A Simulation Tool Using AI Technics and Karnaugh Maps for Learning and Control Digital Pneumatics

António Ferreira da Silva
Research Center of Mechanical Engineering (CIDEM)
Mechanical Department, School of Engineering (ISEP)
Polytechnic of Porto,
Porto, Portugal
afs@isep.ipp.pt

Adriano A. Santos
Research Center of Mechanical Engineering (CIDEM)
Mechanical Department, School of Engineering (ISEP)
Polytechnic of Porto,
Porto, Portugal
0000-0001-8991-6768

***Abstract*—This paper describes a software, developed in PROLOG, called ISEPCPC (Intelligent System for Electric and Pneumatic Control of Pneumatic Circuits). It uses PROLOG clauses, Artificial Intelligence (AI) techniques (an Expert System), and Karnaugh Map's methodology. The main objective is to compute the set of optimized logic equations that enables either an electric or a pneumatic control of any, but mainly complex pneumatic circuits. As the complexity of the circuit increases, as the result of a combinatorial explosion generated by the increase in the number of cylinders involved, the more difficult it becomes to obtain the control equations (often requiring the use of auxiliary memories). This software automatically obtains the equations, which will yield minimized circuit configuration. These equations, can be used either in electro or pneumatic control. A practical application example of the software will be shown to prove its applicability. The minimum number of equations required to control the circuit will be used in programming a PLC that will control the circuit.**

***Keywords*—Prolog; Artificial Intelligence, Karnaugh maps, symbolic pattern manipulation, minimized logic equations, ON/OFF control, pneumatic sequential circuits.**

I. Introduction

The main goal of this paper is to describe an expert system ISEPCPC, that performs a computational symbolic manipulation of Karnaugh Maps (KM) [1], to obtain the set of optimized logical equations that enables the control of pneumatic or electro-pneumatic circuits. The symbolic manipulation of a KM is governed by a few sets of generic rules [2] for signal flow plotting and for logic equation minimization/optimization that are particularly applicable in controlling complex pneumatic circuits. This complexity comes from the combinatorial explosion that results from the increase in the number of cylinders that constitutes the cycle to automate [3]. The symbolic learning approaches was implemented performing a search in the space of symbolic patterns of the KM. This search can use pruning and, obtain the optimal solution for controlling the desired pneumatic or electro-pneumatic circuit.

In developing ISEPCPC method computes of the optimal control unit based on the Karnaugh mapping method. The rules for flow path plotting and the rules for looping to derivate the minimized logic equations are applicable for complex control circuits. The operation of these rules involves the manipulation of symbolic patterns so that the KM can be derived automatically [4]. From these logic equations, it is quite simple for the designer to build the control part of the pneumatic/electro-pneumatic circuit.

An Expert System (ES) developed in Prolog [5], allows the creation of the KM, and obtaining the smallest set of logical command equations. A description of the proposed methodology, as well as its applicability, will be illustrated by a practical example. The applicability of the proposed software, either in industrial or educational environment will be demonstrated.

II. Pneumatic Sequential Processes

The classic methodologies of electro-pneumatic control are essentially based on the Cascade method and a methodology supported by Karnaugh Map [3]. In the first method, the resolution of pneumatic problem is basically performed by dividing the working sequence in groups. In the second method, the approach taken minimizes each of the control logic equations of the set, converting the truth table, to the corresponding logical circuit [2].

In pneumatics, one generally needs to automate sequential processes. In this cases, control of the power parts (cylinder) is not only depending on the state of the present information but also in terms of knowledge of past actions. In practice, ensure a series of sequential movements in an automatic cycle is to find the set of equations witch, when executed ensure the realization of the desired sequence. The application of these methods makes it unnecessary to analyse the speed or course of stakeholder's cylinders (irrelevant features to the materialization of the command sequence). It means therefore that the control equations of the cylinders that will allow their advance and their indentation (called output variables) are function only of two types of variables (input variables) and positioning each intervening cylinder (signal end-of-course) and if necessary memory variables.

III. Karnaugh Maps

KM are often, used to simplify and minimize Boolean functions. Industrially KM can be used to develop control for sequential pneumatic or electro pneumatic circuits, because it is simple to pass from the obtained equations to the implementation phase. In [6] an alternative way to use the KM are proposed, that minimizes the logical conditions of

We acknowledge the financial support of CIDEM, R&D unit funded by the FCT – Portuguese Foundation for the Development of Science and Technology, Ministry of Science, Technology and Higher Education, under the Project UID/EMS/0615/2016.

978-1-7281-3637-0/19/$31.00 ©2019 IEEE

command for pneumatic circuits, whether the command is pneumatic or electric. In the last case, the method also permits the control of the circuit through a programmable controller (PLC).

From a pneumatic point of view, a KM is a table of cells, which allows the representation of the sequential movements of the cylinders involved in a circuit. Cell addresses are a function of the cylinder position (sensors) and memories. Auxiliary memories are necessary when we need to discriminate between identical position sequence valve combinations [4]. The table, is a convenient representation of the movement sequence of cylinders, and to obtain the control part (see Fig.6). The control part is the result of the minimization of the logical equations obtained from de map, establishing a relation between sensors, memories and cylinders orders. For each cylinder of the sequential circuit, we will obtain a function. For example, in the case of the movement A1 (cylinder A advance) the relation is expressed by A1 = f (b0, b1, c0, c1,…, x0, x1,…).

The dimension of the map is dependent of the number of cylinders involved in the circuit. The minimum number of cells is 2^N where N is the number of cylinders, and if necessary auxiliary memories. A KM for four cylinders and two memories is represented in Fig. 1.

				y0		y1	
				x0	x1	x1	x0
d0	c0	b0	a0	1	2	3	4
			a1	5	6	7	8
		b1	a1	9	10	11	12
			a0	13	14	15	16
	c1	b1	a0	17	18	19	20
			a1	21	22	23	24
		b0	a1	25	26	27	28
			a0	29	30	31	32
d1	c1	b0	a0	33	34	35	36
			a1	37	38	39	40
		b1	a1	41	42	43	44
			a0	45	46	47	48
	c0	b1	a0	49	50	51	52
			a1	53	54	55	56
		b0	a1	57	58	59	60
			a0	61	62	63	64

Fig. 1. KM for four cylinders and two memories (64 cells)

Each cell corresponds an only specific position of all the cylinders involved. For example, translating cell 49 corresponds in term of cylinders positioning at a0, b1, c0, d0, that is cylinders A, C and D in returned position, and cylinder B is advanced position.

The rules to flow-path in the map are [4]:

- Occupy a cell in the table only once;
- When a movement (A0, A1, B0, B1, etc.) falls in an already occupied field, the map must be enlarged to the double the existing number of columns. The new columns represents the auxiliary memory;
- Moving a memory, corresponds to moving to another vacant cell (maintaining all the cylinder's positions), and change only the state of the memory. If the memory is X, this new movement is designed by X1 (only changes x0 to x1);
- The memory reset must be done after prioritizing the movements of the cylinders;
- The final state (cell) must coincide with the initial state.

IV. Optimization of Logic Equations

After the map construction, the second and must important part, is to obtain the logic equation minimization, the core of this methodology. To do that, we need to apply a couple of rules (four), to obtain the logic control equations for each cylinder of the cycle (including if necessary the memories).

For the minimization/optimization of the logic equations, we need to choose the maximum number of cells that do not violate any of this four rules:

- The number of cells chosen must be a power of two, that is 2^0, 2^1, 2^2, etc.;
- Empty cells are jokers, that is, may be chosen anytime we want;
- Any occupied cell in the path between the opposite movement we want to control, and the cell that contains the movement is prohibited. Suppose that we want to find the equation for A0. All the occupied cells in the path, between the cell containing A1 and A0 cell is prohibited;
- The movement before the movement we want to do, must be represented in the logical equation of the next movement.

The correct application of these rules allows to determine, for each cylinder (and each eventual memory requirements), to find the write optimal/minimized equations that control each cylinder.

V. Case Study Using ISEPPCPC

The case study involves an electro-pneumatic pick-and-place (Fig.2) used in the quality test of locks in LARI (Laboratório de Automação e Robótica Industrial of ISEP). The system consists of four actuators one for position (cylinder A, horizontal) one to make the approach (cylinder B, vertical) and two other cylinders C and D that give body to the test. C is responsible for the rotation (rotation cylinder) and D for the gripper of the key. After test pick-and-place return to initial position, return of the cylinder A-. An operating sequence may be: A+\ B+\ D-\ C+\ C-\ D+\ B-\ A-.

Fig. 2. The pneumatic manipulator existing in LARI

The corresponding electro-pneumatic circuit is represented in Fig. 3.

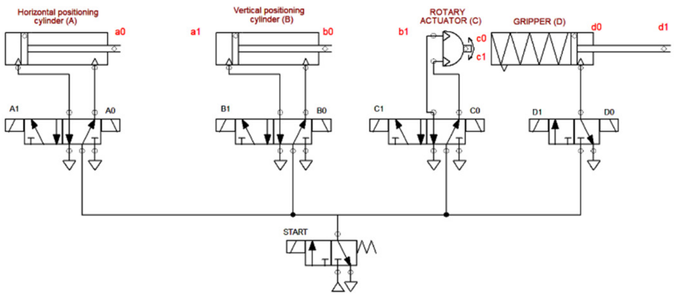

Fig. 3. Electro-pneumatic circuit of the pneumatic manipulator

The KM methodology can quickly calculate the necessary logic equations for programming the PLC that will control the circuit. The procedure is as follows:

- Calculate the number of cylinders in the problem;
- Build the KM for the necessary cylinders;
- Step by step fill the KM with the given sequence;
- Find the minimized logic equations that controls the cylinders;
- Implement the PLC program
- Testing the proposed solution.

So to follow this methodology we need to:

a) Calculate the number of cylinders in the problem.

In this problem as we can see from Fig.3 we need to control four double acting cylinders.

b) Build the KM

It is necessary a KM for four cylinder with 2^4 address fields (without counting eventual needs of memories).

c) Step by step fill the KM with the pretended sequence

Step by step we need to fill the address fields in the KM, represented the necessary movements, and see if memories are needed. The design of the pretended map is shown in Fig. 4.

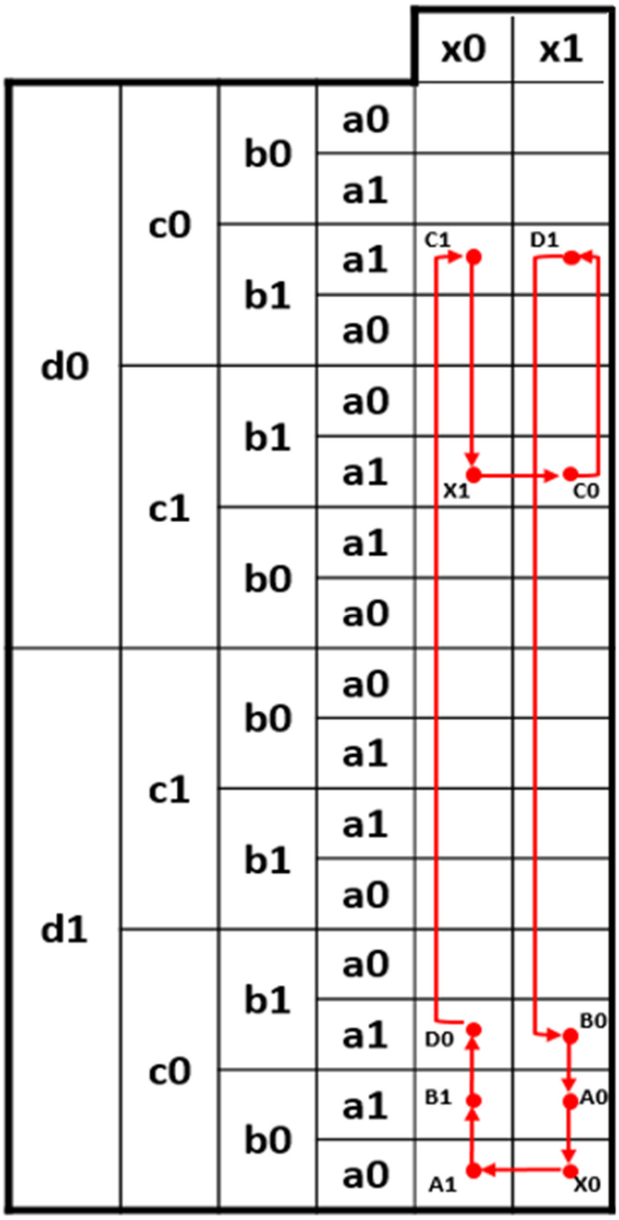

Fig. 4. Karnaugh map for the sequential path

As we can see from Fig. 4, a memory (X) is necessary, to implement the necessary sequential path.

d) Find the minimized logic equations

For controlling the cylinders (and the memory), we need to use the KM to find the minimized logic equations that controls each cylinder and the memory, using the previously described methodology.

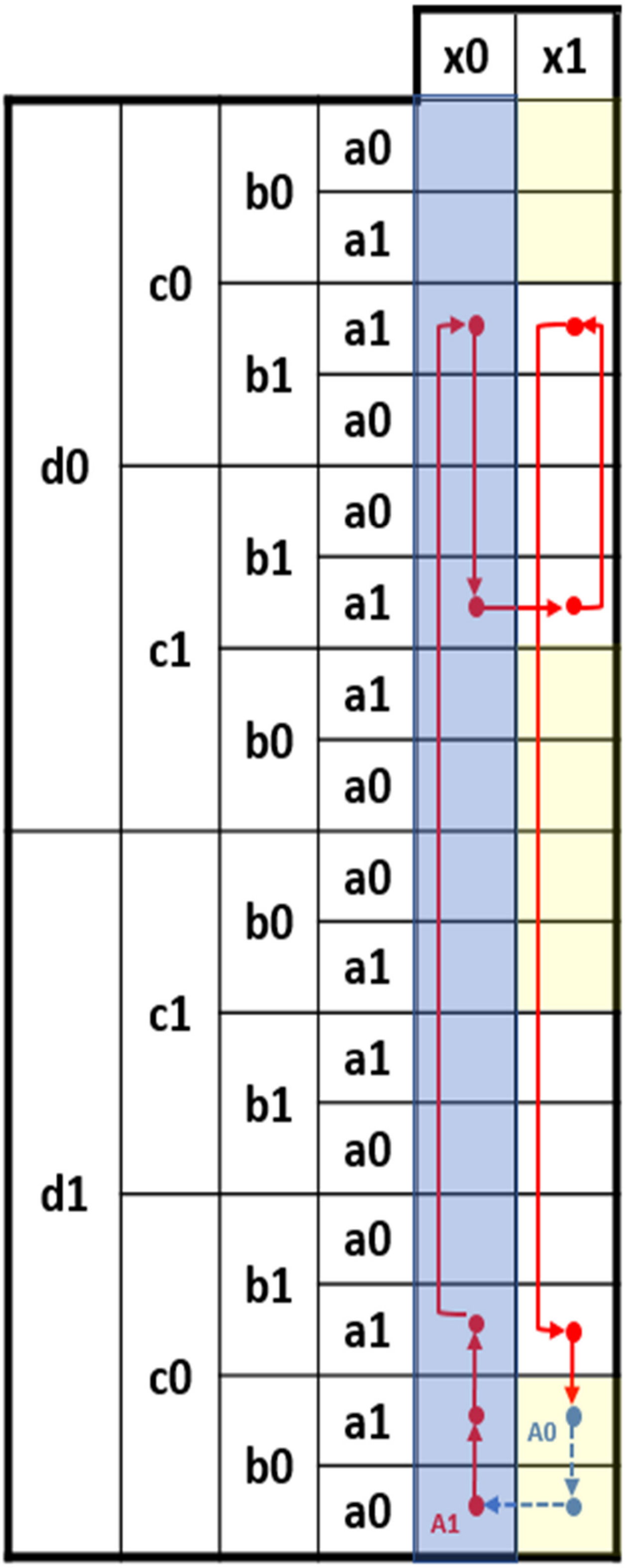

Fig. 5. Karnaugh map for the sequential path (4 cylinders and a memory)

As an example Fig. 5 shows the way to find the areas for implementing the movements of A cylinder (A1 and A0).

Using the same methodology, for all the cylinders and the memory, the final set of equations are shown in (1):

$$\begin{cases} C_1 = d_0.x_0 \\ C_0 = x_1 \end{cases} \quad \begin{cases} D_1 = c_0.x_1 \\ D_0 = b_1.x_0 \end{cases} \quad \begin{cases} X_1 = c_1 \\ X_0 = a_0 \end{cases} \tag{1}$$

VI. Prolog Implementation

The description of the rules that allow, to draw the KM, discover the flow-path that represents the desired sequence of movements and finding the minimized logical equations to control the cylinders, needs the manipulation of symbolic signs and checking for membership of a particular group. Almost all of these situations requires pattern matching of symbols [8]. PROLOG is a programming language with such capabilities.

The construction (dimension) of the KM is only dependent on the number of cylinders and memories. Implementing the rules for flow-path requires that appropriate activated sensors combinations be selected. It is necessary to initiate the path (starting cell), filling the KM in a chronological order and return to the beginning to form a loop, see pseudocode in Fig.6.

```
INPUT: startCell, Karnaught Map, Movements Sequence
OUTPUT: The path (sequence of cells) from initial cell to end cell
   Movements = List of required movements
   Path = Empty
WHILE Movements NOT Empty DO
   MOV = NextMove(MOVEMENTS)
   MOVEMENTS = MOVEMENTS – MOV
   NextCELL = findNextCell(ListOfPossibleMovements, Path)
   IF no possible NextCell THEN
        createNewMemory()
   ELSE
      Path = Path + NextCell
ENDWHILE
IF in FinalCell() THEN
   RETURN: Path
ELSE
   GoTo(EndCell)
   RETURN: Path
```

Fig. 6. Pseudocode to finding the loop path

VII. Implement the PLC program

The program was made in Ladder Logic, using a S7-200 PLC from SIEMENS.

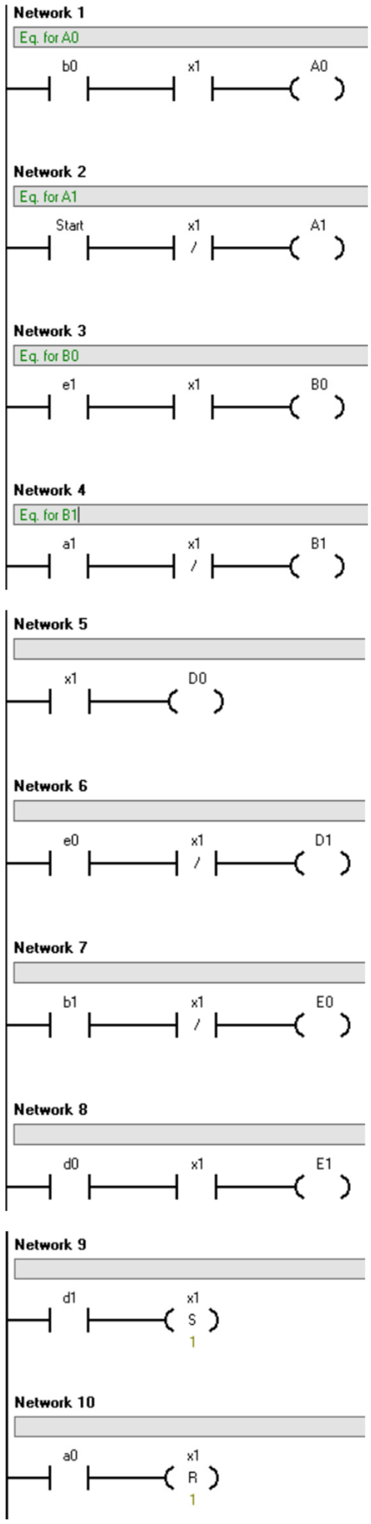

Fig. 7. PLC program in Ladder language

As we can see in Fig.7, the implementation in Ladder language is a pure translation of the logical equations obtained by the KM [7].

VIII. Conclusions

Technological resources used in automation are many and often diverse. It is a constantly changing area. The numerous technological features that potentially allow support automation are extremely diverse and subject to deep and constant changes. In this sense, the main contribution of this work is to use the potential of AI methodologies like symbolic learning approaches, to increase the potential of using traditional pneumatic control methodologies. The methodology described here, will extend its use not only in industrial applications, but also in the educational level in terms of engineering schools allowing a particular use in the automation of laboratories in the field of pneumatic control circuits and sequential electro (it is currently teaching in automation course unit of mechanical engineering in ISEP).

References

[1] Karnaugh, M.,"The map method for synthesis of combinational logic circuits', in Transactions of the American Institute of Electrical Engineers, Vol. 72, No. 1, pp.593–599, 1953.

[2] Santos, A.A. and da Silva, A.F., "Methodology for manipulation of Karnaugh maps designing for pneumatic sequential logic circuits", in International Journal of Mechatronics and Automation, Vol. 6, No. 1, pp.46–54, 2017.

[3] Santos, A.A. and Silva, A.F., Automação Pneumática, 3rd ed., Publindústria, Portugal, 2014, pp. 185-243.

[4] Sim, S-K. and Chua, P.S.K., "Symbolic pattern manipulation of Karnaugh-Veitch maps for pneumatic circuits", in Artificial Intelligence in Engineering, Vol. 10, No. 1, pp.71–83, 1996.

[5] Bratko, I., "Fast prototyping of expert systems using PROLOG", in Topic in Expert System Design, ed. G. Guida & C. Tasso. Elsevier Science Publishers Amsterdam 1989, pp. 69-86.

[6] Silva, A.F. and Santos, A.A., "Teaching control pneumatic and electro-pneumatic circuits – a new method", in International Congress on Education, Innovation and Learning Technologies (ICEILT2015), September 2015, Granada, Spain.

[7] Santos, A.A. and Silva, A.F., "Electro pneumatic control versus grafcet", in International Congress on Education, Innovation and Learning Technologies (ICEILT2015), September 2015, Granada, Spain.

[8] Sim, S-K. and Chua, P.S.K., "Symbolic pattern manipulation of Karnaugh-Veitch maps for pneumatic circuits", in Artificial Intelligence in Engineering, Vol. 10, No. 1, pp.71–83, 1996.