MODEL OF MECHANISM BEHAVIOR FOR VERIFICATION OF PLC PROGRAMS

José M. Machado
University of Minho, Mechanical Engineering Department, Campus de Azurém, 4800-058 Guimarães, Portugal
jmachado@dem.uminho.pt

Bruno Denis
Ecole Normale Supérieure de Cachan, LURPA, 61, Avenue du Président Wilson, 94235 Cachan Cedex, France
denis@lurpa.ens-cachan.fr

Jean-Jacques Lesage
Ecole Normale Supérieure de Cachan, LURPA, 61, Avenue du Président Wilson, 94235 Cachan Cedex, France
lesage@lurpa.ens-cachan.fr

Jean-Marc Faure
Ecole Normale Supérieure de Cachan, LURPA, 61, Avenue du Président Wilson, 94235 Cachan Cedex, France
faure@lurpa.ens-cachan.fr

Jaime F. da Silva
University of Minho, Mechanical Engineering Department, Campus de Azurém, 4800-058 Guimarães, Portugal
jaimefs@dem.uminho.pt

Abstract. *More extensive work on formal methods is now available for checking PLC (Programmable Logic Controller) programs. To verify a PLC program, it is necessary to consider a set of properties to prove and one of the most interesting problems that the designers must deal is to deduce a set of properties that traduces all the safety requirements of the system behavior. In this paper, we explore the contribution of such a plant model within the context of deduction, in a systematized way, of a set of properties to prove, verifying the PLC program. Our study is primarily experimental in nature and based on a case study. A set of properties to be checked based on detailed plant model is proposed. We then analyze how a Symbolic Model-Checking tool (the NuSMV has been selected) ensures verification of these properties either with or without the considered plant model.*

Keywords: *Discrete event systems, Plant model, Formal verification, Model-checking*

1. Introduction

The work presented in this paper lies within the framework of a cooperative research program between the Mechanical Engineering Department of the University of Minho in Portugal (DEM) and the LURPA (Laboratoire Universitaire de Recherche en Production Automatisée) of ENS (École Normale Supérieure de Cachan) in France. This joint program focuses on the topic of "Dependable Control of Manufacturing Systems".

When designing and implementing the control of complex manufacturing systems, automation engineers are required to check that the behavioral models and controller programs they develop indeed fulfill all application requirements, especially those related to dependability. Formal verification methods, such as model-checking (Kowalewski, 1999; Lampérière-Couffin, 2000 and Bornot, 2000) or theorem-proving (Roussel, 2002) may be used to achieve this objective.

Nevertheless, the use of these methods in industrial models or programs requires considerable skill and can lead to combinatory explosion. In order to overcome this problem, several solutions may be envisaged: modular verification, introduction of constraints on variable states, or introduction of a plant model (model of the physical system) (Rausch, 1998).

The work presented herein is intended to highlight the advantages of whether a plant model should be taken into account in formal verification methods for PLC (Programmable Logic Controller) programs. Does the introduction of a plant model allow us to deduce a list of properties in a systematized way? Does the introduction of a plant model allow verifying additional properties? Use of a plant model increases the size of models to be verified. Does this represent an obstacle to verification? The following assumptions will be made:

- The formal verification method is model-checking.
- Controller behavior is described in accordance with the IEC 61131-3 Standard.
- The plant behavior model is built using finite state automata.

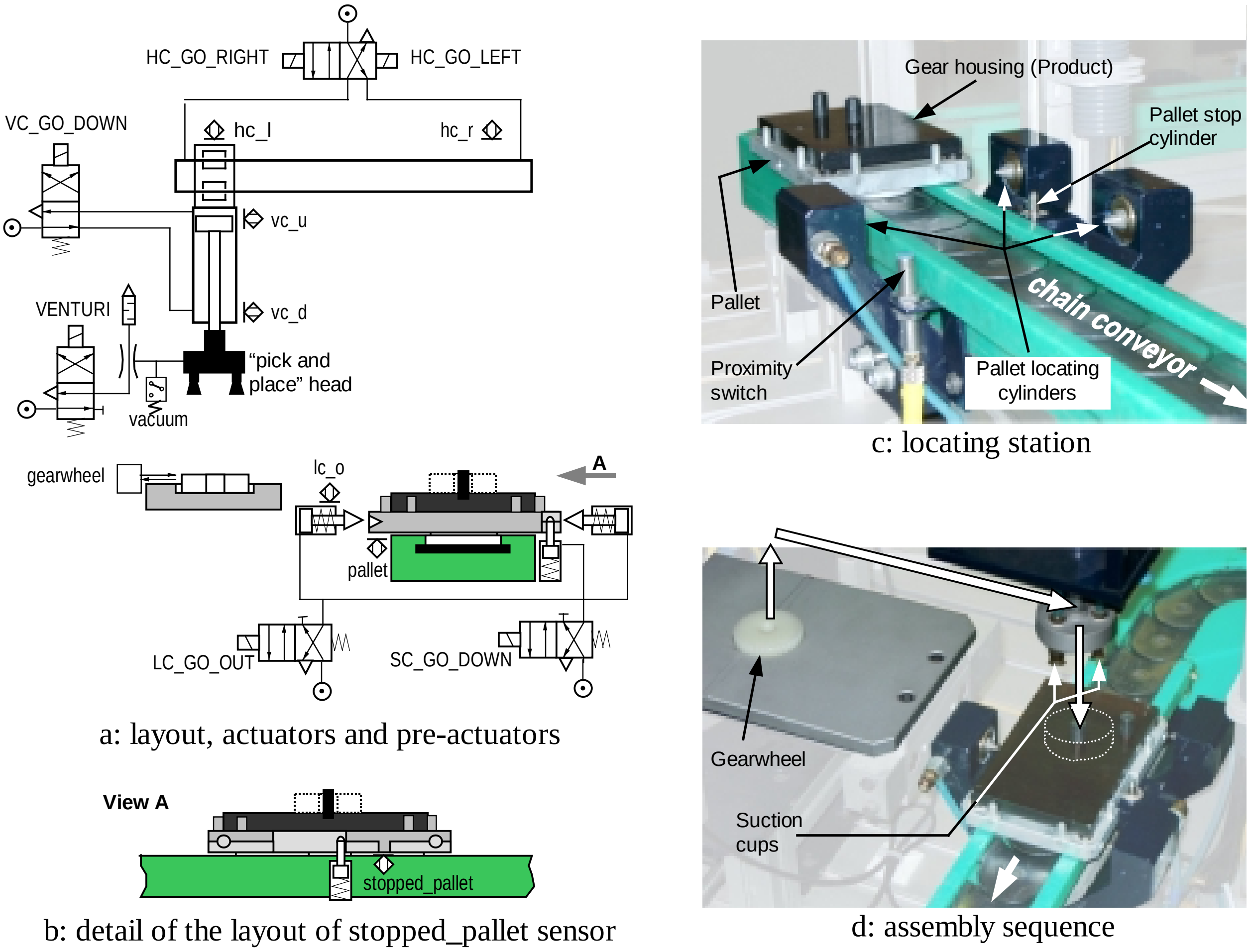
The paper has been organized as follows. In section 1, we present the challenge proposed to our work. Section 2 is devoted to the general presentation of a case study involving a "pick-and-place" workstation. We thus begin with a workstation already designed mechanically, as well as its PLC program in accordance with the IEC 61131-3 Standard. We next provide (section 3) a behavioral plant model of the uncontrolled workstation and we present a list of formal properties to be checked by the PLC program, obtained from a systematized way and based on the plant models presented before. Section 4 discusses model-checking results in order to determine, from this case study, the impact of the uncontrolled plant model within the formal verification of all properties. In section 5 are presented some conclusions and future works.

2. Case study

2.1 Aim and structure of the entire system

The case study presented is based on an assembly line that produces spur gears. A chain conveyor transfers gear housing from one workstation to the next. In this paper, we will focus on a pick-and-place workstation with the main operations consisting of: stop and locate incoming pallets, pick up gearwheels with suction cups, transfer gearwheels to gear housing using two pneumatic cylinders, and release pallets (gearwheel feeding lies beyond the scope of this study).

Figure 1 provides the various views of the workstation. It is important to note the diversity of the actuators technologies employed herein: double-acting cylinders, single-acting spring-loaded cylinders and single-acting spring-retracted cylinders. With respect to the pre-actuators, both single- and dual-solenoid valves are involved.



a: layout, actuators and pre-actuators

b: detail of the layout of stopped_pallet sensor

c: locating station

d: assembly sequence

Figure 1. Overall presentation of the workstation

2.2 Control system behavior

Control system behavior is expressed according to IEC 61131-3 and presented in Figure 2. The control system boundary is composed of a set of logical inputs and outputs, as follows:

Table 1. Studied system inputs and outputs.

Inputs:	**Outputs:**
pallet - Inductive proximity sensor indicating the presence of a pallet within the station-locating area. **stopped_pallet** - Inductive proximity sensor indicating the presence of a pallet at the exact station-locating place. **lc_o** - Magnetic sensor indicating whether or not the locating cylinder is out. **gearwheel** - Optical sensor indicating the presence of the gearwheel in place for holding prior to pallet transfer. **vacuum** - Vacuum sensor indicating whether or not the venturi is activated. **vc_d** - Magnetic sensor indicating the end of the stroke when the vertical cylinder is down (out). **vc_u** - Magnetic sensor indicating the end of the stroke when the vertical cylinder is up (in). **hc_r** - Magnetic sensor indicating the end of the stroke when the horizontal cylinder is on the pallet side. **hc_l** - Magnetic sensor indicating the end of the stroke when the horizontal cylinder is on the gearwheel side.	**SC_GO_DOWN** - Solenoid of the stop cylinder valve serving to free the pallet. **LC_GO_OUT** - Solenoid of the locating cylinder valve serving to locate the pallet. **VC_GO_DOWN** - Solenoid of the vertical cylinder valve. **HC_GO_RIGHT** - Solenoid of the horizontal cylinder valve that moves the “pick-and-place” head in the locating station direction (bistable function). **HC_GO_LEFT** - Solenoid of the horizontal cylinder valve that moves the “pick-and-place” head in the gearwheel-loading direction (bistable function). **VENTURI** - Solenoid of the vacuum system that enables gearwheel lifting.

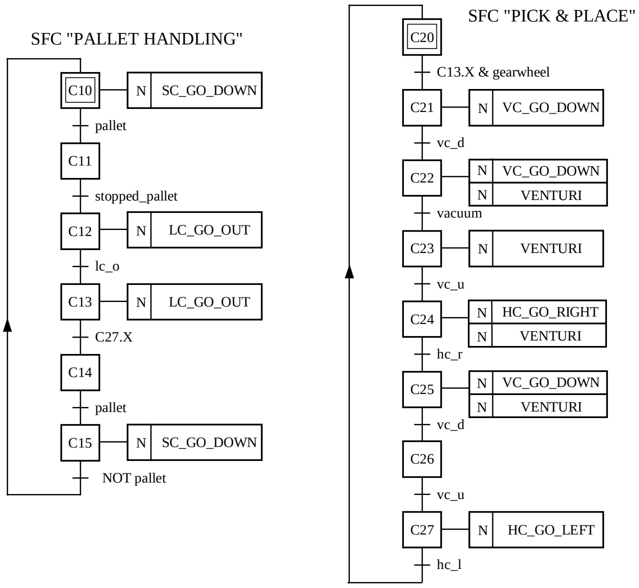


Figure 2. SFC specifications of the described APS control system

3. Uncontrolled plant models

A plant model considers, necessarily, all the physical possibilities or states for which the physical system can evolve. In these models, the set of inputs is considered as being the orders transmitted from the control system to the plant and the outputs as being the information that the sensors transmit from the plant to the control system.

A question that we will clarify in our approach is the fact that we also consider "plant models", where the informations transmitted by the sensors, to the control system, are not considered, i.e., these models are equal to the plant models, but only describe the evolution of the plant without giving information, of this evolution, to the control system. Can these models be considered as plant models? We consider that, although to describe the possibilities or possible configurations of evolution of the plant, these models are not considered as plant models because they do not give information for the inputs of the control system (the inputs of the control system do not observe the states of the plant). In this paper, we are going to assign these models as “*description models of the plant behavior*”. But it exists a question that can be placed: If these models do not supply information to the control system, why we must then consider them? The answer to this question is that these models are used in the "non model based approach" presented in section 4 of this paper.

3.1 Specific models obtained from generic models

When we analyze a plant of an automated production system is usual, to understand it, that we concentrate in the parts that constitute it (modules), to understand easily the entire system. In this system, our analysis starts for being to study the modules to allow us more easily understanding and modeling the entire system. It is pertinent to consider, in our elementary analysis the following modules, because we are able to define, correctly, the frontiers of each modular system considered:

- A cylinder with its respective command valve and its end of stroke sensors associated: the inputs are the orders transmitted to this command valve and the outputs are the information given by the sensors. Of course, that in this group of modules it has diverse variants, as, for example, double-acting cylinders, single-acting spring-loaded cylinders and single-acting spring-retracted cylinders with its respective command valves: both single- and dual-solenoid valves and still with two end of the stroke sensors, one end of the stroke sensor or without end of the stroke sensors.
- System of vacuum, with its respective command valve and the associated vacuum sensor: the input is the order transmitted to the command valve and the output is the information given by the sensor. We may have the variants with a dual- or single- solenoid valve and with or without vacuum sensor.

This kind of approach (modular) can be applied in other components, as electric motors..., that were not here considered, because they are not parts of this system. These modular models, presented here for each module, can be reused in the modeling of other automated systems plant that contains equal modules. This allows us to build a library of modules that can be reused, facilitating the modeling of the plant of other automated systems that contain similar modules.

But, how elaborate these modular plant models? For instance, in a pneumatic cylinder, we start for considering two states: ***in*** and ***out***. But, between these two states, exists another one that characterizes the movement between them, independently of the movement direction. If we think about the previous transitions to this state, we need to define a state preceded by the *in* state and GO_OUT order and other state preceded by the *out* state and GO_IN order. Therefore are considered four states (in, out, going_in and going_out), where it is adopted, for a pneumatic double acting cylinder, the plant model presented in Figure 3.

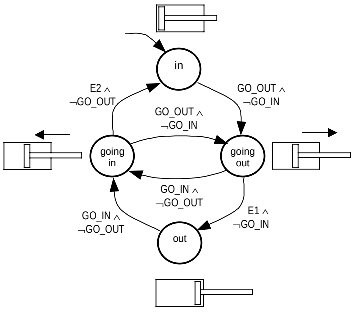


Figure 3. Plant model of a pneumatic double acting cylinder commanded by a dual-solenoid valve.

Normally, the initial state of each one of the corresponding models to this group (cylinders) must be in accordance with the initial position of the component, in automatic mode. Usually, this is the criterion, defined in the elaboration of this kind of models. In the example presented in Figure 3 as it is not directly associated with an automated system, it is considered, for defect, as initial state the state ***in***.

If a cylinder is in the ***in*** state, any that is the order transmitted to the command valve, the only possibility of evolution of this cylinder is for the ***going out*** state. If a cylinder is in the ***going out*** state, this can have two possible evolutions: or it continues in the ***going out*** state until arriving at the ***out*** state or changes to the ***going in*** state, by means of the orders transmitted by the control system to the command valve that is associated to it. In the same way, if the cylinder is in the ***out*** state, only it has the possibility to change to the ***going in*** state and if it is in the ***going in*** state it can continue in this state until reaching the ***in*** state or change to the ***going out*** state.

The logical conditions associated at the transitions between the states are composed by the orders transmitted by the control system to the cylinder command valve and by uncontrollable variables (Ei) that have been introduced to handle unspecified behavior. We can say that this model (with the mentioned states) is a generic model "cylinder" because all the cylinders models have the same states. For us, a generic model is a model that is the base to obtain some specific models of the same kind. For example, a double acting cylinder commanded by a dual-solenoid valve, is a specific model obtained from a generic model "cylinder".

Let us consider the example where a double acting cylinder is commanded by a dual-solenoid valve, where GO_OUT is the order transmitted by the control system to move the cylinder out and GO_IN is the order transmitted by the control system to move the cylinder in: the logical condition associated at the transition between the states ***in*** and ***going out*** is defined by the order GO_OUT, but we must consider that it will not be emitted, at the same time, the order GO_IN, or else the command valve will not be able to move of position and occurs a command error. So, the logical condition associated at the transition between ***in*** and ***going out*** states is defined by GO_OUT ∧ ¬GO_IN. After the cylinder be on the state ***going out*** and it maintains in this state, it is not necessary to keep order GO_OUT, because the command valve is a dual-solenoid valve. If however not exists the GO_IN order, the cylinder will finish for reaching the state ***out***. In this case, the transition between the ***going out*** and ***out*** states is given through the existence of an uncontrollable event. If, when the cylinder is in the state ***going out***, it has the order GO_IN and, at the same time, it has not the order GO_OUT, the cylinder changes to the state ***going in***. The logical condition associated at the transition between these two states is defined by GO_IN ∧ ¬GO_OUT. At the same way, were obtained the other logical conditions associated at the other transitions.

Following the previously cited principles and doing the necessary technological adaptations, relatively to the existing actuators and command valves in our case study, we have elaborated a set of specific models, obtained from generic models “cylinder” and “system of vacuum”), that are presented in Figure 4.

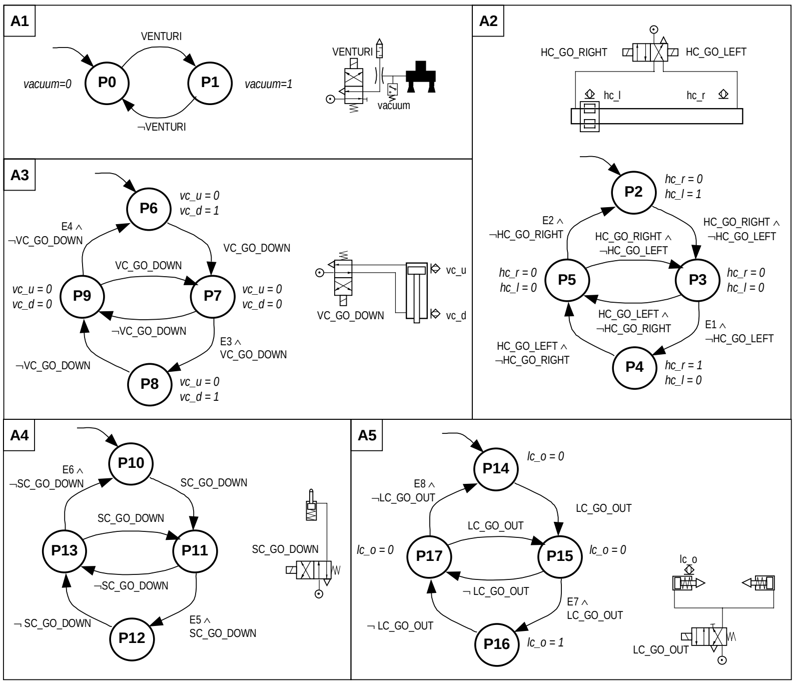


Figure 4. Specific uncontrolled plant models obtained from generic models

Where :

- (A1) suction cups with a venturi grip system driven by a single-solenoid valve (VENTURI) and observed by a vacuum sensor (vacuum);
- (A2) a double-acting horizontal cylinder driven by a dual-solenoid valve (HC_GO_LEFT, HC_GO_RIGHT) and observed by two end-stroke sensors (hc_l, hc_r);
- (A3) a double-acting vertical cylinder driven by a single solenoid valve (VC_GO_DOWN) and observed by two end-stroke sensors (vc_u, vc_d);
- (A4) a single-acting and spring-loaded stop cylinder driven by a single solenoid valve (SC_GO_DOWN) and not observed;
- (A5) a single-acting and spring-retracted locating cylinder driven by a single solenoid valve (LC_GO_OUT) and observed by an extended end-stroke sensor (lc_o).

3.2 Specific models of the desired behavior of the system mechanical components

The scope of these models elaboration is to study and to prevent mechanical collisions or undesirable functioning of the system. In contrast with the models presented in section 3.1, these models are characteristic of each system and, they are not reused, for the modeling of other systems.

In our case study, there are considered as specific models of the desired behavior of the system mechanical components: the gearwheel-loading area, the pallet-locating system and the pick and place head moving area (Figure 5).

In the elaboration of the gearwheel-loading area model (A7, Figure 5), we consider a simplification of the gearwheel-loading system, because that part of the system is not treated in our case study.

The movement of the pick and place head (A8, Figure 5) depends on the horizontal and vertical cylinders states. In this model four corresponding states to the extreme positions are considered of the pick and place head (P27, P29, P31 and P33) and intermediate states (P28, P30 and P32) of movement between the previously cited states. In the state P27 the pick and place head is in the superior position, on the left side; in P29 state the pick and place head is in the inferior position on the left side; in P31 state the pick and place head is in the superior position on the right side and in the P33 state, the pick and place head is in the inferior position on the right side. All the previously related states correspond to the desired behavior of the pick and place head. It is considered as been P27 the initial state, that corresponds to the initial state of pick and place head in the beginning of the automatic cycle. The logical conditions associated at the transitions between the model states depend on the horizontal and vertical cylinders states. Synchronization between automata is archived using state status (active/inactive) on transition labels. For example, XP1 stands for "State P1 is active". Related to the model transitions let us consider, for example, the transitions with origin in the P27 state: If P27 is active, to move the pick and place head in the vertical direction (P27 -> P28), from up to down, the vertical cylinder must begin to go out (XP7) and it must be guaranteed that the horizontal cylinder is not moving to the right side (¬XP3). If P27 is active, to move the pick and place head from the left side to the right side (P27 -> P30) the horizontal cylinder must begin moving itself from the left side to the right side (XP3) and it must be forbidden that the vertical cylinder moves from up to down (¬XP7). All the other logical conditions associated at all other transitions had been gotten following the same principles.

The pallet-locating system model (A6, Figure 5) depends on the evolution of the two plant modules: the stop cylinder module and the locating cylinder module. Consecutively, the logical conditions associated at this model transitions must consider the states of the cited cylinders models. This model elaboration has into account the details related in the elaboration of the pick and place head model.

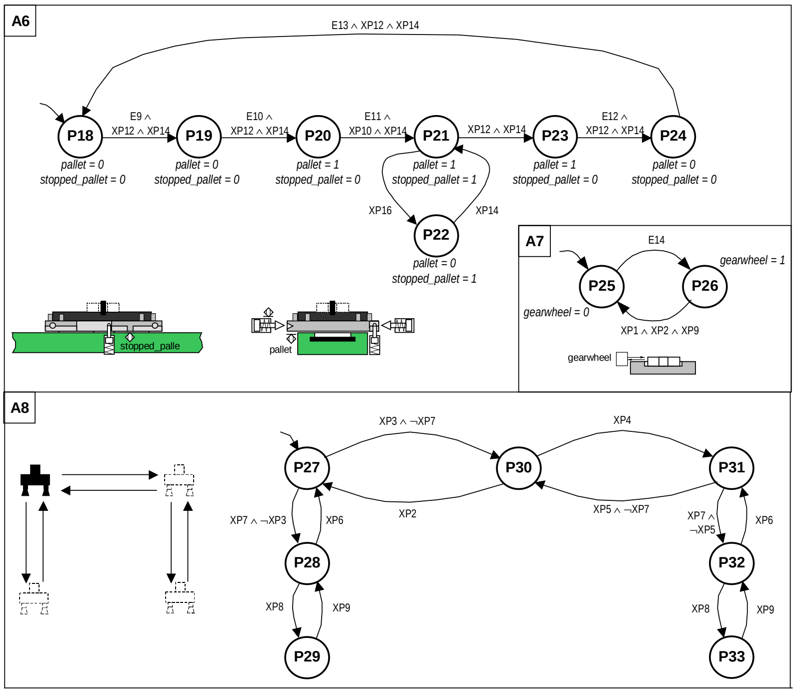


Figure 5. Specific uncontrolled plant models of the desired behavior of the system mechanical components.

Where:

- (A6) pallet-locating system - this automaton yields the behavior for both the "pallet" and "stopped-pallet" sensors;
- (A7) gearwheel-loading area - this automaton yields the behavior for the "gearwheel" sensor.

- (A8) pick and place head moving area – this automaton is obtained regarding the movement possibilities of the pick and place head, based on the vertical and horizontal cylinders states.

3.3 Deduction of properties

Considerable research has dealt with intrinsic PLC program properties, such as "*each step of our model must be reachable*" or "*there is no deadlock*". These properties have been checked in the current case study to ensure PLC program quality yet will not be developed any further in this paper. Interested readers seeking additional information are referred to Bornot et all (2000) and De Smet et all (2002).

The deduced properties, correspondent at the desired system functioning, are usually divided into *safety properties*, *liveness properties* and *fairness properties* and we shall follow this breakdown:

The *safety properties* are those witch must not occur in operation of the system. For example, the two command orders of a dual-solenoid valve must not occur at the same time.

The *liveness properties* describe what the system must do. For instance, the pick and place head must put the gearwheel in pallet only if the pallet is located.

The *fairness properties* describe how nondeterministic operations are to be resolved.

In our approach we are going to focus in ***safety properties*** and systematized way to obtain them, using the plant models.

For the deduction of properties, a careful analysis to each model is done and it is defined a set of undesirable transitions (labeled as UTi) for each one of the considered models (*i* is the number attributed to each UT). The goal is that all model evolutions (undesirable transitions) that take undesired situations of command or situations of mechanical collisions must be identified and be prevented. (see undesirable transitions in Figure 6).

To deduce a list of safety properties, we analyze all models, but we only consider the A2, A6 and A8 models. Why only these three models? The answer is simple: of all the considered models, only the described components by the models A2, A6 and A8 can take situations of undesirable command or mechanical collisions.

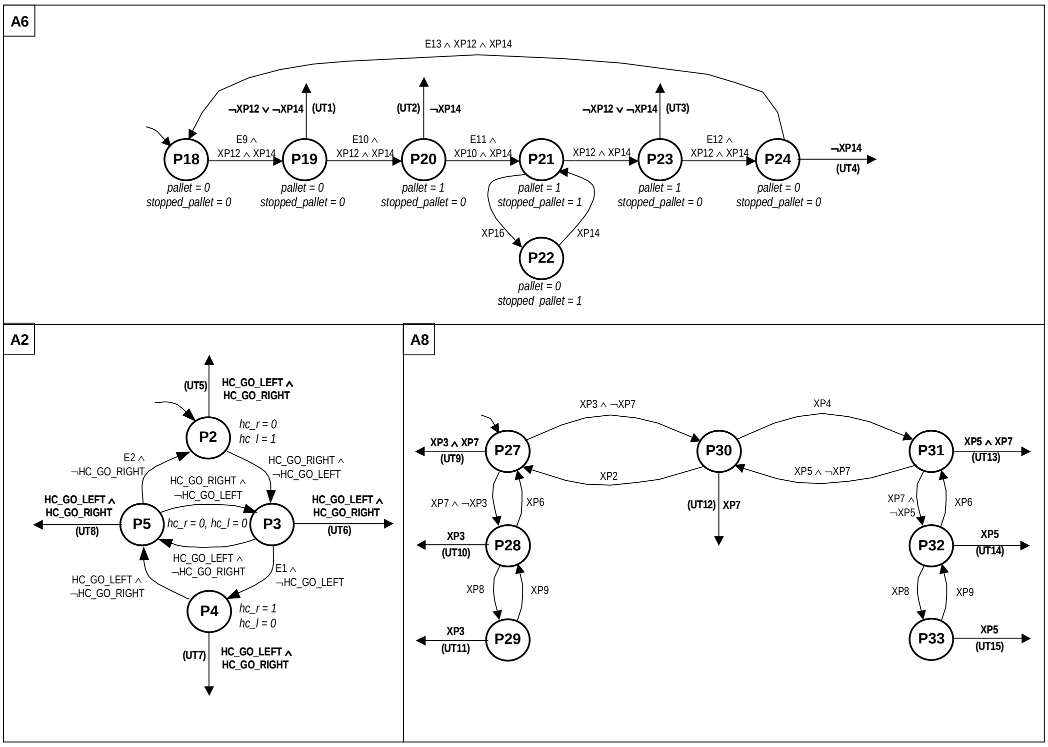


Figure 6. Properties deduction in a systematized way based on the plant models analysis.

In the A2 model, it is intended that in each one of its states it does not verify the occurrence, in simultaneous, of the two orders of horizontal cylinder command valve. For instance, the logical condition (XP2 ∧ HC_GO_LEFT ∧ HC_GO_RIGHT) associated at the transition labeled as UT5, must not occur. We can say that UT5 is an undesirable transition. The same to the transitions labeled as UT6, UT7 and UT8. In the A6 model, the transitions UT1, UT2, UT3 and UT4 are undesirable transitions because they allow evolutions that originate impediments to the pallet movement. For instance, if the P23 state is active, it must not be possible to hinder the advance of pallet originated by occurrence of the logical condition associated at the undesirable transition UT3: $(XP23 \wedge (\neg XP12 \vee \neg XP14))$. In A8 model, the transitions UT9, UT10, UT11, UT12, UT13, UT14 and UT15 are undesirable transitions because they allow evolutions that originate mechanical collisions. For instance, if the P30 state is active (displacement of pick and place head in the horizontal direction) then it is intended that it must be not possible to move the pick and place head in the vertical direction: "$XP30 \wedge XP7$" is associated at the undesirable transition (UT12).

Thus, all the deduced properties consist on identifying all logical conditions associated at all undesirable transitions in an exhaustive way, analyzing all plant models states.

Our list of safety properties, deduced from a systematized, way is (see Figure 6):

- A6 model: $\forall\ t \in \mathrm{IR}^{+*} \mid \neg(XP19 \wedge (\neg XP12 \vee \neg XP14))$ (PROP_UT1)
- A6 model: $\forall\ t \in \mathrm{IR}^{+*} \mid \neg(XP20 \wedge \neg XP14)$ (PROP_UT2)
- A6 model: $\forall\ t \in \mathrm{IR}^{+*} \mid \neg(XP23 \wedge (\neg XP12 \vee \neg XP14))$ (PROP_UT3)
- A6 model: $\forall\ t \in \mathrm{IR}^{+*} \mid \neg(XP24 \wedge \neg XP14)$ (PROP_UT4)
- A2 model: $\forall\ t \in \mathrm{IR}^{+*} \mid \neg(XP2 \wedge \mathrm{HC\_GO\_LEFT} \wedge \mathrm{HC\_GO\_RIGHT})$ (PROP_UT5)
- A2 model: $\forall\ t \in \mathrm{IR}^{+*} \mid \neg(XP3 \wedge \mathrm{HC\_GO\_LEFT} \wedge \mathrm{HC\_GO\_RIGHT})$ (PROP_UT6)
- A2 model: $\forall\ t \in \mathrm{IR}^{+*} \mid \neg(XP4 \wedge \mathrm{HC\_GO\_LEFT} \wedge \mathrm{HC\_GO\_RIGHT})$ (PROP_UT7)
- A2 model: $\forall\ t \in \mathrm{IR}^{+*} \mid \neg(XP5 \wedge \mathrm{HC\_GO\_LEFT} \wedge \mathrm{HC\_GO\_RIGHT})$ (PROP_UT8)
- A8 model: $\forall\ t \in \mathrm{IR}^{+*} \mid \neg(XP27 \wedge XP3 \wedge XP7)$ (PROP_UT9)
- A8 model: $\forall\ t \in \mathrm{IR}^{+*} \mid \neg(XP28 \wedge XP3)$ (PROP_UT10)
- A8 model: $\forall\ t \in \mathrm{IR}^{+*} \mid \neg(XP29 \wedge XP3)$ (PROP_UT11)
- A8 model: $\forall\ t \in \mathrm{IR}^{+*} \mid \neg(XP30 \wedge XP7)$ (PROP_UT12)
- A8 model: $\forall\ t \in \mathrm{IR}^{+*} \mid \neg(XP31 \wedge XP5 \wedge XP7)$ (PROP_UT13)
- A8 model: $\forall\ t \in \mathrm{IR}^{+*} \mid \neg(XP32 \wedge XP5)$ (PROP_UT14)
- A8 model: $\forall\ t \in \mathrm{IR}^{+*} \mid \neg(XP33 \wedge XP5)$ (PROP_UT15)

Machado et all (2003) presents a list of properties, related with the same example here presented, based on the expertise of the designers. In this paper we show that all safety properties of the indicated publication can be included in our list of properties deduced from a systematized way and also, a set of properties not considered in the cited publication, is indicated in this paper. That shows that our systematized way to obtain properties is more complete, because all situations that conduce into undesirable functioning (UTi) are exhaustively analyzed.

The properties presented in Machado et all (2003) are:

PROP1 : $\forall\ t \in \mathbb{R}^{+*} \mid \neg(\mathrm{HC\_GO\_RIGHT} \wedge \mathrm{HC\_GO\_LEFT})$
PROP2 : $\forall\ t \in \mathbb{R}^{+*} \mid \mathrm{VC\_GO\_DOWN} \Rightarrow \neg(\mathrm{HC\_GO\_RIGHT} \vee \mathrm{HC\_GO\_LEFT})$
PROP3 : $\forall\ t \in \mathbb{R}^{+*} \mid (C23.X \vee C26.X) \Rightarrow \neg(\mathrm{HC\_GO\_RIGHT} \vee \mathrm{HC\_GO\_LEFT})$
PROP4 : $\forall\ t \in \mathbb{R}^{+*} \mid \mathrm{vc\_d} \Rightarrow \neg(\mathrm{HC\_GO\_RIGHT} \vee \mathrm{HC\_GO\_LEFT})$
PROP5 : $\forall\ t \in \mathbb{R}^{+*} \mid \mathrm{VC\_GO\_DOWN} \Rightarrow ((\mathrm{hc\_r} \wedge \neg \mathrm{hc\_l} \wedge \neg \mathrm{HC\_GO\_LEFT}) \vee (\neg \mathrm{hc\_r} \wedge \mathrm{hc\_l} \wedge \neg \mathrm{HC\_GO\_RIGHT}))$
PROP6 : $\forall\ t \in \mathbb{R}^{+*} \mid (\mathrm{VC\_GO\_DOWN} \wedge \mathrm{hc\_r}) \Rightarrow \mathrm{lc\_o}$

Doing an analysis of our list, relatively to the cited properties, in the two approachs, we conclude the following:

- Property PROP1 is equivalent to the properties PROP_UT5, PROP_UT6, PROP_UT7 and PROP_UT8. The properties PROP_UT5, PROP_UT6, PROP_UT7 and PROP_UT8 can be converted into one only property (PROP_UT_5_8) such that: PROP_UT_5_8 : $\forall\ t \in \mathrm{IR}^{+*} \mid \neg((XP2 \vee XP3 \vee XP4 \vee XP5) \wedge \mathrm{HC\_GO\_LEFT} \wedge \mathrm{HC\_GO\_RIGHT})$:

Table 2. Comparison between properties PROP1 (Machado, 2003) and PROP_UT_5_8

PROP1	PROP_UT_5_8
$\forall\ t \in \mathrm{IR}^{+*} \mid \neg(\mathrm{HC\_GO\_LEFT} \wedge \mathrm{HC\_GO\_RIGHT})$	$\forall\ t \in \mathrm{IR}^{+*} \mid \neg((XP2 \vee XP3 \vee XP4 \vee XP5) \wedge \mathrm{HC\_GO\_LEFT} \wedge \mathrm{HC\_GO\_RIGHT})$

As the horizontal cylinder is always in one of states XP2 or XP3 or XP4 or XP5, we can conclude that the properties are equivalents. The logical value of $(XP2 \vee XP3 \vee XP4 \vee XP5)$ is "1" (always true).

- Property PROP2 is contained in the properties PROP_UT10 and PROP_UT14:

Table 3. Comparison between properties PROP2 (Machado, 2003) and properties PROP_UT10 and PROP_UT14

PROP2	PROP_UT10	PROP_UT14
$\forall\ t \in \mathbb{R}^{+*} \mid \mathrm{VC\_GO\_DOWN} \Rightarrow \neg(\mathrm{HC\_GO\_RIGHT} \vee \mathrm{HC\_GO\_LEFT})$	$\forall\ t \in \mathrm{IR}^{+*} \mid \neg(XP28 \wedge XP3)$	$\forall\ t \in \mathrm{IR}^{+*} \mid \neg(XP32 \wedge XP5)$

In our model (A8) we do not make distinction between the movements go down and go up of pick and place head, therefore it is only important to observe the pick and place head movement between the extreme positions of the vertical cylinder. If starts the order (VC_GO_DOWN), then we have pick and place head moving down because the command valve of the vertical cylinder is done by a single-solenoid valve. In these conditions, we intend that do not exist the orders HC_GO_RIGHT or HC_GO_LEFT (PROP2). But, if pick and place head is moving down we have the activity of the P28 or P32 states (A8 model). However, in the PROP2, it is considered that if we are descending the pick and place head then we must not have the occurrence of the orders ($\mathrm{HC\_GO\_RIGHT} \vee \mathrm{HC\_GO\_LEFT}$), but it was not considered (because an exhaustive approach was not done) that when the pick and place head is, for example, in the left side, if exists the order HC_GO_LEFT, the pick and place head can move down, because this order does not origins movement of the pick and place head in horizontal direction, so that it is necessary to consider, in this position, only that we do not have the occurrence of the pick and place head movement, from the left side to the right side $\neg(XP28 \wedge XP3)$ (see A8 model, Figure 6). The same is true to the HC_GO_RIGHT order, when the pick and place head is on the right side $\neg(XP32 \wedge XP5)$.

Summarizing, on property PROP_UT10, it is considered that if pick and place head is moving down or up and if it finds on the left side, then it must not move to the right side $\neg(XP28 \wedge XP3)$ and in property PROP_UT14, it is considered that if pick and place head is moving down or up and if it finds on the right side, then it must not move to the left side $\neg(XP32 \wedge XP5)$. We conclude that the property PROP2 is contained in properties PROP_UT10 and PROP_UT14 because these properties also include the possibilities where pick and place head is moving up (because the states P28 and P32 describe the evolutions of moving up and moving down of pick and place head).

- Property PROP3 is contained in the properties PROP_UT10 and PROP_UT14. The justification is the same that to PROP2.
- Property PROP4 is equivalent at properties PROP_UT11 and PROP_UT15. The justification is the same that to PROP2.
- Property PROP5 is equivalent at property PROP_UT12:

Table 4. Comparison between properties PROP5 (Machado, 2003) and PROP_UT12

PROP5	PROP_UT12
$\mathrm{VC\_GO\_DOWN} \Rightarrow ((hc\_r \wedge \neg hc\_l \ \wedge \neg \mathrm{HC\_GO\_LEFT}) \vee (\neg hc\_r \wedge \ hc\_l \wedge \neg \mathrm{HC\_GO\_RIGHT}))$	$\forall\ t \in \ \mathrm{IR}^{+*} \mid \neg(XP30 \wedge XP7)$

In property PROP5 ($\neg hc\_r \wedge hc\_l \wedge \neg \mathrm{HC\_GO\_RIGHT}$) is the corresponding position at P27 state (A8 model) of pick and place head (position up on the left side) and ($hc\_r \wedge \neg hc\_l \ \wedge \neg \mathrm{HC\_GO\_LEFT}$) is the corresponding position at P31 state of pick and place head ((position up on the right side). This is described in property PROP5 that pick and place head only must go down from these states. In property PROP_UT12, it is specified that the moving down of pick and place head in the P30 state is not desired, being allowed its descending from any the other states (P27 or P31); the same behavior that is specified in property PROP5.

- Property PROP6 is not included in our list of properties, because this property is a ***liveness property*** and not a ***safety property***.

4. Verification

The formal verification, by model-checking, may be done considering different approaches (Frey, 2000). In our approach we consider the non model based and the model based approaches, and then the results obtained in each one are compared. The approaches considered are:

Non model based: We consider as inputs for model-checking the control model, the description models of the plant behavior (presented in previous section) and the deduced properties. This approach is considered like “Non model based”, because the used models, although to be models that "describe" the evolutions of the plant, they do not inform the control system on this evolutions. The assignment "non model based" for this kind of approach is arguable, but it is necessary to consider these models, therefore it was from the states of the same ones that the properties had been deduced.

Model based: We consider as inputs for model-checking the control model, the plant models (with information for the control system about its own evolution) and the deduced properties.

It is important to point out that the plant models had been elaborated to allow the systematic deduction of properties (already demonstrated, successfully, in the previous section) and to be used as input in model-checker to the formal verification of the control program, to verify these same properties. Only with the comparison of the results obtained in the two approaches we will be able to conclude about the influence of the use of the plant models in the verification of the desired behavior of the automated system.

The used model-checker is NuSMV, version 2.1.2; this tool has been designed for an automaton as behavior input specification. The uncontrolled plant model is then easily translated into NuSMV code. For the PLC program translation, algebraic equations have been introduced (Marcé, 1993; Lampérière-Couffin, 2000). All properties have been checked in a few seconds on an Intel P4 architecture running on Windows XP operating system with 256 MB of RAM and the following NuSMV options: -reorder -dynamic. The results obtained either with or without the plant model are the following:

- ***Without plant models*** (non model based approach) the properties PROP_UT5, PROP_UT6, PROP_UT7 and PROP_UT8 are ***true*** and all other properties are ***false***. The number of reachable states is: 18141 (2^14.147) out of 8.22084e+08 (2^29.6147) and the time necessary to the verification is 1,8 seconds.
- ***With plant models*** (model based approach) ***all properties are true***. The number of reachable states is: 101 (2^6.65821) out of 8.22084e+08 (2^29.6147) and the time necessary to the verification is 0,8 seconds.

In reviewing the experimental results, our first remark is that certain properties can be true or false depending on whether the plant model is being implemented or not. Does this finding suggest that PLC programs are sometimes correct and sometimes incorrect? Obviously not, because the same program is always being targeted. The reason for this change in behavior of certain properties stems from the fact that the boundary of the system checked by NuSMV is changing. Even if our aim still remains checking the PLC program, in one case NuSMV checks behavior of the unconstrained PLC program, and in the other case, it checks the synchronized behavior of the PLC program with the uncontrolled plant model.

In non model based approach the properties PROP_UT5, PROP_UT6, PROP_UT7 and PROP_UT8 are *true* because they depend only from the orders emitted by the control system, i.e., only with the control system it is possible to guarantee that the orders sent to the command valve of the horizontal cylinder do not happen in simultaneous.

In the model based approach all the properties are true, because the control system is informed, by the sensors, about the evolutions of the plant.

The time for the verification of the properties is reduced, but we don't know if in more complex cases, this time may be strongly increased. The lower number of reachable states in the model based approach is the result that the use of plant models restricts the possibility of evolution of the state space.

5. Conclusions and perspectives

In this paper we have presented the advantages of the use of the plant models in formal verification tasks. These models allow us the elaboration of one properties list to prove (in a systematized way) and the use of them in the formal verification of the PLC program.

Moreover, it was presented, of an exhaustive way, how to elaborate the specific models obtained from generic models - that can be part of a modules library that may be reused - and how to elaborate specific models of the desired behavior of the system mechanical components. For the elaboration of this kind of models it is required, at the designer, a strong technological knowledge of the components, to correctly elaborate them.

As future work, it is necessary to define, on an exhaustive way, the deduction of the liveness and fairness properties.

6. References

Bornot, S., Huuck, R., Lakhnech, Y. and Lukoschus, B., 2000, "Verification of sequential function charts using SMV" Proceedings of *PDPTA* 2000 (special session on Formal Validation), Las Vegas, USA.

De Smet, O. and Rossi, O., 2002, "Verification of a controller for a flexible manufacturing line written in a Ladder Diagram via model-checking", Proceedings of 21th American Control Conference, CDRom paper Nº734, pp. 4147-4152, Anchorage, Alaska, USA.

Frey, G. and Litz, L., 2000, "Formal Methods in PLC programming", Proceedings of 2000 IEEE International Conference on Systems, Man & Cybernetics, pp. 2431-2436, Nashville, USA.

IEC 61131-3, 1998, "Programmable Controllers – Programming languages"

Lampérière-Couffin, S., and Lesage, J.-J., 2000, "Formal verification of the sequential part of PLC programs" Proceedings of Wodes2000 - 5th Workshop on Discrete Event Systems, pp. 247-254, Ghent, Belgium.

Kowalewski, S., Engell, S., Preußig, J. and Stursberg, O., 1999, "Verification of Logic Controllers for Continuous Plants Using Timed Condition/Event-System Models" Automatica, Vol. 35, pp. 505-518.

Machado, J.M., Denis, B., Lesage, J.-J., Faure, J.-M. and Silva, J.C.L.F., 2003, "Increasing the efficiency of PLC Program Verification using a plant model", Proceedings of the 6th edition of the International Conference on Industrial Engineering and Production Management, Porto, Portugal.

Marcé, L. and P. Le Parc, 1993, "Defining the semantics of languages for programmable Controllers with synchronous processes", Control Engineering Practice, Vol. 1, pp. 79-84.

Rausch, M. and Krogh, B. H., 1998, "Formal Verification of PLC Programs" Proceedings of American Control Conference. Philadelphia, USA.

Roussel, J-M., Denis, B., 2002, "Safety properties verification of ladder diagram programs" Journal Européen des Systèmes Automatisés, Vol. 36, pp. 905-917.