Industrial Robot for Controlling and Monitoring of a Serving Activity

Paul Ciprian Patic and Florin Ion Popa

***Abstract*—**The industrial robot is currently the point of intersection of peak results in a number of areas: mechanical, automation, computers and drive systems. This congruent of such different scientific and technological branches is explained by the robot's complexity, both in terms of mechanical architecture and management. Having in view what we said above, one can say that an industrial robot is a result of these technical and scientific developments and can be defined as a technological system capable of assisting or replacing human in the exercise of various actions on machinery or production lines. In this paper one present in detail, especially, the design of a robotic arm that has the task of transporting different objects in an ordered order. The robotic arm will be programmed in LabVIEW to "collect" the pieces in the indicated locations and, after that, to transport different parts to the storage sites or in other industrial places.

***Keywords*—**Graphic Programming, Industrial Robot, Robotic Arm, Stepper Motors, Virtual Instrument.

I. INTRODUCTION

THE robot is the natural result of the evolution of automated machine tools, program-driven machines, automated manufacturing lines etc. when their rigidity and inflexibility no longer met the current requirements of productivity and quality, and the human tried to execute direct, immediate actions on processes that gained a supervisory and control role.

The industrial robots currently used present non-uniform constructive and conceptual solutions, due in particular to the diversity of the required tasks, the technical parameters imposed and the specific applications for which they were designed [14].

With all this apparent unity, the robot through its mechanical structure can be considered as a homogeneous system consisting of elements with well-defined functions that ensure the direct interaction between the robot and the object of its action in the operating space.

The main components of the mechanical structure are: the effector element, the arm and the base of the robot [6].

The effector element sometimes referred to as a griper, prehensile element, hand or simply a terminal element ensures direct contact directly between the robot and the object in the operating space on which it operates. This element differs constructively from the range of applications and the nature of the function.

Thus, the effector elements used in welding differ from those used in handling or painting operations.

Such an element comprises:

- the body itself, with a mechanical structure appropriate to the performed function;
- One or more actuators;
- One or more sensors for determining critical operations of the operation.

It should be noted that the adopted constructive solutions tend to achieve either a multifunctional element with a wide range of applications or a mono-functional effector element with a precise destination.

The robot arm serves for correct positioning of the effector element. For this purpose, the arm is a mechanical structure with a variable geometry obtained by cascading some segments connected by rotation or translation joints. The corresponding drive systems ensure the independent movements of each segment relative to the previous segment. These movements are generally restricted by certain features of mechanical architecture [15].

All these elements and subassemblies are mounted on a special frame that forms the robot base. This base is placed either on a fixed or mobile base (depending on the type of robot), or suspended on a track guideway.

The listed items form the basic structure of any industrial robot. In addition to this "classic" structure, locomotion systems, systems with 2-3 arms, systems with 2-3 effector elements etc. can be found in robot construction [12].

Robots, through their structure and functions, represent a class of systems that synthesize cutting-edge elements from a range of technical and scientific fields. In fact, through its attributions, the robot imitates or replaces the locomotor, manipulation, and intellect functions of man. It is obvious, therefore, that the robot is an extremely complex system, described by sophisticated mathematical models defined by nonlinear differential equation systems with variable, deterministic or stochastic parameters, comprising a large number of input and output variables [3], [4].

The basic function of the robot is its movement in space, so the static and dynamic regimes of the mechanical structure will be the starting point in defining the robot as a driving object.

For example, consider a robot with three rotation joints

Paul Ciprian Patic is with the Valahia University of Targoviste, Romania (corresponding author to provide e-mail: patic@valahia.ro).

Ion Florin Popa is with the Valahia University of Targoviste, Romania (e-mail: p.florin@yahoo.com)

(Figure 1.3). The movement, the robot's evolution, is determined by the three moments M1, M2, M3 applied to the joints, which cause the corresponding segments to rotate and thus obtain a new position of the arm, defined by the new angular values q1, q2, q3 [12].

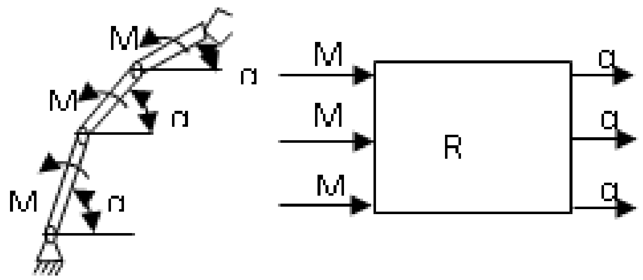


Fig. 1. Kinematic structure of a RRR Robot Type

Thus, as a guiding object, the robot receives an input vector defined by the generalized forces applied to the joints and generates an output vector formed by the angles (or displacements) of the joints.

The mechanical system of a robot consists of a rigid body configuration, the system elements interconnected successively through rotation or translation joints. The relative positions of these elements determine the overall position of the mechanical arm, this position being one of the robot's functional conditions.

The most known versions of mechanical joints encountered in robotic systems are represented by open kinematic chains in which the positioning of the speed and acceleration of an element can be obtained recursively from the parameters of the preceding element. In general, each element contains a single degree of freedom relative to the preceding element so that the transformation relationships between elements contain a single variable parameter. The cascading of all the transformations associated with each element allows the determination of the motion parameters of the entire mechanical configuration and, in general, of the terminal element [16].

Analysis as a driven object also requires the definition of the robot state vector. Generally, this vector is determined by the generalized coordinates established in the joints (angles or displacements) and their derivatives (generalized velocities of motion). Robot-specific input-state-output relationships are given by non-linear differential equations obtained from its dynamic regimes. The deduction of these equations and the quantitative and qualitative analysis of the movement is the subject of the many research in robotic domain [1].

Robot-specific manipulation operations require, firstly, a proper positioning of the mechanical system, thus reaching a point in the workspace, and secondly, require a particular orientation of the terminal element. For example, a thread-fitting operation requires both the tap hole and correct alignment of the screw to perform the assembly. It is therefore necessary to adopt a coordinate system corresponding to the description of these requirements.

II. DESIGNING OF THE ROBOTIC ARM

As one knows, the robot's governing structure is a hierarchical structure. This driving principle is due to the special complexity of the systems that are part of the robot and the difficulties created by the imposed operating tasks.

The hierarchical organization of the robot control systems is vertical, each level hierarchically covering the inferior level in relation to the driving problems addressed. A level of control communicates with the immediate inferior level through control instructions and receives from it characteristic information which, together with the decisions provided by the immediate next level, allows it to determine the future strategy of action.

Robot systems generally include a variable number of hierarchical levels depending on the complexity and degree of intelligence of the driving system used.

The hierarchically superior level is the human operator, as in other complex driving systems. It communicates with the management system in various ways, intervening periodically only in case of changing strategic driving directions, crash situations or in the event of unforeseen external disturbances.

The proper driving system comprises four hierarchical levels. The highest level corresponds to systems that have the ability to recognize obstacles in the operating area and allows for appropriate decisions to change working conditions. The immediate inferior level is called the strategic level in which the breakdown of the expected operation is performed in elementary operations. The next level is called the tactical level, in which it produces the distribution of the elementary movements in the movement on each degree of freedom, thus, within it actually generating the trajectories of motion. The last level, the lower level, is the executive level, coordinating the operation of the various actuators associated with the robot's degree of freedom.

In order to be able to complete this research project we used both hardware and software. We used metallic rods, hard plastic, Perspex, motors, screws and others for the mechanical part of the project. For arm control we opted for the LabVIEW programming environment. The power is supplied separately via a 5 V transformer and not through the acquisition board.

One chose LabVIEW because it is a very advanced, easy-to-learn programming environment with a host of applications that can only be generated when you're a beginner. The program for the robotic arm is divided into two sections. The first part makes the engine command step by step. It operates the robotic arm horizontally to allow it to reach the storage site. The other part of the program deals with the operation of the 6 DC motors. These are also the degrees of freedom of the robotic arm [9].

Besides the software used, one also utilize many hardware elements. From Perspex, hard plastic, metal rods to motors, transistors, resistors. The first attempt of this project was for the robotic arm to work with stepper motors. But one failed, the stepper motors are a bit heavy and the whole arm drops below their weight. So one chose small DC motors.

In the fig. 2 the stepper motors are visible, the gears that make the whole assembly work. One had about three toothed wheels on each engine in order for the arm to have stability. We succeeded with small DC motors. For being functional for left and right way we made 5 H-bridges, one for each engine, such was shown in Fig. 3 from below.

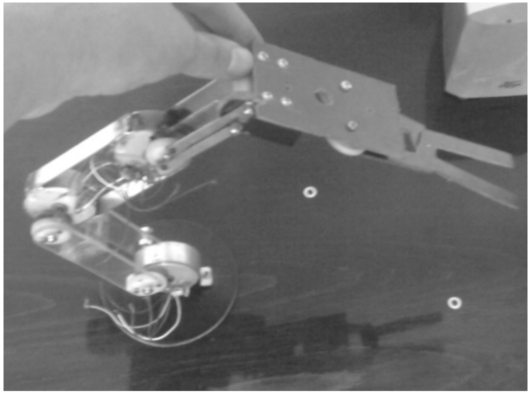


Fig. 2. Creating the Robotic Arm

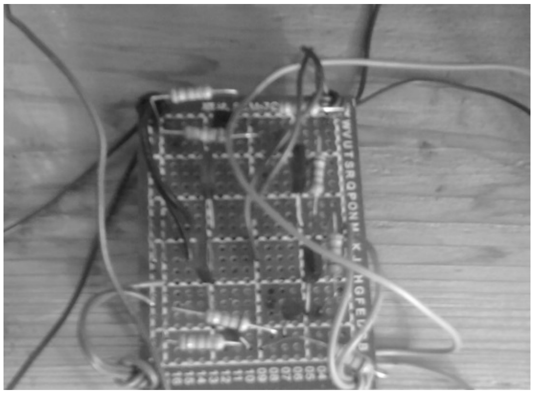


Fig. 3. Using 5 H-bridges for creating Robotic Arm

A. Making hardware components

The entire robotic arm is placed on a 80 x 40 cm hard plastic board. For horizontal movement we used two very fine metallic rods, the support to which it is attached the arm can be moved smoothly [2].

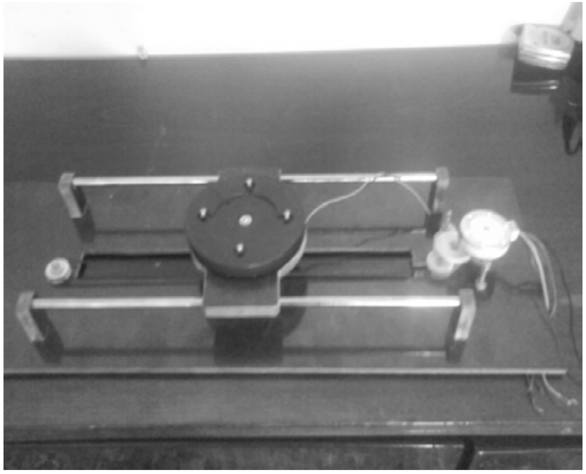


Fig. 4. Building the base of Robotic Arm

As shown in the Fig. 5 from below, a stepper motor is observed. This, together with the toothed wheel assembly, makes it possible to move the robotic arm horizontally.

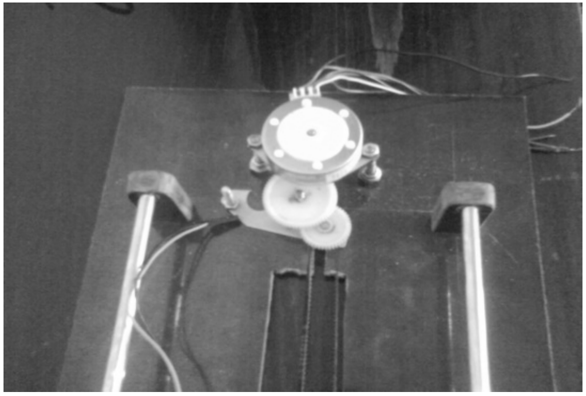


Fig. 5. Mounting the Stepper Motor to drive Robotic Arm

We have opted for this type of engine (stepper motor - Fig. 5) to move horizontally because we needed a powerful engine that has a higher torque and made it possible to move left and right without other electronic components like bridge H.

Because initially one wanted to use only step-by-step engines for the construction of the robotic arm, one found that they had some important drawbacks: fairly high weight and increased gauge. As shown in the figure below (Fig. 6), the robot arm was heavy and unstable. For stability one needed many gears, toothed wheels, but one gave up the idea of using stepper motors [7].

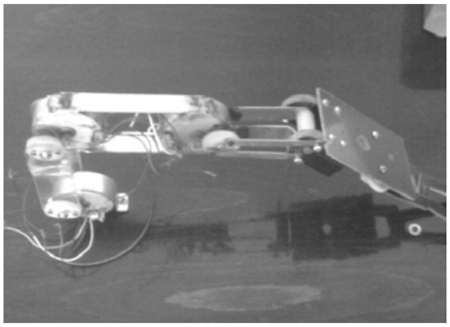


Fig. 6. Showing the possible instability for the Robotic Arm

But we found a solution. Using of small DC motors was a way to follow. With these pieces one started the design of the robotic arm. With some plexiglass, a few screws and motors, one finished our robotic arm.

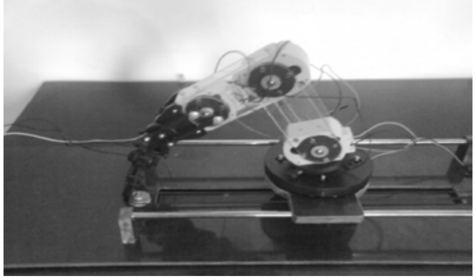


Fig. 7. The final for of a Robotic Arm

For a greater complexity we wanted to attach some sensors to indicate the position of each object to be taken by the robotic arm. When the application was put into operation, a check of the pieces was to be made and then the arm decided where it would be placed, because we placed them randomly, either in the place of storage, or in the place of reception, or where the arm would to take the pieces. We gave up, for the moment, of this idea because of the lack of acquisition plates. But, still now the arm will take the pieces and place them in the indicated places [5], [8].

Having in view the electronic part of this project means one H-bridge for each DC motor for forward and reverse displacement and a circuit to control step-by-step motor (as we shown in Fig. 8) [13].

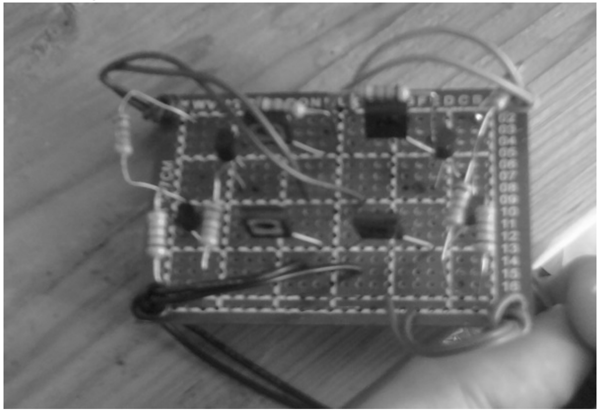


Fig. 8. Controlling the step by step motors for action of the Robotic Arm

Engine control is done through an acquisition board. It is connected to the PC via USB and the impulses for the operation of the motors are given by some pulse generators made in the LabVIEW programming environment.

The organization of management programs is a very important aspect in the robot's management systems, depending largely on the performance and complexity of the hardware solutions adopted.

The simplest way of realizing the programs is based on the sequential exploitation of the component blocks in a fixed structure of constant length.

Each block of information contains the data set that defines a particular command. The sequence of orders, in a well-ordered order, determines a complete technological operation of the robot [12].

The complexity of the management system and the degree of difficulty of the executed operations determine the adoption of specific technologies for the implementation of the driving laws.

The solutions offered by most robots and industrial manipulators can be grouped into two classes: wired logic implementations and flexible logic (scheduled) implementations. The first class is representative of those types of sequential and manipulative robots that have management systems with no more than two hierarchical levels (inferior), executive and tactical levels.

The driving laws are of a sequential type and have a pronounced rigidity, the motion trajectories imposed by the executed operations, keeping their shape and significant size constant. Technologically these systems are made in two variants: fluid and electronic. Fluid solution is encountered in the first types of industrial manipulators, nowadays almost exclusively electronic technologies are preferred.

Flexible driving solutions meet the demands of modern industrial robots. They offer great advantages both in terms of performance and the complexity of the problems being treated and the simplicity of the hardware used. Basically, these systems are implemented in four variants: flexible, microprogrammed, with programmable machines, microprocessors and microcomputers [16].

This scaling also indicates the complexity of the solutions adopted, the microprogrammed structures corresponding to simple driving laws and the management with microprocessors and micro-calculators being specific to the complex systems with an adaptive behavior to the changes in the operating environment.

The complexity of the functions performed and the technological implementation solution adopted are elements that predict a certain management architecture.

Flexible logic structures with microprogrammed logic are made from a sequential machine consisting of a programmable sequencer and a memory whose set of instructions defines both the commands required for each sequence and the future evolution of the automaton and the external conditions that can govern this evolution.

The flexibility of the system is conferred by the elasticity of the information residing in memory both in quantitative and qualitative terms.

A superior driving structure is provided by programmable machines. Instead they replace cabling structures with programmable logic structures such as semiconductor memories, programmable logic networks (PLA), programmable specialized circuits, etc. In this way, there is a high flexibility in operating modes, a maximum utilization of memory capacities, and the introduction of a dialogue unit allows human operator intervention in establishing priority work regimes.

Microprocessor management systems perform a number of robotic leadership and supervision functions using the full range of logic and arithmetic operations, quantitative and qualitative evaluations of some sizes, signal processing by discrete driving algorithms, etc.

These systems have hardware support 8 or 16 bit or microprogrammed microprocessor families.

For mixed processing of binary and numerical signals, such systems are also implemented in bi-processor configurations consisting of different sections specialized on the interpretation of purely sequential or numerical quantities

Such systems are naturally introduced into hierarchical leadership structures, each processor taking up one of the hierarchical levels of robotics [12].

B. Making the software using the LabVIEW Graphic Environment

One chose to command the engines in LabVIEW. LabVIEW is a powerful graphical programming environment, extensively used for signal acquisition, measurement analysis, and data presentation, offering the flexibility of traditional programming languages and a user-friendly interface. LabVIEW is available for a variety of platforms, Windows, Linux, HP-US, and Sun Solaris [2].

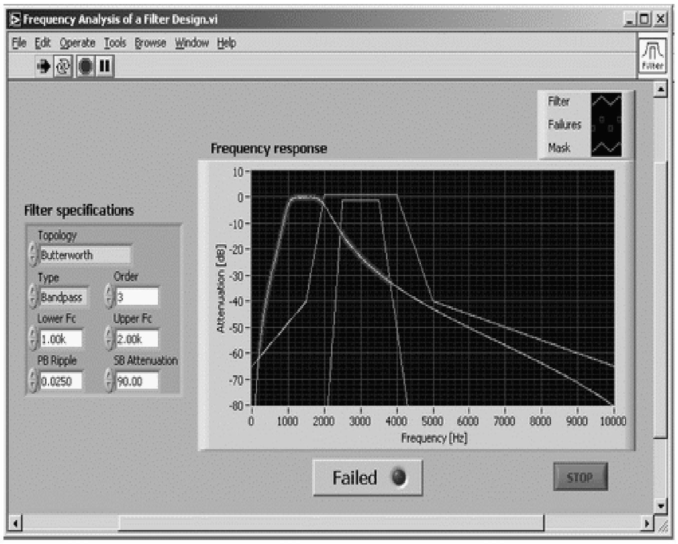
The main feature of LabVIEW is that it uses intuitive front panel icons and block schemes to develop applications. The user develops the software application through the hierarchy of Virtual Instruments. A virtual tool is a package of graphical programs that look and act as a tool.

The front panel (with buttons, switches, switches, toolbars) shows the inputs, outputs, and is the usual interface for interactive operations. Behind the panel is a block diagram, which is the executable program.

LabVIEW is a hierarchical system because a virtual tool can be represented as a graphical symbol and used in the block diagram to construct another virtual tool [10], [11].

LabVIEW presents the applications it contains in libraries, describing them in detail, front panels, block diagrams, and symbols being provided with a complete graphical and functional description.

LabVIEW allows the acquisition of signals from a variety of equipment. Data from GPIB, serial, Ethernet, PXI and VXI tools can be purchased using the included drivers. There is the possibility to communicate with more than 1400 tools from 150 manufacturers using the LabVIEW communication drivers. The program offers high performance and portability for the users.



Filter specifications
Topology: Butterworth
Type: Bandpass
Order: 3
Lower Fc: 1.00k
Upper Fc: 2.00k
PB Ripple: 0.0250
SB Attenuation: 90.00

Fig. 9. A virtual instrument for analyzing the frequency of a filter made with LabView.

Communication drivers simplify the tool control and the period for developing new applications, eliminating the need to learn programming protocols for each instrument. Many drivers use Visual Instrument Software Architecture (VISA) to communicate with a range of communication buses, such as GPIB or serial, using the same LabView code. Regardless of which type of bus is the tool, VISA drivers take control of the respective communication protocols.

The control of physical instrument using LabView is designed with block symbols that can be combined on the screen to build a virtual instrument software (VI). With LabView, control of automated instruments is as simple and intuitive as manipulating physical instrument panels.

The VI consists of reusable modules, the front panels of which can be intuitively used to perform measurements. In addition, each VI can be entered in a simple graphic (icon) and graphically combined with another icon to build a top-level VI.

LabView is a complete system for scientific programming, and includes extensive analysis capabilities that are useful in a wide range of applications.

LabView offers a multitude of integrated functions and additional modules specifically dedicated to measurement analysis and signal processing. With these tools, we can analyze measurements as we perform, extract and process data, and we can endow applications with the ability to make decisions based on measurement results.

Using these functions, it is no longer necessary to write our own algorithm for converting raw data into usable information.

The functions included are: primitive, statistical, digital signal processing, filtering and numerical methods.

Primitive:

- arithmetic operations, logical operations;
- a concatenation, formatting, sharing;
- date, time, alert the user;
- sin, cos etc.

Statistical Functions:

- average, standard deviation, variation, median, RMS, histograms;
- linear regression, polynomial, exponential, residual regression.

Numerical processing of signals:

- Fourrier's transformation, spectral power, convolution, correlation;
- Integration, differentiation, interpolation, decimation;
- Hanning, Hamming, Triangle, Gaussian;
- Impulse generation, pulse, rectangle, triangle, exponential, ramp.

Filters:

- Passes down, passes up, passes band, stops band, ButterWorth, Chebzshev, Elliptic, Hanning, Barlett, Bessel.

Numerical methods:

- Factor analysis, sign, sinc, erf;
- Adding, decreasing, transition from algebra to polar form and vice versa etc.

LabView combines all input and output data on the intuitive panel on the screen. Includes a wide range of viewing tools, including table and graph generation tools, 2D and 3D viewing tools. We can modify the presentation mode, colors, fonts, type of graphics and wheels at any time, zoom in / out or move these graphs with mouse help.

VI’s operate directly on their front panels. Multiple VI’s can be executed in parallel, moving the mouse between their front panels. The front control panel and data can be printed, just as they can be passed or removed from other front panels of other application programs.

LabView includes the ability to publish the web application, make it accessible as a Web page with the Web Publishing Tool. We can then access, view data generated and control our application from anywhere in the world. Input and output data can also be embedded in Microsoft Word or Excel format, easily archived or interchanged with other users.

III. ENGINE CONTROLS

It's very easy to command step-by step and DC motors through Labview. The signal comes from a “.VI” with Boolean variables (True - a command signal is sent to the acquisition board, False - is transmitted to the board a 0 (zero) signal, the motor remains in standby).

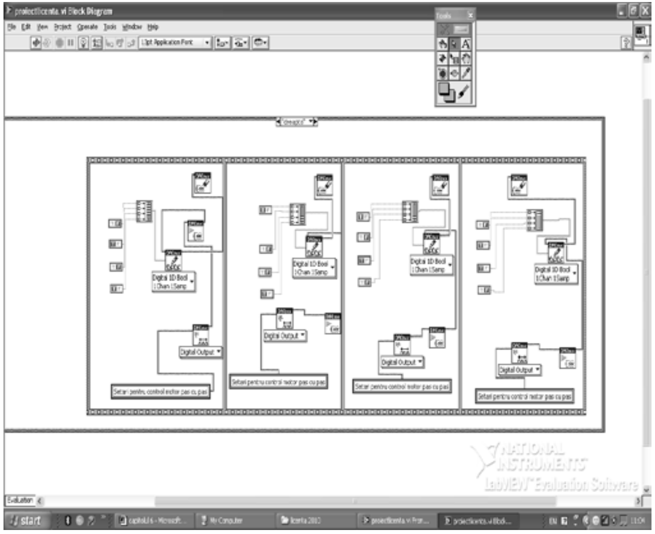


Fig. 10. The stepper motor control made with LabView

The stepper motor control is highlighted in the picture below.

The stepper motor needs to operate two such commands, according with the next figures.

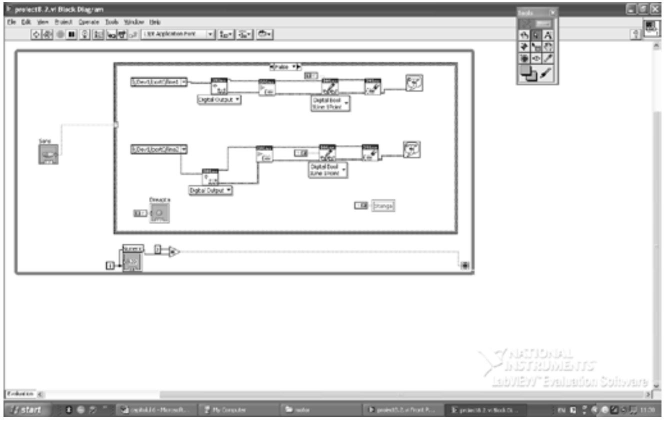


Fig. 11. The DC motor control made with LabView

One for left and one for right. It requires four boolean variables that bind to an array. Then follow the writing sequence on the acquisition board. All this is putting in a sequential structure so the engine receives a signal for each coil.

For DC motors, the program is even simpler. For every sense of the engines we have a True / False boolean variable that determines whether or not to receive the operating signal.

When an engine sense is active, the other is inactive. That is, it receives a true or false signal.

The entire project's operating scheme is presented below.

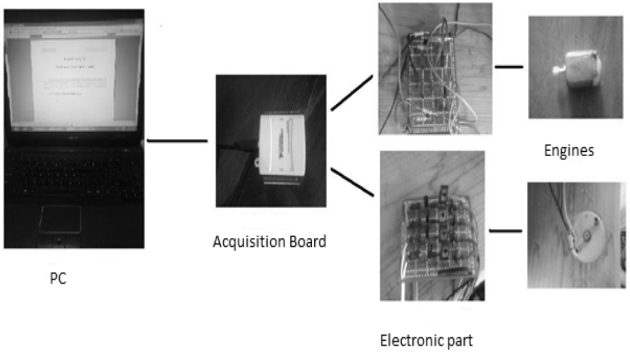


Fig. 12. The assembly operating scheme of the project

IV. CONCLUSIONS

Most robots and industrial manipulators operate in practice under anticipated conditions, running cyclically in line with technological requirements. As a result, it is possible to synthesize a nominal control of a programmed control that implements the desired movement for a particular initial state, considering that no disturbance affects the movement. Such a control can be synthesized using the centralized (global) model of the robot. Since these models are generally quite accurate, it is to be expected that the robot's trajectory by performing this control will be quite correctly executed.

The synthesis of this control is usually done off-line by first defining the motion trajectory in accordance with the technological requirements of the robot and calculating then the control quantities required to operate it. This means that the level of tactical control is reduced to a simple storage of appropriate trajectories and control sequences

Driving robots by measuring forces-moments is required in assembly operations when the robot comes in direct contact with certain objects. In this case, force control leads to a better adaptation to fluctuations in the operating space parameters while achieving satisfactory dynamic performance.

The present robotic arm was programmed in LabVIEW to collect the pieces from the locations indicated above. The work can be thought in the future as the robotic arm thinks alone. More precisely, we can put some presence sensors (LED + photodiode) to indicate the position of each object to be taken by the arm. For this we had to use a lot of conductors, connected to about four acquisition boards. If we have the ability to connect multiple acquisition boards to the robotic arm, we have to automatically check the position of each object before starting the application and then place them in the depot in the empty locations where the sensors did not detect any presence of any object.

REFERENCES

[1] Belea C, System Theory – Non-linear systems, Didactical and Pedagogical Publishing House, Bucureşti, 1985.

[2] Butucea C. A. – Monitoring and stock control in an industrial process, Licence Theme, Targoviste, 2010.

[3] Călin S., Belea C., Automatic adaptive and optimal systems, Technica Publishing House, Bucureşti, 1971.

[4] Călin S., et al, Numerical regulation of technological processes, Technica Publishing House, Bucureşti, 1984.

[5] Clare C., Designing Logic Systems Using State Machine, Mc Graw-Hill Inc. Publishing House, 1972.

[6] Crişan L., Drăgănoiu Gh., Predoi A., Flexible mounting systems with robots and manipulators, Technica Publishing House, Bucureşti, 1988.

[7] Girard P., Naslin P., Constructions des machines séquentielles industrielles, Dunod Publishing House, Paris, 1973.

[8] Hill F. J. Peterson G. R, Introduction to switching theory and logical design, John Willey Publishing House, 1974.

[9] Hill T, J., Peterson R.G., Numerical Computers, Hardware – structure and designing, Technica Publishing House, Bucureşti, 1980.

[10] Ionescu V., System Theory – Linear systems, Didactical and Pedagogical Publishing House, Bucureşti, 1985.

[11] Ionescu V., Popeea C, Structural driving of linear systems, Technica Publishing House, Bucureşti, 1986.

[12] Ivanescu M., Advanced Driving Systems in Robotics, Scrisul Romanesc Publishing House, Craiova, 2003.

[13] Lupu C., Stăncescu S., Microprocessors, circuits, design, Military Publishing House, Bucureşti, 1986.

[14] Oprean Aurel, Drimer Dolphi, Dorin Alexandru, Alexandrescu Nicolae, Industrial Robots and Manipulators, Tehnica Publishing House, 1996.

[15] Patic P.C., Pascale L., Mainea M., Mantescu G., Designing a Mobile Robot used for Access to Dangerous Areas, International Conference on Control, Artificial Intelligence, Robotics and Optimization, ICCAIRO 2017, Prague, Czech Republic, pp. 60 – 65, May 20-22, 2017.

[16] Popa I. F., Patic P. C., Activity Controlling and Monitoring of a Serving Industrial Robot, , International Conference on Control, Artificial Intelligence, Robotics and Optimization, ICCAIRO 2018 Conference, Prague, Czech Republic, May 19-21, 2018.

Paul Ciprian Patic is from Valahia University of Targoviste, Electrical Engineering Faculty, Automatics, Informatics and Electrical Engineering Department. The domain of interest are Mechatronics, Industrial Robotics and Educational Robotics, Manufacturing Flexible Systems. He is president of the Romanian Robotics Society, Targoviste Subsidiary, and, also, a member of International Federation of Robotics, or Romanian Automatics and Informatics Society. He wrote many papers presented in different national and international Conferences and published in journals and conferences proceedings.