Jamie Cho

1. **Figures**

Calibration Curve
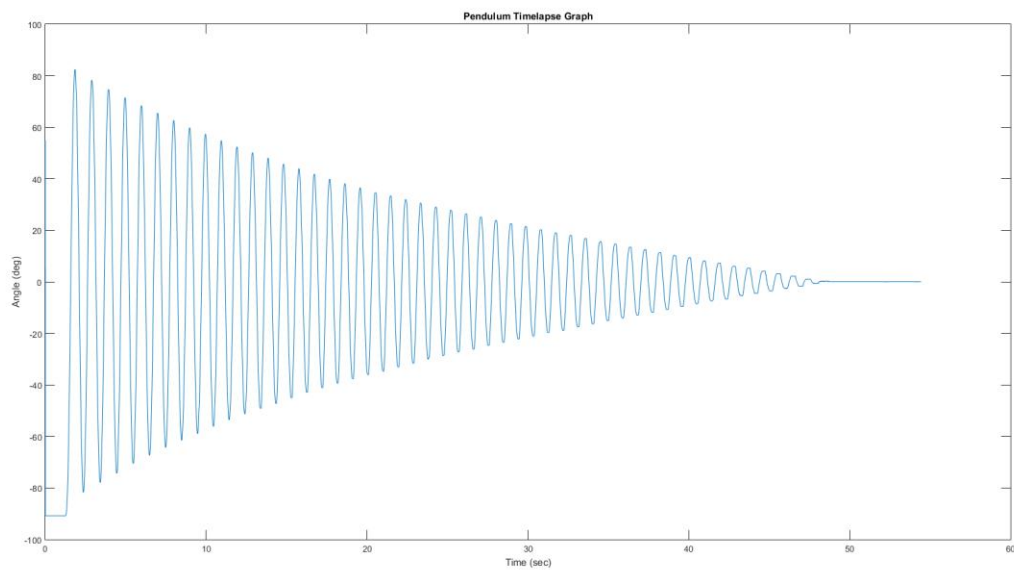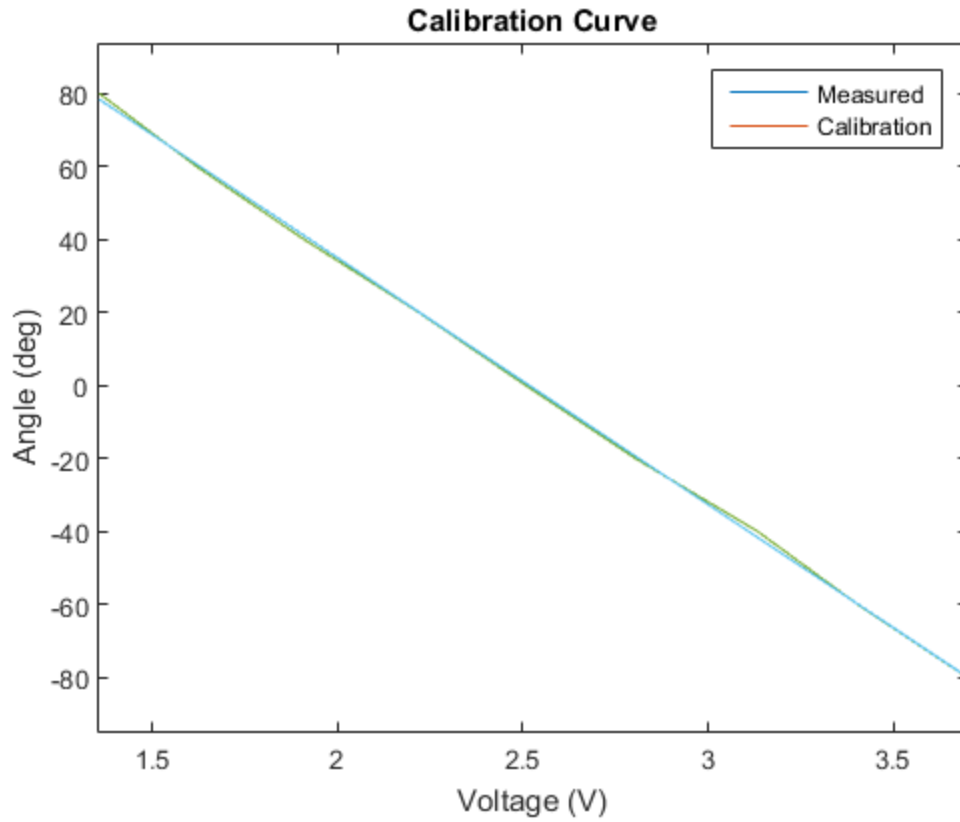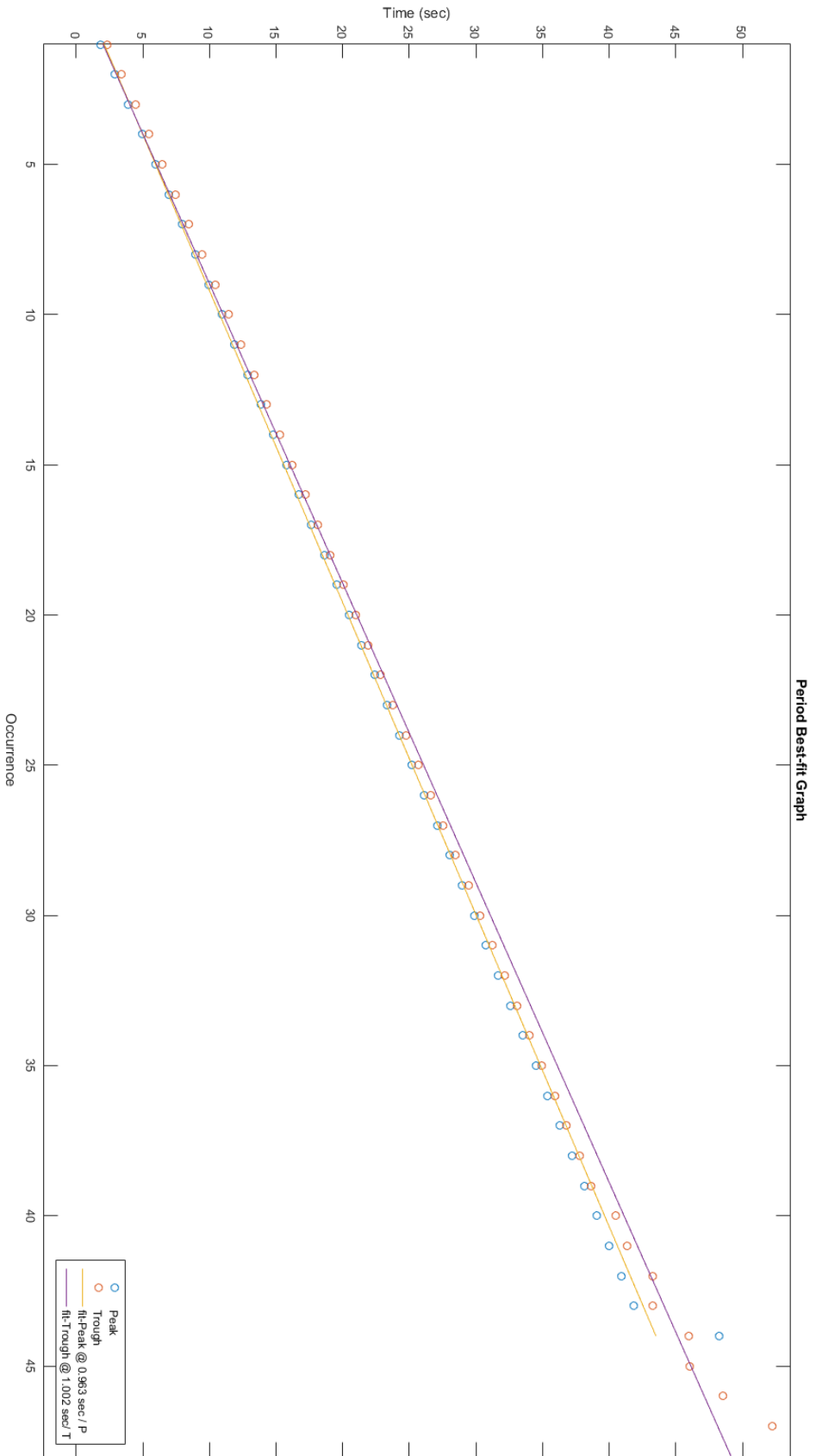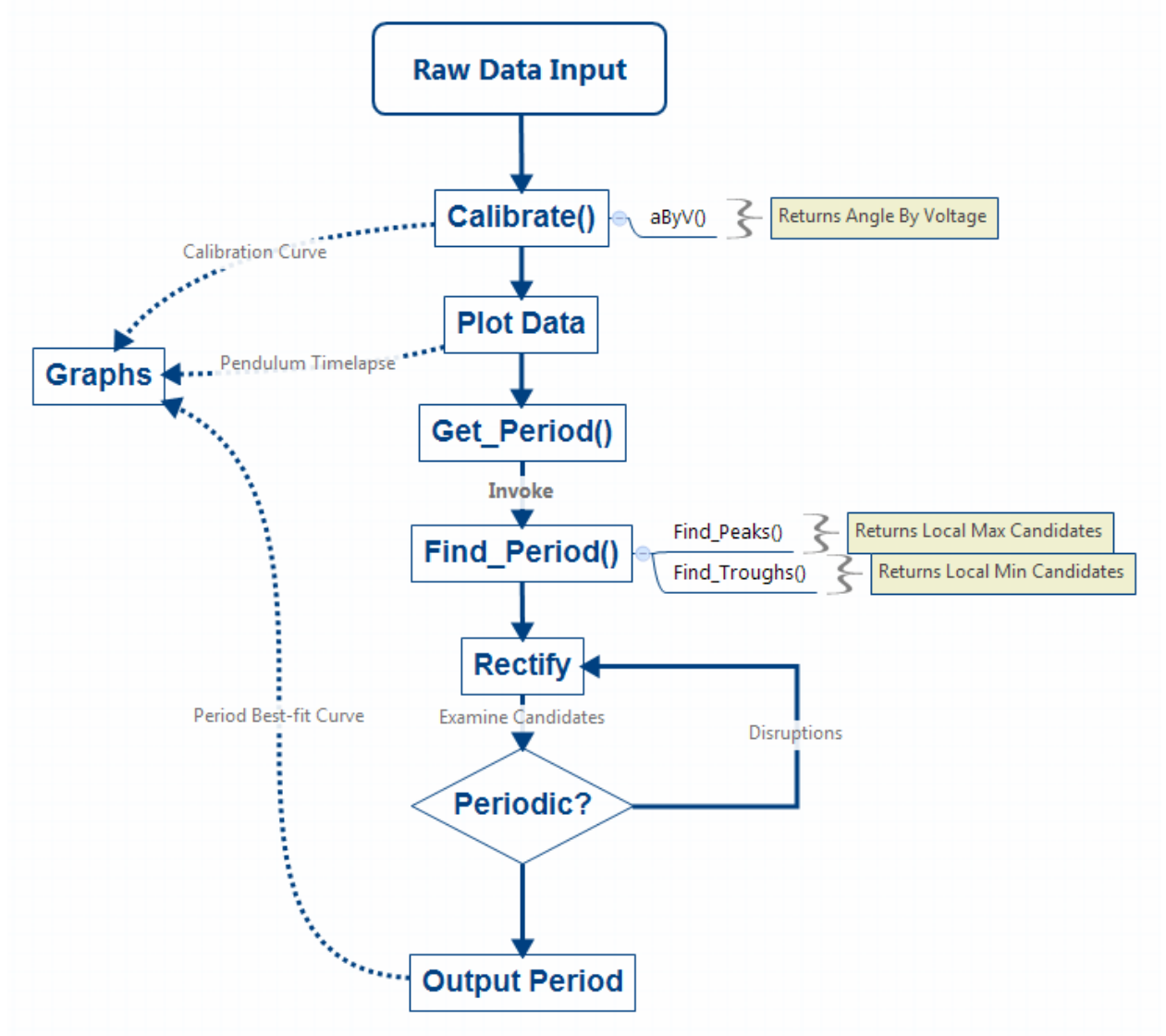

Pendulum Timelapse Graph

Period Best-fit Graph

2. Flowchart

3. MATLAB Command Line & Period Value

```
>> Calibration()
>>  plot(Times+30,aByV(C1V)); %Pendulum Timelapse Graph
>> get_Period()
Period is 0.963 seconds(Peaks), 1.002 seconds(Troughs)
 Ave. to 0.982 seconds
```

4. MATLAB Scripts

Calibration.m
```
%precondition : Cal_Voltage & Cal_Angle are imported Data

c_fit = polyfit(Cal_Voltage, Cal_Angle, 1);
plot(Cal_Voltage, Cal_Angle);
```

```matlab
 hold on
aByV = @(x)x*c_fit(1) + c_fit(2); %Angle By Voltage
ezplot(aByV, Cal_Voltage);
legend('Measured','Calibration');
title('Calibration Curve')
xlabel('Voltage (V)');
ylabel('Angle (deg)');
```

get_Period.m

```matlab
%precondition : C1V is the raw Voltage, Times range in -30:30
%p = Peak, t = Trough
 [p,t] = find_Period(aByV(C1V),Times+30);


flag = true;

while(flag) %rectification feedback loop
    flag = false;
    scale_p = [1:length(p)];
    scale_t = [1:length(t)];

    p_fit = polyfit(scale_p,p,1);
    t_fit = polyfit(scale_t,t,1);

    i = 2;
    while(i < length(p))
        if( p(i+1) - p(i-1) < p_fit(1)*5/4) %in discord with other data
            p(i) = [];
            flag = true; %indicates change
        else
            i = i+1;
        end
    end

    i = 2; %repeat same process with troughs
    while(i < length(t))
        if( t(i+1) - t(i-1) < t_fit(1)*5/4)
            t(i) = [];
            flag = true;
        else
            i = i+1;
        end
    end

end %ends when nothing is out of place


plot(scale_p,p,'o');
hold on
plot(scale_t,t,'o'); %raw periods

fit_func_p = @(x) x*p_fit(1) + p_fit(2);
ezplot(fit_func_p,scale_p);

fit_func_t = @(x) x*t_fit(1) + t_fit(2);
```

```matlab
ezplot(fit_func_t,scale_t); %polynomials from period data

p_label = sprintf('fit-Peak @ %.3f sec / P', p_fit(1));
t_label = sprintf('fit-Trough @ %.3f sec/ T', t_fit(1));

l = legend('Peak','Trough',p_label,t_label, 'Location','SouthEast');
title('Period Best-fit Graph');
xlabel('Occurrence');
ylabel('Time (sec)');

msg = sprintf('period is %.3f seconds(Peaks), %.3f seconds(Troughs)\n Ave.
to %.3f seconds', p_fit(1), t_fit(1), (p_fit(1)+t_fit(1))/2);
disp(msg);
```

find_Period.m
```matlab
function [p,t] = find_Period(dat,Time)
p = find_Peaks(dat,Time);
t = find_Troughs(dat,Time);
end


function ret = find_Peaks(dat, time)
r = 20; %range +-. at the sampling rate of 60/7260 (~0.008) seconds/sample,
this amounts to +- 0.165 seconds.
ret = [];
len = length(dat);
prevval = 0;
    for val = 1+r:len-r
        if(abs(prevval -dat(val)) < 0.2) %ignore plateaus at top, leeway of
0.2 degrees difference.
            continue;
        else
            prevval = dat(val);
        end
        if(isLocalMax(dat(val), dat(val-r:val+r))) % = if Peak
            ret(end+1) = time(val); %append new Peak
            prevval = dat(val);
        else
            %disp(val);
        end
    end
end


function ret = find_Troughs(dat, time) %ditto
r = 20;
ret = [];
len = length(dat);
prevval = 0;
    for val = 1+r:len-r
        if(abs(prevval -dat(val)) < 0.2)
            continue;
        else
            prevval = dat(val);
        end
        if(isLocalMin(dat(val), dat(val-r:val+r))) % = if Trough
            ret(end+1) = time(val); %append new Trough
```

```matlab
                prevval = dat(val);
            else
                %disp(val);
            end
        end
    end


    function ret = isLocalMax(val, arr)
        ret = true;
        for dat = arr'
           if(dat > val)
                ret = false;
                break;
           end
        end

        if(arr(1) == val || arr(end) == val)
            ret = false;
        end

    end
    function ret = isLocalMin(val, arr)
        ret = true;
        for dat = arr'
           if(dat < val)
                ret = false;
                break;
           end
        end

        if(arr(1) == val || arr(end) == val)
            ret = false;
        end

    end
```