Wireless Accelerometer for Stroke Mobility Data Acquisition using Bluetooth and IEEE 802.15.4

Scott E. T. Hadley
York University
seth@cse.yorku.ca

Abstract

Mobility research for stroke patients using Bluetooth (IEEE 802.15.1) sensors is compared with similar ZigBee (IEEE 802.15.4) devices. Two body sensor networks are established to collect acceleration data signals. Based on certain criteria, key factors are measured and compared based on relevance to stroke research. These factors include sensor life, range, sampling frequency, 2.4 GHz ISM interference and others. The power consumption of Bluetooth maintains a strong connection well suited for ISM interference and collects data at 100 Hz from four Bluetooth sensors located on a patient at the cost of sensor lifetime. The ZigBee sensor needs to manage interference in the busy ISM band and with a lower data throughput needs compression to meet stroke research criteria. The low power shown by ZigBee devices does point to an extended sensor lifetime. Further signal processing should solve acquisition issues and this research justifies ZigBee sensors for future research purposes.

1. Introduction

In June 2008, the author was hired by Toronto Rehabilitation Institute (TRI) to write an application for a body sensor network (BSN) of wireless accelerometers used by researchers of stroke rehabilitation. They employed the Bluetooth WiTilt from Sparkfun and decided to explore other available sensors. The ZigBee IMote2 sensor (IMote) from Crossbow Technologies was selected as a possible candidate for this study. While the WiTilt has proven successful as a research tool, future research will require devices to collect data for longer periods of time, not possible with the WiTilt. ZigBee, or IEEE 802.15.4, has a low-power design that should solve this longevity problem. Also, future data collection may occur in a home environment and besides a lifetime exceeding twelve hours, will the ZigBee BSN collect signals from four devices at 100 Hz?

This study will compare the two systems and evaluate the IMote as a replacement for the WiTilt. This will be accomplished by the proposition of four hypotheses:

1. The IMote will last twelve hours.
2. The range of the IMote will be greater than 10 meters.
3. The IMote will produce a sufficient sampling frequency, > 200 Hz.
4. The BSN will operate collecting data from four devices at 100 Hz.

Three key factors were studied to test these hypotheses: power usage, sampling frequency and sensor range. These factors are considered the most important in data collection for stroke research. Three further factors are measured and compared: cost/complexity, quality of measured data and latency for future reference and are not as significant.

2. Motivation

William McIlroy, University of Waterloo, and William H. Gage, York University research stroke rehabilitation at TRI specializing in the collection of acceleration data from wireless sensors. These sensors record the activity of patients outside of structured therapy since it is likely to have a profound influence on their recovery [1].

TRI constructed a BSN from a collection of WiTilts, a commercial wireless Bluetooth accelerometer from Sparkfun, and developed an application in Labview executed on a PocketPC, the sink for the data. It was decided to re-implement an application in Java due to issues with maintainability, discovering of devices and lost connections. An implementation in Java allowed it to run on many operating systems, including Windows Mobile for the PocketPC.

The short lifetime of the WiTilt required other sensors to be evaluated. ZigBee offered a sensor solution for a BSN with a low-power, low-complexity design. It does not have frequency hopping, a feature of Bluetooth, to handle interference but consumes less power. It does offer a deep sleep mode to conserver power when the sensor inactivity is allowed. The lifetime of ZigBee devices can be measured in years but in this scenario the IMote is constantly sampling at 100 Hz and will reduce the lifetime to hours.

A comparative study of the two sensors was suggested with the direction of Gage and A.W. Eckford of York University and funding from a NSERC grant allowed the purchase of the IMote equipment. A BSN based on these devices was constructed to test the capacities of the IMote and compare the two BSNs as platforms for acceleration data collection. This study will summarize with a determination of the suitability of the ZigBee as a replacement for Bluetooth as the basis of the BSN.

3. BSN Development

As discussed, the WiTilt BSN was developed before this study was contemplated and its design will be the basis for an IMote BSN [2]. In other words, the IMote implementation will attempt to replicate the behavior of the WiTilt BSN as closely as possible. This section describes the development of the BSNs required to collect the acceleration signals for research purposes.

Two fundamental differences are allowed to simplify the IMote BSN. First, the collection will be a desktop or laptop, where the WiTilt has the option of a PocketPC. It is assumed moving to a mobile platform only increases complexity and not the ability of the network. Second, the IMote will not perform re-transmission of lost packets but this is a feature of Bluetooth. A comparison of range will compensate by equating a packet success rate of 0.70 with the IMote with the loss of the Bluetooth connection where

$$\mathit{Success\ Rate} = \mathit{Packets}_{Received} \,/\, \mathit{Packets}_{Sent} \qquad (1)$$

3.1 WiTilt

The WiTilt BSN has a data collection application, WiTilt PocketPC Data Application (WPDA), executed on a PDA or Windows XP platform. WPDA, through Bluetooth, establishes connections with surrounding WiTilts and can control their settings, start and stop data streams, and capture and record acceleration data for analysis.

Two important settings of the WiTilt are sampling frequency and data mode. The sampling frequency, or speed, can be set from 10 Hz to a maximum of 350 Hz. Different transmissions of data are binary and raw mode. Binary is most efficient, transmitting the signal with 11 bytes per sample, and raw produces human-readable text with an average of 15 bytes.

Since it is based on Bluetooth, the WiTilt BSN has no packet loss. The Bluetooth Serial Port Profile (SPP) connection established ensures all data is transmitted across the link. When the range is increased past a critical distance, the link will break and results in a loss of the connection. WPDA has the ability to sense this situation and try to reconnect, assuming it is a temporary situation, and resume data collection.

The difficulty of implementing WPDA was increased by the requirement of a PocketPC data logger with its limited resources and operating system, Windows Mobile. The PocketPC ensures Bluetooth connections are not lost during data collection with its BSN since all sensors are within a comfortable range, less than 2 meters. WPDA was specifically designed to work on Windows Mobile and a PocketPC with its li processor and is limit functionality.

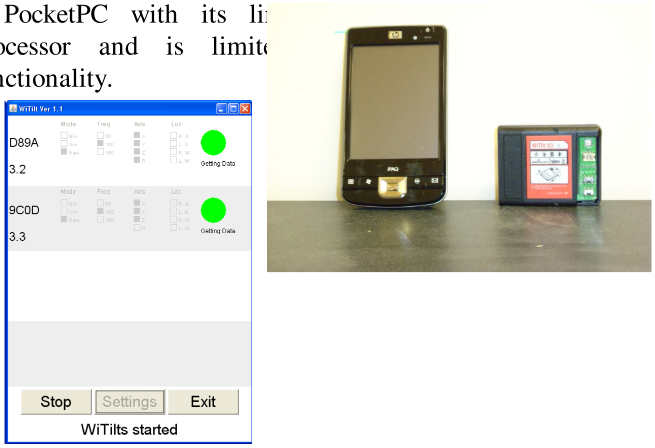


Figure 1: WPDA with PocketPC and WiTilt

WPDA required two open source components. The first component is BlueCove, a JSR 82 implementation of Bluetooth functions. It currently interfaces with many combinations of operating systems and Bluetooth stacks but most importantly, Windows Mobile with the Widcomm stack used on the PocketPC employed by TRI, a HP iPAQ 210. The second required component is a Java Virtual Machine (JVM) for execution of the Java code with Windows Mobile and Mysaifu was chosen, another GPL open source product. It is limited to AWT graphics and reportedly conforms to J2SE.

WPDA connects as a Bluetooth client with up to four WiTilts servers at 100 Hz. The application reads the data stream from each WiTilt and immediately writes this data with an occasional time stamp from WPDA. Another post-processing application, reads the data collected from each device, including the time stamps, and by linear interpolation estimates the time of each acceleration sample allowing synchronization of signals from multiple devices.

For comparative purposes, the only variable, or setting available for the WiTilt is sampling rate. The data mode is restricted to binary mode and sampling three axes since this equates to the proposed IMote firmware. Other settings of the WiTilt are occasionally tested but may not be relevant to stroke research, such as a gyroscope or raw mode.

3.2 IMote

The development of the IMote BSN was more complex compared with the WiTilt system and will be described in more detail. The WiTilt device is pre-programmed and ready to operate out of the box and data can be collected with a Bluetooth connection and a terminal program in minutes. The IMotes required development of the firmware to transmit the signal and a collection program to receive and process them.

The WiTilt BSN has four main functions, discovery of WiTilts, device settings, connection, and data collection. The IMote BSN is much simpler and is implemented with a radio board that sniffs packets and limits its functionality. There will be no discovery or connections made in the Bluetooth sense. The IMote radio channel is pre-selected along with the device firmware or the IMote will not change settings. These deficiencies do not limit this network as a research tool but it does show packet loss and with a sufficient packet success rate can compensate with a retransmission strategy.

The IMotes were purchased as a development kit from Crossbow Technologies and contained three radio boards, two battery boards and two sensor boards to complete two sensors along with a receiving radio. Two additional sensors were purchased for a total of four functioning sensors. Software included with the development kit is Microsoft Visual Studio, Microsoft .NET Micro 2.0 and IMote2.builder SDK. This software allows implementation and flashing of firmware applications. Besides an accelerometer, there are sensors for light, temperature and humidity but not studied.

The majority of developers use TinyOS, but for this report an implementation with the Microsoft products was used. With .NET framework, the applications are relatively easy to implement and flashed through a convenient USB connection. An example solution, *XAccel*, demonstrated an application that essentially replicates the behavior of the WiTilt firmware, sampling at 50 Hz and reports a signal in three dimensions.

3.2.1 IM1: First Application. In the interest of measuring the speed of sampling, the first firmware application written for the IMote is a modified version of *XAccel*. The addition of a packet number, encoded in two bytes, is appended to ten acceleration samples collected and transmitted as a packet. The packet number is useful for determining lost packets.

The following description illustrates the basic behavior of this application.

1. Collect X, Y, Z acceleration samples in packet.
2. Delay for *k* milliseconds.
3. Every tenth sample transmit packet.
4. Increment packet number and repeat 1.

Therefore, the speed of sampling can be adjusted by varying *k*, a designed delay, where *k*=0 represents no delay and the fastest speed possible.

3.2.2 IM2: Second Application. Similar to IM1 but it collects and stores a fixed number of samples and then transmits the data. This removes the delay caused by the transmission and provides an upper limit on the IMote capabilities. It may also be an idea for a long-term design where data would be collected on a device temporarily out of range of a ZigBee network and then downloaded when back in range.

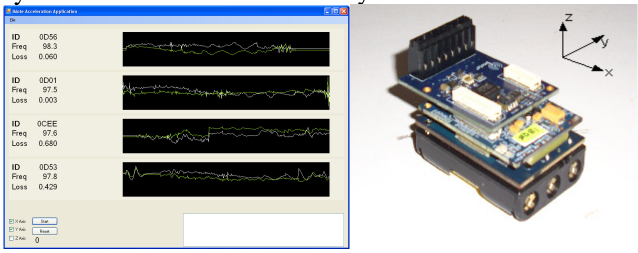
3.2.3 Pendulum Test. To study the behavior of the sensor data a consistent and known acceleration signal was needed. Researchers use "shake tables" but none were available for this project. A signal with a sine wave signal was desired and a pendulum was constructed. It consisted of a large stone paperweight, for mass, and swung from a desk frame. Acceleration data collected from the IM1 and IM2 gave a classic dampening sinusoid primarily in the z-axis signal.

This illustrated a problem with the acceleration data. For IM1 at *k*=5 (126 Hz), it was noted that the sensor values changed only every fourth or fifth sample. The accelerometer sensor appeared to be "sticking". In other words, the processor was working faster then the accelerometer was sampling. The sampling rate was effectively fixed at 40 Hz when $k < 20$.

Research suggested the accelerometer sampling can be varied by altering the decimation factor [3]. This decimation factor can be found in the .NET class *AccelerometerSensor.cs* located in the IMote2.builder libraries. By adjusting initialization of the first register of it was possible to vary the decimation factor [4]. The purpose of the decimation factor is to lower the power resources required by the accelerometer. The designers at Crossbow set it to the highest value and gave the device the lowest power consumption possible.

3.2.4 IM3: Third Application. Similar to IM1, it collects and packages samples with a delay of *k* milliseconds to alter speed but with the ability to changed decimation factor to match the accelerometer and processor. The number of samples in each packet was increased to 14, the maximum allowed. It is more efficient since the same overhead of a packet is constant and lowers the rate of transmission. Two other major modifications were required to fix problems with IM1 and IM2. First, the sampling was uneven or not spaced at regular intervals and corrected by changing the transmission code, shortening its duration. Second, the sensor would freeze after a few minutes, just stop operating, when $k < 15$ and the same revision repaired this issue.

3.2.5 IMote Data Capture. Along with the development kit another application, *SerialDump.exe*, illustrated the capturing of packets transmitted by the IMotes with the additional radio board attached by USB cable to a desktop. This allowed the acceleration data to be sniffed and logged. From this example, IMote GUI Application (IGA) was developed to allow up to four IMotes to be recorded, acting like WPDA. Not limited to a PocketPC, it gives a graphical real-time demonstration of the signals produced by the IMotes. It also records these signals, the sampling frequency and packet loss rate of each device to allow synchronization and further analysis.



ID 0D56
Freq 98.3
Loss 0.060

ID 0D01
Freq 97.5
Loss 0.003

ID 0CEE
Freq 97.6
Loss 0.680

ID 0D53
Freq 97.8
Loss 0.429

☑ X Axis
☑ Y Axis
☐ Z Axis
0

Figure 2: IGA with IMote

4. Methodology

4.1 Power

With the first hypothesis, the determination of the life of a sensor is necessary. The current of each device is measured with a digital multimeter, Gate Crafters DT830-D, and with battery capacity known an estimate of the lifetime can be made.

$$\text{Life (h)} = \text{Capacity (mAh)} / \text{Current (mA)} \qquad (2)$$

The WiTilt and IMote have different battery sources, respectively, a rechargeable LiPo battery rated at 870 mAh and three "AAA" batteries, 900 mAh. To verify these estimates, the WiTilt, fully charged, and the IMote, with fresh batteries, were allowed to run continuously until they stopped working, and gives a practical estimate of lifetime.

4.1.1 WiTilt Current Measurement: Inserting a multimeter into the power connection, as unobtrusively as possible, required exposing the power and ground terminal from the female connection on the WiTilt and the male connection from the LiPo battery. Universal male connectors were found attached to a portable home phone battery, a Wellson 2.4V 300 mAh, and allowed a power connection from the WiTilt. The male connector of the battery was exposed by inserting a small section of 22-gage speaker wire.

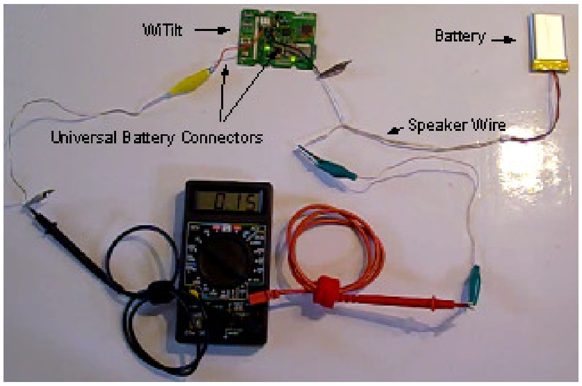


Figure 3: WiTilt current measurement

With the multimeter properly setup, the current could now be measured. Unfortunately, the measurements fluctuated significantly and made the current estimation difficult. Using a digital camera, a QuickTime movie of a transmitting WiTilt, see Figure 3, was recorded and the multimeter readings where read, frame by frame, in a two minute period. Unfortunately, this analysis was too time-consuming and a different approach was taken. Sample the multimeter readings at ten random frames and estimate the current from this sample. This method allowed more trials of different speeds and transmission modes but had a slightly reduced accuracy.

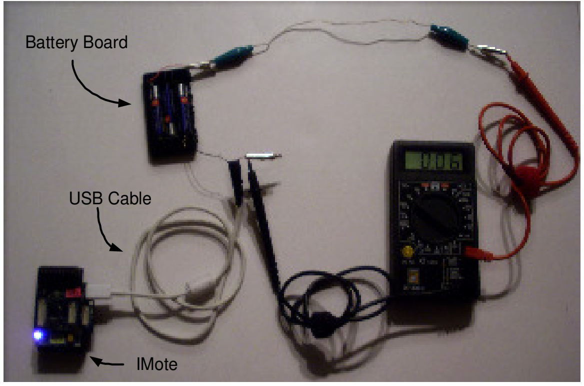


Figure 4: IMote current measurement

4.1.2 IMote Current Measurement: The IMote has a modular battery board that plugs directly into the radio board and the power leads are contained in 40-pin Hirose connector. While this design gives the device good stability and strength, the power connections cannot be isolated easily. But the IMote can also be powered through its USB Mini-B port. Power was supplied with a makeshift battery board, fabricated from an old television remote and three OEM batteries, Panasonic Industrial AAA AM-4PI/C. These batteries have a fresh voltage of 1.53 V and can power the IMote from its battery board, voltage range 3.2-4.7 V, but fall short of the range needed for the USB, range 4.8-5.2.

Fortunately, not all AAA batteries are the same and a generic brand, Presidents Choice Long Life, have a fresh voltage of 1.61V and will start the IMote through its USB connection. All power tests for the IMote were conducted with this battery setup but it was noted the same current measurements were observed when the device was powered through a USB connection, i.e. from a laptop, but may be due to the poor resolution of the multimeter. The IMote current recorded was constant and made estimation simple.

4.2 Range

The second hypothesis requires a ten meters range, more than enough for a BSN. The following tests will confirm this distance plus investigate the IMote and its ability to change radio power, not an option with the WiTilt.

4.2.1 Range test 1: The first range test was performed in an isolated home environment with no other equipment using the 2.4 GHz ISM band. Also during these tests, a clear line of sight was maintained between the IMote and the receiver. The firmware application was XAccel at channel 11 and *k*=20. The effects of distance and radio power were measured with five trials of sixty seconds each and the packet success rate recorded. The maximum distance in this room was 9.3 meters, just short of the hypothesis.

4.2.2 Range test 2: Performed outside, this test was designed to show the maximum range of the networks and would not be typical for stroke research purposes. The surroundings were a typical neighborhood and estimation of ISM interference was not possible but several wireless LAN routers are known.

WiTilt range was measured by leaving the sensor stationary and moving the receiver or WPDA and note where the Bluetooth connection is broken. Three different Bluetooth configurations were used, a Class I USB dongle, a Class II, and a PocketPC with five different trials. The IMote ran firmware IM3 (100 Hz) at 0 dBm and the packet success rate was measured with varied distances.

4.2.3 Range test 3: A home environment will not be free of ISM interference and this test was designed to show theses effects. Three different common sources of interference in a home were studied, a common WiFi router (D-Link DI-524), a portable 2.4 GHz phone (Panasonic KX-TG2621) and a microwave (Frigidaire 950W emitting at 2.45 GHz) in a ground floor setting.

A WiTilt trial had the sensor location fixed and the receiver, a HP iPAQ 210 PocketPC, walking away on a path that intercepted ISM interference and the distance the SPP connection broke was recorded. The IMote trials were all performed by a stationary sensor using IM3 k=20 (50 Hz) at 0 dBm and moved the sniffing board to different distances along the same path as the WiTilt with five 60 second trials at each distance to record the success rate.

An initial test with no interference was performed to give a benchmark. Three follow-up tests for each source of interference was conducted, except for the WiTilt and phone, in isolation to show their single effects on the connection or success rate. The WiFi router was located on the second floor and was set to channel 6 or 2437 MHz and purposely set to overlap with the frequency of the IMote radio set at channel 17 or 2435 MHz. Also, the WiFi network was transferring a large file to a laptop located near the BSNs to simulate heavy wireless traffic. The microwave boiled water during its trials and was located in the path walked by the receivers. Placing a phone call in the vicinity of the sniffing board during a trial simulated the portable phone interference.

4.2.4 Range test 4: Inspired by 4.2.3, it is performed with the IMote only. This test was designed to show the packet success rate by different IMote radio channels with WiFi interference. It was performed at York University and used the AirYork system with heavy traffic simulated by a download of the iTunes setup file, iTunes64Setup.exe or 76 MB, to the laptop equipped with the IMote sniffing board. The distance between receiver and IMote was 1.82 meters to simulate a reasonable distance of a BSN.

4.3 Sampling Frequency

The third hypothesis was investigated by a determination of the maximum speeds of both devices. IM1 and IM2 were initially created to study how fast the sensor could operate and are similar to the WiTilt binary mode. Their sampling rates can be maximized by setting k=0, no delay, and provides an answer to the speed question. But IM2 is fundamentally different since it does not transmit and is limited by the execution time of a sample collection.

IM3 has the added ability to change the accelerometer decimation factor and its effects are studied on the sampling rate. The decimation factor affected the sampling rate by causing the successive samples to have the same value, the "sticky" accelerometer. This sampling rate was estimated by considering all consecutive samples with the same value as one sample.

The WiTilt provides a configuration menu and allows the sensor setting to be changed by increments of 10 Hz to a maximum of 350 Hz for binary and 220 Hz for raw. To provide independent confirmation of the rates, WPDA counted samples received and used its internal clock to determine the sampling rate. It seemed appropriate to set the devices to their maximum settings and record the fastest speeds possible. For reference only, the WiTilt speed was recorded with the gyroscope and battery indicators to show their effects on sampling rate.

Finally, the fourth hypothesis is tested by all four IMotes transmitting at 100 Hz and the resulting packet success is recorded.

4.4 Cost/Complexity

Comparing these two BSN on this basis can be divided into quantitative and qualitative categories. The only quantitative measurement made is cost. For qualitative purposes, ease of use, device design and implementation of software and firmware is discussed and rated on a subjective scale.

4.5 Quality of Measurement

The signals produced by the two systems are compared to estimate the quality of the signal with respect to acceleration. These tests will compare the two devices or, more properly, compare their accelerometers and determine if the IMote signal is usable for stroke research.

4.5.1 Static test: Simply record the signal produced by a still device. Both devices were placed on a desktop and a constant acceleration signal is assumed and an estimation of the signal's standard deviation, σ, is converted to units of g. For this conversion the WiTilt signal needed to be divided by 250 and the IMote by 1000.

4.5.2 Pendulum test: Both devices were simultaneously placed on the pendulum and both BSNs collected acceleration samples. This test allowed a study of the two signals collected from the same event. The signals were also synchronized and scaled to the unit g.

4.6 Latency

Although not an original consideration in the comparison of the devices, the measurement of the latency of the devices was suggested. Define latency as the time difference between when an event is recorded, t_R, and when an event happens, t_E.

$$t_L = t_R - t_E \tag{3}$$

A latency comparison can be estimated by recording a common event by both the IMote and WiTilt. One application produces and records the event time while both systems record the time of the event.

The common event is a "bump" produced by a PC sound system subwoofer. The bump is caused when a sound file, *Latency.wav*, is played. This sound file represents a square waveform rated at 0.5 Hz and was created with the help of shareware, NCH Tone Generator from NCH Software. It produces two sharp pulses, one bump immediately followed by another bump after one second. The bump, with the volume full, causes a vibration in the subwoofer that can be registered by both devices.

With the WPDA, when a button was pressed, the sound file was played and the time recorded by the JVM, t_E. The reporting of this time was inserted just before the last instance the sound file was being written to its output to minimize the latency introduced by the JVM. During this time, data was collected from both devices and t_R was estimated by their respective applications. The WiTilt was run at 100 Hz and the IMote had a firmware implementation with one sample per packet (64 Hz) to make its time estimation simpler without multiple samples in a packet, but this restricted the resolution to 16 ms intervals

In a 45 second period, 12 bumps were produced. The z-axis data showed the best activity since it was in the vertical orientation and a bump would be signified when the signal exceeds some threshold. Assuming the signal is constant, μ_z, calculate its standard deviation, σ_z and define a deviation:

$$Z_i = (z_i - \mu_z) / \sigma_z \tag{4}$$

Using this deviation threshold, when $|Z_i| > 3$ then t_R was recorded as the time of z_i.

Although the total length of time estimated for the latency is measured, it is a result of many events and not just the latency of the sensor. For example, a Java button being pressed, the wave file being read and other processes will have their own effect. It is argued that these unknown latency times will be similar for both devices and therefore the difference in the total time will indicate the difference in the latency times of the devices and allows a comparison.

5. Results

5.1 Power

Table 1: Current comparison by sampling rate

Speed Hz	WiTilt mA	IMote mA	IMote % WiTilt
50	150	60	40%
100	***150***	***60***	***40%***
130	170	70	41%
310	180	100	56%

Table 1 shows both devices increase power consumption with increasing speed due to increased radio and processor activity. At a target rate of 100 Hz, the IMote uses 2.5 times less power and with equivalent batteries should last as long. From these current measurements, an estimation of lifetimes is made in Table 2. The WiTilt lifetime matches closely the estimate but the IMote falls four hours short, or 27%. This may be due to an assumed capacity of 900 mAh for the OEM batteries. A data sheet obtained does not show the rating but graphically estimates the life at 15 hours [5].

Table 2: Lifetime estimates with actual results

Device	Battery (mA)	Est. Life (hr)	Actual Life (hr)	Firmware
WiTilt	LiPo 3.7V 870	5.8	5.7	V3.2 Raw 100 Hz
			5.1	V3.3 Raw 100 Hz
IMote	3 "AAA" 4.5V 900	15.0	10.8	XAccel, k=20 (43 Hz)
			9.9	IM3, k=9 (98 Hz)

5.2 Range

5.2.1 Range test 1: Figure 5 shows the IMote with varied radio power and distance and its effect on packet success rate. At maximum power, the IMote certainly proved the second hypothesis and showed minimal loss of packets. The lower power settings showed less success when less than 10 meters and has an estimated path loss exponent of 2.4. The actual limit of the BSN is the size of the human body or approximately 2 meters and this result suggests –15 dBm as a lower limit when no interference is present.

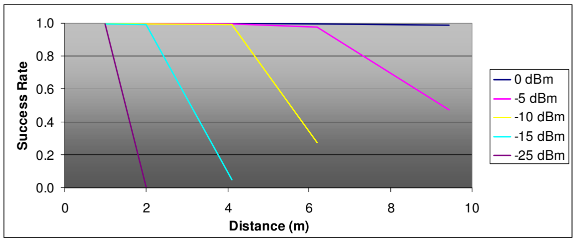


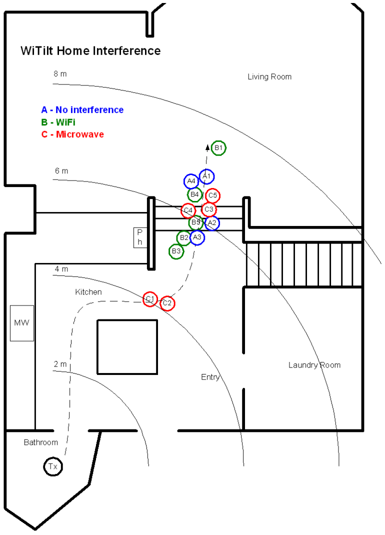
Figure 5: IMote with no interference

5.2.2 Range test 2: Designed to find the maximum range, Table 3 shows the result of different Bluetooth radios available. The USB dongles had only a slight distance reduction expected for power reduction from Class I to II.

Table 3: WiTilt maximum range outside

Receiver	Trial - Distance (m)					Avg (m)
	#1	#2	#3	#4	#5	
IoGear USB Dongle (Class I)	16.7	27.3	31.3	20.7	11.7	**21.5**
TrendNet USB Dongle (Class II)	18.1	18.0	18.0	17.9	20.1	**18.4**
HP IPAQ 210 PocketPC	6.8	9.3	9.5	5.9	9.5	**8.2**

The PocketPC had the worst range and is likely due to reduced radio power necessary for battery conservation and barely fails the second hypothesis. Tests with the IMote suggests the range of the IMote network could extend to 30 meters.



Trial	SPP Break	
	Dist (m)	St. Dev
A	6.30	0.55
B	6.11	0.91
C	5.43	1.22

Figure 6: ISM interference in the home with WiTilt

5.2.3 Range test 3: Figure 6 shows the result of home interference on the WiTilt and IMote. For the WiTilt, the average distance for each trial does not appear to be significantly different and demonstrates the benefit of frequency hopping in Bluetooth as it deals effectively with this interference. It is a completely different result for the IMote. With no interference the system works well out to 10 meters but is adversely affected by all three types of interference. Most disastrous, the WiFi kills the IMote BSN but the channels were directly overlapping and this behavior can be corrected by changing channels. The microwave and phone both had significant adverse results and these findings suggest interference will be a factor for the IMote in a home environment.

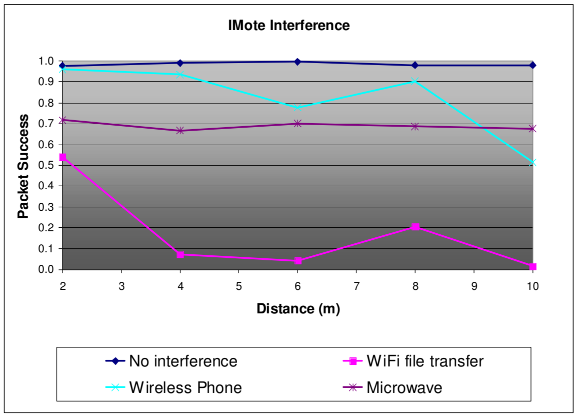


Figure 7: Home interference with IMote

5.2.4 Range test 4: Since WiFi interference has a major effect on the IMote radio, it seemed logical to explore this a bit further and is acknowledged by Crossbow [6]. A collapse of the network was recorded at channels 21 (2.455 GHz), 22 (2.460 GHz) and 23 (2.465 GHz) and is explained by AirYork set at WiFi channel 11 (2.462 GHz). The distance was within the range of a typical BSN and suggests the IMote should have a strategy to deal with interference in a home to avoid a WiFi or other ISM radiation.

5.3 Sampling Frequency

5.3.1 WiTilt: Three different versions of WiTilts were tested for maximum speed in Table 4 and reported with different modes. The older V2.5 does not have a gyroscope or battery indicator and is not tested. With the V2.5 in binary mode the maximum setting is 610 Hz, V3.2 and V3.3 allow 350 Hz. The speeds reported by WPDA are quite different with one exception. It appears the WiTilt uses the same strategy used by IM3 with the insertion of a delay to control sampling rate. At the higher settings, a change to the speed setting of the WiTilt does not produce a change in the sampling rate and can be shown to behave just like IM3 when *k* is small or set to a high sampling rate. Also, note the underachievement of V3.3 where researchers must be careful of the speed setting of the device to get the desired rate. The gyroscope and battery indicator in V3 also further slow the devices. These high speed issues aside, the WiTilt behaved well at 100 Hz. Note: the third hypothesis requires 200 Hz but researchers typically only need 100 Hz.

Table 4: Maximum speeds of WiTilt

Mode	Axis	WiTilt V2.5	WiTilt V3.2	WiTilt V3.3	IMote IM3
		Max Hz	Max Hz	Max Hz	Max Hz
Binary	**XYZ**	**461**	**531**	**162**	**314**
Binary	XYZR	N/A	375	146	N/A
Binary	XYZRB	N/A	350	109	N/A
Raw	XYZ	198	288	126	N/A

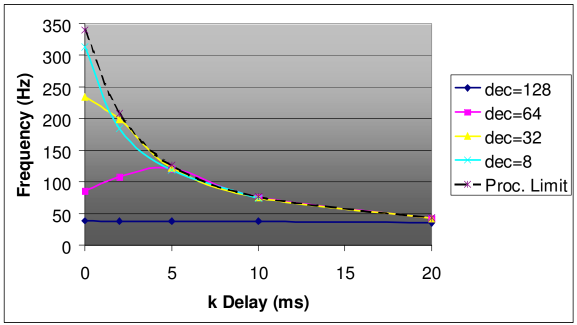


Figure 8: IM3 speed by delay and decimation factor

5.3.2 IMote: The maximum speed achieved by IM3 is shown in Figure 8 and recall this is analogous to the binary mode of the WiTilt. When the decimation setting is set to its lowest, a rate of 314 Hz is achieved and proves the third hypothesis. When the transmission of the packet is removed with IM2 rates close to 900 Hz are observed and this suggests the limiting factor of time to collect a sample is 1.1 ms.

5.3.3 IMote BSN: An issue occurs when more than one device attempts to transmit at high speeds. Four IMotes at 50 Hz transmit 5 packets per second when there are ten samples in each packet. The BSN has no issues with this arrangement but when they are increased to 100 Hz or 10 packets per second the system starts to loss packets. This is caused by collision avoidance since ZigBee uses CSMA/CA and increasing the speed of the devices increases the chance of a busy radio space. This suggested increasing the number of samples to 14 per packet but it still does pass the fourth hypothesis. A compression scheme is needed to halve the data of the network and simulate a 50 Hz system.

5.4 Cost/Complexity

The cost of an IMote is three times more than a WiTilt. The WiTilt, a single purpose sensor, is a commercially available product and uses scales of economy to lower its price. The IMote is a general-purpose research tool with more sensors, not currently needed by stroke researchers, and is expandable for other future sensors. A similar 802.15.4 radio with a single accelerometer sensor mass-produced would be cheaper than the IMote.

The WiTilt is simple to operate with a convenient power switch, built-in rechargeable battery, small profile and pre-programmed firmware. It's size and weight is appropriate for a body sensor. The only complication was the development of a GUI application to log the data. With the BlueCove library, it provided the necessary connections and relieves the developer of the difficult job of dealing with different operating systems and different Bluetooth stacks. The development time is mainly in improving the application's functionality.

The IMote has a modular design but is more bulky than the WiTilt and the IMote's power source is 3 AAA batteries, which increase its weight and profile. The development of the required firmware and logging applications make this BSN moderately more time consuming. The provided development kit helped with firmware application but it is an added level of complexity. The logging application for the IMote, IGA essentially has the same level of complexity of WPDA.

5.5 Quality of Measurement

5.5.1 Static Test: Table 5 shows a marked difference between the two sensors. The IMote produces a much cleaner signal. The standard deviation of WiTilt is almost two orders of magnitude greater when converted to units of g. The IMote would provide a TRI researcher with much cleaner data for analysis.

Table 5: Static test for quality

Device	ID	Prog	Speed	X	Y	Z
			Hz	σ (g)	σ (g)	σ (g)
IMote	0CEE	XAccel k=20	42	2.91E-04	2.59E-04	8.81E-04
IMote	0D56	IM3, k=8	97	4.09E-04	4.94E-04	1.02E-03
IMote	0CEE	IM3, k=8	97	5.79E-04	4.83E-04	1.58E-03
WiTilt	9C0D	50 Hz, Raw	41	2.41E-02	2.53E-02	1.91E-02
WiTilt	9C0D	50 Hz, Raw	41	2.46E-02	2.56E-02	1.96E-02
WiTilt	9BF8	100 Hz, Raw	99	2.02E-02	1.95E-02	1.53E-02

5.5.2 Pendulum Test: Once the two signals of both devices are synchronized and converted to g units in Figure 9 the signal differences can be noted. The noise seen in 5.5.1 can be seen in the maximum and minimums of the WiTilt signal. It never reaches the peaks obtained by the IMote and very poorly approximates a sine wave. Further, the WiTilt has six negative spikes reported which do not appear in the IMote data. This further shows the superiority of the IMote for acceleration research.

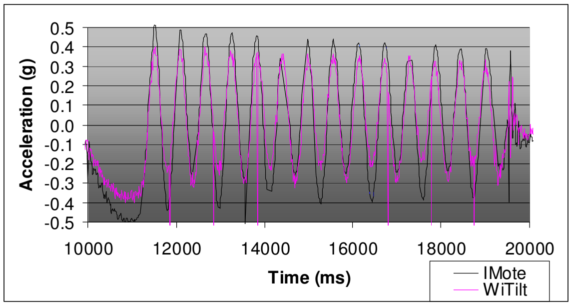


Figure 9: Synchronized signals on pendulum

5.6 Latency

With Figure 10 the IMote clearly shows the two "bumps" separated by a one second interval and represents each time the sound file was played. Note that the eleventh set is missing the first bump due to packet loss. The threshold test easily found all these events for the IMote but the noise, found in the WiTilt, made this determination more difficult and illustrates the increased quality of the IMote. With hints from the IMote data and lowering the threshold to |Z| > 2, the position of the WiTilt bumps were indicated.

Table 6 has converted the times from the two different logging applications and estimated the total latency time from these twelve latency tests for both BSNs. The data suggests a slight advantage for the IMote at 75 ms compared to 85 ms for the WiTilt. For the purposes of stroke research, these latencies are more than sufficient since the data is not time critical. The data is collected for post-processing and latency is a consideration for the signal synchronization.

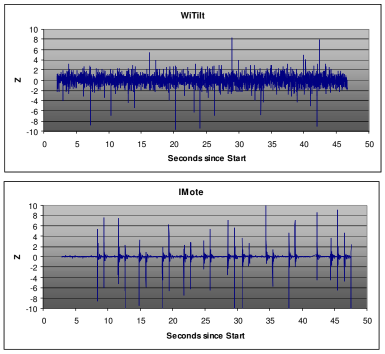


Figure 10: Latency "bumps" for both devices

Table 6: Latency measurements for both devices

Bump		1	2	3	4	5	6	7	8	9	10	11	12	Avg
WiTilt	1st	86	77	77	88	87	71	80	81	92	83	106	103	**86**
	2nd	49	88	84	81	83	73	87	88	92	91	102	97	**85**
IMote	1st	63	63	78	63	78	94	63	94	78	78	Lost	63	**74**
	2nd	79	79	78	63	78	63	78	110	94	63	79	63	**77**

6. Conclusion

The first hypothesis was almost proven by the IMote only lasted 11 hours. With further work and more efficient firmware the lifetime should be extended by one hour. The ability of the device to sleep when the patient is inactive would certainly save power. Both the second and third hypothesis were proven and showed the IMote can operate as a sensor in a BSN. The fourth hypothesis showed a failing of the ZigBee network with heavy traffic. The solution will be in compression or simply have the devices store the data for future transmission.

The WiTilt showed its effectiveness dealing with ISM interference where the IMote indicated home environment causes packet loss will require techniques to overcome this failing. The WiTilt is a third of the price of the IMote but it gives the researcher a better signal for analysis and can be considered a prototype of a mass-produced 802.15.4 sensor. The IMote, with handling of interference and data compression of data, will pass all four hypotheses and with its extended lifetime can be used in stroke

7. References

[1] W.H. Gage , K.F. Zabjek, K.M. Sibley, A. Tang, D. Brooks, W.E. McIlroy: “Ambulatory monitoring of activity levels of individuals in the sub-acute stage following stroke: a case series”, Neuroeng. Rehabil. 2007, 4:41.

[2] S.E.T. Hadley, “WiTilt Reference Manual”, produced TRI internal reference, April 2009

[3] J. A. Rice and B. F. Spencer, Jr, “Structural health monitoring sensor development for the Imote2 platform”, Proc. SPIE Smart Structures/NDE. 2008

[4] STMicroelectronics, “LIS3L02DQ: Digital Output Linear Accelerometer”, Reference Manual, p. 14, 2005

[5] Panasonic, “Alkaline Battery Specifications for AAA”, 2005

[6] Crossbow Technology, “Avoiding RF Interference Between WiFi and ZigBee”, available at http://www.xbow.com/Support/Support_pdf_files/ZigBeeandWiFiInterference.pdf, 2004.