# A LabVIEW Filter for Calculating Mean Arterial Pressure

Skyler Blair, Payton J Thomas

BME 4101 Monday Afternoon Section
Lab 3 Report
2022-09-27

## 1 Abstract

To minimize blood flow without compressing arterial plaques, intra-arterial balloons must track and respond to mean arterial pressure. This task is nontrivial, however, because mean arterial pressure undergoes drastic but transient variation as the heart beats. A device which can track exclusively slow variations in mean arterial pressure is therefore necessary. This project developed a rolling mean filter in LabView to address this need. Verification and validation experiments demonstrated that this filter operates well under physiological conditions. In the future, this filter may be used in conjunction with a pneumatic pump to safely control intra-arterial balloons.

## 2 Introduction and Specifications

The main objective of this project was to develop a Mean Arterial Pressure (MAP) system that could calculate the changing MAP over an adjustable time window while transmitting the calculated MAP values to a pneumatic pump control. The goal is to use this MAP tracker while a patient is walking on a treadmill and performing a stress test. A catheter-based pressure sensor with a 10mV voltage output signal for every 1mmHg input is used. The input signal generates 1mmHg for every 10mV. This must be taken into account when designing the MAP system tracker. This tracker must be able to calculate and display input and output values in current time. As the pressure sensor changes as the patient's heartbeats change, the pneumatic pump must get the transmitted MAP values.

Mean arterial pressure is the average blood pressure which is dependent on cardiac output and the vascular resistance within the heart. In this project, the patient will be exerting energy by walking on the treadmill. As energy is exerted, the MAP will increase as the cardiac output and heart rates increase. To create the MAP system tracker, the MAP calculations must create an input and output signal in real time while taking an increasing MAP into account. LabVIEW was used to create the MAP tracker with data acquisition software. Within this software, objectives were tested. To verify the mean arterial pressure system. These included:

1. Verify that the primary AI channel can sample signals at 100Hz with a 5mV resolution over $\pm$5V.

2. Verify that the MAP can be calculated from the last 0.01-10 seconds.

3. Verify that the reported MAP calculation has a 5mV resolution over a $\pm$5V range and it is delayed less than 20ms from the input signal.

To validate the MAP calculations, heartbeats of 60, 90 and 180 beats per minute (bpm) were simulated and the $N$ value was determined by finding the most consistent MAP for each bpm. This report will contain how the MAP system tracker will contain how the system was implemented, verified and validated. The MAP tracker results will be stated and how the system took in the pressure sensor changes and commanded a pneumatic actuator to change the pressure of an arterial balloon.

## 3 Implementation and Verification

### 3.1 Implementation

The filter is a virtual instrument (VI) implemented in LabVIEW. The VI is intended to receive input from an arterial blood pressure sensor via analog input channel AI, but all input signals considered in this report were artificially generated using a function generator. The processed signal is output via analog output channel AO0 and routed back to the VI for analysis via analog input channel AI1. In practice, this output signal will be routed to the control of a pneumatic pump which in turn will control intra-arterial balloon inflation.

The VI saves all three signals to a local .lvm file as well as displays them on a chart on the LabVIEW front panel.

Within LabVIEW, the VI calculates the output signal $y_n$ as the arithmetic mean of $N$ (selected by the user on the LabVIEW front panel) previous values of the input signal $x_n$. That is,

$$y_n = x_n + x_{n-1} + \cdots + x_{n-N+1}.$$

The filter is therefore linear and time-invariant (LTI) with impulse response

$$h_n = \delta_n + \delta_{n-1} + \cdots + \delta_{n-N+1}.$$

This impulse response yields transfer function

$$H(\hat{\omega}) = \mathscr{F}(h_n) = \sum_{k=0}^{n-N+1} e^{-jk\hat{\omega}},$$

that is, the VI functions as a simple low-pass filter to remove the relatively high-frequency oscillations of MAP with each heartbeat. The control diagram for the filter implementation in LabVIEW and the front panel are included in Appendix A. The VI, all recorded data, all analysis scripts, and all figures are available on GitHub. All MATLAB scripts are also available on GitHub and in Appendix B.

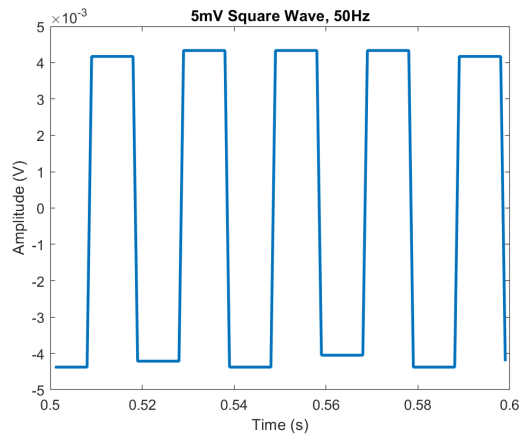
## 3.2 Verification

The VI was tested against its specifications in three verification experiments. Firstly, the frequency, amplitude, and resolution capabilities of the analog input channel was tested (Subsubsection 2.2.1), secondly, the accuracy and precision of the MAP filter was tested for various moving window lengths (Subsubsection 2.2.2), and thirdly, the fidelity of the output signal to the calculated MAP (amplitude and phase delay) was characterized (Subsubsection 2.2.3).

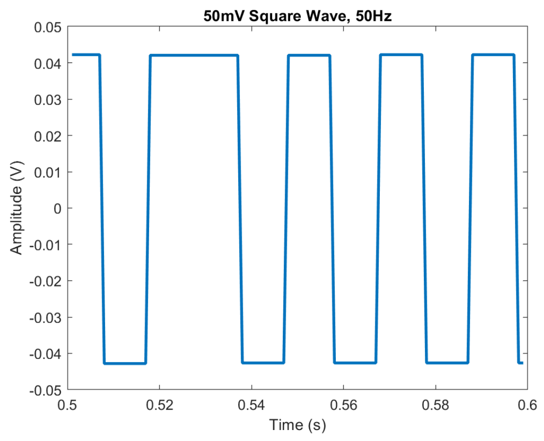
### 3.2.1 BP Testing

To test the analog input capabilities of the system, a function generator produced 50Hz square waves with amplitudes 5mV, 50mV, 500mV, and 5V (Figure 1). Parameters set on the function generator were assumed to describe the true waveform. Fourier analysis of the LabVIEW record of each signal yielded amplitude spectral densities (ASDs) of each signal (Figure 2). ASD plots were used to determine the principal frequency of the recorded signals (Table 1). Likewise, the amplitude of each recorded waveform (from midline) was calculated (Table 1).

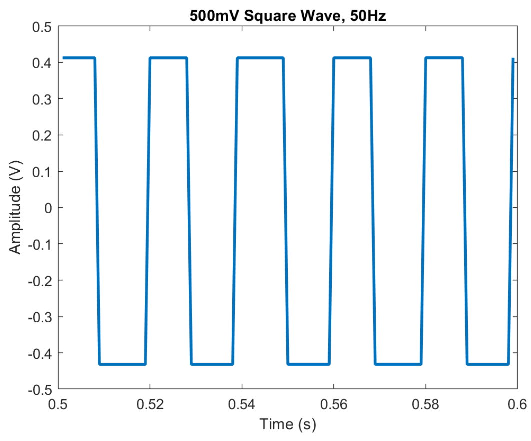
Inspection of these ASDs revealed that the VI resolves frequencies (the odd harmonics of the square wave) well beyond the 50Hz (Nyquist) limit of the 100Hz specification (Figure 2). The results also demonstrated that the VI is capable of recording voltages at and beyond the 5V specification (Figure 1, Table 1), although the recorded voltages typically ran low. Recorded amplitudes were all low, but all within 1.5 standard deviations of the intended value. Likewise, recorded frequencies were all within 2% of the intended value. The results



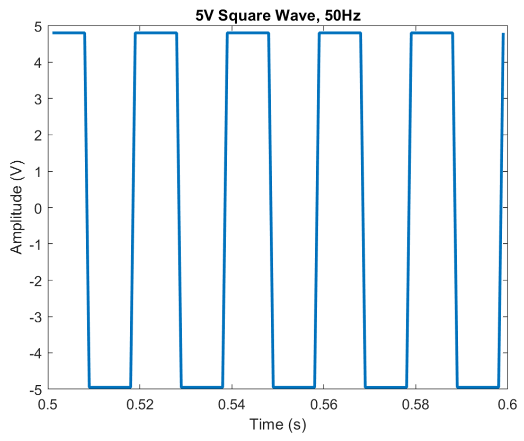
(a) 5mV case



(b) 50mV case



(c) 500mV case

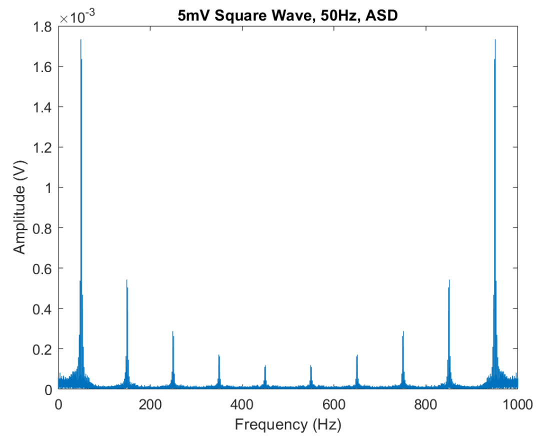


(d) 5V case

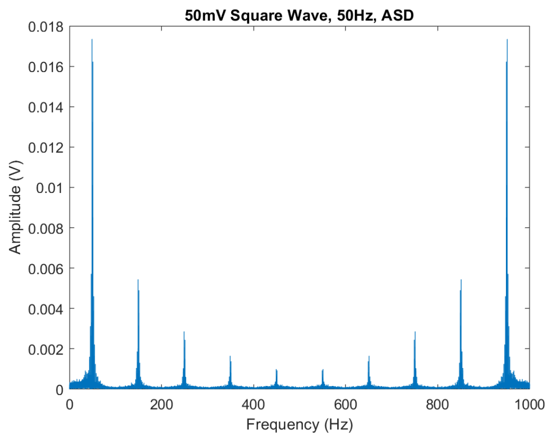
Figure 1: Waveforms produced by the function generator as recorded through the analog input channel.

| Intended Amplitude (V) | Intended Frequency (Hz) | Recorded Midline (V) | Recorded Amplitude (V) | Recorded Frequency (Hz) |
|---|---|---|---|---|
| 0.005 | 50 | -0.0012 | 0.0042± 0.0005 | 49.1 |
| 0.05 | 50 | 0.0000 | 0.0420±0.004 | 49.1 |
| 0.5 | 50 | 0.0202 | 0.4067±0.07 | 49.1 |
| 5 | 50 | 0.0752 | 4.7815±0.5080 | 50.9 |

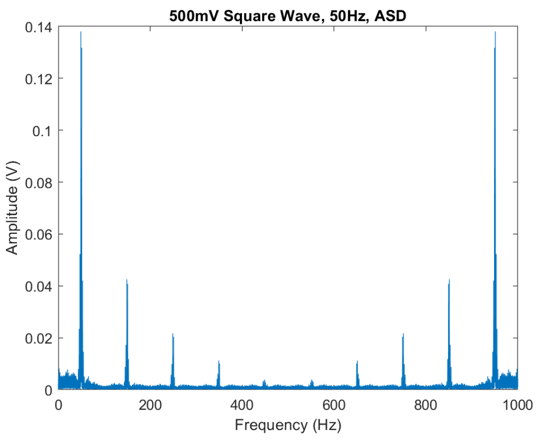
Table 1: Verification of analog input channel. Amplitudes ran low in all cases, but signals were otherwise replicated with high fidelity in software through the analog input channel.



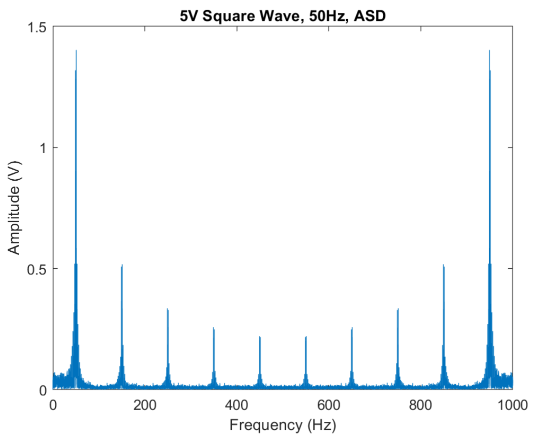
(a) 5mV case



(b) 50mV case



(c) 500mV case



(d) 5V case

Figure 2: Spectra of input test waves, calculated in MATLAB. Note that frequencies well beyond 50Hz are resolved, indicating that the input channel samples at well beyond 100Hz.

also showed that the VI can resolve at the 5mV level because the midline of -0.0012V in the 5mV case is statistically significantly different from the recorded amplitude of $(0.0042 \pm 0.0005)$V. This indicates that the 5mV wave was distinguishable from no signal at all.

### 3.2.2 MAP Testing

To test the ability of the system to calculate MAP, a function generator produced a 0.5Hz sine wave with amplitude 1V. The VI calculated MAP within this setup as the arithmetic mean of the previous $N$ recorded values of arterial blood pressure for $N = 10, 100, 1000, 10000$. A replicated filter in MATLAB (Appendix B) yielded similar MAP results to the LabView (Figure 3). Near the beginning of each signal, the waveforms differ because of differences in how MATLAB and LabVIEW apply a rolling mean when the total length of received signal is less than $N$; this behavior is ultimately of little consequence, as the filter should not be applied to intra-arterial balloon pneumatic control until sufficient data has been collected to

| Input Amplitude (V) | Delay (s) | Offset (V) | Gain (V) |
|---|---|---|---|
| 0.005 | 0.01 | -0.0015 | 0.9765 |
| 0.05 | 0.01 | -0.0015 | 0.9988 |
| 0.5 | 0.01 | -0.0017 | 1.0048 |
| 5 | 0.008 | 0.0010 | 1.0332 |

Table 2: Verification of analog output channel. A consistent offset of around 1.5mV was observed. Gains near unity indicate high amplitude-fidelity in the output signal. All delays are well within the 20ms specification.

apply the filter. These behaviors sufficiently demonstrate that the filter is a rolling window mean filter with window size between 10ms and 10s, per specifications.

As expected, increasing $N$ resulted in attenuation of the signal, with the $N = 10$ case showing no noticeable change in amplitude and the $N = 10000$ case reducing the amplitude by an order of magnitude. Also as expected, increasing $N$ delayed the calculated MAP signal relative to the measured arterial blood pressure. Again, the $N = 10$ case exhibits no noticeable phase shift, whereas the $N = 10000$ case exhibits a delay of around one quarter-period. (Figure 4). Note that the oscillatory behavior of the input signal would be fully attenuated if $N$ were selected such as to 'cover' an integer number of full periods, that is, if

$$N = k\frac{f_{sampling}}{f_{signal}},$$

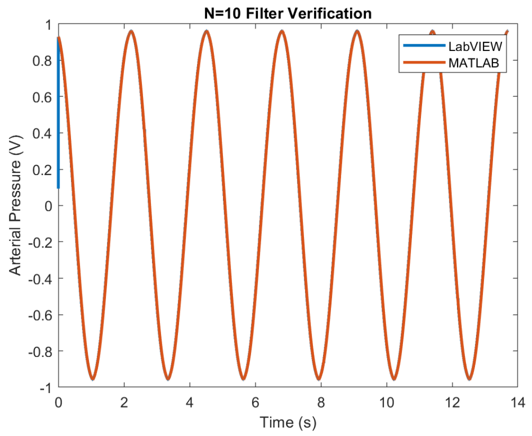
For some $k \in \mathbb{Z}$ where $f_{sampling}$ and $f_{signal}$ are the frequencies of sampling and the signal, respectively.

### 3.2.3 Output Testing

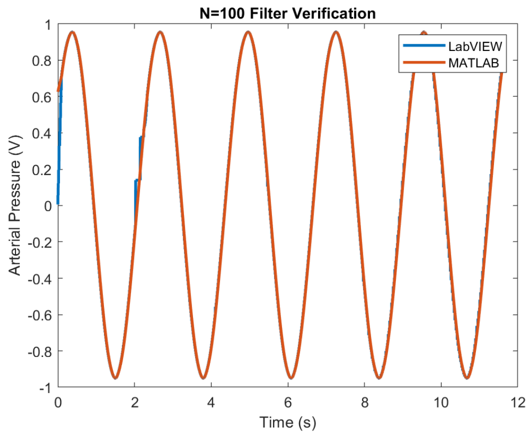
To test the capacity of the system to output a control signal to a pneumatic pump, a function generator produced 5Hz triangle waves with amplitudes 5mV, 50mV, 500mV, and 5V. These signals were treated as inputs to the VI with parameter $N = 10$. The resultant MAP estimation was then output via analog output channel AO0 and immediately read back in via analog input channel AI1. In each case, this recording was compared to the calculated MAP signal to determine the analog output capabilities of the system.

In each case, the mean value of both signals was subtracted off so that amplitudes could be directly compared. The 'offset' is given by the difference of these two means. The MATLAB function `finddelay` was then used to determine the most likely time delay between the output signal and the calculated MAP. The signals were then shifted by this delay such that they align, and the mean multiplicative gain (output signal / input signal) was calculated (Table 2). Multiplicative gain need not be applied when delay is calculated because cross correlation is maximized when the signals maximally align, regardless of whether or not they have the same amplitude.

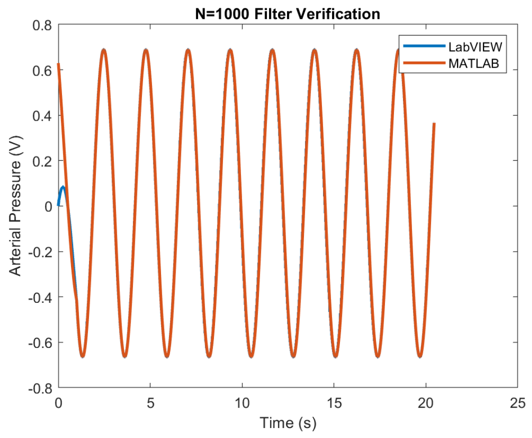
These results indicate that the analog output channel typically produces a voltage waveform about 1.5mV lower than expected, but with no appreciable difference in amplitude.



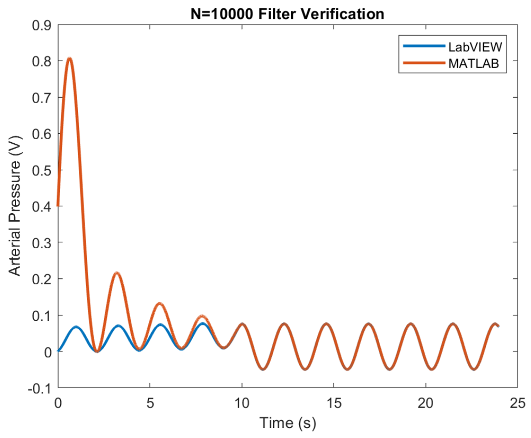
(a) $N = 10$ case



(b) $N = 100$ case

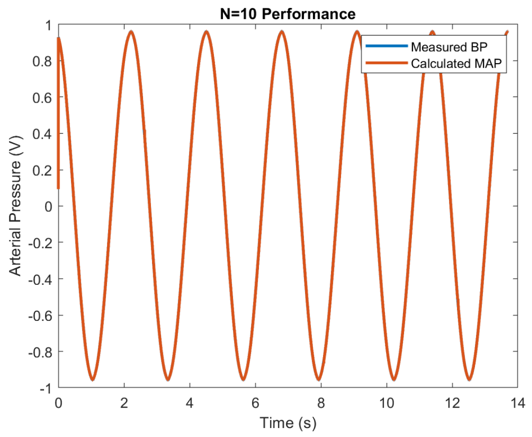


(c) $N = 1000$ case

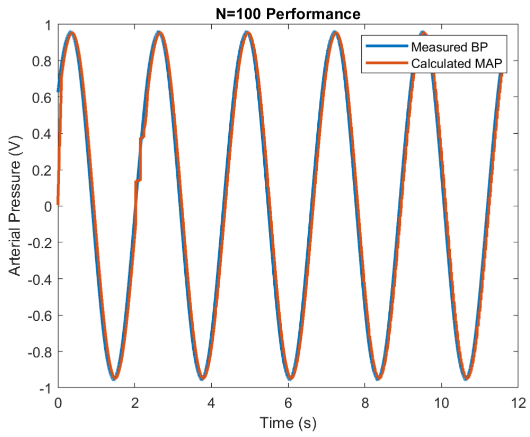


(d) $N = 10000$ case

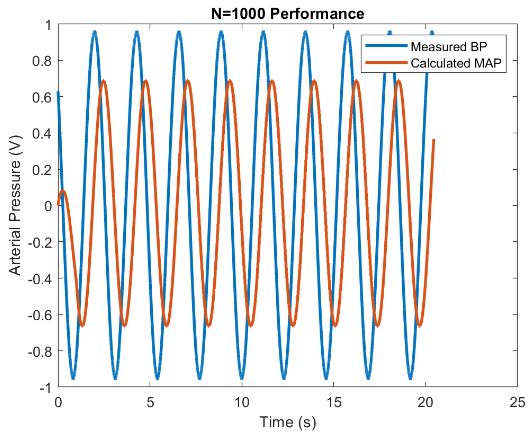
Figure 3: Comparison of MATLAB-calculated moving window means to LabVIEW-calculated moving window means. Transient differences are observed for the first few seconds of signal due to differences in how short signals are handled in each implementation, but these differences are not relevant to the intended application of the filter.



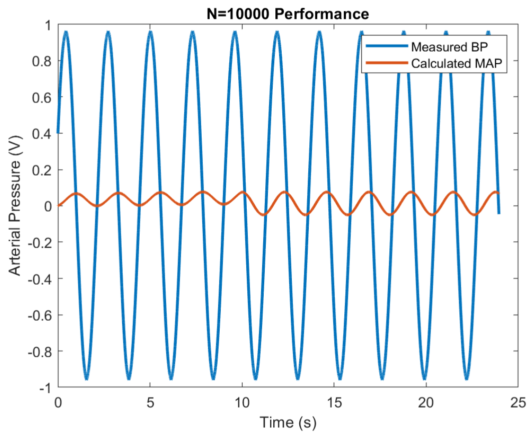
(a) $N = 10$ case



(b) $N = 100$ case



(c) $N = 1000$ case



(d) $N = 10000$ case

Figure 4: Performance of the filter as the number of points considered varies. Increased $N$ increases robustness to oscillations, though and ideal $N$ would cover an integral multiple of the wave period.

|  | Specification | Verified | Met? |
|---|---|---|---|
| Input Resolution | 5mV | 5mV | True |
| Input Range | ±5V | ±5V | True |
| Input Sampling Rate | 100Hz | 1000Hz | True |
| Min Window Time | 10ms | 10ms | True |
| Max Window Time | 10s | 10s | True |
| Output Resolution | 5mV | 5mV | True |
| Output Range | ±5V | ±5V | True |
| Output Delay | 20ms | 10ms | True |

Table 3: Verification of analog output channel. A consistent offset of around 1.5mV was observed. Gains near unity indicate high amplitude-fidelity in the output signal. All delays are well within the 20ms specification.

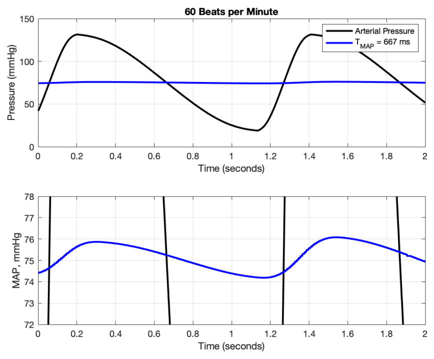
They also show that the analog output channel delays the signal by at most 10ms in all cases, satisfying the 20ms specification. The ability of the output channel to produce 5mV and 5V waveforms with no appreciable change in amplitude demonstrates that the analog output can produce waveforms with 5mV resolution over a $\pm 5V$ range, again satisfying specifications. Future iterations of the filter should adjust for the observed 1.5mV offset, however. Verification results are summarized in (Table 3)

## 4 Product Validation & Results

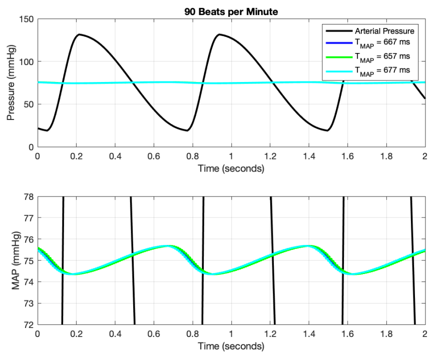
This project was simulated with LABView as intra-arterial catheters were not available in the lab to validate the mean arterial pressure system tracker. A function generator was symbolic for the patient. To validate the MAP tracker, a fixed systolic/diastolic pressure was set to 150/50mmHg and an N of 10,000 was used.

A patient model of 90 bpm was first simulated by manipulating the function generator. The function generator was set to construct a sine wave of 1.5Hz with a peak to peak amplitude of 1.0V. A DC offset was set to 1.0V. Before conducting the experiment, a consistent MAP at N = 667 was determined. As seen in Figure 5Ai, the mean arterial pressure within the heart at 90 bpm is approximately 75.5mmHg. Three different durations were tested by manipulating the data to account for 10 ms of data points in the past, and in the future. As seen in Figure 5Aii, the time duration of 676 ms, the MAP is consistent with the heartbeat interval.

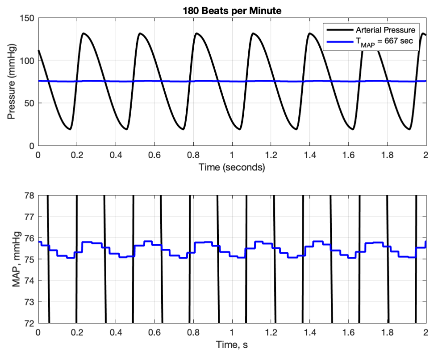
The second patient model that was simulated was a patient with a 180 bpm heart rate. This heart rate mimics a patient conducting exercise on a treadmill. The same function generator parameters except the peak to peak amplitude of 3 Hz was used. This was to simulate the patient exercising on the treadmill to increase heart rate. As seen in Part Bi, the Mean Arterial Pressure is approximately 75mmHg. This demonstrates that the change in heartbeat does not alter how the MAP tracker measures pressure. Because the MAP is calculated by the change in cardiac output and vascular resistance, the same pressure is expected. Therefore, the MAP tracker is validated to accommodate the user needs. As the heartbeat doubles in frequency, the time duration will remain the same at 667 ms as the MAP



(a) 60bpm case



(b) 90bpm case



(c) 180bpm case

Figure 5: Product Validation by simulating patients with a heart rate of A) 90, B) 180 and C) 60 bpm. Figures show how the MAP tracker system responded to varying heart rates while outputting the MAP. Part i portions are an expanded view, and Part ii shows a zoomed in view of Part i. Time durations are seen for each heart rate to determine validation.

is constant over any variation of periods. The last patient model that was used to validate the MAP tracker was to simulate a patient with a heart rate of 60bpm. This was to simulate a patient at rest. The function generator peak to peak amplitude was changed to 1Hz. As seen in Figure 5Ci, the MAP at a time duration of 667 ms remained at approximately 75mmHg. This was consistent with what was determined previously. Although the heart beat frequency changed, the MAP was not altered.

By simulating a patient's heart beat at 90, 180 and 60bpm, it can be seen that the MAP tracker can track and calculate MAPs over varied durations. Because an N of 10,000 was used within the determination of the results, it took time for the past data points summation to get rid of old values. This is something to keep in mind as data points taken at the beginning of data collection need to be disregarded. Overall, this data validated the MAP tracker when using a N of 10,000. When a smaller value of N was used, the MAP was not accurately displayed; this is a limitation that will need to be incorporated into a final design of the MAP tracker.

## 5 Discussion & Conclusion

This lab project's goal was to create a mean arterial pressure system tracker that calculates the changing MAP over an adjustable time window. This rolling mean filter in LabView will transmit calculated MAP values to a pneumatic pump control while also displaying input and output signals in real time. Verification experiments revealed that input range, resolution, and sampling rate; maximum and minimum rolling window length; and output resolution, range, and delay were all within specifications (Table 3).

Validation testing confirmed that the MAP tracker can track and calculate mean arterial pressures within a simulated patient while heartbeat frequency changes. It was determined that as a simulated patient's heart rate varied from 60 to 90 to 180 bpm, the MAP did not vary. However, a limitation of this MAP tracker is the time duration or N value used to calculate the MAP. If too small of a time duration window is used, the MAP will have more variability when calculating the MAP; this makes the MAP inaccurate. This is a key consideration when finalizing this MAP tracker for future use. Improvements should be made to this MAP tracker before it is released into the market. Improvements include designing a MAP tracker that can account for a change in heart rate and respond with carrying time duration values to calculate the MAP in real time. A heart rate monitor should be paired with this system tracker as well.

This MAP tracker system was created with a primary goal of maintaining the pressure on an intra-arterial balloon. By using an appropriate time duration window, the mean arterial pressure will be accurately tracked and calculated in real time. This MAP tracker design met verification and validation testing given the task of the project. Although there was success in the design of the MAP tracker, there still needs to be improvements before a health care provider can use this device in the medical field to maintain a patient's intra-arterial balloon pressure at their local MAP.

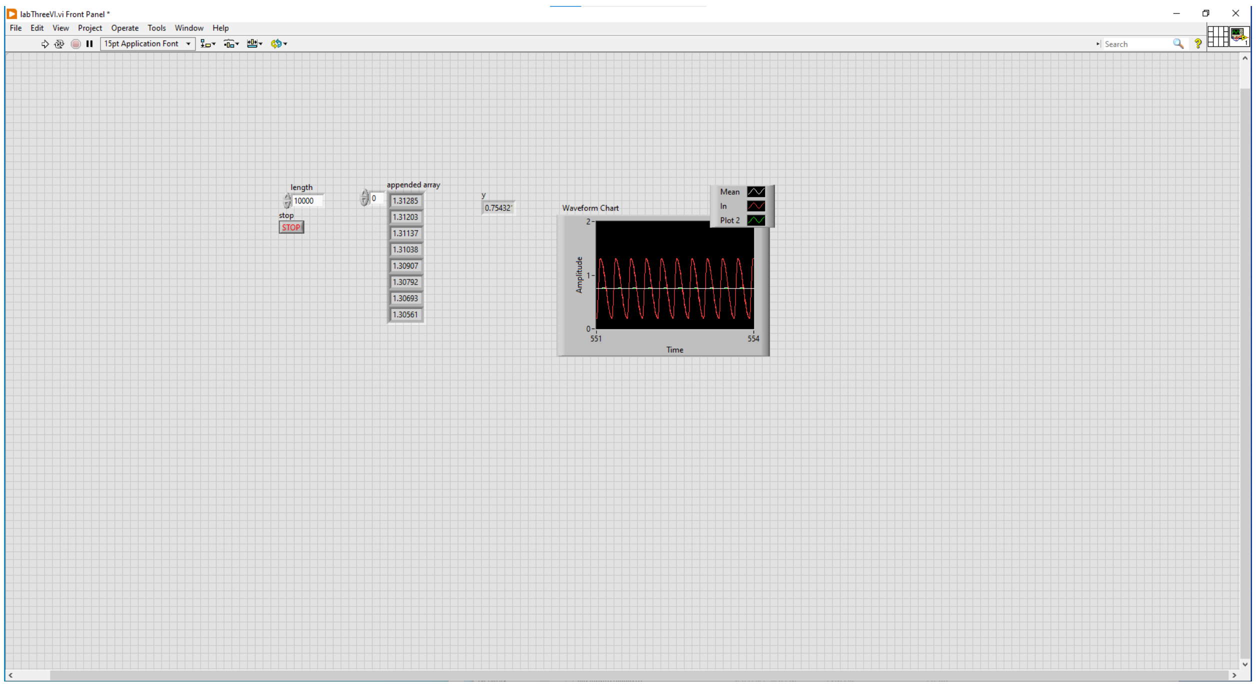
## 6 Appendices

### 6.1 Appendix A

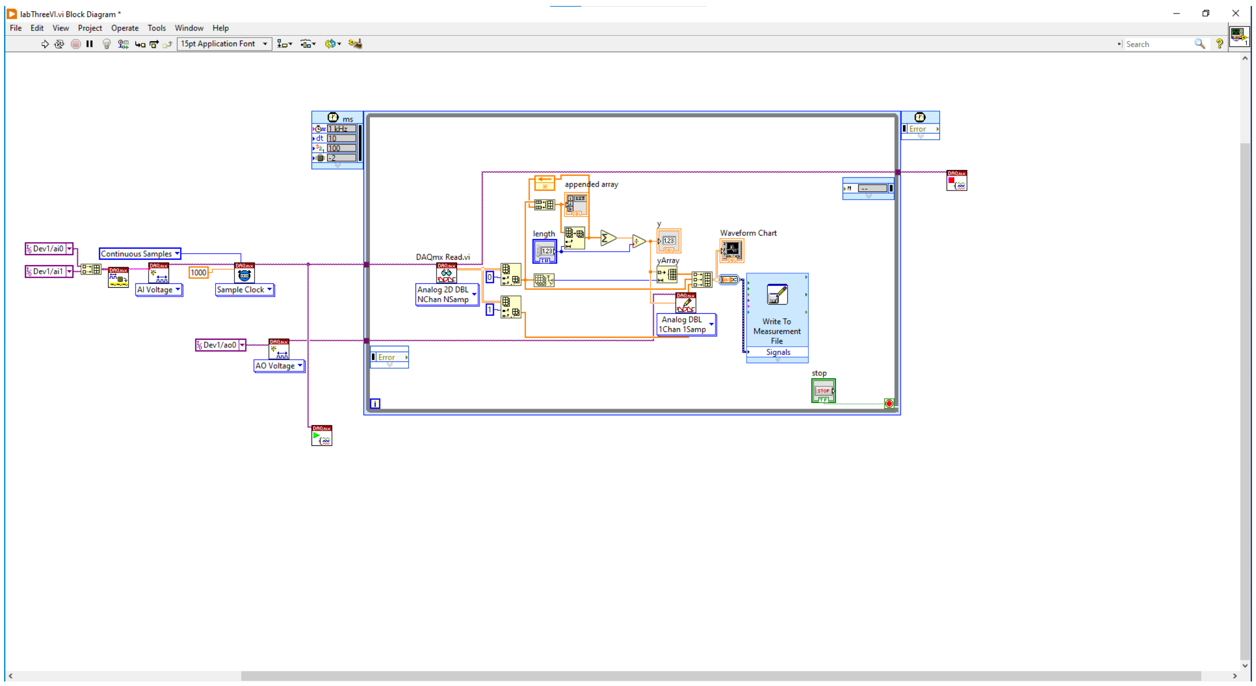
Screenshots of the implementation in LabVIEW (front panel and block diagram) are included in Figure 6. The VI is also available on GitHub.

### 6.2 Appendix B

All MATLAB scripts are attached. The scripts are also available on GitHub.



(a) LabVIEW Front Panel



(b) LabVIEW Block Diagram

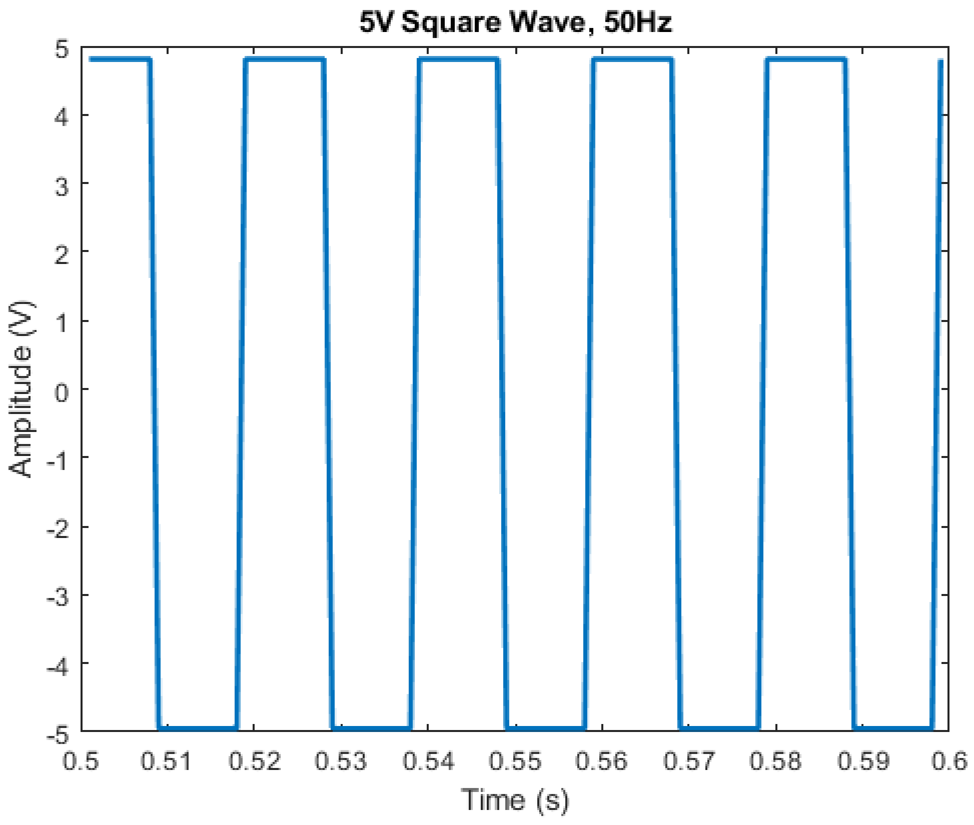
Figure 6: LabVIEW implementation of the filter.

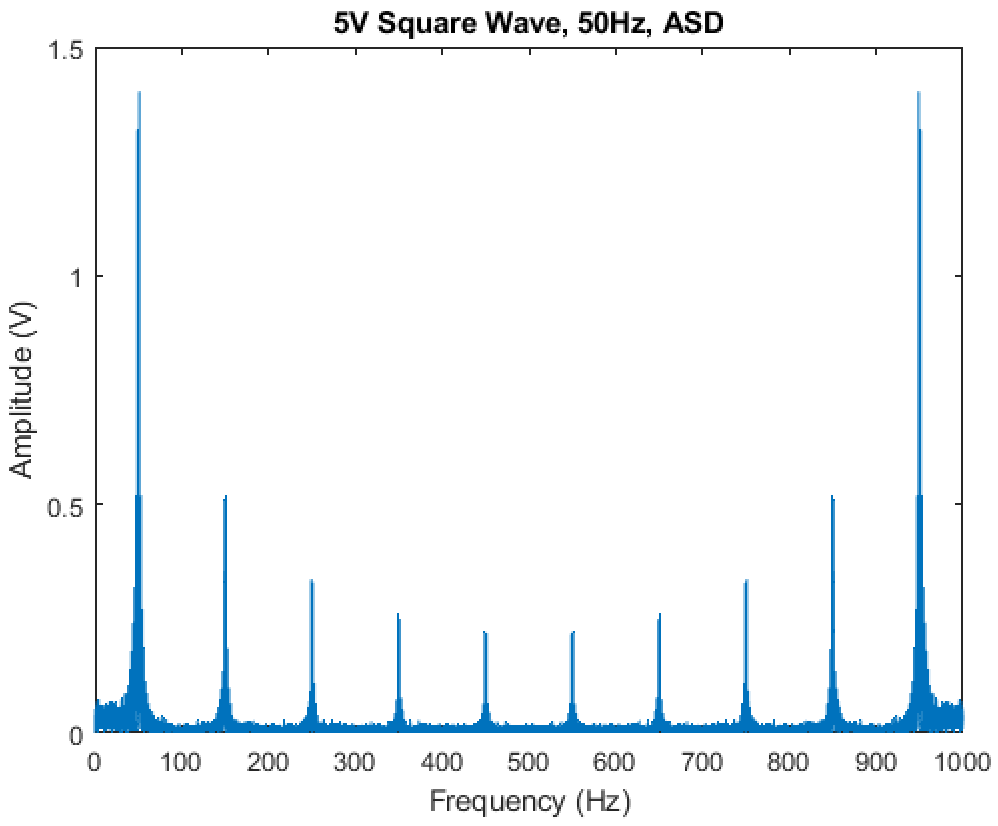
## Contents

## Determining qualities of input signal BP TESTING

```
%determine frequency, amplitude, and mean (report all as mean \pm std)
% e.g. IIIA5mV.lvm

[frequency, amplitude, mean] = inChar('../data/IIIA5V.lvm') %input name of file of interest


%freq is principal frequency, A is a tuple with [mean amplitude, std
%amplitude] and mu is the midline value. frequency in Hz, amplitude +
%midline in V.
```

```
frequency =

   50.8689


amplitude =

    4.7815    0.5080


mean =

    0.0752
```

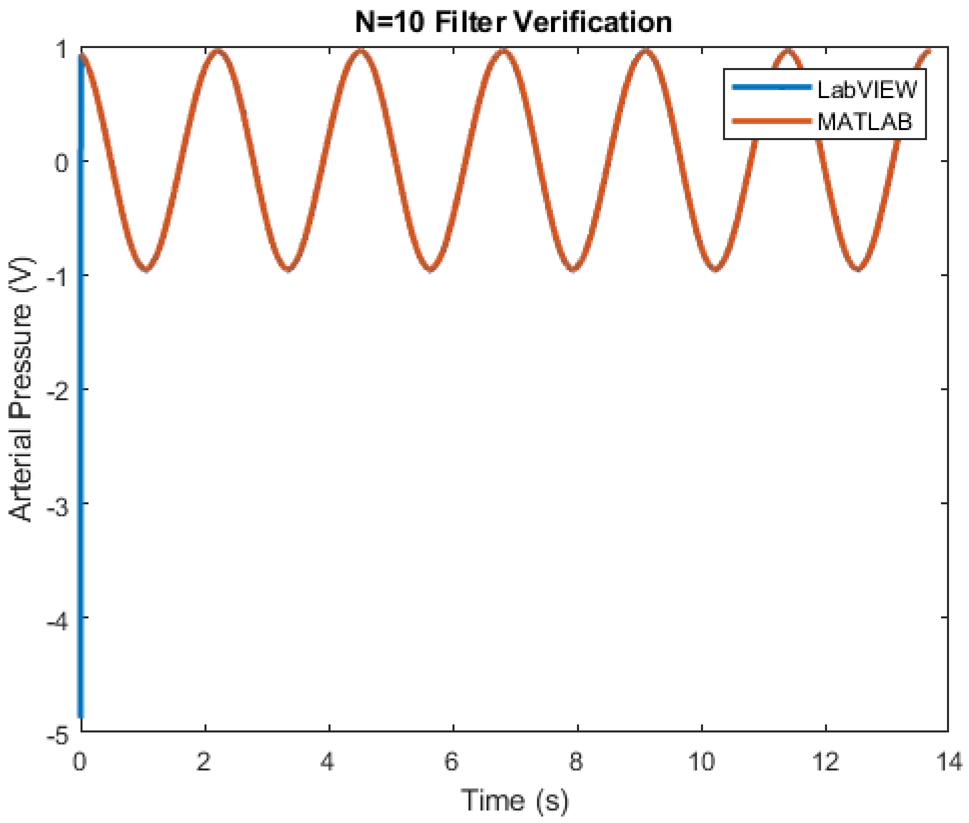


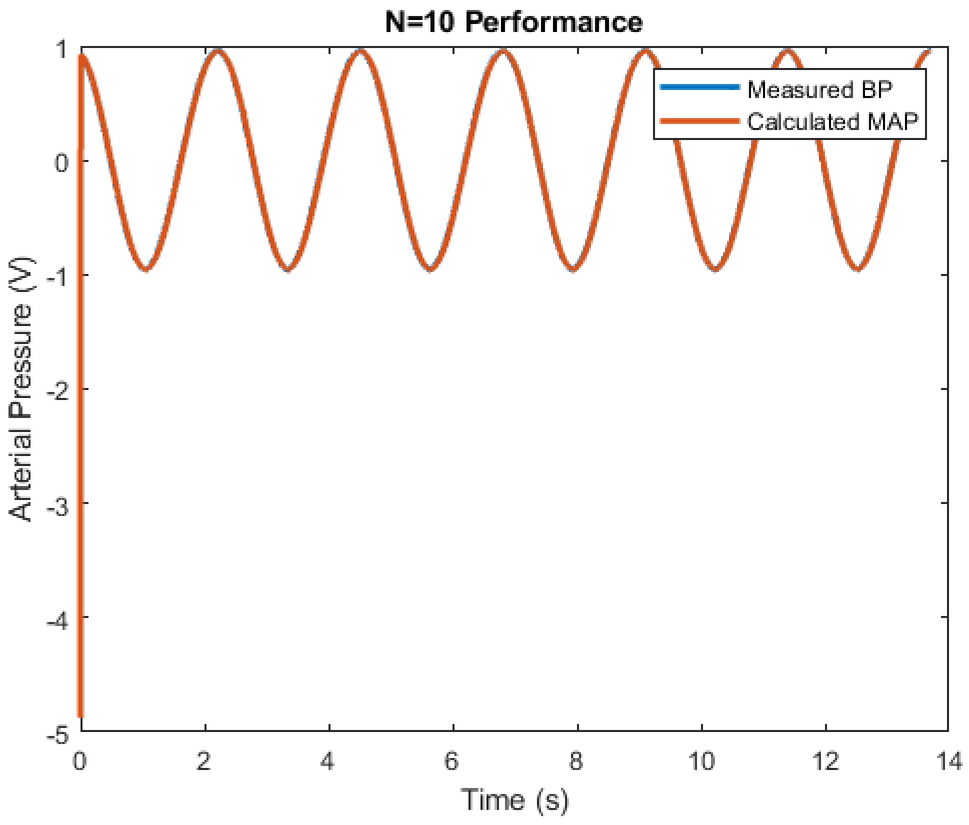


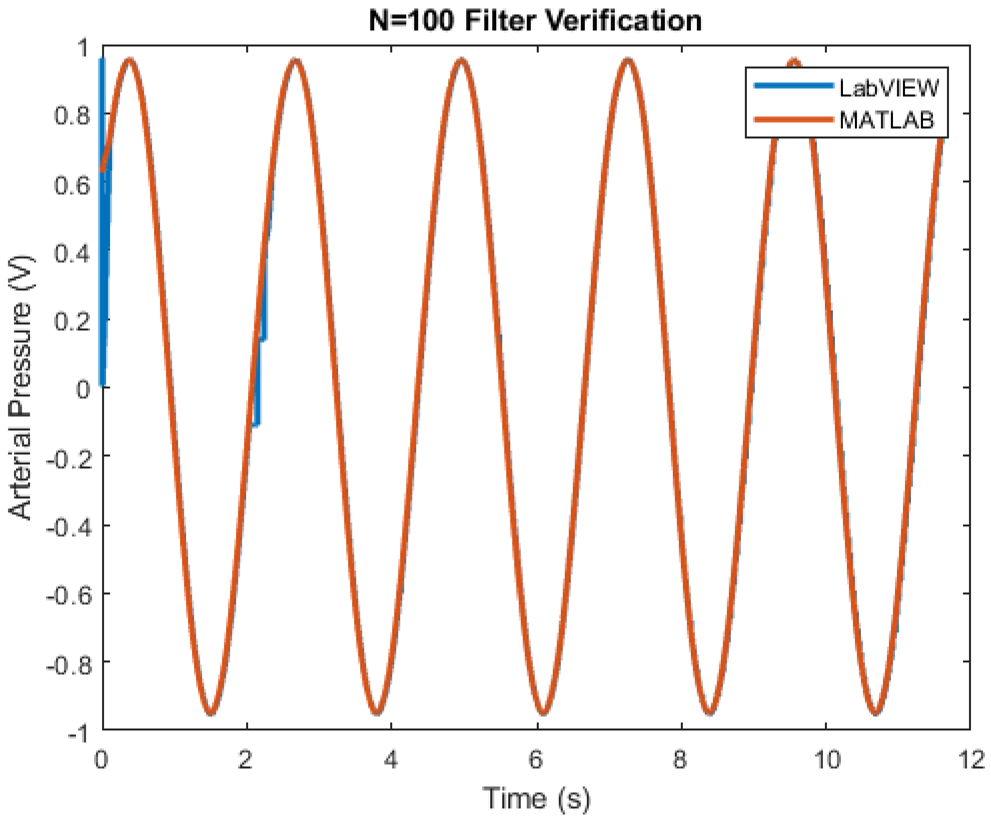
## Checking that filter did what we intended + Characterizing the Transfer Function MAP TESTING

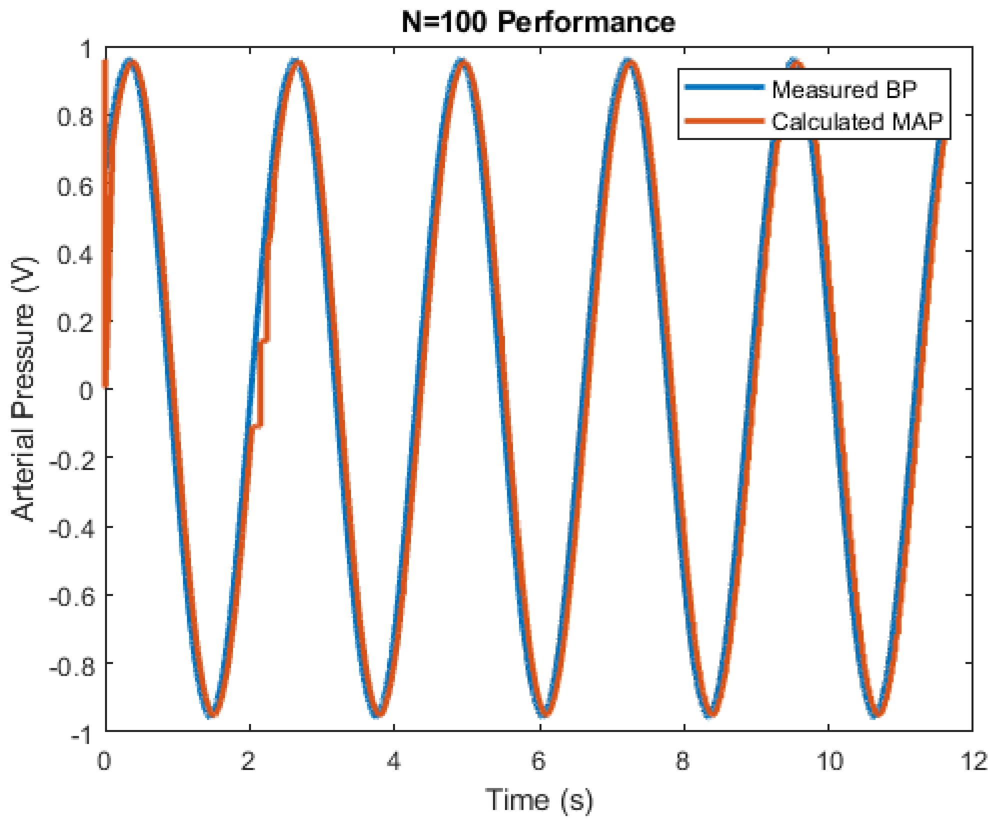
```
%Reproduce results in MATLAB
% e.g. IIIA10.lvm
```

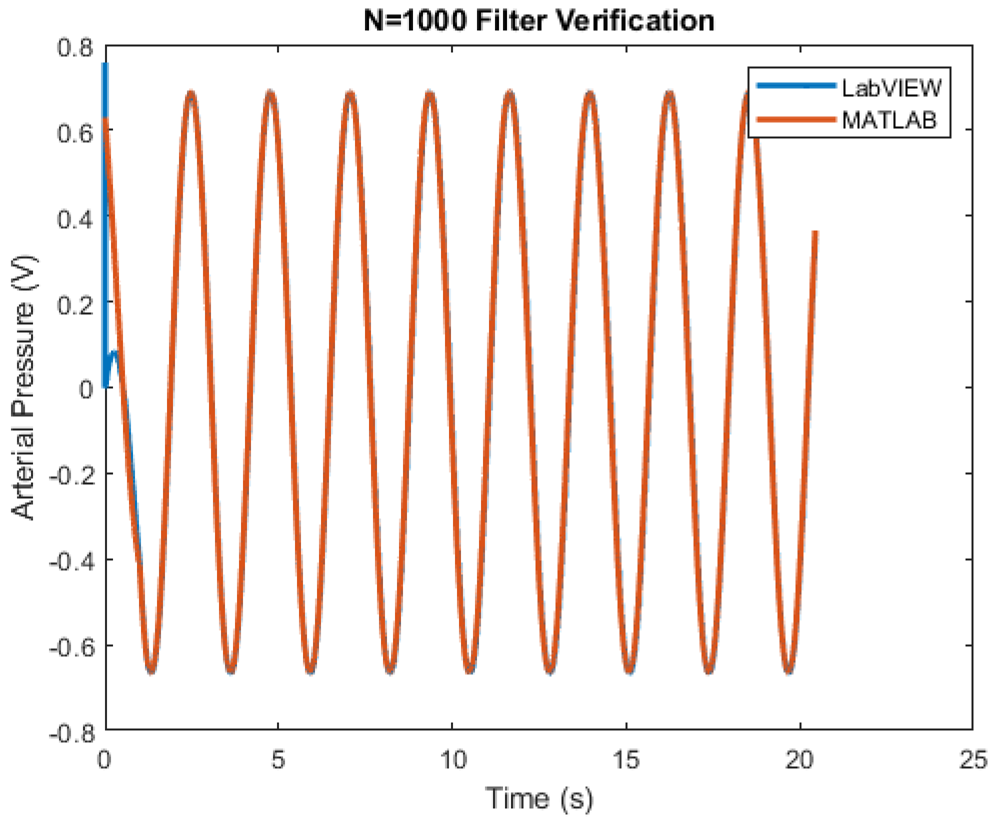
```
for N = [10, 100, 1000, 10000]  %iterate over the number of points used
    filename = "../data/IIIA" + num2str(N) + ".lvm";    %directory to data file
    mapArray = load(filename);  %load data from data file
    x = mapArray(:,3);  %extract x vector (not ordered according to lab instructions)
    y = mapArray(:,2);  %extract y vector
    newY = zeros(size(y));  %initialize MATLAB replication of calculated MAP
    for i = 1:length(y)     %populate new calculated MAP vector
        window = y(max(i-N+1,1):i);
        newY(i) = sum(window)/length(window);
    end

    M = length(x);              %signal length
    fs = 1000;                  %sampling frequency (Hz)
    dt = 1 / fs;                 %sampling period
    t = (0:M-1) * dt;           %Time array (s)

    figure(N)                           %Plot original and recreated MAP
    plot(t,x,t,newY, "LineWidth", 2)
    legend("LabVIEW", "MATLAB")
    xlabel("Time (s)")
    ylabel("Arterial Pressure (V)")
    title("N="+num2str(N)+" Filter Verification")

    figure(N + 1)                       %Plot measured bp against calculated MAP
    plot(t,y,t,x,"LineWidth", 2)
    legend("Measured BP", "Calculated MAP")
    xlabel("Time (s)")
    ylabel("Arterial Pressure (V)")
    title("N="+num2str(N)+" Performance")
end
```

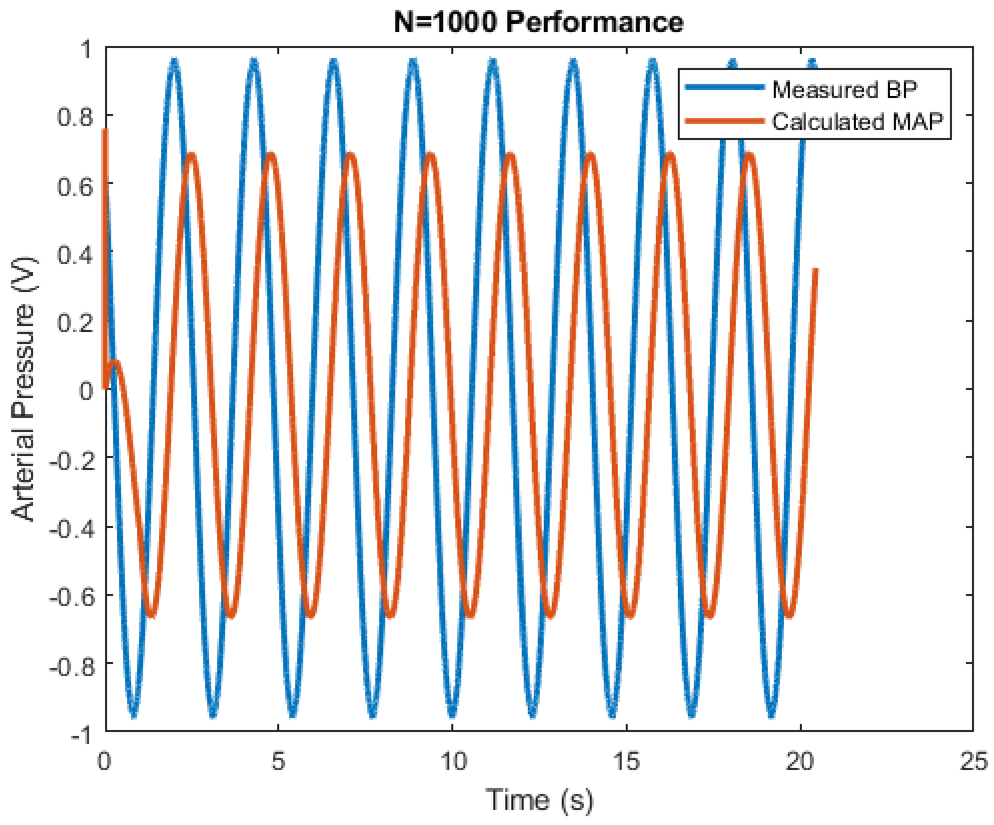


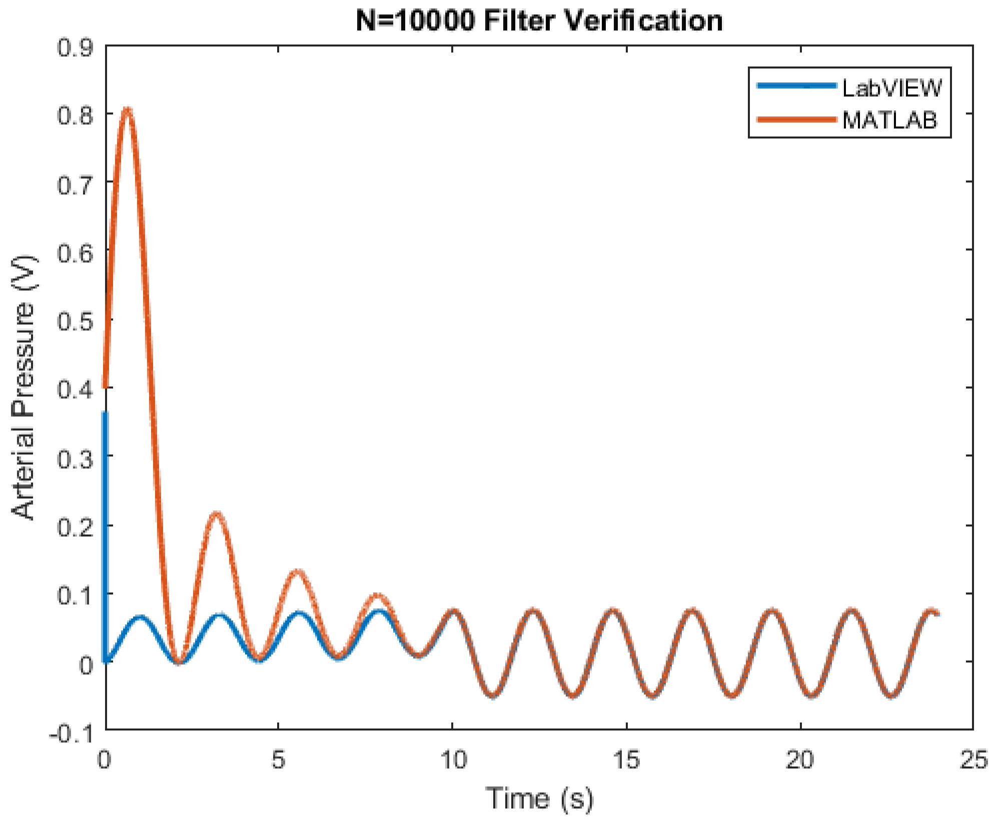

N=10 Performance
Measured BP
Calculated MAP
Arterial Pressure (V)
Time (s)
1
0
-1
-2
-3
-4
-5
0
2
4
6
8
10
12
14

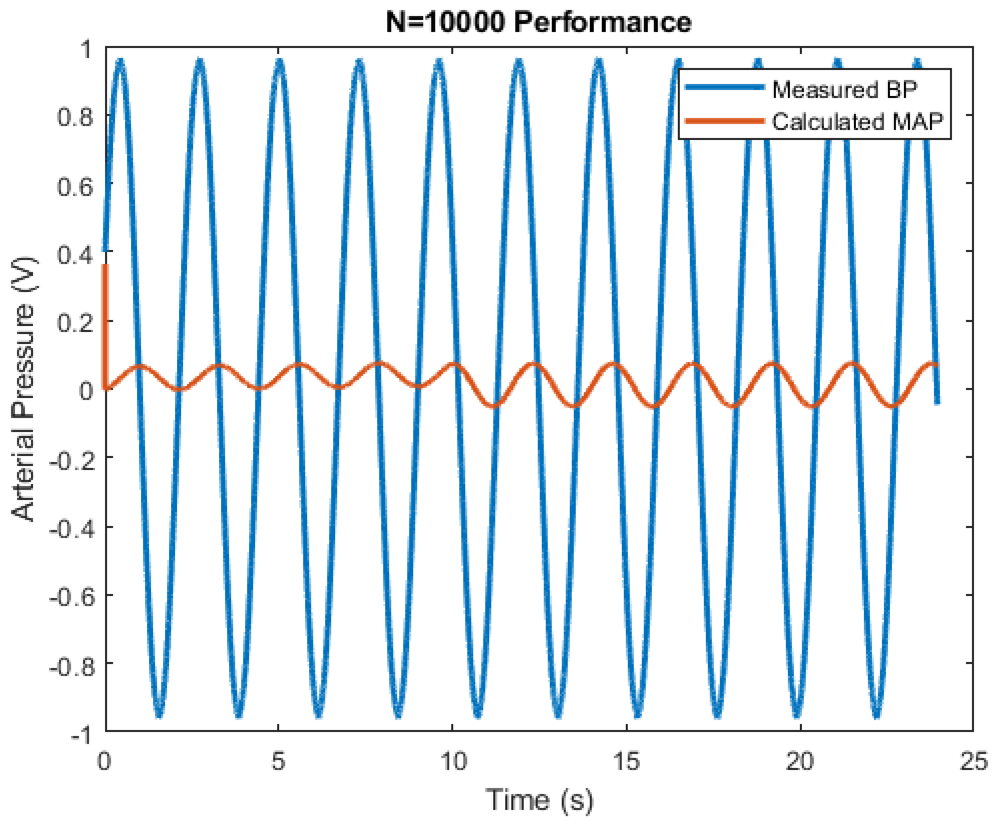

N=100 Filter Verification
LabVIEW
MATLAB
Arterial Pressure (V)
Time (s)
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
0
2
4
6
8
10
12


N=100 Performance
Measured BP
Calculated MAP
Arterial Pressure (V)
Time (s)
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
0
2
4
6
8
10
12


N=1000 Filter Verification
LabVIEW
MATLAB
Arterial Pressure (V)
Time (s)
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
0
5
10
15
20
25


N=1000 Performance
Measured BP
Calculated MAP
Arterial Pressure (V)
Time (s)


N=10000 Filter Verification
LabVIEW
MATLAB
Arterial Pressure (V)
Time (s)



## Finding delay and confirming 5mV resolution OUTPUT TESTING

```
%Find delay as argmax xcov, take delayed difference, form CI
% e.g. IIIAZ5mV.lvm
clear;clc;
outputArray = load("../data/IIIAZ5V.lvm");      %load recorded data
y = outputArray(:,1);                            %extract columns
x = outputArray(:,2);
z = outputArray(:,3);

N = length(x);                   %signal length
fs = 1000;                       %sampling frequency (Hz)
dt = 1 / fs;                      %sampling period (s)
df = fs/N;                       %frequency step size (Hz)
f = (0:N-1) * df;                %frequency array (Hz)
t = (0:N-1) * dt;                %Time array (s)

mu_y = mean(y);                  %reset y to have midline at zero
y = y - mu_y;
mu_z = mean(z);                  %reset z to have midline at zero
z = z - mu_z;

offset = mu_z - mu_y

delay = finddelay(y,z);              %use cross correlation to find most likely delay (indices)
delayTime = delay * dt               %most likely delay (s)

gain = mean(z(delay+1:N) ./ y(1:N-delay))
```

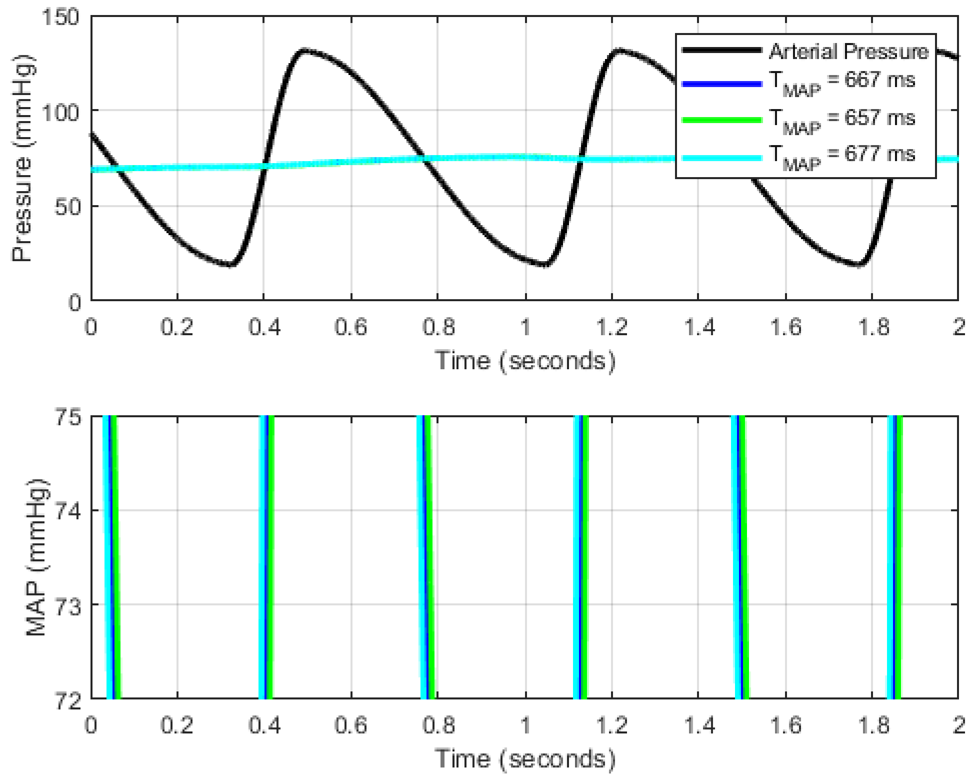
```
offset =
    0.0010


delayTime =

    0.0080


gain =

    1.0332
```

## 90 bpm VALIDATION

```
clear, clc

data = sprintf('../data/IIIB90bpm10000.lvm'); % reading in 90bpm N = 10,000 data
[x y z] = textread(data,'%n %n %n','headerlines',23);
% data1 = sprintf('IIIB90bpm1000.lvm'); % reading in 90bpm N = 1,000 data
% [x1 y1 z1] = textread(data1,'%n %n %n','headerlines',23);
% data2 = sprintf('IIIB90bpm100.lvm'); % reading in 90bpm N = 100 data
% [x2 y2 z2] = textread(data2,'%n %n %n','headerlines',23);

t = 1:2000; % time in milliseconds


xa = x(9001:11000); % N = 10,000 x data
ya = y(9001:11000); % N = 10,000 y data

x1 = x(8991:10990); % shift data back in time by 10 ms
y1 = y(8991:10990);

x2 = x(9011:11010); % shift data forward in time by 10 ms
y2 = y(9011:11010);


length(t)
length(x2)
figure(1)
subplot(2,1,1);
pa = plot(t/1000,ya*100,'-k',t/1000,xa*100,'-b','LineWidth',2);
hold on
pa1 = plot(t/1000,x1*100,'-g','LineWidth',2);
pa2 = plot(t/1000,x2*100,'-c','LineWidth',2);
grid on;
legend('Arterial Pressure','T_{MAP} = 667 ms','T_{MAP} = 657 ms','T_{MAP} = 677 ms');
xlabel('Time (seconds)')
ylabel('Pressure (mmHg)')
hold off



subplot(2,1,2)
pb = plot(t/1000,ya*100,'-b','Linewidth',2);
hold on
pb1 = plot(t/1000,y1*100,'-g','Linewidth',2);
pb2 = plot(t/1000,y2*100,'-c','Linewidth',2);
axis([0 2 72 75]);
```

```
grid on;
xlabel('Time (seconds)')
ylabel('MAP (mmHg)')
hold off
```

```
ans =

        2000


ans =

        2000
```

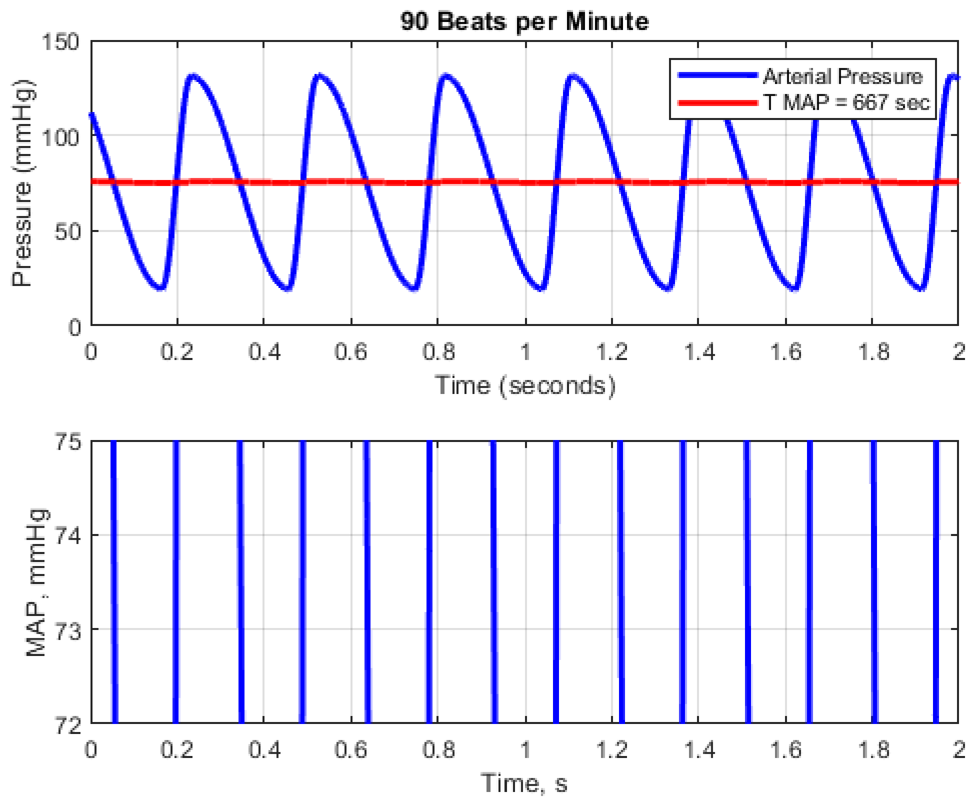


### 180 bpm

```
clear, clc
figure(2)


t = 1:2000; % time

data3 = sprintf('../data/IIIB180bpm10000.lvm'); % reading in 180bpm N = 10,000 data
[X Y Z] = textread(data3,'%n %n %n','headerlines',23);


x3 = X(10001:12000); % N = 10,000 x data
y3 = Y(10001:12000); % N = 10,000 y data
```

```
figure(2)
subplot(2,1,1)
pc = plot(t/1000, y3*100,'-b', t/1000, x3*100, '-r','Linewidth', 2)
legend('Arterial Pressure','T MAP = 667 sec')
grid on;
title('90 Beats per Minute')
xlabel('Time (seconds)')
ylabel('Pressure (mmHg)')


subplot(2,1,2)
pd = plot(t/1000,y3*100,'-b','LineWidth',2);
axis([0 2 72 75]); grid on;
ylabel('MAP, mmHg');
xlabel('Time, s');
```

```
pc =

  2×1 Line array:

  Line
  Line
```



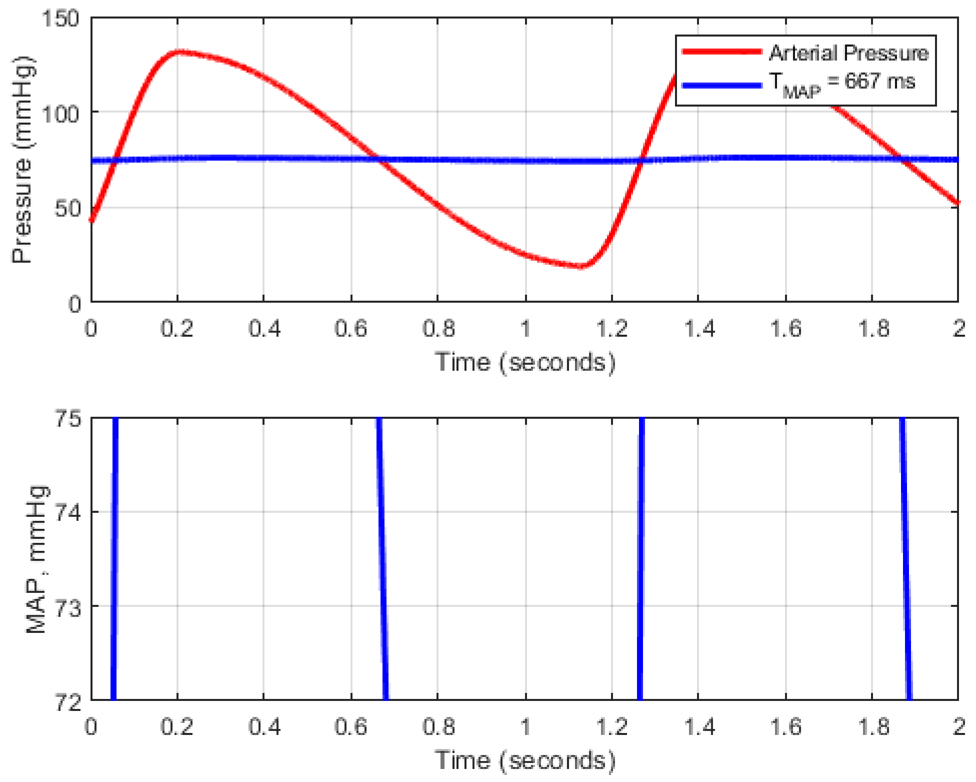
## 60 bpm

```
clear, clc

data3 = sprintf('../data/IIIB60bpm10000.lvm'); % reading in 180bpm N = 10,000 data
[X Y Z] = textread(data3,'%n %n %n','headerlines',23);
```

```matlab
t = 1:2000; % time (miliseconds)
x = X(10001:12000); % N = 10,000 x data
y = Y(10001:12000); % N = 10,000 y data

subplot(2,1,1)
pe = plot(t/1000, y*100, '-r', t/1000, x*100, '-b', 'Linewidth', 2)
grid on;
legend('Arterial Pressure','T_{MAP} = 667 ms');
xlabel('Time (seconds)')
ylabel('Pressure (mmHg)')

subplot(2,1,2)
pd = plot(t/1000,y*100,'-b','LineWidth',2);
axis([0 2 72 75]); grid on;
ylabel('MAP, mmHg');
xlabel('Time (seconds)');



function [freq, A, mu] = inChar(filename)
    bpArray = load(filename);   %load .lvm recorded by LabVIEW
    x = bpArray(:,1);           %input (blood pressure) vector
    y = bpArray(:,2);           %Calculated MAP vector

    mu = mean(x);               %calculate mean value
    x = x - mu;                 %recenter x at midline for amplitude calculations

    N = length(x);              %signal length
    fs = 1000;                  %sampling frequency (Hz)
    dt = 1 / fs;                 %sampling period
    df = fs/N;                  %frequency step size (Hz)
    f = (0:N-1) * df;           %frequency array (Hz)
    t = (0:N-1) * dt;           %Time array (s)
    X = abs(1/N * fft(x));      %find fft of x to determine amplitude and frequency

    figure(1)                   %plotting input waveform
    plot(t(t>0.5& t<0.6),x(t>0.5 & t<0.6), 'LineWidth',2)
    xlabel('Time (s)')
    ylabel('Amplitude (V)')
    title('5V Square Wave, 50Hz')

    figure(2)                   %plotting amplitdue spectral density
    plot(f,X)
    xlabel('Frequency (Hz)')
    ylabel('Amplitude (V)')
    title('5V Square Wave, 50Hz, ASD')

    [~, maxIndex] = max(X(1:round(N/2)));   %Find index of primary frequency
    freq = f(maxIndex);                     %Find primary frequency

    pks = findpeaks(x);                 %find local maxima in signal
    A = [mean(pks), std(pks)];          %Find mean + std of local maxima
end
```

```
pe = 

  2×1 Line array:

  Line
```

```
Line
```



*Published with MATLAB® R2021b*