

Controlling Systems Using PLC Ladder Bond-Graph Component in MATLAB Simulink

Ahmad Naebi¹, Iman Rezazadeh², Aghil E. Kelishomi³,
Moharam H. Korayem⁴, Samaneh Barimani⁵

^{1,2}Department of Electronic and Computer Engineering, Qazvin Branch, Islamic Azad University, Qazvin, Iran

³Department of Computer Engineering, Nour Branch, Islamic Azad University, Nour, Iran

⁴The college of Mechanical Engineering, Iran University of Science and Technology, Narmak, Tehran, Iran

⁵Department of Electronic Engineering, Science and Research Branch, Islamic Azad University, Tehran, Iran

Abstract: The bond graph is a system for modeling in Mechatronics systems. The advantage of using bond graph for PLC programming is the use of a common language for hybrid mechanical and electrical systems. For a better understanding we use Ladder which is a circuit definition language. Ladder program has been implemented in MATLAB Simulink software as an application program. We implement examples for reading and understanding. In this paper implementation of ladder program method with simulation of systems has done.

Keywords: Control Systems, Programmable Logic Controller (PLC), Bond Graph, Matlab Simulink.

Introduction

PLC (Programmable Logic Controller) is software controller, at the input, that received the binary information and they according program store in its memories then process them. The operation result of output Actions to send to the receiver and runner. (In order, a logical controller is that the control logic can define by the program and if changes are necessary, it will change easily (Fig. 1)). It was previously the charge of a relay control circuits. That new industry has tissue the using of their environments [1, 2, 3].

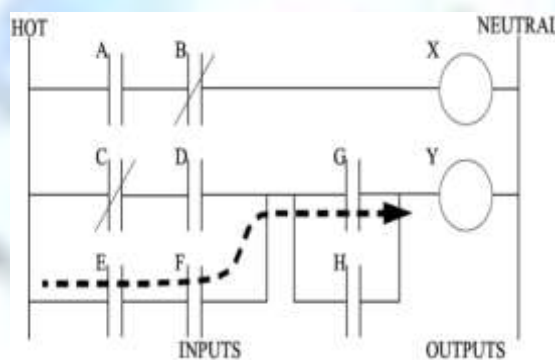


Figure 1. The sample of a simple Ladder Diagram.

Using PLC the change has done easily in production process or performance of the car. In often PLC programming has done using C, Pascal or other programming in special environments relate to them. Often PLC production has used unique language for programming that all the basic functions are related to a series of logical and general rules, the only difference is in the coding and symbols used [1, 2, 3].

Bond Graph presented in late 1950 as a single method modeling tool for modeling and analyzing dynamic systems, particularly systems in a few areas. Bond graph is base diagram method for hybrid, mechanical, electrical, pneumatic, and chemical systems (Fig. 2). In fact, it is a unlimited method in design studies that show explicit topology model, includes a set of specific elements for systems modeling that is will be molding with bind elements system together. Also in terms of bond graph modeling can be used for simulation [4, 5, 6, 7].

As regards the Mechatronics systems modeling (mechanics, electronics, etc.) with bond graph has been done previously. In order that, Mechatronics systems for modeling and programming are done with a single language. Recently a new method called a new method of programming flowchart and Mechatronic systems using bond graph is expressed (Fig. 3A). Approach of Mechatronics systems modeling and programming were done with a bond graph language which is single language [8].

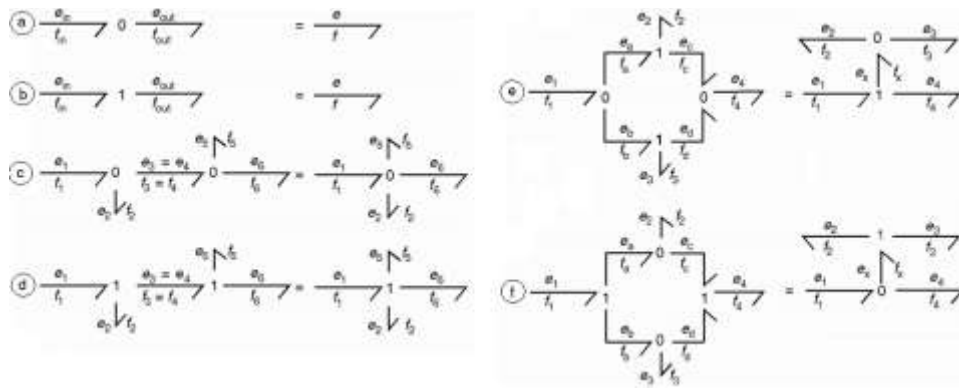


Figure 2. The elements of bond Graph.

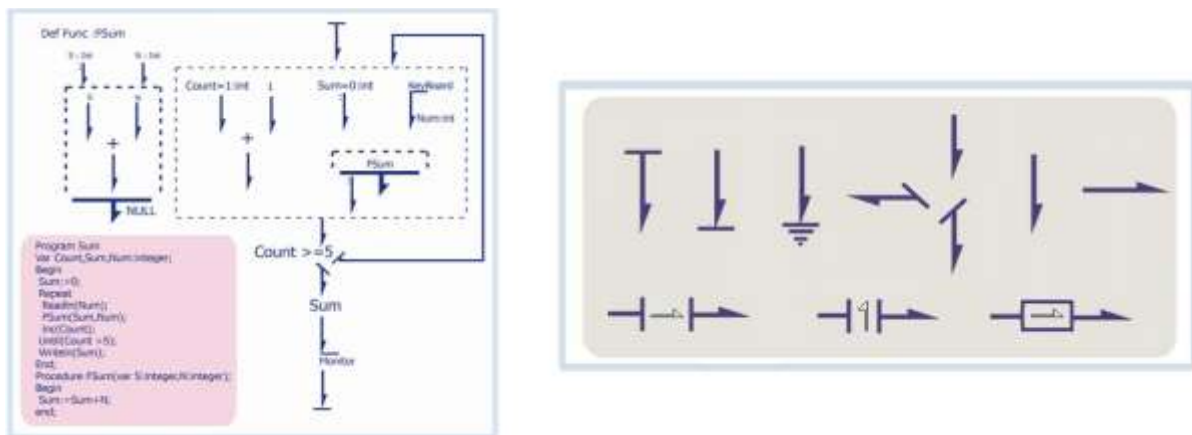


Figure 3. The view of bond graph elements, (A) A sample of programming using bond graph programming elements in left picture. (B) the proposed elements of ladder bond graph in right Picture.

The only difference is used in the coding and symbols. Advantage of this language related to other languages is not unique to PLC language. It is a general language for Mechatronics Systems. First of all, people can use PLC bond-graph component for PLC in Matlab Simulink that called components of Ladder bond-graph. This package is new PLC package for PLC in Matlab.

Related Worked

A control programming is a set of instructions that can issue commands to control the process. So it should be the specific language and understand the rules be written to PLC. An example of this language is called SETP5 that these languages are represented in different ways [1, 2, 3]. Those are: 1. Ladder, 2. Flowchart, 3. Statement.

The ladder display, any command or line program as a symbol of connection and the relay coil circuits are shown. The program is structured in this way is almost like a relay control circuits. The presentment of the old system was a conventional relay, the circuit maps are often drawn to this method. Therefore this method is more and more savory those who believe it is so easy to understand. The map which drawn with this method is like an electrical circuit of the same system. This way is some of the symbols used in this program [9, 10]. In the flowchart method, a program has showed as set of the rectangular form (blocks). This way used more during designing program. In below, a few blocks for the display are showed (Fig. 3B).

In the expression method, the program is as a set of orders which each order called line program or statement (Fig. 4). Any order or a line program contain a combinations of logical of mathematic command as the AND, OR, NOT, and equivalence. In the some program do a special action by some line called a segment. A program includes several segments [9, 10].

According to the bond graph describe physical behavior of the system in different domains of visually express Its four features make Bond graph modeling as a best design options for higher multi-domain systems in both macro and micro scales which includes [11]:

- a) The wide scope of systems that can be created because of the multi- and inter-domain nature of bond graphs, the efficiency of evaluation of design alternatives [10].
- b) Performance estimation and evaluation of design alternatives.

- c) Features of combines links and node components for the production of design alternatives.
- d) Simple of mapping the process of engineering design.

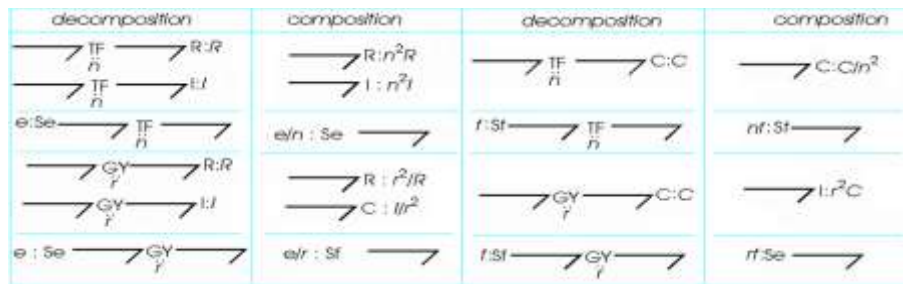


Figure 4. The elements of bond graph in different domains and between domains.

You know that elements implement in Matlab Simulink, are clearly. In fact, any company hasn't implemented elements of ladder elements for PLC in Matlab up to now. The component of Matlab Simulink has showed in Fig. 5

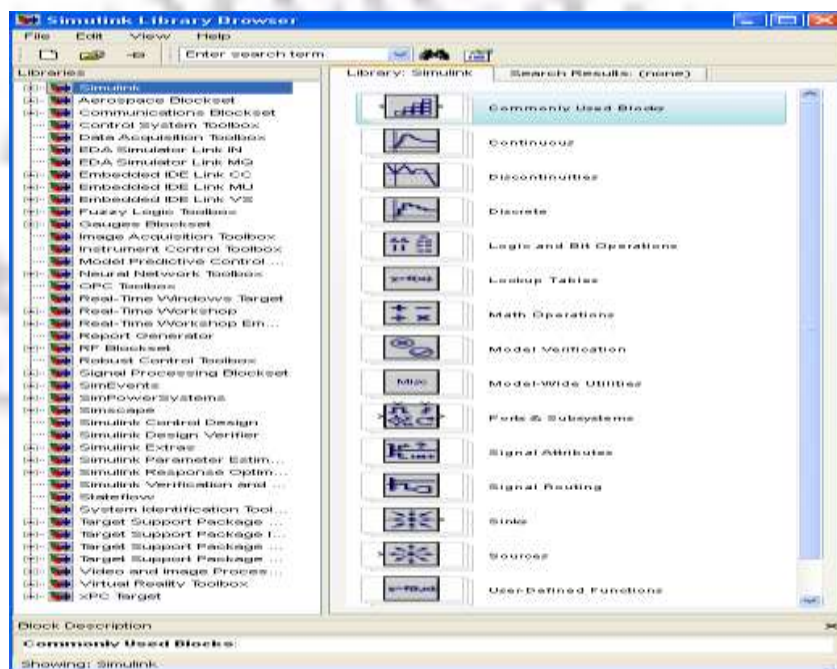


Figure 5. Matlab Simulink Components.

Proposed Methods

First, we will explain ladder bond graph implementation in visual status; we will describe it briefly for better understanding. Program begin with bond graph start element then it will catch data from input, after this step it will move to next step, processing of received data [12].

After data processing (logical and other operations) to be active or inactive status of PLC ports will be specified. Output specific data will send to PLC output port, then program will face with condition element, if program running time is over, program will finish with bond graph finish element, the else program will run while it's running time become over [12].

Another difference between our method and previous methods except common elements, are elements that are shown, which they shown in hidden status in previous methods. In other word start element is starter of all devices, finish element will be finalizing element of all devices.

Fig. 6 shows ladder bond graph elements which are implemented in MATLAB Simulink software. These elements will be used as PLC element in Simulink in simple word; these elements create PLC Pack for Matlab Simulink software [12].

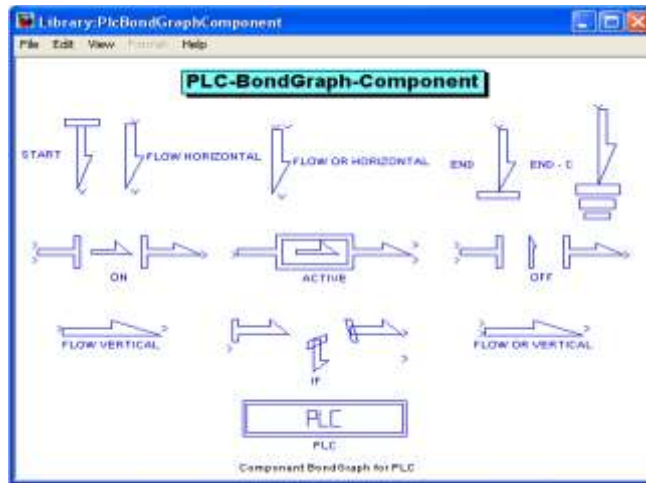


Figure 6. The elements of Ladder program for Matlab Simulink Component.

Created Micro details for elements, is similar to simulated details for other Elements in Simulink application of MATLAB Software. We will explain this element details for making better understanding. This method is implemented as logical elements, ON element has been implemented as following, 2 input which are with Double type enters in Element .AND element effect on 2 input and output, AND element is Boolean. Then, it will be multiplied by a Gain to be converted to Double and exit with ON output from Element (Type of input and output defined as Double because if we define them as Boolean, error will be occurred).

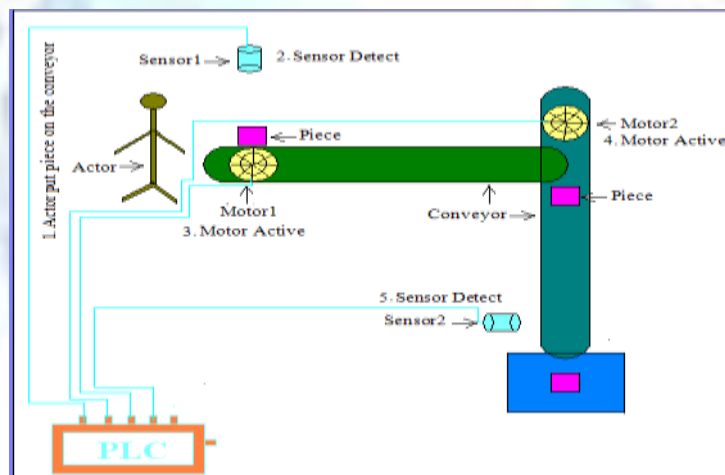


Figure 7. The view of first proposed system.

A. First Proposed System

Our first method is designing of a small system which is component transferor. In this method a worker puts a component on the conveyor, sensor detects is a component is on the conveyor, then motor A start to work followed by the second motor also will active. When component transferred to the next conveyor, again it will be in moving status and when second sensor was activated, it means component is transferred. Causes of the first engine and the second engine are switched off, while a new component is being placed on the conveyor for transport. You can see a view of the created system in Fig. 7.

Because the main transmissivity, powers are engines then we will explain implemented Motors.

DC motors are the first that are used in industries due to their easy control with wide range of power, voltage and nominal speed [13, 14, 15]. There are many methods to control of motors speed, angle, torque and etc. with some advantages and disadvantages. Fig. 8 shows schematic view of a simple DC motor. In the figure R_a , L_a , I_a , V_{in} , B , J and ω are armature resistance, armature inductance, armature current, terminal voltage, viscous friction coefficient, machine speed and Total Inertia respectively.

Following you can see equations which are described motor operations [14, 15]:

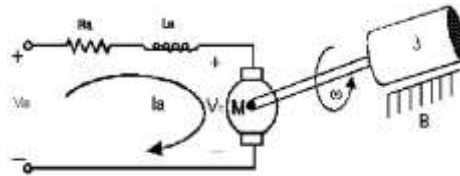


Figure 8. The view of DC motor [14, 15].

In the above T_l and E_a are load torque and counter-electromotive force respectively.

$$V_t = L_a \frac{di_a}{dt} + R_a i_a + E_a \quad (1)$$

$$E_a = K \omega m \quad (2)$$

$$J \frac{d^2 \theta}{dt} + B \frac{d\theta}{dt} - T_l = K i_a \quad (3)$$

Explained DC motor is implemented in MATLAB software in below method (Fig. 9) which is controlled with using a fuzzy controller, PWM model is used for enabling and disabling of electric connector and disconnector chapters to Motor Control [15].

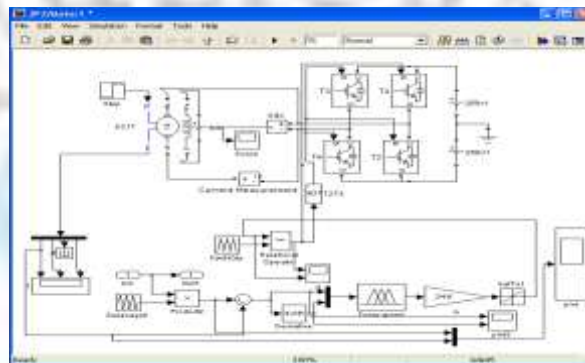


Figure 9. Control of motor DC in Simulink of Matlab [14].

In Fact we will not explain more about fuzzy control details and other issues. Our target is to show this motor is controlled by MATLAB PLC ladder elements.

B. Second Proposed System

First we will bring machine environment component and double click on it a dialog box will open. We will select Attraction and Direct Dynamic and then will bring ground option and by choosing one of Ticks we can join from one side to machine environment (Fig. 10).

We will enter [0 0 0] for ground position and from other side join it to revolute.

Its other side will be joined to body. We consider its amount of mass 20, and its length along the y axis 4. From other side we connect another arm to revolute and its other side will be connected to other body. We consider its amount of mass 15, and its length along the y axis 3. Also we connect each of revolute to 1 body sensor and 1 joint actuator and 1 joint sensor. Joint actuator input is torque. Joint sensor has four outputs, which with using of Demux, we use only the angle. Output of body sensor is position of 3 x, y, z axis that we use y axis. We have one reference angle that is difference with output angle of joint sensor which will get and send to the PID (Fig. 11).



Figure 10. The view of two Degree of Freedom with contest in grand.

A.1. Direct kinematics equations [16]

Direct kinematics equations can be expressed as follows:

$$X(\theta_1, \theta_2) = L_1 \cos(\theta_1) + L_2 \cos(\theta_1, \theta_2) \quad (4)$$

$$Y(\theta_1, \theta_2) = L_1 \sin(\theta_1) + L_2 \sin(\theta_1, \theta_2) \quad (5)$$

Inverse kinematics equations:

$$\theta_1^* = \frac{V_x \cos(\theta_1, \theta_2) + V_y \sin(\theta_1, \theta_2)}{\sin \theta_2} \quad (6)$$

$$\theta_2^* = \frac{V_x [\cos \theta_1 + \cos(\theta_1, \theta_2)] + V_y [\sin \theta_1 + \sin(\theta_1, \theta_2)]}{\sin \theta_2} \quad (7)$$

Jacobian matrix:

$$dx = \frac{\partial x(\theta_1, \theta_2)}{\partial \theta_1} d\theta_1 + \frac{\partial x(\theta_1, \theta_2)}{\partial \theta_2} d\theta_2 \quad (8)$$

$$dy = \frac{\partial y(\theta_1, \theta_2)}{\partial \theta_1} d\theta_1 + \frac{\partial y(\theta_1, \theta_2)}{\partial \theta_2} d\theta_2 \quad (9)$$

$$dx = Jd\theta$$

$$J = \begin{bmatrix} \frac{\partial x}{\partial \theta_1} & \frac{\partial x}{\partial \theta_2} \\ \frac{\partial y}{\partial \theta_1} & \frac{\partial y}{\partial \theta_2} \end{bmatrix} = \begin{bmatrix} -L_1 \sin(\theta_1) - L_2 \sin(\theta_1, \theta_2) & -L_2 \sin(\theta_1, \theta_2) \\ L_1 \cos(\theta_1) + L_2 \cos(\theta_1, \theta_2) & L_2 \cos(\theta_1, \theta_2) \end{bmatrix} \quad (10)$$

We will put DE terminal the Jacobean matrix zero.

$$\text{Det}(J) = 0 \quad (11)$$

$$(-L_1 \sin \theta_1 - L_2 \sin(\theta_1, \theta_2)) * L_2 \cos(\theta_1, \theta_2) + L_2 \sin(\theta_1, \theta_2) * (L_1 \cos(\theta_1) + L_2 \cos(\theta_1, \theta_2)) = 0$$

$$L_1 L_2 (\sin(\theta_1, \theta_2) * \cos \theta_1 - \sin \theta_1 * \cos(\theta_1, \theta_2)) = 0 \quad (12)$$

$$\frac{\sin \theta_1}{\cos \theta_1} = \frac{\sin(\theta_1, \theta_2)}{\cos(\theta_1, \theta_2)} \quad (13)$$

$$\tan \theta_1 = \tan(\theta_1, \theta_2) \Rightarrow \theta_1 = \text{KPI} * (\theta_1, \theta_2) \quad (14)$$

A.2. Dynamical equations

The dynamical equations can be stated as follows:

$$T_1 = \frac{1}{2} I_{1A} \cdot \dot{\theta}_1^2 \Rightarrow T_1 = \frac{1}{2} \bar{I}_2 \cdot \omega_2^2 + \frac{1}{2} m \cdot v_{G2}^2 \quad (15)$$

$$V_{G2} = V_B + \omega_2 * \vec{r}_{B/G2} + \vec{v}_{rel} \Rightarrow V_{G2} = \dot{\theta}_1 \cdot \hat{k} * \vec{r}_{H/B} + \omega * \vec{v}_{B/G2} + \omega_F^B * \vec{v}_{rel} * \vec{r}_{G2/B} \quad (16)$$

$$\omega_F = \dot{\theta}_1 \cdot \hat{k} * \dot{\theta}_2 \cdot \hat{k}, \quad \omega_F^B = 0 \quad (17)$$

$$V_{G2} = \dot{\theta}_1 \cdot \hat{k} * (L \cos \theta_1 \cdot \hat{i} + L \sin \theta_1 \cdot \hat{j}) + (\dot{\theta}_1 + \dot{\theta}_2) \cdot \hat{k} * \frac{L}{2} \hat{i}, \quad i = \cos(\theta_1 + \theta_2) \hat{i} + \sin(\theta_1 + \theta_2) \hat{j} \quad (18)$$

$$V_{G2} = [L \dot{\theta}_1 \cos \theta_1 + \frac{L}{2} (\dot{\theta}_1 + \dot{\theta}_2) \cos(\theta_1 + \theta_2)] \hat{j} - [L \dot{\theta}_1 \sin \theta_1 + \frac{L}{2} (\dot{\theta}_1 + \dot{\theta}_2) \sin(\theta_1 + \theta_2)] \hat{i} \quad (19)$$

$$T = \frac{1}{2} (\frac{1}{3} m L^2) \dot{\theta}_1^2 + \frac{1}{2} [\frac{1}{12} (2m) L^2] (\dot{\theta}_1 + \dot{\theta}_2)^2 + \frac{1}{2} (2m) v_G^2 \quad (20)$$

$$V_{G2}^2 = [L^2 \dot{\theta}_1^2 + \frac{L}{2}(\dot{\theta}_1 + \dot{\theta}_2)^2 + L^2 \theta_1(\dot{\theta}_1 + \dot{\theta}_2) \cos \theta_2] \quad (21)$$

$$T = \frac{1}{2}(\frac{1}{3}mL^2) \dot{\theta}_1^2 + \frac{1}{2}[\frac{1}{12}(2m)L^2](\dot{\theta}_1 + \dot{\theta}_2)^2 + \frac{1}{2}(2m)[L^2 \dot{\theta}_1^2 + \frac{L}{2}(\dot{\theta}_1 + \dot{\theta}_2)^2 + L^2 \theta_1(\dot{\theta}_1 + \dot{\theta}_2) \cos \theta_1 \cos(\theta_1 + \theta_2) + L^2 \theta_1(\dot{\theta}_1 + \dot{\theta}_2) \sin \theta_1 \sin(\theta_1 + \theta_2)] \quad (22)$$

$$U = mg \frac{L}{2} \sin \theta_1 + 2mg[L \sin \theta_1 + \frac{L}{2} \sin(\theta_1 + \theta_2)] \quad (23)$$

$$L = T - U \quad (24)$$

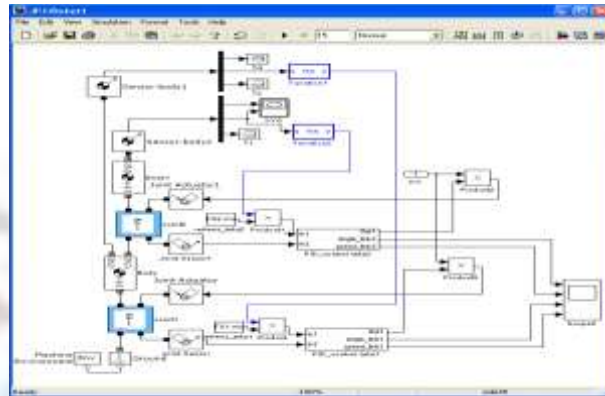


Figure 11. The control of two degree of freedom (DOF) in Simulink of Matlab

The expressed arm is two degree of freedom has been implemented in the MATLAB simulator Software as following (Fig. 11). It includes explained Mechanical elements that are connected to each other which follow rules expressed in the above equations. In this part we won't explain elements in detail, because we explained them in detail in above; our target is to show Control of arm with two degree of freedom by ladder elements in MATLAB.

Simulation and Result

A. First Proposed System

Expressed system in Figure 12, has been implemented as blocks. Whole system has been implemented as five blocks that they can exchange information with each other. The first block is engine, implementation block which move conveyor that user puts component on it. This is DC engine implementation block for transfer. Second Block which is similar to first block engine is used for next conveyor moving and with activation of first engine. Also it will be active immediately. Third block is Sensor 1 block which is include sensor implementation in simple status and its duty is enabling sensor in a specify period of time. 4rd Block again is Sensor 2 which after specified time of first sensor activation, will be active .This interval is average of component transfer time on 2 conveyors. Last block is control block which is control commands implementation block with using of band graph ladder elements in PLC MATLAB, we will describe completely that how it control whole system.

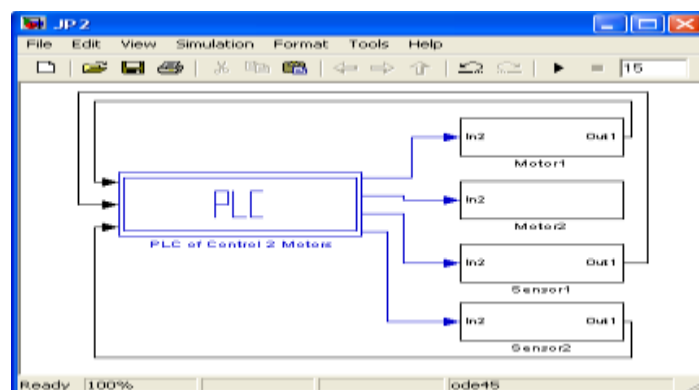


Figure 12. The Control double DC motors and sensors with PLC package.

Fig. 13 show a total view of system control with using of MATLAB band graph elements to transfer components by conveyors by 2 motors and detecting of component placing on conveyor and components full transfer by 2 sensors. System begins with start element then next element will become AND with return element output. Output will transfer to horizontal and vertical elements.

From output of the first row , output 1 will enter to sensor 1 ON element and will become AND with Sensor 1 input and its output will become AND with NOT value of sensor 2. Output will transfer to motor 1 (Sensor 1 will become active with putting component on conveyor and sensor is inactive in result, motor 1 will be active).

From output of the Second row output 1 will enter to ON element of motor 1 and will become AND with motor 1 input. Output will transfer motor 2 Activator (with motor 1 activation, motor 2 will become active too).

From output element of third row, output 1 will become AND with NOT value of Sensor 2 and output will transfer to sensor 1 activator.

From output element of fourth row, output 1 will become AND with NOT value of Sensor 1 and output will transfer to sensor 2 activator.

Outputs with elements connect to ground in vertical element before rows it will face with a conditional element which if you want before specified time system becomes down, it can reach to program End element. In this program 1 considered to execute until End of program.

While sensor 1 is active, sensor 2 activator will be inactive; also while sensor 2 is active sensor 1 activator will be inactive.

The chart of motor speed, after execution is as follows that shown in Fig. 14. In chart Reference velocity diagram is (yellow), actual speed of the motor output is (brown). It takes a little time, that engine speed be converges to actual speed. Before the actual speed of the engine loading, it follows the reference speed. During loading it will create fluctuation, fluctuation, decrease rapidly. This means motor actual speed follow Reference speed fast (this is a short time), loading will be done in motor in starting time.

B. Second Proposed System

Assuming (in conditions) two robots (the arm with two degrees of freedom) cannot simultaneously do the welding and pressing, therefore, welding and pressing should be done with interval. In This method when first robot is welding, which means is active. The next robot is disabled. When second robot becomes active and begin pressing. The first robots will be idle until the second robot done all its work. We consider that there is need specific time to each of them which Action they are doing to successfully finish. Whole system has been implemented in 3 Block that they can exchange information with each other (Fig. 15).

First Block is implemented of arm with two degrees of freedom for Welding and second block is arm with two degrees of freedom for pressing. System works as follow that with specific interval arm with two degrees of freedom for welding will be active and when it is become inactive, pressing arm will become active. Last block is control block which is control commands implementation block with using of band graph ladder elements in PLC MATLAB for above mentioned robots.

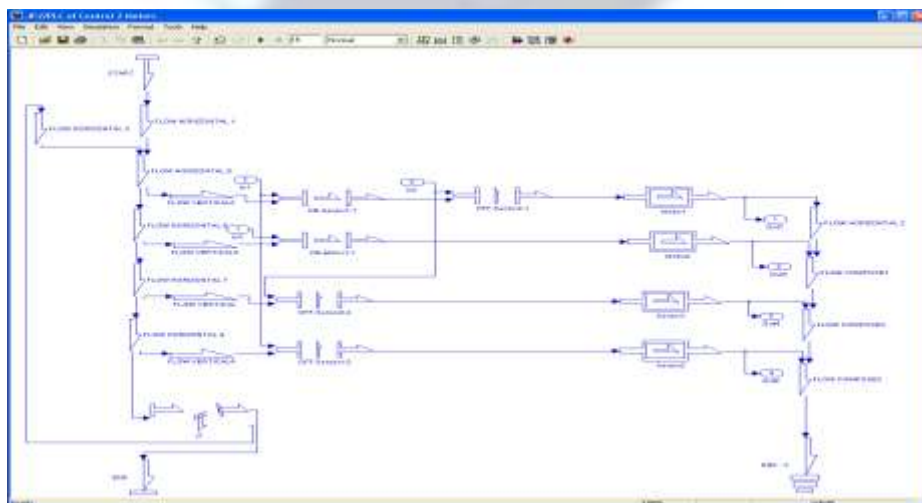


Figure 13. The Control of transfer piece on conveyor using DC motors using ladder elements in Matlab Simulink.

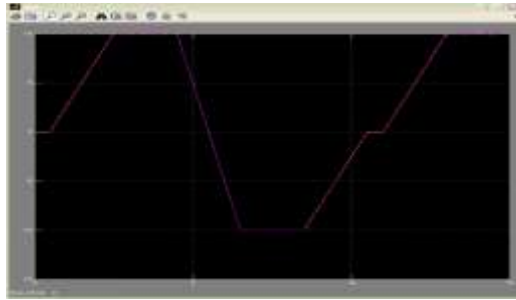


Figure 14. The diagram of speed motor DC with reference value.

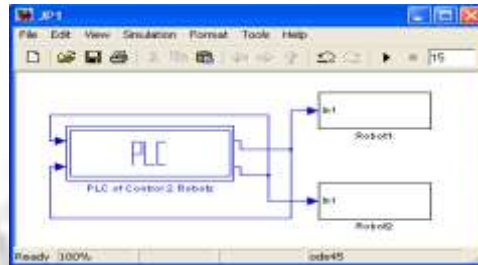


Figure 15. The Control two degree of freedom (DOF) robot with PLC package.

Arms control in the workspace: We have a vector of angles reference which angles are placed with 1 interval from each other. Controller in specific period of time will control angle corresponds to that time. We have two controllers: 1- base Arm angle controller (First arm angle). 2- Connected Arm angle controller (second arm angle).

Controllers are PIDs that their input value is difference between reference angle and arm angle which is obtained by sensor and controllers, output is power which is applied to Stimulus Articulation.

Fig. 16 shows a view of arm with two degrees of freedom which is simulated in MATLAB.

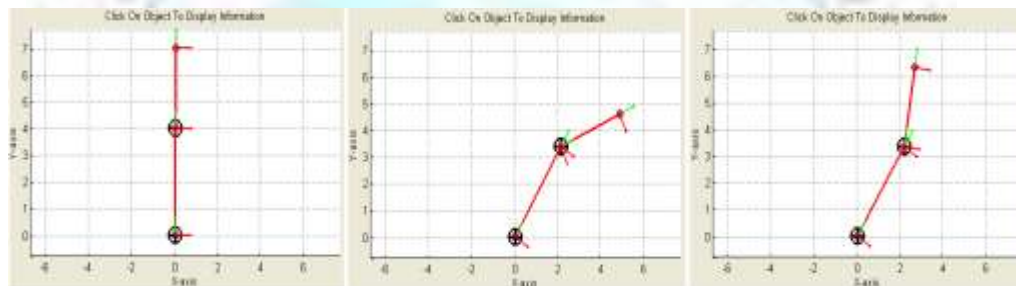


Figure 16. The view of two degree of freedom in three states in Matlab Simulink

In highest figure, the robot is in vertical mode which is starting to move for welding or pressing. In middle and bottom figure robot changes its arms angles and moves in a way that reach desired point for welding or pressing. (Pressing and welding tools are connected to head of arm). Therefore in this 2 figure two valid points are shown, which pressing or which welding has been done there.

Fig. 17 show a total view of control system with using of MATLAB Bond graph Elements for two arms with 2 degree of freedom to welding and pressing.

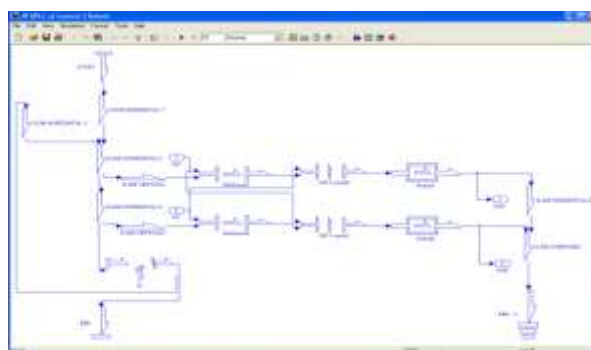


Figure 17. The Control of two degree of freedom (DOF) using ladder elements in Matlab Simulink

Outputs with elements connect to ground. In vertical element before rows it will face with a conditional element which if you want before specified time system become down, it can reach to program End element. In this program 1 considered to execute until End of program

System begins with Start Element then next element will become AND with return element output. Output will transfer to horizontal and vertical elements.

From output of the first row, output 1 will enter to welding robot ON element and become AND with active input of welder robot and its output will become AND with NOT value of pressing robot input. (If welder robot input is active and pressing robot input is inactive then welder robot will stay active).

Output from vertical element will transfer to second row horizontal element. From output of the Second row output 1 will enter to pressing robot ON element and become AND with active input of pressing robot and its output will become AND with NOT value of welder robot input. Output will transfer pressing robot activator. (If pressing robot input is active and welder robot input is inactive then pressing robot will stay active). Outputs with elements connect to ground. In vertical element before rows it will face with a conditional element which if you want before specified time system become down, it can reach to program End element. In this program 1 considered to execute until End of program.

In this controller, two robots cannot be active simultaneously when one of them is active, other one will be inactive.

Fig. 18 shows, head movement of arm with 2 degree freedom which broken parts in 2 lines are welding and pressing points, therefore you can see the paths and points that arm has perform the operation there .

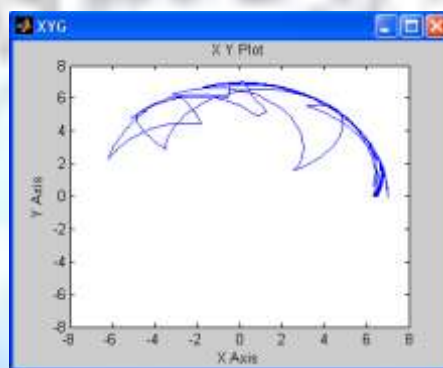


Figure 18. The paths and points for performing of operation using 2 DOF

Compare PLC

This New elements are designed for MATLAB simulator that users be able to run system controllers in a simulated environment in MATLAB simulator software, because PLC simulator for system control is as a software which is same PLC programming language.

The benefits of this new method compared to other PLC software include:

- a) Creating new controlling elements with Bond Graph in Systems control software: which can be explained from 2 point of view, first we can control systems that are implemented with MATLAB elements. Second, with using of these new elements we can control systems that designed with Bond Graph modeling package for MATLAB.
- b) Low cost of software package: a software package which is added to MATLAB, no need to more cost but for preparing a new software will need to more cost.
- c) Whole system create with controller but most PLC software only implement Controlling part (some systems implemented but in most of them this is not considered).
- d) As add-ons of application software which is used with many users, it can be more useful for making users familiar with PLC in a simulated environment before practical work.
- e) Use of specific Mechatronics language (bond graph) for controlling language: Bond Graph which is Bond graph modeling language. It is better to PLC language implemented for Mechatronic field with this language because its understanding will be in high level.

Conclusion

In this paper a new programming language for PLC with using of bond graph is defined. This type of programming has been defined before for mechatronics field, but it has not been defined for PLCs. With definition of this programming language a new programming language will create for PLC which will be unique. New elements in

proposed method section defined, that is easy to understand for readers. Also we explain two application examples in Simulink of Matlab using PLC Ladder component. This is an overall method that will make us needless to using different programming language in mechatronics field. In fact this method will be prepared students use it for better understanding from PLC programming and simulating of systems before they implement actual systems.

References

- [1]. Siemens-Simatic S5-115 U. Programmable Controllers (1989).
- [2]. Festo. Programmable Controllers (1995).
- [3]. Warnock, I.G.: Programmable Controllers Operation and Application. Prentice-hall (1988).
- [4]. Breedveld, P.: Bond Graph. Introduction for Bond Graph.
- [5]. Gawthrop, P.J., Bevan, G.P.: Bond-Graph Modeling. 24 IEEE Control System Magazine (April 2007).
- [6]. Broenink, J.F.: Introduction to Physical Systems Modeling with Bond Graphs (1998).
- [7]. Gawthrop, P.J., Smith, L.P.S.: Metamodeling: Bond Graphs and Dynamic Systems. Prentice Hall, Hemel Hempstead (1996).
- [8]. Naebi, A., Khalegi, F., Hosseinpour, F., Zanjanab, A.G., Khoshravan, H., Kelishomi, A.E., Rahmatdoustbeilankouh, B.: A New Flowchart and Programming Technique Using Bond Graph for Mechatronic Systems. In: 2011 UKSIM 13th International Conference on Modeling and Simulation, Cambridge, United Kingdom, March 30-April 01 (2011) ISBN: 978-0-7695-4376-5.
- [9]. Zalata, M.A., Alia, M.: Realizing Sequential Processes Using Programmable Logic Controllers (PLCS). Simulation 5(3-4), ISSN 1473-804x online, 1473-8031 print.
- [10]. Fan, Z., Seo, K., Rosenberg, R.C., Hu, J., Goodman, E.D.: Exploring Multiple Design Topologies using Genetic Programming and Bond Graphs. In: Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2002, New York, pp. 1073–1080 (2002).
- [11]. Parker, J.K.: Introduction to Programmable Logic Controllers in a Mechanical Engineering Instrumentation Course. In: Proceedings of the 2001 American Society for Engineering Education Annual Conference & Exposition Copyright (C) 2001. American Society for Engineering Education (2001).
- [12]. Ahmad Naebi, Masoud Hassanpour Aghdam, Ali Ghalebhan Zanjanab, Esmaeil Nourani, "A New Programming Language for PLC Using Bond Graph", Advances in Automation and Robotics, Lecture Notes in Electrical Engineering Volume 122, Springer 2011, pp 73-80.
- [13]. Akin Delibaş, Türker Türkiye, Galip Cansever, Real time DC motor position control by fuzzy logic and PID controllers using LABVIEW.
- [14]. Dave Polka. Motors and Drives. Chapter 3. The Instrumentation, Systems, and Automation Society.
- [15]. A. Naebi, H. Khoshravan, M. H. Koryem, "Comparing of PSO, AFSA and hybrid PSO-AFSAmethods for optimizing of fuzzy logic controller of DC motor", Proceeding of the 4thinternational symposium on mechatronics (AISM 2010), pp 297-303.