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). In most of these control schemes, there is the necessity of having a train operator for driving the locomotives according to the signals issued by the traffic control authority, thus resulting in a semi-automatic system, in which only the signaling is automatically controlled.

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 and 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 (Petersen 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 and Xiao (2000).

In order to make it easier to simulate and visuallise the typical problems found in railway traffic control, Jesus and

Aquino (2009) started the implementation of a model railroad (based on model railroad parts by Frateschi™), in which electrical miniature locomotives are driven by the DC electricity collected from energized railway tracks, in a similar manner as some real railways which provide the locomotive energy from a third track. In the model railroad, the train speed is determined by the voltage level at which the railway tracks are energized, differently from a real train, whose speed is controlled by the locomotive operator. Based on that characteristic and considering that the miniature trains would be automatically driven by an external controller, the authors adopted the strategy of assembling the tracks in electrically separated blocks, thus allowing the speed control of the locomotive that was located within each block. A similar approach, of separating the tracks into blocks for energizing purposes, can be found in some railway installations, although the speed control is left to the train operator, following the signals provided by the traffic control system.

This work uses the same apparatus developed by Jesus and Aquino (2009) and presents the study of a railway traffic controller, which was developed to deal with the main safety aspects in that specific railway model. More specifically, the controller was devised to prevent head-on collisions of two locomotives running at opposite directions in a single track railway. Further to modelling part of 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* graphical languages (IEC 1131-3). The developed control algorithm was applied to controlling the physical apparatus, which was assembled 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. The main objective of this work was to show that it is possible to attain the full automatic control of trains running on the same track and implement such a controller in an industrial Programmable Controller, using graphical IEC 1131-3 programming languages.

2. PROBLEM DESCRIPTION

Based on the geometry of the model railroad shown in Fig. 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 Fig. 1) in order to allow the development of a discrete controller to be implemented in a programmable controller.

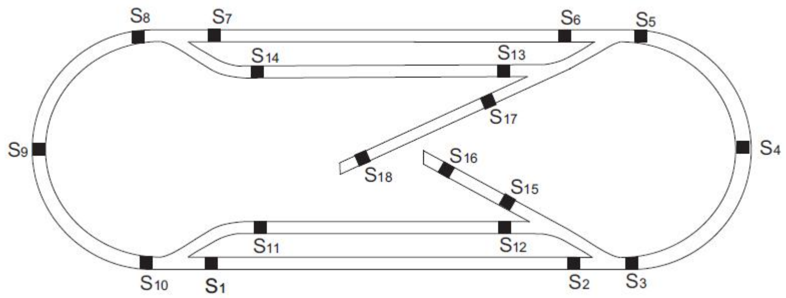

Figure 1. Schematics of the model railroad geometry outlining the discrete sensor positions

In Fig. 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 localization of the locomotives was made possible by installing under each one a set of magnets, which would

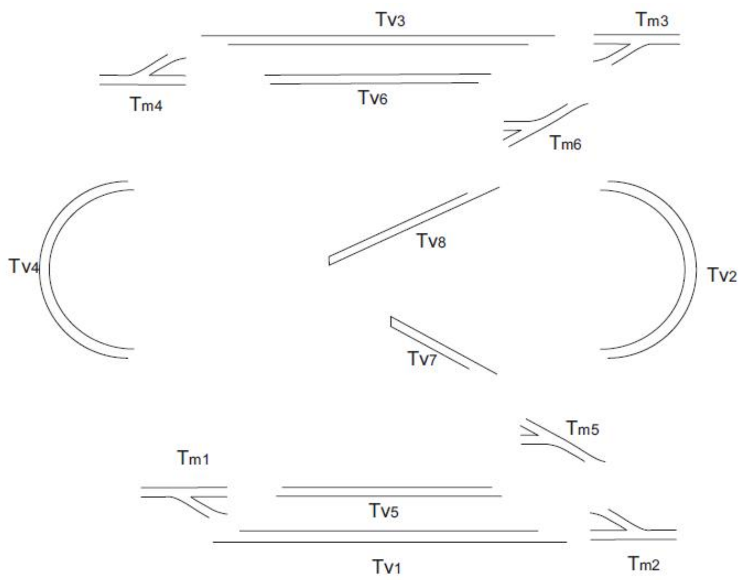

Figure 2. Schematic drawing of the model railroad according to the blocks made of single tracks, Tv_j, and the turnout blocks, Tm_k

activate reed switch sensors distributed on the circuit according to the scheme shown in Fig. 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 turnout actuators 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 1 to 6. They are actuated through a relay output module, by means of discharging a capacitor on one of the turnout actuator 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 a branch or to a passing loop. The distribution of the turnouts along the circuit is shown in Fig. 3.

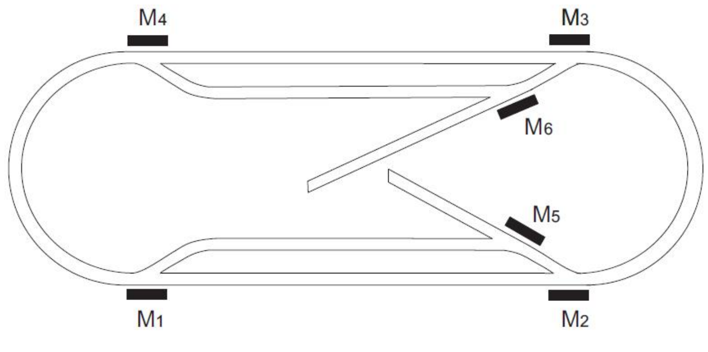

Figure 3. Schematic drawing according to the turnout positioning along the model railroad. M_t

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

3.1 Main characteristics of the problem

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 main control criterion used in the development of this work uses a first to request basis or, in other words, the first locomotive that requests the block authorization, will receive it. This is based on the assumption that the probability of having two locomotives requesting the same block at exactly the same time is very small.

Also, during the study of this problem, the sequential nature of the execution of the different process steps was detected. Such a sequential nature occurs because there is a set of blocks Tv_j and Tm_k and a set of turnout positions $M_{t,p}$, through which the locomotive goes in order to move from Tv_7 position to Tv_8 and vice-versa. 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 to that, the same consideration can be applied to any number of completed circuits. This displacement can be carried out independently of the other locomotive. To do that, the only 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 $M_{5,n}$, $M_{2,r}$, $M_{3,r}$ and $M_{6,n}$. The respective actions for achieving the movement present the following sequence:

1. energize the current block;
2. energize the next block;
3. energize the turnout(s) and switch for one of its (their) possible positions: $normal$ or $reverse$;
4. de-energize the previous sector;
5. de-energize the last turnout(s) traversed;

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

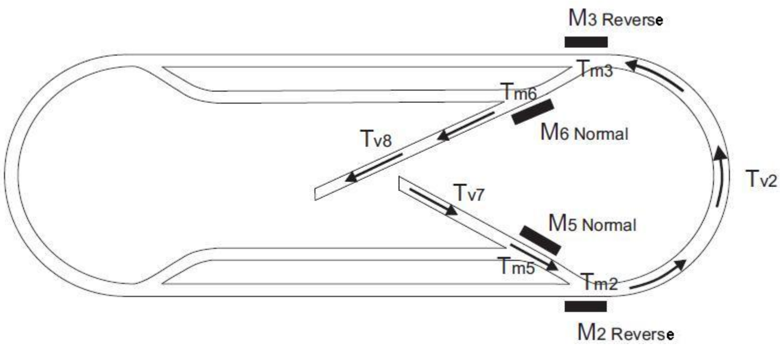

Figure 4. Schematics for the locomotive movement from Tv_7 to Tv_8

3.2 Petri Net modeling

The Petri Net modeling was developed for the specific case illustrated in the previous section in order to show the possible conflicts that would be faced during the development of the controller. Figure 5 depicts the resulting Petri Net model, considering the possibility of displacement in the reverse way. Resource concurrence is explicit in this modeling, given that one can consider the presence of two trains in blocks Tv_7 and Tv_8, both of them 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 static priority determination. Basically, 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 of the respective action for each direction, 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.

Note that the Petri Net shown in Fig. 5 does not have a cyclical behaviour since it models only the movement from one parking track to the other. The initial state shown in the Petri Net only illustrates the case of only a locomotive parked inicially in Tv_7. Therefore, once the locomotive attains the destination, it stops. Also, it should be observed that if two locomotives are set two each of the parking tracks, a conflict will be illustrated, since the model do not include movements along the other railway blocks.

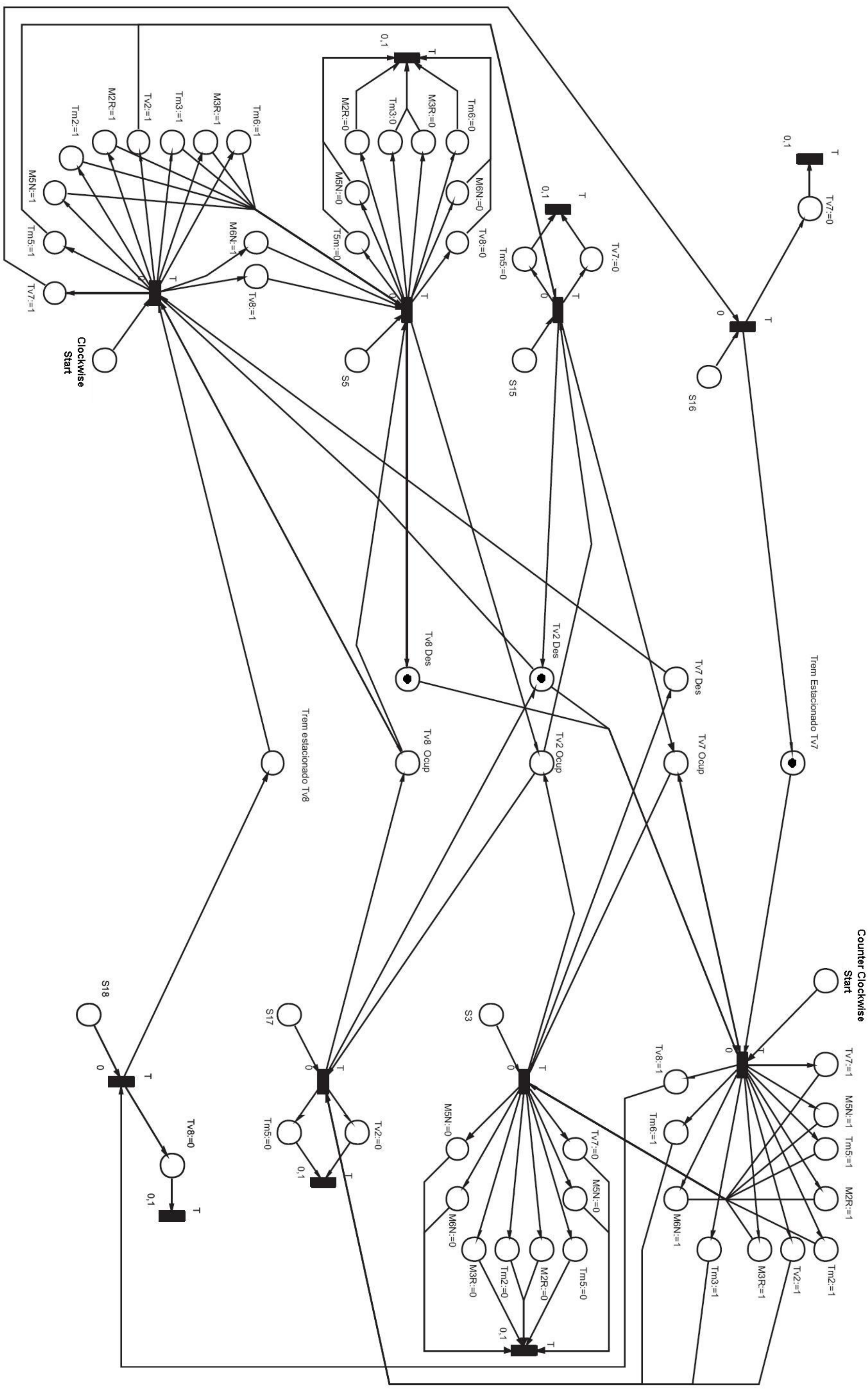

Figure 5. Schematic of the Petri Net and its initial marking

It can be emphasized that the implemented model does not represent the overall project. Although, by using a similar logic, the implementation of the overall case could be accomplished by expanding the specific model shown. Some places related to the controller outputs were deliberately duplicated in order to make it easier to visualize the network, apart from highlighting its symmetric characteristic. This places are not simultaneously active in any simulation time, indicating the exclusiveness of these resources.

4. CODES AND RESULTS

The Petri net modelling was carried out according to the description in the previous section. With such a modelling, the symmetry of the system could be visualized and some clues were obtained about the effectiveness of the devised control strategies. These were implemented using the *SFC* and the *Ladder* graphical IEC 1131-3 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 overall problem dealt with in this work consisted of initially placing the locomotives in the Tv_7 and Tv_8 blocks and driving them automatically to exchange places. Both locomotives would need to complete an arbitrary number of closed circuit turns (not necessarily equal) and then park in the opposite blocks: Tv_8 and Tv_7. For each direction (clockwise and counter clockwise), a corresponding SFC structure was designed. Figure 6 shows the resulting SFC structure for the movement in one direction. It should be noted that the SFC for the opposite direction is symmetric to the one shown in Fig. 6. Due to the large size of the resulting SFC map and to the limited space available in this text, detailed information about the SFC maps was intentionally omitted.

The flux of the states occurs in an independently way when the map executions are started. The action sequence obey to a general standard, similarly to the behaviour described in the previously illustrated case (see section 3.1).

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

1. De-energized: 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. The locomotive stays in the current block until the next block is liberated.

Although, there are some drawbacks related to the previously defined logic. For instance, in the case of sensors S_3 and S_8 (counter clockwise case) or S_5 and S_{10} (clockwise case), the track release mechanism is similar to the above mentioned one. However, the subsequent transitions are used to analyse more than simply checking the energization of blocks Tv_1 and Tv_3. This is due to the fact that, even if there is the intention of moving along these blocks, there is a possibility of bypassing through blocks Tv_5 and Tv_6 if blocks Tv_1 or $Tv3$ are in an occupied or required state. 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, another importante aspect can be observed in the transitions related to S_1 and S_7 sensors. In this case, the management of the cycle counter in both directions, related to the detection circuits S_8 (clockwise) and S_7 (counter clockwise), is accomplished in the final step of the displacements. This verification encompasses, further to the analysis of the energizing status of the Tv_7 and Tv_8 blocks, also the presence of the locomotives in these areas, by means of checking the state of the S_{16} and S_{18} sensors. This is due to the fact that S_1 and S_7) sensors are considered as the last decision opportunity to order if a train should be stopped in a crossing train situation. In this case, due to the increased complexity involved in the logical functions necessary to deal with this problem, *Karnaugh maps* were used for simplifying the analysis.

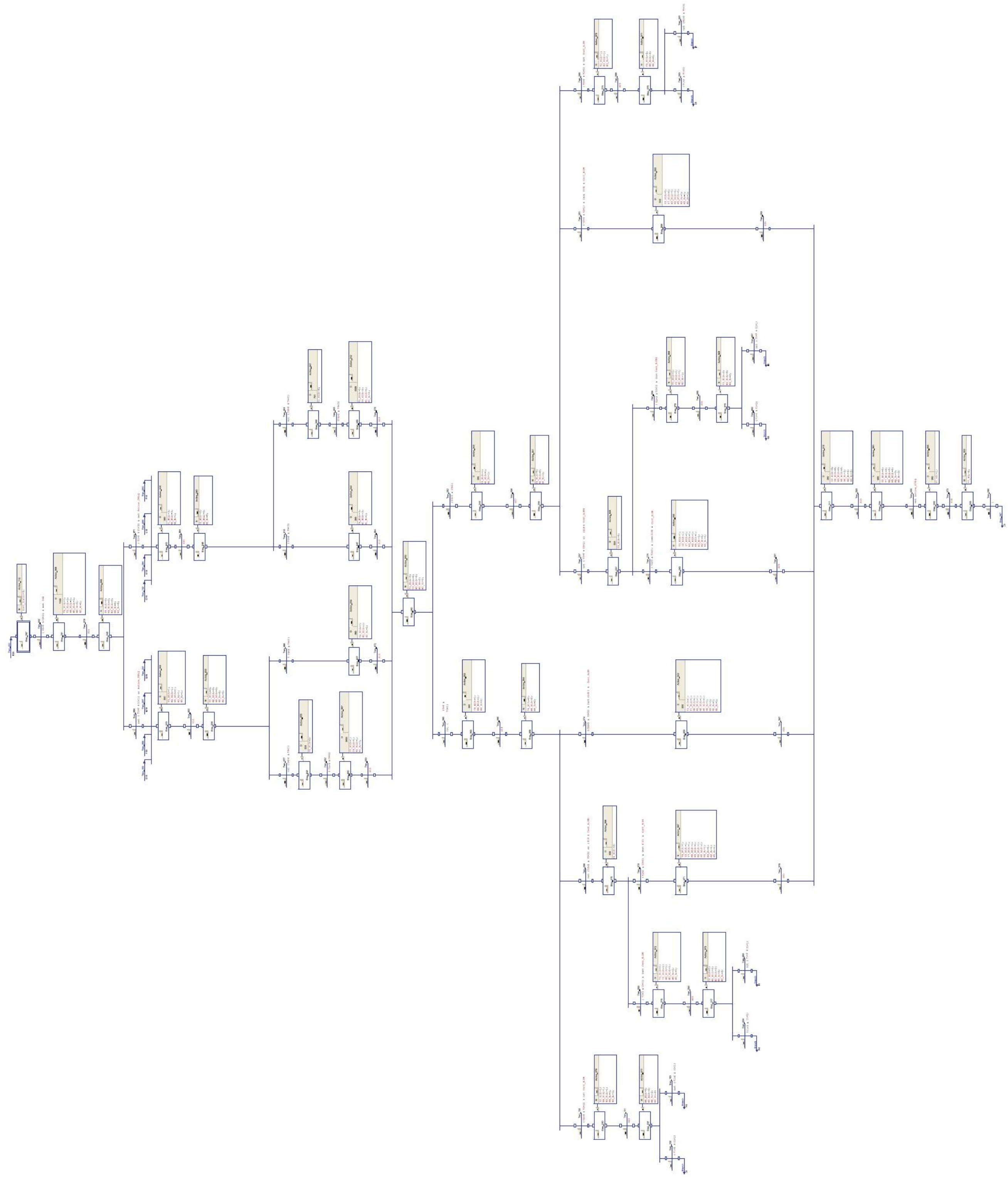

Figure 6. SFC programming the movement of trains in one direction

Note in Fig. 6 the great symmetry in the generated code, which is directly related to the physic architectures as shown in Fig. 1.

Some routines were written in *ladder* language (see Fig. 7) in order to provide support for some aspects with respect to the SFC maps. Such aspects were: (a) to provide an external interface; (b) to standardize the signals coming from the detection circuits; and (c) to facilitate the utilization of counters. The first idea during the development of the code was the utilization of the *ladder* language mainly for providing an external interface to the SFC program. However, during the tests, problems related to concurrency were detected. Despite the fact that there is a very small probability of having two locomotives requesting block authorization at exactly the same time, the concurrency problems might have occurred probably due to the scan cycle of the programmable controller. Coincidently, both locomotives would call for block authorization within the time span that corresponds to the programmable controller scan time for the SFC program execution. In order to deal with this problem, the *ladder* language was also used to implement the concurrence treatment for the overall project. The solution to this problem was implemented as shown in Fig.7, in which the overall sensor logic was implemented, including the all possible combination for simultaneous occurrence of facts, giving precedence to the train that would set the S_3 or the S_8 sensor despite the concurrent train that would set the S_{10} or the S_5 sensor.

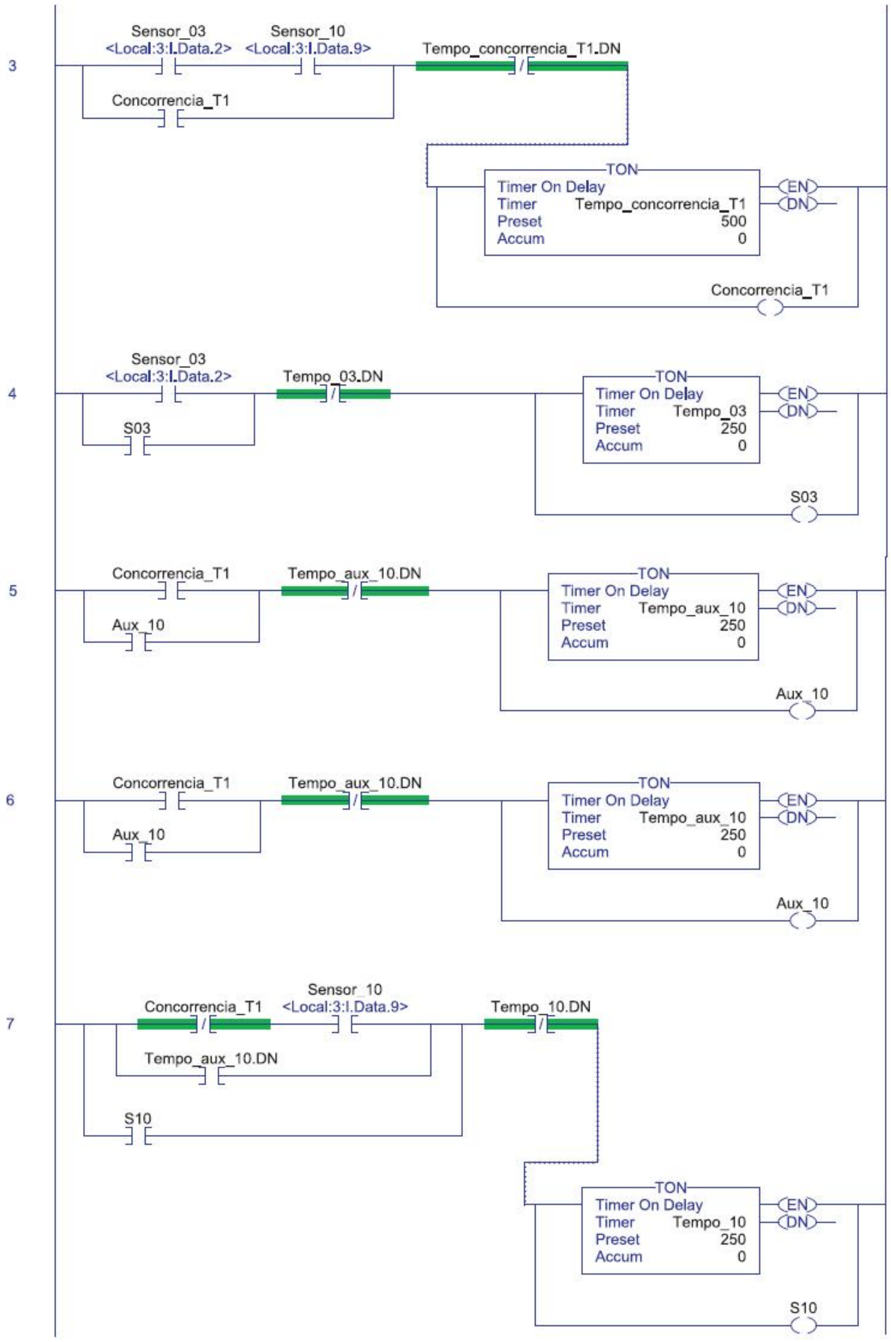

Figure 7. Block Tv_3 energizing concurrency treatment in ladder

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 also 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

ASSIS, W.O. (2002) Geração de Programa Horário e Regulação do Tráfego de Trens de um Sistema de Transporte Metroviário. Doctor of Science Thesis, Campinas State University, Campinas.

CHERNIAVSKY, A.L. A program for timetable compilation by a look-ahead method. Artificial Intelligence, vol. 3, pp. 61-67: 1972.

GAREY, M. R., JOHNSON, D.S. Computers and Intractability: A Guide to the Theory of NP-Completeness. W.H. Freeman and Company, New York: 1979.

GOMIDE, F., BESSA, E., VIEIRA, P., NETO, L. Railway Dispatch Planning and Control. In: Proc. of NAFIPS99, New York, USA, pp. 134-138: 1999.

JESUS, Tiago Romeiro de and AQUINO, José Victor de, 2009, Estratégia de controle de tráfego em vias férreas singelas, utilizando programação SFC sobre arquitetura de controladores industriais programáveis. Undergraduate Project Report. Mechatronic Engineering Course, Universidade de Brasília, DF, Brazil.

PETERSEN, E.,TAYLOR, A. Line block prevention in rail line simulation and optimization. Transportation Science 16(2), pp.192-205: 1982.

TAZONIERO, A. Estratégias de decisão para o planejamento de circulação de trens em tempo real. / Alexandre Tazoniero. Campinas, SP. 2007.