

**Partial Order Control and Optimal Control of Discrete
Event Systems modeled as Polynomial Dynamical
Systems over Galois fields**

Hervé Marchand, Michel Le Borgne

► **To cite this version:**

Hervé Marchand, Michel Le Borgne. Partial Order Control and Optimal Control of Discrete Event Systems modeled as Polynomial Dynamical Systems over Galois fields. [Research Report] RR-3273, INRIA. 1997. <inria-00073416>

HAL Id: inria-00073416

<https://hal.inria.fr/inria-00073416>

Submitted on 24 May 2006

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

*Partial Order Control and Optimal Control of Discrete
Event Systems modeled as Polynomial Dynamical
Systems over Galois fields*

H. Marchand, M. Le Borgne

N° 3273

October 6, 1997

————— THÈME 1 —————

 *rapport
de recherche*

Partial Order Control and Optimal Control of Discrete Event Systems modeled as Polynomial Dynamical Systems over Galois fields

H. Marchand, M. Le Borgne

Thème 1 — Réseaux et systèmes
Projet EP-ATR

Rapport de recherche n3273 — October 6, 1997 — 34 pages

Abstract: In this paper, we propose computational methods for the synthesis of controllers for discrete event systems modeled by polynomial dynamical systems over finite Galois field. The control objectives are specified as order relations to be checked and as minimization of a given cost function over the states through the trajectories of the system. The control objectives are then synthesized using algebraic tools such as ideals, varieties and morphisms. The applications of these methods to the safety specification of a power transformer station controller is finally presented.

Key-words: Discrete event systems, polynomial methods, synthesis, optimal control

(Résumé : tsvp)

This work is supported by Électricité de France (EdF)

Contrôle par ordre partiel et contrôle optimal de systèmes à événements discrets modélisés par des systèmes dynamiques polynomiaux

Résumé : Dans ce rapport, nous proposons des méthodes de calcul pour la synthèse d'automatismes discrets sur des systèmes à événements discrets modélisés par des systèmes dynamiques polynomiaux. Les objectifs de contrôle sont exprimés comme des relations d'ordre à vérifier ou comme la minimisation d'une fonction de coût donnée sur les différentes trajectoires d'états du système. Les objectifs de contrôle sont alors synthétisés à l'aide d'outils algébriques tels que idéaux, variétés et morphismes. Les applications de ces méthodes à la spécification sûre du contrôleur d'un poste de transformation électrique sont alors présentées.

Mots-clé : Systèmes à événements discrets, Méthodes polynomiales, Contrôle, Commande optimale

Introduction

The main purpose of real-time applications is to control reactive systems for which control and physical parts are often merged. Thus, different theories dedicated to the control of discrete event systems had emerged since the 80's [38, 20, 23, 31]. Usually, the starting point of these theories is: given a model for the system and the control objectives, a controller must be derived by various means such that the resulting behavior of the closed loop system meets the control objectives.

In the Ramadge and Wonham theory of discrete events systems [38, 39, 40], the physical model is described in terms of prefix-closed language (often a finite state automaton [41]). The control of the physical model is then performed by inhibiting some events (called controllable events) while the other events can not be prevented from occurring. However, in some situations, the problem is rather different: the physical model (or the system) emits some uncontrollable outputs, called signals, to the controller which in turns emits some other signals to control the model (these signals are controllable and are considered as inputs for the physical model)[2, 3, 29]. Even if it is possible to mix the two approaches in a single model, we feel that the two approaches are complementary in applications.

In our case, the specification of the physical model is realized using the synchronous data-flow language SIGNAL[5, 30], dedicated to the specification of real-time systems. The formal bases used in the definition of this language allow us to check properties on the system. To this purpose, the boolean part of the program (*i.e.*, the system specification) is translated into a polynomial dynamical system over $\mathbb{Z}/3\mathbb{Z}$ [1, 28] ¹ The use of $\mathbb{Z}/3\mathbb{Z}$ instead of the boolean field $\mathbb{Z}/2\mathbb{Z}$ is motivated by the fact that we want to have the notion of presence/absence of a boolean signal considering some other signals and not only the value (true/false). Furthermore, using such algebraic methods and polynomial dynamical system as a formal model, we are also able to synthesize controllers satisfying various kinds of control objectives. As SIGNAL is a data-flow language, it is natural for us to use the input/output approach (however systems defined as finite state automata, like in Ramadge and Wonham, can also be considered within this framework). The physical model is then represented by a polynomial dynamical system while the control of the system is performed by restricting the controllable input values to values suitable for the control goal. This restriction is obtained by incorporating new algebraic equations to the initial system.

In the historical development of control theory for discrete event systems, control objectives are usually expressed as *invariance*, *reachability* and *attractivity* of a given properties[18]. Control equations are then synthesized using algebraic tools such as ideals, varieties and morphisms. In this paper, we will extend the class of control objectives to two classes of optimal control objectives. In the first one, order relations between states of the system implement some kind of preferences between these states. The second approach is more classical and uses cost functions over bounded trajectories of the system.

To validate our approach, we consider in this paper the application of this control theory to the specification of the automatic control system of a power transformer station. It concerns the response to electric defects on the lines traversing it. It involves complex inter-

¹ $\mathbb{Z}/3\mathbb{Z}$ denotes the finite Galois field with 3 elements $\{-1, 0, +1\}$ with usual multiplication and addition modulo 3 ($1 + 1 = -1$, and $3 = 0$).

actions between communicating automata, interruption and preemption behaviors, timers and timeouts, reactivity to external events, ... The functionality of the controller is to handle the power interruption, the redirection of supply sources, and the re-establishment of the power following an interruption. The objective is twofold: protecting the components of the transformer itself, and minimizing the defect in the distribution of power in terms of duration and size of the interrupted sub-network.

The remainder of this paper is organized as follows: the first section is dedicated to the presentation of polynomial dynamical systems and to an overview of the algebraic framework. The second part deals with the presentation of the synthesis of controllers over polynomial dynamical systems and an overview of classical control problems. In the third and fourth parts order relation control and optimal control are presented. Finally, the last section deals with the application of these methods to the incremental construction of a power transformer station controller.

1 Polynomial dynamical systems

The first point concerns the choice of the model on which the control will be performed. We have chosen to represent the system by a polynomial dynamical system, which can be seen as an equational representation of an automaton. Such a polynomial dynamical system originates from the logical abstraction of a SIGNAL program. This abstraction is *automatically* performed by the compiler. The resulting polynomial dynamical system can then be for used as formal basis for verification and optimal controller synthesis purpose. Furthermore using a high level language such as SIGNAL allow us to easily specify the real time system on which control as to be performed.

1.1 General form of Polynomial Dynamical Systems

A polynomial dynamical system (PDS) over Galois fields[26, 27] (in our case $\mathbb{Z}/3\mathbb{Z}$ ²) may be specified as a set of polynomial equations of the form:

$$S = \begin{cases} p_1(X, Y, X') & = 0 \\ \dots & \\ p_n(X, Y, X') & = 0 \end{cases} \quad (1)$$

where X, Y, X' are vectors of variables in $\mathbb{Z}/3\mathbb{Z}$ and $\dim(X) = \dim(X') = n$.

Such a system will be denoted as:

$$P(X, Y, X') = 0 \quad (2)$$

The components of vectors X and X' represent the states of the system and are called *state variables*, whereas Y is a vector of m variables in $\mathbb{Z}/3\mathbb{Z}$, called *event variables*. Such a dynamical system is generally implicit, or equivalently, behavioral in the sense of

²the theory is the same for each finite field $\mathbb{Z}/p\mathbb{Z}$, with p prime

J.C. Willems [46], and no distinction between inputs and outputs is made. A case of particular interest is when it is possible to reexpress the previous PDS (2) as follows:

$$S = \begin{cases} X' & = P(X, Y) \\ Q(X, Y) & = 0 \\ Q_0(X) & = 0 \end{cases} \quad (3)$$

The first equation of (3), $X' = P(X, Y)$ is called the *state transition equation* of the system; it can be considered as a vector-valued function $[P_1, \dots, P_n]$ from $(\mathbb{Z}/3\mathbb{Z})^{n+m}$ to $(\mathbb{Z}/3\mathbb{Z})^n$. It is composed of all the equations on the state variables, and characterizes the dynamical aspect of the system. The second equation of (3), $Q(X, Y) = 0$, is called the *constraint equations* of the system. It is a system of equations $[Q_1, \dots, Q_l]$. It comprises all the equations characterizing the properties of the system involving the current instant only, call it the *static* part of the system (invariant for all instants t). Finally, the third equation $Q_0(X) = 0$ is called the *initialization equation* of the system. It is a system of equations $[Q_{0_1}, \dots, Q_{0_n}]$. This set of equations defines a set of allowed initial states. If the *initialization equation* is not given, all the possible states of the system are initials (*i.e.*, this is equivalent to state that Q_0 is identically zero.).

Such a PDS is said to be an explicit polynomial dynamical system. In the following, we focus our attention on explicit PDS.

Each element $(x, y) \in (\mathbb{Z}/3\mathbb{Z})^{n+m}$ such that $Q(x, y) = 0$ is said to be *admissible* and an event y is admissible in the state x if (x, y) is admissible. A *trajectory* $(x_i, y_i)_{i \in \mathbb{N}}$, initialized at (x_0, y_0) is a sequence (x_i, y_i) , such that the pair (x_i, y_i) is admissible and $x_{i+1} = P(x_i, y_i)$.

Therefore, a PDS can be seen as a finite state transition system. The initial states of this transition system are the solutions of the equation $Q_0(X) = 0$. When the system is in a state $x \in (\mathbb{Z}/3\mathbb{Z})^n$, any event $y \in (\mathbb{Z}/3\mathbb{Z})^m$ such that $Q(x, y) = 0$ can trigger a transition. In this case, the system evolves into state $x' = P(x, y)$ (noted $x \xrightarrow{y} x'$).

The composition of two polynomial dynamical systems is obtained in the following way: let us consider two PDSs S_1 and S_2 , then the composition $S_1 \parallel S_2$ (assuming that the two systems have no common state variable) is simply obtained by combining the equations together. The event variables with the same name in the two PDS are considered as identical as is usual in mathematics.

1.2 The Algebraic Framework

The theory of polynomial dynamical systems uses classical tools in algebraic geometry such as *ideals*, *varieties* and *morphisms*. Using this theory we are allowed to transform the properties of algebraic sets into equivalent properties of the associated ideals of polynomials [26, 27]. Similar work may be found in [19, 14]. Finally, an overview of ideals and varieties is described in [17].

1.2.1 Ideals and Varieties

We wish to work with polynomials in several variables, with coefficients in $\mathbb{Z}/3\mathbb{Z}$. However we need that each variable X_i satisfies the condition $X_i^3 = X_i$, expressing that we indeed consider X_i as a variable living in $\mathbb{Z}/3\mathbb{Z}$. This is formalized next.

Consider first the ring $\mathbb{Z}/3\mathbb{Z}[X, Y]$ of polynomial functions in several variables $X = (X_i)_{i=1, \dots, n}, Y = (Y_j)_{j=1, \dots, m}$. Then, introduce the quotient ring of polynomial functions $A[X, Y] = \mathbb{Z}/3\mathbb{Z}[X, Y] / \langle X^3 - X, Y^3 - Y \rangle$, where all polynomials $X_i^3 - X_i$ (resp. $Y_j^3 - Y_j$) are identified with zero, written for short $X^3 - X = 0, Y^3 - Y = 0$. Equivalently, $A[X, Y]$ can be regarded as the set of polynomial functions with coefficients in $\mathbb{Z}/3\mathbb{Z}$ for which the degree in each variable is lower than 2. We shall also need the $A[X]$ quotient ring, which is defined similarly. For example, let $P(X_1, Y_1) = X_1 + 4Y_1 + 2X_1 * Y_1 + X_1^4 Y_1 - 4Y_1^3$ be a polynomial function in the ring $\mathbb{Z}/3\mathbb{Z}[X_1, Y_1]$. Then the corresponding polynomial function in the quotient ring $A[X_1, Y_1]$ is given by $P(X_1, Y_1) = X_1 - X_1 * Y_1 + X_1^2 Y_1$.

Let E be a set of events and state variables in $(\mathbb{Z}/3\mathbb{Z})^{n+m}$. The set of polynomials such that:

$$\mathcal{I}(E) \stackrel{def}{=} \{p \in A[X, Y] / \forall (x, y) \in E, p(x, y) = 0\} \quad (4)$$

is called the *ideal* of E in $A[X, Y]$. This set represents all the polynomials, for which the set E is a solution. Conversely, to any set of polynomials G , we can associate a set in $(\mathbb{Z}/3\mathbb{Z})^{n+m}$, called the *variety* of G , defined as follows:

$$\mathcal{V}(G) \stackrel{def}{=} \{(x, y) \in (\mathbb{Z}/3\mathbb{Z})^{n+m} / \forall p \in G, p(x, y) = 0\} \quad (5)$$

This set represents all the solutions for a given set of polynomials.

The advantage of using ideals is that there is a direct correspondence between an ideal and the associated variety. In fact, we can easily prove that:

Property 1 *In the quotient ring $A[X, Y]$, for every ideal \underline{a} of $A[X, Y]$, and every variety $V \in (\mathbb{Z}/3\mathbb{Z})^{n+m}$, we have:*

$$\mathcal{V}(\mathcal{I}(V)) = V \quad (6)$$

$$\mathcal{I}(\mathcal{V}(\underline{a})) = \underline{a} \quad (7)$$

This implies that relations between varieties can be translated into relations between the associated ideals of polynomials. Hence, instead of enumerating the elements of sets and manipulating them explicitly, this approach manipulates the polynomial functions characterizing their set. Then, the relations between different sets (*e.g.* inclusion or projection) can be rephrased as operations over polynomials.

1.2.2 Principal generator of an ideal

The previous transformations result in relations between ideals which can be verified using formal calculus. The Gröbner bases [11][12] are a classical tool of effective algebra able to solve problems of this kind. Gröbner bases are general and can be used in any polynomial

ring, but the computation of the canonical generators can be very expensive even for a low number of variables [18]. For this reason, we choose to represent an ideal \underline{a} by a single generator, called the *principal generator* of \underline{a} .

Property 2 *let \underline{a} be an ideal of $A[X]$ and $\{g_1, \dots, g_k\}$ a set of generators of \underline{a} , denoted by $\underline{a} = \langle g_1, \dots, g_k \rangle$. Then the polynomial function*

$$f = g_1 \oplus g_1 \oplus \dots \oplus g_k \quad (8)$$

where the operator \oplus is defined by: $f \oplus f' \stackrel{\text{def}}{=} (f^2 + f'^2)^2$, is the principal generator of \underline{a} , meaning that $\underline{a} = \langle f \rangle$.

The link between principal generators and varieties is:

Property 3 *Let $E \in (\mathbb{Z}/3\mathbb{Z})^n$. A polynomial function f is a principal generator of the ideal $\mathcal{I}(E)$ if and only if, for any element $x \in (\mathbb{Z}/3\mathbb{Z})^n$*

$$\begin{cases} x \in E & \Rightarrow f(x) = 0 \\ x \notin E & \Rightarrow f(x) \neq 0 \end{cases}$$

By coding ideals with a single polynomial, the ideal computations are reduced to simple polynomial operations.

Property 4 *Let V_1 and V_2 be two varieties of $(\mathbb{Z}/3\mathbb{Z})^n$ and g_1 (resp. g_2) be principal generator of $\mathcal{I}(V_1)$ (resp. $\mathcal{I}(V_2)$), then*

1. $V_1 \subseteq V_2 \Leftrightarrow \mathcal{I}(V_1) \supseteq \mathcal{I}(V_2) \Leftrightarrow g_2(1 - g_1^2) = 0$
2. $g_1 \oplus g_2$ is the principal generator of $\mathcal{I}(V_1 \cap V_2) = \mathcal{I}(V_1) + \mathcal{I}(V_2)$
3. $g_1^2 g_2^2$ is the principal generator of $\mathcal{I}(V_1 \cup V_2) = \mathcal{I}(V_1) \cap \mathcal{I}(V_2)$
4. $1 - g_1^2$ is the principal generator of $\mathcal{I}((\mathbb{Z}/3\mathbb{Z})^n - V_1)$

To implement the operations on principal generators, a powerful symbolic calculus system is available. We represent polynomial functions by ternary decision diagrams (TDD) [27], a slight extension of Binary Decisions Diagrams (BDDs) [9, 7], which are very efficient in boolean algebra and other areas. Like for the BDD, we obtain in this way a canonical representation of the polynomial functions.

1.2.3 Elimination of quantifiers

We are given \underline{a} an ideal on $\mathbb{Z}/3\mathbb{Z}[X, Y]$, we introduce the ideal $\exists\text{elim}_Y(\underline{a})$, defined by

$$\exists\text{elim}_Y(\underline{a}) = \underline{a} \cap \mathbb{Z}/3\mathbb{Z}[X].$$

This operation is called the *projection* of the ideal \underline{a} on the X component. The projection can be seen as the \exists -elimination of variables. In terms of variety,

$$\mathcal{V}(\exists\text{elim}_Y(\underline{a})) = \{x/\exists y, (x, y) \in \mathcal{V}(\underline{a})\}$$

Furthermore, we need to introduce the $\forall\text{elim}_Y$ operation on ideals:

$$\mathcal{V}(\forall\text{elim}_Y(\underline{a})) \stackrel{\text{def}}{=} \{x/\forall y, (x, y) \in \mathcal{V}(\underline{a})\}$$

which can be seen as the \forall elimination of variables.

1.2.4 Operations on dynamical behaviors.

To capture the dynamical aspect of a polynomial dynamical system, we introduce the notion of morphism and comorphism. A *morphism* (often called in other communities *post-condition*) is a polynomial function P from $(\mathbb{Z}/3\mathbb{Z})^{n+m}$ to $(\mathbb{Z}/3\mathbb{Z})^n$ (the *state transition equation* $X' = P(X, Y)$ of the system, for example). With the morphism P , there is an associated *comorphism* P^* from $\mathbb{Z}/3\mathbb{Z}[X]$ to $\mathbb{Z}/3\mathbb{Z}[X, Y]$, defined by:

for a polynomial $p \in \mathbb{Z}/3\mathbb{Z}[X]$:

$$\begin{aligned} P^*(p(X)) &= P^*(p(X_1, \dots, X_n)) \\ &\stackrel{\text{def}}{=} p(P_1(X, Y), \dots, P_n(X, Y)) \end{aligned} \quad (9)$$

where P_1, \dots, P_n are the components of P . In other words, $P^*(p(X))$ is obtained by substituting every X_i in p with the corresponding $P_i(X, Y)$.

There are relations between varieties and ideals using morphisms and comorphisms that are used to perform computations on the properties of polynomial dynamical systems.

Property 5 *If E is a subset of $(\mathbb{Z}/3\mathbb{Z})^n \times (\mathbb{Z}/3\mathbb{Z})^m$ and \underline{b} an ideal of $A[X]$, then*

$$\mathcal{I}(P(E)) = P^{*-1}(\mathcal{I}(E)) \quad (10)$$

$$\mathcal{V}(\langle P^*(\underline{b}) \rangle) = P^{-1}(\mathcal{V}(\underline{b})) \quad (11)$$

This section gave an overview of the basic operators which are used by the symbolic calculus system dedicated to the verification and the automatic synthesis of controllers. In this paper, we will not explain how verification can be carried out. Using these algebraic operations, it is possible to check properties such as *liveness*, *invariance*, *reachability* and *attractivity*[28, 1]. It is also possible to express CTL [16] formulae as well as propositional μ -calculus formulae [22]. For a more complete review of the theoretical foundation of this approach, the reader may refer to [27, 25].

Let us now introduce the problem of polynomial dynamical system control.

2 Control of polynomial dynamical systems

Before speaking about control of polynomial dynamical systems, we first need to introduce a distinction between the events. From now on, we distinguish between the *uncontrollable* event variables which are emitted by the system to the controller, and the *controllable* event variables which are emitted by the controller to the system.

Considering the distinction between controllable and uncontrollable events, a polynomial dynamical system is now written as:

$$S : \begin{cases} Q(X, Y, U) &= & 0 \\ X' &= & P(X, Y, U) \\ Q_0(X) &= & 0 \end{cases} \quad (12)$$

where X represents the state variables; Y and U are respectively the set of uncontrollable and controllable event variables. Such a system is called a controllable polynomial dynamic system.

Let n , m , and p be the respective dimension of X , Y , and U . The trajectories of a controllable system are sequences (x_t, y_t, u_t) in $(\mathbb{Z}/3\mathbb{Z})^n \times (\mathbb{Z}/3\mathbb{Z})^m \times (\mathbb{Z}/3\mathbb{Z})^p$ such that $Q_0(x_0) = 0$ and, for all t ,

$$\begin{aligned} Q(x_t, y_t, u_t) &= 0 \\ x_{t+1} &= P(x_t, y_t, u_t). \end{aligned}$$

The events (y_t, u_t) include an uncontrollable component y_t and a controllable one u_t . We have no direct influence on the y_t part which depends only on the state x_t . On the other hand, we have full control over u_t and we can choose any value of u_t which is admissible, i.e. such that $Q(x_t, y_t, u_t) = 0$. The chosen value determines the next state x_{t+1} and indirectly influences the possible values for y_{t+1} .

To distinguish the two components, a vector $y \in (\mathbb{Z}/3\mathbb{Z})^m$ is called an *event* and a vector $u \in (\mathbb{Z}/3\mathbb{Z})^p$ a *control*. This leads to a new notion of admissibility: an event y is *admissible* in a state x if there exists a control u such that $Q(x, y, u) = 0$; such a control is said *compatible* with y in x .

The behavior of such a controllable PDS is the following: at each instant t , given a state x_t and an admissible y_t , we can choose some u_t which is admissible *i.e.*, such that $Q(x_t, y_t, u_t) = 0$. A polynomial dynamical system S can be controlled by first selecting a particular initial state x_0 and then by choosing suitable values for $u_1, u_2, \dots, u_n, \dots$. Different strategies can be chosen to determine the value of the controls. In our case, we will only consider static control policies (*i.e.*, the value of the control u_t is instantaneously computed from the value of x_t and y_t). Such a controller is called a *static controller*.

Formally, the static controller is a system of two equations:

$$\begin{cases} C(X, Y, U) = 0 \\ C_0(X) = 0 \end{cases} \quad (13)$$

where the equation $C_0(X) = 0$ determines initial states satisfying the control objectives and the other one describes how to choose the instantaneous controls; when the controlled system is in state x , and when an event y occurs, any value u such that $Q(x, y, u) = 0$ and $C(x, y, u) = 0$ can be chosen.

The behavior of the system S composed of the controller is then modeled by the following system:

$$S_C : \begin{cases} Q(X, Y, U) = 0 \\ X' = P(X, Y, U) \\ C(X, Y, U) = 0 \\ C_0(X) = 0 \\ Q_0(X) = 0 \end{cases} \quad (14)$$

However, not every controller (C, C_0) can constitute an acceptable controller. First, the controlled system S_C has to be initialized ; thus, the equations $Q_0(X) = 0$ and $C_0(X) = 0$ must have common solutions. Furthermore, due to the uncontrollability of the events Y , any event that the system S can produce must be admissible by the controlled system S_C . This remark leads to the definition of acceptable controller:

Definition 1 *An acceptable controller for a system S*

$$S : \begin{cases} Q(X, Y, U) & = & 0 \\ X' & = & P(X, Y, U) \\ Q_0(X) & = & 0 \end{cases}$$

is given by an initial constraint equation $C_0(X) = 0$ and a control equation $C(X, Y, U)$ such that:

1. The initial constraints $C_0(X) = 0$ and $Q_0(X) = 0$ have common solutions ;
2. for all the states x which can be reached during the evolution of the controlled system, any uncontrollable event y admissible in x for the initial system S is also admissible for the controlled system S_C .

We recognize here the notion of controllable language introduced by Ramadge and Wonham in [39].

We now illustrate the use of the present framework for solving a particular classical control synthesis problem we shall reuse in the sequel. A dynamical system is said to satisfy a *state safety* property E if for every instant n , the current state x_n belongs to some set of "good states" E . To guarantee invariance of a given set of states E , the controller is computed using the following method:

introduce the operator \tilde{pre} , defined by: for any set of states F ,

$$\tilde{pre}(F) = \{x \in (\mathbb{Z}/3\mathbb{Z})^n \mid \forall y \in (\mathbb{Z}/3\mathbb{Z})^m, \exists elim_U(Q((x, y, u)) = 0 \Rightarrow \exists u, Q(x, y, u) = 0 \text{ and } P(x, y, u) \in F\}$$

Consider now the sequence $(E_i)_{i \in \mathbb{N}}$ defined by:

$$\begin{cases} E_0 & = & E \\ E_{i+1} & = & E_i \cap \tilde{pre}(E) \end{cases}$$

The sequence is decreasing. Since all sets E_i are finite, there exists a j such that $E_{j+1} = E_j$. The set E_j is then the greatest control-invariant subset of E . In practice, we transform this computation into an equivalent one over principal generators $(g_i)_{i \in \mathbb{N}}$. and g_j is then the principal generator of E_j .

Finally, $C_0(X) = g_j$ and $C(X, Y, U) = P^*(g_j)$ is an admissible feed-back controller and the system $S_C : S + (C_0, C)$ verifies the invariance of the set of states E . Consequently, using the same methods, we are able to ensure attractivity, reachability or recurrence of a set of states. For more details about this kind of control, the reader could refer to [18, 29].

However, many properties of discrete event system cannot be stated with the help of static relations. for example if we want a signal y never to take the same value two consecutive times, it must satisfy the relation $\forall n, y_{n+1} - y_n \neq 0$. Such relations are called dynamic relations, or *locally testable* properties. These kinds of properties leads to a dynamical controller. The mean idea is to extend the order of the initial system so that the initial control objective is reduced to a static control control objective for the new system. We will come back to this kind of control in section 5.2.1.

After this brief presentation of classical control objectives, the next section will describe some new kinds of control objectives specified by order relations over the states of the systems.

3 Partial Order Control problem

In the historical development of control theory, control objectives are expressed as *invariance*, *reachability* and *attractivity* of the models. Then control equations can be synthesized [18] as presented in section 2. However expressing control objectives as partial order relations is motivated by the fact that some control objectives cannot be expressed as traditional objectives. These control objectives are more concerned with the way to reach a given logical goal, rather than with the goal to be reached. Sometimes, these control objectives can be expressed as an order (or pre-order) relation between the states of the system. Furthermore, in the previous section the obtained controllers are not deterministic, in the sense that for a given state and an admissible uncontrollable event fired, different controllable events can be chosen by the controller. Therefore, in order to obtain explicit control laws over the controllable event variables, this new kind of control objectives must encompass classical control objectives.

3.1 Order relation controller synthesis

In this section, we suggest a general method for the synthesis of a controller for a control objective modeled as a partial order relation. Let S be a PDS as the one described by (12)

$$S : \begin{cases} Q(X, Y, U) & = & 0 \\ X' & = & P(X, Y, U) \\ Q_0(X) & = & 0 \end{cases}$$

Let us suppose that the system evolves to a state x , and that y is an admissible event at x . As the system is generally not deterministic, it may have several controls u such that $Q(x, y, u) = 0$. Let u_1 and u_2 be two controls admissible for the pair (x, y) . Then the system can evolve into two different states x_1 and x_2 such that

$$\begin{cases} x_1 = P(x, y, u_1) \\ x_2 = P(x, y, u_2) \end{cases}$$

The goal of the controller is to choose between u_1 and u_2 , in such a way that the system evolves into the state which is minimal for some given order relation \succeq . Since the set of states is finite, each order relation can be translated into an equation such as: $R_{\succeq}(x, x') = 0 \Leftrightarrow x \succeq x'$.

A strict order relation between the different states is computed, defined as:

$$x \succ x' \Leftrightarrow (x \succeq x') \text{ et } \neg(x' \succeq x). \quad (15)$$

The translation of (15) into polynomial equations is then given by:

$$R_{\succ}(x, x') = R_{\succeq}(x, x') \oplus (1 - R_{\succeq}(x', x)) = 0 \quad (16)$$

A controller can then be computed using the function R_{\succ} . The possible initial states are the maximal states (for the order relation R_{\succ}) among all the solutions of the equation $Q_0(X) = 0$. Such computation is performed as follows:

Let $I = \{x / Q_0(x) = 0\}$ be the set of initial states, then the maximal states (according to the order relation R_{\succ}) are obtained by removing, from the set of states I , all states for which there exists a smaller state for the order relation R_{\succ} :

$$I_{max} = I \setminus \{x / \exists x' \in I, x' \succ x\} .$$

Let Q_0 be the generator of the associated ideal, then I_{max} is generated by

$$Q_{0max}(X) = Q_0(X) \oplus \{1 - \exists elim_{X'}(Q_0(X') \oplus R_{\succ}(X, X'))\} \quad (17)$$

To force the system to choose the best control, we now introduce the following definition:

Definition 2 *A control u_1 is said to be better compared to a control u_2 , if and only if the state $x_1 = P(x, y, u_1)$ is greater than the states $x_2 = P(x, y, u_2)$ for the order relation R_{\succ} .*

In other words, the control chooses, for a pair (x, y) , an admissible control which makes the state x evolve to the state which is maximal for the relation R_{\succ} (note that there can exist more than one maximal state). The controller of the system is then provided by the following polynomial relation:

$$\{R(X, Y, U) = 0\} \Leftrightarrow \{\forall U' \in (\mathbb{Z}/3\mathbb{Z})^p, (Q(X, Y, U') = 0 \Rightarrow P(X, Y, U) \succ P(X, Y, U'))\}$$

This yields for R the formula :

$$R(X, Y, U) = \forall elim_{U'}[(1 - Q(X, Y, U'))(R_{\succ}(P(X, Y, U), P(X, Y, U')))] ,$$

and the controlled system is :

$$S_C : \begin{cases} X' = P(X, Y, U) \\ Q(X, Y, U) = 0 \\ R(X, Y, U) = 0 \\ Q_{0max}(X_0) = 0 \end{cases} \quad (18)$$

This section explained the method used to synthesize a static controller for a given plant modeled by a PDS and a control objective expressed as an order relation over the states. The following section is devoted to the presentation of various useful order relations (or pre-order relations). The use of these examples will be illustrated on the transformer power station application.

3.2 Some useful examples of order relations

Different kinds of order relations (or pre-order relations) can be used to express properties over the states. This section presents a non exhaustive catalog of them. We also show how it is possible to use cost functions over the states or the events to express order relations.

3.2.1 Minimally restrictive constraints on uncontrollable events

Let us now assume that the system S is in a state x and receives the event y ; then the system can choose any control such that $Q(x, y, u) = 0$. Let u_1 and u_2 be two possible controls, and x_1 and x_2 the corresponding successor states. As defined in 3.1, we have:

$$\begin{cases} x_1 &= P(x, y, u_1) \\ x_2 &= P(x, y, u_2) \end{cases}$$

Then a minimally restrictive control can be synthesized by adopting the following strategy: let Ad_1 and Ad_2 be the sets of admissible y events in, respectively, x_1 and x_2 . Using the principal generator of the constraints equations over the uncontrollable event variables denoted $Q'(X, Y) = \exists elim_u(Q(X, Y, U))$, we get

$$\begin{cases} Ad_1 &= \{y \in (\mathbb{Z}/3\mathbb{Z})^m / Q'(x_1, y) = 0\} \\ Ad_2 &= \{y \in (\mathbb{Z}/3\mathbb{Z})^m / Q'(x_2, y) = 0\} \end{cases}$$

We now specify our order relation depending on the following three different cases :

1. $Ad_1 \subset Ad_2$: there is more spontaneous evolution in x_2 than in x_1 . The controller must choose the control u_2 rather than u_1 .
2. $Ad_2 \subset Ad_1$: there is more spontaneous evolution in x_1 than in x_2 . The controller must choose the control u_1 rather than u_2 .
3. u_1 and u_2 are not comparable. The controller can choose either u_1 or u_2 .

We will now translate the above strategy into an order relation between the states of the system. It is possible to define this order relation in terms of ideals and principal generators. The result is a polynomial function R_{\succeq} such that $x \succeq x'$ if and only if every admissible event in the state x' is also admissible in the states x :

$$\forall y \in (\mathbb{Z}/3\mathbb{Z})^m, Q'(x', y) = 0 \Rightarrow Q'(x, y) = 0$$

and,

$$x \succeq x' \Leftrightarrow R_{\succeq}(X, X') = 0, \quad \text{where} \quad R_{\succeq}(X, X') = \forall elim_Y((1 - Q'(X', Y))Q'(X, Y)).$$

By applying the methods described in section 3.1, we can synthesize a controller such that the controlled system respects the control strategy of minimally restrictive constraints on uncontrollable events.

3.2.2 Maximization of the number of state variables equal to 1

Let (X_1, \dots, X_k) be a subset of the set of state variables X ; d_1 and d_2 be two tuples of this subset of $(\mathbb{Z}/3\mathbb{Z})^k$, $k \leq n$, where the integer n represents the number of state variables.

$$\begin{cases} d_1 &= (x_1^1, \dots, x_k^1) \\ d_2 &= (x_1^2, \dots, x_k^2) \end{cases}$$

Definition 3 We say $d_1 \sqsupseteq d_2$ if and only if

$$\forall i \in [1..k], x_i^2 = 1 \Rightarrow x_i^1 = 1 ,$$

this is a preorder relation.

To express this preorder relation, we need to introduce the polynomial function δ from $(\mathbb{Z}/3\mathbb{Z}) \times (\mathbb{Z}/3\mathbb{Z})$ to $\mathbb{Z}/3\mathbb{Z}$ such that

$$\delta(x, y) = (x(x + 1)(1 - y))^2 . \quad (19)$$

It is straightforward to show that $\delta(x, y) = 0 \Leftrightarrow \{(x = 1) \Rightarrow (y = 1)\}$. The previous preorder relation defined by the definition (3), can then be expressed in polynomial terms:

$$d_1 \sqsupseteq d_2 \Leftrightarrow T(d_1, d_2) = 0 \text{ with } T(d_1, d_2) = \bigoplus_{i=1}^k \delta(X_i^2, X_i^1) \text{ and } \oplus \text{ defined in section 1.2.2}$$

In order to extend this preorder relation to all states of the system, let us consider two states of the system x_1 and x_2 , such that $x_1 = (x_1^1, d_1)$ and $x_2 = (x_2^1, d_2)$, where d_1 and d_2 belong to $(\mathbb{Z}/3\mathbb{Z})^k$. Thus, $x_1 \sqsupseteq x_2$ if and only if $d_1 \sqsupseteq d_2$. Finally,

$$R_{\sqsupseteq}(x_1, x_2) = 0 \Leftrightarrow x_1 \sqsupseteq x_2 \text{ with } R_{\sqsupseteq}(x_1, x_2) = T(\exists elim_{[X_{k+1} \dots X_n]}(x_1), \exists elim_{[X_{k+1} \dots X_n]}(x_2))$$

By applying the construction described in section 3.1, it is possible to synthesize a controller which chooses, in a state x , one of the best controls for the relation R_{\sqsupseteq} . In other words, such a control leads the system in a state where the number of state variables equal to 1, for a given subset of X , is maximal. An example of this partial order objective will be given in section 5.2.2.

Though it is always possible to express priorities over the states using algebraic order relations, it is sometimes more useful to express directly the priorities using numerical cost functions, we investigate this next.

3.2.3 Numerical order relations

Let $X = (x_1, \dots, X_n)$ be the state variables of the system. Then, a cost function is a map from $(\mathbb{Z}/3\mathbb{Z})^n$ to \mathbb{N} , which associates to each x of $(\mathbb{Z}/3\mathbb{Z})^n$ some integer k .

Now consider a PDS S such as the one seen in the section 3.1. We assume some cost function c is given. We then have the following definition:

Definition 4 A state x_1 is said to be c -better than a state x_2 for the PDS S (noted $x_1 \succeq_c x_2$), if and only if, $c(x_2) \geq c(x_1)$

In order to express the corresponding order relation as a polynomial relation, let us consider

$$k_{max} = \sup_{x \in (\mathbb{Z}/3\mathbb{Z})^n} (c(x))$$

The following sets of states are then computed

$$A_i = \{x \in (\mathbb{Z}/3\mathbb{Z})^n / c(x) = i\}, \forall i \in [0, ..k_{max}]$$

These sets A_i form a partition of the global set of states. Consequently, we can express the following property

$$x_1 \succeq_c x_2 \quad \Leftrightarrow \quad \exists i \in [0, .., k_{max}], x_1 \in A_i \Rightarrow x_2 \in \bigcup_{j=i}^n A_j \quad (20)$$

Let g_0, \dots, g_n be the principal generator of the ideals associated to the sets A_1, \dots, A_n . The order relation \succeq_c defined by the property (20) can then be expressed as polynomial relation:

$$x \succeq_c x' \Leftrightarrow R_{\succeq_c}(X, X') = 0 \quad \text{where} \quad R_{\succeq_c}(X, X') = \prod_{i=1}^n \{(1 - g_i(X))(\prod_{j=i}^n (g_j(X')))\}$$

As this order relation is now expressed as a polynomial relation, we are able to use the method described in section 3.1 to synthesize the corresponding controller. Some numerical order relation examples will be given in section 5.2.2.

Remark 1 *To compute efficiently such order relations, it was important to rely on the ADD developed by [10] or by [15].*

In this section, the use of cost function is only considered to minimize (resp. maximize) a cost for one step of control. In the next section, we will generalize this method over a bounded states trajectory of the system. This kind of control is named *optimal control*

4 Optimal Control

Although it is very popular in classical control theory, optimal control is a new approach for DES, which has emerged in the nineties. This approach was motivated by the fact that some control objectives could not be expressed as traditional objectives (invariance, attractivity, temporal logic). The graph theoretical formulation of the optimal control problem for a class of DESs represented by automata was first given by Sengupta and Lafortune [44, 45] and a supervisory optimal control by Lin [32], whereas Ionescu [21] presented in an optimization method for a system specified with a temporal logic. Passino and Antaklis [36, 37] have associated a cost with every state transition of a DES and examined the optimization in respect of an event cost function. They use the A* algorithm to perform the controller synthesis computation. Other work in this area can be found in [24, 8, 13, 31].

In this section, the optimal controller synthesis problem for polynomial dynamical systems is presented. It involves constructing a controller which is able to choose a sequence of inputs that will transfer the polynomial dynamical system from a set of initial states to a given set

of final states while minimizing a cost function. In this section, we assume that all the events are controllable (for notational simplicity); that means that there is no perturbation. The control strategy is defined as follows. A cost is attached to each control variable (in our approach, we recall that a control is a vector of controllable variables. It is then possible to have synchronization between different controls). To be more general, a cost is also attached to each state. Finally, the optimal control strategy is based on dynamical programming [6]. A real application example of this is given in the application section.

4.1 The optimal controller synthesis problem for polynomial dynamical systems

Since, in this section, we only consider the case where there is no perturbation, the polynomial dynamical system can be rephrased as follows:

$$\begin{cases} X' &= P(X, U) \\ Q(X, U) &= 0 \\ Q_0(X_0) &= 0 \end{cases}$$

In more usual terms, this defines the dynamical system

$$\begin{cases} x_{i+1} &= P(x_i, u_i) \\ Q(x_i, u_i) &= 0 \\ Q_0(x_0) &= 0 \end{cases} \quad (21)$$

Now we are given a set \mathcal{X}_i of initial states, and a set \mathcal{X}_f of final states. A *valid control sequence* of the system is then a sequence of controls u_0, \dots, u_{n_f} which transfers the system from one of the initial states to one of the final states:

$$\mathcal{X}_i \ni x_0 \xrightarrow{u_0} x_1 \xrightarrow{u_1} \dots \xrightarrow{u_{n_f-1}} x_{n_f-1} \xrightarrow{u_{n_f}} x_{n_f} \in \mathcal{X}_f \text{ (set of final states)}, \quad (22)$$

where n_f is the first hitting time of \mathcal{X}_f , for the considered valid control sequence.

The cost of a trajectory $s_x = (x_0, \dots, x_{n_f})$ associated to the corresponding valid control sequence $s_u = (u_0, \dots, u_{n_f-1})$ is defined as follows:

$$C(s_x, s_u) \stackrel{def}{=} \sum_{i=0}^{n_f} c''(x_i) + \sum_{i=0}^{n_f-1} c'(u_i) \quad (23)$$

Thus we wish to minimize (23) subject to (21,22). This is achieved by computing backward recursively the *value function*, following a variant of the Bellman principle [4]. The actual implementation of this is described next.

4.2 Optimal controller synthesis

This section outlines a computational method for synthesizing control equations which will force the system to evolve from an initial set of states \mathcal{X}_i to a final set of states \mathcal{X}_f with a minimal cost. Finally, the various stages of the algorithm are described in this section. Problem (23,21,22) is indeed a time invariant finite horizon problem, since the final instant n_f is a first hitting time. Hence one should expect a time invariant value function together with a time invariant controller, we calculate both next.

4.2.1 Optimal value function computation

First, we compute a value function $V_{\min}(x)$, the minimal cost, for a given admissible control sequence and initial state x , to reach \mathcal{X}_f . To this end, we consider a sequence of value functions $(V_i)_{i \in \mathbb{N}}$, initialized by:

$$\begin{cases} V_0(x) &= c''(x) & \text{for } x \in \mathcal{X}_f \\ &= \infty & \text{for } x \notin \mathcal{X}_f \end{cases} \quad (24)$$

This sequence of value functions is updated by backtracking from the final states to the initial states. This is described now in terms of our framework.

Set $\mathcal{X}_0 = \mathcal{X}_f$. In a first step, we compute the set of states from which \mathcal{X}_0 can be reached in one transition.

$$\mathcal{X}'_1 = \{x \in (\mathbb{Z}/3\mathbb{Z})^n / \exists u, Q(x, u) = 0 \Rightarrow P(x, u) \in \mathcal{X}_0\}$$

To each state of \mathcal{X}'_1 is attached the cost:

$$\forall x \in \mathcal{X}'_1, V_1(x) = \min\{V_0(x), c''(x) + \min_{u/Q(x,u)=0} \{c'(u) + V_0(P(x, u))\}\}$$

Let us now assume that we are at iteration i , then

$$\begin{cases} \mathcal{X}'_{i+1} &= \{x \in (\mathbb{Z}/3\mathbb{Z})^n / \exists u, Q(x, u) = 0 \Rightarrow P(x, u) \in \mathcal{X}_i\} \\ \mathcal{X}_{i+1} &= \mathcal{X}'_{i+1} \cup \mathcal{X}_i \\ \forall x \in \mathcal{X}'_{i+1}, V_{i+1}(x) &= \min\{V_i(x), c''(x) + \min_{(u/Q(x,u)=0)} \{c'(u) + V_i(P(x, u))\}\} \end{cases} \quad (25)$$

If a cost has already been computed (in a previous step), then this cost is compared with the new one, and the minimum is taken. This way cycles on states can be considered.

It is easy to see, that $\forall i \in \mathbb{N}, V_i(x) \geq V_{i+1}(x)$. Since V_i has values in \mathbb{N} , there exists a k such that $V_{k+1} = V_k \forall x \in (\mathbb{Z}/3\mathbb{Z})^n$ (note that, in this case, we also have $\mathcal{X}_{k+1} = \mathcal{X}_k$). This is our $V_{\min}(x)$ value function. If $V_{\min}(x) < +\infty$ and $x \in \mathcal{X}_i$, then x is a valid initial state.

This description of the algorithm provides us directly with an efficient implementation using both ADD and BDD technologies, see [33] for details.

4.2.2 Control strategy

The classical way to recover the (time invariant) feedback control from Bellman recursion (25) would consist in keeping a table providing, for each state x , the optimal controls u if any. In our case this would result in combinatorial explosion. Instead, we shall regard the value function V_{\min} , computed using (25), as specifying a preorder relation on the set of states, and then we shall reuse the techniques of section 3.

Using the preceding notations, for a given state x and associated set $\{u_1, \dots, u_k\}$ of admissible controls, the pair (x, u) is said to be *preferred* to the other pairs (x, u_i) , if and only if

$$\forall u' \in \{u_1, \dots, u_k\}, c'(u') + V_{\min}(P(x, u')) \geq c'(u) + V_{\min}(P(x, u)) ,$$

i.e., we set :

$$(x, y) \succeq (x', y') \Leftrightarrow c_{xu}(x, u) \geq c_{x'u'}(x', u'), \text{ with } c_{xu}(x, u) = V_{\min}(P(x, u)) + c'(u) \quad (26)$$

Using the same methods used in section 3.1, we are able to synthesize the control equations.

First, relation (26) is translated in terms of varieties and ideals. As was done in section 3.2.3, we introduce the sets $A_1, A_2, \dots, A_{k_{max}} \in \mathbb{Z}/3\mathbb{Z} [X, U]^3$, such that

$$\forall i \in [1..k_{max}], A_i = \{(x, u) \in (\mathbb{Z}/3\mathbb{Z})^{n+m} / c_{x,u}(x) = i\} .$$

Let g_i be the principal generator of A_i then

$$(x, u) \succeq (x', u') \quad \text{if and only if} \quad \exists i \in [0, \dots, k_{max}], (x, u) \in A_i \Rightarrow (x', u') \in \bigcup_{j=i}^n A_j ,$$

whence the following formula to represent relation (26) :

$$(x, u) \succeq (x', u') \quad \Leftrightarrow \quad R_{\succeq}(X, X', U, U') = 0$$

where $R_{\succeq}(X, X', U, U') = \prod_{i=1}^n \{(1 - g_i(X, U))(\prod_{j=i}^n (g_j(X', U')))\}$

$R_{\succeq}(x, x, u, u') = 0$ means that, in a given state x , the trajectory initialized in x with the control u will have a smaller cost than the one with the control u' . The control equations are then given by:

$$R(X, U) = 0 \Leftrightarrow \forall U' \in (\mathbb{Z}/3\mathbb{Z})^3, (Q(X, U') = 0 \Rightarrow R_{\succeq}(X, X, U, U') = 0) .$$

Hence $R(X, U) = \forall elim_{U'} [(1 - Q(X, U'))(R_{\succeq}(X, X, U, U'))]$. The controlled system is, finally :

$$S_C = \begin{cases} X' & = P(X, U) \\ Q(X, U) & = 0 \\ R(X, U) & = 0 \\ I_{max}(X_0) & = 0 \end{cases}$$

where $I_{max}(X_0)$ is the set of initial states with a minimal cost, i.e., such that:

$$V_{\min}(I_{max}(X_0)) \leq V_{\min}(Q_0(X_0))$$

4.3 Example

This section gives an example of the optimal control synthesis problem for a polynomial dynamical system represented here by the figure (1) (for convenience, we choose an automaton representation).

The circles represent the states, the arrows the labeled transitions. The set of initial states is here reduced to the two states X_1 and X_0 . The set of final states, \mathcal{X}_f , contains the

³ k_{max} is the maximum of the value function V_{\min}

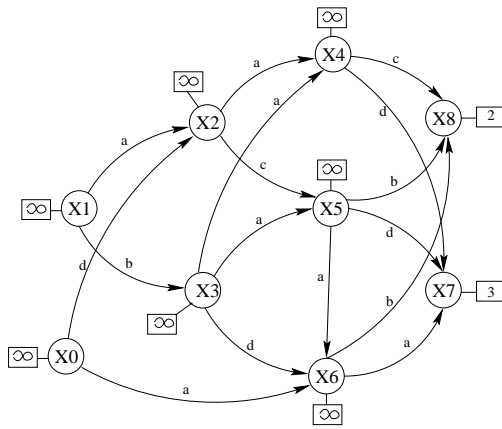


Figure 1: Example: initialization of the value function V_0

states X_7 and X_8 . Finally, the initial cost attached to the events and states are described by the following table:

events	a	b	c	d
c'	1	2	3	4

states	X_0	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8
c''	1	1	1	1	1	1	3	3	2

In figure (1), the value function V_0 , at the first step of recursion (24), is represented in the different boxes. Then all the states have an infinity cost, except for the final states which have their own costs. The figure (2) represents the automata after the computation of the value function V_{\min} : Then the cost attached to each state and put in the box is the minimal cost for a trajectory initialized in these states. A fake arrow represents the optimal control which will be chosen by the controller if the system evolves into these states.

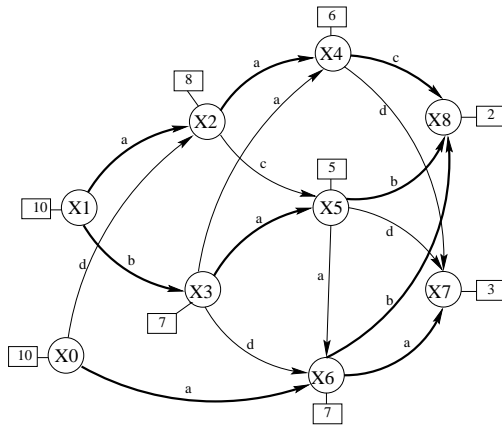


Figure 2: Example: cost attached to each state by the value function V_{\min}

Finally, as controller only allows the optimal trajectories, the controlled system is reduced to the simple automaton represented in figure (3).

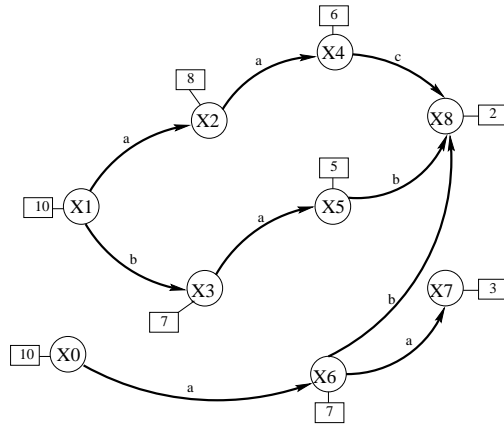


Figure 3: Example: admissible trajectories of the controlled system

4.4 Optimal control with perturbations

If the polynomial dynamical system has uncontrollable event variables, the computation of the value function V_{\min} is quite different. In fact, we cannot minimize directly the cost of uncontrollable events. We take a minmax game theoretic approach. The computation of the value function is then realized by taking, for a given state x , the maximal cost for the admissible uncontrollable event, and for this pair (x, y) , by choosing the control with the minimal cost.

As for the case with no perturbation, the computation of V_{\min} is realized by backtracking from the final states to the initial states:

$$\left\{ \begin{array}{l} \mathcal{X}'_{i+1} = \{x \in (\mathbb{Z}/3\mathbb{Z})^n / \exists(y, u), Q(x, y, u) = 0 \Rightarrow P(x, y, u) \in \mathcal{X}_i\} \\ \mathcal{X}_{i+1} = \mathcal{X}'_{i+1} \cup \mathcal{X}_i \\ \forall x \in \mathcal{X}'_{i+1}, V_{i+1}(x) = \widehat{V} \end{array} \right.$$

where

$$\widehat{V} = \min\{V_i(x), c''(x) + \max_{y/Q'(x,u)=0} \{c_y(y) + \min_{(u/Q(x,y,u)=0)} \{c'(u) + V_i(P(x, y, u))\}\}\}$$

Then control synthesis is quite similar to the one developed in the previous section.

4.5 Examples of optimal control

In this section, we briefly mention some examples which illustrate the use of optimal control.

1. First, the optimal control theory can be used to perform an excursion of minimal duration from the set of initial states. To synthesize such a controller, we attach to each event a cost equal to one. The final states are taken identical to the initial states. By computing the value function V_{\min} , we attach to the initial state the minimal number of transitions which is necessary to come back to one of the set of initial states. This kind of control could be useful to perform quickest resetting of a system when some external event is sent to the system, by a human operator for example.

2. Alternatively, let us consider a control objective expressed as a safety property, modeled by a polynomial F . Suppose now that there does not exist a solution to this control synthesis problem. If this property is a crucial safety property and must be nevertheless satisfied, using the optimal control, we are able to compute a controller (C, C_0) which makes the system evolve to the set of states where the property is satisfied. At this point, if a transition makes the system evolve to a state where the safety property is violated, the controller is activated and forces the system to come back to the “good states” at a minimal cost (e.g., minimal duration).

$$S_c = \begin{cases} Q(X, Y, U) & = 0 \\ X' & = P(X, Y, U) \\ C_0(X) & = 0 \\ C(X, Y, U) * (1 - F(X, Y, U)^2) & = 0 \end{cases}$$

It is straightforward to see that the controller is active if and only if the safety property is violated (*i.e.*, $F(X, Y, U)^2 = 1$). This kind of control objective could be called *restricted safety property*.

3. We can also compose two (or more) optimal control synthesis problems. Suppose that the system is initialized in a set of states \mathcal{X}_0 , and that the first goal to achieve is to reach the set of states \mathcal{X}_1 with a minimal cost. Suppose now that once the goal is achieved, the system must come back to the set of states \mathcal{X}_0 . In order to perform both goals, we first compute a controller C_1 which ensures the first goal for the system S . A second controller c_1 which ensures the second goal for the system S is then computed. Finally by composing the two controller C_1 et C_2 in the following way, we obtain a controlled PDS which achieves the global goal.

$$S_c = \begin{cases} X' = P(X, Y, U) & G' = f(G) \\ Q(X, Y, U) = 0 & Q'(G, X, Y, U) = 0 \\ (1 - G) * C_1(X, Y, U) = 0 & (1 + G) * C_2(X, Y, U) = 0 \\ \mathcal{X}_0(X) = 0 & f_0(G) = 0 \end{cases}$$

where G is a new state variable, which is equal to 1 when the system tries to achieve the first goal and is equal to -1 when the system tries to achieve the second goal.

5 Application to the power transformer station controller

5.1 The power transformer station description

The purpose of an electric power transformer station is to lower the voltage of power so that it can be distributed in urban centers. The kind of transformer we are interested in receives high voltage lines, and several medium voltage lines come out of it and distribute power to end-users. For each high voltage line, a transformer lowers the voltage. In the course of exploitation of this system, several kinds of electrical defects can occur (three types of electrical defects are considered: phase PH, homopolar H, or wattmetric W), due to causes

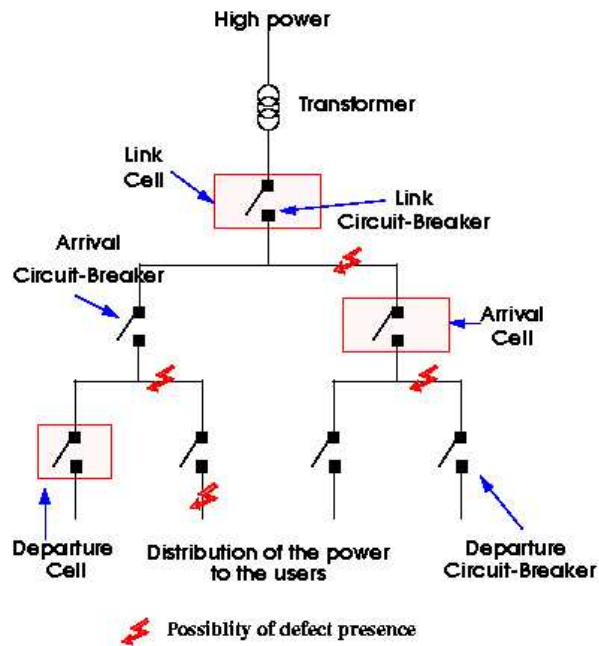


Figure 4: A possible power transformer station topology

internal or external to the station. In order to protect the device and the environment, several circuit breakers are placed in a network of *cells* in different parts of the station (on the arrival lines, link lines, and departure lines). These circuit breakers are informed about the possible presence of defects by sensors at different locations on the lines.

5.1.1 Propagation of Power and Defect

We discuss in this section some physical requirements of the power network located inside the power transformer station controller. Concerning the propagation of the power, it is obvious that the power is “visible” from the different cells if and only if all the upstream circuit-breakers are closed. As consequences, if the link circuit-breaker is opened, the power distribution is cut and then no defect can be seen by the different cells of the power transformer station.

The visibility of the defect by the sensors of the cells is less obvious. In fact, we have to consider two major properties:

On one side, if a physical defect, considered as an input of our system, is seen by the sensors of a cell, then all the downstream sensors are not able to see some physical defects. In fact, the appearance of a defect at a certain level (the departure level in the picture (5) for example) makes increase the voltage on the downstream lines and induces the masking of all the other possible defects.

On the other side, if the sensors of a cell at a given level (for example the sensors of one of the departure cells as illustrated in picture (6)) are informed about the presence of a defect, then all the upstream sensors (here the sensors of the arrival cell) detect this same

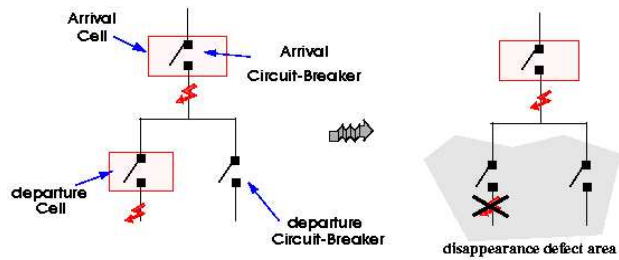


Figure 5: Masking of defects

defect. As a consequence, the arrival cell starts handling the defect while this would not be necessary.

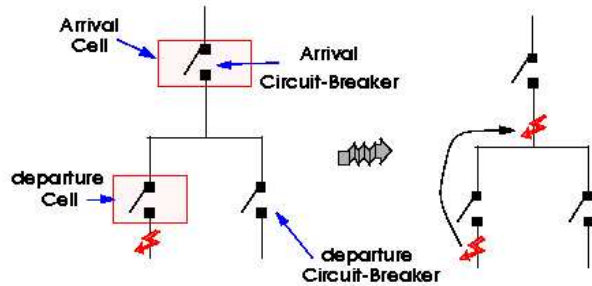


Figure 6: The propagation of the defect

5.1.2 The controller of the power transformer station

The controller of the power transformer station can be divided into two parts. The first part concerns the local controllers (*i.e.*, the cells). We chose to specify each local controller in SIGNAL, because they merge logical and numerical aspects. We only give here a brief description of the behavior of the different cells. The other part concerns more general requirements to be checked by the global controller of the power transformer station. Their specification will be further described in the following.

The local controllers: Each circuit breaker controller (or cell) defines a behavior beginning with the confirmation and identification of the type of the defect. In fact, a variety of defects are transient, *i.e.*, they occur only for a very short time. Since their duration is so short that they do not cause any danger, the manoeuvre of the circuit-breaker is inhibited. The purpose of this confirmation phase is then to take the time to let the transient defects to disappear spontaneously. If the defect is confirmed, the handling consists in opening and closing the circuit-breaker during a given delay for a certain number of cycles. The circuit-breaker is opened in consecutive cycles with an increased opening duration. At the end of each cycle, if the defect is still present, the circuit-breaker is reopened. Finally, in case the defect is still present at the end of the last cycle, the circuit-breaker is opened definitively, and control is given to the remote operator.

The specification of a large part of these local controllers has been performed using the SIGNAL synchronous language and verified using our formal calculus system, named SIGALI [28]. In this article, we are not interested in the description and validation of this part of the power transformer station controller. For a more complete review of the description and verification of the departure cell, the reader may refer to [34, 35, 28].

In the following section, we will be interested in the description of more general requirements for the controller which involve the interactions between different circuit-breakers.

Some global requirements for the controller. Even if it is quite easy to specify the local controllers in SIGNAL, some other requirements are too informal, or their behaviors are too complex to be expressed directly as programs.

One of the most significant problems concerns the appearance of two defects (the kind of defects is not important here) at two different departure cells, at the same time. Double defect is very dangerous, because it implies high defective currents. At the place of the defect, this results in a dangerous path voltage that can electrocute people and other living creatures. The detection of these two defects must be performed as fast as possible as well as the handling of one of the defects.

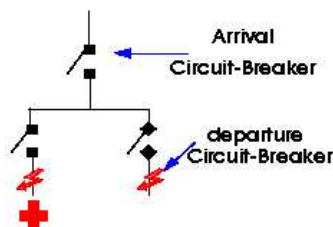


Figure 7: The double defect problem

Another important aspect is to know which of the circuit breakers must be opened. If the defect appears on the departure line, it is possible to open the circuit breaker at departure level, or at link level, or at arrival level. Obviously, it is the interest of users that the circuit be broken at departure level, and not at a higher level, so that the fewest users are deprived of power.

We also have to take into account the importance of the departure circuit-breaker. Assume that some departure line, involved in the double defect problem, supplies with electricity an hospital. Then, if the double defect problem occurs, the controller should not open this circuit-breaker, since electricity must always be delivered to an hospital.

In order to take into account these requirements, with the purpose of obtaining an optimal controller, we choose to rely on our optimal control theory. We will now describe how such controllers can be synthesized.

5.2 Incremental specification of the power transformer station controller

We have seen in the previous section, that one of the most critical requirements concerns the double defect problem. Assume that we have already a polynomial dynamical system coming from the logical abstraction of the physical model specified in SIGNAL. This one is composed by four departure circuit-breakers, two arrival circuit-breakers and one link circuit-breaker. The physical model takes also into account the power and defect propagation described in section 5.1.1. The obtained polynomial dynamical system S is represented by more than 60 states variables, 14 controllable events and 21 uncontrollable events.

5.2.1 Logical Control Objective

In this section, we assume that the circuit-breakers are ideal in the sense that they immediately react to actuators (*i.e.*, when a circuit-breaker receives an opening demand, then at the next instant the circuit-breaker is opened, as well as for a closing demand). With this assumption, the double defect problem can be rephrased as follows:

If two defects are picked up at the same time by two different departure cells, then at the next instant, one of the two defects (or both) must have disappeared.

In order to synthesize the controller, we assume that the only controllable events involve the opening and closing demands of the different circuit-breakers. The other events concern the appearance of the defects and can not be considered as controllable.

The specification of the control objective is performed as follows. After translating the physical model into a polynomial dynamical system S , we introduce a polynomial $double_defect$ which is equal to 1 when two defects are present at the same time, and -1 otherwise. We are then able to compute the set of states, where two defects are present at two consecutive instants. This is performed by computing the following polynomial: $Error_t = double_defect_t \wedge double_defect_{t-1}$. The solutions of this polynomial (*i.e.*, when $Error_t = 0$) are the set of states where two defects are present at two consecutive instants.

The problem of controller synthesis is now to ensure the invariance of the set of states no_double_defect , which is the complementary of the set of states $Error$. Applying the algorithm, described in section 2, we are able to synthesize the controller, given by the pair (C_1, C_0) , which ensure the invariance of the set of states no_double_defect for the controlled system $S_{C_1} = S + (C_1, C_0)$.

5.2.2 Partial order control objectives

However, even if the double defect problem is solved, different requirements have not been taken into account. The first one is induced by the obtained controller itself. Indeed, several solutions are available at each instant. For example, when two defects appear at a given instant, the controller can choose to open all the circuit-breakers, or at least the link circuit-breaker. This kind of solutions is not admissible and must not be considered. The second concerns the importance of the departure lines (possibility to supply with electricity an

hospital for example). The first controller (C_1, C_0) does not look at this kind of problems and can force the system to open the bad circuit-breaker.

As consequences, two new requirements must be added to the first one in order to obtain a real controller:

1. The number of opened circuit-breaker must be minimal
2. The importance of the lines (and of the circuit-breakers) has to be different.

These two requirements introduce a quantitative aspect in the control objectives which can not be solved using traditionnal techniques. We will now describe the solutions we proposed to cope with these problems.

1. *The number of opened circuit-breaker must be minimal:* In order to compute a controller which ensure the first requirement, two possible methods can be used.

First, let us assume that the state of a circuit-breaker is coded with a state variable according to the following convention: the state variable $n^{\circ}i$ is equal to 1 if and only if the corresponding circuit-breaker $n^{\circ}i$ is closed. CB is then a vector of state variables which collects all the state variables which encode the state of the circuit-breakers. Following the method described in section 3.2.2, we are able to maximize the number of state variables of CB equal to one.

Let us consider the system S_{C_1} . We then introduce an order relation over the states of the system: a state x_1 is said to be maximal compared to a state x_2 ($x_1 \supseteq x_2$ ⁴) if and only if for their corresponding sub-vectors CB_1 and CB_2 , we have $CB_1 \supseteq CB_2$. This order relation is then translated in an algebraic relation R_{\supseteq} and by applying the construction described in section 3.1, we obtain a controller (C_2, C'_0) for which the controlled system $S_{C_2} = (S_{C_1} + (C_2, C'_0))$ respects the control strategy.

However, this order relation is in fact quite strong. Let us suppose that the system evolves into a state x . Assume that the controller are able to choose between two different controls which make evolve the system into the states x_1 and x_2 with $CB_1 = (-1, 1, 1, 1, 1, 1, -1)$ and $CB_2 = (1, -1, -1, -1, -1, -1, -1)$ as sub-vectors of x_1 and x_2 . Considering the order relation \supseteq , $CB_1 \supseteq CB_2$ if and only if the state variables equal to 1 in CB_2 are equal to 1 for CB_1 . In the present case, CB_1 and CB_2 can not be compared using this order relation. This way both are allowed by the controller, and it can make evolve the system into a state where all the circuit-breakers are opened except one. In order, to solve this problem, we need to explicitly introduce the numerical aspect.

Using the same hypothesis as the above one, let \sharp be a function from $(\mathbb{Z}/3\mathbb{Z})^n$ to \mathbb{N} which associates to a state $x = (x_1, x_2, \dots, x_n)$, the number $\sum_{i=1}^n \alpha(x_i)$, with $\alpha(x_i) = 1$ when $x_i = 1$ and $x_i \in CB$, 0 otherwise.

This cost function leads to an order relation which is less restrictive than the last one, in the sense that this order relation takes into account the number of state variables

⁴where the order relation \supseteq is defined in section 3.2.2

equal to 1 and does not only compare the value of each of the state variables (see section 3.2.3 for the computation of the algebraic order relation). Using the cost function, it is then possible to compute a controller (C'_2, C''_0) , which will choose at each instant the controls which will make evolve the controlled system, $S_{C'_2} = (S_{C_1} + (C'_2, C''_0))$, into the state where the number of closed circuit-breaker is maximal.

To finish with these requirements, we now have to take into account the importance of the circuit-breakers. The last obtained controller can always choose to open the link circuit-breaker.

2. *The importance of the lines (and of the circuit-breakers) is different.*

As for the previous requirement, we will use a cost function to encode the importance of the line (in fact the both requirements will be taken into account by this cost function). We simply have to encode the fact that the more important is the circuit-breaker, the more important is the cost affected to the state variable which encodes the circuit-breaker. The picture (8) summarize the way we attache the cost.

The cost affected to each state variable in this figure corresponds to the cost when the corresponding circuit-breaker is opened. When it is closed, the cost is equal to 0. The cost of a global state is simply obtained by adding all the circuit-breaker costs. With this cost function that it is always more expensive to open a circuit-breaker at a certain level than to open all the downstream circuit-breakers.

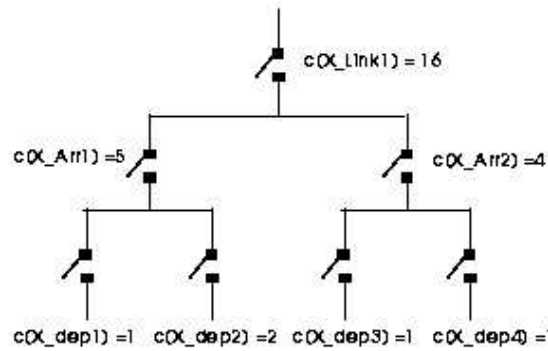


Figure 8: Cost attached to each circuit-breaker

Finally, we can remark that the cost affected to the state variable which encodes the second departure circuit-breaker (encoded by the state variable X_{dep2}) is bigger than the others because the corresponding line supplies with electricity an hospital.

Using the same method as the one described for the first requirement, we obtain a controller (C_3, C_{init}) which is sufficient to solve our problem.

5.2.3 Optimal Control objective

Thus the last controller we have obtained solves the double defect problem. However, we have assumed that all the circuit-breakers are ideal. In the reality, it is never the case. In fact, the response time of each circuit-breaker is different (we have to take into account the

quality of the circuit-breaker). We then have to introduce in our specification the fact that the response time of certain circuit-breakers of the power transformer station is longer than for some other ones. The double defect problem has now to be expressed as follows:

*If two defects are picked up at the same time by two different departure cells,
then one of the two defects must be handled as fast as possible*

In order to perform the synthesis of such a controller the idea is the following: as in section 5.2.1, we introduce the polynomial *double_defect*, which is equal to 0 when two defects are present at the same instant. Now, we also have to take into account the two other requirements (maximum of closed circuit-breakers and the importance of the lines). For that, we will reuse the cost function of the second item in the previous section. Assume that the initial set of states of the system corresponds to the set where the polynomial *double_defect* is equal to 0, and that the final set of states is the set where *double_defect* is equal to 1 or -1. Then, using the optimal control theory described in section 4, we are able to compute a controller (C_3, C_0) , which will force the controlled system to choose the minimal trajectory which makes the system evolve from a state where two defects are present, into a state where less than two defects are present. Assume we have computed this controller, then the final controlled system is given by the following set of equations:

$$S_c = \begin{cases} Q(X, Y, U) & = 0 \\ X' & = P(X, Y, U) \\ C_0(X) & = 0 \\ C_3(X, Y, U) * (1 - double_defect(X, Y, U)^2) & = 0 \end{cases}$$

The controller (C_3, C_0) is only active when two defects are present (*i.e.* $double_defect(X, Y, U)^2 = 0$). Then, when this polynomial becomes null, the controller is activated and forces the system to come back to the “good states” (*i.e.* where $double_defect(X, Y, U)^2 = 1$) at a minimal cost. This kind of control objective is similar to the *restricted safety property* described in section 4.5.

6 conclusion

In this report, we have shown the usefulness of control theory concepts for the class of polynomial dynamical systems over a finite Galois field. As this model results from the translation of a SIGNAL program (not presented here), we have a powerful environment to describe the model for real-time data-flow system. Even if classical control can be performed, we showed that using the same algebraic framework, order relations and optimal control synthesis problem can also be performed.

The order relation controller synthesis problem covers different areas of control. It can first be used to synthesize control objectives which relate more to the way to get to a logical goal, than to the goal to be reached, but can also be used to obtain explicit control laws for the controllable events (by using a strict order relation for example), considering a previous classical control synthesis problem. Concerning the optimal control, its application covers a large area including the IA plant synthesis problem. These methods have finally

been successfully applied to the incremental construction of a power transformer station controller.

The theory of polynomial dynamical systems over finite Galois field deserves much more research. One issue is the control under partial observations as carried out in [32] or in a slightly different domain the control of implicit non-deterministic polynomial dynamical systems. Some other perspectives concern the synthesis of fault tolerance controllers like in [42, 43] or the synthesis of controllers with control objectives expressed as properties that depend on the behavior of numerical variables.

Acknowledgment

This research is supported in part by EdF (Electricité de France) undercontract number M 64/7C8321/E5/11. The authors would like to thank Albert Benveniste of IRISA, France, for his careful and insightful comments.

References

- [1] T. Amagbegnon, P. Le Guernic, H. Marchand, and E. Rutten. SIGNAL – the specification of a generic, verified production cell controller. In C. Lewerentz and T. Lindner, editors, *Formal Development of Reactive Systems – Case Study Production Cell*, number 891 in Lecture Notes in Computer Science, chapter VII, pages 115–129. Springer Verlag, janvier 1995.
- [2] S. Balemi. *Control of Discrete Event Systems: Theory and Applications*. PhD thesis, Swiss Federal Institute of Technology Zurich, 1992.
- [3] S. Balemi, G. J. Hoffmann, H. Wong-Toi, and G. F. Franklin. Supervisory control of a rapid thermal multiprocessor. *IEEE Transactions on Automatic Control*, 38(7):1040–1059, July 1993.
- [4] R. Bellman. *Dynamic programming*. Princeton University, Press, NJ, 1957.
- [5] A. Benveniste and P. Le Guernic. Hybrid dynamical systems and the SIGNAL programming language. *IEEE Trans. Automat. Control*, 35:535–546, May 1990.
- [6] P. Bernhard. *Commande optimale, décentralisation et jeux dynamiques*. Dunod, 1976.
- [7] K. S. Brace, R. L. Rudell, and R. E. Bryant. Efficient implementation of a bdd package. *27th ACM/IEEE design automation conference*, pages 40–45, 1990.
- [8] Y. Brave and Heymann M. On optimal attraction in discrete-event processes. *Information Sciences*, 67:245–276, 1993.
- [9] R.E. Bryant. Graph-based algorithms for boolean function manipulations. *IEEE Transaction on Computers*, C-45(8):677–691, August 1986.
- [10] R.E. Bryant and Chen Y. Verification of Arithmetic Functions with Binary Diagrams. Technical report, School of Computer Science CMU, May 1995.
- [11] B. Buchberger. A theoretical basis for the reduction of polynomials to canonical form. In *ACM SIGSAM bull.*, volume 10(3), pages 19–29, 1976.
- [12] B. Buchberger. Gröbner Bases: An Algorithmic Method in Polynomial Ideal Theory. In N. K. Bose, editor, *Multidimensional Systems Theory*, pages 184–232, Dordrecht, 1985. K. Reidel Publishing Company.
- [13] C. Cassandras, S. Lafortune, and G. Olsder. Introduction to the modelling, control and optimization of discrete event systems. In A. Isidori, editor, *Trends in Control. A European Perspective*, pages 217–291. Springer–Verlag, September 1995.
- [14] J. Chazarain, L. A. Alonso, and E. Briales. Multivalued Logic and Gröbner bases with Applications to Modal Logic. In *Journal of Symbolic Computation*, volume 11, 181-194 1991.

- [15] E. Clarke, Fujita M., and Zhao X. Applications of multi-Terminal Binary Decisions Diagrams. Technical Report CMU-CS-95-160, School of Computer Science CMU, Pittsburgh, April 1995.
- [16] E. M. Clarke, E. A Emerson, and A. P. Sistla. Automatic verification of finite-state concurrent sytems using temporal logic specifications. *ACM Transactions on Programming Languages and Systems*, 8(2):244–263, April 1986.
- [17] D. Cox, J. Little, and D. O’Shea. *Ideals, varieties and algorithms : an introduction to computational algebraic geometry and commutative algebra*. Springer-Verlag, 1993.
- [18] B. Dutertre and M. Le Borgne. Control of polynomial dynamic systems: an example. Technical Report 2193, INRIA, January 1994.
- [19] R. Germundsson. Basic results on ideals and varieties in finite fields. Technical Report LiTH-ISY-I-1259, Linköping University, September 1991.
- [20] A Ichikawa and K. Iraishi. Analysis and control of discrete-events sytems represented by petri nets. In *Discrete Event Sytems: Models and Applications, IIASA conf.*, Sopron, Hungary, August 1987.
- [21] D. Ionescu. Conditions for optimization of discrete event systems using temporal logic methods. In G. Cohen and J. P. Quadrat, editors, *11th Int. Conf. on Analysis and Optimization of Systems, Discrete Event Systems*, volume 199 of *LNCS*, pages 93–100, Berlin, Germany, June 1994. Springer-Verlag. (INRIA Conf., Sophia-Antipolis, France).
- [22] D. Kozen. Results on the propositional μ -calculus. *Theoretical Computer Science*, 27(3):333–354, 1983.
- [23] B. H. Krogh. Supervisory control of petri nets. In *Belgian-French-Netherlands’ Summer School on Discrete Event Systems*, June 1993.
- [24] R. Kumar and V. K. Garg. Optimal control of discrete event dynamical systems using network flow techniques. In *Proc. of 29th Allerton Conf. on Communication, Control and Computing*, Champaign, IL, USA, October 1991.
- [25] M. Le Borgne. *Systèmes dynamiques sur des corps finis*. Thèse, Université de Rennes I, September 1993.
- [26] M. Le Borgne, A. Benveniste, and P. Le Guernic. Polynomial ideal theoric methods in discrete event and hybrid dynamical systems. In *Proc. of 28th IEEE Conf. on Decision and Control*, pages 2665–2700, Tampa, FL, December 1989.
- [27] M. Le Borgne, A. Benveniste, and P. Le Guernic. Polynomial dynamical systems over finite fields. In *Algebraic Computing in control*, volume 165, pages 212–222. Lecture Notes in Control and Information Sciences, G. Jacob et F. Lamnabhi-lagarigue, March 1991.

- [28] M. Le Borgne, H. Marchand, E. Rutten, and M. Samaan. Formal verification of SIGNAL programs: application to a power transformer station controller. In *Proc. of the 5th Int. Conf. on Algebraic Methodology and Software Technology, AMAST'96*, LNCS No 1101, pages 270–285, Munich, Allemagne, juillet 1996.
- [29] Michel Le Borgne, Albert Benveniste, and Paul Le Guernic. Dynamical systems over Galois fields and DEDES control problems. In *Proc. of the 30th IEEE conference on Decision and Control*, pages 1505–1510. IEEE Control System Society, 1992.
- [30] P. Le Guernic and T. Gautier. Data-flow to von Neumann: the SIGNAL approach. In Jean-Luc Gaudiot and Lubomir Bic, editors, *Advanced Topics in Data-Flow Computing*, chapter 15, pages 413–438. Prentice-Hall, Englewood Cliffs, New Jersey, 1991.
- [31] Y. Li. *Control of vector discrete-events systems*. PhD thesis, university of Toronto, July 1991.
- [32] Y. Lin and W. M. Wonham. On observability of discrete-event systems. *Information Sciences*, 44(3):173–198, 1988.
- [33] H. Marchand. *Méthodes de synthèse d'automatismes décrits par des systèmes à événements discrets finis*. PhD thesis, Université de Rennes I, October 1997. (to appear).
- [34] H. Marchand, E. Rutten, and M. Samaan. Specifying and verifying a transformer station in SIGNAL and SIGNALGTi. Technical Report 2521, INRIA, March 1995.
- [35] H. Marchand, E. Rutten, and M. Samaan. Synchronous design of a transformer station controller with SIGNAL. In *4th IEEE Conference on Control Applications*, pages 754–759, Albany, New-york, September 1995.
- [36] K. M. Passino and P. J. Antsaklis. On the optimal control of discrete event systems. In *Proc. of 28th Conf. Decision and Control*, pages 2713–2718, Tampa, Floride, December 1989.
- [37] K. M. Passino and P. J. Antsaklis. Solutions to optimal control problems for discrete event systems. Technical Report n° 70, Dept. of Electrical and Computer Engineering, University of Notre Dame, Indiana, USA, 1990.
- [38] P. J. Ramadge and W. M. Wonham. Supervision of discrete event processes. In *Proc. of 21st IEEE Conf. Decision and Control*, pages 1228–1229, Orlando, FL, USA, December 1982.
- [39] P. J. Ramadge and W. M. Wonham. Supervisory control of a class of discrete event processes. volume 63 of *LNCIS*, pages 477–498. Springer-Verlag , Berlin, Germany, 1983.
- [40] P. J. Ramadge and W. M. Wonham. Modular supervisory control of discrete event systems. In A. Bensoussan and J. L. Lions, editors, *Proc. of 7th Int. Conf. Analysis and Optimization of Syst.*, volume 83 of *LNCIS*, pages 202–214, Antibes, France, June 1986. Springer-Verlag , Berlin, Germany.

- [41] P. J. Ramadge and W. M. Wonham. Modular feedback logic for discrete event systems. *SIAM J. Control Optim.*, 25(5):1202–1218, September 1987.
- [42] M. Sampath, R. Sengupta, S. Lafortune, K. Sinnamohideen, and D. Teneketzis. Diagnosability of discrete event systems. In G. Cohen and J. P. Quadrat, editors, *Proc of the 11th Int. Conf. on Analysis and Optimization of Systems, Discrete Event Systems*, LNCS No 953, pages 73–79, Sophia-Antipolis, France, June 1994. Springer-Verlag.
- [43] M. Sampath, R. Sengupta, S. Lafortune, K. Sinnamohideen, and D. Teneketzis. Failure diagnosis using discrete event models. Technical Report CGR-94-03, Dept. of Electrical Engineering and Computer Science, University of Michigan, USA, May 1994.
- [44] R. Sengupta and S. Lafortune. A graph-theoretical optimal control problem for terminating discrete event processes. *Discrete Event Dynamic Systems: Theory and Applications*, 2(2):139–172, November 1992.
- [45] R. Sengupta and S. Lafortune. Extensions to the theory of optimal control of discrete event systems. In S. Balemi, P. Kozák, and R. Smedinga, editors, *Discrete Event Systems: Modeling and Control*, volume 13 of *Progress in Systems and Control Theory*, pages 153–160. Birkhäuser Verlag, Basel, Switzerland, 1993. (Proc. of WODES'92, Aout 1992, Prague, Czechoslovakia).
- [46] J. C. Willems. Paradigms and Puzzles in the Theory of Dynamical Systems. *IEEE Transactions on Automatic Control*, 36(3):258–294, 1991.

Contents

1	Polynomial dynamical systems	4
1.1	General form of Polynomial Dynamical Systems	4
1.2	The Algebraic Framework	5
1.2.1	Ideals and Varieties	6
1.2.2	Principal generator of an ideal	6
1.2.3	Elimination of quantifiers	7
1.2.4	Operations on dynamical behaviors.	8
2	Control of polynomial dynamical systems	8
3	Partial Order Control problem	11
3.1	Order relation controller synthesis	11
3.2	Some useful examples of order relations	12
3.2.1	Minimally restrictive constraints on uncontrollable events	13
3.2.2	Maximization of the number of state variables equal to 1	13
3.2.3	Numerical order relations	14

4	Optimal Control	15
4.1	The optimal controller synthesis problem for polynomial dynamical systems .	16
4.2	Optimal controller synthesis	16
4.2.1	Optimal value function computation	17
4.2.2	Control strategy	17
4.3	Example	18
4.4	Optimal control with perturbations	20
4.5	Examples of optimal control	20
5	Application to the power transformer station controller	21
5.1	The power transformer station description	21
5.1.1	Propagation of Power and Defect	22
5.1.2	The controller of the power transformer station	23
5.2	Incremental specification of the power transformer station controller	25
5.2.1	Logical Control Objective	25
5.2.2	Partial order control objectives	25
5.2.3	Optimal Control objective	27
6	conclusion	28



Unit é de recherche INRIA Lorraine, Technopôle de Nancy-Brabois, Campus scientifique,
615 rue du Jardin Botanique, BP 101, 54600 VILLERS LÈS NANCY
Unit é de recherche INRIA Rennes, Irista, Campus universitaire de Beaulieu, 35042 RENNES Cedex
Unit é de recherche INRIA Rhône-Alpes, 655, avenue de l'Europe, 38330 MONTBONNOT ST MARTIN
Unit é de recherche INRIA Rocquencourt, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex
Unit é de recherche INRIA Sophia-Antipolis, 2004 route des Lucioles, BP 93, 06902 SOPHIA-ANTIPOLIS Cedex

Éditeur
INRIA, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex (France)
<http://www.inria.fr>
ISSN 0249-6399