ENCM 509 - Laboratory #1

Kyle Derby MacInnis - Mebrhatom Aneya

September 22, 2015

Abstract

The first lab of this course looked at the basics of loading data and displaying it in MATLAB as well as collecting biometric data from a tablet and using it to look at the statistical distributions of different signature variations.

1 Introduction

The lab focuses on signatures as the biometric data of interest, and using a WACOM tablet, ten signatures were collected from both group members and an additional five "forged" signatures were also collected from both members.

The data was collected using an application called SigGet whose main function is to collect and convert the measured information from the tablet and convert into an appropriate .MAT file. The parameters of interest which were collected included pressure, position, and time information.

These signatures make up the base of the data being analyzed in this laboratory, and upon the completion of the collecting process, the lab's next focus was on loading and utilizing the collected data within MATLAB's framework.

Once the data is in a convenient format for MATLAB to process it, performing calculations is easily achieved. For the purpose of this lab, the emphasis was on the collecting process and not on complex calculations. A normal distribution was used as the underlying model to compare the collected data and display it.

2 Procedure

1. Using SigGet collect 10 separate sets of signature data from each member. Save and store as individual .MAT files.
2. Grab an additional 5 sets of "forged" signatures for use later.
3. Using MATLAB and the guidelines in the lab manual, write a script to display the follow:
   - A colour-coded pressure map of the signature data.
   - A colour-coded velocity map of the signature data.
   - A 3D plot of the signature's pressure map.
   - The normal distributions of the velocity data.
4. Discuss any noticeable deviations from the expected outcomes, and give some conclusions on the data observed.

3 Results

1. Pressure Mapped Signatures

Real

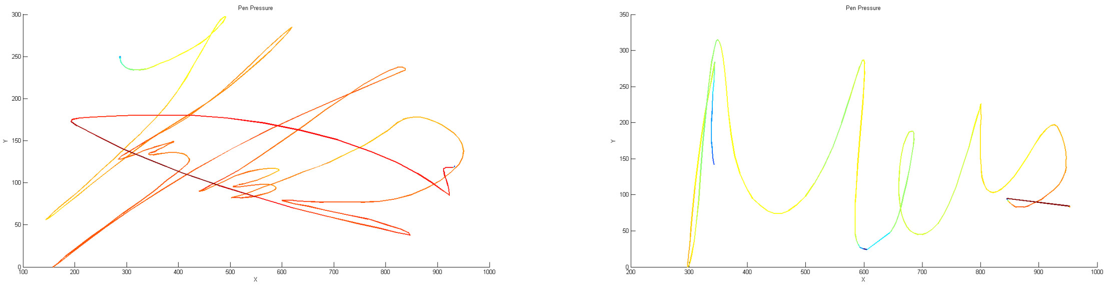

Figure 1: Example "Real" Pressure Mapped Signatures Captured

Forged

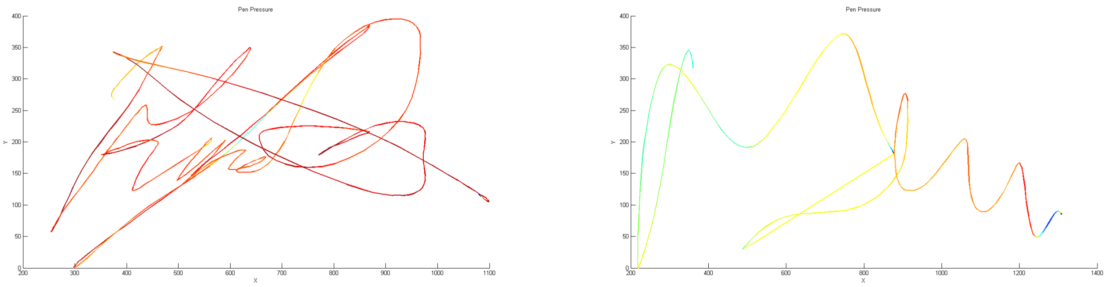

Figure 2: Example "Forged" Pressure Mapped Signatures Captured

2. Velocity Mapped Signatures

Real

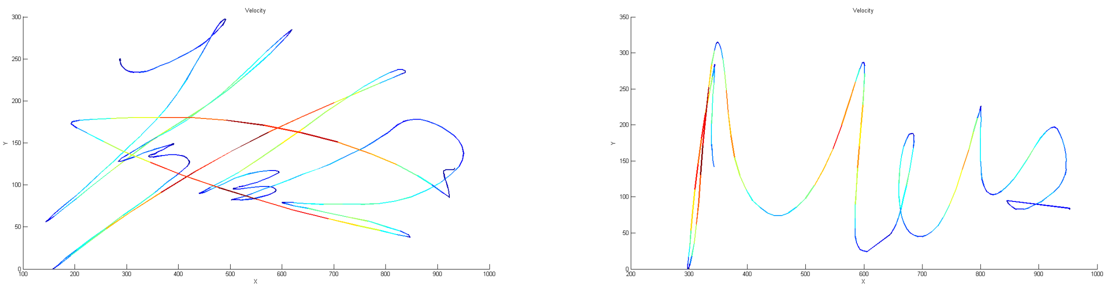

Figure 3: Example "Real" Velocity Mapped Signatures Captured

Forged

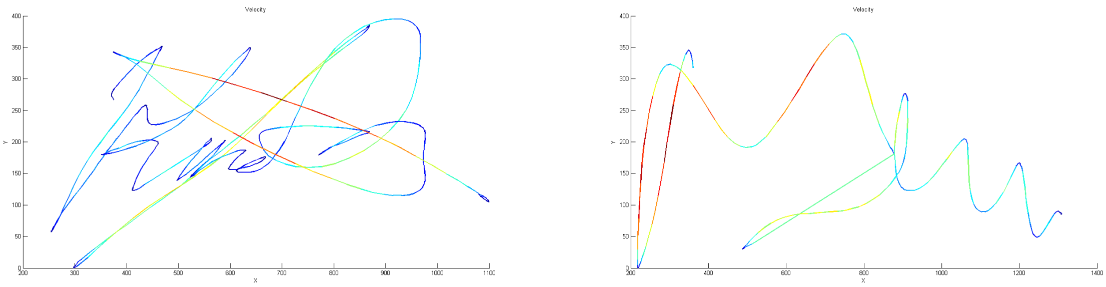

Figure 4: Example "Forged" Velocity Mapped Signatures Captured

3. 3D Pressure Graph

Real

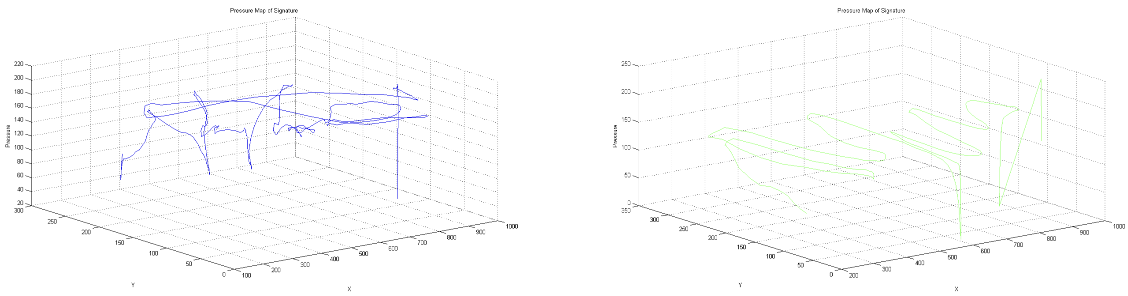

Figure 5: Example "Real" 3D Pressure Mapped Signatures Captured

Forged

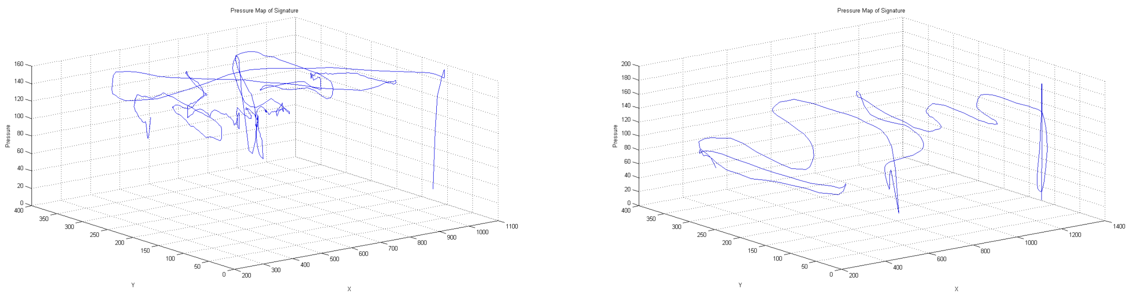

Figure 6: Example "Forged" 3D Pressure Mapped Signatures Captured

4. Statistical Evaluation

Real

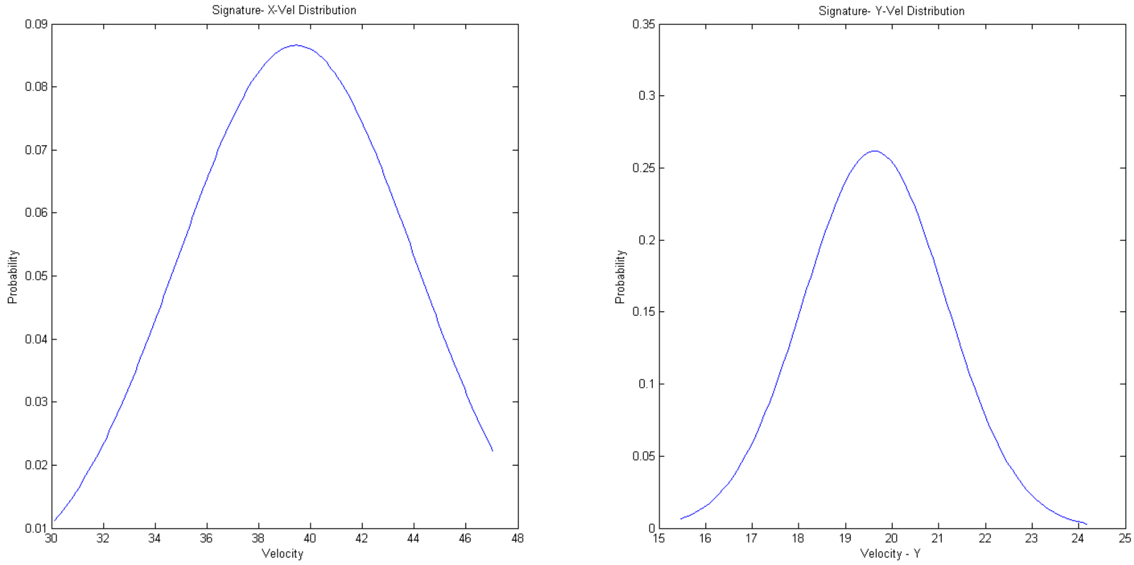

Figure 7: Example "Real" Statistical Distributions of K-Signature

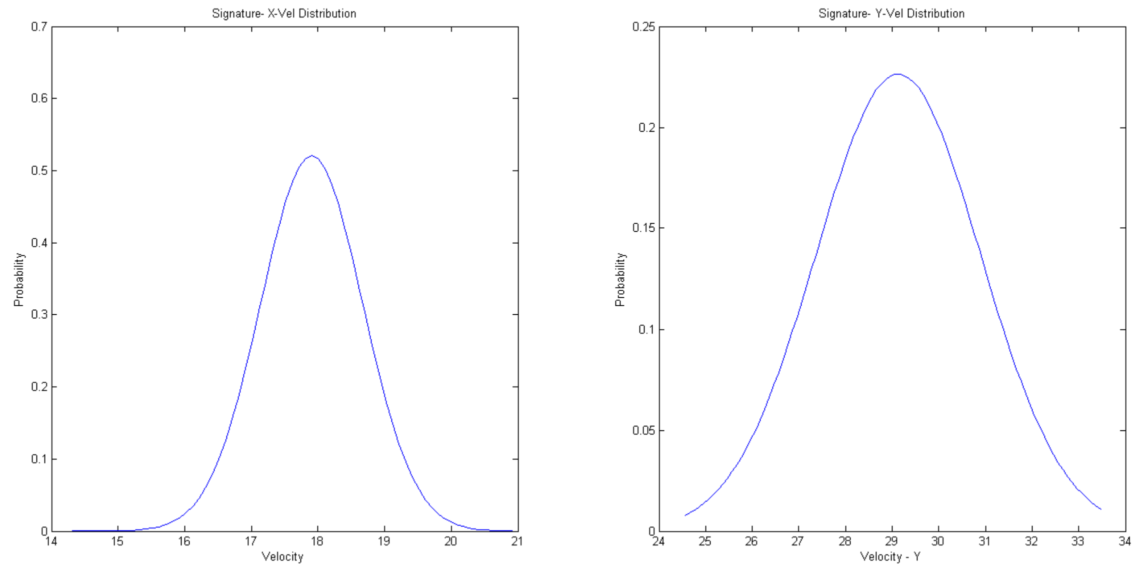

Figure 8: Example "Real" Statistical Distributions of M-Signature

Forged

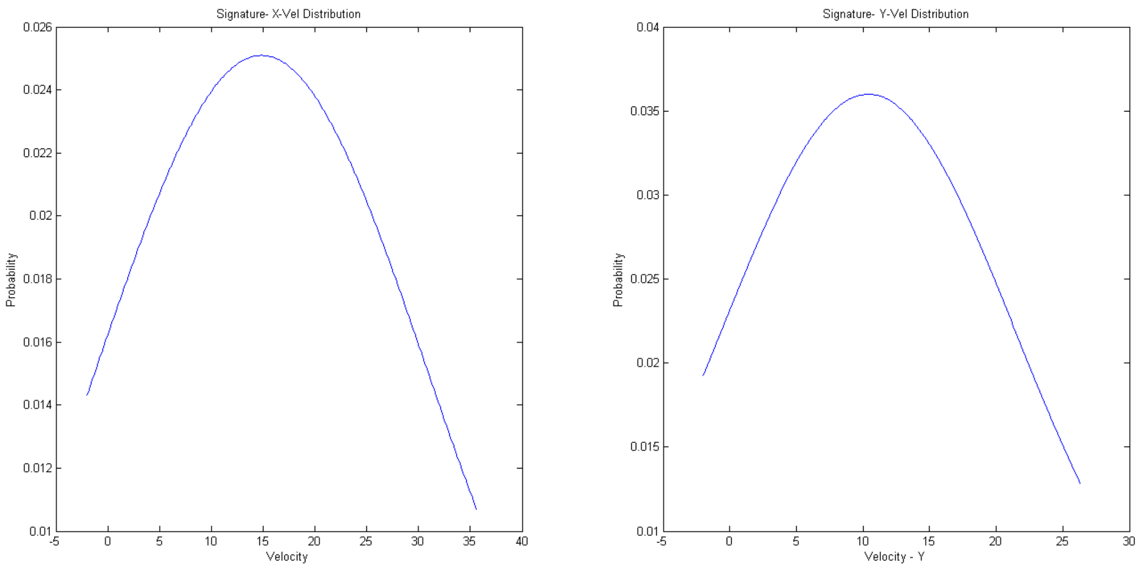

Figure 9: Example "Forged" Statistical Distributions of K-Signature

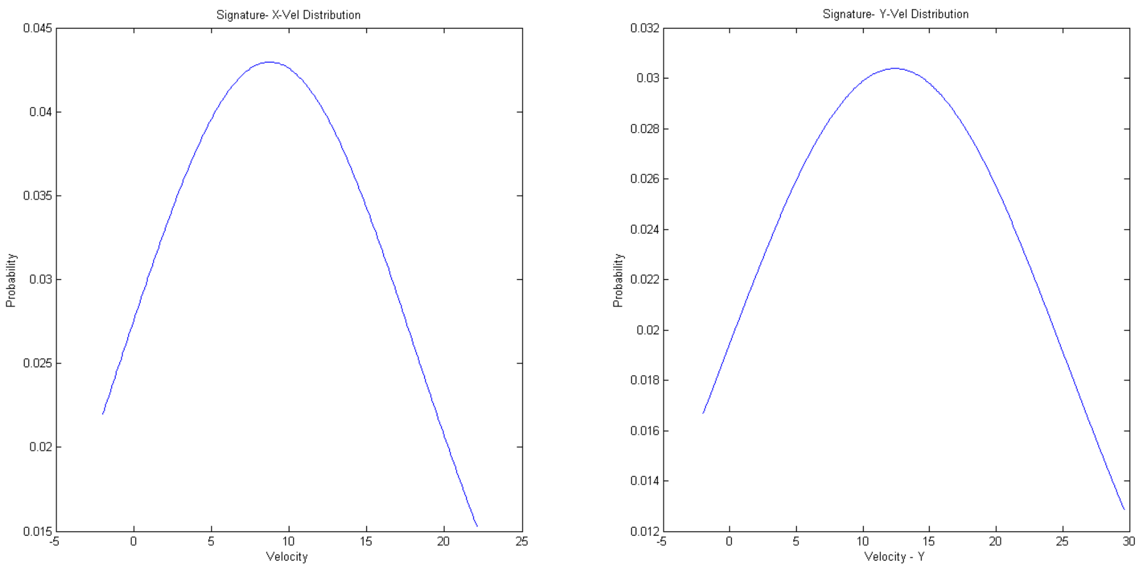

Figure 10: Example "Forged" Statistical Distributions of M-Signature

4 Discussion

- **Problem 1 - Issue with Loss of Contact**
  In order to correct for issues involving a loss of contact, we decided to throw in a conditional check via an If-Statement. Essentially, since MATLAB indexes its information starting with 1, we adjusted the pressure map index value to automatically search for *one (1)* in the event of a *zero (0)*.

- **Problem 2 - Issue with the colour bar**
  During operation of the lab exercises, we as a group were unable to make MATLAB show its colour

bars on any of the figures. This appears to be caused by some sort of path dependency issue as one solution online suggested it may be calling a different version of a built-in function which could be supplying the issue. That or the code needs to be updated to account for the newest features in MATLAB 2015a.

- **Statistical Distributions**
  After performing an average of the velocity in both the X- and Y- directions, the distributions were analyzed and compared amongst each other. The comparison made apparent the vast difference in the statistical distributions of the different signatures and showcased variations in individual tendencies to write quickly in a particular direction. The distributions did in-fact resemble normal Gaussian distributions and as such our assumption was justified.

- **Comparison between Real and Forged**
  Between our two sets of "real" signatures, the distributions were very much different, and upon comparing them to their corresponding "forged" version, they too were also different. This goes to show that statistical analysis would be a sufficient method of comparing and contrasting whilst allowing for variations within the individual signature data.

5 Conclusion/Remarks on the Lab

The lab for the most part went off without any problems and the results were as expected. We did run into issues with the colour bars which we were unable to resolve however we did resolve the issue of indexing and loss of contact. Overall, the lab was quite simple and was a nice refresher for MATLAB as well as providing a good starting point into biometric analysis.

6 Appendix A - MATLAB Code

```
function Full_Lab1()
% function Full_Lab1()
%
%   Performs the Calculations for Lab 1 and calls on minor function
%
%   Authors:    Kyle MacInnis & Mebrhatom Anenya
%   Date:       September 16, 2015
%   Course:     ENCM 509
%

% Close all Windows
close all;

% Make Empty Arrays for velocity, Coords, and Pressure
xVelocity = zeros(1,10,'double');
yVelocity = zeros(1,10,'double');

% xPressure = zeros(1,10,'double');
% yPressure = zeros(1,10,'double');
%
% xCoords = zeros(1,10,'double');
% yCoords = zeros(1,10,'double');

% Load and fill Data from Signatures
for i = 1:5
    % Example Signatures
    %filename = strcat('sy',num2str(i),'.mat');

    % Real Signatures
    %filename = strcat('kSig',num2str(i),'.mat');
    %filename = strcat('msig',num2str(i),'.mat');

    %Forged Signatures
    filename = strcat('ksigf',num2str(i),'.mat');
    filename = strcat('msigf',num2str(i),'.mat');

    % load basic Signature Data
    [COORDi, TIMEi] = Lab1(filename,1);

    % Separate into X and Y data
    coordX = COORDi(:,1);
    coordY = COORDi(:,2);
    % Calculate Range of Data
    maxSize = max(size(COORDi,1))-1;
    % generate empty velocity vectors
    xVel = zeros(1,maxSize,'double');
    yVel = zeros(1,maxSize, 'double');
    % loop through data
    for j = 1:maxSize
        xDiff = abs(coordX(j+1) - coordX(j));
        yDiff = abs(coordY(j+1) - coordY(j));
        timeElapsed = double(TIMEi(j+1) - TIMEi(j));
        % Calculate Directional Velocities
        xVel(j) = (xDiff/timeElapsed)+1;
        yVel(j) = (yDiff/timeElapsed)+1;
    end

    % Calculate the average
    xVelAvg = 0;
    yVelAvg = 0;
    for j = 1:maxSize
        xVelAvg = xVelAvg + xVel(j);
        yVelAvg = yVelAvg + yVel(j);
    end
    xVelocity(i) = xVelAvg/maxSize;
    yVelocity(i) = yVelAvg/maxSize;
end
% Scale up Velocities
xVelocity = xVelocity * 10;
yVelocity = yVelocity * 10;
```

```
%% Perform Statistical Analysis on Data
% X- Direction
RangeX = min(xVelocity)-2:0.1:max(xVelocity)+2;
MeanX = mean(xVelocity);
StdX = std2(xVelocity);
ProbX = normpdf(RangeX,MeanX,StdX);
% Y-Direction
RangeY = min(yVelocity)-2:0.1:max(yVelocity)+2;
MeanY = mean(yVelocity);
StdY = std2(yVelocity);
ProbY = normpdf(RangeY,MeanY,StdY);

% plot Statistical Distribution
figure('name','X-Direction Normal Distribution');
plot(RangeX,ProbX);
xlabel('Velocity');
ylabel('Probability');
tname = strcat('Signature','- X-Vel Distribution');
title(tname);

figure('name','Y-Direction Normal Distribution');
plot(RangeY,ProbY);
xlabel('Velocity - Y')
ylabel('Probability');
tname = strcat('Signature','- Y-Vel Distribution');
title(tname);
```

```
function [COORD, TIME] = Lab1(sigFile,n)
%% Lab 1 - Loads and Performs Analysis of Signatures
%
%   Authors:    Kyle MacInnis & Mebrhatom Anenya
%   Date:       September 16, 2015
%   Course:     ENCM 509
%
%
%   sigFile - filename (.mat file)
%   n - Display Figures or Not (For Full Lab File Run) (0 for figures)
%

% Close Figures
close all;

% Load in Signature Data
Sig1 = load(sigFile);
coord1 = double(Sig1.coord);
time1 = double(Sig1.time);
prs1 = double(Sig1.prs);

% return Time
TIME = time1;

% Reverse Direction of the Y Position (From Bottom Left)
coord1(:,2) = max(coord1(:,2)) - coord1(:,2);

% return Coordinates
COORD = coord1;

%% Plot 2D Signature Pressure
if n == 0
    figure('name','Signature - 1');
    pressureMap = colormap(jet(max(prs1)));
    % Connect Lines between each point
    for i = 1:(size(prs1,1)-1)

        if(prs1(i+1) == 0)
            prsMap = pressureMap(1,:);
        else
            prsMap = pressureMap(prs1(i+1),:);
        end
        line([coord1(i,1) coord1(i+1,1)], [coord1(i,2) coord1(i+1,2)], 'color', prsMap,
    'linewidth', 2);
```

```
    end
    xlabel('X');
    ylabel('Y');
    title('Pen Pressure');
end

%% velocity Calculations
% Create Empty Vector
vel = zeros(size(time1,1)-1, 1);
for i = 1:size(time1,1)-1
    % Find Euclidean Distance
    distance = sqrt((coord1(i+1,1)-coord1(i,1))^2 + (coord1(i+1,2)-coord1(i,2))^2);
    vel(i) = distance/(time1(i+1) - time1(i));
    vel(i) = int32((vel(i)*1000)+1);
end
vel = [1; vel];

%% Plot Velocity Map in 2D
if n == 0
    velmap = colormap(jet(max(vel)));
    figure('name', 'Velocity Map');
    for i = 1:size(coord1,1)-1
        vMap = velmap(vel(i+1),:);
        line([coord1(i,1) coord1(i+1,1)], [coord1(i,2) coord1(i+1,2)], 'color', vMap, '
    linewidth', 2);
    end
    xlabel('X');
    ylabel('Y');
    title('Velocity');
end

%% Plot Pressure in 3D
if n == 0
    figure('name', '3D Pressure');
    plot3(coord1(:,1), coord1(:,2),prs1, 'color', prsMap);
    xlabel('X');
    ylabel('Y');
    zlabel('Pressure');
    title('Pressure Map of Signature');
    grid on;
end
```