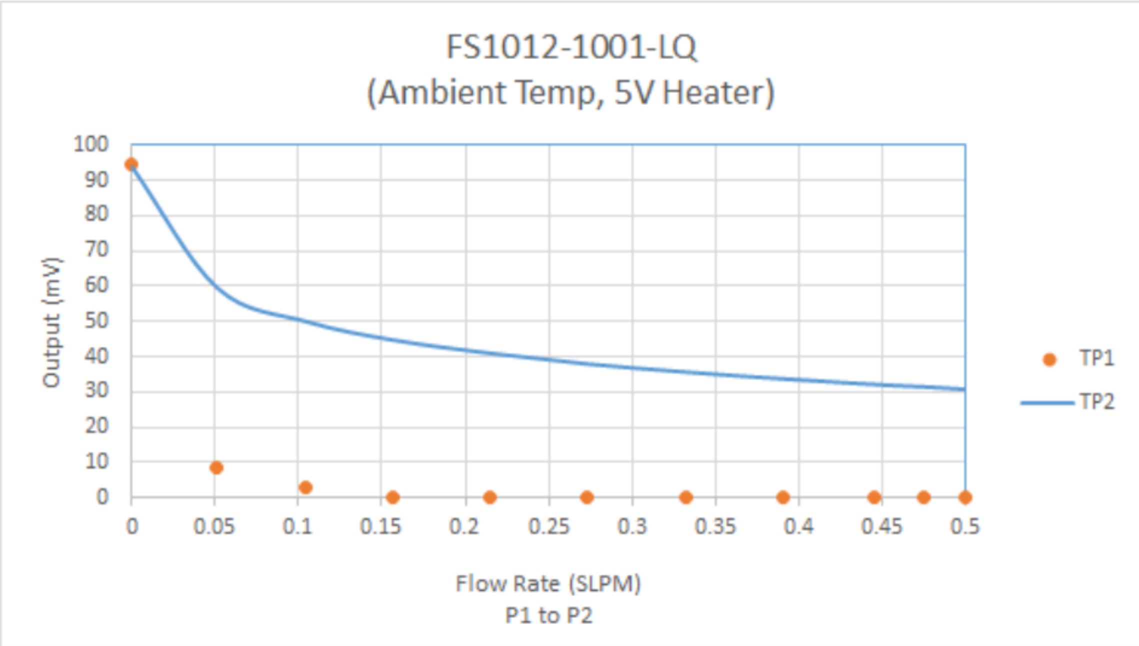
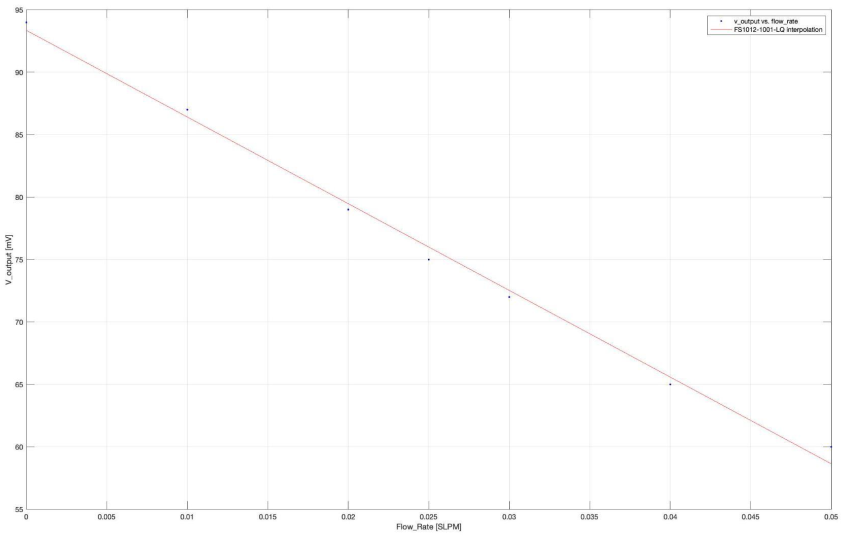
FS1012-1001-LQ – Curve Fitting

The FS1012-1001-LQ is a flow meter capable of measuring a non-corrosive gas or liquid. Since, the proposed device allows drug infusion, it is necessary to characterize sensor's output in reference to liquids. The flow output curves are typical values at room conditions.
For the calibration process, it was decided to set the sensor's working range between *0.00 SLPM-0.05 SLPM*. This choice appears to be in agreement with the project requirements and in line



with typical flow rates used in a real infusion pump.
Under these assumptions, the calibration curve shows the following pattern.



The sensor I/O relationship is given by

$$V[mV] = k\,FR[SLPM] + offset$$

$$\Delta V = V_1 - V_2 = \left(k\, FR_1 + offset\right) - \left(k\, FR_2 + offset\right) = k\left(FR_1 - FR_2\right) = k\Delta FR$$

- k represents the angular coefficient of the line
- the offset value is the intercept of the line with the y-axis (value that the sensor returns when the measurand is zero)

The output is on the order of mV. Considering an amplifier stage with a gain of 40, the I/O relationship can be rewritten as follows:

$$\Delta FR = \frac{\frac{\Delta V}{Gain}}{k}$$

The voltage of the less significant bit is:

$$\Delta V_{LSB} = \frac{V_{cc}}{2^n} = \frac{5\,V}{2^{10}} = \frac{5\,V}{1024} = 5\,mV$$

We calculate the minimum flow rate at the less significant bit:

$$\Delta FR_{min} = \frac{\frac{\Delta V}{Gain}}{k}$$

It is obtained a minimum flux rate of:

$$\Delta FR_{min} = -\ 0.029\,ml/s$$

To validate the accuracy of the model used in order to obtain the sensor I/O relationship, the calculation of the fitting error was performed:

$$error = \frac{V_{experimental} - V_{returned}}{V_{experimental}}$$

The fitting error turns out to be:

$$error = \ 1.02$$

Matlab code used to calculate the minimum flow rate and fitting error is given:

```
clc;

close all;

clear all;


%Experimental Data Matching (Graphical method)
```

```
%[SLPM]

fr_1 = 0.0000;
fr_2 = 0.0100;
fr_3 = 0.0200;
fr_4 = 0.0250;
fr_5 = 0.0300;
fr_6 = 0.0400;
fr_7 = 0.0500;

flow_rate = [fr_1, fr_2,fr_3,fr_4,fr_5,fr_6,fr_7]';

%[mV]

v_out1 = 94.00;
v_out2 = 87.00;
v_out3 = 79.00;
v_out4 = 75.00;
v_out5 = 72.00;
v_out6 = 65.00;
v_out7 = 60.00;

v_output = [v_out1,v_out2,v_out3,v_out4,v_out5,v_out6,v_out7]';
```

```
%% Curve Fitting (from experimental data)

fs_interpolation = cftool(flow_rate,v_output);

k = -694.3;

offset = 93.36;

%% Curve Fitting Plot: 'FS1012-1001-LQ interpolation'.

[xData, yData] = prepareCurveData( flow_rate, v_output );

% Set up fittype and options.
ft = fittype( 'poly1' );

% Fit model to data.
[fitresult, gof] = fit( xData, yData, ft );

% Plot fit with data.
figure( 'Name', 'FS1012-1001-LQ interpolation' );
h = plot( fitresult, xData, yData );
legend( h, 'v_output vs. flow_rate', 'FS1012-1001-LQ interpolation', 'Location', 'NorthEast',
'Interpreter', 'none' );
% Label axes
xlabel( 'Flow_Rate [SLPM]', 'Interpreter', 'none' );
```

```
ylabel( 'V_output [mV]', 'Interpreter', 'none' );

grid on


%% Error definition


%Data matching from curve fitting


%[mV]

v_out1_real = 93.35;

v_out2_real = 86.38;

v_out3_real = 79.46;

v_out4_real = 75.98;

v_out5_real = 72.52;

v_out6_real = 65.58;

v_out7_real = 58.64;


v_output_real =[v_out1_real,v_out2_real, v_out3_real, v_out4_real,
v_out5_real,v_out6_real,v_out7_real]';


%Since v_output e v_output_real have consistent dimension


l = length(v_output);


for i=1:l
```

```
        %error due to the fitting (percentage)

        err_fitting(i) = abs(((v_output(i)-v_output_real(i))/v_output(i))*100);


end


%Mean err_fitting (percentage)

mean_err_fitting = mean(err_fitting);


%Non-linearity error

err_nlinearity = abs((v_output(4)-v_output_real(4))/v_output(4)*100);


%% DELTA FR_MIN MEASURABLE

gain = 40;


v_lsb_min = 5000/2^10; %[mV]

lsb = 5/2^10;%[V]


delta_fr_min = (((v_lsb_min/gain)/k)/60)*1000;


%% Compute for analog output of the sensor


v_min = (94 * gain)/1000; %[mV] --> 94 mV

v_max = (60 * gain)/1000; %[mV] --> 60 mV


v_output_sensor = [v_min : (-lsb): v_max];
```

```
%max number ADC levels

levels = [0: lsb: 5];

level_94mV = v_min/lsb;
level_60mV = v_max/lsb;

%Read data from FS1012-1001-LQ

%v_output_sensor_code = ((output*lsb)/40)*1000;

%use v_output_sensor_code in I/O relationship (from interpolation)
```

to visualise the Matlab code and the plots see the file FS_Interpolation_report.m.