Reducing the Power Consumption of an IMU-Based Gait Measurement System

Shenggao Zhu, Hugh Anderson, and Ye Wang

School of Computing, National University of Singapore, Singapore
{zhusheng,hugh,wangye}@comp.nus.edu.sg

Abstract. This paper presents our approach to reducing the power consumption in our Gait Measurement System (GMS), which is the foundation for various monitoring and assistive systems. Our GMS is a small foot-mounted device based on an Inertial Measurement Unit (IMU), containing an accelerometer and a gyroscope. It can compute gait parameters in real-time, including cadence, velocity and stride length, before transmitting them to a nearby receiver via a radio frequency (RF) module. Our power saving strategy exploits the cooperation between both hardware and software. By realizing on-chip computing, reducing RF usage and enabling sleep mode, the GMS's current consumption was dramatically reduced. In active mode, the GMS consumes about 2.1mA, while in standby mode, the current is only 20μA. Powered by a small rechargeable 110mAh battery, we expect the GMS to last for months of normal usage without recharging; a duration necessary for our intended applications in e-health.

Keywords: Low Power, Gait Measurement, IMU, On-chip.

1 Introduction

Gait measurement techniques aiming to estimate various spatio-temporal walking parameters (e.g., cadence and stride length) are the foundation for numerous monitoring and assistive systems. From a technological perspective, there are three kinds of approaches: machine vision based, floor sensor based and wearable sensor based [1]. The machine vision approach has been used to monitor and track whole body movement, including highly accurate gait parameters. The floor sensor based approach is also widely applied, wherein a series of pressure sensors are placed on the floor or integrated into a mat, such that gait information is collected when people walk across. Both machine vision and floor sensor based approaches are restricted to use in a controlled environment; the gait measurement must be conducted at a specific time or location, only where the video or sensor mats are available.

By contrast, the wearable sensor based approach can move with the subject. Inertial Measurement Units (IMUs) for motion analysis have become relatively accurate and inexpensive. An IMU usually contains an accelerometer and/or a gyroscope and perhaps a magnetometer, each recording along 3 (x,y,z) axes.

W. Lin et al. (Eds.): PCM 2012, LNCS 7674, pp. 105–116, 2012.

The units are attached to the human body (e.g., on the waist, shank or foot) to track motion. Since the units are becoming less expensive, they are finding applications in many fields, where previously you would not have considered using them. For example, by analyzing people's specific walking patterns, gait recognition can achieve very high accuracy in human identification [2]. IMU-based navigation system can be used in combination with GPS to improve accuracy [3], or in situations where GPS is not available [4]. Many context-aware applications use inertial sensors to detect whether the person is stationary, walking or running [5].

Our interest in IMU technology is in the Healthcare field, particularly with people suffering from walking difficulties, such as Parkinson's disease (PD) patients and stroke victims. Apart from the obvious use in measuring gait and recording the improvement over time, we have a specific application in a music therapy technique known as Rhythmic Auditory Stimulation (RAS), which offers an alternative treatment method for PD and stroke patients [6]. To facilitate and automate the music therapy procedures, our research group is developing a RAS-based gait training system, which involves a gait detection sensor and a tempo-based music search engine [7]. When the patients are having gait training, they can follow the rhythm of a selected song whose tempo is related to the patients' walking cadence. Therefore the convenient and real-time gait measurement is a fundamental part for our system.

Power saving is an important problem for practical applications, especially for mobile devices. However, few studies investigate the power consumption of an IMU-based Gait Measurement System (GMS). In our context, it is important to make a GMS that can be used in the field for a long time without maintenance. A related work is the Kalman Filter based power optimization [8], but it is in a quite different scenario.

In this paper, we describe the design and optimizations we used to reduce power consumption in our IMU-based Gait Measurement System. Our GMS measures a range of spatio-temporal gait parameters to assess in detail a patient's walking ability, for use in a clinical environment. In the future, it could also be integrated with other context-aware applications.

Our GMS is a tiny, low-cost and very power-efficient device with the capability of real-time on-chip gait parameters calculation, including cadence, velocity, stride length and so on. Although our GMS is built up with very inexpensive components, the average stride length measurement error is smaller than 3%, and the error of total walking distance measurement is less than 2%. These accuracy results are comparable with those obtained from expensive sensors as reported in literature [4,9,10]. By realizing on-chip computing, reducing RF usage and enabling sleep mode, the GMS's current consumption drops greatly. In active mode, when the patient is walking, the GMS consumes about 2.1mA, while in standby mode, when the patient is stationary, the current is only about 20μA. We expect the unit to last for months of normal usage without recharging, powered by a small rechargeable 110mAh battery. If the unit needs to be recharged it can be connected to a USB port and recharged in an hour or so.

The remainder of the paper is organized as follows. In Section 2, we present an overview of the GMS prototype, and briefly describe the gait parameters measurement. In Section 3, we introduce three versions of GMS with different power-saving strategies, and compare their power efficiency by experiments. Conclusions and future work are provided in Section 4.

2 System Overview

2.1 Prototype Architecture

The GMS prototype consists of an ATmega328 based microcontroller board (Arduino Pro Mini), an IMU, a RF module (RFM12B), and a 3.7 volt Lithium-ion battery. The IMU is a combo board with 6 degrees of freedom (DOF): a 3-axis accelerometer (ADXL345) and a 3-axis gyroscope (ITG-3200). Figure 1 shows these components, communicating over a shared communication link.

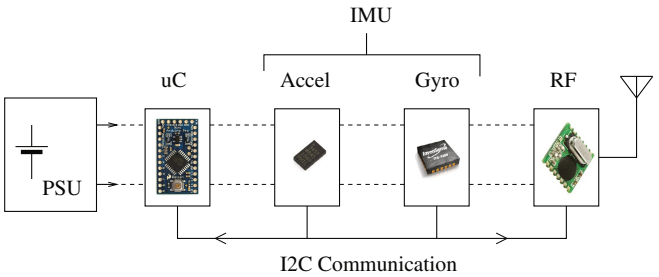

Fig. 1. Block diagram showing the power conditioning and charging circuitry (PSU), microcontroller (uC), accelerometer, gyro, and RF module

The prototype uses I^2C for communication between the microcontroller, the IMU and the RF module. The RFM12B 868MHz RF module provides a bi-directional data service at over 100kb/s over 100 metres (outside), and has an ultra-low power standby mode. We have experimented with techniques to use redundancy to recover from errors in transmission, and have found the unit reliable in our context.

The configuration of the IMU in the prototype is listed in Table 1.

Table 1. Accelerometer and gyroscope configurations

	Accelerometer	Gyroscope
Range	$\pm 16\ g$	$\pm 2000\ ^\circ/s$
Resolution	13 bits	16 bits
Sensitivity	256 LSBs / g	14.375 LSBs / $^\circ/s$
Sampling Rate	100 Hz	100 Hz

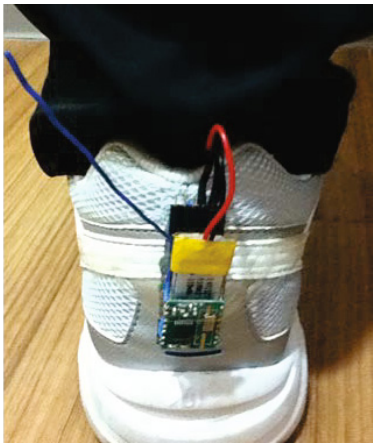

Fig. 2. The GMS prototype attached to a shoe

The complete system measures 34mm×18mm×11mm in size and weighs 9g (battery included). It is shown in use, attached to a shoe in Fig. 2.

After the GMS is assembled, the IMU sensors must be calibrated once, recording the offset and precision of each axis. These parameters are used by the on-chip software, and affect the measurement accuracy. When using the GMS, it is attached to the heel (on the sock, shoe or directly on the calcaneus). The GMS automatically detects if the patient is walking, and if so, calculates and records accumulated gait information. The calculated gait parameters for each stride are sent to a receiver via the RF module. If the patient is stationary, the system goes to sleep, drawing very little current, but waking up from time to time to see if the foot has moved.

2.2 Gait Measurement

We attach the GMS to the heel as shown in Fig. 3. Since we are primarily interested in the patient's horizontal movement, we calculate the horizontal acceleration (a_h) and perpendicular acceleration (a_p), using the measurements of acceleration in the y- and z-axes and the angular rate in x-axis (see Fig. 4).

The specific algorithms used for gait parameters measurement are described in detail in the paper [11]. We assume that at the beginning and the end of each stride, the velocity and acceleration of foot-mounted IMU are both zero, which we call *zero points* [12]. Each stride is detected by using a series of time-varying thresholds and sliding windows based on characteristic points such as the toe-off and heel-strike points (Fig. 4).

The following gait parameters can be derived from this collected sensor data:

1. **Cadence**: The number of steps per minute. It is computed by recognizing each stride in real-time.

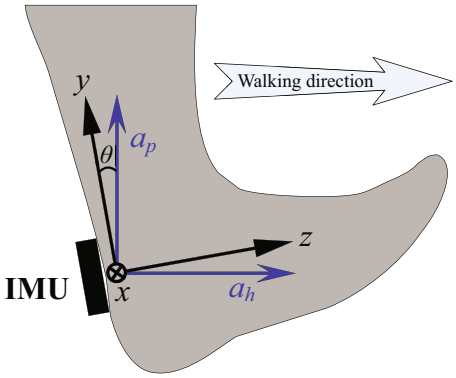

Fig. 3. Illustration of the IMU coordinate system XYZ and the walking direction

2. **Velocity**: The velocity is based on the integration of horizontal acceleration.
3. **Stride Length**: The integral of velocity over the period of the stride cycle.
4. **Swing/Stance Ratio**: The ratio of swing time (from toe-off to heel-strike) to stance time (from heel-strike to toe-off).
5. **Stride Regularity**: The percentage of the standard deviation of stride length in relation to the average stride length in a walking session.

Other useful data (e.g., averages, maximums) can be derived from the above five parameters.

An important element of our system is that our algorithms compute the above gait parameters, in real time, in the microprocessor. This is done by treating each stride as a separate computation, so that the system only needs to keep a small amount of historical data in the limited-memory microprocessor [11].

3 Power Saving Approaches

The prototype GMS is powered by a 110mAh 3.7v Lithium-ion battery. Our main goal is to reduce the power consumption and thus prolong the battery life. In this section we will describe and compare different power saving approaches and the results.

3.1 Approaches Summary

We used three power saving approaches: (1) use low power consumption components; (2) optimize the algorithm and reduce computational workload; and (3) put the components into sleep mode as often as possible.

When initially selecting components to build up a GMS with the desired functions, we considered the following features: size, accuracy, cost and power consumption. We had several false starts to the GMS development. At first we tried the combination of an accelerometer and magnetometer for stride length

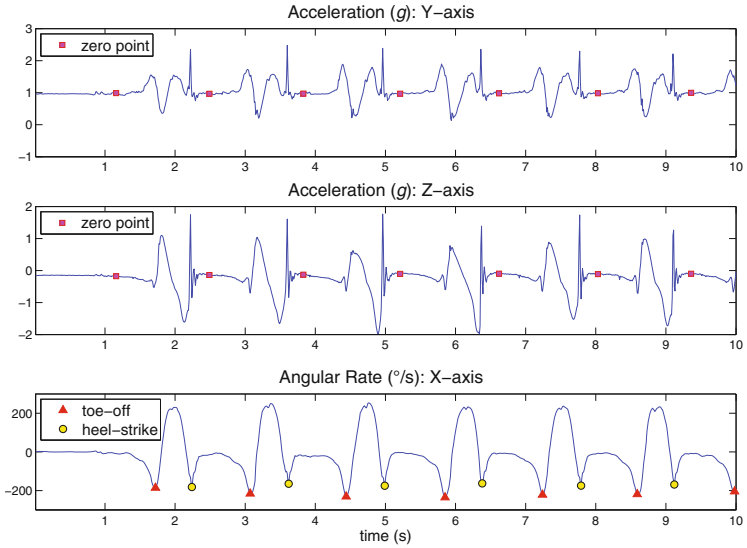

Fig. 4. An example of gait data: acceleration in y- and z-axes and angular rate in x-axis. Sampling rate is 100Hz. Zero points as well as toe-off and heel-strike points are marked on the waveforms.

measurement, because the magnetometer was cheaper and more power efficient than a gyroscope. However, the magnetometer alone was not enough to measure the GMS's exact orientation (and hence stride length). We finally chose an IMU with an accelerometer and a gyroscope.

When we investigated in more detail the power consumption of the individual components in the unit, we found that the current consumption of the regulator on the Arduino board alone was about 80μA. Therefore, we replaced the original regulator with a low-dropout (LDO) low-quiescent current regulator (MCP1703), and managed to reduce the current consumption to 5μA.

In many off-line processing applications, sensor data is sent to a remote computer through a wireless transmission. Usually data packet loss during transmission is inevitable. To reduce measurement error, packet loss must be properly handled. Sending redundant packets may be able to recover the lost data, but it also increases the transmission workload. Another common approach is using interpolation to reconstruct the missing data points. A more elegant solution is to implement on-chip computing, which can completely avoid the sensor data loss problem and simultaneously reduce the power consumption caused by wireless transmission.

Based on the the last two approaches mentioned above, we implemented three versions of the GMS with different configurations. Depending on whether the GMS is moving or not, we divide the GMS's working modes into active mode and standby mode. We measured the average current consumption of each mode,

Table 2. Summary of three versions of GMS

	version 1	version 2	version 3
Sleep Mode	Off	On	On
On-chip Computing	No	No	Yes
RF Usage	High	High	Low
Active Current	11.7mA	10.3mA	2.1mA
Standby Current	11.7mA	20μA	20μA
Active Battery Life	9.4 hours	10.7 hours	2.2 days
Standby Battery Life	9.4 hours	7.6 months	7.6 months

and then estimated the respective battery life of a 110mAh battery. Table 2 summarizes the main features of each version:

3.2 Naive Version (v1)

This is a naive version that only handles the sensor data collection, before transmitting all the sensor data to a remote computer where the gait analysis is conducted. The sampling rate is about 100Hz, so the period between two samples is close to 10 milliseconds. In each period, the GMS reads from the sensors, packetizes the sensor readings together and sends this packet to the receiver via the RF module. Then the GMS waits until new sensor data is available. The receiver is connected to a computer through a USB port, and the sensor data is processed by MATLAB in real-time. To improve measurement accuracy, data loss during transmission is recovered by sending redundant packets. In this version the GMS does not differentiate between active and standby modes, so the current consumption is the same in each case.

To calculate the current consumption, we added a 2Ω resistor in series with the power supply, and measured the voltage across the resistor. The power consumption (waveform of the resistor voltage) is shown in Fig. 5.

In each period, the maximum voltage is 40mV, so the corresponding current is 20mA. This high current consumption occurs during the RF transmission, which has a duty cycle of about 0.45. During the waiting time between RF transmissions the current drops to 5mA. The average current in one period is about 11.7mA. Powered by the 110mAh battery, we estimate that the GMS should work continuously for 9.4 hours. This estimate is consistent with earlier measurements of 5 hours continuous use, when the RF was operated with a duty cycle of 0.9.

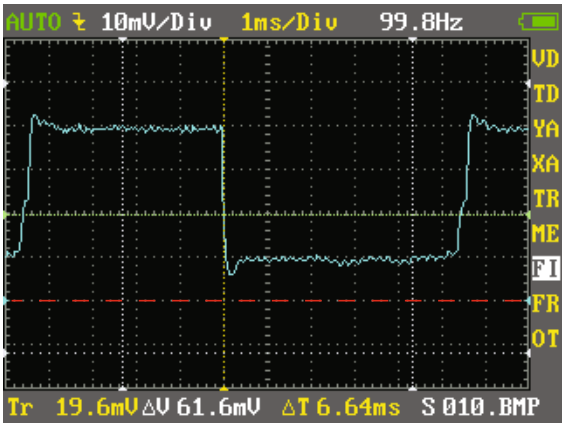

Fig. 5. Power consumption (v1). The green line is the voltage waveform, and the red dashed line is the zero voltage reference.

3.3 Sleep Version (v2)

In this version, to reduce power consumption, we put the unused components into sleep mode whenever possible.

Firstly, the microprocessor goes into sleep immediately after the corresponding data processing. When new sensor data is available, the accelerometer will trigger an interrupt to wake up the microprocessor. As such, the microprocessor is able to sleep about 6 milliseconds during the 10-millisecond interval between two sensor readings.

Secondly, since among the 3-axes gyroscope sensors, only the x-axis sensor is used in the prototype, the other two axes sensors are put into standby mode, reducing the gyroscope consumption from about 6.5mA to 2.2mA.

Thirdly, the RF module also goes to sleep after each data transmission.

Finally, the GMS can detect whether it is moving or stationary, and has different strategies for active mode and standby mode. In active mode, if there is no movement detected for a sufficient period of time (e.g., 30 seconds), the GMS will switch to standby mode. In standby mode, all components (microprocessor, accelerometer, gyroscope and RF module) are in sleep. A watchdog timer wakes up the microprocessor and accelerometer every 0.5 seconds, to check if the GMS is moving.

Table 3 gives a comparison of the components' typical current consumption in normal operating mode and sleep mode.

The GMS's power consumption for version 2 in active mode is shown in Fig. 6.

In version 2, the current during RF transmission and sensor reading stays the same with version 1 (20mA and 5mA respectively), but when the microprocessor goes into sleep, the current is only 2mA. In active mode, the average current is

Table 3. Typical current comparison

	Operating Mode	Sleep Mode
Microprocessor	5.2mA	4.2μA
Accelerometer	140μA	0.1μA
Gyroscope	6.5mA	5μA
RF module	16mA	0.3μA

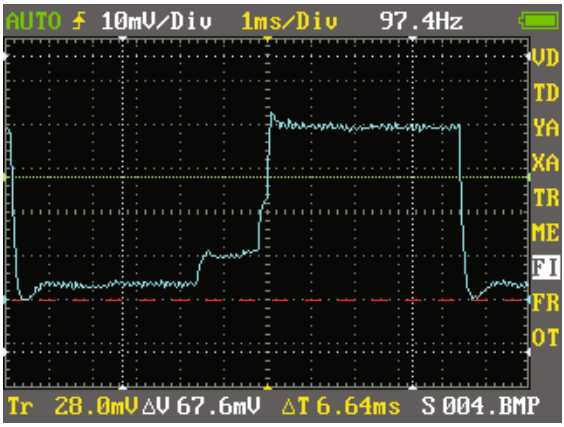

Fig. 6. Power consumption in Active mode (v2)

about 10.3mA, which means the battery can last 10.7 hours, slightly longer than that of version 1.

In standby mode, the GMS's current consumption drops to 12μA for most of the time, as shown in Fig. 7. Every 0.5 second, the microprocessor and accelerometer are awoken by the watchdog timer. Because of the accelerometer's wake-up delay, the GMS waits (in sleep) for another 13 milliseconds before reading the accelerometer to determine if it is moving again (thus there are two positive impulses in the waveform).

The average current consumption in standby mode is about 20μA, and the battery life can be as long as 7.6 months. When not using the GMS, we can simply put it away without unplugging the battery. This makes it possible to encapsulate the whole GMS (battery included) into a small box for easy and convenient deployment, and reduces the need to charge the battery between visits to the clinic.

3.4 On-Chip Version (v3)

We observed the highest current consumption during RF transmission. In order to reduce the RF usage and save more power, we developed a new version based on version 2, which implemented on-chip data processing. All the desired gait parameters are computed by the microprocessor and then sent directly to the

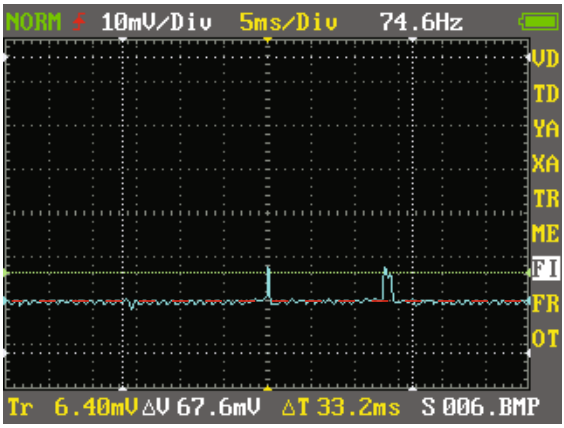

Fig. 7. Power consumption in Standby mode

receiver. In this way, we only need to transmit data for once during each stride (which usually takes longer than one second). By avoiding sending raw sensor data, the total RF transmission time is reduced to one 4.5mS transmission per stride, a duty cycle of less than 0.005.

The biggest difficulty in implementing the on-chip algorithm was the limited RAM (2kB) and computing capability of the ATmega328 microcontroller. The limits meant it was impossible to store large amounts of sensor data in buffers for later analysis, as MATLAB did. In addition, the least possible floating-point numbers and operations should be used while maintaining computing accuracy, due to the microprocessor's low speed with floating-point computation.

Thus in this version we re-designed the data structure and algorithm to strike an appropriate balance between accuracy and simplicity. In the real-time algorithm, each time after reading from the IMU sensors, these readings are processed immediately to recognize the stride cycle and compute gait parameters simultaneously. Thus there is no computation latency because once a stride is completed, it is recognized and all relevant parameters are computed. This turns out to be useful for quick responses to changes in a person's walking patterns. Our current program uses only 1kB of RAM for the computation.

The power consumption in active mode for version 3 is shown in Fig. 8.

As we can see from the waveform, the main current consumption is during sensor reading and data processing (approximately 5mA). Since there is no need for data transmission in every 10mS period, the microprocessor can sleep for a longer time. Even with data transmission in every stride, the average current consumption in active mode is 2.1mA, with a corresponding battery life of 2.2 days. In standby mode, the GMS's behavior is the same with that of version 2 (Fig. 7), and we again estimate the battery would last 7.6 months.

Version 3 is the final version of our GMS prototype. By using these power saving approaches, we decreased the average GMS power consumption to a

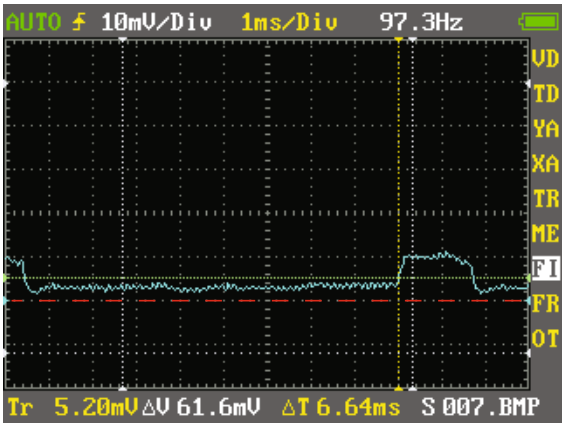

Fig. 8. Power consumption in Active mode (v3)

manageable level. With days of active battery life and months of standby battery life, our GMS can be conveniently operated and widely applied in various situations.

4 Conclusion and Future Work

We have presented a low power gait measurement system, including the prototype architecture and a range of power-saving approaches. The GMS can perform on-chip gait parameters calculation in real-time, and is expected to last for months of normal usage without recharging when powered by a 110mAh battery. Through the experiment results of our GMS prototype, this paper is intended to shed some light on the power saving problem of IMU-based systems, which is an important topic in healthcare. Since our GMS can measure various gait parameters and has a wireless communication interface, it can be easily integrated into other applications.

In our future work, we will design our own printed circuit board (PCB) instead of using the Arduino board. More efforts will be made towards reducing the size of the GMS to encapsulate it into a convenient wearable device. We are cooperating with a local hospital and will use this GMS with patients.

Acknowledgements. The work was supported by Singaporean MOE grant R-252-000-463-112. We thank the anonymous reviewers for their constructive comments.

References

1. Derawi, M.O.: Accelerometer-Based Gait Analysis, A survey. In: Norwegian Information Security Conference, NISK 2010, pp. 33–44 (2010)

2. Bours, P., Shrestha, R.: Eigensteps: A giant leap for gait recognition. In: 2010 2nd International Workshop on Security and Communication Networks, IWSCN, pp. 1–6 (May 2010)
3. Kim, S.B., Lee, S.Y., Choi, J.H., Choi, K.H., Jang, B.T.: A bimodal approach for GPS and IMU integration for land vehicle applications. In: 2003 IEEE 58th Vehicular Technology Conference, VTC 2003-Fall, vol. 4, pp. 2750–2753 (October 2003)
4. Ojeda, L., Borenstein, J.: Non-GPS navigation for security personnel and first responders. Journal of Navigation 60(3), 391–407 (2007)
5. Beach, A., Gartrell, M., Xing, X., Han, R., Lv, Q., Mishra, S., Seada, K.: Fusing mobile, sensor, and social data to fully enable context-aware computing. In: Proceedings of the Eleventh Workshop on Mobile Computing Systems and Applications, HotMobile 2010, pp. 60–65. ACM, New York (2010)
6. Thaut, M.H., McIntosh, G.C., Rice, R.R., Miller, R.A., Rathbun, J., Brault, J.M.: Rhythmic auditory stimulation in gait training for Parkinson's disease patients. Movement Disorders 11(2), 193–200 (1996)
7. Li, Z., Xiang, Q., Hockman, J., Yang, J., Yi, Y., Fujinaga, I., Wang, Y.: A music search engine for therapeutic gait training. In: Proceedings of the International Conference on Multimedia, MM 2010, pp. 627–630. ACM, New York (2010)
8. Udaya Shankar, P.S., Raveendranathan, N., Gans, N.R., Jafari, R.: Towards power optimized kalman filter for gait assessment using wearable sensors. In: Wireless Health 2010, WH 2010, pp. 137–144. ACM, New York (2010)
9. Sabatini, A., Martelloni, C., Scapellato, S., Cavallo, F.: Assessment of walking features from foot inertial sensing. IEEE Transactions on Biomedical Engineering 52(3), 486–494 (2005)
10. Li, Q., Young, M., Naing, V., Donelan, J.: Walking speed estimation using a shank-mounted inertial measurement unit. Journal of Biomechanics 43(8), 1640–1643 (2010)
11. Zhu, S., Anderson, H., Wang, Y.: A Real-Time On-Chip Algorithm for IMU-Based Gait Measurement. In: Weisi, L., Dong, X., Anthony, H., Jianxin, W., Ying, H., Jianfei, C., Mohan, K., Ming-Ting, S. (eds.) PCM 2012. LNCS, vol. 7674, pp. 93–104. Springer, Heidelberg (2012)
12. Peruzzi, A., Croce, U.D., Cereatti, A.: Estimation of stride length in level walking using an inertial measurement unit attached to the foot: A validation of the zero velocity assumption during stance. Journal of Biomechanics 44(10), 1991–1994 (2011)