INTERNATIONAL JOURNAL OF INTELLIGENT CONTROL AND SYSTEMS
VOL. 13, NO. 2, JUNE 2008, 128-135

Solutions Based on Particle Swarm Optimization for Optimal Control Problems of Hybrid Autonomous Switched Systems

Sahbi BOUBAKER and Faouzi M'SAHLI .

Abstract— **In this paper, a novel evolutionary computation technique, the Particle Swarm Optimisation (PSO) is applied to optimal control problems of hybrid autonomous switched systems. In particular, the developed algorithms amount to the computation of the optimal switching instants which are the optimization parameters. The objective is that to minimize a performance index, depending on these instants, over a finite time horizon. We assume that a pre-assigned modal sequence is given and that at each switching instant, a jump in the state space variable may occur and that an additional cost is then associated with it. We demonstrate via numerical examples, the effectiveness of the PSO-based algorithms for such a problem known to be NP-hard and constrained nonlinear optimization one, like many scheduling problems.When compared to results obtained by gradient-based methods, PSO results seem to be very promising without requiring any regularity of the objective function to be minimized.**

Index Terms— **Hybrid Switched Systems, Optimal Control, Particle Swarm Optimization, Autonomous switching sequence**

I. INTRODUCTION

Switched systems are a particular class of hybrid systems consisting of a number of subsystems and a switching law orchestrating the active subsystem at each time instant [1] [2]. They describe many phenomena in the real-world processes such as power electronics, chemical processes[3], [4] , automotive systems and networked control systems[5], [6]. The continuous dynamics described by a set of differential/difference equations interact with the switching law in a unified framework known as the hybrid automaton [5].

The emergence of the hybrid systems modelling framework has provided a new perspective for researches in the field of optimal control. In the last few years, this class of problems has been under intensive investigations. A solution for such a problem consists on determining not only the continuous control inputs, as in a conventional optimal control problem but also on the computation of the discrete control inputs defined in terms of switching instants and their corresponding active subsystems [1]. The special case where the continuous inputs are absent or known is considered as the optimal control of autonomous switched systems on which we focus our attention in this paper. This problem has been considered in many works. In particular, works in [7] and [8] derive the derivatives of the cost function with respect to the switching instants. Non-linear optimization is used to compute optimal switching instants in which state jumps may occur. The proposed method takes advantage of the special structure of linear subsystems and quadratic cost in [7] and considers non-linear continuous dynamics in [8]. In references [9] and [2] , the same problem has been considered. The optimal switching law has been demonstrated to be a feedback control one.Reference [2] demonstrates that it is possible to identify a region in the state space in which a jump should occur. This region is computed via a suitable procedure known as the switching tables. The problem is decomposed into two levels for a discrete-time hybrid automaton. The low level enforces safety and liveness while the high level is used for performance optimization [9]. Works developed in [10] , [11] and [12] concern the computation of the switching times based on a derived formula for the gradient of the cost functional. This technique has been applied to a two-tank system in [10], to the construction of optimal switching surfaces in [12] and for sub-optimal solutions in scheduling problems in [11]. A very interesting work in [13] concerns with the optimization of amounts of jumps while work in reference [14] proposes a version of the hybrid maximum principle based on the Lagrange multiplier rule.

These conventional approaches encounter many difficulties:

(a) According to the formulations in [8] and [11], one has to verify that the continuous dynamics functions and the cost to be minimized are continuously differentiable. In fact, this propriety is needed for the gradient computation of the objective function according to the switching instants and it is sometimes very difficult to demonstrate (see, e.g, in [7],[8] and [12])

(b) There is no existence proof of optimal solutions even in the linear subsystems and quadratic cost cases. many restrictive assumptions must be considered [7].

(c) In the switched system framework, state jumps in the continuous state space may occur and associated difficulties are added to the optimization problem. Conventional methods for optimization handle theses jumps very difficultly [8].

Manuscript received February 22, 2008; revised May 14, 2008.

S. BOUBAKER and F. M'SAHLI are with "Commande Numérique de Processus Industriels" (CONPRI) Research Unit- National School of Engineers of Gabès (ENIG)- Gabès- Tunisia.

S. BOUBAKER is actually an assistant professor at High Institute of Technological Studies of Nabeul (ISETN)- Nabeul- Tunisia (email:shoubabr@yahoo.fr). F.M'SAHLI is currently a professor at departement of Electrical Engineering at National School of Engineers of Monastir (ENIM)- Monastir- Tunisia (email: faouzi.msahli@enim.rnu.tn)

(d) Formulations in [7] and [11] consider the case where the subsystems are linear and the costs are quadratic. In a general framework, subsystems can be non-linear and costs are not necessarily quadratic.

(e) Most of the results on numerical solutions of hybrid optimal control problems are designed for discrete-time hybrid systems or based on discretizations of time or state spaces which may lead to combinatoric explosions [8].

(f) Formulations based on Hamilton-Jacobi-Bellman equations require the solution of two kinds of equations: the state equations and the so called adjoint equations which have to be solved in the opposite direction of time.

(g) If the algorithm falls in a local minima it will be difficult to find the solution wich must be the global minima.

.

Due to the complexity and the unpredictability of the non-linear constrained optimization problems derived from the hybrid optimal control problems [10], a general deterministic optimal solution is impossible [15]. This provides an opportunity for intelligent evolutionary computation techniques such as the Particle Swarm Optimization(PSO).In fact the PSO technique has been successfully used in the field of hybrid optimal control problems because of its attractive features: ease of implementation, fasteness, convergence guarantees and the fact that it does not require any regularity of the index to be optimized. Reference [17] has used PSO for reactive power and voltage control considering voltage security assessment. In reference [18] , PSO has been combined with the lagrangian relaxation scheme to schedule electric power generators. An optimal control problem of a hybrid system has been studied in [19] where a study on the effect of the inertia weight on the convergence time is given. A solution for fed-batch processes optimal control problem was given in [3] , while reference [4] presents a comparison between PSO and other metaheuristic approaches, such as genetic algorithms (GA). The comparison among the algorithms was based on their final results and on the convergence speed.

This paper is organized as follows: in section II, the optimal control problem of a hybrid autonomous switched system is formulated. Section III describes the Particle Swarm Optimization (PSO) algorithm. We focus on its usefulness to solve constrained non-linear optimization problem and we illustrate the convergence aspect on a test-function. In section IV, the PSO based algorithm is used to solve optimal control problem of a switched system. We demonstrate how the difficulties encountered by "conventional" methods are overcome. In section V, we present results on the application of PSO to an internally autonomous switched system (IASS) with jumps and on viewed as autonomous switched system (VASS) without jumps. Experimental comparison among the results beteween PSO and gradient-based method are also given. Section VI concludes the paper and discusses further works.

II. PROBLEM FORMULATION

A. Autonomous Switched Systems

In this paper, we consider autonomous switched systems consisting of subsystems [8] [11]

$$\begin{aligned} \dot{x}(t) &= f_q(x(t)) \qquad (1)\\ f_q &: X \to \mathbb{R} \\ q &\in Q = \{1, ..., M\} \\ X &\subseteq \mathbb{R}^n \end{aligned}$$

where $x(t) \in \mathbb{R}^n$is the continuous state space vector, $\dot{x}(t)$ is its first derivative according to time, f_q is an indexed field of vectors and Q is a set of finite discrete variables which indicate that the system will be in M configurations.

For such a hybrid system, one can control its state trajectory evolution by choosing appropriate switching sequence σ in an interval of time $[0, t_f]$. The switching sequence is defined as

$$\begin{aligned} \sigma &= \{(q_1, t_1), (q_2, t_2), ..., (q_K, t_K)\} \\ t_0 &= 0 \quad \text{and} \quad t_{K+1} = t_f \qquad (2) \end{aligned}$$

σ indicates that the subsystem $q \in Q$ is active in $[t_{q-1}, t_q)$ and that the system switches from subsystem q to subsystem $(q+1)$ at t_q. In such an instant, a jump in the state space variable may occur and

$$x(t_q^+) = g(x(t_q^-)) \qquad (3)$$

An example of execution of a hybrid autonomous switched system is given in Fig 1.

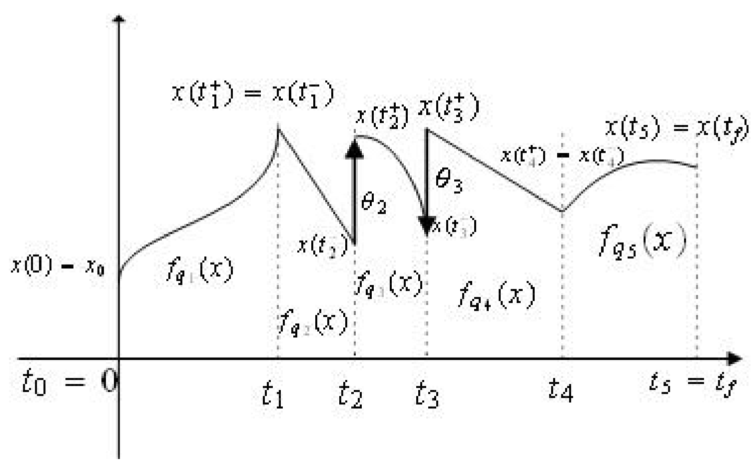

Figure 1: An example of execution with $K = 4$ switches. Note jumps at t_2 and t_3

In this paper, we consider that:

- The whole number of switches, K in the interval of time $[0, t_f]$ is finite and given.
- The modal sequence $\{q_1, q_2, ..., q_K\}$ is also given.
- The jumps are additive

B. An Optimal Control Problem of a Hybrid Autonomous Switched System

Given a hybrid autonomous switched system as described by (1), an initial continuous state, x_0 and an interval of time $[0, t_f]$. A solution for an optimal control problem of such a system amounts to the computation of the optimal switching instants $t_1^*, t_2^*, ..., t_K^*$ such that the following cost functional

$$J(t_1, t_2, ..., t_K) = \psi(x(t_f)) + \int_0^{t_f} L(x(t))dt + \sum_{q=1}^{q=K} \psi_q(x(t_q^-)) \tag{4}$$

is minimized under the constraint

$$0 = t_0 < t_1 < t_2 < ... < t_K < t_{K+1} = t_f \tag{5}$$

The index is composed of three parts:

- a part associated with the final state, $x(t_f)$
- a part relative with the continuous state evolution
- and a part associated to the continuous-state jumps occuring at the switching instants

The problem defined by equations (1), (2), (3), (4) and (5) is descretisized by a sampling period T. The time interval $[0, t_f]$ is then divided in N sampling instants such that

$$N = \frac{t_f}{T} \tag{6}$$

The approximate optimization parameters which are the switching instants are associated with a set of switching indices such that

$$(t_1, t_2, ..., t_K) \approx T \times (N_1, N_2, ..., N_K) \tag{7}$$

The state space equation $\dot{x}(t) = f_q(x(t))$ in (1) is replaced by its first order approximation

$$\begin{aligned} x_{h+1} &= T \times f_q(x_h) + x_h \\ x_{N_q} &= x_{N_q} + \theta_q \end{aligned} \tag{8}$$

where $x_h = x(t_h)$ and $t_h = T \times h$, $h \in \{0, 1, 2, ..., N_1, N_1+1, ..., N_2, N_2+1, ..., N_K, N_K+1, ..., N\}$

An algorithm for solving an optimal control problem of a system defined by equations (6), (7) and (8) amounts to the computation of the optimal switching indices $(N_1^*, N_2^*, ..., N_K^*)$ corresponding to their approximate switching instants $(t_1^*, t_2^*, ..., t_K^*)$ such that the following cost functional

$$J(N_1, N_2, ..., N_K) = \psi(x_N) + T \times \sum_{h=0}^{h=N-1} L(x_h) + \sum_{q=1}^{q=K} \psi_q(x_{N_q}) \tag{9}$$

is minimised under the constraint

$$0 < N_1 < N_2 < ... < N_K < N \tag{10}$$

III. PARTICLE SWARM OPTIMIZATION

Particle Swarm Optimization (PSO) technique is categorized into the family of evolutionary computation. It optimizes an objective function by undertaking a population based search [20]. This novel technique is inspired by social behavior of bird flocking or fish schooling. Unlike genetic algorithms, PSO has no evolution operators such as crossover and mutation. In PSO, the population is initialized randomly and the potential solutions , named particles, freely fly across the multidimensional search space [21]. During flight, each particle updates its own velocity and position by taking benefit from its best experience and the best experience of the entire population. The aim of a PSO algorithm is to minimize an objective function F which depends on a set of unknown variables, $(x_1, x_2, ..., x_n)$. In the PSO formalism, the optimization parameters are coded as a position $\overrightarrow{X}^i = (x_1^i, x_2^i, ..., x_n^i)^t$ of the particle i.The particle i is also characterized by its velocity $\overrightarrow{V}^i = (v_1^i, v_2^i, ..., v_n^i)^t$, its personal best position discovered so far $\overrightarrow{P}^i = (p_1^i, p_2^i, ..., p_n^i)^t$ and the global best position of the entire population $\overrightarrow{G}^i = (g_1^i, g_2^i, ..., g_n^i)^t$.

Let k be the iteration index in the optimization context. The new particle velocity and position are updated according to the move equations [22] [23] [24]

$$\overrightarrow{V}_{k+1}^i = w_k \times \overrightarrow{V}_k^i + b_1 \times r_1 \times (\overrightarrow{P}_k^i - \overrightarrow{X}_k^i) + b_2 \times r_2 \times (\overrightarrow{G}_k^i - \overrightarrow{X}_k^i) \tag{11}$$

and

$$\overrightarrow{X}_{k+1}^i = \overrightarrow{X}_k^i + \overrightarrow{V}_{k+1}^i \tag{12}$$

At each iteration, the behavior of a given particle is a compromise between three possible choices (see Fig 2):

- to follow its own way
- to go toward its best previous position
- to go toward the best neighbor

In this paper, we have chosen the PSO algorithm version consisting of:

- $b_1 = b_2 = 0.7$
- r_1 and r_2 are two random numbers with values in the range $[0, 1]$
- w_k evolves according to the equation

$$w_k = w_{\max} - \frac{w_{\max} - w_{\min}}{k_{\max}} \times k \tag{13}$$

where $w_{\max} = 0.9$, $w_{\min} = 0.4$ and $k_{\max}$ is a pre-fixed maximum number of iterations.

- The swarm size is $s = 20$.
- The swarm has a full connected topology in which, for the i'^{th} particle, all particles of the swarm are considered as neighbors and $\overrightarrow{G}^i = \overrightarrow{G}$ is the best position of the entire population

We also use the following notations for each particle:

- $F(\overrightarrow{X}^i)$ as the actual fitness
- $F(\overrightarrow{P}^i)$ as the best previous personal fitness
- $F(\overrightarrow{G}^i) = F(\overrightarrow{G})$ as the best global fitness

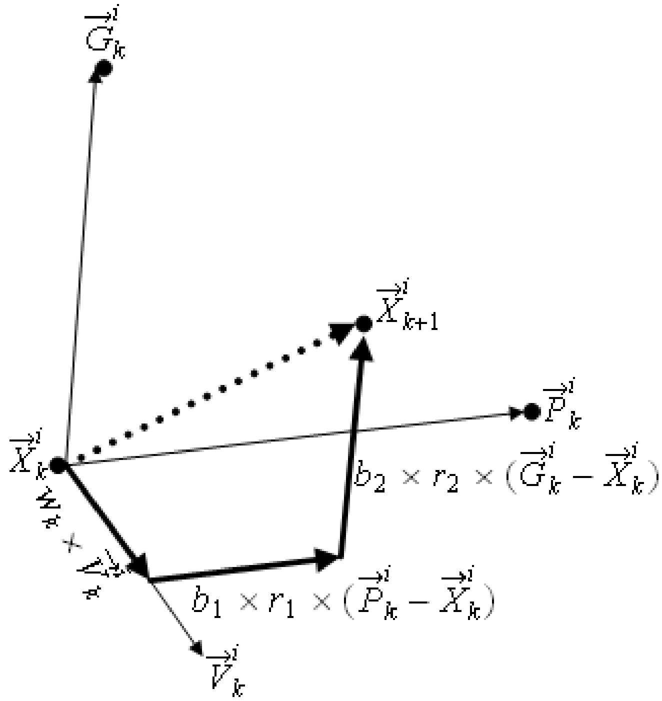

Figure 2: The particle flight illustration

A. Pseudo-Code of the PSO-based algorithm

The detailed operation of the particle swarm optimization is given in the pseudo-code below:

Step 1:

- Initialize randomly the particles by positions and velocities in the search space. The initialization mechanism must take into account eventual constraints on the optimization parameters
- For each particle, i , set $\overrightarrow{P}^i = \overrightarrow{X}^i$ and $F(\overrightarrow{P}^i) = F(\overrightarrow{X}^i)$

Step 2:

- set $F(\overrightarrow{G}) = \min_i(F(\overrightarrow{X}^i))$ with $i \in \{1, 2, ..., s\}$ and s is the swarm size, record I as the corresponding indice
- set $\overrightarrow{G}^i = \overrightarrow{G} = \overrightarrow{X}^I$

Step 3:

- For each particle update velocity and position according to equations (11), (12) and (13)
- test eventual constraints , if violated reinitialize particles by positions in the search-space
- calculate $F(\overrightarrow{X}^i)$ for each particle i, if $F(\overrightarrow{X}^i) < F(\overrightarrow{P}^i)$ set $\overrightarrow{P}^i = \overrightarrow{X}^i$ else if return to step 2.

Step 2 and step 3 are repeated until a stop condition is met, such as a prefixed maximum number of iteration $k_{\max}$, a minimum value for the objective function $F_{\min}$ or a failure to make progress for a certain number of iterations.

Once terminated, the PSO algorithm reports $\overrightarrow{G}$ and $F(\overrightarrow{G})$ as its solution.

B. PSO convergence illustration

The convergence aspect of the PSO algorithm is illustrated in a well-known test function, the Rosenbrock function $F(x_1, x_2) = 100 \times (x_2 - x_1^2)^2 + (x_1 - 1)^2$. The optimization variables x_1 and x_2 are coded as the position for each particle in the PSO formalism. The analytic solution is $(x_1^*, x_2^*) = (1, 1)$. We choose a search-space to be $[0, 5] \times [0, 5]$ on which we initialize randomly the 20 particles of the swarm (see Fig 3(a)). Respectively in Fig 3 (b) and Fig 3 (c), we show the situation of the swarm after 50 and 100 iterations. Note that the particles move towards the analytic solution. At the final iteration (Fig 3 (d)), all the particles come closer to this optimal solution.

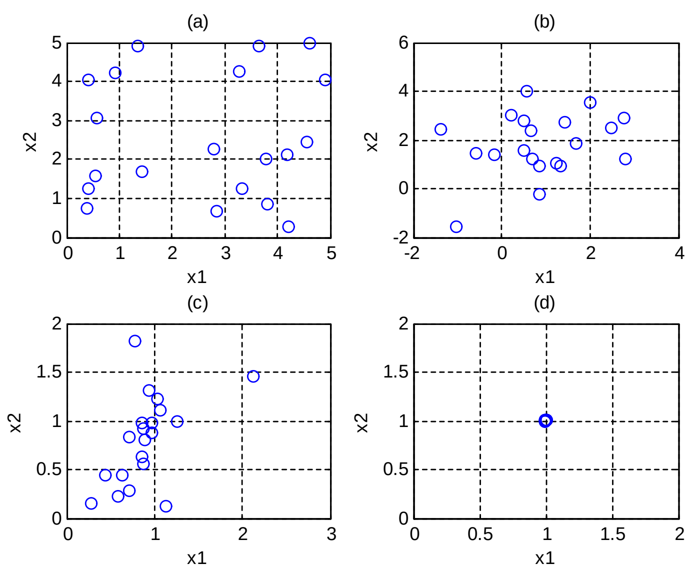

Figure 3: Convergence of PSO: illustration on two-dimensional Rosenbrock function-test

IV. A BASED ON PSO ALGORITHM FOR AN AUTONOMOUS SWITCHED SYSTEM OPTIMAL SWITCHING SEQUENCE

In this paper, we develop an algorithm PSO-based to approximate some sub-optimal solutions for an autonomous switched system bahavior as defined by equations (1), (2), (3), (4) and (5) and transformed to an "equivalent" problem as described by (7), (8), (9) and (10) via the sampling time period as defined by (6).The PSO version used in this paper has been originally designed for continuous optimization problems. An adaptation mechanism is adopted to make it deal with discrete optimization parameters as in our case. In particular, we demonstrate how difficulties encountered by conventional methods based on gradient-projection or dynamic programming are overcome.

A. Optimization parameters definition

In the first step, one should determine the parameters that need to be optimized and give them minimum and maximum ranges taking into account eventual constraints. In PSO formalism, these variables are coded as the position of a particle of the swarm, known to be a potential solution. In our problem, the unknown parameters are the switching indices $(N_1, N_2, ..., N_K)$. For the i'^{th} particle, its position is defined as $\overrightarrow{X}^i = (N_1^i, N_2^i, ..., N_K^i)^t$. Theses variables are initialized as random integers respecting the constraint (10) [25].

B. Update Positions

The update mechanism defined by equations (11), (12) and (13) , generates real continuous values for the position $\overrightarrow{X}^i$. Optimization parameters are integer numbers. A simple way consists in rounding them to their integers parts.

C. Constraints Manipulation

To respect the constraint on optimization parameters , (10), the PSO must reinitialize a particle position which violate the constraint. This procedure has the disadvantageous that the algorithm bahaves for such a particle as it is in step 1 of the pseudo-code (section III.A).

D. Objective Function Evaluation

The PSO algorithm is mainly based on the objective function (to be minimized) evaluation. In opposite of test functions (see illustrative examples in section V), whose expressions are explicit, the problem studied in this paper has complicated objective function because it depends on the dynamic continuous bahavior of the switched system and on its disrete switching sequence. Given a switching sequence, $(1, 2, ..., K)$, an initial continous state, x_0, and a set of some switching indices $(N_1, N_2, ..., N_K)$. The following procedure evaluates the objective function J as defined by (9).

`Procedure for objective function evaluation`

$J_0 = T \times L(x_0)$

`//Initialise the objective function`

---

for h=0:N_1-1

`//q=1,the continuous dynamic f1(.)is active`

$x_{h+1}=T\times f_1(x_h)+x_h$
$J_{h+1} = J_h + T \times L(x_h)$
$x_{N_1} = x_{N_1} + \theta_1$
$J_{N_1} = J_{N_1-1} + \psi_1(x_{N_1})$

---

for h=N_1:N_2-1

`//q=2,the continuous dynamic f2(.)is active`

$x_{h+1}=T\times f_2(x_h)+x_h$
$J_{h+1} = J_h + T \times L(x_h)$
$x_{N_2} = x_{N_2} + \theta_2$
$J_{N_2} = J_{N_2-1} + \psi_2(x_{N_2})$

`//computations are repeated until q=K`

---

for h=N_K:N-1

`//q=K,the continuous dynamic fK(.)is active`

$x_{h+1}=T\times f_K(x_h)+x_h$
$J_{h+1} = J_h + T \times L(x_h)$
$x_N = T\times f_K(x_N)+x_{N-1}$
$J = J_N = J_{N-1} + \psi(x_N)$

The PSO fastness depends on the number of evaluations at each iteration which depends on the particles number in the swarm [22]. But it is obvious that this performance criterion is not so sufficient because the objective function complexity affects seriously the fastness. The problem studied in this paper is known to be NP-hard like many scheduling problems [10] and difficult optimization one like many non-linear constrained optimization problems [15].

E. How PSO overcomes difficulties?

In the following, we give some comments on how PSO overcomes difficulties encountered by conventional (enumerated in section I) methods used to solve the same problem as in this paper:

(a) The functions fq and L continuity and differentiability are not needed because PSO algorithm is based on objective function evaluation and it does not require gradient computation

(b) A solution quality is defined by the PSO user. The solution attained at the stop condition is considered as the "best" one

(c) Additive jumps at the switching instants do not add any difficulty for the PSO because they are considered only in the computational side (see the procedure of objective function evaluation)

(d) PSO considers problems with quadratic or not quadratic problems, linear or nonlinear continuous dynamics. In fact, PSO was demonstrated to be very sufficient for non-linear optimization problems.

(e) Combinatoric explosions are specially encountered by optimization methods which consider all the possible discrete evolutions (Branch and Bound methods). PSO surrounds all the search space with an intelligent way. Only scenarios which ameliorate the objective function are considered and evaluated.

(f) In PSO, there are not any solutions for differential equations solutions. The algorithm uses the objective function evaluation and does not require gradient computation.

(g) PSO algorithm gives always global solutions. The problem of local minima is not raised [16].

V. TWO ILLUSTRATIVE EXAMPLES

In this section, we use the PSO algorithm described in section III for optimal switching sequences of an Internally Autonomous Switched System (IASS) and a Viewed as an Autonomous Switched System (VASS). The illustrative examples have been considered in the literature by algorithms based on gradient-projection. We compare results obtained by our approach with the ones in the literature.

A. Example 1: Internally Autonomous Switched System (IASS) with state jumps

The same example was treated in [8] by the gradient-projection method for optimization.

Consider a hybrid autonomous non linear switched system consisting of

$$\begin{aligned} Subsystem1: \dot{x}_1 &= x_1 + 0.5 \sin x_2 \\ \dot{x}_2 &= -0.5 \cos x_1 - x_2 \end{aligned} \quad (14)$$

$$\begin{aligned} Subsystem2: \dot{x}_1 &= 0.3 \sin x_1 + 0.5 x_2 \\ \dot{x}_2 &= 0.5 x_1 + 0.3 \cos x_2 \end{aligned} \quad (15)$$

$$\begin{aligned} Subsystem3: \dot{x}_1 &= -x_1 + 0.5 \cos x_2 \\ \dot{x}_2 &= 0.5 \sin x_1 + x_2 \end{aligned} \quad (16)$$

Assume that $t_f = 3$ and that the system switches from $Subsystem1$ to $Subsystem2$ at $t = t_1$ and from $Subsystem2$ to $Subsystem3$ at $t = t_2$. Constraint on t_1 and t_2 is defined by

$$0 < t_1 < t_2 < t_f \quad (17)$$

We assume also that state jumps occur at t_1 and t_2 and that they are defined by

$$\begin{aligned} x_1(t_1^+) &= x_1(t_1^-) + 0.2 \\ x_2(t_1^+) &= x_2(t_1^-) + 0.2 \end{aligned} \quad (18)$$

$$\begin{aligned} x_1(t_2^+) &= x_1(t_2^-) + 0.2 \\ x_2(t_2^+) &= x_2(t_2^-) - 0.2 \end{aligned} \quad (19)$$

We want to find optimal switching instants t_1^* and t_2^* such that the cost functional

$$\begin{aligned} J(t_1, t_2) &= \frac{1}{2} x_1^2(3) + \frac{1}{2} x_2^2(3) + \frac{1}{2} \int_0^3 (x_1^2(t) + x_2^2(t)) dt \\ &+ \sum_{q=1}^{q=2} \frac{1}{2} x_1^2(t_q^-) + \frac{1}{2} x_2^2(t_q^-) \end{aligned} \quad (20)$$

is minimized. We take $x_1(0) = x_{10} = 1$ and $x_2(0) = x_{20} = 3$.

To obtain results with the same accuracy as in [8] , we have chosen the sampling period to be $T = 0.001$. The stop condition of the PSO algorithm is taken to be a prefixed maximum number of iterations, $k_{\max} = 10$.Because of the stochastic character of PSO, the algorithm has been run 100 $times$. histograms in Fig 4 show that results for the two optimization parameters are distributed according to normal distributions with $t_1^{average} = 0.4902$ and $t_2^{average} = 1.9321$ as averages.we adopt $t_1^* = 0.4902$ (0.4847 in [8]) and $t_2^* = 1.9321$ (1.9273 in [8]). Their corresponding state trajectory is shown in Fig. 5. In Fig. 6, we give also x_1 and x_2 evolutions. Note the state jumps at t_1 and t_2.

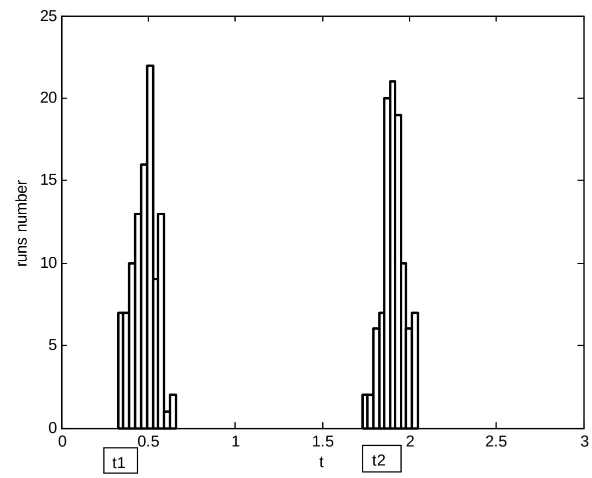

Figure 4: Histogram for results distributions.

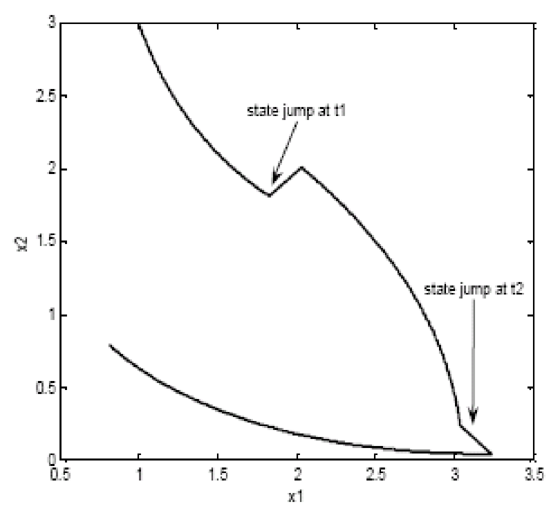

Figure 5: State trajectory for example 1

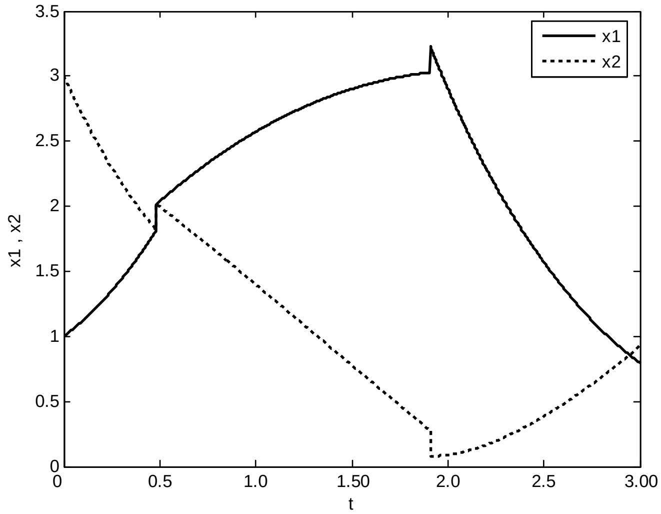

Figure 6: States evolutions for example 1. Note jumps at t_1 and t_2

B. Example 2: A Viewed as Autonomous Switched System (VASS) without state jumps

To illustrate our approach, we consider the same problem considered by many researchers and in particular by [10]. The studied framework consists in adjusting the fluid level in the lower tank by the control of the fluid level in the upper tank. The control variable is only the switching law applied to the valve V (see Fig. 7) which is switched (by external way) between three configurations: closed, half-open and fully-open.

The state vector is composed of the fluid levels in the two tanks, respectively h_1 for the upper and h_2 for the lower.

$$x(t) = \begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix} = \begin{pmatrix} h_1(t) \\ h_2(t) \end{pmatrix} \tag{21}$$

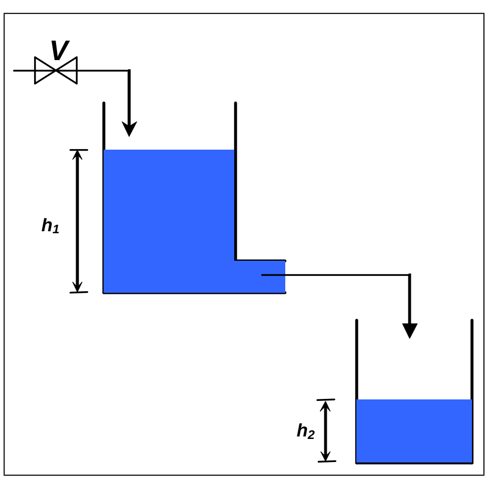

Figure 7: The Two-tank system studied

$x(t)$ evolves according to the following differential equations [10]

$$\dot{x}(t) = \begin{pmatrix} -\alpha_1\sqrt{x_1(t)} + u \\ \alpha_1\sqrt{x_1(t)} - \alpha_2\sqrt{x_2(t)} \end{pmatrix} \tag{22}$$

where

- $u = u_0 = 0$ for f_0 which correspnds to V closed
- $u = u_1 = 0.5 \times u_{\max}$ for f_1 which correspnds to V half-open
- $u = u_2 = u_{\max}$ for $f_2$which correspnds to V fully open

We assume also that $\alpha_1 = \alpha_2 = 1$ and $u_{\max} = 1$

Note that the system is not autonomous because it is controlled by the continuous variable u. But this variable is known of each subsystem so the system is viewed as a hybrid autonomous one.

Given an initial state $x(0) = x_0 = \begin{pmatrix} 0.8 \\ 0.2 \end{pmatrix}$ and a modal sequence $\{f_1, f_2, f_1, f_2, f_0, f_2, f_1, f_2, f_1\}$ as in [10], the PSO algorithm amounts to the computation of optimal switching instants $\{t_1^*, t_2^*, t_3^*, t_4^*, t_5^*, t_6^*, t_7^*, t_8^*\}$ such that the fluid level in the lower tank h_2 tracks a given value denoted by x_r and the cost functional

$$J(t_1, t_2, t_3, t_4, t_5, t_6, t_7, t_8) = K \times \int_0^{t_f} (x_2(t) - x_r)^2 dt \tag{23}$$

is minimized. Here, $K = 10$, $x_r = 0.5$ and $t_f = 5$ is the finite time horizon.The stop condition of the PSO is taken to be a minimum value of the cost functional to be minimized. here $J_{\min} = 0.3$.

the results obtained are reported in table I. Fig. 8 shows the state trajectory associated with the reference x_r. It can be seen that h_2 tracks x_r very well and that $h_2(t_f) = 0.5063$ comes close to the reference x_r.

t_1^*	t_2^*	t_3^*	t_4^*	t_5^*	t_6^*	t_7^*	t_8^*	J^*
0.3	0.9	1.7	2.1	2.6	3.2	3.9	4.5	0.2907

Table I: Results obtained by PSO for example 2

VI. CONCLUSIONS

In this paper, we propose solutions based on Particle Swarm Optimization for sub-optimal control problems of hybrid autonomous switched systems defined on switched-mode dynamical subsystems. The design parameters are the switching instants. In particular, we demonstrate how PSO overcomes difficulties to solve such an NP-hard constrained problem encountered by conventional methods based on gradient projection. In fact, the PSO-based algorithm is demonstrated to be very efficient to optimize switched systems behavior even with non-linear dynamics, irregular objective functions and state jumps occuring at the switching instants. The ease of implementation, the speed and the ease of constraints manipulation open very promising investigation ways in hybrid systems. Further work will focus on optimizing such systems on both discrete and continuous inputs.

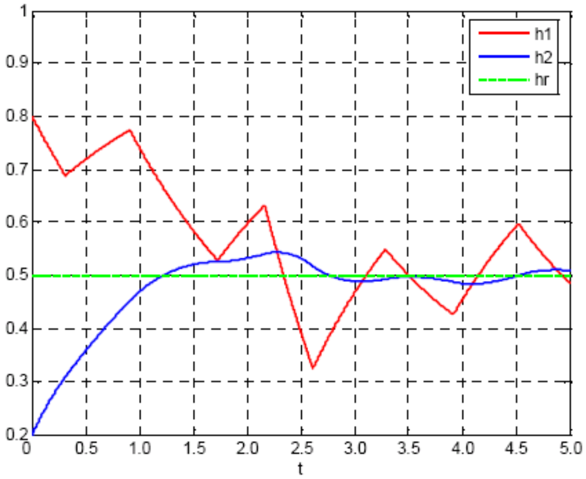

Figure 8: heights and reference evolutions

REFERENCES

[1] X. Xu and P.J. Antsaklis, "Optimal Control of Switched Systems Based on Parametrization of the Switching Instants," IEEE Trans. Automat. Contr.,vol. 49, no. 1,Jan,2004.

[2] A. Giua, C. Seatzu and C. Van Der Mee, ," Optimal Control of Autonomous Linear Systems Switched with pre-assigned finite sequence," Proc. IEEE Int. Symposium on Intelligent Control.2001.

[3] A. Ismael, F. Vaz and E. C. Ferreira, " Optimal Control of fed-batch Processes with Particle Swarm Optimization" ,Sicilia et al. Ed.2006

[4] R. Mendes, I. Rocha, E.C. Ferreira and M. Rocha, "A comparison of algorithms for the optimization of fermentation processes," Proc. IEEE Cong. Evolutionary computation.2006

[5] M.S. Branicky, V.S. Borkar and S.K. Mitter ,"A Unified Framework for Hybrid Control: Model and Optimal Control Theory," IEEE. Trans. Aut. Contr, Jan, 1998.

[6] G. Zheng, W. Zeng and F. Xu, "Computation on the optimal control systems with multiple switching modes over high speed local area networks", Int. Journal of Computers, Communications and control, Vol. I, No. 4 pp. 101-109, 2006.

[7] X. Xuping and P.J Antsaklis , " Quadratic Optimal Control Problems of Hybrid linear Autonomous Systems with State Jumps," Proc. American Cont. Conf, 2003.

[8] X. Xuping and P.J Antsaklis, 2003 ," Optimal Control of Hybrid Autonomous Systems with State Jumps," in Proc. American Cont. Conf, 2003.

[9] D. Gromov, E. Mayer, J. Raish, D. Corona, C. Seatzu and A. Giua, , "Optimal Control of discrete-time hybrid Automata under safety and liveness constraints," in Proc. IEEE int. Symposium on intelligent Control,2005.

[10] H. Axelsson, M. Egerstedt, Y. Wardi and G. Vastsevanos " Algorithm for Switching- Time Optimization in Hybrid Dynamical System", in Proc. IEEE int. Symposium on intelligent Control, 2005.

[11] M. Egerstedt, Y. Wardi and H. Axelsson, "Transition-Time for Switched-Mode Dynamical Systems", IEEE Transactions on Automatic Control, Vol. 51, NO. 1, January 2006.

[12] M. Boccadoro, M. Egerstedt, P. Valigi and Y. Wardi, , " Beyond the construction of Optimal Switching Surfaces for Autonomous Hybrid Systems," in Preprints. 2nd. Conf. on Analysis and Design of Hybrid systems, 2006.

[13] S.A. Attia, V. Azhmyakov and J. Raish, "State jump optimization for a class of hybrid autonomous systems", 16th IEEE International Conference on Control Applications, Singapore, October 2007.

[14] V. Azhmayakov and al, "Necessary Optimality conditions for a Class of hybrid Optimal Control Problems", HSCC 2007, LNCS 4416, pp. 637-640, 2007.

[15] X. Hu and R. Eberhart, "Solving Constrained Optimization Problems with Particle Swarm Optimization", In Procceedings of sixth World conferebce on Systemics, Cybernetics and Informatics, 2002.

[16] J. H. Seo and al, "Multimodal Function Optimization Based on Particle Swarm Optimization", IEEE Transactions on Magnetics, Vol. 42. N0. 4, April, 2006.

[17] H. Yoshida, K. Kawata, Y. Fukuyama, S. Takayama and Y. Nakanishi, , " A particle swarm optimization for reactive power and voltage control considering voltage security assessment," in IEEE trans. On power systems, Vol. N°4, pp. 1932-1939.2001

[18] H. H. Balci and F. George,, " Scheduling electric power generators using particle swarm optimization combined with the lagrangian relaxation method," in Int. J. Appl. Math. Compt. Sci. Vol. 14, No. 3, pp. 411-421. 2004.

[19] M. Senthil Amurugam and M. V. C. Rao, 2004 , " On the performance of the particle swarm optimization algorithm with with various inertia weight variant for computing optimal control of a class of hybrid system," in Hindawi Publisher Corporation in nature and society, pp. 1-17.2004

[20] J. Kennedy R. C. Eberhart, " Particle Swarm Optimization, , " in proc. IEEE. Int. Conf. on Neural-Networks, pp. 1942-1948.1995

[21] K. O. Jones, "comparison of genetic algorithms and particle swarm optimization for fermentation feed profile determination", Int. Conf. on Computer Systems and Tecnologies, 2006.

[22] M. Clerc, L'optimization par essaims particulaires, Ed. Lavoisier, 2005.

[23] Y. Shi, "Particle Swarm Optimization", IEEE Neural Networks Society. 2004,

[24] A.Abraham,H. Guo and H. Liu, ,"Swarm Intelligence: foundations, perspectives and applications," Springer-Verlag.2006

[25] S. Doctor and G. K. Venayagamoorthy, , "Improving the performance of particle swarm optimization using adaptive critics designs," IEEE.2005

Sahbi Boubaker received his engineer diploma in Electrical Engineering and his Master in Automatic from National School of Engineers of Tunis (ENIT) - Tunisia in 1997 and 1999 respectively. He is currently a Phd condidate at National School of Engineers of Monastir (ENIM) - Tunisia. His research interests include Hybrid Dynamical Switched Systems, Identification, Optimal Control and Intelligent Optimization.

Faouzi M'SAHLI received his Mastery of Sciences and DEA from ENSET- Tunis - Tunisia in 1987 and 1997 respectively. In 1995, he obtained his doctorate degree in Electrical Engineering from ENIT- Tunis - Tunisia. He is currently professor of Electrical Engineering at National School of Engineers of Monastir (ENIM)- Tunisia. His research interests include Modeling, Identification, Predictive and Adaptive Control of Linear and Non-Linear Systems. He has published over 80 technical papers and Co-author of a book "Identification et Commande Numérique de Procédés Industriels" Ed- Technip, Paris, France.