Emergence, Complexity and Computation ECC

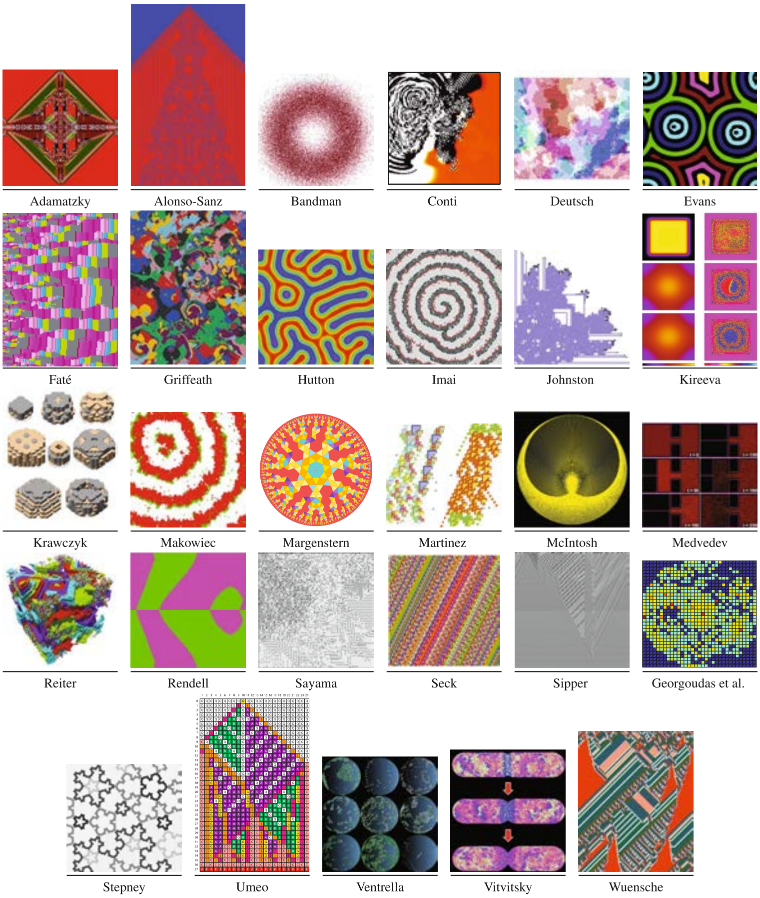
Andrew Adamatzky
Genaro J. Martinez
Editors

Designing Beauty: The Art of Cellular Automata

Springer

generated by the rules. Many of the cellular automata art works have been shown at major art exhibitions, installations and performances; others are newly born and awaiting for their fame to come.

The book offers in-depth insights and first-hand working experiences into production of art works, using simple computational models with rich morphological behaviour, at the edge of mathematics, computer science, physics and biology. We believe the works presented will inspire artists to take on cellular automata as their creative tool and will persuade scientists to convert products of their research into the artistic presentations attractive to general public.



Andrew Adamatzky, Bristol
Genaro Martinez, Mexico City
September, 2015

Reversibility, Simulation and Dynamical Behaviour

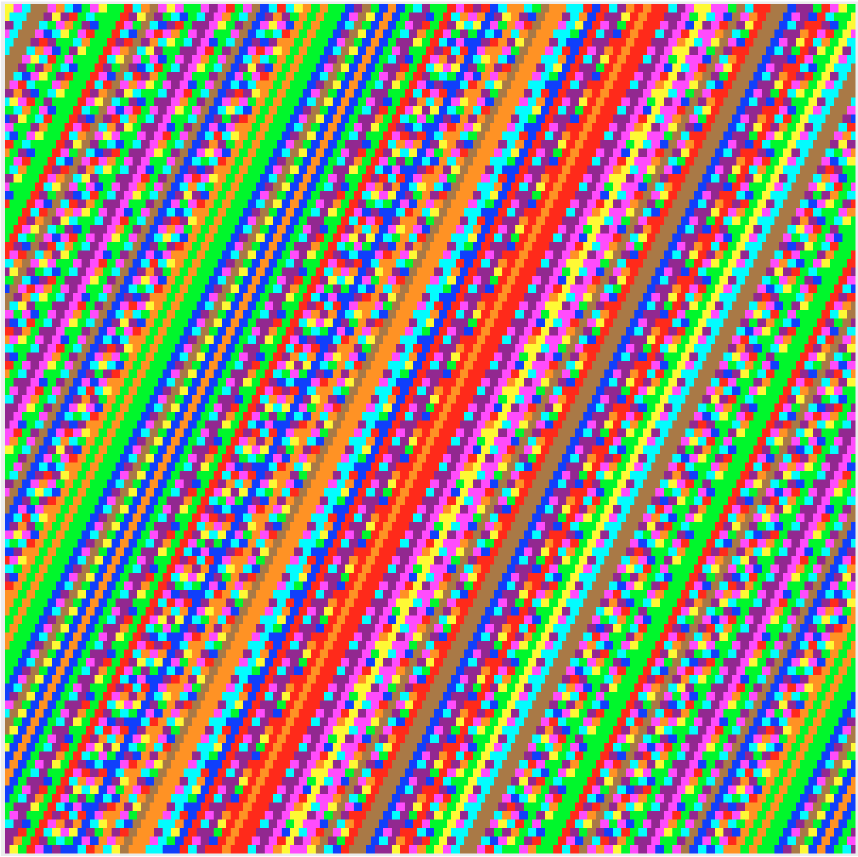
Juan Carlos Seck Tuoh Mora, Norberto Hernandez Romero,
and Joselito Medina Marin

A cellular automaton (CA) is reversible if it repeats its configuration in a cycle. Reversible one-dimensional CA are studied as automorphisms of the shift dynamical system, and analyses using graph-theoretical approaches and with block permutations. Reversible CA are dynamical systems which conserve their initial information. This is why they pose a particular interest in mathematics, coding and cryptography.

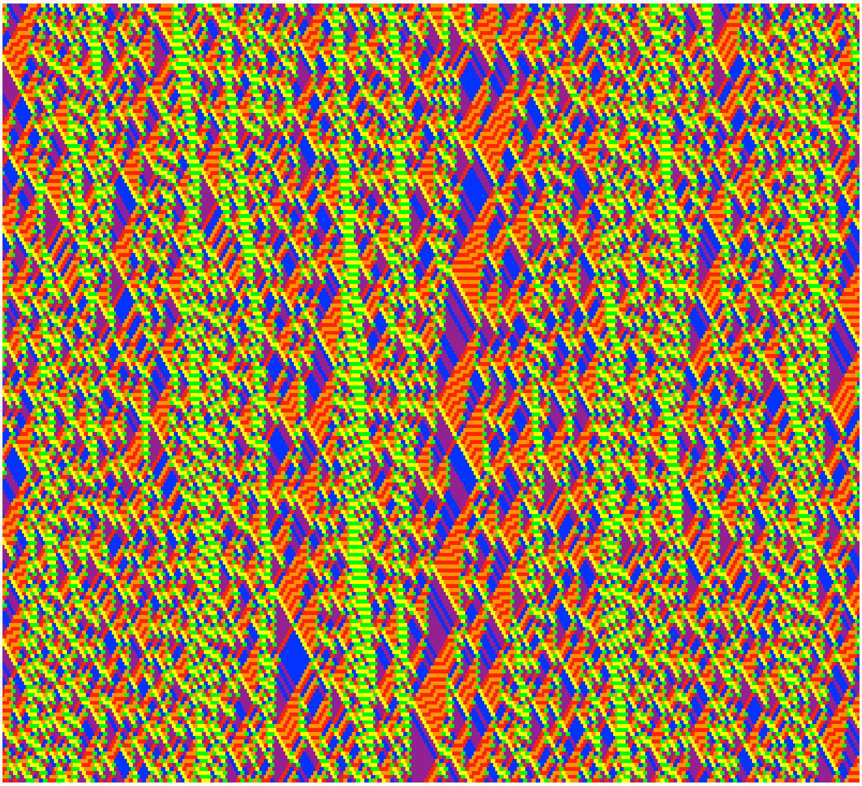
Rule 110 CA are Turing-complete. The Rule 110 is produces morphologically rich patterns composed of a periodic background (ether), on which a finite set of periodic structures (gliders) travel. The gliders collide and annihilate, produce new gliders or stationary localisations in result of the collisions. Rule 110 is a wonderful example of an apparently simple system with complex behaviour. This rule has been analysed using regular expressions, de Bruijn diagrams, and tilings. In particular, a block substitution system with three symbols is able to simulate the behaviour of Rule 110. The dynamics of Rule 110 can be reproduced by a set of production rules applied to blocks of symbols representing sequences of states of the same size, and the shape of the current blocks is useful for predicting the number of blocks in the next step. Another classic problem in CA is the specification of numerical tools to represent and study their dynamical behaviour. Mean field theory and basins of attraction have been commonly used; however, although the mean field theory gives the long-term estimates of density, it does not always give the adequate approximation for the step-by-step temporal behaviour. We present images related to the specification of reversible cellular automata by amalgamations of states and using memory. These are examples describing the dynamics of elementary CA by surface interpolation and the simulation of Rule 110 using a block substitution system.

J.C.S.T. Mora · N.H. Romero · J.M. Marin
Area Academica de Ingenieria, Universidad Autonoma del Estado de Hidalgo,
Carr Pachuca-Tulancingo Km 4.5, Col. Carboneras, Pachuca Hidalgo 42184, Mexico
e-mail: {jseck,nhromero,jmedina}@uaeh.edu.mx

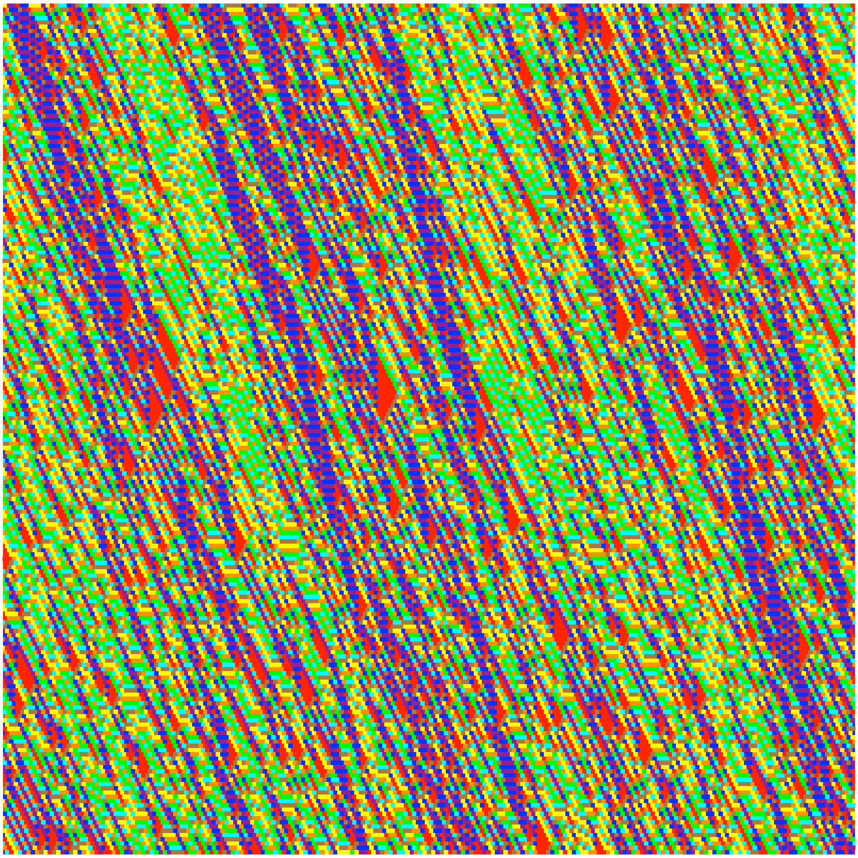
A. Adamatzky and G.J. Martínez (eds.), *Art of Cellular Automata*,
Emergence, Complexity and Computation 20,
DOI: 10.1007/978-3-319-27270-2_22



Reversible one-dimensional CA of 9 **states with neighbourhood size** 2 **and Welch indices** $L = 9$ **and** $R = 1$**.**
Reversible CA conserve the information presented in their initial configuration. This is because these automata perform block permutations and shifts in every step. In one dimension, a reversible CA of s states is equivalent to a full shift of s symbols. Symbolic dynamics operations can be applied over reversible automata in order to obtain and analyse different kinds of behaviours. In the case of a Welch index 1, the evolution rule of a reversible CA can be randomly defined by amalgamations of adequate permutations of states, obtaining an invertible dynamics as the one presented in the figure, characterised by cyclic patterns protected in quiescent barriers [145].

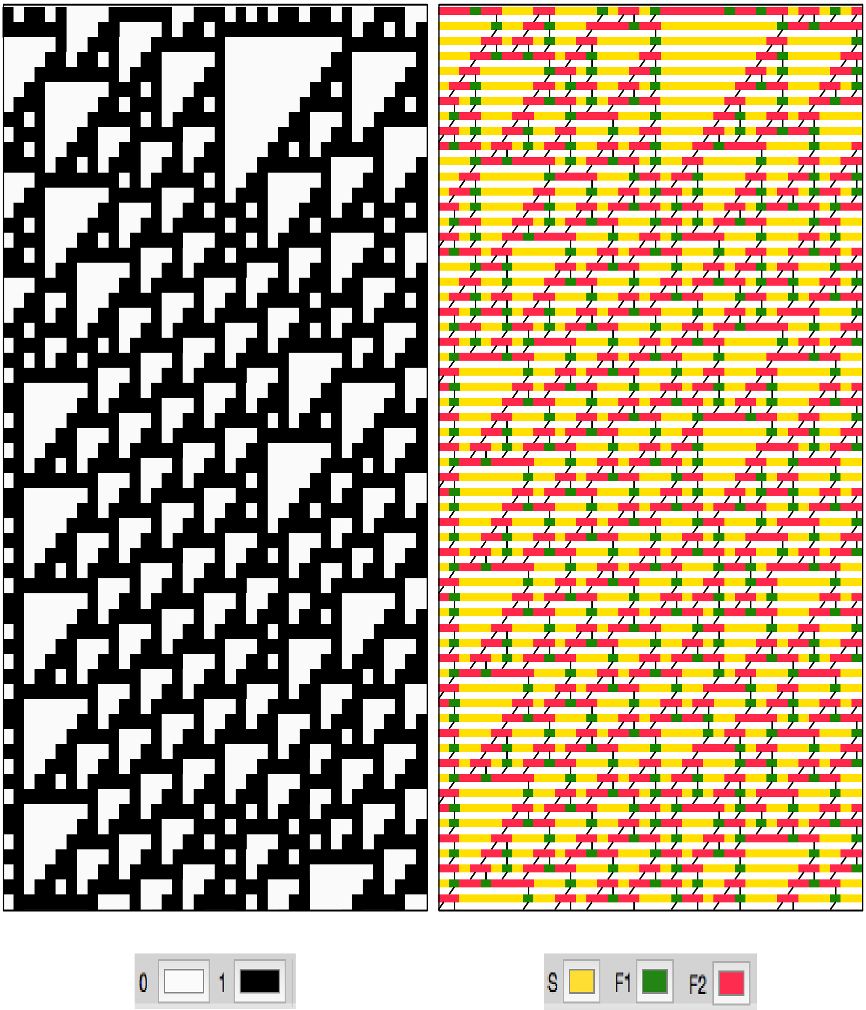


Reversible one-dimensional CA of 6 **states with neighbourhood size** 2 **in both invertible rules and Welch indices** $L = 2$ **and** $R = 3$**.** ©2014 Juan Carlos Seck-Tuoh-Mora.
Reversible CA can be defined with a unitary Welch index, or both Welch indices different from 1. In any case, the Welch indices product is equal to the number of de Bruijn blocks, or blocks with s^{2r} states; where s is the number of states and r is the neighborhood radius. For reversible automata $r = 1/2$ or neighbourhood size 2, the Welch indices product must be equal to the number of states. Even when both invertible rules are defined with $r = 1/2$, there are reversible automata able to generate interesting evolution patters, as the one depicted in the figure with 200 cells in 200 evolutions [144].

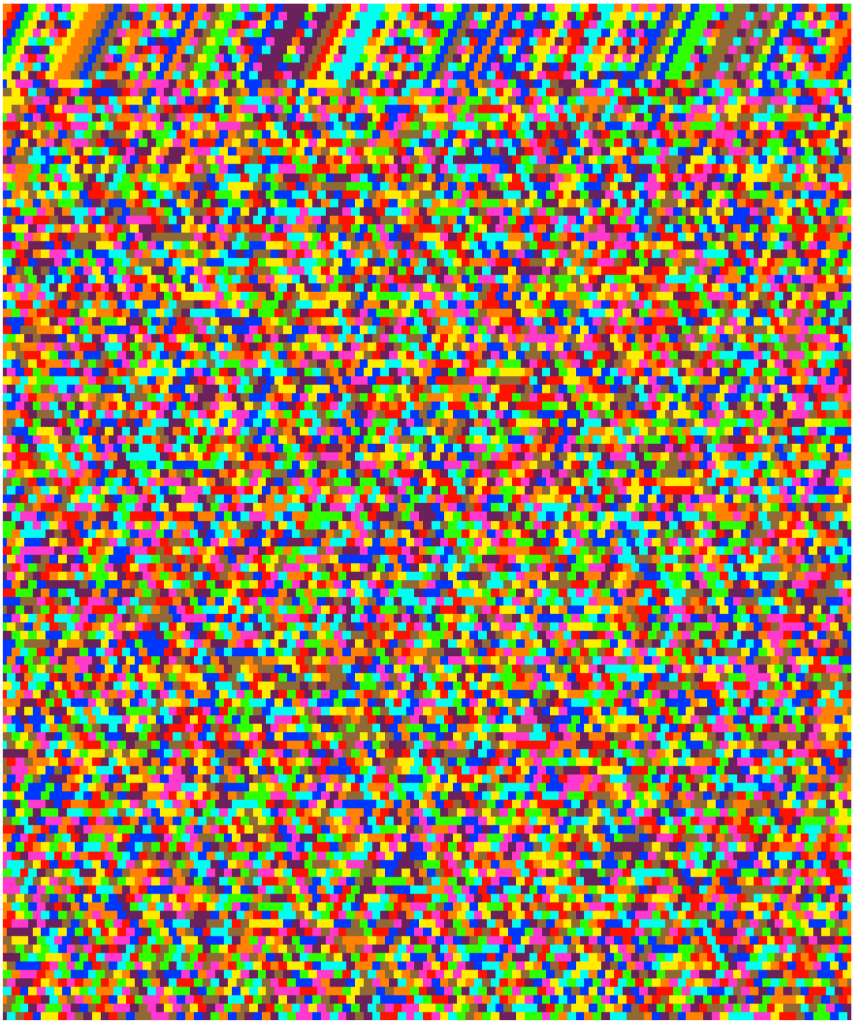


Reversible one-dimensional CA of 8 **states with neighbourhood size** 2 **in both invertible rules and Welch indices** $L = 2$ **and** $R = 4$**.** ©2014 Juan Carlos Seck-Tuoh-Mora.

Reversible CA with more states are able to produce more complicated behaviours, even when a neighbourhood size 2 is defining both invertible rules. The characterisation of reversible cellular automata by means of block permutations using the classical Cantor topology provides a basis for analysing its dynamical behaviour. With this, we obtain useful information about their transitive behaviour, in order to know if the automaton is topologically transitive or mixing [146].

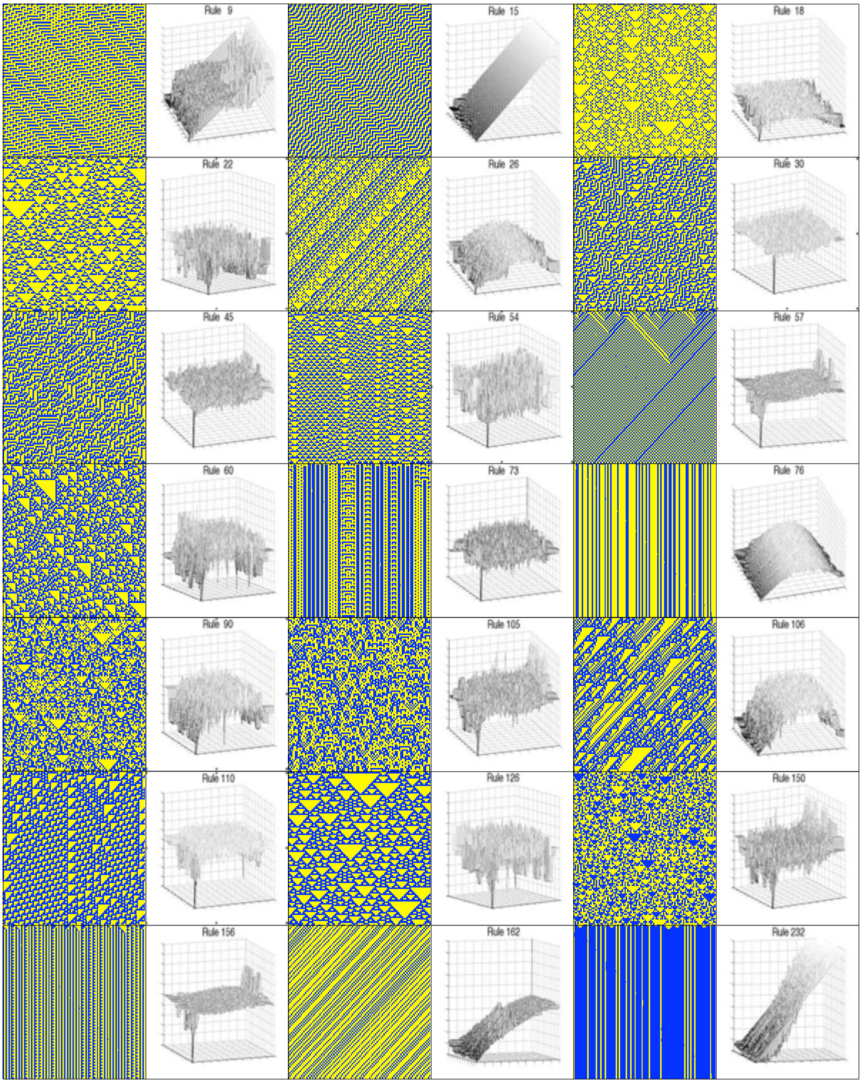


Rule 110 **represented as a block substitution system.** ©2014 Juan Carlos Seck-Tuoh-Mora. Rule 110 can be represented as a block substitution system of three symbols: F_2 representing the sequence 01, F_1 for state 1 iff it is on the left of 0 or F_2, and S_m for sequences of length m of the same state between F_α, for $\alpha \in \{1,2\}$. With these symbols, the production rules reproducing Rule 110 dynamics are: $S_mF_2 \to S_{m-1}F_2F_1$, $S_mF_1 \to S_{m-1}F_2$ and $S_mF_{\alpha_1}F_{\alpha_2}\ldots F_{\alpha_p} \to S_{m-1}F_2S_{q-2}F_1$ where $q = \sum \alpha_i$. The simulation is given by reordering blocks in every step when symbols F_α are concatenated. In some cases, the substitution system conserves the information of the original sequence, although the number of blocks may vary in this process, illustrating the dynamics of Rule 110 as a combination of block mappings and re-orderings [148].



Reversible one-dimensional CA of 9 states, neighbourhood size 2 and invertible memory of size 9. ©2014 Juan Carlos Seck-Tuoh-Mora.

We use a eversible automaton of 9 states and Welch indices $L = 9$ and $R = 1$. The evolution rule is complemented with an invertible memory function which takes into account the current generation and the previous 8 states of each cell. This memory is defined in such a way that the global evolution of the automaton is still reversible. The first nine generations are produced by the application of the classic reversible rule. From the ninth generation, however, the memory produces a more complicated dynamics, but still remains invertible [147].



Surface interpolation for representing the dynamics of elementary CA ©2014 Juan Carlos Seck-Tuoh-Mora.
Certain tools as mean field theory and basins of attraction are limited to explore the complex dynamics of elementary CA. The interpolation techniques can be applied to represent and classify the dynamics of elementary CA. The picture shows different interpolation surfaces obtained for elementary automata with hundred cells. A rich family of surfaces emerges in this analysis [149].

Applied Mathematical Sciences, Vol. 9, 2015, no. 144, 7167 - 7173
HIKARI Ltd, www.m-hikari.com
http://dx.doi.org/10.12988/ams.2015.510622

Study of Renewable Systems Based on Functional Operators with Shift

Oleksandr Karelin, Manuel Gonzalez-Hernandez,
Norberto Hernandez-Romero and Juan Seck-Tuoh-Mora

Hidalgo State University, Institute of Basic Sciences and Engineering
Carretera Pachuca-Tulancingo, Km.4.5, C.P.42184, Pachuca Hgo, Mexico

Abstract

In previous works we proposed a method for the study of systems with one renewable resource. The separation of the individual and the group parameters and the discretization of time led us to scalar linear functional equations with shift. Cyclic models, in which the initial state of the system coincides with the final state, were considered. In this work, we present cyclic models for systems with two renewable resources. In modeling, the interactions and the reciprocal influences between these two resources are taken into account. We applied our results on invertibility of the functional operators with shift to the study of the balance equations The equilibrium state of the system is found.

Mathematics Subject Classification: 47B38, 39B22, 37N99

Keywords: renewable resources, reciprocal influence, equilibrium state, Hölder space, inverse operator

1 Introduction

The interest towards the study of functional operators with shift was stipulated by the development of solvability theory and Fredholm theory of characteristic singular integral equation with non-Carleman shift [3, 4, 5]. Our motivation to return to these investigations has its reasons. On modeling systems with renewable resources, linear functional operators with shift appear [6, 2]. These operators are the adequate mathematical instrument for the investigation of such systems.

In Section 2, we construct a cyclic model of a system with two renewable resources, in which the initial state of the system coincides with the final state. Terms that take into account the reciprocal influence of these resources on each other are introduced. The system of balance equations of the cyclic model is represented throught lineal functional equations with shift.

In Section 3, using some known results [7] on the invertibility of the functional operator with shift in weighted Hölder spaces, a method for solution of the system of balance equations is offered. The equilibrium state of the system is found.

2 Cyclic model of a system with two renewable resources

Let S be a system with two resources λ_1, λ_2, and let T be a time interval. Let t_0 be the initial time and S the system under consideration.

The initial state of the system S at time t_0 is represented by density functions of the distribution of the group parameter by the individual parameter for each resource $v(x,t_0)=v(x), 0< x <x_{max}$, $w(y,t_0)=w(y), 0< y <y_{max}$.

We will now analyze the system's evolution. In the course of time, the elements of the system can change their individual parameter - e.g. fish can change their weight and length. Modifications in the distribution of the group parameters by the individual parameters is represented by a displacement. The state of the system S at the time $t = t_0 + T$ is:

$$v(x,t_0+T) = \frac{d}{dx}\alpha(x)\cdot v(\alpha(x)), \quad w(y,t_0+T) = \frac{d}{dy}\beta(y)\cdot w(\beta(y)). \qquad (1)$$

In the article [6], the appearance of derivatives in (1) was explained.

Over the period $j_0 = [t_0, t_0+T]$, extractions might be taken from the system as a result of human economic activity; these are represented by summands $\rho(x)$, $\delta(y)$. If an artificial entrance of elements into the system has taken place, it shall be accounted for by adding terms $\zeta(x)$, $\xi(y)$. We take natural mortality into account with the coefficients $c(x)$, $d(y)$.

The process of reproduction will be represented by terms

$$\sum_{i=1}^{n} P_i p_i(x), \quad P_i = \int_{\nu_{i-1}}^{\nu_i} v(x)dx,\ 0 = \nu_0 < \nu_1 < ... < \nu_n = x_{max},$$

$$\sum_{i=1}^{m} Q_i q_i(y), \quad Q_i = \int_{\mu_{i-1}}^{\mu_i} w(y)dy, \ 0 = \mu_0 < \mu_1 < ... < \mu_m = y_{max}.$$

We obtain

$$v(x, t_0 + T) = c(x)\frac{d}{dx}\alpha(x)v(\alpha(x)) + \rho(x) + \zeta(x) + \sum_{i=1}^{n} P_i p_i(x),$$

$$w(y, t_0 + T) = d(y)\frac{d}{dy}\beta(y)w(\beta(y)) + \delta(y) + \xi(y) + \sum_{i=1}^{m} Q_i q_i(y).$$

We will account for reciprocal influence of the resources λ_1 and λ_2 by

$$\sum_{i=1}^{k} R_i r_i(x), \quad R_i = \int_{\gamma_{i-1}}^{\gamma_i} w(y)dy, \quad 0 = \gamma_0 < \gamma_1 <, ..., < \gamma_k = y_{max},$$

$$\sum_{i=1}^{l} F_i f_i(y), F_i = \int_{\epsilon_{i-1}}^{\epsilon_i} v(x)dx, \quad 0 = \epsilon_0 < \epsilon_1 <, ..., < \epsilon_l = x_{max}.$$

Thereby, the final state of the system at the moment $[t_0 + T]$ is described as follows:

$$v(x, t_0+T) = c(x)\frac{d}{dx}\alpha(x)v(\alpha(x)) + \rho(x) + \zeta(x) + \sum_{i=1}^{n} P_i p_i(x) + \sum_{i=1}^{k} R_i r_i(x), \quad (2)$$

$$w(y, t_0+T) = d(y)\frac{d}{dy}\beta(y)w(\beta(y)) + \delta(y) + \xi(y) + \sum_{i=1}^{m} Q_i q_i(y) + \sum_{i=1}^{l} F_i f_i(y). \quad (3)$$

Let our goal be to find the equilibrium state of system S, that is, to find such an initial distribution of the group parameters by the individual parameter $v(x, t_0)$, $w(x, t_0)$, that after all transformations during the time interval $(t_0, t_0 + T))$, it would coincide with the final distribution:

$$v(x, t_0) = v(x, t_0 + T) = v(x), \ w(y, t_0) = w(y, t_0 + T) = w(y). \quad (4)$$

From here, substituting relations (2) and (3) into (4), it follows that

$$v(x) = c(x)\frac{d}{dx}\alpha(x)v(\alpha(x)) + \rho(x) + \zeta(x) + \sum_{i=1}^{n} P_i p_i(x) + \sum_{i=1}^{k} R_i r_i(x), \quad (5)$$

$$w(y) = d(y)\frac{d}{dy}\beta(y)w(\beta(y)) + \delta(y) + \xi(y) + \sum_{i=1}^{m} Q_i q_i(y) + \sum_{i=1}^{l} F_i f_i(y). \quad (6)$$

Equations (5), (6) are called equilibrium proportions or balance equations. A model is called cyclic if the state of system S at the initial time t_0 coincides with the state of system S at the final time $t_0 + T$.

Without loss of generality, we assume below that $x_{max} = 1, y_{max} = 1$.

3 About the space in which the cyclic model is considered and conditions of invertibility of functional operators with shift

A function $\varphi(x)$ that satisfies the condition on $J=[0,1]$,

$$\mid \varphi(x_1)-\varphi(x_2) \mid \leq C \mid x_1-x_2 \mid^{\varsigma}, \quad x_1\in J, x_2\in J,\ \varsigma\in(0,\ 1),$$

is called a Hölder function with exponent ς and constant C on J.

Let ϱ be a power function which has zeros at the endpoints $x=0$, $x=1$:

$$\varrho(x)=x^{\varsigma_0}(1-x)^{\varsigma_1}, \quad \varsigma<\varsigma_0<1+\varsigma,\ \varsigma<\varsigma_1<1+\varsigma.$$

The functions that become Hölder functions and turn into zero at points $x=0$, $x=1$, after being multiplied by $\varrho(x)$, form a Banach space. Functions of this space are called Hölder functions with weight ϱ: $H^0_{\varsigma}(J,\varrho),\ J=[0,1]$. The norm in the space $H^0_{\varsigma}(J,\varrho)$ is defined by

$$\| f(x) \|_{H^0_{\varsigma}(J,\varrho)} = \|\varrho(x)f(x)\|_{H_{\varsigma}(J)},\ \|\varrho(x)f(x)\|_{H_{\varsigma}(J)} = \|\rho(x)f(x)\|_C + \|\rho(x)f(x)\|_{\varsigma},$$

where

$$\|\varrho(x)f(x)\|_C = \max_{x\in J} |\varrho(x)f(x)|,$$

$$\|\varrho(x)f(x)\|_{\varsigma} = \sup_{x_1,x_2\in J, x_1\neq x_2} |\varrho(x)f(x)|_{\varsigma},\ |\varrho(x)f(x)|_{\varsigma} = \frac{\mid \varrho(x_1)f(x_1)-\varrho(x_2)f(x_2)\mid}{\mid x_1-x_2 \mid^{\varsigma}}.$$

Let $\beta(x)$ be a bijective orientation-preserving displacement on J: if $x_1<x_2$ then $\beta((x_1)<\beta(x_2)$ for any $x_1\in J,\ x_2\in J$; and let $\beta(x)$ have only two fixed points: $\beta(0)=0,\ \beta(1=1,\ \beta(x)\neq x,\ when\ x\neq 0, x\neq 1$. In addition, let $\beta(x)$ be a differentiable function and $\frac{d}{dx}\beta(x)\neq 0,\ x\in J$.

We consider the equation

$$(A\nu)(x)=f(x),\ (A\nu)(x)\equiv a(x)(I\nu)(x)-b(x)\left(\Gamma_{\beta}\nu\right)(x), \quad x\in[0,1] \qquad (7)$$

where I is the identity operator and Γ_{β} is the shift: $\left(\Gamma_{\beta}\nu\right)(x)=\nu[\beta(x)]$.

Let functions $a(x),\ b(x)$ from operator A belong to $H_{\varsigma}(J)$.

We will now formulate conditions of ivertibility for operator A from (7) in the space of Hölder class functions with weight [6].

Theorem 3.1 *Operator A, acting in Banach space $H^0_{\varsigma}(J,\varrho)$, is invertible if the following condition is fulfilled:* $\theta_{\beta}[a(x),b(x),H^0_{\varsigma}(J,\varrho)]\neq 0, \quad x\in J$, *where the function σ_{β} is defined by* $\theta_{\beta}[a(x),b(x),H^0_{\varsigma}(J,\varrho)]=$

$$\begin{cases} a(x), when\ |a(0)|>[\beta'(0)]^{-\varsigma_0+\varsigma}\ |b(0)|; and, |\,a(1)\,|>[\beta'(1)]^{-\varsigma_1+\varsigma}\ |b(1)|; \\ b(x), when\ |\,a(0)\,|<[\beta'(0)]^{-\varsigma_0+\varsigma}\,|\,b(0)\,|; and, |\,a(1)\,|<[\beta'(1)]^{-\varsigma_1+\varsigma}|b(1)|; \\ 0 \qquad in\ other\ cases. \end{cases}$$

4 Analysis of the solvability of the balance equations and finding the equilibrium state of the system

Let S be the system with two resources considered in Section 2. We find the equilibrium state of the system in which the initial distribution of the group parameters by the individual parameters $v(x), w(y),\ x \in (0,1)$ coincide with the final distribution, after all transformations during the time interval T.

We rewrite the balance equations of the cyclic model (5), (6) for system S in the form

$$(Vv)(x)=\sum_{i=1}^{n} P_i p_i(x)+\sum_{i=1}^{k} R_i r_i(x)+g(x),\ (Vv)(x) = v(x) - c_\alpha(x)v(\alpha(x)) \quad (8)$$

$$(Ww)(y)=\sum_{i=1}^{m} Q_i q_i(y)+\sum_{i=1}^{l} F_i f_i(y)+h(y),\ (Ww)(y)=w(x)-d_\beta(y)w(\beta(y)). \quad (9)$$

Let us study the model in the space of Hölder class functions with weight:

$$H_\varsigma^o(J,\varrho),\ \varrho(x)=x^{\varsigma_0}(1-x)^{\varsigma_1}, \quad 0<\varsigma<1,\ \varsigma<\varsigma_0<1+\varsigma,\ \varsigma<\varsigma_1<1+\varsigma,$$

$$H_\vartheta^0(J,\sigma),\ \sigma(y)=y^{\vartheta_0}(1-y)^{\vartheta_1}, \quad 0<\vartheta<1,\ \vartheta<\vartheta_0<1+\vartheta,\ \vartheta<\vartheta_1<1+\vartheta,$$

considering conditions of invertibility of operators V and W fulfilled

$$\theta_\alpha[1, c_\alpha(x), H_\varsigma^0(J,\varrho)] \neq 0,\ x \in J, \quad \theta_\beta[1, d_\beta(y), H_\vartheta^0(J,\sigma)] \neq 0,\ y \in J. \quad (10)$$

From Theorem 1, inverse operators to operators V and W exist. Specific forms of inverse operators V^{-1} and W^{-1} can be found in [7]. Note that the series representing inverse operators in Lebesgue spaces with weight [4] are also suited for Hölder spaces with weight, when they converge.

For solving the system of equations, let us make use the approach for the solution of integral Fredholm equations of the second type with degenerate kernel [1].

First, let us apply operators V^{-1}, W^{-1} to the left side of equations (8), (9); we obtain

$$v(x) = \sum_{i=1}^{n} P_i(V^{-1}p_i)(x) + \sum_{i=1}^{k} R_i(V^{-1}r_i)(x) + (V^{-1}g)(x), \quad (11)$$

$$w(y) = \sum_{i=1}^{m} Q_i(W^{-1}q_i)(y) + \sum_{i=1}^{l} F_i(W^{-1}f_i)(y) + (W^{-1}h)(y). \quad (12)$$

Having integrated the first equation of system (11) over intervals $[\nu_{j-1}, \nu_j]$, $j = 1, 2, ..., n$ corresponding to constants $P_j = \int_{\nu_{j-1}}^{\nu_j} v(x)dx$, and over intervals $[\epsilon_{j-1}, \epsilon_j], j = 1, 2, ..., l$ corresponding to constants $F_j = \int_{\epsilon_{j-1}}^{\epsilon_j} v(x)dx$, and having subsequently integrated the second equation of system (12) over intervals $[\mu_{j-1}, \mu_j],\ j = 1, 2, ..., m$ corresponding to constants $Q_j = \int_{\mu_{j-1}}^{\mu_j} w(y)dy$, and over intervals $[\gamma_{j-1}, \gamma_j], j = 1, 2, ..., k$ corresponding to $R_j = \int_{\gamma_{j-1}}^{\gamma_j} w(y)dy$, we have a system of $r = n + m + l + k$ linear algebraic equations with the same number of unknowns.

Let us assume that the determinant of this system is different from zero $det\Delta \neq 0$. After finding the unknown constants, we can define the solution of the balance system (11), (12). Thus, we have found the equilibrium state of the cyclic model of the system S, that is, we have obtained the state of the system to which it returns after the time T.

5 Results and conclusions

The theory of linear functional operators with shift is the adequate mathematical instrument for the investigation of systems with renewable resources [6, 2]. In this work, we present cyclic models for systems with two renewable resources. In modeling, the interactions and the reciprocal influences between the resources are taken into account. Balance correlations are found. Analysis of the models is carried out in weighted Hölder spaces. A method for the solution of balance equations, developed through the application of inverse operators to functional operators with shift, is offered. The equilibrium state of the system is obtained.

References

[1] K.E. Atkinson, The Numerical Solution of Integral Equations of the Second Kind, Cambrige University Press, 1997. http://dx.doi.org/10.1017/cbo9780511626340

[2] O. Karelin, A. Tarasenko, M.G. Hernández, Application of functional operators with shift to the study of renewable systems when the reproductive processed is described by integrals with degenerate kernels, *Appl. Math.*, **4** (2013), 1376-1380. http://dx.doi.org/10.4236/am.2013.410186

[3] Yu. Karlovich, V. Kravchenko, Singular integral equations with non-Carleman shift on an open contour, *Diff. Equat.*, **17** (1981), 1408-1417.

[4] V.G. Kravchenko, G.S. Litvinchuk, *Introduction to the Theory of Singular Integral Operators with Shift*, Kluwer Acad. Publ., Dordrecht, Boston, London, 1994. http://dx.doi.org/10.1007/978-94-011-1180-5

[5] G.S. Litvinchuk, *Solvability Theory of Boundary Value Problems and Singular Integral Equations with Shift*, Kluwer Acad. Publ., Dordrecht, Boston, London, 2000. http://dx.doi.org/10.1007/978-94-011-4363-9

[6] A. Tarasenko, A. Karelin, G.P. Lechuga, M.G. Hernández, Modelling systems with renewable resources based on functional operators with shift, *Appl. Math. and Comput.*, **216** (2010), 1938-1944. http://dx.doi.org/10.1016/j.amc.2010.03.023

[7] A. Tarasenko, O. Karelin, On invertibility of functional operators with shift in weighted Hölder spaces, *Advances in Pure Mathematics*, **4** (2014), 595-600. http://dx.doi.org/10.4236/apm.2014.411068

Received: October 18, 2015; Published: December 12, 2015

Int J Adv Manuf Technol
DOI 10.1007/s00170-016-8638-9

ORIGINAL ARTICLE

Simulation and validation of diagram ladder—petri nets

J. C. Quezada[1] · J. Medina[2] · E. Flores[1] · J. C. Seck Tuoh[2] · A. E. Solís[1] · V. Quezada[1]

Received: 11 December 2015 / Accepted: 20 March 2016

Abstract Automated systems based on programmable logic controllers (PLC) are still applied in discrete event systems (DES) for controlling and monitoring of industrial processes signals. PLC-based control systems are characterized for having physical input and output signals coming from and going to sensors and actuators, respectively, which they are in direct contact with the production or manufacturing process. The input subsystem to PLC consists of sensor-wiring-physical inputs module, and it can present two kinds of faults: short circuit or open circuit, in one or more signals of the process physical inputs, which it causes faults in the control and/or in the control algorithms behavior. Ladder diagram (LD) is one of the five programming languages supported by the International Electrotechnical Commission (IEC) through the IEC-61131-3 standard, and it remains being used at industry for control algorithm design of PLC-based systems. This paper proposes the simulation and validation of control algorithms developed in LD by using Petri Nets (PN) in order to deal with the possible fault options (short circuit and/or open circuit) in the physical inputs subsystem of a PLC-based control system. One control algorithms in LD have been analyzed in order to show the advantages of the proposed approach.

✉ J. C. Quezada
jcarlos@uaeh.edu.mx

1 Escuela Superior de Tizayuca, Universidad Autónoma del Estado de Hidalgo, Ext. 5700 km 2.5 Carretera Federal Tizayuca-Pachuca, Hidalgo, México

2 Centro de Investigación Avanzada en Ingenieria Industrial, Universidad Autónoma del Estado de Hidalgo, Hidalgo, Mexico

Keywords Control algorithms · Discrete event systems · Ladder diagram · Petri nets · Programmable logic controller · Simulation · Validation

1 Introduction

Control based on programmable logic controllers (PLC) still remains being used in a large variety of production or manufacturing processes. PLCs can be programmed through different programming languages, namely structured text (ST), instruction list (IL), function block diagram (FBD), sequential function chart (SFC), and ladder diagram (LD), which they are the five languages considered in the IEC-61131-3 standard (International Electrotechnical Commission) [1]. This standard establishes the syntax and semantics of these programming languages, but not the verification and/or validation of the control algorithms, which they have been and they are still developed based on the experience of those responsible for controlling the systems. The problem of guaranteeing safe control algorithms has been treated in theory through different approaches having as main basis the formal specifications of the system being controlled, and its validation or verification is based mainly on theoretical concepts. Approaches recently proposed are mentioned below.

Conversion of control algorithms into machines *B* for their formal analysis of security limitations is presented in [2]. Generating the machine *B* is based on the project's specifications. The informal specifications or non-explicit limitations are "*manually*" incorporated to the control algorithm refinement.

In [3], it is shown the modeling and validation of a PLC-based control system by using the behavior, interaction, and priority (BIP) component framework. The authors propose a monitor per each of the properties being validated; then they integrate all the monitors in order to yield the global system for its respective simulation. If a requirement is violated, the corresponding monitor will change an error state. This approach was applied to a real system where *"errors"* are presented in the system's global design.

A formal verifying method, based on the user's specifications, is presented in [4]. Firstly, it is implemented in Unified Modeling Language (UML), and then it transformed into a Petri Net (PN) for its validation. The verifying process is accomplished through a tool Symbolic Model Checker (NuSMV), where the checker is based on the specifications and properties of the system, and it uses the temporal logic for defining the properties.

System's specifications are divided into operating predictions, operating behavior, exception conditions, exception behavior, and invariants. Temporal logic is the basis for sequencing the system states. All the previous introduces the concept of Reusable Automation Components (RAC) for a scanning of PLC, and the semantics for updating the system signals state, considering the system as valid if all of the behavior operations are successfully completed before the update and all the operating preconditions are not exception conditions, as well as each of exception behavior and each invariant must be successfully completed for each updating [5].

In [6], the authors mention software for validating control algorithms developed in Instruction List language; however, they consider that they are limited by being focused on theoretical attributes (security, liveliness, and reachability). The authors' proposal is to develop an environment that enables the visual verification of the control algorithms through a 3D graphical environment of the system to be controlled which it is based on a mapping from the state of the physical inputs and outputs of the PLC-based system.

An approach on verification and validation off-line of control algorithms is presented in [7]. This proposal is based on the III phase V & V method, which it involves tests on manual, model checker, and virtual commissioning for the system specifications. The authors consider that after fulfilling these proofs, the code may be implemented in a PLC-based system.

Approaches focused on detection and/or locations of faults in control algorithms of PLC-based systems have also been proposed.

In [8], the authors present a new method which it treats sensor fault as state variable to enforce fault diagnosis, it based in the builder of model of sensor fault into state equation to evaluate the control algorithm.

In [9], a diagnosis system for improving the reliability of PLC-based systems is proposed. The authors consider that system developers and programmers are not able to identify each fault that may occur in the system. Their approach, FDS-PLC (Fault Diagnosis System-Programmable Logic Controller), executes in "*parallel*" both the control system in the PLC and the diagnosis system based on a finite state machine, and it runs in a personal computer connected to the PLC. The diagnosis approach proposes an initial state of the system based on the specifications, the input signals' state is copied, the copied input is compared to that of the initial state; if there is no correspondence to the specifications, it is reported as "*fault or unknown status*"; otherwise, the system state is updated, and the reading of input signals as well as the comparison of their state is periodically continued.

In [10], it is considered that the main causes of faults in input signals are short circuit and open circuit due to damage at the connection lines from sensors to PLCs; or due to faults in the mechanical contacts of switches, or by damage in the electronic sensors. For the reliability of the input signals, the authors propose that various sensors have high reliability and to remove the "*causes*" in order to avoid short circuit, open circuit or connection line to PLC. The reliability of the input signal from the PLC production site can be estimated according to the control system characteristics, as well as the relationship between signals.

An example of sequence in LD is considered in [11], showing the "*vulnerability*" of the control algorithm. The system opens a door with the sequence of pushing four pushbuttons, a sensor detecting the door state (closed-open), as well as a button to reset the system conditions. It is considered that by pushing all the buttons at one time and in the same PLC scans the door would open because the control algorithm is executed each cycle from the left to right and from the top to bottom. The proposal of pushing all the buttons at one time is equivalent to the extreme case of short circuit fault for all input sensors to PLC; however, for this example, the door would not open, since the control algorithm in the PLC is executed each cycle based on the copy of the states of the input signals of a same "*moment*" (reading stage of input signals in the scan). The language Cadence SMV is used for validating control algorithms developed in LD. The modeling basis is the conversion of the control algorithm into LD, in logic AND, OR, and NOT.

The operation and states of sensors and actuators are continuously monitored through Framework OPC Server connected to the PLC. A vector of normal operating values of signals is compared to the real-time observed values; if a discrepancy exists, it will be indicated through an alarm [12]. A fault condition can coincide with the corresponding state at this moment of the process, which it would allow a sequence more in the process.

The use of real-time PN allows reading the states of process inputs and outputs, which they are compared to predetermined values; if a difference exists, the information will be treated with fuzzy PN in order to diagnose and find the root cause of fault. For the state equation, it is added an equalization between the possible values mapping of the set of inputs and outputs, and the reachable markings from an initial marking [13].

A general procedure for fault detecting in PLC-based systems is presented in [14]. The authors consider some hardware and software problems for determining a generic fault, supported by light indicators at the modules integrating the PLC. It is important to highlight that a better understanding of the system allows an effective and efficient solution of faults.

In general, as far as we know, the proposed approaches on validation do not take into account physical faults of short circuit and open circuit in the input subsystems (sensor-wiring-physical input module) in the PLC-based systems. In the present work, the concept of validation proposed in [15] it is considered, establishing that "the process of evaluating a model, simulation, or federation of models and simulations throughout the development and execution process to determine how well it satisfies the acceptability criteria within the context of the referent; the process of determining the degree to which a model is an accurate representation of the problem space from the perspective of the intended uses of the model".

In this paper, we show the simulation of control algorithms considering the behavior of scan of the PLC, besides, a method for validating control algorithms developed in LD in fault conditions in the physical inputs subsystem in a PLC-based control system is proposed. The proposed validation has been evaluated in a real application control algorithm, and it has allowed obtaining safety results about what conditions must be included in the LD in order to avoid they occur in case of fault.

This work has been organized as following. Sections 2 and 3 introduce concepts about PLC- and PN-based systems, respectively. Section 4 explains the faults of short circuit and open circuit, the signals characterization in PN elements, and their considerations in incidence matrix as well as the validation proposal of control algorithms developed in LD. Section 5 shows the validation in two real cases and the obtained results.

2 PLC-based control systems

PLC is a "digitally operating electronic system, designed for use in an industrial environment, which it uses a programmable memory for the internal storage of user-oriented instructions for implementing specific functions such as logic, sequencing, timing, counting and arithmetic, to control, through digital or analogue inputs and outputs, various

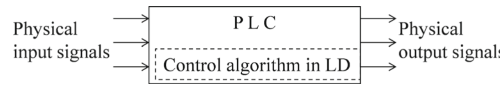


Fig. 1 PLC-based control system

types of machines or processes. Both the PLC and its associated peripherals are designed so that they can be easily integrated into an industrial control system and easily used in all their intended functions" and PLC-based system is a "user-built configuration, consisting of a programmable controller and associated peripherals, that is necessary for the intended automated system. It consists of units interconnected by cables or plug-in connections for permanent installation and by cables or other means for portable and transportable peripherals" [16].

PLC-based systems for DES are characterized by having physical input signals coming from the process (sensors, switches, selectors, among others), connected to the PLC input modules. Based on the state of these signals, the control algorithm is executed, and its results are reflected in the modules of physical output signals which they are connected to the process actuators (relays, contactors, electro-valves or solenoid valves, among others). Figure 1 shows a PLC-based control system.

2.1 Ladder diagram

LD is one of the five programming language supported by the standard IEC-61131-3 for developing PLC control algorithms. LD is considered a graphic-type language having as functioning basis the behavior of an electromechanical relay. In [17], it is defined like "modeling networks of simultaneous functioning electromechanical elements, such as relay contacts and coils, timers, counters, etc.".

A contact can be normally close (NC) or normally open (NO). For a PLC-based system, a NO and/or NC contact may come from a mechanical or electrical sensor, which it closes or opens the electrical circuit, to the physical inputs module, which it detects voltage presence or absence for the state (0 or 1) of the corresponding variable. Both the voltage level and signal type (direct or alternating) are in function of the input module. Also, a NO and/or NC contact may be a memory internal signal that is linked to a coil, internal too. A physical input signal might be considered as many times as necessary in the control algorithm through NO and/or NC contacts.

2.2 Scan of a program

The periodic or cyclic execution of a control algorithm is the operating basis of the PLC-based systems. Figure 2 [18]

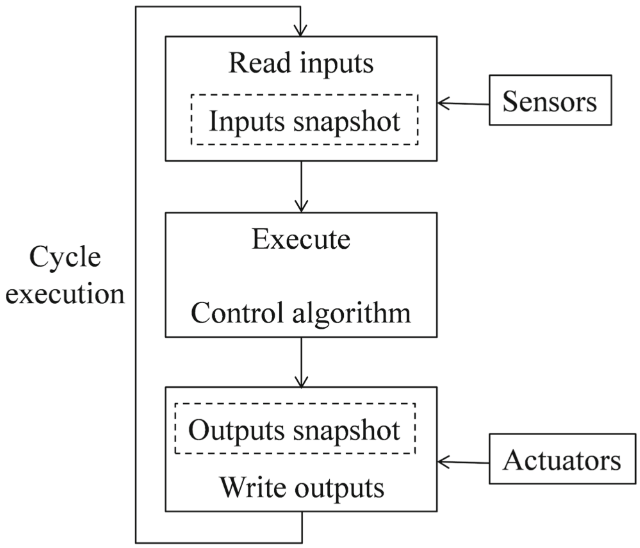


Fig. 2 Cyclic running of a PLC control algorithm

shows, in a general way, the scan of the control algorithms, standing out the image of the states of physical input signals, with which the control algorithm is evaluated.

Ideally, during the evaluation time of the control algorithm at the scan period, a change in the state of the physical input signals does not affect the control execution, but until the new image of the states of input signals is updated. This allows evaluating, in an independent way, the control algorithm in function of possible states of the physical input signals.

3 Petri nets

PNs are a graphic and mathematical tool for modeling the DES behavior. From [19], Table 1 considers the formal definition of a PN in its basic form, as well as its analysis tools, which they are subsequently described.

As part of their formal definition, PNs offer tools for carrying out the analysis of the modeled system. Some of them are described following.

Table 1 Formal definition of a PN

A Petri net is a 5-tuple, $PN = (P, T, F, W, M)$ where:
$P = \{p_1, p_2, ..., p_m\}$ is a finite set of places,
$T = \{t_1, t_2, ..., t_n\}$ is a finite set of transitions,
$F \subseteq (P \times T) \cup (T \times P)$ is a set of arcs,
$W : F \rightarrow \{1, 2, 3, ...\}$ is a weight function,
$M_0 : P \rightarrow \{0, 1, 2, ...\}$ is an initial marking, and
$P \cap T = \emptyset$ and $P \cup T \neq \emptyset$

3.1 Coverability tree

The coverability tree allows finding the possible markings of a PN from an initial marking M_0. The PN will have M_k markings depending on which transitions are enabled, which ones are being enabled, and in which sequence each enabled transition is fired. The result of the firings sequence may be represented by means of a tree, where the root is the initial marking M_0, and depending on the transitions' firing sequence, the tree branches with their respective new markings are generated [19].

3.2 Incidence matrix

In order to represent the dynamic behavior of the PNs, the incidence matrix is used, which relates the weightings of the input and output arcs from transitions to places and vice versa. For a PN with n transitions and m places, its incidence matrix $A = [a_{ij}]$ is an integer numbers matrix representing the weighting of the input and output arcs; ${a_{ij}}^+$ represents the weighting of output arcs from transitions, and ${a_{ij}}^-$ represents input arcs to transitions. Equation 1 represents how the incidence matrix values are obtained.

$$a_{ij} = {a_{ij}}^+ - {a_{ij}}^- \tag{1}$$

3.3 State equation

The state equation shows the marking in a sequence state through the relationship between the vector of a preceding state with certain system marking M_{k-1}, the transpose of the incidence matrix A and a firing vector u_k determining the process firing sequence. Equation 2 shows the relationship between them.

$$M_k = M_{k-1} + A^T u_k \tag{2}$$

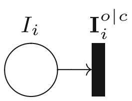
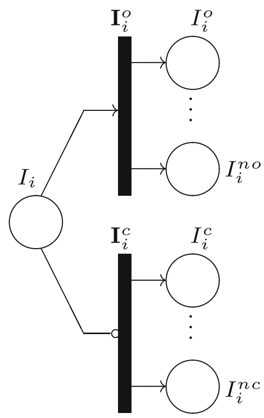
4 Simulation control algorithms in LD with PN

In this section, we propose the mathematics equations to simulate the dynamic behavior of control algorithms development in LD with PN.

4.1 Characterization of signals

LD has as basis the behavior of an electromechanical relay, so contains NO and NC contacts and coils. A signal (of physical input and/or output or of memory) in a LD may have elements at diverse lines. In [18], a signal distribution based on the relay behavior is proposed; that is to say, if the signal is activated, the NO contacts close, and those

Table 2 Representation of a physical input by PN elements

Signal	Contact distribution
	

NC open. Also, we consider the definition of the net *LDPN* (Ladder Diagram Red de Petri).

Table 2 shows the distribution of a physical input signal by employing PN elements. Where I_i is a place representing a physical input signal, and I_i^o and I_i^c are places representing the NO and NC contacts of the signal, respectively. The use of the inhibitor arc allows that only one of transitions, $\mathbf{I}_i^o$ or $\mathbf{I}_i^c$, are enabled, modeling the behavior of that only one type of contact of a same signal can be activated in a scanning. Such a distribution is analogue for physical output signals O_o as well as of internal memory B_b of the PLC. In general, the types of contacts of a signal are represented by the Eqs. 3a–3f.

$$I_i^o = \#\text{ contacts NO of physical inputs signals} \tag{3a}$$

$$I_i^c = \#\text{ contacts NC of physical inputs signals} \tag{3b}$$

$$O_o^o = \#\text{ contacts NO of physical outputs signals} \tag{3c}$$

$$O_o^c = \#\text{ contacts NC of physical outputs signals} \tag{3d}$$

$$B_b^o = \#\text{ contacts NO of memory signals} \tag{3e}$$

$$B_b^c = \#\text{ contacts NC of memory signals} \tag{3f}$$

The signals distribution must fulfill the following characteristics:

1. PN is binary, only may have as maximum, one token in each place, $W : F \longrightarrow 0, 1$,
2. Only one transition from $\mathbf{I_i^o}$ or $\mathbf{I_i^c}$ of a signal may be activated at a time, and its marking fulfills for Eqs. 4a–4c,

$$M(I_i) = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \text{then} \begin{Bmatrix} M(I_i^o) = 0 \text{ and } M(I_i^c) = 1 \\ M(I_i^o) = 1 \text{ and } M(I_i^c) = 0 \end{Bmatrix} \tag{4a}$$

$$M(O_o) = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \text{then} \begin{Bmatrix} M(O_o^o) = 0 \text{ and } M(O_o^c) = 1 \\ M(O_o^o) = 1 \text{ and } M(O_o^c) = 0 \end{Bmatrix} \tag{4b}$$

$$M(B_b) = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \text{then} \begin{Bmatrix} M(B_b^o) = 0 \text{ and } M(B_b^c) = 1 \\ M(B_b^o) = 1 \text{ and } M(B_b^c) = 0 \end{Bmatrix} \tag{4c}$$

5 Accumulation tokens problems

Propose of this investigation to the accumulation tokens problems, it is set logical functions to enable marking for places O_o and B_b. In Eqs. 5a and 5b, they are to enable marking in the output place O_o y B_b respectively, when input structure PN is logical, and if input structure PN is logical OR, then the equations 6a y 6b will be enable marking for places O_o y B_b; however, if input structure PN has both logicals AND and OR, the Eqs. 7a and 7b will be enable marking for places O_o y B_b.

$$O(t^\bullet)_{AND} = \prod M(^\bullet t) = 1\ AND\ O(t^\bullet) = 0 \tag{5a}$$

$$B(t^\bullet)_{AND} = \prod M(^\bullet t) = 1\ AND\ B(t^\bullet) = 0 \tag{5b}$$

$$O(t^\bullet)_{OR} = \sum M(^\bullet t) = 1\ AND\ O(t^\bullet) = 0 \tag{6a}$$

$$B(t^\bullet)_{OR} = \sum M(^\bullet t) = 1\ AND\ B(t^\bullet) = 0 \tag{6b}$$

$$\begin{aligned} O(t\bullet))_{ANDOR} &= \sum\Big(\prod(M(\bullet t)_{L1} = 1)), \ldots, (\prod(M(\bullet t)_{Li} = 1)\Big) \\ &= 1\ AND\ O(t\bullet) = 0 \end{aligned} \tag{7a}$$

$$\begin{aligned} B(t\bullet))_{ANDOR} &= \sum\Big(\prod(M(\bullet t)_{L1} = 1)), \ldots, (\prod(M(\bullet t)_{Li} = 1)\Big) \\ &= 1\ AND\ B(t\bullet) = 0 \end{aligned} \tag{7b}$$

6 Reset places problems

The Eqs. 4a–4b to model the behavior of energize or de-energize contacts NO and/or NC of one coil, when this is energize or de-energize in control algorithm in LD.

To consume the mark of outputs places O_o y B_b in one structure PN is considered the marking of inputs places and logical type. The Eqs. 8a and 8b are to reset outputs places O_o y B_b, respectively, with logical and in the structure PN. If structure PN is logical or, then the Eqs. 9a and 9b are to reset outputs places O_o y B_b, respectively; however, the Eqs. 10a and 10b are to reset outputs places O_o y B_b, respectively, when structure has both logical AND and OR.

$$G(L^{\bullet})_{AND} = \prod M(^{\bullet}t) = 0 \ AND \ O(t^{\bullet}) = 1 \tag{8a}$$

$$G(t^{\bullet})_{AND} = \prod M(^{\bullet}t) = 1 \ AND \ B(t^{\bullet}) = 0 \tag{8b}$$

$$G(t^{\bullet})_{OR} = \sum M(^{\bullet}t) = 0 \ AND \ O(t^{\bullet}) = 1 \tag{9a}$$

$$G(t^{\bullet})_{OR} = \sum M(^{\bullet}t) = 0 \ AND \ B(t^{\bullet}) = 1 \tag{9b}$$

$$G(t\bullet)))_{ANDOR} = \sum \left(\prod (M(\bullet t)_{L1} = 1)), ..., (\prod (M(\bullet t)_{Ll} = 1)\right) = 0 \ AND \ O(t\bullet) = 1 \tag{10a}$$

$$G(t\bullet)))_{ANDOR} = \sum \left(\prod (M(\bullet t)_{L1} = 1)), ..., (\prod (M(\bullet t)_{Ll} = 1)\right) = 0 \ AND \ B(t\bullet) = 1 \tag{10b}$$

7 Ordinary ladder diagram petri net

The formal definition of the Ladder Diagram Petri Net is:

Ordinary LDPN is 5-tuple $(\mathbf{P}, \mathbf{T}, \mathbf{W}, \mathbf{F}, \mathbf{M}_0)$, where:

$\mathbf{P} = \{I \cup O \cup B \cup G\}$ is a finite set of places, where:

$I = \{I_1, I_2, I_3, ..., I_i\}$ is a finite set of places that represent physical inputs signals, and by Eqs. 3a and 3b: $I_1 = \{I_1^o \cup I_1^c\}$, $I_2 = \{I_2^o \cup I_2^c\}$, $I_3 = \{I_3^o \cup I_3^c\}$, ... , $I_i = \{I_i^o \cup I_i^c\}$ are places that represent contacts NO and NC of each physical input signal and its marking it in function of the Eq. 4a.

$O = \{O_1, O_2, O_3, ..., O_o\}$ is a finite set of places that represent physical outputs signals, and by Eqs. 3c and 3d: $O_1 = \{O_1^o \cup O_1^c\}$, $O_2 = \{O_2^o \cup O_2^c\}$, $O_3 = \{O_3^o \cup O_3^c\}$, ... , $O_o = \{O_o^o \cup O_o^c\}$ are places that represent contacts NO and NC of each physical output signal and its marking it in function pf the Eq. 4b.

$B = \{B_1, B_2, B_3, ..., B_b\}$ is a finite set of places that represent memory signals, and by Eqs. 3e and 3f: $B_1 = \{B_1^o \cup B_1^c\}$, $B_2 = \{B_2^o \cup B_2^c\}$, $B_3 = \{B_3^o \cup B_3^c\}$, ... , $B_b = \{B_b^o \cup B_b^c\}$ are places that represent contacts NO y NC of each memory signal and its marking it function of the Eq. 4c.

$G = \{G(\mathbf{T}_1), G(\mathbf{T}_2), G(\mathbf{T}_3), ..., G(\mathbf{T}_g)\}$ is a finite set of places to reset outputs places and its marking it in function of the Eqs. 8a, 8b, 9a, 9b, 10a y 10b.

$\mathbf{T} = \{\mathbf{I}^{c|o} \cup \mathbf{O}^{c|o} \cup \mathbf{B}^{c|o} \cup \mathbf{L} \cup \mathbf{R}\}$ is a finite set of transitions, where:

$\mathbf{I}^{c|o} = \{\mathbf{I}_1^{c|o}, \mathbf{I}_2^{c|o}, \mathbf{I}_2^{c|o}, ..., \mathbf{I}_i^{c|o}\}$ is a finite set of transitions that have inputs places I, where $\mathbf{I}_1^{c|o} = \{\mathbf{I}_1^c \cup \mathbf{I}_1^o\}$, $\mathbf{I}_2^{c|o} = \{\mathbf{I}_2^c \cup \mathbf{I}_2^o\}$, $\mathbf{I}_3^{c|o} = \{\mathbf{I}_3^c \cup \mathbf{I}_3^o\}$, ..., $\mathbf{I}_i^{c|o} = \{\mathbf{I}_i^c \cup \mathbf{I}_i^o\}$ are transitions with inputs places I_i^c and I_i^o taht represent contacts NC and NO respectively.

$\mathbf{O}^{c|o} = \{\mathbf{O}_1^{c|o}, \mathbf{O}_2^{c|o}, ..., \mathbf{O}_o^{c|o}\}$ is a finite set of transitions that have inputs places O, where $\mathbf{O}_1^{c|o} = \{\mathbf{O}_1^c \cup \mathbf{O}_1^o\}$, $\mathbf{O}_2^{c|o} = \{\mathbf{O}_2^c \cup \mathbf{O}_2^o\}$, $\mathbf{O}_3^{c|o} = \{\mathbf{I}_3^c \cup \mathbf{O}_3^o\}$, ..., $\mathbf{O}_o^{c|o} = \{\mathbf{O}_o^c \cup \mathbf{O}_o^o\}$ are transitions with inputs places O_o^c y O_o^o that represent contacts NC and NO, respectively.

$\mathbf{B}^{c|o} = \{\mathbf{B}_1^{c|o}, \mathbf{B}_2^{c|o}, ..., \mathbf{B}_b^{c|o}\}$ is a finite set of transitions that have both inputs and outputs places B, where $\mathbf{B}_1^{c|o} = \{\mathbf{B}_1^c \cup \mathbf{B}_1^o\}$, $\mathbf{B}_2^{c|o} = \{\mathbf{B}_2^c \cup \mathbf{B}_2^o\}$, $\mathbf{B}_3^{c|o} = \{\mathbf{B}_3^c \cup \mathbf{B}_3^o\}$, ..., $\mathbf{B}_b^{c|o} = \{\mathbf{B}_i^c \cup \mathbf{B}_b^o\}$ are transitions with inputs places B_b^c y B_b^o that represent contacts NC and NO, respectively.

$\mathbf{L} = \{\mathbf{L}_1, \mathbf{L}_2, ..., \mathbf{L}_l\}$ is a finite set of auxiliary transitions that may have both inputs an outputs places I, O, y B.

$\mathbf{R} = \{\mathbf{R}_1, \mathbf{R}_2, ..., \mathbf{R}_r\}$ is a finite set of transitions that have input place G to reset outputs places.

$F \subseteq (P \times T) \cup (T \times P)$ is a set of arcs.

$W : F \to \{1\}$ all weights of the arcs are equal to 1. and,

$M_0 = P \to \{0, 1\}$ initial marking.

7.1 Marking of the LDPN

The Eqs. 4a–4c to characterization of signals, Eqs. 5a, 5b, 6a, 6b, 7a and 7b to problem of accumulation tokens and Eqs. 8a, 8b, 9a, 9b, 10a and 10b to reset outputs places, these should be evaluated after of each evaluation M_{k+1} of the state matrix to update marking of the LDPN and simulate the dynamic behavior of the cycle PLC-based system. The Fig. 3 shows the relation between places and equations.

The marking of I places this in function of the physical inputs signal (sensors).

The obtained *LDPN* of the control algorithm in LD, graphically is ordinary because it has the unit weight in all its arcs, and all its places can only have one token for each scan in the PLC. In the incidence matrix, the number of output places for physical inputs signal transitions correspond to NO and/or NC contacts.

7.2 Rules to simulation of the LDPN

Contacts NC allow energy flow in a control algorithm in LD, therefore, places I_i^c, O_o^c, B_b^c have token initial. Add a this marking token in places of protections of system is

$$M = \begin{bmatrix} I_i & I_i^o & I_i^c & \ldots & B_b & B_b^o & B_b^c & \ldots & O_o & O_o^o & O_o^c & \ldots & G_{AND} & \ldots & G_{OR} & \ldots & G_{ANDOR} \\ & & & & & & & & & & & & & & & & \\ Process & 4a & 4a & & & 4b & 4b & & & 4c & 4c & & & & & & \\ & & & & 5b & & & & 5a & & & & & & & & \\ & & & & 6b & & & & 6a & & & & & & & & \\ & & & & 7b & & & & 7a & & & & & & & & \\ & & & & & & & & & & & & 8a & & 9a & & 10a \\ & & & & & & & & & & & & 8b & & 9b & & 10b \end{bmatrix}$$

Fig. 3 Equations to simulate the LDPN

obtained initial marking M_0 of the LDPN. Next marking is in function of inputs places, which they are in function of the activation o de-activation process sensors.

To describe and simulate the dynamic behavior of a control algorithm in LD through LDPN are considered the following transition firing rules:

a) A transition $\mathbf{T} = \{\mathbf{I}^{c|o}, \mathbf{O}^{c|o}, \mathbf{B}^{c|o}, \mathbf{L}, \mathbf{R}\}$ is enable if each input place $\mathbf{P} = (I, O, B, G)$ de $\mathbf{T}$ has token, i.e., $M(\mathbf{P}) = \mathbf{W}(P, T) = 1$.
b) All transitions enabled should be fired in one same evaluation.
c) LDPN is binary, so that one enabled transition fired $\mathbf{T}$ consumes unique token $\mathbf{W}(P, T) = 1$ of each input place $\mathbf{P}$ of $\mathbf{T}$, and put one token $\mathbf{W}(T, P) = 1$ to each output place $\mathbf{P}$ of $\mathbf{T}$.
d) To update marking of the LDPN should be considered Eqs. 4a–4c to drain tokens of signal distribution ($I_i^{c|o}$, $O_o^{c|o}$, $B_b^{c|o}$), the Eqs. 5a, 5b, 6a, 6b, 7a and 7b to resolver problem of accumulation tokens and Eqs. 8a, 8b, 9a, 9b, 10a and 10b to problem of reset places.

7.3 Analysis of the incidence matrix for signal distribution

Based on the above described conditions, the inhibitor arc may be treated as an ordinary arc in the incidence matrix and in the state equation. The generalized incidence matrix, for the signals distribution from Table 2, is shown in Eq. 11, which it is analogue for the signals of physical output O, and of memory B.

$$a_{ij} = \begin{bmatrix} & I_i & I_1^o & I_2^o & \ldots & I_i^{no} & I_1^c & I_2^c & \ldots & I_i^{nc} \\ \mathbf{I}_i^o & -1 & 1 & 1 & \ldots & 1 & 0 & 0 & \ldots & 0 \\ \mathbf{I}_i^c & -1 & 0 & 0 & \ldots & 0 & 1 & 1 & \ldots & 1 \end{bmatrix} \quad (11)$$

where I_i^{no} y I_i^{nc} represent the number of contacts NO and NC of the signal I_i, which allows reducing the incidence matrix, as shown in Eq. 12.

$$ar_{ij} = \begin{bmatrix} & I_i & I_i^o & I_i^c \\ \mathbf{I}_i^o & -1 & \sharp NO & 0 \\ \mathbf{I}_i^c & -1 & 0 & \sharp NC \end{bmatrix} \quad (12)$$

where:

$i(o) = 0, 1, 2, ..., \#NO$

$i(c) = 0, 1, 2, ..., \#NC$

Graphically, the reduction is not possible since each place I_i^{no} e I_i^{nc} it is independent and it has relationship with different transitions in the PN. Two or more places I_i^{no} y/o I_i^{nc} as input to a same transition are equivalent to have two contacts NO and/or NC of the same signal in a same control line, which it is an inoperative redundancy.

The reduced incidence matrix ar_{ij} can validate the control algorithm's behavior in fault conditions of short circuit and/or open circuit in the input subsystem of the PLC-based control system. The following section describes the proposed validation algorithm.

8 Validation approach

For control algorithms design in LD, two types of specifications, formal and informal, they are mainly considered. Formal specifications include the process safety and operation signals. Informal specifications are proposed by the designer who analyzes the process and develops the corresponding control algorithm, for later testing it in the commissioning of the production system. Therefore, designing control algorithms in LD is developed heuristically based on the experience of the programmer or responsible for the process control [20]. Figure 4 presents the context for control algorithms design in LD for DES.

All system has the possibility of faults in the inputs subsystem, it includes sensors-wire-inputs module, the faults may be short-circuit or open-circuit on one signal. We consider that a risk condition is the unwanted drive of one actuator in process industrial. In control algorithm risk condition is an energized coil, which it is connect with an actuator.

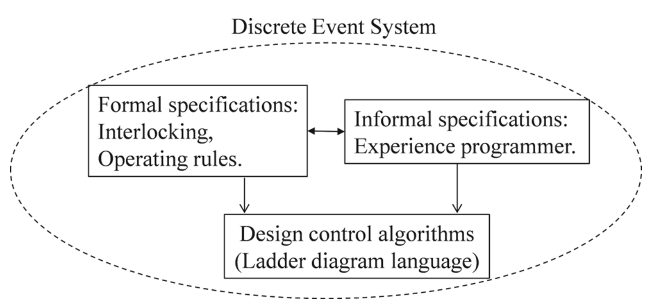


Fig. 4 Context to design control algorithms in LD

Proposed approach includes both failures in two situations, independents faults, and combination of faults in inputs signals. The LDPN is evaluated with a marking of fault, if there is token in any output place O_o is will be necessary to verify the fault condition that originates this and decide whether it should be considered in the control algorithm.

In the PLC-based systems, sensors and their connections to input modules, and output modules and their connections to actuators, can mainly represent two fault types, which they are analyzed in the following section.

8.1 Physical failures in PLC-based control systems

Regardless of the operating principle of sensors and actuators, subsystems sensor-wiring-physical input module and physical output module-wiring-actuator may represent two types of faults: short circuit or open circuit, for each of sensors and/or actuators of the process.

8.1.1 Short circuit fault

Short circuit fault at the input subsystem may occur at a sensor, at wiring, or at one of the input module sections. The fault causes that the corresponding physical input signal remains activated to the control algorithm; that is to say, in each scan of the PLC, the short-circuited signal will always be 1 for its NO contacts, and 0 for those NC.

In case the fault occurs at the output subsystem, if the fault is at wiring, then the actuators would not energize, the fault produces an overload at the corresponding output of the module; however, if the short circuit is in an output module section, then the output in fault would always be active and consequently the corresponding actuator also. Figure 5 shows the short circuit fault for both cases.

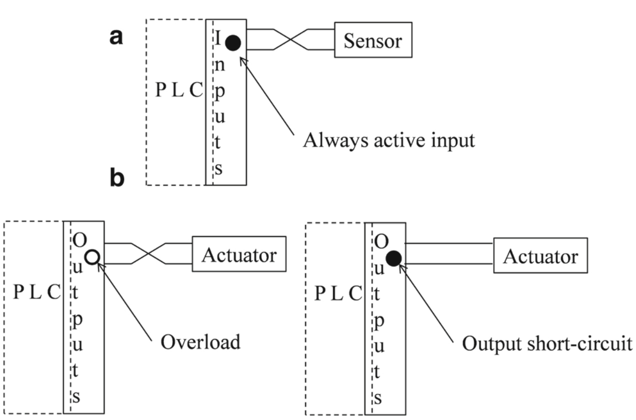


Fig. 5 Short circuit fault in subsystems of **a** inputs and **b** outputs physical signals.

8.1.2 Open circuit fault

Open circuit fault at the input subsystem may also occur at a sensor, at wiring, or at one of the input module sections, causing that the corresponding physical input signal remains disabled to the control algorithm, which produces that the NO contacts will always be open, and the NC always closed.

In the case of open circuit fault at the output subsystem, regardless of where it occurs, output module section, wiring, or actuator, the corresponding action in the process will never be accomplished, since the actuator will never energize. Figure 6 shows the open circuit fault for both cases.

Based on the described analysis about the effects causing faults, it may be determined that the affectation on the control algorithm behavior (not in the process) is mainly at the inputs subsystem, for both fault conditions. Thus, the present research proposes the validation of control algorithms considering only short-circuit and open-circuit faults at the input subsystem to PLCs.

8.2 Validation of control algorithms

A control algorithm has N-number of physical inputs, which may present fault of open circuit and/or short circuit. An input signal can only present one fault at a time. Various signals may present the same fault at a time, or some they are shorted, and the remaining be open-circuited. Equation 13 determines the number of fault possibilities Ft that may occur at the inputs subsystem of the PLC-based control, considering that the operating signal or signals may have value of 1 for active signals, and 0 for those non-active.

$$Ft = \sum_{1}^{n-1}[(2N_I)n] + 2^{N_I} \tag{13}$$

where

$n = 1, 2, ..., N_I$

N_I = number of physical input signals.

However, if it was considered that either the short circuit or open circuit fault may be presented in the input

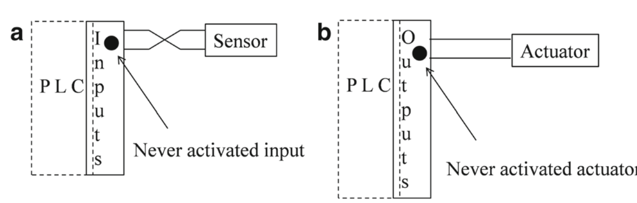


Fig. 6 Open circuit fault in subsystems of (**a**) inputs and (**b**) outputs physical signals

signals, regardless the state it has, then the possible fault combination is incremented, as shown in Eq. 14.

$$Ft = 4^{N_I} - 2^{N_I} \tag{14}$$

Each fault option is a situation to evaluate. Considering M_{Ftk} as an initial marking, by using the state equation of PN a marking in fault condition M_{Ftk+1} is obtained (Eq. 15), with which a set of markings in fault conditions $\mathbf{M}_{Ft}$ can be generated.

$$M_{Ftk+1} = M_{Ftk} + ar_{ij}^{T} * u_k \tag{15}$$

where ar_{ij} is the reduced incidence matrix, and u_k is the firing vector, whit $k = 1, 2, ..., Ft$.

From the formal operating specifications of control algorithms, of their periodic execution, and of their evaluation with the image of the states of physical input signals, the valid markings Mv of system operation can be obtained by using the coverability tree. If a marking Mv is within the set $\mathbf{M}_{Ft}$, this must be excluded from the validation in fault conditions. For the validation, it should be verified if the PNs places have mark and the fault conditions causing it, that is to say, which sensors are shorted, and which ones are open-circuited; if this is a risk condition, it should be included line or lines of control in the algorithm in order to prevent that combination of faults arises in system operation. It is noteworthy that, in the proposal, the risk condition and its corresponding proposed solution are based on the proficiency and knowledge of the process programmer. The flowchart in Fig. 6 shows the markings generation in terms of LDPN, considering the possible fault conditions of short circuit (sc) and/or open circuit (oc) of the physical input signals of a PLC-based system.

The initial markings of the physical output signals $M_0[O]$ and of memory $M_0[B]$ are not affected and should be considered together with each of the fault markings F_t for the system global evaluation.

The next section is analyzed an example to show the efficiency of the approach proposed.

9 Case study 1: carwash system

From [21], it is taken the example of an automatic control for a carwash train, as shown in Figs. 7 and 8. The system is composed of:

a) Reversible main motor, for moving the washing machine along the rail. MP1 for displacement from right to left, and MP2 vice versa.
b) Brush motor (MC), for car washing.
c) Fan motor (MV), for car drying.
d) Electro-valve (XV), for wash liquid applying.
e) Presence sensor (S3), for car detecting.
f) Limit switches (S1 and S2), for stopping the machine at the rail endings.
g) Two pushbuttons (M and P), for machine starting and stopping.

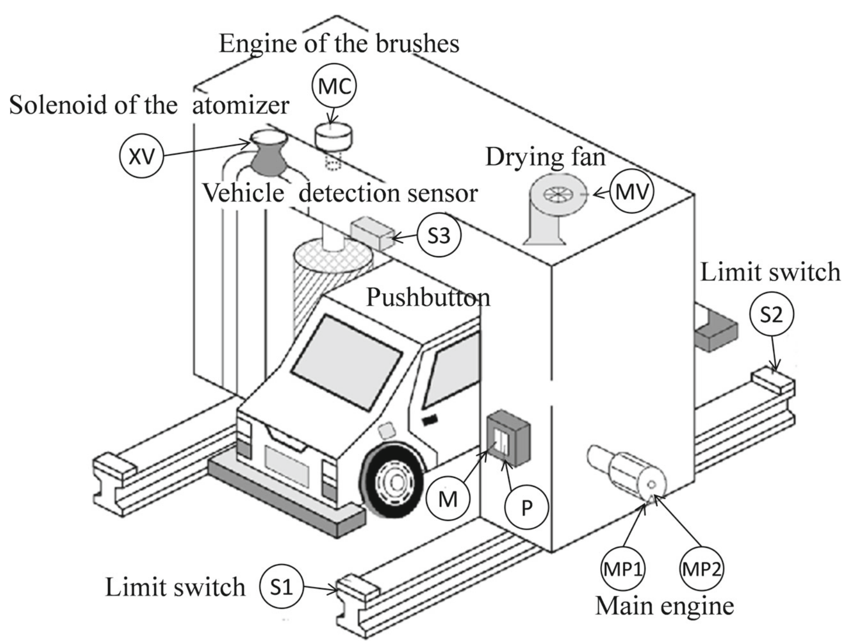


Fig. 7 Carwash system

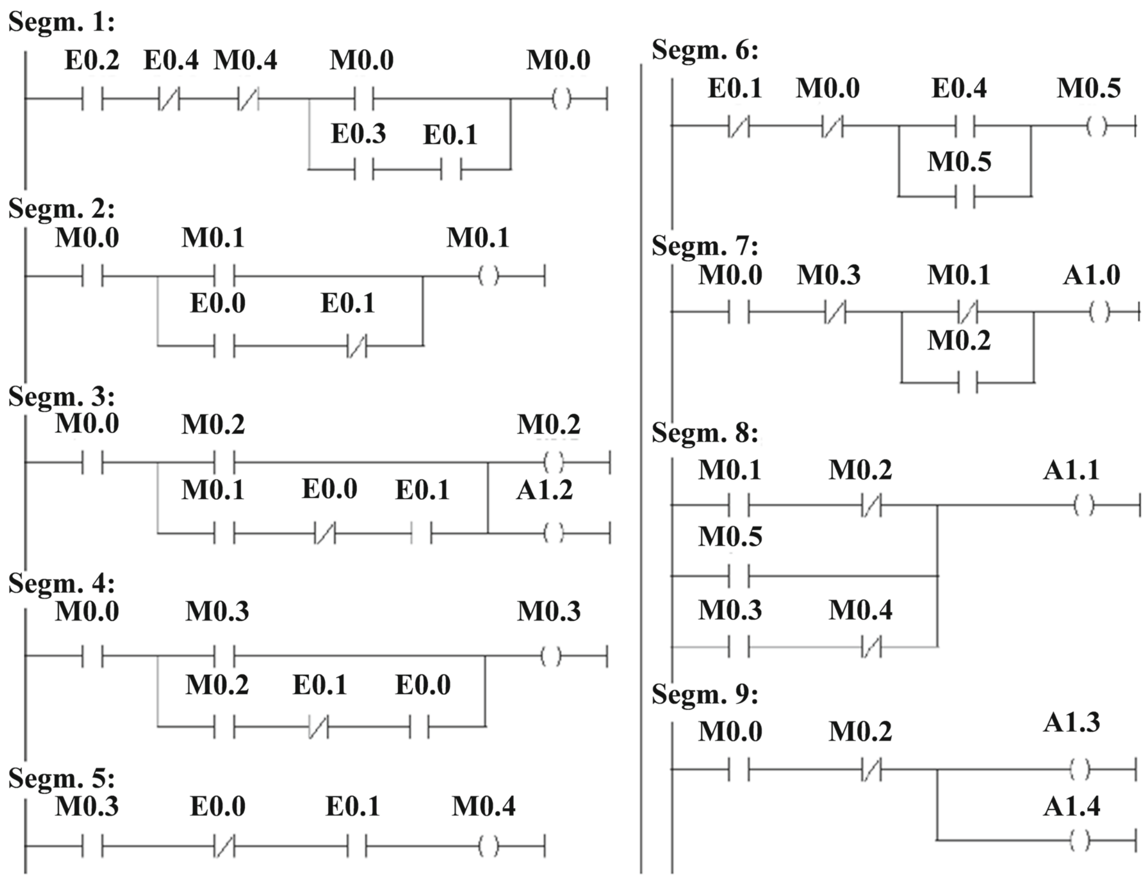


Fig. 8 Control algorithm of the carwash system

The machine formal specifications are:

1. The machine initial state is in the right limit (S2 activated),
2. Car in washing position (S3 activated), push the pushbutton M to start operation,
3. Machine must accomplish a go-and-back trip with the electro-valves XV and the brush motor MC in operation,
4. When the machine goes back to the right limit (S2 is activated again), it must accomplish another go-and-back trip in which only the fan motor MV is running. After the trip, the machine stays in its initial state,
5. If the stop pushbutton P is activated, the machine must automatically go back to its initial position.

Table 3 shows the variable assignation for physical input and output signals of the washing system. Variables of the LDPN definition are included.

The carwash system has five physical input signals, so that, based on Eq. 13, the possible fault number is $Ft = 132$, and by Eq. 14 it would be of $Ft = 992$. Combinations that, when summed to the initial markings M_0 of the places B_b and O_o, are the fault markings M_{ft} to be evaluated. The carwash system at initial conditions only has mark in the place I_1, corresponding to sensor S2 activated, indicating the machine is in the right limit. Based on the transforming approach LDPN, the corresponding networkof the control algorithm of the carwash system is obtained, as Fig. 9 shows, from which the reduced incidence matrix ar_{ij} of the system can be

Table 3 Addressing of physical input and output signals

Signal	Address	Description	LDPN
S1	E0.0	left limit switch	I_0
S2	E0.1	right limit switch	I_1
S3	E0.2	vehicle detection sensor	I_2
M	E0.3	start pushbutton	I_3
P	E0.4	stop pushbutton	I_4
MP1	A1.0	main engine left turn	O_0
MP2	A1.1	main engine right turn	O_1
MV	A1.2	drying fan	O_2
MC	A1.3	engine of the brushes	O_3
XV	A1.4	solenoid of the atomizer	O_4

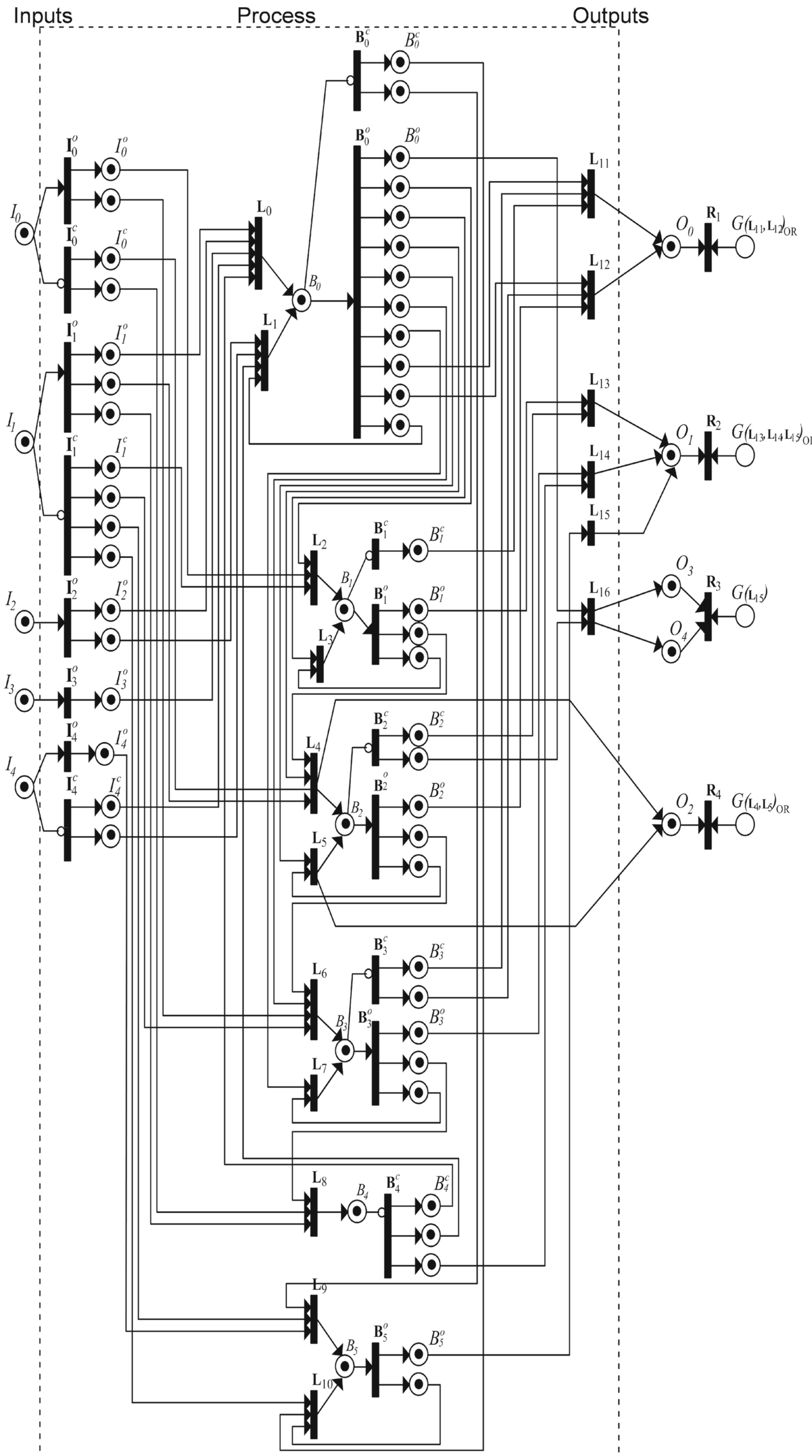


Fig. 9 LDPN control algorithm of carwash system

obtained, which is not presented by reasons of size and space.

Based on the results from the fault conditions evaluation, matrix from Eq. 16 shows the risk conditions. The open circuit fault does not generate marking at places O_o of system output.

$$\begin{bmatrix} I_0 & I_1 & I_2 & I_3 & I_4 & \ldots & O_0 & O_1 & O_2 & O_3 & O_4 \\ 0/1 & 0/1 & 0/1 & 0/1 & sc & \ldots & 0 & 1 & 0 & 0 & 0 \\ sc & 0/1 & 0/1 & 0/1 & sc & \ldots & 0 & 1 & 0 & 0 & 0 \\ 0/1 & 0/1 & sc & 0/1 & sc & \ldots & 0 & 1 & 0 & 0 & 0 \\ 0/1 & 0/1 & 0/1 & sc & sc & \ldots & 0 & 1 & 0 & 0 & 0 \\ sc & sc & 0/1 & 0/1 & sc & \ldots & 0 & 1 & 0 & 0 & 0 \\ sc & 0/1 & 0/1 & sc & sc & \ldots & 0 & 1 & 0 & 0 & 0 \\ sc & 0/1 & sc & sc & sc & \ldots & 0 & 1 & 0 & 0 & 0 \\ sc & sc & sc & sc & 0/1 & \ldots & 1 & 0 & 0 & 1 & 1 \end{bmatrix} \quad (16)$$

Where sc represents the short circuit fault. It can be observed that the place I_4, regardless of if other places have mark either by fault or normal system operation, it sets mark in the place O_1 corresponding to actuating the motor MP_2 moving the machine toward the right side, which it is a hazardous condition for both people and for the system. The stop signal P must completely stop the machine and not starting the motor toward the right side, which it will not stop if $S2$ has short circuit fault.

Furthermore, if places I_0, I_1, I_2 e I_3, they are in short circuit fault, a mark will be placed at the output places O_0, O_3, and O_4 corresponding to actuating the motor MP_1 moving the machine toward the left side, as well as the brush motor MC and of the energizing of the electro-valve XV, which it is also a hazardous condition for both people and for the system.

10 Conclusions

Having safe control algorithms for people as well as for the industrial machines or processes still remains a problem addressed by researchers from universities and research centers of proprietary firms related to the development of PLCs and their programming interfaces. Semantics and syntax of the interfaces cover the security aspects so that the control algorithm is executed on PLC; however, it still remains indispensable an updated and experienced knowledge of the responsible of designing the control algorithms in order to ensure the processes safety.

The validation proposal allows evaluating the behavior of the control algorithm in possible fault conditions of short circuit and/or open circuit in the physical input signals (sensors) in order to determine risk and/or danger conditions that may occur in the industrial process, and thus take the appropriate security measures before their implementation, or even if these are already implemented on the PLC-based systems.

As far as we know, control algorithms validation is mainly carried out based on theoretical concepts, such as, liveliness, coverability, among others. The presented validation approach is based on the possibility of that real faults (short circuit and/or open circuit) occur at the subsystem sensor-wiring-input module, of PLC-based systems, which allows predicting risk or danger conditions in industrial machines and processes.

Furthermore, it is important to evaluate the formal specifications of the processes in order to take security measures in fault conditions of the physical input signals, even though this could represent an additional cost due to having to consider more sensors.

References

1. International Electrotechnical Commision IEC61131-3 (2003) Programmable Controllers: Programming Languages, International standard, segunda edición
2. Barbosa H, Déharbe D (2012) Formal verification of PLC programs using the B method, Lecture Notes Computer Science, vol 7316, pp 353–356
3. Wang R, Zhou M, Yin L, Zhang L, Sun J, Ming G (2012) Modeling and validation of PLC-controlled systems: a case study. In: IEEE 6th international symposium on theoretical aspects of software engineering, pp 161–166
4. Grobelna I, Grobelny M, Adamski M (2010) Petri nets and activity diagrams in logic controller specification - transformation and verification. In: 607–612
5. Ljungkrantz O, Akesson K, Fabian M, Yuan C (2010) Formal specification and verification of industrial control logic components. IEEE Trans Autom Sci Eng 7(3):538–548
6. Park SC, Park CM, Wang G (2008) PLCStudio: simulation based PLC code verification. In: Winter simulation conference, pp 222-228
7. Thapa D, Park CM, Dangol S, Wang G (2006) III-phase verification and validation of IEC standard programmable logic controller. In: International conference on computational intelligence for modelling control and automation,and international conference on intelligent agents, web technologies and internet commerce
8. Hou Y, Cheng Q, Qiu A, Jin Y (2015) A new method of sensor fault diagnosis for under-measurement system based on space geometry approach. Int J Control Autom Syst 13(1):39–44
9. Bao J, Wu H, Yan Y (2014) A fault diagnosis system-PLC design for system reliability improvement. Int J Adv Manuf Technol
10. Huai L, Cheng C (2013) Reliability design of PLC-based control system. In: IEEE 9th international conference on natural computation, pp 1671–1675
11. Kuzmin EV, Sokolov VA (2013) On construction and verification of PLC programs. Autom Control Comput Sci 47(7):443–451
12. Qin S, Wang G (2012) A study of fault detection and diagnosis for PLC controlled manufacturing system. In: IEEE international conference semantic computating, Part I, pp 373–382
13. Wu Z, Hsieh S (2012) A realtime fuzzy Petri net diagnoser for detecting progressive faults in PLC based discrete manufacturing system. Int J Adv Manuf Technol 61:405–421
14. Malik AH, Mehmood T, Choudhry MA, Hanif A (2010) A generic procedure for troubleshooting of PLC based control systems. In:

IEEE 11th international conference control, automation, robotics and vision, pp 732–737
15. IEEE Recommended Practice for Validation of Computational Electromagnetics Computer Modeling and Simulations, IEEE Std 1597.2-2010,pp 1,124 (2011)
16. International Electrotechnical Commision IEC61131-1 (2003) General Information, International standard, segunda edición
17. International Electrotechnical Commision IEC61131-8:ProgrammableControllers (2003) Guidelines for the application and implementation of programming languages, international standard, segunda edición
18. Quezada JC, Medina J, Flores E, Seck Tuoh JC, Hernández N (2014) Formal desing methodology for transforming ladder diagram to Petri nets. Int J Adv Manuf Technol 73:821–836
19. Murata T (1989) Petri nets: properties, analysis and applications. Proc IEEE 77(4):541–580
20. Lee J, Lee JS (2009) Conversion of ladder diagram to petri net using module synthesis technique. Int J Model Simul 29 (1)
21. Mandado E, Acevedo JM, Fernández C, Armesto JI Autómatas Programables y Sistemas de Automatización, Segunda edición, Alfaomega, ISBN: 978-607-7686-73-6