. Power Applied (Absolute)
Cubic Fit
3500
3000
2500
2000
1500
1000
500
0
-500
-1000
10
20
30
40
50
60
70
80
90

Published with MATLAB® R2015b