TEXAS INSTRUMENTS

3-V Accelerometer Featuring TLV2772

Application Report

1998

Advanced Analog Products

SLVA040

IMPORTANT NOTICE

Texas Instruments and its subsidiaries (TI) reserve the right to make changes to their products or to discontinue any product or service without notice, and advise customers to obtain the latest version of relevant information to verify, before placing orders, that information being relied on is current and complete. All products are sold subject to the terms and conditions of sale supplied at the time of order acknowledgement, including those pertaining to warranty, patent infringement, and limitation of liability.

TI warrants performance of its semiconductor products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are utilized to the extent TI deems necessary to support this warranty. Specific testing of all parameters of each device is not necessarily performed, except those mandated by government requirements.

CERTAIN APPLICATIONS USING SEMICONDUCTOR PRODUCTS MAY INVOLVE POTENTIAL RISKS OF DEATH, PERSONAL INJURY, OR SEVERE PROPERTY OR ENVIRONMENTAL DAMAGE ("CRITICAL APPLICATIONS"). TI SEMICONDUCTOR PRODUCTS ARE NOT DESIGNED, AUTHORIZED, OR WARRANTED TO BE SUITABLE FOR USE IN LIFE-SUPPORT DEVICES OR SYSTEMS OR OTHER CRITICAL APPLICATIONS. INCLUSION OF TI PRODUCTS IN SUCH APPLICATIONS IS UNDERSTOOD TO BE FULLY AT THE CUSTOMER'S RISK.

In order to minimize risks associated with the customer's applications, adequate design and operating safeguards must be provided by the customer to minimize inherent or procedural hazards.

TI assumes no liability for applications assistance or customer product design. TI does not warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right of TI covering or relating to any combination, machine, or process in which such semiconductor products or services might be or are used. TI's publication of information regarding any third party's products or services does not constitute TI's approval, warranty or endorsement thereof.

Application Report

Jim Karki
Advanced Analog Products
Advanced Analog Applications Group

Abstract

This paper describes a complete solution for digital measurement of acceleration. An AMP accelerometer sensor is used for the conversion between mechanical acceleration and electrical analog. This electrical signal is then conditioned using Texas Instruments' TLV2772 op amp (on the Universal Op Amp EVM), digitized using the TLV1544 ADC EVM, and processed with the TMS320C5X EVM. This provides the user with a quick and easy way to evaluate a complete 3-axis accelerometer solution.

Contents

TEXAS INSTRUMENTS

Figures

Tables

1 Introduction

Accelerometers are used in aerospace, defense, automotive, household appliances, instrumentation, audio, transport, material handling, etc. This application report develops a data collection system for acceleration in 3 axis.

Figure 1 shows a block diagram of a typical analog data collection system. This application presents information about the sensor, signal conditioning, ADC, processor, display, and memory.

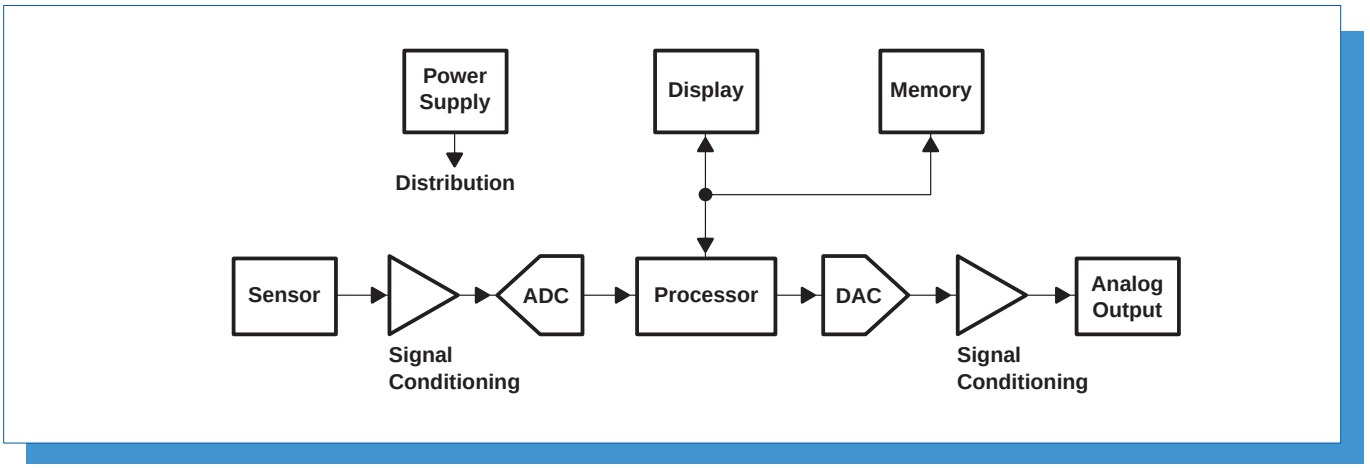

Figure 1–1. Typical Analog Data Collection System

2 System Description

Figure 2.1 shows the accelerometer system diagram

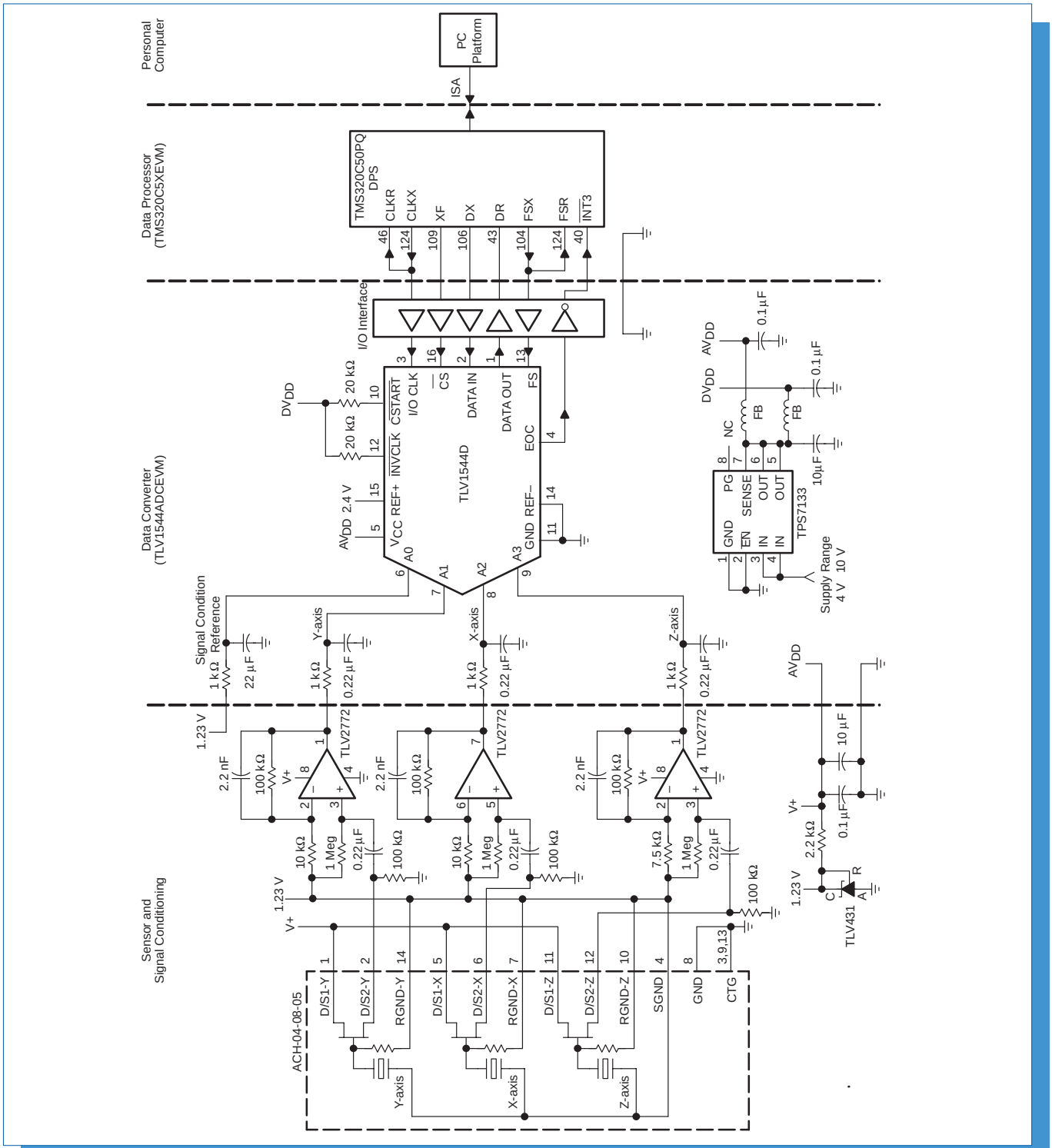

Figure 2–1. Accelerometer System Diagram

2.1 Sensor

An AMP ACH04-08-05 shock sensor converts mechanical acceleration into electrical signals. The shock sensor contains three piezoelectric sensing elements oriented to simultaneously measure acceleration in three orthogonal linear axes. The sensor responds from 0.5 Hz to above 5 kHz. An internal JFET buffers the output. Typical output voltage for x and y axis is 1.80 mV/g. Typical output voltage for the z axis is 1.35 mV/g. Refer to AMP's web site at http://www.amp.com/sensors for in-depth information about this sensor, piezo materials in general, and other related products.

2.2 Signal Conditioning

Circuitry using the Texas Instruments TLV2772 operational amplifier provides amplification and frequency shaping of the shock sensor output. Due to its high slew rate and bandwidth, rail-to-rail output swing, high input impedance, high output drive and excellent dc precision the TLV2772 is ideal for this application. The device provides 10.5 V/μs slew rate and 5.1 MHz gain bandwidth product while consuming only 1 mA of supply current per amplifier. The rail-to-rail output swing and high output drive make this device ideal for driving the analog input to the TLV1544 analog-to-digital converter. The amplifier typically has 360 μV input offset voltage, 17 nV/vHz input noise voltage, and 2 pA input bias current. Refer to Texas Instruments' web site at http://www.ti.com and search on TLV2772 to download a TLV2772 data sheet, literature #SLOS209.

The Universal Operational Amplifier EVM is used to construct the ACH04-08-05 shock sensor and TLV2772 operational amplifier circuitry. The Universal Operational Amplifier EVM facilitates construction of surface mount operational amplifier circuits for engineering evaluation. Refer to Texas Instruments' web site at http://www.ti.com to download the Universal Operational Amplifier EVM User's Manual, literature #SLVU006.

2.3 ADC

The TLV1544 EVM provides analog-to-digital conversion. The TLV1544 is a low-voltage (2.7 V to 5.5 V dc single supply), 10-bit analog-to-digital converter (ADC) with serial control, 4 analog inputs, conversion time = 10 μs, and programmable 1 μA power down mode. Refer to Texas Instruments' web site at http://www.ti.com to download a TLV1544 data sheet, literature #SLAS139B, TLV1544 EVM User's Manual, literature #SLAU014, and related information.

2.4 Processor, Memory, and Display

The TMS320C5x EVM controls and collects data samples from the TLV1544. Refer to Texas Instruments' web site at http://www.ti.com to download the TMS320C5x EVM Technical Reference, literature #SPRU087, and related information .

The TMS320C5x EVM installs into a PC platform. The PC provides programming and control of the TMS320C5x EVM , and provides resources for file storage or other processing of the collected data.

3 System Specification Requirements

The following system specification requirements were derived to guide the design.

3.1 G-Force Measurement Requirements

Range: ±50g

Noise: 0.05g pk-pk (equivalent input noise measured in g)

Resolution: 0.1g

Frequency Response: 1 Hz to 500 Hz (min 3 dB BW)

3.2 Power Requirements

Input Voltage: 3 V ±10%, Noise = 30 mV pk-pk (20 MHz BW)

Input Current: 5 mA (max) (power requirements for sensor and signal conditioning circuitry)

4 Sensor and Signal Conditioning Design

Circuit design is a 3-step process:

1. Hand analysis
2. SPICE simulation
3. Circuit breadboard and lab testing

The signal from the sensor must be amplified and frequency shaped to provide a signal that the ADC can properly convert into a digital number.

The schematic in Figure 4–1 shows the topology used in this application for 1 axis of the sensor and signal conditioning circuit.

Input power is 3 V and ground. The TLV431 precision voltage regulator, when configured as shown, produces a nominal 1.23-V reference voltage. This voltage provides the signal reference for the signal conditioning circuitry and the bias voltage for the internal JFETs in the shock sensor.

The transfer function of the signal conditioning circuit is derived by several means, the easiest of which may be by using super position. Perform a dc analysis, perform an ac analysis, and superimpose the results.

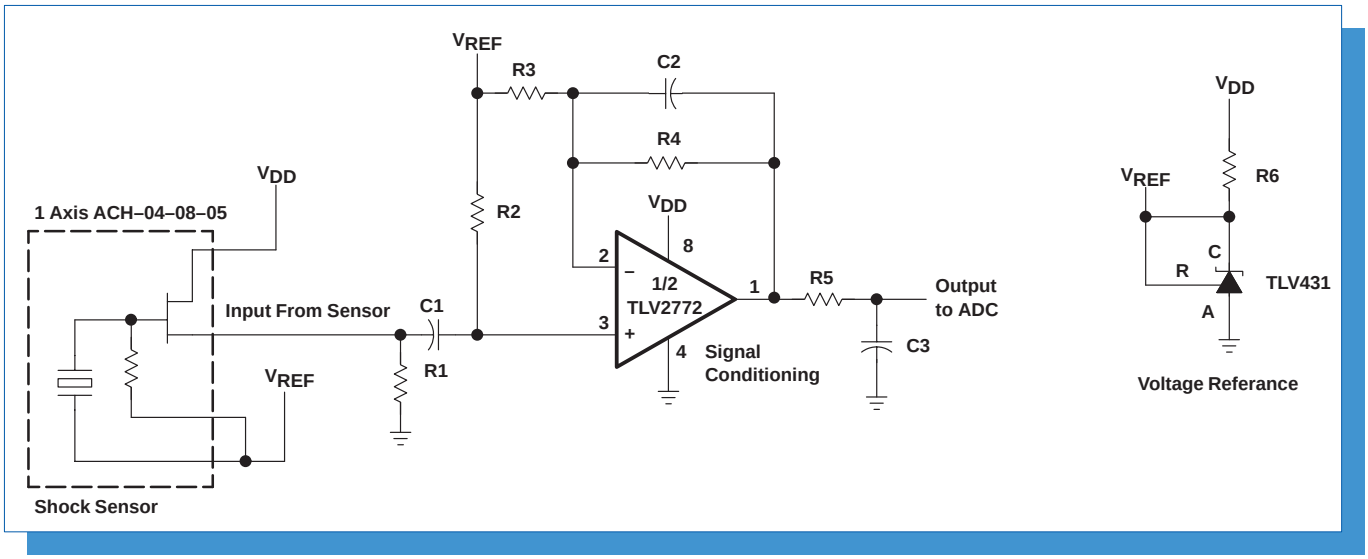

Figure 4–1. 1-Axis Accelerometer Sensor and Signal Conditioning Circuit

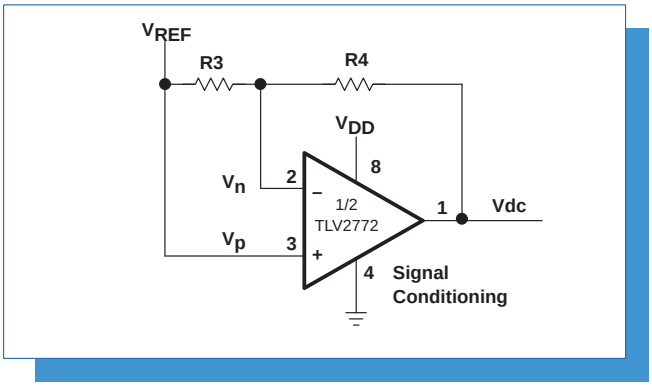

Figure 4–2. DC Circuit Model

4.1 Hand Analysis

In hand analysis, simplifying assumptions make solutions easier to derive. If the circuit does not function as anticipated, these assumption must be reevaluated.

4.1.1 DC Analysis

To perform a dc analysis, assume all inductors are short circuits and all capacitors are open circuits.

Assume that the resistance of R2 is insignificant compared to the input impedance of the op amp. Therefore, Vref will appear at the positive input to the op amp.

Assuming the ADC input does not impose a significant load on the circuit, the voltage divider formed between the ADC input and R5 can be disregarded.

The dc model shown in Figure 4–2 is based on the assumptions made above.

The gain of the amplifier with reference to the negative input, Vn, is:

$$\frac{Vdc}{Vref} = -\frac{R4}{R3} \quad or \quad Vdc = -Vref\,\frac{R4}{R3}$$

The gain of the amplifier with reference to the positive input, Vp is:

$$\frac{Vdc}{Vref} = \left(1 + \frac{R4}{R3}\right) \quad or \quad Vdc = Vref\,\left(1 + \frac{R4}{R3}\right)$$

Superimposing the positive and negative dc gains of the amplifier results in:

$$Vdc = Vref$$

The output of the amplifier is referenced to Vref. The ac response is superimposed upon this dc level.

4.1.2 AC Analysis

For ac analysis break the circuit into 3 parts and determine the transfer functions;

- H1(s) = Vp / Vi
- H2(s) = Vo / Vp
- H3(s) = Vadc / Vo

Combine the results to obtain the overall transfer function:

- H(s) = Vadc / Vi

Where Vi is the input signal from the shock sensor and Vadc is the output signal to the analog-to-digital converter. Figure 4–3 shows the ac model with the dc sources shorted.

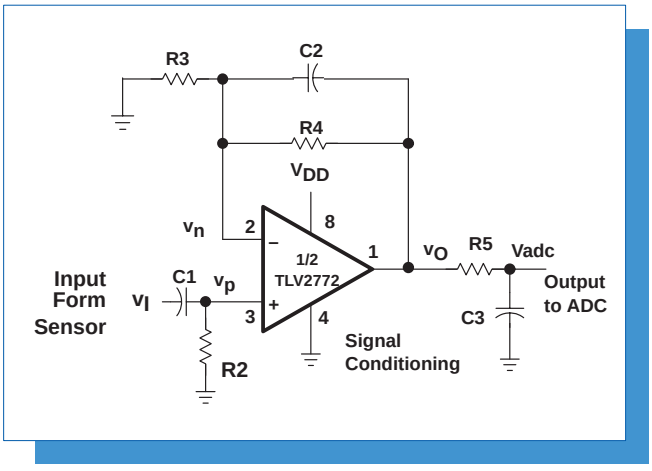

Figure 4–3. AC Circuit Model

4.1.2.1 H1(s)=Vp /Vi

Capacitor C1 and resistor R2 form a passive high-pass filter from the sensor input to the high impedance positive input to the TLV2772 where:

$$Vp = Vi\,\frac{R2}{R2 + 1/sC1} \quad or$$

$$H1(s) = \frac{Vp}{Vi} = \frac{R2}{R2 + 1/sC1}$$

4.1.2.2 H2(s) = Vo / Vp

The amplifier gain is found by solving for H2(s) = Vo/Vp. The solution is a non-inverting amplifier with:

$$H2(s) = \frac{Vo}{Vp} = \left(1 + \frac{Z}{R3}\right)$$

Where:

$$Z = \frac{R4}{1 + sC2R4}$$

substituting

$$H2(s) = \frac{Vo}{Vp} = 1 + \frac{R4}{R3}\left(\frac{1}{1 + sC2R4}\right)$$

4.1.2.3 H3(s) = Vadc / Vo

Assuming that the input impedance to the TLV1544 ADC is very high in comparison to the impedance of C3 and R5, C3 and R5 form a passive low-pass filter where:

$$Vadc = Vo\,\frac{1}{1 + sC3R5} \quad or$$

$$H3(s) = \frac{Vadc}{Vo} = \frac{1}{1 + sC3R5}$$

4.1.2.4 H(s) = Vadc / Vi

Superimposing the results from above gives the overall transfer function:

$$H(s) = \frac{Vadc}{Vi} = \left(\frac{R2}{R2 + 1/sC1}\right) \times \left(1 + \frac{R4}{R3}\left(\frac{1}{1 + sC2R4}\right)\right) \times \left(\frac{1}{1 + sC3R5}\right)$$

To find the complete response, add the ac and dc components so that:

$$Vadc = Vi\,H(s) + Vref$$

4.1.3 Gain Calculation

Since the TLV2772 is capable of rail-to-rail output, with a 3 V supply, Vout min = 0 V and Vout max = 3 V. With no signal from the sensor, Vout nom = reference voltage = 1.23 V. Therefore, the maximum negative swing from nominal is 0 V – 1.23 V = –1.23 V, and the maximum positive swing is 3 V – 1.23 V = 1.77 V.

Model the shock sensor as a low impedance voltage source with output of 2.25 mV/g max in the x and y axis and 1.70 mV/g max in the z axis, and calculate the required amplification of the signal conditioning circuit as follows:

Gain = Output Swing ÷ (Sensor Sensitivity · Acceleration)

To avoid saturating the op amp, base the gain calculations on the maximum negative swing of –1.23 V and the maximum sensor output of 2.25 mV/g for the x and y axis, and 1.70 mV/g for the z axis. Therefore:

Gain(x,y) = –1.23 V ÷ (2.25 mV/g × (–50g)) = 10.9 and
Gain(z) = –1.23 V ÷ (1.70 mV/g × (–50g)) = 14.5

Choosing R3 = 10 kΩ and R4 = 100 kΩ, gives a gain of 11 in the x and y channels. Choosing R3 = 7.5 kΩ and R4 = 100 kΩ, gives a gain of 14.3 in the z channel.

4.1.4 Bandwidth Calculations

To calculate the component values for the frequency shaping characteristics of the signal conditioning circuit, use 1 Hz and 500 Hz as the minimum required 3-dB bandwidth from the specifications requirements.

4.1.4.1 Component Values for H1(s) = Vp / Vi

To minimize the value of the input capacitor required to set the lower cutoff frequency, a large value resistor is required for R2. A 1 MΩ resistor is used here. To set the lower cutoff frequency, the value for capacitor C1 must be:

C1 = 1 ÷ (2π × lower cutoff frequency (Hz) × R2 (Ω))

C1 = 1 ÷ (6.28 × 1 Hz × 1 MΩ) = 0.159 µF

A more common 0.22-µF capacitor is used for C1. This moves the lower cutoff frequency to 0.724 Hz.

Figure 4–4 shows the bode plot approximation to the input transfer function, H1(s) = Vp/Vi.

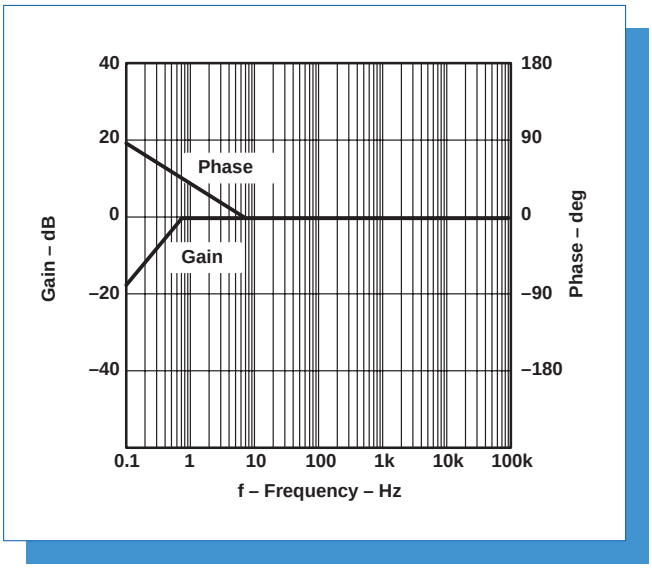

Figure 4–4. Bode Plot H1(s) = Vp/Vi

4.1.4.2 Component Values for H2(s) = Vo/Vp

To minimize phase shift in the feedback loop caused by the input capacitance of the TLV2772, it is best to minimize the value of feedback resistor R4. Also, to reduce the required capacitance in the feedback loop, a large-value resistor is required for R4. A compromise value of 100 kΩ is used for R4. To set the upper cutoff frequency, the required capacitor value for C2 is:

C2 = 1 ÷ (2π × upper cutoff frequency (Hz) × R4 (Ω))

C2 = 1 ÷ (6.28 × 500 Hz × 100 kΩ) = 3.18 nF

A more common 2.2 nF capacitor is used for C2. This changes the upper cutoff frequency to 724 Hz.

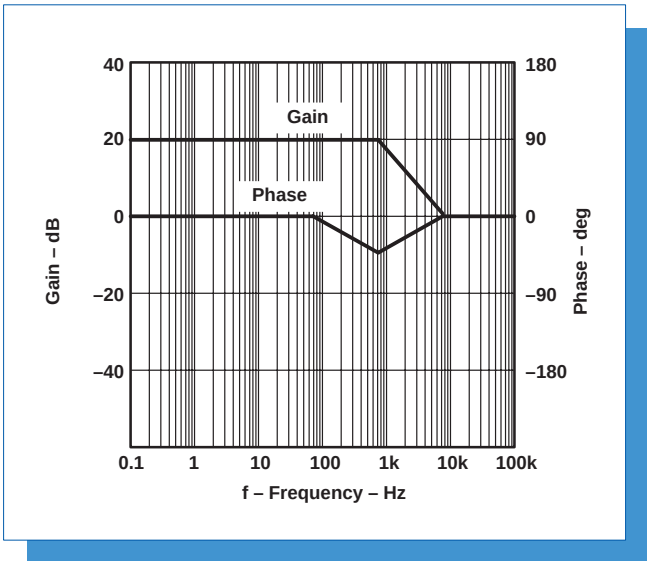

Figure 4–5. Bode Plot of H2(s) = Vo/Vp

Using the amplifier gain calculated above, Figure 4–5 shows the bode plot approximation to the transfer function H2(s) = Vo/Vp for the x and y channels. The z channel is the same except the gain is slightly higher in the pass band.

4.1.4.3 Component Values for H3(s) = Vadc/Vo

Resistor R5 and capacitor C3 cause the signal response to roll-off further. To set the frequency for this roll off to begin at the upper cutoff frequency, select 1÷(2π·C3(F) × R4(Ω)) = upper cutoff frequency(Hz).

With R5 = 1 kΩ and C3 = 0.22 µF, the roll-off frequency = 1 ÷ (6.28 × 0.22 µF × 1 kΩ) = 724 Hz.

Figure 4–6 shows the bode plot approximation to the transfer function H3(s) = Vadc/Vo.

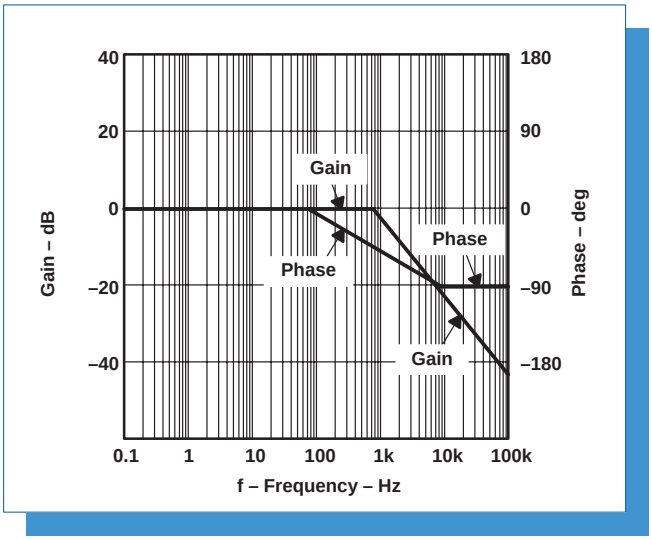

Figure 4–6. Bode Plot of H3(s) = Vadc/Vo

4.1.4.4 Transfer Function H(s) = Vadc / Vi

Superimposing the previous bode plot approximations results in the bode plot approximation for the overall transfer function to be expected from the signal conditioning circuit as shown in Figure 4–7.

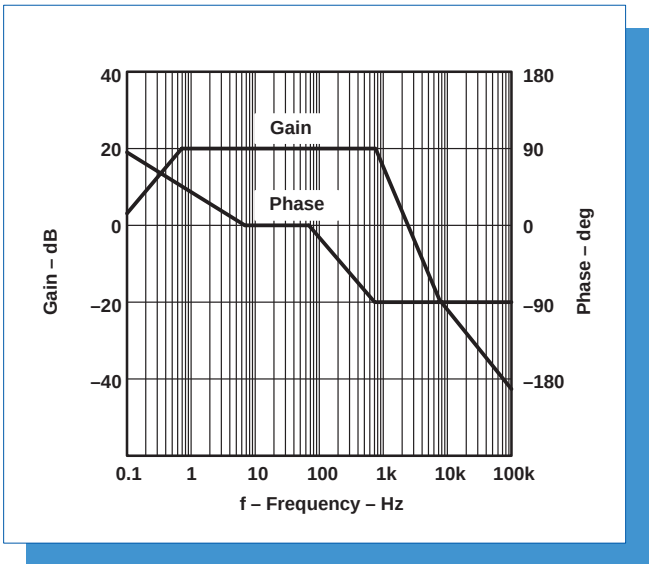

Figure 4–7. Bode Plot of H(s) = Vadc/Vi

4.2 Spice Simulation

Spice simulation verifies the results of hand analysis and provides a more accurate result than what is practical by hand.

The proper models must be used to perform SPICE simulation of the shock sensor and signal conditioning circuit.

The data sheet for the ACH-04-05-08 shock sensor states that the internal JFET used to drive the output is similar to the industry standard 2N4117. To model the sensor, a signal source is used to drive the gate of a 2N4117 JFET and proper bias is applied.

Modeling the signal conditioning circuit is straight forward except that most available SPICE versions do not have a library model for the TLV2772. This is easily remedied. Place a similar part on the schematic and modify its model with the model editor to match that of the TLV2772. Figure 4–8 below shows the TLV2772 sub-circuit model (as printed in the TLV2772 data sheet).

```
.SUBCKT TLV2772-X 1  2  3  4  5
    C1      11    12   2.3094E-12
    C2      6     7    8.0000E-12
    CSS     10    99   2.1042E-12
    DC      5     53   DY
    DE      54    5    DY
    DLP     90    91   DX
    DLN     92    90   DX
    DP      4     3    DX
    EGND    99    0    POLY(2)  (3,0)  (4,0)  0  .5  .5
    FB      7     99   POLY(5)  VB  VC  VE  VLP  VLN  0  19.391E6 – 1E3  1E3  19E6 – 19E6
    GA  6   0     11   12    150.80E-6
    GCM 0 6     10   99    7.5576E-9
    ISS     3     10   DC    116.40E-6
    HLIM    90    0    VLIM  IK
    J1      11    2    10    JX1
    J2      12    1    10    JX2
    R2      6     9    100.0E3
    RD1     4     11   6.6315E3
    RD2     4     12   6.6315E3
    RO1     8     5    17.140
    RO2     7     99   17.140
    RP      3     4    4.5455E3
    RSS     10    99   1.7182E6
    VB      9     0    DC 0
    VC      3     53   DC .1
    VE      54    4    DC .1
    VLIM    7     8    DC 0
    VLP     91    0    DC 47
    VLN     0     92   DC 47
    .MODEL DX D(IS=800.0E-18)
    .MODEL DY D(IS=800.0E-18  Rs=1m  Cjo=10p)
    .MODEL JX1 PJF(IS=2.250E-12  BETA=195.36E-6  VTO=–1)
    .MODEL JX2 PJF(IS=1.750E-12  BETA=195.36E-6  VTO=–1)
    .ENDS
```

Figure 4–8. TLV2772 Sub-Circuit Model

Figure 4–9 shows the schematic used for SPICE simulation. Figures 4–10 show the simulation results.

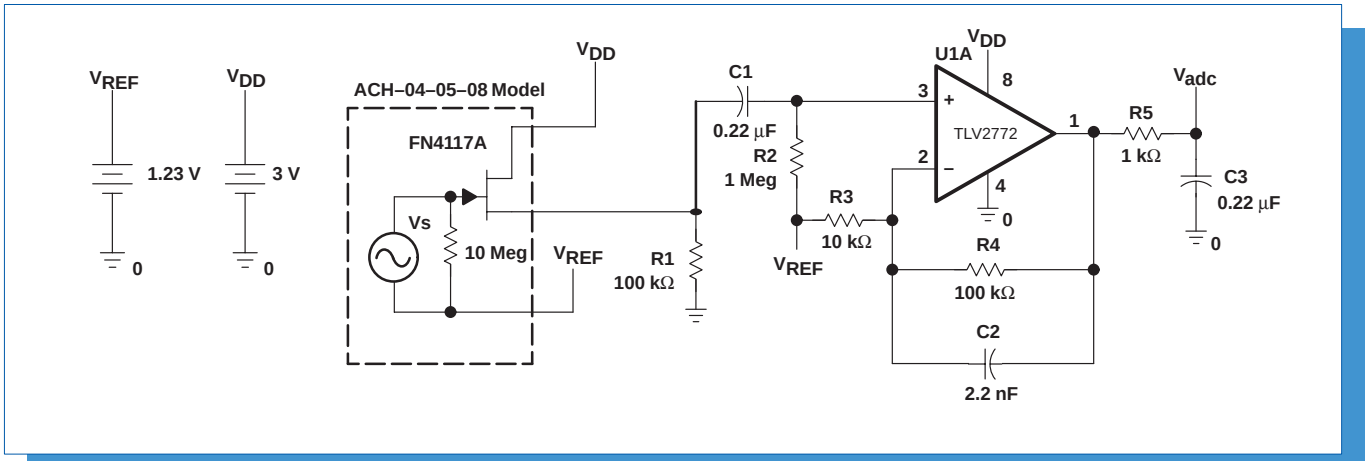

Figure 4–9. SPICE Simulation Schematic

The SPICE simulations show agreement with the hand calculations and the bode plot approximations.

The analyses above do not consider the frequency response of the shock sensor, which falls off below 0.5 Hz and above 5 kHz. The data sheet for the ACH-04-08-05 does not specify the role-off characteristics of the shock sensor's frequency response, but in general it is expected that there will be more attenuation in the signal below 0.5 Hz and above 5 kHz.

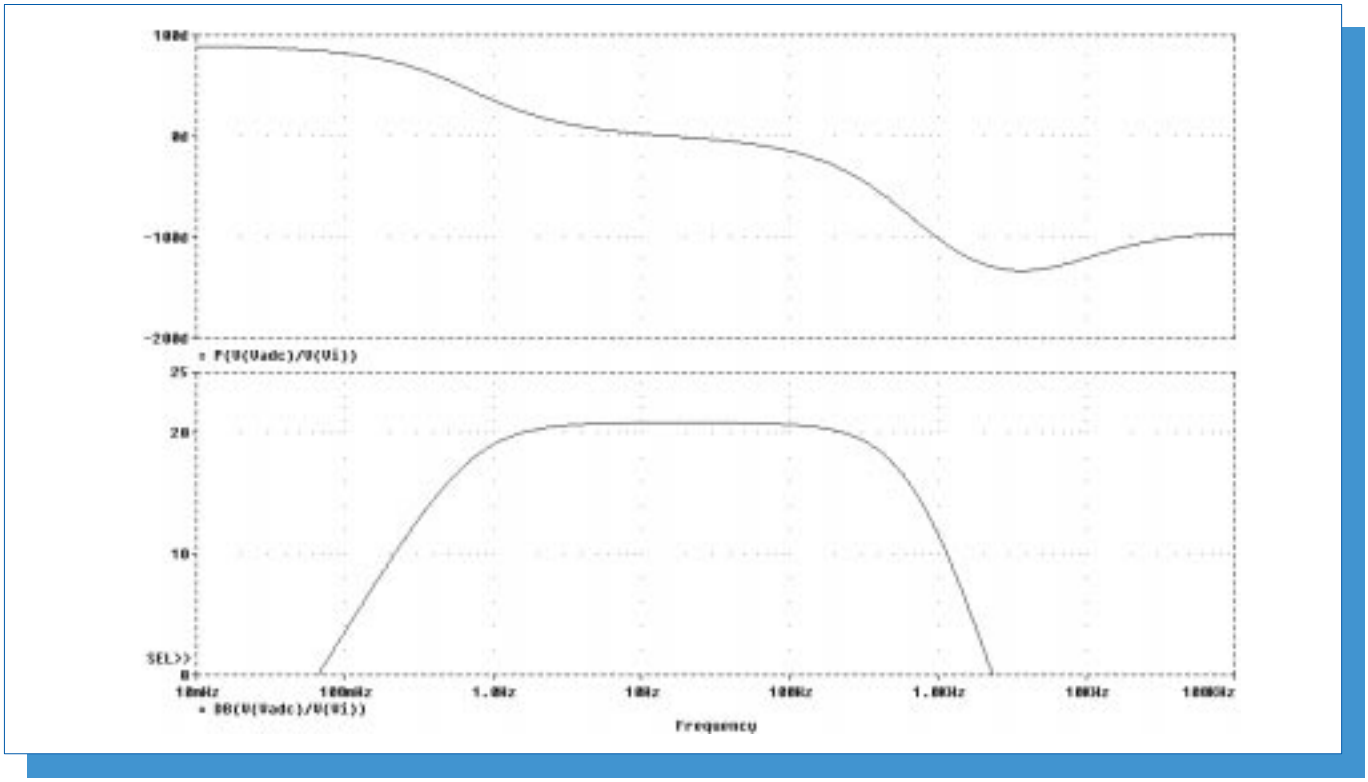

Figure 4–10. SPICE Simulation Results

5 Circuit Realization

The shock sensor and signal conditioning circuits are built on area 100 of two Universal Operational Amplifier EVM boards (SLOP 120–1). One EVM board holds a TLV2772, an ACH-04-05-08, a TLV431, and required ancillary devices. The other board has only a TLV2772 and required ancillary devices. The signal conditioning circuit is constructed by installing required components and wiring. The two boards share signals and sources through board-to-board connectors. Standoffs and screws secure the two boards together. Figure 5–1 shows the schematic diagram with reference designators for using area 100 on two Universal Operational Amplifier EVM boards.

Tables 5–1 and 5–2 summarize reference designator part descriptions to construct the shock sensor and signal conditioning circuits.

Table 5–1. Board 1 – Universal Operational Amplifier EVM Area 100

REFERENCE DESIGNATOR	DESCRIPTION	REFERENCE DESIGNATOR	DESCRIPTION
R101	1 MΩ 1% SMT	C101	Not used
R102	10 kΩ 1% SMT	C102	2200 pF 5% NPO SMT
R103	Not used	C103	2.2 µF 20% Y5V SMT
R104	100 kΩ 1% SMT	C104	0.1 µF 10% X7R SMT
R105	100 kΩ 1% SMT	C105	2200 pF 5% NPO SMT
R106	Not used	C106	Not used
R107	10 kΩ 1% SMT	C107	Not used
R108	1 MΩ 1% SMT	C108	Not used
R109	Use 0.22 µF 10% X7R SMT Capacitor	C109	0 Ω or Jumper
R110	Use 0.22 µF 10% X7R SMT Capacitor	C110	Not used
R111	0 Ω or Jumper	C111	Not used
R112	Not used	C112	Not used
R113	Not used	U101	TLV2772CD
R114	2.2 kΩ 5% SMT	U102	TLV431ACDBV5
R115	0 Ω or Jumper	V1+	3 V power input
R116	Not used	VREF1	Signal conditioning reference
R117	Not used	GND1	Signal and power ground
R118	Not used	R119	0 Ω or Jumper
A1OUT	X Axis output	B104+	Y Axis input form ACH-04-05-08 pin 2
A101–	Not used	B103+	Jump to VREF1
A102–	Jump to VREF1	B102–	Jump VREF1
A103+	Jump to VREF1	B101–	Not used
A104+	X Axis input from ACH-04-05-08 pin 6	B1OUT	Y Axis output

Table 5–2. Board 2 – Universal Operational Amplifier EVM Area 100

REFERENCE DESIGNATOR	DESCRIPTION	REFERENCE DESIGNATOR	DESCRIPTION
R101	Not used	C101	Not used
R102	Not used	C102	Not used
R103	Not used	C103	Not used
R104	0 Ω or Jumper	C104	0.1 µF 10% X7R SMT
R105	100 kΩ 1% SMT	C105	2200 pF 5% NPO SMT

Table 5–2. Board 2 – Universal Operational Amplifier EVM Area 100

REFERENCE DESIGNATOR	DESCRIPTION	REFERENCE DESIGNATOR	DESCRIPTION
R106	Not used	C106	Not used
R107	7.5 kΩ 1% SMT	C107	Not used
R108	1 MΩ 1% SMT	C108	Not used
R109	Use 0.22 μF 10% X7R SMT Capacitor	C109	0 Ω or Jumper
R110	Not used	C110	Not used
R111	Not used	C111	Not used
R112	Not used	C112	Not used
R113	0 Ω or Jumper	U101	TLV2772CD
R114	Not used	U102	Not used
R115	Not used	V1+	3 V from Board 1
R116	Not used	VREF1	Signal conditioning reference from Board 1
R117	Not used	GND1	Circuit common connect to board 1 GND1
R118	Not used	R119	0 Ω or Jumper
A1OUT	Z Axis output	B104+	Not used
A101–	Not used	B103+	Not used
A102–	Jump to VREF1	B102–	Not used
A103+	Jump to VREF1	B101–	Not used
A104+	Z Axis input from ACH-04-05-08 pin 12	B1OUT	Not used

The ACH-04-08-05 shock sensor mounts in the breadboard area on board 1. Table 5–3 summarizes the connections.

Table 5–3. Board 1 – Universal Operational Amplifier EVM Area 100 ACH-04-08-05 Connections

ACH-04-08-05		
PIN	NAME	BOARD 1 CONNECTIONS
1	D/S1–Y	V1+
2	D/S2–Y	B104+
3	CTG	GND1
4	SGND	VREF1
5	D/S1–X	V1+
6	D/S2–X	A104+
7	RGND–X	VREF1
8	GND	GND1
9	CTG	GND1
10	RGND–Z	VREF1
11	D/S1/Z	V1+
12	D/S2–Z	To board 2
13	CTG	GND1
14	RGND–Y	VREF1

Board-to-board connectors provide required connections between board 1 and board 2. The required board-to-board connections are GND1, V1+, and VREF. Also, pin 12 of the ACH-04-08–05 must be connected to A104+ on board 2. This is accomplished by using board-to-board connectors in the breadboard area and connecting the appropriate nodes thereto.

Figure 5–1 is the schematic of the 3-axis realization of the 1-axis circuit shown in Figure 4–1, except for output filter components R5 and C3. To ease component placement, the output filters for each channel are placed on the TLV1544 EVM. For standalone testing of the signal conditioning circuit without the TLV1544 EVM, temporarily place the filter components on board 1 and board 2. Remove them for final integration.

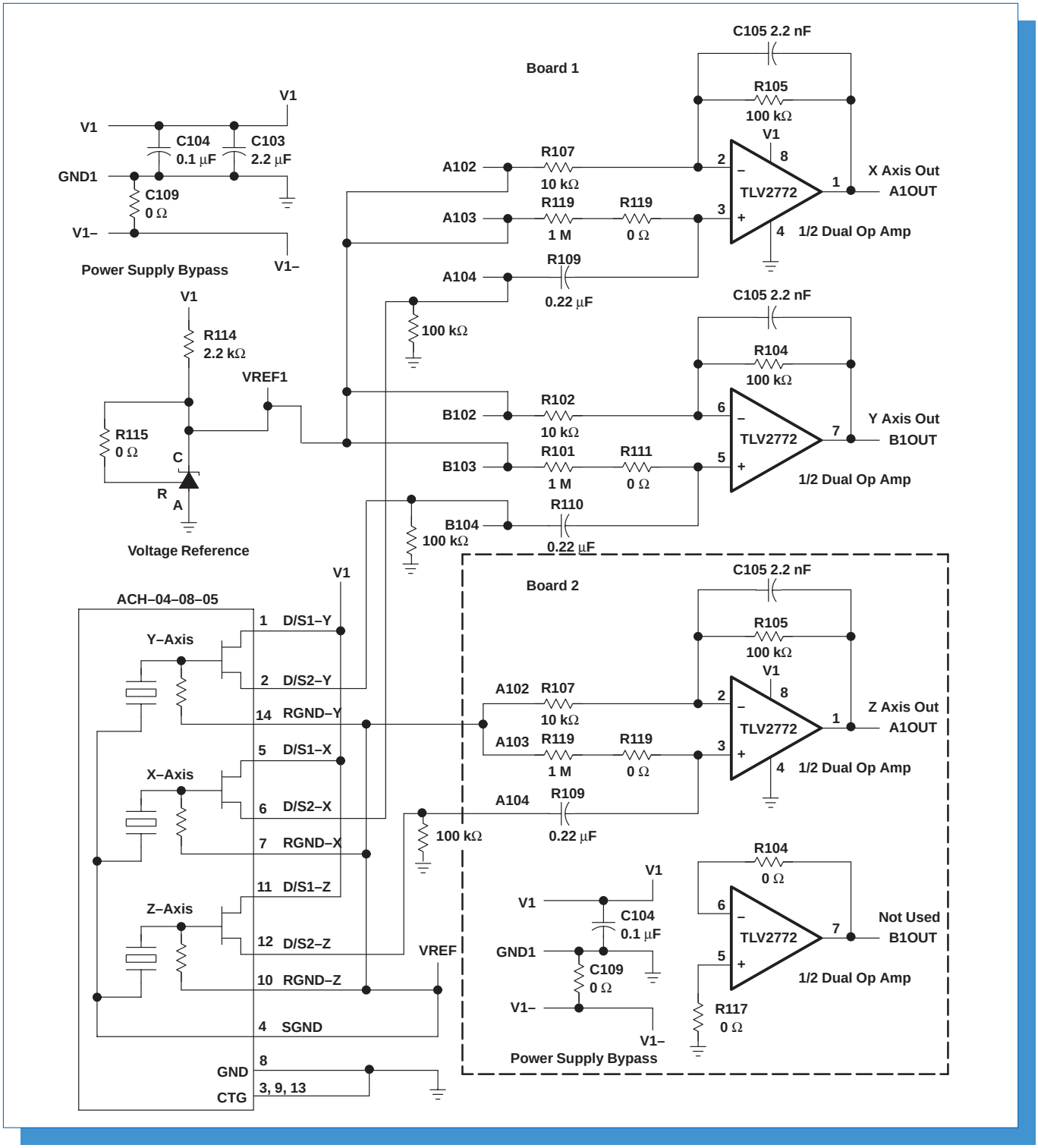

Figure 5–1. Signal Conditioning Schematic using Two Universal Operational Amplifier EVM Boards

5.1 Test of Signal Conditioning Circuit

After constructing the signal conditioning circuit, and before connecting the shock sensor, tests were run to verify the signal conditioning circuit performed as expected. An AP Instruments network analyzer was used to measure the gain and phase response. Figure 5–2 shows the results are very much the same as the SPICE simulation predicted.

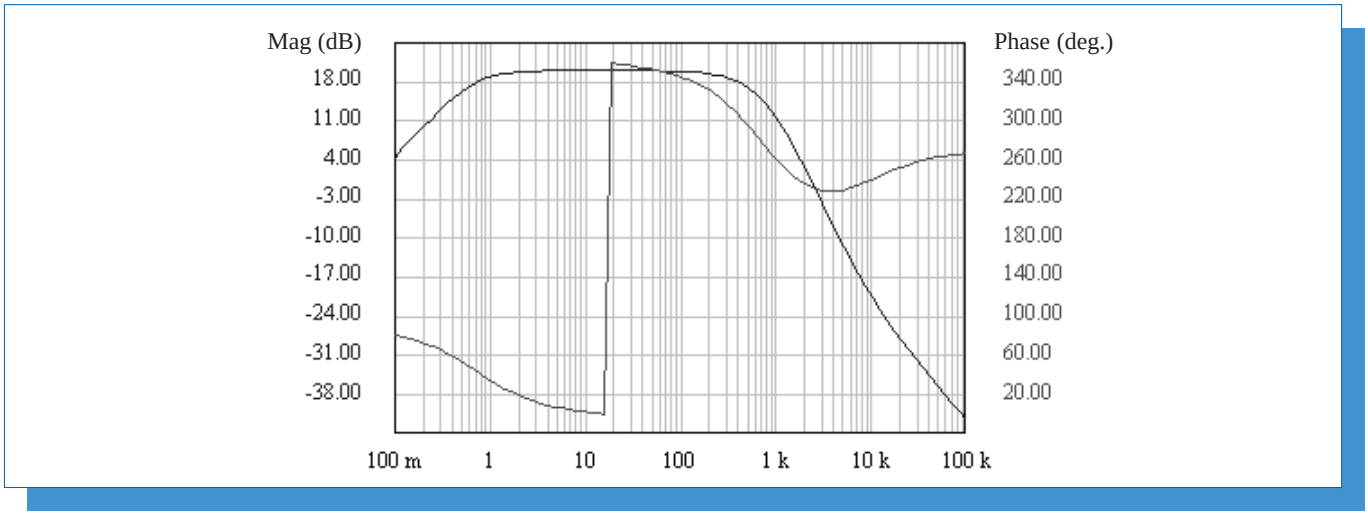

Figure 5–2. Network Analyzer Display

5.2 Test of Shock Sensor and Signal Conditioning Circuit

After connecting the shock sensor, the measurement of acceleration was verified. A spring loaded test fixture was constructed to induce simple harmonic motion into the sensor. Figure 5–3 shows a diagram of the test fixture. By measuring the deflection from center and the frequency of the oscillation, the acceleration is calculated as:

$$a = -\omega^2 \times \cos(\omega t + \phi)$$

Where

a is acceleration

ω is frequency of oscillation in radians

x is the maximum deflection from center

ϕ is the phase

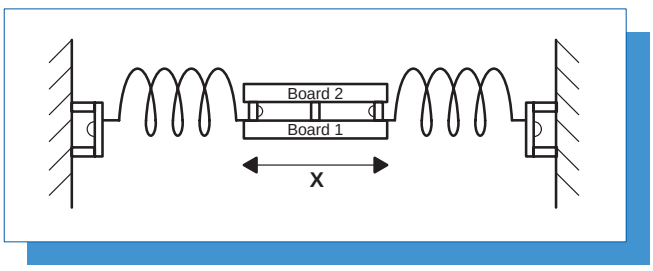

Figure 5–3. Spring Test Fixture

While monitoring the output of all 3 axes with an oscilloscope, the system was set into motion along the x-axis. Figure 5–4 shows the result displayed on an oscilloscope. Channel 1 is the x-axis, channel 2 is the y-axis, and channel 3 is the z-axis.

About 0.25” mechanical deflection was measured visually. An oscilloscope was used to measure the frequency of oscillation and the peak voltage amplitude. Using the above formula to find the peak acceleration expected from the mechanical movement (ignoring the sign):

$a\ (peak) = (2\pi \times 16)^2 \cdot (0.25 in/s^2) = 2524 in/s^2 = 6.58g$

Where $g = 32\ ft/s^2$

To check the output of the signal conditioning circuit, use the formula:

$$a\ (peak) = \frac{Peak\ voltage\ output}{Sensor\ output \cdot Gain}$$

$$= 128\ mV \frac{g}{1.80\ mV} \frac{1}{11} = 6.46\ g$$

Using the typical x–axis output = 1.80mV/g, there is good agreement between the two calculations.

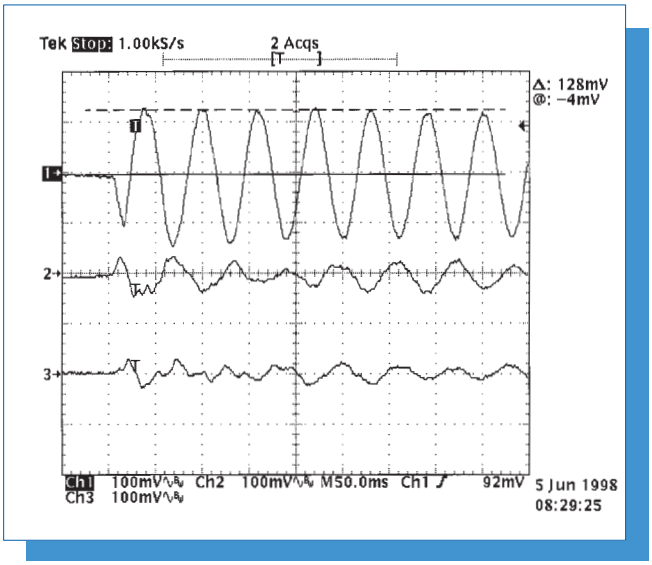

Figure 5–4. Output Displayed on Oscilloscope

Note that output is seen on the y-axis and z-axis. The ACH-04-08-05 data sheet specifies a transverse sensitivity of 15%. Fifteen percent of the x-axis peak signal output is 19.2 mV. The peak y axis signal is 40 mV and the peak z-axis signal is 26 mV. Transverse sensitivity does not account for the observation. While the primary motion was in the x-axis, motion was noticed in the other 2 axes as well.

5.3 TLV1544 EVM

For this application the fast conversion mode of the TLV1544 ADC is used and three of the four analog inputs are converted one after the other. This results in a sampling frequency of 28 ksps. By Nyquist's sampling theorem the frequency content of the input analog signal must be insignificant for frequencies ½ the sampling frequency and above. In this system, the signal is attenuated by over 40 dB at the Nyquist frequency.

5.3.1 Interfacing the Signal Conditioning Circuit to the TLV1544 EVM

The TLV1544 EVM provides a power supply, analog signal conditioning circuitry, ADC (TLV1544), and circuitry to support the interface to a DSP or micro controller.

Refer to the TLV1544 EVM User's Guide, literature #SLAU014, for detailed descriptions of the board and its functions.

To interface the shock sensor and signal conditioning circuit to the TLV1544 EVM, minor modifications are required on the TLV1544 EVM. To ease circuit realization, place the output filter for each axis signal, R5 and C3 in Figure 4–1, on the TLV1544 EVM.

The TLV1544 has 4 analog inputs. In this application, channel 0 samples the reference voltage from the signal conditioning circuit, channel 1 samples the x–axis output, channel 2 samples the y–axis output, and channel 3 samples the z-axis output. Make the following modifications to the TLV1544 EVM, REV–:

Channel 0 Modifications:

- Remove U4, R26, R27, and R28.
- Jumper U4–2 to U4–3.
- Place a 1-kΩ resistor for R28.
- Place a 22-µF capacitor between A0 and ground.

Channel 1 Modifications:

- Remove U5, R29, and C19.
- Jumper U5–3 to U5–6.
- Place a 1-kΩ resistor for R29.
- Place a 0.22-µF capacitor for C19.

Channel 2 Modifications:

- Place a 1-kΩ resistor for JP3.
- Place a 0.22-µF capacitor across D6.

Channel 3 Modifications:

- Place a 1-kΩ resistor for JP4.
- Place a 0.22-µF capacitor across D8.

Switch SW1 to operate the TLV1544 EVM at 3 V. Set the TLV1544 reference voltage input to 2.4 V by installing jumpers JP1–1, 2 and JP2–1, 2. Use the 3 V power supply on the TLV1544 EVM to supply power to the shock sensor and the signal conditioning circuit.

Table 5–4 shows the interface connections between the signal conditioning circuit and the TLV1544 EVM, REV–.

Table 5–4. Interface Between Signal Conditiioning Circuit and TLV1544 EVM

SIGNAL CONDITIONING CIRCUIT	TLV1544 EVM
Board 1 V1+	AV_{DD}
Board 1 GND1	GND
Board 1 VREF1 (signal reference)	JP19 (U2–2)
Board 1 A1OUT (X axis output)	Analog Input 1
Board 1 B1OUT (Y axis output)	Analog Input 2
Board 2 A1OUT (Z axis output)	Analog Input 3

Figure 5–5 is a schematic drawing for the interface between the signal conditioning circuit and the TLV1544.

5.4 Interfacing the TLV1544 EVM to the TMS320C5C EVM

Figure 5–6 shows the connections required to interface the TLV1544 EVM to the TMS320C5X EVM. For further details refer to *Interfacing the TLV1544 Analog-to-Digital Converter to the TMS320C50 DSP Applications Report* literature # SLAA025.

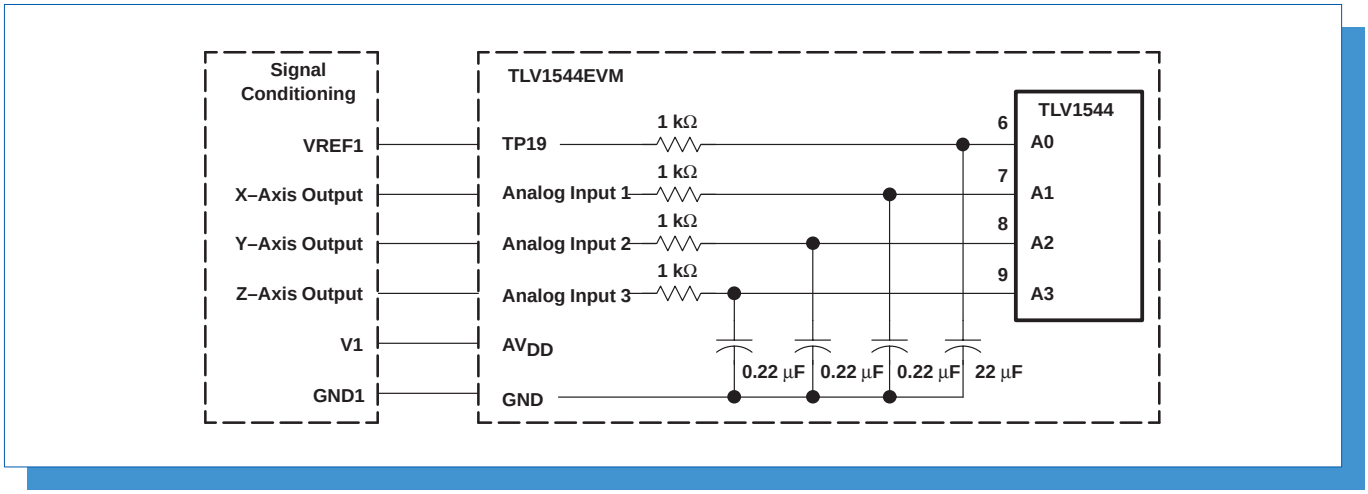

Figure 5–5. Schematic – Signal Conditioning to TLV1544 ADC

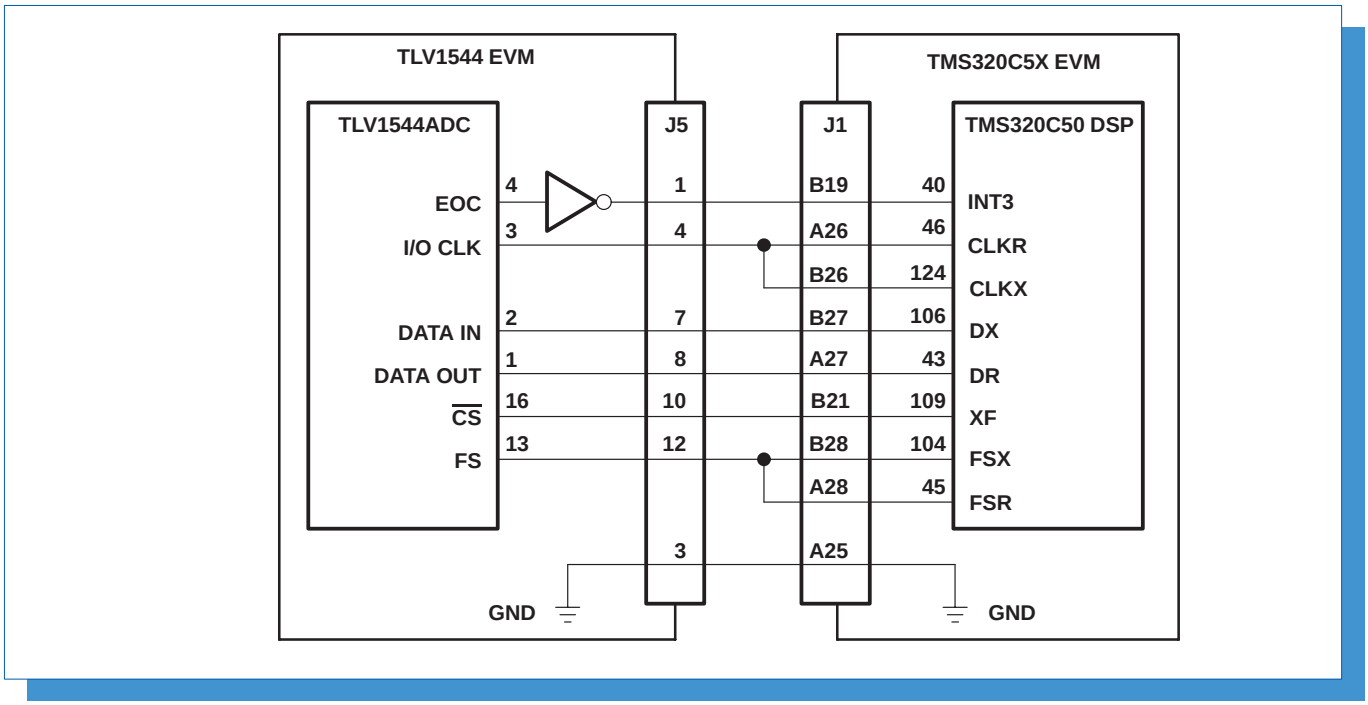

Figure 5–6. Interface Between TLV1544 EVM and TMS320C5X EVM

5.5 The TMS320C5X EVM

The TMS320C5X EVM plugs into an ISA slot in a PC. Refer to *TMS320C5X Evaluation Module Technical Reference* literature # SPRU087 for detailed information on its use. The TMS320C5X C Source Debugger is also required. Refer to *TMS320C5X C Source Debugger User's Guide* literature # SPRU055B for user information.

6 Error and Noise Analysis

Error analysis can predict the errors to be expected under normal operating conditions. Two analyses are presented: gain error and noise.

Gain errors contribute to uncertainty in the system output given a known input.

Noise is a random electrical event that occurs in all conductors. Noise imposes a statistical uncertainty on the system's electrical signals.

The system can be calibrated to compensate for the effect of gain errors. The uncertainty associated with noise, on the other hand, prohibits any type of compensation.

Two of the most widely used types of error analysis are worst case analysis and sum of squares analysis. The major task in performing an error analysis is to identify and quantify the error sources.

Worst case analysis assumes that all errors compound, one on top of the other. Therefore, the maximum error in all circuit elements is added to arrive at the total system error.

Sum of squares analysis assumes that some errors will tend to cancel each other. By summing the square of all the errors and taking the square root, the result takes this cancellation effect into account. A sum of squares analysis gives a realistic idea of how well the typical system will perform. This is the type of analysis that will be presented. A sum of squares analysis uses the following formula;

$$Total\ System\ Error = \sqrt{\sum_{n=1}^{k} Error_n^2}$$

Where the subscript n indicates an individual error component and k is the total number of error components.

System gain error analysis is presented first, followed by system noise analysis.

6.1 System Gain Error Analysis

Gain errors are errors that contribute to uncertainty in the system output, given a known input.

The system is essentially an ac system in which dc errors can be compensated. Therefore, dc errors caused by reference voltage errors, offset voltages, and the like are not included in this error analysis.

6.1.1 Sensor Gain Error

The shock sensor's output varies, and this contributes to uncertainty in the system output given a known input. The data sheet for the ACH-04-08-05 states the output sensitivity:

CHARACTERISTIC	MIN	TYP	MAX	UNIT
X-axis sensitivity	1.35	1.80	2.25	mV/g
Y-axis sensitivity	1.35	1.80	2.25	mV/g
Z-axis sensitivity	1	1.35	1.70	mV/g

This is a ±25% tolerance in the output voltage for the same acceleration input. Thus the error assigned to this component is 25%.

6.1.2 Signal Conditioning Gain Error

Errors in the signal conditioning circuit also contribute to uncertainty in the system output, given a known input. The analysis is limited to the pass band of the amplifier circuit. In its pass band the gain of the amplifier is given by:

$$A = \left(\frac{1}{b}\right)\left[\frac{1}{1 + \frac{1}{ab}}\right]$$

Where:

a = open loop amplification of the operational amplifier

And

$$b = \frac{R4}{R4 + R3}$$

The term 1/ab is seen as an error term in this equation because it is desired to set the gain of the circuit with the 1/b term (ratio of R3 to R4). To minimize this error, the open loop amplification of the operational amplifier should be as large as possible so that 1/ab is very close to 0. In this application the TLV2772's open loop amplification is the lowest at the upper cutoff frequency. At 1 kHz the open loop amplification is about 75 dB or 5600. Using this value: 1/ab = 0.000016, and the deviation of the amplifier gain from ideal is 0.002%.

The error caused by the op amp gain setting resistors is the tolerance of the resistors. In this case, 1% resistors add an error of 1%.

6.1.3 ADC Gain Errors

Errors caused by sampling of the signal by the ADC also need to be analyzed. Figure 6–1 shows a simplified schematic to use for analysis.

Set initial conditions where Vadc = Vo and Vsample =0. To simplify the analysis, notice that the time constant R5C3 = 0.22 ms is much longer than the time constant RadcCadc = 55 ns, because C3 is 4000 times larger than Cadc. In this system the TLV1544's I/O CLK is being driven at 5 MHz. The sampling period = 6 I/O CLK periods = 1.2 µs. Because C3 >> Cadc, when the switch closes during sampling, C3 acts like a voltage reservoir for charging Cadc via Radc. During the sampling period very little charge is transferred from Vo. The sampling period, 1.2 µs = 29 × 55 ns or 29 RadcCadc time constants. Therefore, at the end of the sampling the period, the voltage across Cadc and C3 is essentially equal, the sample voltage is set by the ratio of C3 to Cadc;

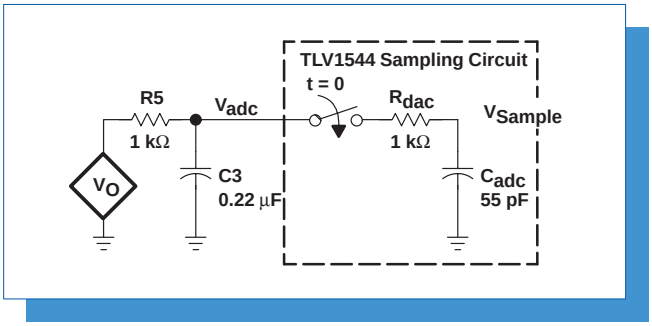

Figure 6–1. Sampling Input Model

$$Vsample = Vadc\left(\frac{C3}{Cadc + C3}\right) = 0.99975\ Vadc$$

The error attributed is 0.025%.

The other sources of error in the ADC can be read from the data sheet. As stated earlier, the system is essentially an ac system, and dc errors are not included in the error analysis. Therefore, only the maximum linearity error from the data sheet, which is ±1 LSB, is included. With 10 bits of resolution or 1024 codes the error = 0.0976%.

6.1.4 Total Gain Error

From this point on in the system, the signal is digital, and it is assumed there are no errors which cannot be corrected for.

Calculating the total error as the square root of the sum of the error squared results in:

$$\sqrt{25\%^2 + 0.002\%^2 + 1\%^2 + 0.025\%^2 + 0.0976\%^2} = 25.02\%$$

The uncertainty associated with the shock sensor output dominates the error analysis. Calibration is required to insure accurate measurement of acceleration.

6.2 System Noise Analysis

The first step to analyzing system noise is to qualify and quantify noise sources within the system. Next, the components must be added to see the overall effect on the system.

In this system the noise sources are the shock sensor, the resistors, and the amplifier. Since all three channels are essentially the same, the noise analysis is performed on one channel. Reference designators from Figure 4–1 are used with the component values as calculated above for channel z.

Because the z-axis of the shock sensor has about 30% less sensitivety than either x-axis or y-axis, 30% more gain is required in channel z. The results of the analysis show the dominant noise source is the sensor. The noise specification for the sensor is given in mg/$\sqrt{Hz}$ and converted to equivalent input noise voltage. Therefore, the noise voltage, as calculated in this analysis, will be higher in channel z, but not significantly so.

It is typical in amplifier circuits to identity equivalent input noise voltage per root Hertz for system components. This is then multiplied by the equivalent noise bandwidth and amplifier noise gain. The result is the total system noise. It is convenient to work in RMS values to compute the total system noise, then to convert to peak-to-peak (pk-pk) and compare to 1 LSB to see the effect on the ADC's output. The following formula is used:

$$\text{Total System Noise (in RMS Volts)} = \sqrt{\sum_{m=1}^{k} ENB \cdot en_m^2 \cdot An^2}$$

Where ENB is the equivalent noise bandwidth in hertz, the subscript m indicated an individual equivalent input noise source, en, in RMS volts per root Hertz, An is the noise gain of the amplifier, and k is the total number of noise sources referred to the input.

Normally, noise in RMS voltage is multiplied by numbers ranging from 5 to 7 to convert to peak-to-peak (pk-pk) voltage. This accounts for the large crest factor seen in noise signals. The pk-pk value of noise voltage is of concern when driving the input to an ADC converter. A multiplication factor of 6 is used in this analysis to convert RMS into pk-pk.

The equivalent noise bandwidth (ENB) accounts for the noise transmitted above the upper cutoff frequency as a result of the finite roll-off, or tail of the frequency response curve. For a 2nd order system, ENB is 1.11 times the normal bandwidth. For this system, ENB = 1.11(724 – 0.7) = 802.9 Hz. It is convenient and acceptable to use ENB = 803 Hz.

The amplifier noise gain is the noninverting gain of the amplifier, which is equal to 14.3.

6.2.1 Sensor Noise Analysis

From the ACH-04-08-05 data sheet the shock sensor has a specification for equivalent noise at 100 Hz equal to 0.2 mg/$\sqrt{Hz}$. Using the typical z-axis value for sensitivity, 1.35 mV/g, the equivalent noise in volts/$\sqrt{Hz}$ is 0.27 μV/$\sqrt{Hz}$. The data sheet gives no further details as to the spectral shape of the noise from the shock sensor. For this analysis, assume the noise voltage from the ACH-04-08-05 is flat over the bandwidth of the system.

6.2.2 Signal Conditioning Noise Analysis

For Vdd = 2.7 V, the TLV2772 data sheet specifies equivalent input noise voltage of 147 nV/vHz at 10 Hz and 21 nV/$\sqrt{Hz}$ at 1 kHz. Because the TLV2772 is a CMOS input op amp, the input noise current is insignificant with the circuit values being used.

The resistors in the signal conditioning amplifier's input circuit contribute thermal noise voltage:

$$= \sqrt{\frac{4kTR}{Hz}}$$

Where k is Boltzmann's Constant = 1.38×10^{-23}J/K

T is temperature in Kelvin

R is resistance in ohms

Hz if frequency in Hertz

R2 contributes 128 nV/$\sqrt{Hz}$. R3 and R4 appear in parallel at the negative input for noise calculations. Together they contribute 10.7 nV/$\sqrt{Hz}$. Resistor R1 adds 40 nV/$\sqrt{Hz}$.

Resistor R5 adds thermal noise voltage = $\sqrt{(4KTR/Hz)}$ to the system. The value of the resistor is small and the noise is not amplified by the amplifier. Therefore the effect of R5 on the overall system noise is negligible.

Adding the foregoing is straightforward for all components except the amplifier's input noise. Referring to the equivalent input noise voltage vs frequency graph in the data sheet for the TLV2772 (Figure 39) it can be seen that the equivalent input noise voltage is characterized by 1/f noise over the bandwidth of the system's operation. The input noise is a combination of the 1/f noise and the white noise of the op amp: $en^2 = (K^2/f)+e_w^2$ where e_n is the equivalent input noise voltage per root Hertz, K is a proportionality constant and f is frequency in Hertz, and e_w is the white noise specifiction for the op amp. K can be calculated from the values on the equivalent input noise vs frequency graph to be about 450 nV. By integrating the noise density over the frequency range of operation, the input noise can be computed by:

$$e_{oa} = \sqrt{ENB\ e_w^2 + K^2\ In\ \frac{f_H}{f_L}}$$

Where e_{oa} is the equivalent input noise voltage over the bandwidth in RMS volts

K is a proportionality constant, equal to 450 nV RMS

f_L is the lower cutoff frequency, 0.724 Hz and f_H is the upper cutoff frequency, 724 Hz.

The result is e_{oa} = 1.32 μV RMS.

6.2.3 Total Noise Analysis

Adding the system noise sources results in:

$$\textit{Total System Noise (in RMS Volts)} = 14.3 \times \sqrt{803\left[(0.27E\text{-}6)^2 + (128E\text{-}9)^2 + (10.7E\text{-}9)^2 + (40E\text{-}9)^2\right] + (1.32E\text{-}6)^2}$$

$$= 0.124\ mV\ RMS$$

To convert to pk-pk voltage, multiply by 6. Then Total System Noise pk-pk = 0.743 mV pk-pk.

The TLV1544 ADC has 10 bits of resolution. With a voltage reference = 2.4 V, 0.743 mV pk-pk of noise is equivalent to about 1/3 LSB.

7 System Test and Evaluation

With the shock sensor and signal conditioning circuit interfaced to the TLV1544 EVM and the TMS320C5X EVM, proceed with testing and evaluation of the system performance.

The program listings as published in the *Interfacing the TLV1544 Analog-to-Digital Converter to the TMS320C50 DSP Applications Report* literature #SLAA025 can be used as a starting point. Appendix A contains program listings based on these, except they modify program flow so that the TLV1544 performs consecutive conversions of three of its four channels. When motion in any axis is detected, the program stores 3000h samples in memory from each channel.

Once the data is stored, it can be saved as a COFF file using the “store data” utility provided with the TMS320C5X C-Source Debugger. To read the data into an Excel spread sheet, a utility program is required to convert the hexadecimal numbers into decimal ASCII format. Source code listings are included in Appendix A.

With the data converted and loaded into Excel, it can be converted into equivalent acceleration in g using the formula:

$$g = \left(\frac{ADC\ Sample \times ADC\ Ref\ Volts}{2^{10}} - Sig\ Ref\ Volts \right) \times \left(\frac{1}{Sensor\ output \times Gain} \right)$$

Figure 7–1 shows the x-axis acceleration graphed in Excel from data collected when testing the signal conditioning and shock sensor circuits. It can be seen that the results match the results as presented previously.

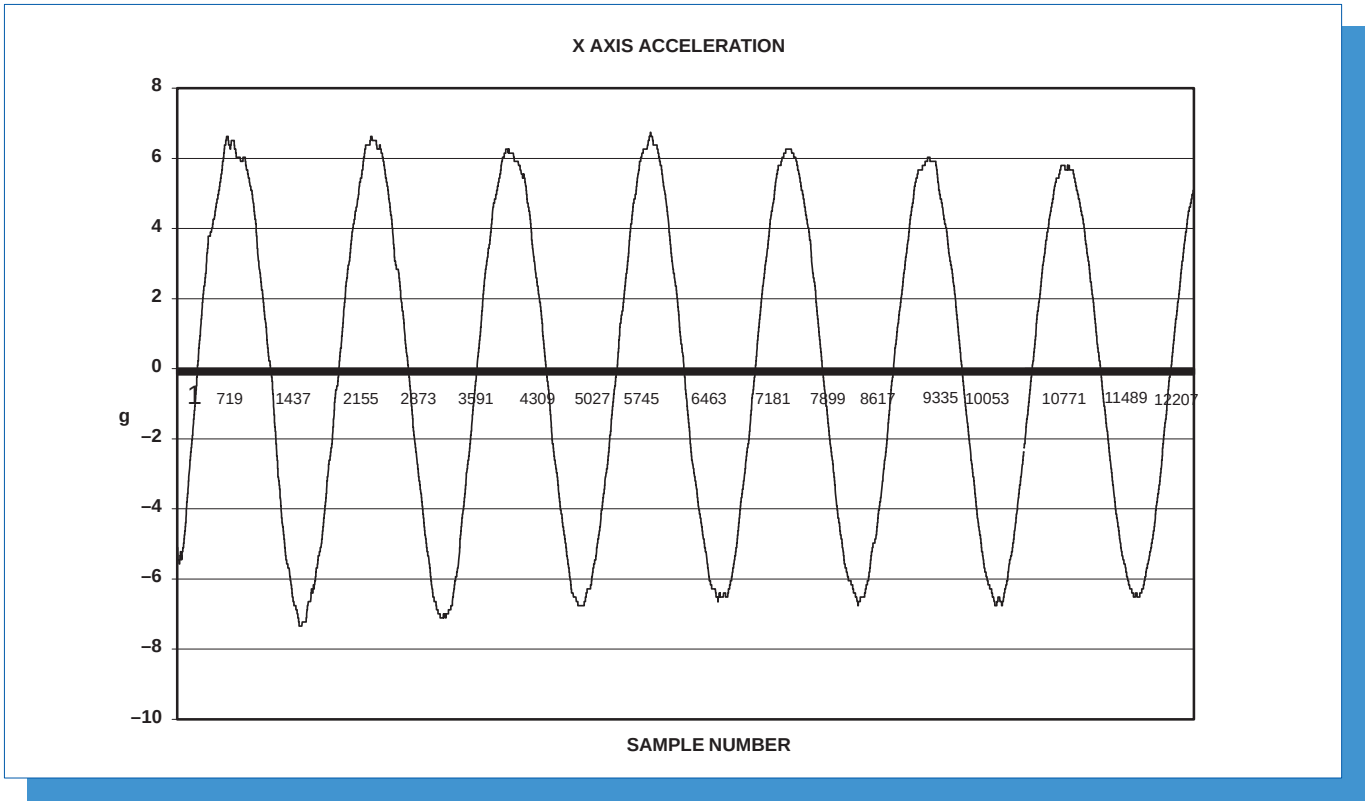

Figure 7–1. X-Axis Acceleration Graphed in Excel

8 Calibration Data/Analysis

8.1 Calibration

The acceleration sensor and the signal conditioning circuitry were housed in a Bud CU-234 box, which was machined to mount on a fixture plate for calibrated testing. The calibrated acceleration source used was an Unholtz Dickie T-1000 vibration tester. The specifications for this machine are:

- 1 inch maximum displacement
- 70 in/s maximum velocity
- 75g maximum sine acceleration ±10% operational tolerance
- 5 Hz to 5 kHz frequency range ±1/4 Hz frequency operational tolerance
- Cross axis output 25% maximum

Signals from the signal conditioning circuitry were routed to the TLV1544 ADC via coaxial cables. Sine wave acceleration was used and data was collected at various acceleration levels and frequencies on all 3 axis. Data for each axis was plotted over frequency and acceleration using the mean sensitivity. The data was analyzed and the errors calculated. Tables 8–1 through 8–6 and Figures 8–1 through 8–6 show the result. Figure 8–7 shows the output vs. input for each axis when averaged over frequency.

A maximum cross axis sensitivity of 17.6% from y-axis to x-axis was observed. Resonance was observed in the z-axis at 2 kHz.

Table 8–1. X-Axis Output Vs. Input Using X-Axis Mean Sensitivity = 1.16 mV/g

g/f	10 Hz	20 Hz	50 Hz	100 Hz	500 Hz
1	1.20	1.20	1.20	1.05	0.83
10		11.50	11.00	10.56	8.70
20			21.86	20.94	17.45
50			53.80	51.15	42.60

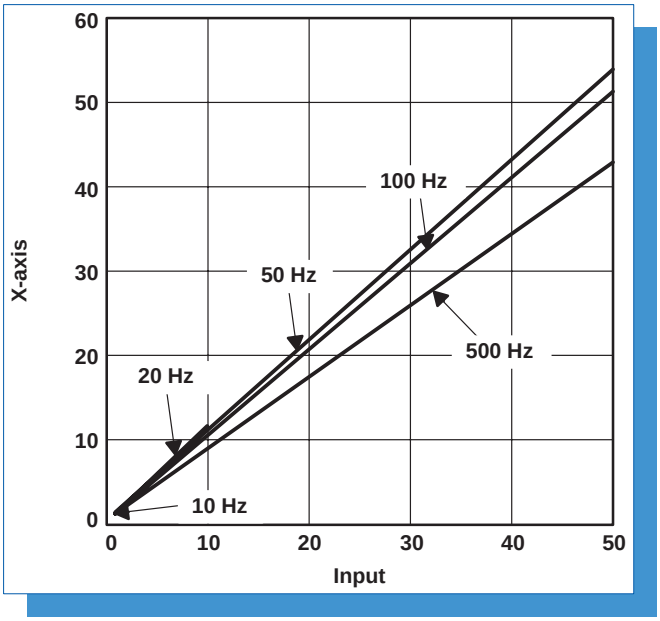

Figure 8–1. X-Axis Output vs. Input

Table 8–2. X-Axis % Error

g/f	10 Hz	20 Hz	50 Hz	100 Hz	500 Hz
1	20.00%	20.00%	20.00%	5.00%	–17.50%
10		15.00%	10.00%	5.60%	–13.00%
20			9.30%	4.70%	–12.75%
50			7.60%	2.30%	–14.80%

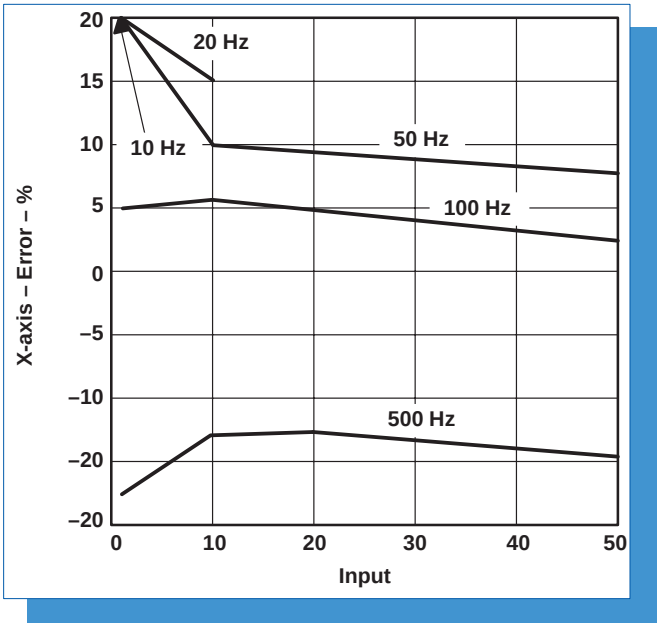

Figure 8–2. X-Axis % Error

Table 8–3. Y-Axis Output Vs. Input Using Y-Axis Mean Sensitivity = 1.35 mV/g

g/f	10 Hz	20 Hz	50 Hz	100 Hz	500 Hz
1	1.03	1.03	1.11	1.03	0.88
10		10.50	9.80	9.70	9.85
20			19.65	19.18	20.10
50			49.65	47.00	52.55

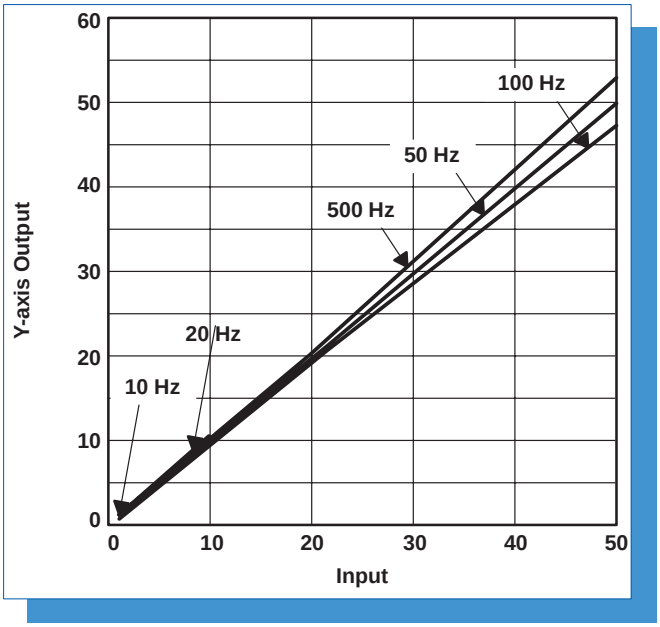

Figure 8–3. Y-Axis Output vs. Input

Table 8–4. Y-Axis % Error

g/f	10 Hz	20 Hz	50 Hz	100 Hz	500 Hz
1	2.50%	2.50%	10.50%	–2.50%	–12.50%
10		5.00%	–2.00%	–3.00%	–1.50%
20			–1.75%	–4.13%	0.50%
50			–0.70%	–6.00%	5.10%

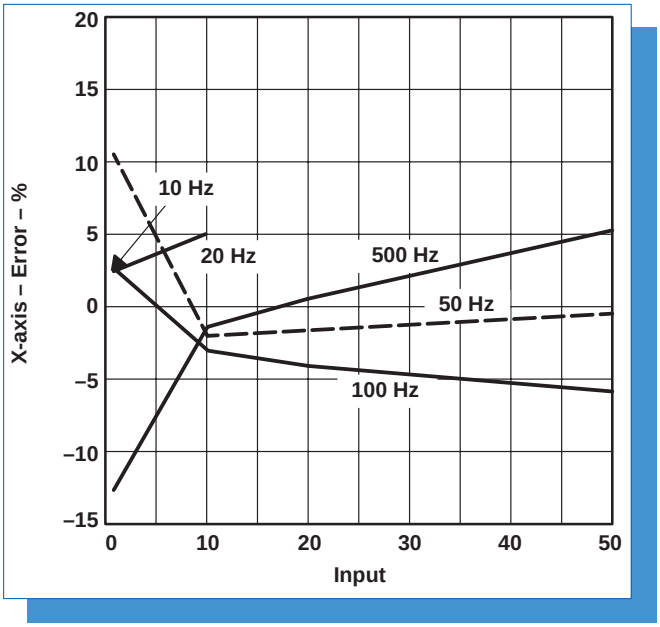

Figure 8–4. Y-Axis % Error

Table 8–5. Z-Axis Output Vs. Input Using Z-Axis Mean Sensitivity = 1.01 mV/g

g/f	10 Hz	20 Hz	50 Hz	100 Hz	500 Hz
1	0.89	0.97	0.97	0.97	1.13
10		8.20	8.35	8.20	11.70
20			16.80	16.55	23.40
50			41.30	41.55	59.10

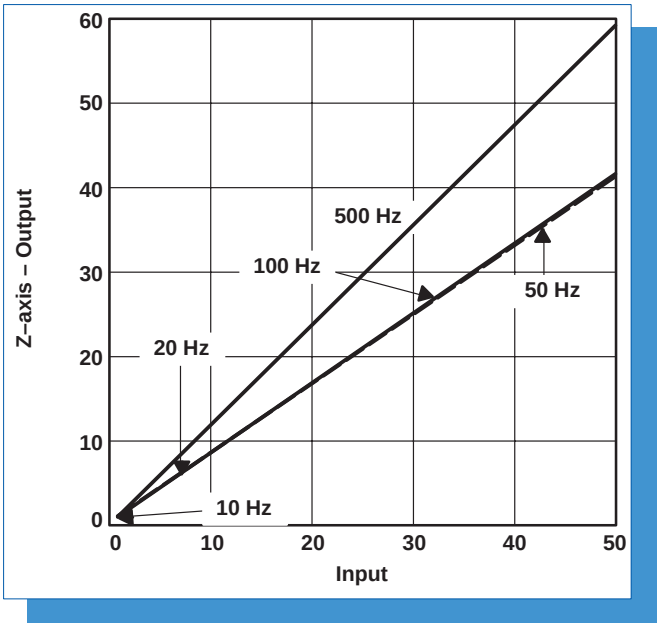

Figure 8–5. Z-Axis Output vs. Input

Table 8–6. Z-Axis % Error

g/f	10 Hz	20 Hz	50 Hz	100 Hz	500 Hz
1	–11.00%	–3.00%	–3.00%	–3.00%	13.00%
10		–18.00%	–16.50%	–18.00%	17.00%
20			–16.00%	–17.25%	17.00%
50			–17.40%	–16.90%	18.20%

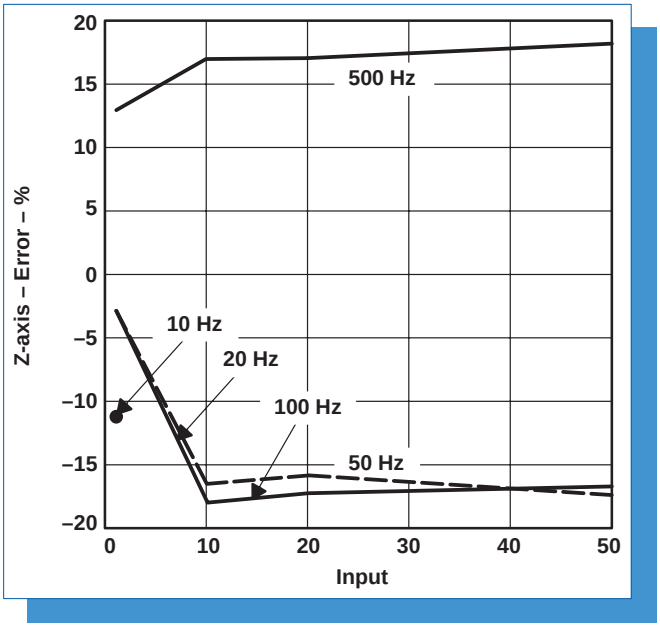

Figure 8–6. Z-Axis % Error

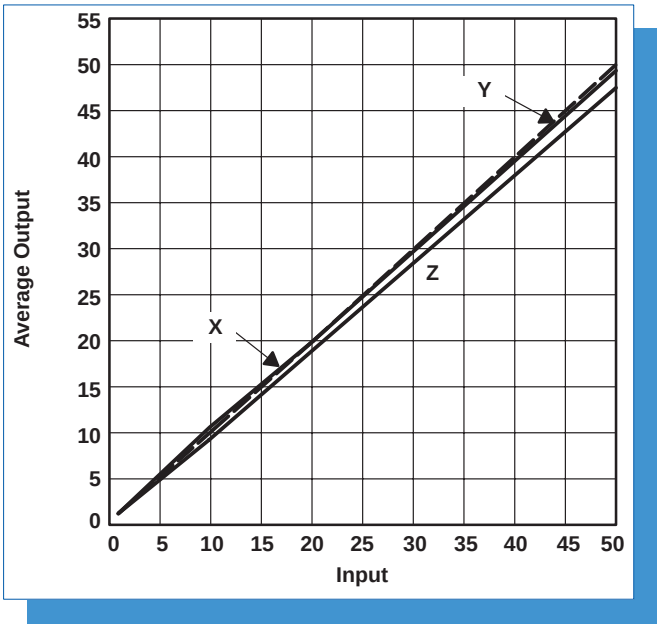

Figure 8–7. Average Output vs. Input Over Frequency for Each Axis

9 References

1. *ACH04-08-05 Data Sheet* http://www.amp.com/sensors
2. *TLV2772 Data Sheet*, literature #SLOS209
3. *Universal Operational Amplifier EVM User's Manual*, literature #SLVU006
4. *TLV1544 Data Sheet*, literature #SLAS139B
5. *TLV1544 EVM User's Manual*, literature #SLAU014
6. *Interfacing the TLV1544 Analog-to-Digital Converter to the TMS320C50 DSP Applications Report*, literature #SLAA025
7. *TMS320C5X Evaluation Module Technical Reference*, literature #SPRU087
8. *TMS320C5X C Source Debugger User's Guide*, literature #SPRU055B
9. *TMS320C5X EVM Technical Reference*, literature #SPRU087

Appendix A. Source Code Listings

```
*******************************************************************************************
* TITLE         : ADC Interface routine                                                   *
* FILE          : ACCELER1.ASM                                                            *
* FUNCTION      : MAIN                                                                    *
* PROTOTYPE     : void MAIN ()                                                            *
* CALLS         : Wait ()                                                                 *
*                 Channels ()                                                             *
* PRECONDITION  : N/A                                                                     *
* POSTCONDITION : N/A                                                                     *
* DESCRIPTION   :
* AUTHOR        : AAP Application Group, Dallas                                           *
*                 CREATED 1998(C) BY TEXAS INSTRUMENTS INCORPORATED.                      *
* REFERENCE     : TMS320C5x User's Guide, TI 1997                                         *
*               : Data Aquisation Circuits, TI 1998                                       *
*******************************************************************************************
    .mmregs
    .sect ".vectors"
    .copy "vectors.asm"

    .sect ".data"
    .copy "values.asm"
    .def    _MAIN
    .def    ACT_CHANNEL
    .def    ADWORD
    .def    ADCOUNT
    .def    ADMEM
    .def    isr_save
    .def    FINITO
    .def    data_loc_point0
    .def    data_loc_point1
    .def    data_loc_point2
    .def    data_loc_point3
    .def    MEMCOUNT
    .def    TST_CH1;
    .def    TST_CH2;
    .def    TST_CH3;
AD_DP       .usect ".variabl", 0        ;
ACT_CHANNEL .usect ".variabl", 1        ; jump address to init. new channel
ADWORD      .usect ".variabl", 1        ; send-bytes to the ADC
ADCOUNT     .usect ".variabl", 1        ; counter for one channel
ADMEM       .usect ".variabl", 1        ; points to act. memory save location
```

```
isr_save    .usect ".variabl", 1              ; memory location to save AR7 during interrupts
FINITO      .usect ".variabl", 1              ; shows, that the sampling is completed
MEMCOUNT    .usect ".variabl", 1              ; counter for samples per channel
TST_CH1     .usect ".variabl", 1              ;
TST_CH2     .usect ".variabl", 1              ;
TST_CH3     .usect ".variabl", 1              ;
data_loc_point0 .usect ".variabl", 1          ;
data_loc_point1 .usect ".variabl", 1          ;
data_loc_point2 .usect ".variabl", 1          ;
data_loc_point3 .usect ".variabl", 1          ;
    .sect ".text"
_MAIN:
*******************************************************************************************
* INITIALIZATION BODY
*******************************************************************************************
* DSP INITIALIZATION
    SETC    INTM                        ; DISABLE GLOBAL INTERRUPTS
    LDP     #0;
    OPL     #0038h, PMST                ; Configure PMST (allocate IRQ)
    APL     #0000h, PDWSR               ; clear PDWSR (zero wait states)
    APL     #00F0h, CWSR                ; clear CWSR  (zero wait states)

* SERIAL PORT INITIALIZATION
SPI_INI:
    APL     #07F38h,SPC                ; clear Res, DLB, FO, XRST, RRST  and FREE in SPC
    OPL     #00038h,SPC                ; Set Burst Mode,
                                       ; CLKX=1/4CLKOUT1,
                                       ; FSX generated by DSP
    OPL     #080C0h,SPC                ; activate transmitter and receiver
* INITIALIZE USER INFORMATIONS FOR DATA AQUISITION
    CALL    CHANNELS                   ; load informations about ADC channel and
                                       ; memory location for data saving
* Enable Interrupts
    LDP     #0
    OPL     #00014h,IMR                ; Unmask RINT and INT3
    CLRC    INTM                       ; enable global interrups
* FIRST SEND OPERATION MODE TO TLV1544
    LDP     #ADWORD
    LACL    ADWORD
    CLRC    XF                        ; Enable Chip Select
    SAMM    DXR                       ; move ADWORD into DXR (DXR = fast conversation mode)
*******************************************************************************************
* initialize MEMCOUNT and data_loc_pointx
*******************************************************************************************
```

```
    SPLK    #data_loc_0,data_loc_point0  ;
    SPLK    #data_loc_1,data_loc_point1  ;
    SPLK    #data_loc_2,data_loc_point2  ;
    SPLK    #data_loc_3,data_loc_point3  ;
    SPLK    #TRIG_REF,TST_CH1  ;
    SPLK    #TRIG_REF,TST_CH2  ;
    SPLK    #TRIG_REF,TST_CH3  ;
    SPLK    #numb_data_all,MEMCOUNT  ;
*********************************************************************************************
* MAIN BODY FOR A PROGRAM, UNRELATED TO AD-CONVERTING                                       *
* PROCESSOR SLEEP AT NONACTIVITY TIMES                                                      *
* INSTEAD OF THE IDLE INSTRUCTION, USER CAN RUN THEIR OWN PROGRAM                           *
*********************************************************************************************
USERINTERFACE:
    SPLK    #00000h,FINITO      ; sampling not finished yet                                   *

*********************************************************************************************
* testing the input samples against trigger values to start                                 *
* the final sampling of the accelerator                                                     *
*********************************************************************************************
TRIP_LOOP:
    LDP     #TST_CH1            ;
    LACC    TST_CH1             ;
    SUB     #TRIP_HIGH          ;
    BCND    TRIG,GEQ            ;
    LACC    TST_CH1             ;
    SUB     #TRIP_LOW           ;
    BCND    TRIG,LEQ            ;
    LDP     #TST_CH2            ;
    LACC    TST_CH2             ;
    SUB     #TRIP_HIGH          ;
    BCND    TRIG,GEQ            ;
    LACC    TST_CH2             ;
    SUB     #TRIP_LOW           ;
    BCND    TRIG,LEQ            ;
    LDP     #TST_CH3            ;
    LACC    TST_CH3             ;
    SUB     #TRIP_HIGH          ;
    BCND    TRIG,GEQ            ;
    LACC    TST_CH3             ;
    SUB     #TRIP_LOW           ;
    BCND    TRIG,LEQ            ;
    B       TRIP_LOOP           ;
*********************************************************************************************
```

```
* start the final sampling of the accelerator
*********************************************************************************************
TRIG:
    SPLK    MAIN_CHANNEL0,ACT_CHANNEL  ; set the next INTERRUPT-BRANCH-ADDRESS to the
                                       ; MAIN sampling
M1:
    LDP     #FINITO;
    LACC    FINITO ;
    SUB     #00001h                   ;
    BCND    M1, LT
*********************************************************************************************
* sampling is done, wait until user interrupt
*********************************************************************************************
M2:
    NOP                             ; no operation
    B       M2;
*********************************************************************************************
* END_CONV_IRQ:
*   Interrupthandler for the external INT3-IRQ, which is the end of conversation signal
*   of the ADC. The Routine will enable the Chip Signal again and send the next sample
*   instruction, contained in ADWORD, to the ADC.
*********************************************************************************************
END_CONV_IRQ:
* MAIN IRQ ROUTINE
    LDP     #ADWORD
    LACL    ADWORD
    CLRC    XF                      ; Enable Chip Select
    SAMM    DXR                     ; move ADWORD into DXR
    RETE                            ; Return to Wait loop
*********************************************************************************************
* RECEIVE_IRQ:
*   Interrupthandler for serial receive of the SPI interface. The Routine will disable the
*   Chip Signal and store the received sample into the by ADMEM specified memdory location.
*   It also checks, how many samples were made of the actual channel and if all done, it
*   will call the procedure CHANNELS to load the next channel specifications.
*********************************************************************************************
RECEIVE_IRQ:
* save AR7 before handling the IRQ routine
    ldp     #isr_save
    sar     ar7,isr_save          ; save AR7 in isr_save

* MAIN IRQ ROUTINE (Data collection from ADC)
    CLRC    SXM                   ; Clear sign bit
    LDP     #0;
```

```
    LACC    DRR,10                ; load content of DRR in ACCh; shift 10
    LDP     #ADMEM;
    MAR     *,AR7;
    LAR     AR7, ADMEM            ; AR7 = ADMEM
    SACH    *+                    ; save sample in the memory
    SAR     AR7, ADMEM            ; ADMEM++
    LAR     AR7, ADCOUNT;         ; AR7 = ADCOUNT
    LACL    ADCOUNT;
    SUB     #1                    ; increment ACC
    SACL    ADCOUNT               ; ADCOUNT=ADCOUNT–1
    BCND    KEEP_ON,NEQ           ; if ADCOUNT not 0 jump to KEEP_ON
    LACC    ACT_CHANNEL           ; load addr. to set new sampling channel
    CALA                          ; jump into the subroutine
KEEP_ON:
    SETC    XF                    ; Disable Chip Select
* restore AR7 before jump back into the user software
    ldp     #isr_save
    lar     ar7,isr_save          ; load old AR0
    RETE                          ; Return from Interrupt

   .copy ”channels.asm”
   .end
*********************************************************************************************
* TITLE         : TLV1544C ADC Interface routine                                            *
* FILE          : Channels.ASM                                                              *
* FUNCTION      : CHANNELS                                                                  *
* PROTOTYPE     : void CHANNELS ()                                                          *
* CALLS         : N/A                                                                       *
* PRECONDITION  : N/A                                                                       *
* POSTCONDITION : N/A                                                                       *
* DESCRIPTION   :
* AUTHOR        : AAP Application Group, Dallas                                             *
*                 CREATED 1998(C) BY TEXAS INSTRUMENTS INCORPORATED.                        *
* REFERENCE     : TMS320C5x User’s Guide, TI 1997                                           *
*               : Data Aquisation Circuits, TI 1998                                         *
*********************************************************************************************
    .text
CHANNELS:
WAKE_UP:
    LDP     #AD_DP                  ; point to AD Data Page
    SPLK    #fast_conv,ADWORD       ; The first Word to send is fast_conv to wake up the ADC
                                    ; from power down
    SPLK    #tresh,ADMEM            ; DATA LOCATION = THRES, because first received byte
                                    ; is meaningless
```

```
    SPLK    #1,ADCOUNT                  ; NUMBER OF DATA 0
    SPLK    #one_more,ACT_CHANNEL       ; set new jump level
    RET
one_more:
    LDP     #AD_DP                      ; point to AD Data Page
    SPLK    #channel_1,ADWORD           ; The first Word to send is fast_conv to
                                        ; wake up the ADC from power down
    SPLK    #tresh,ADMEM                ; DATA LOCATION = THRES, because first
                                        ; received byte is meaningless
    SPLK    #4,ADCOUNT                  ; NUMBER OF DATA 0
    SPLK    #CHANNEL1,ACT_CHANNEL       ; set new jump level
    RET
CHANNEL1:
    LDP     #AD_DP                      ; point to AD Data Page
    SPLK    #channel_1,ADWORD           ; SELECT CHANNEL 0
    SPLK    #TST_CH3,ADMEM              ; DATA LOCATION 0
    SPLK    #00001h,ADCOUNT             ; NUMBER OF DATA 0
    SPLK    #CHANNEL2,ACT_CHANNEL       ; set new jump level\
    RET

CHANNEL2:
    LDP     #AD_DP                      ; point to AD Data Page
    SPLK    #channel_2,ADWORD           ; SELECT CHANNEL 0
    SPLK    #TST_CH1,ADMEM              ; DATA LOCATION 0
    SPLK    #00001h,ADCOUNT             ; NUMBER OF DATA 0
    SPLK    #CHANNEL3,ACT_CHANNEL       ; set new jump level
    RET
CHANNEL3:
    LDP     #AD_DP                      ; point to AD Data Page
    SPLK    #channel_3,ADWORD           ; SELECT CHANNEL 0
    SPLK    #TST_CH2,ADMEM              ; DATA LOCATION 0
    SPLK    #00001h,ADCOUNT             ; NUMBER OF DATA 0
    SPLK    #CHANNEL1,ACT_CHANNEL       ; set new jump level
   RET
MAIN_CHANNEL0:
    LDP     #AD_DP                      ; point to AD Data Page
    SPLK    #channel_0,ADWORD           ; SELECT CHANNEL
    LACL    data_loc_point3             ;
    SACL    ADMEM                       ; load actual memory pointer
    ADD     #1                          ; increment ACC
   SACL    data_loc_point3             ; save new memory pointer
    SPLK    #00001h,ADCOUNT             ; NUMBER OF DATA
    SPLK    #MAIN_CHANNEL1,ACT_CHANNEL  ; set new jump level
    RET
```

```
MAIN_CHANNEL1:
    LDP     #AD_DP                      ; point to AD Data Page
    SPLK    #channel_1,ADWORD           ; SELECT CHANNEL
    LACL    data_loc_point0             ;
    SACL    ADMEM                       ; load actual memory pointer
    ADD     #1                          ; increment ACC
   SACL    data_loc_point0             ; save new memory pointer
    SPLK    #00001h,ADCOUNT             ; NUMBER OF DATA
    SPLK    #MAIN_CHANNEL2,ACT_CHANNEL  ; set new jump level\
    RET

MAIN_CHANNEL2:
    LDP     #AD_DP                      ; point to AD Data Page
    SPLK    #channel_2,ADWORD           ; SELECT CHANNEL
    LACL    data_loc_point1             ;
    SACL    ADMEM                       ; load actual memory pointer
    ADD     #1                          ; increment ACC
   SACL    data_loc_point1             ; save new memory pointer
    SPLK    #00001h,ADCOUNT             ; NUMBER OF DATA
    SPLK    #MAIN_CHANNEL3,ACT_CHANNEL  ; set new jump level
    RET
MAIN_CHANNEL3:
    LDP     #AD_DP                      ; point to AD Data Page
    SPLK    #channel_3,ADWORD           ; SELECT CHANNEL
    LACL    data_loc_point2             ;
    SACL    ADMEM                       ; load actual memory pointer
    ADD     #1                          ; increment ACC
   SACL    data_loc_point2             ; save new memory pointer
    SPLK    #00001h,ADCOUNT             ; NUMBER OF DATA
    LACL    MEMCOUNT                    ;
    SUB     #1                          ; increment ACC
    SACL    MEMCOUNT                    ; ADCOUNT=ADCOUNT–1
    BCND    FINITISIMO,EQ               ; if ADCOUNT not 0 jump to KEEP_ON
    SPLK    #MAIN_CHANNEL0,ACT_CHANNEL  ; set new jump level
   RET
FINITISIMO:
    SPLK    #00001h,FINITO              ; set finish sign
    SPLK    #CHANNEL_END,ACT_CHANNEL    ; set new jump level
   RET
* TLV1544 POWER DOWN AFTER TRANSITIONS
CHANNEL_END:
    LDP     #AD_DP                      ; point to AD Data Page
    SPLK    #power_down,ADWORD          ; SELECT CHANNEL 0
    SPLK    #0,ADCOUNT                  ; NUMBER OF DATA 0
```

```
    SPLK    #CHANNEL_END,ACT_CHANNEL ; set new jump level
    RET
*************************************************************************************
* TITLE         : TLV1544C ADC Interface routine                                    *
* FILE          : values.ASM                                                        *
* FUNCTION      : N/A                                                               *
* PROTOTYPE     : N/A                                                               *
* CALLS         : N/A                                                               *
* PRECONDITION  : N/A                                                               *
* POSTCONDITION : N/A                                                               *
* DESCRIPTION   : contains the control data for the TLV 1544, the starting address to *
*                 save the samples to the DSP memory and the number of samples      *
*                 according each channel                                             *
* AUTHOR        : AAP Application Group, Dallas                                     *
*                 CREATED 1998(C) BY TEXAS INSTRUMENTS INCORPORATED.                *
* REFERENCE     : TMS320C5x User's Guide, TI 1997                                   *
*               : Data Aquisation Circuits, TI 1998                                 *
*************************************************************************************
channel_0       .set    0000h                   ; Select TLV1544 channel 0
channel_1       .set    2000h                   ; Select TLV1544 channel 1
channel_2       .set    4000h                   ; Select TLV1544 channel 2
channel_3       .set    6000h                   ; Select TLV1544 channel 3
power_down      .set    8000h                   ; Software Power Down
fast_conv       .set    9000h                   ; Fast Conversion Rate
slow_conv       .set    0A00h                   ; Slow Conversion Rate
test_200        .set    0B00h                   ; (Vreg+ – Vreg–)/2
test_000        .set    0C00h                   ; Vreg–
tets_3FF        .set    0D000h                  ; Vreg+
num_data_0      .set    1FFh                    ; Number of Data from channel 0
num_data_1      .set    200h                    ; Number of Data from channel 1
num_data_2      .set    200h                    ; Number of Data from channel 2
num_data_3      .set    200h                    ; Number of Data from channel 3
tresh           .set    1000h                   ; address to waste the first input sample
                                                ; after initialization
data_loc_0      .set    01000h                  ; Start data location for channel 0
data_loc_1      .set    04000h                  ; Start data location for channel 1
data_loc_2      .set    07000h                  ; Start data location for channel 2
data_loc_3      .set    0A000h                  ; Start data location for channel 3
numb_data_all   .set    03000h                  ; how many samples per channel to do
TRIP_HIGH       .set    023bh                   ; high trigger value = 1.34 V
TRIP_LOW        .set    01e7h                   ; low trigger value = 1.14 V
TRIG_REF        .set    0211h                   ; mean value of no triggering = 1.24 V
*************************************************************************************
* TITLE         : TLV1544C ADC Interface routine                        *
```

```
* FILE          : vectors.ASM                                                    *
* FUNCTION      : N/A                                                            *
* PROTOTYPE     : N/A                                                            *
* CALLS         : N/A                                                            *
* PRECONDITION  : N/A                                                            *
* POSTCONDITION : N/A                                                            *
* DESCRIPTION   : defines the interrupt vector table.                            *
*                 if RINT occurs: vector points to RECEIVE_IRQ subroutine        *
*                 if INT3 occurs: vector points to END_CONV_IRQ subrout.         *
* AUTHOR        : AAP Application Group, Dallas                                  *
*                 CREATED 1998(C) BY TEXAS INSTRUMENTS INCORPORATED.             *
* REFERENCE     : TMS320C5x User's Guide, TI 1997                                *
*********************************************************************************
   .mmregs

RS      b   _MAIN             ;0x00; RESET
INT1    b   INT1              ;0x02; external user interrupt #1
INT2    b   INT2              ;0x04; external user interrupt #2
INT3    b   END_CONV_IRQ      ;0x06; external user interrupt #3
TINT    b   TINT              ;0x08; internal timer interrupt
RINT    b   RECEIVE_IRQ       ;0x0A; Serial Port receive interrupt
XINT    b   XINT              ;0x0C; Serial Port transmit interrupt
TRNT    b   TRNT              ;0x0E; TDM receive interrupt
TXNT    b   TXNT              ;0x10; TDM transmit interrupt
INT4    b   INT4              ;0x12; external user interrupt #4
   .space  14*16             ;0x14–0x21; reserved area
TRAP    b   TRAP              ;0x22; trap instruction vector
NMI     b   NMI               ;0x24; non–maskable interrupt
/*********************************************************************************/
/* TITLE         : Hexadecimal to ASCII – Decimal Converter                      */
/* FILE          : B_H_C502.PAS                                                  */
/* FUNCTION      : N/A                                                           */
/* PROTOTYPE     : N/A                                                           */
/* CALLS         : N/A                                                           */
/* PRECONDITION  : N/A                                                           */
/* POSTCONDITION : N/A                                                           */
/* DESCRIPTION   : strip header from COFF file and convert                       */
/*                              hexadecimal numbers into ASCII–decimal           */
/* AUTHOR        : AAP Application Group, Dallas                                 */
/*                 CREATED 1998(C) BY TEXAS INSTRUMENTS INCORPORATED.            */
/* REFERENCE     :                                                               */
/* NOTE       : Written in PASCAL                                                */
/*********************************************************************************/
uses crt,dos,printer;
```

```
type
   positionstyp=record
   AAAA :word;
   END;
var
   pos          :positionstyp;
   satz1  :file of positionstyp;
   inpfile,ffile        :string[20];
   na,c         :char;
   s2           :string[2];
   i            :integer;
   A,n          :char;
   AA           :byte;
   tin          :file of byte;
   tout         :text;
   offset       :byte;
   nb           : byte;
   number       :longint;
   number_low   :longint;
   number_high  :longint;
begin
repeat
  clrscr;
  writeln;
  writeln(' TEXAS INSTRUMENTS (C) SOFTWARE');
  writeln(' 1998 ');
  writeln;
  write(' hard drive for input binary hexadecimal file(16bit) : ');  na:=readkey;
writeln(na,':);
  write(' file name: ');
  readln(ffile);ffile:=na+':+ffile;
  assign(tin,ffile);{$i-} reset(tin);{$i+}
  i:=IoResult;
  if (i<>0) then
    begin
     repeat
       writeln(' file not found, new Input Name? [Y/N]');
       n:=upcase(readkey);
     until ((n='Y') or (n='N'));
    end;
  if i=0 then n:='w';
until ((n='N') or (n='w'));
if n<>'N' then
begin
```

```
inpfile:=ffile;
repeat
  writeln;
  write(' hard drive for decimal output file : ');  na:=readkey; writeln(na,':);
  write(' file name: ');  readln(ffile);ffile:=na+':+ffile;
  assign(tout,ffile);{$i-} reset(tout);{$i+}
  i:=IoResult;
  if (i=0) then
    begin
     repeat
       writeln(' file already found, should I overwrite it? [Y/N]');
       n:=upcase(readkey);
     until ((n='Y') or (n='N'));
    end
  else n:='Y';
until n='Y';
    rewrite(tout);writeln;
    writeln(tout,'binary Hex_dec-convertion, made by  TI: ');
    writeln(tout,'Input_file: ',inpfile);
    read(tin,nb);
    reset(tin);
   for i:=1 to 70 do read(tin,nb);
   repeat
     read(tin,nb);
     number_low:=nb;
     read(tin,nb);
     number_high:=nb;
     number:=number_high*256+number_low;
     writeln(number);
     writeln(tout,number,';');
   until eof(tin);
   close(tin);
   close(tout);
   readln;
end;
end.
{A=97 0=48}
```