Design and Automatic System Control Implementation for an Imitator Robot in a Humanoid Platform through Image and Video Acquiring and Processing

Parra N. * **Rosales A.** * **Burbano P.** *

* *Escuela Politécnica Nacional, Departamento de Automatización y Control Industrial, Quito, Ecuador (e-mail: nelson.epn@gmail.com; andres.rosales@epn.edu.ec; carlos.burbano@epn.edu.ec)*

Resumen: *En la actualidad, los robots teledirigidos son utilizados para realizar acciones que son peligrosas o muy complejas para los seres humanos, tal es el caso del manejo de substancias radioactivas, búsqueda y rescate de personas en zonas inaccesibles, desactivación de bombas, entre otros. Además, los robots teledirigidos se utilizan en la rehabilitación de personas con problemas en sus extremidades, ya que se incentiva de forma interactiva el movimiento de las mismas, mientras el robot emula dichos movimientos o ayuda a dichas personas a realizarlos. El presente trabajo trata acerca de la emulación de los movimientos de las extremidades de una persona en tiempo real, mediante el sensor kinect el cual obtiene el registro de los datos de las articulaciones de la persona sin ningún tipo de contacto. Posteriormente, se interpretan los datos y se transforman mediante la técnica de cinemática inversa en ángulos, los cuales se envían a través de comunicación inalámbrica a la plataforma humanoide RoboPhilo, la misma que utiliza dichos ángulos para el movimiento final del robot. Todo el proceso se presenta como la teleoperación de un robot humanoide.*

Palabras clave: *Robot, Imitador, Cinemática Inversa, Control, Teleoperación, Kinect.*

Abstract: *Nowadays, teleoperated robots are used to execute dangerous or complex tasks assigned for human beings; a few examples of these tasks are managing of radioactive substances, searching and rescuing people in inaccessible zones, bombs disarming and so on. Teleoperated robots are also used for helping people from limb injuries because it interactively incentivizes the movement, meanwhile the robot emulates user's movements and helps the user to achieve them. This paper refers about emulation of limb movements in real time by using Kinect sensor which obtains joints position data from the user without any contact. Afterwards the sensor uses the data and transforms it using inverted kinematics giving final data as angles between limbs in degrees, those angles are send by wireless communication to RoboPhilo humanoid robot and it finally gets the Kinect data and make the servomotors move. All this process is presented as a teleoperated humanoid robot.*

Keywords: *Control, Robot, Humanoid Platform, Kinect, Teleoperation.*

1. INTRODUCTION

From the beginning of the fourteenth century, humans have been interested in means to perform repetitive tasks on machines that are accessible and have a level of security and performance. With the emergence of the first prototypes of robots could take a big step towards automation to industrial and let the world enjoy products, especially electronics at very affordable prices. Then he decided to apply robotics to other areas, including: exploration, medicine, entertainment, etc., with a very positive result.

Robotics now forms a key role both in industry and in the home, because of the easy handling, adaptation and performance that has different types of robots that the user requires. Robots today serve many functions critical to the development of new technologies, thanks to its speed and the level of security with which they work. One of the most important tasks of the robots is to replace workers in dangerous places and difficult to reach. This project intends to use basic teleoperation to show that there is a wide range of applications where robots can be used as in some dangerous environments where people can suffer injuries.

2. HUMANOID PLATFORM PERFORMANCE ANALYSIS

At present we have various types of robots, which can perform various activities, depending on the user need. The Humanoid robot platform designed by the company RoboPhilo Robobrothers Inc. is ideal for imitation tasks basic movements of a person, and has been designed with the following specifications:

- Height: 13" (330.2mm).
- Weight: 1.2kg (1200g) battery included.
- Controller: ATmega32 – 16PU.
- 24 available channels.
- Battery: 6V 700mAh NiMH.
- Charger 7.2V 1000mA (100-240 VAC 50/60Hz).
- Servomotors: SV 4032 (4.1kg-cm), SV 4104 (6.5kg-cm), SV 2030 (1.3kg-cm)[1].

Figure 1: RoboPhilo humanoid platform, taken from [1].

2.1 Chassis and Internal Structure

The humanoid platform chassis RoboPhilo has been made of plastic, very light material compared with metal designs, which previously has been given certain characteristics so that it can withstand the pressure exerted on it to be at rest and when performing certain movements.

2.2 Locomotion System

In order to perform the movements, the humanoid robot uses 20 servomotors RoboPhilo analog that provides 20 degrees of freedom (DOF). Servomotors have a DC motor that rotates at high speed, a series of gears to produce reduced rotational speed and increase torque capacity, a potentiometer connected to the output shaft (encoder) and a control circuit feedback.

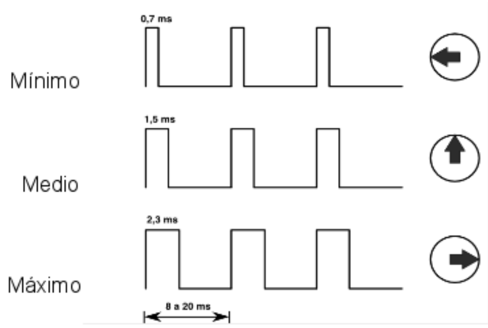

Figure 2: Signal and servo position [2].

2.3 Control Unit

This control unit is responsible for generating square wave signals by (PWM) for the movement of the actuators, plus get the data sent from the computer to then interpret and perform some control action.

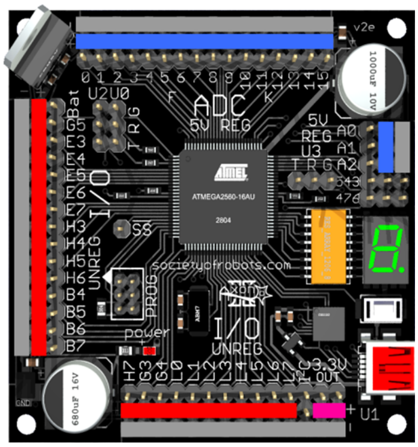

Figure 3: Axon II Control Unit.

The control unit Axon-II development has the following technical characteristics:

- 58-pin input / output in total.
- 16 channels of analog / digital conversion.
- Able to control over 25 servomotors *
- Communication I2C, SPI.
- Serial communication 3 + USB UART.
- 8 external interrupts.
- 15 PWM channels.
- 64 kB Flash.
- 4 kB EEPROM.
- 8 kB SRAM.
- 6 Timers (four 16-bit, and two 8-bit).
- Pre-programmed with bootloader programmer is not required.
- LED numeric display.
- 3.3V power bus, voltage 5V and unregulated.
- Support external memory (Port A) [3].

The servomotors specify the connection between PWM generated by hardware and software generated PWM.

2.4 Serial Communication

RoboPhilo humanoid platform has a serial communication port (Figure 4) through which you can make the programming of the different movements and calibration via a special cable (original control plate).

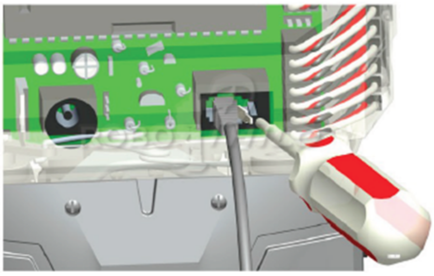

Figure 4: Wiring to the serial port RoboPhilo.

3. SYSTEM CONTROL DESIGN AND IMPLEMENTATION

The system control in Kinect sensor acquires video images with color CMOS sensor at a frequency of 30 Hz in color 32-bit RGB and VGA resolution of 640 480 pixels. The CMOS monochrome video channel is 16-bit, resolution.

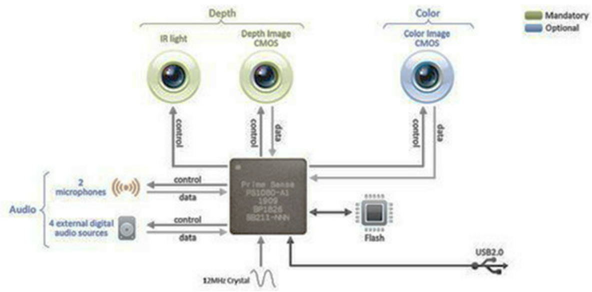

Figure 5: Internal composition, Kinect sensor.

The angle of view is 58^{o} horizontal and 45^{o} vertical. In addition the sensor has an internal servomotor allows oriented upwards or downwards by increasing the angle of view up to 27^{o}.
The array has four microphone capsules and operates with each channel processing in 16-bit audio with a frequency range of 16 kHz. The Kinect camera works with hardware and software for recognizing own image.

The camera has two main functions:

- Generate a 3D map of the image you have in your visual field.
- Recognize human movement between image objects from different segments of the joints of the body and a grayscale scheme.

In a large field of view with objects, the Kinect camera is about recognizing how far away the different objects, distinguishing movements in real time.

The Kinect sensor can distinguish the depth of each object with a resolution of 1 cm and estimates of the height and width with an accuracy of about 3 millimeters. The Kinect hardware consists of the camera and infrared spotlight, added to the firmware and a processor that uses algorithms to process three-dimensional images [4].

3.1 Signal Processing

The Kinect sensor processor is able to interpret the movements of objects that are recorded caught on camera in "meaningful events" on the screen. The movement sought by the algorithm is contextualized. For example, if you are using the Kinect sensor to a game that requires bending over or lying down, the algorithm will search for the identification of these movements in real time to produce events in display [5].
The Kinect sensor takes the data provided by the identification chip depth and estimates the position of each of the joints of a person and then the person takes possession at a given instant. The estimation of the pose of a person succeeds, by testing patterns with different types of people of different sizes and textures.

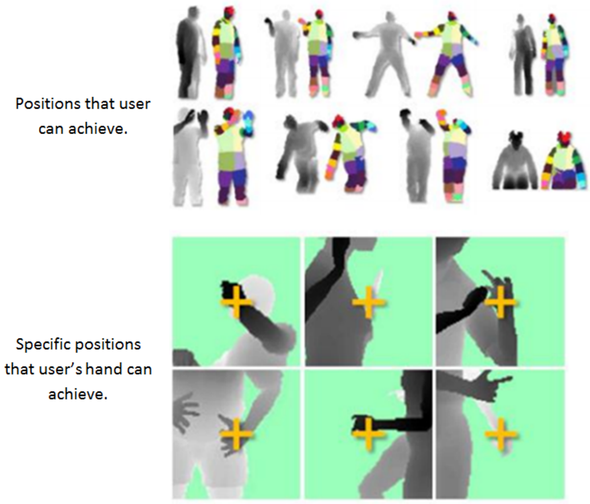

Figure 6: Estimating positions using the Kinect sensor.

3.2 Inverse Kinematics

In order to send the correct angles to actuators RoboPhilo, humanoid platform should be performed prior to process data obtained from the Kinect sensor, a process known as inverse kinematics.
Most basic robots have usually relatively simple kinematic chains, and the first three degrees of freedom (DOF), which position the robot in space, usually have a planar structure. This condition facilitates the resolution of the n - tuple. Furthermore, the last three degrees of freedom are used for guidance, which allows the resolution decoupled (kinematic decoupling) of the tip position of the robot. As an alternative to solve the same problem can use to directly

manipulate the equations for the direct kinematic problem. That is, from the relationship between the transformation matrix and the equations based on the joint coordinates $q = [q_1, q_2, ..., q_n]$ n may clear the joint variables "qi" in terms of the components of vectors n, o, and p as shown in Equation (1) [6].

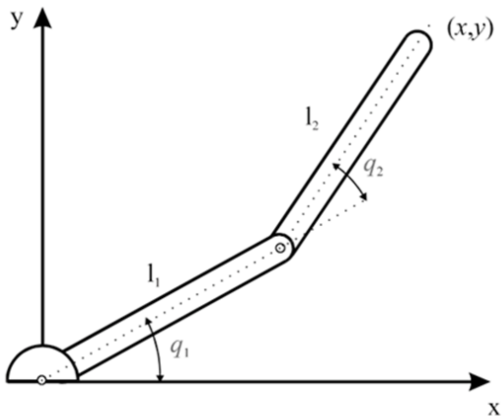

Figure 7: Joint coordinates of a robotic arm[5].

$$\begin{bmatrix} n & o & a & p \\ 0 & 0 & 0 & 1 \end{bmatrix} = [t_{ij}(q_i....q_n)] \tag{1}$$

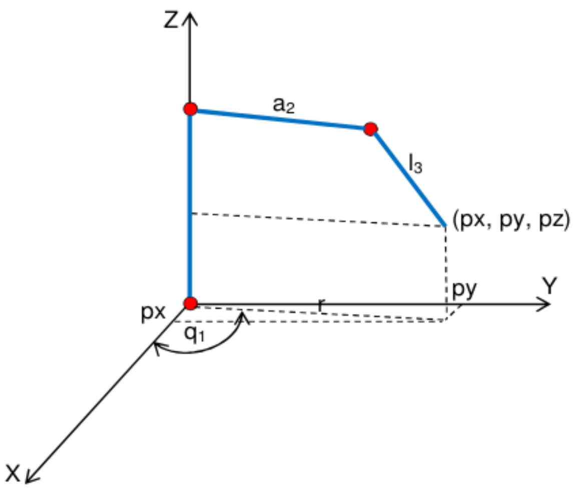

Figure 8: JRobot with three rotational degrees of freedom.

RoboPhilo, the humanoid robot has three degrees of freedom which correspond to the kinematic pairs of shoulder, elbow and wrist in the upper body and hips, knee and ankle in the lower body which is why we deduce the inverse kinematic equations as shown below

Referencing Fig. 9, it is located on the first triangle to find the value of q_1 as shown in Fig.10.

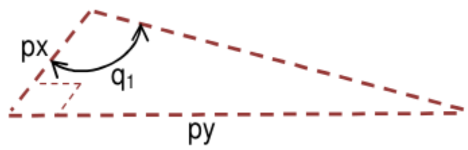

Figure 9: Right triangle used to derive the angle q_1.

Right triangle of Fig. 10 shows the following:

$$tg(q_1) = \frac{py}{px} \tag{2}$$

Therefore:

$$q_1 = tg^{-1}\left(\frac{py}{px}\right) \tag{3}$$

Whereas link of Fig. 10 are located in one plane and using the law of cosines, it will:

For the same triangle in Figure 10:

$$r^2 = px^2 + py^2 \tag{4}$$

Now using the triangle formed between the radius r and the z-axis (Fig. 11) for the distance between a link and the end effector:

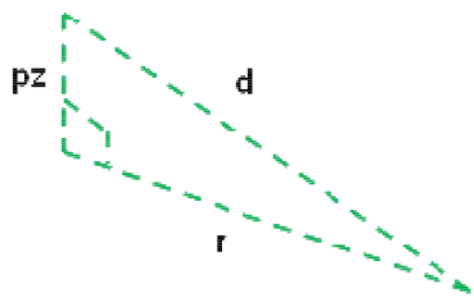

Figure 10: Triangle used for deriving the distance between a link and the end effector.

Using the right triangle in Fig. 11 we can deduce the following:

$$d^2 = r^2 + pz^2 \tag{5}$$

r^2 is replaced in Equation 4, and it is obtained:

$$d^2 = px^2 + py^2 + pz^2 \tag{6}$$

For the derivation of the second angle, the distance found is used in Equation 4, and the triangle of Fig. 12

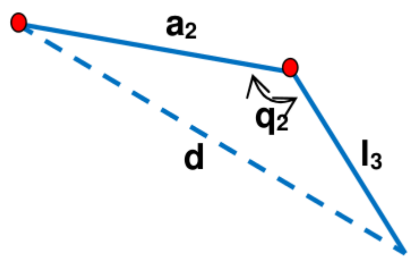

Figure 11: Triangle used to derive the angle between links.

Cosine law is used for the angle q_2, as shown in the following discussion:

$$d^2 = a_2^2 + l_3^2 - 1a_2l_3cos(q_2)$$

Therefore:

$$q_2 = cos^{-1}\left(\frac{a_2^2 + l_3^2 - d^2}{2a_2l_3}\right) \quad (7)$$

d^2 replacing Equation 6 is obained:

$$q_2 = cos^{-1}\left(\frac{a_2^2 + l_3^2 - (px^2 + py^2 + pz^2)}{2a_2l_3}\right) \quad (8)$$

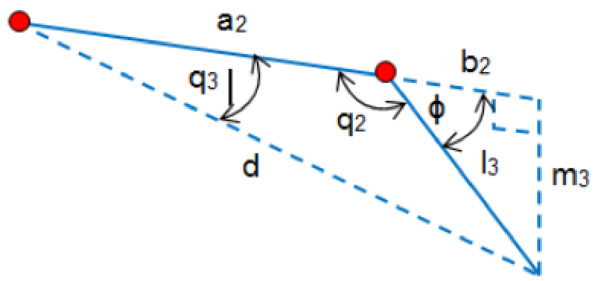

Figure 12: Triangle used to derive q_3.

The angle q_3 can be obtained from the angle found in Equation 8, using supplementary angles, as follows:

$$\phi = 180 - cos^{-1}\left(\frac{a_2^2 + l_3^2 - (px^2 + py^2 + pz^2)}{2a_2l_3}\right) \quad (9)$$

$$q_3 = tan^{-1}\left(\frac{l_3cos\left(180 - cos^{-1}\left(\frac{a_2^2l_3^2(px^2+py^2+pz^2)}{2a_2l_3}\right)\right)}{a_2 + l_3sin\left(180 - cos^{-1}\left(\frac{a_2^2+l_3^2(px^2+py^2+pz^2)}{2a_2l_3}\right)\right)}\right) \quad (10)$$

Equation 10 is able to find the angles according to the movement of joints and in some cases it could have the inverted value of the angle due to the position of the joints, the solution of this equation is always bounded because of the simple kinematic chain as was explained before.

3.3 Implementation of Inverse Kinematics equations in RoboPhilo Humanoid Platform

To implement inverse kinematics equations in RoboPhilo humanoid platform, we present a scheme (Figure 13), so then analyze movements and angles required for operation.

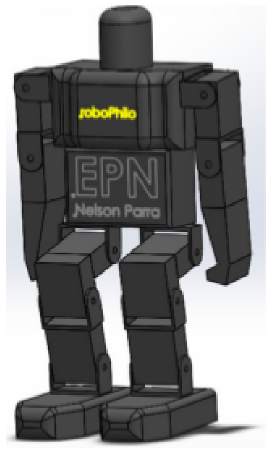

Figure 13: Platform RoboPhilo initial position.

As seen in Figure 13, the platform humanoid is at rest or initial state, and it is from this position that the analysis part. Analyzing the movements of the platform is made by taking into account each separate limb.

Table 1: Description of joints and abbreviation.

Joints Description		
Joint number	Joint Name	Abbreviation
1	Head	cbz
2	Center shoulder	hmc
3	Left shoulder	hmi
4	Right shoulder	hmd
5	Left elbow	cdi
6	Right elbow	cdd
7	left wrist	mui
8	right wrist	mud
9	Left hand	mni
10	Right hand	mnd
11	spine	sp
12	Center hip	crc
13	Left hip	cri
14	Right hip	crd
15	Left knee	rdi
16	Right knee	rdd
17	Left ankle	tbi
18	Right ankle	tbd
19	Left foot	pii
20	Right foot	pid

It analyzes the movement of the leg in the yz plane, the same algorithm is applied to all levels analyzed in each robot motion.

$$pri = tg^{-1}\left(\frac{py}{pz}\right) \quad (11)$$

The movement of the left knee angle, is taken from Equation 11.

$$rdi = cos^{-1}\left(\frac{a_2^2 + l_3^2 - (px^2 + py^2 + pz^2)}{2a_2 l_3}\right) \tag{12}$$

While:

a_2 = distance between cri and rdi.
l_3 = distance between rdi and tdi.

Points px, py and pz refer to the position of the ankle, each of the coordinate axes, that is:

$$\begin{aligned} px &= tbi_x \\ py &= tbi_y \\ pz &= tbi_z \end{aligned}$$

For the movement of the left leg in the yz plane, take the Equation 3.

$$cri = \tan^{-1}\left(\frac{px}{\sqrt{pz^2 + py^2}}\right)\tan^{-1}\left(\frac{l_3\sin(180 - rdi)}{a_2 + l_3\cos(180 - rdi)}\right) \tag{13}$$

4. CONTROL SOFTWARE DEVELOPMENT

To manage the development board C, Axon II is used as the programming language for the advantages it has over other languages. Due to the efficiency of the code it produces, is one of the most used programming languages for the development of systems and applications. The C language is deeply rooted characteristics of a mid-level language but with peculiarities of a low-level language, i.e. in the same code in which an algorithm is implemented in C can add blocks of programming in assembly language, if the programmer so required [7].

The implementation of the algorithms of inverse kinematics is in C # as a programming language, since it enables the realization of complex calculations. The sensor Kinect for Windows works at an optimal level in this language and is recommended by Microsoft due to Software Development Kit (Software Development Kit - SDK), which is incorporated in this language.

4.1 Software Implementation

Control algorithms implementation are on the microcontroller, using an Atmel ATMEGA640 AVR series (Fig. 14), which receives only the data already processed in the computer and transformed into pulses to the servo.

The location of the 12 servo motors (with their respective nomenclature) used in the development board Axon II are shown in Fig. 14.

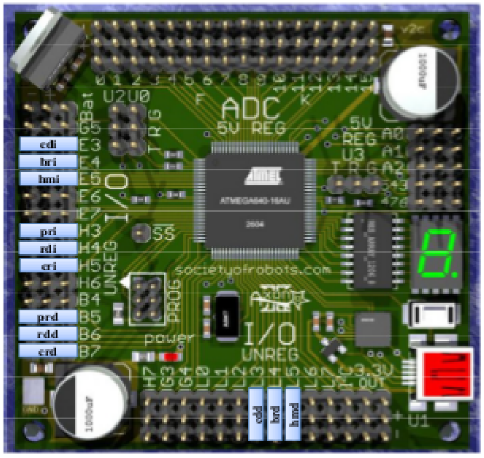

Figure 14: Development scheme Axon II plate with the position of the servomotors.

Each trio servomotors, the same as shown in Fig. 14 belong to a timer 16-bit microcontroller, as shown in Table 2.

Table 2: Servomotors location in axon II.

Servomotor's number	Servomotor's name	Microcontroller's Port	Microcontroller's pin	Microcontroller's timer	
1	cdi	E3	5	OC3A	Timer 3
2	bri	E4	6	OC3B	
3	hmi	E5	7	OC3C	
4	cdd	L3	38	OC5A	Timer 5
5	brd	L4	39	OC5B	
6	hmd	L5	40	OC5C	
7	rdi	H4	16	OC4B	Timer 4
8	pri	H3	15	OC4A	
9	cri	H5	17	OC4C	
10	rdd	B6	25	OC1B	Timer 1
11	prd	B5	24	OC1A	
12	crd	B7	26	OC1C	

The imitator robot, has two modes of operation. The first mode is "Suspended" in the platform which mimics the movements by all four limbs, while hanging from a stand provided in the kit RoboPhilo. The second mode is "Plain" in which humanoid platform has its upright lower limbs, while their upper limbs are moving.

5. WIRELESS COMMUNICATION

5.1 Serial Asynchronous Communication

One of the first devices designed to implement an asynchronous serial communication, and so far one of the most popular is the Universal Asynchronous Receiver-Transmitter, UART. This circuit consists of two separate systems: a receiver and a transmitter of data, each with its own data bus, serial port and clock output. Control logic that

determines the number of data bits, the parity type and number of stop bits, it is common to the receiver and transmitter. The UART architecture allows it to operate in "Full Duplex" with different transmission speeds [8].

5.2 XBEE Wireless Communication Module

Major modules features Xbee are the following:

- Good Range: Up to 300ft (100 meters) in line of sight for Xbee modules and up to 1 mile (1.6 km) to the Xbee Pro modules.
- 9 inputs / outputs, analog and digital.
- Low power consumption < 50mA when in operation and < 10uA when in sleep mode.
- Serial interface.
- 65,000 addresses for each of the 16 available channels.

You can have many of these devices on the same network [10].

Figure 15: Xbee Wireless Communication Module [9].

5.3 Data Frame

The data frame is sent from the computer to the microcontroller shown in Fig. 16.

1	2	3	4	5	6	7	8	9	10	11	12	13
Identificador	cdi	bri	hmi	cdd	brd	hmd	rdi	pri	cri	rdd	prd	crd
	Ángulos de servomotores											

Figure 16: Data frame sent from the Computer to microcontroller.

The identifier corresponds to 200 and because from the computer restrictions are made for the angles sent to the microcontroller no larger than 180 ^{o} or below 0 ^{o}, this identifier is distinguished and lets you know where the first data begins to separate then each of the bytes and interpret the microcontroller.

5.4 Human Machine Interface HMI

The Human Machine Interface or HMI is used to display the status of certain variables in an instant of time, as well as monitoring and control actions for a specific process. The programming language used to implement the HMI is C # in Microsoft Visual Studio. This interface is selected for the HMI because it is recommended by Microsoft for developing applications, performance and versatility.

The HMI created in Visual C # chose to place indicators that allows the user to observe if you are in the correct position to begin the process of imitation, the range of distance that the person should be positioned for correct operation is in between 2.3 to 2.8 meters, which has a complete view of the joints by the sensor kinect.

Besides, it is showing the status of each of the angles that are sent to the microcontroller, it can identify the perpendicular distance in centimeters that the person finds the sensor (Dz), which can also be identified depending on the distance from the sensor, there are two Dh and Dw distance calls, which measure the height and width, respectively, of the screen shown to the user. In this way you can know the approximate height of the person and the room you are in. Additionally places a virtual button to start the process of imitation, eliminating the need for a person close to press a button on the computer.

Finally the HMI shows an additional feature, in which a person can take up to 10 pictures while the robot is imitating.

Each button is described in the application implemented:

(1) Virtual button to start the application.
(2) Ad buttons (away, ok and get close).
(3) Control the angle of inclination of the kinect sensor camera.
(4) Box to the operation mode.
(5) Button user to take pictures with photo number indicator in which it is located.
(6) Box to the serial port.
(7) Distances from the user relative to the sensor.
(8) Current value of each of the angles of the limbs.
(9) Video screen.
(10) Screen depth.
(11) Points representing each joint of the user.
(12) Been sent from the microcontroller to the computer ("MIMIC", "INACTIVE").

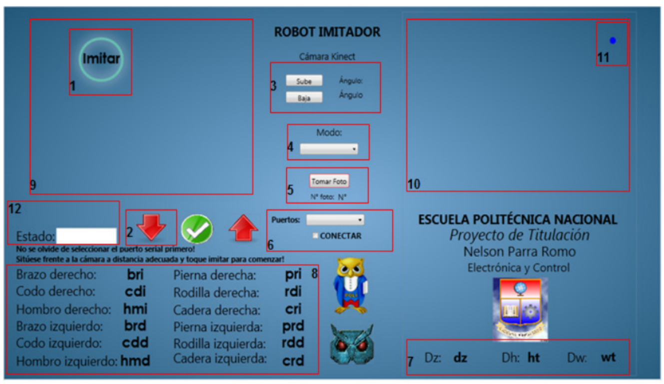

Figure 17: Description of each area of the HMI implemented.

6. TESTS AND RESULTS

6.1 Suspend Mode

Measurements of the angles use a large grader within 0.5^o.

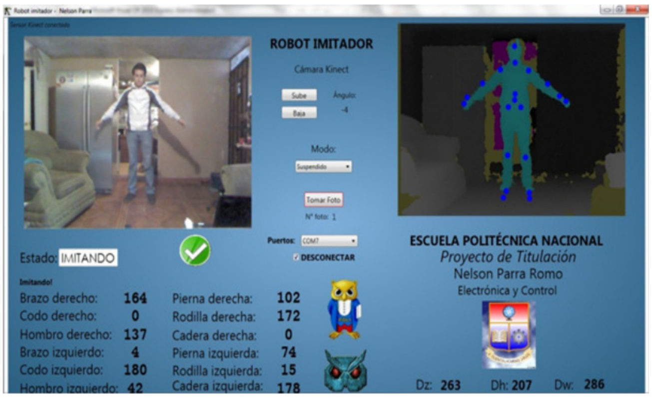

Figure 18: Test shrug xy plane, in suspended mode.

The test of Fig. 18 shows the operational part of the system thus obtains little change in upper limb movements of the robot and the angles measured and computed by software are as follows:

Table 3: Angle measures taken to upper limb.

Hombro Izquierdo	X_1	X_2	X_3	X_4	X_5	$\bar{X}$
Usuario	43.5°	43°	42.5°	43°	42.5°	42.9°
Cinemática Inversa	42°	43°	42°	41°	42°	42°
Plataforma	40.5°	41.5°	41°	40.5°	41°	40.9°
Hombro Derecho						
Usuario	43°	42.5°	43°	43°	42.5°	42.8°
Cinemática Inversa*	43°	43°	42°	43°	41°	42.5°
Plataforma	40.5°	41.5°	41°	41.5°	42°	41.3°

Table 3 is taken from a real user movement angle. It follows that for the left shoulder, the maximum variation is 2 , which represents 1.1% of the total movement of the actuator and to the right shoulder; the maximum variation is 1.5^o, which represents 0.8% of the total movement of the servomotor.

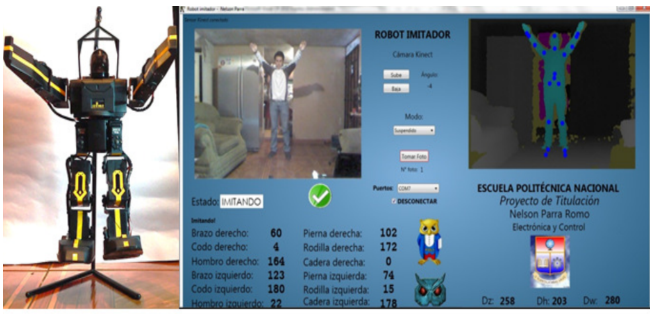

Figure 19: Arm movement test of the 3 axes, suspended mode.

In Fig. 19, it can be appreciated that the oscillations are imperceptible and the robot follows the motion of the person in a continuous manner. Measurements of the angles of the arms are shown in Table 4.

Table 4 is taken from a real user movement angle. It follows that for the left arm, the maximum variation is 2^o, which represents 1.1% of the total movement of the servomotor, and maximum right arm 2.2^o variation, which represents 1.2% of the total movement of the servomotor.

Table 4: Angle measures taken to upper limb.

Brazo Izquierdo	X_1	X_2	X_3	X_4	X_5	$\bar{X}$
Usuario	56.5°	56°	55.5°	56°	55.5°	55.9°
Cinemática Inversa*	57°	57°	56°	58°	56°	56.8°
Plataforma	58.5°	57.5°	58°	57°	58.5°	57.9°
Brazo Derecho						
Usuario	59°	59°	58.5°	58°	58.5°	58.6°
Cinemática Inversa	60°	59°	58°	59°	60°	59.2°
Plataforma	60.5°	61°	60°	61°	61.5°	60.8°

6.2 Plain Mode

The test of Fig. 20 is observed with slight oscillations, continuous perceptible movements and lower extremities are shown upright throughout the period of time for which such test measures the angles of the upper limbs, as are shown in Table 5.

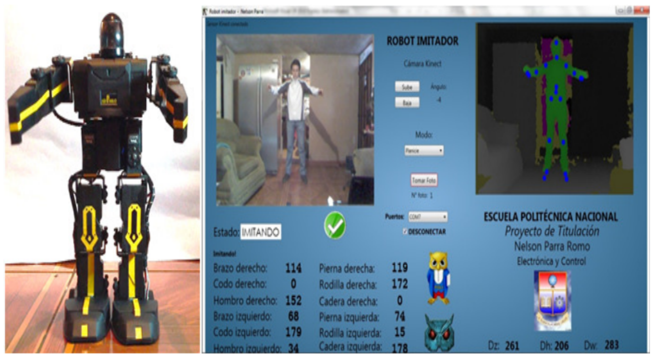

Figure 20: Test arm movement on 3 axes plain mode.

Table 5: Angle measures taken to upper limb.

Brazo Izquierdo	X_1	X_2	X_3	X_4	X_5	$\bar{X}$
Usuario	70.5°	71°	69.5°	68.5°	70°	69.9°
Cinemática Inversa	68°	68°	67°	69°	67°	67.8°
Plataforma	69.5°	69°	68.5°	70°	71.5°	69.7°
Brazo Derecho						
Usuario	69°	68°	68.5°	69°	68°	68.5°
Cinemática Inversa*	66°	67°	66°	65°	65°	65.8°
Plataforma	68.5°	69°	67°	67°	68.5°	68°

In Table 5, actual measurement is taken as the angle of movement of the user, it follows that for the left arm, the maximum variation is 2.1^o, which represents 1.2% of the total movement of the servomotor. For the right arm, the maximum variation is 2.7^o, which represents 1.5% of the total movement of the servomotor.

7. CONCLUSIONS

The maximum variation of the servo movement under normal conditions is 3.6^o which represent 2% of the total movement of an actuator, it indicates that using inverse kinematics calculations as the angles of movement of the platform is accurate to 98%.

The cost of this type of engineering projects is high, due to the research and development of applications, so titling this project can be used to rehabilitate children, youth and adults to promote movement of the limbs of fun and flashy, this project also provides the basis for the implementation of ideas to help teleoperation actions.

Kinect sensor is used for data acquisition and considering that this is a major step for the successful implementation of this project qualification, after analyzing the final data, we conclude that this makes a pretty accurate estimation of data and validates the data correctly with the interpretation of pixels it has in its internal programming, it allows reliable data, which are subsequently manipulated.

This project's main objective was to make a humanoid robot to imitate at real time the movements of a person standing in front of a Kinect sensor, this objective has been successfully reached and as shown in the results and test chapter, the accuracy of the movements are in the accepted range with the main method that has been used (Inverted Kinematics).

The next steps to take in this project would be the implementation of a closed loop control with inertial sensors and accelerometers which could make the humanoid robot autonomous when imitating, as it can control its position when the movements it is making, are detected as an alert of falling.

REFERENCES

[1] http://www.robobrothers.com/

[2] http://robots-argentina.com.ar/MotorServo_basico.htm

[3] http://www.societyofrobots.com/

[4] MATHE L., SIMBÁN D., GÓMEZ G., "Estudio del funcionamiento del sensor Kinect y aplicaciones para bioingeniería", Argentina, 2012.

[5] http://www.uhu.es/fernando.gomez/proyecto_manipulador

[6] http://www.disclab.ua.es/robolab/EJS2/RRR_Intro_3.

[7] http://es.wikipedia.org/wiki/C_(lenguaje_de_programac)

[8] http://www.buenastareas.com/ensayos/Comunicaci%C3%B3n-Serial-Asincrona/2882934.html

[9] http://www.olimex.cl/product_info.php?products_id=

[10] http://www.xbee.cl/