FACTA UNIVERSITATIS (NIŠ)
SERIES: ELECTRONICS AND ENERGETICS VOL. 11, NO.1 (1998), 25-49

TOWARDS AN UNIVERSAL COMPUTER INTERLOCKING SYSTEM

Dejan Lutovac and Tatjana Lutovac

Abstract. During twenty years of applications, computer railway interlocking systems have offered many advantages, but they have not fully satisfied expectations. This paper gives a summary of possible areas of improvement that have been observed and proposals to resolve many of them. The solutions are presented in the form of a Computer Interlocking System (CIS) which could become a standard system, independent of track layout and country of application. The main contribution is conversion of the operational, functional and safety requirements of an interlocking system into general interlocking software. There is also a new way of presenting a control table which can be entered as a simple data file and control table conversion to the interlocking functions suitable for computer application. An advanced method of screen design showing the layout of a railway station is proposed as well.

1. Introduction

The standardization of the safety and functional requirements for a CIS is under consideration in Europe [3]. A CIS promises standardization by nature of realization of interlocking functions by software instead by hardware (relay contacts), but there is still disagreement between systems developed and applied in different countries. The purpose of this paper is to try to contribute to the standardization of CIS. The system proposed in this paper is the result of many years work in the railway signalling field. The experience gained from designing and applying the various Relay Interlocking Systems (RIS), CIS and solid state railway equipment in various countries, has been used to identify and define the common requirements and principles in the form of general interlocking software.

Manuscript received November 11, 1997.
Manuscript revised April 2, 1998.

D. Lutovac is with Transportation Systems, Siemens Ltd., Melbourne, Australia. T. Lutovac is with Department of Computer Science, RMIT University, Melbourne, Australia.

2. Description of operation

A railway interlocking system controls the traffic in a railway station, and between adjacent stations. The control includes train routes, shunting moves and the movements of all other railway vehicles in accordance with railway rules, regulations and technological processes required for the operation of the railway station.

The CIS gives the authority for moves through the station areas and defines the route and the speed of the move in every particular situation. The number of different routes depends on the track configuration. All routes that can be given are defined by the control table. All safety and functional demands are included as well. The correct implementation of the control table in the logic of the interlocking system makes setting of the routes possible only if all safety requirements are met.

The request to set a route comes from the signalman, but the decision to allow the move is made by the interlocking system on the basis of in–built safety logic represented by control table requirements.

The continuous monitoring of the state of the system is provided by the interlocking. All relevant information about system, power supply, trackside elements (signals, points, track circuits etc.) is presented to the signalman on a VDU or mimic panel. The signalman, in accordance with the train time table, selects the route to be set on the basis of the indicated information and gives a command to the interlocking. The interlocking sets the points and the signals for the requested route in accordance with the state of the system (availability of the route) and control table requirements (safety).

The final goal of the control system is safe passage of a train through the controlled area. The indication of the position of the train is achieved by detection of the state of track circuits in the route. The communication between the signalman and the interlocking system is interactive, providing the means for the signalman to undertake all necessary actions at the appropriate time. The train releases the route by the occupation and release of the track circuits. If normal release by the train is not possible, for any reason, the signalman can initiate an action to resolve the problem.

The process described above represents a closed loop of the real time control that consists of signalman, the interlocking and the train. A continuous control system monitoring train movements is required to update the status of the controlled elements on time. The time cycle is defined by the shortest track circuit in the system and by the maximum train speed. If maximum line speed is $160km/h$ and the shortest track circuit $24m$, then the time between two consecutive readings can not be longer then $1s$. This figure restricts the total number of elements that can be used in the system.

3. Proposed hardware

The system has been designed from a signalling design engineer's point of view using, as far as possible, the latest computer technology and experience gained from the application of existing computer interlocking systems. The proposed system is a modular, distributed, fail–operational computer system. The block diagram of the system is shown in Figure 1. The CIS generally consists of: keyboard, Video Display Units (VDUs) for signalman display and maintenance terminal, printer, CPU, recording media, remote processing units, trackside equipment (signals, points, track circuits etc.) with appropriate interfaces and a power supply. The CIS can be remotely controlled as a part of Central Traffic Control (CTC) system and can allow remote maintenance access as a part of Central Maintenance System (CMS).

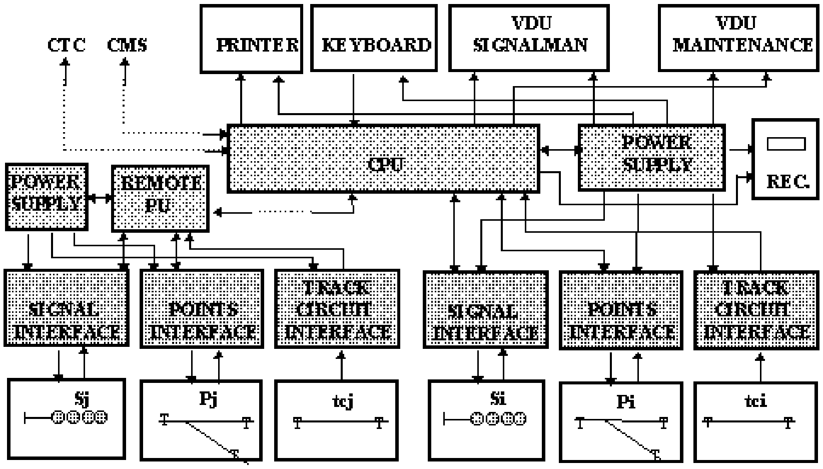


Figure 1.

3.1 Two–out–of–three fault tolerant system

From the safety point of view the two-out-of-two system is a fail–safe system equivalent to the relay interlocking system, but a two–out–of–three system gives much better reliability and availability characteristics [43], [46]. Therefore many approved systems, as well as systems under development, in which safety is based on the redundancy of the hardware, utilize this concept and it seems that such a configuration has become the standard [8], [16], [42]. The high price of specially designed processor modules was the main reason for all other developers to adopt solutions with lower numbers of processor modules. We believe this is not an issue any more. Due to significant reduction of prices of computer's equipment generally, redundancy of the hardware, in the aim of achieving safety, appears as a cost effective solution. The CPU of the proposed system consists of three identical processor modules which operate as a triple redundant fault–tolerant system with redundancy management. A typical example of triple redundant CPU is presented in Figure 2.

In the majority voting system the processor modules operate in parallel, all receiving the same inputs and performing the same tasks. Their outputs are compared and the system output is derived in accordance with the majority vote. The comparison and voting is achieved by redundancy management hardware which is able to isolate any module which is in disagreement with other two. The system will continue to work as a fail–safe system in two–out–of–two configuration, until the failed module is repaired or replaced, when the system reverts to the triple system. Failure of a second processor module before the first failed module is repaired will cause a system failure, result in a complete system shut–down and all equipment will lose power as a safety precaution. For example, one of the axioms of the railway signalling rules is that a dark signal has to be considered as a signal showing red (stop) aspect. On the railway it is generally safe to bring a train to rest in the event of failure.

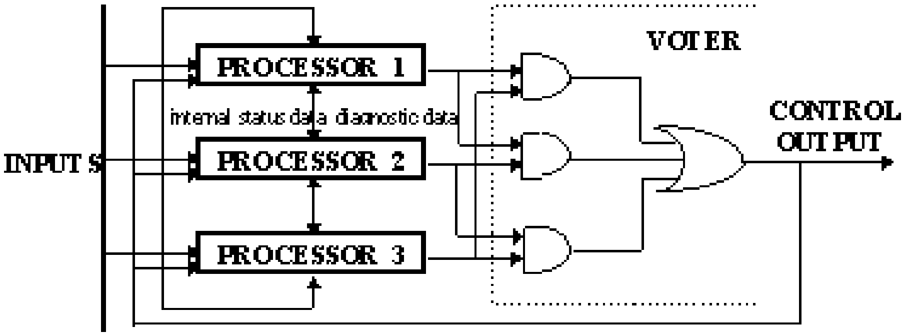


Figure 2.

3.2 Two–out–of–two fail–safe remote processing unit

A remote processing unit is an intelligent dual fail–safe interface controller which acts as a concentrator for a group of trackside elements. A clock synchronized work with redundancy clock arrangement is proposed [21], [23]. The simplified architecture of the unit is illustrated in Figure 3.

The main purpose is to allow connection of remote equipment without expensive signalling multicore cables and improve the speed of reading the state of the elements from the field. Low cost industrial controllers can be used for realization of the unit, due to very limited local functional requirements. They can also satisfy various temperature requirements of the environment. Data collected by the unit can be stored in local RAM and transmitted later at the one time. Communication is based on vital serial transmission with protective coding and application of Cyclic Redundant Code (CRC) Check–sum. The existing CIS have a communication speed which is generally low and has little possibility of being increased with current hardware used [40]. The proposed solution would allow increase of the speed up to 28.8 KBaud with possibilities for further increases. Communication uses a master/slave protocol wherein the units talks only if questioned by the CPU. Direct Memory Access (DMA) and interrupt handler could be used to increase the speed, where required. A two–out–of–two system with fail–safe comparator is sufficient to provide traditional railway fail–safety. More redundancy and availability is not required on this level. In the case of failure the unit will be isolated and graceful degradation will take place. Careful grouping of elements can minimize the affect of graceful degradation to availability of the whole system.

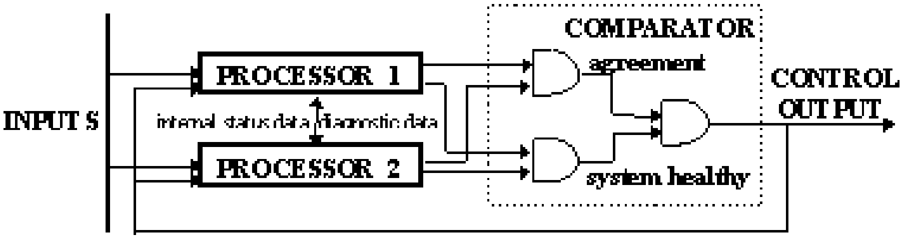


Figure 3.

3.3 General interface for trackside elements

A general type of interface is proposed to provide compatibility with different types of trackside signalling equipment and presented on Figure 4.

The interface is not fail–safe, but its function can be proved in a closed loop of real–time control before and after the change of state of any element [19]. Various fail–safe design techniques are available for protection against wrong side failures of the interface including the duplication of inputs [32], [23]. The general interface have been designed to provide the digital I/O connection of various types of trackside equipment, found in different countries, to the interlocking system, with minimum design changes and minimum affects on the system. In addition it ensures galvanic isolation between a trackside element and the system. The application of memory mapped I/O space and vector interrupts can be used in order to increase the number of outside elements and maximize performances of the system [12].

Three basic types of the trackside elements (signals, points and track circuits) have been considered. The basic fail–safe concept of relay interlocking can be retained. The appropriate simple circuits have to be designed to enable connection to the interface and thus, microcomputer's control of the elements as controlled objects in a real time system. The circuits of trackside elements, in some cases, should be slightly modified to suit the application and could be simplified. One example of the fail–safe trackside circuits design and their connection to the system with the interface was presented in previous published papers. [20], [25].

Power supply for this type of equipment is an independent system based on redundancy. A change over period for the computer equipment is covered by the UPS (Uniterruptible Power Supply) unit. The control and monitoring is achieved by the general interface [19].

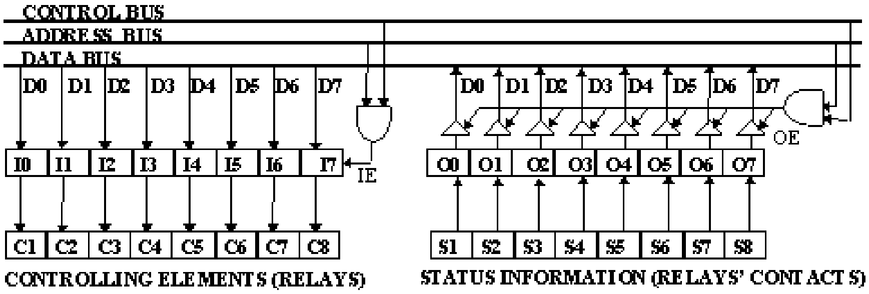


Figure 4.

3.4 Safety techniques

The redundancy management hardware does not have to be fail–safe as a unit. It can be checked by software using specially designed techniques which will prove a correct working state just before the required voting process, as well as immediately after the voting process, giving the possibility to cancel the previous decision if a failure is detected [23]. Fail–safe redundancy management hardware is still desirable because of the difficulties in proving the fail–safety of the software. It can, also, release software whose task is permanent checking and improve the speed of the system.

3.5 Asynchronous work and comparison before safety critical actions

The loose synchronization is anticipated for the fault tolerant system [14], [47]. The comparison of the outputs does not have to be at each cycle. It is satisfactory to prove agreement of the modules only when a safety critical activity is demanded. This will happen whenever the system sends a command to an element to change its state. This will eliminate clock synchronization problems as well as superfluous cross-checking between processor modules [6], [44]. The reduction of the comparison frequency allows simplification of the redundancy management hardware and/or use of faster processors. Therefore the features of the systems can be improved without any loose in safety or in functionality.

3.6 'At least as safe' as a conventional relay system

The most important safety approach is based on the classic and worldwide adopted safety analysis of relay systems. The main characteristics of the safety analysis are single channel information flow and system resistance to a single fault. This means that a single failure can not cause unsafe conditions under any circumstances. Calculated safety, expressed as Mean Time Between Unsafe Failure (MTBUF), required by the Office for Research and Experiments (ORE) of the International Union of Railways (UIC) is 100 years ($870000h$). Reliability required by ORE UIC is 4 months ($2880h$) [34], [35].

The described system concept offers the same treatment of a microprocessor interlocking system as a relay interlocking system from the fail–safety point of view. Unpredictability of the failure modes of the electronic components is covered by isolating the faulty module, giving the system at least the same level of the fail–safety as the level of an equivalent relay interlocking system. Calculated safety, even for a two–out–of–two system, will satisfy ORE UIC requirements [19].

3.7 Reliability and availability

The redundancy technique improves reliability and availability of the system. Modular structure makes replacement of the faulty module fast and easy, contributing to the higher availability of the system.

The probability of the same error occurring simultaneously in two different processor modules is very small. In addition, diversity in hardware can be used to protect against common mode failures. Theoretically it is possible to reduce the probability further by increasing the number of elements which must be in agreement, three–out–of–four or four–out–of–five, but the cost of the system will be higher and could not be justified. Mathematical analysis of the triple redundant systems with repair has indicated that expected failure rates for the system as well as for the parts will satisfy ORE UIC requirements [22].

3.8 Compatibility and cost

Use of the latest technology

Most of the existing CIS use old 8 bit microprocessors with low speed of only few MHz and therefore have limited address and memory space [8], [16]. The proposed concept allows for use of the latest technological products, as faster and the higher performance processor modules with 486, p5, p6 ... , while staying open to future improvement. Software is virtually independent of the hardware of the system. Hence, software with advanced capabilities can also be used. As a result the system is very flexible and upwardly compatible for both hardware and software upgrades. This approach offers all the benefits of the use of the latest products, at the same time cutting the cost of the system from day to day. High speed processors with huge memory space, together with higher speed of transmission of data significantly improve the speed and capability of the system. The limitation present in most of the existing CIS regarding the number of elements that can be used is almost completely avoided.

The low cost solution based on commercially available components

The proposed CIS is a low cost solution based on commercially available microcomputer hardware. Instead of use of specially designed hardware, railways can now benefit from general industrial development, without disadvantages related to maintenance because of fast advances in technology. This is achieved by using a high level programming language instead of an assembler language which is dependent on a particular processor [11]. Therefore software is independent of hardware. As a result hardware can follow advances of technology and the price of the system can be reduced.

4. Software

In this paper special attention is paid to interlocking software. The interlocking software is divided broadly in two groups: application driven data and general interlocking software. Application driven data are used to define the layout of the station and to create the computer control table. The most important part is general interlocking software which is independent of the application.

4.1 Definition of the control table

Control table design is the most important task in the design of an interlocking system. All operational, functional, and most importantly the safety requirements are listed in the control table. There are many different methods of presentation of control tables. They are understandable only by signalling engineers. The translation of control tables into a form acceptable for use by the computer system should be done by them, since it is a very significant, difficult and time consuming task. Especially if we take into consideration that they would need only very limited computer knowledge to design the control table and to present it in the required format. On the other hand, for anyone else, would require a big effort to get familiar with the railway signalling field.

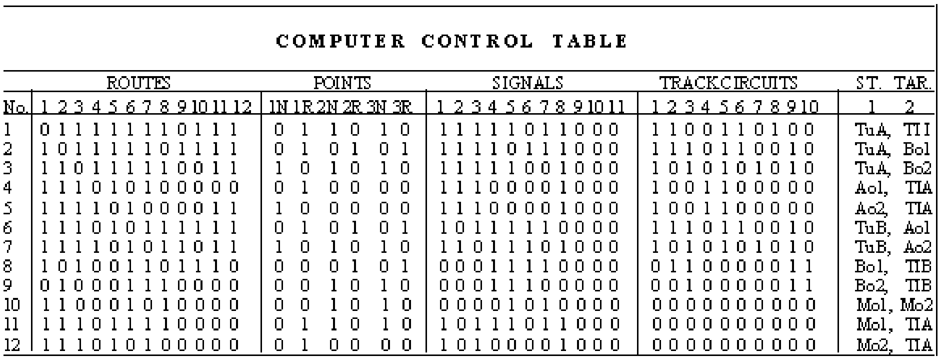
For most of the existing CIS a traditional control table is retained and slightly modified to suit a computer application. A specific design language has been developed for use by signal design engineers in describing the requirements of particular installations [8]. Therefore the data preparation process and their transformation in the form which is readable by computer are time consuming tasks.

The proposed control table, presented in Table 1, resolves many of the problems mentioned above. The table is ready for computer application immediately. Some additional effort is required from a designer to fill out the proposed control table form, but releases him/her to do other tasks until the commissioning of the system.

A control table designed this way becomes a simple data base for a CIS. The complexities of the various traditional control tables are avoided. The ability to define their own requirements will be given to the signal engineers from the country, but their outcome will be the control table presented in the proposed form and ready to enter into the computer.

The interlocking safety principles, rules and regulations are very complex and particular. A brief explanation is given to provide a better understanding of the control table design. Some parts of the control table, such as approach time locking, time release etc. are omitted to simplify presentation and not to confuse the average reader by complex interlocking requirements. The most important facet is that all the requirements will be stated in a similar manner.

The control table consists of all routes required for the station. Each route is dependent on the states of other routes and the positions of the relevant elements. The route requires clear track circuits, points set to the correct position and clears the signal to allow the move. The most important safety principle that defines interlocking between routes requires that any two routes can not share any portion of the track. If two routes have a shared portion they are conflicting routes and can not be set at the same time. The proposed control table is based on this principle and all other applicable safety principles [28], [19].



COMPUTER CONTROL TABLE

	ROUTES												POINTS						SIGNALS											TRACKCIRCUITS										ST. TAR.	
No.	1	2	3	4	5	6	7	8	9	10	11	12	1N	1R	2N	2R	3N	3R	1	2	3	4	5	6	7	8	9	10	11	1	2	3	4	5	6	7	8	9	10	1	2
1	0	1	1	1	1	1	1	1	0	1	1	1	0	1	1	0	1	0	1	1	1	1	1	0	1	1	0	0	0	1	1	0	0	1	1	0	1	0	0	TuA,	TII
2	1	0	1	1	1	1	1	0	1	1	1	1	0	1	0	1	0	1	1	1	1	1	0	1	1	1	0	0	0	1	1	1	0	1	1	0	0	1	0	TuA,	Bo1
3	1	1	0	1	1	1	1	1	0	0	1	1	1	0	1	0	1	0	1	1	1	1	1	0	0	1	0	0	0	1	0	1	0	1	0	1	0	1	0	TuA,	Bo2
4	1	1	1	0	1	0	1	0	0	0	0	0	0	1	0	0	0	0	1	1	1	0	0	0	0	1	0	0	0	1	0	0	1	1	0	0	0	0	0	Ao1,	TIA
5	1	1	1	1	0	1	0	0	0	0	1	1	1	0	0	0	0	0	1	1	1	0	0	0	0	1	0	0	0	1	0	0	1	1	0	0	0	0	0	Ao2,	TIA
6	1	1	1	0	1	0	1	1	1	1	1	1	0	1	0	1	0	1	1	0	1	1	1	1	1	0	0	0	0	1	1	1	0	1	1	0	0	1	0	TuB,	Ao1
7	1	1	1	1	0	1	0	1	1	0	1	1	1	0	1	0	1	0	1	1	0	1	1	1	0	1	0	0	0	1	0	1	0	1	0	1	0	1	0	TuB,	Ao2
8	1	0	1	0	0	1	1	0	1	1	1	0	0	0	0	1	0	1	0	0	0	1	1	1	1	0	0	0	0	0	1	1	0	0	0	0	0	1	1	Bo1,	TIB
9	0	1	0	0	0	1	1	1	0	0	0	0	0	0	1	0	1	0	0	0	0	1	1	1	0	0	0	0	0	0	0	1	0	0	0	0	0	1	1	Bo2,	TIB
10	1	1	0	0	0	1	0	1	0	0	0	0	0	0	1	0	1	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Mo1,	Mo2
11	1	1	1	0	1	1	1	1	0	0	0	0	0	1	1	0	1	0	1	0	1	1	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	Mo1,	TIA
12	1	1	1	0	1	0	1	0	0	0	0	0	0	1	0	0	0	0	1	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	Mo2,	TIA

The route and non conflicting routes are represented by a "0". The conflicting routes are represented by a "1". Points are divided into normal and reverse columns and the required states are denoted by a "1". Track circuits required to be clear are denoted by a "1". The conflicting signals are represented by a "1" and non conflicting signals by a "0". Starting and destination points are defined and entered into the table as well.

4.2 Input of the control table

A user friendly program has been developed to make input of the control table related data such as: number of routes, number of elements, values of the timers etc. in an interactive fashion with confirmation. The control table itself will be entered into the software as a data file in matrix form as prepared by the signal design engineer. Additional programming is not necessary. This way is chosen to allow the simple and easy input of data, as well as to simplify checking and the quality assurance (QA) process. As entered, the version of the data will be produced by the program to allow immediate checking and corrections until a version free of errors is obtained.

4.3 Conversion of the control table to general interlocking software

The programming of the control table presented as stated above becomes very simple. Any control table can be presented as a set of interlocking functions FI_i in Boolean form:

$$FI_i = FR_i \times FP_i \times FS_i \times FT_i, \quad i = 1, \dots n,$$

where:

n - is the total number of the routes,

FR_i - is the interlocking subfunction between route No. i and the other routes,

FP_i - is the interlocking subfunction between route No. i and points condition,

FS_i - is the interlocking subfunction between route No. i and state of the signals,

FT_i - is the interlocking subfunction between route No. i and of the track circuits.

The subfunctions are defined as follows:

$$FR_i = \prod_{j=1}^{k} R_j,$$

where k is the total number of routes Rj interlocked with route No. i.

$$FP_i = \prod_{j=1}^{l} P_j(N \text{ or } R),$$

where l is the total number of points $P_j(N$ or $R)$ interlocked with route No. i.

$$FS_i = \prod_{j=1}^{m} S_j,$$

where m is the total number of signals S_j interlocked with route No. i.

$$FT_i = \prod_{j=1}^{o} T_j,$$

where o is the total number of track circuits T_j interlocked with route No. i.

An example of definition the interlocking of route No. 2. from the control table is given. The route starts from entrance signal Au and extends up to the exit signal Bo1. The appropriate interlocking function of the route is:

$$FI_2 = FR_2 \times FP_2 \times FS_2 \times FT_2,$$

where subfunctions have the following values:

$$\begin{aligned}
&FR_2 = R_1 \times R_3 \times R_4 \times R_5 \times R_6 \times R_7 \times R_9 \times R_{10} \times R_{11} \times R_{12},\\
&FP_2 = P1R \times P2R \times P3R,\\
&FS_2 = S_1 \times S_2 \times S_3 \times S_4 \times S_6 \times S_7 \times S_8,\\
&FT_2 = T_1 \times T_2 \times T_3 \times T_5 \times T_6 \times T_9.
\end{aligned}$$

Interlocking functions and subfunctions for all other routes can be developed accordingly.

The described functions are general and they are included in the appropriate subprograms. They are part of the general interlocking software and independent of the track layout, but they are defined with data from the computer control table which defines the particular configuration. Therefore the design task is greatly simplified.

To set the route the main program checks whether the interlocking function and subfunctions of the route are satisfied (true) and if the route is available it gives a command for the elements to be set accordingly.

4.4 Minimization of the interlocking subfunctions

Total minimization of the interlocking subfunctions is not possible, but by considering the variables some improvements can be achieved. Most complexity is introduced by interlocking between the routes. It is easily noticeable even for a small station. For complex stations this will be exaggerated.

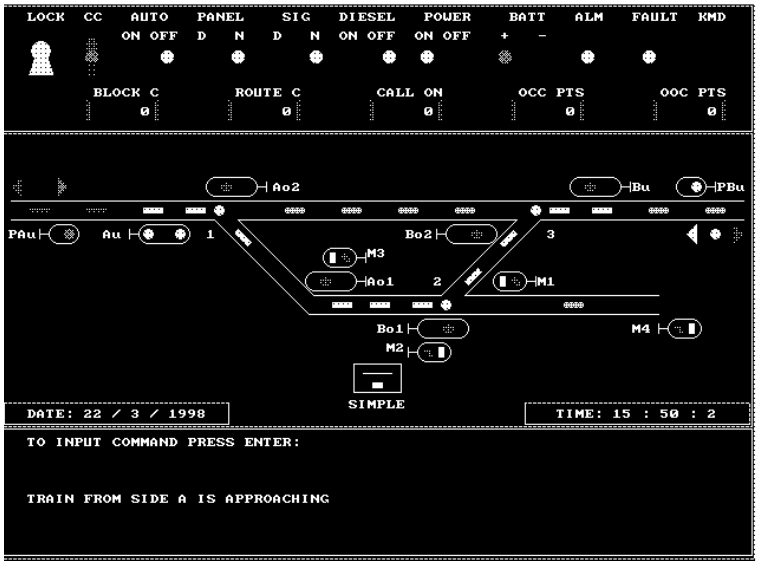
If we consider the routes that can be set at the same time rather then those that cannot, the interlocking subfunction between routes is simplified. This is illustrated in the previous example of route No. 2. If we form the subfunction FR_i' which has the same form as FR_i, but now k is the total number of routes that are not interlocked with the route No. i, the function for route No. 2 will become:

$$FR_2' = R_2 \times R_8.$$

This means that only one route can run simultaneously with route No. 2. From this simple example it is clear that the number of variables can be significantly reduced and therefore the memory space required and checking time within the program can be reduced.

4.5 Development of screen layout

A user friendly program has been developed to make the design of the screen layout as easy as putting the predefined indication element modules together to form the appropriate picture. All the required elements have been developed and they are available in an elements library. The elements are placed by specifying coordinates to suit the signalling arrangement. The development of the a station layout, even for very complicated configurations, is reduced to only few hours. The VDU Layout of the analyzed typical railway station is presented in Figure 5.



DATE: 22 / 3 / 1998

TIME: 15 : 50 : 2

TO INPUT COMMAND PRESS ENTER:

TRAIN FROM SIDE A IS APPROACHING

Figure 5.

The VDU indications are provided by an extra care. The actual indication, as in the field, including flashing aspects, will be available to the signalman all the time. This can be an improvement to the existing CIS and RIS. Generally the systems do not differentiate between proceed aspects, and only one proceed indication is given. This could, for example, disable a signalman from undertaking preventive action in the case of failure. In some systems flashing aspects are indicated by adding a flashing letter next to the aspect. The non-appearance of the letter, which could be caused by an error in the indication part of the system , could confuse a signalman.

The indications are not the fail–safe portion of the system, because a system itself must be fail–safe, but they are a very important basis for the signalman's decisions. Therefore they should be reliable and accurate. The proposed system has the indication of the RGB primary colors on the screen all the time as required by ORE UIC recommendations [36].

4.6 General interlocking software

The basic idea is to use, as far as possible, knowledge and experience of the signal design engineer and implement that into the software of the system. This way the system incorporates expert knowledge, basic signalling principles and common railway rules and regulations, and replaces the signal design engineer through all the phases of the design of an interlocking system

The software of the system is designed in the form of a main program and various functionally oriented subprograms, representing the synthesis of safety as well as functional and logical requirements based on the single–channel operating principle. The general–purpose interlocking program embodies most of the standard signalling principles and contains the rules for operating on the control table data to produce the precise signalling controls required. This enables easier safety, functional and logical analysis of the software. A simplified block diagram of the main program is shown in Appendix A.

Railway signalling principles

Railway signalling knowledge and experience is expressed through safety railway signalling principles. The general principles are valid for all systems regardless of the country of the application of an interlocking system. There are some differences between different countries systems, but they are more functional and operational than safety in nature. The principles are complex and unique to the railway signalling field, but it is important that they can be generalized and converted into software with the aim of developing a general CIS [28].

Signalling principles and safety functions are in–built in completely developed general interlocking software. The designer is released from that task. By implementation of the control table and track layout all jobs are done. The program itself puts all parts together and makes all required interlockings.

Route approach

There are, generally, two well known ways of designing the signalling circuits: "geographical" and "free-wired". The "geographical" system has in–built redundancy which contributes to generality and makes the design easier, but increases the cost of hardware. The "free-wired" system does not have spare parts and therefore hardware is less expensive, but the design is more expensive due to need to solve the problem from case to case. It is important to emphasis that there is no basic difference between the system's functions. The pattern is a chain of self–contained circuits operating in cascade, each performing a function and passing the result to the next in the chain.

Route approach is utilized in the software. The route concept combines both general "geographical" approach and efficient "free wired" approach. It allows generalization without spare parts whether in hardware or in the software. Direct correlation is established between the interlocking function and the route. Interlocking functions and subfunctions are written very simple, but they are used extensively in many different ways in various software modules. Software modules are developed to cover all "chains". Program flow is designed to follow the approved cascade operation of relay circuits.

It is impossible here to give a comprehensive account of the structure of all software modules and how the interlocking functions have been used. It is hoped that use of interlocking functions in the route setting module will serve to illustrate the principle. A block diagram of route setting algorithm is presented in appendix B. Required activities are listed below:

- Route R_i is determined by the request which defines the starting point of the route and its destination. The starting point and the destination point are unique for the route. Table 1 shows the correlation between the route's number and starting and destination point.
- If R_i route is an exit route than the block section up to the next station has to be available and locked for the direction of the route.
- Route R_i has to be normal (not engaged) at the time the route has been requested. This will prove that the previous movement allowed by the route has been completed.
- All conflicting routes must be normal. This condition will be proved by satisfying the interlocking function Fr_i.
- The starting signal of the route has to be in normal (showing red aspect) conditions. This will prove that the signal is operational and not in use by any other route.
- Protection of the route by proving specified signals normal will be achieved by satisfying the interlocking function FS_i.
- Proving that the portion of tracks which belongs to route R_i is vacant will be done by satisfying the interlocking function FT_i.
- Checking of points condition will be done in accordance with the interlocking function FP_i. First, all specified points will be proved unlocked and able to operate. Then, all points detected in positions which are not correct for the route will be called and moved to the correct position. At the end, all points have to be detected in their correct positions to satisfy the interlocking function.

- Checking of all requirements will be proved by satisfying the overall interlocking function FI_i.
- If the request to set route R_i has been canceled then further actions will not take place. The request for cancellation can be accepted if it has been received up to this stage, before the route locking takes place.
- After all requirements have been satisfied, locking of route Ri can take place. Points locking will be done in accordance with the interlocking function FP_i. Locking of track sections will be done in accordance with the interlocking function FT_i. Conflicting routes will be locked in accordance with the interlocking function FR_i. After all the required locking has been done the route itself will be locked too.
- The starting signal of route R_i has to be set to proceed aspect. This will be done by the appropriate subprogram. The aspect will be given taking in account aspects of relevant signals ahead and speed restrictions within the route. The country specific requirements will be implemented in the module. The rest of the route setting module is generalized.
- If the signal has a repeater controlled by the signal, then the repeater's aspect will be set accordingly.
- If relevant signals behind the signal have already been set their aspects will be changed accordingly.

All these main activities are required in the aim to set the route. Beside these, there are many other activities which have to be performed. The other activities have not been mentioned here, because the increase in complexity would not help to better understanding of the matter.

4.7 Automatic route setting

The automatic route setting releases the signalman from unnecessary presetting of the elements required by the route and gives him/her time for other activities. The remote control of an interlocking is simplified due to the use of the one route command instead of a set of commands for the elements and a command for the route itself. The automatic setting of the elements in the route, when that route is called, is a feature in–built in the general interlocking software of the proposed CIS. The feature is achieved in the interlocking software rather than by the panel processor software as in most of the existing systems [8], [40], [42]. This approach gives functional consistency on the route level inside the interlocking as well as individual manipulation of the element if required.

4.8 Modular structure

The CIS described in this paper adopts a modular structure of hardware and software as well as a structured programming language application which is implemented as recommended by ORE UIC [4], [5], [35], [36], [37]. The system can be built to required capacity from modules and configured to suit the application. Thus, only the amount of hardware actually required for an installation needs to be supplied. Modular structure minimizes cost and size of the system, maximizes flexibility and makes the system easy to maintain.

Program modules have been designed in the form of short independent and functionally driven subprograms using a high level structured programming language. The main program calls required subprograms to fulfill functional requirements. This makes software checking, testing and validation easier and faster.

Modular design of hardware and software makes the applications for various countries more adaptable. The country specific railway signalling rules will require alterations of only a few subprograms with little effort and without virtually any influence on the system. For example, country-based aspect diversities can be resolved by designing the appropriate subprograms for the aspects.

4.9 Advanced simulation

The simulation and testing in laboratory conditions is simplified. There is no requirement for a special work station and thick manuals. All activities can be done on the system itself, or any PC, with simulation software which is independent of the hardware. The simulation software is actual general interlocking software which uses simulated software modules as inputs. The trackside equipment can be tested with the system, or separately by the simulation of outputs and checking of the received inputs. For this purpose the latest results of the checking of the real time controlled objects can be applied. The system is dealing with the elements as controlled objects through input and output registers making the hardware independent of the control system requirements. This means that various elements found in different countries can be connected to the system by provision of appropriate low cost interfaces. Thus the proposed system can be utilized for various countries without significant alterations to the hardware and almost no alteration to the software.

The simulation software can be used as a powerful independent checking tool for the checking of the conventional control table design. This will uncover many problems before installation and save time and money, especially for the conventional RIS, where changes affect hardware.

The simulation software can be a very useful training tool for young designers and checkers, giving them the possibility to correct their own mistakes and develop their skills. This is particularly important, bearing in mind that experienced signalling engineers are very hard to find and that there were no universities providing railway signalling system training until recently [38].

4.10 Verification and validation

Verification and validation of the safety critical software is required. This task has to be done in accordance with the latest standards [5], [13]. Part of the process should be safety analysis of the general interlocking software in accordance with the safety analysis of RIS. This proving should be done by experienced signal design engineers [41]. Validation of the software modules, representing special requirements of one country, can be done by signal design engineers from that country to allow the best translation into the software. Once software of the system has been approved, design becomes easy and simpl. Instead of development of advanced techniques to make data preparation and checking easy [7], design, checking and testing are significantly reduced.

5. Advantages of the system

5.1 Flexibility

The hardware and software modules used in the system are flexible and can be configured to suit the most diverse of customer requirements. The control table exactly determines the size of the software and hardware of the system. There is no need to provide superfluous hardware and software modules for predefined size of an interlocking system. Therefore the size of the proposed CIS is optimized to an application.

5.2 An expert knowledge consisting system

The most significant advantage of the proposed system is the much greater implementation of railway interlocking safety principles and general knowledge and experience of the signal design engineer into the software of the system. This approach makes the system design more independent of the signalling knowledge and reduces the need for the presence of signal engineers who are not easily available. The reason for the shortage of signal design engineers and therefore high cost, is that they are getting their skills only through work experience since there was no opportunity to get them educated in universities until recently [38]. On the other hand, for data preparation of existing CIS, computer knowledge is very important. This excludes a significant number of the signal design engineers who can easily handle relay circuits [18]. The proposed solution overcomes this disagreement by dividing the job at a natural border. A valuable experienced signal design engineer can continue to deal with principles and safety requirements including validation of the system. Young engineers can do programming and try to get as much of the safety principles knowledge as they can.

Of course, the signalling arrangement and control table still have to be produced by a signal design engineer. The signalling arrangement will be the basis for the design of the station layout for the VDU. The control table will be designed and presented as in Table 1, by a signal design engineer to be used as a knowledge base for the system. All other stages of the design and the development of the system, until the final testing, can be undertaken by programmers. For the final testing and commissioning, the presence of an signalling expert is required. The final test will prove all safety locking and functional requirements against the control table and point out alterations to be made until complete satisfaction of the design requirements. This process is very similar to checking of existing relay interlockings. Therefore a checker does not have to be an computer interlocking expert, signalling knowledge alone is sufficient. Considering the very short time required for design and checking activities, all the required corrections can be done on the same day. This reduces unexpected delays to the commissioning and allows better planning of the whole project.

5.3 Quick design, checking, testing, commissioning and quality assurance

Data preparation and input time is greatly reduced. The programmer will be asked by the program to input all required data. Through this process all geographical and other relevant data will be entered. The same method will be used to develop the layout for the VDU. The time for design of the station layout, even for very complicated configurations, is reduced to only few hours.

The control table design time requires only a few hours. As the control table has already been presented in the way required for computer implementation, its input is as simple as input of an ordinary data file. The computer will than print the entered control table data. The control table produced by the computer will have the same format and layout as the originally designed one. This will make checking as easy as a simple comparison of two numbers. The appropriate corrections can be made straight–away and the checking process is repeated until the computer generated control table is exactly the same as the original one. An error free control table can be obtained by two independent checkers, minimizing the subjectivity of the checkers. The time required for this process is dependent on size of the control table, but in most cases can be done within few hours.

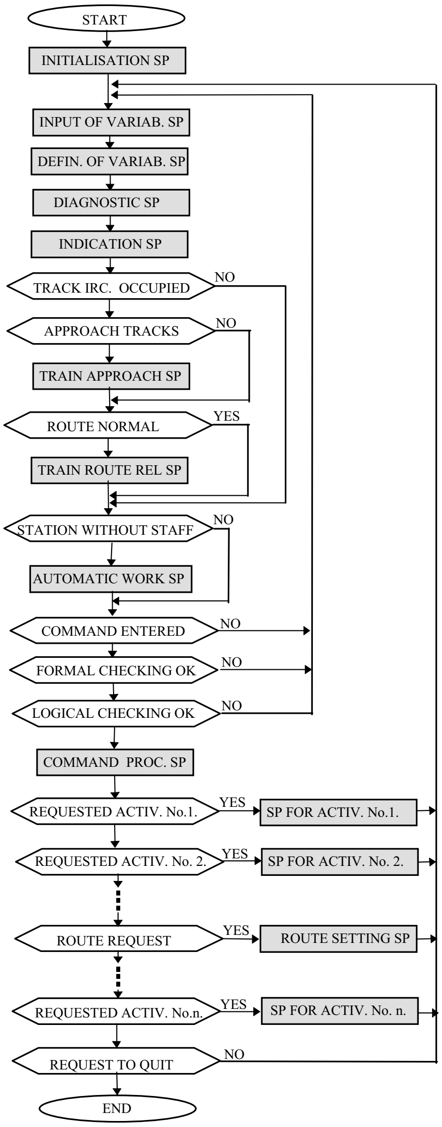
The same procedure is applicable for further alteration of the control table caused by the alterations after the commissioning. This way is very easy, fast and less expensive then the alterations of the existing CIS. The paper work, design and checking documentation are minimized too by the proposed presentation of the control table.

6. Conclusions and further work

A universal CIS with commercially available computer hardware and general interlocking software written in high level structured programming language independent of the hardware has been proposed. The safety of the system is based on the, proven by practice, triple redundant system with defined repair time and appropriate safety techniques. Trackside equipment interfaces are developed as the interfaces to the real time controlled objects and generalized to suit a various types of elements used by different countries. The system is upwards compatible and open for further development and alteration to satisfy the most diverse railway requirements. The hardware and software are designed on a functional module basis to allow all advantages of the modern concept of fault–tolerant real time controlled systems. The most important railway signalling knowledge and experience is implemented into the software of the system making it general and independent of the layout of the railway station. A simple and easy method of definition of the control table and VDU layout design is proposed to make the development of the system very quick, cutting the cost of the whole project. The design, checking, testing and commissioning time is reduced to a few days from over a month.

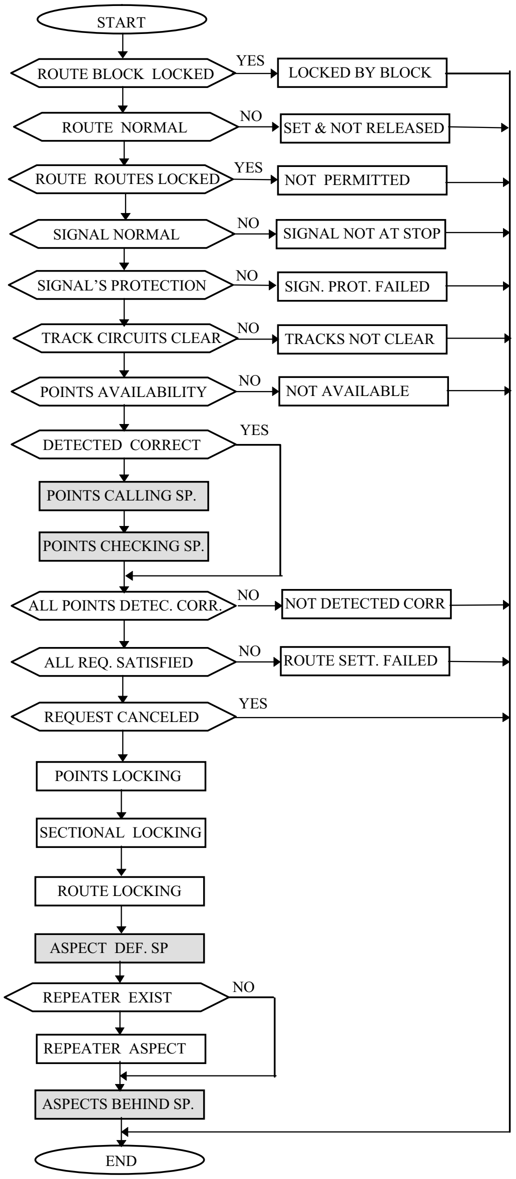
Further work is required to perform validation and verification of the proposed safety critical software. Selection of the most efficient redundancy management software, diagnostic and self diagnostic software will be required. Further improvement and standardization of both hardware and software modules could also be considered.

Appendix A



A simplified flow chart of the main program

Appendix B



A flow chart of route setting algorithm

REFERENCES

1. Z. Avramovic, D. Lutovac: *Fail–safe technics and concepts of microprocessors railway interlocking systems.* XXXII Yugoslav Conference on electronic and telecommunications - ETAN, Sarajevo 1988 (in Serbian).
2. Z. Avramovic, D. Lutovac: *Railway electronic interlocking systems.* Symposium Yugoslav Science Association - JAZU, Zagreb, 1988. pp. 271–274, (in Serbian).
3. J. Berger, P. Middelraad, A. Smith: *EURIS: European Railway Interlocking Specification.* UIC 7A/16, Utrecht, 19 May 1992.
4. BRB/LU Ltd./RIA Technical Specification No. 19.: *Technical Requirements Standard for Non–Safety Signalling Software.* 1988.
5. BRB/LU Ltd./RIA Technical Specification No. 23.: *Safety Related Software for Railway Signalling.* Consultative Document 1991.
6. S.R. McConnel, D.P. Siewiorek: *Synchronization and Voting.* IEEE Transactions on Computers, Vol. C–30, 1981, pp. 161–164.
7. A.H. Cribbens, I.H. Mitchell: *Railway Engineers Forum Meeting.* The Application of Advanced Computing Techniques to the Generation and Checking of SSI Data, 6th November 1991, London.
8. A.H. Cribbens: *Microprocessors in railway signalling: the Solid–State Interlocking.* Microprocessors an Microsystems, Vol. 11, No 5, pp. 264-272, June 1987.
9. ERICSSON: *Interlocking Processing Unit JZH 850.* Printed in Sweden Satherlund & Krook 85. LZT. 11810.Ue.
10. ERICSSON: *Fully Electronic Interlocking.* Printed in Sweden Satherlund & Krook 84. LZT. 11814.Ue.
11. A. Goltz: *Safety Demands Formal Software Engineering.* Railway Gazette International, pp. 495–497, July 1986.
12. D.V. Hall: *Microprocessors and interfacing, programming and hardware.* McGraw–Hill, 1986.
13. IEC 65A (Secretariat) 122 Working Group 9: *Software for Computers in the Application of Industrial Safety–Related Systems.* August 1989.
14. B.W. Johnson: *Fault–tolerant Microprocessor–based Systems.* IEEE Micro, Vol. 4, December 1984, pp. 6–21.
15. B.W. Johnson: *Design and Analysis of Fault–tolerant Digital Systems.* Addison Wesley, Reading Mass., 1989.
16. A. Katsuji: *Practical Use of Computerized Interlocking System "SMILE" in JNR.* Japanese Railway Engineering No. 94, pp. 21–24, June 1985.
17. P.K. Lala: *Fault–tolerant and Fault–testable Hardware Design.* Prentice Hall, N.J., 1985.
18. D. Lamb D, R. Davies: *Are Microprocessors and Signal Engineers Incompatible?.* IRSE Conference, 14th February, London, 1995.
19. D. Lutovac: *Microprocessors control of railway interlocking systems in the passengers stations.* MSc Thesis, Faculty of Electrical Engineering, University of Belgrade, 1988 (in Serbian).

20. D. Lutovac: *Concept of the microprocessor interlocking system of Kirilo Savić Institute.* International Symposium IKS Belgrade 1988 (in Serbian).

21. D. Lutovac: *Fail–safe central processing unit for the railway microprocessor interlocking system.* XXXIII Conference ETAN, Novi Sad 1989. part VIII, pp. 105–112 (in Serbian).

22. D. Lutovac,D. Živković: *Model for the estimation of the reliability, safety and availability of the microprocessors railway interlocking system.* JUREMA, Šibenik 1989. part IV, pp. 32–34 (in Serbian).

23. D. Lutovac: *The fail–safe railway computer based interlocking system.* JUREMA, Šibenik 1989. IV, pp. 15–20 (in Serbian).

24. D. Lutovac: *Microprocessor interlocking system as a factor of the railway traffic safety.* XI Yugoslavian Symposium on electronic of traffic, LJubljana 1989. pp. 37–40 (in Serbian).

25. D. Lutovac: *Outside devices for station microprocessor interlocking system.* XXXIV Conference ETAN, Zagreb 1990. part II–III, pp. 177–184 (in Serbian).

26. D. Lutovac: *Control table transformation for microprocessor's control of railway signalling system.* XXXV Conference ETAN, Ohrid 1991. part IX, pp. 505–512 (in Serbian).

27. D. Lutovac: *Software for a railway station microcomputer interlocking system.* XXXVI Conference ETAN, Kopaonik 1992. pp. 505–512 (in Serbian).

28. D. Lutovac, Z. Avramovic: *Fail–safe British Rail Signalling Principles.* Železnice, Year 51, No. 5, Belgrade, May 1995, pp. 488–494. (in Serbian).

29. D. Lutovac, T. Lutovac: *Hardware of an advanced computer interlocking system.* Proceedings of JUŽEL - The 3rd International Scientific Conference of Railway Experts, Nis, Yugoslavia, 3–4 Oct. 1996, pp. 91–95.

30. D. Lutovac: *Universal Computer Interlocking System.* IRSE Australasian Conference, Launceston, Tasmania, 15–16 Nov. 1996.

31. D. Lutovac, T. Lutovac: *Software of an advanced computer interlocking system.* Proceedings of JUŽEL - The 4th International Scientific Conference of Railway Experts, V. Banja, Yugoslavia, 2–4 Oct. 1997, pp. 74-79.

32. W.F. McGill, S.E. Smith: *Fault Tolerance in Continuous Process Control.* IEEE Micro, Vol. 4, December 1984, pp. 22–33.

33. V.P. Nelson, B.D. Caroll, eds.: *Tutorial: Fault–tolerant Computing.* IEEE Computer Society Press, Washington DC, 1987.

34. ORE UIC: *Safety and reliability considerations for individual warning installations.* Utrecht, 1983.

35. ORE UIC: *Individual warning systems for personnel working on the track.* Utrecht, 1984.

36. ORE Committee A118, UIC: *Reports: 1–12.*

37. ORE Committee A155, UIC: *Reports: 1–13.*

38. Railway Gazette International: *Postgraduate study & Continuous professional development in Rail Systems Engineering.* pp. 312, May 1995.

39. B.D. Rutherford: *Fail–safe microprocessor interlocking - an application of numerically integrated safety assurance logic.* Railway Safety Control and Automation

Towards the 21st Century, IRSE International Conference, London, Sept. 1994, pp. 72–76.

40. H.A. Ryland: *WESTRACE - A Second Generation Electronic Interlocking.* International Conference on Advanced Railway Control "Aspect 95", Sec. 10. pp. 35–40, IRSE, London, 25–27 September 1995.

41. R.C. Short: *Software validation for a railway signalling system.* in Bayliss J A (ed.) Proc. IFAC Conf. Safety of Computer Control Systems, Cambridge, UK (1983) pp. 183–193.

42. SIEMENS A G: *SICAS - The New Microcomputer Interlocking System.* 1995.

43. D.P. Siewiorek, R.Z. Swarz: *The theory and practice of reliable system design.* Digital Press, Bedford, Massachusetts, 1982.

44. N. Storey: *Safety–Critical Computer Systems.* Addison Wesley Longman, England, 1996.

45. H. Strelow, G. Scharfenberg: *Das Mikrocomputersystem SIMIS/MES 80.* Signal+Draht 75 (1983) Hft 12, S.229–234 (in German).

46. D.B. Turner, R.D. Burns, H. Hecht: *Designing micro–based systems for fail–safe travel.* IEEE Spectrum, pp. 58–63, 1987.

47. J.H. Wensley et al: *SIFT: Design and Analysis of a Fault–Tolerant Computer for Aircraft Control.* Proceedings of the IEEE, Vol. 66, No. 10, October 1978.