American J. of Engineering and Applied Sciences 4 (2): 252-264, 2011
ISSN 1941-7020

Petri Nets and Ladder Logic for Fully-Automating and Programmable Logic Control of Semi-Automatic Machines and Systems

[1]Mahmoud A. Barghash, [2]Osama M. Abuzeid,
[3]Anas N. Al-Rabadi and [1]Ahmad M. Jaradat
[1]Department of Industrial Engineering,
[2]Department of Mechanical Engineering,
[3]Department of Computer Engineering,
University of Jordan, Amman, Jordan

Abstract: Problem statement: Automating semi-automatic dynamic machines and complex systems, where some functionalities are already automated but the machine doesn't run fully in automatic mode, represents another challenge to automating dynamic and manual complex systems. This study deals with a special case of this general problem, where the system's semi-automatic and automatic functionalities run separately in a sequence with either automatic and/or semi-automatic functionality running at any point of time, but not both. **Approach:** Petri-nets can successfully represent the operations of both the semi-automated and fully-automated functionalities of such dynamic systems. Three roles for the programmable logic controllers were suggested: (a) Ignoring the presence of the semi-automatic part, (b) simulating the semi-automatic part and recognizing discrepancies and(c) simulating the semi-automatic functionalities and supporting it, where the last choice was most costly and most reliable. **Results:** This study presents a case study for a PVC mixing dynamic process to illustrate the three suggested controller design possibilities, where a Petri net model and related ladder logic program were developed to show these three controller design options. In this respect, the semi-automatic functionality is an intermediate step between the two automated functionalities, within which an important decision should be made to end the previous automatic step and to begin the next automatic step. **Conclusion/Recommendations:** The automation method that was introduced in this study is applicable to a large number of machines within industrial and mechanical systems that were built using older semi-automatic control systems. Various decision analyses {DA1, DA2, DA3} were shown to produce basic choices for such types of applications. Petri-nets and ladder logic in this respect can successfully represent the operations of both the semi-automated and fully-automated functionalities, where a Petri net model and the consequent ladder logic program, used to program the utilized programmable logic controller, are developed to show three possible controller designs, where the choice between three designs depend on the designers objective of cost, reliability and fault tolerance.

Key words: Automation method, control systems, decision analysis, dynamic systems, fully-automated functionalities, ladder logic, petri net, programmable logic controller, semi-automatic functionalities

---

INTRODUCTION

The development of fully automated machines and systems posses the possibility that the Programmable Logic Controllers (PLCs) (Stenerson, 1999) which are cyclic controllers that go through all basic inputs and outputs in a repetitive manner running a Boolean logic (Al-Rabadi, 2004) may not have the full control of the machine as some machine functionalities may still be running by semi-automatic hardware Ladder Logic (LL) (Pollard, 1994). The challenge is slightly different from building a control system from scratch, where automating a semi-automatic machine takes shorter time than building the control system from scratch and it also requires less costly PLCs, that makes this choice more appealing to the factories' managers as the machine may still be available while the new controller is installed with little or no down time, with

Corresponding Author: Mahmoud A. Barghash, Department of Industrial Engineering, University of Jordan, Amman, 11942, Jordan

less time and less cost for achieving the project. However, this does not make this type of implementation any less challenging to the designer or the implementer; on the contrary, the stress will be higher not disrupting any of the already running semi-automatic functionalities especially when the system is a capacity bottleneck, i.e., other machines may be short supplied if this machine is off for a long period of time.

This study is addressing a case study of automating a special type of semi-automatic machines when the semi-automatic and automatic functionalities do not run concurrently, instead they can be represented in sequence with any of the semi-automatic functionalities being preceded and followed by manual functionalities to be automated. As shown in Fig. 1 the basic requirement for automation is to replace the manual operation, while the semi-automatic operation remains intact.

Petri Nets (PNs) is one of several mathematical modeling languages for the representation of distributed and asynchronous systems (Zurawski and Zhou, 1994). Like other industrial standards such as the UML activity diagrams, BPMN and EPCs, PNs offer a graphical notation for stepwise processes that include choice, iteration and concurrent execution. Unlike these standards, PNs have an exact mathematical definition of their execution semantics, with a well-developed mathematical theory for process analysis. In this study, PN is used to model both of the semi-automatic and the automatic operations. This in turn is used to generate the Ladder Logic to program the Programmable Logic Controllers. The PLCs were introduced to replace the more costly conventional controllers, where PLCs include the PLC logic timing and sequencing in an analogous way to traditional electric circuits and systems.

The reason for choosing ladder diagrams was the familiarity of such technique in manufacturing (Pollard, 1994). The International Electrotechnical Commission (IEC) proposed the IEC 61131-3 standard for the Industrial Process Measurement and Control Systems (IPMCS) which was adopted by most of the PLC manufacturers including Schneider Electric, Rockwell Automation ICS Triplex ISaGRAF, Infoteam, KW Software, Mitsubishi Electric, Panasonic, Smart Software Solutions, ABB or WAGO (Estévez *et al*., 2007).

The international standard (IEC1131-3) included sequential function charts, function block diagrams, structured texts, instruction lists and ladder diagrams (Pollard, 1994). The control language that is predominantly used in PLCs is still the LL. Using LL, it is possible to represent the control processes both sequentially and graphically, where field engineers have utilized this language to represent process control efficiently.

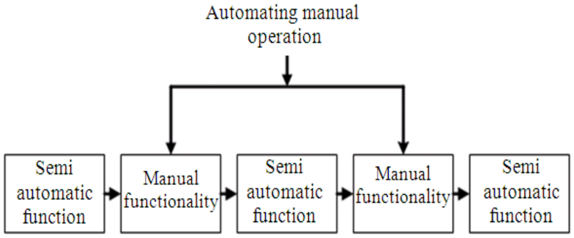


Fig. 1: Automating a semi-automatic machine with separate non-concurrent manual and semi-automatic functions

The simplicity of LL, which makes it so transparent, is also its greatest downfall. This is because, when developing complex control systems involving parallel tasks, which interact periodically, LL offers little in the way of structural constructs to deal with the complexity problem. In order to analyze and model the complexity problem more systematically, PN representation techniques are introduced and compared with LL programs. The main drawback with LL is that it is an implementation technique and not a design methodology, thus the interpretation of the requirement may not be correct from the first time. It is also challenging to design the controller program correctly from the first time using LL as compared to PNs. Criticism to LL and the IEC1131-3 was also for (a) difficulty to fit safely to a job (b) the lack of analytical ability (Jafari and Boucher, 1994; Nagao, 1993) and (c) its poor timing abilities (Halang, 1989). The IEC proposed the IEC 61499 as a successor of the IEC 61131 standard. This new design is object-oriented and uses UML to define a 4-layered architecture for designing such type of systems (Thramboulidis and Tranoris, 2001). This standard is not yet fully adopted by the industry.

Previous approaches for PLC programs can be categorized into two groups: 1 verification of a given PLC program and 2 generation of a dependable PLC program. In the first group, various software tools have been developed for the verification of PLC-based systems via the use of timed automata such as UPPAAL2k, KRONOS, Supremica and HyTech, mainly for programs written in a statement list language also termed Boolean. These software tools verify PLC programs to a certain extent; however, they remain limited. Since they are mainly focusing on the checking of theoretical attributes such as safety, liveness and reach ability, it is not easy for users to determine whether the PLC programs actually achieve the intended control objectives. In the second group many researchers have focused on the automatic generation of PLC programs from various formalisms including state diagrams Petri nets and IDEF0. These formalisms can help the design process of control logics, however, it is compared to LL which makes it difficult to find hidden errors, which are the most difficult part of the verification of a control program. To cope with the problem, one needs a more transparent PLC programming environment that helps the users to recognize hidden errors (Park *et al*., 2008).

The IEC PN models reside among the most widely used tools to model and evaluate the behavior of discrete complex industrial and dynamic systems. Petri net models help in understanding the interactions and relations of stochastic events, to visualize conflicts and problems with buffers andto detect deadlocks. They also provide quantitative analysis methods for resource utilization, consequences of system failures, system throughput rates andmore. Being a well-defined and a well-developed tool based on mathematical formulation, PN modeling is ideal as a method to implement real-time control for discrete industrial systems (Tzafestas *et al*., 2002). Boucher *et al*. (1989) a controller for a robot and numerically controlled lathe with the PN, LL loaders and compilers were developed, where the researchers compared the control performance of each of them and reported that the PN described the flow of the process more efficiently than the corresponding LL. In (Baker and Song, 1992) a Programmable Logic Controller Net (PLCNet) was proposed as a replacement for the LL. Ferrarini *et al*. (1994) a PN-based modular simulation for distributed and real-time control was presented. Venkatesh and Ilyas (1995) a Real-Time Petri Nets (RTPNs) was proposed. Jiang *et al*. (1996) a technique to translate PN into Sequential Function Charts (SFC) was proposed. In (Zhou and Twiss, 1998) a comparison was made between LL and PN-based control systems which revealed that PN-like diagram methods were better in terms of legibility and flexibility in the control design. Stenerson (1999) a stage-based programming was proposed as a substitute to LL which reduced programming and troubleshooting time. Yao *et al*. (2005) a mobile manufacturing system with Personal Digital Assistance (PDA) and PLC was developed using PN as a tool for modeling.

Several previous works were performed that studied different algorithms and various control methods that can be used for modeling and optimization within the industrial and non-industrial applications. For example, in (Al-Rabadi, 2004) important Boolean and multi-valued algorithms were developed that could be used within various modeling and optimization applications. Salem *et al*. (2010) the Field Programmable Gate Array (FPGA)-based System-On-Chip (SOC) for real-time power process control was studied. The development of a semi-industrial multi fruit dryer system using simultaneous intelligent control was studied in (Javanmard *et al*., 2010). The study in (Mukherjee *et al*., 2010) proposed a unique studied attempt to characterize and model Bheri (shallow flat bottom waste water fed fishery) as a complex biological system. The research in (Youssef and Peng, 2010) proposed a method that could deal with fault-tolerant control system by using the decentralized control theory. The study in (Chami *et al*., 2010) proposed a way that revealed the reason behind a haptic illusion called the Velvet Hand Illusion (VHI) by fem analysis. The research in (Alfred, 2009) presented the optimizing feature construction process for dynamic aggregation of relational attributes. The publication in (Alfred, 2010) summarized the relational data using semi-supervised genetic algorithm-based clustering techniques. The study in (Furukawa, 2010) presented an adaptable user interface which is based on the ecological interface design concept for multiple robots operating with uncertainty. The study in (Mohammed *et al*., 2010) aimed at presenting principal attacks classifications, especially the study of classification towards evaluation for which suggested some improvements that may allow the generation of test cases selection about attacks by using the classification tree method. The study in (Jaya and Thanushkodi, 2011) studied the implementation of computer-aided diagnosis system based on parallel approach of ant-based medical image segmentation. The results in (Carifio and Perla, 2010) studied the modeling and analyses for the decline and fall of radical and educational constructivism. The research in (Monprapussorn *et al*., 2009) presented a multi-criteria Decision Analysis (DA) and geographic information system framestudy for hazardous waste transport sustainability. The study in (Hu *et al*., 2009) presented a modified IPA for order-winner criteria improvement. The publication in (Liu and Du, 2009) presented Logical Time Interaction Petri Nets (LTIPN) that were designed to describe multimedia synchronization. Improving QoS calculation strategy of web service composition by a dynamic configuration method is presented in (Yang and Li, 2010). A behavior-aware trust reasoning methods that are based on Associate Petri Net (APN) are presented in (Liu and Yang, 2010). The research published in (Meng *et al*., 2011) presented cross-organization task coordination patterns and models of urban emergency response systems.

Petri nets: In general, a Petri net (Stenerson, 1999) is composed of: (1) circles containing token which are analogous to external states of the machine such as sensors, buttons or logical states such as production stages, (2) arcs (not ending with circles), (3) inhibitor arcs (ending with a circle) representing a compulsory relation between a state and a transition, (4) a transition that consists of three types of (a) thin line (plain), (b) unfilled rectangle (timed relation or counter) and (c) filled rectangle (operation-based transition) and (5) double arcs where self-loops are assumed as shown in Fig. 2.

Figure 3 shows an example of a PN model. In Fig. 3, {A, B, C, D, E, F, J} are places, {G, H, I} are untimed transitions, T is timed transition, where D will become on only if {A, B, C} have tokens, that is when they are on. Once D is on, {A, B, C} lose tokens and become off. If D becomes on, F becomes on directly, followed by E. On the other hand, J has a timed transition, that is, it becomes on and E loses its token after some delay time T.

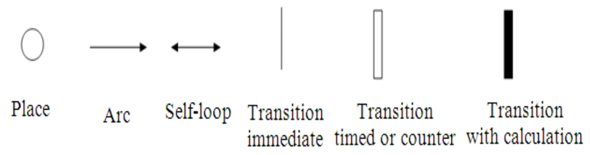


Fig. 2: The basic components of the Petri net representation

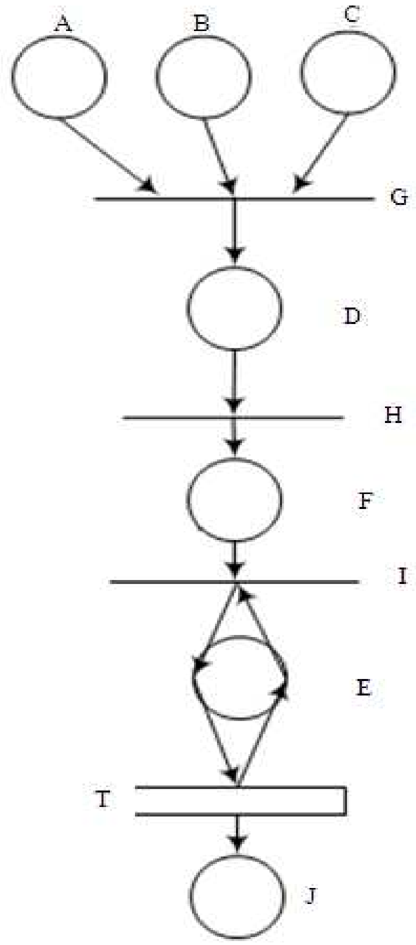


Fig. 3: An example of a Petri net model

MATERIALS AND METHODS

This section presents the case study of polymer raw-material mixing machine, including materials and methods, detailed flow chart and system stages, Petri net representation and the utilized ladder logic, the three system design possibilities for the interaction variables, control system implementation, the results and evaluation of the new system anddiscussion of the proposed implementation methods.

The interactions between semi-automatic and fully-automatic stages: In many cases, the PLC could not always be allowed a full access to machine control where, in some stages of the operation, the PLC can have only a monitoring role. This is a valid statement in machine development especially for older machines, where replacing the controllers presents higher cost when compared to using a new controller in the operation. Thus, in this case, the operating machine will have more than one controller with the possibility of interaction and conflict between them. This study suggests three solutions to handling this conflict. Furthermore, the reliability and costs of such solutions will be addressed in more details.

The suggestions in this research are only valid when the semi-automatic stages have clear beginning and end conditions and clear assignment to the PLC automatic system. Consequently, as shown in Fig. 4, the system operation can be classified into stages (i-1) and (i+1) which are semi-automatic and stage (i) which is an automatic intermediate stage. In this way, the interaction variable denoted in this study by "E" serves as a trigger for the transfer between the automatic and semi-automatic control operations.

However, in most cases there will be several options for the variable E, where the choice among these options will affect both of the reliability and the cost of the controller.

Three possible Decision Analysis (DA) choices will be suggested to achieve the objectives of this study, which are denoted as {DA1, DA2, DA3}, as shown in Fig. 5 where:

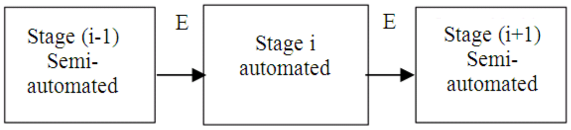


Fig. 4: An illustration for the interaction of the semi-automatic and automatic stages

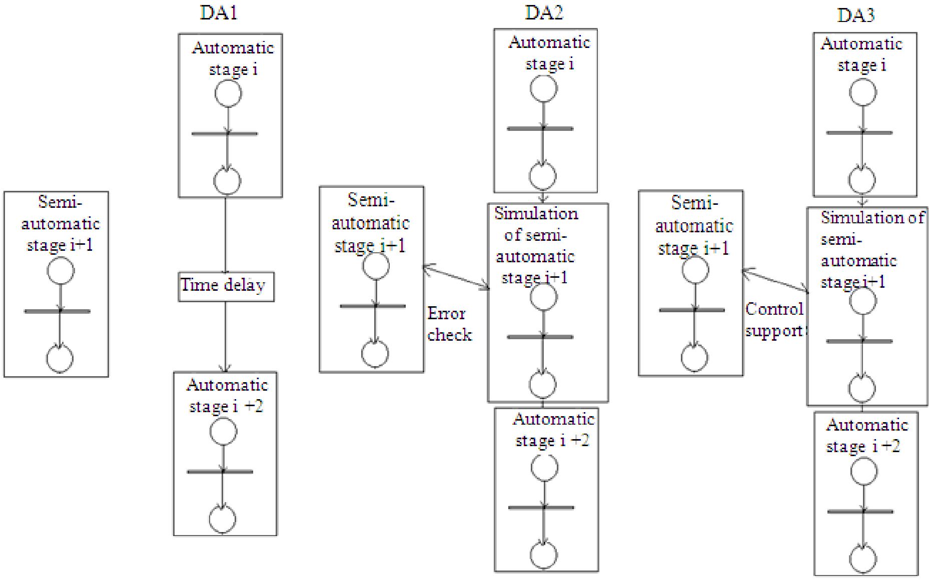


Fig. 5: A schematic illustration of the three choices of decision analysis of DA1 (i.e., ignoring the semi-automatic operation), DA2 (i.e., simulation and discrepancy check) andDA3 (i.e., simulation and control support)

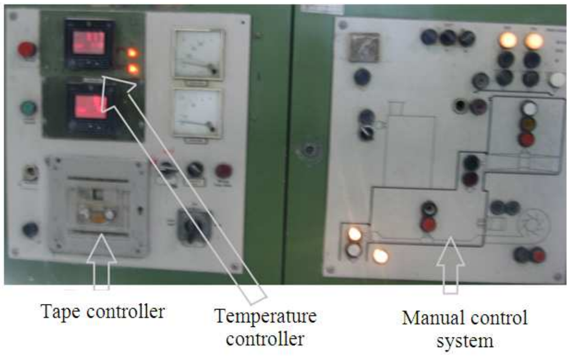


Fig. 6: Control cabinet for KN400

DA1: Ignoring the semi-automatic operation. In this case, the PLC ignores the semi-automatic operation completely and allows only time for the semi-automatic portions to finish. This case is applicable when the semi-automatic stages are trivial and need only a small amount of time.

DA2: Simulation and discrepancy check. The semi-automatic stages are programmed into the PLC ladder logic but without making any control orders. A discrepancy check control maybe added for the diagnosis check to check the semi-automatic operation. In case there is a non-trivial error between the simulation and the semi-automatic operation, an alarm is displayed to the operator. A non-trivial error could be of several forms such as (a) missing control signal, (b) wrong control signal and(c) wrong timing for a control signal.

DA3: Simulation and control support. In this part, PLC makes the simulation and checks that the semi-automatic part had made the orders and, in case of a discrepancy, the PLC issues the control signal instead.

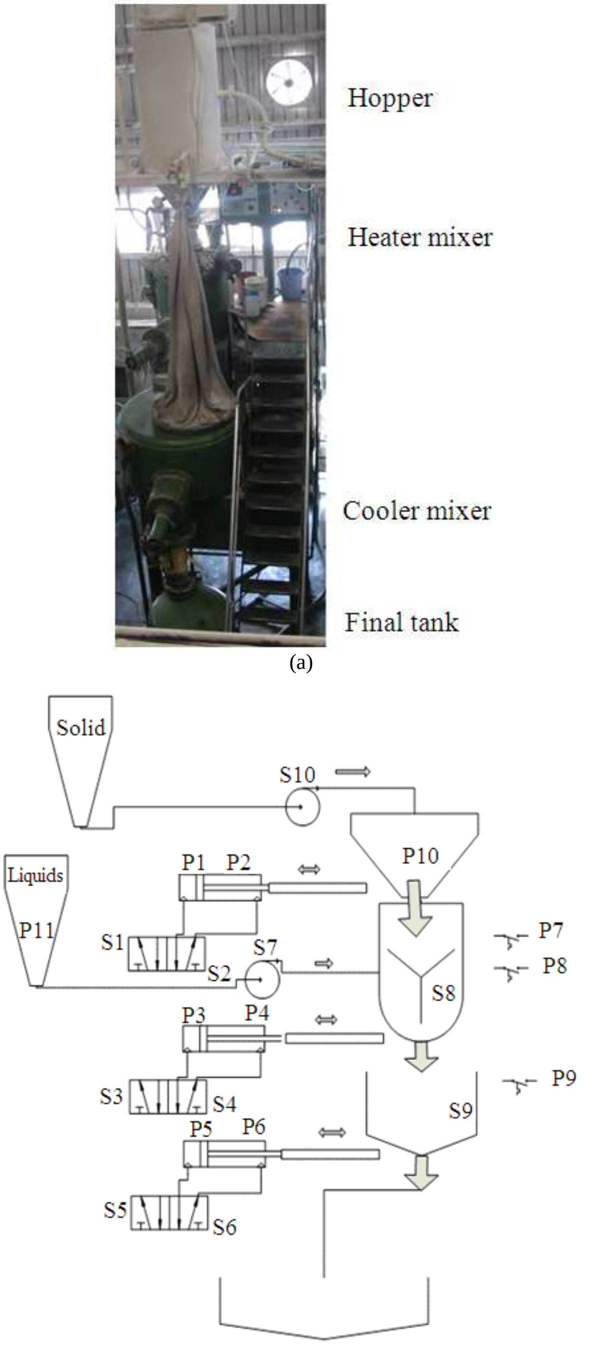
Case study: Polymer raw material mixing machine: The case study in this study is applied to the mixer control KN400 shown in Fig. 6 and the PVC raw-material mixing machine shown in Fig. 7a which is manufactured by Diosna (Dierks and Sohnen Machinenfabrik), Germany. The machine was initially operating as a PVC pre-mixer in CABELCO company in Zarqa, Jordan. Prior to this study, the machine was operating semi-automatically and the operators presence was compulsory. Originally, the machine was timely automated using track-based tape where each motor rotates a tape with hollow parts representing the time of operations for the various parts of the machine. Thus, the machine had a DA1-type automation system. Each part of the machine ran at a specific instance of time without sensing what is happening to the other parts leaving the semi-automatic parts of the machine to do the self-check. All functions of the machine were controlled electrically and electro-pneumatically. All control parts are accommodated in the switchboard cabinet except for the pneumatic cylinders, limit switches andtemperature probes which are fitted to the machine. The control cabinet, which is shown in Fig. 6

includes the manual control part, tape controller (which is replaced by the PLC system) andthe temperature controllers for the heater mixer and the cooler mixer. The temperature controller is the KS92 model that includes four switches (H1-H4) and a single probe fitted to the machine. Two such probes are used; one for the heater mixer and one for the cooler mixer.

The machine under discussion mixes PVC powder and fillers (including calcium carbonate) with a plasticizer and a plasticizer agent. The PVC powder is initially heated to 87°C prior to adding the liquid plasticizer agent then heated further to 118°C. The material is then cooled in the cooler mixer and then passed to the final tank.

A schematic of the mixing machine is shown in Fig. 7. The machine is composed of a hopper, a heater mixer anda cooler mixer, where these have a pneumatic piston connected to a steel plate that opens and closes the various containers. If this plate is moved, then the material is passed to the next stage. Table 1 shows the various sensors and controls for the machine.

Flow chart and system stages: In our case study, the basic mixing machine operations are as follows: Solid addition, initial heating, liquid addition, further heating and cooling, as shown in Fig. 8. The proposed automatic stages are designated by M1-M5 that are related directly to the aforementioned basic operations. Each of the basic operations is followed by a semi-automatic stage, subject to trigger conditions. For example, M2 is currently a manual stage to be automated which is related to solid addition by issuing an S2 command that retracts (i.e., opens) the first piston and solids are passed from the hopper to the heater mixer. The semi-automatic stage is triggered by sensor P1 (i.e., it is open and showing that the piston had finished retracting), a hardware timer is then activated followed by S1 command which causes the piston to extend and close the hopper. This hardware timer stage is designated as M2-T and it is an intermediate stage between the automatic stage M2 (solid addition) and M3 (liquid addition). M3 is triggered by the heater mixer reaching to a certain temperature and activates the S7 command which triggers the semi-automatic stage M3-T which is run by a hardware counter to pass some preset amount of material through a pulsed timer. The trigger condition for the following manual operation (that is to be automated) is reaching the temperature P8 sensor.



(a)

(b)

Fig. 7: A schematic diagram for the PVC raw-material mixing machine

The rest of the machine operation is performed in the same manner.

The semi-automatic operations are hardwired and denoted as Mi_T, where "i" stands for the subscript of the prior automatic stage and“T” stands for transfer, that is the material transfer from one location to another. There will be a hardware trigger for each semi-automatic stage and a hardware signal. However, the automatic stage Mi (which we design) does not always have a definite hardware trigger for starting and ending.

Petri net representation and the ladder logic program: Figure 9 shows the Petri net representation for the mixing machine. It is designed to be compatible with the flow chart shown in Fig. 8. The flow chart states of Mi or Mi_T are represented as places. M1 only requires the finish of the last stage and the automatic switch on. As for the second stage M2, it requires the solid and liquid material be available (sensors P10, P11) and the top most piston be in the extend situation (P2 on) in addition to the initial stage being true (M1 on).

Table 1: The control variables and sensors for the machine operation

Inputs/Sensors	Description
P1	Sensor for hopper open (passing solids to heater mixer)
P2	Sensor for hopper closed
P3	Sensor for heater mixer open (passing heater material to cooler mixer)
P4	Sensor for heater mixer closed
P5	Sensor for cooler mixer open (passing material to final tank)
P6	Sensor for cooler mixer closed
P7	Temperature sensor switch for pumping liquid material
P8	Temperature sensor switch for finishing heater mixer work and passing material to cooler mixer
P9	Temperature sensor switch for finishing cooler mixer work and passing material to final tank
P10	Proximity sensor for solid material available
P11	Proximity sensor for available liquid material
Outputs/Controls	Description
S1	Control signal for closing the hopper
S2	Control signal for opening the hopper to pass material to heater mixer
S3	Control signal for closing the heater mixer
S4	Control signal for opening the heater mixer to pass material to cooler mixer
S5	Control signal for closing the cooler mixer
S6	Control signal for opening the cooler mixer to pass material to final tank
S7	Control signal for pumping the liquid material
S8	Control signal for starting the heater mixer motor
S9	Control signal for starting the cooler mixer motor
S10	Control signal for pumping the solid material

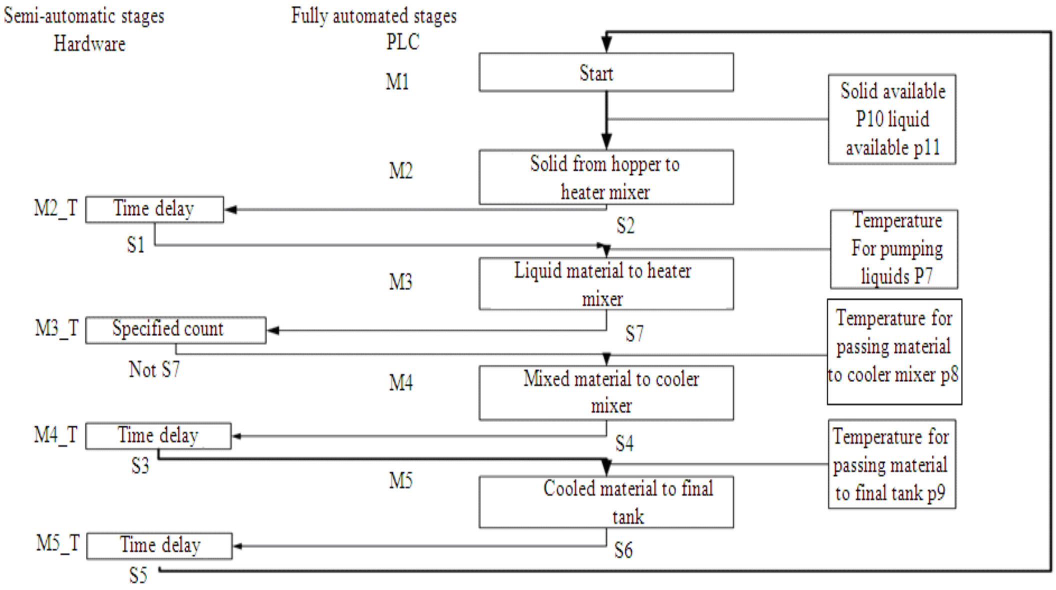


Fig. 8: A general flow chart representing different stages for the operation of the mixing machine

A semi-automatic stage follows which is represented as an independent portion of the system separated by a dashed line. The stage M2-T is accessed when the piston retracts (P1 on) and causes a timer to delay the extending of the piston to allow the material from the hopper to the heater mixer. Within the time delay studying of the semi-automatic stage, two automatic operations must be achieved which represents a dilemma, that is, the end of stage M2 and the beginning of stage M3. The trigger action is being investigated and affects both the reliability and the cost of the automation system. The same can be said in relation to the other automatic and semi-automatic stages.

The ladder logic program in Fig. 10 is compatible with the Petri net representation. For example, the conditions of stage M2 can be given by Eq. 2. Thus, it is represented as a series of normally open (make contact) and normally closed (break contact). The same can be said about the other states. The representation of the semi-automatic stage is for simulation purpose or control support purpose. State M2-T is triggered when piston1 is in retract situation (P1 on) and starts an On-delay timer which causes a delayed signal to extend the cylinder back (S1 on). State M2-T is ended when the piston is extended (P2 on). This is totally compatible with the Petri net representation.

The basic equations, for the operation within our case study of the mixing machine, are as follows:

M1= Auto And E7 (1)

M2 = S2 = ((P2 And P10 And P11 And M1) Or M2) And (Not E1) (2)

M2-T = (P1 Or M2-T) And (Not P2) (3)

M3 = S7 = ((P7 And E2) Or M3) And (Not E3) (4)

M3-T = (S7 Or M3-T) And (Not P8) (5)

M4 = S4 = ((P4 And P8) Or M4) And (Not E4) (6)

M4-T = (P3 Or M4-T) And (Not P4) (7)

M5 = S6 = ((P9 And P6 And E5) Or M5) And (Not E6) (8)

M5-T = (P5 Or M5-T) And (Not P6) (9)

where {(Not P2), (Not P4), (Not P6), (Not P8)} are implied connections in the semi-automatic operations in Fig. 9 and{S1, S3, S5} are implied and issued by the semi-automatic operations.

The three system design possibilities for the interaction variables: The interaction variables E1-E7 represent the decision analysis perspectives for our case study system. These variables indicate the interface choice between the semi-automatic stages and the automatic stages. It is compulsory that, within the semi-automatic stages, the pre-automatic stages must finish and the post-automatic stages must be started. For example, let's take stages M2 and M3. In one case, M2 can be assumed not to finish within the semi-automatic stage M2-T and it is only ended when P7 is triggered, that is when M3 is initiated. In this case, no extra internal memory is needed andtime is allowed for the semi-automatic stage to finish with no action taken within the semi-automatic stages (that is totally ignoring the semi-automatic stages). Another choice is to simulate the semi-automatic stage operation and define M2-T using the ladder logic. In this type of choice, it is possible to check whether the cylinder is extended back in the proper time, if not then an alarm signal is issued. The third choice type is to simulate the working of the semi-automatic stages and in case the semi-automatic stages failed, proper signals are issued to support key aspects of the semi-automatic stages. This last choice will require more internal memory relay in addition to more output and input ports. The choices for the interaction variables E, according to the three DA choices are as follows:

- DA1 (Totally ignoring): E1 = M3, E2 = M2, E3 = M4, E4 = M5, E5 = M4, E6 = M1 andE7 = M5
- DA2 (Simulation): E1 = M2-T, E2 = M2-T, E3 = M3-T, E4 = M4-T, E5 = M4-T, E6 = M5-T andE7 = M5-T. Yet, in this case, signals {S1, Not S7, S3, S5} are not the actual output signals
- DA3 (Control support): E1 = P1, E2 = P2, E3 = M3-T, E4 = P3, E5 = P4, E6 = P5 andE7 = P6. Yet, in this case, signals {S1, Not S7, S3, S5} are the actual output signals

The three upper choices represent the different choices available for the designer when automating semi-automatic machines. The first choice ignores the hardware through allowing time to the finishing of the semi-automatic part. The second choice uses the PLC to simulate the process where any discrepancy should be handled as an error. The third choice makes excessive use of the inputs and outputs.

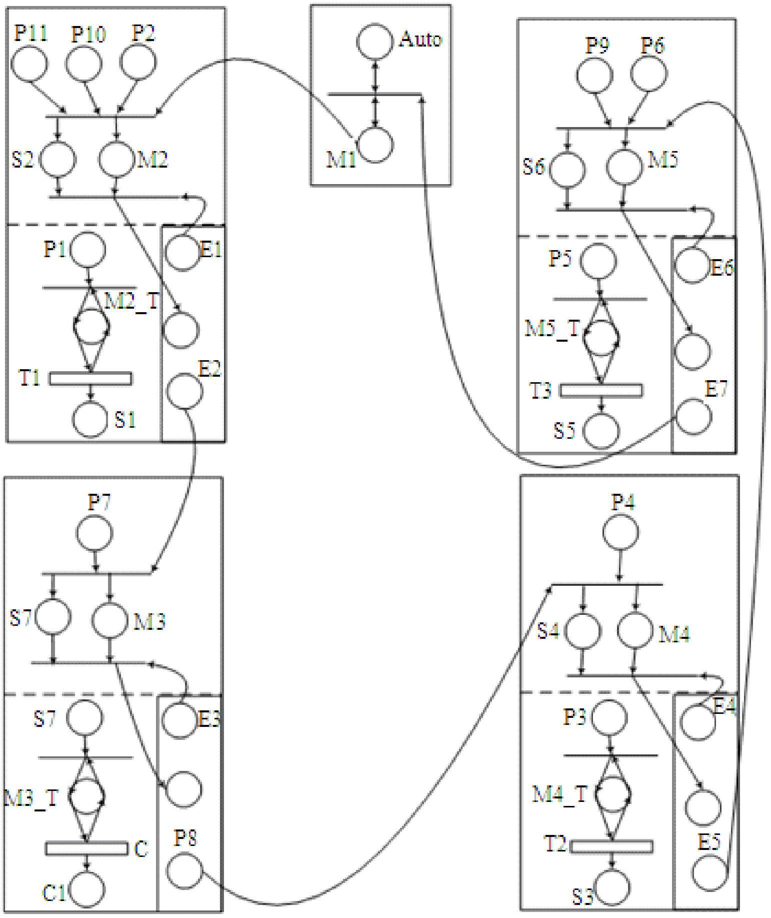


Fig. 9: Petri net representation for the total operation (semi-automatic and fully-automatic) of the mixing machine

There are three main objectives for choosing between the three decisions {DA1, DA2, DA3}. These are: (1) use of resources (memory, inputs, timers), (2) reliability of the design and (3) failure modes. The first design uses the least resources but it completely ignores the studying of the semi-automatic part. However, if one part of the semi-automatic system failed, such as if the door of the heater mixer did not close back, the system will drop solid material that will not heat and will drop directly into the cooler mixer. The second choice of the simulation uses more memory positions, but no extra inputs and outputs. It can give more insight into the behavior of the system to the operator and any discrepancy can be detected by the operator. As for the last choice, it uses the largest of memory resources as it keeps track of the system and issues control signals. On one hand this choice is the most reliable, however, it increases the cost of the design and the hardware appreciably as it requires more memory, inputs, timers and counters. As for the current case study, the cost was of most importance and the presence of the operator for monitoring was mandatory. In case of any problem in the line, the operator will shut down the line and make a maintenance order, thus DA1 was chosen.

Control system implementation: Figure 11 shows the basic equipment used for the PLC control system implementation. The sensors shown in Table 1 are used to switch a contactor which passes an on signal (220 Volt), which is connected to the common port (Com) to the normally open port (NO) and then to the input of the used PLC. The designed ladder logic program in Fig. 10 which was implemented using LOGO!Soft Comfort v6.0.4 Siemens software tool, is then downloaded to the PLC and responds by switching on and off the pneumatic valves and contactors to control the pneumatic cylinders. The used PLC in this study is the Siemens LOGO PLC OBA0 series as shown in Fig. 11.

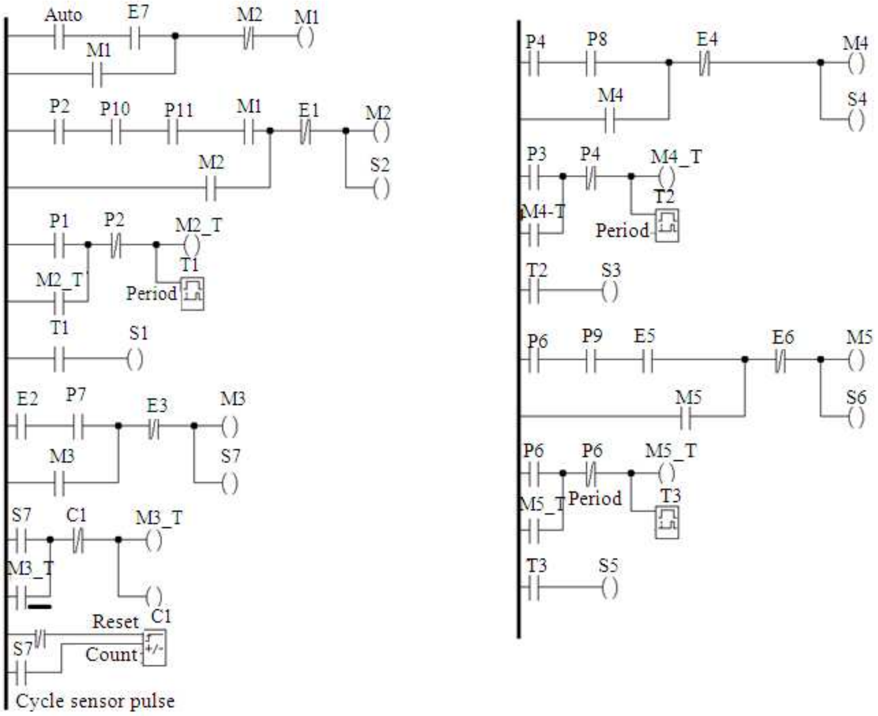


Fig. 10: A schematic of the ladder logic control program for the mixing machine operation

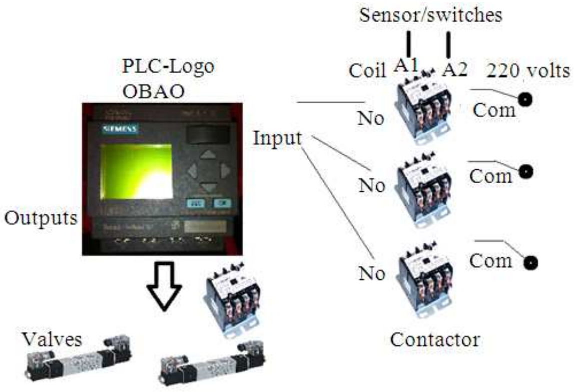


Fig. 11: The basic configuration for the proposed PLC control system

RESULTS

A comparison between the manual and the automatic operation in a state-time diagram is shown in Fig. 12 where this Figure represents a sample of the whole picture starting from state M2. Currently, the operator must start the semi-automatic stage and start the following automatic stage. On average, the operator will have a delay period D due to the need for the operator detection of the end of the stage.

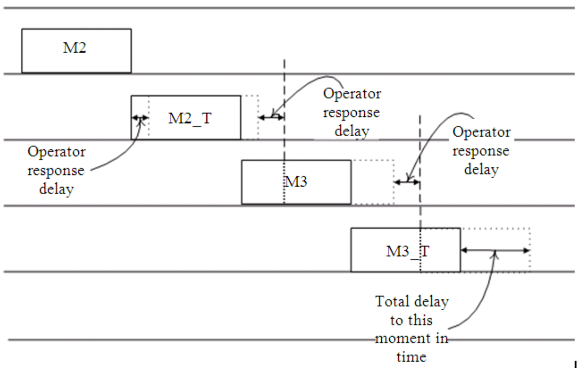


Fig. 12: State-time diagram for the manual part (the dotted line) and the automatic part (the continuous line)

It should be considered here that the operator in general andespecially in this case study, may be assigned to other work duties in addition to running the machine. The total delay will be cumulative as shown in Fig. 12, where the total operator response delay (T_D) is computed as:

$$T_D = \sum_{i=1}^{N} D_i \tag{10}$$

Where:

N = Number of operator actions (= 9) for this case study
D_i = i^{th} operator response delay assumed on average to be 0.1 min

Then, the total operator response delay is 0.9. The average cycle time (T_c) is nearly 15 min. The cycle time is highly variable; it is much higher in winter as the raw material will be cold as compared to hot summer time where the raw material temperature will be high.

The lack of efficiency, ς, due to the operator response could be given as (where $T_C > T_D$):

$$\varsigma = \frac{T_c - T_D}{T_c} * 100\% \qquad (11)$$

from which the lack of efficiency is equal to 0.6. Thus implementing the automatic control system will increase the efficiency by 6% on average. But, this amount will be less in winter as the cycle time increases and will be more in summer when the cycle time is lower.

The cost which is reduced due to the automation process can be considered as an important evaluation scheme. The cost reduction is based on two parts: (a) increasing productivity which is implicating some profit due to this increased productivity in addition to cost reduction due to the reduced energy and other costs because the machine is operating for lesser periods of time to produce the same amount of product and (b) reducing the number of operators needed by one, where the factory has three time shifts so this number is multiplied by three. Equ. 12 represents this profit increase, due to the full automation, as follows:

$$\text{Increased Profit} = 3 \cdot YS + p \cdot (\varsigma \cdot Q) \qquad (12)$$

Where:

YS = Average yearly salary for an operator
p = Profit/unit produced
ς = Lack of efficiency
Q = Quantity produced yearly

DISCUSSION

The automation method that was introduced in this study is applicable to a large number of machines within industrial and mechanical systems that were built using old control systems as it was previously common to heavily rely on semi-automatic operations. These types of machines where extremely common in the 1970’s and 1980’s and even through early 1990’s as the cost of automatic control systems was relatively high. Currently, automatic control systems had shown appreciable reduction in cost which made the change to fully-automated systems very feasible. In the majority of cases, semi-automatic machines tend to be separate from the manual operations, which need to be automated and this complies with the constraints and assumptions that are dealt with in this study.

As with any investment, there are several options for the decision makers which determine the quality and the cost of the project, where decision analysis {DA1, DA2, DA3} show the basic choices for this type of application. This study shows the detailed transformation for a process, using PN modeling and LL programming for the used PLC control, for the change to full-automation which reduces the cost and improves the efficiency.

CONCLUSION

When developing and using older machines and systems, the full automation of a semi-automatic machine, where the semi-automatic functionalities and automatic functionalities run separately in a sequence, is an important challenge. In this regards, Petri Nets (PNs) can successfully represent the operations of both the semi-automated functionalities and the fully-automated functionalities. In this study, a PN model and the corresponding ladder logic programming, as a means to program Programmable Logic Controllers (PLCs), were developed to show three controller possibilities, where the following conclusions were observed:

- The Petri Net-based controller can be successfully used for representing both of the semi-automatic and fully-automated stages.
- The first role of the PLC can be to run the automatic part while ignoring the semi-automatic part, where this case is the least reliable. The second role of the PLC can be to simulate the semi-automatic part (uses more timers, counters, functional blocks and memory). The third role of the PLC can be to simulate and sense the semi-automatic part output (using more internal resources plus more input and output ports) and to issue control signals if needed.
- The decision in relation to the controller depends on three main objectives: (a) cost, (b) reliability and(c) possible faults. For the current case study, the cost was the most important objective, thus the first decision analysis (DA1), of ignoring the semi-automatic operation was chosen.

Future study will include the following items: (1) the investigation of using the introduced methods to other types of complex dynamic systems; (2) more utilization of the mathematical matrix-based PN formulations for the purpose of representing and controlling the operations of complex machines and systems; and (3) the investigation of the development of colored Petri Nets (CPNs), that offer hierarchical system descriptions and can explicitly represent both of the system states and actions, for the representation and control of complex machines and dynamic systems.

REFERENCES

Alfred, R., 2009. Optimizing feature construction process for dynamic aggregation of relational attributes. J. Comput. Sci., 5: 864-877. DOI: 10.3844/jcssp.2009.864.877

Alfred, R., 2010. Summarizing relational data using semi-supervised genetic algorithm-based clustering techniques. J. Comput. Sci., 6: 775-784. DOI:10.3844/jcssp.2010.775.784

Al-Rabadi, A.N., 2004. Reversible Logic Synthesis. 1st Edn., Springer, New York, ISBN: 3540009353, PP: 427

Institution of Electrical Engineers, 1992. Colloquium on Discrete Event Dynamic Systems-A New Generation of Modeling, Simulation and Control Applications. 1st Edn., Institution of Electrical Engineer, Lodnon, pp.

Boucher, T.O., M.A. Jafari and G.A. Meredith, 1989. Petri net control of an automated manufacturing cell. Comput. Indus. Eng., 17: 459-463. DOI:10.1016/0360-8352(89)90105-8

Carifio, J. and R.J. Perla, 2010. Towards the decline and fall of radical and educational constructivism (Mark I). Cur. Res. Psy., 1: 1-15. DOI: 10.3844/crpsp.2010.1.15

Chami, A., M. Ohka, Y. Kawabe and H.B. Yussof, 2010. Response of SAI afferents may play a role in the perception of velvet hand illusion. J. Comput. Sci., 6: 934-939. DOI:10.3844/jcssp.2010.934.939

Estévez, E., M. Marcos and D. Orive, 2007. Automatic generation of PLC automation projects from component-based models. Int. J. Adv. Manuf. Tech., 35: 527-540. DOI: 10.1007/s00170-007-1127-4

Ferrarini, L., M. Narduzzi and M. Tassan-Solet, 1994. A new approach to modular liveness analysis conceived for large logic controllers' design. IEEE Trans. Robotics Automation, 10: 169-184. DOI: 10.1109/70.282542

Furukawa, H., 2010. Adaptable user interface based on the ecological interface design concept for multiple robots operating works with uncertainty. J. Comput. Sci., 6: 904-911. DOI:10.3844/jcssp.2010.904.911

Halang, W.A., 1989. Languages and tools for the graphical and textual system independent programming of programmable logic controllers. Microprocess. Microprogramm., 27: 583-590. DOI: 10.1016/0165-6074(89)90115-4

Hu, H.Y., S.I. Chiu and T.M. Yen, 2009. Modified IPA for order-winner criteria improvement: A MICMAC approach. J. Applied Sci., 9: 3792-3803. DOI: 10.3923/jas.2009.3792.3803

Jafari, M.A. and T.O. Boucher, 1994. A rule-based system for generating a ladder logic control program from a high-level systems model. J. Intell. Manufact., 5: 103-120. DOI: 10.1007/BF00572403

Javanmard, M., J. Endan, K.A. Abbas and F. Arvin, 2010. Development of a semi-industrial multi fruit dryer system using simultaneous intelligent control. Am. J. Applied Sci., 7: 160-166. DOI: DOI: 10.3844/ajassp.2010.160.166

Jaya, J. and K. Thanushkodi, 2011. Implementation of computer aided diagnosis system based on parallel approach of ant based medical image segmentation. J. Comput. Sci., 7: 291-297. DOI: DOI: 10.3844/jcssp.2011.291.297

Jiang, J., D. Azzopardi, J. Holding, G.F. Carpenter and J.S. Sagoo, 1996. Real-time synchronisation of multiaxis high-speed machines, from SFC specification to Petri net verification. Proc. IEE Control Theory Appl., 143: 164-170.DOI: 10.1049/ip-cta:19960266

Liu, W. and Y. Du, 2009. Modeling multimedia synchronization using petri nets. Inform. Technol. J., 8: 1054-1058. DOI: 10.3923/itj.2009.1054.1058 http://www.doaj.org/doaj?func=abstract&id=601213

Liu, X. and L. Yang, 2010. Behavior-aware trust reasoning based on associate Petri net. Inform. Technol. J., 9: 1178-1183. DOI: 10.3923/itj.2010.1178.1183 http://docsdrive.com/pdfs/ansinet/itj/2010/1178-1183.pdf

Meng, D., Q. Zeng, F. Lu, J. Sun and J. An, 2011. Cross-organization task coordination patterns of urban emergency response systems. Inform. Technol. J., 10: 367-375. DOI: 10.3923/itj.2011.367.375 http://docsdrive.com/pdfs/ansinet/itj/2011/367-375.pdf

Mohammed, M., T. Bouchentouf, A. Benazzi and M. Azizi, 2010. Amelioration of attack classifications for evaluating and testing intrusion detection system. J. Comput. Sci., 6: 716-722. DOI: 10.3844/jcssp.2010.716.722

Monprapussorn, S., D. Thaitakoo, D.J. Watts and R. Banomyong, 2009. Multi criteria decision analysis and geographic information system framework for hazardous waste transport sustainability. J. Applied Sci., 9: 268-277. DOI: 10.3923/jas.2009.268.277 http://www.doaj.org/doaj?func=abstract&id=604581

Mukherjee, I., P. Bhaumik, M. Mishra, A.R. Thakur and S.R. Chaudhuri, 2010. Bheri-A unique example of biological complex system. OnLine J. Biol. Sci., 10: 1-10.

Nagao, Y., H. Urabe, S. Nakano and S. Kumagai, 1993. Application of Petri nets to sequence control. IEICE Trans. Fund. Elec. Commun. Comput. Sci., 76: 1598-1606. http://search.ieice.org/bin/summary.php?id=e76-a_10_1598

Park, S.C., C.M. Park and G.N. Wang, 2008. A PLC programming environment based on a virtual plant. Int. J. Adv. Manuf. Tech., 39: 1262-1270. DOI: 10.1007/s00170-007-1306-3

Pires, J.N. and J.M.G.S. da Costa, 2000. Object-oriented and distributed approach for programming robotic manufacturing cells. Robotics Comput. Integ. Manuf. J., 16: 29-42. DOI: 10.1016/S0736-5845(99)00039-3

Pollard, J.R., 1994. Ladder logic remains the PLC language of choice. Ctrl. Eng., 4: 77-79. http://md1.csa.com/partners/viewrecord.php?requester=gs&collection=TRD&recid=0158966CI

Salem, A.K.B., S.B. Othman and S.B. Saoud, 2010. Field programmable gate array -based system-on-chip for real-time power process control. Am. J. Applied Sci., 7: 127-139. DOI: 10.3844/ajassp.2010.127.139

Stenerson, J., 1999. Fundamentals of Programmable Logic Controllers, Sensors and Communications. 2nd Edn., Prentice-Hall, Upper Saddle River, ISBN: 0137461240, pp: 562.

Thramboulidis, K. and C. Tranoris, 2001. A function block based approach for development of distributed IPMCS applications. Proceedings of the 10th IEEE International. Conference on Advanced Robotics, Aug. 22-25, Budapest, Hungary, pp: 1-1. http://seg.ee.upatras.gr/thrambo/dev/PaperAbstracts/ICAR-01.pdf

Tzafestas, S.G., M.G. Pantelelis and D.L. Kostis, 2002. Design and implementation of a logic controller using Petri nets and ladder logic diagrams. SAMS, 44: 135-167. DOI: 10.1080/02329290212164

Venkatesh, K. and M. Ilyas, 1995. Real-time Petri Nets for modeling, controlling, and simulation of local area networks in flexible manufacturing systems. Comput. Indus. Eng., 28: 147-162. DOI:10.1016/0360-8352(94)00034-K

Yang, H. and Z. Li, 2010. Improving QoS of web service composition by dynamic configuration. Inform. Technol. J., 9: 422-429. DOI: 10.3923/itj.2010.422.429 http://www.doaj.org/doaj?func=abstract&id=601302

Yao, A.W.L., S.D. Strombeck and J.S.C. Chi, 2005. Development of a mobile manufacturing system with PDA and PLC. Int. J. Adv. Manuf. Tech., 25: 723-729. DOI: 10.1007/s00170-003-1914-5

Youssef, R. and H. Peng, 2010. Piecewise sliding mode decoupling fault tolerant control system. Am. J. Applied Sci., 7: 102-109. DOI:10.3844/ajassp.2010.102.109

Zhou, M.C. and E. Twiss, 1998. Design of industrial automated systems via relay ladder logic programming and petri nets. IEEE Trans. Syst., Man Cybernetics, 28: 137-150.DOI: 10.1109/5326.661096

Zurawski, R. and M. Zhou, 1994. Petri nets and industrial applications: A tutorial. IEEE Trans. Indus. Elect., 41: 567-583. DOI: 10.1109/41.334574