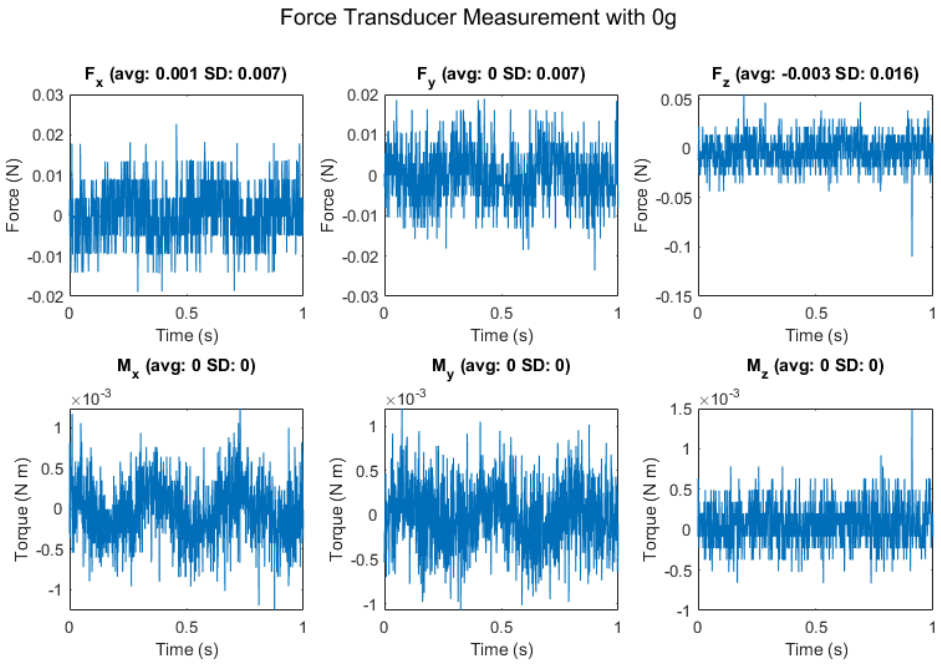
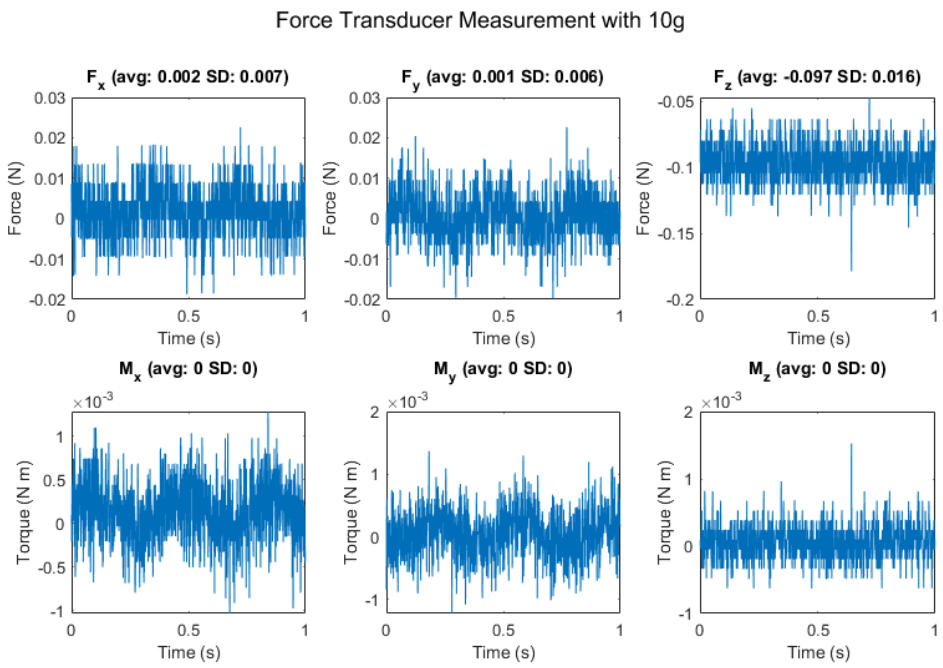
```
clear
close all

% This file is used to produce plots from the force transducer data
 saved
% in xlsx files
files =
 ["0g_mass.xlsx" "10g_mass.xlsx" "50g_mass.xlsx" "100g_mass.xlsx" "200g_mass.xlsx"

for i = 1:length(files)

    % Get mass from filename
    mass = erase(files(i), "_mass.xlsx");
    force = (str2num(mass{1}(1:end-1)) / 1000) * 9.81;

    % Get data from file
    data = readmatrix(files(i));

    these_raw_times = data(4:end,7);
    force_vals = data(4:end,1:6);
    force_means = round(mean(force_vals), 3);
    force_SDs = round(std(force_vals), 3);

    % Open a new figure.
    f = figure;
    f.Position = [200 50 900 560];

    % Create three subplots to show the force time histories.
    subplot(2, 3, 1);
    plot(these_raw_times, force_vals(:, 1));
    title("F_x (avg: " + force_means(1) + " SD: " + force_SDs(1)
 + ")");
    xlabel("Time (s)");
    ylabel("Force (N)");
    subplot(2, 3, 2);
    plot(these_raw_times, force_vals(:, 2));
    title("F_y (avg: " + force_means(2) + " SD: " + force_SDs(2)
 + ")");
    xlabel("Time (s)");
    ylabel("Force (N)");
    subplot(2, 3, 3);
    plot(these_raw_times, force_vals(:, 3));
    title("F_z (avg: " + force_means(3) + " SD: " + force_SDs(3)
 + ")");
    xlabel("Time (s)");
    ylabel("Force (N)");

    % Create three subplots to show the moment time histories.
    subplot(2, 3, 4);
    plot(these_raw_times, force_vals(:, 4));
    title({"M_x (avg: " + force_means(4) + " SD: " + force_SDs(4)
 + ")" ""});
```

```
    xlabel("Time (s)");
    ylabel("Torque (N m)");
    subplot(2, 3, 5);
    plot(these_raw_times, force_vals(:, 5));
    title({"M_y (avg: " + force_means(5) + " SD: " + force_SDs(5)
+ ")" ""});
    xlabel("Time (s)");
    ylabel("Torque (N m)");
    subplot(2, 3, 6);
    plot(these_raw_times, force_vals(:, 6));
    title({"M_z (avg: " + force_means(6) + " SD: " + force_SDs(6)
+ ")" ""});
    xlabel("Time (s)");
    ylabel("Torque (N m)");

    % Label the whole figure.
    sgtitle("Force Transducer Measurement with " + mass);

    % Calculate percent error using the z-component of the force
(gravity)
    % and the known weight of the mass. (z-comp is negative)
    Absolute_Error = (force_means(3) + force);
    Percent_Error = round(((force_means(3) + force) / force) * 100,
3);
    disp('For ' + mass + ' the absolute error was ' +
Absolute_Error ...
    + 'N and the percent error was ' + Percent_Error + '% (Calculated
from F_z)');
```

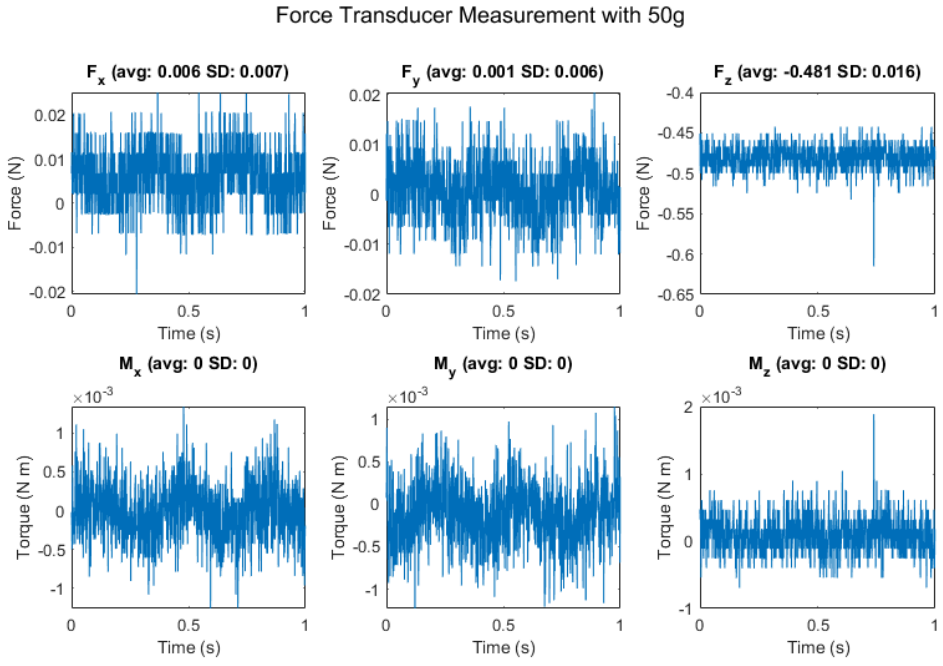
For 0g the absolute error was -0.003N and the percent error was -Inf% (Calculated from F_z)



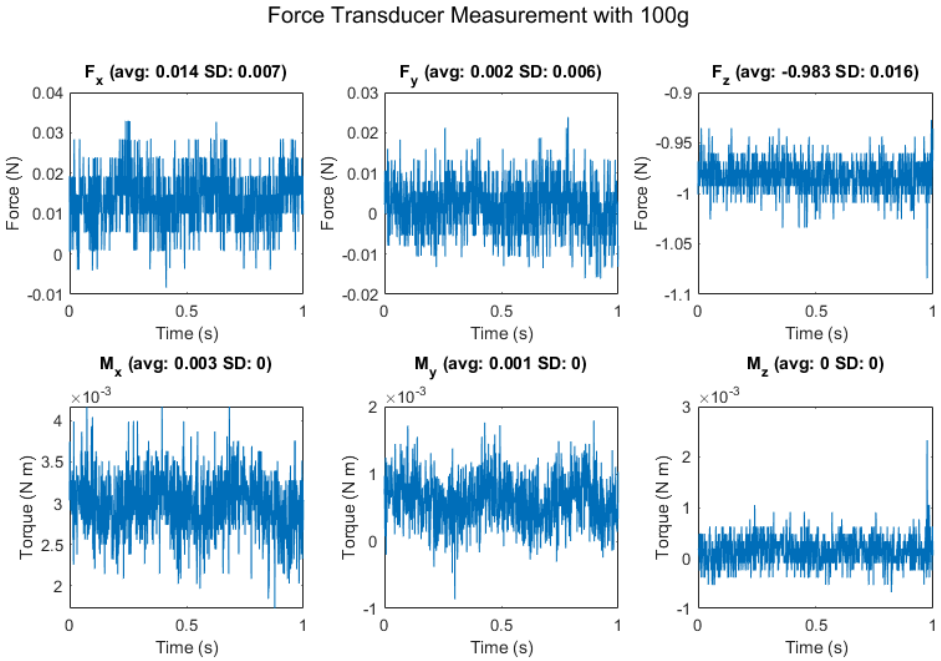
For 10g the absolute error was 0.0011N and the percent error was 1.121% (Calculated from F_z)



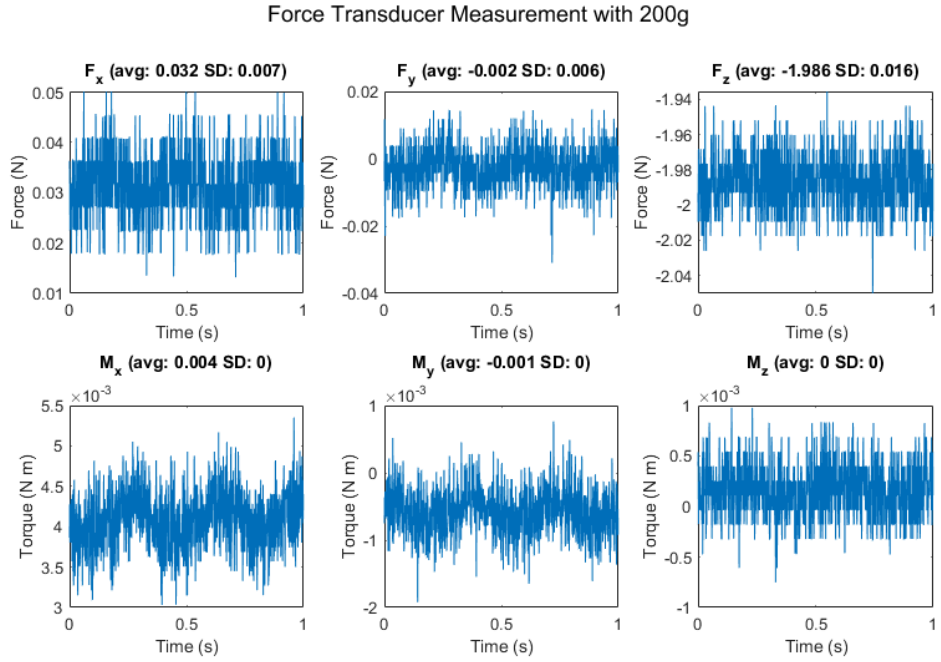
For 50g the absolute error was 0.0095N and the percent error was 1.937% (Calculated from F_z)



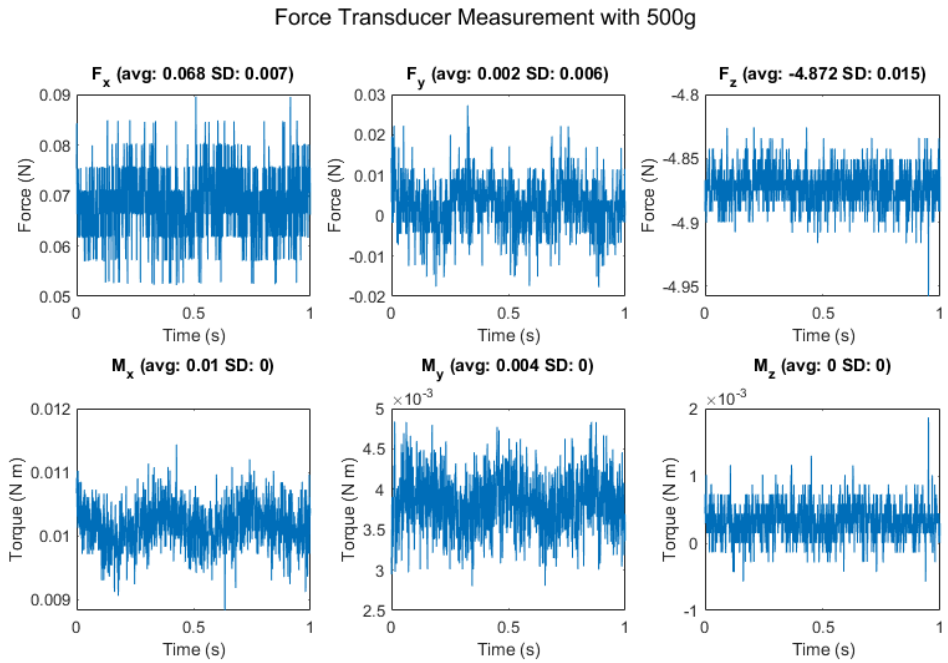
For 100g the absolute error was -0.002N and the percent error was -0.204% (Calculated from F_z)



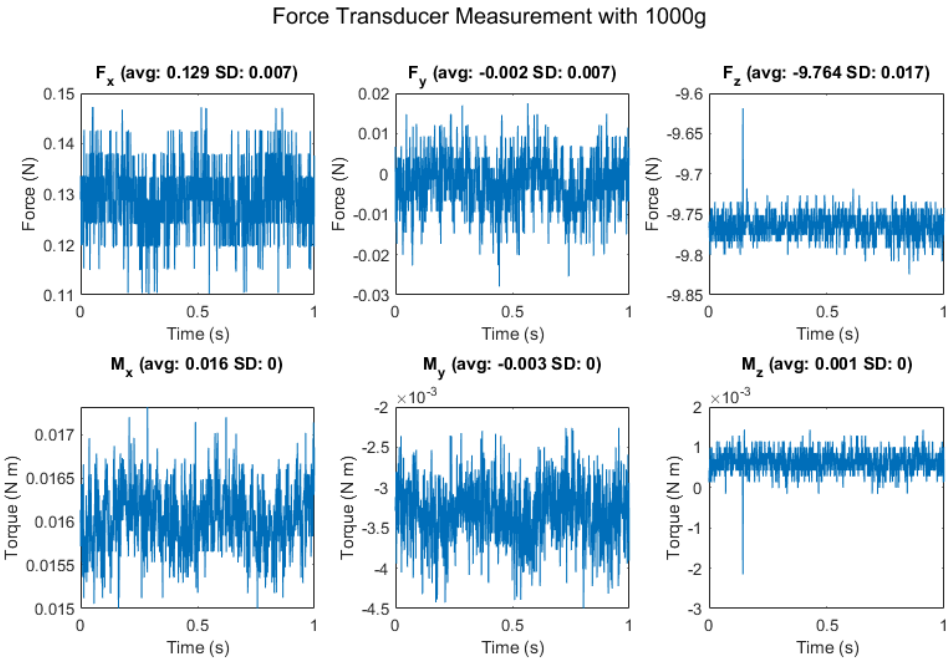
For 200g the absolute error was -0.024N and the percent error was -1.223% (Calculated from F_z)



For 500g the absolute error was 0.033N and the percent error was 0.673% (Calculated from F_z)



For 1000g the absolute error was 0.046N and the percent error was 0.469% (Calculated from F_z)



end

Published with MATLAB® R2020b