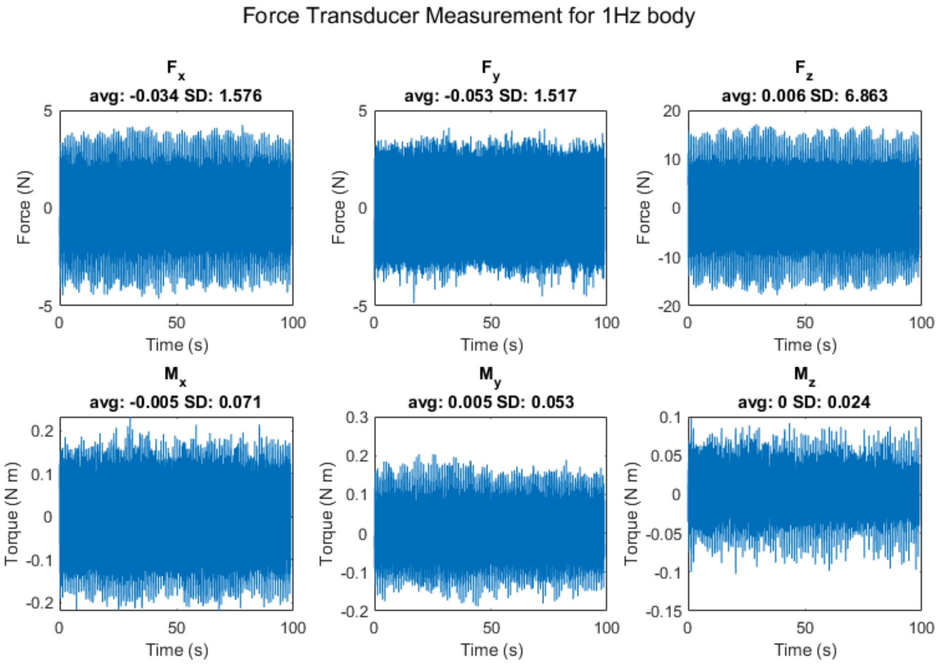
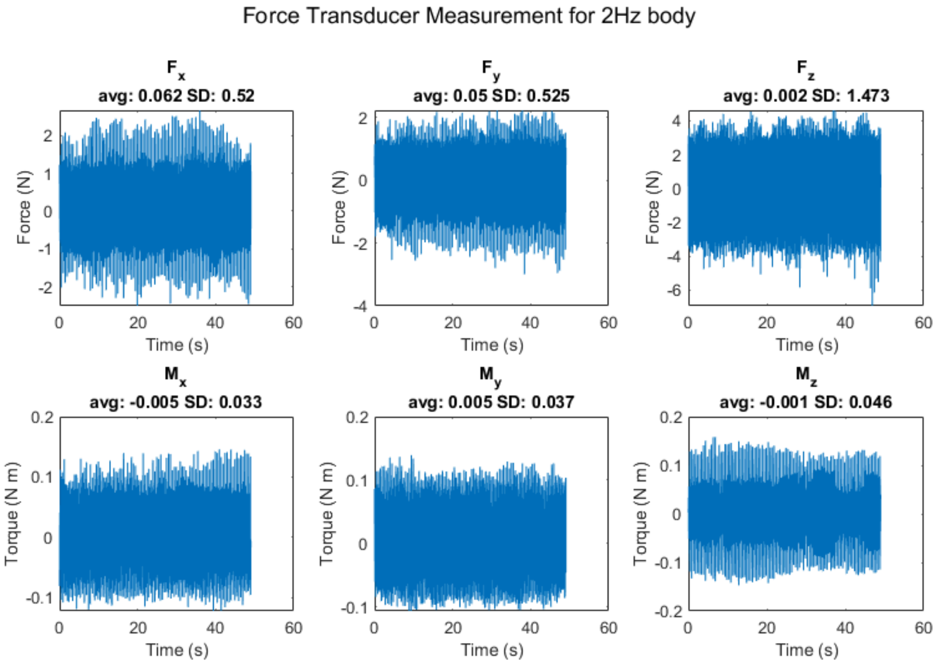
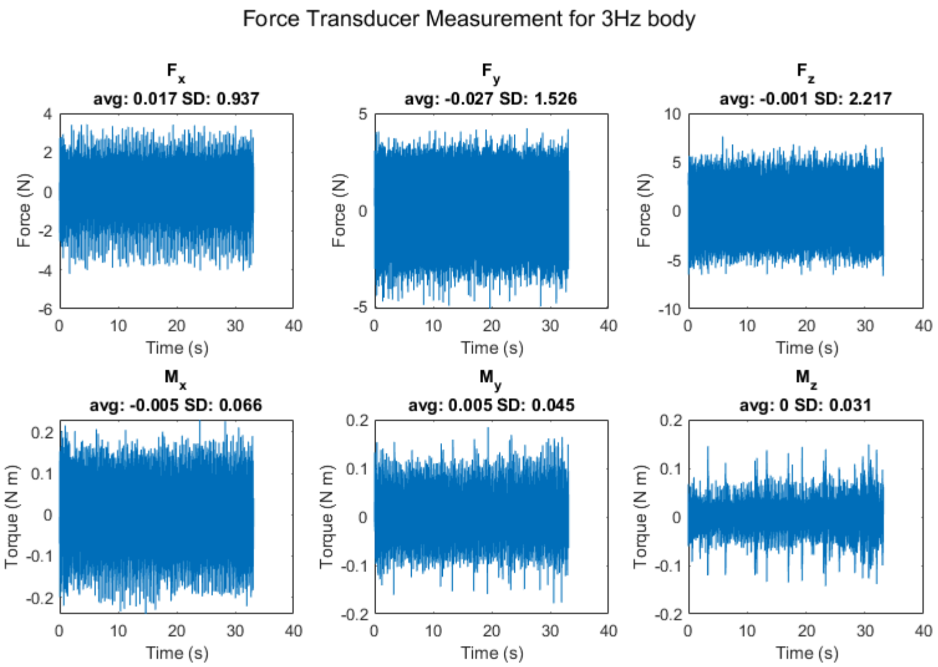
```
clear
close all

% Ronan Gissler January 2023

% This file is used to analyze the data from the experiments Sakthi
% and I ran with the 1 DOF flapper robot with and without
% Polydimethylsiloxane (PDMS) wings on January 19th 2023. We tested
% flapping speeds between 1 Hz and 4 Hz with the PDMS wings, at 4 Hz
% the whole system was shaking and grinding loudly. We test flapping
% speeds between 1 Hz and 6 Hz with no wings attached, at 6 Hz the
% whole system was shaking and grinding loudly (although less
% dramatically than at 4 Hz with the wings attached).

% ----------------------------------------------------------------
% ------------------------Plot All Data---------------------------
% ----------------------------------------------------------------

files = ["1Hz_body_experiment_011923.csv"
         "2Hz_body_experiment_011923.csv"
         "3Hz_body_experiment_011923.csv"
         "4Hz_body_experiment_011923.csv"
         "5Hz_body_experiment_011923.csv"
         "6Hz_body_experiment_011923.csv"
         "1Hz_PDMS_experiment_011923.csv"
         "2Hz_PDMS_experiment_011923.csv"
         "3Hz_PDMS_experiment_011923.csv"
         "4Hz_PDMS_experiment_011923.csv"];

for i = 1:length(files)
    % Get case name from file name
    case_name = erase(files(i), "_experiment_011923.csv");
    case_name = strrep(case_name,'_',' ');

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

    times = data(1:end,1);
    force_vals = data(1:end,2:7);

    % Trimming off end of data (it appears beginning is already
    % trimmed)
    count = 0;
    vertical_diffs = diff(force_vals(:,3));
    for j = 1:length(vertical_diffs)
        if (abs(vertical_diffs(j)) < 0.05)
            count = count + 1;
        else
            count = 0;
        end
        if (count > 5)
            data = data(1:j-1000, :);
```

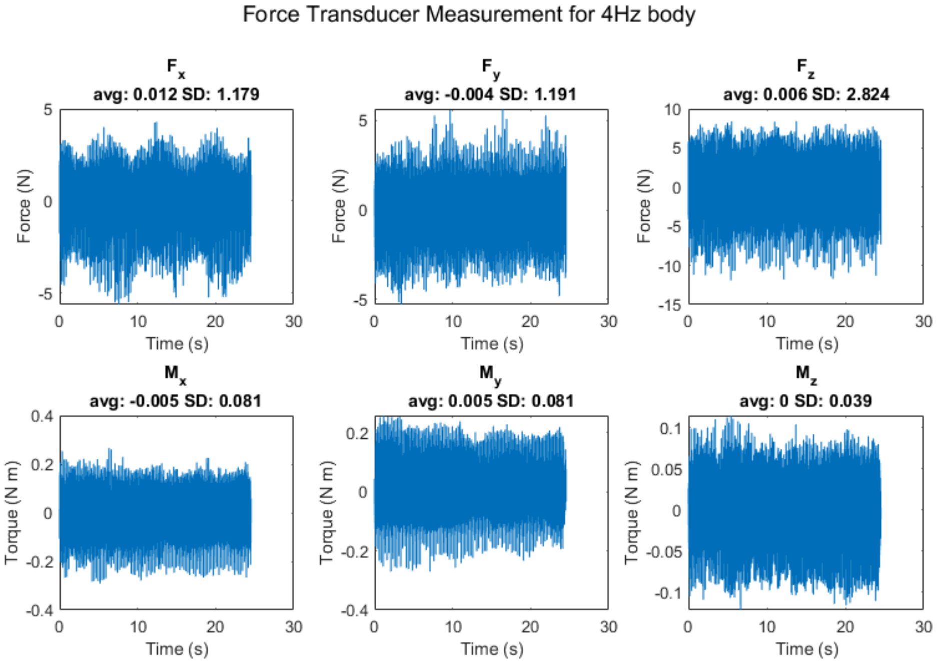
```
           break
       end
   end

   times = data(1:end,1);
   force_vals = data(1:end,2:7);

   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(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(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(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(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(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(times, force_vals(:, 6));
   title(["M_z" ("avg: " + force_means(6) + " SD: " +
force_SDs(6))]);
   xlabel("Time (s)");
   ylabel("Torque (N m)");
```

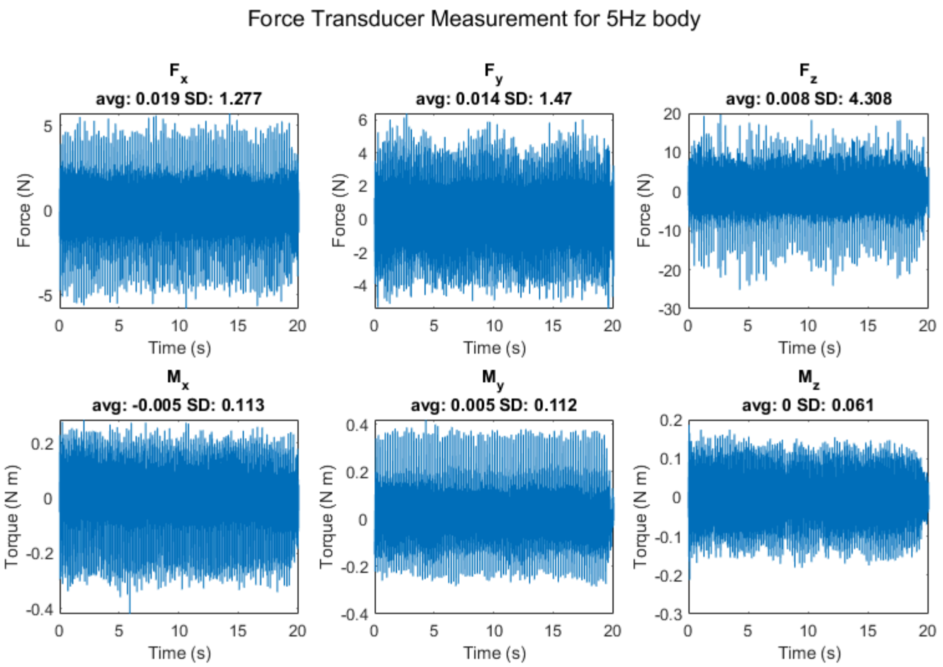
```
    % Label the whole figure.
    sgtitle("Force Transducer Measurement for " + case_name);

    case_parts = char(split(case_name));
    save([case_parts(2,:),'_',case_parts(1,1:end-1),'.mat'], 'data')
end
```

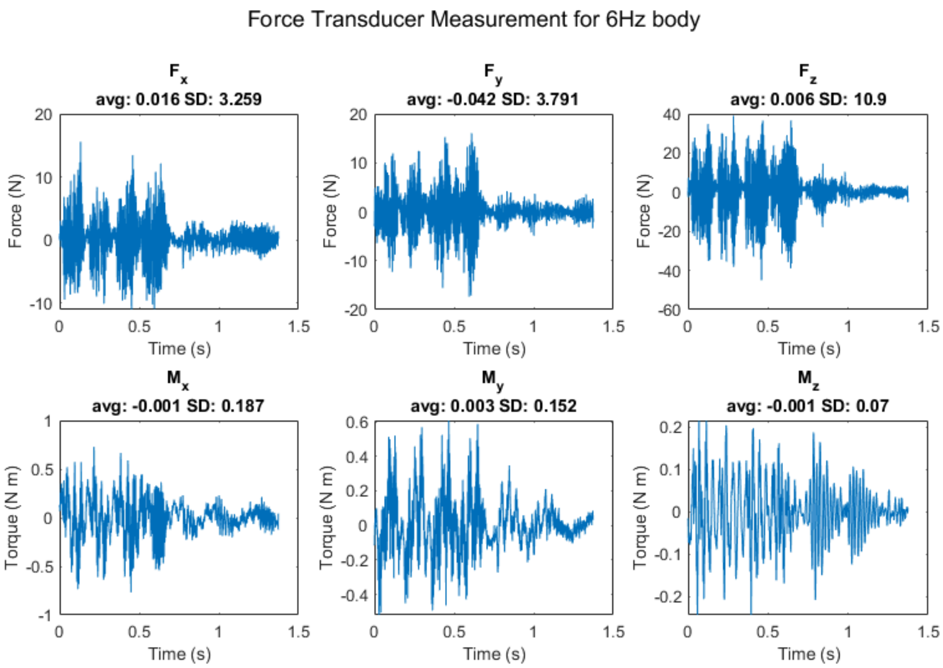


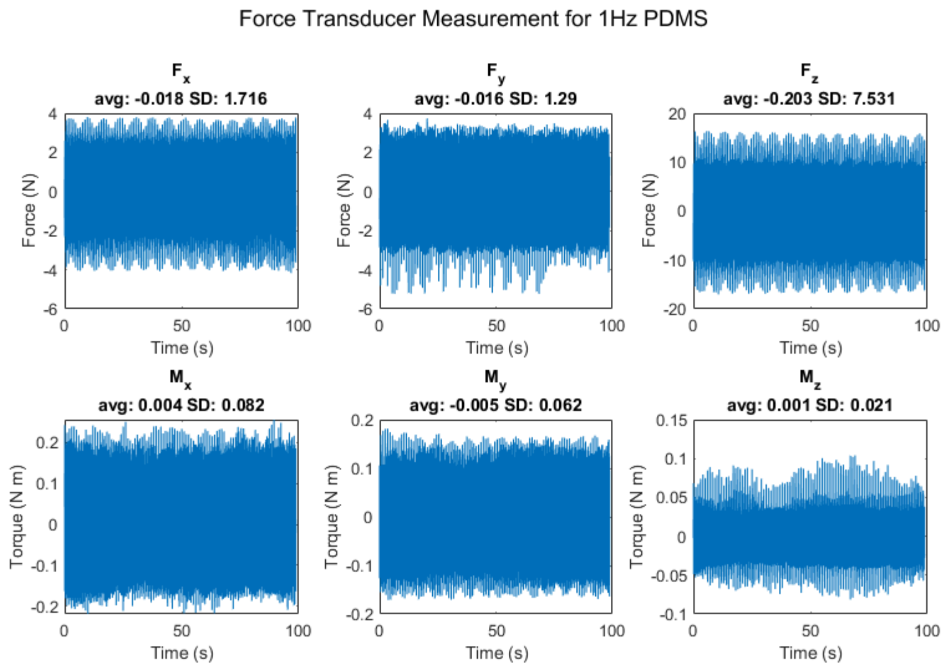


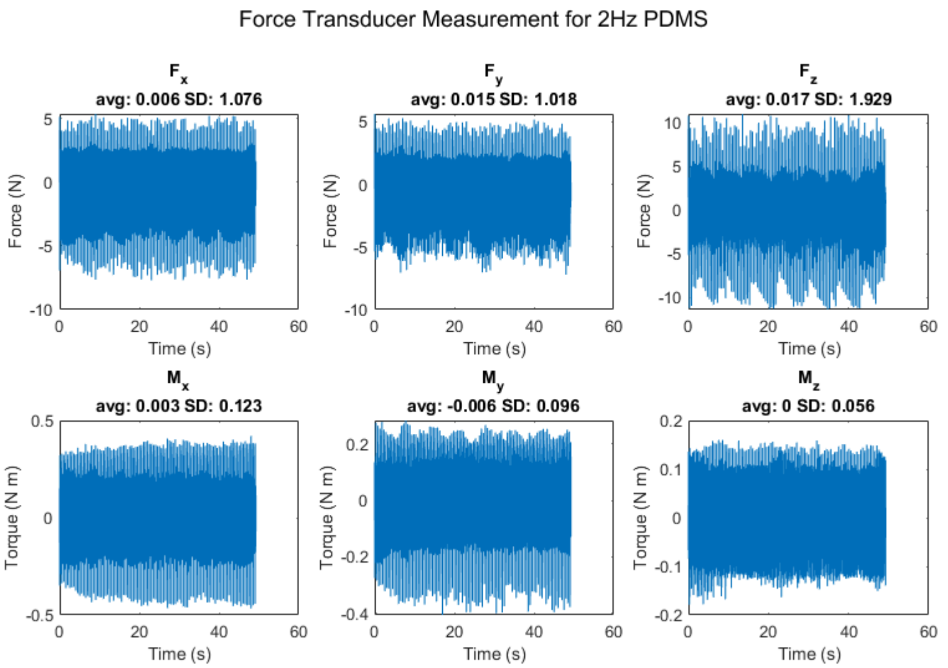

Force Transducer Measurement for 3Hz body
F_x
avg: 0.017 SD: 0.937
F_y
avg: -0.027 SD: 1.526
F_z
avg: -0.001 SD: 2.217
Force (N)
Time (s)
M_x
avg: -0.005 SD: 0.066
M_y
avg: 0.005 SD: 0.045
M_z
avg: 0 SD: 0.031
Torque (N m)
Time (s)
0
10
20
30
40

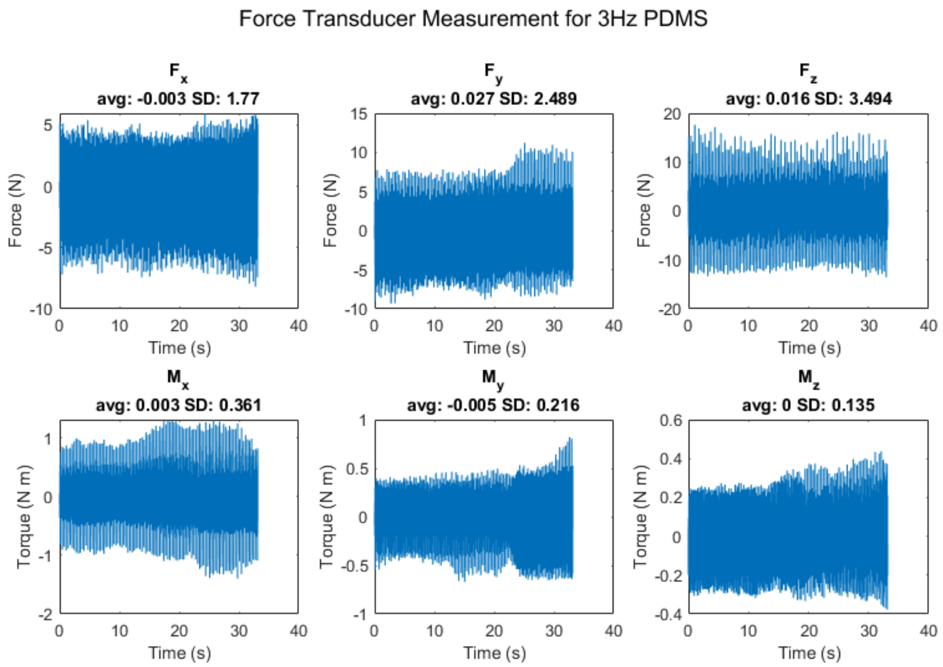

Force Transducer Measurement for 4Hz body
F_x
avg: 0.012 SD: 1.179
F_y
avg: -0.004 SD: 1.191
F_z
avg: 0.006 SD: 2.824
Force (N)
Time (s)
M_x
avg: -0.005 SD: 0.081
M_y
avg: 0.005 SD: 0.081
M_z
avg: 0 SD: 0.039
Torque (N m)
Time (s)
0
10
20
30

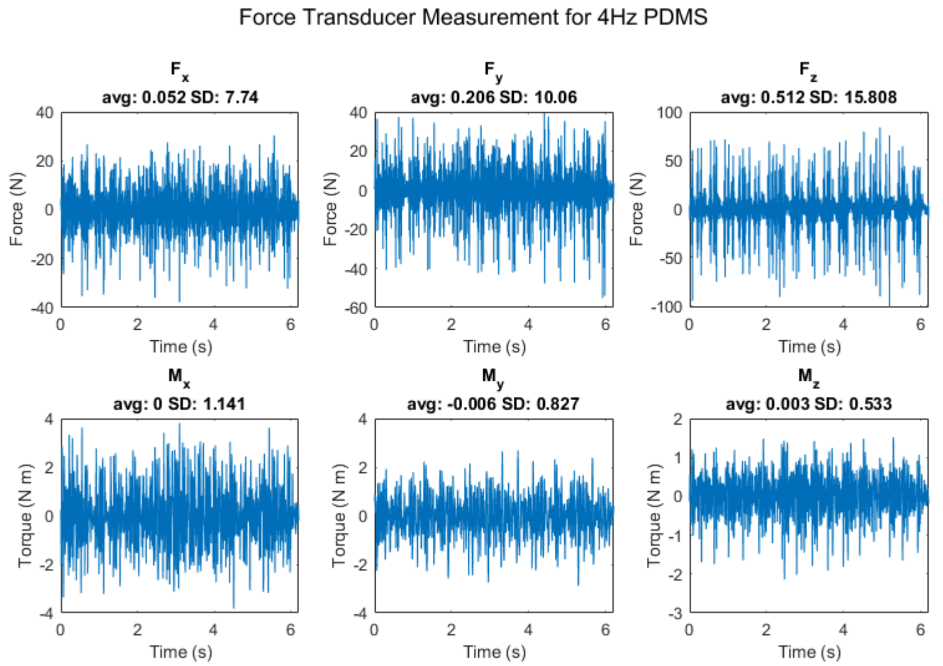

Force Transducer Measurement for 5Hz body
F$_x$
avg: 0.019 SD: 1.277
F$_y$
avg: 0.014 SD: 1.47
F$_z$
avg: 0.008 SD: 4.308
Force (N)
5
0
-5
6
4
2
0
-2
-4
20
10
0
-10
-20
-30
0
5
10
15
20
Time (s)
M$_x$
avg: -0.005 SD: 0.113
M$_y$
avg: 0.005 SD: 0.112
M$_z$
avg: 0 SD: 0.061
Torque (N m)
0.2
0
-0.2
-0.4
0.4
0.2
0
-0.2
-0.4
0.2
0.1
0
-0.1
-0.2
-0.3
Time (s)


Force Transducer Measurement for 6Hz body
F$_x$
avg: 0.016 SD: 3.259
F$_y$
avg: -0.042 SD: 3.791
F$_z$
avg: 0.006 SD: 10.9
Force (N)
20
10
0
-10
20
10
0
-10
-20
40
20
0
-20
-40
-60
0
0.5
1
1.5
Time (s)
M$_x$
avg: -0.001 SD: 0.187
M$_y$
avg: 0.003 SD: 0.152
M$_z$
avg: -0.001 SD: 0.07
Torque (N m)
1
0.5
0
-0.5
-1
0.6
0.4
0.2
0
-0.2
-0.4
0.2
0.1
0
-0.1
-0.2
Time (s)


Force Transducer Measurement for 1Hz PDMS
F$_x$
avg: -0.018 SD: 1.716
F$_y$
avg: -0.016 SD: 1.29
F$_z$
avg: -0.203 SD: 7.531
M$_x$
avg: 0.004 SD: 0.082
M$_y$
avg: -0.005 SD: 0.062
M$_z$
avg: 0.001 SD: 0.021
Force (N)
Torque (N m)
Time (s)
4
2
0
-2
-4
-6
20
10
-10
-20
0.2
0.1
-0.1
-0.2
0.15
0.05
-0.05
50
100


Force Transducer Measurement for 2Hz PDMS
F$_x$
avg: 0.006 SD: 1.076
F$_y$
avg: 0.015 SD: 1.018
F$_z$
avg: 0.017 SD: 1.929
M$_x$
avg: 0.003 SD: 0.123
M$_y$
avg: -0.006 SD: 0.096
M$_z$
avg: 0 SD: 0.056
Force (N)
Torque (N m)
Time (s)
10
5
0
-5
-10
0.5
-0.5
0.2
-0.2
-0.4
0.1
-0.1
20
40
60

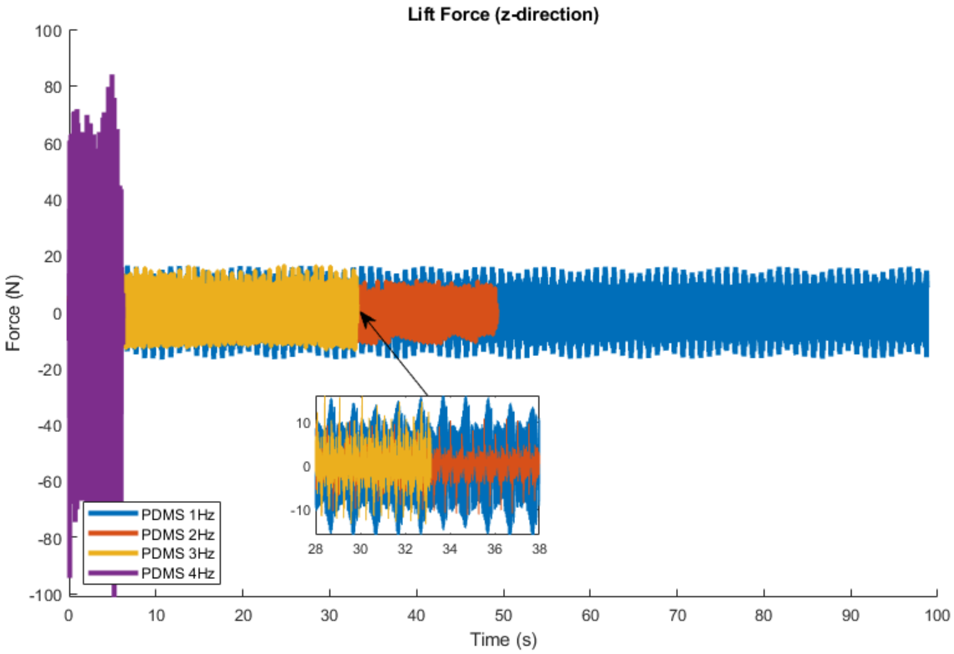




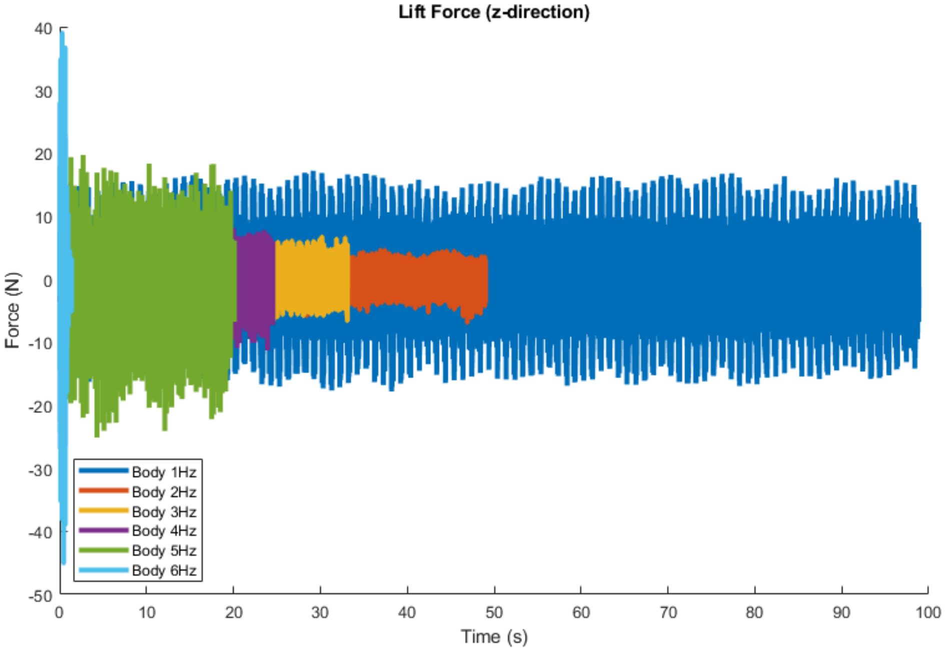
```
% ---------------------------------------------------------------
% -------------------------Plot PDMS Data------------------------
% ---------------------------------------------------------------

cases = ["PDMS_1Hz", "PDMS_2Hz", "PDMS_3Hz", "PDMS_4Hz"];

% Open a new figure.
f = figure;
```

```
f.Position = [200 50 900 560];
title("Lift Force (z-direction)");
xlabel("Time (s)");
ylabel("Force (N)");
hold on

for i = 1:length(cases)
    % Load data
    mat_name = cases(i) + ".mat";
    load(mat_name);

    case_name = strrep(cases(i),'_',' ');

    % Trimming off end of data (it appears beginning is already
    % trimmed) when differences between measurements are small
    count = 0;
    vertical_diffs = diff(data(:,4));
    for j = 1:length(vertical_diffs)
        if (abs(vertical_diffs(j)) < 0.05)
            count = count + 1;
        else
            count = 0;
        end
        if (count > 5)
            data = data(1:j-1000, :);
            break
        end
    end

    % resave data after trimming it
    save(cases(i) + ".mat", 'data')

    times = data(1:end,1);
    force_vals = data(1:end,2:7);

    % Plot lift force
    plot(times, force_vals(:, 3), 'DisplayName',
 case_name, "LineWidth",3);
end
legend("Location","Southwest");
ax1 = axes('Position',[0.35 0.2 0.2 0.2]);
hold on
for i = 1:length(cases)
    % Load data
    mat_name = cases(i) + ".mat";
    load(mat_name);

    case_name = strrep(cases(i),'_',' ');

    times = data(1:end,1);
    force_vals = data(1:end,2:7);

    % Plot lift force
    plot(ax1, times, force_vals(:, 3))
```

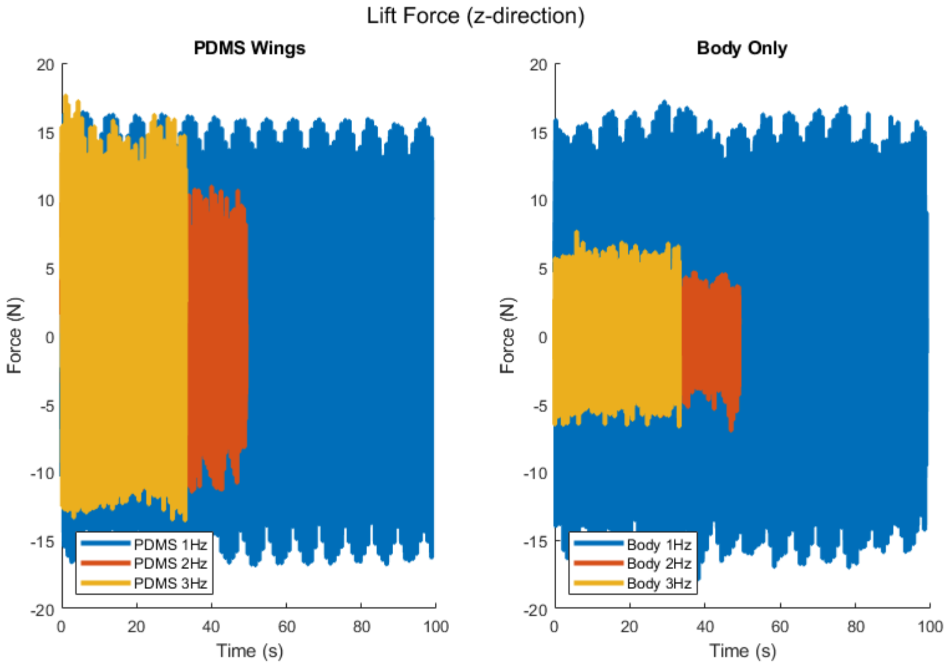
```
end
xlim([28, 38])
ylim([-16, 16])
box on
annotation('arrow',[0.45 0.39], [0.4 0.52])

% The data shows a positive correlation between wingbeat frequency and
% aerodynamic force, with the exception of 1 Hz. This exception is
% explained by the fact that the robot's natural frequency appeared to
% lie around 1 Hz so the system vibrated loudly for the 1 Hz test,
% obscuring the aerodynamic force production.
```



```
% -----------------------------------------------------------------
% ----------------------Plot Wingless Data------------------------
% -----------------------------------------------------------------

cases = ["Body_1Hz", "Body_2Hz", "Body_3Hz", ...
         "Body_4Hz", "Body_5Hz", "Body_6Hz"];

% Open a new figure.
f = figure;
f.Position = [200 50 900 560];
title("Lift Force (z-direction)");
xlabel("Time (s)");
ylabel("Force (N)");
hold on

for i = 1:length(cases)
    % Load data
    mat_name = cases(i) + ".mat";
    load(mat_name);
    case_name = strrep(cases(i),'_',' ');

    % Trimming off end of data (it appears beginning is already
    % trimmed)
    count = 0;
    vertical_diffs = diff(data(:,4));
    for j = 1:length(vertical_diffs)
        if (abs(vertical_diffs(j)) < 0.05)
            count = count + 1;
        else
            count = 0;
        end
        if (count > 5)
            data = data(1:j-1000, :);
            break
        end
    end

    % resave data after trimming it
    save(cases(i) + ".mat", 'data')

    times = data(1:end,1);
    force_vals = data(1:end,2:7);

    % Plot lift force
    plot(times, force_vals(:, 3), 'DisplayName',
 case_name, "LineWidth",3);
end
legend("Location","Southwest");
```



```
% -----------------------------------------------------------------
% -------Plot PDMS and Wingless Data at 1 Hz, 2 Hz, and 3 Hz------
% -----------------------------------------------------------------

cases = ["PDMS_1Hz", "PDMS_2Hz", "PDMS_3Hz"];

% Open a new figure.
f = figure;
f.Position = [200 50 900 560];
subplot(1,2,1)
title("PDMS Wings");
xlabel("Time (s)");
ylabel("Force (N)");
hold on
for i = 1:length(cases)
    % Load data
    mat_name = cases(i) + ".mat";
    load(mat_name);

    case_name = strrep(cases(i),'_',' ');

    times = data(1:end,1);
    force_vals = data(1:end,2:7);

    % Plot lift force
    plot(times, force_vals(:, 3), 'DisplayName',
 case_name, "LineWidth",3);
end
legend("Location","Southwest");

%-------------------------Body Only Plot-----------------------------
cases = ["Body_1Hz", "Body_2Hz", "Body_3Hz"];

subplot(1,2,2)
title("Body Only");
xlabel("Time (s)");
ylabel("Force (N)");
hold on
for i = 1:length(cases)
    % Load data
    mat_name = cases(i) + ".mat";
    load(mat_name);

    case_name = strrep(cases(i),'_',' ');

    times = data(1:end,1);
    force_vals = data(1:end,2:7);

    % Plot lift force
    plot(times, force_vals(:, 3), 'DisplayName',
 case_name, "LineWidth",3);
end
legend("Location","Southwest");
sgtitle("Lift Force (z-direction)");
```

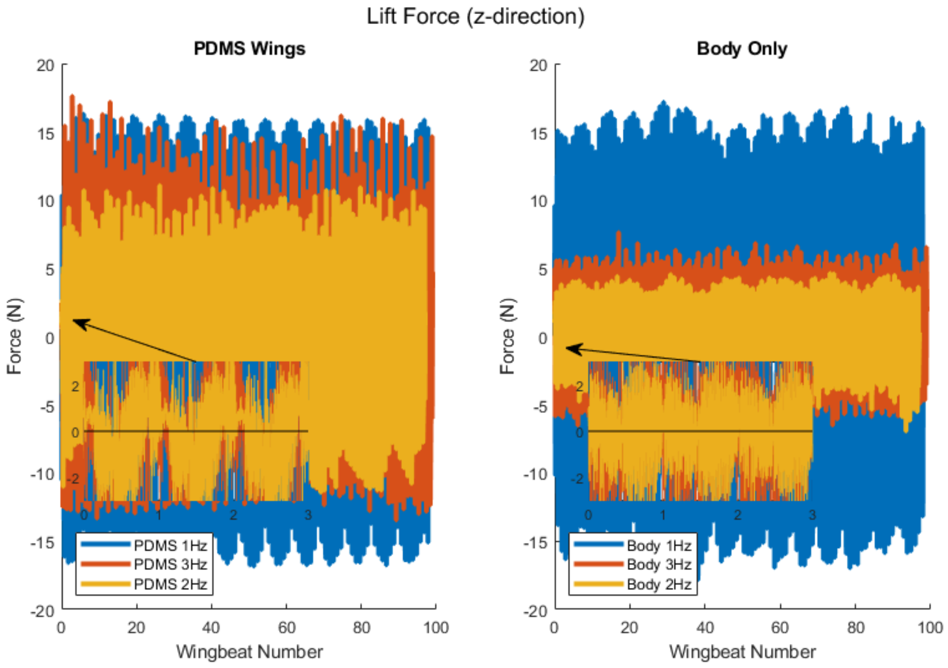


```
% ----------------------------------------------------------------
% -------Plot PDMS and Wingless Data at 1 Hz, 2 Hz, and 3 Hz------
% ------------------normalized by wing cycles--------------------
% ----------------------------------------------------------------

cases = ["PDMS_1Hz", "PDMS_3Hz", "PDMS_2Hz"];

% Open a new figure.
f = figure;
f.Position = [200 50 900 560];
subplot(1,2,1)
title("PDMS Wings");
xlabel("Wingbeat Number");
ylabel("Force (N)");
hold on
for i = 1:length(cases)
    % Load data
    mat_name = cases(i) + ".mat";
    load(mat_name);

    case_name = strrep(cases(i),'_',' ');

    times = data(1:end,1);
    force_vals = data(1:end,2:7);

    % Filtering force data with moving average filter
    window = 100;
    b = 1/window*ones(window,1);
    filtered_lift_vals = filter(b, 1, force_vals(:, 3));

    % Count the number of wingbeats
    wingbeat_count = 0;
    case_name = char(case_name);
    speed = str2double(case_name(6));
    window = round(400 / speed);
    start_index = 0;
    end_index = 0;
    for j = (1 + window):(length(filtered_lift_vals) - window)
        if (filtered_lift_vals(j) == max(filtered_lift_vals(j-window:j
+window)))
            wingbeat_count = wingbeat_count + 1;
            if (start_index == 0)
                start_index = j;
            else
                end_index = j;
            end
        end
    end

    data = data(start_index:end_index,:);
    filtered_lift_vals = filtered_lift_vals(start_index:end_index);
    wingbeats = linspace(0,wingbeat_count,length(filtered_lift_vals));
    disp("For the " + case_name + " trial, " + wingbeat_count + ...
         " wingbeats were identified elapsing " + ...
         length(filtered_lift_vals) + " frames.");

    force_vals = data(1:end,2:7);
    % Plot lift force
    plot(wingbeats, force_vals(:,3), 'DisplayName',
 case_name, "LineWidth",3);
    save(cases(i) + ".mat", 'data', 'filtered_lift_vals','wingbeats');
end
legend("Location","Southwest");
ax1 = axes('Position',[0.15 0.26 0.2 0.2]);
hold on
for i = 1:length(cases)
    % Load data
    mat_name = cases(i) + ".mat";
    load(mat_name,'data','wingbeats');

    force_vals = data(1:end,2:7);

    % Plot lift force
    plot(ax1, wingbeats, force_vals(:,3));
end
xlim([0, 3])
ylim([-3, 3])
y_axis = line(xlim, [0 0], 'Color','black');
box on
annotation('arrow',[0.25 0.14], [0.46 0.52])

%-------------------------Body Only Plot-----------------------------
cases = ["Body_1Hz", "Body_3Hz", "Body_2Hz"];

subplot(1,2,2)
```

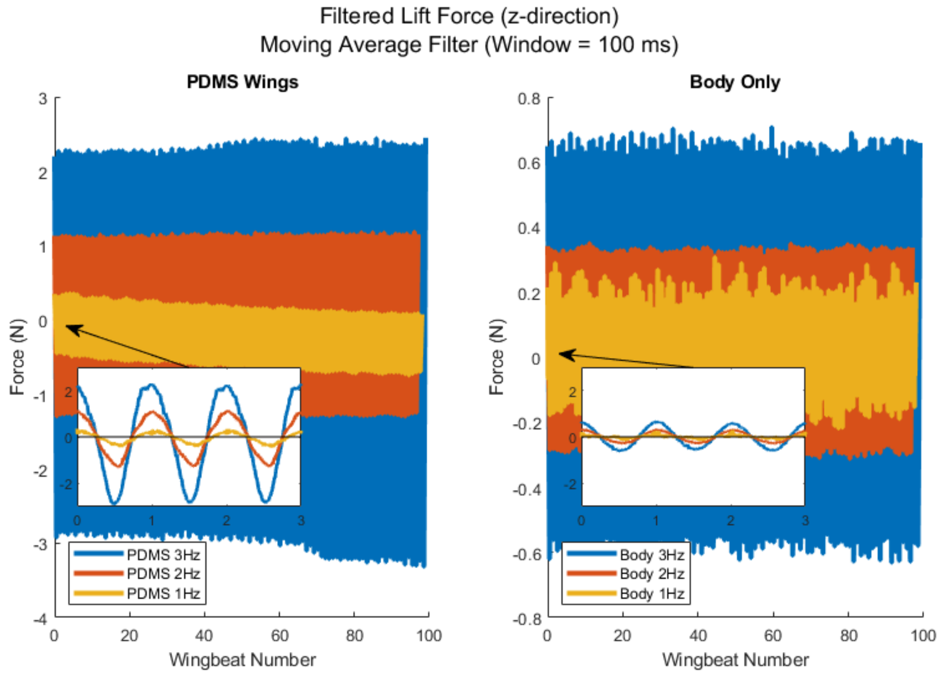
```
title("Body Only");
xlabel("Wingbeat Number");
ylabel("Force (N)");
hold on
for i = 1:length(cases)
    % Load data
    mat_name = cases(i) + ".mat";
    load(mat_name);

    case_name = strrep(cases(i),'_',' ');

    times = data(1:end,1);
    force_vals = data(1:end,2:7);

    % Filtering force data with moving average filter
    window = 100;
    b = 1/window*ones(window,1);
    filtered_lift_vals = filter(b, 1, force_vals(:, 3));

    % Count the number of wingbeats
    wingbeat_count = 0;
    case_name = char(case_name);
    speed = str2double(case_name(6));
    window = round(400 / speed);
    start_index = 0;
    end_index = 0;
    for j = (1 + window):(length(filtered_lift_vals) - window)
        if (filtered_lift_vals(j) == max(filtered_lift_vals(j-window:j
+window)))
            wingbeat_count = wingbeat_count + 1;
            if (start_index == 0)
                start_index = j;
            else
                end_index = j;
            end
        end
    end

    data = data(start_index:end_index,:);
    filtered_lift_vals = filtered_lift_vals(start_index:end_index);
    wingbeats = linspace(0,wingbeat_count,length(filtered_lift_vals));

    disp("For the " + case_name + " trial, " + wingbeat_count + ...
         " wingbeats were identified elapsing " + ...
         length(filtered_lift_vals) + " frames.");

    force_vals = data(1:end,2:7);
    % Plot lift force
    plot(wingbeats, force_vals(:,3), 'DisplayName',
 case_name, "LineWidth",3);
    save(cases(i) + ".mat", 'data', 'filtered_lift_vals','wingbeats');
end
legend("Location","Southwest");
ax2 = axes('Position',[0.6 0.26 0.2 0.2]);
```

```
hold on
for i = 1:length(cases)
    % Load data
    mat_name = cases(i) + ".mat";
    load(mat_name,'data','wingbeats');

    force_vals = data(1:end,2:7);

    % Plot lift force
    plot(ax2, wingbeats, force_vals(:,3));
end
xlim([0, 3])
ylim([-3, 3])
y_axis = line(xlim, [0 0], 'Color','black');
box on
annotation('arrow',[0.7 0.58], [0.46 0.48])

sgtitle("Lift Force (z-direction)");
```

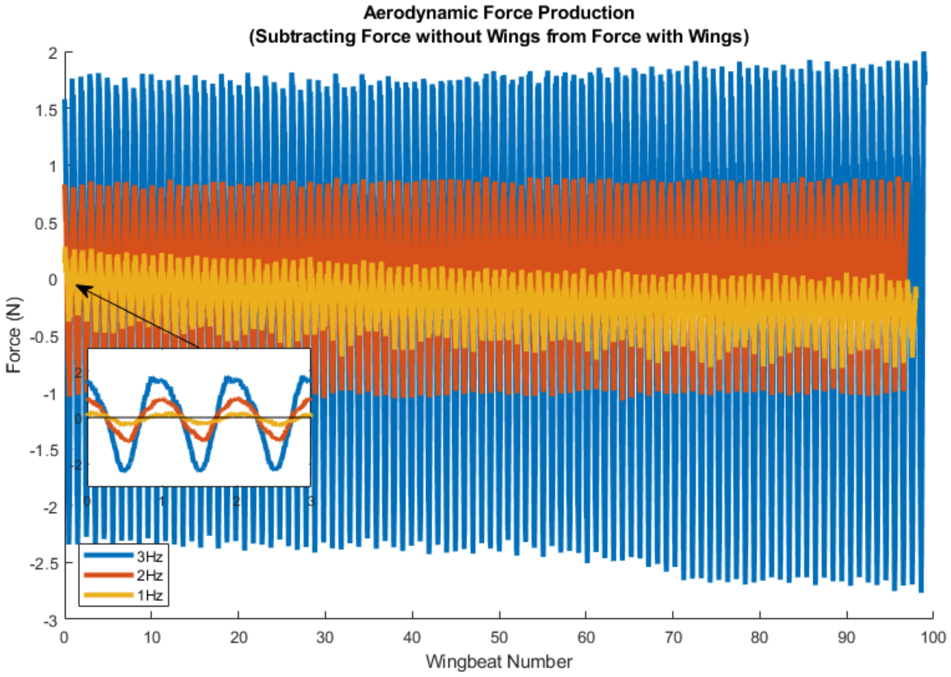
```
For the PDMS 1Hz trial, 98 wingbeats were identified elapsing 97030
 frames.
For the PDMS 3Hz trial, 99 wingbeats were identified elapsing 32666
 frames.
For the PDMS 2Hz trial, 97 wingbeats were identified elapsing 48008
 frames.
For the Body 1Hz trial, 98 wingbeats were identified elapsing 97015
 frames.
For the Body 3Hz trial, 99 wingbeats were identified elapsing 32669
 frames.
For the Body 2Hz trial, 97 wingbeats were identified elapsing 48013
 frames.
```



```
% ----------------------------------------------------------------
% -------Plot PDMS and Wingless Data at 1 Hz, 2 Hz, and 3 Hz------
% -------------normalized by wing cycles and filtered-------------
% ----------------------------------------------------------------

cases = ["PDMS_3Hz", "PDMS_2Hz", "PDMS_1Hz"];

% Open a new figure.
f = figure;
f.Position = [200 50 900 560];
subplot(1,2,1)
title("PDMS Wings");
xlabel("Wingbeat Number");
ylabel("Force (N)");
hold on
for i = 1:length(cases)
    % Load data
    mat_name = cases(i) + ".mat";
    load(mat_name);

    case_name = strrep(cases(i),'_',' ');

    % Plot lift force
    plot(wingbeats, filtered_lift_vals, 'DisplayName',
 case_name, "LineWidth",3);
end
legend("Location","Southwest");
ax1 = axes('Position',[0.15 0.26 0.2 0.2]);
hold on
for i = 1:length(cases)
    % Load data
    mat_name = cases(i) + ".mat";
    load(mat_name,'filtered_lift_vals','wingbeats');

    % Plot lift force
    plot(ax1, wingbeats, filtered_lift_vals, "LineWidth",2);
end
xlim([0, 3])
ylim([-3, 3])
y_axis = line(xlim, [0 0], 'Color','black');
box on
annotation('arrow',[0.25 0.14], [0.46 0.52])

%-------------------------Body Only Plot-----------------------------
cases = ["Body_3Hz", "Body_2Hz", "Body_1Hz"];

subplot(1,2,2)
title("Body Only");
xlabel("Wingbeat Number");
ylabel("Force (N)");
hold on
for i = 1:length(cases)
    % Load data
    mat_name = cases(i) + ".mat";
    load(mat_name);

    case_name = strrep(cases(i),'_',' ');

    % Plot lift force
    plot(wingbeats, filtered_lift_vals, 'DisplayName',
 case_name, "LineWidth",3);
end
legend("Location","Southwest");
ax2 = axes('Position',[0.6 0.26 0.2 0.2]);
hold on
for i = 1:length(cases)
    % Load data
    mat_name = cases(i) + ".mat";
    load(mat_name,'filtered_lift_vals','wingbeats');

    % Plot lift force
    plot(ax2, wingbeats, filtered_lift_vals, "LineWidth",2);
end
xlim([0, 3])
ylim([-3, 3])
y_axis = line(xlim, [0 0], 'Color','black');
box on
annotation('arrow',[0.7 0.58], [0.46 0.48])

sgtitle(["Filtered Lift Force (z-direction)" "Moving Average Filter
 (Window = 100 ms)"]);
```



```
% ----------------------------------------------------------------
% -------Plot Wingless Data Subtracted from PDMS Data-------------
% ---------------------at 1 Hz, 2 Hz, and 3 Hz--------------------
% ----------------------------------------------------------------

body_cases = ["Body_3Hz", "Body_2Hz", "Body_1Hz"];
wing_cases = ["PDMS_3Hz", "PDMS_2Hz", "PDMS_1Hz"];

% Open a new figure.
f = figure;
f.Position = [200 50 900 560];
title(["Aerodynamic Force Production" "(Subtracting Force without
 Wings from Force with Wings)"]);
xlabel("Wingbeat Number");
ylabel("Force (N)");
hold on
for i = 1:3
    case_name = char(body_cases(i));
    speed = case_name(6:end);

    % Load body data
    mat_name = body_cases(i) + ".mat";
    load(mat_name, 'filtered_lift_vals');
    lift_body = filtered_lift_vals;

    % Load wing data
    mat_name = wing_cases(i) + ".mat";
    load(mat_name,'filtered_lift_vals','wingbeats');
    lift_PDMS = filtered_lift_vals;

    min_length = min(length(lift_body), length(lift_PDMS));
    lift_sub = lift_PDMS(1:min_length) - lift_body(1:min_length);
    wingbeats_sub = wingbeats(1:min_length);

    % Plot lift force
    plot(wingbeats_sub, lift_sub, 'DisplayName',
 speed, "LineWidth",3);
end
legend("Location","Southwest");
ax1 = axes('Position',[0.15 0.3 0.2 0.2]);
hold on
for i = 1:3
    % Load body data
    mat_name = body_cases(i) + ".mat";
    load(mat_name, 'filtered_lift_vals');
    lift_body = filtered_lift_vals;

    % Load wing data
    mat_name = wing_cases(i) + ".mat";
    load(mat_name,'filtered_lift_vals','wingbeats');
    lift_PDMS = filtered_lift_vals;

    min_length = min(length(lift_body), length(lift_PDMS));
    lift_sub = lift_PDMS(1:min_length) - lift_body(1:min_length);
    wingbeats_sub = wingbeats(1:min_length);

    % Plot lift force
    plot(ax1, wingbeats_sub, lift_sub, "LineWidth",3);
end
xlim([0, 3])
ylim([-3, 3])
y_axis = line(xlim, [0 0], 'Color','black');
box on
annotation('arrow',[0.25 0.14], [0.5 0.59])
```



```
% % ----------------------------------------------------------------
% % ---------------------------Trashed Code-------------------------
% % ----------------------------------------------------------------
%
% % Fast Fourier Transform stuff I was looking at for a bit...
%
% % f = figure;
% % f.Position = [200 50 900 560];
% % %instfreq(force_vals(:, 3),fs)
% % Y = fft(force_vals(:, 3));
% % L = 2000;
% % P2 = abs(Y/L);
% % P1 = P2(1:L/2+1);
% % P1(2:end-1) = 2*P1(2:end-1);
% %
% % freqs = fs*(0:(L/2))/L;
% % plot(freqs,P1)
% % title("Single-Sided Amplitude Spectrum of Force Transdcuer Data")
% % xlabel("f (Hz)")
% % ylabel("|P1(f)|")
% % xlim([0,10])
%
% % instantaneous frequency stuff I was trying...
```

```
%
% % Open a new figure.
% f = figure;
% f.Position = [200 50 900 560];
% title("Instantaneous Frequency of Force in z-direction");
% xlabel("Time (s)");
% ylabel("Force Frequency i.e. Wing Speed (Hz)");
% hold on
%
% for i = 1:length(files)
%     % Get case name from file name
%     case_name = erase(files(i), ["12_02_2022_benchtop_test/",
 "_experiment_120222.csv"]);
%     case_name = strrep(case_name,'_',' ');
%
%     % Get data from file
%     data = readmatrix(files(i));
%
%     times = data(1:end,1);
%     force_vals = data(1:end,2:7);
%
%     % Filtering force transducer data with a butterworth filter
%     fc = 3; % cutoff frequency
%     fs = 1000; % sample frequency
%
%     [b,a] = butter(6,fc/(fs/2)); % 6th order
%     force_vals = filter(b, a, force_vals);
%
% %     [s,f,t] = stft(force_vals(:, 3),fs);
% %     stft(s(64,:),fs);
% %     force_vals(:, 3) = force_vals(:, 3) - mean(force_vals(:, 3));
% %     stft(force_vals(:,
 3),fs,'Window',kaiser(1024,5),'OverlapLength',500,'FFTLength',1024,
 'FrequencyRange','centered');
%
% %     pspectrum(force_vals(:, 3),fs,'spectrogram');
% %     ylim([0,10]);
%
%     [ifq,t] = instfreq(force_vals(:, 3),fs);
%
%     % Filtering instantaneous frequency with a butterworth filter
%     window = 100;
%     b = 1/window*ones(window,1);
%     ifq = filter(b, 1, ifq);
%
%     plot(t, ifq, 'DisplayName', case_name, "LineWidth", 3)
% end
% legend("Location","Southwest");
```

Published with MATLAB® R2020b