

DESIGN AND CONTROL OF A DUAL-ARM ROBOT

¹Jebli Tarek, ²Zaoui Chiheb, ¹Aref Maalej

¹Laboratory of Electromechanical Systems of University of Sfax, Tunisia

²U.R Automatique et Robotique Marine (ARM), Académie Navale, 7050 Menzel Bourguiba, Tunisia

Abstract- This paper describes the design and control of a robot equipped with two arms and having 7 dof. The robot was primarily developed to study the dynamic contribution of arms during compensating of external disturbances. This study allows us to determine on-line, the required movements and accelerations of arms in order to maintain the robot stability and shows the importance of the arms for the robot stability. This mechatronic project contains the implementation of control algorithms, mechanical design, and electronic control boards under operating multiple processors together. The mechanical design is a complete CAD solids model, with specific motors and transmission systems selected. The simulations of the dynamic behavior of the robot ADAM are presented using SimMechanics software so a high fidelity graphical simulator has been developed, providing important early feedback on critical design decisions. The control of robot is achieved by IPOS 4808 intelligent servo drives based on the latest DSP technology. The motion is programmed using the Technosoft Motion Language (TML); a high-level set of codes allowing parameterizing and executing specific motion operation. This paper shows also the way to implement a network between the drives.

Keywords – Robot ; Simulation; Control; IPOS; TML

I. INTRODUCTION

In the recent years we have witnessed an explosive development of robotics, In addition to single-arm serial robots, several multi-arm systems were designed to perform advanced complex cooperative tasks in industrial and domestic environments. Furthermore, to imitate the human body structure, dual-arm manipulators and humanoid robots were designed and adapted to perform the desired robotic applications. Recent industrial development of dual-arm robots (e.g. Motoman SDA-series, DLR-KUKA Justin etc.) has considerably leveraged interest of researchers and users for this new class of robotic systems. The main advantages and benefits of dual arm-robots over single arm systems are obvious: *multitasking* (the arms can operate independently or synchronously performing complex bimanual assembly tasks), *cost saving* (a dual-arm robot can replace at least two convenient robots, as well as expensive fixture and regrasping devices) and *space saving* (a dual-arm robot requires smaller operating space). There are, however, several practical difficulties in understanding, controlling, planning and programming of dual-arm operations [1]. The research presented in this paper discusses the development of a dual arms robot from the planning stage to the motion control. This robot provided a reliable platform to experiment with dynamic balance control algorithms. This paper is broken down into sections as follows: in section 2, the specification of the design and implementation of ADAM robot and its constituent bodies and instrumentation of the prototype will be detailed. Section 3 is dedicated to the simulation of the robot using SimMechanics and VRML. The rest of this paper is reserved to the control of the robot using the IPOS 4808 BX-CAN drives programmed with C++ language and TML libraries.

II. FUNCTIONAL DESCRIPTION OF THE MECHANISM

The robot that we designed and built is shown in Figure 1. The final design is 48 cm tall, weighs 12 kg, and has 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) [2]. This robot was manufactured in the Electromechanical System Laboratory (LASEM). It comprises a chest, two arms, two forearms and two pliers.



Fig. 1 Adam Robot

Publication History

Manuscript Received : 19 December 2015
 Manuscript Accepted : 23 December 2015
 Revision Received : 26 December 2015
 Manuscript Published : 31 December 2015

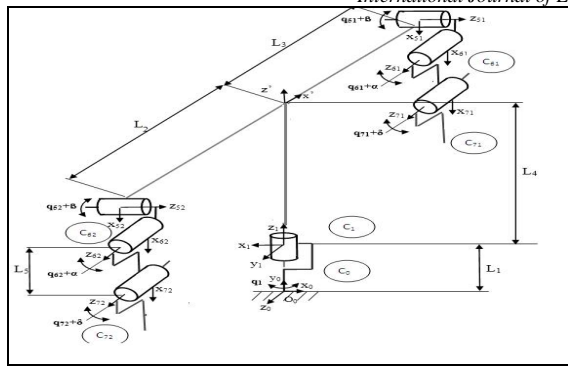


Fig. 2 Modelling of Adam robot

The properties of ADAM mechanism are presented in table 1.

TABLE 1 CHARACTERISTICS 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

III. PROCEDURES USED TO DESIGN THE MECHANISM

To design a system, the following procedure must be organized. This step is a schematic formulation of the functions to be studied as part of this action. It has a considerable interest in the orientation of the designer field of reflection.

Environment and service functions:

The first step is to define the interactions of the system with its environment. Figure 3 shows schematically the different elements put into play in the different phases of the creation of the system. The main function is to compensate for external disturbances in particular ensured the stability of the robot under disturbance.

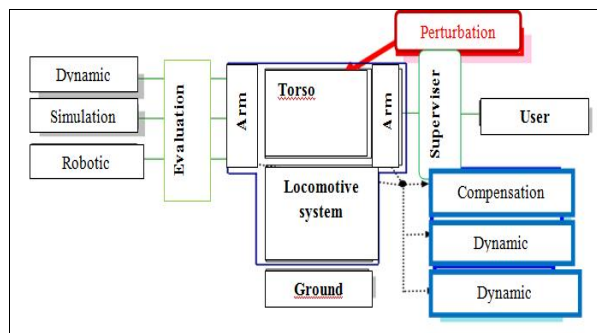


Fig. 3 Environment and service functions

Three main sources are necessary for ADAM prototype design;

- Dynamic studies on the upper part of the robot (torso and both arms) will occur in primitive design phases to give qualitative assessment tools (sizing, joints, paces, etc.).
- The simulation tool will occur downstream of the analysis of joint movement for quantitative assessment of these functions (schedules laws, joint torques, powers, etc.). The design and technology choices will be in harmony with the robotic technical aspects (actuation, control, perception). A phase of experimentation and interaction with the user are illustrated by the oversight function.

Technical Feature

In this step, the system will be studied in terms of detailed technical functions. Like any robotic system, the experimental platform will consist of three main parts: mechanics, electronics and computer (Figure 4).

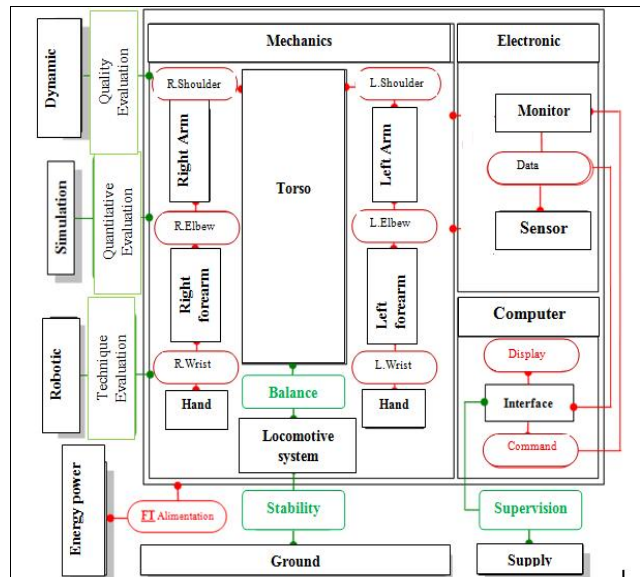


Fig. 4 Technical functions

The mechanical part, which is the functional aspect of the upper part, is composed of a torso provided with two arms (right and left), two forearms and two hands. The joints between these bodies represent technical functions to study (Shoulder, Elbow and Wrist).

Quantitative and qualitative assessments of these technical functions will be carried out respectively with the analytical calculation of dynamic equations and using the numerical simulation tool. Moreover, the electronic part consists of a control unit (Controller) and collecting means (sensors). Two technical functions (Control and Perception) bind these bodies to the mechanical part. Finally the computer part is composed mainly of a man-machine interface that will intervene in sending orders (Controller), hence the technical function (command). In addition, the technical function (Display) will help the user monitor the experimental platform. Data exchange between controller, sensor and interface are realized by the technical function. It should be noted that the energy supply needed to supply different parts

of the system is placed between the experimental platform and its environment. The technological aspect will be also taken into account in all investigated parts.

Final Prototype:

After presenting the various stages of design of the ADAM robot (Figure 5), the technical description of the individual modules will highlight the results of this design.

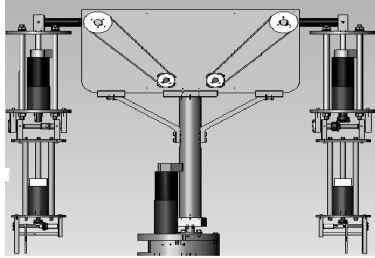


Fig. 5 Prototype of Adam robot

Shoulder:

As noted, the shoulder module has two degrees of freedom and can perform two movements:

- 1st movement: flexion – extension
This movement occurs in the sagittal plane (Figure 5), the actuating system is provided by:
 - A Maxon DC motor (power 120 W),
 - A planetary gear (reduction ratio = 230).

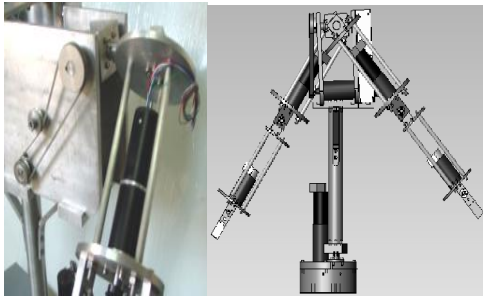


Fig. 6 Flexion extension movement

2nd movement: abduction

This movement is in the frontal plane (Figure 6), the movement is ensured by:

- A direct current motor (power 120 W)
- A planetary gear (reduction ratio = 230)
- A belt pulley transmission system.

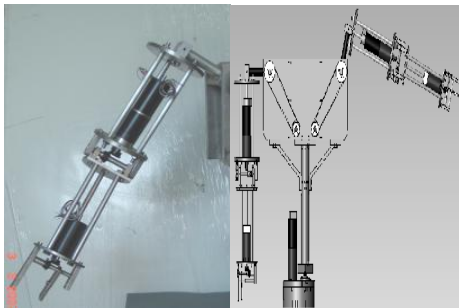


Fig. 7 abduction movement

Elbow

The elbow (Figure 7) provides flexural movement; has only one degree of freedom

Linear actuators consist of the following three elements:

- A Maxon DC motor (power 120 W),
- A parallel gear train (reduction ratio = 74)
- A transmission bevel gear (transmission ratio = 1).

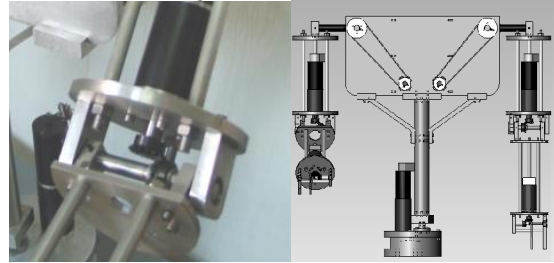


Fig. 8 Elbow

Pliers

having a single degree of freedom, the clip module (Figure 8) allows the robot to grasp objects whose mass does not exceed 1 kg.

- Actuating of the chain is ensured by:
 - A direct current motor (power 120 W),
 - A transmission bevel gear (reduction ratio = 1).

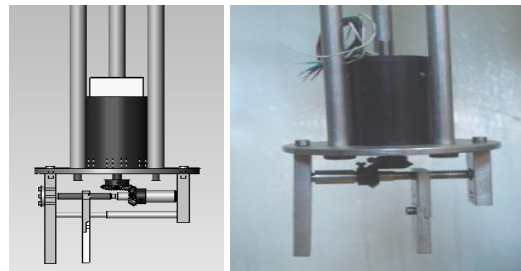


Fig. 9 Pliers

Rotation block:

Having a single degree of freedom active, the rotation block (Figure 9) allows the robot to perform rotational movement around the vertical axis.

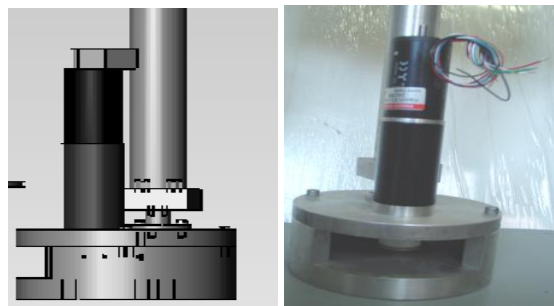


Fig. 10 Rotation block

IV.SIMULATION OF ADAM ROBOT

Modeling and simulation of the robot can greatly assist in making progress on all levels of design, implementation and operation. In this article some parts of simulation and modeling the robot dynamics as well as measures for stability and performance of arms has done by SimMechanics.

Simulink is a block diagram environment for multi-domain simulation and model-based design. It supports system-level design, simulation, automatic code generation, and continuous test and verification of embedded systems. Simulink provides a graphical editor, customizable block libraries, and solvers for modeling and simulating dynamic systems. It is integrated with MATLAB, enabling us to incorporate MATLAB algorithms into models and export simulation results to MATLAB for further analysis. Therefore, Simulink is a good choice to build and simulate Adam motion control system as the modeling tool. In addition, it is easy and convenient to generate the code automatically by Simulink which is a potential method for the deeper development of the robot [3].

Rigid body blocks are interconnected by blocks seals of different types. For example, a robot arm can be modelled by five blocks in SimMechanics. 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).

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.

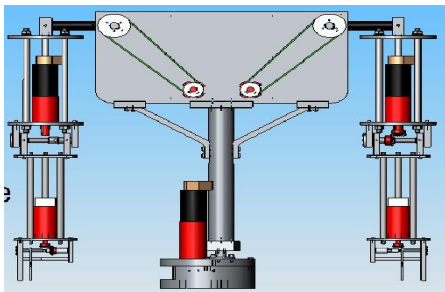


Fig. 11 CAD assembly

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

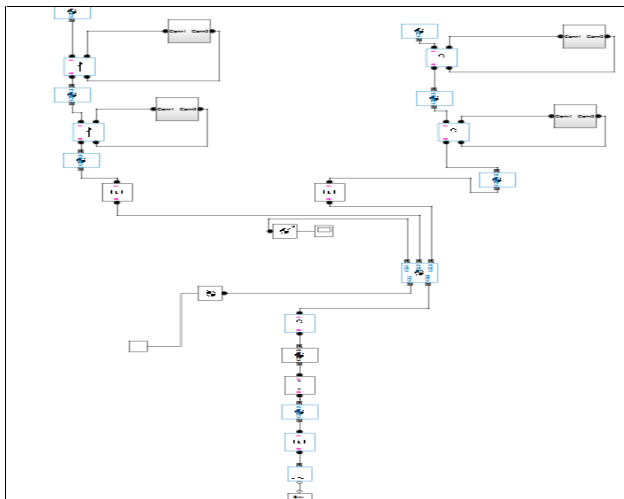


Fig. 12 SimMechanics model

Based on the values of the external forces detected by the sensor, we apply for each joint of the model the corresponding torque in order to maintain stability e.g. Keep $F_x = 0, F_y = \text{trunk weight}, F_z = 0, M_x = M_y = M_z = 0 \text{Nm}$.

The model of each external force is represented in Fig.3.

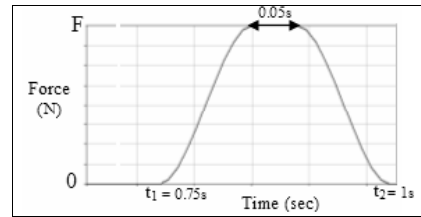


Fig. 13 External forces model

The external force is applied between $t_1=0.75\text{s}$ and $t_2=1\text{s}$. It increases from 0 until the selected maximum amplitude F , then the force is maintained for 0.05s and finally it decreases from F to 0.

During the simulation of the dynamic behavior of the robot, a PD controller is used to compute joint torque based on the current state and the desired motion. The PD controller determines the output torque according to the difference between the desired state q_d, \dot{q}_d and the current state q, \dot{q} (respectively positions and velocities vector) Where the design K_{pi} and K_{vi} are respectively called position and velocity gains, are symmetric positive definite and suitably selected.

$$\tau_i = K_{pi} (q_{di} - q_i) + K_{vi} (\dot{q}_{di} - \dot{q}_i)$$

The diagram below present the blocks used to simulate output torque (Fig. 14).

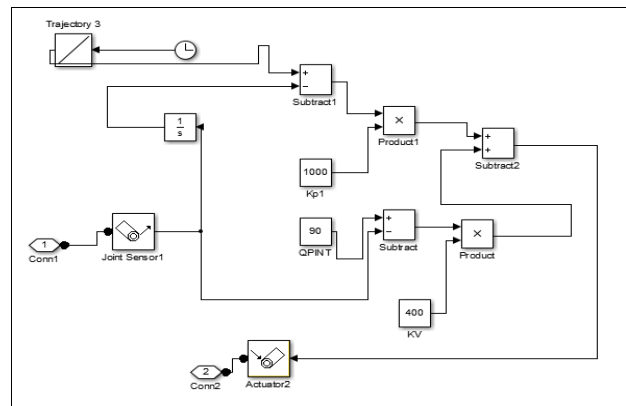


Fig. 14 PD model

Some simulations are presented, showing the flexion / extension and the abduction / adduction of the shoulder.

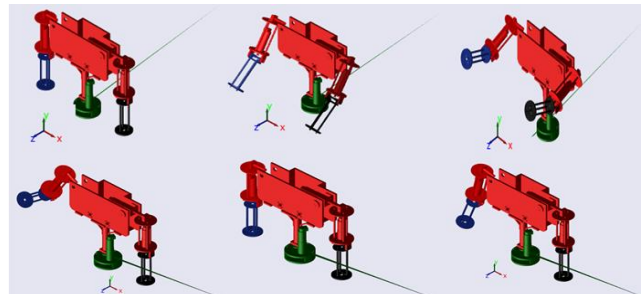


Fig. 15 Arms motion simulation

V. POWER ELECTRONICS

Motor Torque Analysis

The motor selection was very important in the design of the robot. In general, the motors should be very strong, lightweight, and easy to use. We used the EC 40, Ø40 mm, 120 Watt from Maxon motors. The order number of the motor is 118896. It is widely recognized at present that use of brushless DC (BLDC) motors as actuators in robotics presents a number of advantages with respect to use of brushed DC motors. A few of these are:

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

However, it is also known that control of BLDC motors is more complicated because of the nonlinear and multivariable nature of their model [4]. The parameters are extracted from the datasheet of this motor with corresponding relevant parameters used. Find below in table 2 the major extracted 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

Encoder

Fig. 16 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. 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 [5].



Fig. 16 Encoder

IPOS servo drive

The iPOS4808 BX-CAN is a new member of the iPOS family of Technosoft intelligent drives. It is based on a new design concept for closed-frame drives, offering a very

compact and cost effective solution for the control of rotary or linear brushless, DC brush, and step motors of powers up to 400 W. Complex motion sequences can be programmed and executed automatically at power-up from the non-volatile memory of the drive. Advanced positioning profiles like the PVT or electronic coming, I/O and program flow control, data transfer between axes, subroutines ISRs and multiple homing modes ease the motion application implementation task. [6]



Fig. 17 IPOS 4808 BX- CAN

-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. 18 IPOS 4808 linked to motor

Before starting to send motion commands from the PC, we need to perform 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.

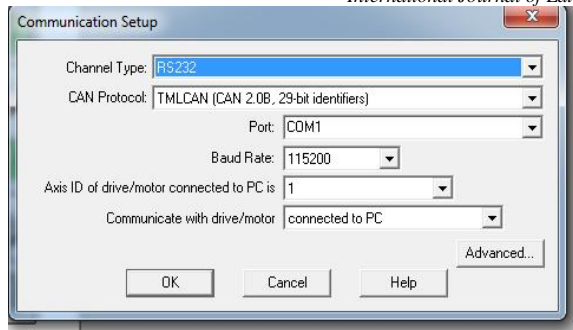


Fig. 19 Communication window

Tests and results

To make the setup of the motor and drive figure 20, 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

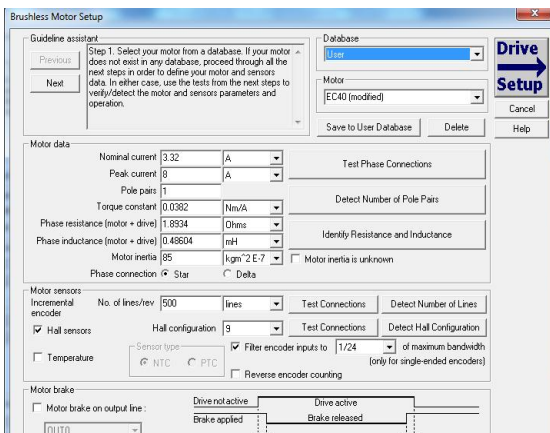


Fig. 20 Motor setup window

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

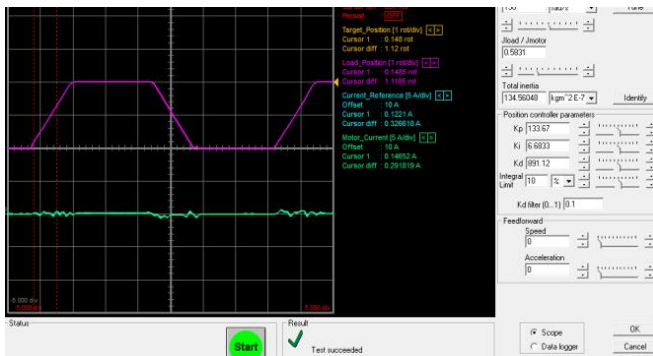


Fig. 21 Motor test

VI. CAN BUS

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 [7]. 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 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 connects 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.

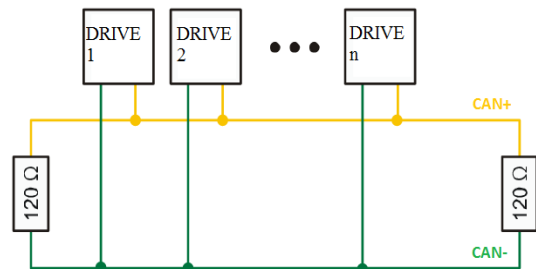


Fig. 22 Multiple axes network

VII. 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 for correct opening of the communication channel (serial RS-232).

```

BOOL InitCommunicationChannel(void)
/* Open the communication 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 .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 [8].

```
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.

VIII. CONCLUSIONS

This article deals with the design and realization of a robot's mechatronic system. The issues covered include the mechatronic design concept, electrical drive systems, design and analysis of the mechanical structure, and contributions to the computer and electronics architecture. The joint drives form a basis for high dynamic performance and acceleration behavior of the robot. The powerful yet lightweight servo actuators combine a high-dynamic brushless servo motor with precision gearing and sensors into a compact package. A simulator of Adam robot was also developed to show the motion of the arms. We used the TML and the IPOS4808 drive to set various motion applications, change the motion modes, execute homing sequences and also perform data transfers between axes. 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 disturbance.

REFERENCES

- [1] D. Surdilovic, Y. Yakut, T-M. Nguyen, X. B. Pham, A. Vick and R. Martin Martin "Compliance Control with Dual-Arm Humanoid Robots: Design, Planning and Programming", 2010 IEEE-RAS International Conference on Humanoid Robots Nashville, TN, USA, December 6-8, 2010.
- [2] 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
- [3] MATH WORK, 2001" Introduction to MATLAB, the Mathworks.
- [4] P. Yedamale, "Brushless DC (BLDC) Motor Fundamentals", Microchip Technology Inc 2003
- [5] 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 POS 4808 BX CAN Intelligent Servo Drive for Step, DC, Brushless DC and AC Motors, Technical reference, Technosoft 2013.
- [6] 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
- [7] TML_LIB Motion Control Libraries for Technosoft Intelligent Drives, user Manuel, Technosoft 2009.