Head Impact Criterion Calculation

Final Report for KIN472 Winter 2010

& Other things

Mr. Andrew Laing

Prepared by: Ali Saeed | 20177598 | 4B Mechatronics Engineering

University of Waterloo

Monday, April 12, 2010

Prof. Andrew Laing
Assistant Professor
Department of Kinesiology
Faculty of Applied Health Sciences
University of Waterloo
200 University Ave, West
Waterloo, ON N2L 3G1

Dear Mr. Laing,

RE: KIN472 – Final Report (Winter, 2010)

Attached please find a copy of the final report which I am submitting to receive credit forKIN472. This report was solely prepared by me to detail my work at the IBAL Lab.

I started the term working with Mr. Wendell Prime, who proved to be of great help. He calibrated the signal conditioner and provided his support throughout the term. Once the calibration was complete, Matlab and Visual Basic code was developed to calculate HIC values. Mr. Sandy Wright gladly assisted me throughout this period with his expertise in NIAD.

Initially it was decided that I would be mainly responsible for rehabilitating the crash-test dummy. However, due to delays in ordering its components the focus was instead moved to the Head Impact Testing apparatus. The current state of progress on the crash test dummy is summarized in the last section of this report.

This report and all code developed are hosted online at http://code.google.com/p/hic-calculator-ibal-lab-uw/, so a copy can be obtained any time, in case they ever get lost or deleted.

I would like to thank you for letting me work with you. It was a great experience working in the IBAL Lab and so much more joyful with so many nice people around.

Yours truly,

Ali Saeed
4B Mechatronics Engineering
mohammadalipak@gmail.com
(519) 635-9055

Contents

1. Signal Conditioner Calibration

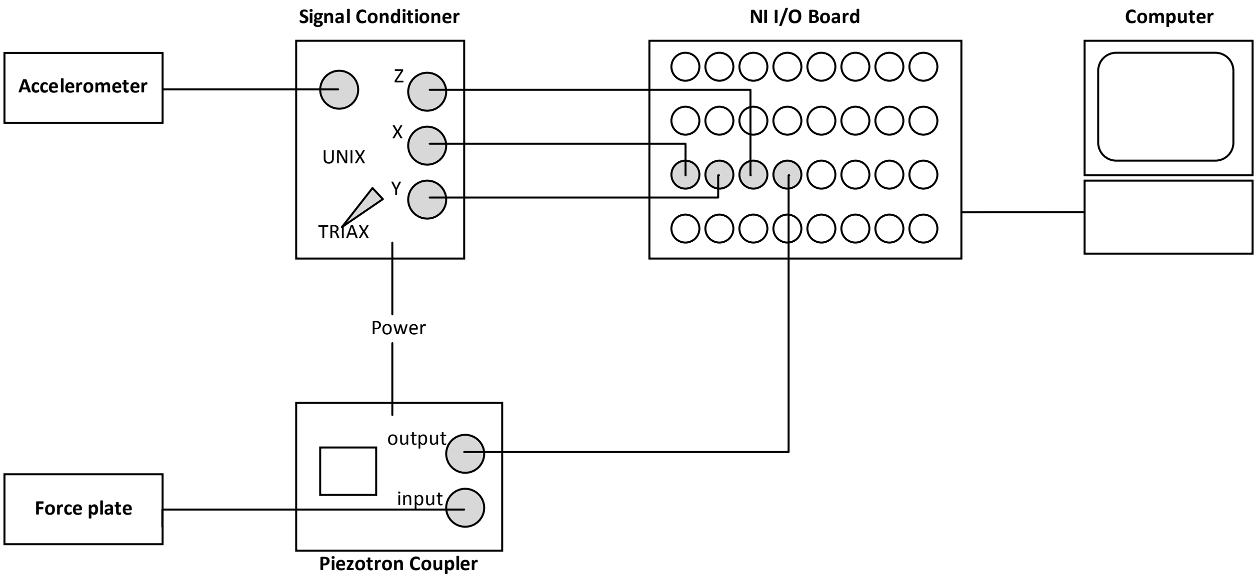
The calibration of the signal conditioner was performed by Mr. Wendell Prime. He replaced the old resistors with lower tolerance resistors to improve precision. The 8 channels of the signal conditioner are actually 4 channels reciprocated, meaning that channel 5 is the negative of channel 1, channel 6 is the negative of channel 2, and so on. The results of his calibration are tabulated below. The complete results sheet that he prepared can be found in Appendix A.

Channel 1	calibFactorX = 312.2 calibFactorY = 305.1 calibFactorZ = 296	Channel 5	calibFactorX = -312.2 calibFactorY = -305.1 calibFactorZ = -296
Channel 2	calibFactorX = 156.5 calibFactorY = 153 calibFactorZ = 148.4	Channel 6	calibFactorX = -156.5 calibFactorY = -153 calibFactorZ = -148.4
Channel 3	calibFactorX = 78.47 calibFactorY = 76.68 calibFactorZ = 74.4	Channel 7	calibFactorX = -78.47 calibFactorY = -76.68 calibFactorZ = -74.4
Channel 4	calibFactorX = 15.73 calibFactorY = 15.38 calibFactorZ = 14.91	Channel 8	calibFactorX = -15.73 calibFactorY = -15.38 calibFactorZ = -14.91

2. Using NIAD to Collect Data

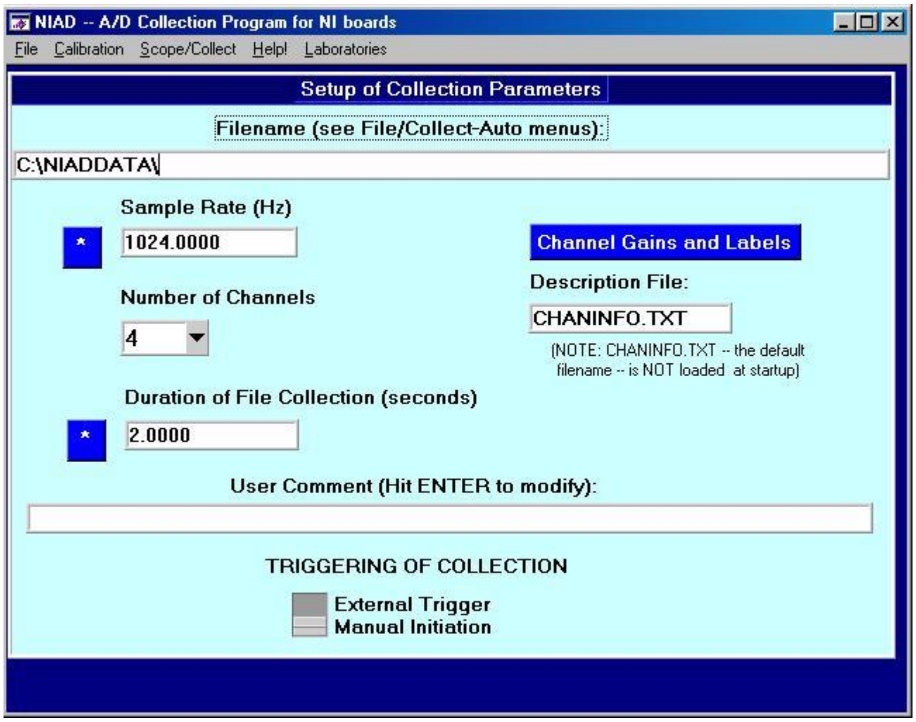
NIAD stands for National Instruments Analog-to-Digital. NIAD is a data acquisition program for National Instruments boards that converts analog readings into digital values. For the head impact testing, accelerometer and force plate readings are gathered using NIAD. Below is a step-by-step procedure for data acquisition through NIAD.

Step 1: Make sure everything is connected
See the schematic below to ensure all connections are properly made.



Step2: Power up
Turn on the computer. The password is “kin”. Also, power up the signal conditioner for accelerometer and the Piezotron Coupler for the force plate.

Step3: Launch NIAD
There is a NIAD Collection icon on the Desktop. The actual application is located at: C:\niad32\NIADWIN32.exe. You should see this startup screen when NIAD loads:



Step4: Specify the location of data storage
Go to File>New File Entry. Type in a name for the new empty file and hit OK. This will create an empty file to be filled later when data is gathered.

Step5: Test using Oscilloscope
Go to Scope/Collect>Oscilloscope to view live readings of the accelerometer and force plate.

Step6: Gather Data
Go to Scope/Collect>File Collection to collect data for the specified duration and save it in a file.

3. Pseudo-code for Calculating HIC value

The pseudo-code as given in the ASTM standard for calculating the HIC value is reproduced below. The only deviation from this pseudo code is the condition to make sure the counter i does not equal j to prevent divide by zero error as explained in the code sections of Matlab and Visual Basic sections.

```
//GLOBAL VARIABLES
      var
//Data Acquistion Information
      SampleFreqency: integer; //Data acquisition rate, samples/second
      nSamples      : integer; //Number of acquired data samples
//Input Data
      AccelData: array [0..nSamples] of real; //Array of acceleration in g
      units
//Outputs
      HICmax        : real;   //HIC score
      HICinterval  :real;    //HIC interval

//HIC CALCULATION PROCEDURE
      Procedure HIC_Calculation;

      //LOCAL VARIABLES
      Var

      // Intermediate Results
      integral         :array [0..nSamples-1] of real; //HIC Integral Values
      iHIC0,iHIC1      :integer;    //HIC interval boundaries
      HIC              :real;    //Intermediate HIC result

      //Counters
      i,j              :integer;

      begin

      //initialize results
      iHIC0 := 0;
      iHIC1 := 0;
      HICmax := 1.0;

      //Calculate integral
      integral[0]:=0.0;
      for i:=1 to nSamples do integral[i]:=integral[i-1] +
      (AccelData[i]+AccelData[i-1])/2;

      //Scan all possible HIC intervals for maximum score
      for i := 0 to nSamples-1 do
            for j := i+1 to nSamples do
                  begin
                        HIC:=(integral[j]-integral[i])/(j-i);
                        If HIC>0.0
                              Then HIC:=Power(HIC,2.5)
                              Else HIC:=0.0;
                        HIC:=HIC*(j-i)/SampleFrequency;
                        If HIC>HICmax then
                              begin
                                    HICmax:=HIC;
```

```
                                    iHIC0:=i;
                                    iHIC1:=j;
                              end;
                  end;

            //Calculate the HIC interval
            HICinterval := (iHIC1-iHIC0)/SampleFrequency;
            end;

      end;
```

3.1 Butterworth Filter Pseudo Code

```
//GLOBAL VARIABLES
const nSamples;       //Number of acquired data samples
var
// Data Acquisition Information
SampleFrequency: integer; //Data acquisition rate, samples/second
nSamples       : interger; //Number of acquired data samples
//Input Data which will be replaced with the filtered data
AccelData: array[0..nSamples] of real; //Array of acceleration data in g
units

//Butterworth Filter
Procedure Butterworth_Filter
//LOCAL VARIABLES
var temp: array [0..nSamples] of real; //Intermediate results
a,b: array [0..2] or real; //Filter coefficients
i,j: integer;             //Counters
begin
a[0] = 0.071893;
a[1] = 0.143786;
a[2] = 0.071893;
b[1] = 1.111586;
b[2] = -0.399159;

//First pass in forward direction
temp:=AData;
for i:=2 to ScanSize-1 do
AData[i]:=a[0]*temp[i] + a[1]*temp[i-1] + a[2]*temp[i-2] + b[1]*Adata[i-1] +
b[2]*Adata[i-2]

//Second pass in backward direction
temp = AData;
for i:= ScanSize-3 downto 0 do
AData[i]:=a[0]*temp[i] + a[1]*temp[i+1] + a[2]*temp[i+2] + b[1]*Adata[i+1] +
b[2]*Adata[i+2]
```

4. Matlab HIC Routine

Matlab R2009b is used to create 3 functions for calculating the HIC value. Two of the functions are used to calculate HIC based on sinusoidal data and the 3rd is to be used for real empirical data. These functions are saved in the My Documents folder of the actlaing user on the JENLAB1 workstation. A copy can be obtained from: http://code.google.com/p/hic-calculator-ibal-lab-uw/ for all code & this report at any time.

mysine.m	Helper function for calculateSinusoidalHIC.m
calculateSinusoidalHIC.m	Calculates HIC based on sinuisoidal waves
calculateHIC.m	Calculates HIC based on real raw data

4.1 Sinusoidal Data Operation

To get HIC values based on sinusoidal waves, run calculateSinusoidalHIC.m by changing the Current Folder to where this file is saved (e.g. C:\Documents and Settings\actlaing.JENLAB1\My Documents\HIC) and typing the following at the Matlab command line:

```
T = 0.01
gmax = 100
[HICmax,HICinterval] = calculateSinusoidalHIC(T,gmax)
```

4.2 Real Data Operation

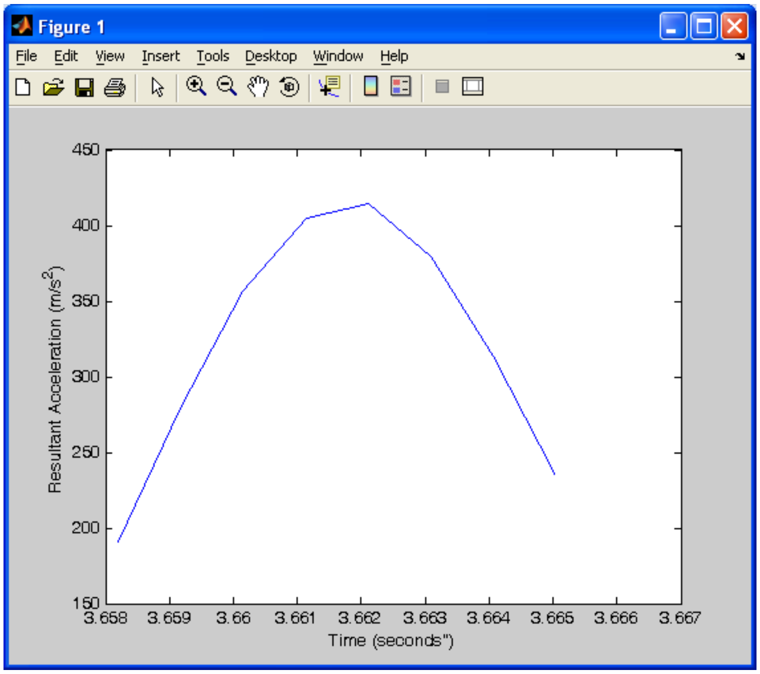
Open Matlab, change the Current Folder to where the calculateHIC.m file is stored (e.g. C:\Documents and Settings\actlaing.JENLAB1\My Documents\HIC). Then type the following on the command line:

```
voltRange = 10;
sampleFrequency = 1024;
signalConditionerChannel = 1;
filePath = 'C:\Documents and Settings\actlaing.JENLAB1\My
Documents\HIC\Ali_firstWorking'
[HICmax,HICinterval,gmax] =
calculateHIC(voltRange,sampleFrequency,signalConditionerChannel,filePath)
```

For more info on the routine, type `help calculateHIC`. Make sure the raw data file is formatted properly (see Data File Format under *Visual Basic HIC Program* section. Also, see the sample data file format under *Raw Data File* section). If the file contains the header, Matlab will throw the following `error: Number of columns on line 4 of ASCII file C:\dataFile must be the same as previous lines.`

4.3 Matlab Graph

The graph that Matlab outputs looks like the following:



4.4 Matlab Code

Below is the code for mysine.m, which is a helper function for calculateSinusoidalHIC.m:

```
function x = mysine(A,T,fs)%f,phi,fs,d)
% mysine   Function to generate a cosine wave
% http://www.eas.asu.edu/~hasancam/courses/Spring-2002/ece194/intro_matlab.
% doc
% usage:
%         x = mysine(A,f,phi,d)
%         A = desired amplitude
%         T = target pulse width
%         fs = sampling frequency (number of samples per second)
%         d = duration of the wave in seconds

%         t = time
%         f  = desired frequency
%         phi = desired phase

t = 0: 1/fs: T;
%t = 0: 0.001: t;
%x = A*(1-cos(2*pi*f*t + phi));
x = A*(1-cos(2*pi*t/T));
plot(t,x,'r');
```

Below is the code for calculateSinusoidalHIC.m for calculating HIC based on sinusoidal input:

```
function [HICmax,HICinterval] = calculateSinusoidalHIC(T,gmax)
```

```matlab
%GLOBAL VARIABLES
    %Data Acquisition Info
        sampleFrequency = 20000.0; %1/sec %int
        %T = 0.01;
        nSamples = T * sampleFrequency; %size(ax,2); %60 %int
        %gmax = 150;
        %c = 0.5; %0.1589;
        A = gmax/2; % A is not amplitude, it's the peak to peak range
    %Input from Cosine Wave
        ax = mysine(0.015*100/2, T, sampleFrequency);
        ay = mysine(0.015*100/2, T, sampleFrequency);
        az = mysine(0.015*100/2, T, sampleFrequency);
        %plot(ax, 'r');hold on;
    %Resultant Acceleration
        accelData = zeros(nSamples);
        for ii = 1:nSamples
            accelData(ii) = sqrt(ax(ii)^2 + ay(ii)^2 + az(ii)^2); %60 values
%double array
        end
        accelData = mysine(A,   T,   sampleFrequency)';

    %ButterWorth Filter
        [bw_a,bw_b] = butter(2,0.20775)
        accelData = filter(bw_a,bw_b,accelData);
        %plot(t, accelData, 'g');
    %Outputs
        %HICmax;
        %HICinterval;
%HIC CALCULATION PROCEDURE
    %function [HICmax,HICinterval] = calculateHIC(filename)
    %double integral[];
    %int iHIC0, iHIC1;
    %double HIC;
    %int i,j;

    iHIC0 = 0;
    iHIC1 = 0;
    HICmax = 1.0;
    integral = zeros(nSamples);
    integral(1) = 0.0;
    for ii = 2:nSamples
        integral(ii) = integral(ii-1) + (accelData(ii)+accelData(ii-1))/2;
    end

    %Scan all possible HIC intervals for maximum score
    warning off MATLAB:divideByZero
    for ii=1:nSamples-1
        for jj=1:nSamples
            HIC = (integral(jj)-integral(ii))/(jj-ii);
            if (HIC>0.0)
                HIC = HIC^2.5;
            else
                HIC = 0.0;
            end
            HIC = HIC*(jj-ii)/sampleFrequency;
```

```
            if (HIC>HICmax)
                HICmax = HIC;
                iHIC0 = ii;
                iHIC1 = jj;
            end
        end
        HICinterval = (iHIC1-iHIC0)/sampleFrequency;
    end
```

Below is the code for calculateHIC.m, which calculates HIC based on real raw data collected in an experiment. The line `if (ii ~= jj)` is added to ensure there is never a divide-by-zero error when ii and jj are the same.

```
function [HICmax,HICinterval,gmax] =
calculateHIC(voltRange,sampleFrequency,signalConditionerChannel,filePath)
%function [HICmax,HICinterval] =
calculateHIC(voltRange,sampleFrequency,signalConditionerChannel)
%voltRange - the voltage range specified in NIAD
%sampleFrequency - the sample frequency at which the data was collected
%signalConditionerChannel is the channel at which the data was gathered
%filePath - is the path to the data file. Remember to format the data file so
%it doesn't have any header lines. Also the data file should have ax, ay
%and az in columns 2, 3 and 4 respectively
        D2A = voltRange*2/4096;
        switch signalConditionerChannel
            case 1
                calibFactorX = 312.2;
                calibFactorY = 305.1;
                calibFactorZ = 296.0;
            case 2
                calibFactorX = 156.5;
                calibFactorY = 153.0;
                calibFactorZ = 148.4;
            case 3
                calibFactorX = 78.47;
                calibFactorY = 76.68;
                calibFactorZ = 74.40;
            case 4
                calibFactorX = 15.73;
                calibFactorY = 15.38;
                calibFactorZ = 14.91;
            case 5
                calibFactorX = -312.2;
                calibFactorY = -305.1;
                calibFactorZ = -296.0;
            case 6
                calibFactorX = -156.5;
                calibFactorY = -153.0;
                calibFactorZ = -148.4;
            case 7
                calibFactorX = -78.47;
                calibFactorY = -76.68;
```

```
                calibFactorZ = -74.40;
            case 8
                calibFactorX = -15.73;
                calibFactorY = -15.38;
                calibFactorZ = -14.91;
        end
     %Input from Data File
        %dataFile = load('Ali_firstWorking'); % loads data from file to the
workspace
        dataFile = load(filePath);
        ax = dataFile(:,2) * D2A * calibFactorX; % extract Ax
        ay = dataFile(:,3) * D2A * calibFactorY; % extract Ay
        az = dataFile(:,4) * D2A * calibFactorZ; % extract Az
        % Using the first value as ZERO point, normalizing all data
        ax = ax-ax(1);
        ay = ay-ay(1);
        az = az-az(1);
    %Number of Samples
        %if using Data File
            nSamples = size(ax,1);
    %Resultant Acceleration
        %accelData = zeros(nSamples); % zeros(n) gives nxn matrix, can be
improved to give nx1
        for ii = 1:nSamples
            accelData(ii) = sqrt(ax(ii)^2 + ay(ii)^2 + az(ii)^2); %60 values
%double array
        end
        accelData = accelData';
        % Butterworth Filter
        [bw_a,bw_b] = butter(2,0.20775);
        accelData = filter(bw_a,bw_b,accelData);

        %Butterworth Filter Numerical Code (if not using Matlab)
        %bw_a(1) = 0.071893;
        %bw_a(2) = 0.143786;
        %bw_a(3) = 0.071893;
        %bw_b(2) = 1.111586;
        %bw_b(3) = -0.399159;

        %==============FORWARD PASS
        %temp = accelData;
        %for ii = 3:nSamples
        %    accelData(ii) = bw_a(1)*temp(ii) + bw_a(2)*temp(ii-1) +
bw_a(3)*temp(ii-2)...
        %        + bw_b(2)*accelData(ii-1)+bw_b(3)*accelData(ii-2);
        %end
        %==============BACKWARD PASS
        %temp = accelData;
        %for ii = nSamples-2:1
        %    accelData(ii) = bw_a(1)*temp(ii) + bw_a(2)*temp(ii+1) +
bw_a(3)*temp(ii+2)...
        %        + bw_b(2)*accelData(ii+1)+bw_b(3)*accelData(ii+2);
        %end
```

```
    %HIC CALCULATION=======================

gmax = max(accelData);
iHIC0 = 0;
iHIC1 = 0;
HICmax = 1.0;
integral = zeros(nSamples);
integral(1) = 0.0;
for ii = 2:nSamples
    integral(ii) = integral(ii-1) + (accelData(ii)+accelData(ii-1))/2;
end

%Scan all possible HIC intervals for maximum score
for ii=1:nSamples-1
    for jj=1:nSamples
        if (ii ~= jj)
            HIC = (integral(jj)-integral(ii))/(jj-ii);
            if (HIC>0.0)
                HIC = HIC^2.5;
            else
                HIC = 0.0;
            end
            HIC = HIC*(jj-ii)/sampleFrequency;
            if (HIC>HICmax)
                HICmax = HIC;
                iHIC0 = ii;
                iHIC1 = jj;
            end
        end
    end
    HICinterval = (iHIC1-iHIC0)/sampleFrequency;
end
%plot graph
clf; %delete old figure
timearray = iHIC0:1:iHIC1;
timearray = timearray/sampleFrequency;

plot(timearray,accelData(iHIC0:iHIC1));
xlabel('Time (seconds")');
ylabel('Resultant Acceleration (m/s^2)');
```

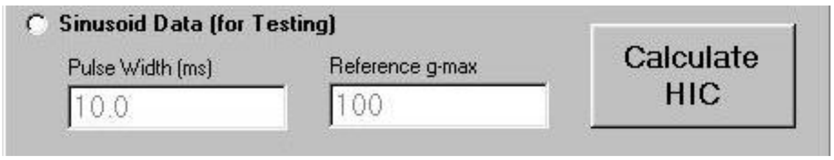
5. Visual Basic HIC Program

Visual Basic 6.0 is used to develop a replica of the Matlab code that can run on the machine that is part of the Head impact testing apparatus.

The program can be operated using real data or sinusoidal test data. Launch the application by double-clicking on the HIC.exe icon on the Desktop or locate it in C:\My Documents\HIC VB Program\HIC.exe.

5.1 Sinusoidal Data Operation

To use this mode, click on the radio button labeled *Sinusoid Data (for Testing)*. Then enter the desired Pulse Width and Reference g-max. When ready to calculate, click *Calculate HIC*.



The sinusoidal test data is generated by the following formula with supplied period (T, also known as Pulse Width) and the Reference g-max (A, which is two times the amplitude):

$$V = A\left(1 - \cos\left(2\pi\frac{t}{T}\right)\right)$$

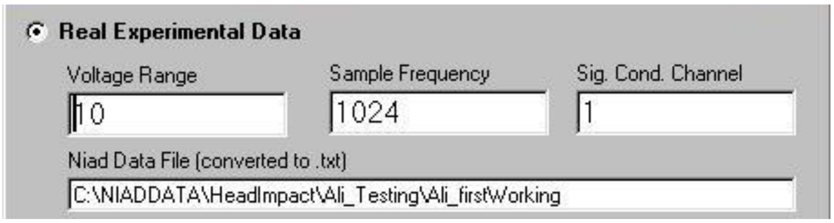
(Standard Specification for Impact Attenuation of Surfacing Materials Within the Use Zone of Playground Equipment, p15)

The table below compares the theoretical HIC values with the ones generated through the code's implementation in Visual Basic:

Pulse Width (T) ms	Reference g-max	Theoretical HIC	Calculated HIC in VB	Error	Error %	Theoretical HIC Interval (ms)	Calculated HIC Interval in VB (ms)
10.0	100	302.9	**302.9**	0.0	0.0000	5.08	**5.10**
10.0	150	834.8	**834.7**	-0.1	-0.0120	5.08	**5.10**
10.0	200	1713.7	**1713.5**	-0.2	-0.0117	5.08	**5.10**
20.0	100	605.9	**605.9**	0.0	0.0000	10.15	**10.20**
20.0	150	1669.6	**1669.5**	-0.1	-0.0060	10.15	**10.20**
20.0	200	3427.4	**3427.2**	-0.2	-0.0058	10.15	**10.20**

5.2 Real Data Operation

For operating on the real data acquired through the accelerometer and saved in ASCII format through NIAD, click of the radio button labeled *Real Experimental Data*.

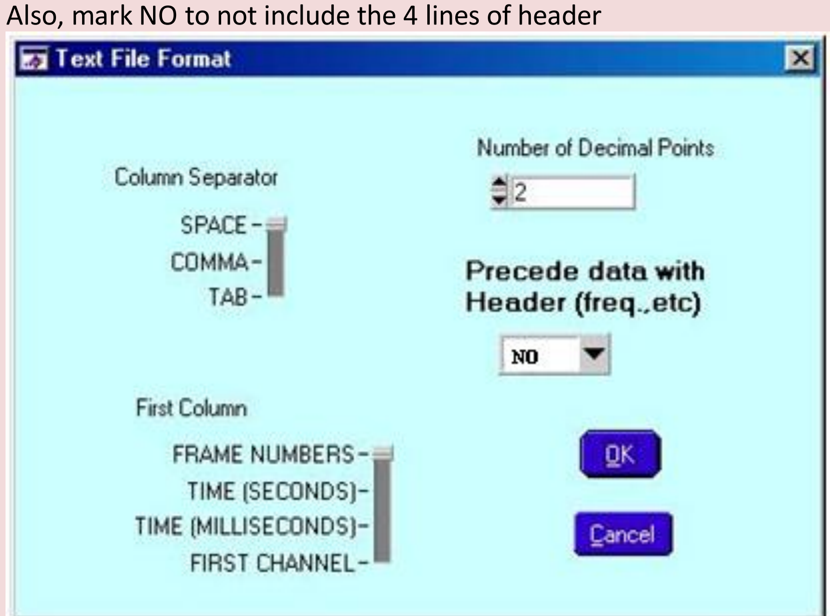


Then supply the voltage range, sample frequency (both of these come straight from the settings in NIAD) and the signal conditioner channel used at the time of data acquisition. Lastly, enter the path to the data file and click *Calculate HIC.*

5.3 Data File Format

Make sure that the file is saved in an ASCII format and is space-delimited. NIAD stores data files in a format that VB cannot understand. To convert NIAD files to ASCII format, click **File>Conversion of file to TEXT format** in the NIAD application. Locate the NIAD file to convert. Type in any extension (e.g. .txt, or .csv). NIAD will then give you some options for conversion.

- Choose space as Column Separator
- Also, mark NO to not include the 4 lines of header



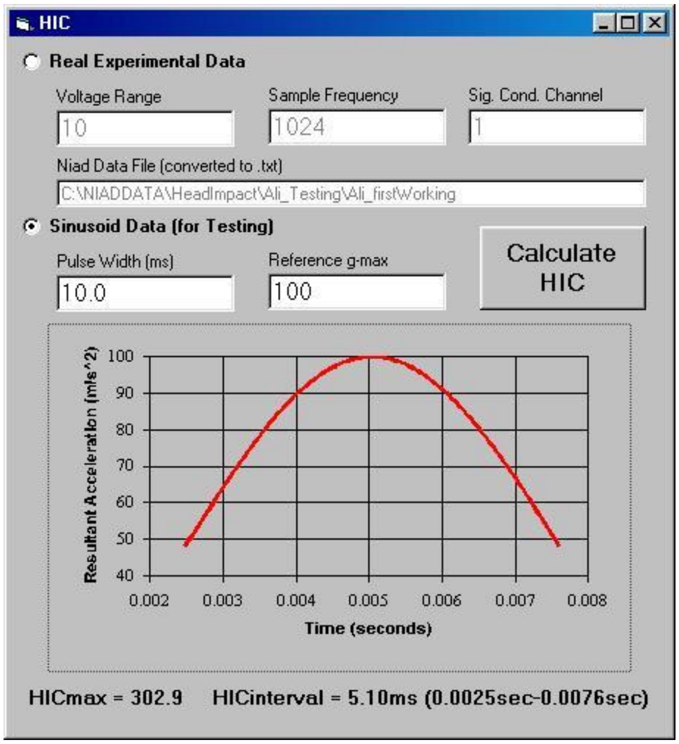
For a sample raw data file, see the ***Raw Data File*** section.

Progress of HIC calculation will be shown as the program is busy. When completed, the resultant acceleration will be plotted against time during the interval of maximum HIC value.

5.4 Butterworth Filter

The Implementation of the Butterworth Filter in Visual Basic was carried out as specified in the ASTM document. The Butterworth Filtering pseudo-code can be found in the ***Pseudo-Code*** section.

5.5 Visual Basic GUI



5.6 Visual Basic Code

Below is the visual basic code. The line `If (i <> j) Then` is added to ensure there is never a divide-by-zero error when i and j are the same.

```
Option Explicit

Private Sub Command1_Click()
Command1.Enabled = False 'disable CalculateHIC button
pb.Visible = True        'show progress bar
'===============================================
'================VARIABLES==================
'Data conversion from raw to acceleration
Dim accelData(), ax(), ay(), az(), force, integral() As Double
Dim sampleFrequency, voltRange, nSamples As Integer
Dim i, j, num, D2A, sigCondChannel, calibFactorX, calibFactorY, calibFactorZ
As Double
'ButterWorth Filter
Dim bw_a(3), bw_b(2) As Double
'HIC
Dim iHIC0, iHIC1, HICmax, currentHIC, HICinterval As Double
'Sinusoidal
Dim T, smallT, A, Pi As Double
Dim gmax As Integer
'========================================================
'===============VARIABLE INITIALIZATION================
Pi = 4 * Atn(1)
sampleFrequency = txtSampleFreq.Text
voltRange = txtVoltRange.Text
sigCondChannel = txtSigCondChannel.Text
```

```
D2A = voltRange * 2 / 4096
Select Case sigCondChannel
Case 1:
    calibFactorX = 312.2
    calibFactorY = 305.1
    calibFactorZ = 296
Case 2:
    calibFactorX = 156.5
    calibFactorY = 153
    calibFactorZ = 148.4
Case 3:
    calibFactorX = 78.47
    calibFactorY = 76.68
    calibFactorZ = 74.4
Case 4:
    calibFactorX = 15.73
    calibFactorY = 15.38
    calibFactorZ = 14.91
Case 5:
    calibFactorX = -312.2
    calibFactorY = -305.1
    calibFactorZ = -296
Case 6:
    calibFactorX = -156.5
    calibFactorY = -153
    calibFactorZ = -148.4
Case 7:
    calibFactorX = -78.47
    calibFactorY = -76.68
    calibFactorZ = -74.4
Case 8:
    calibFactorX = -15.73
    calibFactorY = -15.38
    calibFactorZ = -14.91
End Select

nSamples = 0

'==================================================
'=====================READING DATA FILE===========
Dim TextLine As String
If optRealData.Value = True Then                'Real Data
    Open txtFilePath.Text For Input As #1
    i = 1
    ReDim Preserve ax(1 To i)
    ReDim Preserve ay(1 To i)
    ReDim Preserve az(1 To i)
    'NO LONGER NEED TO IGNORE FIRST 4 LINES, AS RAW DATA FILE SHOULD NOT HAVE
HEADER
    'Line Input #1, TextLine 'ignore first line (header)
    'Line Input #1, TextLine 'ignore second line (header)
    'Line Input #1, TextLine 'ignore third line (header)
    'Line Input #1, TextLine 'ignore fourth line (header)
    Input #1, num, ax(i), ay(i), az(i), force ' first row of actual data
    ax(i) = ax(i) * D2A * calibFactorX '- ax(1)
```

```
    ay(i) = ay(i) * D2A * calibFactorY '- ay(1)
    az(i) = az(i) * D2A * calibFactorZ '- az(1)
    Do While Not EOF(1)
        i = i + 1
        'Expand arrays to hold the new row
        ReDim Preserve ax(1 To i)
        ReDim Preserve ay(1 To i)
        ReDim Preserve az(1 To i)
        'Parse Data for each line
        Input #1, num, ax(i), ay(i), az(i), force
        ax(i) = ax(i) * D2A * calibFactorX - ax(1)
        ay(i) = ay(i) * D2A * calibFactorY - ay(1)
        az(i) = az(i) * D2A * calibFactorZ - az(1)
    Loop
    nSamples = i
    Close #1

    ax(1) = 0
    ay(1) = 0
    az(1) = 0
    '================================================
    '==============RESULTANT ACCELERATION============
    ReDim accelData(1 To nSamples)
    For i = 1 To nSamples
        accelData(i) = Sqr(ax(i) ^ 2 + ay(i) ^ 2 + az(i) ^ 2)
    Next i

    '================================================
    '=================BUTTERWORTH FILTER=============
    bw_a(0) = 0.071893
    bw_a(1) = 0.143786
    bw_a(2) = 0.071893
    bw_b(1) = 1.111586
    bw_b(2) = -0.399159
    '==================Forward pass
    Dim temp() As Double
    ReDim temp(nSamples)
    For i = 1 To nSamples
        temp(i) = accelData(i)
    Next i
    For i = 3 To nSamples
        accelData(i) = bw_a(0) * temp(i) + bw_a(1) * temp(i - 1) + bw_a(2) *
temp(i - 2) _
                        + bw_b(1) * accelData(i - 1) + bw_b(2) * accelData(i
- 2)
    Next i
    '==================Backward pass
    For i = 1 To nSamples
        temp(i) = accelData(i)
    Next i
    For i = nSamples - 2 To 1
        accelData(i) = bw_a(0) * temp(i) + bw_a(1) * temp(i + 1) + bw_a(2) *
temp(i + 2) _
                        + bw_b(1) * accelData(i + 1) + bw_b(2) * accelData(i
+ 2)
```

```
    Next i

'===================================================
'===================SINUSOIDAL DATA=================
ElseIf optSine.Value = True Then               'Cosine wave
    ' GET 'T' & 'gmax' from user
    T = txtPulseWidth.Text / 1000 '0.01
    gmax = txtRefGmax.Text '100
    sampleFrequency = 20000
    nSamples = sampleFrequency * T
    A = gmax / 2
    ReDim accelData(1 To nSamples + 1)
    accelData(1) = 0
    For i = 1 To nSamples
        smallT = i / sampleFrequency
        accelData(i + 1) = A * (1 - Cos(2 * Pi * (smallT / T)))
    Next i
Else
    MsgBox ("Error: Please select either Real Expermental Data or Sinuisoidal
Data")
End If

'===============================================
'==================HIC CALCULATION==============
iHIC0 = 0
iHIC1 = 0
HICmax = 1
'==============g-MAX
Dim rgmax As Double
rgmax = 0
For i = 1 To nSamples
    If rgmax < accelData(i) Then
        rgmax = accelData(i)
    End If
Next i
'==============INTEGRATION
ReDim integral(1 To nSamples)
integral(1) = 0
For i = 2 To nSamples
    integral(i) = integral(i - 1) + ((accelData(i) + accelData(i - 1)) / 2)
Next i

'==============HIC CALCULATION LOOPS
For i = 1 To nSamples - 1
    For j = 1 To nSamples
        If i <> j Then
            currentHIC = (integral(j) - integral(i)) / (j - i)

            If currentHIC > 0 Then
                currentHIC = currentHIC ^ 2.5
            Else
                currentHIC = 0
            End If
            currentHIC = currentHIC * (j - i) / sampleFrequency
```

```
            If currentHIC > HICmax Then
                HICmax = currentHIC
                iHIC0 = i
                iHIC1 = j
            End If
        End If
        'DoEvents
    Next j
    DoEvents
    HICinterval = (iHIC1 - iHIC0) / sampleFrequency
    pb.Value = Int(100 * i / nSamples)
Next i

'=====================================================
'=================DISPLAYING RESULTS ON SCREEN========
pb.Visible = False 'hide progress bar
lblOutput.Caption = "HICmax = " & Format(HICmax, "#0.0") & "     " &
"HICinterval = " & Format(HICinterval * 1000, "#0.00") & "ms (" &
Format(iHIC0 / sampleFrequency, "#0.0000") & "s-" & Format(iHIC1 /
sampleFrequency, "#0.0000") & "s)" & vbCrLf & "g-max: " & rgmax
Command1.Enabled = True 'enable CalculateHIC button
'=========CHART
chart.RowCount = iHIC1 - iHIC0 + 1
For i = iHIC0 To iHIC1
    chart.DataGrid.SetData i - iHIC0 + 1, 1, i / sampleFrequency, False
    chart.DataGrid.SetData i - iHIC0 + 1, 2, accelData(i), False
Next i
chart.Plot.UniformAxis = False
'chart.Title.Text = "Resultant Acceleration during max HIC interval"
chart.Plot.Axis(VtChAxisIdX).AxisTitle.Text = "Time (seconds)"
chart.Plot.Axis(VtChAxisIdX).AxisTitle.VtFont.Name = "Arial"
chart.Plot.Axis(VtChAxisIdX).AxisTitle.VtFont.Style = 1
chart.Plot.Axis(VtChAxisIdY).AxisTitle.Text = "Resultant Acceleration
(m/s^2)"
chart.Plot.Axis(VtChAxisIdY).AxisTitle.VtFont.Name = "Arial"
chart.Plot.Axis(VtChAxisIdY).AxisTitle.VtFont.Style = 1
End Sub


Private Sub Form_Unload(Cancel As Integer)
End
End Sub

Private Sub optSine_Click()
'routine to Disable Real Data Fields
    txtPulseWidth.Enabled = True
    txtRefGmax.Enabled = True
    Me.txtVoltRange.Enabled = False
    Me.txtSampleFreq.Enabled = False
    Me.txtSigCondChannel.Enabled = False
    txtFilePath.Enabled = False
End Sub

Private Sub optRealData_Click()
'routine to Disable Sinusoid Data Fields
```

```
    txtPulseWidth.Enabled = False
    txtRefGmax.Enabled = False
    Me.txtVoltRange.Enabled = True
    Me.txtSampleFreq.Enabled = True
    Me.txtSigCondChannel.Enabled = True
    txtFilePath.Enabled = True
End Sub
```

6. Raw Data File

Here is a sample of what the raw data file looks like, once it is converted to TEXT/ASCII format from NIAD.

```
0  -3.00 2.00 -9.00 -1.00
1  -2.00 1.00 -9.00 -1.00
2  -2.00 2.00 -8.00 0.00
3  -3.00 1.00 -9.00 -1.00
4  -2.00 1.00 -9.00 -1.00
5  -2.00 2.00 -8.00 -1.00
6  -3.00 2.00 -9.00 -1.00
7  -2.00 1.00 -9.00 -1.00
8  -2.00 2.00 -7.00 -1.00
9  -2.00 3.00 -9.00 -1.00
10  -1.00 1.00 -9.00 -1.00
11  -1.00 2.00 -8.00 -1.00
12  -2.00 2.00 -8.00 -1.00
13  -2.00 1.00 -9.00 -1.00
14  -2.00 3.00 -8.00 -1.00
15  -2.00 1.00 -9.00 -1.00
16  -2.00 1.00 -10.00 0.00
17  -2.00 2.00 -9.00 0.00
18  -3.00 2.00 -9.00 -1.00
19  -2.00 1.00 -8.00 -1.00
20  -2.00 2.00 -9.00 -1.00
21  -3.00 1.00 -9.00 -1.00
```

7. Journal Articles

The three journal articles related to the Head Impact Testing I consulted are:

1. Brian G. McHenry. (2004). *Head Injury Criterion and the ATB.* 1-8.
2. ASTM Committee. (2004). *Standard Specification for Impact Attenuation of Surfacing Materials Within the Use Zone of Playground Equipment.* 1-23.
3. Patrick J. Bishop, Robert W. Norman and John W. Kozey. (1984). *An evaluation of football helmets under impact conditions*. The American Journal of Sports Medicine. 233-236.

8. Crash-Test Dummy

The crash test dummy is about 25 years old (as per Denton Inc.) and is missing two cables for Denton model 1716 sensor (S/N 083). Unfortunately, Denton no longer produces cables that can fit on the connectors currently present at the dummy sensor. Newer model cables are the only choice, which can be used if the connectors on the dummy's sensor are also updated. The cost to update the connections and purchase new cables (including labor and shipping) is roughly $1,100. See full quote in Appendix B. Also, the email communication between Mr. Wendell Prime and Denton Inc. is included in Appendix C.

Appendix A: Mr. Wendell Prime's Calibration Sheet

Endevco Triaxial Accelerometer
Model 7276A - S/N BB23
with Endevco Amplifier #1012 & #1010 Mode Card

Endevco Output Sensitivity is in mv/g (with 10V excitation)
Output Sensitivity X = 0.1589 mv/g Y = 0.1626 mv/g Z = 0.1676 mv/g

CAL SHUNT VALUES = microvolts/volt ÷ (Output Sensitivity(mv/g)/10)(1000)

Calib 1 & 5	X = 312.2 g	Y = 305.1 g	Z = 296.0 g
Calib 2 & 6	X = 156.5 g	Y = 153.0 g	Z = 148.4 g
Calib 3 & 7	X = 78.47 g	Y = 76.68 g	Z = 74.40 g
Calib 4 & 8	X = 15.73 g	Y = 15.38 g	Z = 14.91 g

R (SHUNT)

Calib 1 & 5	49.9 Kohms	(R51 on Mode Card)
Calib 2 & 6	100 Kohms	(R52 on Mode Card)
Calib 3 & 7	200 Kohms	(R53 on Mode Card)
Calib 4 & 8	1 Mohms	(R54 on Mode Card)

R (GAUGE) = 500 ohms
R (Completion) = 1000 ohms
NOTE: R(SHUNT) is connected across R(Completion)

$$\text{microvolts/volt} = \left| \frac{1}{2} - \frac{R\,(\text{SHUNT})}{R\,(\text{Completion}) + 2*(R\,(\text{SHUNT}))} \right| * 10^{6}$$

microvolts/volt (49.9 Kohms) = 4960.3175
microvolts/volt (100 Kohms) = 2487.5622
microvolts/volt (200 Kohms) = 1246.8828
microvolts/volt (1 Mohms) = 249.8751

Appendix B: Quote for Crash Test Dummy's Neck Sensor

Denton ATD, Inc.
2967 Waterview Drive
Rochester Hills, MI USA 48309

DENTON

Phone (248)852-5100
Fax (248)852-6060

Quotation

Quote No : SO29702
Quotation Date: 1/13/2010
Terms: Net 30
Freight: Prepay & Add OR Collect

Revision #1 1/20/10

Quote To:
University of Waterloo

Attn: Mr. Wendell Prime

Line #	Item ID	Description	Qty	UM	Disc %	Disc. Price	Amount
10	**TF-50-30**	Model 50-30' Teflon Cable	1	Ea.	20 %	$453.60	$453.60
20	**1716AJ-TF-50-30**	Neck, Upper, 6 Channel Load Cell	1	Ea.	20 %	$8,200.00	$8,200.00

Model TF-50-30 Cable with 19pin connector. Does not include connector on the data acquisition side.
Used in Model 1716 Upper Neck Load Cell.

Estimated Lead Time: 2 Weeks After Receipt of Order

NOTE: Please refer to this quotation number on future correspondence including purchase order.
This quotation is valid for 30 days. Please feel free to contact me if you require more information.
Refer to website www.RADenton.com for Standard Terms & Conditions of Product Sales.

SIGNATURE ______________________________

Total USD **$8,653.60**

Page 1 of 1

Appendix C: Communication between Denton Inc. & Mr. Wendell Prime

From: Wendell Prime <wprime@uwaterloo.ca>
To: Ali Saeed <amsaeed@uwaterloo.ca>
Date: Tue, 6 Apr 2010 10:22:25 -0400
Subject: FW: Picture 1716 Sensor connectors

From: Mike Kulwicki [mailto:mikek@radenton.com]
Sent: 2010 01 5:15 PM
To: Wendell Prime
Cc: George Stanton; Jami Herbert; Mark Brown; 'Michael Gratopp'; Rebecca Muller; Sharon Cook
Subject: RE: Picture 1716 Sensor connectors

Hi Wendell,

We estimate the cost to replace the current connectors and recal the load cell is about $1100 (that would also include the new cable assembly I originally quoted). I have attached the updated quote for a new load cell, the cable is included with the new load cell.

Please let me know if you have any questions and if decide to send in your load cell to be updated we'll issue you an RMA number.

Mike

[attached quote can be found in appendix B]

From: Wendell Prime [mailto:wprime@uwaterloo.ca]
Sent: Wednesday, January 20, 2010 12:07 PM
To: mikek@radenton.com
Subject: RE: Picture 1716 Sensor connectors

Mike,

A rough estimate will do. Also, please quote a new sensor and cable.

Wendell

From: Mike Kulwicki [mailto:mikek@radenton.com]
Sent: 2010 01 12:05 PM
To: Wendell Prime
Cc: George Stanton
Subject: RE: Picture 1716 Sensor connectors

Hi Wendell,

I couldn't give you a price to convert the connectors until we received the load cell and have a chance to look at it, this is our normal process with returned goods. Are you looking for a rough estimate? We just can't be

precise because of the labor involved.

Mike

From: Wendell Prime [mailto:wprime@uwaterloo.ca]
Sent: Monday, January 18, 2010 2:26 PM
To: mikek@radenton.com
Subject: RE: Picture 1716 Sensor connectors

Mike,

Please provide a quotation to change the connectors and to supply a non terminated cable.

Thank you for your assistance with this matter,

Wendell

From: Mike Kulwicki [mailto:mikek@radenton.com]
Sent: 2010 01 2:23 PM
To: Wendell Prime
Cc: George Stanton
Subject: RE: Picture 1716 Sensor connectors

Hi Wendell,

Unfortunately we don't have a source for this connector any longer, this load cell is about 25 years old. We would have to remove the old connector from the load cell and install a new one.

Mike

From: Wendell Prime [mailto:wprime@uwaterloo.ca]
Sent: Friday, January 15, 2010 1:13 PM
To: mikek@radenton.com; George Stanton
Subject: Picture 1716 Sensor connectors

Mike & George,

I have attached 4 pictures as requested.

Wendell

From: Mike Kulwicki [mailto:mikek@radenton.com]
Sent: 2010 01 9:51 AM
To: Wendell Prime
Subject: RE: Univ of Waterloo CAN: 1716 Sensor cable

Hi Wendell,

Can you take a picture of the top view on the 1716 load cell, where the moment and force connectors are? The

16 pin connector is our standard so I'm wondering if this was a special order.

Mike

From: Wendell Prime [mailto:wprime@uwaterloo.ca]
Sent: Thursday, January 14, 2010 10:22 AM
To: mikek@radenton.com
Subject: Univ of Waterloo CAN: 1716 Sensor cable

Mike,

The TF-50-30 cable in your quotation SO29702 does not seem to be the cable that I need. I need two 12 wire cables as the 1716 sensor that we have has two 12-pin male round connectors. One is for the moments and one is for the forces. The documentation indicates the following color coding for the non terminated ends.

	FX or MX	**FY or MY**	**FZ or MZ**
+Ex	**Brown**	**Red Stripe**	**Green**
+Sig	**Red**	**Black**	**Blue**
-Ex	**Orange**	**White**	**Violet**
-Sig	**Yellow**	**Black Stripe**	**Grey**

Wendell

From: Mike Kulwicki [mailto:mikek@radenton.com]
Sent: 2010 01 4:14 PM
To: Wendell Prime
Cc: George Stanton; Jami Herbert; Mark Brown; 'Michael Gratopp'; Rebecca Muller; Sharon Cook
Subject: RE: Univ of Waterloo CAN: 1716 Sensor cable

Hi Wendell,

Here is the quote for the load cell cables. Please let me know if you have any questions.

Mike

[attachment not included in this report, as the quote is for wrong cables]

From: Wendell Prime [mailto:wprime@uwaterloo.ca]
Sent: Wednesday, January 13, 2010 4:02 PM
To: mikek@radenton.com
Subject: RE: Univ of Waterloo CAN: 1716 Sensor cable

Mike,

The two mating cables do not need to be terminated as we will add our own amplifier connectors.

Thanks,

Wendell

From: Mike Kulwicki [mailto:mikek@radenton.com]
Sent: 2010 01 2:48 PM
To: wprime@uwaterloo.ca
Subject: RE: Univ of Waterloo CAN: 1716 Sensor cable

Hi Wendell,

Will you need connectors installed or do you plan on doing them internally?

Mike

Michael J. Kulwicki
Sales Manager – Americas
Denton ATD, Inc.
2967 Waterview Dr.
Rochester Hills, Mi. 48309
Office Phone: (248)-243-5101
Cell Phone: (248)-884-3390
mikek@radenton.com

From: Wendell Prime [mailto:wprime@uwaterloo.ca]
Sent: Tuesday, January 12, 2010 4:32 PM
To: info@radenton.com
Subject: 1716 Sensor

Sirs,

We are missing the two cables for a Denton model 1716 sensor (S/N 083) that is mounted in Hybrid III dummy. Please provide pricing and availability information for these two cables.

Wendell Prime

Technical Manager 519-888-4567 ext.33563
Kinesiology Dept. BMH1402 FAX 519-888-4849
University of Waterloo wprime@uwaterloo.ca
Waterloo, ON CANADA N2L 3G1