Enforcing safety of cyberphysical systems using flatness and abstraction

Citation: Colombo, Alessandro, and Domitilla Del Vecchio. "Enforcing Safety of Cyberphysical Systems Using Flatness and Abstraction." ACM SIGBED Review 8, no. 2 (June 1, 2011): 11–14, New York, NY, USA, Association for Computing Machinery (ACM), 2011.

As Published: http://dx.doi.org/10.1145/2000367.2000369

Publisher: Association for Computing Machinery (ACM)

Persistent URL: http://hdl.handle.net/1721.1/119163

Version: Author's final manuscript: final author's manuscript post peer review, without publisher's formatting or copy editing

Enforcing safety of cyberphysical systems using flatness and abstraction *

Alessandro Colombo
Department of Mechanical Engineering, MIT
77 Massachusetts Avenue, Cambridge, USA
acolombo@mit.edu

Domitilla Del Vecchio
Department of Mechanical Engineering, MIT
77 Massachusetts Avenue, Cambridge, USA
ddv@mit.edu

ABSTRACT

The diffusion of cyberphysical systems acting in human-populated environments brings to the fore the problem of implementing provably safe control laws, to avoid potentially dangerous collisions between moving parts of the system, and with nearby obstacles, without compromising the system's functionality. The limiting factor in most implementations is the model's complexity, and a common workaround includes the reduction of the physical model, based on differential equations, to a finite symbolic model. Following this strategy, we are investigating ways to exploit the specific structure of many mechanical systems (the differentially flat systems) to achieve this simplification. Our objective is to construct a supervisor enforcing a given set of safety rules, while imposing as little constraints as possible on the system's functionality. In this paper, we outline our approach, and present an example – a collision avoidance algorithm for a fleet of vehicles converging to an intersection. Our approach improves on previous results by providing a deterministic symbolic model for a class of system, regardless of their stability properties, and by addressing explicitly the problem of safety enforcing.

1. INTRODUCTION

From industrial manipulators to automated transportation networks, cyber-physical systems are becoming an ever more common sight in many portions of modern society, be it in the industry, in the military, or in the civil context. As these systems become widespread, and their interaction with human beings grows tighter, it becomes important to develop efficient, provably safe control algorithms, that guarantee given specifications without being overly restrictive. Such control algorithms should guarantee that the behaviour of the system complies with given safety constraints, for example avoiding collisions between different moving parts or with obstacles, while ensuring that the system can complete its task. A common framework for the problem is the supervisory control of discrete event systems [7, 2], where the above requirements take the form of illegal states or sequences of states, and of nonblocking requirements on the controlled system. The advantage of casting the control problem in this framework lies in the relative simplicity of formally verifying discrete event systems, as opposed to dealing with geometric constraints on sets of differential equations. Additionally, the discrete event structure couples naturally with the digital systems that implement the controller.

One of the greatest challenges in this approach resides in finding an efficient way to map the physical, continuous-time system onto a discrete event system, without losing too much structure along the way. Typical solutions involve defining a space and time discretization of the continuous-time model, restricting space and input sets so that the discretization is finite, and then proving an equivalence relation between the discretized system and a suitable discrete event system. The discrete event equivalent is then called a *finite abstraction* of the continuous-time system. An interesting approach, that proceeds along these lines and applies to incrementally stable systems, has been proposed in [8, 9, 4], based on the discretization of the state space on a regular grid. The stability requirement has been relaxed in [11], however the obtained discrete event system is nondeterminsitic. A similar approach, based on the partition of the state space along tangent and transverse foliations to the flow, has been proposed in [1]. It applies to weakly integrable systems, proving their equivalence to nondeterministic discrete event systems. These approaches do not address explicitly the issue of safety enforcement.

We plan to exploit the specific structure of a general class of systems –the differentially flat systems– to construct a symbolic model of our system. Safety, the ability of the system to avoid entering a specified *bad set*, is addressed explicitly and ensured by construction. Furthermore, safety is ensured irrespective of the chosen discretization step, so that a coarse discretization can be chosen, for the purpose of reducing the computational complexity, without compromising the specifications. Additionally, we give conditions under which an upper bound on the distance of the allowed trajectories from the collision set can be computed.

In the text, the symbol $\|\cdot\|$ is the infinity norm of a vector or matrix, a subscripted index (e.g., x_i) indicates an element of a vector, and a superscripted index (e.g., x^i) indicates a vector out of a set of vectors.

2. PROBLEM STATEMENT

*This work was supported by the NSF Award # CNS 093081

.

We analyse a system of the form

$$\begin{aligned} \dot{x} &= f(x,a) \\ y &= h(x), \end{aligned} \tag{1}$$

with $x(t) \in X \subset \mathbb{R}^m$, $a(t) \in A \subset \mathbb{R}^n$, $y(t) \in Y \subset \mathbb{R}^n$. Functions f and h are C^k for sufficiently large k. The vector $x(t)$ is the state of the system, $a(t)$ is the input, and $y(t)$ is the output. We assume that (1) is differentially flat [3, 10, 6], with y as flat output. This means that (1) satisfies the following assumption:

ASSUMPTION 2.1. *Function h has rank n. Additionally, there exist two functions $\Gamma : (\mathbb{R}^n)^{q+1} \mapsto \mathbb{R}^m$ and $\Theta : (\mathbb{R}^n)^{q+2} \mapsto \mathbb{R}^n$ of rank m and n, respectively, in their domains, such that the integral curves of (1) identically satisfy the equations*

$$\begin{aligned} x &= \Gamma(y, \dot{y}, ..., y^{(q)}) \\ a &= \Theta(y, \dot{y}, ..., y^{(q+1)}). \end{aligned} \tag{2}$$

Here $y^{(i)}$ denotes the i-th derivative of y.

Call $\mathcal{A}$, $\mathcal{X}$, and $\mathcal{Y}$ the sets of all possible functions of time a, x, and y, respectively. We assume that A, X, and Y are compact sets, and that $\mathcal{A}$ is the space of piecewise polynomial functions of t that are polynomial in intervals $(k\tau, (k+1)\tau)$ for a given τ. We consider the two sets $\Omega \subset Y$, called the terminal set, and $B \subset Y$, called the bad set, and assume that B is the union of a finite number of convex polytopes. This is the setting, for example, of mechanical systems obeying Newton's law $\dot{x}^1 = x^2$, $\dot{x}^2 = f(x^1, x^2, a)$, where the flat output is $y = x^1$, and the bad set is a set of collision points in configuration space. We will analyse one such system in the example section.

We introduce the function $\phi_t(x(0), a) \in \mathbb{R}^m$, called the *flow* of system (1), such that $\phi_0(x(0), a) = (x(0))$ and x with $x(t) = \phi_t(x(0), a)$ for all t satisfies equation (1). We also introduce the symbol $\phi_{[t_1,t_2]}(x(0), a)$ to indicate a *trajectory* passing through $(x(0))$, that is, $\phi_{[t_1,t_2]}(x(0), a) := \bigcup_{t\in[t_1,t_2]} \phi_t(x(0), a)$. Finally we denote by $x([0, t])$ function x with domain restricted to $[0, t]$.

A function $\sigma : \mathcal{X} \times \mathbb{R} \mapsto 2^A$, that maps $x([0, t]) \in \mathcal{X}$ and $t \in \mathbb{R}$ to $\sigma(x([0, t]), t)$ is called a *supervisor* [7, 2] of (1). To specify the desired properties of the supervisor we need first to introduce the concept of ϵ-safe trajectory:

DEFINITION 2.1. *Consider the bad set $B \subset Y$, and let $b \in B$. Consider a trajectory $\phi_{[0,T]}(x(0), a)$ of (1), with $T \in \mathbb{R}_+$. The trajectory is ϵ-safe provided*

$$\inf_{t\in[0,T]} \inf_{b\in B} \|h(\phi_t(x(0), a)) - b\| > \epsilon.$$

In particular, the trajectory is 0-safe if $\epsilon = 0$.

PROBLEM 2.1 (SUPERVISOR DESIGN). *Determine a set $X_0 \subset X$, such that there exists an $a \in \mathcal{A}$ that makes $\phi_{[0,\infty)}(x(0), a)$ 0-safe for all $x(0) \in X_0$, and such that $\phi_t(x(0), a) \in \Omega \cap X_0$ for some $t > 0$. Then determine a supervisor $\sigma : \mathcal{X} \times \mathbb{R} \mapsto 2^A$ that attaches to each pair $(x([0, t]), t)$ a (possibly empty) set of inputs $\sigma(x([0, t]), t)$, such that, for all $x(0) \in X_0$, $a(t) \in \sigma(x([0, t]), t)$ implies $\phi_t(x(0), a) \in X_0$ for all $t > 0$.*

3. OUTLINE OF THE SUPERVISOR DESIGN

The design of the supervisor σ proceeds through five main steps. Here we outline the main points of each step:

(i) We introduce a system of the form

$$\dot{\chi} = u, \tag{3}$$

with state $\chi(t) \in Y$ and input $u : \mathbb{R} \mapsto U \subset \mathbb{R}^n$. We define U as a set of vectors with elements in $u_{adm} := \{k\mu, \mu > 0, k \subset \mathbb{Z}\}$, and assume that u is constant over intervals $(k\tau, (k+1)\tau]$. We call $\mathcal{U}$ the set of all such signals u. By hypothesis the bad set B and the terminal set Ω are defined in the set Y, so we can design a supervisor σ_C ensuring ϵ-safety of trajectories of (3). The concept of ϵ-safety, defined before, can be extended to (3) simply by substituting the flow $\phi_t(\chi, u)$ of (3) to the function $h(\phi_t(x(0), a))$ in Definition 2.1.

(ii) We obtain a time-τ discretization of (3), called Σ_{DT}. The state space of Σ_{DT} is Y (as for (3)), and can be divided into a regular grid with hypercubic cells of side η for some $\eta > 0$. The quantity η is chosen so that any transition of Σ_{DT} maps all states in a cell into states in another cell. This is possible by taking $\eta = \tau\mu$, given the form of the signals u and their codomain U.

(iii) We construct the discrete event system $\Sigma_{DE} := (G, W, \psi)$, where G is the set of states, W the set of events, and ψ the transition function defined as follows. A transition is a pair (g, w) with $g \in G$ and $w \in W$. We construct G so that it is has one state for each cell in which we have divided Y and we define the function $\ell : Y \mapsto G$ that takes states $\chi(t) \in Y$ of a cell to the corresponding state $g \in G$. We set $W = U$. Finally, we construct ψ so that there exists a transition (g^1, w) between two states $g^1 \in G$ and $g^2 \in G$ if and only if there exist two states $\chi(0), \chi(\tau) \in Y$ such that $g^1 = \ell(\chi(0))$, $g^2 = \ell(\chi(\tau))$, and $\chi(\tau) = \phi_\tau(\chi(0), u)$ when $u(t) = w$ for all $t \in [0, \tau]$. A transition (g, w) is defined ϵ-safe if, for all $\chi(0)$ such that $\ell(\chi(0)) = g$, the trajectory $\phi_{[0,\tau]}(\chi(0), u)$ with $u(t) = w$ for all $t \in [0, \tau]$ is an ϵ-safe trajectory of (3). This can be checked easily by checking that the trajectory $\phi_{[0,\tau]}(\bar{\chi}(0), u)$ is $(\epsilon + \eta/2)$-safe, where $\bar{\chi}(0)$ lies at the centre of the hypercube $\ell^{-1}(g)$. Checking safety, in turn requires to verify if $\phi_{[0,\tau]}(\bar{\chi}(0), u)$, which is a straight segment, intersects an $(\epsilon + \eta/2)$-neghbourhood of B, which is a union of polytopes. This is equivalent to checking a set of linear inequalities. With this structure, Σ_{DE} can be proved to be bisimilar to Σ_{DT} (see [2] for a definition of bisimilarity).

(iv) We construct a supervisor $\sigma_C(\chi([0, t]), t) \in 2^U$ for (3) using Σ_{DE}. We compute all transitions of Σ_{DE} that are not ϵ-safe. Then, we classify as terminal any state $g \in G$ such that for all $\chi(t) \in Y$ with $\ell(\chi(t)) = g$, $\chi(t) \in \Omega$. Using Dijkstra's algorithm we identify all the states of Σ_{DE} that can reach a terminal state using only ϵ-safe transitions. Call $S \subset G \times W$ the set of all ϵ-safe transitions between such states. We define $\sigma_C(\chi([0, t]), t) := \{u(t) \in U : u(t) = w$ for all $k\tau < t \leq (k+1)\tau$, and $(\ell(\chi(k\tau)), w) \in S\}$.

(v) We construct the supervisor σ for (1). The trajectories allowed by σ are obtained, through a suitable mapping, from the trajectories of (3) allowed by σ_C. This mapping is, in turn, constructed exploiting the flatness property of (1). Having set $\chi(0) = h(x(0))$,

we construct y as a piecewise polynomial signal, C^q everywhere, passing through the points $\chi(k\tau)$, $k \in \mathbb{N}$. In each interval $(k\tau, (k+1)\tau]$, y is equal to a polynomial $p^k(t, h(x([0,t])), u(t))$, whose coefficients are chosen so that y passes through $\chi(k\tau)$ and $\chi((k+1)\tau)$, and such that the derivatives of y up to order q at $t = (k+1)\tau$ are either null or expressed as a function of $u(t) \in \sigma_C(\chi([0,t]), t)$, which is constant in the interval, and the derivatives at $t \to k\tau$ ensure $y \in C^q$. Notice that these coefficients can be calculated for $t \in (k\tau, (k+1)\tau]$ knowing only t, $h(x([0,t]))$, and $u(t)$, since $\chi(k\tau) = h(x(k\tau))$ and $\chi((k+1)\tau) = \chi(k\tau) + u(t)\tau$ with $u(t) \in \sigma_C(\chi([0,t]), t)$. The coefficients of $p^0(t, h(x([0,t])), u(t))$ are taken so that y at $t \to 0$ satisfies the first equation of (2).

The set X_0 that solves Problem 2.1 is $X_0 = \{x(0) \in X : (\ell(h(x(0))), w) \in S \text{ for some } w \in W\}$. Let us define

$$\sigma(x([0,t]), t) := \Theta(y(t), \dot{y}(t), ..., y^{(q+1)}(t))$$

with

$$y(t) := p^k\Big(t, h(x([0,t])), \sigma_C(h(x([0,t]))), t)\Big), \quad k\tau < t \leq (k+1)\tau.$$

Since the polynomials p^k have coefficients and domain in a compact set, the quantity

$$\epsilon_a = \sup_{u\in\mathcal{U}} \sup_{t>0} \inf_{s>0} \|y(t) - \phi_s(\chi(0), u)\|, \tag{4}$$

which is the maximum distance of $y(t)$ from $\chi(t)$, is finite. Setting $\epsilon > \epsilon_a$ and constructing σ_C to enforce ϵ-safety, the supervisor σ solves Problem 2.1.

The supervisor can be shown to be the least restrictive (in the family of supervisors with piecewise polynomial output defined above), meaning that any input $a(t) \notin \sigma(x([0,t],t)$ violates the requirements of Problem 2.1. Additionally, if for all $\chi(0) \in Y$ and for all $u \in \mathcal{U}$ there exists a fixed $T > 0$ such that $\ell(\phi_T(\chi(0), u)) \in G$ is a terminal state, we can prove an upper bound on the distance between the flat output of orbits allowed by σ and the bad set B. Specifically, consider the set u_{adm} with elements in increasing order, and define δ as the maximum difference between two successive elements. Then, we can prove that, if $a(t) \in \sigma(x[0,t],t)$ for all t, then $\min_{t\in[0,\infty)} \min_{b\in B} \|h(\phi_t(x(0), a)) - b\| \leq 2\epsilon + \tau\delta$.

The complexity of the supervisor design discussed above grows linearly with the number of transitions of Σ_{DE}, which is equal to the cardinality of W times the cardinality of G. The first is $O(p^n)$, where p is the cardinality of u_{adm}, and n the dimension of Y; the second is $O(c^n)$, where the constant $c > 1$ depends on the number of steps of length η subdividing Y in each direction. The overall complexity is thus $O((pc)^n)$.

4. EXAMPLE

Consider a set of n vehicles travelling in straight lines along m roads that intersect at a common point, as in the example in Fig. 1. A vehicle's position at time t is represented by $x_i(t) \in \mathbb{R}, i \in \{1, ..., n\}$. Let $x(t) = (x_1(t), ..., x_n(t))$. The set of indices $\{1, ..., n\}$ of x can be partitioned into m sets $\Delta^k, k \in \{1, ..., m\}$, where k indicates the road along which vehicle i is travelling. If $i \in \Delta^k$ and $j \in \Delta^k$, then vehicles i and j travel along the same road (e.g., in Fig. 1 indices 1 and 2 are in Δ^1, indices 3 and 4 in Δ^2 and Δ^3, respectively). The set of vehicles obeys the law

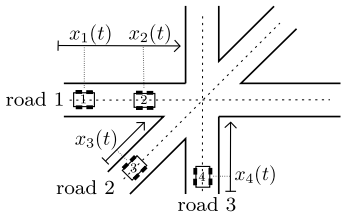

Figure 1: Four vehicles along three roads, heading towards a common intersection. The quantities $x_i(t)$ represent the positions of the vehicles at time t.

$$\begin{aligned} \dot{x} &= v \\ \dot{v} &= a, \end{aligned} \tag{5}$$

which is a flat system with flat output $y = x$ and

$$\begin{aligned} (x, v) &= \Gamma(y, \dot{y}, ..., y^{(q)}) = (y, \dot{y}) \\ a &= \Theta(y, \dot{y}, ..., y^{(q+1)}) = y^{(2)}. \end{aligned} \tag{6}$$

Our objective is to design a control law that prevents vehicle collisions, while ensuring that all vehicles pass the intersection and that their velocities remain nonnegative. States in the bad set satisfy one of two sets of inequalities. The first one is used for two vehicles, $i \in \Delta^k$ and $j \in \Delta^l$, driving along two different roads, where $y_i(t) \in [\alpha_k, \beta_k]$ if vehicle i is in the intersection. Then a collision occurs if

$$\begin{aligned} &\exists i \in \Delta^k, j \in \Delta^l, k \neq l \text{ such that } \alpha_k < y_i(t) < \beta_k \\ &\text{and } \alpha_l < y_j(t) < \beta_l, \alpha_k, \alpha_l, \beta_k, \beta_l \in \mathbb{R}. \end{aligned} \tag{7}$$

The second condition is used for two vehicles driving on the same road. Let $\gamma \in \mathbb{R}$ be the minimum safe distance between vehicles i and j. Then a collision occurs if

$$\exists i, j \in \Delta^k \text{ such that } -\gamma < y_i(t) - y_j(t) < \gamma. \tag{8}$$

A state $y \in B$ if (7) or (8) is satisfied. The control problem is solved once each vehicle $i \in \Delta^k$ passes a predetermined position p_i, where $p_i \geq \beta_k$. Defining the vector p of elements p_i, we can set Y equal to a hypercube and $\Omega := \{y \in Y \text{ such that } y \geq p\}$.

We shall assume that x and v are continuous signals, while a can be only piecewise continuous. This implies that y satisfying (6) must be of class C^1. We can thus approximate each trajectory $\phi_{[k\tau,(k+1)\tau]}(\chi(k\tau), u)$, with a cubic polynomial $p^k(t, (x([0,t]), v([0,t])), u(t))$ with the following boundary conditions:

$$\begin{aligned} &\lim_{t\to k\tau} p^k(t, (x([0,t]), v([0,t])), u(t)) = x(k\tau) \\ &p^k(t, (x([0,t]), v([0,t])), u(t))|_{t=(k+1)\tau} = x(k\tau) + \tau u(t) \\ &\lim_{t\to k\tau} \dot{p}^k(t, (x([0,t]), v([0,t])), u(t)) = v(k\tau) \\ &\dot{p}^k(t, (x([0,t]), v([0,t])), u(t))|_{t=(k+1)\tau} = u(t). \end{aligned}$$

The corresponding signal y is given by

$$\begin{aligned} y(t) = x(k\tau) + tv(k\tau) - \frac{2t^2}{\tau}(v(k\tau) - u(t)) + \\ \tfrac{t^3}{\tau^2}(v(k\tau) - u(t)), k\tau < t \leq (k+1)\tau. \end{aligned}$$

We set $u_{adm} = \{1, 2, 3, 4\}$, which with the equation above ensures that v is nonnegative. Assuming $v(0) \in [1, 4]$, one

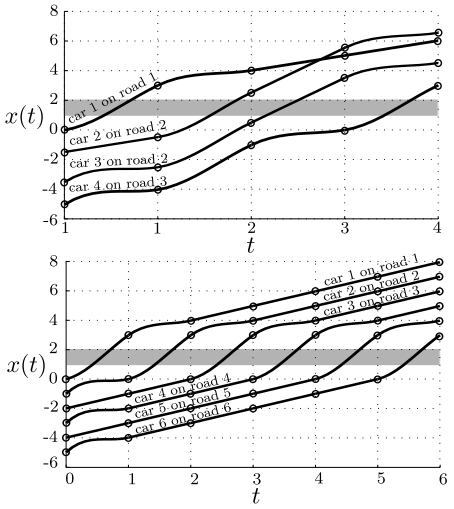

Figure 2: Trajectories of 4 cars on 3 roads (top), and 6 cars on 6 roads (bottom), with $\alpha = 1$, $\beta = 2$ (equal for all cars), $\gamma = 1$, $\eta = 1$. The gray area is the crossing $\alpha < x_i < \beta$. Two components corresponding to vehicles on different roads must never be simultaneously in the crossing, and two components corresponding to vehicles on the same road must never be closer than $\gamma = 1$. The initial conditions are: $x(0) = (0, -1.5, -3.5, -5)$, $v(0) = (1, 1, 4, 4)$ (top), and $x(0) = (0, -1, -2, -3, -4, -5)$, $v(0) = (1, 4, 1, 4, 1, 4)$ (bottom). The discrete event system had 390 states and 256 transitions per-state (top), and 7626 states and 4096 transitions per-state (bottom), and the running time on a single 2.4GHz processor was respectively 1.14 seconds, and 397 seconds.

can easily prove the following:

$$\begin{aligned}&\sup_{u\in\mathcal{U}} \sup_{t>0} \inf_{s>0} \|y(t) - \phi_s(\chi(0), u)\| < \\ &\sup_{u\in\mathcal{U}} \sup_{t>0} \inf_{t>0} \|y(t) - \phi_t(\chi(0), u)\| \leq \\ &\sup_{u\in\mathcal{U}} \|v(0) - u(\tau)\| \frac{4\tau}{27} < 0.45.\end{aligned}$$

Thus $\epsilon_a < 0.45$. If σ_C ensures 0.45-safety, the supervisor σ ensures 0-safety. We have applied the algorithm detailed above to the case of 4 vehicles driving along 3 roads, and to the case of 6 vehicles on 6 different roads. Parameters α and β have been set to 1 and 2, respectively, for all vehicles, while $\gamma = 1$, $\eta = 1$, $\mu = 1$, and $\tau = 1$. Figure 2 portrays the different components of the trajectory of (5) for both cases, while the crossing is represented in grey. The trajectories lie outside of the bad set if two components corresponding to vehicles on different roads are never in the gray region simultaneously, and two components corresponding to vehicles on the same road maintain a distance greater than $\gamma = 1$.

5. CONCLUSION

We have proposed an algorithm for the supervisory control of differentially flat systems, using model abstraction. By merging the trajectory planning techniques and the abstraction approach, we have obtained an algorithm that can handle relatively large systems, yet it provides guarantees on the safety of the allowed trajectories. Our approach starts by considering a simple system living in the space of the flat output variables. We construct an abstraction of this system, which by virtue of the simple dynamics can be made deterministic, and design a supervisor based on the abstraction. The control inputs allowed by the supervisor are then mapped back onto the original system using flatness. The allowed trajectory are safe by construction. The algorithm can be applied to differentially flat systems in the form specified in Section 2, as long as the bad set is expressed in terms of the output variables. The bad set is an arbitrary but finite union of polytopes.

Unlike previous abstraction-based approaches [1, 8, 9, 4, 11], our technique provides a deterministic abstraction without requirements on the system's stability. This is a definite advantage when the complexity of the abstraction is a bottleneck. Moreover, unlike all these approaches our algorithm provides safety guarantees for the allowed orbits.

Our approach is similar to a roadmap-based motion planning algorithm [5]. However, rather than a single trajectory, our algorithm provides a set of acceptable trajectories, which can be proved to be the largest within the considered family, and with additional conditions, we can provide a minimal distance between the allowed orbits and the bad set.

The main limit of our approach is the complexity, which is exponential in the number of variables. A decisive step forward will be the exploitation of the geometric and dynamic properties of a system to reduce the complexity class of the algorithm.

6. REFERENCES

[1] M. Broucke, M. D. Di Benedetto, S. Di Gennaro, and A. Sangiovani-Vincentelli. Efficient solution of optimal control problems using hybrid systems. *SIAM J. Contr. Opt.*, 43:1923–1952, 2005.

[2] C. G. Cassandras and S. Lafortune. *Introduction to Discrete Event Systems*. Springer-Verlag, 2008.

[3] M. Fliess, J. Lévine, P. Martin, and P. Rouchon. Flatness and defect of non-linear systems: Introductory theory and examples. *Int. J. Control*, 6:1327–1361, 1995.

[4] A. Girard, G. Pola, and P. Tabuada. Approximately bisimilar symbolic models for incrementally stable switched systems. *IEEE Trans. Autom. Control*, 55:116–126, 2010.

[5] J. C. Latombe. *Robot motion planning*. Kluwer Academic Publishers, 1991.

[6] J. Lévine. *Analysis and control of nonlinear systems: A flatness-based approach*. Springer, 2009.

[7] P. J. Ramdage and W. M. Wonham. Supervisory control of a class of discrete event processes. *SIAM J. Contr. Opt.*, 25:206–230, 1987.

[8] P. Tabuada. An approximate simulation approach to symbolic control. *IEEE Trans. Autom. Control*, 53:1406–1418, 2008.

[9] P. Tabuada. *Verification and control of hybrid systems*. Springer-Verlag, 2009.

[10] M. van Nieuwstadt, M. Rathinam, and R. M. Murray. Differential flatness and absolute equivalence of nonlinear control systems. *SIAM J. Contr. Opt.*, 36:1225–1239, 1998.

[11] M. Zamani, G. Pola, M. Mazo Jr., and P. Tabuada. Symbolic models for nonlinear control systems without stability assumptions. *arXiv:1002.0822v3*, 2010.