ADC Sigma Delta: Data Processing

Script for importing data from the following text file:

```
filename: D:\Backup\Users_gennai\Documents\MATLAB\SigmaDelta\default.csv
```

Auto-generated by MATLAB on 22-Apr-2022 12:45:26

Setup the Import Options and import the data

```
opts = delimitedTextImportOptions("NumVariables", 2);

% Specify range and delimiter
opts.DataLines = [7, Inf];
opts.Delimiter = ",";

% Specify column names and types
opts.VariableNames = ["Times", "Data"];
opts.VariableTypes = ["double", "string"];

% Specify file level properties
opts.ExtraColumnsRule = "ignore";
opts.EmptyLineRule = "read";

% Specify variable properties
opts = setvaropts(opts, "Data", "WhitespaceRule", "preserve");
opts = setvaropts(opts, "Data", "EmptyFieldRule", "auto");

% Import the data
myData = readtable("D:\Backup\Users_gennai\Documents\MATLAB\SigmaDelta\default.csv", opts);
```

```
% Clear temporary variables
clear opts
```

Convert data in a bit stream

```
nbit=8;
sz_myData=size(myData);
myStream=zeros((nbit-1)*(sz_myData(1)+1),1);
for i=1:sz_myData(1)-1
    str=myData(i,2).Data{1};
    for j=1:nbit
        myStream(1+(i-1)*nbit+j-1)=str2double(str(j+1));
    end
end
```

Compute sampling period

```
dt=myData{end,1}-myData{end-1,1}; % Sampling period
dr=dt/nbit; % Output data rate
fprintf('Sampling Frequency: %g SPS\nOutput data rate: %g SPS\n', 1/dr,1/dt);
```

```
Sampling Frequency: 50000 SPS
Output data rate: 6250 SPS
```

```

```

Clean first sample and plot data vs time

```
myStream(1)=[];
ttime=0:dr:(length(myStream)-1)*dr;
figure(1);
clf;
plot(ttime,myStream,'.');
title('Input Bitstream')
xlabel('Time (s)');
ylabel('bit value');
xlim([0.03 0.05])
ylim([-0.5 1.5])
```

Compute moving average 1

```
clear ma1;
ma1=movmean(myStream,8);
ma1(end-nbit+1:end)=[];
ttime1=ttime(1:end-nbit);
figure(2);
clf;
plot(ttime1,ma1);
title('First Moving Average')
xlabel('Time (s)');
ylabel('A.U.');
```

Compute second moving average

```
clear ma2;
ma2=movmean(ma1,8);
ma2(end-nbit+1:end)=[];
ttime2=ttime1(1:end-nbit);
%remove mean value to simplify fit
ma2z=ma2-mean(ma2);
figure(3);
plot(ttime2',ma2)
title('Second Moving Average')
xlabel('Time (s)');
ylabel('A.U.');
```

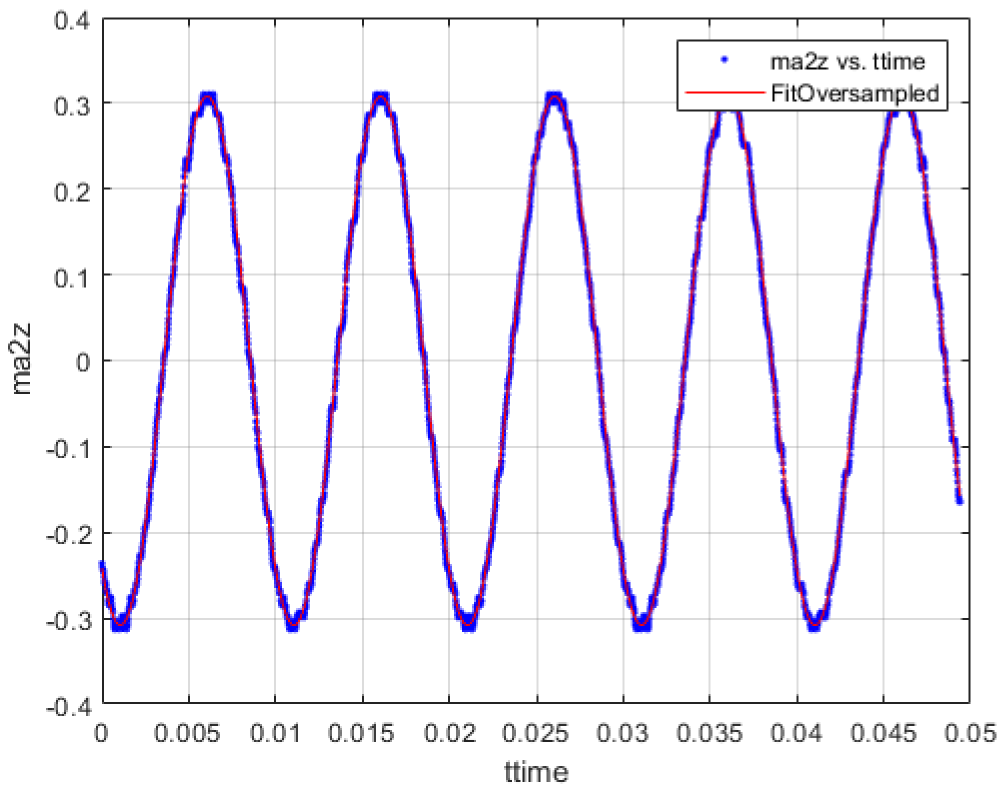
Compute third moving average

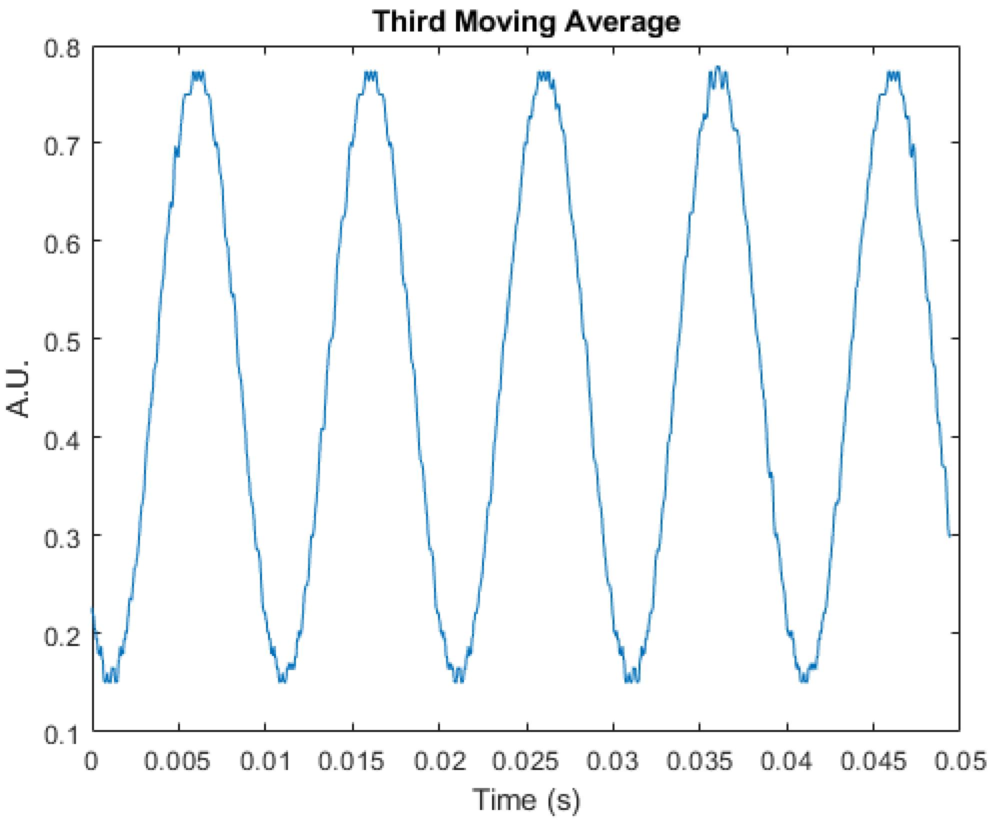
```
clear ma3;
```

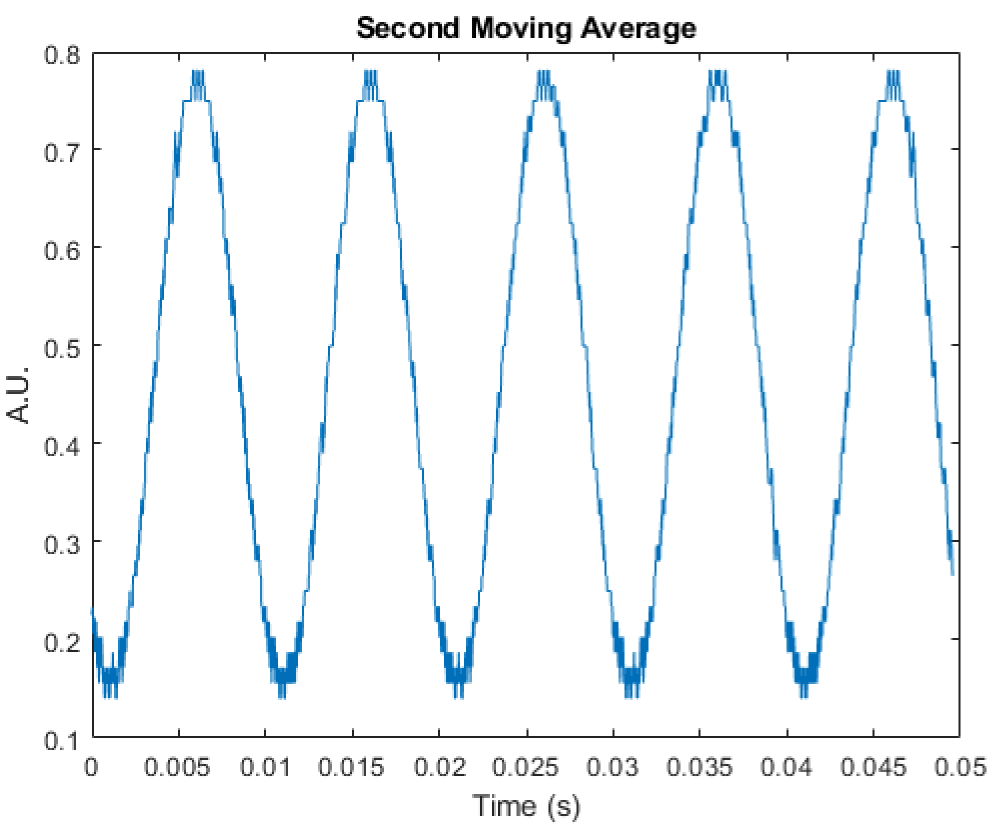
```
ma3=movmean(ma2,8);
ma3(end-nbit+1:end)=[];
ttime3=ttime2(1:end-nbit);
%remove mean value to simplify fit
ma3z=ma3-mean(ma3);
figure(4);
plot(ttime3,ma3)
title('Third Moving Average')
xlabel('Time (s)');
ylabel('A.U.');
```

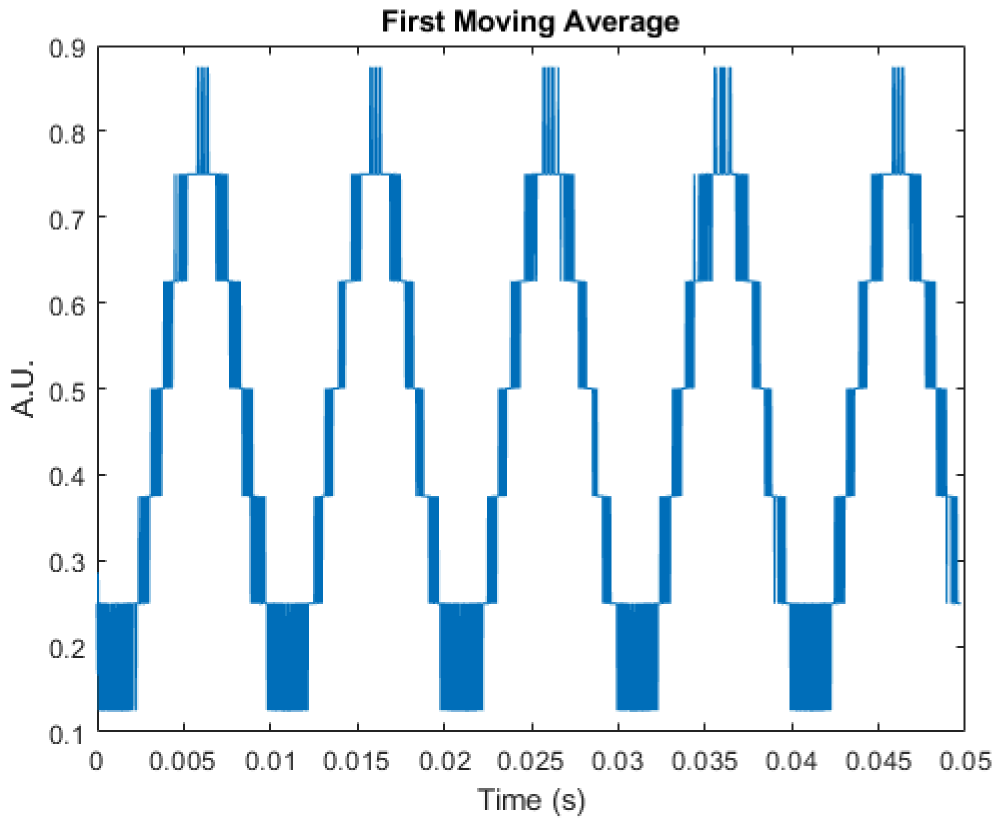
Fit of third moving average

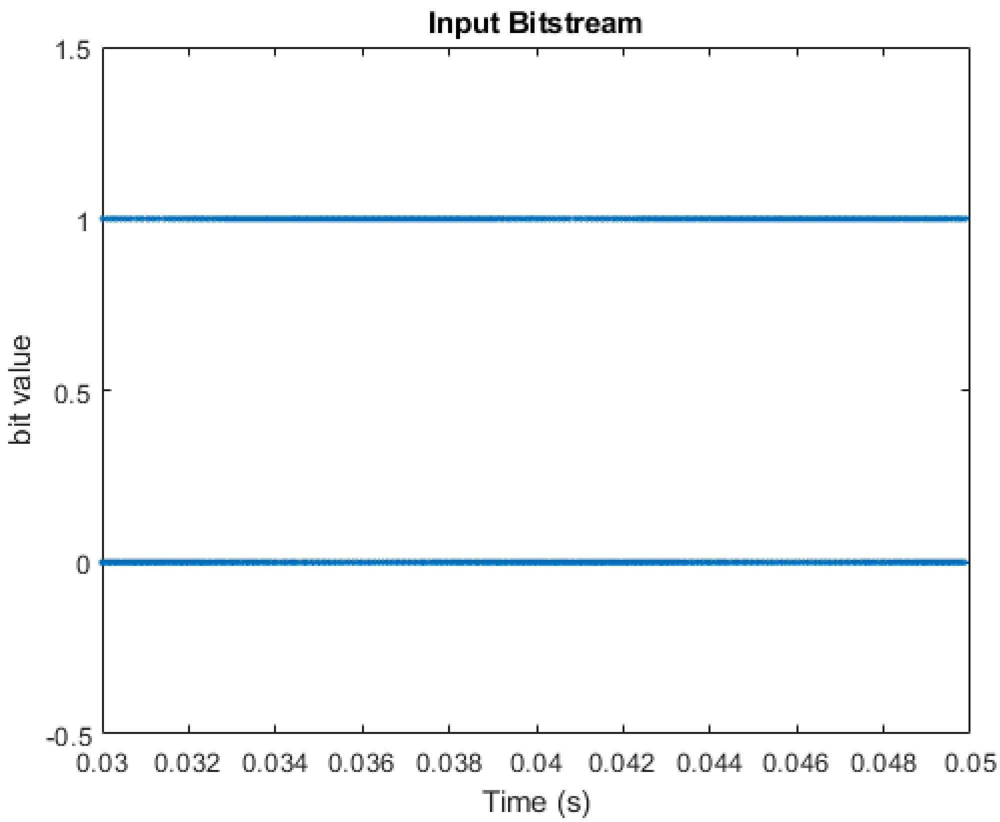
```
[fitresult, gof] = createFit(ttime3', ma3z)
```




Third Moving Average
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
A.U.
0
0.005
0.01
0.015
0.02
0.025
0.03
0.035
0.04
0.045
0.05
Time (s)


Second Moving Average
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
A.U.
0
0.005
0.01
0.015
0.02
0.025
0.03
0.035
0.04
0.045
0.05
Time (s)






```
fitresult =
     General model Sin1:
     fitresult(x) =  a1*sin(b1*x+c1)
     Coefficients (with 95% confidence bounds):
       a1 =       0.308  (0.3076, 0.3084)
       b1 =       628.5  (628.4, 628.6)
```

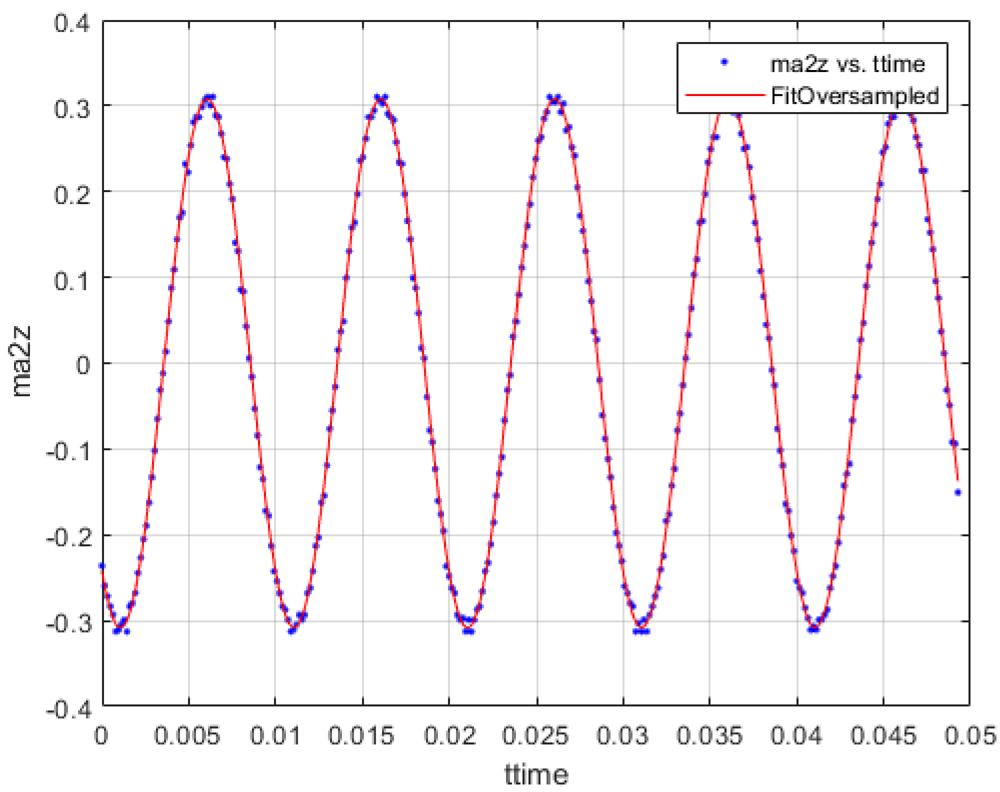
```
        c1 =      -2.237  (-2.24, -2.234)
gof = struct with fields:
           sse: 0.1348
       rsquare: 0.9989
           dfe: 2468
    adjrsquare: 0.9989
          rmse: 0.0074
```

Downsample second moving average by a factor nbit (8)

```
ma3zD=downsample(ma3z,8);
ttimeD=downsample(ttime3,8);
```

Fit of decimated data

```
[fitresultD, gofD] = createFit(ttimeD', ma3zD)
```




```
fitresultD =
     General model Sin1:
     fitresultD(x) =  a1*sin(b1*x+c1)
     Coefficients (with 95% confidence bounds):
       a1 =       0.308  (0.3068, 0.3092)
       b1 =       628.5  (628.3, 628.8)
       c1 =      -2.237  (-2.245, -2.229)
gofD = struct with fields:
           sse: 0.0175
       rsquare: 0.9988
```

```
         dfe: 306
  adjrsquare: 0.9988
        rmse: 0.0076
```

Compute ENOB (Effective Number Of Bits)

```
residuals=ma3zD-fitresultD(ttimeD');
rmsNoise=std(residuals);
rmsSigMax=0.5*sqrt(2)/2;
snrmax=rmsSigMax^2/rmsNoise^2;
enob=(10*log10(snrmax)-1.76)/6.02;

fprintf('This ADC is equivalent to an ideal ADC with:\n\tENOB = %g\n\tSamplig Frequency %g SPS\
```

```
This ADC is equivalent to an ideal ADC with:
 ENOB = 5.3325
 Samplig Frequency 6250 SPS
```

Calibration

Only a short note about calibration. Since the peak-to-peak analog input signal must be limited to the saturation levels of opampa used as single bit DAC in our circuit, the digital range 0-1 corresponds to about 3.5-(-3.5)= 7Vpp. Therefore 0.1 for the digital signal corresponds to about 0.7 V. The amplitude of sine wave estimated by the fit is 0.308 (+/- 0.0012) and corresponds to an input anlog sine wave with a amplitude of about 216 mV (+/- 0.8 mV). This value must be compared with the actual sine wave generated by AnalogDiscovery 2.