12th IFAC/IEEE Workshop on Discrete Event Systems
Cachan, France. May 14-16, 2014

Control design pattern based on safety logical constraints for manufacturing systems: application to a palletizer

B. RIERA*, R. COUPAT*, A. PHILIPPOT*, D. ANNEBIQUE**, and F. GELLOT***

** CReSTIC (EA3804), UFR Sciences Exactes et Naturelles, Reims University (URCA), Moulin de la Housse, BP 1039, 51687 Reims - France (bernard.riera@univ-reims.fr).*
*** CReSTIC (EA3804), IUT de Troyes, 9 rue de Québec, BP 396, 10026 TROYES, Cedex, France*
****PSIGT-TE (CES), Direction de l'Ingénierie, Société Nationale des Chemins de Fer Français, 6, avenue François Mitterrand – 93574 La Plaine Saint Denis CEDEX, France*

Abstract: This paper presents an original approach for safe controller design for manufacturing systems controlled by PLC (Programmable Logic Controller). In this work, manufacturing systems are considered as Discrete Event Systems (DES) with logical Inputs (sensors) and logical Outputs (actuators). The proposed approach, which separates the functional control part from the safety control part, is easy to implement and ensures that the designed controller is safe. The methodology is based on the use of safety constraints in order to get a permissive safe controller which can be validated off line by model-checking. This controller is then constrained by functional constraints. The approach is illustrated with a palletizer simulated process using the ITS PLC software from the Real Games Company (www.realgames.pt). The control algorithm is presented and allows resulting in a safe control using a standard control design pattern, may be simpler than a conventional approach based on a complete specification in GRAFCET (IEC 60848) that does not distinguish the functional aspect from the safety aspect. This approach presents interesting perspectives like the management of several operating modes linked to a Manufacturing Execution System (MES) or the manual modes through Human-Machine Interfaces (HMI) or Supervisory Control and Data Acquisition (SCADA) systems.

Keywords*:* Discrete-Event Systems, Control, Safety, Programmable Logic Controllers, Manufacturing Systems.

1. INTRODUCTION

This paper presents an original approach of control design for manufacturing systems controlled by PLC (Programmable Logic Controller). In this work, manufacturing systems are considered as Discrete Event Systems (DES) [Cassandra *et al.* 1999] with logical Inputs (sensors) and logical Outputs (actuators) which can be seen respectively as uncontrollable events (logical Inputs) and controllable events (logical Outputs). This is an extension of the research work that the CReSTIC (Research Centre in Information and Communication Science and Technologies) has led for several years on the definition and design of guard conditions placed at the end of the PLC program which act as a logic filter in order to be robust to control errors. These safety constraints can be formally checked off line by using a model checker. The proposed approach, which separates the functional control part from the safety control part, enables to get a control design pattern easy to implement and ensuring that the designed controller is safe. The methodology is based on the use of safety constraints in order to get a permissive safe controller. This controller is then constrained by functional constraints. This paper proposes several improvements of the control algorithm presented in [Riera *et al.* 2012] particularly in the management of combined safety constraints. The approach is illustrated by using one example: a virtual palletizer using the ITS PLC software from the Real Games Company (www.realgames.pt). This control synthesis approach allows to result in a safe control, may be simpler than a conventional approach based on a complete specification in GRAFCET (IEC 60848) that does not distinguish the functional aspect from the safety aspect. This approach presents interesting perspectives like the management of several operating modes linked to a Manufacturing Execution System (MES) or the manual modes through Human-Machine Interfaces (HMI) or Supervisory Control and Data Acquisition (SCADA) systems.

2. BOOLEAN SAFETY CONSTRAINTS FOR ROBUST PLC CONTROL

Since a PLC is a dedicated controller, it will only process the program over and over again. One cycle through the program is called a scan time and involves reading the inputs (i.e. uncontrollable events) from the other modules, executing the logic based on these inputs and then updating the outputs (controllable events) accordingly. The memory in the CPU stores the program while also holding the status of the I/O and providing a means to store values. The notations used in this paper are:

- t: current scan time (from PLC point of view), *t-1* previous PLC scan time.
- $o_k = o_k(t)$: logical variable corresponding to the value of k^{th} PLC Boolean output (actuator) at t.

- $o_k^* = o_k(t-1)$: logical variable corresponding to the value of k^{th} PLC Boolean output (actuator) at time *t-1* (previous scan time).
- $i_j = i_j(t)$: logical variable corresponding to the value of j^{th} PLC Boolean input (sensor) at time *t*.
- $i_j^* = i_j(t-1)$ logical variable corresponding to the value of j^{th} PLC Boolean input (sensor) at time *t-1*.
- "$.$", "$+$", "$\overline{\ }$" are respectively the logical operators AND, OR, and NOT.
- 0 means FALSE and 1 means TRUE.
- $\sum$ and $\prod$ are respectively the logical sum (OR) and the logical product (AND) of logical variables.
- $\sum\prod$ is a logical polynomial (sum of products expression also called SIGMA-PI).
- $\uparrow x$ means rising edge of Boolean variable x (in the PLC, $\uparrow x = \overline{x^*}.x$).
- $\downarrow x$ means falling edge of Boolean variable x (in the PLC, $\downarrow x = x^*.\bar{x}$).
- O: set of output variables at *t*
- O^*: set of output variables at *t-1*
- I: set of input variables at *t, t-1, t-2 ...*
- OBS: set of observers at *t, t-1, t-2 ...*
- N_o : number of PLC Boolean outputs
- N_i: number of PLC Boolean inputs
- N_{CSs}: number of Simple Safety Constraints
- N_{CSc}: number of Combined Safety Constraints

The proposed methodology to design safe controllers is based on the use of logical safety constraints, which act as logical guards placed at the end of the PLC program, and forbids sending unsafe control to the plant [Marangé *et al.* 2010]. The set of safety constraints acts as a control filter.

Constraints (or guards) are always modeled with the point of view of the control part (PLC), and it is assumed that the PLC scan time is sufficient to detect any changes of the input vector (synchronous operation, possible simultaneous changes of state of PLC inputs). In addition, the plant is considered functioning normally without failure.

In this approach, safety constraints are expressed in the form of a logical monomial function (product of logical variables, $\prod$) which must always be equal to 0 (FALSE) at the end of each PLC scan time, before updating the outputs, in order to guarantee the safety. It is considered in this work that the initial safe state for all the actuators (o_k) is defined to 0. The constraints have to be defined in order to leave the system controllable. This means that, even with the set of safety constraints, it is possible to design a controller which matches the specifications. For example, considering the previous hypothesis about the safe initial state, a set of safety constraints which resets all outputs is safe but does not ensure the controllability. Some guards involve a single output at time *t* (called simple safety constraints *CSs*), other constraints involve several outputs at time *t* (combined safety constraints *CSc*). Constraints require the knowledge of I/O at the current time *t* and possibly previous times (presence of edge (*t-1*) for instance).

It may be necessary to define observers due to the lack of system observability. This is especially true when there are flows of products. Observers correspond ideally to a sequential function of PLC inputs and allow coming back to a combinatory constraint.

The set of safety constraints is considered as necessary and sufficient to guarantee the safety. In this approach, it is assumed that the safety constraints can always be represented as a monomial and depend on the inputs (at *t, t-1, t-2*…), outputs (at *t, t-1, t-2*…) and observers (depending ideally on only inputs at *t, t-1, t-2*…). In the initial methodology [Marangé *et al.* 2010], the control filter is validated offline by model checking [Behrmann *et al.* 2002] and stops the process in a safe state if a safety constraint (*CSs* and *CSc*) is violated.

As proposed, *CSs* and *CSc* are represented (equations (1) and (2)) as logical monomial functions ($\prod$, products of variables but not necessarily minterms) which have always to be FALSE at the end of each scan time to guarantee the safety. It is important to note that each *CSs* depends only on one controllable event (output: o_k) and that each *CSc* depends on several controllable events (outputs: o_k, o_l …).

$$\forall m \in [1, \mathrm{N_{CSs}}],\ \exists!\, k \in [1, \mathrm{N_O}]\ /\ CSs_m = \prod(o_k, \mathrm{I}, \mathrm{OBS}, \mathrm{O}^*) = 0 \quad (1)$$

$$\forall n \in [1, \mathrm{N_{CSc}}],\ \exists!\, (k, l, \dots) \in [1, \mathrm{N_O}]\ with\ k \neq l \neq \cdots\ /\ CSc_n = \prod(o_k, o_l, \dots, \mathrm{I}, \mathrm{OBS}, \mathrm{O}^*) = 0 \quad (2)$$

There are only 2 exclusive forms of simple safety constraints (*CSs*) because they are expressed as a monomial function, and they only involve a single output at time *t* (equation (3) or (4)):

$$\forall m \in [1, \mathrm{N_{CSs}}],\quad \exists!\, k \in [1, \mathrm{N_O}]\ /\ CSs_m = o_k.h_{0m}(\mathrm{I}, \mathrm{OBS}, \mathrm{O}^*) \quad (3)$$

xor

$$CSs_m = \overline{o_k}.h_{1m}(\mathrm{I}, \mathrm{OBS}, \mathrm{O}^*) \quad (4)$$

These simple safety constraints (*CSs*) express the fact that if $h_{0m}(\mathrm{I}, \mathrm{OBS}, \mathrm{O}^*)$ which is a monomial (product) function of only uncontrollable events at *t*, is TRUE, o_k must be necessarily FALSE (equation (3)) in order to keep the constraints equal to 0. If $h_{1m}(\mathrm{I}, \mathrm{OBS}, \mathrm{O}^*)$) is TRUE, o_k must be necessarily TRUE (equation (4)).

For each output, it is possible to write equation (5) corresponding to a logical OR of all simple safety constraints.

$$\sum_{i=1}^{N_{CSs}} CSs_i = \sum_{k=1}^{N_o} \left(f_{sk}(o_k, \mathrm{I}, \mathrm{OBS}, \mathrm{O}^*)\right) = 0 \quad (5)$$

$f_{sk}(o_k, \mathrm{I}, \mathrm{OBS}, \mathrm{O}^*)$ is a logical $\sum\prod$ function independent of the other outputs at *t* because only *CSs* are considered. $f_{sk}(o_k, \mathrm{I}, \mathrm{OBS}, \mathrm{O}^*)$ can be developed in equation (6) where f_{s0k} and f_{s1k} are polynomial functions (sum of products, $\sum\prod$) of I (inputs at *t*, *t-1*, *t-2* ...), O^* (previous outputs) and OBS (observers at *t, t-1,* …). Equation (6) has always to be FALSE because all simple safety constraints must be FALSE at each PLC scan time.

$$f_{sk}(o_k, \mathrm{I}, \mathrm{OBS}, \mathrm{O}^*) = o_k.f_{s0k}(\mathrm{I}, \mathrm{OBS}, \mathrm{O}^*) + \overline{o_k}.f_{s1k}(\mathrm{I}, \mathrm{OBS}, \mathrm{O}^*) = 0 \quad (6)$$

Taking into account all *CSs*; it is possible to write equation (7).

$$\sum_{i=1}^{N_{CSs}} CSs_i = \sum_{k=1}^{N_o} \left(o_k.f_{s0k}(\mathrm{I}, \mathrm{OBS}, \mathrm{O}^*) + \overline{o_k}.f_{s1k}(\mathrm{I}, \mathrm{OBS}, \mathrm{O}^*)\right) = 0 \quad (7)$$

It is important to note that the simple safety constraints have to respect the following mathematical property (equation 8):