A Low-Power Opportunistic Communication Protocol for Wearable Applications

Andrea Gaglione, Shanshan Chen, Benny Lo, Guang-Zhong Yang
The Hamlyn Centre, Institute of Global Health Innovation,
Imperial College London,
London, United Kingdom,
{a.gaglione, shanshan.chen, benny.lo, g.z.yang}@imperial.ac.uk

***Abstract*—Recent trends in wearable applications demand flexible architectures being able to monitor people while they move in free-living environments. Current solutions use either store-download-offline processing or simple communication schemes with real-time streaming of sensor data. This limits the applicability of wearable applications to controlled environments (e.g., clinics, homes, or laboratories), because they need to maintain connectivity with the base station throughout the monitoring process. In this paper, we present the design and implementation of an opportunistic communication framework that simplifies the general use of wearable devices in free-living environments. It relies on a low-power data collection protocol that allows the end user to opportunistically, yet seamlessly manage the transmission of sensor data. We validate the feasibility of the framework by demonstrating its use for swimming, where the normal wireless communication is constantly interfered by the environment.**

Keywords—body sensor networks; wearable sensors; data collection; data analysis

I. Introduction

Body sensor networks and wearable devices are being widely used in health and sport applications to monitor both physiological parameters and kinematic motion of subjects [1]. Micro electro-mechanical systems (MEMS) based wearable accelerometers have now been introduced widely in healthcare for activity recognition [2]. For patients with chronic diseases such as diabetes and chronic pulmonary disease, mobility, metabolic energy expenditure and activity profiling are important indices for guiding clinical management decisions [3]. In orthopaedics and rehabilitation, gait analysis through wearable sensing is employed to track the patient recovery progress and detect possible complications and deteriorations [4], [5]. In sports, they are used to support the training of elite athletes [6], [7], [8].

Early generations of wearable sensors mainly rely on store-download for post-processing or analysis/visualisation. The introduction of low-power communication hardware and protocols has enabled wider use of wireless communications, allowing real-time streaming or periodic uploads of data. Thus far, the use of wearable sensors generally involves the use of a wireless link to communicate with a base station (e.g., a PC/laptop [9], a tablet [10], a mobile phone [11], or a customised embedded system [12]). Continuous monitoring of patients is achieved through real-time streaming of sensor data. In this arrangement, wearable sensors need to maintain connectivity with the base station throughout the monitoring process, thus limiting their applicability to controlled environments such as homes, clinics or laboratories. In sports, real-time measurement of athlete parameters is mainly performed in dedicated setups [13]. However, these cannot always recreate the experience in natural training environment (e.g., the differences in the kinematics between treadmill and overground running are well-known [14]). For certain applications, reliable wireless communication is not always possible. For example, in the case of swimming, real-time wireless streaming of sensor data is problematic because of the water disturbing the communication [15]. Some wearable devices use their own storage unit (e.g., flash memory [7] or SD-micro cards [16]) to record sensor values, which can be downloaded to a computer afterwards, through a wired connection, for data analysis. However, such tethered systems are not practical, as users often demand seamless data integration and online data analysis.

There is a need for flexible architectures, being able to monitor patients or athletes in free-living or natural training environments. These applications require a reliable and seamless method of data communication. For example, in sports, sensor data needs to be transmitted immediately after the training session. Data needs to be transmitted reliably and swiftly, so that athletes could benefit from prompt information feedback from the coach to optimise the training strategy.

This paper presents a communication framework that addresses these needs. It is designed to simplify both the adoption and usage of wearable technology in practical deployments with due consideration of power usage during wireless transmission. The framework relies on an opportunistic asynchronous data collection (OADC) protocol through which the user—that can be a trained clinician or coach—opportunely triggers the transmission of data (e.g., when the subject wearing the sensing device is close enough to the base station). We use the ear-worn activity recognition (e-AR) sensor as our wearable sensing platform, which has been already used successfully in both medical and sport applications [5], [7], to demonstrate the proposed concept.

We introduce the e-AR sensor technology and illustrate the general architecture of the framework in Section II, outlining

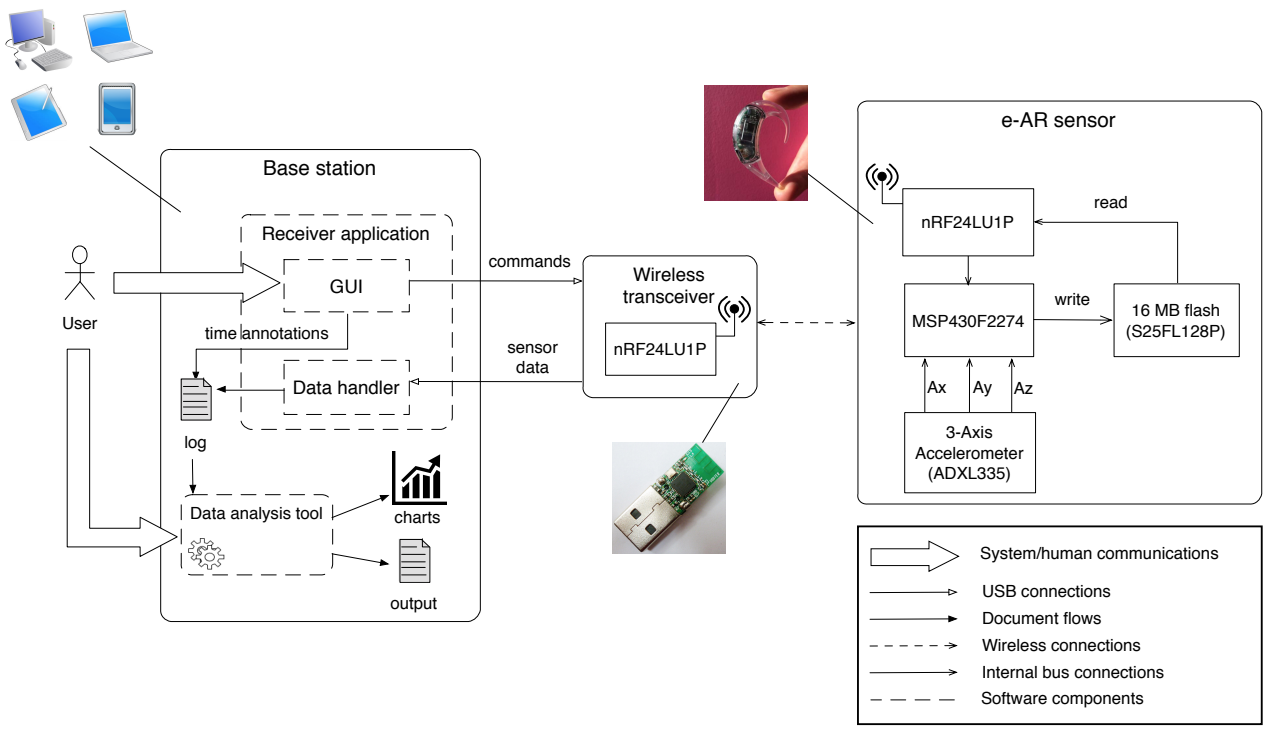


Fig. 1. System architecture of the proposed framework.

its behaviour and providing some implementation details. We present the OADC protocol in Section III. In Section IV, we evaluate the framework in terms of transmission efficiency and reliability. In Section V, we demonstrate the feasibility of the proposed framework for swimming performance monitoring, and also show the results of example post-processing data analysis, enabling a prompt feedback to the swimmers. Finally, the paper is completed by concluding remarks in Section VI.

II. System Overview

A. Sensor Hardware

The e-AR sensor is a miniaturised ear-worn device, which reduces the discomfort experienced by the subject wearing it. At the core of the device is a Nordic nRF24LU1P microcontroller [18] (8-bit 8051 compatible microcontroller, referred as *nRF24* from now on). The nRF24 is mainly used to control a built-in 2.4 GHz RF transceiver, and is coupled with a TI MSP430F2274 microcontroller [19] (16-bit *MSP430* RISC microcontroller), which performs data sampling and in-node processing operations. The e-AR sensor is equipped with a 3D accelerometer (Analog Devices ADXL330) having a range in SI unit from -3*g* to 3*g*, a Spansion S25FL128P 16MB flash memory, which provides extensive storage capability (15 hours of 100 Hz data), and a 55mAhr Li-Polymer battery. The device is waterproof and weighs 7.4 grams, thus allowing for an easy way of using it both in the water and in free-living environments.

B. System Architecture

The e-AR sensor and the other main components of our framework are illustrated in the overall system architecture, shown in Fig. 1. The e-AR sensor runs two different programs in parallel on its two microcontrollers and operates in two different modes: `recording` and `transmitting`. In the `recording` mode, the nRF24 is initially switched off, while the program running on the MSP430 performs two tasks. First, it samples the 3D accelerometer data and stores them into the flash memory. Second, it periodically switches the nRF24 on to check for incoming commands from the user, according to our OADC protocol, as described in Section III.

The e-AR sensor switches to `transmitting` mode on demand or when opportunistically detects there is a reliable window for transmission. In that case, the MSP430 program stops, while the program running on the nRF24 starts to read the sensor data from the flash memory and send them over the radio. The reception of a command by the nRF24 is signalled to the MSP430 through a shared port. The two microcontrollers could also exchange data through the internal SPI bus, thus enabling the implementation of real-time monitoring applications, where sensor data are sampled and transmitted straight away. Both programs are written in embedded C.

The *base station* allows for data collection, user interaction, online data visualisation and analysis. It runs a *receiver application*—currently implemented in C++—that lets the *user* send *download* commands (requesting data transmission) to the e-AR sensor through a *graphical user interface* (GUI). An nRF24 module controls the *wireless transceiver* and is connected to the base station via USB. It runs an embedded C program that forwards commands coming from the receiver application to the e-AR sensor, and forwards sensor data coming from the e-AR sensor to the receiver application. The *data handler* then stores the sensor data in a *log* file. The GUI also allows the user to write the log file during the actual observation period (e-AR sensor in `recording` mode) to

annotate local timestamps (*time annotations*), which can be used to track specific events (e.g., in a swimming application, it might be useful to annotate the timestamps as the swimmer pushes off the pool wall in order to calculate the lap times). The log file is then analysed offline locally (alternatively, it could be sent to a back-end system and cloud service for remote analysis or storage). Finally, the user interprets the analysis results, in *charts* and/or textual *output* file.

We implement a lossless differential pulse code modulation (*DPCM*) compression scheme [20] in order to save memory on the e-AR sensor and speed up the transmission time. DPCM stores only the difference between two consecutive samples in the flash memory of the e-AR sensor, and is particularly suited for health monitoring and sport applications. Such applications usually require a high sampling rate, and the values of differences between two consecutive samples are smaller than those of the original samples. Our scheme uses 8 bits to store the differences, which means that our accelerometer samples—represented in 16 bits—are compressed 2 to 1. Original data are then reconstructed by the receiver application. In the current implementation, the user could enable DPCM compression on both the embedded programs and the receiver application at compilation time.

III. OADC PROTOCOL

Energy efficiency is a major requirement for sensor networks and wearable embedded devices, since limitation on energy resources directly affects the system lifetime. Therefore, the design of low power communication protocols is required, as it controls the transceiver operation, which is often the most energy-consuming component in a sensor node [21]. Significant effort has been devoted by the research community to the development of medium access control (MAC) protocols, especially designed for sensor networks [22]. Among them, asynchronous protocols relying on the preamble sampling techniques (see [23], [24]), also called *low power listening* (LPL), are widely used because of their simple design and energy saving capabilities [25]. They link together a *sender* with data to a *receiver* that is duty cycling. When the sender has data, it transmits a preamble that is at least as long as the sleep period of the receiver. The latter only wakes up to sample the medium, that means listening to the radio channel for a very short duration (e.g., time to receive a packet): if the channel is found idle, the receiver goes back to sleep; if a preamble is detected, the receiver continues to listen until the packet is received. Such protocols drastically reduce idle listening, a state of the node when the radio is turned on and in receive mode, but not receiving any packets.

The proposed OADC protocol regulates the communication between the base station and e-AR sensor and follows the LPL wake/sleep schedule. Specifically, our approach is a simplified version of the MX-MAC [26], according to which the sender, instead of transmitting a long preamble, transmits repetitions of the data message. In our case, the LPL scheme is only used when the e-AR sensor is in `recording` mode, while we keep the radio always on when in `transmitting` mode. The base station acts as the sender, while the e-AR sensor acts as the receiver. Before proceeding with the description of the protocol, let us describe the main operating modes of the nRF24 radio transceiver:

- `POWER DOWN`, the transceiver is disabled with minimal current consumption;
- `RX`, the transceiver is used as a receiver, listening to the channel and trying to demodulate signals;
- `TX`, the transceiver is transmitting packets.

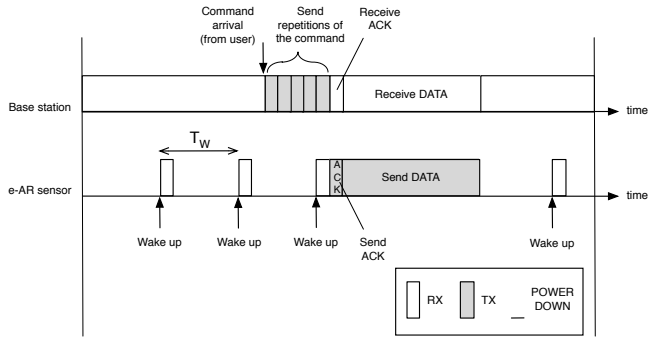


Fig. 2. The OADC protocol timeline between the e-AR sensor and base station.

A schematic representation of the OADC protocol is shown in Fig. 2. The base station transceiver is always on, by default in `RX` mode. At start-up, the e-AR sensor is in `recording` mode, and its transceiver is in `POWER DOWN` mode. The transceiver wakes up to sample the radio channel (transceiver switches to `RX`) with a constant period T_W. On the other hand, the base station transceiver switches to `TX` as soon as it receives a download command from the user. Hence, it starts transmitting repetitions of the command, until the e-AR sensor ACKs the packet. At that point, the base station transceiver switches to `RX`, while the e-AR sensor transceiver switches to `TX`. The e-AR sensor—now in `transmitting` mode—stops recording data, and starts transmitting all the sensor data recorded so far to the base station, as fast as it can, without duty-cycling.

In the current implementation, when all the data is sent, the e-AR sensor transceiver switches to `POWER DOWN` mode. The e-AR sensor starts recording data again, as well as listening to the radio channel according to the described LPL scheme. It is possible that the e-AR sensor loses the connectivity during transmission. In this case, retransmission needs to be initiated. The e-AR sensor transceiver periodically wakes up, but now it switches to `TX` mode and sends a specific beacon over the radio channel in order to check if the base station is in range. When an ACK is received, the transmission of the most recent sensor data starts again. The proposed protocol greatly simplifies the use of a specific wearable technology in practical deployments, as the user opportunely triggers data collection. Although we use the e-AR sensor as our sensing platform, the protocol could be exploited by any other sensing device with flash storage capabilities.

IV. System Evaluation

The performance of the communication framework is assessed in order to show how the system achieves the overall goal of ensuring a reliable and relatively fast data transmission. Experiments were conducted in an open space laboratory by also evaluating the effect of DPCM compression on the performance. Firstly, we compare the system with and without DPCM compression w.r.t. data transmission time. Then, we analyse the reliability of data collection at different distances between the e-AR sensor and the base station. In all the experiments, we set T_W = 5 seconds and use a sampling frequency of 100 Hz.

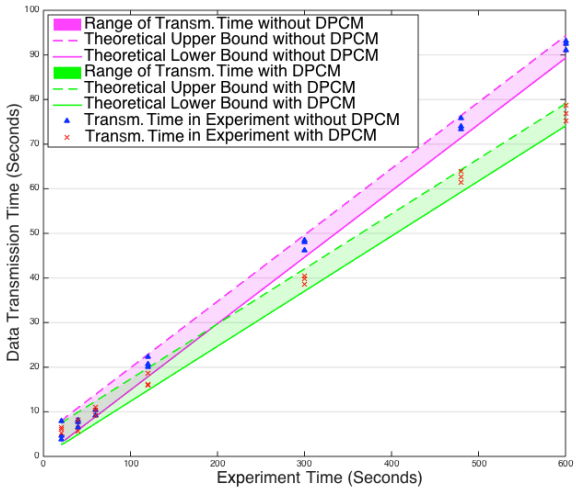


Fig. 3. Data transmission efficiency of the OADC protocol with and without DPCM.

We define data transmission time as the time interval from when the download is triggered to when the last data sample is received by the receiver application. We run 7 sets of experiments to evaluate the data transmission time w.r.t. the experiment times of 20, 40, 60, 120, 300, 480, and 600 seconds. The experiment time indicates the time interval that the e-AR sensor is in `recording` mode. For each set of experiments, we execute 3 runs of data collection with and without DPCM compression. We then compare the results with those obtained by theoretical models, deduced by the data transmission rate captured by the receiver application. Specifically, data transmission rate is 6.72:1 without DPCM compression, and 8.1:1 with DPCM compression. In other words, with DPCM scheme, 8.1 seconds of sensor data are transmitted in 1 second. The receiver application runs on a Windows 8 PC, which acts as the base station. Results are illustrated in Fig. 3. It is worth noting that effective data transmission times measured during the experiments are not the same as predicted by theoretical models. These only provide lower bounds for the transmission times with and without DPCM. In fact, data transmission times in the experiments have a random overhead falling in the range 0-5 seconds, due to T_W. However, the overhead becomes less influential to the transmission times when the experiment sessions are longer.

Although the DPCM scheme doubles the information carried in a single packet, in our practical case it only increases the data transmission rate by 20%. This is due to operation overhead of the compression algorithm implemented on the embedded microcontrollers. Therefore, theoretical models of data transmission time—with and without DPCM—indicate that it is convenient to employ DPCM compression when the experiment time is longer than ~198 seconds. In such cases, the effective data transmission time with DPCM will be lower than the predicted one without DPCM. If we consider a 10-minute experiment with DPCM compression, we are able to transmit all the data recorded in less than 80 seconds, which is sufficient for most wearable applications. For example, in an application deployment for swimming performance monitoring, the transmission time is comparable with the rest intervals required by swimmers between repeats.

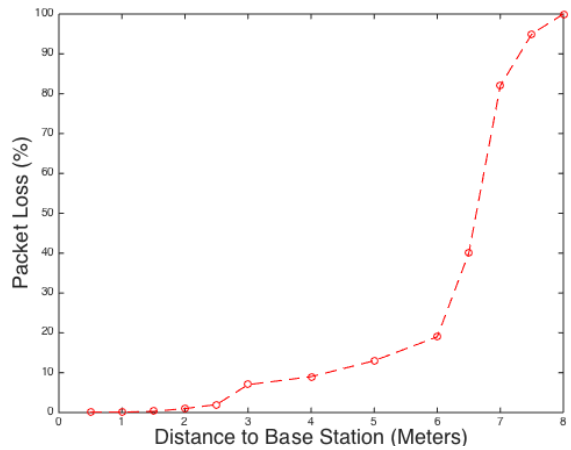


Fig. 4. Packet loss rate vs. distance of the OADC protocol.

To analyse the reliability of data collection, we evaluate the packet loss rate (PLR) at different distances between the e-AR sensor and the base station. It is worth noting that PLR analysis of the system with and without DPCM compression is the same, since the latter is a lossless scheme. Figure 4 shows that the PLR increases with the distance. Specifically, PLR is stable in the range of 0-1 meter, where its value is 0.001%. At two meters distance, PLR=0.01%. Between 2 and 7 meters PLR increases substantially, until reaching 100% at 8 meters, which means that the devices are completely out of their communication range. We do not use retransmissions to recover packets lost. However, PLR less than 0.01% and a distance between the base station and the wearable sensor less than two meters are acceptable in most wearable scenarios. For example, in a swimming application, a PLR less that 0.01% does not affect the accuracy of the stroke detection algorithm, as discussed in the next section. Further, a distance of two meters is reasonable between the swimmer and the coach during data transmission.

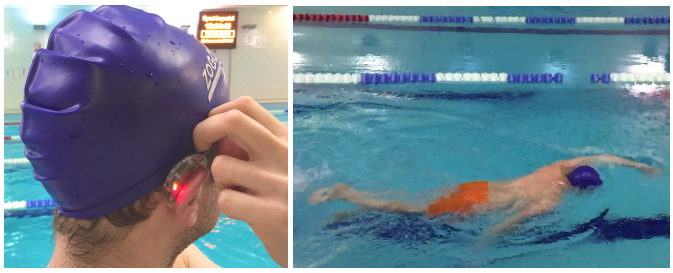
V. Evaluation with Swimming Performance Monitoring

In this section, we describe a real-world deployment of the framework in order to show how it is used as a training support tool for swimming performance monitoring. In our scenario, the swimmers wear the e-AR sensor that collects the head motion at a frequency of 100 Hz during different swimming styles. A Windows 8 tablet acts as the base station, and is used by the experiment conductor (referred as the *coach*

from now on) to collect data from the e-AR sensor, whenever wireless transmission is possible (e.g., when the swimmer is out of the water and close to the tablet). Therefore, the coach can see the swimming data immediately after each swimming session and process them through a lightweight on-tablet data analysis. Finally, he could provide a feedback to the swimmers in order to support their training strategy.

A. Experiment Design

Two male subjects wearing e-AR sensor were recruited in our swimming experiments. Although the e-AR sensor can be secured around ears by itself, they both wore the sensor under the cap for extra security and waterproof, as shown in Fig. 5. The subjects both swam in breaststroke style and freestyle for performance comparison. For each style, the subjects try to use different stroke rates and body positions in order to simulate different swimming performance.



(a) (b)

Fig. 5. (a) Subject wearing the e-AR sensor; (b) Subject in the pool.

B. Data Analysis for Performance Feedback

The data collected from the e-AR sensor is in the format of raw Analog-to-Digital (ADC) values. This data is firstly calibrated using Newton-Raphson method [27] and converted to g's. The calibrated data is then low-passed at 0.3Hz to capture only the static trend for inclination information. The *pitch* and *roll* angles of the head can thus be computed using trigonometry rules [7]. We consider that each stroke is associated with a distinctive head movement (e.g. lifting up in breaststroke style and turning to the side in freestyle), thus the stroke rate can be extracted by peak detection algorithm from the head angle signals, shown in Fig. 6. Stroke rate per minute is expressed after the swimming cycles are segmented:

$$stroke\ rate\ = \frac{60}{average\ stroke\ \ time} \tag{1}$$

Stroke length is then extrapolated knowing the pool lane length and the number of strokes per swimming session.

$$stroke\ length = \frac{pool\ lane\ length}{number\ of\ strokes\ per\ lap} \tag{2}$$

Finally, swimming speed is computed as:

$$speed\ =\ stroke\ rate * stroke\ length \tag{3}$$

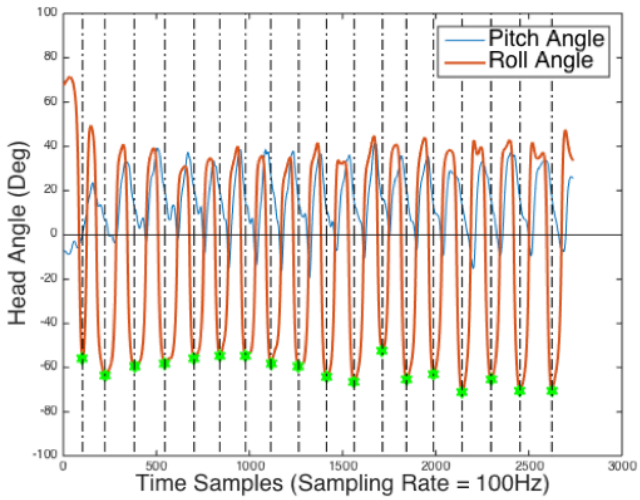


Fig. 6. Swimming cycle segmentation.

Equation (3) shows that swimming speed is proportional to both stroke rate and stroke length, with the latter two parameters competing with each other. We use a radar plot to visualise the relationship among the above three parameters. This allows the coach to read the chart, consider a performance optimisation strategy, and eventually provide prompt feedback to the swimmers. As an example, the overall radar plot in Fig. 7 shows the performance of the two swimmers' breaststroke, considering that they used three different body positions during the swimming session to simulate different swimming performance (in our experiments, higher values of the swimming speed correspond to better performance).

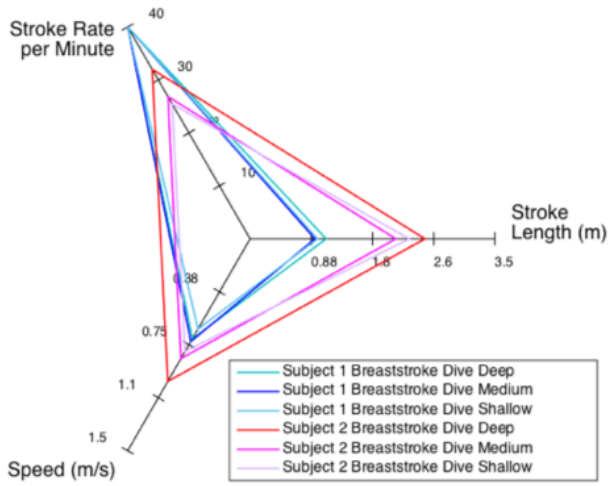


Fig. 7. Radar plot of swimmers' performance as expressed in terms of speed, stroke length, and stroke rate.

VI. Conclusion

In this paper, we have presented a monitoring framework aimed at simplifying the use of a specific wearable technology in practical deployments. The framework relies on the OADC protocol that allows the end user to opportunely trigger the transmission of sensor data. We evaluate the system w.r.t. the

overall goal of ensuring reliable and fast data transmission in order to highlight its use in practical applications. To further demonstrate the feasibility of the framework and highlight the benefits of the OADC protocol, we also deployed an application for swimming performance monitoring. Finally, we have shown the results of example post-processing data analysis for online swimming performance assessment, allowing prompt feedback to the swimmers to support their training strategy.

REFERENCES

[1] G.-Z. Yang, *Body Sensor Networks*, 2nd Edition, Springer, 2014, IBSN 978-1-4471-6374-9.

[2] M.J. Mathie, A.C. Coster, N.H. Lovell, and B.C. Celler, "Accelerometry: providing an integrated, practical method for long-term, ambulatory monitoring of human movement," *Physiological measurement*, vol. 25, pp. R1-R20, 2004.

[3] L. Atallah, B. Lo, G.-Z. Yang, "Real-Time Activity Classification Using Ambient and Wearable Sensors," *IEEE Transactions on Information Technology in Biomedicine*, vol. 13, pp. 1031-1039, 2009.

[4] M.P.M. Steultjens, J. Dekker, M.E. van Baar, R.A.B. Oostendorp, and J.W.J. Bijlsma, "Range of joint motion and disability in patients with osteoarthritis of the knee or hip," *Rheumatology*, vol. 39, pp. 955-961, 2000.

[5] B. Lo, L. Atallah, O. Aziz, M. ElHelw, A. Darzi, and G.-Z. Yang, "Real-Time Monitoring for Postoperative Care," in *Proc. 4th Int. Workshop on Wearable and Implantable Body Sensor Networks (BSN 2007),* RWTH Aachen University, Germany, pp. 122-127, 2007.

[6] D.A. James, N. Davey, and T. Rice, "An accelerometer based sensor platform for insitu elite athlete performance analysis," in *Proc. IEEE Sensors*, Vienna, Austria, pp. 1373-1376, 2004.

[7] J. Pansiot, B. Lo, and G.-Z. Yang, "Swimming Stroke Kinematic Analysis with BSN," in *Proc. Int. Conference on Body Sensor Networks (BSN 2010)*, Biopolis, Singapore, pp. 153-158, 2010.

[8] D. Morris, S. Coyle, Y. Wu, K.T. Lau, G. Wallace, D. Diamond, "Bio-sensing textile based patch with integrated optical detection system for sweat monitoring," *Sensors and Actuators B: Chemical*, vol. 139, pp. 231-236, 2009.

[9] E.M. Tapia, S.S. Intille, W. Haskell, K. Larson, J. Wright, A. King, and R. Friedman, "Real-Time Recognition of Physical Activities and Their Intesities Using Wireless Accelerometers and a Heart Rate Monitor," in *Proc. 11th IEEE Int. Sysmposium on Wearable Computers (ISWC 2007)*, Boston, MA, USA, pp. 37-40, 2007.

[10] L. C. Jatobá, U. Großmann, C. Kunze, J. Ottenbacher, and W. Stork, "Context-Aware Mobile Health Monitoring: Evaluation of Different Pattern Recognition Methods for Classification of Physical Activity," in *Proc. 30th Annual Int. Conference of the IEEE Engineering in Medicine and Biology Society (EMBS 2008)*, Vancouver, BC, Canada, pp. 5250-5253, 2008.

[11] O.D. Lara, M.A. Labrador, "A survey on human activity recognition using wearable sensors," *IEEE Communications Surveys & Tutorials*, vol. 15, pp. 1192-1209, 2013.

[12] T.P. Kao, C.W. Lin, and J.S. Wang, "Development of a Portable Activity Detector for Daily Activity Recognition," in *Proc. IEEE Int. Symposium on Industrial Electronics (ISIE 2009)*, Seoul, Korea, pp. 115-120, 2009.

[13] S. Coyle, D. Morris, K.T. Lau, D. Diamond, and N. Moyna, "Textile-based wearable sensors for assisting sports performance," in *Proc. 6th Int. Workshop on Wearable and Implantable Body Sensor Networks (BSN 2009)*, Berkeley, CA, USA, pp. 307-311, 2009.

[14] B.M. Niggs, R.W. De Boer, and V. Fisher, "A kinematic comparison of overground and treadmill running," *Medicine & Science in Sport & Exercise*, vol. 27, pp. 98-105, 1995.

[15] A.S. Silva, A.J. Salazar, C.M. Borges, and M.V. Correia, "Wearable Monitoring Unit for Swimming Performance Analysis," *Biomedical Engineering Systems and Technologies*, vol. 273, pp. 80-93, 2013.

[16] M. Ermes, J. Pärkkä, J. Mäntyjärvi, and I. Korhonen, "Detection of daily activities and sports with wearable sensors in controlled and uncontrolled conditions," *IEEE Transactions on Information Technology in Biomedicine*, vol. 12, pp. 20-26, 2008.

[17] B. Lo, J. Pansiot, and G.-Z. Yang, "Bayesian analysis of a sub-plantar ground reaction force with BSN," in *Proc. 6th Int. Workshop on Wearable and Implantable Body Sensor Networks (BSN 2009)*, Berkeley, CA, USA, pp. 133-137, 2009.

[18] nRF24LU1P Product Specification. Web page—https://www.nordicsemi.com/eng/content/download/2723/34042/file/nRF24LU1P_O_Product_Spec_v1.0.pdf.

[19] MSP430F2274 Product Specification. Web page—http://www.ti.com/lit/ds/symlink/msp430f2274.pdf.

[20] C.C. Cutler, "Differential Quantization of Communication Signals," *U.S. Patent 2605361*, 1952.

[21] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer Networks*, vol. 38, pp. 393–422, 2002.

[22] K. Kredo II and P. Mohapatra, "Medium Access Control in Wireless Sensor Networks," *Computer Networks*, vol. 51, pp. 961-994, 2007.

[23] A. El-Hoiydi, "Aloha with preamble sampling for sporadic traffic in Ad hoc wireless sensor networks," in *Proc. IEEE Int. Conference on Communications (ICC 2002)*, New York City, NY, USA, pp. 3418-3423, 2002.

[24] J. Polastre, J. Hill, and D. Culler, "Versatile Low Power Media Access for Wireless Sensor Networks," in *Proc. 2nd Int. Conference on Embedded Networked Sensor Systems (SenSys 2004)*, Baltimore, MD, USA, pp. 95-107, 2004.

[25] G. Halkes, T.V. Dam, and K. Langendoen, "Comparing Energy-Saving MAC Protocols for Wireless Sensor Networks," *Mobile Networks and Applications*, vol. 10, pp. 783-791, 2005.

[26] C. Merlin and W. Heinzelman, "Network-Aware Adaptation of MAC Scheduling for Wireless," in *Proc. Int. Conference on Distributed Computing in Sensor Systems (DCOSS 2007), Poster Session*, Santa Fe, NM, USA, pp. 24-28, 2007.

[27] S. Chen, J.S. Brantley, T. Kim, and J. Lach, "Characterizing and minimizing synchronization and calibration errors in inertial body sensor networks," in *Proc. 5th Int. Conference on Body Area Networks (BodyNets 2010)*, Corfu Island, Greece, pp. 138-144.