

ELEX 3120/3321: Electric Circuits 2

LAB 4 - SCA

|  |  |  |
| --- | --- | --- |
| Student Name: Enze Xu | Student Number:A01336393 | Set:B |

Table of Contents

[1 Introduction 3](#_Toc181441576)

[2 SCA Design 3](#_Toc181441577)

[2.1 SCA Design Equations 3](#_Toc181441578)

[2.2 SCA Circuit Implementation 3](#_Toc181441579)

[2.3 MATLAB SCA System 4](#_Toc181441580)

[3 Calibrate SCA Amplifier 5](#_Toc181441581)

[4 Test Calibrated SCA 5](#_Toc181441582)

[5 Conclusions 7](#_Toc181441583)

Table of Figures

[Figure 1 - SCA Circuit Implementation LTspice Schematics 3](#_Toc181441560)

[Figure 2 - SCA Circuit LTspice Simulation – Vout[V] Versus Vxd[V] 4](#_Toc181441561)

[Figure 3 - MATLAB Simulation of Vxd[V] and Vo[V] Versus Gravity[g] 5](#_Toc181441562)

[Figure 4 - MATLAB PLOT Vxd vs SCA Output From -1 g to 1 g 6](#_Toc181441563)

Table of Tables

[Table 1 - Predicted and Measured Output Voltage of Calibrated SCA 5](#_Toc181441591)

# Introduction

This lab involves designing and calibrating a signal conditioning amplifier (SCA) circuit for an accelerometer to ensure its output matches the input range of an ADC. Accelerometers are widely used in devices like smartphones and fitness trackers to measure acceleration, but their analog output requires conditioning for digital processing. Using TL-084 op-amps, the SCA circuit will be built, calibrated, and tested across the range of -1g to +1g. MATLAB will be used for data acquisition and analysis to verify the system's accuracy, and the accelerometer's performance will be measured in real-time.

# SCA Design

## SCA Design Equations

## SCA Circuit Implementation



Figure 1 - SCA Circuit Implementation LTspice Schematics

A graph with a line

Description automatically generated

-1g

+1g

Figure 2 - SCA Circuit LTspice Simulation – Vout[V] Versus Vxd[V]

## MATLAB SCA System

close all; % Close all figures

% Define the range of Gravity from -1g to 1g in 0.1g steps

Gravity = -1:0.1:1;

% Initialize SCA\_Output array with zeros

Vo = zeros(size(Gravity));

Vxd = zeros(size(Gravity));

% FOR loop to implement functions of Vxd and Vo

for i = 1:length(Gravity)

Vxd = 0.3 \* Gravity + 1.5;

Vo = (25/3) \* Vxd - 10;

end

% Generate the VTD plot (SCA\_Output vs Gravity)

figure;

plot(Gravity, Vxd, 'LineWidth', 2); % VTD of Vxd vs Gravity

grid on;

hold on;

plot(Gravity, Vo, 'LineWidth', 2); % VTD of Vo vs Gravity

% Label the axes

xlabel('Gravity(g)','FontSize',12);

ylabel('SCA\_Output(V)','FontSize',12);

% Title and legend for the plot

title('ADXL335 Tilt System','FontSize', 14);

legend('Vxd','Vo');

A graph with a red line and blue line

Description automatically generated

Figure 3 - MATLAB Simulation of Vxd[V] and Vo[V] Versus Gravity[g]

# Calibrate SCA Amplifier

Set Vxd to the minimum value 1.2 V by tuning the offset potentiometer, until the output reaches the minimum voltage 0.0 V. Set Vxd to maximum value 1.8 V, decrease or increase gain by tuning the Ri potentiometer, until the output reach the maximum voltage 5.0 V. After adjusting he gain, check back whether Vxd (min) matches with Vout (min), start over the calibration process from adjusting the offset until Vout is close enough to the desired value.

# Test Calibrated SCA

|  |  |  |
| --- | --- | --- |
|  | Predicted | Measured |
| Vxd (V) | Vout (V) | Vout (V) |
| 1.3 | 0.83 | 0.79 |
| 1.4 | 1.67 | 1.64 |
| 1.5 | 2.50 | 2.44 |
| 1.6 | 3.33 | 3.20 |
| 1.7 | 4.17 | 4.06 |

Table 1 - Predicted and Measured Output Voltage of Calibrated SCA

A graph with a line and numbers

Description automatically generated

Figure 4 - MATLAB PLOT Vxd vs SCA Output From -1 g to 1 g

% DAQ Configuration

daqSession = daq.createSession('ni'); % 'ni' for National Instruments, adjust if different DAQ

addAnalogInputChannel(daqSession, 'myDAQ1', 'ai0', 'Voltage'); % Replace 'Dev1' and 'ai0' with your device and channel ID

addAnalogInputChannel(daqSession,"myDAQ1","ai1","Voltage");

% Set DAQ properties

daqSession.Rate = 100; % Set the sampling rate (samples per second)

% Continuous loop to read and calculate tilt angle

while true

% Read Vo from DAQ (single scan)

Vo = inputSingleScan(daqSession);

Vxd = inputSingleScan(daqSession);

% Calculate g from Vxd, then limit g within [-1, 1] to avoid out-of-range errors

Vo = (25/3) \* Vxd - 10;

g = (Vxd – 1.5) / 0.3

g = max(min(g, 1), -1); % Limit g between -1 and 1

% Calculate tilt angle in degrees

theta = asind(g); % arcsin of g to find theta in degrees

% Display results

str = sprintf('Vxd = %.2f, 'Vo = %.2f, g = %.2f, theta = %.1f°', Vxd, Vo, g, theta);

disp(str);

pause(0.5); % Slow down display for readability

end

Vxd = 1.18, Vo = -0.17, g = -1.00, theta = -90.0°

Vxd = 1.17, Vo = -0.21, g = -1.00, theta = -90.0°

Vxd = 1.18, Vo = -0.16, g = -1.00, theta = -90.0°

Vxd = 1.21, Vo = 0.08, g = -0.97, theta = -85.9°

Vxd = 1.20, Vo = 0.03, g = -0.98, theta = -87.3°

Vxd = 1.25, Vo = 0.41, g = -0.83, theta = -76.1°

Vxd = 1.31, Vo = 0.88, g = -0.64, theta = -61.0°

Vxd = 1.37, Vo = 1.42, g = -0.43, theta = -42.0°

Vxd = 1.43, Vo = 1.95, g = -0.22, theta = -21.8°

Vxd = 1.59, Vo = 3.27, g = 0.30, theta = 30.3°

Vxd = 1.61, Vo = 3.41, g = 0.36, theta = 35.6°

Vxd = 1.67, Vo = 3.88, g = 0.55, theta = 52.7°

Vxd = 1.70, Vo = 4.17, g = 0.68, theta = 62.6°

Vxd = 1.74, Vo = 4.50, g = 0.80, theta = 73.3°

Vxd = 1.76, Vo = 4.68, g = 0.87, theta = 78.8°

Vxd = 1.77, Vo = 4.76, g = 0.90, theta = 81.2°

Vxd = 1.78, Vo = 4.80, g = 0.92, theta = 82.4°

Vxd = 1.77, Vo = 4.78, g = 0.91, theta = 81.8°

Vxd = 1.78, Vo = 4.83, g = 0.93, theta = 83.2°

Vxd = 1.78, Vo = 4.84, g = 0.94, theta = 83.5°

Vxd = 1.78, Vo = 4.83, g = 0.93, theta = 83.2°

Vxd = 1.77, Vo = 4.79, g = 0.92, theta = 82.1°

Vxd = 1.77, Vo = 4.79, g = 0.92, theta = 82.1°

Vxd = 1.79, Vo = 4.88, g = 0.95, theta = 84.7°

Vxd = 1.79, Vo = 4.91, g = 0.96, theta = 85.6°

Vxd = 1.79, Vo = 4.94, g = 0.98, theta = 86.4°

Vxd = 1.80, Vo = 4.96, g = 0.98, theta = 87.0°

Vxd = 1.80, Vo = 5.01, g = 1.00, theta = 88.4°

Vxd = 1.82, Vo = 5.13, g = 1.00, theta = 90.0°

Vxd = 1.82, Vo = 5.20, g = 1.00, theta = 90.0°

# Conclusions

The lab successfully implemented a Signal Conditioning Amplifier (SCA) that accurately mapped the ADXL335 accelerometer output from -1g to +1g to a 0V–5V range. Through careful calibration, the SCA met the desired specifications, with experimental results closely aligning with the simulated predictions. This practical exercise effectively demonstrated the use of signal conditioning techniques to prepare sensor outputs for digital processing.