Safe and near optimal controller synthesis for Stochastic Hybrid Games

Richard Valentín Yantas Alcantara

Advisor: PhD. Marco Muñiz

Committee Members:

Thesis submitted to the
Department of Computer Science
in partial fulfillment of the requirements for the degree of
Master in Computer Science.

Universidad Católica San Pablo – UCSP
July of 2021 – Arequipa – Peru

To my parents, Mario and Maria, who never stop giving of themselves in countless ways. To my brothers and sister, who encourage and support me.

Abbreviations

Acknowledgments

First and foremost, I want to thank God for having guided me throughout these two years of study.

I would like to thank my family for their continuous support and encouragement during these years of study.

I would like to thank in a special way to the National Council for Science, Technology and Technological Innovation (CONCYTEC-PERU) and to the National Fund for Scientific Development, Technological and Technological Innovation (FONDECYT-CIENCIACTIVA), which through the Management Agreement 234-2015-FONDECYT have allowed the grant and financing of my studies in the Master Program in Computer Science at Universidad Católica San Pablo (UCSP).

I would like to express my gratitude and appreciation to my advisor Marco Muñiz for giving his guidance and support throughout the Master program and the preparation of this thesis.

Finally, I thank all the people who directly or indirectly helped me in the preparation and presentation of this work, either by exchanging ideas, giving me advice and recommendations or by encouraging me to continue.

Abstract

Nowadays, hybrid systems are widely used in industry problems like optimization and safety are some of the most important, due to some advances in machine learning and model Checking was possible to implement a methodology to improve a real time system using reinforcement learning and boundary computation to explore limits in order to guarantee a safe and near optimal behaviour. Hybrid systems that are most common in engineering applications(*e.g.*, logic-dynamic controllers, internet congestion even physical systems with impact,*etc.*). Hybrid systems have been used to model several cyber-physical systems. For this purpose, in this document we implement a"Solar water heating" as case study in order to analize metrics which allow us to evaluate our results comparing with traditional controllers. In additionm we present a toolbox UPPAAL STRATEGO to explore multiple strategy in efficient way and get patterns which allow us to optimize.

Keywords: Control Theory, Game Theory, Machine Learning, Model Checking.

Resumen

TRADUCIR DESPUES DE DEFINIR EL ENGLISH ABSTRACT .....

Palabras clave: Teoría de juegos, Aprendizaje automático, Verificación de modelos, Teoría de control.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

In Section 1.1 we describe the motivation and context of our work, Section 1.2 presents our problem statement. Section 1.3 shows the objectives of this work. Finally, Section 1.4 describes the structure of this thesis document.

1.1 Motivation and Context

Hybrid systems are widely used in engineering applications and its importance has grown up considerably these last years, because of their ease of implementation for controlling cyber-physical systems. A switched systems is a set of dynamical systems, each with its own dynamical behaviour controlled by a parameter mode u whose values are in a finite set U(See Liberzon (2003)). However, due to the composition of many switched systems together, the global switched systems has a number of modes and dynamics which increases exponentially. Switched systems have numerous applications in control of mechanical systems, the automotive industry, and many other fields.

1.2 Problem Statement

Nowadays, there is a large number of methods to solve switched systems; however, it does not have guarantee in safety. For that reason, we propose a new approach to solve switched systems.

1.3 Objectives

General Objective

Our main objective is to propose a pipeline to solve switched systems guarantee safety and near optimal synthesis controllers.

Specific Objectives

To achieve our main objective, we have the following specific objectives:

- Define a case of study as a stochastic hybrid game and get its mathematical model.
- Implement and explore safe patterns for the model.
- Implement an API between the simulator and UPPAAL to Optimize the model.
- Compare results with traditional methods with our methodology.

1.4 Contributions

This thesis proposes a novel approach to solve switched systems guaranteeing safety and optimal controller synthesis.

- Find safe patterns for our case study using decomposition algorithm (Le Coënt et al. (2017)).
  - We propose a method to guarantee safe controller. This methods consider three regions to have reachability and safety in the system.
  - We also demonstrate the utility of the proposed method comparing with traditional methods.
- Find a near optimal patterns to online control(Larsen et al. (2016)).
  - Implement an API between simulator and UPPAAL.
  - Build a timed automata MDP in UPPAAL for a real time MPC.

1.5 Outline

This thesis document is divided into five chapters. After this introduction and problem formulation, in Chapter 2 we survey the literature about the recent research in safety controllers and near optimal online controller synthesis. Chapter 3 presents some basic concepts about the hybrid systems, traditional controllers, stability criteria and optimal controller technique. Next, in Chapter x we describe in detail the corpus, techniques used by our pipeline and their evaluation results. Finally, the limitations, future works, and conclusions of this work are presented in Chapter.

Chapter 2

Related Works

Our work draws on prior research in the areas of hybrid systems, safe patterns and online control synthesis.

2.1 Online and Compositional Learning of Controllers with Application to Floor Heating

This work has proposed one method to perform *optimal controller synthesis for stochastic hybrid switched systems* e.g. a floor heating system in a house Larsen et al. (2016). Is proposed a general and scalable methodology for controller synthesis for such systems. Instead of off-line synthesis of a controller for all possible input temperatures and an arbitrary time horizon, is proposed an on-line synthesis methodology, where it periodically compute the controller only for the near future based on the current sensor readings.

2.2 An Improved Algorithm for the Control Synthesis of Nonlinear Sampled Switched Systems

In this paper is presented and algorithm for the control synthesis for nonlinear switched systems using and existing procedure of state-space and made available for nonlinear systems with the help of guaranteed integration, the algorithm has been improved to be able to consider longer patterns of modes with a better running approach.

This approach permits to deal with stability, reachability, safety and forbidden region constraints. The approach was numerically validated on several examples taken from the literature.Le Coënt et al. (2017)

2.3 Distributed Synthesis of State-Dependent Switching Control

In this part is presented a correct-by-design method of state-dependent control synthesis for linear discrete-time switching systems. Given an objective region R of the state space, the method builds a region S. Le Coënt et al. (2016)

2.4 Final Considerations

This chapter presented some recent proposals related to our thesis work. Some research works have been focused on analyzing safety controller over switched systems. On the other hand, other works have focused on optimize controller synthesis using some techniques related to model checking.

The next chapter will present some concepts needed to understand better our work, those concepts are related to the modelling physical system over our case of study and represent as a state space.

Chapter 3

Background

In this chapter we present basic concepts that are needed to understand better our work. In Section 3.1 we define some definitions about switched systems and the difference between hybrid system a basic concept that we will use for our analysis,

3.1 Switched systems

Switched systems are loosely defined as dynamical system whose state has two components, one of which evolves in a continuous set such as $\mathbb{R}$ while the other evolves in a discrete set such as $\mathbb{N}$ according to some transition logic based rule. Hybrid System is an bastract meaning of switched system which is a system which changes acording to switch rule, this rule comprises of discrete events, (Branicky, 1994).

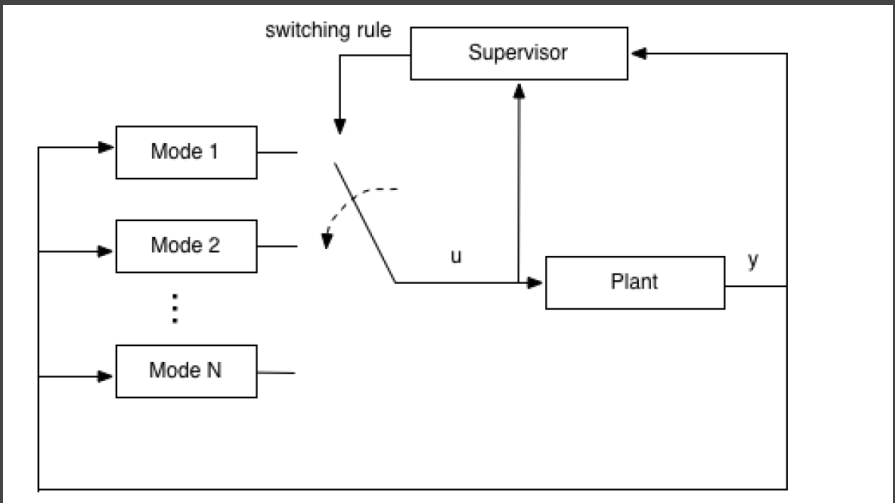


Figure 3.1: atched System Schematic

Let us consider nonlinear switched systems such that

$$\dot{x} = f_{\sigma(t)}(x(t), d(t)) \tag{3.1}$$

defined for all $t \geqslant 0$, where $x(t) \in \mathbb{R}^n$ is the state of the system, $\sigma(.) : \mathbb{R}^+ \to U$ is the switching rule, and $d(t) \in \mathbb{R}^m$ is a bounded perturbation. The finite set $U = \{1, 2, ..N\}$ is the set of switching modes of the system. We focus on sampled switched systems: given a sampling period $\tau > 0$, switching will occur at time $\tau, 2\tau, ...$ Switching depend only on time, and not on states: *this is the main difference with hybrid system.*

We work in the *synchronous* setting for discrete events. This means that all the discrete events are supposed to occur at periodic instants: $\tau, 2\tau, 3\tau, ...$ The switching rule $\sigma(.)$ is thus piecewise constant, we shall consider that $\sigma(.)$ is constant

on the time interval $[(k-1)\tau, k\tau[$ for $k \geqslant 1$. We call *"pattern"* a finite sequence of modes $\pi = (i_1, i_2, ...i_k) \in U^k$. With such a control input, and under a given perturbation d, we shall denote by $x(t; t_0; x_0, d, \pi)$ the solution at time t of the system.

$$\begin{aligned} &\dot{x}(t) = f_{\sigma(t)}(x(t), d(t)), \\ &x(t_0) = x_0, \\ &\forall j \in \{1, ..., k\}, \sigma(t) = i_j \in U \text{ for, } t \in [1, 2[\end{aligned} \tag{3.2}$$

We will use 3.2 in order to have a clear understanding about some definitions that will be useful for our methodology.

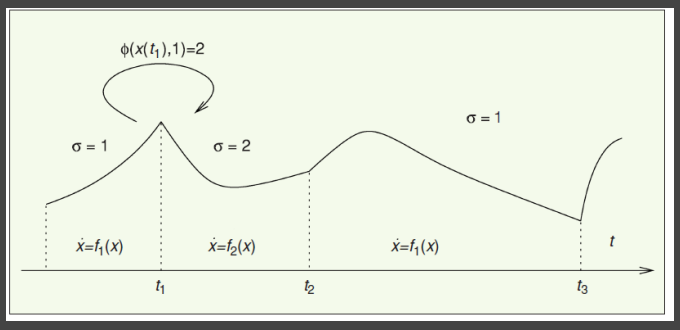


Figure 3.2: Trajectory of a hybrid system. The switching signal $\sigma(t)$ takes on integer values that change at discrete-time instances.(Liberzon, 2003)

3.2 Safe patterns computation

In this part is presented a some definitions based on an existing procedure of state-space bisection and made available for nonlinear systems with the help of guaranteed integration, the algorithm has been improved to be able to consider longer patterns of modes with a better pruning approach. The method consist of given a objective region R, a set S and a forbidden reigion B, the method find a pattern π_i, $i \in \mathbb{N}$. Le Coënt et al. (2016), some definitions are shows as follow:

Definition 1 *((R,S) - Stability Problem).* Given a switched system as shown in figure before, a set of recurrence $\mathbb{R}^n$ and a safe set $S \subset \mathbb{R}^n$,**there are** a control rule $\sigma : \mathbb{R}^+ \to U$ such that, for any initial condition $x_0 \in R_1$ and any perturbation $\varpi : \mathbb{R}^+ \to U$ the following holds:

- *Recurrence in* R*: There are a monotonically strictly increasing sequence of (positive) integers* $\{k_l\}_{l\in\mathbb{N}}, t \in \mathbb{N}$ *such that for all* $l \in \mathbb{N}, \phi(k_l\tau; t_0, x^0, \sigma, w) \in R$.
- *Stability in* S*: For all* $t \in \mathbb{R}^n, \phi(t; t_0, x^0, \sigma, w) \in S$.

Definition 2 *((*R_1, R_2, S*) - Reachability). Given a switched system of the form shown above, two sets* $R_1 \subset \mathbb{R}^n$ *and* $R_2 \subset \mathbb{R}^n$ *and a safety set* $S \subset \mathbb{R}^n$*,* R_2 *is reachable if there is a control rule* $\sigma : \mathbb{R}^+ \to U$ *such that, for any initial condition* $x_0 \in R_1$ *and any perturbation* $\varpi : \mathbb{R}^+ \to U$*, the following holds:*

- *Reachability from* R_1 *to* R_2*: there exists an integer* $k \in \mathbb{N}$ *such that we have* $\phi(k_l\tau; t_0, x^0, \sigma, w) \in R_2$.
- *Stability in S: for all* $t \in \mathbb{R}^+, \phi(t; t_0, x^0, \sigma, w) \in S$.

Definition 3*(Safe Control Synthesis).* Let us consider a sampled switched system. Given three sets R, S and B, with $R \cup B \in S$ and $R \cap B = \varnothing$. The system is safe if there is a rule $\sigma(.)$ such that, for any $x(0) \in R$.

- τ*-stability:* $x(t)$ *return in* R *infinitely often, at some multiples of sampling time* τ.
- *safety:* $x(t)$ *always stays in* S/B.

Algorithm 1 Decomposition

1: **procedure** DECOMPOSITION(W, R, S, B, D, K) ▷ List of set of patterns
2: **Input:** A box W, a box R, a box S, a box B, a degree D of bisection, a length K of input pattern
3: **Output:** $\{(V_i, \pi_i)\}_i, True$ or $<, False>$
4: $(\pi, b) := FindPattern(W, R, S, B, K)$
5: **if** $b = True$ **then**
6: return $< \{(W, pat)\}, True >$
7: **else**
8: **if** $D = 0$ **then**
9: return $<, False>$
10: **else**
11: Divide equally W into (W_1, W_2)
12: **for** $i = 1, 2$ **do**
13: $(, b_i) = Decomposition(W, R, SB, D - 1, K)$
14: **return** $<, False>$.

Algorithm 2 Find Pattern with forbidden transitions

1: **procedure** FINDPATTERN(W, R, S, B, K) ▷ List of set of patterns
2: **Input:** A box W, a box R, a box S, a box B, a length K of input pattern
3: **Output:** $< \prod, True >$ *or* $<, False >$
4: $\mathcal{S} = \{\emptyset\}$
5: $\mathcal{L} = \{(W, W, \emptyset)\}$
6: **while** $\mathcal{L} \neq \emptyset$ **do**
7: $e_{current} = takeHead(\mathcal{L})$
8: **for** $i \in U$ **do**:
9: **if** $Pair(l, i) = False$ **then** ▷ l is the current mode
10: **continue** ▷ Not considered transition l, i
11: **if** $Post_i(e_{current}.Y_{current}) \subseteq R$ **and** $Tube_i(e_{current}.Y_{current}) \cup B = \emptyset$ **and** $Tube_i(e_{current}.Y_{current}) \cup B \subseteq S$ **then**
12: putTail($\mathcal{S}e_{current}. \prod + i$)
13: **else**
14: **if** $Tube_i(e_{current}.Y_{current}) \cap B \neq \emptyset$ **or** $Tube_i(e_{current}.Y_{current}) \nsubseteq S$ **then**
15: discard $e_{current}$
16: **if** $Tube_i(e_{current}.Y_{current}) \cap B = \emptyset$ **and** $Tube_i(e_{current}.Y_{current}) \subseteq S$ **then**
17: **if** $Length(\prod) + 1 < K$ **then**
18: putTail($\mathcal{L}, (e_{current}.Y_{init}.Post_i(e_{current}.Y_{current}), e_{current}. \prod + i)$)
19: **return** $<, False >$ ▷ if no solution is found, or $< \prod, True >$, $\prod$ beaing any pattern validated in *Solution*.

3.3 Near optimal Switched Controller synthesis

Definition 1.(Stochastic Hybrid Game). A stochastic hybrid game $\mathcal{G}$ is a tuple $(\mathcal{C},\mathcal{U},\mathcal{X},\mathcal{F},\delta)$ where:

1. $\mathcal{C}$ is a controller with a finite set of (controllable) modes C.
2. $\mathcal{U}$ is the environment with a finite set of (uncontrollable) modes U.
3. $\mathcal{X} = \{x_1, x_2, ..x_n\}$ is a finite set of continuous (real-valued) variables
4. for each $c \in C$ and $u \in U, \mathcal{F}_{c,u} : \mathbb{R}_{>0} \times \mathbb{R}^X \to \mathbb{R}^X$ is the flow-function that describes the evolution of the continuous variables over time in the combined mode (c,u), and
5. δ is a family of density functions, $\delta_\gamma : \mathbb{R}_{\geqslant 0} \times U \to \mathbb{R}_{\geqslant 0}$, where $\gamma = (c, u, v) \in C \times U \times \mathbb{R}^X$. more precisely, $\delta_\gamma(\tau, u')$ is the density that $\mathcal{U}$ in the global configuration $\gamma = (c, u, v)$ will change to the uncontrollable mode u' after a delay of τ [1].

We shall assume that among the continuous variables $\mathcal{X}$, there is a variable **time** measuring global time, i.e $\mathcal{F}_{c,u}(\tau, v)(\mathbf{time})$ for any mode-configuration (c, u). In the above definition, the syntatic specification of flow functions—e.g. using ODEs—has been left open. In the game $\mathcal{G}$, the controller $\mathbb{C}$ will only be permitted to change controllable mode at time-points being a multiple of some given period P (hence the term switched control). In contrast, the environment $\mathbb{U}$ will change its uncontrollable mode according to the family of density function δ_γ.

Example. we use a one-room heating control problem as a running example to demonstrate our techniques in a simple setting: we model the problem, explain the necessary theory behind the model, show how the model fits the theory and show how Uppaal Stratego can be used to solve the problem.

The one-room system consists of room with walls, a window, heater and its controller. The objective of the controller is to mantain the room temperature at the goal level(21C°). Due to temperature sensor energy considerations the controller receives temperatura readings only once every 15 minutes and then it has two options either to turn the heater on(mode "HeatOn") and keep it there or switch the heater off (mode "HeatOff"). Consequently the temperature evolution will be different in these modes due to different energy supply from the heater. There is also a continuous leak of energy through the walls and the window to the ourside environment. In short, the temperature dynamics can be described by the following differential equation:

[1] Note that $\sum_u, \int_\tau \delta_{c,u,v}(\tau, u')d\tau = 1$ for all (c, u, v).

$$\frac{d}{dt}T(t) = (T_e(t) - T(t))A(t) + H(t) \tag{3.3}$$

where $T(t)$ is the room temperature at time $t, T_e(t)$ is the outdoor temperature, $A(t)$ is the heat exchange factor specific to the walls and windows, and $H(t)$ is the power of the heater.

Fig1b. shows such differential equation with heater step functions modelled in UPPAAL STRATEGO as hybrid automaton with two discrete modes. The continuous dynamics of $T(t)$ is typeset as an invariant constraint over the clock variable T derivative under the respective modes. The periodic behaviour of the controller is enforced by the invariant $x <= p$ and guard $x == p$ over clock x with default derivative of 1. For the sake of simplicity, we assume static outdoor temperature fixed to a specific value and modelled by the constant floating point variable T_e. All model variables (their types and initial values) are declared as C structures in **Fig1.a**. The window step function A(t) is modelled in **Fig. 1c** as stochastic automaton with transitions between "Open" and "Closed" modes and changing the floating point variable A. Thus the window process can change the value of A discretely distribution over time, but only at intervals specified by a user profile. The profile is stored in arrays **closedL/U** and textbfclosedL/U denoting the lower and upper bounds of time intervals when the switch may happen. For example, one can read the profile arrays by columns: the window starts and stays closed during the night time, but it will open somewhere between 6 and 7 o'clock in the morning and close between 7 and 8 o'clock, then it will open again between 11 and 12, and close between 12 and 13, etc.

The whole system model is then a composition of the controlleed heating process with the stochastic window process where temperature depends on the heating mode and the mode of the window.

Example 1. In our one-room example, the controllable modes are HeatOff and HeatOn with controllable transitions (using solid lines) between them, the uncontrollable are Open and Closed with uncontrollable transitions (using dashed lines). We also have a number of continuous variables: temperature T and clocks t,x and w. The differential equations together with discretely changing variables are part of the flow-functions definition.

Now let $\mathbb{C}$ denote the set of global configurations $C \times U \times \mathcal{R}^X$ of the game $\mathbb{G}$. Then a (memoryless) strategy σ for the controller $\mathcal{C}$ is a function $\sigma : \mathbb{C} \to \mathcal{C}$, i.e. given the current configuration $\gamma = (c, u, v)$, the epression $\sigma(\gamma)$ is the controllable mode to be used in the next period.

Let $\gamma = (c, u, v)$ and $\gamma' = (c', u', v')$. We write $\gamma \xrightarrow{\tau} \gamma'$ in case $c' = c, v' = \mathcal{F}_{(c,u)(\tau,v)}$ and $\delta_\gamma(\tau, u') > 0$. Let $\sigma : \mathbb{C} \to C$ be a (memoryless) strategy. Consider an interleaved sequence π of configurations and relative time-delays of the form:

$$\pi = \gamma_0 :: \tau_1 :: \gamma_1 :: \tau_2 :: \lambda_2 :: \tau_3 :: \gamma_3 ...$$

where $\gamma_i = (c_i, u_i, v_i), \tau_i \in \mathbb{R}_{\geqslant 0}$ and for all n there exist i st. $\sum_{j \geqslant i} \tau_j = nP$. Then π is a run according to the strategy σ if for all i either $\gamma_i \xrightarrow{\tau_{i+1}} u\gamma_{i+1}$ or $\sum_{j \geqslant i+1} \tau_j$ is multiple of P and $\gamma_i \xrightarrow{\tau_{i+1}} (c_i, u_i, v_{i+1})$ with $c_{i+1} = \sigma(c_i, u_i, v_{i+1})$ and $u_{i+1} = u_i$.

In fact, under a given strategy σ of the game $\mathcal{G}$ becomes a completly stochastic process $\mathcal{G}|\sigma$, inducing a probability measure on sets of runs. Thus, if $H \in \mathbb{N}$ is a given time-horizon, and D is a random variable on runs–e.g. measuring the integrated deviation of the continuous variables wrt. given target values– then $\mathbb{E}^{\mathcal{G},\gamma}_{\sigma,H}(D) \in \mathbb{R}_{\geqslant 0}$ is the expected value of D with respect to random runs of $\mathbb{G}|\sigma$ of length H starting in the configuration γ. We want to obtain a strategy σ^H which minimizes (or maximizes) this expected value.

Example 2. The one-room controller's goal is to keep the room temperature as close as possible to the goal set point, therefore a desired controller would minimize the absolute difference $T(t) - T_g$. In order to encourage the minimization even more we use a quadratic difference function to measure the distance between the room T and the goal T_g temparatures, and then integrate it to achieve a distance function over complete trajectories. Conveniently, our distance function is modelled using differential equation in *fig* as a separate process. Before we synthetize anything, we can inspect how does a uniform random choice fare in *fig*: the temparature curve is at the top and heating and window mode trajectories are bellow and they jump up when the heating is on the window is open respectively. The result is that the room temperature responds to the mode changes and varies widely, tending to overshoot above the goal, and hence the distance function after $24h$ period is about 4200 on average. In order to synthetize a strategy we pose the following query in UPPAAL STRATEGO:

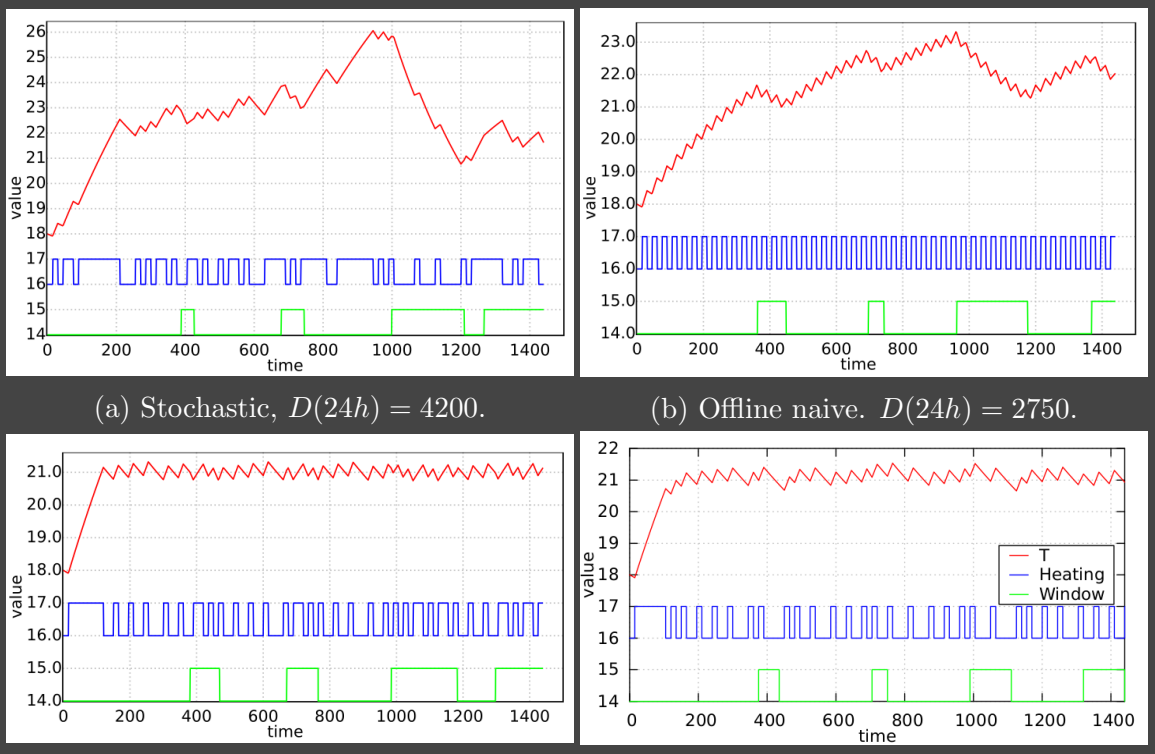
```
strategy opt = minE (D) [<=24h]: <> t==24*h
```

which asks to find the strategy that will minimize the expected value of D when we reach a state with $t == 24 * h$ while considering simulations of up to $24 * h$ in duration. Once the strategy is available, we may inspect it by requesting a simulation plot:

```
simulate 1 [<=24*h]{T, Window.Open+14, Room.HeatOn+16} under opt
```

For example, the synthetized 24h strategy using the ”naive” learning method yields the distance od 2750 on average as shown in *Fig.* The result is even more improved by the ”splitting” learning method in *Fig* where the temperature oscillates around the goal very closely.

Uppaal Stratego offers four learning methods focusing on various parts of the model, therefore we consider the quality and the cost of each method before we focus on our industrial-scale example. **Table** shows a summary of the evaluation of various methods on two variants of a one-room example: the purely dynamical model is show in **Fig1** and another one that has an extra counter incremented at each period P. The result is that among that offline methods (discussed so far) the ”splitting” method provides the smallest distance solution, however it is costlier than others in **CPU** time and memory. The right side **Table** shows that if we add a period counter to our model, then other methods dominate and the ”splitting” method is no longer as good and the ”naive” computation costs significantly less. Offline-6 section (strategy for six days) requires twice as many resources as offline-3 (strategy for three days) which means that a linear number of resources is needed in terms of duration of the strategy while using the same number of runs, but the quality (distance) degraded almost four times with a period counter.



(a) Stochastic, $D(24h) = 4200$.

(b) Offline naive. $D(24h) = 2750$.

(c) Offline splitting, $D(24h) = 468$.

(d) Online naive, $D(24h) = 471.5$.

Figure 3.3: One-room 24h trajectories of various control strategies

Online Synthesis Uppaal Stratego [5,6] provides a method for aproximating $\mathbb{E}^{\mathcal{G},\gamma}_{\sigma,H}(D) \in \mathbb{R}_{\geqslant 0}$ by computing a near-optimal strategy σ^H for a given horizon H using reinforcemente learning. However, the effort needed to learn the strategy σ^H with a desired precision and confidence-level grows exponentially in the number of dimensions (variables). The quality of the learned control degrades sharply after the control options outnumber the number of simulation runs during learning, making this direct application of Uppaal Stratego limited in the time horizon. For instance, given a realistic setting of eleven heating switches as considered in our case study, the controller is faced with $2^11 = 2048$ distinct options at each 15min period and thus Uppaal Stratego manages to compute sensible heating configuration only for the first two periods (yieldings $2048^2 = 4194304$ combinations in total) and then it simply resolves to default option of no heating at all. Instead of learning — at great computational expense – the entire strategy σ^H, we propose a method for attentively and online (i.e while playing the game G in the real setting) to compute a near-optimal strategy for controllable mode-change at the next period. More precisely, the resulting online and periodic strategy σ^O will base the mode-change at time nP not only on the configuration at that point

(γ_n) but also on the configuration (γ_{n-1}) at time $(n-1)P$[2], which will be used as the basis for online learning of short-horizon ($h << H$) strategies.

Table 3.1: Performance evaluation of one room controller synthesis: offline-3(-6) methods synthetize strategy for entire 72 hours (144 hours respectively) at once, strategy distance is evaluated on 70 simulations; online-3 methods synthetize a strategy for 5 periods of 15 min ahead and repeat synthesis and execution until 72 hours are covered, the distance is averaged over 70 online simulations.

	Synthesis method	**Purely dynamical model**			**Extra period counter**		
		Distance	**cpu,s**	**mem,kB**	**Distance**	**cpu,s**	**mem,kB**
Offline-3	naive	10227.8	1555.15	11884	3671.84	566.4	9448
	splitting	517.9	1640.06	13424	2361.80	1608.48	90740
	covariance	10227.8	1298.66	11896	1091.81	1668.45	22820
	regression	10227.8	1368.34	11480	1387.84	1767.50	19196
Offline-6	naive	19668.8	1855.36	11836	8032.86	1316.08	20820
	splitting	593.7	3200.38	13112	8260.19	3120.03	167308
	covariance	20234.3	2039.30	11528	2468.91	3258.09	39580
	regression	19007.2	2525.13	12148	3425.62	3488.26	28416
Online-3	naive	584.7±1.0	1046.5±5.0	7240	526.6±0.5	1227.1±3.2	7328
	splitting	547.7±0.6	1136.4±3.6	7384	526.1±0.6	1240.8±2.5	7384
	covariance	587.5±1.2	1084.0±3.9	7272	527.1±0.6	1158.5±2.5	7624
	regression	585.3±1.0	1173.9±3.4	9052	527.9±0.5	1337.1±2.5	7380

Formally:

$$\sigma^O(\gamma_{n-1}, \gamma_n) = \text{def } \mathbf{let}(\sigma^h = \mathbf{argmin}_\sigma \mathbb{E}^{\mathcal{G},\gamma_{n-1}}_{\sigma,h}(D)) \ \mathbf{in} \ \sigma^h(\gamma_n)$$

We leave the formal definition of runs under the one-step-memory strategy σ^O to the reader (slightly more complicated version of runs under a memoryless strategy given above). However, we note that σ^O may be used for an arbitrary finite horizon H or even as a strategy of infinite horizon. To maximize the quality of σ^O, the choice of the small horizon h should be such that it just allows the learning of σ^h to be completed between the two configurations γ_{n-1} and γ_n, i.e. within the period P.

[2]Note that there way be several configurations between γ_{n-1} and γ_n due to the environment $\mathcal{U}$ changing the uncontrollable mode.

3.4 Safe and near optimal controller for sthocastic hybrid games

Model Predictive Control In principle, model predictive control is very effective in discrete control. Now consider the issue of critical systems where is mandatory a safe evolve along the control. This suggest to incorporate a sequence of modes that guarantee a safe system. For this reason the study incorporat e a methodology to find patterns for non linear hybrid systems [ref adrien].

Chapter 4

Safe and Near Optimal Controller Synthesis for Stochastic hybrid Games

In this part, we propose a case study *Sthocastic Hybrid Solar Water Heating*, to apply our methodology, following the next procedure: *System modelling* using physics criteria Tsilingiris (1996). The use of mechanical concepts to have a *stochastic hybrid game*. Some engineering assumptions either int the model and the configuration parameters are considered to facilate our analysis. A fundamental characteristic of the model is the incorporation of uncontrollable events for instance open the aperture valve or close it. Secondly, our system interact with an environment, for this reason we set some paremeters to get the environment conditions so as understand the temporal data such as *Irradiance*, *external and internal temperatures*, these perturbations are preprocessed to interact with our simulator Section 4.1. To establish a safe sthocastic hybrid system, we start to compute patterns as Section 3.2 explains. It is important to bear in mind the uncontrollable events in our patterns computation to verify the safety in our system. Lastly, the computation of a near optimal controller consist of an exhaustive exploration of different strategies in order to either minimize or maximize a cost function of a random variable in a specific horizon and stochastic hybrid game, this computation is using UPPAAL STRATEGO. In control experiments we found that This methodology help to concatenate the two approaches implementing a toolbox for real time simulation in PYTHON Larsen et al. (2016).

4.1 Solar Water Heating Case study

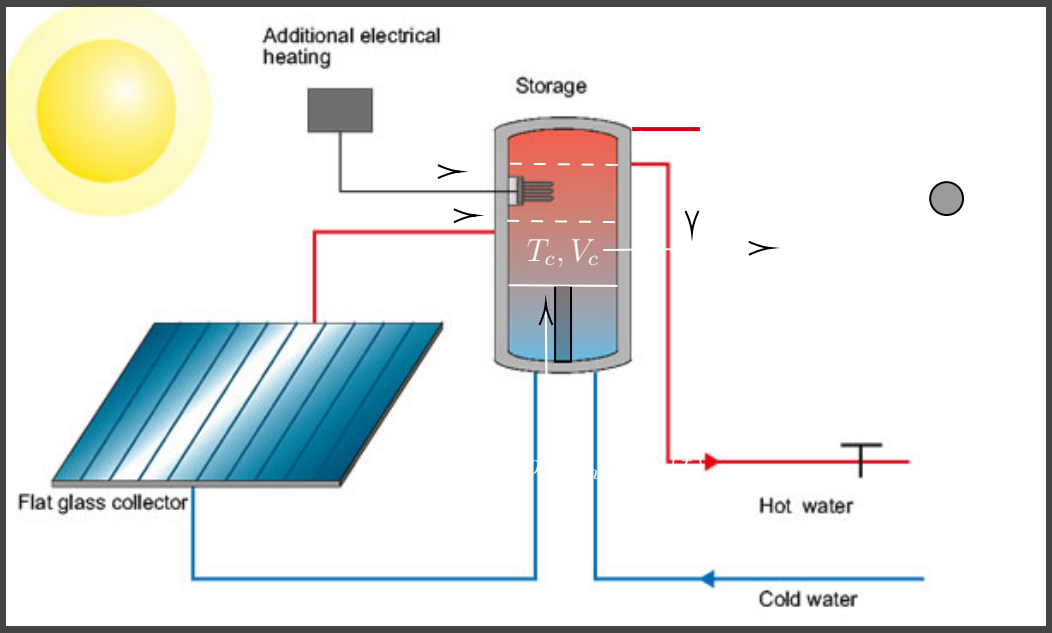


Figure 4.1: Solar Water Heating diagram

4.1.1 data analysis

Stochastic events as historical data. In practice we notice unpredictable discrete events which can perturbate a dynamical system, these stochastic discrete perturbations we call uncontrallable actions and is managed using a stochastic approach and machine learning to optimize the behaviour. Generate this actions with a realist criteria for our case study we choose three gaussian distributions with $\mu_1 = 7h$, $\mu_2 = 12h$ and $\mu_3 = 19h$ and $\sigma_1 = 12, \sigma_2 = 15, \sigma_3 = 10$ respectively. Finally we add a uniform distribution around the day.

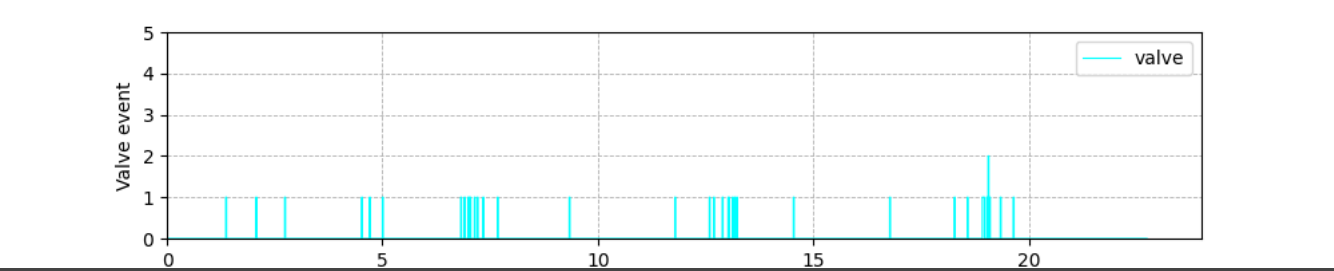


Figure 4.2: State changes of valves along the day.

Solaris Data and preprocessing The data collected is from spain solaris, which consist of a set of environment features such as environment temperature and irradiance that would be useful for the simulation. However, we preprocess the data for our purpose such us modify the time step from 15 minutes to 1 minute trough interpolation.

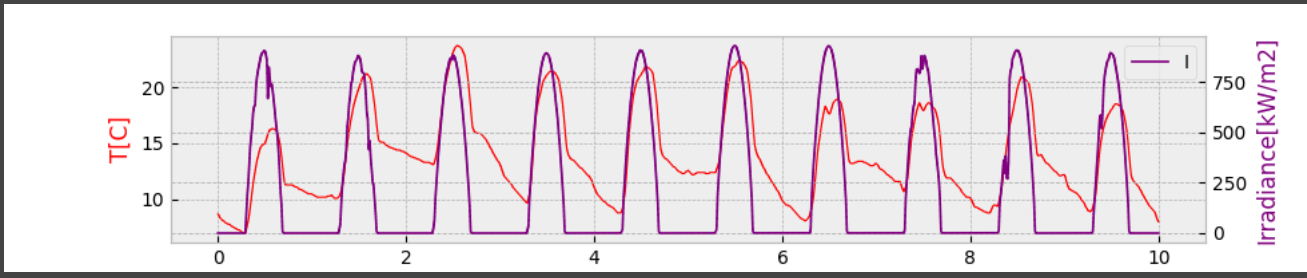


Figure 4.3: Weather time series, irradiance and temperature

Prediction Data. To integrate a model predictive controller approach we had to implement a forecaster in real time in order to get a sequence of environment data on future which will be used to control future states an get the correct patterns on present. ARIMA state model is used with a realtime pivot .. bla bla .

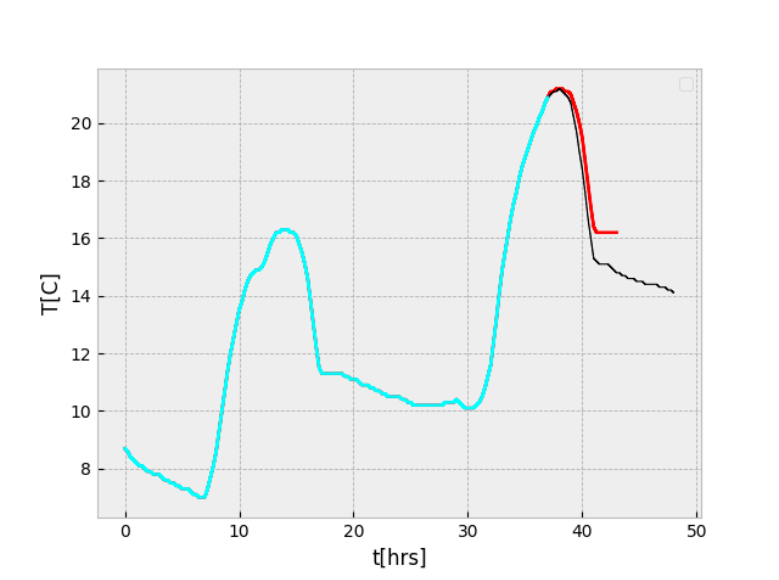


Figure 4.4: An example of prediction using ARIMA forecaster

States			
Variable	Description	Initial Value	Units
E	Energy consumption	0.0	J
V	Container Volumen	0.13	m^3
T	Container Temperature	50.0	°C
Constants			
Variable	Description	Value	Units
C_e	The factor heat of the water in	4186	$J\,°C^{-1}\,kg^{-1}$
$\dot{m}$	Mass flow rate input/output	0.1	$kg\,s^{-1}$
M_c	the mass of the container in	100	kg
A_c	Area of colector in	1	m^2
A_t	Area of total of surface in	5.56	m^2
k_c	Conduction coeficient	16	$W\,m^{-1}\,°C^{-1}$
P_{aux}	Auxiliary heat power in	1000	W
Input values			
Variable	Description	Space values	
v	Controllable action to release water	$\{0, 1\}$	
r	Controllable action to heat	$\{0, 1\}$	
p	Uncontrollable action to change capacity water	$\{1, 2, 3\}$	
Disturbances			
Variable	Description	Range	Units
$T_{in}(t)$	the temperature of water input/output in	$[20 - 25]$	°C
$I_{env}(t)$	the irradiance in	$[0 - 920]$	$W\,m^{-2}$
$T_{env}(t)$	the outside temperature in	$[0 - 16.5]$	°C

Table 4.1: Parameter values are used to compute c_1, c_2, c_3 and c_4.

4.1.2 Hybrid Solar Water Heating as a Sthocastic Hybrid Game

. The hybrid solar water heating scenario with 12 modes of operations is defined like this: $\mathcal{G}_{n,m} = (\mathcal{C}, \mathcal{U}, \mathcal{X}, \mathcal{F}, \delta)$, where the controller $\mathcal{C}$ has a finite set of controllable modes, given by resistance state $r \in \mathbb{B}$ and piston position $p \in \{1, 2, 3\}$. The environment $\mathcal{U}$ has a finite set of uncontrollable modes $v \in \mathbb{B}$, that means the valve state for opening/closing water aperture. We assume that $\mathcal{U}$ given δ can switch among modes with equal probability at every period. The state variables in $\mathcal{X}$ are given by $\{T, E, V\}$, container temperature, energy used and container volumen respectively.

Given the container temparature and volume, a controllable modes $r \in \mathbb{B}$ and $p \in \{1, 2, 3\}$ and uncontrollable mode $v \in \mathbb{B}$ and a time delay τ Figure 4.1.

$$\begin{aligned}\frac{d}{dt}T(t) = \frac{1}{V(t)}[&-c_1(T(t) - T_{env}(t)) - vc_2(T(t) - T_{in}(t)) \\ &-fc_3(pV_{min} - V(t))(T(t) - T_{in}(t)) + c_4 I_{env}(t) + rc_5]\end{aligned} \tag{4.1}$$

$$\frac{d}{dt}V(t) = pV_{min} - V(t); \tag{4.2}$$

$$\frac{d}{dt}E_{used} = rc_3; \tag{4.3}$$

The equation 4.1 has some constants $\{c_1, c_2, c_3, c_4\}$ this parameters are computed with the parameteres defines in table 4.1 and equal to $\{2.44e^{-5}, 4.77e^{-6}, 0.0024, 0.01\}$ respectively. Another important variable to consider is f, which means the transition between event p and p' which create a term that affect the total energy Equation 4.1, this term measure the changes of energy when the piston is in expansion thus, the water entrance and the container temperature decrement so as the increment of the volume, or contraction which cause water dispense the loss of total energy decreasing the volumen but not the container temperature. $p < p' \rightarrow f' == 1$, which means a piston expansion otherwise a piston contraction $p \geqslant p' \rightarrow f' == 0$, in initial conditions we set $f == 0$.

According to our case study we find some transisions such as $\gamma_2 \xrightarrow{\tau} \gamma_0$, $\gamma_3 \xrightarrow{\tau} \gamma_0$, $\gamma_2 \xrightarrow{\tau} \gamma_1$, $\gamma_3 \xrightarrow{\tau} \gamma_0$ so as $\gamma_6 \xrightarrow{\tau} \gamma_3$, $\gamma_7 \xrightarrow{\tau} \gamma_3$, $\gamma_6 \xrightarrow{\tau} \gamma_5$, $\gamma_7 \xrightarrow{\tau} \gamma_5$

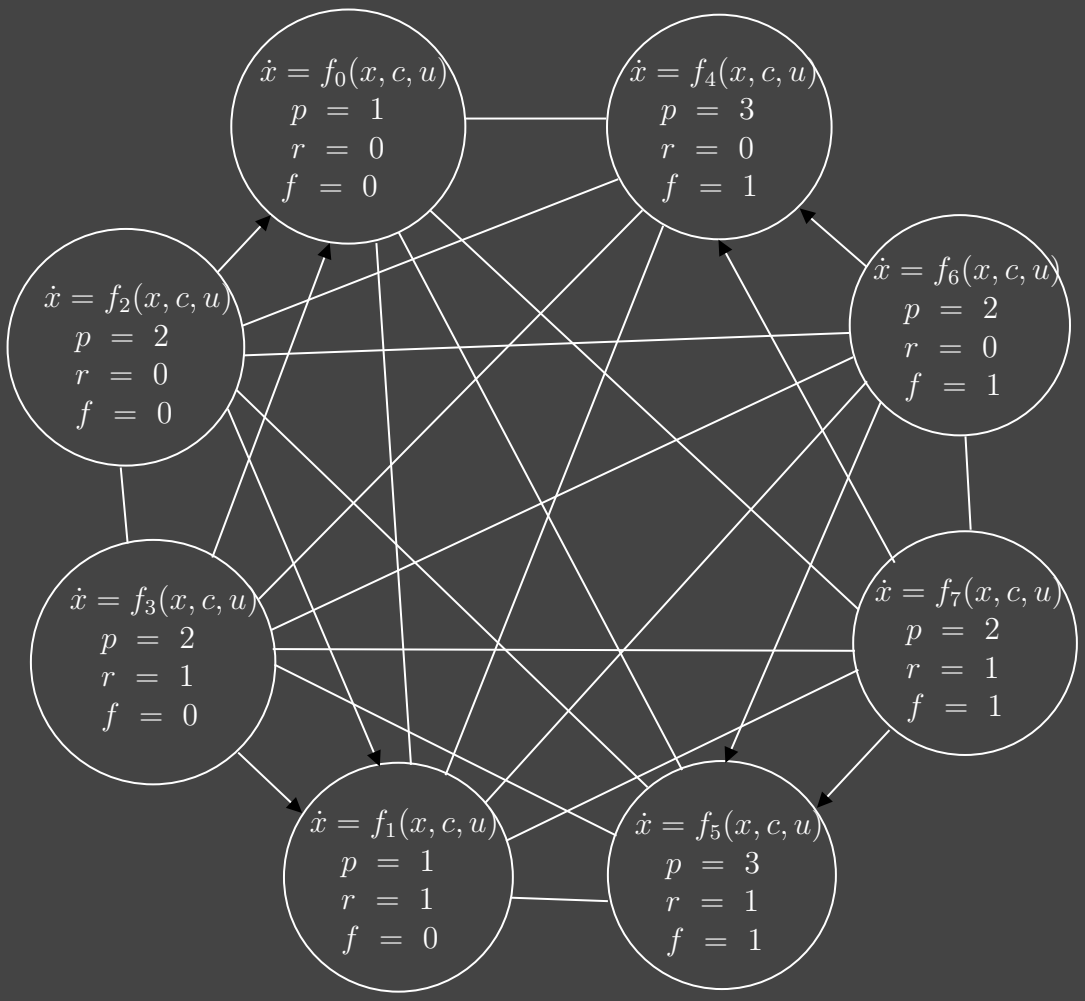


Figure 4.5: Automaton with forbidden transitions because of the addition of a discrete variable f which facilate the system modelling in cases such as piston expansion and compression.

Finally we set as controllable actions $c = (p, r, f) \in \{1, 2, 3\} \times \{0, 1\} \times \{0, 1\}$ and uncontrollable actions $u = \sum_{i=0}^{3} p_i(t)$, $p_1(t)$ as shown at Figure 4.2.

4.2 Experiments:

4.2.1 Patterns computation

To establish a safe sthocastic hybrid system, we start to compute patterns as Section 3.2 explains. It is important to bear in mind that the stochastic events shall not be considered as discrete state but as interval, in other words the sthocastic action will be composed by lower case when valve is close doing the loss energy as zero and the higher case when valve is open doing loss energy due to water dispense. Safe patterns are computed and associated to zonotopes,which are regions where patterns verify the next: Given $D = 20, K = 3$ and by constructing a law $\sigma(.)$, such that for all $x_o \in R$, and under the unknown bounded perturbation d, there exists $\pi = \sigma(.) \in U^k$ for some k such that:

$$\begin{gathered} x(t_0 + k\tau; t_0, x_0, d, \pi) \in R \\ \forall t \in [t_0, t_0 + k\tau], x(t; t_0, x_0, d, \pi) \in S \\ \forall t \in [t_0, t_0 + k\tau], x(t; t_0, x_0, d, \pi) \notin B \end{gathered}$$

Such a law permits to perform a safe control guaranteeing limits for each action in each step conditioning a safe sequence of modes called *pattern*. We configurate $R = \begin{bmatrix} 40 & 70 \\ 0.1 & 0.2 \end{bmatrix}$, $S = \begin{bmatrix} 30 & 80 \\ 0.09 & 0.31 \end{bmatrix}$, $W = \begin{bmatrix} 40 & 70 \\ 0.1 & 0.2 \end{bmatrix}$, $B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$, after to run the

Proposition 1. *Algorithm Decomposition with input (R,R,S,B,D,K) returns, when it successfully terminates, a decomposition $\{V_i, \pi_i\}_{i \in I}$ of R which solves Problem 1.*

There are forbidden transitions in the automaton, for this reason we modify the algorithm which only permit to evaluate allowed transitions, otherwise the algorihtm ignore the mode exploring other modes to complete the pattern.

Algorithm 3 Find Pattern with forbidden transitions

1: **procedure** FINDPATTERN(W, R, S, B, K) ▷ List of set of patterns
2: **Input:** A box W, a box R, a box S, a box B, a length K of input pattern
3: **Output:** $< \prod, True >$ *or* $<, False >$
4: $\mathcal{S} = \{\emptyset\}$
5: $\mathcal{L} = \{(W, W, \emptyset)\}$
6: **while** $\mathcal{L} \neq \emptyset$ **do**
7: $e_{current} = takeHead(\mathcal{L})$
8: **for** $i \in U$ **do**:
9: **if** $Pair(l, i) = False$ **then** ▷ l is the current mode
10: **continue** ▷ Not considered transition l, i
11: **if** $Post_i(e_{current}.Y_{current}) \subseteq R$ **and** $Tube_i(e_{current}.Y_{current}) \cup B = \emptyset$ **and** $Tube_i(e_{current}.Y_{current}) \cup B \subseteq S$ **then**
12: putTail($\mathcal{S}e_{current}.\prod + i$)
13: **else**
14: **if** $Tube_i(e_{current}.Y_{current}) \cap B \neq \emptyset$ **or** $Tube_i(e_{current}.Y_{current}) \nsubseteq S$ **then**
15: discard $e_{current}$
16: **if** $Tube_i(e_{current}.Y_{current}) \cap B = \emptyset$ **and** $Tube_i(e_{current}.Y_{current}) \subseteq S$ **then**
17: **if** $Length(\prod) + 1 < K$ **then**
18: putTail($\mathcal{L}, (e_{current}.Y_{init}.Post_i(e_{current}.Y_{current}), e_{current}.\prod + i)$)
19: **return** $<, False >$ ▷ if no solution is found, or $< \prod, True >$, $\prod$ beaing any pattern validated in $Solution$.

The algorithm find pattern second version on Le Coënt et al. (2017) is improved filtering invalid transition in the line 10.

PYTHON SIMULATOR. We implement a *toolbox* which support sthocastic hybrid models $\mathcal{G}_{n,m}$ and controllers $\mathcal{C}$. The models works with seasonal data disturbances for this reason we incorporate modules to preprocess the data. **Simulation** In the initial step of the Model predictive algorithm to control, we use the first pattern corresponding to the current zonotope. In the subsequent iterations e use the STRATEGO to find a near optimal strategy from a selected set of safe patterns as show 4.2.2. It is very important to bear in mind the way of model predictive controler operate which is an asynchrnous call before finish the last mode in the

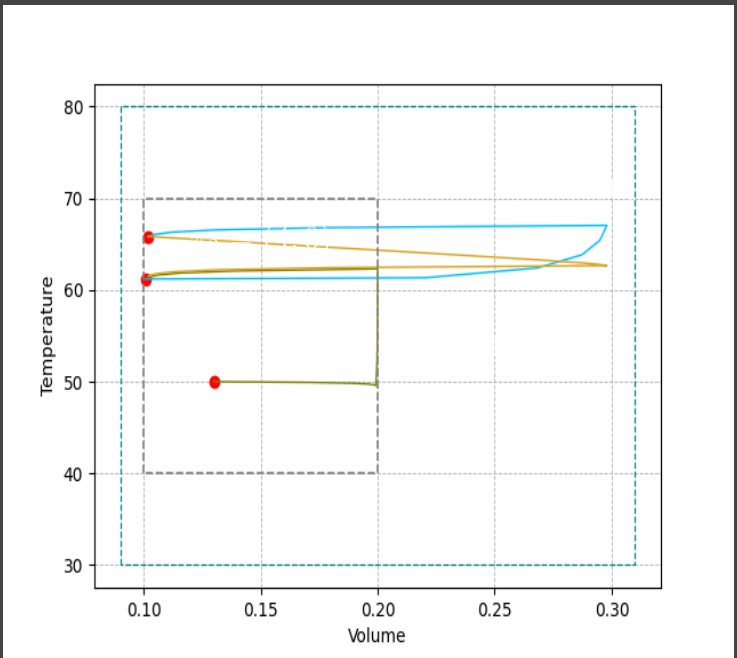


Figure 4.6: Simulation for the firsts three patterns

current pattern, sending to STRATEGO the current dynamical states such as *mode, volume, temperature* and so on with the corresponding environment predictions 4.4. After the near optimal pattern computation for stochastic hybrid game by STRATEGO the process repeat.

UPPAAL STRATEGO.

Learning strategies for PTMPDs The algorithm used to learn will generate an approximation of the optimal strategy. The algorithm has five main phases, and one optional. All these can be seen in FIG . We will now shortly each of these steps. In the following sections.

check learning optimal schedullign for time uncertain cet **Simuation Filtering Learning Determinization Evaluation**

uncontrollable action si defined in a interval [min, max] at the moment to run the find pattern.

add this part add the pseudocode about the methodoloy pattern and near optimal.

add the uppaal template in this part add a figure with the template and describe it.

4.2.2 Near optimal Experiments

Regarding our experiments, we have two major components: a simulator written in PYTHON and a number of controllers, including the ones produced by UPPAAL STRATEGO. The simulator implements a solar water heating as stochastic hybrid game $\mathcal{G}_{n,m}$. For our experiments, in the simulator we fix a time horizon H of 75 minutes with a period P of 5 minutes. As in the real house, every 5 minutes, the simulator outputs the current container values such as temperature T, volume V, energy E, mode and predicted environment data Subsequently, the controller inputs the control actions such as resistor r, piston p which are used by the simulator for the next 5 minutes. The house has a desired temperature T^g and alpha parameters which denotes the importance to minimize either target or energy consumption. Our goal is to optimize the comfort for the water consumption reducing the energy consumption. To define this cost function subject to controller (strategy) σ and $\mathcal{G}_{n,m}$ of the form $\pi = \gamma_0 \xrightarrow{t_1} \gamma_1 \xrightarrow{t_2} ... \xrightarrow{t_{k-1}} \gamma_{k-1} \xrightarrow{t_k} \gamma_k$ where $k = H/P$ is the number of control steps in the run π. Let $T_i(\gamma_j)$ denote the container temperature T_i at configuration γ_j. Then the cost function is defined by

$$cost(\pi) = \alpha(E) + (1 - \alpha)(T - T_g) \tag{4.4}$$

In our experiments, we evaluate a number of different controllers. The simulator uses the cost function to compare the different controllers.

Controllers. In the following we introduce a number of controllers which we use in out experiments. We present the current controller operating in the house, two controllers proposed by engineers and the controllers synthetized using online synthesis and UPPAAL-STRATEGO.

Safe Greddy Controller with Patterns. The greddy controller explore pattern by pattern and is chosen those which minimize the cost function, due to the use of complete patterns it guarantee a safe behaviour and is optimize without consider the uncontrollable action valve thus it could be optimize even more from a sthocastic approach.

Stategо Online Controller. The Figure environment "floats" to find an optimal position on the page . When you begin a figure, you can use the options [htbp!] to specify whether the figure should go "here", the "top" of this page, the "bottom" of this page, or a special "page" reserved for floats. LaTeX, in its infinite wisdom, will sometimes prevent you from putting a figure in an "ugly" place, but you can (sort of) override its decision using the "!" option. When you haven't used

any option, the environment assumes that you provided [tbp], and it's choosing to put the figure at the top of the page, above where your section goes.

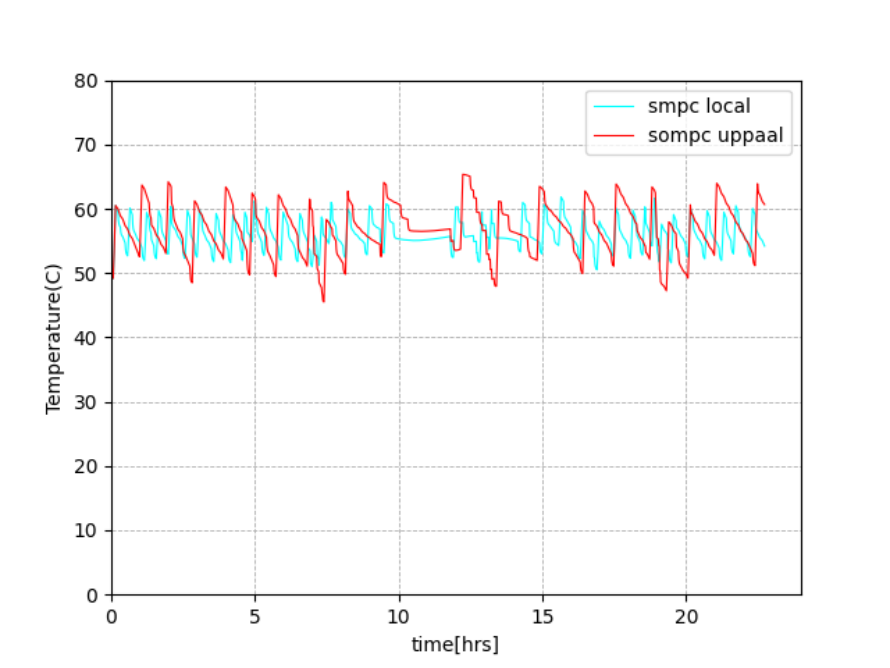


Figure 4.7: Container temparature comparison between greedy controller with uppaal stratego controller, both approaches are safe.

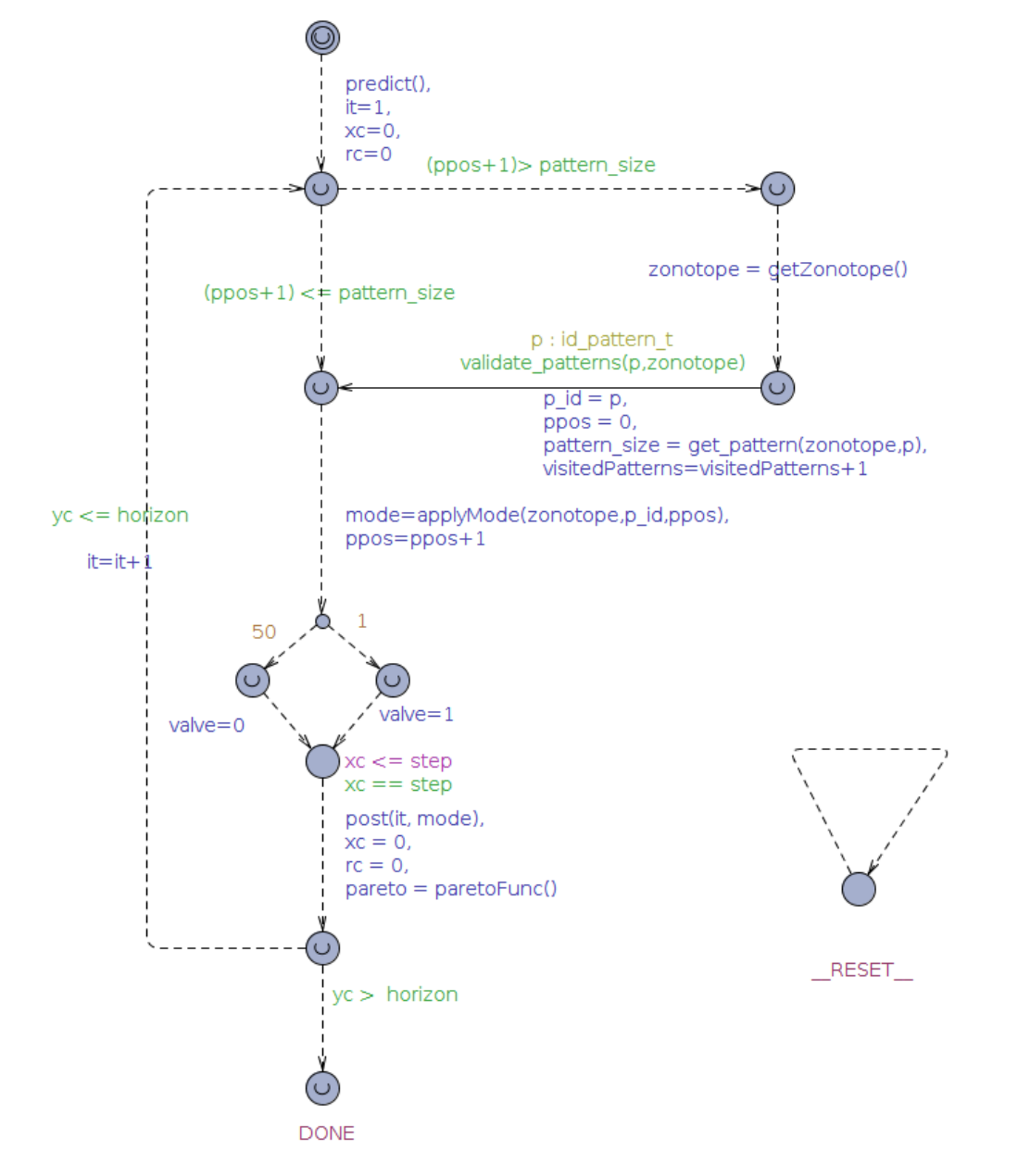


Figure 4.8: Template of the safe an near optimal algorithm

We can observe an algorithm that run on STRATEGO, this template is adapted to find near optimal strategies for stochastic hybrid games, some heuristics are running for an exhaustive stochastic exploration and reinforcement learning is used to

Safe and Near Optimal Controller.

Evaluation Scenarios

Controller Evaluation

Chapter 5

Discussion and Conclusions

In the *Hybrid Solar Water Heating* case study we evaluated the existing UPPAAL STRATEGO controller synthesis subject to certain operation modes called patterns wich ensure a safe system behaviour, this methodology helps enginners not only guarantee a safe controller but also some strategies which optimize trough reinforcement learning given by UPPAAL STRATEGO. The proposal was successfully concatenated finding safe and near optimal control strategies for our controller.

5.1 Limitations and Future Work

To reduce the complexity to compute patterns on way to face problems related dynamic setpoint configurations would be to have more than one R regions.

Solar car The latest advances in solar machines are cars, this system interact with disturbances which is irradiances, the roads to stop the car lose energy. This would work with spatial trajectories as spatio temporal series of data.

Bibliography

Branicky, M. S. (1994). Stability of switched and hybrid systems. In *Proceedings of 1994 33rd IEEE Conference on Decision and Control*, volume 4, pages 3498–3503. IEEE.

Larsen, K. G., Mikučionis, M., et al. (2016). Online and compositional learning of controllers with application to floor heating. In *International Conference on Tools and Algorithms for the Construction and Analysis of Systems*, pages 244–259. Springer.

Le Coënt, A., dit Sandretto, J. A., et al. (2017). An improved algorithm for the control synthesis of nonlinear sampled switched systems. *Formal Methods in System Design*, pages 1–21.

Le Coënt, A., Fribourg, L., et al. (2016). Distributed synthesis of state-dependent switching control. In *International Workshop on Reachability Problems*, pages 119–133. Springer.

Liberzon, D. (2003). *Switching in systems and control*. Springer Science & Business Media.

Tsilingiris, P. (1996). Solar water-heating design—a new simplified dynamic approach. *Solar Energy*, 57(1):19–28.