UDC 656.25:519.68
Preliminary comunication
Received: 02.12.2008.

Automatic generation and verification of railway interlocking control tables using FSM and NuSMV

Ahmad Mirabadi and **Mohammad Bemani Yazdi**
School of Railway Engineering, Iran University of Science and Technology, Tehran, IRAN
e-mails: mirabadi@iust.ac.ir; m_bemani@rail.iust.ac.ir

SUMMARY

Due to the important role in providing safe conditions for train movements, railway interlocking systems are considered as safety critical systems. The reliability, safety and integrity of these systems, relies on reliability and integrity of all stages in their lifecycle including the design, verification, manufacture, test, operation and maintenance. In this paper, the automatic generation and verification of interlocking control tables, as one of the most important stages in the interlocking design process has been focused on, by the safety critical research group in the School of Railway Engineering. Three different subsystems including a graphical signalling layout planner, a control table generator and a control table verifier, have been introduced. Using NuSMV model checker, the control table verifier analyzes the contents of control table besides the safe train movement conditions and checks for any conflicting settings in the table. This includes settings for conflicting routes, signals, points and also settings for route isolation and single and multiple overlap situations. The latest two settings, as route isolation and multiple overlap situations are from new outcomes of the work comparing to works represented on the subject recently.

***Key words**: railway signalling, interlocking, safety critical systems, formal methods, model checking, NuSMV.*

1. INTRODUCTION

Railway interlocking systems are categorized as safety critical systems with SIL-4, based on EN 50126 and IEC 61508 standards. Functional specification of the railway interlocking systems is introduced in interlocking control tables. Control tables have an important role in the signalling design process.

They clarify what conditions must be met before a train move can be permitted on the railway lines and stations. Control tables are considered as an interlocking design specification, to be used by the interlocking designers and also as a test specification, to be used by the tester. These tables contain the key functional safety requirements for the interlocking system. The development process of these interlocking tables, especially for medium to large scale stations, is an intensive labour requiring specialized skills, it is currently an entirely manual process. Obviously, this can cause a major source of human errors in the design process of interlocking system. Mechanization of the generation and verification of the control tables can be an efficient approach to improve the reliability of the overall interlocking system. The work introduced in this paper is an introduction to a toolset, designed for automatic generation and verification of control tables.

In contrast to the works represented by other researchers such as Eisner [1], Simpson et al. [2] and Hubber [3], this paper proposes an easier approach in modeling the interlocking system and its verification to identify and by comparing it to the work represented by Tombs et al. [4] a further step in identifying the settings for route isolation and flank protection.

2. INTERLOCKING CONTROL TABLE

In signalling point of view, a railway station consists of a collection of functions including different types of signals, track sections (monitored by train detection systems such as track circuits and axle counters), points, level crossing equipment and etc. Each of the objects in a railway can attain a certain number of states:

- a track section can be either occupied or clear;
- a three-aspect main signal can be red (ON), yellow or green (OFF);
- a point can be in reverse or normal position.

Figure 1 depicts a schematic view of signalling objects arrangement (signalling layout plan) in a typical railway station. Each separated object in this figure is provided with a unique identification code. The layout plan of the stations is considered as the first stage of the interlocking design, based on the operation requirements provided by the railway operator.

In setting a route for a particular train movement (i.e. a signal to become green or yellow) the following are the minimum pre-settings, required to be implemented and verified [5]:

- all tracks in the route and in the overlap should be clear;
- all points in the route and in the overlap should be set, clear, locked and checked;
- all conflicting signals and opposing signals should be ON (red);
- all in-route signals should be OFF (clear);
- the route should be isolated from all potential conflicting movements.

An interlocking control table is a structured, tabular presentation of the rules and pre-settings, governing route settings. It is used as a reference for identification of interrelation between different signalling functions (i.e. signals, points and track sections) in generation and verification of interlocking.

All possible and required routes in the stations, which are derived from the signalling layout plans, are represented in the route table of the stations.

Generation and verification of control table is the design stage after the route table generation and before the wiring diagram is designed in relay based interlocking systems (or software flowchart development in computer based interlocking). The format and the contents of tables are not standardized, and may vary even within the same railway administration. Nevertheless, general principles of control table design are evident.

A route is defined by an entrance signal and exit signal. Each row of the table consists of the pre-settings required by one particular route which can be defined in the station. The required settings for a route between signals *S1* and *S9* in Figure 1, as one row of the interlocking control table, is shown in Table 1.

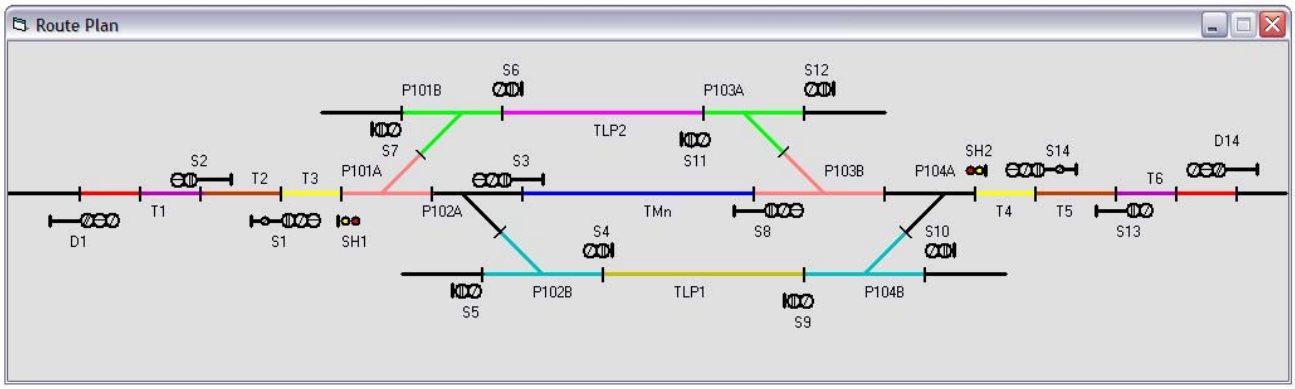

Fig. 1 Signalling layout plan for a typical simple railway station

Table 1 A row of control table for the station shown in Figure 1

Start Signal	Route Name	Exit Signal	Signals		Points		Track Section		Overlap	Conf. Routes	Flank Protection	
			ON	OFF	Normal	Reverse	Clear	Occ.			Normal	Reverse
S1	*S1(m1)*	*S9*	*S9* *S3* *S4* *S5* *S6* …	*S1* *Sh1*	*P101A*	*P102A* *P102B*	*T3* *TLP1*	*T2*	*P104B[N]*	*S3(m1)* *S4(m1)* *S5(m1)* *S5(m2)* *S6 (m1)* …	*P101A* *P104A*	*P103B*

3. AUTOMATIC CONTROL TABLE GENERATION AND VERIFICATION

Figure 2 shows the flow diagram of an automatic control table generation and verification system. The system is basically designed in three subsystems as:

- Graphical Signalling Layout Planner (SLP);
- Route Table Generator (RTG);
- Control Table Generator (CTG);
- Control Table Verifier (CTV).

3.1 Signalling Layout Planner (SLP)

Signalling Layout Planner (SLP) is a software tool to plan the signalling layout of any given station, based on its topographic map, using a user friendly graphical interface. Using SLP, the user is able to generate a model of the station as a combination of track sections and then to position the signalling objects (i.e. signals, points…) on the specified locations, based on the operational and signalling safety requirements.

SLP provides the signalling layout plan in Extensible Markup Language (XML) format.

3.2 Route Table Generator (RTG)

Route Table Generator (RTG) is a software system to analyze the signalling layout plan and to identify all routes possible to be defined in the station. Each route is defined as the distance between a start and an exit signal. The system is able to identify routes initiated from main, colling-on and subsidiary signals in the station. The operator will be able to alter the table according to operational requirements and limitations.

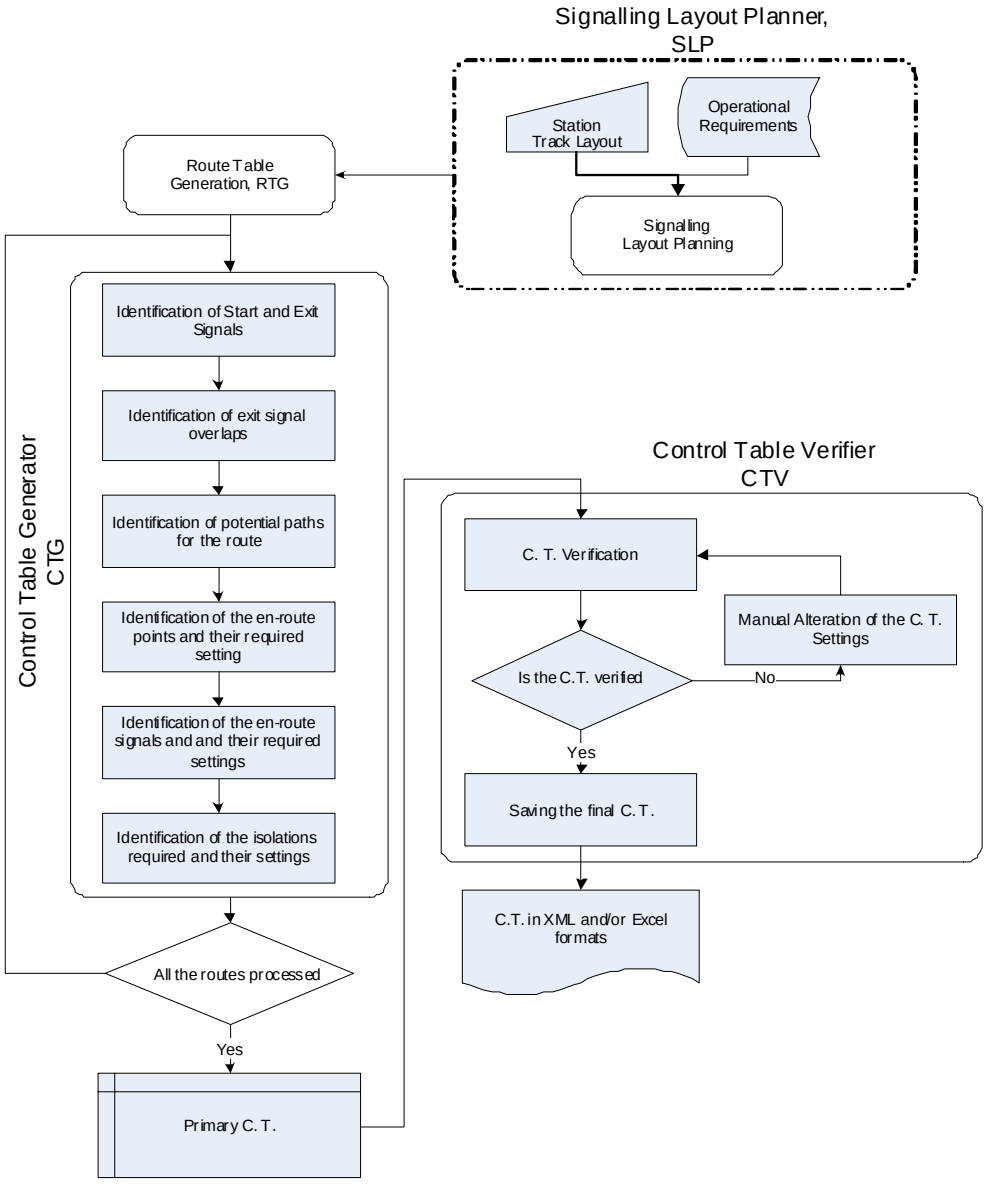

Fig. 2 Flow diagram of automatic control table generation and verification

3.3 Control Table Generator (CTG)

The Control Table Generator (CTG) determines all required settings for each particular route specified in the route table. For this purpose, CTG scans the route between the entrance and the exit signals, and identifies all track sections, signals and points which have filled the route. In other words, the CTG algorithm identifies the sequence of track sections, points and signals within each route and its overlap and their corresponding situations (Normal or Reverse).

Since, in some situations, there are more than one path to reach from the entrance signal to the exit signal (e.g., s_1 to s_{13} in Figure 1), CTG algorithm is designed in a way to identify and record both paths as two different routes.

Running the CTG over the route table and the signalling layout plan, a majority of control table field, including the following will be completed:

- overlaps;
- track sections within the route and overlap that should be clear or occupy;
- points lying in the route and overlap and their direction;
- signal replacement track sections (track sections which are placed after the entry signal);
- en-route shunt signals.

Table 2 shows a sample of CTG output for the route connecting signals *S1* and *S9*.

3.4 Control Table Verifier (CTV)

After generation of primary control table consisting of basic setting for all routes identified by RTG or by the user, Control Table Verifier (CTV) will check the generated control table against a set of signalling principles, to ensure the integrity of the settings and also to fill the remaining fields of the control table.

For this purpose, CTV benefits from the NuSMV model checker. CTV checks the possibility of a collision of train moving on a particular route with all other routes identified in the station.

4. CONTROL TABLE VERIFICATION

Automatic verification of control tables is one of the key functions of the signalling design toolset. In this toolset, the automatic verification is performed by using the formal language Finite State Machines (FSM) as well as the symbolic model checker NuSMV.

FSM is used to model the train movement as a sequence of states which train should go through from entrance signal to the exit signal, while NuSMV is used for detection of any conflict between the routes, in the same or in the opposite directions.

The input language of NuSMV is designed to allow for the description of Finite State Machines (FSM) as transition relations. This relation describes the evolutions of the FSM states.

4.1 CTL model checking

In order to check the model developed for a system satisfies the desired properties and conditions specified by the user, a model checker is used. These specifications need to be defined for the system in a suitable manner. In NuSMV, the specifications to be checked can be expressed in two different temporal logics: the Computation Tree Logic (CTL) and Linear Temporal Logic (LTL). The specifications represented in CTL or LTL will be evaluated by NuSMV, which determines whether they are true or false in FSM. If the NuSMV recognizes that a specification is false, it will provide the trace of the FSM that falsifies that property as an output. In this paper CTL is used to express the specifications of the model.

CTL provides the opportunity to express the properties that should hold for all the paths, starting in a particular state and also properties that should hold just for some paths. For example, consider for instance the formula *AF p* in CTL. It expresses the condition that, for all the paths *(A)* starting from a state, eventually in the future *(F)* condition *p* must hold. That is, all the possible evolutions of the system will eventually reach a state satisfying condition *p*. The *EF p* formula in CTL, on the other hand, requires that there exists some path *(E)* that eventually satisfies *p* in the future.

Table 2 Sample of control table, generated by CTG

Start Signal	Route Name	Exit Signal	Signals		Points		Track Circuits		Overlap
			ON	OFF	Normal	Reverse	Clear	Occupied	
S1	S1(m1)	S9	S9	S1	P101A	P102A P102B	T3, TLP1	T2	P104B[N]
S1	S1(m2)	S9	S9	S1	P101A	P102A P102B	T3, TLP1	T2	P104B[R]

Similarly, formula *AG p* requires that condition *p* is always, or globally, true in all the states of all the possible paths, while formula *EG p* requires that there exist some paths along which condition *p* is globally true. More information on CTL logic can be found in Refs. [6] and [7].

4.2 Verification of the safety requirements

The general safety requirements of railway interlocking system are explained in Section 2 of this paper. In order to formalize the problem of a train moving on ah particular route from one state to another, while at the same time a second train is moving on all other routes sequentially. The specifications will be verified in case of collision between the two mentioned trains.

A train collision is simply specified for two trains (*t1* and *t2*) occupying the same track section in the station. The CTL formulas to ensure train movement without collision and derailment are given in the Table 3.

Table 3 A sample of CTL formula

AG! (t1.location = t2.location)
AG! (t1.location = derailment)
AG! (t2.location = derailment)
AF (t1.locatin = last track & t2.locatin = last track)

In Table 3, AG! can be read as never and AF as at least one time. These formulas guarantee that:

- Two trains should never be located in the same track section, otherwise collision will happen;
- A train should occupy all the track sections of the route and its overlaps, sequentially, until it reaches the last track section. In other words, trains should completely pass the routes without any collision or derailment.

After finishing the above checking of all routes in the station, all detected conflicts will be represented as a list of conflicting routes.

The control table and consequently the interlocking system should provide all necessary settings in order to ensure that no two conflicting routes can be set at the same time. During the checking the FSM model, the NuSMV model checker goes through each route of the route table. The conflicting routes are detected and represented as counter-example outputs by the NuSMV. A counter-example is a list of states that lead to a state violating the checked safety requirements (i.e. in this case a front-to-front collision).

For each state only the changes of the previous state are given. Figure 3 shows the key parts of states that finally lead to a collision of trains *t1* and *t2* on the track P101A (see state 1.4).

The last stage of the process, through which the primary control table will be completed, new settings for the control table, ensuring that no two conflicting routes will be set at the same time, will be added.

```
-- specification AG !(t1.location = t2.location)  is false
-- as demonstrated by the following execution sequence
Trace Description: CTL Counterexample
Trace Type: Counterexample
-> State: 1.1 <-
  t1.location = T3
  t2.location = TLP1
-> Input: 1.2 <-
  _process_selector_ = t1
  running = 0
  t2.running = 0
  t1.running = 1
-> State: 1.2 <-
  t1.location = P101A
-> Input: 1.3 <-
  _process_selector_ = t2
  running = 0
  t2.running = 1
  t1.running = 0
-> State: 1.3 <-
  t2.location = P101B
-> Input: 1.4 <-
  _process_selector_ = t2
  running = 0
  t2.running = 1
  t1.running = 0
-> State: 1.4 <-
  t2.location = P101A
-- specification AF (t1.location = TLP2 & t2.location = T3)  is true
-- specification AG !(t1.location = derailment)  is true
-- specification AG !(t2.location = derailment)  is true
```

Fig. 3 counter-example output by NuSMV

As a result of the model checking implementation, the detected conflicting routes are added to the associated list and then all required settings for isolation of the specified route, such as point settings, will be identified and added to the control table.

In other words control table verifier, does not verify only the settings in the primary control table, it also completes the table with settings concerning the route isolation as one of the safety requirements.

Table 4 shows the new control table, with additional information in bold, which has been added by the verified program.

5. CONCLUSION

This paper introduces an algorithm for generation and verification of interlocking control table. Required settings for route isolation and also multiple overlaps are the problems this paper has tried to solve. Using the toolset developed by the safety critical research group in the School of Railway Engineering (SRE), human interference in the design, development and verification of control table have been minimized.

6. REFERENCES

[1] C. Eisner, Using symbolic model checking to verify the railway stations of hoornkersenboogerd and heerhugowaard, *Proc. of the Conf. on Correct Hardware Design and Verification Methods* (CHARME'99), Vol. 1703 of LNCS, Springer-Verlag, 1999.

[2] A. Simpson, J. Woodcock and J. Davies, The mechanical verification of solid state interlocking geographic data, *Proc. of the Conf. on Formal Methods Pacific'97* (FMP'97), Discrete Mathematics and Theoretical Computer Science Series, Springer-Verlag, pp. 223-243, 1997.

[3] M. Hubber, Towards an industrially applicable model checker for railway signalling data, Masters Thesis, University of York, 2001.

[4] D. Tombs, N. Robinson and G. Nikandros, Signalling control table generation and verification, *Proc. of the Conf. on Railway Engineering* (CORE2000), Railway Technical Society of Australasia, November 2002.

[5] Office of Rail Regulation, *Railway Safety Principles and Guidance, Part 2, Section D, Guidance on Signalling*, HSE Books, London, April 2006.

[6] R. Cavada, A. Cimatti, G. Keighren, E. Olivetti, M. Pistore and M. Roveri, *NuSMV 2.2 Tutorial*, IRST, Povo, 2006.

[7] E.A. Emerson, Temporal and modal logic, In: *Handbook of Theoretical Computer Science*, Ed. J. van Leeuwen, Vol. B, Elsevier Science Publishers, Amsterdam, pp. 996-1072, 1990.

Table 4 Additional columns filled after the verification process by CTV

Enter Signal	Route Name	Exit Signal	Signals		Points		Track Circuits		Overlap	Conflicting Routes	Flank Protection	
			N	R	N	R	Clear	Occ.			N	R
S1	S1(m1)	S9	S9, S3, S4, S5, S6, …	S1, SH1	P101A	P102A P102B	T3, TLP1	T2	P104B[N]	**S3(m1), S4(m1), S5(m1), S5(m2), S6 (m1), …**	**P101A P104A**	**P103B**

AUTOMATSKO GENERIRANJE I PROVJERA ŽELJEZNIČKIH KONTROLNIH PLOČA ZA BLOKIRANJE KORISTEĆI FSM I NUSMV

SAŽETAK

Željeznički sustavi za blokiranje smatraju se vrlo važnim sigurnosnim sustavima zbog njihove uloge u osiguravanju uvjeta kretanja vlakova. Pouzdanost, sigurnost kao i cjelovitost ovih sustava ovisi o pouzdanosti svih faza njihovog životnog ciklusa uključivši projektiranje, provjeru, proizvodnju, testiranje, rad i održavanje. Ovaj rad se usredotočuje na automatsko generiranje i provjeru kontrolnih ploča za blokiranje, budući da one predstavljaju jedan od najvažnijih faza u procesu projektiranja blokiranja. Ovo je ujedno i osvrt istraživačkog tima koji se bavi sigurnošću željezničkog prometa u Školi željezničkog prometa u Teheranu, Iran. Prezentiraju se tri različita podsistema koji uključuju shemu signalizacije, generator kontrolne ploče, te uređaje za provjeru kontrolne ploče. Pomoću NuSMV modela kontrole, uređaj za provjeru kontrolne ploče analizira sadržaj kontrolne ploče pored sigurnosnih uvjeta gibanja vlaka te provjerava i proturječne postavke na ploči. To uključuje postavke proturječnih ruta, signala, točaka kao i postavke zbog izolacije rute, te jednostavnih i višestrukih situacija preklapanja. Posljednje dvije postavke, kao što su izolacija rute i višestruke situacije preklapanja, predstavljaju nešto novo u ovom radu u usporedbi s nedavno predstavljenim radovima u ovom području.

***Ključne riječi**: željeznička signalizacija, blokiranje, kritični sustavi sigurnosti, formalne metode, provjera modela, NuSMV.*