Human Arm Motion Capture using Gyroscopic Sensors

Arun Fabian Panaite[1], *Monica* Leba[1*], *Marius Leonard* Olar[1], *Remus Constantin* Sibisanu[1] and *Lilla* Pellegrini[1]

[1]University of Petroșani, România

Abstract. By using the most rudimentary microcontroller chips, that receive data from sensors, and transmit the data to a computer system, thorough a virtual serial port, motion of many objects, bodies and joints can be captured. Capturing the motion and reproducing it live is not the only destination for the data usage. Recording and studying the motion data, can reduce a lot of work in a wide range of domains. Using the simplest methods to capture the data, also means making it so widely accessible for learning, editing and also developing systems that use very little processing power, granting data access for the less efficient computers. We propose using the MPU-6050 MEMS sensor in a dual instance, and the Arduino UNO microcontroller, connected to a computer for data acquisition, to capture the motion of a human arm, and reproduce it in a projected environment. Other experiments, conducted by other researchers and developers have used a higher number of sensors, and the data acquisition and recording systems were much more complex, but our research reduced the number of sensors to just two. One of the high impact innovations brought by this system, in particular, is that we've virtually hooked the end of one sensor to the tip of the other, creating a virtual motion chain.

1 Introduction

During the last decade, multiple approaches on the techniques and methods of motion capture have been the guidelines of the motion capture processes, regardless of the work domains they've been incorporated in [1]. Multiple inertial sensors [2,3], ranging from accelerometers, magnetometers, gyroscopes, cameras, infrared sensors have all been put to the test in the race to which of the sensors were most fit for the job [4-6].

These sensors each have their own way of perceiving motion or displacement.

Here's a little brief approach of each category:
   1) Accelerometers measure the acceleration that they are moved with, simply said.
   2) Magnetometers, even if only useful in magnetic fields can theoretically detect the mechanical torque from a magnetic perspective.
   3) Gyroscopes perceive angular velocity, by the type of their construction, they can be built for a) spinning rotors; b) spring lasers; c) vibrating mass.

---

* Corresponding author: monicaleba@yahoo.com

4) Cameras, usually stereoscopic, can have depth of field perception, for motion capture in particular.

5) Infrared sensors can also work as if they were distance measuring sensors due to the response time, using amplitude response.

The technique we are using is, however, one that employs the MPU6050 a 3-axis accelerometer and gyroscope sensor, capable of communication with the I2C interface, requiring only two connections. Two such devices are connected to an Arduino UNO development board, all these, connected to a computer's USB port, while using the Arduino IDE and the Processing IDE. Not only have we chosen the above-mentioned devices and software for very low resource consumption reason, but also for operator safety, the voltages never getting higher than 5V.

2 State of the art

Wearable devices for motion capture have proliferated incredibly during the last few years [7, 8] yet, in the future years may become obsolete, as non-invasive technologies are on the rise [9]. Gait (or other motion analysis) is well-established [10] in clinical assessment of movement disabilities. The causes being numerous for the disabilities, reproducing the correct movements is an achievable desire. The motion capture systems used vary, depending on multiple forms of demand, for example we have Fig. 1 [11] and regarding, of course the body area treated, the distribution chart below can express the demand results - stating this as leverage for sustaining the idea that no work of motion capture, treatment, or reproduction of motion capture and treatment technology is to be researched without demand, implying purpose.
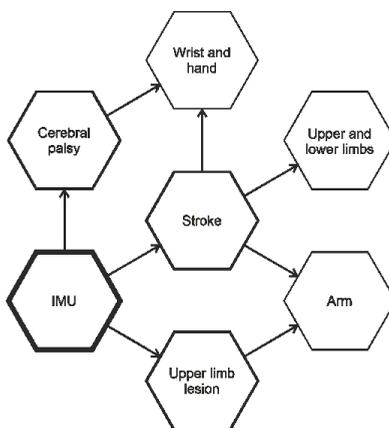
Fig. 1. Some clinical conditions that have been studied using IMU (inertial measurement unit).

The trajectory of research that we've taken is shown in the figure below (Fig. 2).

Our system is a non-optical motion capture system that uses multiple mathematical and computational approaches. Non-optical systems are mostly based on inertial sensors with incorporated accelerometers, gyroscopes and magnetometers, to allow recordings of the movement associated data, in an integrated device, being low cost, precise and wearable. IMU sensors are already part of the medical domain, in the detriment of requiring supervised try-outs and drift-avoidance configurations [12].
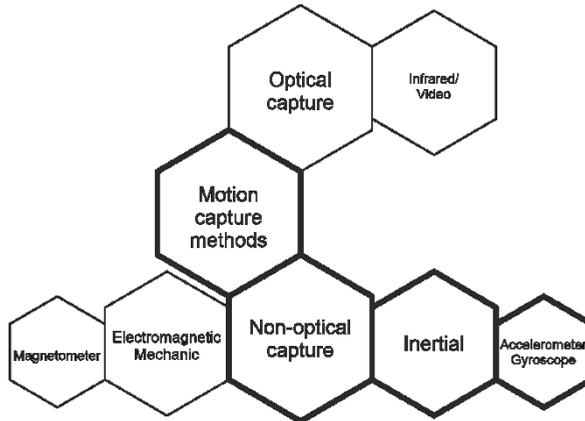
Fig. 2. Main motion capture system methods.

In the study of exoskeletons and human-computer interaction, the motion-capture technologies can also involve and employ machine learning and neural networks [13]. That kind of an approach is also important not only to analyse the recorded datasets, but also to generate more natural motion and, of course, more ergonomic motion, respectively. In deep learning there are a few different architectures that can be used for different purposes [14]. Deep Belief Networks are composed of several fully-connected layers. Convolutional Neural Networks are inspired by the human visual cortex hierarchical structure. Recurrent Neural Networks are specialized in processing value series and sequences, having the ability to capture long-distance dependencies.

Virtual reality [15] is also a domain where our work can be applied, more so, virtual reality has the power of transforming the data that our system records and generates, into whatever data can be imported, reproduced and manipulated in a virtual environment. In the field of virtual reality, the benefit can be indoor exercising in confined spaces, yet give the users (patients) better insight [16] of their movement.

3 System design

The data trajectory from our system works according to the following figure:
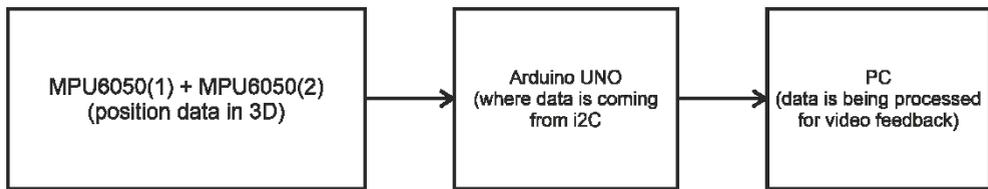
Fig. 3. A concise version of the data trajectory, from the data acquisition, to the user's interpretation of the movement.

The data from the two MPU6050 sensor kits is being picked up, sent to the Arduino Uno board thorough the I2C Interface, the Arduino UNO sends the data to the PC thorough a serial port (in our case 115200) through port COM 5 (a virtual serial port). In the PC, the quaternion data is being processed in the software (Processing IDE) here the data is not only processed in the sense that it's stored, but also in the sense that the data is being processed into the motion of the 3D rendered arm shaped object, this way, the data is used, live, in a discrete type of processing (8kHz sample rate), which displays the motion of the 3D rendered arm at a desired framerate (up to 120 FPS). We used the data samples with the

default error rate published in the MPU6050 datasheet, taken "as is" therefore we did not test the sensor data precision, but the captured motion seemed as natural as one could see. Future experiments and research will subject all the errors to exhaustive testing, dedicated to bettering the system. One of the major advantages regarding the video processing output is that it uses the classic OpenGL rendering interface that comes, nowadays with any stock video card for desktops and laptops, cross platform and running smoothly even in software rendering for the non-OpenGL capable video cards. But there's a catch: if the Arduino Uno board isn't tested with the File.ino script for reading the data from known addresses from the sensors and known virtual ports, the Processing IDE program will not know how to acquire the data. Neither would the Arduino Uno, on its own, be able to properly separate the read values.
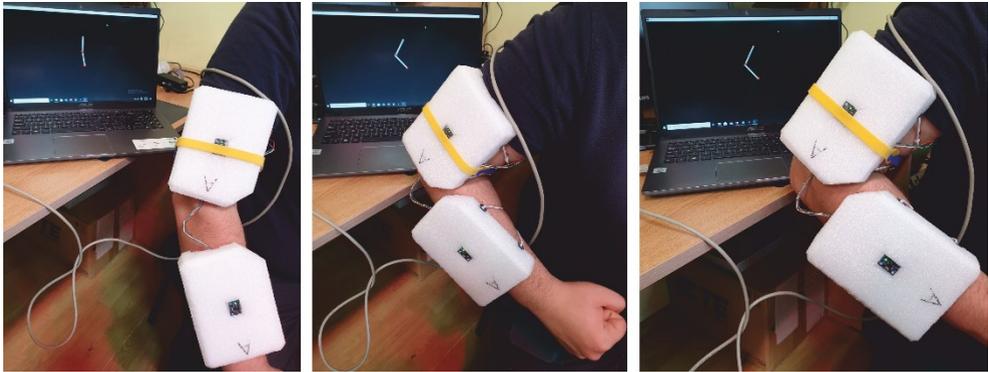

Fig. 4. Our experiment.

Connecting both sensors to the Arduino UNO was very thicky, as the addressing system is hardwired into the sensor cards, therefore there is a special pin configuration that changes the port's address. In the Processing IDE software, replicating the MPU6050 movements to the animated arm it is necessary to convert the serial data from the two sensors into quaternion data values.

//The serial data we receive from the sensors (the two MPU6050) needs to be converted to quaternion data values

```
q[0] = norm(values[3],-255, 255);
q[1] = norm(values[0],-255, 255);
q[2] = norm(values[1],-255, 255);
q[3] = norm(values[2],-255, 255);
for (int i = 0; i < 4; i++){
if (q[i] >= 2){
q[i] = -4 + q[i];}}
quat.set(q[0], q[1], q[2], q[3]); // Setting up the quat count, to update the arm's positions in space
println("--Begin Data 1--");
println("q:\t" + round(q[0]*100.0f)/100.0f + "\t" + round(q[1]*100.0f)/100.0f + "\t" + round(q[2]*100.0f)/100.0f + "\t" + round(q[3]*100.0f)/100.0f);
println("--End Data 1--");
```

4 Mathematical model

Based on the frames from Fig. 5, applying the Denavit-Hartenberg formalism for the human upper limb, we get the following parameters (Table 1):
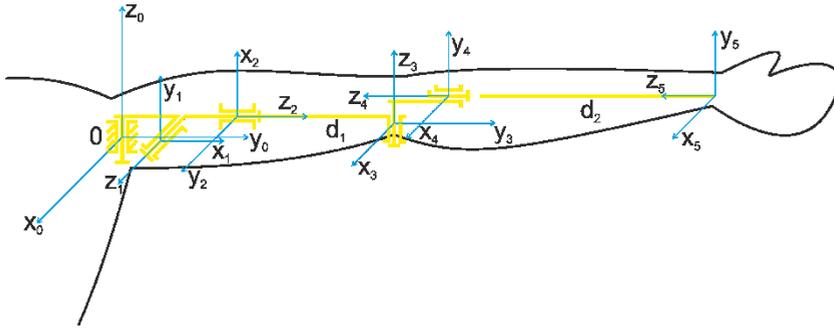
Fig. 5. The Denavit-Hartenberg model.

Table 1. Denavit-Hartenberg parameters

		Coordinate			
		a_i	α_i	d_i	θ_i
Element	1	0	$\dfrac{\pi}{2}$	0	$\dfrac{\pi}{2} + \theta_1$
	2	0	$\dfrac{\pi}{2}$	0	$\dfrac{\pi}{2} + \theta_2$
	3	0	$\dfrac{\pi}{2}$	d_i	$\dfrac{\pi}{2} + \theta_3$
	4	0	$\dfrac{\pi}{2}$	0	θ_4
	5	0	0	$-d_2$	θ_5

The general form of the Denavit-Hartenberg matrix that depicts the relative movement from one reference frame to the following one is:

$$T_{i-1,i} = \begin{bmatrix} cos\theta_i & -sin\theta_i \cdot cos\alpha_i & sin\theta_i \cdot sin\alpha_i & a_i \cdot cos\theta_i \\ sin\theta_i & cos\theta_i \cdot cos\alpha_i & -cos\theta_i \cdot sin\alpha_i & a_i \cdot sin\theta_i \\ 0 & sin\alpha_i & cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1}$$

So, we get the following matrixes, that, multiplied together, give us the movement matrix T_{05} of the final element in the base reference frame, that is the direct kinematic mathematical model:

$$T_{01} = \begin{bmatrix} -sin\theta_i & 0 & cos\theta_i & 0 \\ cos\theta_i & 0 & sin\theta_i & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}; T_{12} = \begin{bmatrix} -sin\theta_2 & 0 & cos\theta_2 & 0 \\ cos\theta_2 & 0 & sin\theta_2 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix};$$

$$T_{23} = \begin{bmatrix} -sin\theta_3 & 0 & cos\theta_3 & 0 \\ cos\theta_3 & 0 & sin\theta_3 & 0 \\ 0 & 1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}; T_{34} = \begin{bmatrix} cos\theta_4 & 0 & sin\theta_4 & 0 \\ sin\theta_4 & 0 & -cos\theta_4 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix};$$

$$T_{45} = \begin{bmatrix} cos\theta_5 & -sin\theta_5 & 0 & 0 \\ sin\theta_5 & cos\theta_5 & 0 & 0 \\ 0 & 0 & 1 & -d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix};$$

$$T_{05} = T_{01} \times T_{12} \times T_{23} \times T_{34} \times T_{45} \tag{2}$$

5 Conclusions

The use of robust, small sized components and the present technology allows us to build compact devices towards the advancement of wearables. The improvements of wearables, regardless of purpose, increasingly integrates wearable technology into everyday life. The multiple use sensors can provide a lot of data, therefore, in the future, the recording and management of datasets is more accessible, and the use of machine learning would be facilitated. The degrees of freedom regarding motion capture are easily enlarged by the methods of data transmission, conversion and processing.

Regarding affordability, not only is the cost per components very low, but for skilled users it could always become a "do it yourself project". It can also be stated that this project can be done with free software from one end to the other, this option becoming one of the very few equivalents to a professional exercise solution for both rehabilitation, recreational, and technical purposes.

Future research and development will involve testing the system on multiple users, and the recording of their opinions and experiences while using this motion capture system.

References

1. F. Brunetti, et al. Inter. Conf. of the IEEE Eng. in Med. and Bio. Soc., (2006)
2. M. Risteiu, M. Leba, A. Arad, Calitatea, (2019)
3. M. N. Risteiu, Leba M., World Con. on Inf. Sys. and Tech., (2020)
4. D. Gouwanda et al. 4th Kuala Lumpur Inter. Conf. on Biomed. Eng. (2008)
5. M. Risteiu, M. Leba, O. Stoicuta, A. Ionica, IEEE 20th (MELECON), (2020)
6. S. Rosca, M. Risteiu, M. Leba, N. Negru, M. Ridzi. MATEC Web of Conf. (2020)
7. J. He, et al. Inter. Journal of Adapt. Contr. and Sign. Process., (2019)
8. A. F. Panaite, M. N. Rişteiu, M.L. Olar, M. Leba, A. Ionica, IOP Conf., (2019)
9. F. Öhberg, et al. Sensors, (2019)
10. S. D. Rosca, M. Leba, A.F. Panaite, World Conf. on Info. Sys. and Tech. (2020)
11. A. C. Alarcón-Aldana, M. Callejas-Cuervo, A. Padilha, L. Bo, Sensors, (2020)
12. J. L. Samper-Escudero et al. Soft robotics, (2020)
13. J. H. Geissinger, Virginia Tech (2020)
14. E. Sansano, R. Montoliu, OB. Fernández, Computational Intelligence, (2020)
15. K. Vogel., Univ. of Twente, (2020)
16. M.Leba, A. Ionica, M. Risteiu, Industria Textila, (2020)