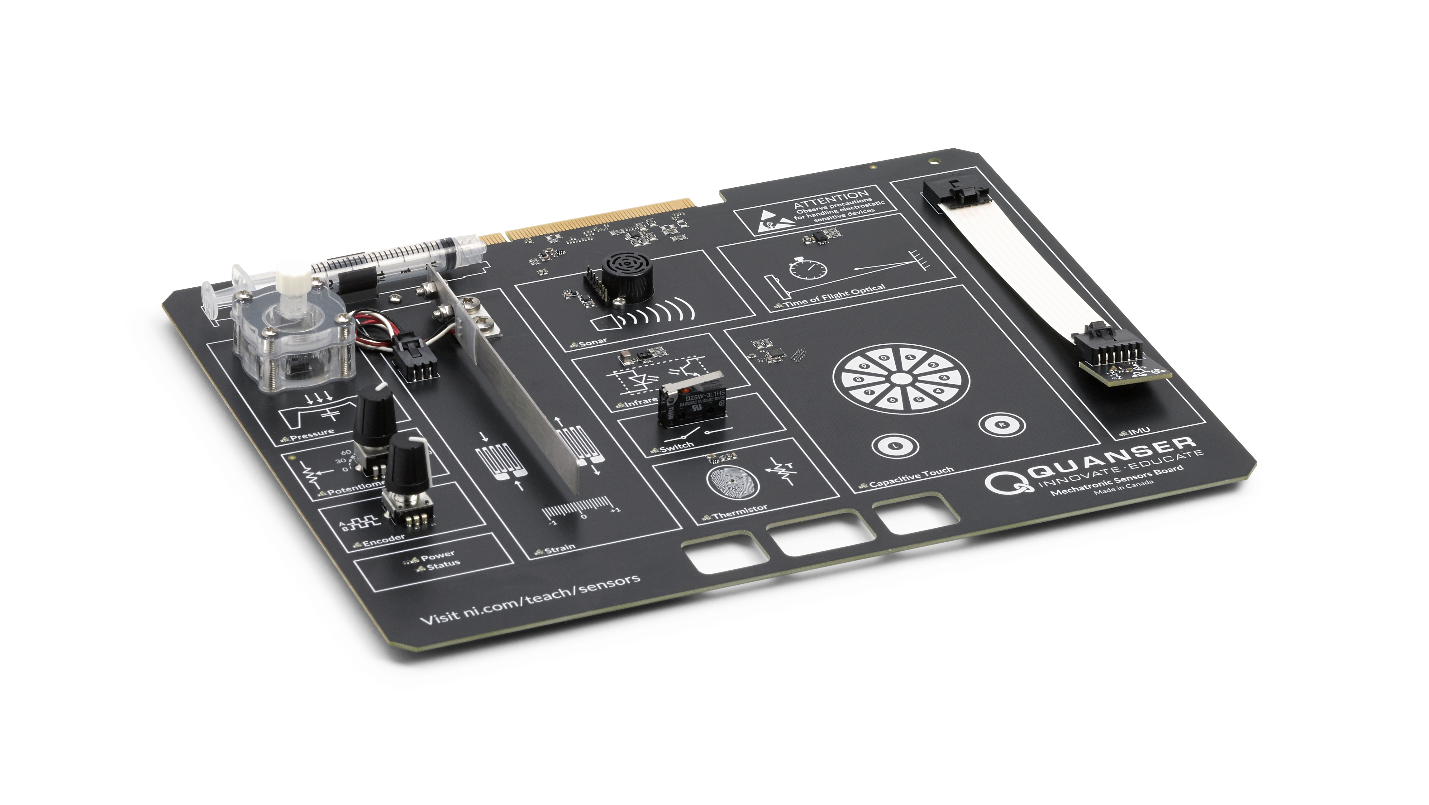


Lab Manual: Fundamentals of Mechatronic Sensors

Using the Quanser Mechatronic Sensors Board for NI ELVIS III



© 2018 Quanser Inc., All Rights Reserved

Printed in Markham, Ontario.

This document and the software described in it are provided subject to a license agreement. Neither the software nor this document may be used or copied except as specified under the terms of that license agreement. Quanser Inc. grants the following rights: a) The right to reproduce the work, to incorporate the work into one or more collections, and to reproduce the work as incorporated in the collections, b) to create and reproduce adaptations provided reasonable steps are taken to clearly identify the changes that were made to the original work, c) to distribute and publically perform the work including as incorporated in collections, and d) to distribute and publicly perform adaptations. The above rights may be exercised in all media and formats whether now known or hereafter devised. These rights are granted subject to and limited by the following restrictions: a) You may not exercise any of the rights granted to You in above in any manner that is primarily intended for or directed toward commercial advantage or private monetary compensation, and b) You must keep intact all copyright notices for the Work and provide the name Quanser Inc. for attribution. These restrictions may not be waved without express prior written permission of Quanser Inc.

LabVIEW and National Instruments are trademarks of National Instruments.

All other trademarks or product names are the property of their respective owners.

**Additional Disclaimers:** The reader assumes all risk of use of this resource and of all information, theories, and programs contained or described in it. This resource may contain technical inaccuracies, typographical errors, other errors and omissions, and out-of-date information. Neither the author nor the publisher assumes any responsibility or liability for any errors or omissions of any kind, to update any information, or for any infringement of any patent or other intellectual property right.

Neither the author nor the publisher makes any warranties of any kind, including without limitation any warranty as to the sufficiency of the resource or of any information, theories, or programs contained or described in it, and any warranty that use of any information, theories, or programs contained or described in the resource will not infringe any patent or other intellectual property right. THIS RESOURCE IS PROVIDED “AS IS.” ALL WARRANTIES, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, ANY AND ALL IMPLIEDWARRANTIES OFMERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT OF INTELLECTUAL PROPERTY RIGHTS, ARE DISCLAIMED.

No right or license is granted by publisher or author under any patent or other intellectual property right, expressly, or by implication or estoppel.

IN NO EVENT SHALL THE PUBLISHER OR THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL, COVER, ECONOMIC, OR CONSEQUENTIAL DAMAGES ARISING OUT OF THIS RESOURCE OR ANY INFORMATION, THEORIES, OR PROGRAMS CONTAINED OR DESCRIBED IN IT, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES, AND EVEN IF CAUSED OR CONTRIBUTED TO BY THE NEGLIGENCE OF THE PUBLISHER, THE AUTHOR, OR OTHERS. Applicable law may not allow the exclusion or limitation of incidental or consequential damages, so the above limitation or exclusion may not apply to you.

Table of Contents

[Introduction 7](#_Toc516839756)

[Learning Objectives 7](#_Toc516839757)

[Prerequisites 7](#_Toc516839758)

[Completed Courses 8](#_Toc516839759)

[Hardware, Software, and Tool Knowledge 8](#_Toc516839760)

[Organization of the Lab Manual 8](#_Toc516839761)

[Lab 1: Angular Displacement 8](#_Toc516839762)

[Lab 2: Distance and Proximity 8](#_Toc516839763)

[Lab 3: Temperature 8](#_Toc516839764)

[Lab 4: Strain 9](#_Toc516839765)

[Lab 5: Pressure 9](#_Toc516839766)

[Lab 6: Contact 9](#_Toc516839767)

[Lab 7: Inertial Measurement 9](#_Toc516839768)

[Lab Tools and Technology 10](#_Toc516839769)

[Platform: NI ELVIS III 10](#_Toc516839770)

[Hardware: Quanser Mechatronic Sensors Board 11](#_Toc516839771)

[Software: LabVIEW 12](#_Toc516839772)

[Lab 1: Angular Displacement 13](#_Toc516839773)

[Learning Objectives 13](#_Toc516839774)

[Required Tools and Technology 14](#_Toc516839775)

[Expected Deliverables 15](#_Toc516839776)

[Section 1: Measuring Angular Displacement using a Potentiometer 16](#_Toc516839777)

[1.1 Theory and Background 16](#_Toc516839778)

[1.2 Implement 19](#_Toc516839779)

[1.3 Analyze 21](#_Toc516839780)

[Section 2: Measuring Angular Displacement using an Encoder 23](#_Toc516839781)

[2.1 Theory and Background 23](#_Toc516839782)

[2.2 Implement 29](#_Toc516839783)

[2.3 Analyze 31](#_Toc516839784)

[Section 3: Design Considerations 33](#_Toc516839785)

[Lab 2: Distance and Proximity 35](#_Toc516839786)

[Learning Objectives 35](#_Toc516839787)

[Required Tools and Technology 36](#_Toc516839788)

[Expected Deliverables 37](#_Toc516839789)

[Section 1: Measuring Long-range Distance using a Sonar 38](#_Toc516839790)

[1.1 Theory and Background 38](#_Toc516839791)

[1.2 Implement 41](#_Toc516839792)

[1.3 Analyze 44](#_Toc516839793)

[Section 2: Measuring Short to Mid-range Distance using a Time-of-Flight Sensor 45](#_Toc516839794)

[2.1 Theory and Background 45](#_Toc516839795)

[2.2 Implement 47](#_Toc516839796)

[2.3 Analyze 49](#_Toc516839797)

[Section 3: Measuring Proximity using an Infrared Proximity Sensor 51](#_Toc516839798)

[3.1 Theory and Background 51](#_Toc516839799)

[3.2 Implement 53](#_Toc516839800)

[3.3 Analyze 55](#_Toc516839801)

[Section 4: Design Considerations 56](#_Toc516839802)

[Lab 3: Temperature 57](#_Toc516839803)

[Learning Objectives 57](#_Toc516839804)

[Required Tools and Technology 58](#_Toc516839805)

[Expected Deliverables 59](#_Toc516839806)

[Section 1: Measuring Temperature using a Thermistor 60](#_Toc516839807)

[1.1 Theory and Background 60](#_Toc516839808)

[1.2 Implement 65](#_Toc516839809)

[1.3 Analyze 67](#_Toc516839810)

[Section 2: Design Considerations 68](#_Toc516839811)

[Lab 4: Strain Gage 71](#_Toc516839812)

[Learning Objectives 71](#_Toc516839813)

[Required Tools and Technology 73](#_Toc516839814)

[Expected Deliverables 74](#_Toc516839815)

[Section 1: Measuring Strain 75](#_Toc516839816)

[1.1 Theory and Background 75](#_Toc516839817)

[1.2 Implement 84](#_Toc516839818)

[1.3 Analyze 87](#_Toc516839819)

[Section 2: Design Considerations 88](#_Toc516839820)

[Lab 5: Pressure 91](#_Toc516839821)

[Learning Objectives 91](#_Toc516839822)

[Required Tools and Technology 92](#_Toc516839823)

[Expected Deliverables 93](#_Toc516839824)

[Section 1: Pressure Measurement and Calibration 94](#_Toc516839825)

[1.1 Theory and Background 94](#_Toc516839826)

[1.2 Implement 100](#_Toc516839827)

[1.3 Analyze 104](#_Toc516839828)

[Section 2: Design Considerations 106](#_Toc516839829)

[Lab 6: Contact 108](#_Toc516839830)

[Learning Objectives 108](#_Toc516839831)

[Required Tools and Technology 109](#_Toc516839832)

[Expected Deliverables 110](#_Toc516839833)

[Section 1: Switch Debouncing 111](#_Toc516839834)

[1.1 Theory and Background 111](#_Toc516839835)

[1.2 Implement 116](#_Toc516839836)

[1.3 Analyze 118](#_Toc516839837)

[Section 2: Capacitive Touch Sensor 119](#_Toc516839838)

[2.1 Theory and Background 119](#_Toc516839839)

[2.2 Implement 123](#_Toc516839840)

[2.3 Analyze 124](#_Toc516839841)

[Section 3: Design Considerations 125](#_Toc516839842)

[Lab 7: Inertial Measurement 126](#_Toc516839843)

[Learning Objectives 126](#_Toc516839844)

[Required Tools and Technology 128](#_Toc516839845)

[Expected Deliverables 129](#_Toc516839846)

[Section 1: Measuring Pose and Magnetic Field using an IMU 130](#_Toc516839847)

[1.1 Theory and Background 130](#_Toc516839848)

[1.2 Implement 136](#_Toc516839849)

[1.3 Analyze 139](#_Toc516839850)

[Section 2: Design Considerations 140](#_Toc516839851)

# Thí nghiệm 1: Góc chuyển vị



Hình 0: Tính toán chính xác góc chuyển vị là việc cần thiết trong nhiều ứng dụng liên quan tới máy móc

Trong bài này, ta sẽ tìm hiểu về phép đo góc chuyển vị bằng cách sử dụng chiết áp và bộ mã hóa tương đối. Hai bộ cảm biến được hiệu chuẩn trước khi thực hiện đo góc chuyển vị. Khảo sát các thuật toán giải mã cho bộ mã hóa.

## Mục tiêu nghiên cứu

Sau khi hoàn thành thí nghiệm, sinh viên có thể :

1. Nắm được đặc tính phân áp của chiết áp
2. Hiệu chỉnh đầu ra của chiết áp dựa vào góc chuyển vị
3. Xác định độ phân giải của một bộ mã hóa
4. Khảo sát giải mã không vuông pha, X2 và X4
5. Hiệu chỉnh đầu ra của bộ mã hóa dựa vào góc chuyển vị
6. Quan sát ảnh hưởng của tốc độ lấy mẫu thấp tới sự chính xác của dữ liệu đo được

## Yêu cầu công cụ và công nghệ

|  |  |
| --- | --- |
| Nền tảng: NI ELVIS III | * Xem hướng dẫn sử dụng   http://www.ni.com/en-us/support/model.ni-elvis-iii.html |
| Phần cứng: Mạch cảm biến cơ điện tử Quanser | * Xem hướng dẫn sử dụng   http://www.ni.com/en-us/support/model.quanser-mechatronic-sensors-board-for-ni-elvis-iii.html |
| Phần mềm: LabVIEW Phiên bản 18.0 hoặc mới hơn  Công cụ và Modules:   * Module thời gian thực LabVIEW * Công cụ NI ELVIS III | * Liên hệ với giảng viên hay trưởng phòng thí nghiệm về thông tin bản quyền và cơ sở hạ tầng phần mềm trước khi tải và cài đặt * Tải & cài đặt NI ELVIS III * <http://www.ni.com/academic/download> * Xem hướng dẫn tại * http://www.ni.com/academic/students/learn-labview/ |

## Kết quả thu được

Trong bài thí nghiệm, ta sẽ thu được các kết quả sau:

* Ghi lại đầu ra chiết áp dạng thô
* Màn hình đồ thị hiệu chuẩn chiết áp biểu diễn đường hiệu cong hiệu chỉnh phù hợp
* Ghi lại hệ số hiệu chuẩn của chiết áp
* Tính toán độ nhạy của chiết áp theo đơn vị mV/độ
* Quan sát và đếm số sườn tín hiệu mã hóa sử dụng giải mã không vuông pha, X2 và X4
* Tính số xung/vòng của bộ mã hóa
* Tính độ phân giải góc của bộ mã hóa
* Màn hình phản hồi tín hiệu của bộ mã hóa

Người hướng dẫn có thể yêu cầu hoàn thành báo cáo thí nghiệm. Liên hệ với người hướng dẫn để có mẫu báo cáo hay yêu cầu cụ thể.

## Phần 1: Đo góc chuyển vị bằng cách sử dụng chiết áp

### 1.1 Cơ sở lý thuyết

#### Chiết áp là gì ?

Chiết áp xoay, viết tắt là POT, là biến trở được điều khiển bằng tay. Theo hình 1-1, cấu tạo chiết áp thông thường gồm một trục ngoài, ba cực (A, W và B), một phần tử điện trở được bọc bên trong có hình tròn và một tiếp điểm trượt được gọi là chổi tiếp xúc. Bằng cách vặn trục, phần chổi nằm phía trong sẽ tiếp xúc với điện trở ở các vị trí khác nhau làm thay đổi giá trị điện trở khi đo cực giữa W với một trong hai cực bên (A hoặc B). Tổng trở của chiết áp có thể tính bằng cách kẹp đầu đo của đồng hồ vạn năng với hai cực A và B.

|  |  |
| --- | --- |
| C:\Users\amolki\Desktop\QNET Mechatronic Sensors - Potentiometer - Concept Review.pdf - Adobe Acrobat Pro.jpg | C:\Users\amolki\Desktop\QNET Mechatronic Sensors - Potentiometer - Concept Review.pdf - Adobe Acrobat Pro.jpg |

Hình 1-1: Một chiết áp vặn điển hình

Hình 1-2 giới thiệu hai loại biến trở phổ biến khác là biến trở thanh trượt và biến trở Trimmer.

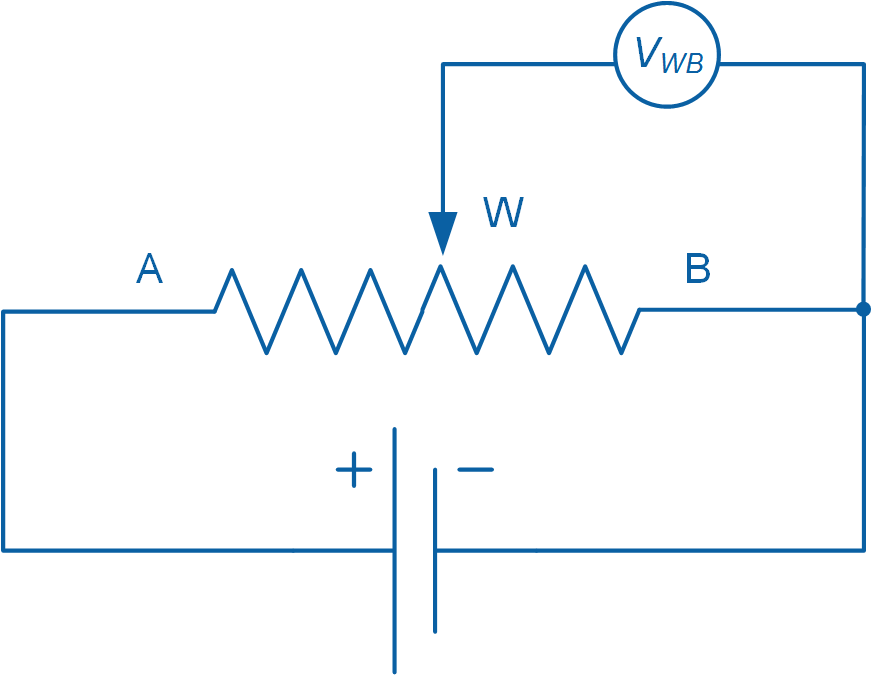
|  |  |
| --- | --- |
| C:\Users\amolki\Desktop\QNET Mechatronic Sensors - Potentiometer - Concept Review.pdf - Adobe Acrobat Pro.jpg  (a) Biến trở Trimmer | C:\Users\amolki\Desktop\QNET Mechatronic Sensors - Potentiometer - Concept Review.pdf - Adobe Acrobat Pro.jpg  (b) Biến trở thanh trượt |

Hình 1-2: Các loại biến trở phổ biến khác (nguồn: DigiKey)

Sơ đồ nguyên lý phân áp của chiết áp được thể hiện ở hình 1-3. Bằng cách gắn điện áp VAB đã biết giữa hai cực A và B, VAW và VWB có thể tính bằng công thức:

Công thức 1-1

Khi nối với một trục ngoài, chiết áp xoay có thể đo được góc chuyển vị tuyệt đối. Trong trường hợp này, bằng cách đặt một điện áp đã biết vào các cực ngoài của chiết áp, ta có thể xác định được vị trí cảm biến dựa trên điện áp ra VAW hoặc VWB mà tỉ lệ với vị trí của trục xoay. Một ưu điểm của việc sử dụng chiết áp như một cảm biến vị trí tuyệt đối là sau khi ngắt nguồn, vị trí đó vẫn giữ nguyên do bởi điện trở chiết áp là không đổi.



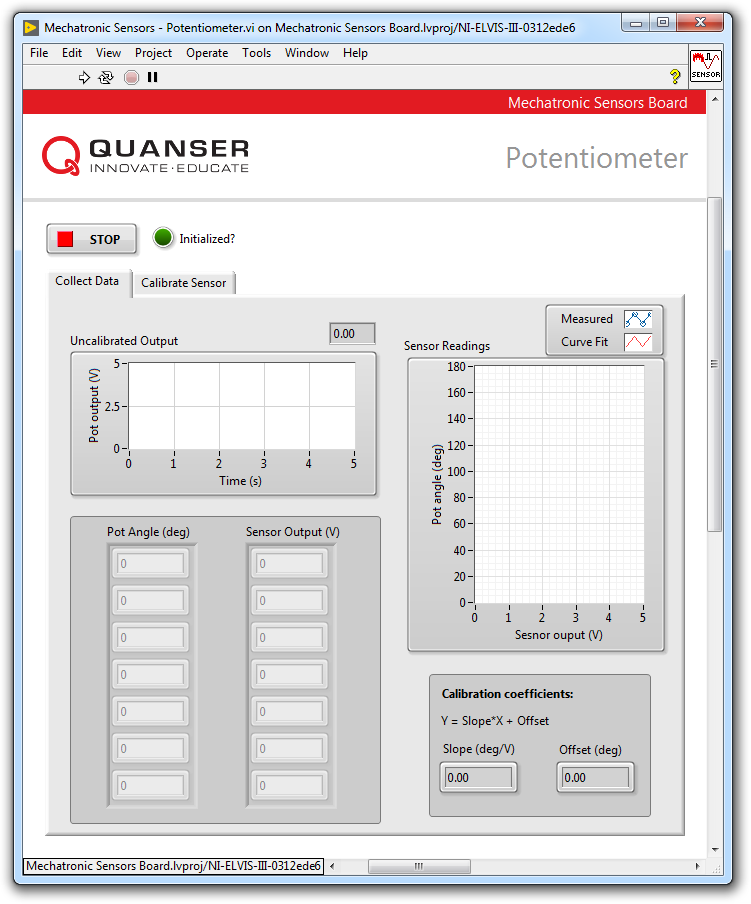
Hình 1-3: Sơ đồ nguyên lý phân áp của một chiết áp xoay

Dựa vào cấu trúc, một số chiết áp có khóa vật lý giúp tránh việc vặn trục xoay hoàn toàn một góc 360°. Để khắc phục hạn chế này, người ta sử dụng chiết áp không có khóa vật lý khi muốn tiếp tục xoay. Một hạn chế khác là sự xuất hiện dải chết, tức là điện trở không thay đổi khi ta xoay chiết áp. Đối với chiết áp xoay liên tục, dải chết xuất hiện khi chiết áp đạt giới hạn và dẫn tới kết quả bị gián đoạn khi tiến hành đo góc chuyển vị.

Tuổi thọ của chiết áp thông thường khoảng vài nghìn lần vặn, điều này là do bởi chổi tiếp xúc có tác động vật lý tới phần tử điện trở bên trong chiết áp và gây mài mòn. Tác động vật lý cùng với bụi bẩn gây ra tiếng ồn và tạp nhiễu điện. Trong khi tiếng ồn thường không nghe thấy được ở các chiết áp thế hệ mới, tạp nhiễu điện lại gây sai số trong việc đo đầu ra. Trong ứng dụng âm thanh, khi chiết áp đóng vai trò điều khiển âm lượng và tông nhạc, tạp nhiễu điện xuất hiện dưới dạng các tiếng nổ lép bép có thể nghe được

### 1.2 Tiến hành thí nghiệm

Hình 1-4 là Công cụ ảo VI dùng để lấy dữ liệu và hiệu chuẩn chiết áp.



Hình 1-4: VI dùng để lấy dữ liệu từ chiết áp

#### Lấy dữ liệu

1. Mở **Mechatronic Sensors Board.lvproj**
2. Từ cửa sổ **Project Explorer**, mở **Mechatronic Sensors - Potentiometer.vi**
3. Chọn thanh **Collect Data**.
4. Tiến hành chạy VI.
5. Đợi tới khi đèn LED chỉ thị **Initialized?** bật sáng.
6. Đặt giá trị nút bấm chiết áp về điểm 0.
7. Nhập 0 vào **Pot Angle (deg)**.
8. Sử dụng biểu đồ dạng sóng **Uncalibrated Output**, đọc đầu ra phản hồi của cảm biến và nhập giá trị vào **Sensor Output (V)**.
9. Tiếp tục đo bằng cách xoay chiết áp tới 30°. Nhập giá trị góc và đầu ra cảm biến đo được vào **Pot Angle (deg)** và **Sensor Output (V)**. Chụp lại màn hình kết quả.

Lưu ý : Khi nhập vào tất cả dữ liệu đo được, một đường cong tuyến tính được tự động khởi tạo để phù hợp với dữ liệu. Đường này thể hiện đường cong hiệu chuẩn của cảm biến và được biểu diễn trong đồ thị dạng sóng **Sensor Readings.**

1. Độ dốc và độ lệch của đường cong hiệu chuẩn được tự động tính toán với VI và biểu diễn qua **Slope (deg/V)** và **Offset (deg)**. Ghi lại các giá trị này.
2. Ghi kết quả thu được vào bảng 1-1.
3. Chụp lại màn hình đồ thị **Sensor Readings**.
4. Chuyển sang phần tiếp theo.

Bảng 1-1: Giá trị chiết áp đo được

|  |  |
| --- | --- |
| Góc (độ) | Kết quả (V) |
| 0° |  |
| 30° |  |
| 60° |  |
| 90° |  |
| 120° |  |
| 150° |  |
| 180° |  |

#### Hiệu chỉnh chiết áp

1. Chọn thanh **Calibrate Sensor** để hiệu chuẩn đầu ra của chiết áp dựa vào vị trí góc (đơn vị độ).
2. Sử dụng điều khiển số **Slope (deg/V)** và **Offset (deg)** để nhập độ dốc và độ lệch đã có trong bước lấy dữ liệu.
3. Kiểm tra tính chính xác của hiệu chỉnh. Để làm việc này, đặt nút bấm chiết áp ở các góc khác nhau và chắc chắn rằng vị trí góc được hiển thị đúng ở biểu đồ dạng sóng **Calibrated Output** và đồng hồ đo **Pot Angle (deg).**
4. Ấn nút **Stop**.

### 1.3 Phân tích

1-1 Trình bày kết quả ở bảng 1-1.

1-2 Trình bày đường cong hiệu chỉnh trong đồ thị sóng *Sensor Readings* ở bước 12.

1-3 Trình bày công thức hiệu chỉnh thu được?

1-4 Độ nhạy của cảm biến theo đơn vị mV/độ là bao nhiêu?

## Phần 2: Đo góc chuyển vị bằng cách sử dụng bộ mã hóa Encoder

### 2.1 Cơ sở lý thuyết

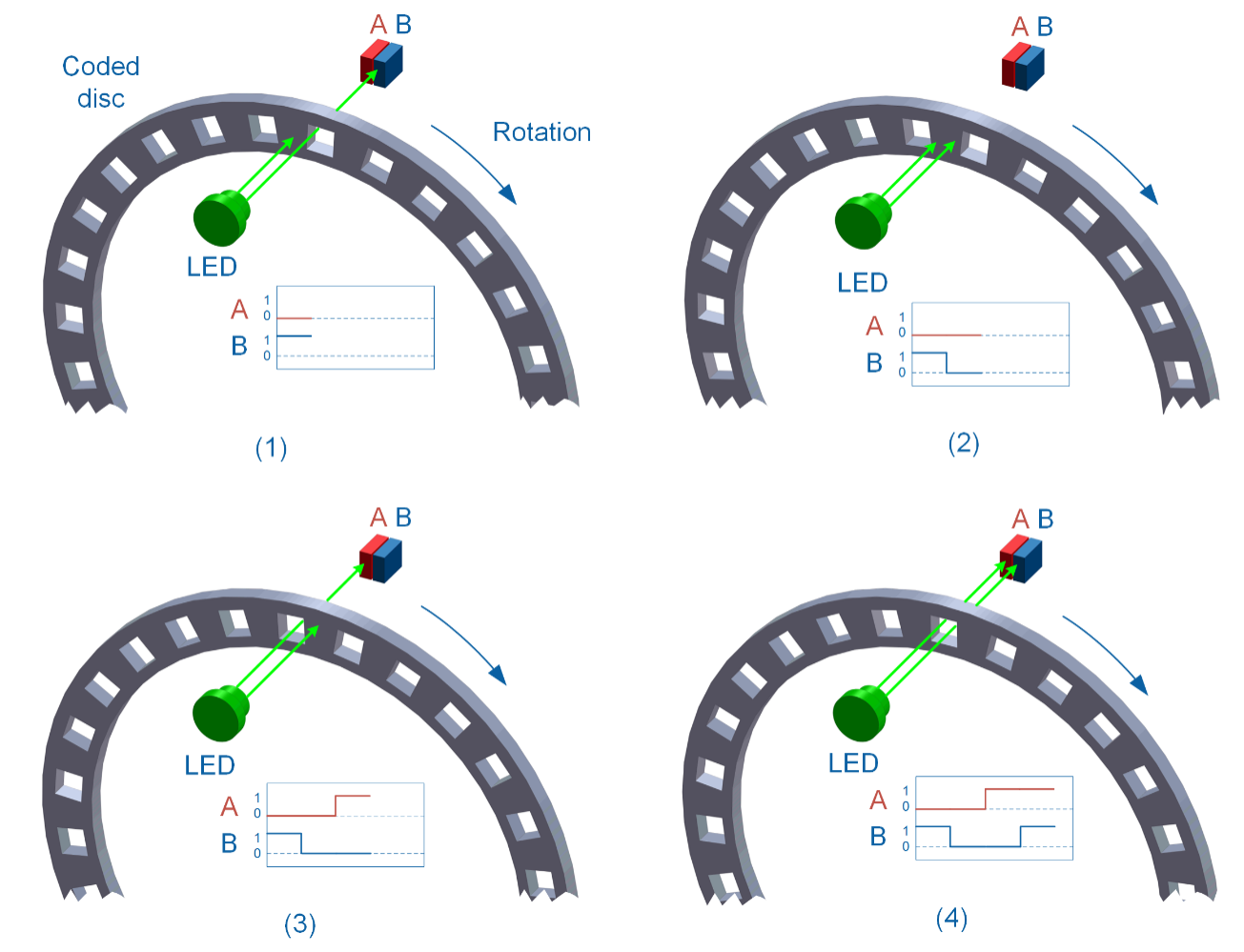
#### Bộ mã hóa là gì ?

Một bộ mã hóa tương đối quang (hình 2-1), là một cảm biến dùng để đo góc chuyển vị tương đối với một giá trị đã biết cho trước. Không giống bộ mã hóa tuyệt đối, bộ mã hóa tương đối không giữ lại thông tin vị trí của nó khi ngắt nguồn. Một bộ mã hóa tương đối cho đầu ra là chuỗi xung liên quan tới sự thay đổi tương đối vị trí góc. Bộ mã hóa thường được sử dụng để đo góc chuyển vị của trục tải xoay. Thông tin trích xuất từ bộ mã hóa tương đối cũng có thể sử dụng để suy ra tốc độ quay tức thời.



Hình 2-1: Một bộ mã hóa tương đối quang được sản xuất bởi US Digital

Một bộ mã hóa tương đối quang thông thường cấu tạo bởi một đĩa mã hóa, một đèn LED hồng ngoại (IR) và hai bộ cảm biến quang. Đĩa được mã hóa với mô hình tia sáng và tối xen kẽ đóng vai trò như một màn trập. Theo như nguyên lý ở hình 2-2, ánh sáng phát ra từ LED hồng ngoại bị ngắt bởi việc mã hóa khi mà đĩa xoay quanh trục của nó.



Hình 2-2: Đầu ra của bộ mã hóa tương đối cho hai tín hiệu A và B khi quay theo chiều kim đồng hồ

Hai cảm biến quang A và B được đặt sau đĩa mã hóa sẽ nhận biết ánh sáng hồng ngoại từ LED , điều đó dẫn tới bốn trạng thái khác nhau của tín hiệu/xung A và B dưới bảng 2-1:

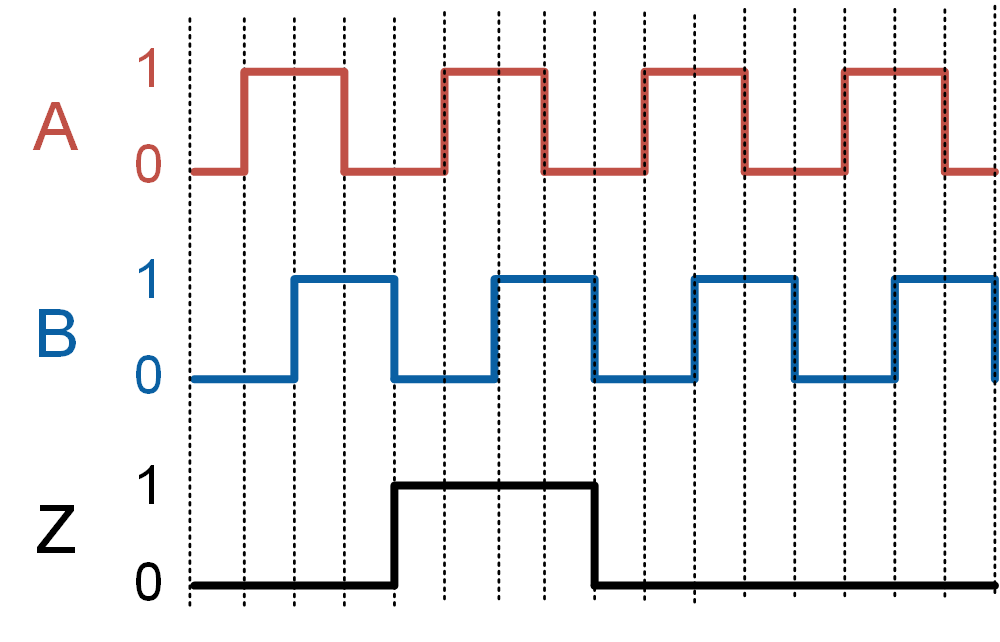
Bảng 2-1: Các trạng thái vuông pha khác nhau

|  |  |  |
| --- | --- | --- |
| Trạng thái | Tín hiệu A | Tín hiệu B |
| 1 | OFF | ON |
| 2 | OFF | OFF |
| 3 | ON | OFF |
| 4 | ON | ON |

Bộ mã hóa có đầu ra tín hiệu A và B thường được gọi là bộ mã hóa vuông pha do bởi các tín hiệu vuông pha và kết quả là có 4 trạng thái khá nhau. Bộ mã hóa không vuông pha chỉ có 1 tín hiệu ra, do đó không thể xác định hướng. Độ phân giải của bộ mã hóa được xác định qua số lượng mô hình sáng và tối trên đĩa, phương pháp đo được dựa trên số xung mỗi vòng (PPR).

Có những bộ mã hóa sử dụng một xung chỉ số (kênh Z), được kích hoạt một lần sau mỗi vòng quay đầy đủ của đĩa (xem hình 2-3). Xung chỉ số có thể được sử dụng cho hiệu chuẩn hay được gọi là hệ thống định hướng, cũng như bộ đếm vòng quay. Dựa vào bộ mã hóa, độ rộng của xung chỉ số có thể được gióng thẳng với một trong bốn trạng thái vuông pha bất kỳ. Ví dụ, xung chỉ số có thể có độ rộng kéo dài tới một chu kỳ (4 trạng thái), nửa chu kỳ (2 trạng thái) hay một phần tư chu kỳ (1 trạng thái). Trong ví dụ ở hình 2-3, độ rộng xung chỉ số được gióng thẳng ứng với một chu kỳ tín hiệu B.

Có hai phương pháp mà bộ mã hóa ghi lại xung chỉ số : (a) sử dụng trạng thái đã xác định trước của tín hiệu A và B, hay (b) sử dụng trạng thái tín hiệu A và B do người dùng xác định, trong trường hợp này người dùng phải chọn một trạng thái tổ hợp A và B mà chỉ xảy ra một lần duy nhất trong bề rộng của xung chỉ số.



Hình 2-3: Đầu ra của bộ mã hóa vuông pha với xung chỉ số

#### Giải mã bộ mã hóa

Để tiến hành đo bộ mã hóa, ta cần nối đầu ra bộ mã hóa với một bộ đếm. Sau đó, sử dụng một thuật toán giải mã để xác định số lần đếm và có thể là cả hướng quay.

Bốn thuật toán giải mã phổ biến được sử dụng: Không vuông pha, X1, X2, and X4.

##### Không vuông pha

Khi sử dụng bộ giải mã không vuông pha, chỉ có sườn lên của tín hiệu A được đếm khi trục quay. Bộ đếm tăng khi có sườn lên của tín hiệu A. Vì tín hiệu B không được sử dụng, bộ mã hóa không thể nhận diện được hướng quay. Ví dụ, sử dụng bộ giải mã không vuông pha, bộ mã hóa 9 PPR sẽ đếm tổng cộng 9 lần cho mỗi vòng trục mã hóa quay. Số lần đếm tiếp tục tăng cho dù bất kể trục quay về hướng nào.

##### Bộ giải mã X1

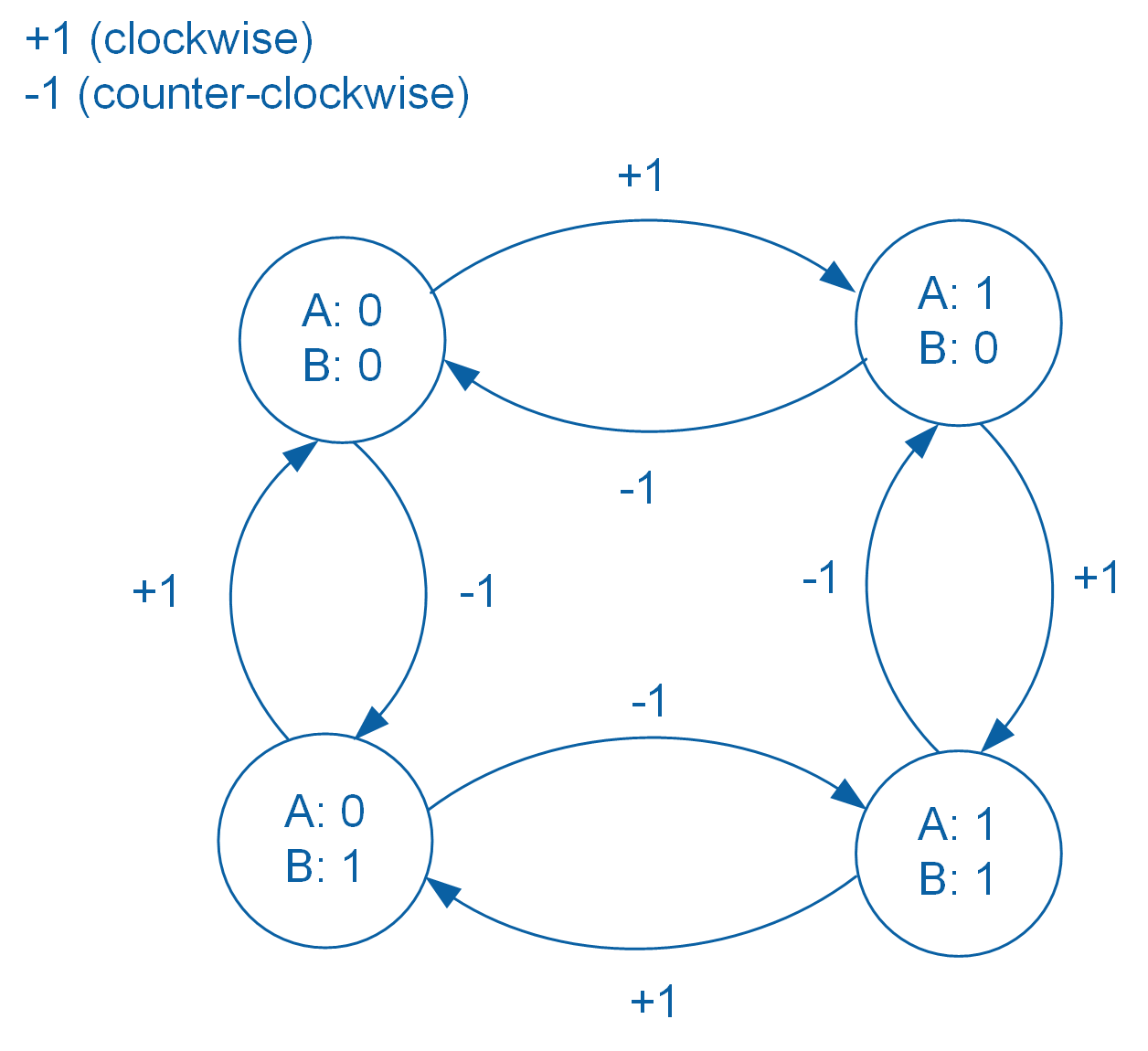
Khi sử dụng bộ giải mã X1, chỉ sườn lên của tín hiệu A được đếm khi trục quay. Khi có tín hiệu sườn lên A, thuật toán sẽ quan sát trạng thái hiện tại của tín hiệu B. Nếu tín hiệu B là thấp, bộ đếm sẽ tăng. Ngược lại, khi tín hiệu B là cao, bộ đếm sẽ giảm. Khi sử dụng bộ giải mã X1, một bộ mã hóa 9 PPR sẽ có kết quả đếm tổng cộng 9 lần cho mỗi vòng trục mã hóa quay

##### Bộ giải mã X2

Khi sử dụng bộ giải mã X2, cả sườn lên và xuống của tín hiệu A được đếm khi trục quay. Khi có tín hiệu sườn lên A, thuật toán quan sát trạng thái hiện tại của tín hiệu B. Nếu tín hiệu B là thấp, bộ đếm sẽ tăng. Ngược lại, khi tín hiệu B là cao, bộ đếm sẽ giảm. Khi có tín hiệu sườn xuống A, nếu tín hiệu B là cao thì bộ đếm tăng và ngược lại. Khi sử dụng bộ giải mã X2, bộ mã hóa 9 PPR sẽ đếm tổng cộng 18 lần cho mỗi vòng trục mã hóa quay.

##### Bộ giải mã X4

Khi sử dụng bộ giải mã X4, cả sườn lên và xuống của hai tín hiệu A và B được đếm khi trục quay. Bằng cách sử dụng sơ đồ máy trạng thái, hình 2-4 mô tả bộ đếm thay đổi phụ thuộc vào trạng thái của A và B. Một bộ giải mã X4 đếm nhiều gấp 4 lần so với bộ giải mã X1, do đó có độ phân giải cao nhất trong ba loại giải mã. Khi sử dụng bộ giải mã X4, một bộ mã hóa 9 PPR sẽ đếm tổng cộng 36 lần cho mỗi vòng trục mã hóa quay.



Hình 2-4: Biểu diễn máy trạng thái của thuật toán giải mã X4

#### Tính góc chuyển vị

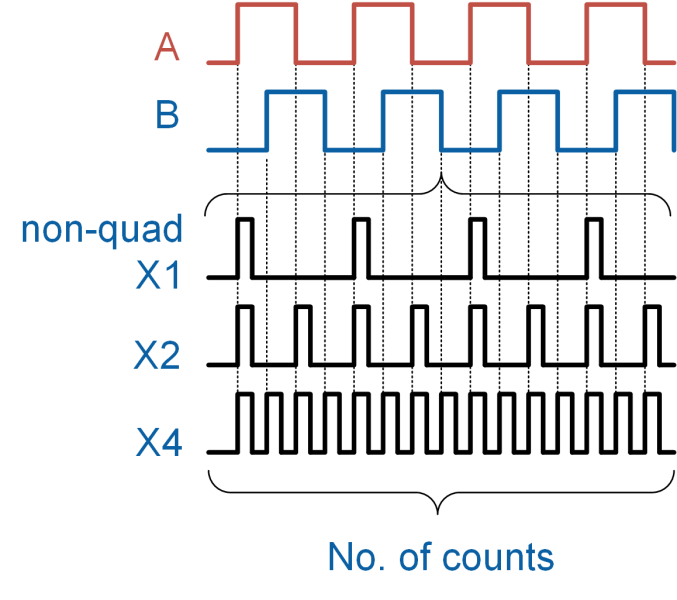
Xung tạo bởi một bộ mã hóa có thể chuyển đổi thành vị trí góc bằng công thức 2-1:

Công thức 2-1

trong đó *Counts* là số sườn đếm được; *N* = 1, 2, hoặc 4 tương ứng lần lượt với bộ giải mã không vuông pha/X1, X2, X4; PPR là giá trị PPR của bộ mã hóa. Độ phân giải góc của một bộ mã hóa (tránh nhầm lẫn với độ phân giải bộ mã hóa, hoặc PPR) phụ thuộc vào PPR của bộ mã hóa và thuật toán giải mã được sử dụng, và được tính bằng công thức 2-2 :

Công thức 2-2

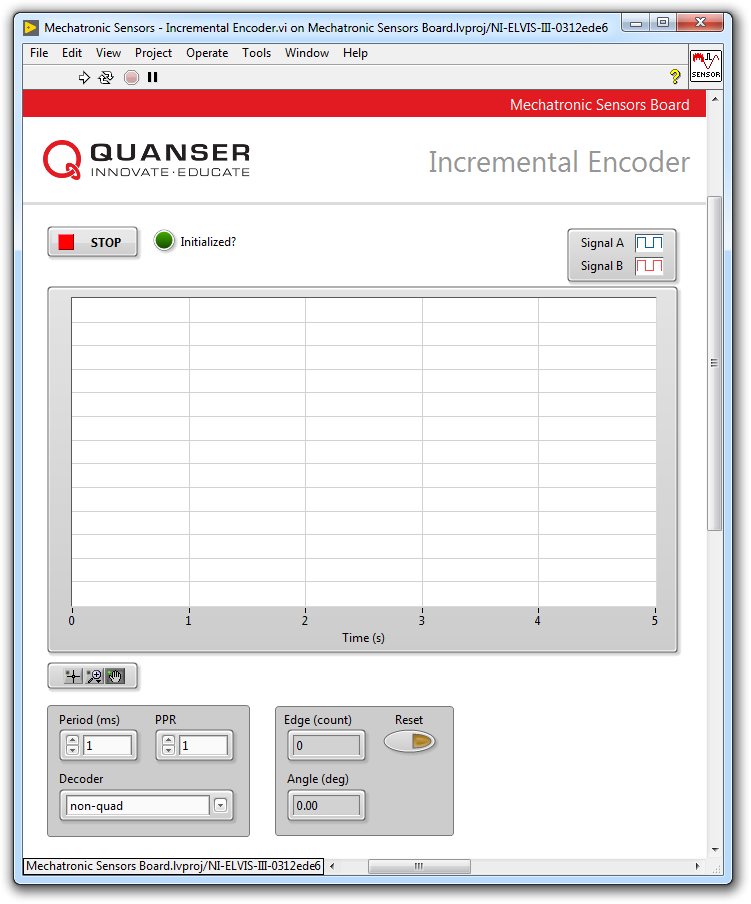
Hình 2-5 so sánh số lần đếm được khởi tạo giữa các bộ giải mã không vuông pha, X1, X2 và X4 :



Hình 2-5: So sánh số lần đếm khởi tạo bởi các thuật toán giải mã khác nhau

### 2.2 Tiến hành thí nghiệm

Hình 2-6 là Công cụ ảo VI dùng để lấy dữ liệu và hiệu chuẩn từ bộ mã hóa



Hình 2-6: VI dùng để lấy dữ liệu từ bộ mã hóa

#### Giải mã không vuông pha

1. Mở **Mechatronic Sensors Board.lvproj**
2. Từ cửa sổ **Project Explorer**, mở **Mechatronic Sensors - Incremental Encoder.vi**
3. Trong bảng chọn **Decoder** , chọn **non-quad**.
4. Tiến hành chạy VI.
5. Đợi tới khi đèn LED chỉ thị **Initialized?** được bật sáng.
6. Trong giải mã không vuông pha chỉ sử dụng tín hiệu A. Xoay núm bộ mã hóa theo chiều kim đồng hồ. Màn hiển thị đếm số **Edge (count)** thay đổi như thế nào?
7. Xoay núm ngược chiều kim đồng hồ. Màn hiển thị đếm số **Edge (count)** thay đổi như thế nào ?

*Lưu ý :* Có thể ấn nút Reset để đặt lại bộ đếm bất cứ lúc nào. Điều này sẽ đặt giá trị hiển thị **Edge (count)** và **Angle (deg)** về 0

1. Khi bộ đếm số **Edge (count)** hiển thị, xác định số xung mà bộ mã hóa khởi tạo trong mỗi vòng quay (PPR)

*Lưu ý:* PPR được xác định trong chế độ không vuông pha và liên hệ với tổng số xung tạo ra bởi *tín hiệu A* khi bộ mã hóa thực hiện hết một vòng quay. Giá trị PPR được sử dụng để hiệu chỉnh xung của bộ mã hóa dựa trên góc chuyển vị (đơn vị độ)

1. Chuyển sang phần tiếp theo.

#### Hiệu chỉnh bộ mã hóa

1. Hiệu chỉnh xung của bộ mã hóa dựa vào góc chuyển vị. Để làm điều này, nhập giá trị PPR đã tính toán ở phần trước trong bộ điều khiển số **PPR** và nhấn **Enter**.
2. Kiểm chứng độ chính xác của hiệu chuẩn. Để làm điều này, đầu tiên ấn nút **Reset** sau đó xoay núm bộ mã hóa và chắc chắn rằng vị trí góc được hiển thị chính xác ở phần hiển thị số **Angle (deg)**.
3. Chuyển sang phần tiếp theo.

#### Giải mã X2

1. Trong bảng chọn **Decoder**, chọn **X2**.
2. Nhấn nút **Reset**.
3. Trong giải mã X2, cả hai tín hiệu A và B được sử dụng. Xoay núm bộ mã hóa theo chiều kim đồng hồ. Màn hiển thị số **Edge (count)** và **Angle (deg)** thay đổi như thế nào?

*Lưu ý:* Một bộ mã hóa sẽ có giá trị PPR cố định bất kể ta sử dụng thuật toán giải mã nào.

1. Xoay núm ngược chiều kim đồng hồ. Màn hiển thị số **Edge (count)** và **Angle (deg)** thay đổi như thế nào ?
2. Quan sát tín hiệu A và B
3. Độ phân giải của góc chuyển vị đo được là bao nhiêu?
4. Chuyển sang phần tiếp theo.

#### Giải mã X4

1. Trong bảng chọn **Decoder**, chọn **X4**.
2. Nhấn nút **Reset**.
3. Xoay núm theo chiều kim đồng hồ và ngược chiều kim đồng hồ. Màn hiển thị số **Edge (count)** và **Angle (deg)** thay đổi như thế nào ?
4. Độ phân giải của góc chuyển vị đo được là bao nhiêu?
5. Quan sát tín hiệu A và B khi từ từ xoay núm bộ mã hóa theo chiều kim đồng hồ. Riêng với trường hợp này, so sánh trạng thái tín hiệu A và B với sơ đồ máy trạng thái ở hình 2-4. Chụp lại màn hình kết quả.
6. Nhấn nút **Stop**.

### 2.3 Phân tích

2-1 Màn hiển thị số **Edge (count)** thay đổi như thế nào khi xoay núm theo chiều kim đồng hồ ở bước 6 ?

2-2 Màn hiển thị số **Edge (count)** thay đổi như thế nào khi xoay núm ngược chiều kim đồng hồ ở bước 7 ? Giải thích trạng thái quan sát được.

2-3 Giá trị PPR của cảm biến đã tính ở bước 8 là bao nhiêu ?

2-4 Trong trường hợp giải mã không vuông pha, khi xoay bộ mã hóa ở bước 11, màn hiển thị số **Angle (deg)** có hiển thị chính xác vị trí góc không ?

2-5 Trong trường hợp giải mã X2, khi xoay bộ mã hóa ở bước 15, màn hiển thị **Edge (count)** và **Angle (deg)** thay đổi như thế nào ?

2-6 Khi xoay bộ mã hóa ngược chiều kim đồng hồ ở bước 16, màn hiển thị **Edge (count)** và **Angle (deg)** thay đổi như thế nào ? Giải thích trạng thái quan sát được

2-7 Độ phân giải của góc chuyển vị đã tính ở bước 18 là bao nhiêu ? Kết quả có phù hợp với giá trị tính bằng công thức 2-2 không ?

2-8 Trong trường hợp giải mã X4, khi xoay bộ mã hóa theo chiều kim đồng hồ và ngược chiều kim đồng hồ ở bước 22, màn hiển thị **Edge (count)** và **Angle (deg)** thay đổi như thế nào ?

2-9 Độ phân giải của góc chuyển vị đã tính ở bước 23 là bao nhiêu ? Kết quả có phù hợp với giá trị tính bằng công thức 2-2 không ?

2-10 So sánh độ phân giải góc của thuật giải mã X2 và X4.

2-11 Sử dụng màn hình kết quả đã chụp ở bước 24, so sánh trạng thái A và B với sơ đồ máy trạng thái ở hình 2-4. Kết quả có khớp với chuỗi máy trạng thái không ?

## Phần 3: Phương án thiết kế

3-1 Bạn được giao nhiệm vụ lựa chọn một bộ mã hóa tương đối để đo vị trí của động cơ một chiều. Độ phân giải tối thiểu là 0.01 radians. Thêm vào đó, hệ thống dữ liệu thu được (DAQ) và phần mềm chỉ có thể thực hiện giải mã X2. Giá trị PPR tối thiểu yêu cầu là bao nhiêu để đáp ứng được hạn chế này ?

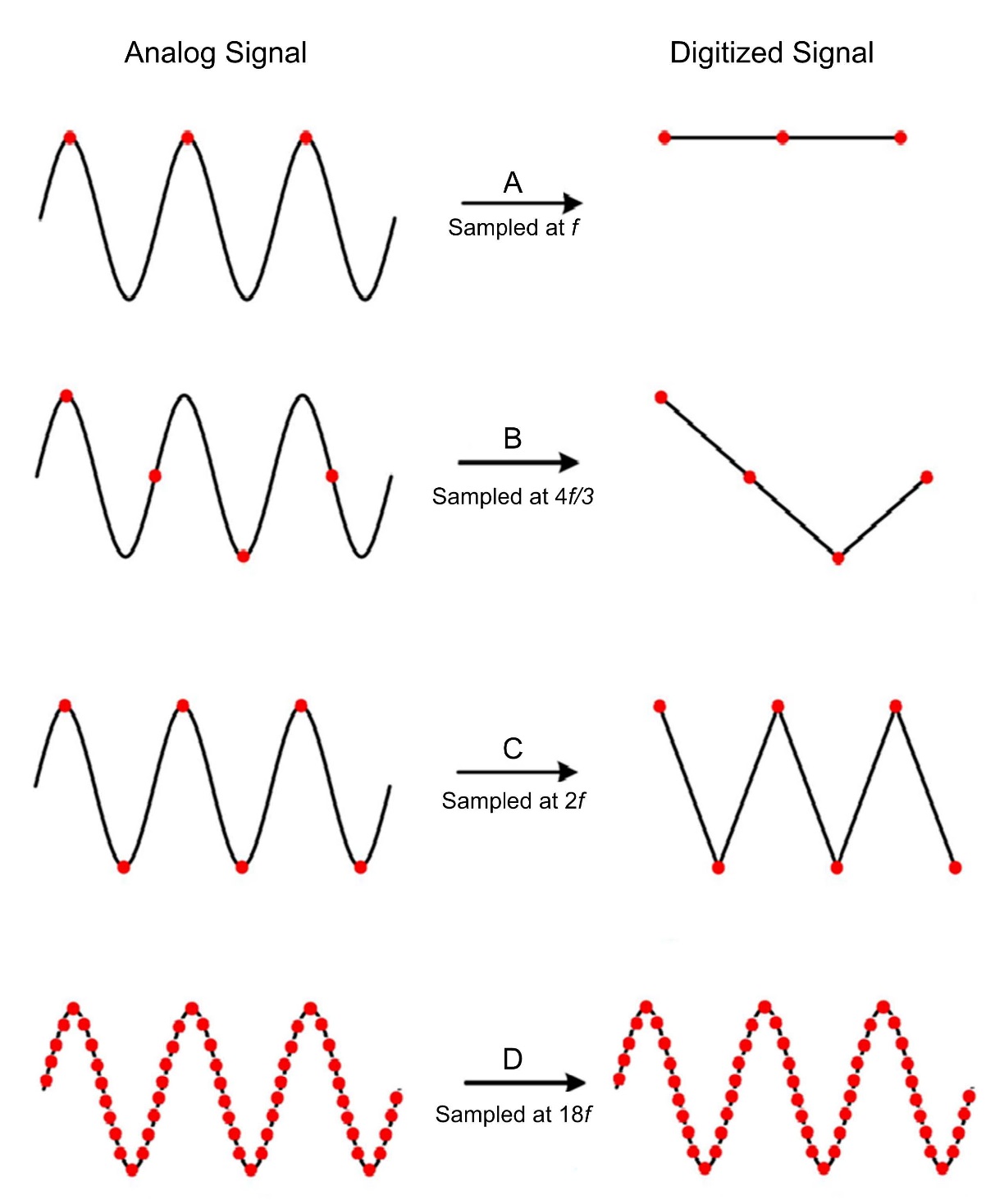
3-2 Như đã lưu ý ở phần lý thuyết, một bộ đếm được sử dụng để đếm xung khởi tạo bởi bộ mã hóa. Bộ đếm sử dụng một kích cỡ bit nhất định. Ví dụ, bộ đếm 24-bit có thể đếm được 224 = 16,777,216 lần. Nếu bộ đếm vượt quá số lần đến sẽ dẫn tới hiện tượng tràn số.

Giả sử bạn có bộ mã hóa 1000 PPR và một bộ đếm 24-bit. Ta sẽ sử dụng bộ mã hóa để theo dõi tốc độ quay của trục có tốc độ tối đa 500 vòng/phút. DAQ của bạn thực hiện giải mã X4. Tính thời gian tối thiểu để bộ đếm có thể theo dõi tốc độ trục quay trước khi xảy ra tràn số.

3-3 Yêu cầu tốc độ lấy mẫu dữ liệu thích hợp được đặt ra nhằm có phép đo chính xác. Tốc độ lấy mẫu hay tần số lấy mẫu là tốc độ mà DAQ đọc hay số hóa một tín hiệu đầu vào. Đơn vị của tốc độ lấy mẫu là Hz, hay mẫu/giây. Tốc độ lấy mẫu thường được xác định bởi người dùng, mặc dù mỗi DAQ sẽ có giới hạn trong tốc độ lấy dữ liệu. Có thể coi mẫu DAQ như các bức ảnh, giống với khung hình trong một bộ phim. DAQ phân tích tín hiệu thực càng nhanh, độ phân giải và chi tiết có thể thấy trong tín hiệu số càng rõ nét hơn.

Hình 2-7 cho thấy ảnh hưởng của tốc độ lấy mẫu khác nhau khi thu thập tín hiệu hình sin, việc lấy mẫu ở tốc độ khác nhau cho ra các dạng sóng số hóa khác nhau. Khi tốc độ lấy mẫu tăng, mẫu đầu ra càng giống chính xác với hình dạng tín hiệu ban đầu hơn.

Quan sát ảnh hưởng của tốc độ lấy mẫu thấp đối với đo góc chuyển vị sử dụng bộ mã hóa. Để làm điều này, ấn **Stop** để ngừng VI. Chu kỳ mẫu mặc định trong VI được đặt giá trị 1 ms (tương đương với tốc độ lấy mẫu 1000 Hz). Thay đổi chu kỳ mẫu tới **20 ms** (tương đương với tốc độ lấy mẫu 50 Hz). Chọn giải mã **X4** và đặt bộ điều khiển số PPR tới giá trị đã tính trước đó. Tiến hành chạy lại VI. Nhấn nút **Reset**. Giờ xoay từ từ trục bộ mã hóa 360 độ theo chiều kim đồng hồ. Màn hiển thị số **Angle (deg)** sẽ hiển thị một góc chuyển vị có giá trị xấp xỉ 360 độ. Tiếp tục Reset bộ mã hóa và xoay như bước trên và xoay thật nhanh. Lúc này giá trị hiển thị trên **Angle (deg)** là bao nhiêu? Giải thích kết quả.



Hình 2-7: Ảnh hưởng của tốc độ lấy mẫu đối với thu thập tín hiệu hình sin

# Lab 2: Distance and Proximity



Figure 0: Distance measurement is an integral part of mobile robotic applications

This lab explores long-range distance measurement using a sonar, mid- to short-range distance measurement using a Time-of-Flight (ToF) sensor, as well proximity measurement using an infrared proximity sensor. Sensor calibration, measurement scatter, as well as the effect of target reflectivity, will be examined.

## Learning Objectives

After completing this lab, you should be able to complete the following activities:

* Calibrate the output of a sonar in terms of distance
* Characterize Time-of-Flight sensor scatter
* Determine sensor resolution
* Measure target proximity using an infrared proximity sensor
* Examine the effect of target reflectivity on the sensor output

## Required Tools and Technology

|  |  |
| --- | --- |
| Platform: NI ELVIS III | * View User Manual   http://www.ni.com/en-us/support/model.ni-elvis-iii.html |
| Hardware: Quanser Mechatronic Sensors Board | * View User Manual   http://www.ni.com/en-us/support/model.quanser-mechatronic-sensors-board-for-ni-elvis-iii.html |
| Software: LabVIEW Version 18.0 or Later  Toolkits and Modules:   * LabVIEW Real-Time Module * NI ELVIS III Toolkit | * Before downloading and installing software, refer to your professor or lab manager for information on your lab’s software licenses and infrastructure * Download & Install for NI ELVIS III * <http://www.ni.com/academic/download> * View Tutorials * http://www.ni.com/academic/students/learn-labview/ |
| Accessories:  * Sturdy 12 in by 12 in cardboard (white color) * Sturdy 12 in by 12 in cardboard (black color) | |

## Expected Deliverables

In this lab, you will collect the following deliverables:

* Record calibrated sonar data
* Screenshot of sonar calibration curve
* Record sonar calibration coefficients
* Calculate sonar resolution
* Calculate sonar sensitivity
* Record ToF sensor scatter data
* Screenshot of ToF scatter behavior
* Calculate the standard deviation of ToF sensor scatter data
* Record proximity sensor threshold data
* Plot curve relating proximity sensor pulse count to a proximity threshold
* Examine the effect of target reflectivity on proximity threshold
* Experimental plan to compare sonar, ToF, and proximity sensors for distance measurement

Your instructor may expect you to complete a lab report. Refer to your instructor for specific requirements or templates.

## Section 1: Measuring Long-range Distance using a Sonar

### 1.1 Theory and Background

#### What is a Sonar?

A sonar is a sensor that uses the propagation of ultrasonic waves to detect objects. Figure 1-1 shows an integrated sonar distance sensor. The sensor consists of an ultrasonic transmitter, receiver, and signal conditioning circuitry.



Figure 1-1: The MaxBotix sonar sensor used in the Quanser Mechatronic Sensors Board

As illustrated in Figure 1-2, the transmitter generates a spherical cone-shaped ultrasonic pulse or ping which, when reflected off a nearby object, is picked up the receiver. The signal conditioning circuit measures the time it takes for the ultrasonic ping to travel to the object, bounce off, and be picked up by the receiver. Since sound waves travel at a constant speed (343 m/s in dry air at 20 °C), the speed of the transmitted and reflected ping will be the same, and hence the distance, *d*, from the reflecting object can be determined using Equation 1-1:

Equation 1-1

Where *d* is the distance between the sensor and the reflecting object, *c* is the speed of sound, and *t* is the time it takes for the transmitted ultrasonic ping to travel from the transmitter back to the receiver. Sonars typically generate several synchronized ultrasonic pings per second.

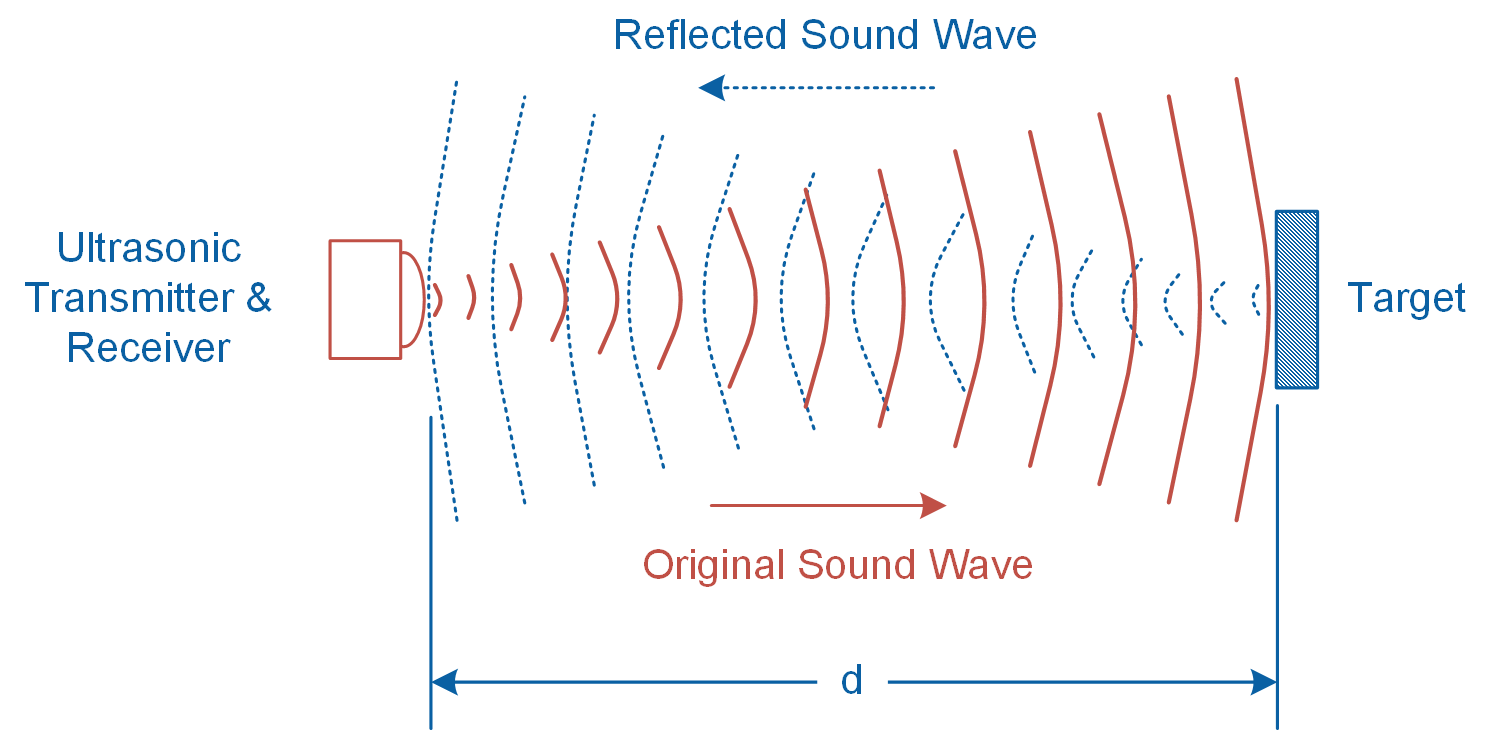


Figure 1-2: Sonar technique transmits and receives ultrasonic pings

Sonar distance sensors are ideal for long-range distance measurements as well as detecting large objects. The sensor mounted on the Quanser Mechatronic Sensors Board can measure objects ranging from 6 to 254 inches with a resolution of ±1 inch. Prior to being measured by the data acquisition system, the signal generated by the sonar is amplified using the non-inverting amplifier illustrated in Figure 1-3. The relationship between the input and output voltages of the amplifier is determined using Equation 1-2:

Equation 1-2

Where *Vin* is the signal generated by the sonar sensor, *VO* is the amplified signal, and resistors *R1* = 10 k and *R2* = 10 k determine the gain of the amplifier as follows: *G* = 1 + *R2*/ *R1.*

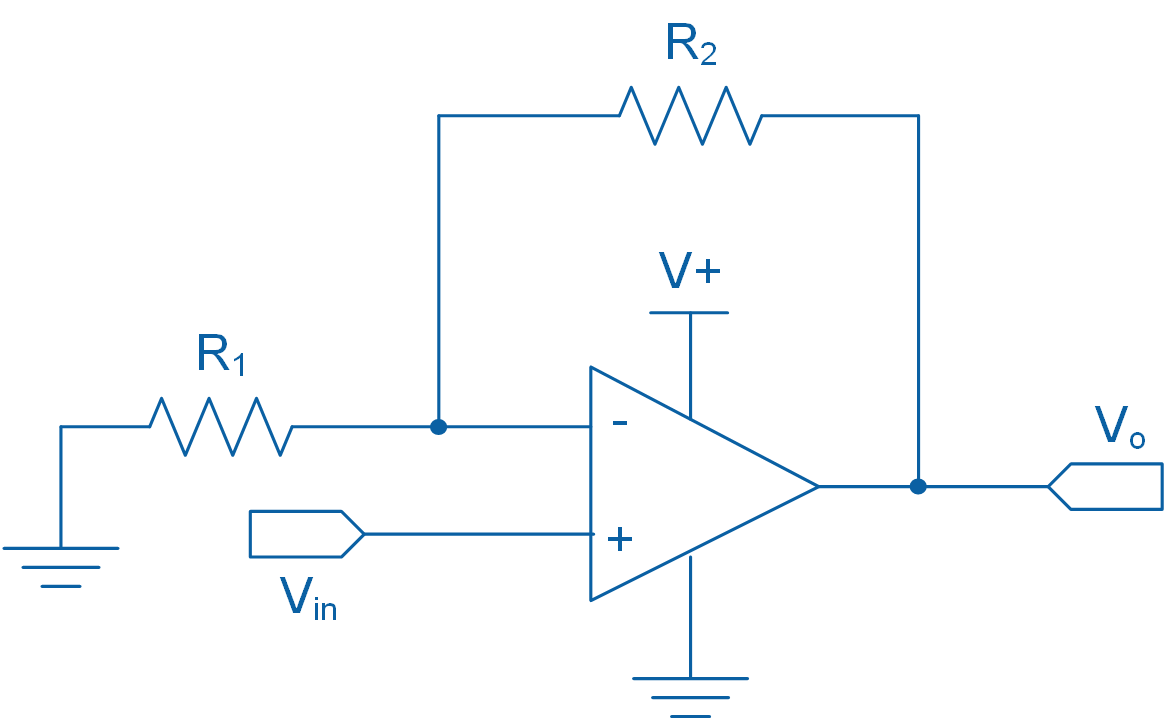


Figure 1-3: Non-inverting amplifier circuit diagram

In general, sonar sensors are not suitable for close-range measurements and their resolution is relatively coarse. Furthermore, they may not detect soft objects such as clothing, blankets, and porous materials. Since the sonar technique relies on measuring reflected ultrasonic sound waves, sonar does not work in space.

### 1.2 Implement

The Virtual Instrument (VI) used to collect data from and calibrate the sonar is shown in Figure 1-4.

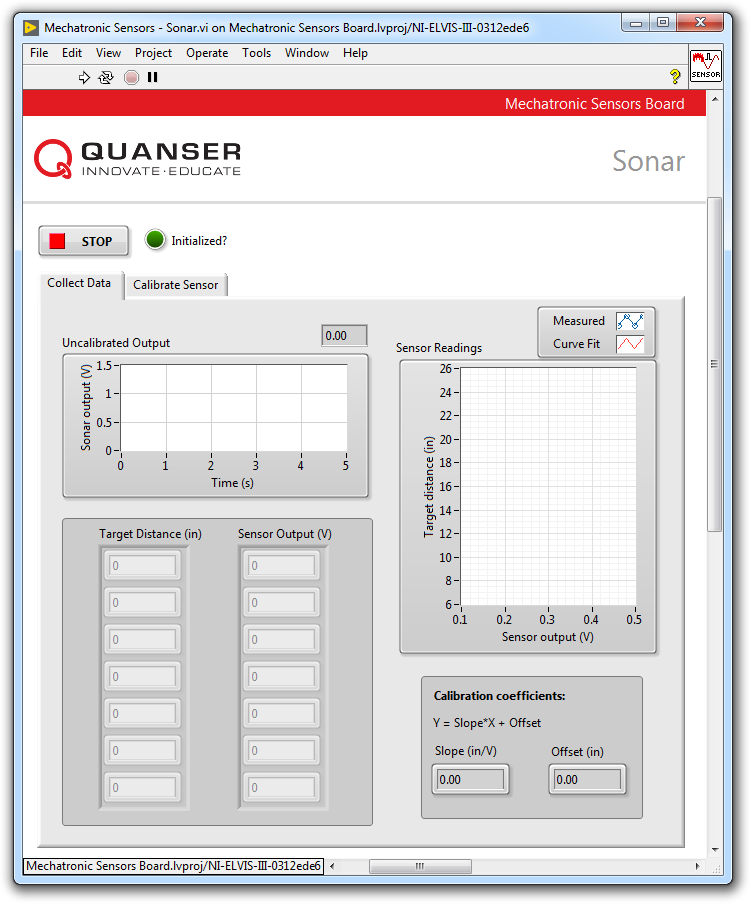


Figure 1-4 VI for collecting data from the sonar

#### Collect Data

1. Prior to conducting this lab, use the **Application Board Power** switch to cycle the Quanser Mechatronic Sensors Board power. The MaxBotix sonar sensor performs a self-calibration on power-up to compensate for ambient temperature and humidity. Ensure that there are no objects close to the sensor while cycling the power. Best sensitivity is obtained when the sensor’s detection area is clear of any objects for 14 inches during power-up.
2. Open **Mechatronic Sensors Board.lvproj**
3. From the **Project Explorer** window, open **Mechatronic Sensors - Sonar.vi**
4. Click on the **Collect Data** tab.
5. Run the VI.
6. Wait for the **Initialized?** LED indicator to turn on.
7. Enter **7** in the **Target Distance (in)** array.
8. Hold a sturdy cardboard target that is sized 12 in by 12 in at a distance of 7 inches away from the surface of the application board. Using the **Uncalibrated Output** waveform chart, read the corresponding sensor output and enter the value in the **Sensor Output (V)** array.  
     
   *Note:* Since the sensor generates a wide cone-shaped pulse, do not stand too close to the sensor while taking measurements. It will cause the sensor to detect your body which will interfere with your measurements. When measuring targets at longer distances, you will need to stand back and stretch out your arm while holding the target to get a valid measurement.
9. Continue taking measurements by moving the target in 3 inch intervals away from the sensor. Each time, enter the target distance and measured sensor outputs in the **Target Distance (in)** and **Sensor Output (V)** arrays respectively.  
     
   *Note*: Once the measured readings are entered, a linear curve is automatically generated to fit the data. The curve is shown in the **Sensor Readings** waveform graph. This curve represents the calibration curve of the sensor.
10. Record the collected data in Table 1-1.
11. The slope and offset of the calibration curve are automatically calculated by the VI and displayed in the **Slope (in/V)** and **Offset (in)** indicators. Record these values in Table 1-2.
12. Take a screenshot of the **Sensor Readings** graph.
13. Continue to the next section.

Table 1-1 Recorded sonar measurements

|  |  |
| --- | --- |
| Target Distance (in) | Sensor Output (V) |
| 7 |  |
| 10 |  |
| 13 |  |
| 16 |  |
| 19 |  |
| 22 |  |
| 25 |  |

Table1-2: Calibration coefficients

|  |  |
| --- | --- |
| Slope (in/V) | Offset (in) |
|  |  |

#### Calibrate the Sonar

1. Click on the **Calibrate Sensor** tab to calibrate the output of the sonar in terms of linear displacement of the target (in inches).
2. Use the **Slope (in/V)** and **Offset (in)** numeric controls to enter the slope and offset values you obtained during the data collection process.
3. Test the accuracy of your calibration. To do this, place the target at different known positions within the calibrated range, and verify that the correct distance is displayed in the **Calibrated Output** waveform chart as well as the **Distance (in)** slider indicator.
4. Using the **Calibrated Output** waveform chart, approximate the resolution of the sensor (in inches). Start by holding the target 7 inches away from the sensor. At a steady rate, slowly move the target away from the sensor and observe its response. The calibrated output of the sensor will have a step-like response. What is the smallest change in distance that it can detect? Take a screenshot of the response of the sensor. Compare your result with the resolution of the sensor provided in Section 1.1.
5. Press the **Stop** button.

### 1.3 Analyze

1-1 Present the results you recorded in Table 1-1.

1-2 Attach the screenshot of the Sensor Readings waveform graph showing the fitted calibration curve from Step 12.

1-3 Present the calibration coefficients that you recorded in Table 1-2.

1-5 What is the resolution of the sonar that you determined in Step 17? Compare your result with the resolution of the sensor provided in Section 1.1. Attach the screenshot of the step-like response of the sensor.

1-6 As noted in Section 1.1, the Quanser Mechatronic Sensors Board implements the non-inverting amplifier circuit shown in Figure 1-3. Using Equation 1-2 and the calibration coefficients recorded in Table 1-2, calculate the sensitivity of the sensor in terms of V/in? Assume R1 = R2 = 10 k.

## Section 2: Measuring Short to Mid-range Distance using a Time-of-Flight Sensor

### 2.1 Theory and Background

#### What is a Time-of-Flight Sensor?

A Time-of-Flight (ToF) sensor, pictured in Figure 2-1, uses the Time-of-Flight principle to measure short to mid-range distances. Time-of-Flight is a measurement technique used for measuring the distance between a sensor and a target, based on the time difference between the emission of a signal and its return to the sensor when reflected by the target. ToF sensors typically have a small footprint (several millimeters) and measure short to medium distances ranging from 100 mm to 2,000 mm. Most ToF sensors are digital sensors, which means communication with the sensors is achieved using SPI or I2C protocols.



Figure 2-1: A ToF sensor manufactured by RF Digital Corporation

As illustrated in Figure 2-2, a ToF sensor contains an infrared LED or low-power laser source. At regular intervals, the sensor pulses the light source. Each time the light source bounces off an object, the built-in receiver measures the “time-of-flight”, or how long it has taken for the beam to travel to the target and bounce back onto the receiver. The target distance can be determined using Equation 2-1:

Equation 2-1

where *d* is the measured target distance in meters, *td* is the time for the bounced beam to be detected by the receiver in seconds, and *c* is the speed of light in meters per seconds. For example, assuming *c* = 300,000,000 m/s, for an object located 2 meters away from the sensor, it takes approximately 13 nanoseconds for the pulsed beam to travel from the source to the receiver.

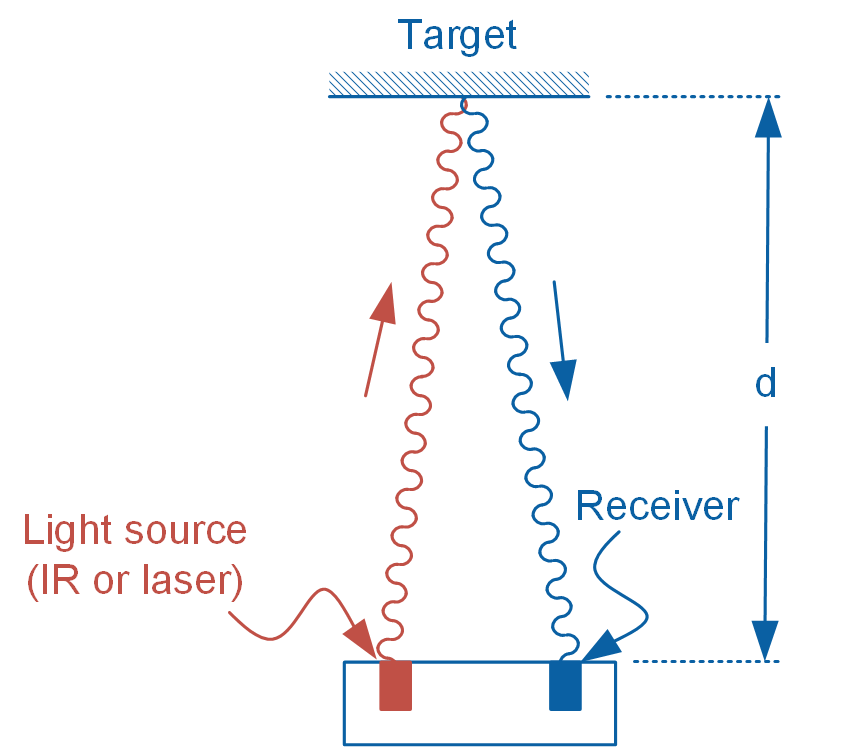


Figure 2-2: The principle of ToF

Since ToF technology uses light rather than a sonic pulse to detect objects, unlike a sonar, it can operate in vacuum. ToF sensors are widely used in robotics and consumer electronic goods. In robotic applications, for example, ToF sensors are used for obstacle detection and avoidance. White goods type of applications include hand detection in automatic faucets and soap dispensers. More advanced applications include gesture recognition, directional movement detection and volume or height control.

When designing a ToF measurement system, it is important to consider how environmental conditions may impact your measurements. For example, since ToF sensors typically contain an infrared light source, the presence of other light sources that contain a rich infrared component, such as a halogen light bulb, may cause the sensor to output erroneous values.

### 2.2 Implement

The Virtual Instrument (VI) used to collect data from the ToF sensor is shown in Figure 2-3.

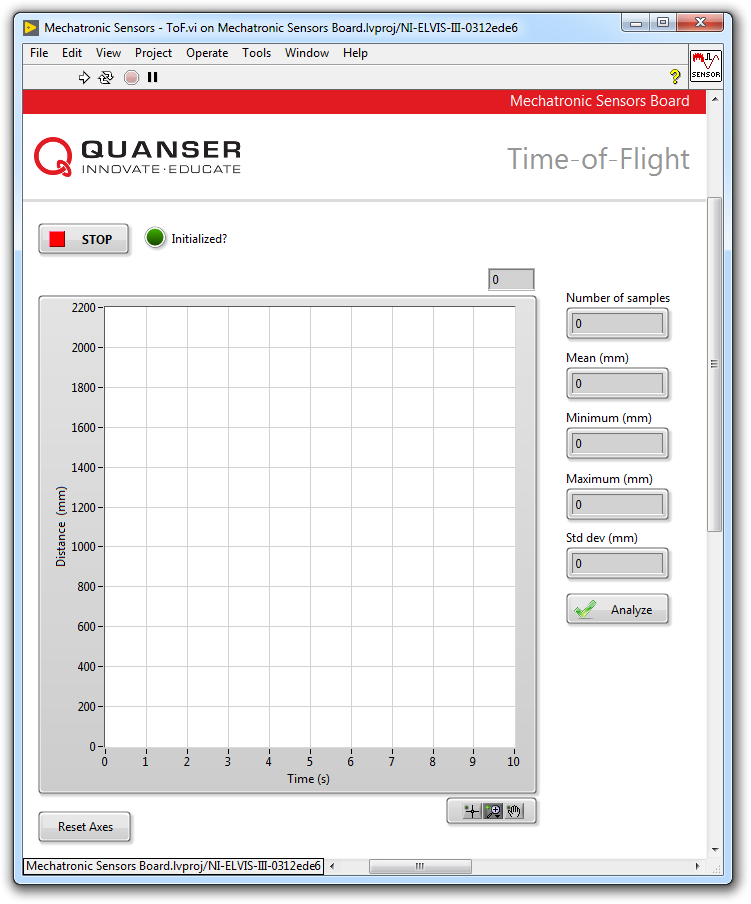


Figure 2-3: VI for collecting data from the ToF sensor

#### Observe Measurement Scatter

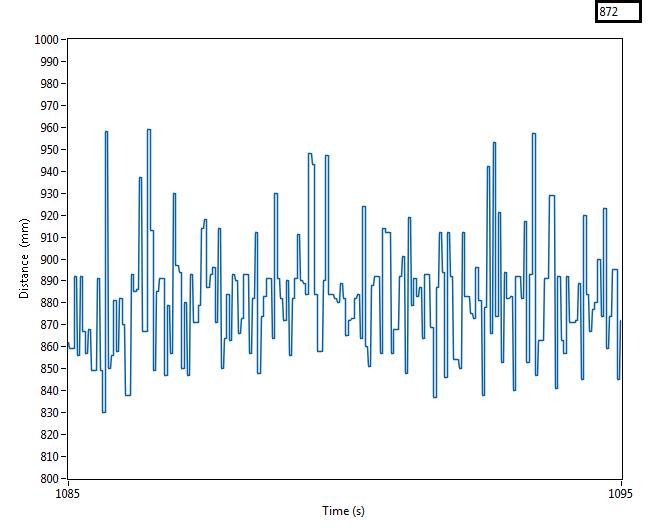
1. Open **Mechatronic Sensors Board.lvproj**
2. From the **Project Explorer** window, open **Mechatronic Sensors - ToF.vi**
3. Run the VI.
4. Wait for the **Initialized?** LED indicator to turn on.
5. The Quanser Mechatronic Sensors Board’s digital ToF sensor outputs an 11-bit value ranging between 0 and 2048, which directly corresponds to the target distance in millimeters. As such, you do not need to calibrate the output of the sensor in terms of measured distance.  
     
   However, the output of the ToF sensor exhibits random variation, or scatter, as shown in Figure 2-4. Observe this scatter by holding a sturdy cardboard target that is sized 12 in by 12 in at various random positions ranging between 100 mm and 1000 mm. Takes screenshots of your observations. Does the level of scatter change with target distance?  
   

Figure 2-4 Measurement scatter associated with the output of the ToF sensor

#### Quantify Measurement Scatter

1. Quantify the level of measurement scatter. To do this, steadily hold the target at an approximate distance of 100 mm from the ToF sensor. Wait for the readings to stabilize for at least **3 seconds,** and then click the **Analyze** button. The VI is programmed to collect 300 data points at a rate of 100 Hz. When the *Analyze* button is pressed, the VI calculates and displays the mean, minimum, maximum, and standard deviation of the acquired data. Record these values as your first trial in Table 2-1. Repeat the measurement twice more and record the data in Table 2-1.
2. Now repeat the previous step but this time hold the target at a distance of approximately 1000 mm. Record your results in Table 2-1.
3. Press the **Stop** button.

Table 2-1: Scatter data

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Approx. distance (mm) | Trial | Mean  (mm) | Max  (mm) | Min  (mm) | Std Dev  (mm) |
| 100 | 1 |  |  |  |  |
| 2 |  |  |  |  |
| 3 |  |  |  |  |
| 1000 | 1 |  |  |  |  |
| 2 |  |  |  |  |
| 3 |  |  |  |  |

### 2.3 Analyze

2-1 Detail your observations of the output of the ToF sensor as the target is placed at different distances from the sensor (Step 5). What trend did you visually observe as the target is positioned further from the sensor? Attach screenshots of your results.

2-2 Present the results you recorded in Table 2-1.

2-3 Standard deviation is a statistical measure of the amount of variation, or scatter, within a set of measured data points. A larger standard deviation implies larger scatter within the acquired data. In this lab, since measurements for a given target distance were repeated multiple times, you must apply *pooled statistics* to provide a single best statistical estimate of the measured data.

For each target distance, calculate the *pooled standard deviation* using the data recorded in Table 2-1 and the formula below:

where Spooled is the pooled standard deviation for a given target distance, N is the number of trials, and Si is the standard deviation of each trial. Compare the pooled standard deviations for the two different target positions. What do your results indicate about the scatter when the target is placed further away from the sensor?

## Section 3: Measuring Proximity using an Infrared Proximity Sensor

### 3.1 Theory and Background

#### What is an Infrared Proximity Sensor?

A typical reflective optical sensor is pictured in Figure 3-1. It consists of an infrared emitting diode (IRED) and a photodiode (detector). The emitting diode and detector are mounted side-by-side on parallel axes.



Figure 3-1: A digital proximity sensor manufactured by Broadcom Limited

When a surface is placed in the proximity of the sensor, the surface reflects the infrared light emitted by the IRED onto the photodiode. The further the surface is placed from the sensor, the less light is reflected onto the photodiode. The sensor will either output an analog signal that is proportional to the measured distance, or will output a digital equivalent of the measured distance.



Figure 3-2: Schematic diagram of a digital proximity sensor

Figure 3-2 schematically illustrates the operation of a digital proximity sensor. When powered, the IRED emits pulsed infrared light, part of which is reflected by the reflective target onto the photodiode. A photodiode is a semiconductor that converts light photons to electrical current. The induced current is then converted to a digital signal (typically counts) by the built-in Analog-to-Digital (ADC) converter. In the illustrated example, the output of the ADC is read using the I2C protocol.

The Mechatronic Sensors Board uses a Broadcom APDS-9190 digital IR proximity sensor. The user can set the number of IRED pulses (between 1 and 255) generated by the sensor during each cycle of operation. Each pulse has a period of 16.3 s. A higher number of pulses increases the sensitivity of the sensor. The sensor outputs a count value ranging between 0 and 1023. The value depends on the distance of the target and is a measure of the amount of reflected IR light. A count value of 1023 means the target has reached the proximity detection threshold of the sensor. Once the output saturates at 1023 counts, it will not increase even if the target moves closer to the sensor.

Proximity sensors are typically used in near field proximity applications. For example, in mobile phones, the sensor can detect when the user positions the phone close to their ear.

### 3.2 Implement

The Virtual Instrument (VI) used to collect data from the IR proximity sensor is shown in Figure 3-3.

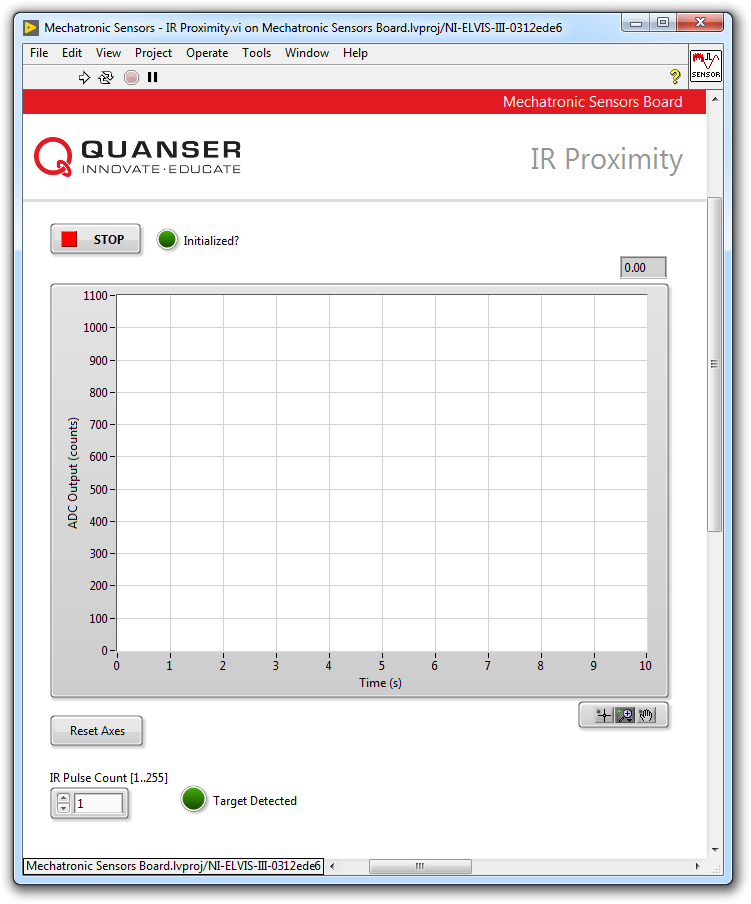


Figure 3-3: VI for collecting data from the IR proximity sensor

#### Observe Sensor Behavior

1. Open **Mechatronic Sensors Board.lvproj**
2. From the **Project Explorer** window, open **Mechatronic Sensors – IR Proximity.vi**
3. Run the VI.
4. Wait for the **Initialized?** LED indicator to turn on.
5. The waveform chart shows the raw count of the sensors analog-to-digital output.
6. Ensure the **IR Pulse Count [1..255]** numeric control is set to **1**. This setting causes the sensor to generate 1 IRED pulse during each cycle of operation.
7. Hold your hand at a distance of about 30 cm the sensor. Slowly move your hand toward the sensor and observe the response of the sensor; in particular, observe how the number of output counts increases as well as any scatter in the sensor’s output. Once your hand reaches the proximity threshold, the number of output counts will reach a maximum value of 1023. Using a ruler, estimate the threshold distance in terms of millimeters and record the value in Table 3-1. Take screenshots of your results.

Table 3-1: Sample recorded data

|  |  |
| --- | --- |
| IR Pulse Count | Proximity Threshold (mm) |
| 1 |  |
| 10 |  |
| 50 |  |
| 100 |  |
| 150 |  |
| 255 |  |

1. Stop the VI by pressing the **Stop** button.
2. As noted in section 3.1, a higher pulse count results in larger sensor sensitivity as well as a larger proximity threshold limit. To observe this, set **IR Pulse Count [1..255]** numeric control to **10** and re-run your VI and repeat step 7. Record your results in Table 3-1.
3. Repeat steps 8 through 9 for the remaining values of **IR Pulse Count** listed in Table 3-1.  
     
   *Note:* Prior to entering a new pulse count value, you must stop the VI each time. Enter a new value and re-run the VI.
4. Press the **Stop** button.

### 3.3 Analyze

3-1 Present the results you recorded in Table 3-1.

3-2 Detail your observations of the output of the IR proximity sensor as you moved your hand closer to the sensor for different pulse count values. Attach screenshots of your results.

3-3 Comment on the degree of scatter you observed for different IR pulse count values. Attach screenshots of your results.

3-4 Using the data recorded in Table 3-1, plot a curve that relates the sensor’s IRED pulse count to its proximity threshold. What type of curve best fits the data? Using this curve estimate the number of pulse count required to result in a proximity threshold of 70 mm. Run the VI and validate your results.

## Section 4: Design Considerations

4-1 You are asked to design a non-inverting amplifier for a sonar sensor similar to the one shown in Figure 1-3. The sensor has a sensitivity of 0.01 V/in and can measure a maximum distance of 254 in. At your disposal, you have a DAQ capable of measuring analog signals between 0 V and 10 V. Determine the maximum allowed gain of the circuit while satisfying the design constraints. Recommend appropriate values for the gain resistors R1 and R2.

4-2 Ideally, a proximity sensor must yield reproducible results regardless of the target color, surface texture, and surface reflectivity. In practice, however, IR proximity sensors are highly dependent on the ability of its target to reflect IR light. Examine and comment on the behavior of the IR proximity sensor that you used in Section2 using targets with different reflectivity. In particular, examine the effect of surface reflectivity on the proximity threshold. Use a white color cardboard and a black color cardboard as good and poor IR reflectors respectively. During your test set the **IR Pulse Count [1..255]** numeric control to **255**.

4-3 Devise an experimental plan to compare the performance and suitability of the sonar, ToF, and proximity sensors as distance sensors. Several key parameters to examine for each of the sensors include:

* Range
* Resolution
* Sensitivity
* Interface type (digital or analog)
* Target compatibility
  + Surface finish
  + Geometry
  + Color
* Operating environmental conditions
* Linearity

# Lab 3: Temperature



Figure 0: Many industries rely on the accurate measurement of temperature

This lab explores the thermo-resistive properties of a thermistor. A two-point calibration will be applied to the sensor and it thermal time-constant will be determined using a step input.

## Learning Objectives

After completing this lab, you should be able to complete the following activities:

* Apply a 2-point calibration to a thermistor
* Determine the -constant of a thermistor
* Apply a low-pass filter to the thermistor output
* Determine the thermal time-constant of a thermistor using a step input

## Required Tools and Technology

|  |  |
| --- | --- |
| Platform: NI ELVIS III | * View User Manual   http://www.ni.com/en-us/support/model.ni-elvis-iii.html |
| Hardware: Quanser Mechatronic Sensors Board | * View User Manual   http://www.ni.com/en-us/support/model.quanser-mechatronic-sensors-board-for-ni-elvis-iii.html |
| Software: LabVIEW Version 18.0 or Later  Toolkits and Modules:   * LabVIEW Real-Time Module * NI ELVIS III Toolkit | * Before downloading and installing software, refer to your professor or lab manager for information on your lab’s software licenses and infrastructure * Download & Install for NI ELVIS III * <http://www.ni.com/academic/download> * View Tutorials * http://www.ni.com/academic/students/learn-labview/ |
| Accessories:  * Thermometer * A small piece of clear tape |  |

## Expected Deliverables

In this lab, you will collect the following deliverables:

* Fingertip temperature measured using a thermometer
* Steady-state thermistor output when measuring fingertip temperature
* Calculate the resistance of the thermistor when measuring finger temperature
* Determine the -parameter of the thermistor
* Determine the thermal time constant of the thermistor under different conditions

Your instructor may expect you to complete a lab report. Refer to your instructor for specific requirements or templates.

## Section 1: Measuring Temperature using a Thermistor

### 1.1 Theory and Background

#### What is a Thermistor?

There are several different types of transducers available to measure temperature: thermocouple, resistance temperature detector (RTD), thermistor, and integrated circuit type. Each has their own advantages and disadvantages. Integrated circuit type sensors are low cost and have a linear output, but because they are mounted on a PCB they are board-layout dependent. Thermocouples have a wide temperature range, are relatively cheap, and are easy to use but are the least stable and sensitive. An RTD, on the other hand, is most stable and accurate of the sensors but is slow, has a fragile construction, and is relatively the most expensive. A thermistor responds very quickly, and has the highest sensitivity but has a limited temperature range. Figure 1-1 compares the typical sensitivity of the three temperature sensors.

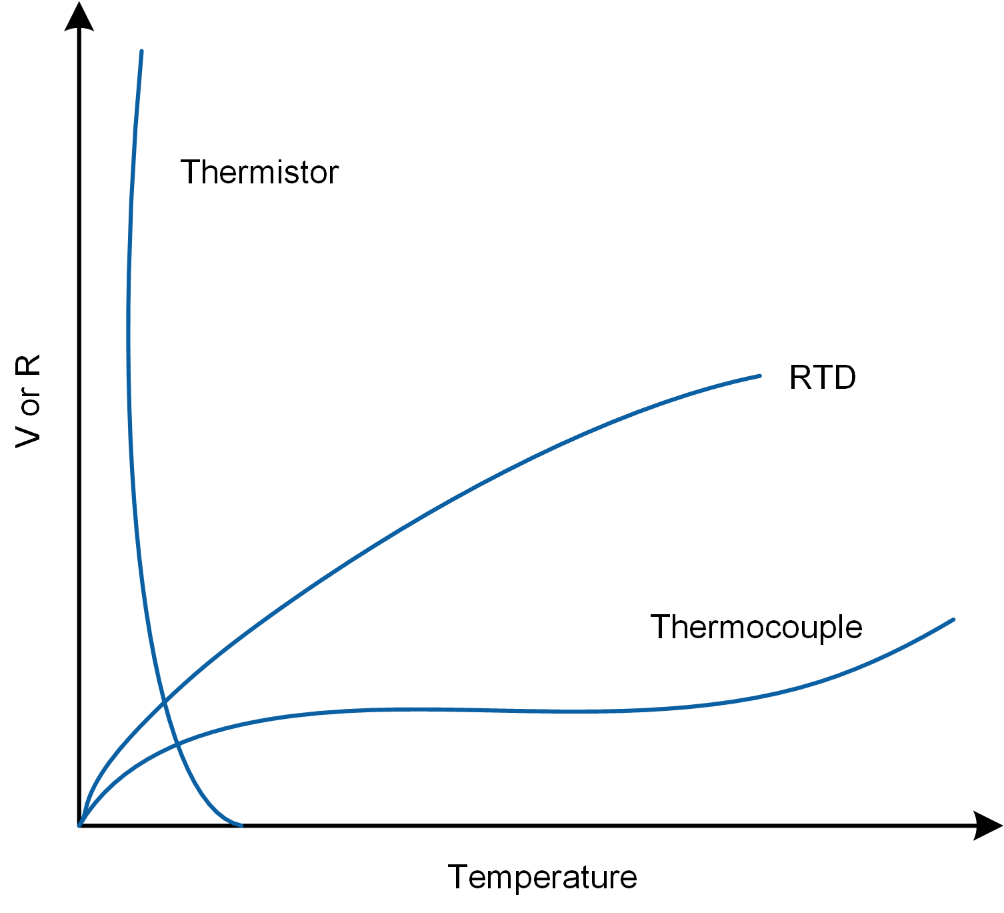


Figure 1-1: Comparison between the sensitivity of thermistors, RTD, and thermocouples

Thermistors differ from thermocouples in that they are resistive sensors with the latter being a voltage generating sensor. Because a current must be passed through a thermistor, it is susceptible to Ohmic heating. This self-heating manifests itself in the form of measurement error. Figure 1-2, illustrates various shapes of common thermistors.

|  |  |
| --- | --- |
| C:\Users\amolki\Desktop\QNET Mechatronic Sensors - Thermistor - Concept Review.jpg  (a) board-mount type | C:\Users\amolki\Desktop\QNET Mechatronic Sensors - Thermistor - Concept Review.jpg  (b) string type |

Figure 1-2: Different types of common thermistors manufactured by Murata

As a resistive sensor, the resistance of a thermistor is dependent on temperature. The relationship between the resistance of the thermistor, *RT*, and temperature, *T*, can be described using the -parameter equation:

Equation 1-1

where *R0* is the nominal resistance of the sensors at temperature *T0* in Kelvin, and  is a parameter which depends on the material, temperature, and construction of the thermistor and typically ranges between 3,500-4,700 K. For the thermistor on the Mechatronic Sensors board, the nominal sensor resistance is:

*R0* = 47,000 ohm

when the temperature is at 25 degrees Celsius or:

*T0 =* 298.15 K

The  -parameter equation provides an acceptable estimation of the measured temperature in applications that have a narrow temperature span (typically less than 20°C). The -parameter can be established using two temperature reference points.

A more accurate estimation temperature (±0.01°C over a 100°C span) can be made using the Steinhart-Hart equation as given in Equation 1-2:

Equation 1-2

where *A*, *B*, and *C* are the Steinhart-Hart parameters which are determined by means of a 3-point calibration process, and *RT* is the resistance of the thermistor at temperature *T* in Kelvin.

#### NTC and PTC Thermistors

Two types of thermistors are available: Negative Thermal Coefficient (NTC) and Positive Thermal Coefficient (PTC). The resistance of an NTC thermistor decreases with an increase in temperature. Conversely, the resistance of a PTC thermistor increases when the temperature increases. NTC thermistors are more sensitive compared to PTC thermistors, and therefore exhibit a much higher change in resistance when exposed to the same levels of temperature.

#### Measuring the Output of a Thermistor

Similar to most resistive sensors, the output of a thermistor is measured using a voltage dividing circuit. Figure 1-3 illustrates the voltage dividing circuit used in the Mechatronic Sensor board, where the thermistor is labeled as *RT*.

Using the voltage divider rule, the output voltage of the circuit shown in Figure 1-3 is by Equation 1-3:

Equation 1-3

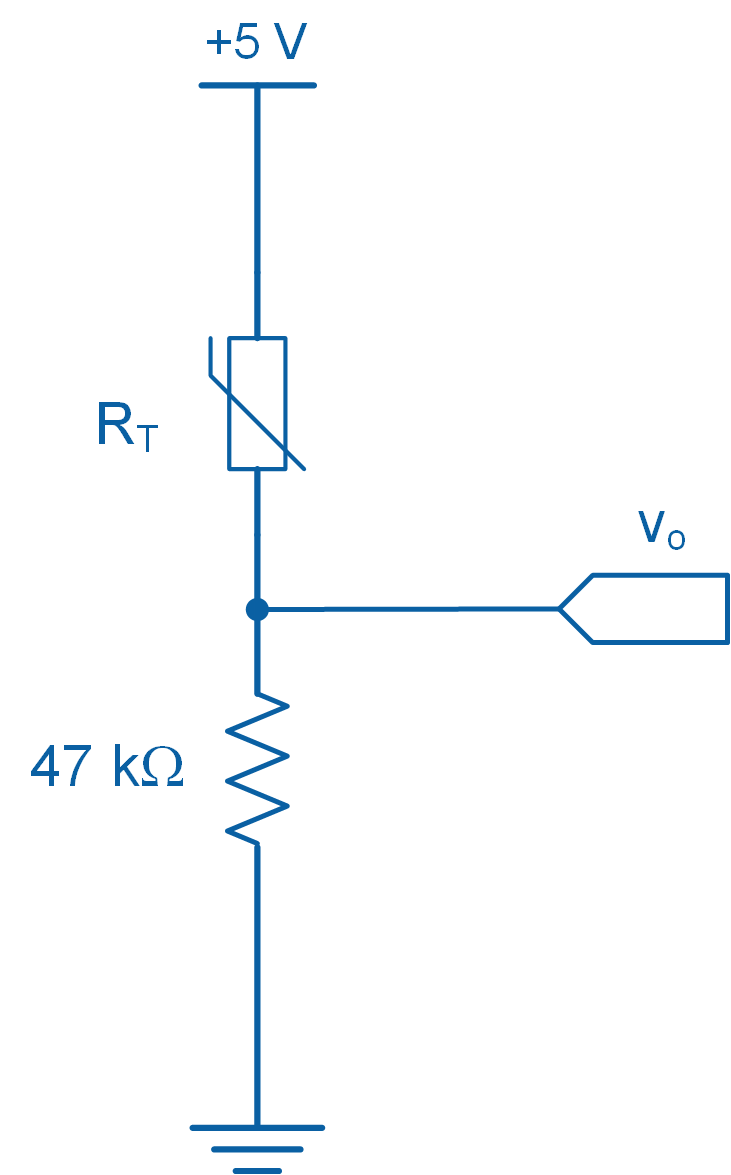


Figure 1-3: Thermistor circuit used in the Mechatronic Sensors board

#### Time Response of a Thermistor

Temperature sensors respond differently to changes in their input. For example, a liquid-filled bulb thermometer when taken outside on a cold winter day responds slowly to the change in temperature. In other words, the bulb thermometer does not instantaneously respond to a change in temperature. Time response is a measure of the time it takes for a sensor to respond to change. Typically sensors with fast response times are desirable.

#### Thermal Time Constant

Thermal time constant () is a measure of a sensor’s response to change and is defined as the time it takes for the sensor’s output to reach 63.2% (= 1-1/e*)* of the steady-state condition from an initial condition.

Figure 1-4, illustrates a typical time response curve. It shows how a sensor’s output behaves as it reaches steady-state. As shown in Figure 1-4, after one time constant () the system output reaches 63.2% of its steady-state value, after 2 time constants (2) the system reaches 86.5% of its steady-state value, and 95% is reached after 3 time constants (3). Theoretically, steady-state is achieved after infinite time, in practice, however, one waits until the output of the system or sensor is within an acceptable margin of error (typically 3 time constants). In thermal applications, time constants are typically large. Thermistors typically have a time constant of 0.5-4 seconds.

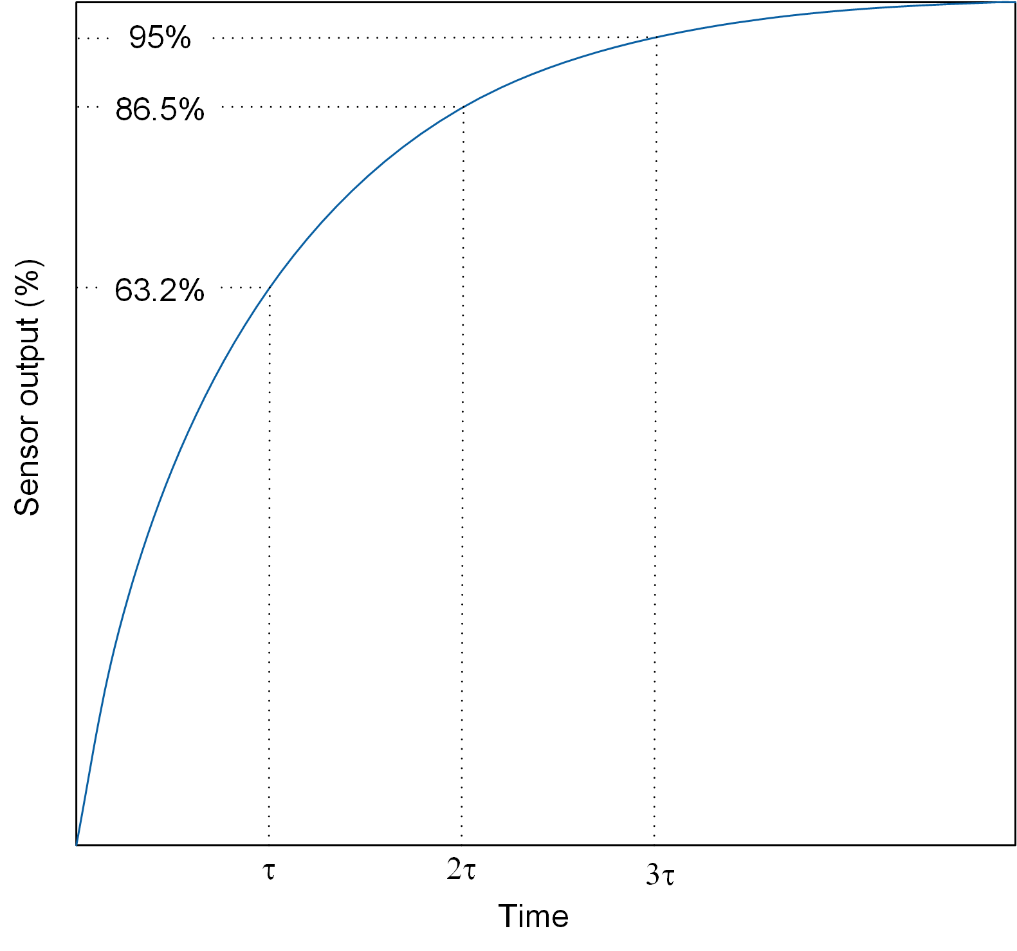


Figure 1-4: Time constant curve

### 1.2 Implement

The Virtual Instrument (VI) used to collect data from and calibrate the thermistor is shown in Figure 1-5.

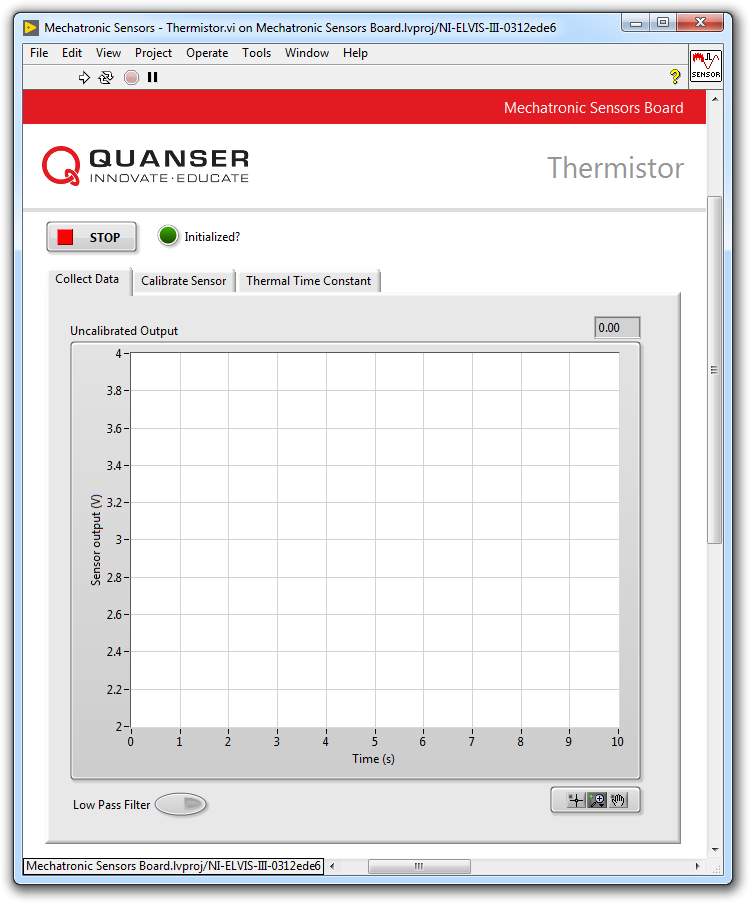


Figure1-5: VI for collecting data from the thermistor

#### Collect Data

1. Prior to starting this experiment, measure the temperature of your thumb using a thermometer and record the value in Table 1-1.  
     
   Note: In order to successfully conduct this experiment, your fingertip temperature must exceed 30 °C.
2. Open **Mechatronic Sensors Board.lvproj**
3. From the **Project Explorer** window, open **Mechatronic Sensors - Thermistor.vi**
4. Click on the **Collect Data** tab.
5. Run the VI.
6. Wait for the **Initialized?** LED indicator to turn on.
7. Gently place your fingertip on the thermistor and examine and take a screenshot of the response.  
     
   Note: If the output of the sensor is noisy, enable the **Low Pass Filter** button. The process filtering attenuates unwanted components from the sensor output, resulting in a “smoother” output signal.
8. Using the **Uncalibrated Output** waveform chart, observe the corresponding sensor output when it reaches steady-state (after approximately 15 s). Record the value in Table 1-1
9. Continue to the next section.

Table 1-1: Recorded fingertip temperature and steady-state sensor output

|  |  |
| --- | --- |
| Fingertip Temperature (°C) | Sensor Output (V) |
|  |  |

#### Calibrate the Thermistor

1. Using the steady-state output of the sensor from the data collection section and Equation 1-3, determine the resistance of the thermistor. Record this value in Table 1-2.
2. Using the temperature of your fingertip, which you determined in the data collection step and Equation 1-1, determine the -parameter of the thermistor. Record this value in Table 1-2. The nominal resistance of the sensor is R0 = 47,000 ohm at 25 °C.

Table 1-2: Sensor resistance and -parameter

|  |  |
| --- | --- |
| Sensor Resistance (ohm) | Calculated -parameter |
|  |  |

1. In the VI, click on the *Calibrate Sensor* tab to calibrate the output of the thermistor in terms of temperature (in °C).
2. Enter the -parameter you calculated for the thermistor using the *B* numeric control.
3. Test if the calibrated temperature closely matches your fingertip temperature which you measured earlier. To do this, gently place your fingertip on the sensor and verify that the correct fingertip temperature is displayed in the *Calibrated Output* waveform chart as well as the *Temperature (C)* thermometer indicator.
4. Press the *Stop* button.

### 1.3 Analyze

1-1 What is the temperature of your fingertip that you recorded in Table 1-1?

1-2 What is the steady-state voltage output of the thermistor that you recorded in Table 1-1? Was the signal noisy? Attach a screenshot of your results.

1-3 What is the resistance of the sensor that you recorded in Table 1-2? Show your calculations.

1-4 What is the -parameter of the thermistor that you recorded in Table 1-2? Show your calculations.

1-5 In step 14, how closely did the calibrated fingertip temperature match the temperature you recorded in Table 1-1?

## Section 2: Design Considerations

2-1 Assume that you are tasked to select a thermistor to measure the temperature of the air inside a duct. A design constraint requires that the output of the sensor reaches steady-state in less than 5 seconds so that a safety valve can be operated to avoid overheating the system. Determine if the thermistor used in the Mechatronic Sensors Board meets this design criterion. Assume steady-state is reached after 3 time constants.

*Hint:* In order to examine the suitability of the sensor you must determine its thermal time constant by following these steps:

* Run the VI and ensure the following:
  + If the sensor output is noisy, click on the *Collect Data* tab to activate the low-pass filter.
  + Click on the *Calibrate Sensor* tab and enter the -parameter you determined earlier in the *B* numeric control.
* Click on the *Thermal Time Constant* tab.
* Gently place your fingertip on the thermistor and examine the response using the *Calibrated Output* waveform chart.
* Once the sensor output has reached steady-state, click the *Plot* button. The response of the sensor will be captured in the *Thermal Time Constant* waveform graph as shown in Figure 2-1.
* Using the *Cursor* tool and the information provided in the Theory and Background section, determine the thermal time constant of the thermistor.

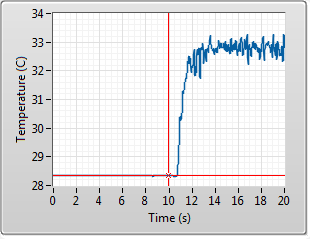


Figure 2-1: Typical thermistor response

2-2 While conducting this lab, you may have observed noise in the sensor output. To remedy this problem, the VI implements a software-based low-pass filter. A low-pass filter allows frequencies that are lower than a predefined cut-off to “pass”, while “blocking” or attenuating the remaining frequency information. This results in a “smoother” output signal. The downside of filtering is that it slows the response of the sensors. In this experiment, when you place your fingertip on the thermistor, your body acts as an antenna, attracting unwanted electromagnetic interference (EMI) from the environment. A common source of EMI includes AC power lines.

An alternative method of attenuating thermistor noise is to electrically insulate the sensor from touch. This can be achieved by covering the thermistor with a small piece of clear tape or cling wrap. Placing such barrier will electrically insulate the sensor from your fingertip, while allowing heat to be exchanged with the sensor.

Figure 2-2 contrasts the thermistor’s raw (unfiltered) output with the following cases: when LPF is switched on, and when the clear tape is placed on the sensor. The results indicate that placing a piece of clear tape on the sensor does a reasonably good job at filtering unwanted noise!

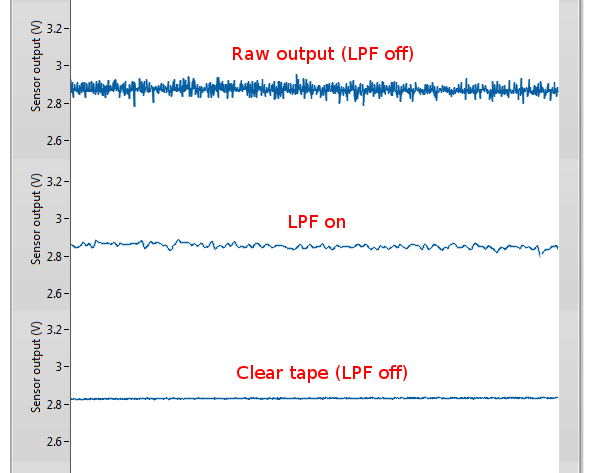


Figure 2-2: Thermistor raw and filtered responses

Your task is to observe the effectiveness of using a small piece of clear tape or cling wrap to attenuate thermistor noise. Record screenshots similar to those shown in Figure 2-2. Additionally, determine the thermal time constant of the thermistor when the clear tape is placed on the sensor (turn off the LPF during your investigation). Does your result indicate a slower responding thermistor?

# Lab 4: Strain Gage



Figure 0: Strain measurement is crucial in engineering design, as well as testing and maintenance practices

This lab explores the concept of strain measurement using a strain gage. The output of a strain gage mounted on a cantilever beam, placed in a quarter-bridge Wheatstone configuration, will be calibrated in terms of beam tip displacement. Furthermore, the natural frequency of the beam assembly will be measured by applying a fast Fourier transform to the response of the beam due to an impulse.

## Learning Objectives

After completing this lab, you should be able to complete the following activities:

* Understand the difference between quarter, half, and full Wheatstone bridge circuits
* Calibrate the output of a strain gage in terms of beam displacement
* Determine the natural frequency of a cantilever beam using a strain gage by applying a fast Fourier transform

## Required Tools and Technology

|  |  |
| --- | --- |
| Platform: NI ELVIS III | * View User Manual   http://www.ni.com/en-us/support/model.ni-elvis-iii.html |
| Hardware: Quanser Mechatronic Sensors Board | * View User Manual   http://www.ni.com/en-us/support/model.quanser-mechatronic-sensors-board-for-ni-elvis-iii.html |
| Software: LabVIEW Version 18.0 or Later  Toolkits and Modules:   * LabVIEW Real-Time Module * NI ELVIS III Toolkit | * Before downloading and installing software, refer to your professor or lab manager for information on your lab’s software licenses and infrastructure * Download & Install for NI ELVIS III * <http://www.ni.com/academic/download> * View Tutorials * http://www.ni.com/academic/students/learn-labview/ |
| AccessoriesFew small paper clips |  |

## Expected Deliverables

In this lab, you will collect the following deliverables:

* Record the zero offset of the bridge circuit
* Record the 5-point calibration data used to calibrate the strain gage
* Record the obtained calibration curve coefficients
* Screenshot of the calibration data showing the fitted curve
* Determine sensor sensitivity
* Determine the natural frequency of beam assembly
* Observe the relationship between mass and natural frequency

Your instructor may expect you to complete a lab report. Refer to your instructor for specific requirements or templates.

## Section 1: Measuring Strain

### 1.1 Theory and Background

#### What is Strain?

Strain is a measure of deformation in a solid body due to an applied force. Figure 1-1 illustrates a rectangular bar being subjected to an axial tensile stress ().

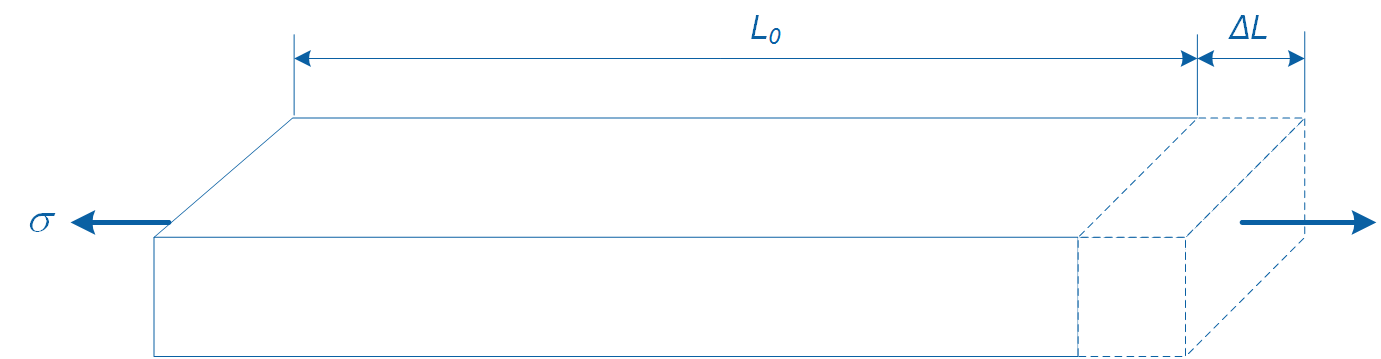


Figure 1-1: Change in length due to an axial tensile force

This stress causes a change in the original length of the bar from *L0* to *L0* + *L*. We define strain (**) using Equation 1-1:

Equation 1-1

where *L0* is the original length and *L* is the change in length due to the applied force. Strain is dimensionless and expressed as a percentage (%) or in mm/mm. However, since strain values are typically very small, strain is expressed in micro-strain () by multiplying strain by 106.

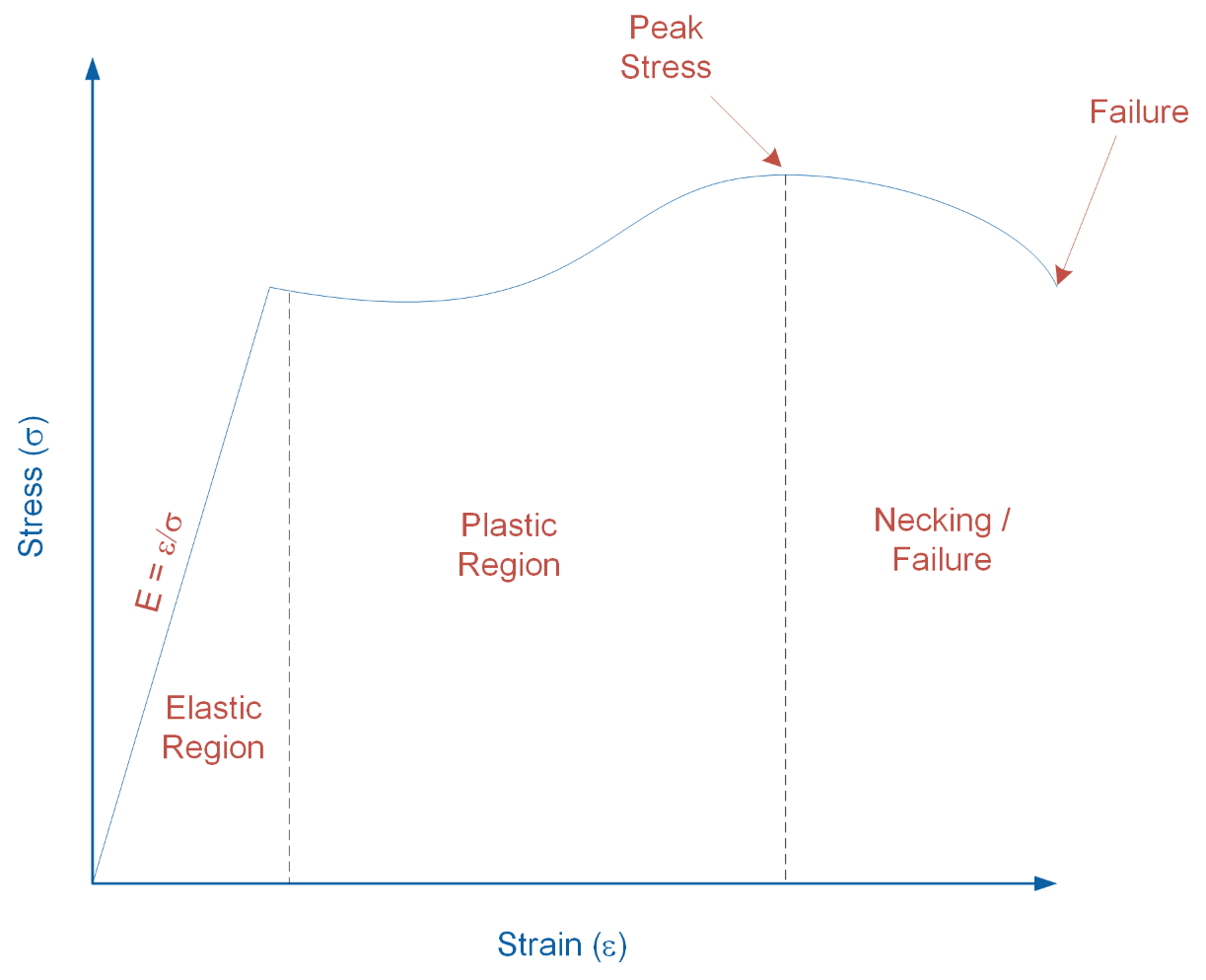


Figure 1-2: Stress-strain curve

Figure 1-2 illustrates the relationship between stress and strain when applied to a solid body. As shown in the graph, as stress increases, the solid body undergoes various deformation stages. In the elastic region, the body does not experience permanent physical change. In this region, the stress-strain relationship exhibits a linear relationship and we define the slope of the relationship as the modulus of elasticity (*E*=**) of the body. In the plastic region, the body permanently deforms, followed by the necking region where necking occurs prior to fracture.

#### Strain Gage

Strain gage is a sensor used for measuring strain in solid bodies. As shown in Figure 1-3, it is constructed from a fine metallic foil element formed into a grid pattern and mounted on a thin backing called a carrier. Strain gages are commonly bonded to test specimens using cyanoacrylate based adhesives or two-part epoxies. When an appropriate bond between the gage and specimen is established, any deformation in the specimen is transferred to the gage. This causes the resistance of the strain gage to change. When a strain gage is under tension its resistance increases, while under compression its resistance decreases.

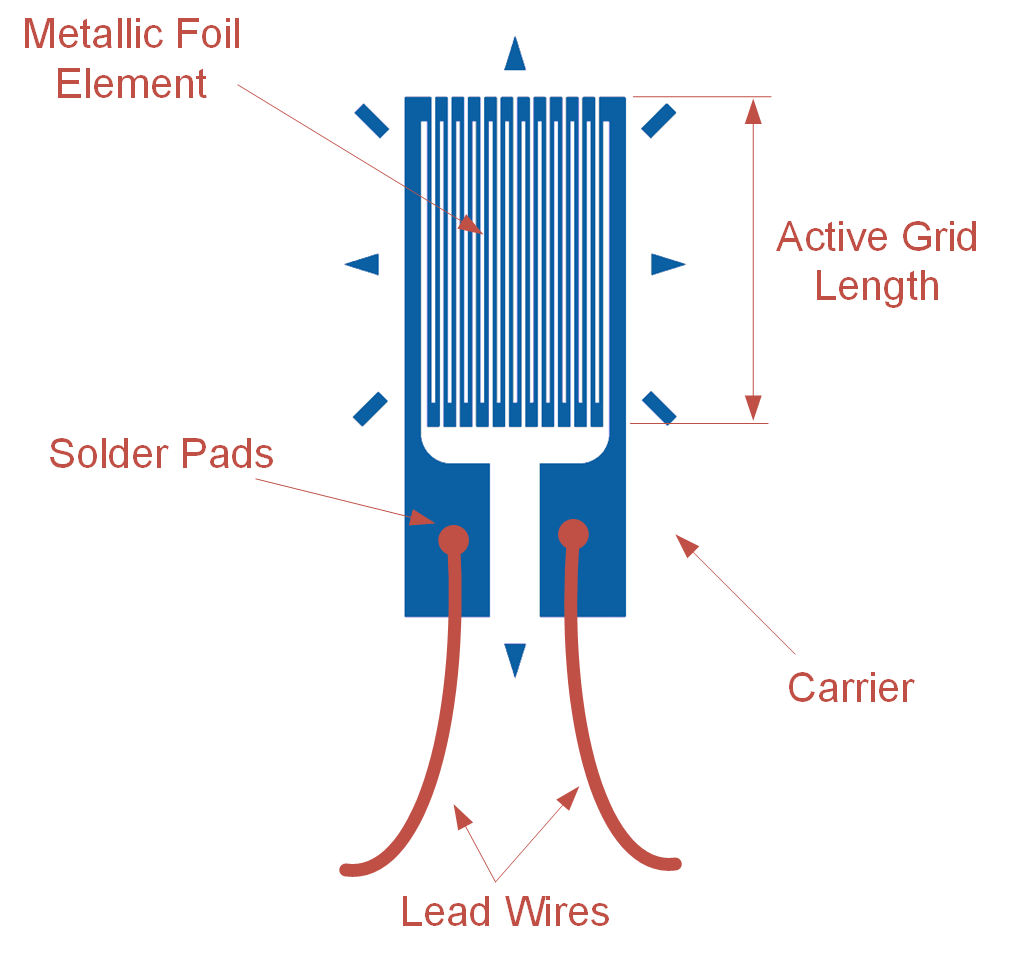


Figure 1-3: Schematic diagram of a strain gage

Strain gages vary in shape, orientation, and number depending on the type of strain being measured. The output of a strain gage is measured by connecting its lead wires to a dedicated signal conditioning circuit and DAQ, or dedicated strain measuring instrument.

Most strain gages have nominal resistances of 120 or 350 ohm. A higher nominal resistance and lower excitation voltage are desirable since that decreases measurement error due to lower Ohmic/self-heating effects.

The sensitivity to strain of a strain gage is called Gage Factor (GF) and is defined using Equation 1-2:

Equation 1-2

where *R* is the change in resistance when the strain gage is deformed, *RG* is the nominal resistance of the gage, and  is the induced strain. Typical gage factor values are approximately 2. For example, *GF* = 2 means if 1% strain is induced in a specimen, the gage’s relative resistance will change by 2%. Strain gages typically measure strains of up to 5% or 50,000 .

#### Wheatstone Bridge

The output of a strain gage is not directly measured; rather the voltage drop due to the change in the sensor’s resistance is measured using a Wheatstone bridge circuit (Figure 1-4). It offers several advantages over voltage dividing circuits, which are typically used for measuring the output of resistive sensors. One benefit is that a Wheatstone bridge allows for higher measurement sensitivity and lower measurement error. Another advantage is that it removes large fixed voltage drops which are present in a typical voltage dividing circuit. Since the output of Wheatstone bridge circuits is very low (typically in the range of microvolts), removing a large fixed voltage drop allows for the signal to be amplified using an amplifier.

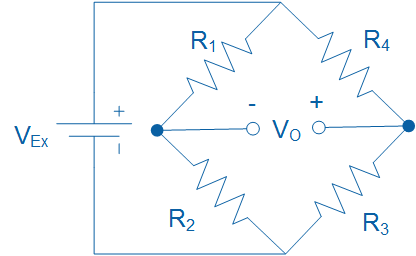


Figure 1-4: A Wheatstone bridge circuit for measuring the output of resistive sensors

The relationship between the resistors (*R1*, *R2*, *R3*, and *R4*), excitation voltage (*VEx*) and output voltage (*VO*) is governed by Equation 1-3:

Equation 1-3

For *VO* to be zero, the following relationship must hold true, in which case the bridge is said to be balanced:

Equation 1-4

However, when the resistance of one of the resistors changes value, the bridge circuit generates an output voltage and is said to become unbalanced. Generally, the following three distinct Wheatstone bridge configurations are used for measuring the output of strain gages: (a) quarter-bridge, (b) half-bridge, and (c) full-bridge configuration.

##### Quarter-bridge Configuration

Figure 1-5 illustrates a quarter-bridge Wheatstone bridge configuration. It consists of a single active strain gage (*RG*) and three fixed external precision resistors. It is the simplest strain measurement configuration and offers the lowest measurement sensitivity. Typically *RG* = 120 or 350 ohm when unstrained and the fixed resistors each equal 120 or 350 ohm, respectively.

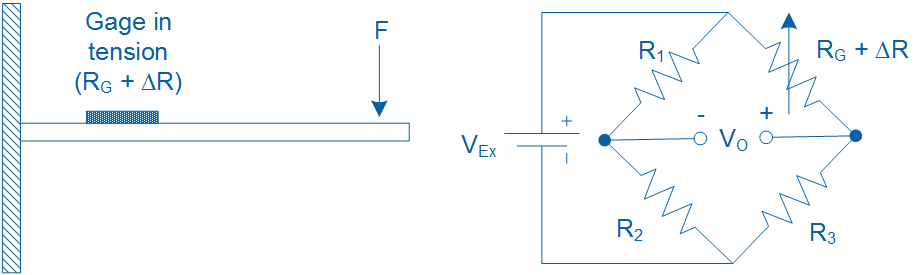


Figure 1-5: Quarter-bridge configuration for measuring strain in a cantilever beam

When a bending force is applied to the beam, it causes the beam along with the strain gage to deform. As a result, the resistance of the gage changes and a voltage output (*VO*) is generated which can be measured using a DAQ. The voltage output is proportional to the strain induced in the beam.

For example, assume that a bending force causes *RG*, which has a nominal resistance of 350 ohm, to increase by 0.0085 ohm. If the bridge is excited at 5 V, using Equation 1-3, the output voltage of the bridge circuit will be *V0* = -30 microvolts.

In practice, the output voltage of a quarter-bridge configuration is very minute and will require amplification to increase measurement resolution before being measured using a DAQ. Typical strain gage circuits or DAQs have built-in amplifiers to increase the signal levels to 10 mV/V (10 mV per each volt of excitation).

Assuming *R1* = *R2* = *R3* = *RG*, and substituting Eq. 1-2 in Eq. 1-3, the output voltage (*VO*) of the quarter-bridge circuit can be expressed in terms of *VEx*, *GF*, and the measured strain () as shown in Equation 1-5:

Equation 1-5

Note that the presence of the term 1/(1+GF∙/2) indicates non-linearity in the output of a quarter-bridge configuration with respect to strain.

##### Half-bridge Configuration

A half-bridge configuration uses two active strain gages and two fixed external resistors. Depending on the type of strain being measured (e.g. bending, torsion, tension, etc.), strain gages in a half-bridge configuration are mounted differently on a specimen. It offers the benefit of twice the sensitivity of a quarter-bridge configuration. Figure 1-6 illustrates a half-bridge configuration for measuring bending strain in a cantilever beam with two active gages mounted on opposite sides of the beam. When a bending force is applied to the beam, it causes tension in one of the gages while the other gage compresses.

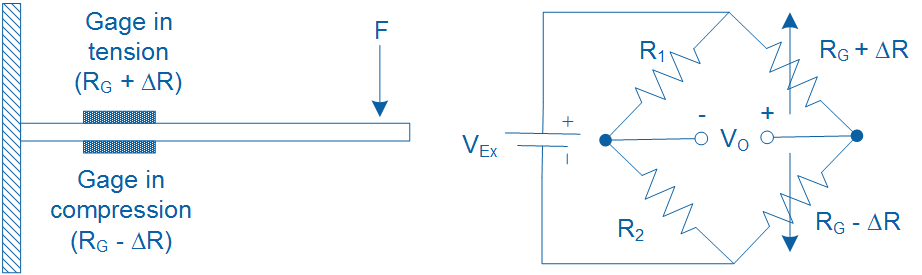


Figure 1-6: Half-bridge configuration for measuring strain in a cantilever beam

Assuming *R1* = *R2* = *RG*, and substituting Eq. 1-2 in Eq. 1-3, the output voltage (*VO*) of the half-bridge circuit can be expressed in terms of *VEx*, *GF*, and the measured strain () as shown in Equation 1-6:

Equation 1-6

##### Full-bridge Configuration

As illustrated in Figure 1-7, a full-bridge configuration uses 4 active strain gages of equal resistance (*RG*) and thus does not employ any external fixed resistors to complete the bridge circuit. Substituting Eq. 1-2 in Eq. 1-3, the output voltage of the full-bridge circuit (*VO*) can be expressed in terms of *VEx*, *GF*, and strain () as shown in Equation 1-7:

Equation 1-7

A full-bridge configuration produces twice the sensitivity of a half-bridge configuration and four times the sensitivity of a quarter-bridge configuration.

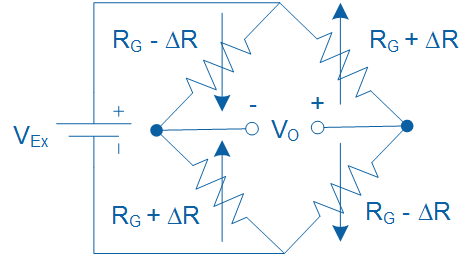


Figure 1-7: Full-bridge configuration for measuring strain

#### Effect of Temperature

In practice, change in temperature has a noticeable effect on the resistance of a strain gage, resulting in temperature induced strains. Such strains are caused by either self-heating of a strain gage, or due to differential thermal expansion between the strain gage and the specimen on which it is mounted.

Several practical methods exist to compensate for temperature-induced strain. One configuration, called quarter-bridge type II, uses one active gage and one dummy gage. This configuration is illustrated in Figure 1-8. The dummy gage is either mounted onto an identical unstrained secondary specimen which is placed in the proximity of the strained specimen, or it is mounted on the same specimen but in the transverse direction. Assuming *R1* = *R2* = *RG*, the output voltage (*VO*) of the quarter-bridge type II circuit can be expressed in terms of *VEx*, *GF*, and the measured strain () using Equation 1-5.

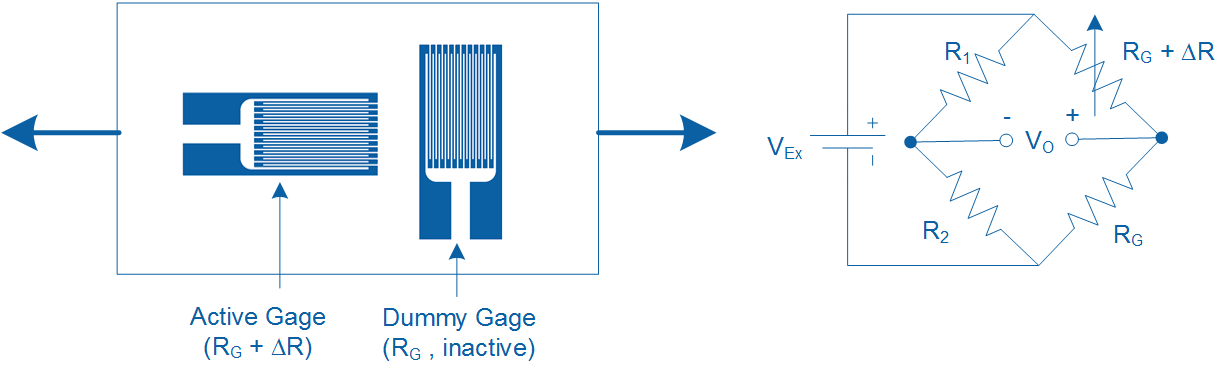


Figure 1-8: Quarter-bridge type II temperature compensating configuration

In this configuration, both the active and dummy gages experience the same fluctuations in temperature, with the resulting temperature induced strain canceling each other in the bridge configuration. Thus, any measured strain is due to active gage experiencing load-induced strain.

A more practical method of compensating for temperature-induced strain involves using a 3-wire self-temperature compensating strain gage. Such gages are made of alloys whose change in resistance due to temperature counters the change in resistance due to the differential thermal expansion between the gage and specimen. One of the limitations of such gages is that they should be mounted only on certain types of specimen.

#### Strain Gage Calibration

Strain gage calibration is the process of determining the mathematical relationship between the output of the Wheatstone bridge circuit versus the physical quantity being measured. Depending on the application the output of the bridge circuit may be calibrated to indicate strain (), deflection (mm), or mass (kg) by applying a series of known forces, displacements, or masses respectively. As part of the calibration process, the user must first adjust the *zero offset* and *full-scale span*.

Zero offsetting is the process of adjusting a Wheatstone bridge output to zero under no-load conditions. Such zero offsets exist because of normal tolerances in strain gage assemblies. This process, which establishes a reference point for measurement, is also referred to as null offsetting. Null offsetting is done using external resistors or via potentiometers built into the amplifier circuit of the strain measurement DAQ. Alternatively, zero offsetting can be achieved by means of software compensation rather than removing the offset off the bridge. However, if the offset is large enough, software compensation will limit the dynamic range of the measurement.

Full-scale span is the output range of a Wheatstone bridge circuit when the gage is subjected to maximum and minimum deflection. In practice, setting full-scale span requires the user to deflect the beam/gage assembly to its maximum or minimum positions and adjust the Wheatstone bridge output to a desired level using a gain potentiometer. Full-scale span is sometimes referred to as full-scale output (FSO) in sensor documentation.

Once zero offset and span have been adjusted, the user must apply three to five known inputs (e.g. deflection or load) to the specimen/strain gage assembly, and record the corresponding output of the bridge circuit. A calibration equation is then obtained by fitting a line to the measured points. Once the calibration equation is determined, it can be used to calculate the calibrated physical quantity for any given output of the bridge circuit.

### 1.2 Implement

The Virtual Instrument (VI) used to collect data from and calibrate the strain gage is shown in Figure 1-9.

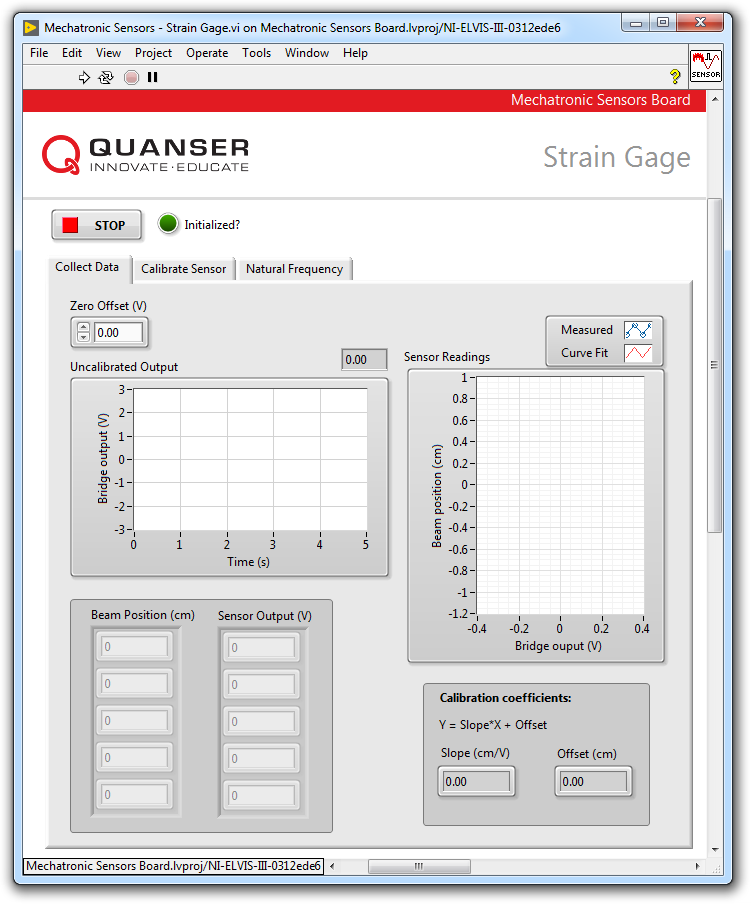


Figure1-9: VI for collecting data from the strain gage

#### Collect Data

1. Open **Mechatronic Sensors Board.lvproj**
2. From the **Project Explorer** window, open **Mechatronic Sensors – Strain Gage.vi**
3. Click on the **Collect Data** tab.
4. Run the VI.
5. Wait for the **Initialized?** LED indicator to turn on.
6. Using the **Uncalibrated Output** waveform chart, read the initial strain gage bridge output.
7. Balance the output of the strain gage bridge. To do this, make sure position and hold the tip of the cantilever beam at the 0 cm mark. Adjust the **Zero Offset (V)** numeric control such that the uncalibrated output of the bridge circuit is as close as possible to 0.00 V. Record the zero offset value in Table 1-1.
8. Enter -1 in the **Beam Position (cm)** array.
9. Flex the tip of the cantilever beam to the -1 cm mark.
10. Read the corresponding strain gage output and enter the value in the **Sensor Output (V)** array.
11. Repeat the process by moving the tip of the beam to the following positions: -0.5 cm, 0 cm, +0.5 cm and +1 cm. Each time, enter the beam position and measured sensor outputs in the **Beam Position (cm)** and **Sensor Output (V)** arrays respectively.
12. Once the measured readings are entered, a linear curve is automatically generated to fit the data. The curve is shown in the **Sensor Readings** waveform graph. This curve represents the calibration curve of the sensor. Take a screenshot of the graph.
13. The slope and offset of the calibration curve are automatically calculated by the VI and displayed in the **Slope (cm/V)** and **Offset (cm)** indicators. Make a note of these values in Table 1-2.
14. Record the collected data in Table 1-3.
15. Take a screenshot of the **Sensor Readings** graph.
16. Continue to the next section.

Table 1-1: Recorded bridge zero offset

|  |  |
| --- | --- |
| Zero offset (V) |  |

Table1-2: Calibration coefficients

|  |  |
| --- | --- |
| Slope (cm/V) | Offset (cm) |
|  |  |

Table 1-3: Recorded bridge output

|  |  |
| --- | --- |
| Beam Position (cm) | Bridge Output (V) |
| -1.0 |  |
| -0.5 |  |
| 0.0 |  |
| +0.5 |  |
| +1.0 |  |

#### Calibrate the Strain Gage

1. Click on the **Calibrate Sensor** tab to calibrate the output of the strain gage bridge circuit in terms of linear displacement of the tip of the cantilever beam (in cm).
2. Use the **Slope (cm/V)** and **Offset (cm)** numeric controls to enter the slope and offset values you obtained during the data collection step.
3. Test the accuracy of your calibration. To do this, flex the cantilever beam to different positions and verify that the correct position is displayed in the **Calibrated Output** waveform chart as well as the **Beam Position (cm)** slider indicator.
4. Press the **Stop** button.

### 1.3 Analyze

1-1 What is the initial bridge zero offset that you recorded in Table 1-1?

1-2 Present the calibration coefficients that you recorded in Table 1-2.

1-3 Present the calibration data you recorded in Table 1-3.

1-4 Attach a screenshot of the Sensor Readings waveform graph showing the fitted calibration curve from step 12.

1-5 What calibration equation did you obtain?

1-6 What is the sensitivity of the amplified bridge circuit in V/cm?

1-7 How well did your calibrated output match the actual beam tip position in step 19?

1-8 Based on the data you collected in Table 1-1, use Equation 1-5 in to determine the maximum and minimum strains induced in the cantilever beam as it’s flexed between -1 cm and +1 cm. For each case, determine if the strain gage is under tension (or compression). Assume a gage factor of GF = 2, a bridge excitation voltage of VEx = +5 V, and an amplification gain of 100.

*Hint:* The output of the bridge circuit is amplified prior to being displayed in the VI. All calculations done using Equation 1-5 must be done using actual (i.e. pre-amplified) bridge output values.

## Section 2: Design Considerations

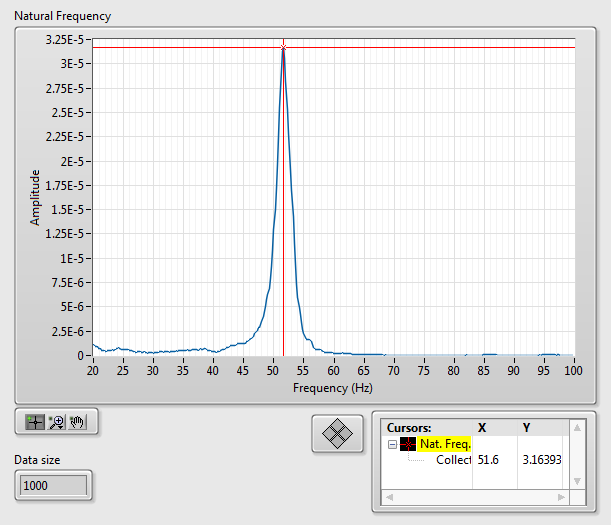
2-1 A strain gage can be used to indirectly measure other physical quantities such as vibration. Accurate measurement of vibration is imperative in ensuring the health of electro-mechanical systems. For example, excessive vibration can cause fractures in the fuselage of an airplane, or cause solder joints to break free in electronic circuit boards. Strain-based vibration switches are commonly used as a simple protection device that senses vibration and triggers an alarm or shuts down a machine if vibration exceeds a certain threshold.

The strain gage mounted on the cantilever beam in the Mechatronic Sensors Board can be used to determine the natural frequency of the beam assembly. Natural frequency is a property of an object that quantifies the frequency at which it “wants” to naturally vibrate when subjected to a disturbance. If a system has a natural frequency that matches normal environmental vibration, then the system will vibrate more violently and may prematurely fail.

  
A structure may fail if its natural frequency matches environmental vibration

Determine the natural frequency of the beam assembly by following the steps below:

* Run the VI.
* Click on the **Natural Frequency** tab.
* Ensure the beam is at rest (i.e. not vibrating).
* With one finger gently bend and then release the tip of the beam.
* Wait for a couple of seconds for the beam to stop vibrating then promptly press the **Stop** button.
* The VI will apply a fast Fourier transform to the captured data and display the result in the **Power Spectrum** waveform graph. The response should look similar to the figure below.
* Using the **Cursor** tool, measure the peak frequency.



Sample natural frequency response

2-2 Natural frequency of a cantilever beam can be calculated using the following equation:

where *fn* is natural frequency in Hz, *k* is stiffness of the beam in N/m, and *m* is the mass of the beam in kg. The equation indicates that natural frequency has an inverse relationship with the square root of mass. Validate this relationship by attaching a small paper clip to the tip of the cantilever beam and measuring the natural frequency of the modified beam. Does increasing the beam mass decrease its natural frequency?

# Lab 5: Pressure



Figure 0: Pressure measurement is an essential part of industrial automation

This lab explores pressure measurement using a capacitive pressure transducer. Reference points based on Boyle’s law will be established prior to calibrating the transducer. Both gage and absolute pressure will be measured.

## Learning Objectives

After completing this lab, you should be able to complete the following activities:

* Establishing calibration reference points based on Boyle’s law
* Conducting a 5-point sequential calibration of a pressure transducer
* Comparing upscale and downscale calibration curves
* Measuring pressure using both gage and absolute scales

## Required Tools and Technology

|  |  |
| --- | --- |
| Platform: NI ELVIS III | * View User Manual   http://www.ni.com/en-us/support/model.ni-elvis-iii.html |
| Hardware: Quanser Mechatronic Sensors Board | * View User Manual   http://www.ni.com/en-us/support/model.quanser-mechatronic-sensors-board-for-ni-elvis-iii.html |
| Software: LabVIEW Version 18.0 or Later  Toolkits and Modules:   * LabVIEW Real-Time Module * NI ELVIS III Toolkit | * Before downloading and installing software, refer to your professor or lab manager for information on your lab’s software licenses and infrastructure * Download & Install for NI ELVIS III * <http://www.ni.com/academic/download> * View Tutorials * http://www.ni.com/academic/students/learn-labview/ |

## Expected Deliverables

In this lab, you will collect the following deliverables:

* Determine local atmospheric pressure adjusted for elevation
* Calculate and record pressure reference points using the syringe by applying Boyle's law
* Record calibration data (up-scale and down-scale directions)
* Screenshot of calibration graphs showing fitted curves
* Record calibration coefficients (up-scale and down-scale directions)
* Record maximum and minimum generated absolute and gage pressures
* Calculate sensor sensitivity in V/kPa
* Calculate theoretical capacitance for given sensor specifications
* Compare obtained sensor calibration with manufacturer's calibration

Your instructor may expect you to complete a lab report. Refer to your instructor for specific requirements or templates.

## Section 1: Pressure Measurement and Calibration

### 1.1 Theory and Background

#### What is a Pressure?

Pressure is the amount of force (F) acting per unit area (A) as defined in Equation 1-1:

Equation 1-1

Various units are used to express pressure. The SI unit of pressure is pascal (Pa), which is newtons per meter squared (N/m2). Other common units include pounds per square inch (psi), bar, and atmosphere (atm). Pressure at sea level under standard atmospheric conditions is defined as *patm* = 1 atm and equals 101.32 kPa.

Pressure is measured using two different scales: gage or absolute. Gage pressure (*pgage*) is pressure relative to local atmospheric pressure (*p0*), while absolute pressure (*pabs*) is pressure relative to perfect vacuum. The relationship between absolute and relative pressure is given in Equation 1-2 and illustrated in Figure 1-1.

Equation 1-2

Note that the value of *p0* depends on atmospheric and geographical conditions and may or may not be higher than standard atmospheric pressure.



Figure1-1: Comparison between different pressure scales

#### Boyle’s Law

Boyle’s law states that the product of volume and pressure of a confined gas is constant. It is mathematically expressed as:

Equation 1-3

where *p* is the pressure of the gas, *V* is the volume of the gas, and *k* is a constant. An alternative form of Boyle’s law is commonly used to compare a gas under two different conditions, and expressed as:

Equation 1-4

where *p1* and *V1* are the pressure and volume of the gas under condition 1, and *p2* and *V2* are the pressure and volume of the gas under condition 2. Figure 1-2 illustrates Boyle’s law using a pressure chamber. In this example, as the plunger is pushed into the chamber, the pressure of the gas increases from its initial state of *p1* to *p2*, while the volume of the gas decreases from *V1* to *V2*.

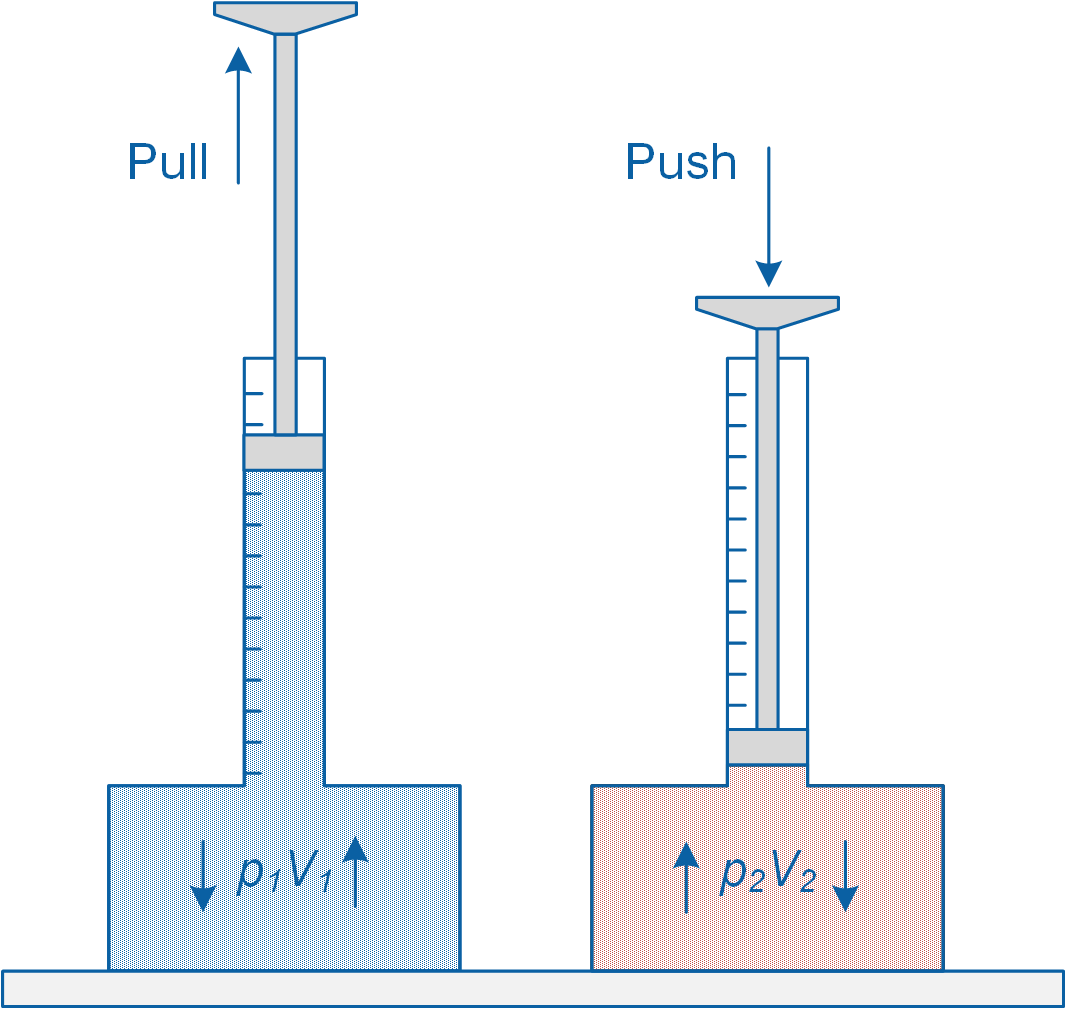
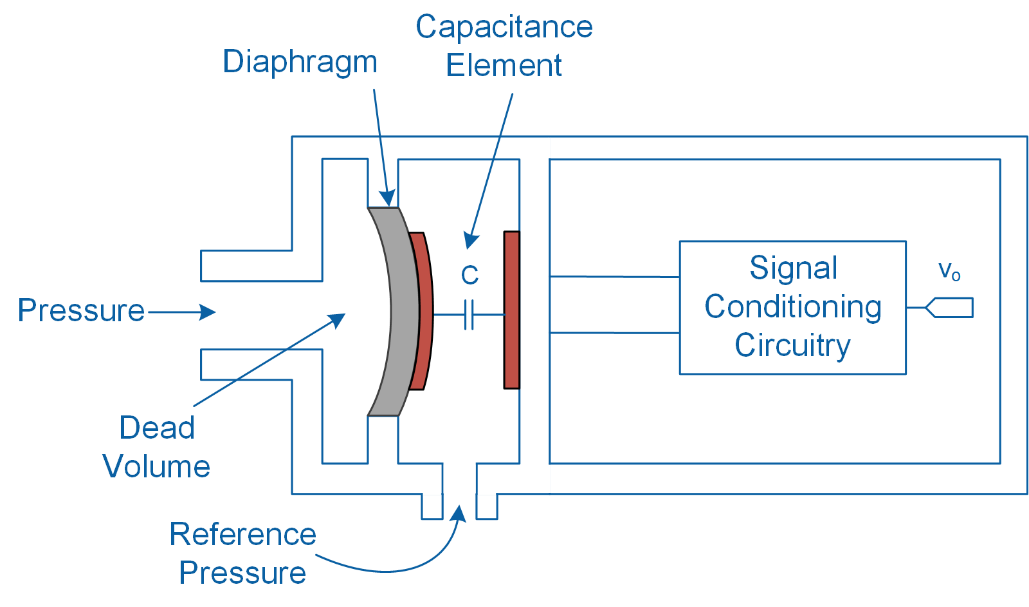


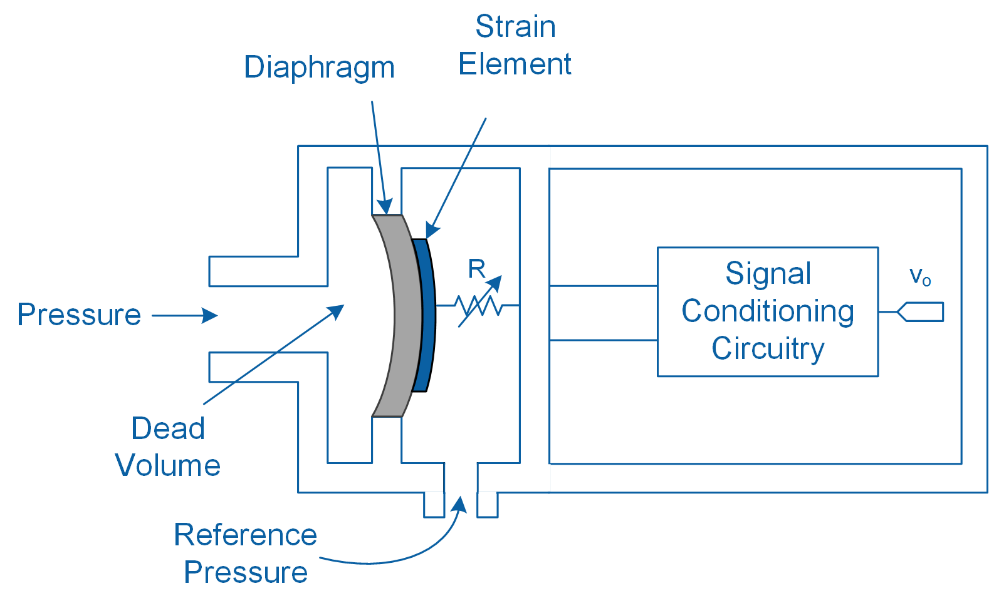
Figure1-2: Demonstration of Boyle’s law

#### Pressure Transducer

Pressure transducers are used in modern mechatronic applications to measure gage, absolute, and differential pressure. They are electromechanical devices that output an electrical signal proportional to the sensed pressure. A pressure transducer has an internal diaphragm or membrane that flexes due to an applied pressure.



(a) Capacitive type



(b) Resistive type

Figure1-3: Schematic illustration of two common types of pressure transducers

As schematically illustrated in Figure 1-3, a pressure transducer typically consists of either a strain/piezoresistive element or a parallel plate capacitor that is attached to a diaphragm. Depending on the type of sensing element used, any deformation in the diaphragm due to an applied pressure causes a change in the sensing element’s resistance or capacitance. Changes in the sensing element are then converted into a detectable electrical signal using signal conditioning circuitry.

Pressure transducers typically have a fast response behavior and allow for high precision measurements. However, because of their built-in electronics, they require a power source for operation.

As illustrated in Figure 1-3, a pressure transducer incorporates a built-in reference pressure port. When measuring gage pressure, the port is exposed to atmospheric pressure. However, when measuring absolute pressure, the port is sealed so that measurements are referenced against perfect vacuum. In differential pressure measurement the main and reference pressure ports are exposed to two different sources of pressure.

#### Pressure Transducer Calibration

A convenient way to convert the output of a pressure transducer to a unit of pressure is to use the manufacturer’s published sensitivity and offset values for the transducer. The sensitivity of a pressure transducer, which relates the output of the sensor to the measured pressure, is typically given in terms of volts per unit of pressure.

Alternatively, a sequential calibration test can be performed, which involves applying a sequential series of known pressures (known as *standards*) to the transducer and recording the corresponding output. The calibration can be done by increasing the input pressure (upscale direction), or by decreasing the input pressure (downscale direction).

The relationship between the input and output of the sensor is mathematically expressed in the form of a first order polynomial equation that is fitted to the recorded data. Using the calibration equation, unknown pressures can be determined for any given output signal.

Table 1-1: Results of a 5-point calibration

|  |  |
| --- | --- |
| Pressure (psi) | Output (V) |
| 0.0 | 0.100 |
| 5.0 | 1.255 |
| 10.0 | 2.550 |
| 15.0 | 3.505 |
| 20.0 | 4.950 |

As an example, Table 1-1 shows results of a 5-point calibration in the upscale direction. The sensors’ calibration equation is established by fitting a line to the points as shown in Figure 1-4.

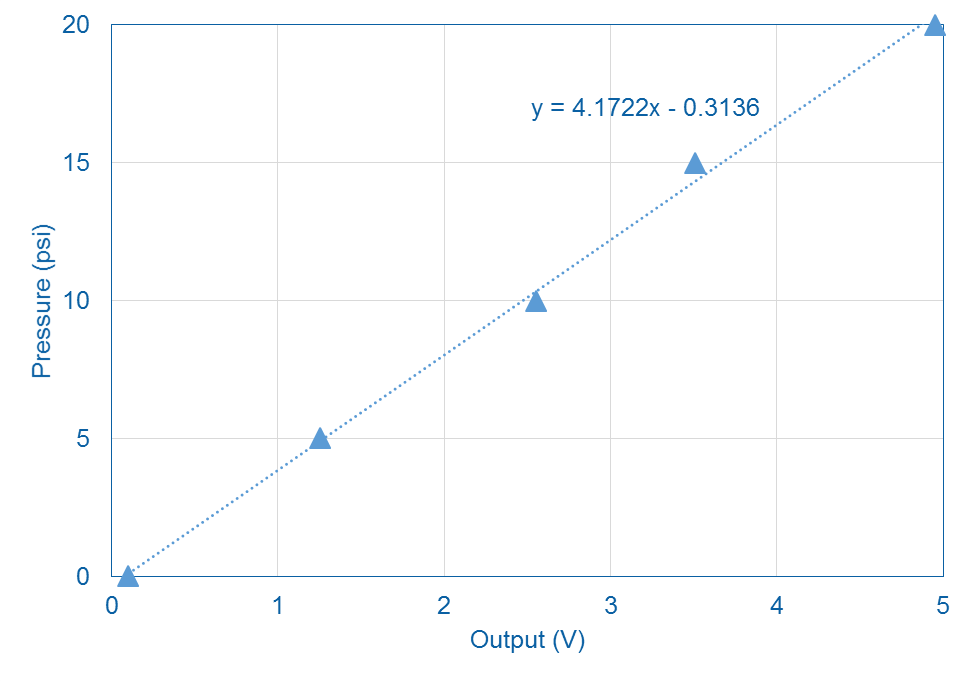


Figure 1-4: Line fitted to the recorded data

### 1.2 Implement

The Virtual Instrument (VI) used to collect data from and calibrate the pressure transducer is shown in Figure 1-5.

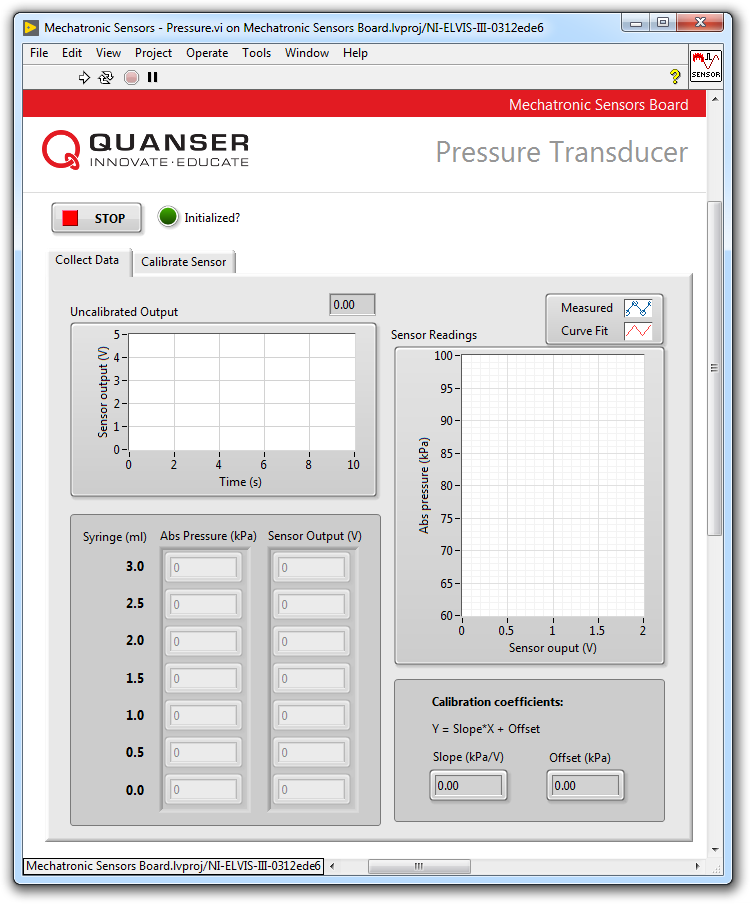


Figure1-5: VI for collecting data from the pressure transducer

#### Establish Reference Points

1. Determine your local atmospheric pressure (in kPa) using a barometer or look up the atmospheric pressure at your closest weather station (e.g. using www.weather.com). If you do not have access to this data, assume a local atmospheric pressure of 101.3 kPa.  
     
   *Note:* Weather stations typically report local atmospheric pressure corrected to sea-level conditions (0 m altitude and 15 °C ambient temperature). In such case, correct the reported sea-level pressure to your local atmospheric pressure by taking into consideration your altitude and ambient temperature using Equation 1-5:

Equation 1-5

where *plocal* is the local atmospheric pressure adjusted for temperature and altitude in kPa, *psea* is the sea-level pressure reported by the local weather station in kPa, *h* is your local altitude in meters, and *T* is ambient temperature in Celsius.

1. In the next section, you will be applying a series of known absolute pressures to the transducer using the syringe. Initially, you will start at atmospheric pressure with the syringe plunger aligned with the 3 ml mark. You will then increase the applied pressure by depressing the syringe plunger.  
     
   Using the adjusted local atmospheric pressure that you determined in the previous step and Equations 1-3 and 1-4, calculate the pressures that can be generated when the syringe plunger is positioned at the following volumetric markings: 3.0, 2.5, 2.0, 1.5, 1.0, 0.5, and 0.0 ml. Note that the pressure chamber has a dead volume of 5 ml. Record your calculations in Table 1-2.  
     
   *Note:* The dead volume refers to the volume of the pressure chamber. It excludes the volume of the attached syringe.
2. Continue to the next section.

Table1-2: Reference pressures using the syringe

|  |  |
| --- | --- |
| Syringe (ml) | Absolute Pressure (kPa) |
| 3.0 |  |
| 2.5 |  |
| 2.0 |  |
| 1.5 |  |
| 1.0 |  |
| 0.5 |  |
| 0.0 |  |

#### Collect Data

1. Open **Mechatronic Sensors Board.lvproj**
2. From the **Project Explorer** window, open **Mechatronic Sensors – Pressure.vi**
3. Click on the **Collect Data** tab.
4. Run the VI.
5. Wait for the **Initialized?** LED indicator to turn on.
6. If attached, unscrew the syringe from the pressure chamber.
7. Position the syringe plunger at the 3 ml mark.
8. If present, remove the protective cap from the pressure sensor chamber.
9. Attach the syringe to the Luer lock connector on the pressure sensor chamber by gently twisting the syringe in the clockwise direction. Ensure a secure connection is made. The applied pressure inside the chamber is now equal to atmospheric pressure.  
     
   *Note:* Attach the syringe such that the graduation markings are facing forward.
10. Enter the first value from Table 1-1into the **Abs Pressure (kPa)** array located in the VI front panel. Now using the **Uncalibrated Output** waveform chart, read the corresponding sensor output and enter the value in the **Sensor Output (V)** array.
11. Continue taking measurements by depressing the syringe plunger at 0.5 ml intervals. Enter the applied pressures (using the values from Table 1-1) and the corresponding sensor outputs in the **Abs Pressure (kPa)** and **Sensor Output (V)** arrays respectively.  
      
    *Note:* Once all of the measured readings have been entered, a linear curve is automatically generated to fit the data. The curve is shown in the **Sensor Readings** waveform graph. This curve represents the calibration curve of the sensor.
12. Record the collected data in Table 1-3.
13. The slope and offset of the calibration curve are automatically calculated by the VI and displayed in the **Slope (kPa/V)** and **Offset (kPa)** indicators. Make a note of these values in Table 1-4.
14. Take a screenshot of the **Sensor Readings** graph showing the fitted curve.
15. Continue to the next section.

Table1-3: Recorded calibration data

|  |  |  |
| --- | --- | --- |
| Syringe (ml) | Absolute Pressure (kPa) | Output (V) |
| 3.0 |  |  |
| 2.5 |  |  |
| 2.0 |  |  |
| 1.5 |  |  |
| 1.0 |  |  |
| 0.5 |  |  |
| 0.0 |  |  |

Table1-4: Calibration coefficients

|  |  |
| --- | --- |
| Slope (kPa/V) | Offset (kPa) |
|  |  |

#### Calibrate the Pressure Transducer

1. Click on the **Calibrate Sensor** tab to calibrate the output of the transducer in terms of pressure in kPa.
2. Use the **Atmospheric Pressure (kPa)** numeric control to enter the adjusted atmospheric pressure you determined in step 1.
3. Use the **Slope (kPa/V)** and **Offset (kPa)** numeric controls to enter the slope and offset values you obtained during the data collection steps.
4. Move the plunger to different positions and observe the displayed pressures in the **Calibrated Output** waveform chart as well as the **Absolute** and **Gage** gauge indicators. What is the difference between the absolute and gage readings?
5. What is the maximum absolute and gage pressures (in kPa) that you can generate?
6. Can you generate negative gage pressure using the syringe? What is the lowest absolute and gage pressure (in kPa) that you can generate?
7. Press the **Stop** button.

### 1.3 Analyze

1-1 What is the local atmospheric pressure (in kPa), adjusted for altitude and temperature, that you determined in step 1? Show your calculations.

1-2 Present the reference points that recorded in Table 1-2. Show sample calculations.

1-3 Present the calibration data that you recorded in Table 1-3.

1-4 Present the calibration coefficients that you recorded in Table 1-4.

1-5 Attach a screenshot of the *Sensor Readings* waveform graph showing the fitted calibration curve.

1-6 What calibration equation did you obtain?

1-7 How did you generate negative gage pressure using the syringe?

1-8 What is the maximum and minimum absolute as well as gage pressures (in kPa) that you were able to generate in steps 23 and 24?

1-9 What is the sensitivity of the sensor in V/kPa?

## Section 2: Design Considerations

2-1 In this lab you obtained a calibration curve for the pressure transducer by means of a sequential test where pressure was increased in the upscale direction. When designing a mechatronic system, the following question may arise: can one use the same calibration equation to infer pressures outside of the applied lower and upper pressure limits? Put differently, can one use the same calibration equation to extrapolate beyond the calibration range?  
  
Partially examine this question by repeating this lab but this time calibrating the transducer using negative gage pressures (downscale direction). How does the downscale calibration curve compare with the upscale curve originally obtained in this lab? Can the upscale curve be used to extrapolate negative pressures?

2-2 The Mechatronic Sensors Board uses an Infineon KP236N6165 pressure sensor. The manufacturer provides the following calibration equation relating the sensor’s output voltage (Vout) to the measured pressure (P):

where VDD = 5 V, a = 0.00876 1/kPa, and b = -0.48571. The calibration equation applies to an operating range of 60 to 165 kPa. How does the manufacturer’s calibration equation compare with the equation that you obtained during this lab? Measure atmospheric pressure using the calibration equation that you experimentally obtained as well as using the manufacturer’s calibration equation. How do they compare? Comment on potential sources of error. The manufacturer reports a sensor accuracy of +/-1.0 kPa when the temperature is between 0 and 85 degrees Celsius.

2-3 As noted in the Theory and Background section, a capacitive pressure sensor contains a parallel plate capacitor that is attached to the sensor’s diaphragm. When the sensor is subjected to different pressures, it causes the sensors’ diaphragm to flex, and as a result the capacitance of the parallel plate capacitor changes. The sensors’ capacitance, C, is determined by the following equation:

where *d* is the distance between the plates, *A* is the overlapping area of the two plates,  is the proportionality constant (= 8.85 x 10-15 F/mm when the area is measured in mm2 and distance is measured in mm), and *c* is the dielectric constant of the material between the capacitor plates.

Assume you are designing a new capacitive pressure transducer. Estimate the theoretical capacitance of the sensors if the plate area is A = 1 mm2, plate distance is d = 500 microns, and the dielectric material between the plates is air c = 1.

# Lab 6: Contact



Figure 0: Touch pads and keyboards are examples of commonly used contact sensors

This lab explores contact sensors including a snap action switch and a capacitive touch sensor. Mechanical switch debouncing, as well as scroll and single-button touch action using a capacitive touch sensor, are examined.

## Learning Objectives

After completing this lab, you should be able to complete the following activities:

1. Apply software-based debouncing to a snap action switch
2. Use capacitive touch sensing to simulate single-button touch and scroll actions

## Required Tools and Technology

|  |  |
| --- | --- |
| Platform: NI ELVIS III | * View User Manual   http://www.ni.com/en-us/support/model.ni-elvis-iii.html |
| Hardware: Quanser Mechatronic Sensors Board | * View User Manual   http://www.ni.com/en-us/support/model.quanser-mechatronic-sensors-board-for-ni-elvis-iii.html |
| Software: LabVIEW Version 18.0 or Later  Toolkits and Modules:   * LabVIEW Real-Time Module * NI ELVIS III Toolkit | * Before downloading and installing software, refer to your professor or lab manager for information on your lab’s software licenses and infrastructure * Download & Install for NI ELVIS III * <http://www.ni.com/academic/download> * View Tutorials * http://www.ni.com/academic/students/learn-labview/ |

## Expected Deliverables

In this lab, you will collect the following deliverables:

* Screenshots of switch debouncing for different debounce periods
* Screenshot of the response of the touchpad due to contact
* Estimate of the distances at which the virtual buttons indicate contact for different threshold values
* Pseudo code describing the virtual button counter implemented in the VI
* Pseudo code for determining scroll direction

Your instructor may expect you to complete a lab report. Refer to your instructor for specific requirements or templates.

## Section 1: Switch Debouncing

### 1.1 Theory and Background

#### Types of Switches

Switches are devices that create or break electrical contact by means of mechanical actuation, photo interruption, or magnetic actuation. Switches are found in most electronic applications. Depending on the type of switch used, it can create or break one or more electrical contacts. Switches can be classified into different types based on several factors such as method of actuation, type of operation, and construction. Figure 1-1 shows several common types of mechanical switches that are used in mechatronic applications. They include snap action, slide, rocker, pushbutton, and toggle switches. These switches typically contain internal metallic contacts and springs, which require physical actuation to operate.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| C:\Users\amolki\Desktop\QNET Mechatronic Sensors - Switch Debouncing - Concept Review.jpg(a) snap action | C:\Users\amolki\Desktop\QNET Mechatronic Sensors - Switch Debouncing - Concept Review.jpg(b) slide | C:\Users\amolki\Desktop\QNET Mechatronic Sensors - Switch Debouncing - Concept Review.jpg (c) rocker | C:\Users\amolki\Desktop\QNET Mechatronic Sensors - Switch Debouncing - Concept Review.jpg(d) push button | C:\Users\amolki\Desktop\QNET Mechatronic Sensors - Switch Debouncing - Concept Review.jpg (e) toggle |

Figure 1-1: Common types of mechanical switches (source: DigiKey)

As illustrated in Figure 1-2, switches can be classified based on their switching action. In this type of classification, pole refers to the number of circuits that can be controlled when the switch is actuated, and throw refers to the number of possible contacts for each of the poles.

|  |  |  |  |
| --- | --- | --- | --- |
| C:\Users\amolki\Desktop\1.png | C:\Users\amolki\Desktop\2.png | C:\Users\amolki\Desktop\3.png | C:\Users\amolki\Desktop\4.png |
| (a) single-pole, single-throw (SPST) | (b) single-pole, double-throw (SPDT) | (c) double-pole, single-throw (DPST) | (d) double-pole, double-throw (DPDT) |

Figure 1-2: Common switch configurations

Another type of classification of switches is whether they are normally open (NO) or normally closed, (NC). They refer to the state of the switch prior to being actuated. The contacts of a normally open switch are not engaged before being activated, e.g. a doorbell switch; while the contacts of a normally closed switch are engaged, e.g. an emergency stop switch.

Figure 1-3, schematically illustrates another popular type of mechatronic switch called a relay. A relay is an electromechanical switch which contains a coil. When the coil is energized it generates a magnetic force which causes an internal switch to latch. Relays are supplied in various configurations such as SPST, SPDT, and DPST.

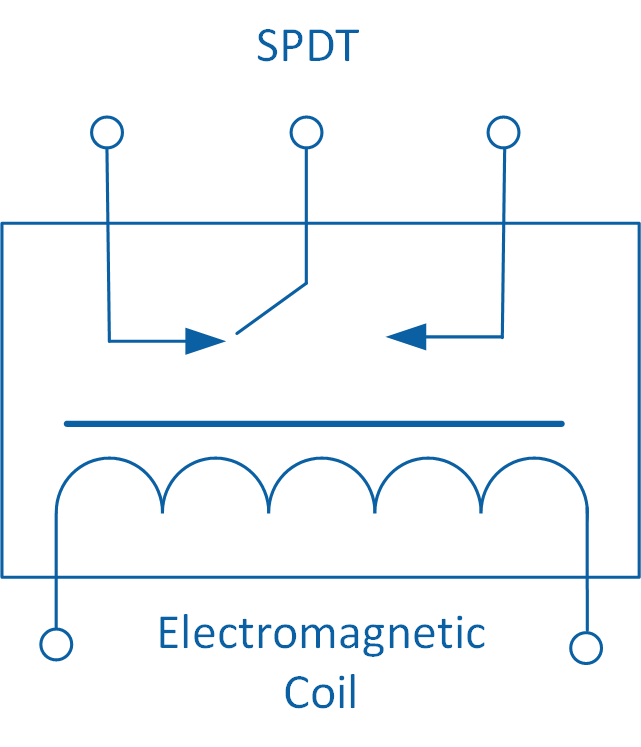


Figure 1-3: Schematic of an SPDT relay

Unlike mechanical and electromechanical switches, the optical switch does not contain moving components. They are classified as being either reflective or transmissive. As shown schematically in Figure 1-4, a reflective optical switch uses a reflective surface to bounce an Infrared LED beam onto its phototransistor. In contrast, in a transmissive optical switch, the Infrared LED shines directly onto the phototransistor. In both types of switches, once the Infrared light is interrupted, switching action is initiated which stops the flow of current.

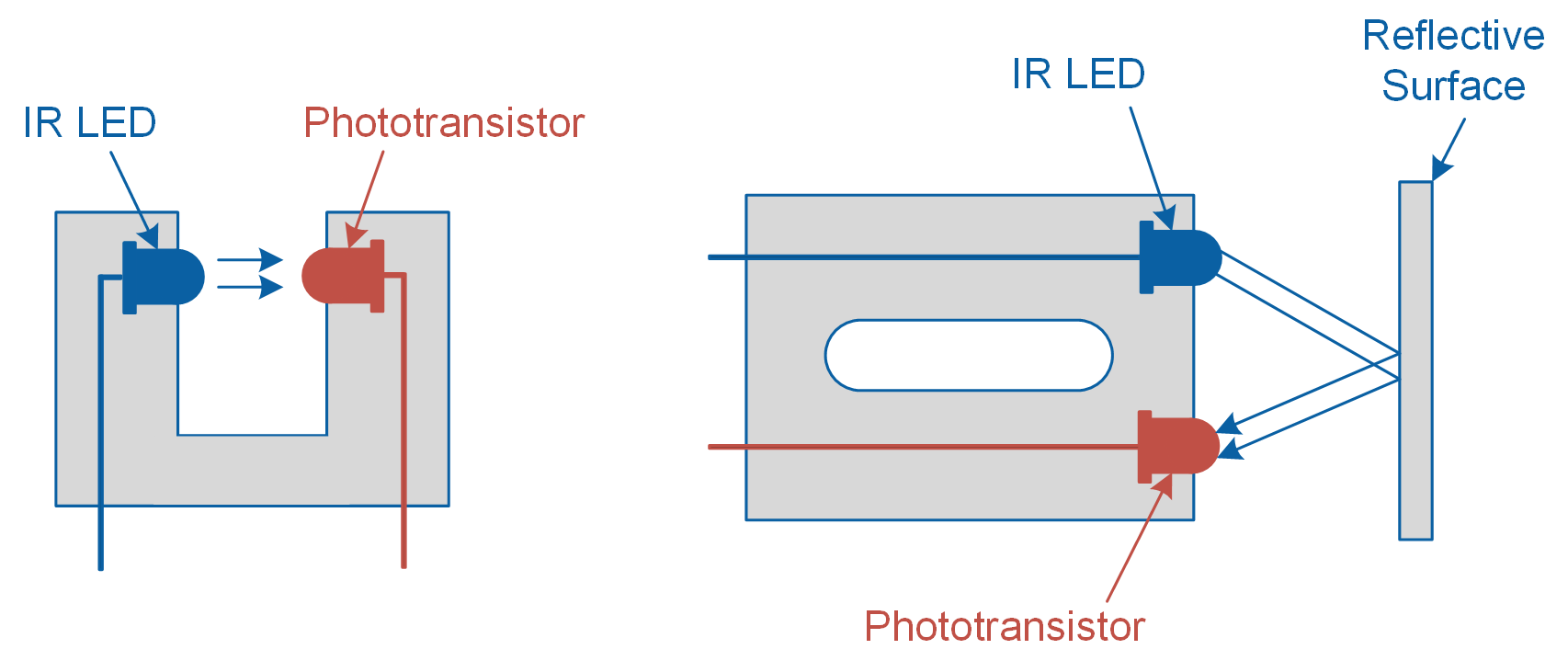


Figure 1-4: Schematic of a transmissive (left) and reflective (right) photo switch

#### Switch Bouncing

All switches suffer from a phenomenon called “switch bouncing”, which refers to the back and forth bouncing between ON and OFF states prior to settling on a final state. Bouncing in mechanical switches occurs because of the flexible components used inside the switch which physically bounce until a secure mechanical contact is made. Figure 1-5 shows the result of switch bouncing in a snap action switch. In this example, the switch is initially *high*. When the switch is depressed, the output bounces between *high* and *low* states for approximately 2,000 microseconds before it settles on a low state. Switch bouncing is an undesired characteristic, since a data acquisition system may think that you’ve actuated the switch button multiple times. Such spurious on/off actions may cause other processes to trigger unwantedly. To counter bouncing, a signal conditioning technique, called switch debouncing, is implemented.

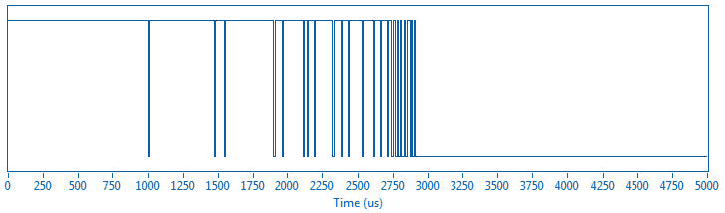


Figure 1-5: Switch bouncing when pressing a snap action switch

#### Debouncing Techniques

Switch debouncing can be implemented using both hardware and software. Figure 1-6 shows a digital debouncing circuit for a single-pole, double-throw switch. This circuit uses two cross-coupled NAND gates that create a simple Set-Reset (SR) latch. An alternative (analog) debouncing circuit is shown in Figure 1-7. This circuit, known as an RC debouncer, uses a capacitor and resistor to filter out rapid changes in the switch output.

|  |  |
| --- | --- |
| C:\Users\amolki\Desktop\2.png  Figure 6: SR debouncer | C:\Users\amolki\Desktop\1.png  Figure 7: RC debouncer |

Alternatively, debouncing can be done using software. A simple method involves using a timer to sample output of a switch and look for n sequential stable readings. If the switch signal consistently remains low (or high) during each of the sequential samples, then the switch is considered to be stable. If a bounce is detected in one of the samples, the timer is reset, and the sequential samples are retaken until n stable samples are read. This technique is illustrated in Figure 1-8, where the output of a switch is sampled at 10 ms intervals and when 2 consecutive stable reading is encountered the debounced switch turns from low to high.

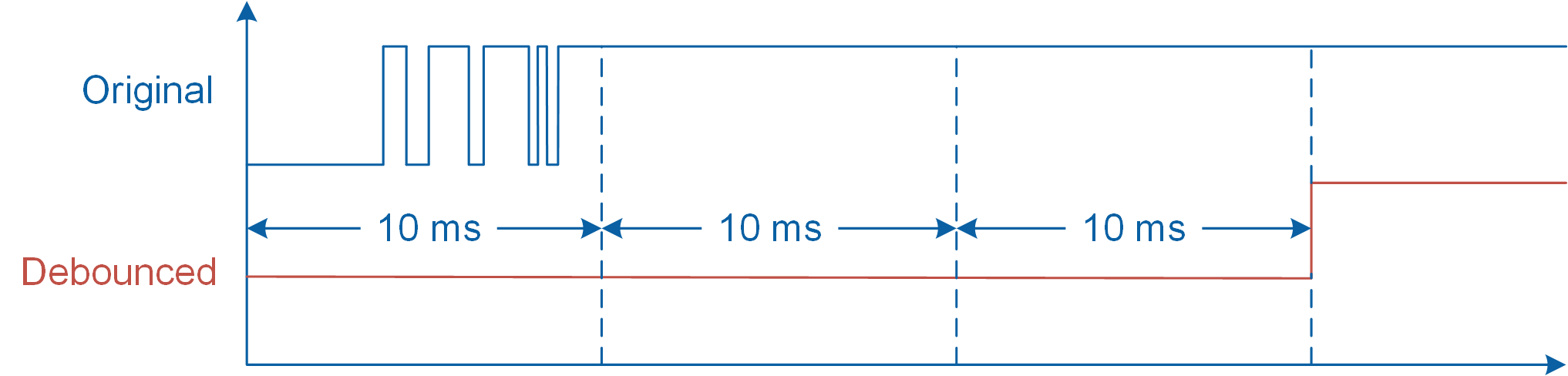


Figure 1-8: Software switch debouncing

### 1.2 Implement

The Virtual Instrument (VI) used to examine and debounce the snap action switch is shown in Figure 1-9.

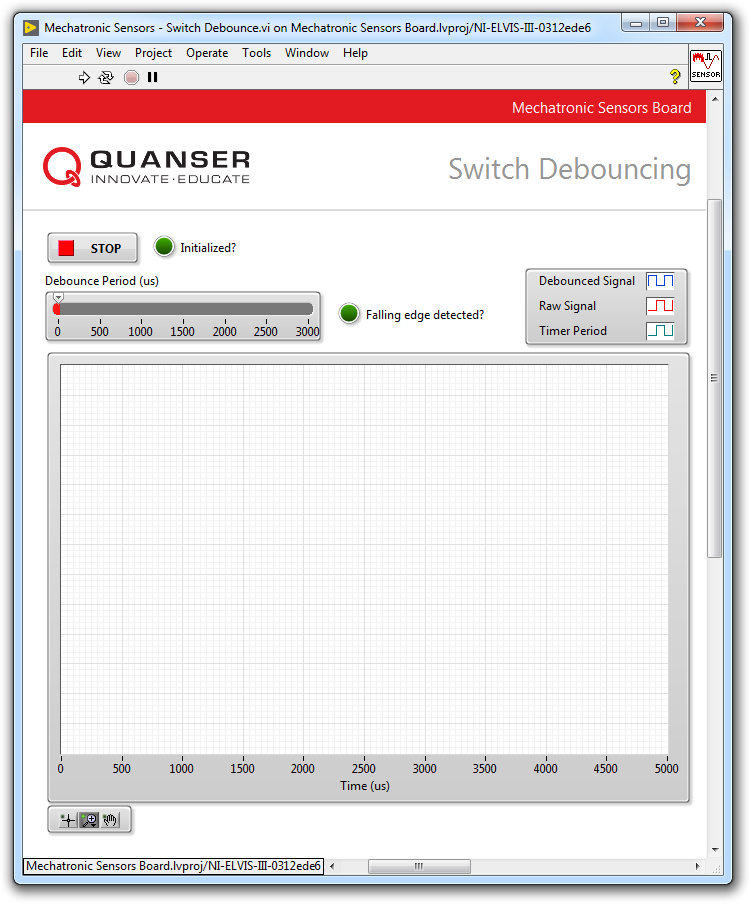


Figure 1-9: VI for examining and debouncing the snap action switch

#### Observe Switch Bouncing

1. Open **Mechatronic Sensors Board.lvproj**
2. From the **Project Explorer** window, open **Mechatronic Sensors – Switch Debounce.vi**
3. Run the VI.
4. Wait for the **Initialized?** LED indicator to turn on.
5. Click the snap action switch until the **Falling edge detected?** LED flashes momentarily. This indicates that a sample data containing a falling edge was successfully acquired. Once acquired, the results will be displayed in the waveform graph. Since by default the debounce period is set to 0 microseconds, the acquired switch data will not be debounced by the VI. Therefore, the graph will display the same results for **Raw Signal** and **Debounced Signal**.
6. Take screenshots of your results.
7. With the debounce period still set to 0 microseconds, press the snap action switch several times and observe switch bouncing. Is switch bouncing different each time you press the switch? Take screenshots of your results.
8. Continue to the next section.

#### Switch Debouncing

1. Set the **Debounce Period (us)** slider to **2000**.
2. Click the snap action switch until the **Falling edge detected?** LED flashes momentarily. Examine the response of the timer, raw signal, and the debounced signal. Take a screenshot of your results.  
     
   *Note:* The Mechatronic Sensors Board implements software debouncing. The VI implements a timer which has a period equal to the *Debounce Period (us)* slider value. When an edge change is detected in the raw signal, the software starts the timer and samples the raw signal again when the timer expires. If the measured raw signal value is different from the signal value prior to starting the timer (e.g. the signal value changed from high to low), the VI assumes that the switch has stopped bouncing and the debounced signal is generated. Otherwise, if the output of the switch value is still the same as prior to starting the timer (e.g. the signal value remains high), the timer is started again and the raw signal is sampled accordingly until the debounced signal is generated.
3. Set the **Debounce Period (us)** slider to **1000**, **500**, and **100**. Examine the response of the timer, raw signal, and the debounced signal. Take screenshots of your results.
4. Press the **Stop** button.

### 1.3 Analyze

1-1 Attach screenshots of the switch bouncing effect that you observed when the debounce period was set to 0 microseconds. Is the switch bouncing behavior different each time you pressed the switch?

1-2 Was the VI successful in debouncing the output of the switch when the debounce period was set to 2000 microseconds? Attach a screenshot of your results. Comment on your results.

1-3 Was the VI successful in debouncing the output of the switch when the debounce period was set to 1000, 500, and 100 microseconds? How do different debounce period values affect the debounced signal? Attach a screenshot of your results.

## Section 2: Capacitive Touch Sensor

### 2.1 Theory and Background

#### What is a Capacitive Touch Sensor?

A capacitance touch sensor measures changes in electrical capacitance to detect the presence of a finger that is either placed directly on the sensor or in its proximity. It is used in numerous applications ranging from consumer electronic touchpads to fluid-level monitoring systems in the automotive industry.

#### Self-Capacitance Sensing

A variety of methods are used to measure the capacitance between two points. A common method is called self-capacitance sensing. A capacitive touch sensor that implements this method, incorporates a single pin and measures the capacitance between that pin and a ground reference. This type of sensor is primarily used when single-touch (e.g. button) or sliding action is required.

Figure 2-1 illustrates the operating principle of a capacitive touch sensor that uses self-capacitance sensing. The sensor assembly typically consists of a conductive sensor pad, a ground hatch, and an overlay, all of which are mounted on a printed circuit board (PCB). The overlay is a protective dielectric layer that covers the ground hatch and sensor pad and prevents direct finger contact. Overlay material with a dielectric constant ranging between r = 2.0 and 8.0 are recommended, which include acrylic, PET film, and ceramic.

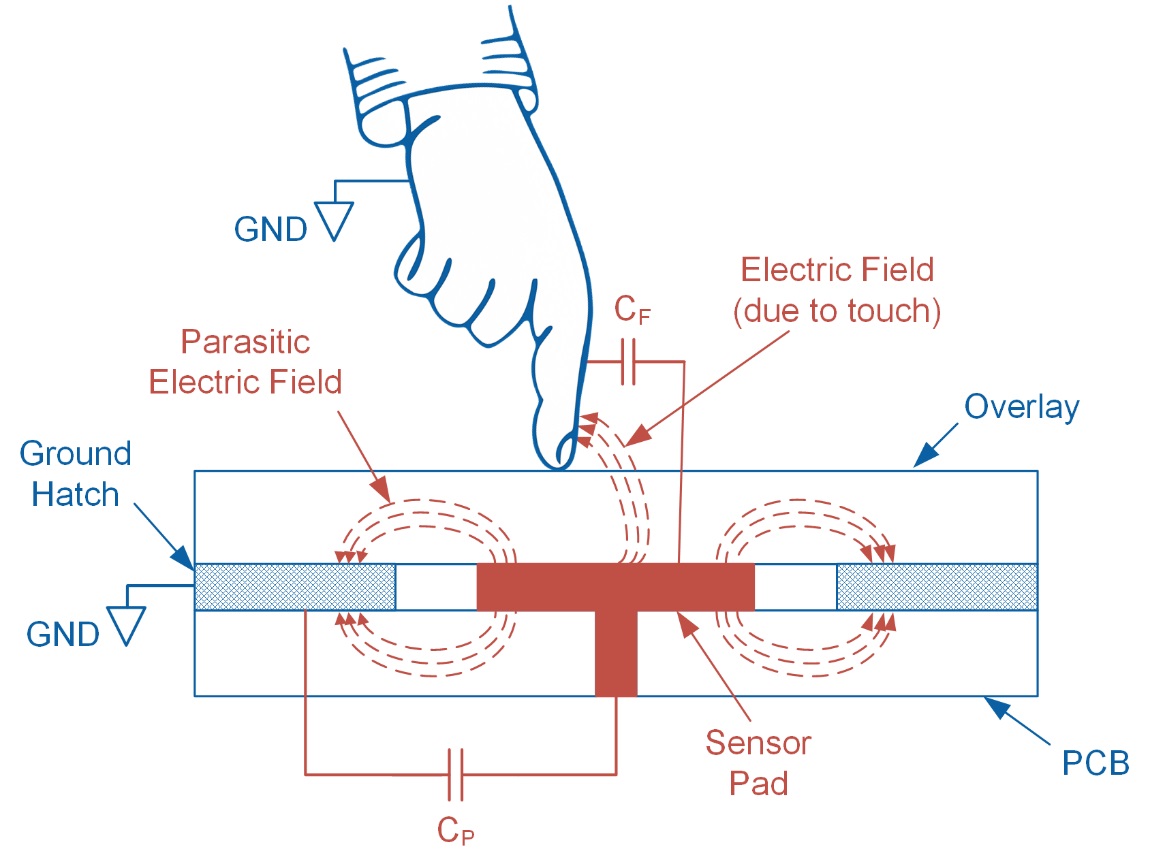


Figure 2-1: The operating principle of a self-capacitance touch sensor

In the absence of physical contact with the sensor, the parasitic electric field between the sensor pad and ground hatch results in the parasitic capacitance *CP*. This parasitic capacitance is a simplification of the distributed capacitance that includes the effects of the sensor pad, the overlay, and the PCB. In this case, the overall measured sensor capacitance, *CS*, is equal to *CP*. However, when a finger makes contact with the sensor pad, the two form an equivalent parallel plate capacitor. The resulting finger capacitance (*CF)* causes the overall sensor capacitance to become:

Equation 2-1

*CF* can be calculated using Equation 2-2:

Equation 2-2

where *0* is the free space permittivity, *r* is the dielectric constant of the overlay, *A* is the contact area between the finger and the sensor pad, and *D* is the overlay thickness.

#### Measuring Capacitance

The Quanser Mechatronic Sensors Board uses a Cypress CY8CMBR3116-LQXI capacitive sensor with an I2C interface. The sensor integrates a circular touch pad that is divided into ten sectors labeled **0** to **9,** as well as two individual buttons labeled **L** and **R**. The sensor converts capacitance into a digital count ranging between 0 and 255. An output of 0 indicates zero finger capacitance, while an output of 255 indicates the maximum measurable capacitance typically induced when a finger makes contact with the touchpad. The sensitivity of the sensor is set to 50 counts/0.1 pF.

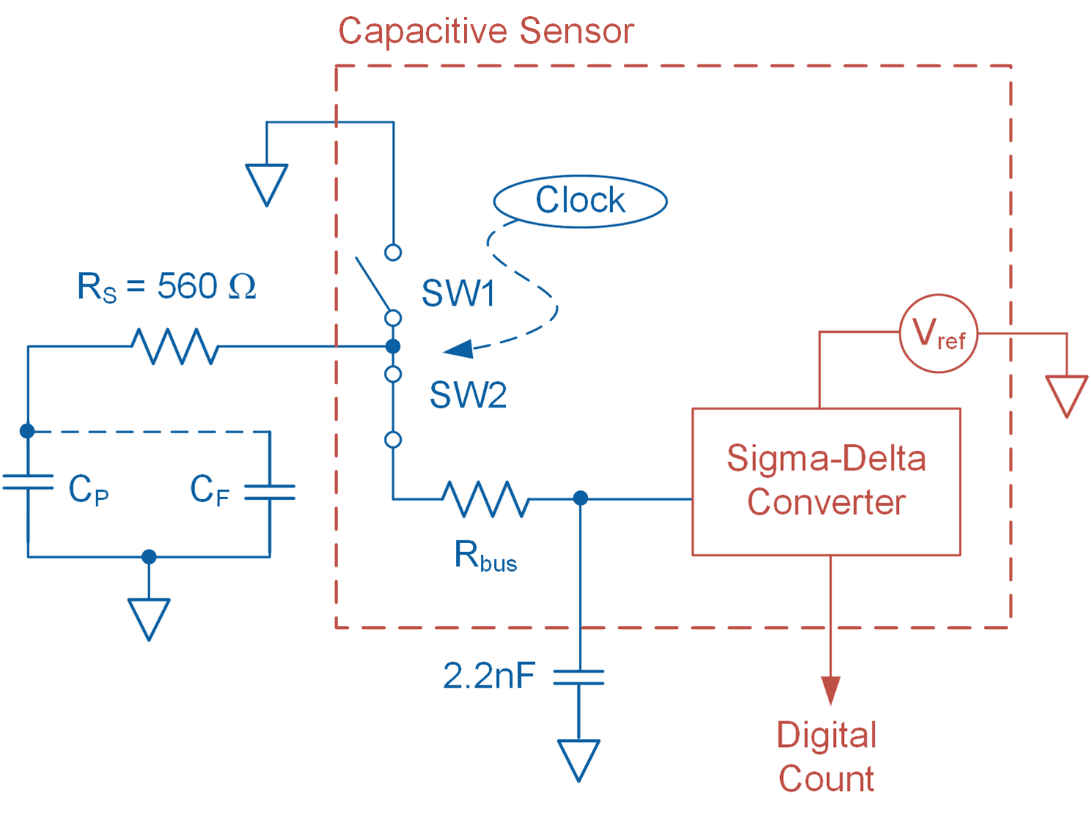


Figure 2-2: Schematic diagram of the capacitive sensor

A schematic diagram of the sensor and signal conditioning circuit is shown in Figure 2-2. The external resistor *RS* and internal resistor *Rbus* help reduce noise. Internally, a high-speed clock continuously switches *SW1* and *SW2*. This switching action results in the formation of an equivalent resistance (*Req*). The sensor then uses a Sigma-Delta modulator to convert the equivalent current (*ICS*) measured through *Req* into a digital count. Details of the equivalent resistor model are shown in Figure 2-3. In practice, when a finger is placed on the sensor pad, the presence of *CF* causes the value of *CS* to increase. This action causes *Req* to decrease, therefore increasing *ICS*, and resulting in a higher output count.

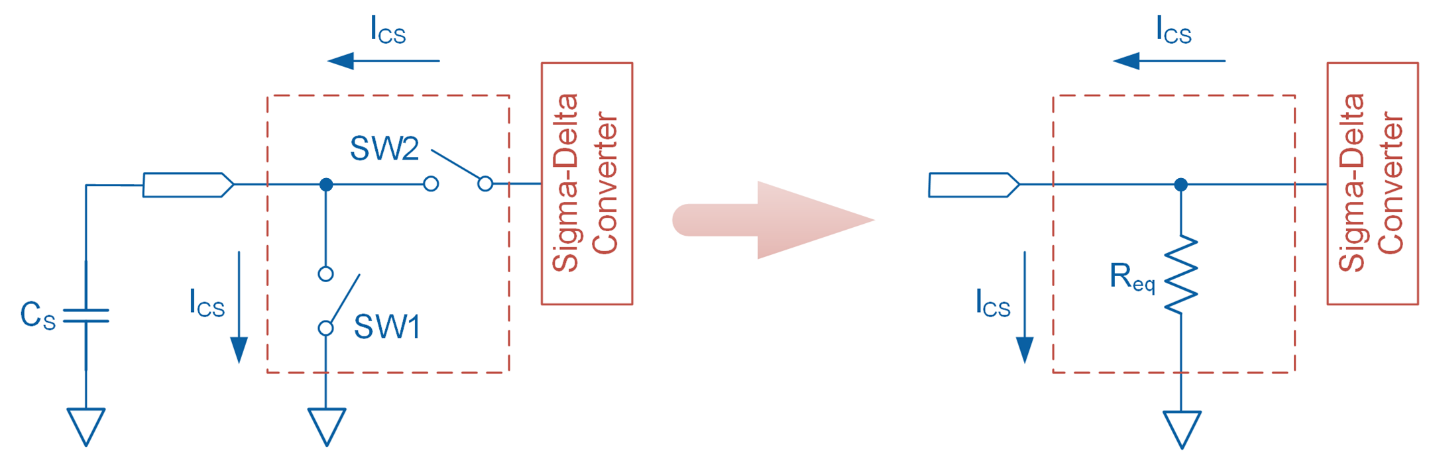


Figure 2-3: Equivalent resistor model

#### Advantages of Capacitive Touch Sensors

Capacitive touch sensors have several advantages over mechanical switches. Some of these advantages include:

* Since touch sensors lack moving components, they are less prone wear over time and therefore do not exhibit the type of mechanical failure that is common in mechanical switches.
* Most mechanical switches have cases that are not sealed, therefore they are more prone to environmental degradation due to moisture and dust build-up.
* Dust build-up over time may cause mechanical switches to become sticky, requiring a larger actuation force to operate.
* The absence of mechanically switching components means that capacitive touch sensors do not suffer from switch bouncing.

### 2.2 Implement

The Virtual Instrument (VI) used to examine the behavior of the capacitive touch sensor is shown in Figure 2-4.

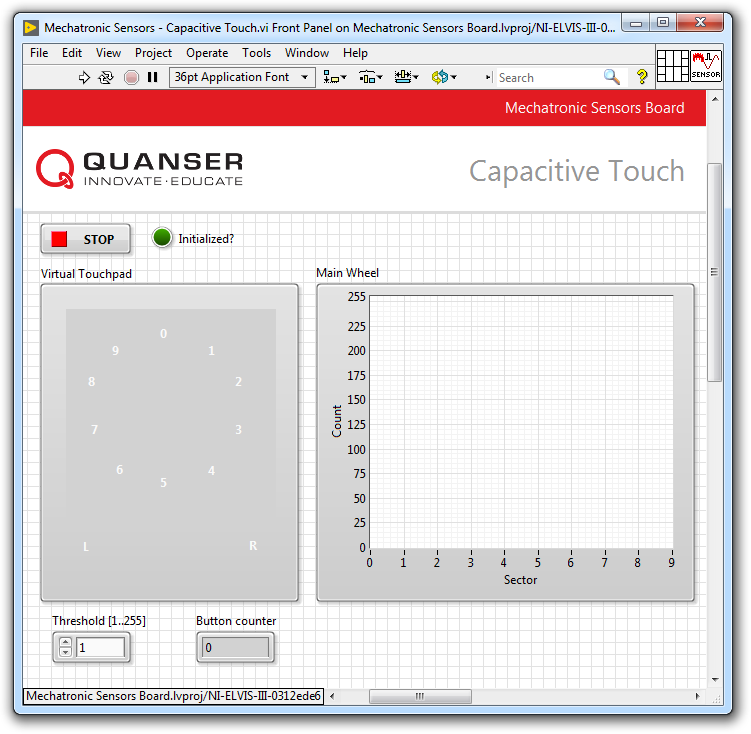


Figure 2-4: VI for examining the behavior of the capacitive touch sensor

#### Observe Sensor Behavior

1. Open **Mechatronic Sensors Board.lvproj**
2. From the **Project Explorer** window, open **Mechatronic Sensors – Capacitive Touch.vi**
3. Run the VI.
4. Wait for the **Initialized?** LED indicator to turn on.
5. The **Main Wheel** waveform graph displays the output of each of the ten sectors of the circular touch pad. As noted earlier, the sensor outputs a digital count that ranges between 0 and 255.  
     
   Slowly move your finger toward sector **0** and observe the response of the sensor using the **Main Wheel** waveform graph. The sensor will initially output a count value of 0, but will gradually increase to a value of 255 as your finger almost makes full contact with the touch pad. Take screenshots of your results.
6. Use the **Threshold [1..255]** numeric control to control the behavior of the virtual touch pad in the VI. The VI is programmed to compare the output of the capacitive touch sensor with the threshold: if the output of the sensor is equal to or greater than the threshold, corresponding individual sectors and touch buttons turn **red** indicating contact. Otherwise, they remain **black** indicating no contact.  
     
   With the threshold set to its default value of 1, slowly move one finger toward the **right** touch button and examine the behavior of the **virtual** touch button. Estimate the physical distance at which the virtual button turns **red**.
7. Change the threshold value to 255 and again estimate the physical distance at which the virtual button turns **red**.
8. Press the **Stop** button.

### Analyze

2-1 Describe the behavior of the sensor as you moved your finger close to the touch pad in Step 5. Attach screenshots of your results.

2-2 With the threshold set to 1 and 255 respectively, at what physical distances did the virtual button turn red (Step 6 and 7)? Describe the effect of the threshold setting on the behavior of the virtual touch pad. Is a higher threshold value desirable?

## Section 3: Design Considerations

3-1 The capacitance sensor implemented in the Quanser Mechatronic Sensors Board can be used to virtually perform single-button and scroll actions. To show this, the VI used in this lab implements a button counter. Each time the right button is tapped, the **Button counter** numeric indicator is incremented by a value of 1 and decremented by the same value if the left button is tapped. Run the VI and examine this functionality.

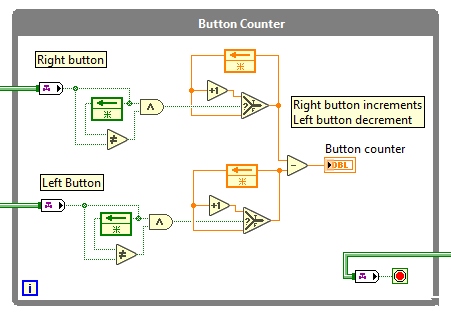


Figure 3-1: LabVIEW code that performs button counting

A screenshot of the block diagram that performs the counting is shown in Figure 3-1. Wire channels are used to pass Boolean values representing the status of the right and left buttons to the While Loop (a value of 0 indicates no contact, and a value of 1 indicates contact). Present the pseudo-code equivalent of this LabVIEW code.

3-2 Present strategies on how you would implement a LabVIEW VI that determines the direction of scrolling based on the output of the touchpad’s individual sectors.

# Lab 7: Inertial Measurement



Figure 0: Precise attitude measurement is instrumental to the control of unmanned aerial vehicles

This lab explores acceleration, rotation, and magnetic field measurements using an Inertial Measurement Unit (IMU) sensor. In particular, roll, pitch, and yaw measurements will be determined using the output of the accelerometer and gyroscope sensors. Also, earth’s magnetic field direction will be approximated using the magnetometer.

## Learning Objectives

After completing this lab, you should be able to complete the following activities:

1. Identify components of an Inertial Measurement Unit
2. Estimate roll and pitch using an accelerometer
3. Estimate yaw by integrating the output of a gyroscope
4. Analyze gyroscopic drift
5. Approximate earth’s magnetic field direction using a magnetometer
6. Determine magnetometer offset
7. Observe Euler geometric singularities when estimating roll and pitch

## Required Tools and Technology

|  |  |
| --- | --- |
| Platform: NI ELVIS III | * View User Manual   http://www.ni.com/en-us/support/model.ni-elvis-iii.html |
| Hardware: Quanser Mechatronic Sensors Board | * View User Manual   http://www.ni.com/en-us/support/model.quanser-mechatronic-sensors-board-for-ni-elvis-iii.html |
| Software: LabVIEW Version 18.0 or Later  Toolkits and Modules:   * LabVIEW Real-Time Module * NI ELVIS III Toolkit | * Before downloading and installing software, refer to your professor or lab manager for information on your lab’s software licenses and infrastructure * Download & Install for NI ELVIS III * <http://www.ni.com/academic/download> * View Tutorials * http://www.ni.com/academic/students/learn-labview/ |

## Expected Deliverables

In this lab, you will collect the following deliverables:

* Record raw accelerometer data
* Examine and calculate roll and pitch angles using accelerometer output
* Examine yaw angle polarity
* Measure maximum and minimum magnetometer outputs
* Determine magnetic North
* Estimate gyro drift
* Estimate magnetometer offset
* Observe and estimate Euler singularities when determining roll and pitch

Your instructor may expect you to complete a lab report. Refer to your instructor for specific requirements or templates.

## Section 1: Measuring Pose and Magnetic Field using an IMU

### 1.1 Theory and Background

#### What is an IMU?

An internal measurement unit, or IMU, is a sensor that measures a body’s acceleration, angular velocity, as well as magnetic field. Using an IMU, it is possible to determine the attitude of a body. An attitude of a body comprises of its pitch, roll, and yaw. Figure 1-1 illustrates that attitude of an aircraft. In this example, pitch is the degree of tilt in the aircraft’s nose (rotation about the pitch axis), roll is the degree of rotation about the longitudinal axis of the aircraft’s fuselage (rotation about the roll axis), and yaw refers to the aircraft’s heading (rotation about the yaw axis). It should be noted that different applications may use different conventions for the orientation of the axes and polarity of rotation. IMUs are used also used in a wide range consumer goods such as smartphones, tablets, motion-based game controllers, and many wearable technologies.

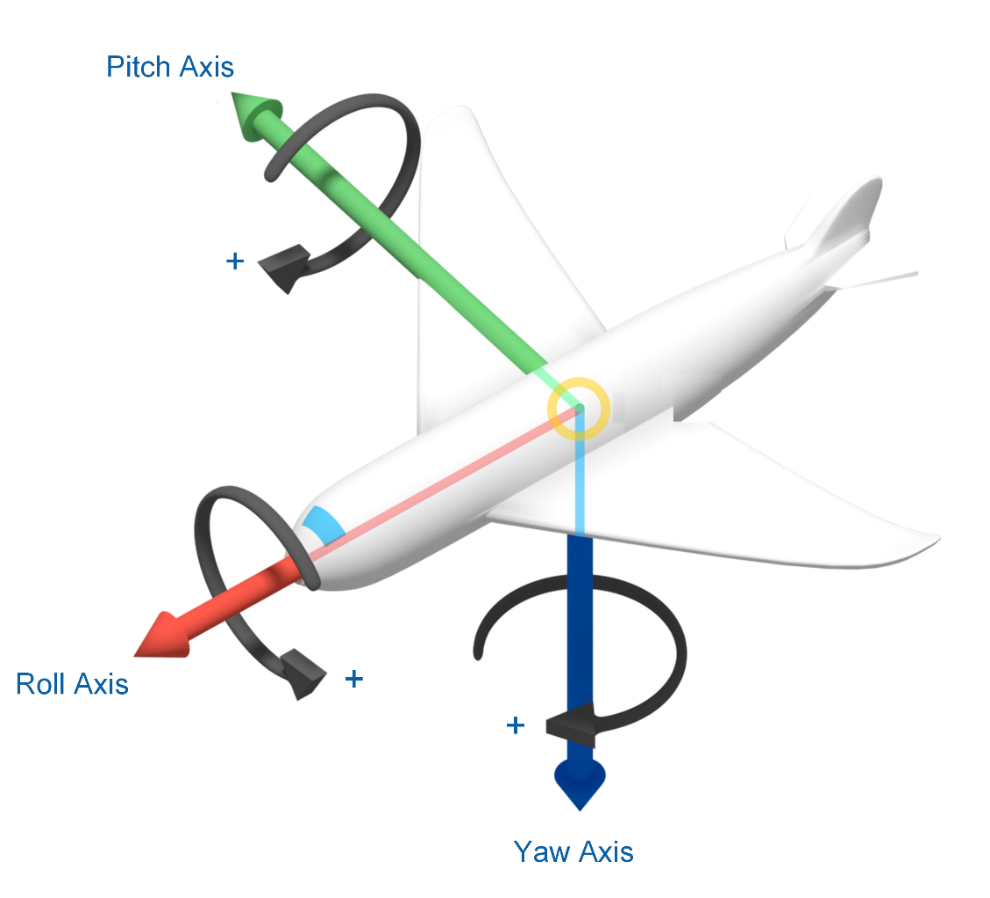


Figure 1-1: An aircraft’s attitude (roll, pitch, and yaw)

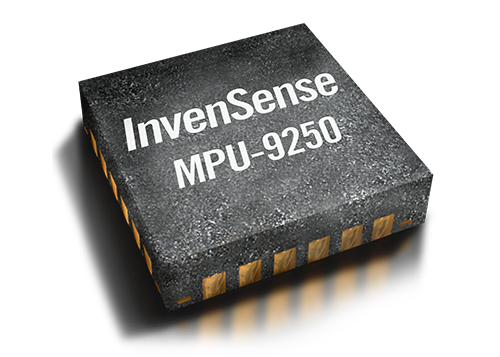


Figure 1-2: Image of a TDK InvenSense MPU-9250

The Quanser Mechatronic Sensors Application Board incorporates a 9-axis TDK InvenSense MPU-9250 IMU sensor (Figure 1-2). The sensor uses three independent vibratory MEMS rate gyroscopes to detect motion about the X-, Y-, and Z-axes. When the gyros are rotated about any of the sensor’s axes, the Coriolis Effect causes a vibration that is detected by a capacitive pickoff. The resulting signal is amplified, demodulated, and filtered to produce a voltage that is proportional to the angular rate. The sensor includes an Analog-to-Digital converter to output digital values of the measured angular velocities up to 2,000 deg/sec.

The sensor also incorporates a 3-axis accelerometer that measures acceleration in units of *g*. The accelerometers use separate proof masses for each axis. Acceleration along a particular axis induces displacement on the corresponding proof mass, and capacitive sensors detect the displacement differentially. Figure 1-3 illustrates the orientation of axes of sensitivity and polarity of rotation for the accelerometer and gyroscope. The accelerometer outputs a value of +1 g for any axis that is vertically pointing upward, and outputs a value of -1 g when the axis is vertically pointing downward. For any arbitrary inclination, the accelerometer outputs the magnitude of the X, Y, and Z acceleration vectors. For example, when the sensor is placed on a flat horizontal surface with its Z-axis facing upward, the sensor outputs AX = 0 g, AY = 0 g, and AZ= +1 g. Similarly, when the sensor’s X-axis is facing vertically downward, the sensor outputs AX = -1 g, AY = 0 g, and AZ = 0 g.

The sensors also include a 3-axis magnetometer capable of measuring up to ±4,800 T. Communication with the sensor is done using the I2C protocol. Figure 1-4 illustrates the orientation of axes of sensitivity for the magnetometer.

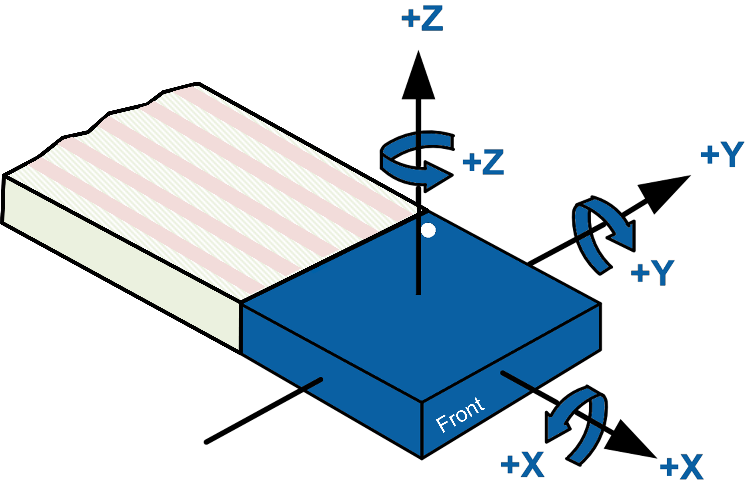


Figure 1-3: Orientation of axes of sensitivity and polarity of rotation for the accelerometer and gyroscope

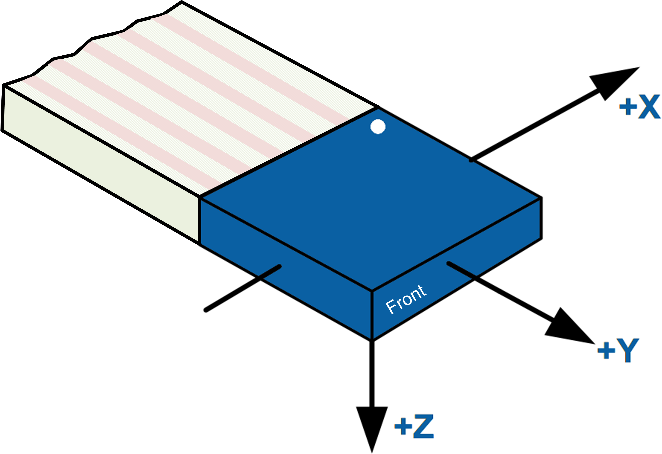


Figure 1-4: Orientation of axes of sensitivity for the magnetometer

#### Estimating Pitch and Roll using an Accelerometer

Measurements from a 3-axis accelerometer can be used to determine the pitch and roll of a body. Figure 1-5 illustrates how to determine the roll angle (ϕ) using the output of an accelerometer and basic trigonometric functions. As shown in the figure, when the accelerometer is held horizontally with its Z-axis pointing upward, the accelerometer outputs a value of AZ = +1 g. In this orientation, the magnitude of the X and Y gravitational acceleration vectors equal AX = AY = 0 g, resulting in a summed magnitude of Asum = +1 g since the sensor is always subjected to gravitational force.

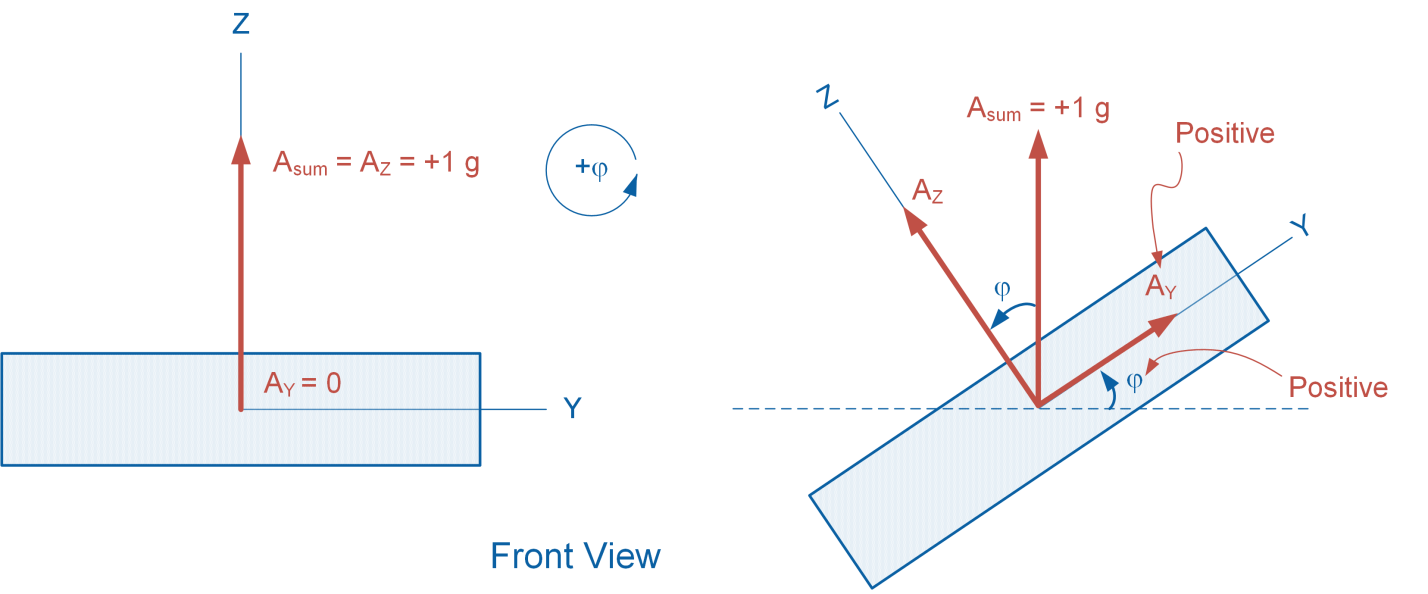


Figure 1-5: Determining roll angle using accelerometer output

However, when the accelerometer is rolled about its X-axis by angle ϕ, the magnitudes of AY and AZ will vary depending on the roll angle, while the magnitude of AX remains 0. Equation 1-1 shows the relationship between the roll angle and AY and AZ acceleration components:

Equation 1-1

*Note:* The *atan2* function computes the arctangent for angles in any of the four quadrants of a 2-dimensional plane, whereas the *atan* function computes the arctangent in only two quadrants. It is defined as follows:

Figure 1-6 illustrates how to determine the pitch (ϕ) angle using the output of an accelerometer and basic trigonometric functions. Similar to the previous example, when the accelerometer is held horizontally with its Z-axis facing upward, the accelerometer outputs a value of AZ = +1 g. In this orientation, the magnitude of the X and Y gravitational acceleration vectors equal AX = AY = 0 g, resulting in a summed magnitude of Asum = +1 g since the sensor is only subjected to gravitational force.

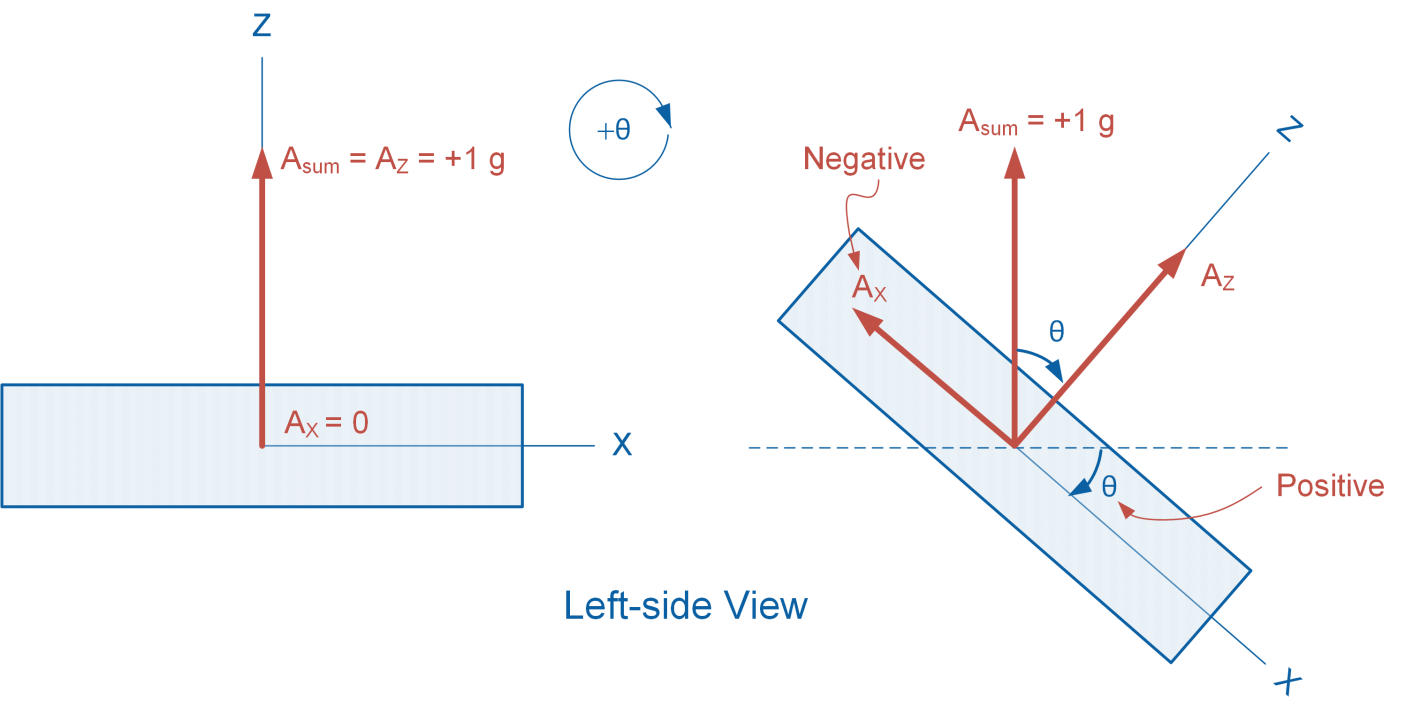


Figure 1-6: Determining pitch angle using accelerometer output

However, when the accelerometer is pitched about its Y-axis by angle θ, the magnitudes of AX and AZ will vary depending on the pitch angle, while the magnitude of AY remains 0. Equation 1-2 shows the relationship between the pitch angle and AX and AZ acceleration components:

Equation 1-2

#### Estimating Yaw using a Gyroscope

Unlike roll and pitch, yaw cannot be deduced from accelerometer measurements. This is because as a body yaws about the Z-axis, an accelerometer does not register changes in AX, AY, AZ. To counter this problem, the yaw of a body can be estimated using a gyroscope. A gyroscope measures the rate of angular change, or angular velocity. As such, in order to determine angular position, or yaw, one must integrate angular velocity about the Z axis with respect to time as shown in Equation 1-3:

Equation 1-3

#### Determining Earth’s Magnetic Heading using a Magnetometer

A magnetometer is a sensor that measures the strength of a magnetic field. The 3-axis magnetometer incorporated in the application board uses highly sensitive Hall sensor technology for detecting terrestrial magnetism along the X-, Y-, and Z-axes. With the advent of MEMS technology, magnetometers have been miniaturized to the extent that they can be incorporated in integrated circuits as micro-size digital compasses. Magnetometers are widely used in military and aerospace applications for positioning and establishing directional references.

It is important to note the existence of a discrepancy between Earth's magnetic north (the direction the north end of a compass needle points, corresponding to the direction of the Earth's magnetic field lines) and Earth's true north (the direction along a meridian towards the geographic North Pole). This discrepancy, called magnetic declination, is defined as the angle on the horizontal plane between magnetic north and true North. Magnetic declination varies both with time and geographical location.

Earth’s magnetic heading can be estimated by using the HX or HY components of a magnetometer output. This can be done by holding the magnetometer horizontally such that its Z-axis is pointed vertically downward. While monitoring the HX output, rotate the sensor on the horizontal plane. When the sensor outputs the maximum value of HX, the X-axis is pointing toward magnetic north.

### 1.2 Implement

The Virtual Instrument (VI) used to examine the behavior of the capacitive touch sensor is shown in Figure 1-7.

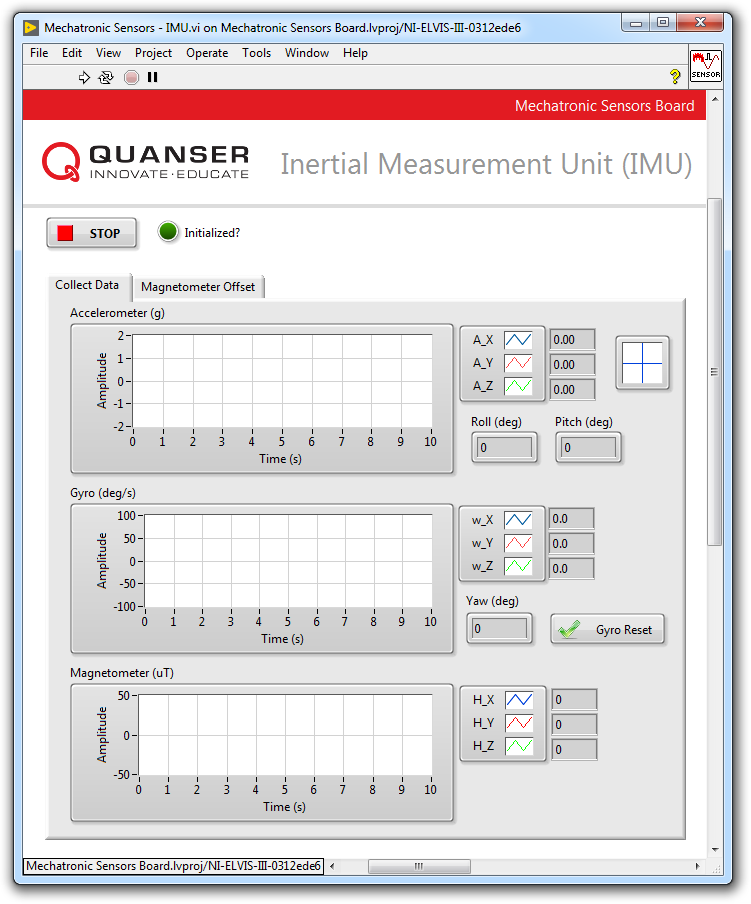


Figure 1-7: VI used for measuring the output of the IMU

#### Raw Accelerometer Output

1. Open **Mechatronic Sensors Board.lvproj**
2. From the **Project Explorer** window, open **Mechatronic Sensors – Pressure.vi**
3. Click on the **Collect Data** tab.
4. Run the VI.
5. Wait for the **Initialized?** LED indicator to turn on.
6. The direction of the axes of the accelerometer is silk screened on the IMU sensor. Hold the IMU sensor such that the accelerometer X-axis is pointing vertically upward. Record the value displayed in the **A\_X** numeric indicator in Table 1-1.
7. Hold the IMU sensor such that the accelerometer X-axis is pointing vertically downward. Record the value displayed in the **A\_X** numeric indicator in Table 1-1.
8. Repeat steps 7 and 8 for the accelerometer Y and Z axes and record your results in Table 1-1.

Table1-1: Raw accelerometer output

|  |  |
| --- | --- |
| Accelerometer Axis Direction | Raw Output (g) |
| X upward |  |
| X downward |  |
| Y upward |  |
| Y downward |  |
| Z upward |  |
| Z downward |  |

#### Examine Roll Angle

1. The VI implements Equation 1-1 to estimate the roll of the sensor. Hold the sensor in the horizontal plane with the accelerometer Z-axis pointing vertically upward. The **Roll (deg)** numeric indicator must output a value of **0**. Slowly roll the sensor sideways (about the accelerometer’s X-axis) and examine changes in the **Roll (deg)** numeric indicator. Does the indicated roll angle polarity match the diagram shown in Figure 1-3?
2. Holding the sensor at a +45 degree roll angle. Make note of the **A\_Y** and **A\_Z** accelerometer outputs and use Equation 1-1 to estimate the roll angle. Does your calculation agree with the roll angle displayed in the VI?

#### Examine Pitch Angle

1. The VI implements Equation 1-2 to estimate the pitch of the sensor. Hold the sensor in the horizontal plane with the accelerometer Z-axis pointing vertically upward. The **Pitch (deg)** numeric indicator must output a value of **0**. Slowly pitch the sensor downward (about the accelerometer’s Y-axis) and examine changes in the **Pitch (deg)** numeric indicator. Does the indicated pitch angle polarity match the diagram shown in Figure 1-3?
2. Pitch the sensor downward holding it at a +45 degree angle. Make note of the **A\_X** and **A\_Z** accelerometer outputs and use Equation 1-2 to estimate the pitch angle. Does your calculation agree with the pitch angle displayed in the VI?

#### Examine Yaw Angle

1. The VI estimates the yaw angle by integrating the Z-axis component of the gyro output. Hold the sensor in the horizontal plane and click the **Gyro Reset** button to zero the yaw angle.
2. While keeping the sensor level in the horizontal plane, slowly yaw it sideways. Examine changes in the **Yaw (deg)** numeric indicator. Does the indicated yaw angle polarity match the diagram shown in Figure 1-3?

#### Find Magnetic North

1. Determine magnetic North by holding the sensor in the horizontal plane with the magnetometer’s Z-axis pointing downward. Slowly rotate the sensor about the Z-axis and examine the **H\_Y** numeric indicator. Determine the maximum and minimum values indicated by the **H\_Y** numeric indicator. The direction of the Y-axis at which maximum **H\_Y** value occurs points toward magnetic North.
2. Validate your finding with a device that indicates magnetic North (e.g. a compass).
3. Stop the VI.

### 1.3 Analyze

1-1 Present the raw accelerometer data you recorded in Table 1-1.

1-2 When examining roll angles in Step 9, did the roll angle polarity match the diagram shown in Figure 1-3?

1-3 What are the A\_Y and A\_Z accelerometer outputs noted in Step 10? Using these values and Equation 1-1, calculate the sensor's roll angle. Show your calculations. Did your calculation agree with the roll angle displayed in the VI?

1-4 When examining pitch angles in Step 11, did the polarity match the diagram shown in Figure 1-3?

1-5 What are the A\_X and A\_Z accelerometer outputs noted in Step 12? Using these values and Equation 1-2, calculate the sensor's pitch angle. Show your calculations. Did your calculation agree with the pitch angle displayed in the VI?

1-6 When examining yaw angles in Step 14, did the polarity match the diagram shown in Figure 1-3?

1-7 What are the maximum and minimum H\_Y values recorded in Step 16? Did you successfully validate your magnetic North finding?

## Section 2: Design Considerations

2-1 As explained earlier in Section 1.1, angular position, or yaw, can be deduced by integrating the output of the gyroscope about the Z axis with respect to time. This process, however, leads to a common problem known as gyroscopic drift. Gyroscopic drift is unavoidable. Even expensive high-end models will have significant drift. The drift occurs because the process of integration not only integrates angular rate, but also integrates any bias and noise that is present in the sensor. Gyroscopic drift is more evident when the sensor is at rest.

Your task is to observe and quantify the drift present in the Sensor board’s gyroscope. To do this, run the VI, hold the sensor in the horizontal plane with the accelerometer’s Z axis pointing in the upward direction. Click the **Gyro Reset** button to zero the output of the **Yaw (deg)** numeric indicator. Using a stopwatch, estimate the average drift that you observe in terms of degrees per minute.

2-2 A sensor that outputs a fixed value when it should be measuring zero is said to exhibit a *zero offset error*. Magnetometers often exhibit zero offset errors along their axes of measurement. Two factors contribute to the error: hard-iron effects and soft-iron effects. Hard iron interferences are due to the sensor’s internal construction, which causes a constant additive value to the measured in addition to earth’s magnetic field. Soft iron interferences are caused by environmental conditions, e.g. the presence of nearby material which influences the measured magnetic field.

In practice, zero offset errors must be determined and corrected for when conducting an experiment. Your task is to quantify the zero offset error present in the magnetometer along its Y and Z axes. To do this, run the VI, click the **Magnetometer Offset** tab, and click the **Plot** button. Gently move the sensor within the magnetometer’s Y-Z plane. As you move the sensor, the waveform graph will plot the Y and Z components of the magnetometer. Once sufficient data has been plotted, you should expect to obtain a circular or elliptical pattern as shown in Figure 2-1. Identify the center of the ellipse and from there estimate the zero offset errors along the Y and Z axes. Take a screenshot of your results.

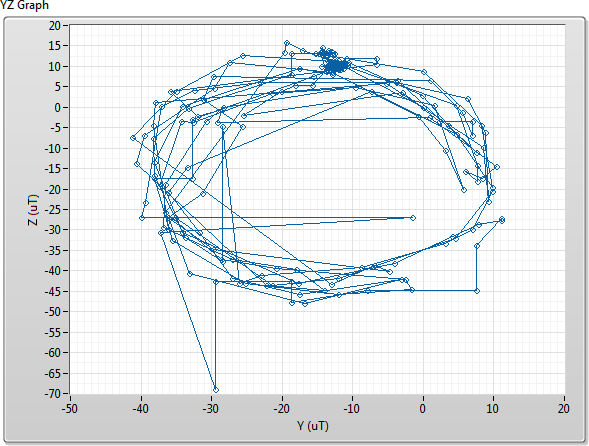


Figure 2-1: Sample magnetometer offset error along the Y and Z axes

2-3 Deducing roll and pitch angles by applying Equations 1-1 and 1-2 to the accelerometer output introduces unwarranted mathematical singularities. Known as Euler singularities, they manifest themselves as incorrect roll and pitch angle measurements when the sensor is posed in certain orientations.

Your task is to observe these singularities. Run the VI and wait for the **Initialized?** LED to turn on. Hold the IMU in the horizontal plane with the accelerometer Z-axis pointing upward such that the indicated roll and pitch angles are close to zero. Slowly pitch the IMU downward, while preventing any roll in the sensor. Observe how the indicated pitch value increases in the VI as the tilt increases. At the same time, closely observe the roll of the sensor. Since you are not rolling the sensor you would expect the VI to indicate a roll angle of close to 0 deg. However, as the pitch angle approaches +90 deg you will notice the roll angle very quickly increases. This unwarranted roll measurement is caused by the Euler singularity problem. Observe a similar singularity in the measured pitch as you roll the sensor sideways.

Using the raw accelerometer outputs (A\_X, A\_Y, and A\_Z) and Equations 1-1 and 1-2, explain how these singularities occur.