15th Triennial World Congress, Barcelona, Spain

A BOTTOM-UP METHODOLOGY FOR TESTING COMPLEX CONTROL FUNCTIONS OF PROCESS AND POWER PLANTS

Emanuele Carpanzano

Institute of Industrial Technologies and Automation
National Research Council
V.le Lombardia 20/A, 20131, Milan, Italy
Email: e.carpanzano@itia.mi.cnr.it

Luca Ferrarini, Claudio Maffezzoni

Dipartimento di Elettronica e Informazione
Politecnico di Milano
P.za L. da Vinci 32, 20133 Milan, Italy
E-mail: ?ferrarin,maffezzo?@elet.polimi.it

Abstract: In the present work a modular simulation-based technique for the automatic verification of logic control functions is introduced and exploited to define a structured bottom-up methodology for the testing of the overall control functions of process and power plants. The proposed method has been implemented in a CACSD environment which uses the Matlab toolboxes Simulink and Stateflow for the control functions testing via simulation. The testing of the logic control functions of a portion of a thermal power plant is considered as an application example of the proposed framework. *Copyright © 2002 IFAC*

Keywords: power plants, testing, simulation, CACSD, logic control.

1. INTRODUCTION

The problem of control system testing, both at the design stage and at the commissioning stage, is a very critical task within the control system design of modern industrial plants (Mok and Stuart, 1996, Carpanzano, Ferrarini, Maffezzoni, 2001). This mainly because of the intrinsic complexity due to the non trivial interaction among process components and the large number of input and output signals involved. Furthermore, the control functions are distributed throughout the plant, and characterized by complex hierarchical structures (Dieterle, Kochs and Dittmar, 1995, and Maffezzoni, Ferrarini and Carpanzano, 1999). So, a structured method is needed, which allows the testing of the control functions in a modular and hierarchical way according to the structure of the control system (Gravenstein, 1994.). In this respect, CACSD (Computer Aided Control Systems Design) tools that effectively support the engineer in the development and testing of an industrial control system, in the process of specification and design, are also becoming more and more necessary (James, Cellier, Pang, Gray and Mattsson, 1995, and Maffezzoni, Ferrarini; and Carpanzano, 1999). Recent approaches apply formal methods (Holzmann, 1997, and Mok and Stuart, 1996), and though promising, they need further study before finding large consensus in the industrial field.

In the present paper a modular simulation-based technique for the automatic verification of logic control functions is first presented. According to such a technique, whenever a single module of the control system has been designed, the designer can easily build up a testing scheme to verify the compliance of the designed control module with respect to its specification (Carpanzano, Ferrarini, Maffezzoni, 2001). Such a scheme is defined by properly parametrizing simple templates, that represent a closed-loop system, where the controlled process is suitably simplified. Then, an automatic procedure can be launched to automatically test the designed module. Such a framework is then exploited to define a structured bottom-up methodology to test the

overall control functions of process and power plants. The proposed method has been implemented in a CACSD environment which, for the control functions testing via simulation, employs the Matlab environment (*Using MATLAB*, 2000), and in particular, the toolboxes Simulink (*Using MATLAB*, 2000) and Stateflow (*Stateflow User's Guide*, 2000). The testing of the control logic functions of a portion of a thermal power plant is considered as an application example of the proposed technique.

2. THE BASIC TESTING SCHEME

In (Carpanzano, Ferrarini, Maffezzoni, 2001) a method for the automatic verification of logic control functions with Matlab is proposed. According to such a method the scheme reported in Fig. 1 is introduced to check by means of simulations that a designed control module (CM) is coherent with its specification. Therefore a model of the controlled object (VP, Virtual Plant) has to be defined. A second problem to face is how to evaluate the correctness of the results obtained by simulating the CM by comparing such results with the desired ones, using a suitable Simulink model denoted in the above scheme with CA (Comparison Analyser). More in detail, CM is the logic control function to be tested: it is specified with Boolean logic operators in the CACSD tool, and its description through standard Simulink blocks is automatically generated by the tool. VP represents a simplified model of the controlled sub-system, to be specified by the user by properly connecting simple blocks defined through Stateflow charts. Finally, CA contains a formal description of the desired behaviour, whose definition is up to the designer, and compares the simulation results with such a desired behaviour. The last may be properly specified by means of the Stateflow toolbox, too.

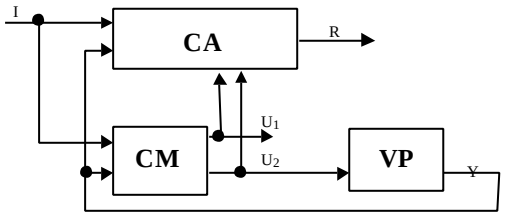
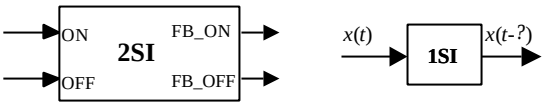


Fig. 1. The testing scheme.

The VP block describes the system portion controlled by the considered CM; usually, such a system portion is constituted by both process and control components. In order to test the CM it is not necessary to define an accurate model for the controlled object, with detailed description of physical phenomena and of control function execution. Often, a simple model that represents the feedback reaction of the field to the CM commands is enough for the purpose. In particular, in the application to power plants, only two types of basic blocks have been found to be necessary for creating VP block:

1) two-signal interaction block (called 2SI, Fig. 2a), where the CM sends a pair of commands (OF and OFF) to VP and VP reacts with a pair of feedbacks (FB_ON and FB_OFF) corresponding to the actuation of the commands; suitable delays are introduced between the input and output pairs;
2) one-signal interaction block (called 1SI, Fig. 2b), where VP sends a delayed feedback to CM (in the figure, the delay is denoted with ?, while x is a generic variable defined over time t).

Both basic blocks are endowed with output override possibilities with exogenous signals, to account for faults or disturbances (not shown in Fig. 2 for simplicity). So, VPs are realised by proper interconnection of many instances means of the two blocks represented in Fig. 2.



(a) (b)

Fig. 2. 2SI (a) and 1SI (b) basic blocks to define VPs.

Block 2SI may be used, for example, to model the responses of an on/off valve, the activation and deactivation of a sequence, the start and stop of a pump, and so on. Similarly, block 1SI may be used whenever a single response is required.

The defined Simulink scheme can be used to automatically check that, for a given input value or sequence, CM computes the outputs according to the desired specification. An exhaustive testing of CM is often unfeasible for a significant portion of the control system, because of the excessive number of possible combination (Daga and Birmingham, 1995, and Gravenstein 1994). As a consequence, another problem to deal with is the definition of the *test cases*, i.e. of the input signals I sequences to generate when testing the CM by means of simulations.

In conclusion, once a CM has been designed, in order to apply the proposed testing method, the following problems have to be faced:

1. modelling the controlled objects (VP);
2. definition of the comparison criteria (CA);
3. generation of suitable histories of exogenous input signals.

Such problems have been addressed with reference to the testing of a generic CM in (Carpanzano, Ferrarini, Maffezzoni, 2001), and are not discussed here in detail for the sake of brevity.

3. THE BOTTOM-UP METHODOLOGY

To improve the design process, the control system is structured according to some hierarchy, suggested by plant functional decomposition (Taylor and Chenney,

1999). What is proposed here is the testing of a control scheme composed by more control modules through a bottom-up approach. In simple words, each hierarchical level can be composed of more CMs, and CMs are tested "one level at a time". First, CMs which are at the lowest level with respect to the functional hierarchy are tested according to the concept shown in Fig. 1. Therefore, the required VPs have to be defined to model the field response. Then, the higher level CMs can be tested assuming that CMs of the lower levels have been already tested. In such a case, VPs for testing the current level CMs can be realised directly through the already tested lower level CMs and the VPs representing the field of the lowest level.

To show the idea, consider the control scheme of Fig. 3 as an illustrative example, where three hierarchical levels may be identified.

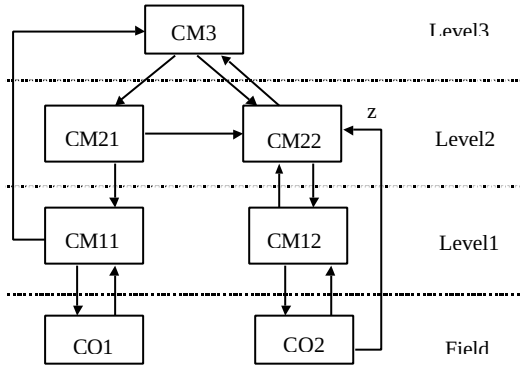


Fig. 3. A hierarchical control scheme (CM = Module, CO = Controlled Object).

When testing level 1 modules, the VP model consists of CO1 and CO2 only, while, when testing level 2 modules, CM11 and CM12 modules plus CO1 and CO2 are considered as VP model. Conversely, signals coming from the upper levels are dealt with as external inputs, while signals going to the upper levels are made available for CA block.

Following the above ideas, the complete bottom-up testing procedure for a control system can be realized through the following steps.

1. Preliminarily, each defined CM is individually tested according to the procedure derived in (Carpanzano, Ferrarini, Maffezzoni, 2001). In doing this, a suitable VP, a proper analysis method (CA), and a suitable test case generation method are defined for any considered CM. By this preliminary analysis local bugs, not depending on the interconnection with other CMs, can be found for ant CM. Of course, VPs' set-up for this preliminary local analysis are discarded in the subsequent bottom-up procedure, but for VPs at the lowest level.
2. The whole designed control system, i.e. all the designed CMs and their connections, is tested by going through its successive hierarchical levels according to the following bottom-up rule. Build the Testing Scheme of level 1 (say, TS_1) as the scheme of Fig. 1 with CM as the aggregation of all the control modules belonging to level 1 and VP (CA) as the aggregation[1] of VPs (CAs) used for the testing of individual control modules of level 1 (see point 1 above). Here, input signal I are either exogenous signals or signals coming from the upper level, while U_1 is the set of signals sent to higher levels. TS_1 can be considered as an aggregate block whose inputs are constituted of the vector-signal I and whose output is the collection of U_1, U_2 and Y. Once the whole level n-1 (n?2) is tested, define the Testing Scheme of level n (say, TS_n), as the scheme of Fig. 1 with:
   a) CM as the aggregation all the control modules belonging to level n;
   b) VP as the aggregation of TS_{n-1} with possible additional field models generating those feedbacks (from field) not yet involved in the lower levels (like signals z in Fig. 3);
   c) CA as the aggregation of CAs defined for the individual testing of the n-th level CM blocks (see point 1 above).

Note that the VPs' output of level n (named Y in Fig. 1) consists of the collection of signals (Y, U_2, U_2) of level (n-1) plus the output of the additional field models.

4. A CASE STUDY: TESTING THE LOGIC CONTROL OF A POWER PLANT

For the sake of example, in this section the presented technique is applied to an industrial plant. In particular, the testing of the control logic for the extraction system of a thermal power plant will be studied. The structure of the considered portion of the plant is shown in Fig. 4.

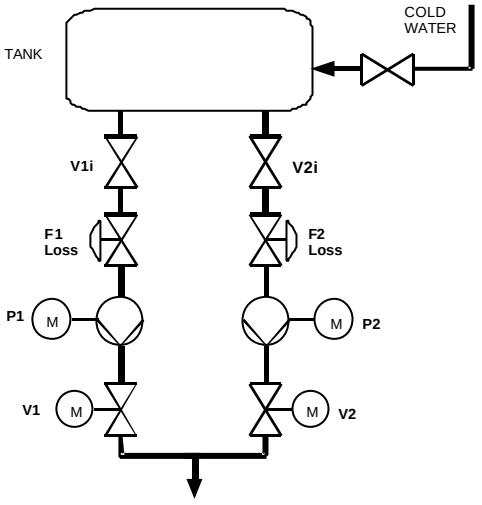


Fig. 4. Considered water extraction system.

[1] Usually a simple collection

The extraction system is constituted by two branches connected in parallel, and the flow in each branch is controlled by means of a valve (V1, V2) and a pump (P1, P2). In detail, the control logic is structured according to the scheme reported in Fig. 5, where four hierarchical levels may be distinguished: system control, branch selector, sequence control and drives. The different modules constituting the control system are shown, as well as their connections; exogenous binary signals are denoted by thick arrows. In particular, PCn represents a process condition, like valve open/closed or tank level<lmin, while Cn represents a signal sent by a pump or valve drive actuator, like actuator disturbed, MCn represents a signal sent by the control logic, e.g. pump vibrations>Vmax. The meaning of each signal is not reported for the sake of brevity. Notice also that the same exogenous signal may interest different modules, so such signals have to be generated properly when simulating the control scheme.

4.1 Testing CMs: the System Control module

All the CMs have been tested individually according to the method presented in section 3. As an example the scheme for the System Control module testing is reported in Fig. 6. The CA block is here not reported for the sake of brevity. Notice that the VP is realized by properly using the two basic blocks defined in section 2. Furthermore, starting from the CM specification the CA has been defined and suitable test cases have been generated semi-automatically, using the methods described in (Carpanzano, Ferrarini, Maffezzoni, 2001). So, the CM correctness has been verified through simulations.

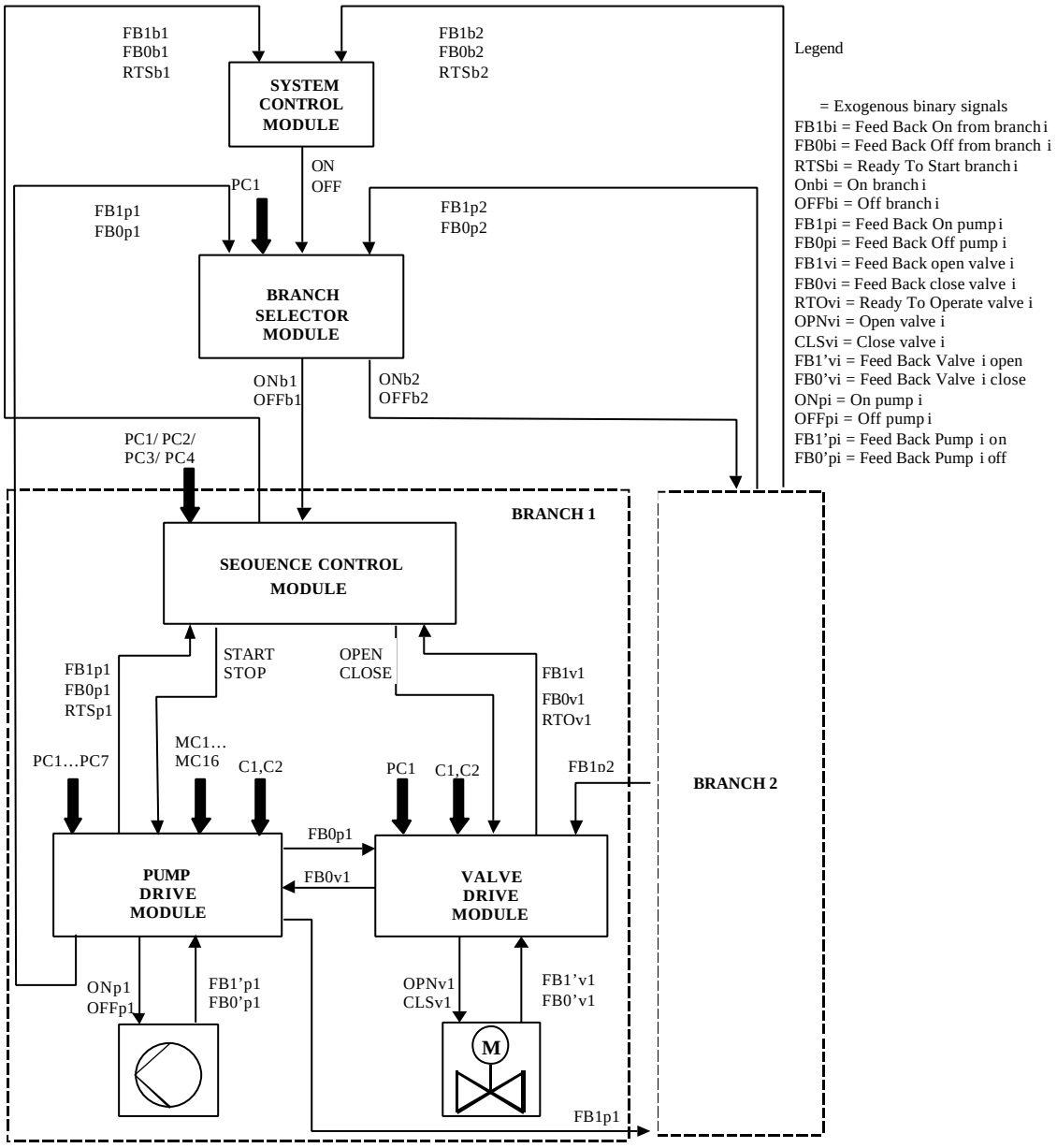


Fig. 5. Extraction system control logic structure.

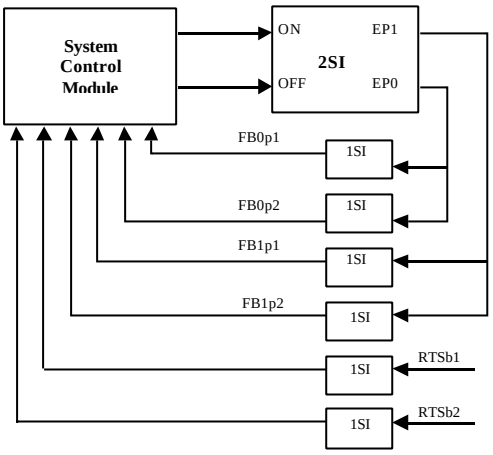


Fig. 6. Scheme for testing System Control Module.

4.2 Testing the whole CS through the bottom-up approach

Once all the CMs have been individually verified, the whole considered control system has been tested by following the presented bottom-up approach. In particular, the hierarchical level 2 (Sequence Control) has been tested through the scheme of Fig. 7.a.

In such a scheme, the Pump and Valve Drive Modules, are considered, as well as the Sequence Control Module, which in turn contains also the Sequence Control Master and two steps' sequences (one for the start-up and one for the shut-down of a branch) and some auxiliary logic functions.

Once the testing of the control functions of level 2 has been completed the testing of the control functions of level 3 (Branch Selector), has been performed according to the scheme of Fig. 7.b. In such a scheme two blocks called "brach1" and "branch2" represent the sequences, pumps and valves CMs and VPs of the two branches of the extraction system, as they appear in the testing scheme of level 2 shown in Fig. 7.a.

Finally, the highest control level, i.e. level 4 (System Control), has been tested according to the scheme reported in Fig. 7.c. In such a figure, the block on the right-hand side, representing the lower level control functions, has already been tested: actually, it is the testing scheme of Fig. 7.b. As for the branch selector and the sequence control modules, also the system control module has been individually tested, and unfortunately, the VPs used in such individual tests can not be reused, at least with a reasonable effort.

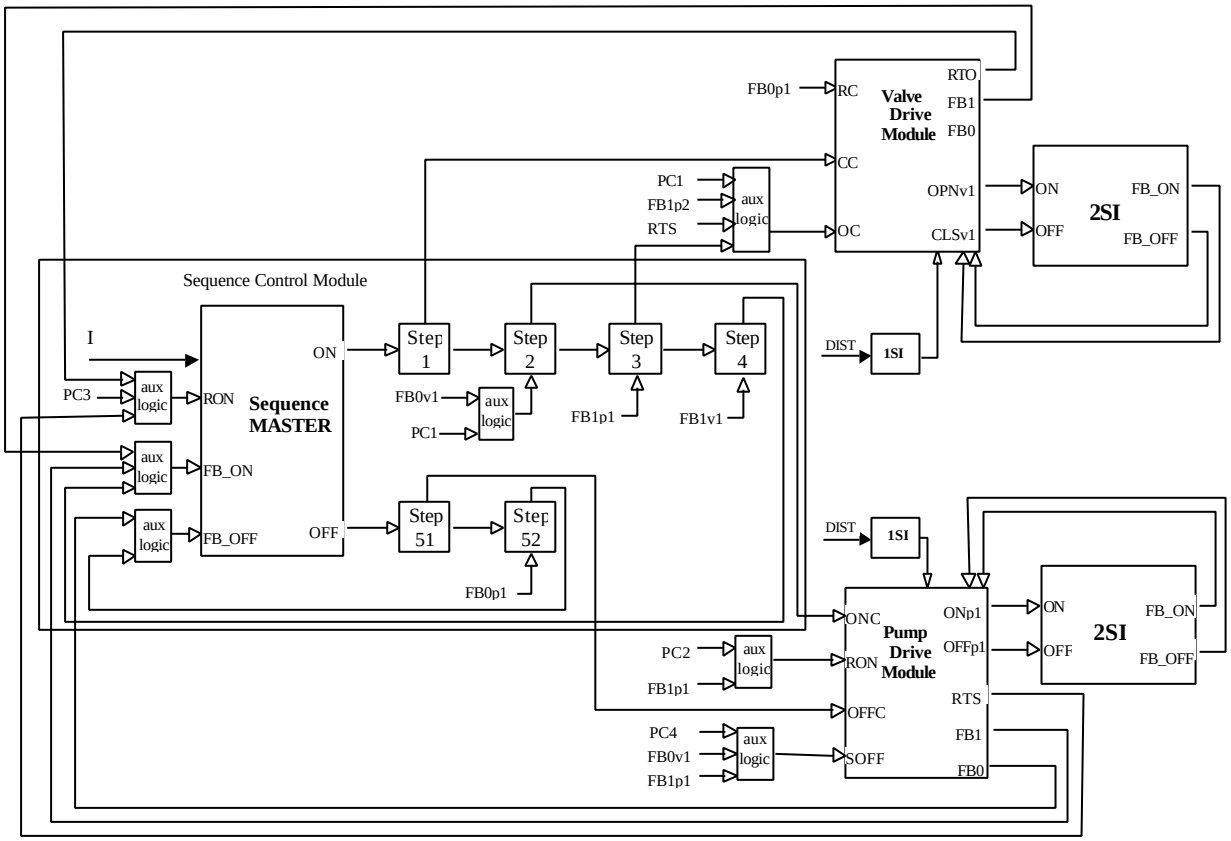
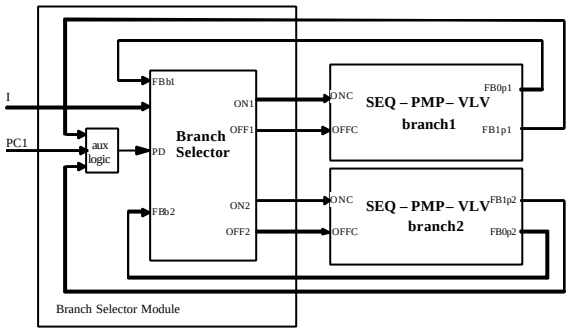


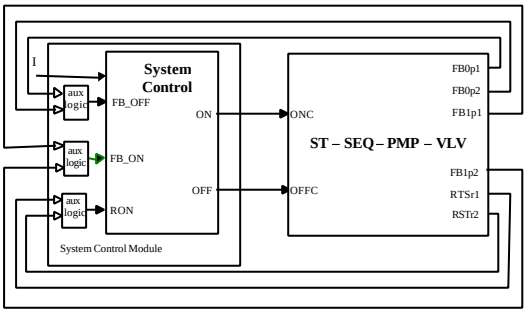
Fig. 7.a. Testing the Sequence Control and Steps – Level 2.

For the schemes of Fig. 7, the simulation results have been analyzed by comparing them with the expected ones according to the considered control system specification, which has been formally described by means of rules, and implemented by means of Stateflow charts, as discussed in (Carpanzano, Ferrarini, Maffezzoni, 2001). Moreover, the test cases for the bottom-up testing of the considered control system, i.e. the exogenous inputs I, have been generated manually by the user. The application of the above methods for the test cases (semi)automatic generation will be subject of future work.



Legend: Branch selector: PD = process disturbance, FBb1(2) = Feedback branch 1(2). Branch 1(2): ONC = ON Command, OFFC = OFF Command.

Fig. 7.b. Testing the Standby Selector – Level 3.



Legend: System Control: RON = Release ON, FB_ON(OFF) = Feedback ON(OFF). ST-…-VLV: ONC = ON Command, OFFC = OFF Command.

Fig. 7.c. Testing the Group Control – Level 4.

5. CONCLUDING REMARKS

A simulation based technique for testing logic control function of process and power plants using Simulink/Stateflow has been presented. Such a technique has been exploited to define a bottom-up testing methodology to support the engineer in the structured design and testing of industrial control systems. In particular, the application of the proposed technique to a portion of a power plant has been discussed. Subject of future work will be the study of suitable methods to formalise the control systems specification and to use such formal models to analyse the obtained simulation results. Automatic test cases generation methods and model checking will also be further investigated.

Acknowledgements

The work has been partially supported by Alstom Power Italia.

References

1. Carpanzano, E.; L., Ferrarini; C. Maffezzoni. 2001. “Modular Testing of Logic Control Functions with Matlab.” *Proceedings of the 13th European Simulation Symposium and Exhibition*, (Marseilles, France, Oct. 18-20), SCS.
2. Carpanzano, E.; L., Ferrarini; C. Maffezzoni; et al. 1999. “Testing industrial distributed control systems, with hardware-in-the-loop simulators.” *Proceedings of the 11th European Simulation Symposium and Exhibition*, (Erlangen-Nuremberg, Germany, Oct. 26-28), SCS, 574-578.
3. Daga, A.J. and W.P. Birmingham. 1995. “A symbolic-simulation approach to the timing verification of interacting FSMs”. *Proceedings of the IEEE International Conference on Computer Design*, ICCD’95, 584-589.
4. Dieterle, W.; H.D. Kochs and E. Dittmar. 1995. “Communication architectures for distributed computer control systems.” *Control Engineering Practice*, Vol. 3, 1171-1176.
5. Gravenstein, M. 1994. “Modeling techniques to support system level simulation and a top-down development methodology.” *Proceedings of the International Verilog HDL Conference*, 43-50.
6. Holzmann, G.H. 1997. “The Model Checker SPIN.” *IEEE Transactions on Software Engineering*, Vol. 23, No. 5.
7. James, J.; F. Cellier; G. Pang; J. Gray and S.E. Mattsson. 1995. “The State of Computer Aided Control System Design.” *IEEE Control Systems, Special Issue on Computer Aided Control System Design*, Vol. 15, No. 2, 6-7.
8. Maffezzoni, C.; L. Ferrarini; and E. Carpanzano. 1999. “Object-Oriented Models for Advanced Automation Engineering.” *Control Engineering Practice*, Vol. 7, No. 8, 957-968.
9. Mok, A.K. and D. Stuart. 1996. “Simulation vs. verification: getting the best of both worlds.” *Proceedings of the 11th annual Conference on Computer Assurance, COMPASS’96*, 12-22.
10. *Stateflow User's Guide*, Mathworks, 2000.
11. Taylor, J.H. and C. Chenney. 1999. “An Intelligent Implementation Aid for Industrial Process Control Systems.” *Proceedings of the American Control Conference*, San Diego, June, 3605-3609.
12. *Using MATLAB*, Mathworks, 2000.
13. *Using Simulink* , Mathworks, 2000.