Al-Yad: A Wearable Sensor Network over DDS Middleware for Industrial Application

Basem Almadani\textsuperscript{a}, Farouq Muhammad Aliyu\textsuperscript{a,}\textsuperscript{*}, Elhadi M. Shakshuki\textsuperscript{b}

\textsuperscript{a}Computer Engineering Department, King Fahd University of Petroleum and Minerals (KFUPM), Dhahran, Saudi Arabia
\textsuperscript{b}Jodrey School of Computer Science, Acadia University, Wolfville, NS Canada

Abstract

Recently, it is estimated that in the coming five years, 700 million wearable devices for all kinds of applications will be sold worldwide. They owe their fame to their ease of accessibility, which in turn ensures productivity. Meanwhile, the need for increase in productivity in industries is at its highest. Many companies are looking for tools to increase the productivity of their employees. One of the main tools welcomed by the industries is handheld devices. This includes tablets, PDAs and mobile phones. However, these tools are not designed for applications that require marathon use — for applications such as machinery maintenance and inventory, require more subtle tools. This paper presents a proposed intelligent glove named Al-Yad. Al-yad is designed to control actuators in an industrial environment.

Keywords: ALYad; RTI; DDS; RIT Connext DDS; ubiquitous computing; Internet of Things; Embedded systems.

1. Introduction

Wearable Sensor Network (WeSN) is a field of Wireless Sensor Networks (WSN), where sensor nodes are designed to gather physiological and kinetic data from wearers' body\textsuperscript{1}. Embedded devices are becoming more intelligent and soon become the third layer of the Internet, i.e. Internet of Things\textsuperscript{2}.

Despite the advantages of WeSN, little research is available in this field with regards to industrial applications. This is due to the common belief that wireless communication is not reliable and therefore is not usable in such applications. However, recent advances in wireless technology enable it to compete with its wired counterpart.

This paper presents and discusses the development and analysis of a proposed, reliable industrial based Networked Control System (NCS), using WeSN. NCS is a Distributed Control System (DCS) where sensors, actuators and controllers communicate over a network\textsuperscript{3}. In order to ensure real-time performance, Real Time Innovations (RTI) Connext DDS is utilized to allow the system to connect with other devices over wireless network\textsuperscript{4}. The main aim of this work is to design intelligent wearable device with the ability to replace the current hand-held devices used in industrial NCS applications. This provides a great help in the industrial environments that require mobile users with...
tasks such as maintenance, mobile monitoring and control. We do believe that users are in favor and keen to use wearable devices than handheld devices. This is because they are lighter, cheaper, consume less energy and more convenient to carry around.

The remaining sections of this paper are as follows: Section 2 presents recent research carried out on wireless/wearable sensor network over DDS middle-ware. Section 3 provides detail description of the proposed Al-Yad wearable sensor. Section 4 explains the performed experiments to evaluate the performance of the proposed system. Section 5 discusses the achieved results. Finally, section 6 concludes the achievements of the proposed research and presents the proposed future work.

2. Previous Work

Wired control systems have been engineers’ favorite, because they are fast and reliable. However, this method of control is expensive to install and maintain. Current wireless communication systems are as reliable and fast as most wired systems, thereby making handheld devices applicable to NCS. Görlich et al.7 developed a NCS where a Profibus-to-Bluetooth connection is used to control and parameterize a Flow Unit via Nokia 6280 mobile phone. The system suffers from incompatibility with other phones, limited communication range and low bandwidth. Huang et al.5, developed an oil and gas storage and transportation simulation system using wireless local area network and an Android tablet PC, with the main objective to address the shortcomings of range and bandwidth problems encountered in7. The tablet PC communicates with programmable logic controllers (PLC) via PC station. However, the PC station that is between tablet PC and the PLC delays communication.

Recent developments in embedded systems allow researchers to replace tablets with Single Board Computers (SBC) like Beagle Bone, Intel Galileo and Raspberry Pi. SBCs have smaller footprint yet they are powerful processing devices capable of replacing phones, tablets and computers in controls systems and WeSN11. Wang et al., developed a wearable sensor network system using Raspberry Pi12. The developed system is used for Crowdsensing. It uses transcoding algorithm to reduce resolution of video files, as its memory is getting full. Khelil et al.13, developed a Body Sensor Network using Raspberry Pi. The system uses Cooking Hacks e-Health shield, which contains an airflow sensor for measuring wearer’s respiratory rate, pulse oximeter for measuring oxygen saturation of hemoglobin and position sensor for motion tracking. Jutila et al.14, use Intel Galileo as a sink node (or gateway) for a wearable system that helps to track children at school. All the aforementioned SBC systems showed promising results. However, they have one common problem that is incompatibility between the different types of SBCs. This forces scientists to stick to one type of device throughout their system.

In order to reduce the incompatibility and mask the heterogeneity of sensor nodes in mission-critical applications, Park et al. in15 proposes a Middleware. This Middleware is capable of hiding complexity and heterogeneity between system components. It eases the management of the system’s resources and improves the predictability of the system. Zhai and colleagues developed a publish/subscribe middleware called Wireless Message-Oriented System on top of TinyOS. They managed to reduce overall the energy consumption of the network through the use of self-adaptive QoS and content/topic model. Zug et al.18 successfully applied a publish/subscribe middleware called Family of Adaptive Middleware for autonomOUs Sentient Objects to an industrial automation system using Katanta robot. The robot publishes its coordinates and angles and it subscribes to movement–, speed– and emergency-stop commands. In all cases, authors reported increase in the performance using the proposed middleware.

The proposed wearable sensor system architecture is developed in the form of intelligent hand glove that accepts hand gesture as input, process it using ARM microprocessor (Raspberry Pi) and publish it using RTI connext professional (publish/subscribe middleware) to all subscribing equipment over wireless network. With the proposed system architecture we are able to address the data processing constraints, the large hardware footprint problem and the communication delays issue.

3. The Proposed System

The proposed system architecture is shown in Figure 1. It consists of the following four main components; 1) A wearable sensor, Al-Yad. The main function of this component, is to convert the physical movements into electric signal. The electric signal is then conditioned so that it can be processed by the 3.3v ARM-6 microprocessor, Raspberry
2) A Real-time Innovations Connext Professional is selected as a middleware to mediate between the wearable sensor and other computing devices of the system. 3) A computing device is connected to the targeted actuator. The computing device can be any device capable of running Connext Professional (e.g. PC, Beagle Bone, Intel Galileo or Raspberry Pi). 4) Actuator that represents any industrial machinery and that is required to be controlled remotely.

3.1. Al-Yad

Figure 2 shows an intelligent hand glove named Al-Yad. It comprises of a combination of light sources, e.g. Light Emitting Diode (LED) and light sinks, e.g. Light Dependent Resistor (LDR). The LDR and the LED are in line-of-sight when fingers are straight, thereby sending a high signal to the Raspberry Pi through a signal conditioning circuit, as shown in Figure 3. However, if the user bends his finger the light stops reaching the LDR and this raises
the resistance of the LDR. Hence, sending a low output to the Raspberry Pi. In order to ensure no ambient light contaminates the signals, the LDR is placed in a hollow cylinder of 1.45cm depth. Furthermore, the light source is pulsed at a frequency of 5KHz, which is the highest frequency of PWM allowed by "wiringpi.h" implementation. The highest frequency available is chosen, because the higher the frequency the smaller the size of the capacitors required. This approach ensures a lighter hand glove.

The signal conditioning circuit consists of a network of capacitors, diodes and inverting buffers, as shown in Figure 4. The circuit is responsible for smoothing the pulsing signal to Direct Current (DC) signal, which can be understood by the Raspberry Pi. The Raspberry Pi is programmed to scan the glove five times in one second (5Hz), which is little higher than value used in. Consequently, the Raspberry Pi converts the gloves signals to an integer value and publishes it over the middleware.

3.2. Middleware

Research has shown that DDS Middleware can achieve up to 75% efficiency in mobile nodes. This efficiency value has encouraged us to utilize a middleware approach in our proposed solution. The middleware chosen for
Table 1. QoS used in the proposed systems middleware.

<table>
<thead>
<tr>
<th>Quality of Service</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Durability</td>
<td>kind = VOLATILE</td>
</tr>
<tr>
<td>Reliability</td>
<td>kind = RELIABLE, max_blocking_time = 100ms</td>
</tr>
<tr>
<td>Liveliness</td>
<td>kind = AUTOMATIC, lease_duration = 1s</td>
</tr>
<tr>
<td>History</td>
<td>kind = KEEP_LAST, depth = 5</td>
</tr>
<tr>
<td>Presentation</td>
<td>access_scope = TOPIC, ordered_access = True, coherent_access = False, kind = EXCLUSIVE</td>
</tr>
<tr>
<td>Owner</td>
<td>5</td>
</tr>
<tr>
<td>Owner_Strength</td>
<td>minimum_separation = 0.2s</td>
</tr>
</tbody>
</table>

this task is RTI Connext Middleware. This middleware is chosen because of its strict compliance to the Object Management Group (OMG) specification for Data Distribution Service middleware. Moreover, certain Quality of Services (QoS) is set in order to ensure maximum efficiency, as recommended in.

Table 1 shows the quality of services used in the proposed system. In order to ensure compatibility, the QoS is configured in both the sensor and the actuator.

4. Experiments

The LDRs casings are designed using Blender2.6 3D modeling software and printed with Replicator2 3D printer. The different electronic components are assembled and described in an environment, as shown in Figure 5. The RTI subscriber is installed on the PC labeled Testbed and the publisher on the Raspberry Pi. The PC receives the gesture commands sent by the Raspberry Pi. All experiment is carried out base on this setup.

During these experiments, three key parameters are measured in order to assess the performance of the proposed system. This includes throughput, latency and power consumption of the system.

In order to test latency and throughput, RTI’s performance toolkit ‘Perftest’ is used. To test latency of the system, packets are sequenced and time stamped before they are transmitted to the Data-reader (receiver). The packets are immediately echoed back to the transmitter where the Round Trip Time (RTT) is calculated. In other to calculate the throughput the receiver counts the number of packets it receives per second. At the end of the experiments, the transmitter reports the latency of the system while the receiver reports its throughput. The smallest packets size allowed by RTI connext is chosen (i.e. 28 byte). This is due to the fact that, our proposed system is sending integer values only. This experiment is carried out using QoS settings (see also: Table 1) and without QoS so as to investigate the cost of applying QoS to the system. Figures 6 and 7 show the results obtained for both throughput and latency of the system respectively.

![Fig. 5. Experiment setup for measuring Latency and Throughput](image-url)
The second experiment is carried out to ascertain the time response of the transducer itself. To carry out this test, the hand glove is connected to an oscilloscope and the output voltage across the GPIO port is fed to an oscilloscope. A delay of 20ms is observed.

Finally, an experiment is setup to measure power consumption of the whole system. A 1Ω resistor is connected in series to the system so that the same amount of current passes through the system and the resistor, as such the voltage drop across the 1Ω resistor equals to the current passing through the system. An Analogue to Digital Converter is connected across the 1Ω resistor. It samples the voltage drop across the resistor every 200ms. The acquired data is then transmitted to a computer via USB, where it is stored. The current flowing through the system is recorded for two transmissions after which the Wi-Fi dongle is immediately removed so as to record the energy overhead of the dongle itself.

![Throughput without QoS Vs No.of Packets](image1)

![Throughput with QoS Vs No.of Packets](image2)

**Fig. 6. Throughput of the proposed system**

![Latency Vs No.of Iterations](image3)

**Fig. 7. Latency of the proposed system**
5. Results Discussion

During our experiments, we observed that there is a little difference in the throughput of the system with or without using QoS as shown in Figure 6. It is found that the average throughput of the system is 769 Mbps with QoS and 778 Mbps without QoS. In both cases no packets were lost during the 100,000 packets transmission. Since we are only sending integer values, this result is acceptable. Moreover, the QoS setting has little effect on the performance of the system.

With respect to our experiments towards latency, the results achieved are demonstrated in Figure 7 for the system with and without QoS. From Figure 7, it can be seen that there is also a little difference (1ms). The total latency of the system is the sum of latency due to transmission and latency due to the hand glove. In this paper, the latency of the smart glove is define as the duration from a fixed hand gesture to when a steady signal is available on the output bus of the glove. The latency of the hand glove is found to be 20 ms. Therefore, the total delay of the system is 23 ms.

Finally, Figure 8 shows the current (top) and the moving average of the current (bottom) that is flowing through the system during and after data transmission. Immediately after the second transmission, the WiFi dongle is removed; hence the spike at 375th second. However, no change of flow in current is encountered after removing the dongle. This shows that the energy overhead comes from data transmission itself. The average power dissipated is 310mW during transmission and 220mW while the system is not transmitting.

6. Conclusions and Future Work

In this paper, the possibility of using WeSN in wireless industrial control systems to complement mobile phones, tablets and computers is investigated. It is found that the system has a delay of approximately 23ms, which considered is fast enough for controlling motors, pneumatic actuators and other mechanical actuators. Furthermore, the maximum power dissipated by the system is 310mW. Therefore, the system can be powered by a mobile phone’s Lithium-ion battery (3100mAh) for up to 10hrs of continuous data transmission. As such, the system can be battery powered, since it can go into sleep mode when it is not transmitting.

Our plan for future work is to focus on developing a smart actuator with the ability to publish its state. This makes the smart glove able to receive feedback from the actuator. Moreover, the actuator can be programmed such that it can publish system failure and its possible cause.
Acknowledgment

The authors would like to acknowledge King Fahd University of Petroleum and Minerals and Real-Time Innovations Inc. for their support.

References


22. (OMG), O.M.G. Data distribution service for real-time systems version 1.2. 2007. URL: http://www.omg.org/spec/DDS/1.2/PDF/.

