International Journal of Latest Research in Science and Technology **ISSN (Online):2278-5299**
Volume 4, Issue 4: Page No.50-55, July-August 2015
https://www.mnkpublication.com/journal/ijlrst/index.php

MODELLING, SIMULATION AND CONTROL OF ADAM ROBOT ARM ACTUATED BY EC 40 BRUSHLESS DC MOTORS

[1]Jebli Tarek, [2] Zaoui Chiheb, [3] Aref Maalej
[1]Laboratory of Electromechanical Systems of University of Sfax, Tunisia
[2]Laboratory of Electromechanical Systems of University of Sfax, Tunisia
[3]Laboratory of Electromechanical Systems of University of Sfax, Tunisia

Abstract*– *This paper presents simulation and control of a robotic arm actuated by three BLDC motors wich are controlled by three IPOS 48080 BX-CAN drives. The modeling problem is necessary before applying control techniques to guarantee the execution of any task according to a desired input with minimum error. The simulations of the dynamic behaviour of the robot ADAM arm are presented using SimMechanic software. The motion is programmed using the Technosoft Motion Language (TML); a high-level set of codes allowing to parameterize and execute specific motion operation. The integrated development environment for the configuration of the drives EasySetUp and C++ language are used to setup and send commands from the PC. This paper shows also the way to implement a network between the drives to synchronize the communication between then and the PC.

Keywords – Robot arm; Control; Brushless; Drives; TML

I. INTRODUCTION

Large number of control researches and numerous control applications were presented during the last years, concentrated on control of robotic arms. Building a robotic arm is not a new idea, but still the design and the spécifications can differ from other designs. For instance, the circuitry, degree of freedom (DoF), algorithm, program, attachments, equipment, accuracy and speed, completely depend on the designer's tact [1]. Concerning the 3 degree of freedom arms of Adam robot is actuated by three EC 40 bruchess motor. This motor is used on different robots like the Hubo robot manufactured by the Korea Advanced Institute of Science and Technology, we found this motor installed on the firm Xi'an Chaoren Robots. Many surgical manipulators are also equipped with this motor. Each BLDC motor is controlled by an IPOS 4808 BX- CAN drive. This drive is part of a family of fully digital intelligent servo drives, based on the latest DSP technology and it offer unprecedented drive performance combined with an embedded motion controller. It is also suitable for control brushless AC (vector control), DC brushed motors and step motors [2]. This paper is broken down into sections as follows: in section 2, we introduce the modeling of the robot ADAM. The dynamics equations of the trunk are developed. Section 3 is dedicated to the simulation of the arms motions using SimMechanics. An experiment validation is given in the section 4 to demonstrate the control of the arm using the IPOS 4808 BX-CAN drives programmed with C++ language and TML libraries.

II. DESCRIPTION AND MODELLING OF ADAM ROBOT

The robot considered in this paper is a simple original mechanism having 7 dofs: 7 rotational movements "Fig.1" and "Fig.2". The element C_1 is connected by a rotational joint (R) to the body C_0. Each arm is connected to the body C_1 and has three degrees of freedom, 2 rotary joints on the shoulder and 1 rotary joint on the elbow. The angles α, β and δ are variables and allow us to choose an initial configuration for the two arms. The sagittal plane of trunk R3P is (O_0, X_0,Y_0) and the frontal plane is (O_0,Y_0,Z_0) [3]. This robot was manufactured in the Electromechanical System Labotaory (LASEM). It weights 12 kg for a 48 cm height. It comprises a chest, two arms, two forearms and two pliers.

Fig. 1 Adam Robot

Publication History
Manuscript Received : 20 August 2015
Manuscript Accepted : 25 August 2015
Revision Received : 26 August 2015
Manuscript Published : 31 August 2015

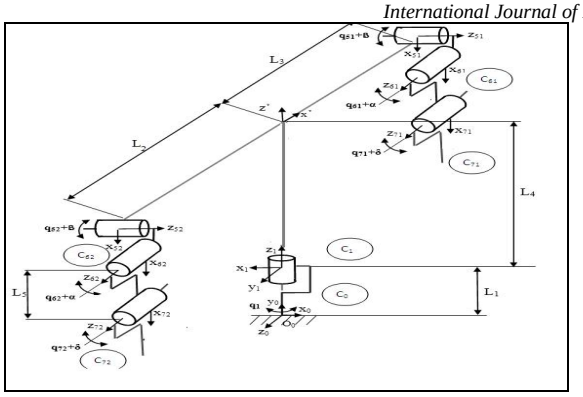

Fig. 2 Modelling of Adam robot

The properties of ADAM mechanism are presented in table 1.

TABLE 1 CARACTERESTICS OF ADAM ROBOT

C1 (kg)	10
C61 (kg)	0,42
C71 (kg)	0,41
C62 (kg)	0,42
C72 (kg)	0,41
L1 (m)	0,1
L2 (m)	0,38
L3 (m)	0,4
L61 and L62 (m)	0,201
L71 and L72 (m)	0,172

The dynamic equations of the upper part of the robot are developed using the Newton-Euler formulation. The efforts applied to the trunk located on point O_0 are given by:

$$\begin{cases} \vec{F}_0 = \sum_{i=0}^{1} m_i(\vec{\gamma}_i - \vec{g}) + \sum_{i=6}^{7}\sum_{j=1}^{2} m_{ij}(\vec{\gamma}_{Gij} - \vec{g}) \\ \vec{M}_0 = \sum_{i=0}^{4}(\vec{h}(i,O_0) - \vec{M}(m_i\vec{g},O_0)) + \sum_{i=6}^{7}\sum_{j=1}^{2}\left[(\vec{h}(ij,O_0) - \vec{M}(m_{ij}\vec{g},O_0)\right] \end{cases}$$

With:

- $\vec{h}(i,O_0)$: Derivation of element Ci angular momentum calculated at point O0.
- $\vec{F}_0 = [F_x, F_y, F_z]^T$ and $\vec{M}_0 = [M_x, M_y, M_z]^T$: Efforts exerted by the frame on C0 at point O0.
- $\vec{\gamma}_i$: Acceleration of the Ci elements.

Direct and inverse kinematics are obtained using DH parameters modified by Khalil –Kleinfinger for arborescent open linkages. The motion equations based on the Newton-Euler formulation are calculated for each element Ci (i= 61, 71, 62 and 72) of the robot at the rigid support point O0. They can be written in the following form:

$$\begin{bmatrix} \ddot{q}_{51} \\ \ddot{q}_{61} \\ \ddot{q}_{71} \\ \ddot{q}_{52} \\ \ddot{q}_{62} \\ \ddot{q}_{72} \end{bmatrix} = \begin{bmatrix} g_1(q_1,...q_6,\dot{q}_1,...\dot{q}_6, L_{ge}, L_{in}, F_x, F_y, F_z, M_x, M_y, M_z) \\ . \\ . \\ . \\ g_6(q_1,...q_6,\dot{q}_1,...\dot{q}_6, L_{ge}, L_{in}, F_x, F_y, F_z, M_x, M_y, M_z) \end{bmatrix}$$

With:

q , $\dot{q}$ and $\ddot{q}$: Position, velocity and acceleration of the joint.

Lge and Lin: Geometrical and inertial parameters of Adam robot.

Elements Ci (i= 61, 71, 62 and 72) represent the mobile masses which are used to compensate the external disturbances.

III.SIMULATION OF THE ROBOT ADAM

SimMechanic graphical interface

SimMechanics software allows us to obtain a model of multiple rigid body dynamics without the latter having to derive the equations of rigid body system. Unlike Simulink, SimMechanics does not require a model of the system block diagram representing equations in the Laplace domain. The patterns obtained with SimMechanics rather represent the physical interaction between two bodies. Thus, the rigid body blocks have inertia and mass properties that can be modified in the way of the user.

Rigid body blocks are interconnected by blocks seals of different types. For example, a robot arm can be modeled by five blocks in SimMechanics (Figure 3). The first block is the fixed inertial body. Between the ceiling and the first rod, there would be a seal block. In this latter would be a rigid body attached to block representing the first rod and a final rigid body blocks representing the second rod. Both body, arm and forearm are connected together by a pivot joint (Revolute) (Figure 3).

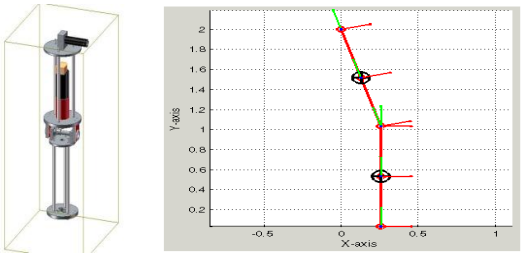

Fig. 3 Graphical interface of the robot arm

Finally, we can add the blocks Body Actuator allow to interface the SimMechanics model with traditional Simulink blocks to exert a torque on the two bodies, while blocks Body Sensor allow to take different measures on the bodies, for example their speed or position.

The SolidWorks model of Adam robot shown in figure below is translated into SimMechanics model for simulation and analysis. This process is called CAD translation. By translating a CAD assembly into a SimMechanics model, we leverage the strengths of CAD platform with the strengths of SimMechanics software.

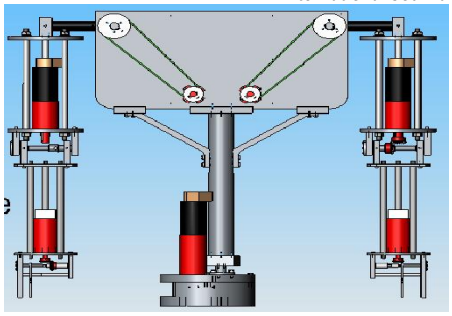

Fig. 4 CAD assembly

CAD translation is a two-step process. First, we export a CAD assembly in XML format. Then, we import the XML file into SimMechanics. SimMechanics uses the XML file to automatically generate a model that replicates the original CAD assembly [4].

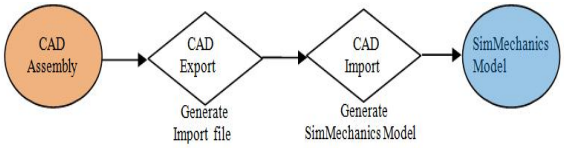

Fig. 5 Export to SimMechanics

In the model, each CAD part maps into a rigid body subsystem. Each CAD constraint or set of CAD constraints, map into a joint. Block names for SimMechanics subsystems are based on the original CAD parts and subassemblies which the subsystems represent. SimMechanics appends the suffix RIGID to the stem of a rigid body name. The following figure shows the imported SimMechanics model of the CAD assembly of Adam robot.

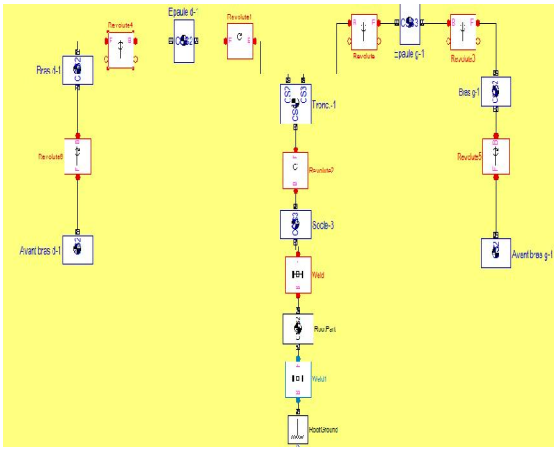

Fig. 6 SimMechanics model

To animate the model performed by SimMechanics we add blocks that create movement that can be reached angular position or a desired trajectory.

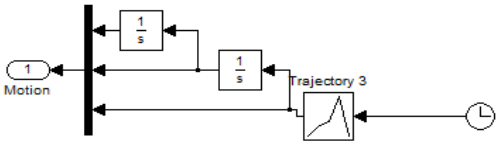

Fig. 7 block generator of trajectory

Using the SimMechanics interface and with adding the blocks the point constituting the trajectory, simulating flexion extension of the shoulder and elbow are performed also the movements of abduction / adduction of the shoulder is carried out.

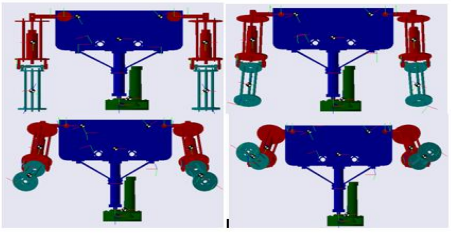

Fig. 8 Simulation of arms motion

VRML graphical intereface

First, the assembly already drowed in Solidworks is saved in VRML format (* .wrl). Then it is connected to the VRML environment with Virtual Reality Toolbox, to create animations of high quality and record videos (figure 9).

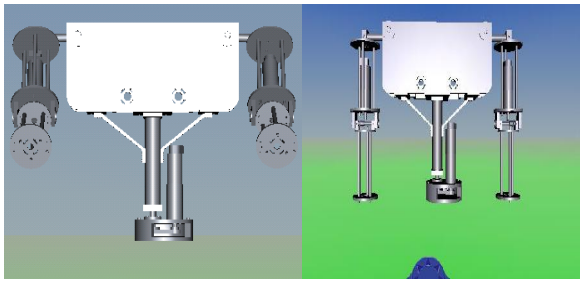

Fig. 9 VRML interface

IV. COMMAND OF ADAM ROBOT ARM

Motor Choice

A robotic arm is a robotic manipulator, usually programmable, with similar functions to a human arm. It has about same number of degree of freedom as in human arm. The robot arm use a central control box that contains the power electronics, the drives and the coordinating computer [5].Each arm of robot Adam is actuated by three EC 40 BLDC Motors. The brushless DC motor provided in this paper is the EC 40, ∅40 mm, 120 Watt from Maxon motors. The order number of the motor is 118896. The parameters are extracted from the datasheet of this motor with corresponding relevant parameters used. Find bellow in table 2 the major extracted parameters.

TABLE 2 CAEC 40 PARAMETERS

Nominal voltage	42 V
No load speed	10400 rpm
No load current	258 mA
Nominal speed	9290 rpm
Starting current	33.5 A
Max. efficiency	84 %
Terminal inductance phase to phase	1.25 Ω
Terminal inductance phase to phase	0.319 mH

As the name implies, BLDC motors do not use brushes for commutation; instead, they are electronically commutated. BLDC motors have many advantages over brushed DC motors and induction motors. A few of these are:

- Better speed versus torque characteristics
- High dynamic response
- High efficiency
- Long operating life
- Noiseless operation
- Higher speed ranges

In addition, the ratio of torque delivered to the size of the motor is higher, making it useful in applications where space and weight are critical factors [6].

Encoder

Fig. 9 shows a HEDS-5500 encoder generally found attached to many DC motors. This device converts the angular position of a motor shaft into a digital code. This encoder is high performance, low cost, two and three channel optical incremental encoders. it emphasizes high reliability, high resolution, and easy assembly [6].There are two types of encoders: incremental and absolute. On incremental encoders position is given by pulses from a start pulse zero with no relation to the shaft angle. On the other hand absolute encoders supply a unique code that is relative to each position of its course [7].

Fig. 10 Encoder

Ipos4808 BX drive

The iPOS4808 BX-CAN is part of a family of fully digital intelligent servo drives, based on the latest DSP technology and they offer unprecedented drive performance combined with an embedded motion controller. It is suitable for control of brushless DC, brushless AC (vector control), DC brushed motors and step motors, the iPOS4808 BX-CAN drives accept as position feedback incremental encoders (quadrature or sine/cosine) and linear Halls signals.

-The iPOS4808 BX-CAN always requires two supply voltages: Vlog and Vmot.
-The logic power supply Vlog =24V 1 A is connected to J4 pin 8.
-The motor power supply Vmot =42V 8A is connected to J1 pin 2.
-The cable RS232 is connected to J7 (Tx pin 1, Rx pin3 and Gnd pin 2 et4).
-The motor connection and the hall sensor are connected to J4 and concerning the encoder is connected to J3.

Fig. 11 Ipos 4808 linked to motor

Establish communication

Before starting to send motion commands from the PC, we need to do the drive/motor setup. For this operation we have to use Easy Setup, the integrated development environment for the configuration of the Technosoft drives and motors. EasySetUp works with setup data. A setup contains all the information needed to configure and parameterize the IPOS4808 drive. This information is preserved in the drive EEPROM in the setup table. The setup table is copied at power-on into the RAM memory of the drive and is used during runtime Easy Setup tries to communicate via RS-232 and COM1 with a drive having axis ID=1.

Because that our PC doesn't contain a DB9 connector, we used an USB RS-232 adapter to establish the communication.

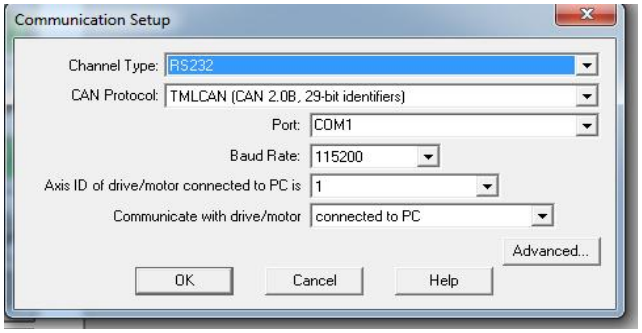

Fig. 12 Communication window

Motor setup

Data introduction is accompanied by a series of tests having as goal to check the connections to the drive and/or to determine or validate a part of the motor and sensors parameters. In the motor setup window we set the following parameters:

-Check of the values for nominal and peak currents.

-Check of the number of motor pole pairs.

-Check of the number of motor pole pairs.

-Check of the number of motor pole pairs.

-Check of the value of the motor inertia.

-Use the <<Test Connections>> button in order to verify the operation of the A and B encoder signals.

All the values of these parameters are described in the figure 13.

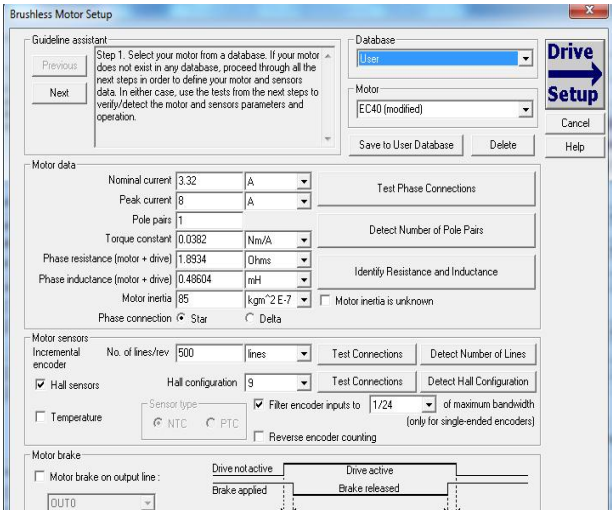

Fig. 13 Motor setup window

Drive setup

In the Drive setup dialogue we can configure and parameterize the drive for the application. In the drive setup window, we set the folloing parameter:

-Select of the type of mode (trapezoidal or sunisoidal)

-Set of the speed controller parameters Kp (proportional), Ki (integral), Integral limit (saturation limit for the integral term of the controller) and the acceleration Feed forward.
-Select and parameterize the protections that we want to activate.

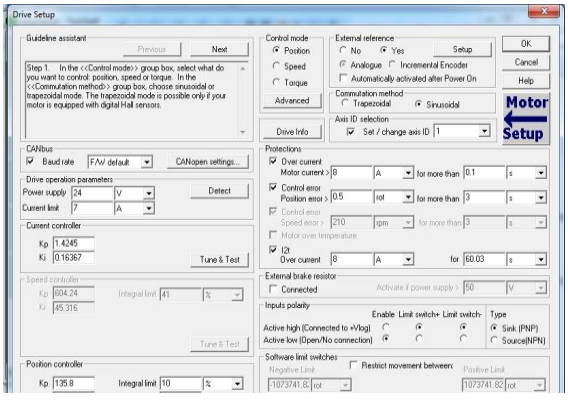

Fig. 14 Drive setup window

Tests and results

After making setup the motor and driver, some tests are mandatory to check if hall sensors are mounted on the motor also to verify the operation of the 3 Hall sensor inputs. After setting the position controller parameters Kp (proportional), Ki (integral), Kd (derivative), Kd filter (filtering coefficient for the derivative term), we compute the position controller parameters and we test the position loop behavior. The results of the tests are satisfactory and indicate that communication is well established and the motor runs normally.

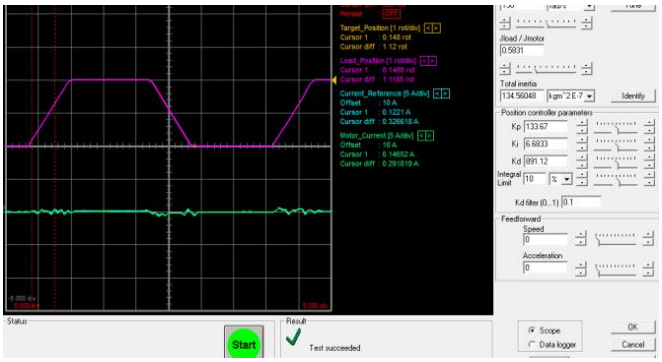

Fig. 15 Motor test

V. INTERNAL NETWORK

The CAN bus is a highly reliable standard developed by Robert Bosch GmbH for use in the automotive environment. It is a multi-master system, with sophisticated error checking and arbitration, so that any high priority message will always get through first without corruption by other messages. All data contained in each packet (up to eight bytes) is also checked with a Cyclic Redundancy Check (CRC) error-checking scheme that can correct up to five random errors, and will be automatically retransmitted if not correct. The network operates at up to 1 Mbit/sec [8].To control one arm, we built a CAN network using two cables with twisted wires (2 wires/pair), with CAN-Hi twisted together with CAN-Lo.The cable impedance was be 100 ohms and a capacitance below 30pF/meter. The CAN network is terminated at either end with 120W resistors, to eliminate signal reflections in the wiring. The first cable is mounted between the J5 connector of the first drive and the J6 connector of the second drive while the second cable connect the J5 connector of the second drive with the J6 connector of the third drive. After we have correctly connected the two drives in CAN, the next step is to make sure that the drives have different axis IDs and the same CAN baudrate.

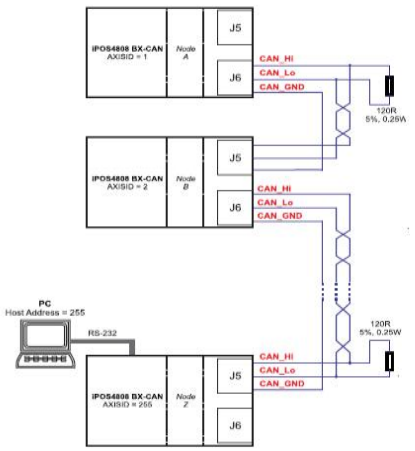

Fig. 15 Multiple axes vetwork

VI. MOTION PROGRAMMING

The IPOS 4808 drive is programmable using the Technosoft Motion Language (TML). TML consists of a high-level set of codes allowing the user to parameterize and execute specific motion operations. A central element of the library is the communication kernel, which is responsible of correct opening of the communication channel (serial RS-232).

```
BOOL InitCommunicationChannel(void)
/* Open the comunication channel:
COM1, RS232, 1, 115200 */
if(TS_OpenChannel (CHANNEL_NAME,
CHANNEL_TYPE, HOST_ID, BAUDRATE) < 0)
Return FALSE
```

After setting the communication, the setup information is required by the library functions in order to check if there are incompatibilities between the drive and the operation to be executed. The setup data is generated by EasySetup based on the actual configuration. The setup information is in the form of archive files with the .t.zip extension. The setup data of the drive/motor are declared using the TS_LoadSetup function in the PC application. The TS_LoadSetup has as argument the *.t.zip file. The function must be called for each axis controlled through TML_lib [9].

```
WORD sAxiOn_flag = 0;
idxSetup =
TS_LoadSetup("Test2ex4.t.zip");
if(idxSetup < 0)
return FALSE;
```

Once, the communication was established and the axis was initialized, we start sending the motion to the motor using some programming functions. The motion is described through AbsPosition parameter for position to reach, Speed for slew speed and Acceleration for acceleration/deceleration rate. The position to reach can be positive or negative. The Speed and Acceleration can be only positive. Once set, the motion parameters are memorized on the drive/motor[9].

VII. CONCLUSIONS

Simulation and controlling of robot arm actuated by three brushless DC motors was the main objective of this paper. A simulator of Adam robot was also developped using VRML to show the motion of the arms. The communication between the motors and the PC was perfermed using RS232 cable. Due to the intelligent drives IPOS 4808 and technosoft motion libraries (TML) software, the programming of arms becames easily and instead of trying to command each step of an axis movement, we can program the drives/motors using TML to execute complex tasks and inform the master when these are done. Future work will consist in using the movements and the accelerations of the parts of the arms to show their importance for stabilization in the case of external disturbances.

REFERENCES

[1] A.Faravar “Design implementation and Control of Robotic Arm Using PIC 16F877A Microcontrollerl,” Eastern Mediterranean University 2014

[2] IPOS 4808 BX CAN Intelligent Servo Drive for Step, DC, Brushless DC and AC Motors, Technical reference, Technosoft 2013.

[3] Ch. Zaoui, O.Bruneau, Fb. Ouezdou, A. Maalej, "Simulation of the dynamic behavior of a bipedal robot with trunk and arms subjected to 3D external disturbances in a vertical posture, during walking and during object handling” Journal of Multibody-Dynamics, 2009

[4] MATH WORK, 2001” Introduction to MATLAB, the Mathworks.

[5] G. Wyeth, J. Lilliwhite, “Distributed digital control of a robot arm”, Gordon Wyeth Retrieved on: 18 July 2015.

[6] P. Yedamale, “Brushless DC (BLDC) Motor Fundamentals”, Microchip Technology Inc 2003.

[7] G. Lopes, F. Ribeiro, B. Matos, “ Controller and actuator of three independent DC motors in closed loop,” Universidade do Minho, Departamento de Electronic Industrial, Guimarães, Portugal.

[8] G. Wyeth, D. Kee, M. Wagstaff, N. Brewer, J. Stirzaker, T. Cartwright, B. Bebel**.** “Design of an Autonomous Humanoid Robot,” School of Computer Science and Electrical Engineering University of Queensland.

[9] TML_LIB Motion Control Libraries for Technosoft Intelligent Drives, user Manuel, Technosoft 2009.