

Applications based on electromyography sensors

Robert Kristof^{1,*}, Cristian Moldovan¹, Valentin Ciupe¹, Inocențiu Maniu¹ and Magdalena Banda¹

¹Politehnica University Timisoara, Mechatronics Department, 1 Mihai Viteazu Boulevard, Timisoara, Romania

Abstract. This paper presents our work regarding two different applications that use the electromyography sensors incorporated in the Thalmic Labs Myo Armband. The first application is about the Human-Machine Interface (HMI) for controlling an industrial robot by creating the environment for the user to control a robot's gripper position just by moving his arm. The second application refers to the real time control of a tracked mobile robot that is built with the Arduino development board. For each application the system design and the experimental results are presented.

1 Introduction

Industrial robots have found their place in a wide range of technological processes, replacing the human operator in performing auxiliary or basic operations. The most important applications are in the following areas: in mechanical machining processes for the automatic supply of parts, tools or devices for CNC machines, or for drilling or grinding operations [1]. As technology developed, industrial robots can be connected with sensors and different tools that could enhance the operation.

Aside from the conventional sensors and vision methods, the use of biological surface electromyography (sEMG) sensors is a low-cost method for detecting and identifying human motions, such as hand and limb motions. The electrical activity of muscle fibers during a contraction generates the sEMG signals, and then the electrodes attached to the skin record the sEMG signals in a non-invasive manner [2]. The corresponding human motions can be detected and recognized by detecting certain muscle contraction patterns, and the detected motion can be remotely duplicated using artificial limbs or robotic hands [3].

In order to analyse the EMG signal, for these applications we have used the Thalmic Labs Myo Armband (Fig. 1) which is an armband that is fixed on the user's forearm, and by moving the arm and fingers, the MYO armband detects muscle activity signals which are processed and used to control different objects and software. The MYO is made out of a series of rectangular pieces of plastic, that are fixed together with rubber. Inside each plastic piece there are three Electromyography Sensors (EMG) that detect electrical

* Corresponding author: robert.kristof@yahoo.com

impulses from hand muscles. The bracelet can be adjusted depending on the wearer's hand by means of tightening the elements. The bracelet has a LED indicator that turns green to show that the bracelet is charging and blue when connected to the computer. From the dimensional point of view, the MYO has a circumference of 19.5 cm and can be extended to 33 cm. The bracelet weighs 93.55 grams only and each rectangular element of the bracelet is 1.14 cm wide. [4]

In this paper the main purpose is to establish a communication between the Thalmic Labs Myo Armband and different devices, in order to control them with gestures. As objectives, we will analyse, using the Myo Armband, how to control the gripper of a KUKA KR15 industrial robot and a tracked mobile robot.



Fig. 1. MYO Armband [5]

2 System design

2.1 KUKA KR15 industrial robot

This application consists of three elements: The Myo Armband, a PC and the KUKA KR15 industrial robot. The armband is communicating with the PC using Bluetooth. The data is received using the Bluetooth dongle, then the information is interpreted, using a C++ program, into movement data that is sent to the KUKA KR 15 industrial robot, using a RS232 serial connection. (Fig. 2. a.)

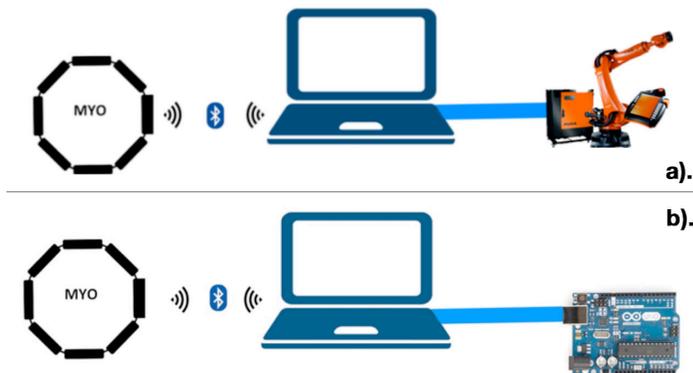


Fig. 2. Connectivity between the Myo Armband and a). the KUKA KR15 industrial robot; b). the Arduino development board

The robot system, shown in Fig. 3, consists of the following components: 1 – robot; 2 - connecting cables; 3 - robot controller, 4 - teach pendant (KCP – KUKA Control Panel), software, options and accessories. All motor units and current-carrying cables are protected against dirt and moisture beneath screwed on cover plates [6].

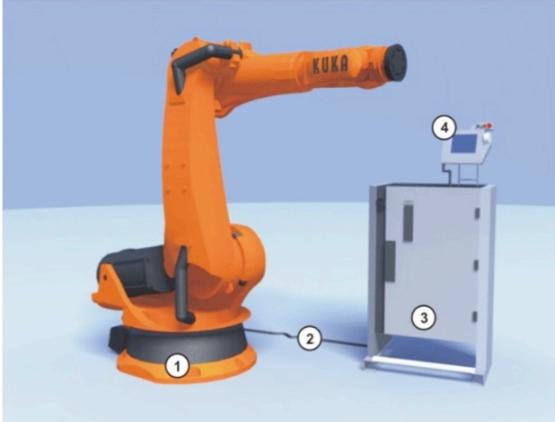


Fig. 3. KUKA KR15 industrial robot elements

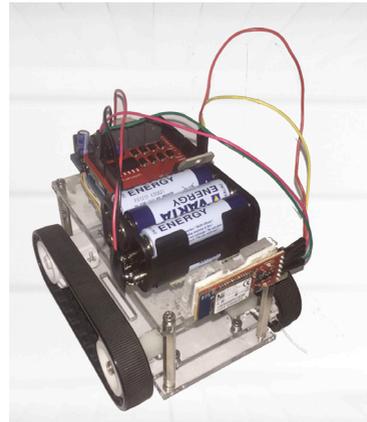


Fig. 4. Arduino tracked mobile robot

2.2 Tracked mobile robot

In this case we created a tracked mobile robot which uses two DC motors connected to the L298 motor driver shield which is linked to the Arduino development board. (Fig. 4) This application has a similar architecture to the first, but instead of sending the data to the KUKA KR15, the PC sends the information to the Arduino development board which manipulates the data into sending the suitable power to the DC motors in order to execute movements. (Fig. 2. b.)

3 Experimental results

Similar research involving a virtual robotic arm created in Unity 3D and controlled with the Myo Armband show that the best way to detect electro activities in muscles is to use the EMG sensor. The EMG sensor allowed to obtain very clear and important data from forearm muscles that were used to control a virtual robotic arm. [7]

3.1 KUKA KR15 industrial robot

The Myo Armband offers the possibility to receive information like gyroscope, accelerometer, orientation, pose or vibration.

In order to send the appropriate data, we calculated the roll pitch and yaw values, starting from the unit quaternion of the Myo Armband.

Euler and roll-pitch-yaw angles are routinely used to represent the orientation of rigid bodies in aerospace, navigation, and robotics because they minimize the dimensionality of the control problem. Both representations, however, introduce unwarranted mathematical singularities which are identified in this paper. Trajectory-tracking algorithms break down at singularities and cause loss of control. Since mathematical singularities do not reflect physical limitations of orientation, remedial measures can be implemented in the controller. [8] The yaw, pitch and roll elements are represented in Fig. 5.

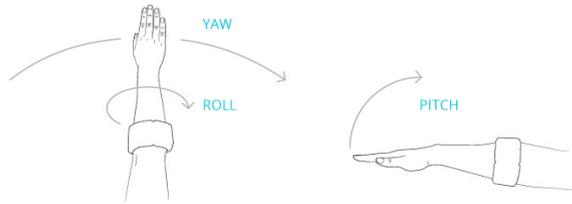


Fig. 5. Yaw, pitch and roll elements [9]

We started calculating the roll, pitch and yaw angles using the code shown in Fig. 6.

```
float r = atan2 ( 2.0f * (quat.w() * quat.x() + quat.y() * quat.z()),
                1.0f - 2.0f * (quat.x() * quat.x() + quat.y() * quat.y()));
float p = asin (max ( -1.0f, min (1.0f, 2.0f * ( quat.w() * quat.y() - quat.z() * quat.x()))));
float y = atan2 ( 2.0f * (quat.w() * quat.z() + quat.x() * quat.y() ),
                1.0f - 2.0f * (quat.y() * quat.y() + quat.z() * quat.z()));
```

Fig. 6. Code to determine the roll, pitch and yaw angles [10]

To determine the angles, we transformed the floating point angles to radians and scaled the elements from 0 to 700, in order to move the KUKA KR15 industrial robot from 0 to 700 on the x, y, z axis. (Fig. 7)

```
roll = static_cast <int> ( ( r + (float) M_PI ) / ( M_PI * 2.0f ) * 700);
pitch = static_cast <int> ( ( p + (float) M_PI / 2.0f ) / M_PI * 700);
yaw = static_cast <int> ( ( y + (float) M_PI ) / ( M_PI * 2.0f ) * 700);
```

Fig. 7. Code to transform roll, pitch and yaw angles to radians [10]

Then the data was sent to the KUKA KR15 industrial robot via serial connection. Using the CREAD function we managed to get the x, y, z values and transform them into positioning data for the gripper. The KUKA KR15 motion program flowchart is presented in Fig. 8.

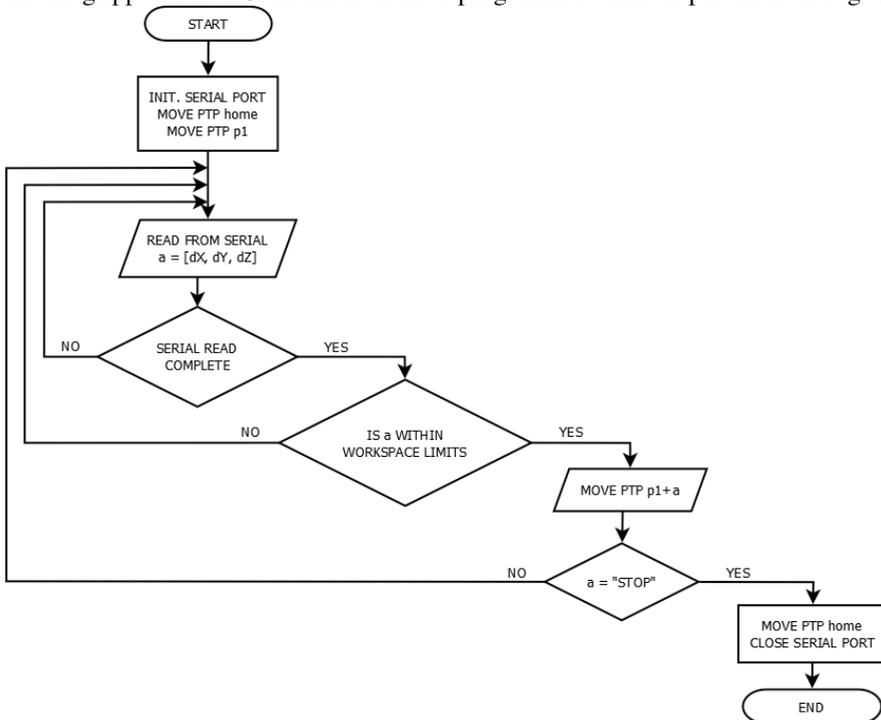


Fig. 8. The KUKA KR15 motion program flowchart

The values resulted from the formulas presented in Fig. 7 are sent to the KUKA robot and represent the distance travelled by the gripper on the x, y, z axis. As we move the hand and modify the roll, pitch and yaw values, the KUKA industrial robot moves its gripper in real time as presented in Fig. 9.

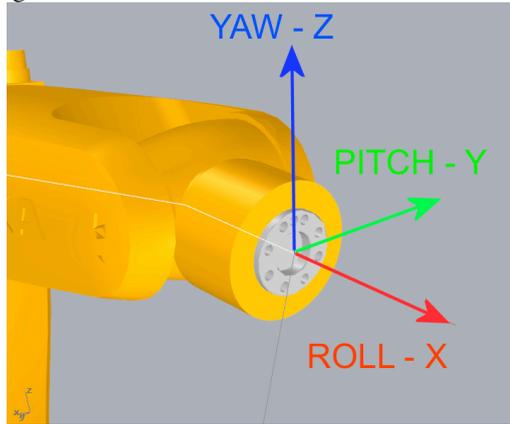


Fig. 9. Correlation between roll, pitch, yaw and KUKA axis [11]

3.2 Tracked mobile robot

The program flowchart (Fig. 10) shows how every gesture is converted into data that is sent to the Arduino board and then into power that is sent to the DC motors.

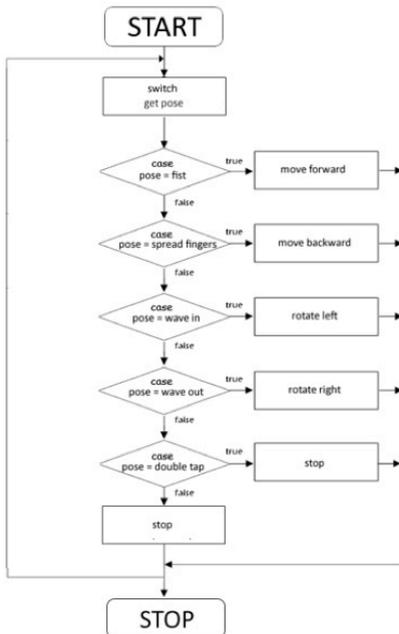


Fig. 10. Arduino program flowchart

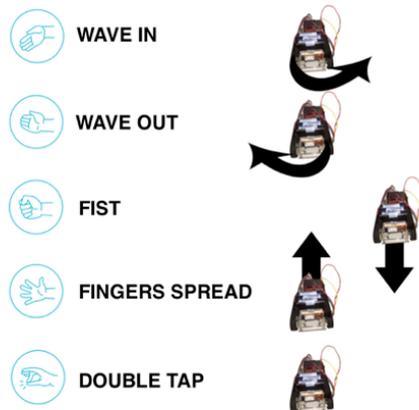


Fig. 11. Tracked mobile robot movement by gesture

The robot executes moves depending on the gesture done by the user (Fig. 11):

- Wave in – rotate left;
- Wave out – rotate right;

- Fist – move forward;
- Fingers spread – move backward;
- Double tap – stop.

4 Conclusions

Electromyography sensors are great tools to help you transform gestures to movement, especially if you use the Thalmic Labs Myo Armband which is a friendly environment due to all of the software capabilities.

This paper presented two applications that transformed gestures to movement. The first application was about moving the KUKA KR15 gripper position which was a very interesting experiment as we had to establish the communication between the Myo Armband and the KUKA robot. Because of the serial communication and the way KUKA handles information using only standard software, there was a delay sending information, so we encountered difficulties into establishing a smooth movement of the gripper. As future work we must find a more appropriate way to synchronise the movement of the hand with the motion of the gripper. In the case of the Arduino tracked mobile robot, we can recall some difficulties on the communication and control part due to the big amount of data that is transferred over USB. As future work, we could transmit the data to the Arduino tracked mobile robot using a wireless network.

References

1. M. Crenganiş, A. Bârsan, S. G. Racz, M. D. Iordache, *Single point incremental forming using kuka kr6-2 industrial robot-a dynamic approach*, Proceedings in Manufacturing Systems, **13**, 133-140 (2018)
2. R. Merletti, P. Parker, *Electromyography: Physiology, Engineering and Noninvasive Applications*, 82–89 (John Wiley & Sons, Inc: Hoboken, NJ, USA, 2004)
3. C. Cipriani, C. Antfolk, M. Controzzi, G. Lundborg, B. Rosen, M.C. Carrozza, F. Sebelius, *Online Myoelectric Control of a Dexterous Hand Prosthesis by Transradial Amputees*, IEEE Trans. Neural Syst. Reh. En, **19**, 260–270 (2011)
4. Myo techspecs, accessed on 25.05.2018, <https://www.myo.com/techspecs>
5. Thalmic Myo Press kit, accessed on 25.05.2018, <https://www.thalmic.com/press>
6. M. Sathiyarayanan, S. Rajan, *MYO Armband for physiotherapy healthcare: A case study using gesture recognition application*, COMSNETS (2016)
7. A. Ganiev, H.-S. Shin, K.-H. Lee, *Study on virtual control of a robotic arm via a myo armband for the selfmanipulation of a hand amputee*, Int. J. Appl. Eng. Res, **11**, 775-782 (2016)
8. M. H. Ang, V. D. Tourassis, *Singularities of Euler and roll-pitch-yaw representations*, IEEE Trans. on Aeros. and Electr. Sys, **23**, 317-324 (1987)
9. GUI without the G: Going Beyond the Screen with the Myo™ Armband, accessed on 25.05.2018, <https://developerblog.myo.com/gui-without-g-going-beyond-screen-myotm-armband>
10. Myo Connect, SDK and firmware downloads, accessed 24.01.2019, https://support.getmyo.com/hc/article_attachments/360014130291/myo-sdk-win-0.9.0.zip
11. Robot programming with Kuka|prc, accessed on 24.01.2019, <http://mkmra2.blogspot.com/2016/01/robot-programming-with-kukaprc.html>