
Table of Contents

.....	1
File Setup	1
Averaged Velocities	1
Plot difference scatter in-out	2
Compare AQD - VEC	3
Compare AQD - M2	4
Compare VEC - M2	5
Functions	9
Defaults	10
Time Series Plot Setup	10
Quiver Plot Setup	11
EOF	13

```
clear all;
close all;
```

File Setup

```
files = dir(' ../../../../Kelp_data/Summer2025/Rooker/Release2/L0/*.mat');
colors = {[0, 0, 1], [1, 0, 0], [1, 0, 1], [0, 1, 0]};

for i = 1:length(files)
    dataCell{i} = load(fullfile(files(i).folder, files(i).name));
    labels{i} = dataCell{i}.Config.SN;

    % Determine appropriate bin
    if size(dataCell{i}.Velocity_East, 2) == 1
        dataCell{i}.bin = 1;
    else
        dataCell{i}.bin = round((1.237 - dataCell{i}.Config.blank) /
dataCell{i}.Config.binSize);

    end
end
```

Averaged Velocities

```
addpath(' ../code')
[T_all, U_all, V_all] = LPF(dataCell, labels, colors);

% normalizing vector size
[~, sizeLimit] = size(U_all{2}); % gonna make this a max function later

AQD = [U_all{1}(1, 1:sizeLimit); V_all{1}(1, 1:sizeLimit)];
M1 = [U_all{2}(1, 1:sizeLimit); V_all{2}(1, 1:sizeLimit)];
M2 = [U_all{3}(1, 1:sizeLimit); V_all{3}(1, 1:sizeLimit)];
VEC = [U_all{4}(1, 1:sizeLimit); V_all{4}(1, 1:sizeLimit)];
```

```
% Compare inside instruments to M1 (Inst - M1)
AQDvsM1 = AQD-M1;
M2vsM1  = M2-M1;
VECvsM1 = VEC-M1;
```

Plot difference scatter in-out

```
figure
scatter(AQDvsM1(1,:), AQDvsM1(2,:), 'b.')
hold on
scatter(M2vsM1(1,:),M2vsM1(2,:) , 'm.')
scatter(VECvsM1(1,:),VECvsM1(2,:) , 'g.')

% Generate and plot trendlines
AQDtrend = polyfit(AQDvsM1(1,:), AQDvsM1(2,:), 1);
yfit = polyval(AQDtrend, AQDvsM1(1,:));
plot(AQDvsM1(1,:), yfit, 'b-', 'LineWidth', 2)

M2trend = polyfit(M2vsM1(1,:), M2vsM1(2,:), 1);
yfit = polyval(M2trend, M2vsM1(1,:));
plot(M2vsM1(1,:), yfit, 'm-', 'LineWidth', 2)

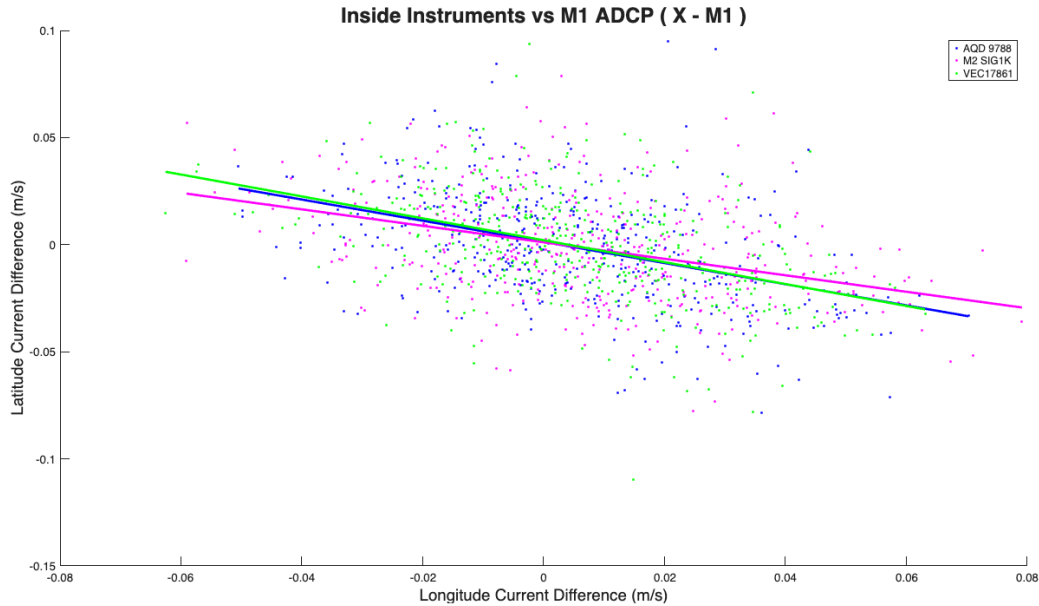
VECTrend = polyfit(VECvsM1(1,:), VECvsM1(2,:), 1);
yfit = polyval(VECTrend, VECvsM1(1,:));
plot(VECvsM1(1,:), yfit, 'g-', 'LineWidth', 2)

slope = mean([AQDtrend M2trend VECTrend]);
theta = atan2d(slope, 1);
sprintf('Angle of max difference variability is %f degrees', theta)

% Formatting and such
title("Inside Instruments vs M1 ADCP ( X - M1 )", "FontSize", 18)
legend(labels{[1 3 4]})
ylabel('Latitude Current Difference (m/s)', 'FontSize', 14)
xlabel('Longitude Current Difference (m/s)', 'FontSize', 14)

ans =

'Angle of max difference variability is -13.003541 degrees'
```



Compare AQD - VEC

```
AQDvsVEC = AQD-VEC;
figure, scatter(AQDvsVEC(1,:), AQDvsVEC(2,:), '.')
```

```
% mean and covariance
mu = [mean(AQDvsVEC(1,:)), mean(AQDvsVEC(2,:))];
Sigma = cov(AQDvsVEC(1,:), AQDvsVEC(2,:));
```

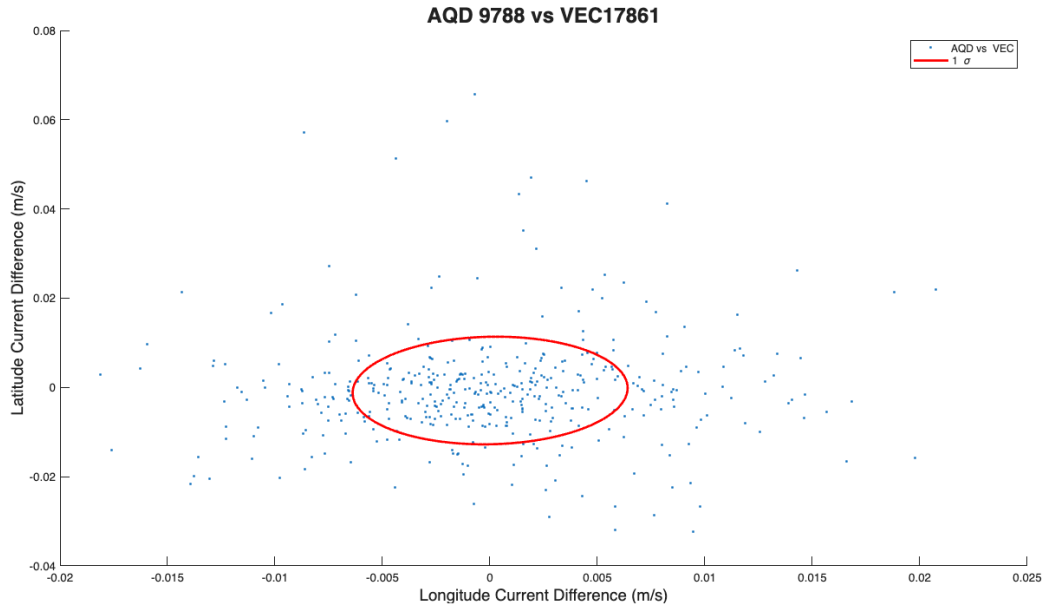
```
% eigen decomposition
[eigvec,eigval] = eig(Sigma);
```

```
% parametric ellipse
theta = linspace(0,2*pi,200);
ellipse = [cos(theta); sin(theta)]';
```

```
% scale by sqrt of eigenvalues (std dev)
ellipse = ellipse * sqrt(eigval) * eigvec';
```

```
% shift to mean
ellipse = ellipse + mu;
hold on
plot(ellipse(:,1), ellipse(:,2), 'r','LineWidth',2)
```

```
% Formatting
title([labels{1} ' vs ' labels{4}], 'FontSize', 18)
ylabel('Latitude Current Difference (m/s)', 'FontSize', 14)
xlabel('Longitude Current Difference (m/s)', 'FontSize', 14)
legend('AQD vs VEC', '1 \sigma')
```



Compare AQD - M2

```
AQDvsM2 = AQD-M2;
figure
scatter(AQDvsM2(1,:), AQDvsM2(2,:), '.')
```

```
% mean and covariance
mu = [mean(AQDvsM2(1,:)), mean(AQDvsM2(2,:))];
Sigma = cov(AQDvsM2(1,:), AQDvsM2(2,:));
```

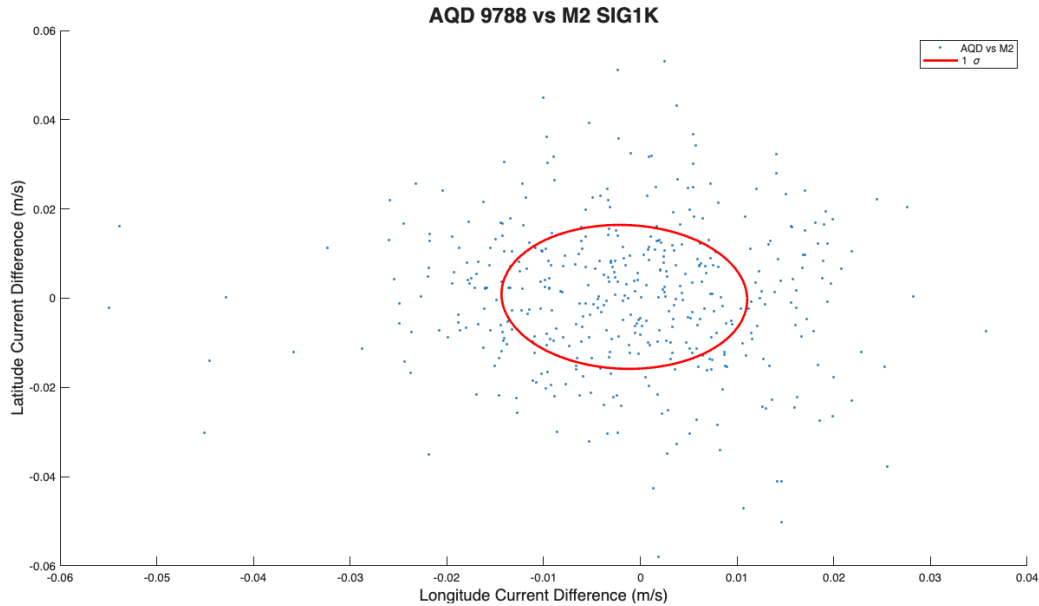
```
% eigen decomposition
[eigvec,eigval] = eig(Sigma);
```

```
% parametric ellipse
theta = linspace(0,2*pi,200);
ellipse = [cos(theta); sin(theta)]';
```

```
% scale by sqrt of eigenvalues (std dev)
ellipse = ellipse * sqrt(eigval) * eigvec';
```

```
% shift to mean
ellipse = ellipse + mu;
hold on
plot(ellipse(:,1), ellipse(:,2), 'r','LineWidth',2)
```

```
% Formatting
title([labels{1} ' vs ' labels{3}], 'FontSize', 18)
ylabel('Latitude Current Difference (m/s)', 'FontSize', 14)
xlabel('Longitude Current Difference (m/s)', 'FontSize', 14)
legend('AQD vs M2', '1 \sigma')
```



Compare VEC - M2

```

VECvsM2 = VEC-M2;
figure
scatter(VECvsM2(1,:), VECvsM2(2,:), '.')
```

% mean and covariance

```

mu = [mean(VECvsM2(1,:)), mean(VECvsM2(2,:))];
Sigma = cov(VECvsM2(1,:), VECvsM2(2,:));
```

% eigen decomposition

```

[eigvec,eigval] = eig(Sigma);
```

% parametric ellipse

```

theta = linspace(0,2*pi,200);
ellipse = [cos(theta); sin(theta)]';
```

% scale by sqrt of eigenvalues (std dev)

```

ellipse = ellipse * sqrt(eigval) * eigvec';
```

% shift to mean

```

ellipse = ellipse + mu;
hold on
plot(ellipse(:,1), ellipse(:,2), 'r','LineWidth',2)
```

% Formatting

```

title([labels{4} ' vs ' labels{3}], 'FontSize', 18)
ylabel('Latitude Current Difference (m/s)', 'FontSize', 14)
xlabel('Longitude Current Difference (m/s)', 'FontSize', 14)
legend('VEC vs M2', '1 \sigma')
```

```

% figure(3)
exportgraphics(ffigure(1), '../../../../../Kelp_data/Summer2025/Rooker/figures/
Release2/timeseries/10_min_avg.png')
exportgraphics(ffigure(2), '../../../../../Kelp_data/Summer2025/Rooker/figures/
Release2/QuiverPlot.png')
exportgraphics(ffigure(3), '../../../../../Kelp_data/Summer2025/Rooker/figures/
Release2/DifferenceScatter.png')
exportgraphics(ffigure(4), '../../../../../Kelp_data/Summer2025/Rooker/figures/
Release2/AQDvsVEC.png')
exportgraphics(ffigure(5), '../../../../../Kelp_data/Summer2025/Rooker/figures/
Release2/AQDvsM2.png')
exportgraphics(ffigure(6), '../../../../../Kelp_data/Summer2025/Rooker/figures/
Release2/VECvsM2.png')

filestem = '../../../../../Kelp_data/Summer2025/Rooker/Release2/LPF';

labels = replace(labels, ' ', '');

disp('Saving has been disabled to prevent accidental overwriting...')

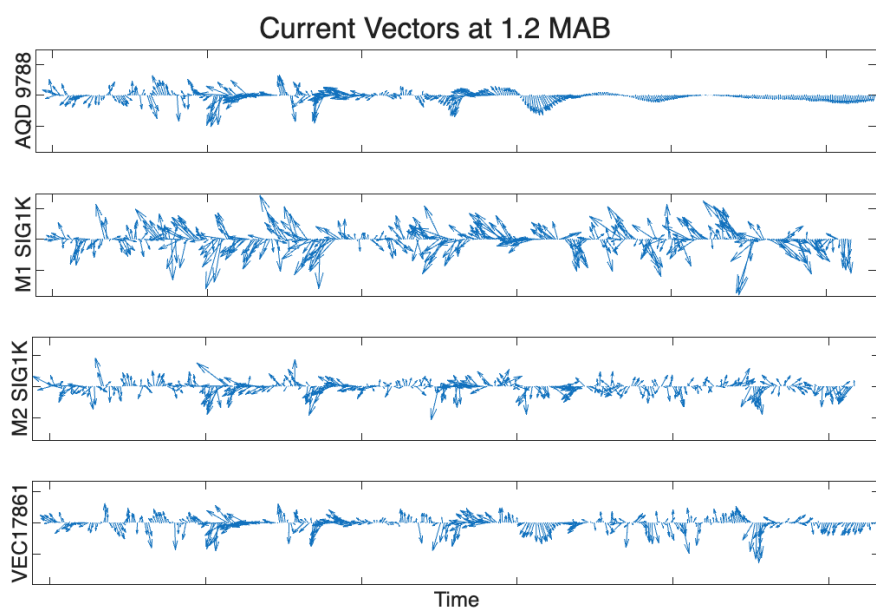
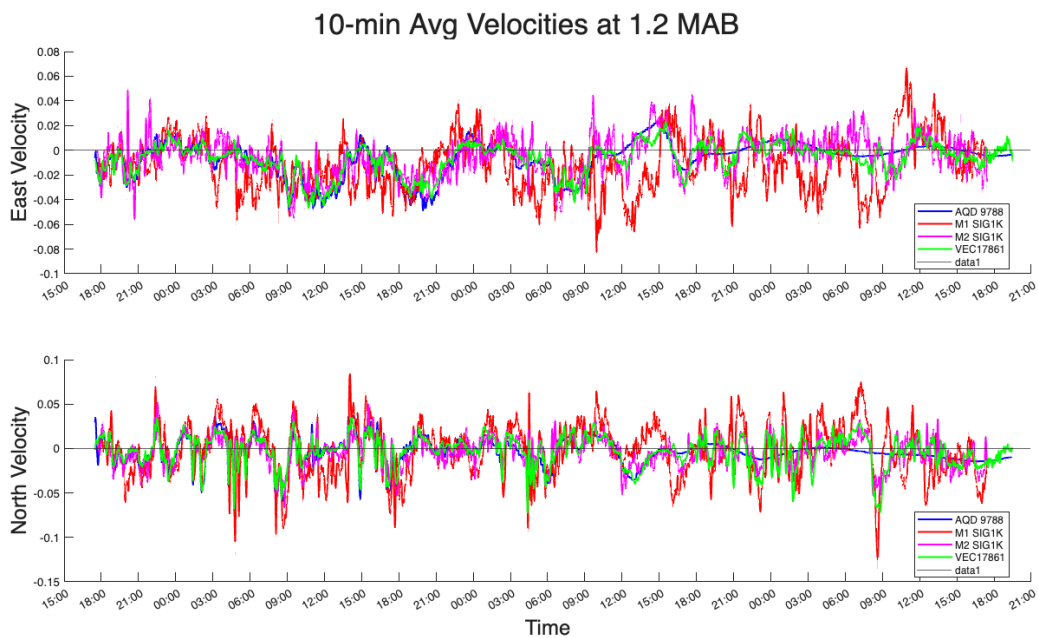
return

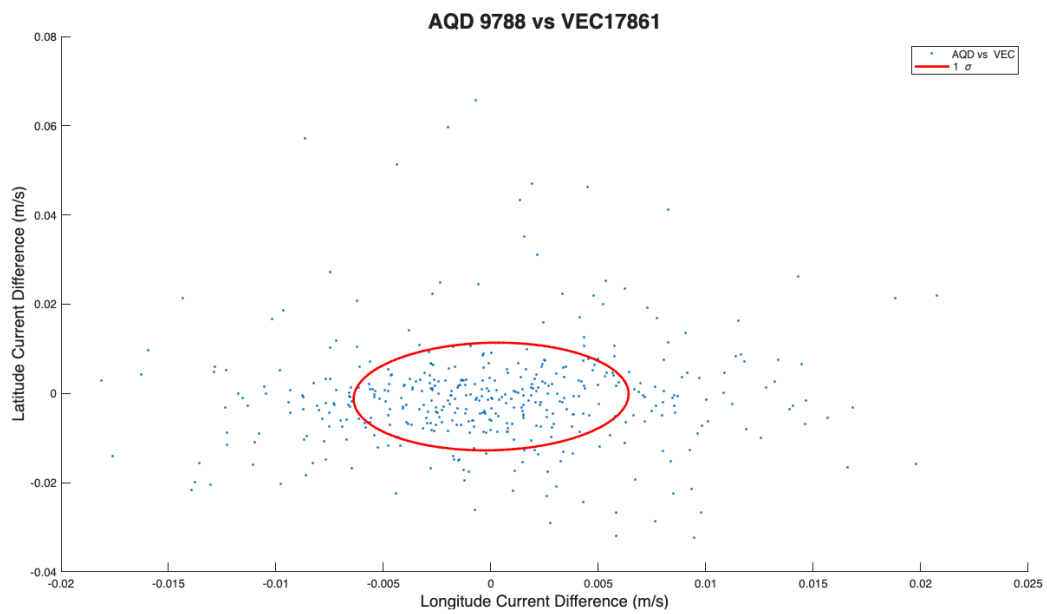
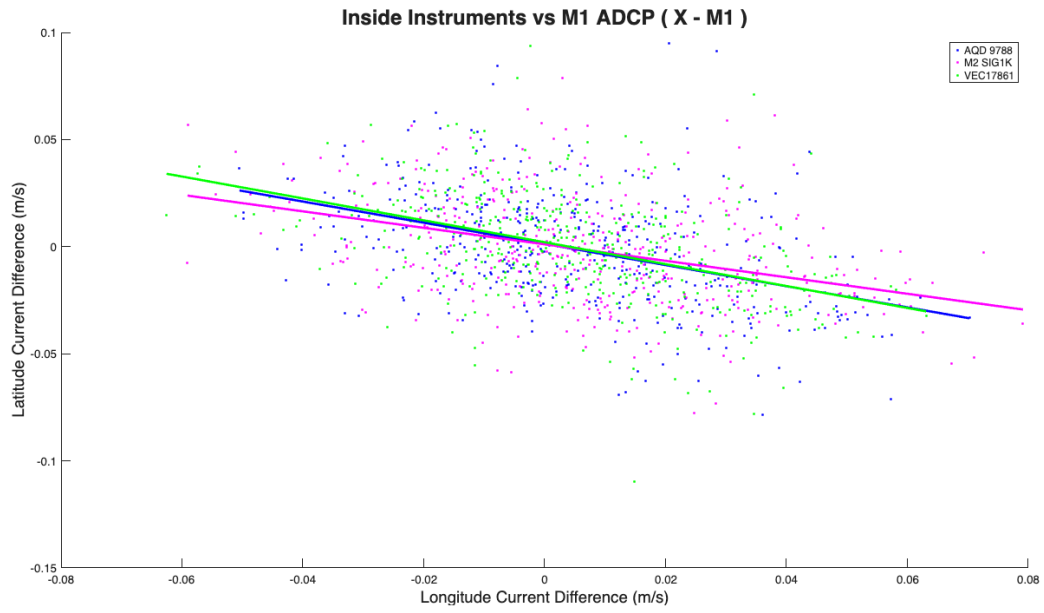
for i = 1:length(labels)

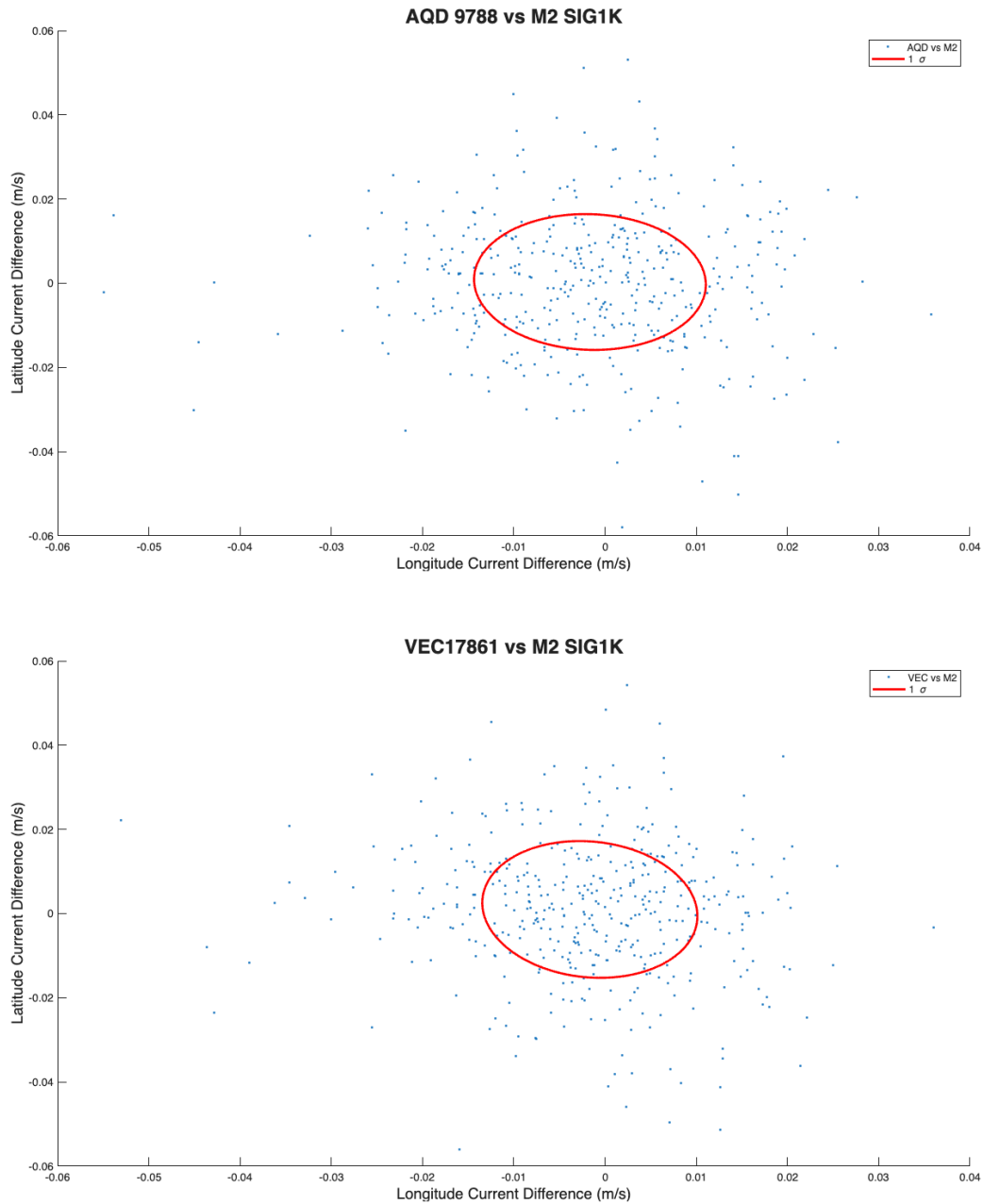
    Velocity_X = U_all{i}';
    Velocity_Y = V_all{i}';
    Time = datenum(T_all{i}');
    label = labels{i};
    sprintf('Saving %s...', labels{i})
    save([filestem '/' labels{i} '.mat'], ...
        'Velocity_X', 'Velocity_Y', 'Time', 'label');
    pca_function(filestem, labels{i})
    title(labels{i})
end

Saving has been disabled to prevent accidental overwriting...

```







Functions

```
function [T_all, U_all, V_all] = LPF(dataCell, labels, colors)

%
% USAGE: [T_all, U_all, V_all] = LPF(dataCell, labels, colors)
%
%   takes time series data and
%   returns 3 cell arrays, containing 10-min averaged data, u, v, and time
%
%   dataCell = cell array (# of files x 1) containing data structures
```

```

% labels = {optional} Cell array of labels for graphs
% colors = {optional} Cell array of RGB vals for graphs
%
%
% T_all = interpolated time
% U_all = East Velocities
% V_all = North Velocities
%

```

Defaults

```

if nargin < 3
    colors = num2cell(lines(length(dataCell)), 2);
    if nargin < 2
        labels = num2cell(1:length(dataCell));
    end
end
end

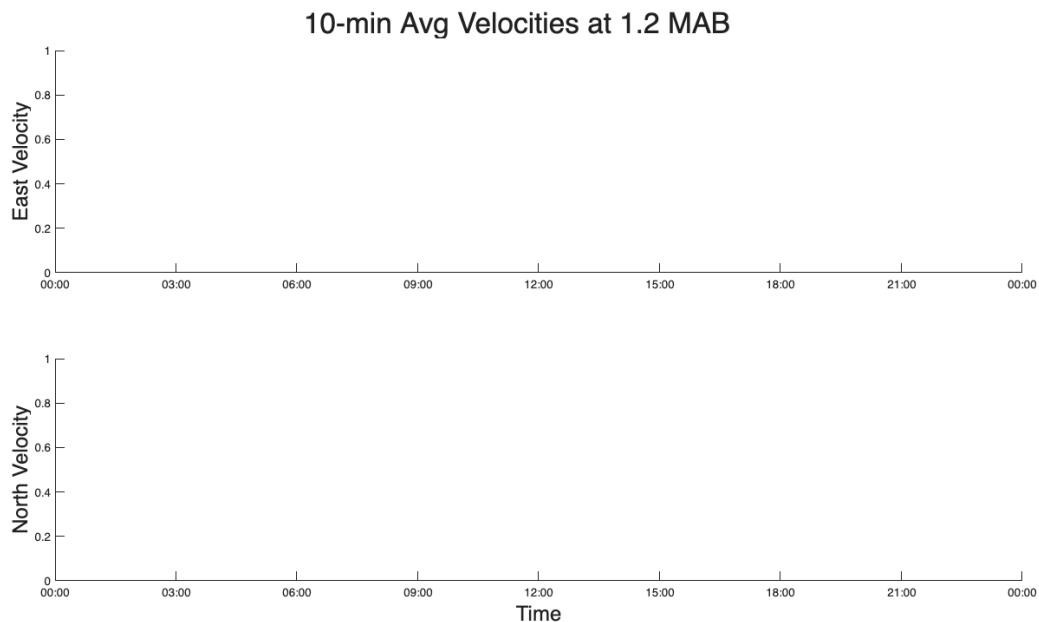
```

Time Series Plot Setup

```

tp = figure;
ax1 = subplot(2, 1, 1); hold(ax1, 'on'); ylabel(ax1, 'East Velocity',
'FontSize', 18);
ax2 = subplot(2, 1, 2); hold(ax2, 'on'); ylabel(ax2, 'North Velocity',
'FontSize', 18);
xlabel(ax2, 'Time', 'FontSize', 18);
datetick(ax1, 'x', 'keeplimits'); datetick(ax2, 'x', 'keeplimits');
sgtitle('10-min Avg Velocities at 1.2 MAB', 'FontSize', 25)

```



Quiver Plot Setup

```
qp = figure;
for i = 1:length(dataCell)

    data = dataCell{i};

    % Extract and clean data
    u = data.Velocity_East(:, data.bin);
    v = data.Velocity_North(:, data.bin);
    t = data.Time;
    dt = double(data.Config.dt);

    % Lowpass filter
    Nf = 600 / dt;
    flt = hamming(Nf); flt = flt / sum(flt);
    nanFlag = ~isnan(u);
    flag_flt = conv(nanFlag, flt, 'same');
    u(~nanFlag) = 0; v(~nanFlag) = 0;
    u_filt = conv(u, flt, 'same') ./ flag_flt;
    v_filt = conv(v, flt, 'same') ./ flag_flt;

    % Plot time series
    figure(1);
    h1(i) = plot(ax1, t, u_filt, 'LineWidth', 1.5);
    h2(i) = plot(ax2, t, v_filt, 'LineWidth', 1.5);
    linkaxes([ax1 ax2], 'x')

    % Downsample to 10-minute intervals
    [~, unique_idx] = unique(minutes(t - t(1)));
    t = t(unique_idx);
    u_filt = u_filt(unique_idx);
    v_filt = v_filt(unique_idx);
    tq = t(1):minutes(10):t(end);
    u_ds = interp1(t, u_filt, tq, 'linear');
    v_ds = interp1(t, v_filt, tq, 'linear');

    % Plot quiver per instrument
    figure(2)
    subplot(length(dataCell), 1, i)

    % Time relative to start, in seconds
    tq_rel = seconds(tq - tq(1));

    % Convert velocity to displacement per 10 min
    dt = 600*100*3; % scaling
    u_disp = u_ds * dt;
    v_disp = v_ds * dt;

    % Plot quiver
    quiver(tq_rel, zeros(size(tq_rel)), u_disp, v_disp, 'AutoScale', 'off')
    axis equal % ensure same scale in x and y for correct arrow directions
```

```

yticklabels([])
xticklabels([])
ylabel(data.Config.SN, 'FontSize', 18)

% Save Global Vars
U_all{i} = u_ds;
V_all{i} = v_ds;
T_all{i} = tq;

end
xlabel('Time', 'FontSize', 18)
sgtitle('Current Vectors at 1.2 MAB', 'FontSize', 25)

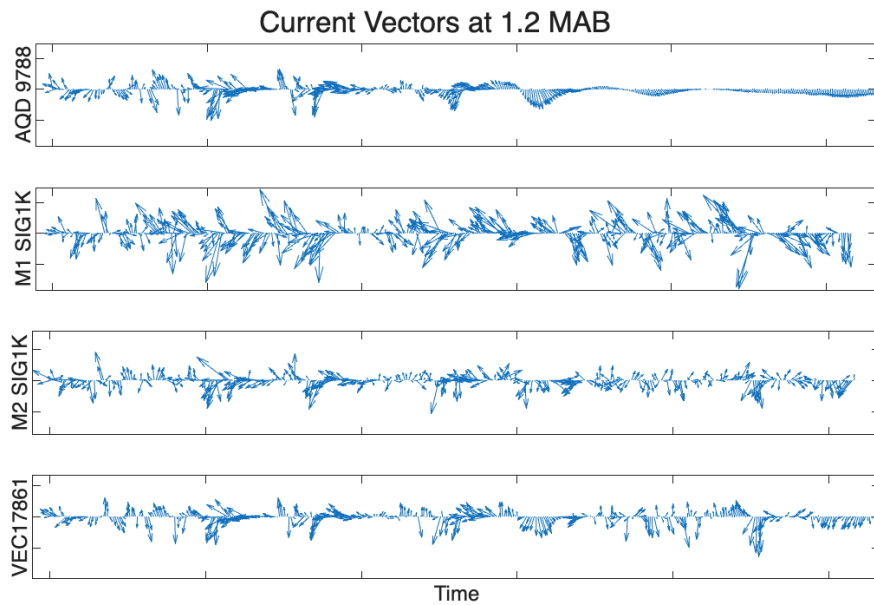
quivers = get(qp, 'Children');
linkaxes(quivers, 'x')

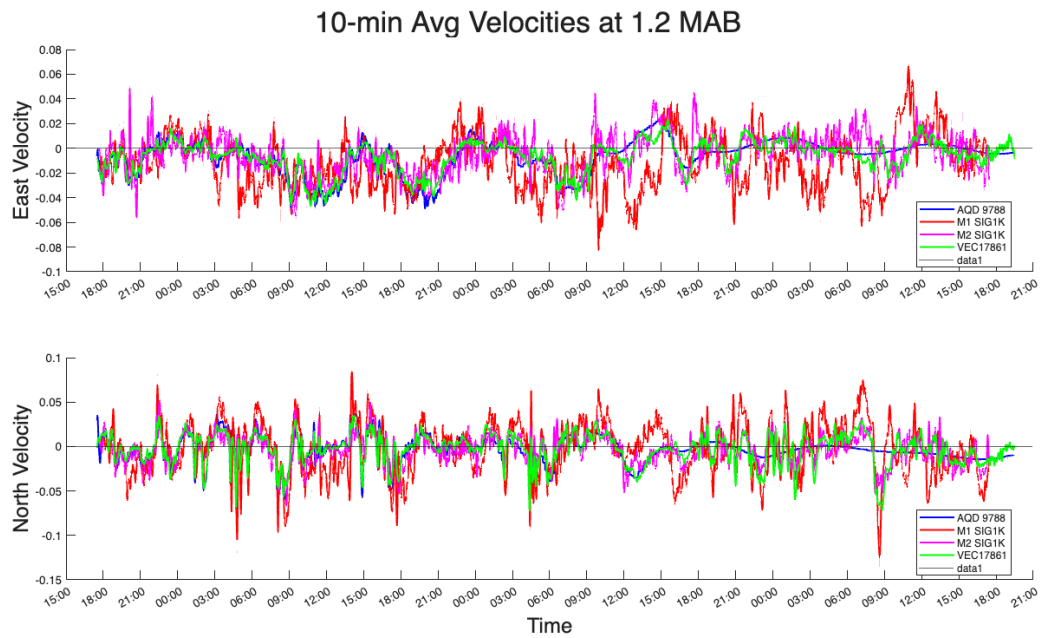
% Add legends
figure(1)
legend(ax1, h1, labels, 'Location', 'best')
legend(ax2, h2, labels, 'Location', 'best')
datetick(ax1); datetick(ax2)

% Set colors
for i = 1:length(dataCell)
    h1(i).Color = colors{i};
    h2(i).Color = colors{i};
end
yline(ax1, 0); yline(ax2, 0);

Warning: Excluding ColorBars, Legends and non-axes

```





end

EOF

Published with MATLAB® R2025a