ABCM Symposium Series in Mechatronics - Vol. 4 - pp.480-489

AUTOMATIC TRAFFIC CONTROL STRATEGY FOR SINGLE TRACK RAILWAYS IMPLEMENTED IN SEQUENTIAL FLOW CHART ON AN INDUSTRIAL PROGRAMMABLE CONTROLLER

Tiago Romeiro de Jesus, tiagoromeirodejesus@hotmail.com
José Adriany Victor de Aquino, victorunbmkt@gmail.com
Guilherme Caribé de Carvalho, gccarval@unb.br
Carlos Humberto Llanos Quintero, llanos@unb.br

Faculdade de Tecnologia, Universidade de Brasília, Brasília-Distrito Federal

Abstract. *This work presents the development and implementation of a traffic control strategy for controlling the assignment of driving rights to two miniature locomotives (1:87 scale) running on a single track railway at opposite directions. The strategy was modelled using a Petri Net approach and was implemented on an industrial programmable controller, utilizing the sequential function chart (SFC) language. For testing the control strategy, an experimental apparatus, based on a model railroad miniature, was implemented. The electrical miniature locomotives are energised via their pair of railroad trucks, thus collecting the power from the electrified rails. Based on this characteristic, a closed circuit miniature single track railroad, including two passing loops, two parking tracks and six turnouts connecting the main track to the passing loops and to the parking tracks, was devised and assembled in such a way as to divide the railroad in several different sections with independent power sources. The energising of each track section as well as the passing loops was controlled by a programmable logic controller (PLC), as a result of the state of the different sensors used to monitor the condition of the railroad. The control strategy was devised so that one train, at most, may be in a section at any time. Each locomotive has its independent SFC implementation such that no priority was predefined for any of the trains. The track section allocation was realized on a first to request basis, thus giving the occupation authorization to the first locomotive that would request it. The system was tested by programming it to control the switch of position of the locomotives, which were parked, as an initial state, each one at one of the parking tracks. Different numbers of loops in the closed circuit track were programmed for each locomotive and good results were obtained with the control strategy, which adequately prevented the occurrence of head-on collisions, always stopping one of the trains at a passing loop while the other train was passing on the main track.*

Keywords: *railway automation; single track railway traffic control; SFC programming; Petri Nets; Programmable Controller.*

1. INTRODUCTION

Many different approaches can be used for controlling underground railways traffic. These generally consist of isolated systems in terms of physical implementation, but intimately connected if a macro-operational point of view is used. Among such approaches, some can be outlined: regulating train headways, for example by means of defining time intervals (Assis, 2002); using safety principles such as block violation control, interlocking control; track occupancy and yard limits control; and strategic planning approach, such as planning train schedule, including traveling speeds at each block in such a way as to restrict the block authority to the train scheduled to that specific timetable (Tazoniero, 2007). Model parameters such as productivity, cost and time consist in variables that impose a high level of complexity in the train scheduling and control in real cases. The optimal planning of train dispatch involves the solution of complex NP-Complete optimization problems (Garey e Johnson, 1979). According to (Tazoniero, 2007), the search for optimal solutions when dealing with real time control problems in a railway system is not always feasible, since optimization problems of such a type demand a considerable computational effort, which demands time that is not always available. Such processing time cold result in loss of railway traffic information since the measured variables could have changed their state, thus leading to non-optimal or to a wrong solution for the new situation. This could result in instability problems which could imply in loss of control. In attempting to better understand and solve this kind of problem, some authors have adopted some discrete event modelling techniques for simulation purposes (Peterson and Taylor, 1982). Due to the complexity involved in such problems, artificial intelligence (Cherniavsky, 1972) and fuzzy logic (Gomide, 1999) have been applied in order to solve them. The use of Petry nets for studying railways typical processes in a specific model have been reported by Tang, Chen e Xiao (2000). This work presents a

ABCM Symposium Series in Mechatronics - Vol. 4 - pp.480-489

railway automation project, which deals with the main safety aspects in a specific railway model. Further to modelling the system in Petri Nets, a Central Traffic Control system was developed and implemented in an industrial programmable controller (Allen Bradley ControLogix 5550), using *SFC* and *ladder* IEC graphical languages. The developed control algorithm was applied to controlling a physical apparatus built, as a closed circuit single track railroad, with to two passing loops and two parking tracks. A supervisory system was also developed in RSView in order to allow the visualization of the concurrency problems and also to evaluate the capability of the developed control system to deal with an actual potential conflict caused by the sharing of a single track railroad between two opposite direction running locomotives.

2. PROBLEM DESCRIPTION

Based on the geometry of the model railroad shown in ***figure* 1**, several discrete sensors where chosen to monitor the state of the tracks and the locomotives. The track sensors were allocated in key positions (see ***figure* 1**) in order to allow the development of a discrete controller to be implemented in a programmable controller.

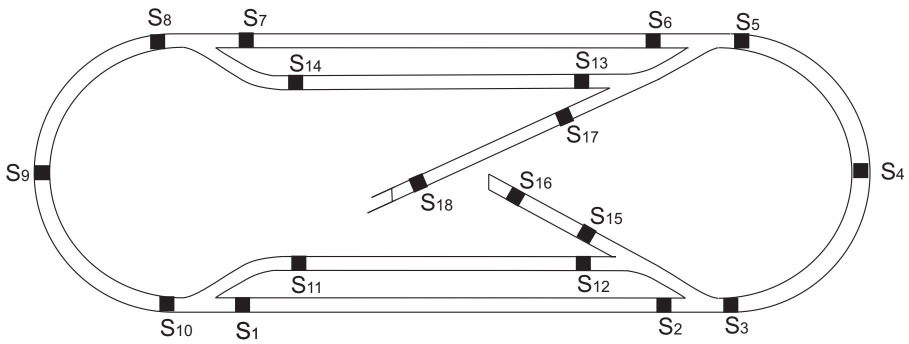

Figure 1. **Schematics of the model railroad geometry outlining the discrete sensor positions**

In ***figure* 1** each sensor is called S_i, such that the subscript i is an integer and ranges from 1 to 18.

***Figure* 2** shows the final configuration of the simulation aparatus in terms of railroad blocks (sections into which the railroad was divided for measurement and control purposes). The blocks composed of single tracks were called Tv_j, where j is an integer ranging from 1 to 8. The turnouts or switches were called Tm_k, such that k ranges from 1 to 6. The model railroad kit used to build such an apparatus is based on electrically driven locomotives whose power is collected from electrified conductive tracks. Based on this characteristic, the approach used in this work, to divide the railroad into blocks for control purposes, was to electrically separate the corresponding railroad sections and to provide, in a controlled manner, independent power sources for each one.

Based on this approach, four different power sources, with voltage levels at $12V$, $-12V$, $5V$ e $-5V$, were used to power the tracks. Obviously, such a connection could only be carried out in a excluding manner: only one voltage level could be applied to each block at any time. Those voltage levels were chosen such that the locomotives could have two speed levels in both directions of movement, clockwise or anti-clockwise. All the control tasks, including the connection of the power sources to the correct tracks, the control of the turnouts and the monitoring of the state of the model railroad were carried out by means of an Allen Bradley ControLogix 5550 programmable controller. The plant was monitored through digital and analogue input modules and was controlled through relay output modules.

The analogue input module was used to monitor the voltage levels applied to each block as a result of the control actions. This was necessary during the programming phase of the system, in order to make it easier to detect programming errors.

The location of the locomotives was made possible by installing under each one a set of magnets, which would activate reed switch sensors distributed on the circuit according to the scheme shown in ***figure* 1**. Such sensors were coupled with switching electronic circuits (transistor circuits) in such a manner as to produce a pulsed signal whenever a locomotive with a set of magnets would pass over them. Each reed switch sensor was connected to an input on the digital input modules

The turnouts are denoted by $M_{t,p}$, such that p can assume the values n, for normal position, or r, for reverse position, and t is a natural number ranging from 0 e 7. They are actuated through a relay output module, by means of discharging a capacitor on one of the turnout coils, which produce in turn the force needed to position the turnout in one of its two possible states: normal or reverse. Normal position, $M_{t,n}$, corresponds to the locomotive passing straight through the turnout, whereas the reverse position, $M_{t,r}$, deviates the locomotive to

ABCM Symposium Series in Mechatronics - Vol. 4 - pp.480-489

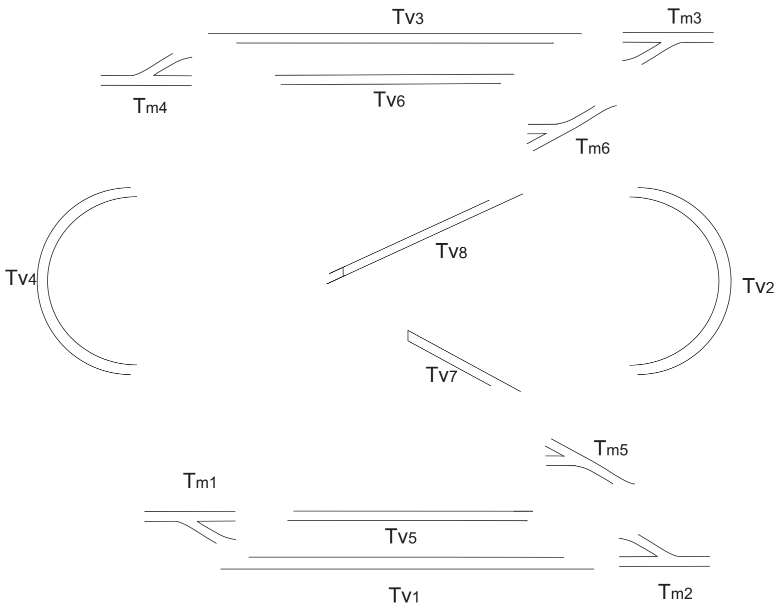

Figure 2. **Schematic drawing of the model railroad according to the blocks made of single tracks, Tv_j, and the turnout blocks, Tm_k**

a branch or to a passing loop. The distribution of the turnouts along the circuit is shown in ***figura* 3**.

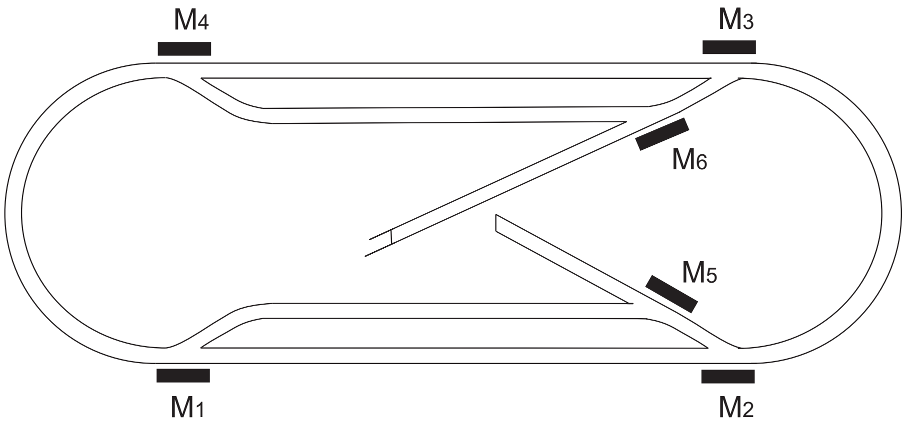

Figure 3. **Schematic drawing according to the turnout positioning along the model railroad.** M_t

After finishing the physical implementation of the experimental apparatus, control criteria were devised based on the identification of the possible plant states which could be allowed to exist and also based on the sensory data that would be used to detect block occupation and block requisition. These information were organized in such a way as to allow the use of combinatorial logic to solve some of the control problems.

The modelling and the devised control strategies were implemented using the *SFC* and the *Ladder* graphical IEC programming languages. The use of one or another or both in each part of the software development work was decided based on the requirements of each routine, in terms of performance, and also, on friendliness e robustness offered by each language.

The problem simulated in this work consists in changing the positions of the two locomotives located initially at Tv_7 and Tv_8 after individually dispatching each locomotive by means of issuing independent start commands. Each locomotive is required to attain a pre-defined number of loops in the closed circuit, before being directed to the destination parking track. Such a problem was devised in order to produce the typical resource sharing and concurrency problems which would have to be treated before a secure exchange of position could be made.

3. THE PROPOSED PROBLEM SOLUTION

Main characteristics of the problem

In the study of this problem was detected the sequential nature of the execution of the different steps of the process . This is because there exists a set of blocks Tv_j and Tm_k; and locomotives positions $M_{t,p}$, in which

ABCM Symposium Series in Mechatronics - Vol. 4 - pp.480-489

the locomotive goes in order to transfer from Tv_7 position to Tv_8 one and viceversa. For instance, one can consider the displacement of a locomotive from Tv_7 to Tv_8 without at least finishing a cycle in the road, due that the same consideration it can be applied to any number of completed circuits. This displacement can be carry out independently of the another locomotive. To do that, the only one possibility would be the sequential occupation of the blocks Tv_7, Tm_5, Tm_2, Tv_2, Tm_3, Tm_6 and Tv_8. The locomotives will follow the sequence $M_{6,n}$, $M_{5,n}$, $M_{2,r}$, $M_{3,r}$ and $M_{6,n}$. The respective actions that results for achieving the the displacement have the sequence as following:

1. turn on the current sector;
2. turn on the next sector;
3. turn on the locomotive for one of the *normal* positions or *reverso*;
4. turn off the previous sector;
5. turn off the last locomotiva(s) de chave percorrida(s);

The execution of the related actions depends on both the Boolean funcions of the detection sensors and the activaton block signals. Fig. ***figura*** 4 depicts the locomotive activity sequence for translating from Tv_7 to Tv_8.

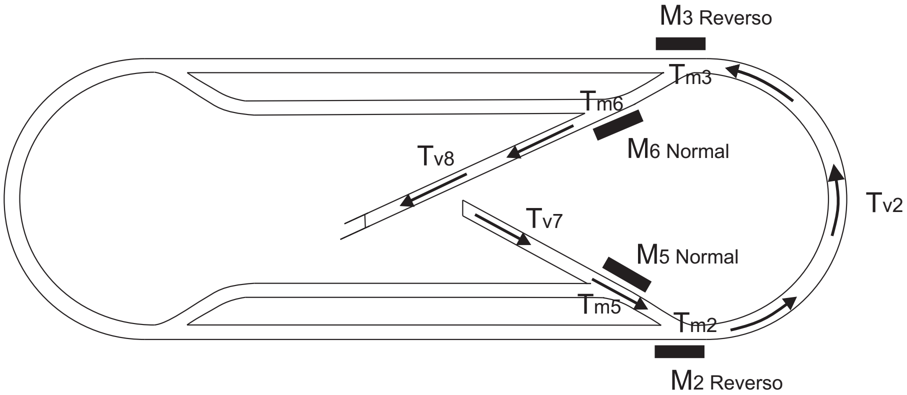

Figure 4. **Esquemático segundo o deslocamento da locomotiva do Tv_7 para o Tv_8**

3.1 Petri net modeling

Figure ***figura*** 5 depict the proposed system considering the possibility of displacement in the reverse way. Then, resource concorrence is explicity in this modeling, given that one can consider the presence of two trains in blocks Tv_7 and Tv_8, both of them with pointing out for traveling the same blocks, however in oposite ways.

It can be observed in the implemented Petri net a symmetry based on a imaginary central vertical axis, showing the absence of priority static determination. Basically, the the decision is defined by the time in which the resources are required.

The network has three vertical columns and the external ones are modeling the execution the respective action for each directions, exposing the occupation/activation of the required blocks and the train circulation over the detection sensors. The central column shows the unified evolution of the occupation and unoccupation in both ways.

A tabela 1 organiza os lugares da Rede de Petri expressa na ***figura*** 5 segundo seus rótulos e funções.

The main objective of this Petri net modeling approach was the verification and easier utilization of the input and output resources and the scheduling of the the main system objects. Using a environment simulation *Visual Object Net* it can be observed that with only the initial inputs and initial block releases (token settings in the suitable places) the net stabilization only leading with a appropriated concurrence. With regard to the collision treating it can observe the presence of the tokens in the places Tv_2, Tv_7 and Tv_8 depicts the evidence of a real collision, and this is no really certificated in the project simulation.

We can emphasize that the implemented model does not represent the overall project. Although, the corresponded used logic in the implementation of the overall case can be accomplished by mean of the expansion of the specific model. Some places related to the controller outputs were deliberately duplicated in order to make easier the visualization of the network, apart from highlighting its symmetric characteristic. This places are not simultaneously actives in any simulation time, indicating the exclusivity of these resources.

Proceedings of COBEM 2009

20th International Congress of Mechanical Engineering
November 15-20, 2009, Gramado, RS, Brazil

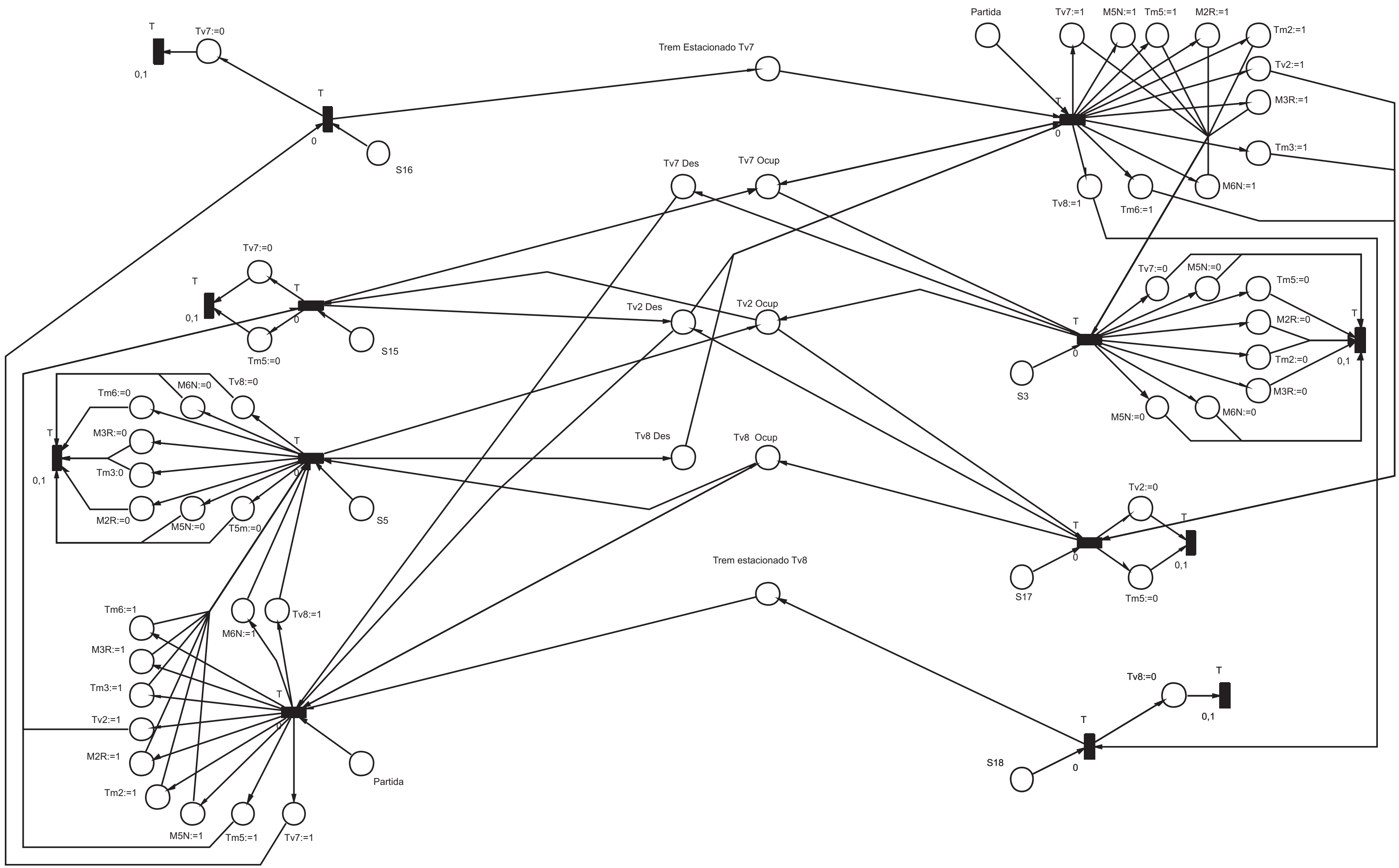

Figure 5. Rede de Petri

ABCM Symposium Series in Mechatronics - Vol. 4 - pp.480-489

Input		**Output - Blocks**		**Output - Turnouts**	
Place	**Description**	**Place**	**Description**	**Place**	**Description**
$S3$	Sensor 3 enable	$Tv_2 := 1$	Energizes Tv_2	$M6R := 1$	Energizes $M_{6,r}$
$S5$	Sensor 5 enable	$Tv_7 := 1$	Energizes Tv_7	$M2N := 0$	Unergizes $M_{2,n}$
$S15$	Sensor 15 enable	$Tv_8 := 1$	Energizes Tv_8	$M2R := 0$	Unergizes $M_{2,r}$
$S16$	Sensor 16 enable	$Tm_2 := 1$	Energizes Tm_2	$M3N := 0$	Unergizes $M_{3,n}$
$S17$	Sensor 17 enable	$Tm_3 := 1$	Energizes Tm_3	$M3R := 0$	Unergizes $M_{3,r}$
$S18$	Sensor 18 enable	$Tm_5 := 1$	Energizes Tm_5	$M5N := 0$	Unergizes $M_{5,n}$
$Part.Hor$	Clockwise Start	$Tm_6 := 1$	Energizes Tm_6	$M5R := 0$	Unergizes $M_{5,r}$
$Part.Anti.Hor$	Anticlockwise Start			$M6N := 0$	Unergizes $M_{6,n}$
Output - Blocks		**Output - Turnouts**		**Output - Turnouts/occupation**	
Place	**Description**	**Place**	**Description**	**Place**	**Description**
$Tv_2 := 0$	Unergizes Tv_2	$M2N := 1$	Energizes $M_{2,n}$	$M6R := 0$	Unergizes $M_{6,r}$
$Tv_7 := 0$	Unergizes Tv_7	$M2R := 1$	Energizes $M_{2,r}$	Tv_2 Ocup	Tv_2 Occupied
$Tv_8 := 0$	Unergizes Tv_8	$M3N := 1$	Energizes $M_{3,n}$	Tv_2 Des	Tv_2 Unoccupied
$Tm_2 := 0$	Unergizes Tm_2	$M3R := 1$	Energizes $M_{3,r}$	Tv_2 Ocup	Tv_7 Occupied
$Tm_3 := 0$	Unergizes Tm_3	$M5N := 1$	Energizes $M_{5,n}$	Tv_2 Des	Tv_7 Unoccupied
$Tm_5 := 0$	Unergizes Tm_5	$M5R := 1$	Energizes $M_{5,r}$	Tv_2 Ocup	Tv_8 Occupied
$Tm_6 := 0$	Unergizes Tm_6	$M6N := 1$	Energizes $M_{6,n}$	Tv_2 Des	Tv_8 Unoccupied

Table 1. Tabela dos principais *lugares* da Rede Petri

4. CODES AND RESULTS

The overall project consists of the previous placement of the locomotives in the blocks Tv_7 e Tv_8. The same locomotives must cover an arbitrary cycle number (not necessarily the equal cycle number) and then parking in the opposite blocks Tv_7 e Tv_8. For each direction was designed corresponded sequential map (clockwise and counterclockwise directions). A em *ladder* code supports the following aspects with respect to the project maps : (b)external interface, signal standardization coming from the detection circuits, (c) the utilization of the counter and (d) simultaneous resource sharing.

The flux of the states happens in an independently way when the map executions are started. The action sequence obey to a general standard similarly to the cited case the section xxx.

The SCF code is shown in ***figure*** 8.

The achieved SCF state tree has states, transitions and actions. The states represent steps to which actions are associated. The transition to new state is depending on a logic function, which is related to a transition.

In this project the transitions are related to (a) a voltage level, (b) the detection sensors and (c) internal variables of the map. The voltage levels are variables from the controller that are gathered thought the interface module analogous input. In order to defined a Boolean value of the respective functions the specific criterion were defined: (a) level between $0V$ e $1V$ are considered as logic 0(zero) or unenergized, (b) Other levels are considered as being logic 1(one) or energized.

The displacement logic of the locomotives is achieved through energizing requirements. The transition of a locomotive to an adjacent block this is depending if the respective block is energized. The process of energizing the block can be related to both a really presence of the locomotive or future occupation. The voltage level verification of a required block leads to two possibilities:

1. Unenergized: the block is free. Viabilization commands to perform the transition from the current block to the adjacent block are ommited.

2. Energized: Waiting to be liberated.

There are several aspects in the implemented logic. For instance, in the case of sensors S_3 e S_8 (counterclockwise case), S_5 e S_{10} (clockwise case), the liberation mechanism is similar to the early described one. However the next transitions evaluate mores aspects than the only energizing of the blocks Tv_1 e Tv_3. This is due to fact that there exist also the possibility to bypass themselves through the blocks Tv_5 e Tv_6 , apart from going away them, whenever the blocks (Tv_1 ou $Tv3$) are occupied or required.

Therefore, there exist more than one possibility apart from waiting. Then the locomotives are alocated depending on the new alternative route, apart from the respective energization commands.

Additionally, it can observed the transitions related to S_1 e S_7 sensors. In this case, the management of the cycle counter in both ways that is related to the detection circuits S_8 (clockwise) e S_7 (counterclockwise) is accomplished in the final step of the displacements. This verification has a energizing analysis of the Tv_7 e Tv_8 blocks, apart from the presence of the locomotives in this areas, through the S_{16} and S_{18} sensors. This due to the fact that S_1 e S_7) sensors are shown as the last opportunity to decide the wait state of a train in a cross-rail. In this case, *Karnaugh maps* were used given the complexity of the functions for simplifying tasks.

The Table 2 classifies the sensors according to the corresponding blocks, the direction of movement and analysis variables.

Sensor	Evaluated Block	Direction	Analysed variable
S_1	Tv_2 e Tv_8	anti-clockwise	voltage level, counter and static presence
S_2	Tv_4	time	voltage level
S_3	Tv_3	anti-clockwise	voltage level and static presence
S_5	Tv_1	time	voltage level e static presence
S_6	Tv_4	anti-clockwise	voltage level
S_7	Tv_2 e Tv_7	time	voltage level, counter and static presence
S_8	Tv_1	anti-clockwise	voltage level e static presence
S_{10}	Tv_3	time	voltage level and static presence

Table 2. Function of the sensors in the implemented *SFC* logic

The first idea the utilization of the *ladder* code was only restricted to the external intercase. However the *ladder* code was also used to implement the concurrence treatment for the overall project. In the case of the Tv_1 e Tv_3 block requirements in which the actualization window of the variable was large enough to force the power-supply to energize the same block com different voltage levels. This yields to a sudden stop of the displacement of both trains, given that the current supplied by the power-supply grown to high levels, leading to the actuation of the circuit protections.

The solution to this problem was implemented in *ladder* as shown in figures ***figures*** 6 and 7, in which the overall sensor logic was implemented, including the all possible combination for simultaneous occurrence of facts, give precedence to the train that set sensors S_3 and S_8 with regard to the concurrent S_{10} and S_5, respectively.

Note in Tab. 2 and in Fig. ***figure*** 8 the great symmetry in the generated code, which is directly related to the physic architectures as shown in ***figure*** 1.

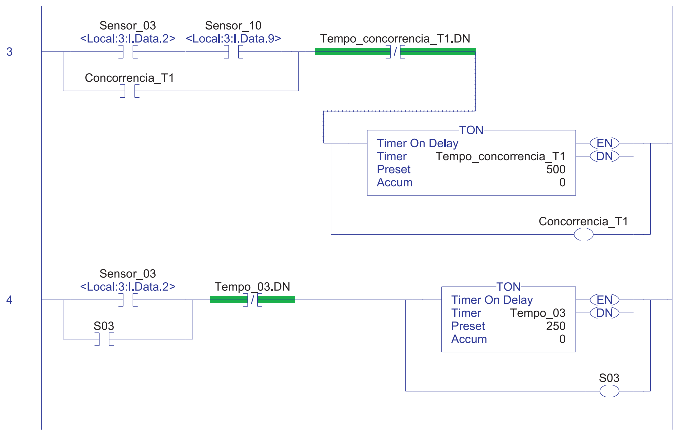

Figure 6. Block Tv_3 energizing concurrency treatment in *ladder*

ABCM Symposium Series in Mechatronics - Vol. 4 - pp.480-489

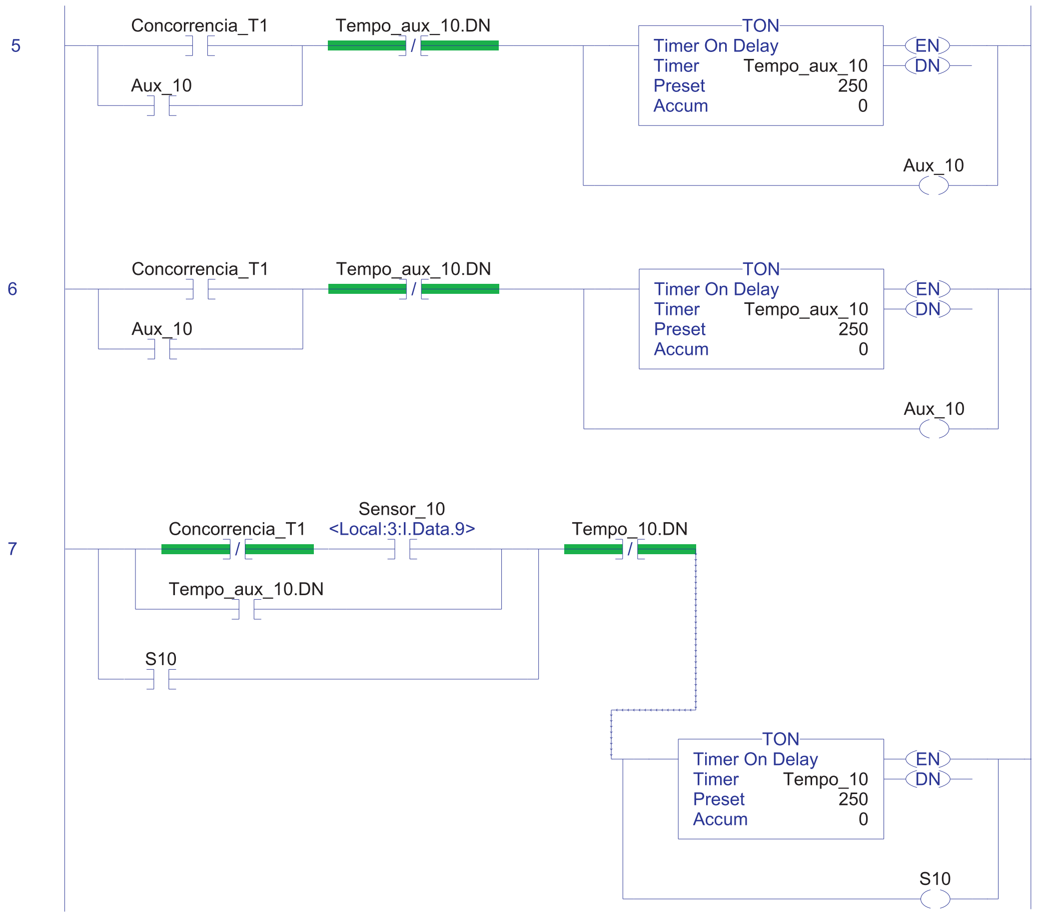

Figure 7. Block Tv_3 energizing concurrency treatment in *ladder*

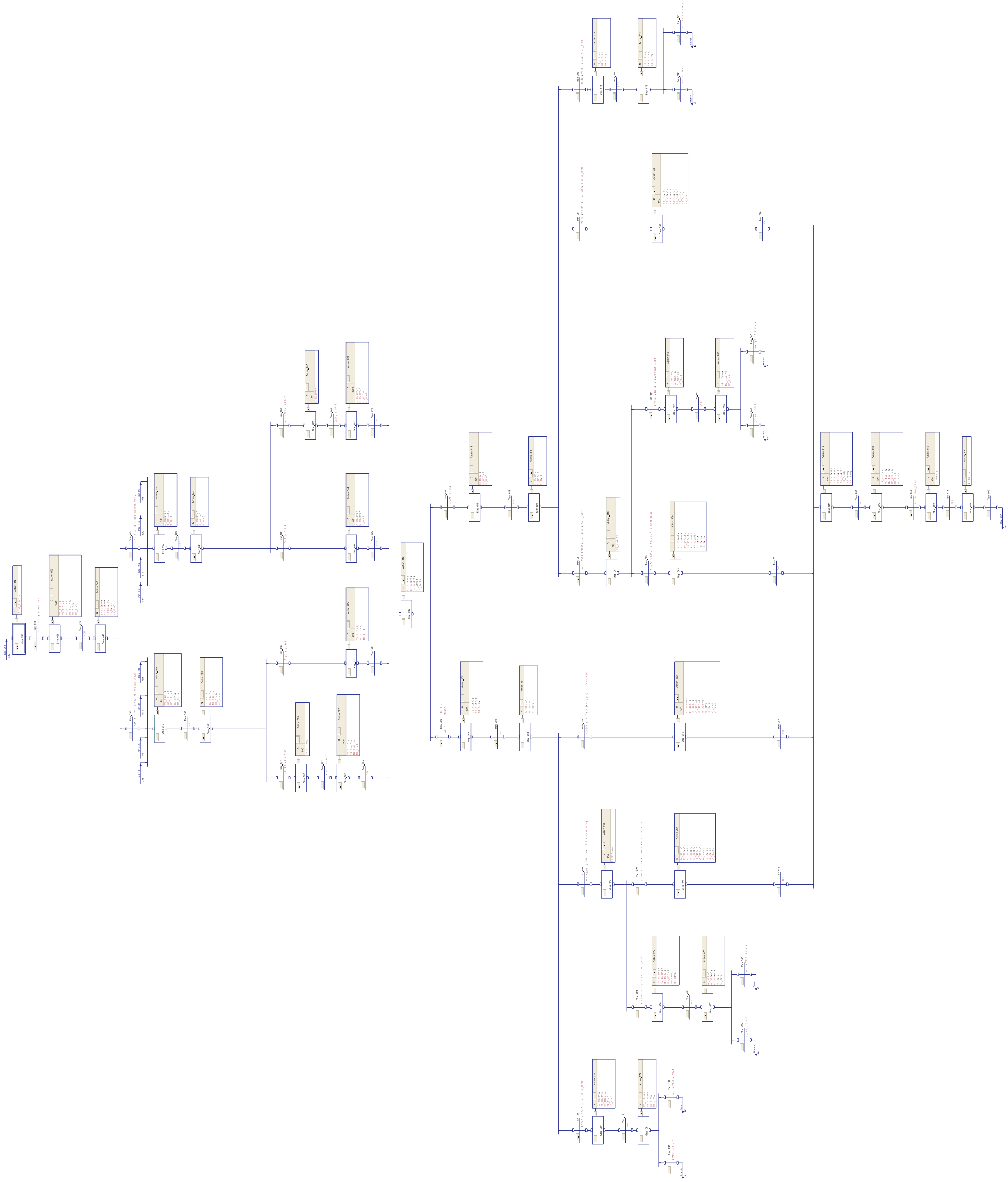

Figure 8. SFC Maps for clockwise movement

ABCM Symposium Series in Mechatronics - Vol. 4 - pp.480-489

5. CONCLUSIONS

Based on the obtained results, the initial objective of this work, of applying a safety based control strategy for controlling the block authorization to two different locomotives running at opposite directions, was fully attained. Concurrency problems were treated in a simple manner and the resulting control system was able to cope with the resource sharing problems encountered in the model railroad constructed. Based on the proposed control architecture and on the results obtained its clear that the industrial programmable controller is effective in highly complex applications such as the railroad traffic control automation. The programming languages available in these controllers are more effective than the traditional structured ones, since they provide a means of visualization of the whole control structure, allowing the easy detection of problems. They are also compatible with the discrete event system modelling techniques, which make it easier for the automation engineer to develop applications after modelling the complex parts of the plant. The Petri Net modelling technique was an important tool in the validation phase of the Project. However, the SFC language, further to being a programming language, was successfully used for modelling purposes. The practical result of this work can be viewed through the internet link: *http://www.youtube.com/watch?v=I1bhZRs7ZUQ*

6. ACKNOWLEDGEMENTS

The authors would like to acknowledge the material and financial support provided by Rockwell Automation do Brasil Ltda, without whose support this work would not have been possible.

7. REFERENCES

Tazoniero, A., 2007, ”Estratégias de Decisão para o Planejamento de Circulação de Trens em Tempo Real”, http://libdigi.unicamp.br/document/?code=vtls000428998.

Lei, T. and Tanglong, C. and Jian, X., 2000, ”Analysis of the concurrent model of train station based on Petrinet”, Autonomous Decentralized Systems, 2000. Proceedings. 2000 International Workshop on Volume , Issue , 2000 Page(s):92 - 96. `http://ieeexplore.ieee.org/Xplore/login.jsp?url=http%3A%2F%2Fieeexplore.ieee.org%2Fiel5%2F7071%2F19066%2F00880892.pdf%3Farnumber%3D880892&authDecision=-203`.

Gomide, F., Bessa, E., Vieira, P. e Neto, L. Railway Dispatch Planning and Control. In: Proc. of NAFIPS99, New York, USA, pp. 134-138: 1999.

Cassandras, C G., Lafortune, S. Introduction to Discrete Event System. Kluver Academic Publishers, Boston,USA: 1999.

8. Responsibility notice

The author(s) is (are) the only responsible for the printed material included in this paper