RCA: Real-Time Concussion Analyzer

Timothy E. Coyle, EE, Justin D. Kober, EE, Scott A. Rosa, CSE, and Kenneth W. Van Tassell, EE

Abstract—**We introduce RCA (Real-Time Concussion Analyzer), a real-time system that will allow a football coach to remotely monitor the impacts a player experiences during a game. This system will provide the likelihood that a player has experienced a concussion, allowing coaches to make more informed decisions pertaining to player safety. RCA incorporates an array of accelerometers inside each player's helmet. The sensor data from each helmet is wirelessly transmitted to an Android device, where an application will query a player database on a server, and determine the likelihood of concussion.**

I. INTRODUCTION

CONCUSSIONS in sports have become a growing concern in recent years, despite advancements in safety equipment. According to the Center for Disease Control and Prevention, a concussion is a type of traumatic brain injury, or TBI, caused by a bump, blow, or jolt to the head that can change the way your brain normally works. ñWhile all concussions are serious, most occur without loss of consciousness.ò [1] Recognition and proper response to concussions when they first occur can help prevent further injury or even death. Current concussion detection relies on a coach to constantly check a playerôs reported symptoms after each impact. They monitor a wide array of symptoms from memory to balance. However, many players will try to mask their symptoms [2], [3]. This problem is not localized to any age of player, though RCA is aimed at the high school level. In June of 2012, over 2000 former players filed a lawsuit against the NFL claiming that the league hid the link between football-related head trauma and permanent brain injuries [4]. This lawsuit brought concussion awareness to the forefront of sports medicine.

Preventing concussions in football seems to be an extremely challenging goal, but detecting them is slightly less challenging. RCA is a system intended on detecting concussions in the hopes that a coach will remove an injured player before they may suffer any further damage. However, a system such as RCA could possibly influence how the game of football is played. If such a system was to be employed, players may find themselves removed from games more frequently. This could lead to opposing teams targeting key players, knowing that the key player would have to be removed from the game after receiving a certain concussive force. This problem has been addressed within the last decade by not only researchers but also private companies. There have been a few systems developed to address this, most notably the Head Impact Telemetry (HIT) System [5]. The HIT System takes a similar approach to this problem, utilizing an array of linear accelerometers with wireless transmission to an off-field base station computer. This is a data collection system that lacks portability at the base station and is not cost efficient. Providing only data on forces, there is no likelihood of a concussion presented.

To begin the design of RCA, we developed system specifications by reviewing the current solutions and their limitations. None of the current systems had a way to adapt for multiple sub-concussive impacts, which may be as dangerous as a single serious concussive impact. With this in mind, RCA will not only have wireless transmit capability, but also have the capability for wireless receiving. From their data sheets, the combination of the sensor network, processor, and radio in the helmet is restricted to using 302 mA and 5 V [6], [7], [8]. This means RCA will have a maximum power of 1.51 W. The Android device has its own internal battery and is not included in this power analysis. RCA is installed in a playerôs helmet in such a way that the player will not notice the system. There is a threshold setting in the microcontroller, imbedded in the helmet, to account for minor accelerations that are not related to impacts. In the event of an impact, this threshold will be broken and trigger transmission of the impact data to the Android device. From the impact data, accelerations of all six sensors, the device will compute a resultant hit vector **H**. This is a single vector representation for the acceleration of the headôs center of mass. With the components of **H**, the device then calculates the probability of a concussion. Finally, the Android device stores the results to a player database. The coach will receive an alert with the playerôs name and number, as well as the probability of concussion and show the impact location. This application on the device will have a user menu for the coach to add or remove players, as well as make notes and query the

T. E. Coyle from Easthampton, Ma (e-mail: tcoyle@student.umass.edu)
J. D. Kober from Easthampton, Ma (e-mail: jkober89@gmail.com)
S. A. Rosa from Peabody, Ma (e-mail: sarosa@student.umass.edu)
K. W. Van Tassell from Chelmsford, Ma (e-mail: kvantass@student.umass.edu)

data after the games. Table I shows a list of specifications.

TABLE I
SPECIFICATIONS

Specification	Initial Requirement	Actual Prototype
Weight	<5% increase (typically 102 grams)	120g
Range	25 m	40m
Response Time	<2 s	X
Battery Life	>5 hours	>5 hours
Cost	<$5000 for full team of 52 players	$5040
Power Consumption	<2 W	1.37 W
Acceleration Range	+/- 70 g	+/- 70 g
Sensitivity	Only measure actual impacts	Threshold of 10 g
Durable Packaging	Stable and waterproof	Stable & water resistant

II. Design

A. Overview

Our approach to this problem is to develop a rugged sensor network to be deployed in a helmet. This network consists of six accelerometers, a microcontroller, a wireless radio, and a battery to supply power. We have ruled out the use of gyroscopes and triple-axis accelerometers after speaking with Professor Steven Rowson from Virginia Tech. Both of those sensors proved to be too technically challenging to implement, with the calibration and stability requirements of each. Instead, RCA utilizes an array of single-axis accelerometers oriented towards the center of mass of the head. This array provides a robust solution; if one sensor shifts the system will be less prone to error. From the linear accelerations that these sensors collect, RCA calculates rotational accelerations, which are needed for the risk calculation. We decided on using an 8-bit microcontroller as our impact data processor instead of an FPGA or other means, due to our familiarity with these devices. Microcontrollers have the ability to interface with analog sensors, process information, and transmit data to other devices. A Bluetooth radio is interfaced with the microcontroller, as this type of radio can easily communicate with an Android device. Bluetooth provides a proof of concept, but with range limitations of these radios RCA is not a full scale product. As mentioned previously, we have selected an Android device for our user interface, as Android is the most prevalent operating system for smartphones [9] and it supports open source programming. To store the data and player history, a server was developed with two separate databases, as to not overload the deviceôs memory. We had also initially considered a base station type of approach with a receiver and a laptop, but with the portability requirement we decided to make the system as mobile as possible.

RCA includes three main blocks:

1) Impact Data Collection: An array of sensors placed in the helmet, along with a microcontroller, Bluetooth radio and power supply. Together, these devices detect an impact and transmit the data to be processed.

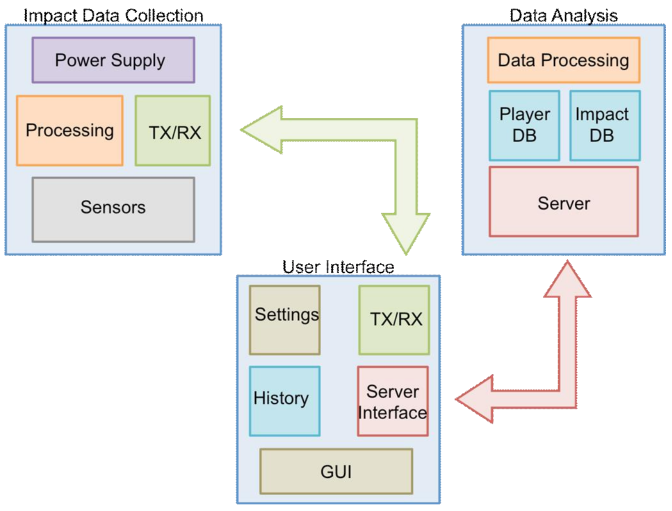

Fig. 1. This RCA Block Diagram shows the organization of the project.

2) Data Analysis: This block is responsible for the calculations of risk, as well as storing and receiving the raw data and player information. It communicates with the Android device and the backend server. By having the data analysis preformed on the Android device, we minimized the processing done in the Impact Data Collection block. This leads to a low-profile system, keeping the increase in helmet weight minimal.

3) User Interface: The user interface is an Android device. Here, the coach can input player rosters, retrieve information about the impacts a player has sustained, and receive real-time alerts with the probability that a player has suffered a concussion. This block communicates with other blocks via Bluetooth and the Internet.

By utilizing Bluetooth and Android devices we can keep the cost to a minimum, as most users will have a capable Android device. These three main blocks are then divided among the team into four sub-blocks.

B. Sensor Network and Power Supply

This sub-block of the Impact Data Collection block is concerned with measuring forces and powering the electrical components in the helmet. Sensors were placed inside the helmet as close to the player's head as possible, for the most accurate measurements. From research at Virginia Tech [15], the maximum linear acceleration found that a player experienced over the two year study was slightly less than 200 gôs. For our prototype we will not be inflicting such high g forces. Our initial sensors were the ADXL 193 [10] which had a tolerance of +/- 250 g, but due to the scalability, we decided to use more sensitive sensors. The sensors being used are micro-electro-mechanical systems (MEMS) accelerometers. The ADXL78 [6] is a low powered, single-axis, MEMS accelerometer, with a tolerance of +/- 70 g, and a sensitivity of 27 mV/g.

We spoke with an applications engineer from Analog Devices, the company that manufactures our sensors, and confirmed that each sensor needed to be tested for its own sensitivity value. The sensor output was measured for two orientations. From the output, we directly calculated the sensitivity each sensor. Using the data sheet for the sensors [6], we determined the correct orientation to provide the most accurate results; the pin for Vdd is the axis of measurement. Our initial Vdd was 5046.2 mV. We placed each sensor so Vdd was pointing up, recorded its output, place the sensor so Vdd was pointing down, and recorded its output. Using Equation (1), we were able to characterize each sensorǒs sensitivity. Table II, shows our results for each sensor.

$$Sensitivity\ (mV/g) = \frac{(Vdd\ Down - Vdd\ Up)}{2} \quad (1)$$

TABLE II
CHARACTERIZATION OF SENSORS SENSITIVITY

Sensor	Vdd Up (mV)	Vdd Down (mV)	Sensitivity (mV/g)
0	2441.9	2497.1	27.6
1	2467.2	2522.2	27.5
2	2485.3	2540.6	27.6
3	2479.0	2534.2	27.6
4	2464.1	2519.7	27.8
5	2480.2	2535.4	27.6

With the sensors partially characterized, we strategically placed them in the helmet to capture the x-axis, y-axis, and z-axis of a player's head; the axis of measurement is normal to the center of gravity. The placement of the sensors came from Joseph Criscoǒs paper on locating the impact [17]. They explained where their sensors were placed, theta and alpha, and what angles gave the least amount of error. To make our packaging easier, some sensors were moved by a few degrees.

TABLE III
PLACEMENT OF THE SENSORS

Sensor	Theta (ɗ$_H$)	Alpha (Ŭ$_H$)
0	0	20
1	-90	15
2	-180	20
3	90	15
4	75	50
5	-69	50

Table III and Fig.2 show the final placement of each sensor in our helmet in terms of theta and alpha.

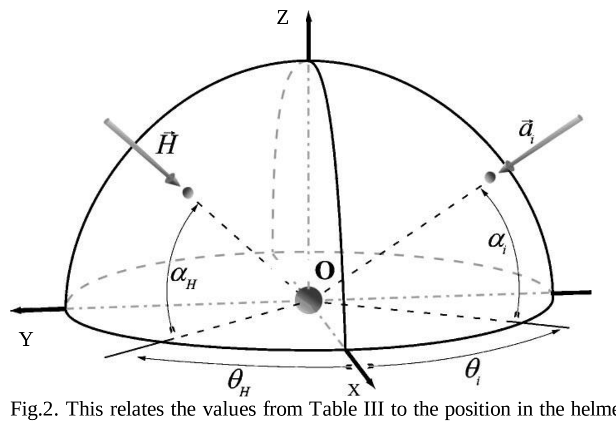

Fig.2. This relates the values from Table III to the position in the helmet; x-axis is back to front, y-axis is left to right, z-axis runs up the spine to top.

To finish characterizing the sensors we conducted another test to focus on response time of the sensor, primarily the time to peak acceleration. Throughout the test, we saw how the sensor responded to multiple impacts of the same force, impacts from different directions, and what kind of realistic forces we can apply on demo day. Performing the same test multiple times also proves the sensor is functioning properly. Fig. 3 below shows a realistic impact reading from the oscilloscope, the peak acceleration happens within 2 ms.

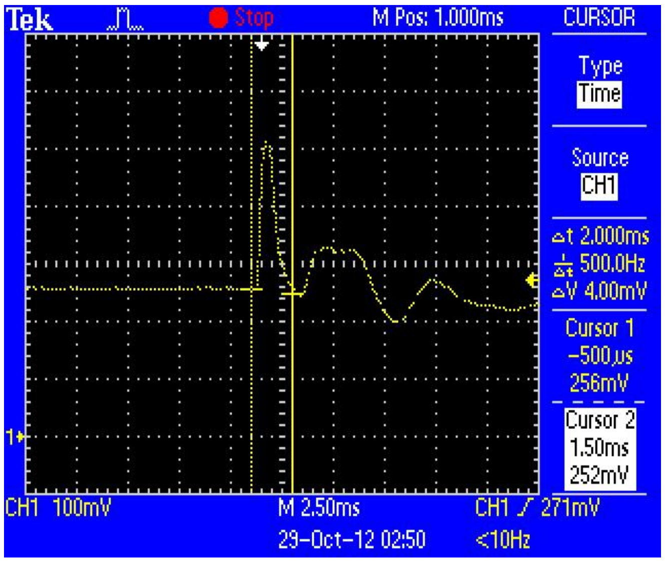

Fig. 3. This shows the peak acceleration of ADXL78.

The test we conducted utilized a basic pendulum to generate an impact of known magnitude. The sensor was placed, in a helmet, so it measured in the direction of the swinging pendulum. The pendulum impacted the helmet giving us a measurement of the impact. This measurement was sent to the processor and transmitted to a computer for analysis, which showed that we can repeatedly inflict an impact with the same magnitude. Once we understood how one sensor worked we built the network of sensors to go in the helmet.

The other aspect of this block deals with powering the system in the helmet. We conducted a power analysis to find the total energy our system consumed. Using the data sheet of every component in the helmet, we calculated the worst case power. Then we converted the power into Joules. From Joules, we were able to find the total mAh that our system uses. Table IV below shows the worst case power analysis of our system. Most batteries are rated in mAh, so knowing that value helped to determine a type of battery to use; our initial thought was to use a coin cell battery because of it small

TABLE IV
WORST CASE POWER ANALYSIS OF OUR SYSTEM

Device	Voltage (V)	Current (mA)	Power (W)	
ATmega32	5	200	1	
Bluetooth	3.3	100	.33	
ADXL 78	5	1.3	.039 (6 sensors)	
			Total Power (W)	1.369
			mAh Needed	1369

weight and size. Unfortunately, the coin cells could not output enough current for our system to function properly, so we began looking into other options.

We found a USB battery pack [18] that has 2000 mAh, outputs 700 mA peak at 5.5 V, and has an added bonus of being rechargeable. To be sure this battery would work; we analyzed our systemôs typical energy consumption by measuring the changing voltage from a Laboratory DC power supply across a small resistor in series with our system. Using Ohms law we found the current, calculated power, and converted to energy.

Both the worst case power analysis and typical power analysis proved the battery would work, so we implemented it into our system. As a final test for the battery, we powered the helmet for more than five hours with the battery. During this time we measured Vdd, to be sure there was no fluctuation, then transmitted data, to be sure the correct data was being sent from an impact. We took these measurements every 15 minutes. After five hours, Vdd remained constant and the data was still being sent correctly. With the sensors characterized and integrated into the helmet and the battery for the system working properly, the focus of this sub-block can now shift to final helmet packaging of the Impact Data Collection block.

C. Impact Processing and Communication

As a sub-block of the Impact Data Collection block, the impact processing and communication block is responsible for detecting an impact and then transmitting the sensor data to the User Interface block. At the heart of this sub-block is a microcontroller that is used to process the incoming signals from the accelerometers described above. Once the signals have been processed by the microcontroller they will be either discarded, or in the case of an impact, they are serially transmitted to a Bluetooth radio. This radio communicates with the impact data to the user interface.

The microcontroller selected for this block is the ATmega32U4 by ATMEL [7]. This microcontroller was selected for RCA after careful review of its features. There are six analog signals coming from the sensor array that will need to be converted to a digital signal for transmission. This microcontroller has 12, 10-bit ADC (analog to digital converter) channels, with six being used for the ADC conversions of the sensor signals. The ATmega32U4 also has a programmable serial USART, which allows the communication between the microcontroller and the Bluetooth radio. This microcontroller also has 2.5 kB SRAM, which can be used to buffer the impact data before transmitting. All this is in a TQFP package, which allows us to have a small circuit to fit inside the helmet.

The programming of this microcontroller in C is familiar, as we have use a similar 8-bit AVR microcontroller, the ATmega32, in previous course work. There are many features of this microcontroller that we utilize for RCA: the ADC, USART, Timers, and Interrupts. While we had previous course work, we found online tutorials essential in configuring some elements of the ATmega32U4 [11], [12]. We use an external 16 MHz clock so that we can utilize the fastest prescaler of the ADC. To increase throughput, the granularity of the ADC was set to have 8-bits of resolution, thus the outputs range from 0-255. With 5 V supply, each incremental value of the ADC output carries a weight of 17.6 mV. With the sensitivity of the chosen accelerometer, 27 mV/g, this leads to a sample granularity of 0.7 gôs per ADC value.

Initially the design for communication was going to be XBee radios, for the full range of 100 m, but due to our decision to have a scaled down prototype we decided to use Bluetooth. The Bluetooth radio selected for this block is the BlueSMiRF Gold by Sparkfun [13], which utilizes the Roving Networks RN-41 Bluetooth module [8]. This Bluetooth radio is a class one Bluetooth device with an advertised range of approximately 100 m. We found this range to be quite exaggerated once the device was configured, and could only connect at a maximum range of 40 m. This radio is Bluetooth version 2.1+EDR with built in error correction for the 8-bit packet transmission and 128-bit encryption. This module has an advertised maximum data rate of 240 Kbps.

To interface this radio with the microcontroller, we first had to initialize the USART feature within the ATmega32U4. Once we configured the Bluetooth radio and microcontroller, we tried to read the serial stream on a laptop terminal window. We could communicate between the microcontroller and the laptop over Bluetooth, although the symbols were being distorted. To fix this issue, the baud rates on both devices were synchronized at 9600.

Once we completed the basics of this sub-block, an experiment was conducted to determine the accuracy to which the user can receive the sensor data. The output of the sensor was measured directly with an oscilloscope. Simultaneously, the sensor data was recorded on the laptop after the Bluetooth transmission. From this experiment, we were able to learn that our sample rate was too low, 166.7 Hz. We had conducted measurements of the sensor directly with an oscilloscope and found that the average response time to reach peak acceleration was approximately 2.5 ms, 400 Hz. This corresponds with the specification in the accelerometers data sheet for the 2-pole Bessel filter at the output, stating that the 3 dB cutoff is at 400 Hz. By enabling double speed operation of the USART, optimizing the baud settings of the ATmega32U4 and the BlueSMiRF, buffering the data before transmitting, and modifying the configuration of the ADC code, we were able to achieve a sample rate of 1.5 kHz for each sensor. This sample rate is above the Nyquist rate of each sensor.

Upon completion, Impact Processing and Communication sub-block was tested to ensure accurate sampling. The initial results from our experiments using the sensor proved to be misleading, as they are not true sinusoids and do not have predictable amplitudes. To fully test this block, RCA was removed from the helmet and sensor 0 was replaced with a sine wave of 333 Hz, 2V peak-to-peak amplitude with a 1V offset from a function generator. Similarly, sensor 1 was also removed and replaced with a square wave of the same parameters as the sine. This frequency was chosen as it represents the typical impact duration measured with a single sensor and the oscilloscope.

This experiment consisted of sampling ADC channels 0 and 1 to try to accurately sample the square and sine waves respectively. It can clearly be seen from the initial results that when sampling 2 channels, each at 4.5 kHz, the system could not accurately sample; in fact the two functions were being combined, see Fig.4.

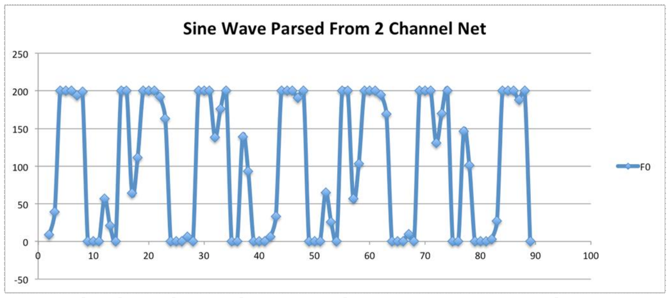

Fig. 4. This shows the initial ADC sampling a sine wave incorrectly.

To solve this issue we had to input even simpler functions than the sine and square wave. We used a 3V DC signal on channel 0 and a 1V DC signal on channel 1. Even these values would sometimes be sampled incorrectly, in a repeating manor. This led us to research more into the sample and hold times of the ADC. The data sheet says that the ADC takes 13 clock cycles to complete a conversion. However, after careful review of this section, we found that 13 clock cycles was the ideal time but there was a variance to this time that the data sheet did not have quantified.

The solution to this issue was to rewrite the ADC sampling code to utilize a more efficient waiting period. By using an interrupt vector in the microcontroller, instead of a while loop, we were able to obtain accurate samples. The interrupt vector triggers only when a conversion is complete, and hence ready to be accurately processed. In that vector the code then buffers that result and starts the next conversion. Again the square wave and sine wave were the only channels being sampled at 4.5 kHz each. This time the samples fit the sine wave. The final test was to sample both the sine and square functions, along with the remaining 4 sensors, effectively reducing the sample rate of each to 1.5 kHz. Again, the sine wave sample points fit the expected curve, see Fig. 5.

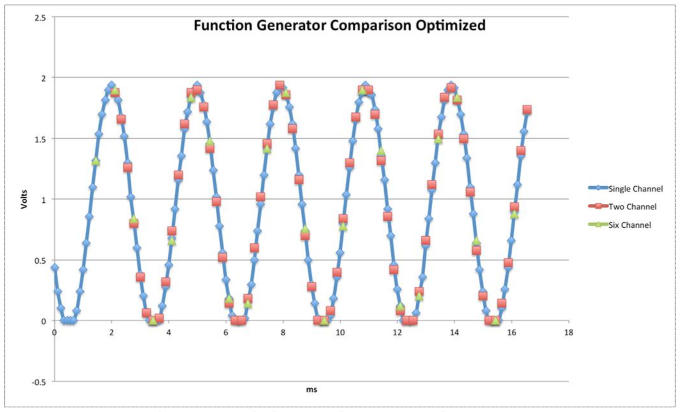

Fig. 5. Sine wave being sampled correctly at 3 sample rates.

After proving the microcontrollers accuracy, the final portion of the code to this block was addressed. The threshold triggering was implemented to avoid continuously streaming data to a device, a necessary feature with the number of players on the field at any time. The threshold for an impact event was selected to be 10 gôs after reviewing the research from Virginia Tech [15]. This is achieved by continuously sampling the ADC and checking the force measured at that time instance. If any sensor samples are at or above this threshold, the microcontroller starts a timer and then samples each sensor in a continuous loop starting with sensor 0, incrementing in order. Each value is buffered within the SRAM until the timer reaches 50 ms, a time chosen based on osciliscope readings from a single sensor impact and verified with research [15]. Once the timer reaches 50 ms, another interrupt vector triggers, in which the buffer is emptied one sample at a time and transmitted in order. All the counters are reset and RCA goes back into monitoring mode to await the next impact.

D.Data Analysis

The purpose of this block is to determine the probability of a concussion for a specific player based on the hits the player has taken. It is broken into two main parts, the database and the risk algorithm. The database stores the raw accelerometer data, player information, and the hit vector **H**. The risk algorithm takes in sampled points from each accelerometer and interpolates on the data points. The algorithm calculates the hit vector at every point and selects the largest magnitude. The algorithm calculates risk using the resultant vector which it stores in the database. Lastly, the algorithm queries the database for hit vectors to calculate the cumulative risk. Once the risk algorithm was written, it was tested using black box testing methods. The data processing and storage, acts as the backbone for the Android application, which displays the risk of a concussion for each hit and the cumulative risk.

The database is a MySQL, My Structured Query Language, database that is on a remote server. The database is controlled by PHP scripts, which can connect, read, and write to the database [14]. The risk algorithm uses the PHP scripts to store the hit vector, player name, player age, player number, as well as the raw accelerometer data into the database. The hit vector and raw data tables in the database provide a time stamp for each entry. With all this information stored into the database, the risk algorithm [15] can use this information and professionals can see data to help diagnose a concussion.

To continue developing the risk algorithm, we needed a way to correlate linear acceleration to probability of concussion. We found a study [15] done by Steve Rowson, a professor of the School of Biomedical Engineering & Sciences at Virginia Tech. Rowson performed a study on concussive impacts by equipping 314 collegiate playersô helmets with the HIT System, which consists of six accelerometers and calculates a resultant linear head acceleration at the CG, center of gravity, of the head. Also, 21 playersô helmets were equipped with the 6DOF, Six Degrees of Freedom, system that had twelve accelerometers instead of six. The 6DOF is able to measure linear and rotational acceleration. The study had monitored players between 2007 and 2009 and noted when players were diagnosed with a concussion [15].

In this study an equation was derived that converts linear acceleration to rotational acceleration from the equations of motion modeling force acting on a head [Equation (2)].

$$\alpha = \frac{m\sqrt{ax^2 + ay^2}}{I}d \qquad (2)$$

In Equation (2), Ŭis rotational acceleration, m is the mass of the head, ax is the peak acceleration along the anterior-posterior axis of the head, ay is the peak acceleration along the medial-lateral axis of the head, I is the moment of inertia of the head, and d is the perpendicular distance from the head CG to the impact vector [15]. The unknown variables of m, d, and I were determined through a regression model analysis of recorded 6DOF acceleration data and confirmed with laboratory validation experiments. A least squares technique was used to equate (m*d/I) to 6.48 m^{-1} [15].

The study also developed an equation that correlated rotational acceleration to risk. To develop the risk equation, the study performed a statistical analysis on occurrences of sub-concussive impacts and concussive impacts that were reported. A logistic regression analysis based on weighted sub-concussive and concussive head acceleration distributions were used to express risk as a function of rotational head acceleration [Equation (3)] [15].

$$\text{risk} = \frac{1}{1 + e^{-(c_1 + c_2\alpha)}} \qquad (3)$$

In Equation (3), Ŭ is rotational acceleration; c_1 and c_2 are regression coefficients. The regression coefficients were determined using a generalized linear model technique. The c_1 coefficient was calculated to be -12.531, and the c_2 coefficient was calculated to be 0.002 [15].

After reviewing his study and speaking with Steven Rowson over the phone, the risk algorithm was developed to model the equations derived in the study. First, the algorithm must calculate the hit vector that is used to find ax and ay from Equation (2). The algorithm receives the raw accelerometer data and has to interpolate each sensor graph. In order to do this, the algorithm uses a sliding quadratic interpolation to create an accelerometer graph from every three accelerometer sample points. The algorithm then calculates the hit vector at every time interval along the accelerometer graphs [Equation (4)].

$$0 = \sum_{i=1}^{n} \left(\|H\|\left(\cos(\alpha_i)\cos(\alpha_H)\cos(\theta_i - \theta_H) + \sin(\alpha_i)\sin(\alpha_H)\right) - \|a_i\|\right)^2 \qquad (4)$$

Equation (4) is a least squares model derived to calculate the magnitude, alpha and theta, of the hit vector from multiple nonorthogonal single axis accelerometers [17]. In Equation (4), alpha is the angle of elevation and theta is the angle of azimuth. The coefficients in Equation (4) with a subscript ñiò represent components of an accelerometer, and the coefficients with a subscript ñHò represent components of the hit vector. To perform this calculation, the algorithm breaks the equation into known and unknown variables to create three different matrices. The first matrix, C, is composed of the parts in Equation (4) involving locations of each accelerometer. The second matrix, X, is composed of the parts in Equation (4) involving the hit vector components. The third matrix, A, is composed of the value for each accelerometer. The algorithm computes the solutions for matrix X using matrix C and matrix A [Equation (5)].

$$pinv\left(C^T\right) * A = \begin{matrix} a \\ b \\ c \end{matrix} \qquad (5)$$

The algorithm uses the solutions of matrix X, a, b, and c, to solve the hit vector components in matrix X [See Equation (6)].

$$\begin{matrix} H\cos(\alpha_H)\cos\theta_H \,|\, a \\ H\cos(\alpha_H)\sin\theta_H \,|\, b \\ H\sin\alpha_H \,|\, c \end{matrix} \qquad (6)$$

After the hit vector is calculated at every time interval, the correct time to calculate the hit vector is selected. The algorithm looks for the time when the magnitude of the hit vector is maximized to select the corrected hit vector data. Once the hit vector is selected, the x and y components are calculated from the hit vector. The x and y components are inserted into ax and ay in Equation (2), respectively. The rotational acceleration produced from this data is then used to calculate risk with Equation (3).

When the user requests the cumulative risk, the algorithm queries the database for all of the peak linear accelerations for that player. The algorithm then converts the hit vector that was returned from the query into a rotational acceleration [Equation (2)]. The rotational acceleration is then used to calculate a risk [Equation (3)]. The risk algorithm then uses a weighted average to return the cumulative risk for the user interface code to display. This average of risk is currently our way to incorporate the possible dangers of multiple sub-concussive impacts; this will be subjected to change with future research.

The entire risk algorithm was black box tested by developing a Java test program that would parse input text files containing simulated impact data, clear the database, and run the impact data through the risk algorithm. The Java test program then writes the rotational accelerations and risks to output text files. A python script was used to generate this simulated impact data. Another python script was written to parse the output text files, store the calculated rotational accelerations, and calculated risks into an excel file. The script also stores the expected rotational accelerations and risks into the excel file. The results of this test were graphed; they matched the graph of the risk function from Rowson's study [Equation (3)].

E. User Interface and Communication Block

This block's purpose is to display concussion data to the user and allow user input to control the application and system settings. The two main parts of this block are the Android application and the application's interface with the Android device's internal Bluetooth module. Currently, the ability for the application to receive a full data set and analyze the information is operational. The application then displays the impact's magnitude, location, and risk of injury in a pop-up notification that is shown to the user, shown in Fig.6. The application uses the default display as its idle screen, and displays the last few hits that were detected. The pop-up occurs on the idle screen, which gives the user the option to inspect the impact information in depth. In order to wrap up this block of the project, a method to accept user settings and a visual improvement of the application will be executed.

The application is written using Java in the Android programming language, utilizing the Android Development Tools plug-in for Eclipse IDE. For programming, we took advantage of prior coding experience and the development tools available to write the application. The application currently has six main activities that function together to handle the data and provide meaningful interpretations. First is the main screen activity, where the user can connect and disconnect to the Bluetooth module, housed in the helmet, display previous hit history, and monitor for newly detected impacts.

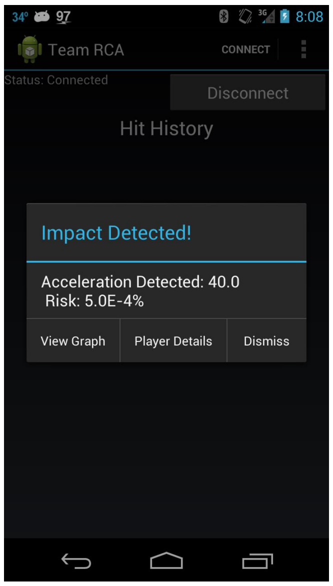
34° 97 8:08
Team RCA CONNECT
Status: Connected
Disconnect
Hit History
Impact Detected!
Acceleration Detected: 40.0
Risk: 5.0E-4%
View Graph | Player Details | Dismiss

Fig. 6. Impact notification screen

The next activity is the graph view, which allows the user to view a plotted graph of the received acceleration data for each of the six accelerometers. This graph activity for a single accelerometer, shown in Fig. 7. comes from an open source graphing library that is available for use in Android development.

Another activity in the application is the cumulative hit histogram display. This activity can be displayed upon the user's request, and can show a histogram of hits for each player. The histogram was recommended by staff of the University's Athletic Department as a tool to observe the effects and trends as a result of multiple hits at varying intensities. The cumulative risk activity, which retrieves the impact data from the server and shows a comprehensive analysis of the player, displays their impact history, and cumulative risk of concussion. It is from this display that a user can make a call to the histogram activity described above to see information of multiple hits.

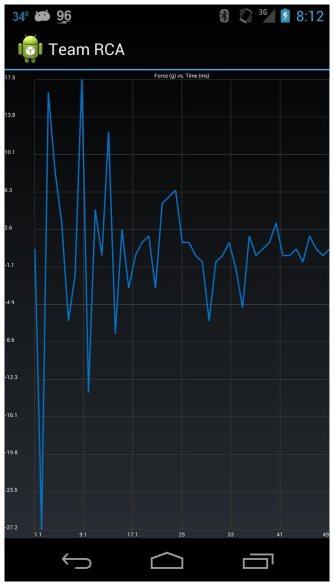

Fig. 7. Accelerometer graph

The user settings screen is a simple but important piece to the application that allows the user to set and change different options within the application, such as a coach vs. trainer view. Throughout the development of the application many bugs were fixed that caused reliability issues and forced the app the close. The functionality and reliability of the application has been improved greatly and the final focus of the application will be on improving its ease of use and aesthetic appearance. Finally, the application's Bluetooth interface was created in reference to the tutorial on Android device and Arduino Bluetooth Communication [16]. This tutorial provided the necessary code examples to detect, connect, and disconnect an Android to an Arduino. We were able to adapt this code to connect to our ATmega microcontroller and interface it with its USART capabilities.

As we tested the Bluetooth module for the distance, as outlined earlier in the document, we also took that opportunity to test the robustness of the Bluetooth data transfer itself. We observed that at distances before losing connection, we saw a significant drop in the rate in which data was being acquired. The limitation of the Bluetooth device has proven to be the main factor in restricting the range that our system will perform. The response time of the data transfer across the Bluetooth and through the data processing unit was measured to characterize the system response time. The average response time was measured to be X ms, which describes the time it takes for the application to receive all data point via Bluetooth and perform the necessary computations to make that data meaningful and display it. This response is acceptable and if we make some adjustments to the code for efficiency, it will be sufficient time to warn a coach and remove a player before they begin another play.

III. Project Management

Utilizing effective tools, such as timelines, open lines of communication, and a Gantt chart, the Impact Data Collection Block is functioning properly. An open line of communication is proving to be are most important tool as we troubleshoot the Data Analysis Block.

The team members of RCA compliment each other very well, and there is a great group dynamic. Every member brings something unique to the table and all help each other out and try to become involved with every facet of the project. On more than one occasion a team member may have been at a low point in their workload and has helped offset anotherôs. We have regular team meetings once per week, and one team meeting with our advisor, Professor Hollot per week. Scott Rosa is the CSE of the group and is the data processing and server expert, as well as maintains the website. Kenneth Van Tassell is the Android device programmer of the group, and working on the wireless communications on the Android device end. Justin Kober is responsible for the sensor network and power for the Impact Data Collection block, as well as validation testing of the project. Tim Coyle is responsible for the microcontroller and wireless transmission of data to the Android device, and is also the team manager.

TABLEV
RCA VS. COMPETITORS

Specification	RCA	HITS
Imbedded Sensor Network in Helmet	Yes	Yes
Simple User Interface (Android)	Yes	No
Accurate Data Collection Transmission	Yes	Yes
Reliable Data Analysis	Partial	Yes
Real-time Solution (alerts)	Yes	No
Two-way Communication Capable	Yes	No
Cost Efficient	Yes	No
Full Range of Field	No	Yes
Waterproof	No	Yes

IV. CONCLUSION

Since MDR, we have integrated the sensor array into the helmet, which is now sampled by the microcontroller and powered by the battery. The sample rate has been optimized and threshold triggering has been implemented. The user interface will need to be configured to input players and recall their information at a userôs request. The possibility for two-way communication between the Android device and helmet exists, but for this iteration of RCA it will not be implemented due to the project deadline and debugging issues.

The majority of the project is completed, though through integration we have found issues that need to be resolved. The Impact Data Collection Block is constructed and produces accurate results; all that remains is final packaging. The User Interface Block has the basic GUI design and underlying structure to interface with the other blocks. The application will be refined and have other menus and options added for a better user experience. The Data Analysis Block has been implemented with an updated algorithm to increase speed but needs optimization for accurate results.

Currently, upon impact, RCA transmits the data to the Android device where the user will see the correctly measured force and corresponding risk. Between now and Demo Day we will be fixing the data analysis issues, characterizing the system error, revising the Android application, and conducting validation testing of our complete system. We plan on reaching all of our goals for RCA by continuing to stay ahead of deadlines and keep communication lines open with the team and advisor.

APPENDIX

A. Application of Engineering

There are many areas of math, science, and engineering that apply to RCA, most notably: data structures and algorithms, classical mechanics, circuit analysis, electronics, computer networks, hardware organization, communications, and signal processing. For our software development portion of RCA the server, databases and Android development were all done in Java, while the software for the microcontroller and Bluetooth control are written in C. We have exposure to these programming languages through the coursework in ECE 122, ECE 242, ECE 353 and ECE 354. The sensor network and the power regulation circuitry design were both essential in our data collection block. These elements of RCA would not have been implemented correctly had we not had previous knowledge of circuit design, which we gained in courses like ECE 211, ECE 212, ECE 323 and ECE 324. Courses like ECE 313, ECE 314, ECE 333, ECE 374 and ECE 563 helped us to better understand the fundamentals needed for successful wireless communication.

B. RCA Cost

Below is our initial cost analysis for the project. All of the costs are in the helmet network, as this system assumes the user already has an android device.

TABLE VI
HELMET NETWORK COST

Device	Model	Unit Cost	Total	
Accelerometer	ADXL78	$5.58	$33.48	analog cost at 1000ct
Microcontroller	ATmega32U4	$5.56	$5.56	sparkfun cost at 100ct
BlueTooth Modem	BlueSMiRF Gold	$51.96	$51.96	sparkfun cost at 100ct
BlueTooth Module	*RN-41*	*$19.96*		sparkfun cost at 100ct
PCB		$33.00	$4.13	4PCB.com 60 sq in
Estimated Costs				
Misc Hardware	Caps, Clock, Resistors		$1.80	DigiKey
Total Cost		**$96.93**		
X 52 Players				
= Helmet Network Cost Per Team		**$5,040.10**		

C. Packaging

The most important aspect of packaging our system into a helmet was making the system ñinvisibleò to player wearing the helmet. We wanted to keep the helmet as normal as possible so we used the padding and open space to house our system. Wherever the sensors needed to be, we inserted them into a pad so Vdd was pointing out the back of the pad. The pad provides protection from the force of impact and stops the sensors from contacting the playerõs head. The pads will be sealed with glue to hold them in place and keep out water near the end of the project. The battery has its own case to provide protection from the force of impact. To install it in the helmet we cut a slit in a pad, the width of the battery, and inserted the battery to stop it from contacting the playerõs head. The last piece of equipment in the helmet is the PCB. We wrapped this in foam and inserted it into open space between the pads in the helmet. We sealed it with electrical tape to keep out as much water as possible; with more time we could completely seal the system and make it waterproof.

D. Weight Analysis

With our system in the helmet, we did not want to increase the overall weight by more than 5%. We used a scale and measured the original helmet and then the helmet with our system in it. Using this we could characterize the weight of our helmet. See table VII below for the different weights and overall weight increase to helmet. The battery and its case is what puts us over our requirement specification, but with more time we could implement the guts of the battery and shed the weight of its case. With the case gone we would meet our specification.

TABLE VII
WEIGHT ANALYSIS OF RIDDELL SPEED HELMET

	Weight (g)	
Helmet w/out our system	1927	Official weight from Riddell
Sensors, battery, protoboard	120	
Helmet w/our system	2047	
Battery unit	75	
Empty battery case	30	
Components in the case	45	
Percent increase (with case)		6.2% (1.2% over spec)
Percent increase (without case)		4.7% (spec met)

ACKNOWLEDGMENTS

Team RCA would like to thank Holyoke Catholic High School and the University of Massachusetts Amherst Football team for supplying football equipment to our project. We would also like to thank Professor Steven Rowson for allowing us to use his risk function and speaking with us about our project and his research.

REFERENCES

[1] CDC Injury Prevention & Control: Traumatic Brain Injury. (2012, January 30). *Concussion and Mild TBI* [Online]. Available: http://www.cdc.gov/concussion/ [Accessed Web. 17 Nov. 2012.]

[2] Health Day News. (2012, January 30). *Many High School Football Players Ignore Signs of Concussion: Survey* [Online]. Available: http://consumer.healthday.com/Article.asp?AID=669798 [Accessed Web. 5 Dec. 2012.]

[3] McCrea, M., T. Hammeke, G. Olsen, P. Leo, and K. Guskiewicz. *Unreported concussion in high school football players: implications for prevention.* Clin. J. Sport Med. 14:13ï 17, 2004.

[4] B. Wilner. (2012, June 7). *NFL Concussions Mega-Lawsuit Claims League Hid Brain Injury Links From Players* [Online]. Available: http://www.huffingtonpost.com/2012/06/07/nfl-concussion-brain-trauma-lawsuit-players_n_1577497.html [Accessed Web. 17 Nov. 2012.]

[5] System and method for measuring the linear and rotational acceleration of a body part, by J.J. Crisco, III and R.M. Greenwald. (2001, Oct 10). *Patent US6826509B2* [Online]. Available: http://www.freepatentsonline.com/6826509.pdf [Accessed Web. 30 Sept. 2012.]

[6] Analog Devices, "ADXL78 Datasheet and Product Info.,*"ADXL78: Â Single-Axis, High-g, IMEMSÂ® Accelerometers.* [Online]. Available: http://www.analog.com/en/mems-sensors/mems-accelerometers/adxl78/products/product.html. [Accessed Web. 17 Nov. 2012.]

[7] ATMEL. (2010, Nov.). ATmega16U4/32U4 Preliminary Datasheet. [Online]. Available: http://www.atmel.com/Images/doc7766.pdf [Accessed Web. 6 Oct. 2012.]

[8] Roving Networks. (2012, Oct.). RN-41 Datasheet. [Online]. Available: http://www.rovingnetworks.com/resources/download/18/RN_41 [Accessed Web. 16 Oct. 2012.]

[9] IDC ï Press Release. (2012, November 1). *Android device Marks Fourth Anniversary Since Launch with 75.0% Market Share in Third Quarter, According to IDC* [Online]. Available: http://www.idc.com/getdoc.jsp?containerId=prUS23771812 [Accessed Web. 17 Nov. 2012.]

[10] Analog Devices, "ADXL193 Datasheet and Product Info.,*"ADXL193: Â Single-Axis, High-g, IMEMSÂ® Accelerometers.* [Online]. Available: http://www.analog.com/en/mems-sensors/mems-accelerometers/adxl193/products/product.html. [Accessed Web. 17 Nov. 2012.]

[11] P. Hood-Daniel. (2012, October 10). Microcontroller Tutorial. [Online]. Available: http://newbiehack.com/MicrocontrollerTutorial.aspx [Accessed Web. 20 Oct. 2012.]

[12] HeKilledMyWire. (2011, January 5). Using the USART/serial. [Online]. Available: http://hekilledmywire.wordpress.com/2011/01/05/using-the-usartserial-tutorial-part-2/#more-31 [Accessed Web. 26 Oct. 2012.]

[13] Sparkfun. (2011, Nov.). Bluetooth Mode ï BlueSMiRF Gold. [Online]. Available: https://www.sparkfun.com/products/10268 [Accessed Web. 16 Oct. 2012.]

[14] Tamada, R. (2011). *How to Connect Android device with PHP, MySQL* [online]. Available: http://www.Android devicehive.info/2012/05/how-to-connect-Android device-with-php-mysql/ [Accessed Web. 16 Nov. 2012.]

[15] Rowson, S, et al. ñRotational Head Kinematics in Football Impacts: An Injury Risk Function for Concussion,ò *Annals of Biomedical Engineering*, vol. 40, no. 1, pp. 1-13, Jan. 2012. [Accessed Web. 21 Sept. 2012.]

[16] Bell, M. (2012). *Android device and Arduino Bluetooth Communication* [online]. Available: http://bellcode.wordpress.com/2012/01/02/Android device-and-arduino-bluetooth-communication/ [Accessed Web. 16 Nov. 2012.]

[17] Crisco, J, et al. ñAn Algorithm for Estimating Acceleration Magnitude and Impact Location Using Multiple Nonorthogonal Single-Axis Accelerometers,ò *Journal of Biomechanical Engineering*, vol.126, no. 1, pp. 1-6, Dec. 2012. [Accessed Web. 3 Feb. 2013.]

[18] Sparkfun. (2012, Nov.). USB Battery Pack ï 2000maH. [Online]. Available: https://www.sparkfun.com/products/11359 [Accessed Web. 4 Feb. 2013.]