2013 European Control Conference (ECC)
July 17-19, 2013, Zürich, Switzerland.

An approximate abstraction approach to safety control of differentially flat systems

Alessandro Colombo and Antoine Girard

Abstract— **Control for safety specifications of large nonlinear systems is a challenging task. By reducing the system to a discrete abstraction the computational demands of the controller can be greatly reduced. We propose a supervisor for differentially flat systems, based on an approximate abstraction of the flat output. By defining the abstraction on the flat output space, we simplify the design of the abstraction and further reduce the computational complexity of the resulting supervisor, and by exploiting approximate simulation techniques we obtain a controller that is simpler and more flexible than previously proposed solutions. The resulting algorithm is tested on an eight-dimensional nonlinear planar crane model.**

I. Introduction

The design of complex and safety-critical engineering systems demands provably safe control algorithms, guaranteed to keep the system's state outside of an unsafe set under all operating conditions. Supervisory control represents an interesting solution to this problem, particularly in the context of multi-objective optimal control, where a supervisor enforcing safety can act as a layer of a more complex controller, or when the controller must interface with human-provided control decisions, disallowing unsafe ones. A supervisor designed for this purpose must be computationally efficient, to cope with the system complexity, and must be as permissive as possible, to avoid imposing unnecessary restrictions on the set of allowed control decisions. A common approach to the design of such algorithms is based on reducing the continuous system to an *abstraction*. This is a simplified representation of the system's dynamics that preserves some significant properties, such as reachability or set invariance. By mapping a system's dynamics onto a simpler one, typically lower dimensional, discrete, or finite dimensional, the problem of verifying or controlling the system is simplified. The process of abstraction usually relies on some particular properties of the abstracted system, which is required to have linear or affine dynamics [1], [2], to be monotone [3], incrementally stable [4], [5], [6], or to have a weakly integrable vector field [7], though more general approaches have been proposed [8]. Frequently, the process of abstraction is based on the concept of *bisimulation* [9], which is an equivalence relation between the dynamics structure of two systems.

In this paper, we design a supervisory control based on a discrete abstraction, that can be applied to differentially flat systems. These are linear or nonlinear systems whose state and input can be written as a function of an output (called the *flat output*) and a finite number of its derivatives. Expressing the flat output and its derivatives as the state of a chain of integrators, trajectories of a flat system correspond to trajectories of the chain of integrators. What is more important, designing inputs for a differentially flat system to drive its flat output along a given trajectory is trivial, as the problem is reduced to evaluating an algebraic map of the trajectory of the chain of integrators. In [10], [11], we proposed a supervisory control algorithm based on the abstraction of differentially flat systems. Here, we extend the results above by exploiting the property, observed in [12], that any n-th order integrator can be rendered *approximately bisimilar* to a first order integrator by the use of a suitable feedback map (see e.g. [13] for a definition of approximate bisimilarity). Based on this observation, we define a first order approximate bisimulation of a generic differentially flat system. Then, we derive a finite abstraction of the first order approximation. The abstraction is, by construction, approximately bisimilar to the original system with feedback control. By bounding the approximation error introduced by the bisimulation, we design a supervisory control algorithm for the original system based on its discrete abstraction. The supervisor is proved correct (safety-enforcing and nonblocking) and can be made arbitrarily permissive by refining the abstraction.

The approach proposed here has several advantages over that initially proposed in [10], [11]. The new approach allows inputs that change continuously within some bounded set, rather than requiring a discrete set of inputs. Thus, this supervisor can be easily overlaid to other control algorithms as the safety-enforcing layer in a multi-objective control scheme [14], or it can be implemented as a safety-enforcing filter to the allowed inputs, without requiring to redesign the input set. Moreover, the additional degrees of freedom provided by the design of the approximate bisimulation allow to tune the set of allowed inputs to meet physical constraints, for example by reducing the required control efforts. The proposed algorithm is tested on an eight-dimensional nonlinear model of a planar crane.

The paper is organised as follows. In the next section, we formally introduce the class of systems addressed by our algorithm. Then, in Section III, we follow the approach presented in [12] to approximate the dynamics of (1) with that of a first order integrator. In Section IV, we design a finite abstraction of the first order integrator. In Section V, we use the abstraction to define a supervisor for (1) that solves the two problems above. Finally, in Sections VI we

This work was supported by the Agence Nationale de la Recherche (VEDECY project - ANR 2009 SEGI 015 01) and the Université Joseph Fourier (SYMBAD project).

A. Colombo is with DEIB, Politecnico di Milano, via Ponzio 34/5, 20133 Milano, Italy. alessandro.colombo@polimi.it

A. Girard is with Laboratory Jean Kuntzmann, University of Grenoble, B.P. 53, 38041 Grenoble, France Antoine.Girard@imag.fr

discuss an example application of the proposed algorithm.

II. Problem Statement

Consider the system

$$\dot{x} = f(x,a), \quad y = h(x), \tag{1}$$

with $x(t) \in X \subseteq \mathbb{R}^m$, $a(t) \in A \subseteq \mathbb{R}^n$, $y(t) \in Y \subseteq \mathbb{R}^n$. Functions f and h are C^k for sufficiently large k. Call $\mathcal{A}$ the space of input signals a, and let x be the state of the system and y be the output. We denote x, y, and a the state, output, and input signals of (1), that is, maps $\mathbb{R}_+ \to \mathbb{R}^m$, $\mathbb{R}_+ \to \mathbb{R}^n$, and $\mathbb{R}_+ \to \mathbb{R}^n$ respectively. The symbols $x(t)$, $y(t)$, and $a(t)$ denote the same quantities evaluated at time t. When referring to a specific element of the spaces X, Y or A, we overline the corresponding symbol, as in $\bar{y} \in Y$. When the input corresponding to a given output or state must be specified, we write $y(t,a)$ or $x(t,a)$, and when the initial conditions $x(0)$ need to be explicitly specified, we use the notation $y(t,a,x(0))$ or $x(t,a,x(0))$. Finally, $y([t_1,t_2]) := \bigcup_{t\in[t_1,t_2]} y(t)$. This notation extends trivially to the other formalisms introduced for state, output, and input. We assume that (1) is differentially flat [15], [16], [17], with y as the flat output. This means that function h has rank n (i.e, it's Jacobian has rank n uniformly), and 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

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

We also require that $n(q+1) = m$, which together with the rank condition implies that the function Γ is invertible.

We consider the set $B \subset Y$, called the *bad set*. Our objective is to design a supervisor [18], [9] for system (1) that prevents output trajectories from entering the bad set, as long as the trajectories are within a compact subset $\hat{Y}$ of Y . This requirement can be formally expressed using the concept of ϵ-*safe* trajectory:

Definition 2.1: An output trajectory $y([0,T]) \subset \hat{Y}$ is ϵ-safe provided $\inf_{t\in[0,T]} \inf_{b\in B} \|y(t) - b\|_\infty > \epsilon$.

Given a grid of hypercubic cells of side η defined on Y, let $\hat{Y}$ be a compact subset of Y composed by a finite number of such cells. We shall design a supervisor $\sigma : X \to 2^{\mathcal{A}}$ for (1) that enforces 0-safety within $\hat{Y}$. The supervisor will be based on an approximately bisimilar abstraction of (1), and will depend on space and time discretization parameters η and τ. More precisely, we aim to solve the following problem.

Problem 2.1 (Correctness): Define a supervisor that attaches to each $x(k\tau)$ a set of inputs a defined in the interval $[k\tau,(k+1)\tau]$, with the following properties:

(P.1) If $a \in \sigma(x(k\tau))$ and $y(k\tau) \in \hat{Y}$, then $y([k\tau,(k+1)\tau],a)$ is 0-safe

(P.2) If $\sigma(x(k\tau)) \neq \emptyset$, $a \in \sigma(x(k\tau))$, and $y((k+1)\tau,a) \in \hat{Y}$, then $\sigma(x((k+1)\tau,a)) \neq \emptyset$ (non-blockingness)

A supervisor solution of the above problem is correct by design, that is, allowed inputs are guaranteed to exist for all positive time and the corresponding solutions are guaranteed to lay outside of the bad set. One may additionally wish to design the supervisor such that the set of allowed inputs is as large as possible. This can be achieved asymptotically, by refining the abstraction on which the supervisor is based. This property is captured by the following problem.

Problem 2.2 (Optimality): Define a supervisor solution of Problem 2.1 with the following additional property:

(P.3) For any δ-safe trajectory $y^*([0,T]) = h(x^*([0,T])) \subset \hat{Y}$, there exists a $d > 0$ and a supervisor based on a grid of cells of side $\eta < d$, such that setting $x(0) = x^*(0)$, some input $a \in \mathcal{A}$ verifies $a \in \sigma(x(\lfloor t/\tau \rfloor \tau, a))$ for all $t \in [0,T']$ and such that the infinity-norm Hausdorff distance[1] $\mathcal{H}(y([0,T'],a), y^*([0,T])) \leq \delta$.

We solve the above problems by designing a suitable feedback control for system (1), such that the controlled system is approximately bisimilar to a first order integrator. Then we design a supervisor for a discrete abstraction of the first order integrator.

III. Herarchical Control Using Simulation Functions

We know that the output trajectories of the flat system (1) coincide with the trajectories of the linear system: $y^{(q+1)} = u$ that we should rewrite under the form

$$\left\{ \begin{array}{rcl} \dot{\theta} & = & A\theta + Bu, \\ y & = & C\theta \end{array} \right. \tag{3}$$

where

$$\theta = \begin{bmatrix} y \\ \dot{y} \\ \vdots \\ y^{(q)} \end{bmatrix}, \; A = \begin{bmatrix} 0_n & I_n & 0_n & \dots & 0_n \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & 0_n \\ 0_n & \dots & \dots & 0_n & I_n \\ 0_n & \dots & \dots & \dots & 0_n \end{bmatrix},$$

$$B^T = [0_n \; \dots \; 0_n \; I_n], \quad C = [I_n \; 0_n \; \dots \; 0_n],$$

where I_n denotes the $n \times n$ identity matrix, 0_n denotes the $n \times n$ zero matrix. Notice that, by the invertibility of Γ in (2), $\theta = \Gamma^{-1}(x)$. We want to approximate (3) by the first order integrator

$$\dot{z} = v, \tag{4}$$

with $z \in Z = Y \subset \mathbb{R}^n$ and $v \in V \subset \mathbb{R}^n$. We use for z and θ the same notation introduced in Section II for x, y and a.

Definition 3.1: Let $\mathcal{V} : \mathbb{R}^{(q+2)\times n} \to \mathbb{R}^+$ be a smooth function and $u_{\mathcal{V}} : \mathbb{R}^{(q+2)\times n} \to \mathbb{R}^n$ be a continuous function. $\mathcal{V}$ is a simulation function of (4) by (3) and $u_{\mathcal{V}}$ is an associated interface if there exists a $\mathcal{K}$ function[2] γ such that for all $(z,\theta) \in \mathbb{R}^{(q+2)\times n}$,

$$\mathcal{V}(z,\theta) \geq \|z - C\theta\|_2 \tag{5}$$

and for all $v \in \mathbb{R}^n$, satisfying $\gamma(\|v\|_2) < \mathcal{V}(z,\theta)$,

$$\frac{\partial \mathcal{V}(z,\theta)}{\partial z} \cdot v + \frac{\partial \mathcal{V}(z,\theta)}{\partial \theta} \cdot (A\theta + Bu_{\mathcal{V}}(v,z,\theta)) < 0 \tag{6}$$

[1]The infinity-norm Hausdorff distance $\mathcal{H}(X,Y)$ between two sets X and Y is $\max\{\sup_{x\in X} \inf y \in Y \|x-y\|_\infty, \sup_{y\in Y} \inf x \in X \|x-y\|_\infty, \}$

[2]A function $\gamma : \mathbb{R}_+ \to \mathbb{R}_+$ is a $\mathcal{K}$ function if it is continuous, strictly increasing and satisfies $\gamma(0) = 0$.

A simulation function allows us to bound the distance between output trajectories of (4) and (3).

Theorem 3.1 (proved in [12]): Let $\mathcal{V}$ be a simulation function of (4) by (3) and $u_{\mathcal{V}}$ an associated interface. Let $v : \mathbb{R}_+ \to V \subset \mathbb{R}^n$ be an input of (4), let z and y be the trajectories of (4) and (3) given by

$$\left\{ \begin{array}{rcl} \dot{z} & = & v, \\ \dot{\theta} & = & A\theta + Bu_{\mathcal{V}}(v, z, \theta), \\ y & = & C\theta \end{array} \right. \tag{7}$$

Then, for all $t \in \mathbb{R}_+$, $\|z(t) - y(t)\|_2 \leq \max\left\{\mathcal{V}(z(0), \theta(0)), \sup_{s\in[0,t]} \gamma(\|v(s)\|_2)\right\}$, and $\mathcal{V}(z(t), \theta(t)) \leq \max\left\{\mathcal{V}(z(0), \theta(0)), \sup_{s\in[0,t]} \gamma(\|v(s)\|_2)\right\}$.

We use the method proposed in [12] to compute a simulation function of (4) by (3) and an associated interface. Let us assume K is a stabilizing gain for (3) (i.e. $A + BK$ is Hurwitz), K can be obtained easily for instance by pole placement or LQ synthesis. Let $\lambda > 0$ and M a positive definite symmetric matrix such that

$$M \geq C^T C, \tag{8}$$

$$(A + BK)^T M + M(A + BK) \leq -2\lambda M. \tag{9}$$

Proposition 3.1 ([12]): The function defined by $\mathcal{V}(z, \theta) = \sqrt{(C^T z - \theta)^T M (C^T z - \theta)}$ is a simulation function of (4) by (3) and an associated interface is given by

$$u_{\mathcal{V}}(v, z, \theta) = Rv + K(\theta - C^T z) \tag{10}$$

where R is an arbitrary $n \times n$ matrix. The function γ is then given by

$$\gamma(\nu) = \frac{\left\|\sqrt{M}(BR - C^T)\right\|_2}{\lambda}\nu \tag{11}$$

and is minimal for $R = (B^T M B)^{-1} B^T M C^T$.
The entries of the matrices M and K can be tuned to meet further dynamic constraints. Given the bound $\epsilon \geq \|z - y\|_2$, $\|(z-y)^{(i)}\|_2 \leq \alpha_i \epsilon$ provided that $M \geq C_i^T C_i / \alpha_i^2$, where C_i is the i-th versor, while the $\|u\|_2 \leq \alpha_k \epsilon$ by imposing $M \geq K^T K / \alpha_k^2$. With the above construction, the flat system (3) controlled by the feedback map $u_{\mathcal{V}}$ as in (7) is approximately bisimilar to (4).

IV. Approximately Bisimilar Finite Abstraction

Using the approach of Section III, we can approximate the dynamics of system (1) with that of the first order integrator (4). We now design a discrete abstraction of (4), that is, a discrete event system that is (exactly) bisimilar to (4), and is therefore approximately bisimilar to (3) (and hence to (1)), once the latter is controlled through the feedback map $u_{\mathcal{V}}$. By discretizing the dynamics of (4), we can turn the control problem into the problem of controlling a finite discrete event system.

Consider a regular lattice Q of step η over Z, such that an element of the lattice lies in the centre of each hypercubic cell composing Z. Let q denote an element of Q. Since both $q \in Q$ and $z \in Z$ are elements of $\mathbb{R}^n$, the infinity norm defines a distance for any pair (q, z). The lexicographical order is a total order on the elements of Q, so that any subset of Q has a unique minimum. Let $\ell(\bar{z}) := \min_{q\in Q}\{q : \|\bar{z} - q\|_\infty \leq \eta/2\}$. Define the discrete event system

$$G := (Q, V_d, \psi) \tag{12}$$

with states $q \in Q$, events $v_d \in V_d$, and transition function $\psi(q, v_d)$. To define V_d and $\psi(q, v_d)$, we proceed as follows. Given a signal v constant and equal to $\bar{v}$ over an interval of length τ, we denote $v_d \overset{\bar{\tau}}{\Leftrightarrow} \bar{v}$ if $\ell(\bar{z} + \bar{v}\tau) = \ell(\bar{z}) + v_d\tau$. We define $V_d \subset \mathbb{R}^n$ as the set $\{v_d \in \mathbb{R}^n : \exists \bar{z} \in Z, \bar{v} \in V$ such that $v_d \overset{\bar{\tau}}{\Leftrightarrow} \bar{v}\}$, The transition function is then defined as follows: $\psi(q, v_d) := q'$ if $\exists\, \bar{z} \in Z, \bar{v} \in V$ such that $q = \ell(\bar{z})$, $q' = \ell(\bar{z} + \bar{v}\tau)$, and $v_d \overset{\bar{\tau}}{\Leftrightarrow} \bar{v}$. The map ℓ relates each $z \in Z$ to a state of G and *vice versa*, and the definition of ψ ensures that for each trajectory of (4) of duration τ there exists a corresponding transition of G, and *vice versa*. Thus, the discrete event system G and the time-τ discretization of (4) are bisimilar (see [9] for a formal definition of bisimilarity), and consequently G and the time-τ discretization of (1) are approximately bisimilar. The symbol s is used to denote a generic string $v_d^1 v_d^2 \ldots$, finite or infinite. Also, given a state $q \in Q$, we denote by (q, v_d) a transition of (12) from state q with event v_d, and by (q, s) an execution of (12) starting from initial state q, with events string $s = v_d^1 v_d^2 \ldots$. We use the notation $\psi(q, s)$ to denote the last state reached by the finite execution (q, s).

In general, the cardinality of the state set Q is infinite. However, we only need the abstraction to control the system (1) within the output subset $\hat{Y}$. Thus, we define the restricted abstraction $\hat{G} := (\hat{Q}, V_d, \psi)$, obtained by restricting the state set to the subset $\hat{Q}$ of Q of all states q such that $\ell(\bar{z}) = q$ for some $\bar{z} \in \hat{Y}$. Thus, $\hat{G}$ is a finite approximate abstraction of (1), with output restricted to $\hat{Y}$.

V. Synthesis of the Supervisor

By reducing (1) to a finite abstraction, we can solve Problems 2.1 and 2.2 by selecting a suitable set T of executions of $\hat{G}$. Let $P := (CMC^T)^{-1}CM$, and call $u_\tau(\bar{v}, \bar{\theta})$ the signal $u_{\mathcal{V}}(v, z, \theta)$ obtained by solving (7) in the interval $[0, \tau]$ with initial conditions $z(0) = P\bar{\theta}$ and $\theta(0) = \bar{\theta}$, and with constant input $v = \bar{v}$. Then, call $a(u_\tau(\bar{v}, \theta(k\tau))) = a(u_\tau(\bar{v}, \Gamma^{-1}(x(k\tau))))$ the input corresponding to $u_\tau(\bar{v}, \theta(k\tau)))$ through the map Θ in (2). Given a set T, construct the supervisor $\sigma(x(k\tau))$ for each time interval $[k\tau, (k + 1)\tau]$ as the union of all signals

$$a(u_\tau(\bar{v}, \Gamma^{-1}(x(k\tau)))) \tag{13}$$

with $\bar{v}$ such that $\bar{v} \overset{P\theta(k\tau)}{\Leftrightarrow} v_d$ for some $v_d \in V_d$ and $s \in 2^{V_d}$, $q = \ell(P\theta(k\tau))$ and $(q, v_d s) \in T$. To ensure that the supervisor meets the specifications of Problems 2.1 and 2.2, we endow T with the following properties.

Definition 5.1: An execution (q, s) is forward-maximal if $\psi(q, v_d^1 \ldots v_d^n) \in \hat{Q}$ for all $n < m$, and $\psi(q, v_d^1 \ldots v_d^m) \notin \hat{Q}$, or if $\psi(q, v_d^1 \ldots v_d^m,) \in \hat{Q}$ for all $m > 0$.

Definition 5.2: A set T of executions is ϵ-non-escaping if $(q, v_d s) \in T$ implies that, for all $\bar{z} \in Z$ such that $\|\bar{z} - \psi(q, v_d)\|_\infty \leq \epsilon + \eta/2$, $(\ell(\bar{z}), s') \in T$ for some s'.

Definition 5.3: A transition (q, v_d) such that $\psi(q, v_d) = q'$ is ϵ-safe if $\inf_{t\in[0,\tau], b\in B} \|t(q'-q)/\tau + q - b\|_\infty \geq \epsilon + \eta/2$. An execution (q, s) is ϵ-safe if all the transitions that compose it are ϵ-safe.

We first prove that the supervisor σ based on a set T of executions with the above properties solves Problem 2.1. For simplicity, we split the proof in four lemmas.

Lemma 5.1: Take $\epsilon \geq \max\{\mathcal{V}(z(k\tau), \theta(k\tau)), \sup_{\bar{v}\in V} \gamma(\|\bar{v}\|_2)\}$. If $a = a(u_\tau(\bar{v}, \theta(k\tau)))$ for some $\bar{v} \in V$ in the time interval $[k\tau, (k+1)\tau]$, then $\mathcal{V}\left(P\theta((k+1)\tau), \theta((k+1)\tau)\right) \leq \epsilon$.

Proof: From Theorem 3.1 $\mathcal{V}(z(t), \theta(t)) \leq \max\{\mathcal{V}(z(0), \theta(0)), \sup_{s\in[0,t]} \gamma(\|v(s)\|_2)\}$ for all $t \geq 0$. Hence, if $\epsilon \geq \max\{\mathcal{V}(z(k\tau), \theta(k\tau)), \sup_{\bar{v}\in V} \gamma(\|\bar{v}\|_2)\}$, then $\mathcal{V}(z((k+1)\tau), \theta((k+1)\tau)) \leq \epsilon$. Moreover, $\mathcal{V}(P\theta((k+1)\tau), \theta((k+1)\tau)) \leq \mathcal{V}(z((k+1)\tau), \theta((k+1)\tau))$, since $\mathcal{V}(\bar{z}, \bar{\theta})$ is minimized when $\bar{z} = P\bar{\theta}$. ■

Lemma 5.2: Take $\epsilon \geq \max\{\mathcal{V}(z(k\tau), \theta(k\tau)), \sup_{\bar{v}\in V} \gamma(\|\bar{v}\|_2)\}$. If $a = a(u_\tau(\bar{v}, \theta(k\tau)))$ for some $\bar{v} \in V$ in the time interval $[k\tau, (k+1)\tau]$, then $\|z((k+1)\tau) - P\theta((k+1)\tau)\|_\infty \leq 2\epsilon$.

Proof: By Lemma 5.1, $\mathcal{V}\left(P\theta((k+1)\tau), \theta((k+1)\tau)\right) \leq \epsilon$, while by the definition of $\mathcal{V}$, $\|P\theta((k+1)\tau) - y((k+1)\tau)\|_2 \leq \mathcal{V}\left(P\theta((k+1)\tau), \theta((k+1)\tau)\right)$. Therefore, $\|P\theta((k+1)\tau) - y((k+1)\tau, a)\|_\infty \leq \epsilon$. Also, by Theorem 3.1, $\|z((k+1)\tau) - y((k+1)\tau)\|_\infty \leq \epsilon$. Therefore, $\|z((k+1)\tau) - P\theta((k+1)\tau)\|_\infty \leq 2\epsilon$. ■

Lemma 5.3: Set $z(0) = P\theta(0)$. If T is a set of ϵ-safe executions for some $\epsilon > 0$, and $\max\{\mathcal{V}(z(0), \theta(0)), \sup_{\bar{v}\in V} \gamma(\|\bar{v}\|_2)\} \leq \epsilon$, then σ has property (P.1).

Proof: Given $z(0) = P\theta(k\tau)$, let $q := \ell(z(0))$. Consider $a = a(u_\tau(\bar{v}, \theta(0))) \in \sigma(x(0))$ in the time interval $[0, \tau]$, and let $v_d \overset{z(0)}{\Leftrightarrow} \bar{v}$. By assumption, if $(q, v_d s) \in T$ for some s, then (q, v_d) is ϵ-safe. By Theorem 3.1, $\|z(0) + t\bar{v} - y(t, a)\|_2 \leq \max\{\mathcal{V}(z(0), \theta(0)), \sup_{\bar{v}\in V} \gamma(\|\bar{v}\|_2)\}$. This in turn implies that $\|z(0) + t\bar{v} - y(t, a)\|_2 \leq \epsilon$ and hence $\|z(0) + t\bar{v} - y(t, a)\|_\infty \leq \epsilon$ for all $t \in [0, \tau]$. Let $q' = \psi(q, v_d)$. Then, $\inf_{t\in[0,\tau]} \|t(q'-q)/\tau + q - (z(0) + t\bar{v})\|_\infty \leq \eta/2$, therefore $\|t(q'-q)/\tau + q - y(t, a)\|_\infty \leq \epsilon + \eta/2$, which implies that $y(t, a)$ is 0-safe in the time interval $[0, \tau]$. By Lemma 5.1, since $\max\{\mathcal{V}(z(0), \theta(0)), \sup_{\bar{v}\in V} \gamma(\|\bar{v}\|_2)\} \leq \epsilon$, setting $z(\tau) = P\theta(\tau)$, we have that $\max\{\mathcal{V}(y(\tau), \theta(\tau)), \sup_{\bar{v}\in V} \gamma(\|\bar{v}\|_2)\} \leq \epsilon$. Thus, we can repeat the reasoning above for each interval $[k\tau, (k+1)\tau]$, completing the proof. ■

Lemma 5.4: Set $z(0) = P\theta(0)$. If T is a set of forward-maximal executions 2ϵ-non-escaping for some $\epsilon > 0$, and $\max\{\mathcal{V}(z(0), \theta(0)), \sup_{\bar{v}\in V} \gamma(\|\bar{v}\|_2)\} \leq \epsilon$, then σ in (13) has property (P.2).

Proof: Given $x(0)$, assume that $\sigma(x(0)) \neq \emptyset$ and consider $a = a(u_\tau(\bar{v}, \theta(0))) \in \sigma(x(0))$ in the time interval $[0, \tau]$, for some $\bar{v} \in V$. Call $q := \ell(z(0))$ and $q' := \ell(P\theta(\tau))$, and let $v_d \overset{z(0)}{\Leftrightarrow} \bar{v}$. By Lemma 5.2, $\|z(0) + \tau\bar{v} - P\theta(\tau)\|_\infty \leq 2\epsilon$, hence $\|\ell(z(0) + \tau\bar{v}) - P\theta(\tau)\|_\infty \leq 2\epsilon + \eta/2$. Since T is 2ϵ-non-escaping and all its executions are forward-maximal, $(q', s) \in T$ for some sequence of events s, or $q' \notin Q$. In the first case, $\sigma(x(\tau, a)) \neq \emptyset$, while in the second case $y(\tau, a) \notin \hat{Y}$. Now, since $\mathcal{V}(z(\tau), \theta(\tau)) \leq \epsilon$ and hence $\max\{\mathcal{V}(z(\tau), \theta(\tau)), \sup_{\bar{v}\in V} \gamma(\|\bar{v}\|_2)\} \leq \epsilon$, we can repeat the reasoning above for each interval $[k\tau, (k+1)\tau]$, completing the proof. ■

Theorem 5.5: Given $z(0) = P\theta(0)$ and $\epsilon \geq \max\left\{\mathcal{V}(z(0), \sup_{\bar{v}\in V} \gamma(\|\bar{v}\|_2)\right\}$, if T is the largest 2ϵ-non-escaping set of ϵ-safe and forward-maximal executions of (12), then the supervisor (13) solves Problem 2.1.

Proof: The results follows from the above Lemmas. ■

The supervisor described above is also solution of Problem 2.2, as long as the derivatives of the output y are naught at time 0, and the input set V for (4) can be selected as a neighbourhood of the origin, as proved in the following theorem.

Theorem 5.6: Let $x(0)$ be such that all derivatives of $y(0)$ are naught, let the input set V of (4) be an open hypercube of side $2d$ centred on the origin, set $z(0) = P\theta(0)$, and take $\epsilon = \max\{\mathcal{V}(z(0), \theta(0)), \sup_{\bar{v}\in V} \gamma(\|\bar{v}\|_2)\}$. If T is the largest 2ϵ-non-escaping set of forward-maximal ϵ-safe executions, then the supervisor σ in (13) has property (P.3), and solves Problem 2.2.

Proof: Given the constant $d > 0$, fix $\eta < d$ and set

$$\tau := 2\frac{\|\sqrt{M}(BR - C^T)\|_2\sqrt{n}}{\lambda} + \frac{3\eta}{2d} + d \qquad (14)$$

so that, according to (11), $\tau = 2\gamma(\sqrt{n}d)/d + 3\eta/2d + d$. By assumption V is an open hypercube of side $2d$ centred on the origin, so that for all $\bar{v} \in V$, $\|\bar{v}\|_2 \leq \sqrt{n}d$. Since $x(0)$ is such that all derivatives of $y(0)$ are naught, we have that $\mathcal{V}(z(0), \theta(0)) = 0$, so that $\max\{\mathcal{V}(z(0), \theta(0)), \sup_{\bar{v}\in V} \gamma(\|\bar{v}\|_2)\} \leq \max\{\mathcal{V}(z(0), \theta(0)), \gamma(\sqrt{n}d)\} = \gamma(\sqrt{n}d)$, and $\epsilon \leq \gamma(\sqrt{n}d)$. Let V_r be an open hypercube of side $2(d - 2\epsilon/\tau - 3\eta/2\tau)$ centred on the origin. Given the value of τ assigned in (14) the above quantity is always positive, so $V_r \subset V$.

Assume at first that the arclength of $y^*([0, T])$ is finite. Since V_r is an open neighbourhood of the origin, we can construct a piecewise linear interpolation $z^*([0, m\tau], v^*)$ of $y^*([0, T])$ as a trajectory of (4) with input v^*, constant and equal to $\bar{v}^* \in V_r$ in intervals of length τ. The trajectory $z^*([0, m\tau], v^*)$ is taken so that it lies on $y^*([0, T])$ at each $t = k\tau$ with $k \in \{1, \ldots, m\}$, and such that $z^*(0, v^*) = y^*(0)$ and $z^*(m\tau, v^*) = y^*(T)$. Notice that, in general, $m\tau \neq T$. The infinity norm distance between two subsequent interpolation points is bounded by $d\tau$, and since by (14) τ is bounded as $d \to 0$, this distance goes to 0 with d. Thus, the interpolation error $e(d) := \mathcal{H}(y^*([0, T]), z^*([0, m\tau], v^*))$ converges to 0 as $d \to 0$. Additionally, both η and ϵ go to 0 with d, and for each d the execution $\left(\ell(z^*(0)), v_d^{*,1} \ldots v_d^{*,m-1}\right)$, with

$v_d^{*,k} \overset{z(k\tau)}{\Leftrightarrow} v^*(k\tau)$, is $(\delta - e(d))$-safe. Thus, for d small enough, the execution is ϵ-safe. We can extend the execution $\left(\ell(z^*(0)), v_d^{*,1} \dots v_d^{*,m-1}\right)$ by concatenating an infinite string of events $v_d^{*,k} = 0$, $k \geq m$, making the execution forward-maximal.

Now, consider an arbitrary step k, and consider the family of transitions starting from states $\ell(\bar{z})$, with $\bar{z} \in Z$ such that $\|\bar{z} - \ell(z^*(k\tau, v^*))\|_\infty \leq 2\epsilon + \eta/2$. We have that $\|z^*(k\tau, v^*) - z^*((k+1)\tau, v^*)\|_\infty \leq (d\tau - 2\epsilon - \eta)$, so that $\|\ell(z^*(k\tau, v^*)) - z^*((k+1)\tau, v^*)\|_\infty \leq (d\tau - 2\epsilon - \eta/2)$, and hence $\|\bar{z} - z^*((k+1)\tau, v^*)\|_\infty \leq d\tau$, so we can always choose a signal v constant and equal to $\bar{v}(\bar{z})$ in the interval $[k\tau, (k+1)\tau]$, such that $\bar{z} + \bar{v}(\bar{z})\tau = z^*((k+1)\tau, v^*)$. For d sufficiently small all transitions $(\ell(\bar{z}), v_d(\bar{z}))$ with $v_d(\bar{z}) \overset{\bar{z}}{\Leftrightarrow} \bar{v}(\bar{z})$ are ϵ-safe. Repeating this reasoning for all $k \geq 0$, we see that the set of executions $E := \{(\ell(\bar{z}(k\tau)), v_d^k(\bar{z}(k\tau)) v_d^{*,k+1} v_d^{*,k+2} \dots)\}$ for all $\bar{z}(k\tau)$ such that $\|\bar{z}(k\tau) - z(k\tau), v^*\|_\infty \leq \epsilon$ is a 2ϵ-non-escaping set of ϵ-safe and forward maximal executions. Since T is the largest 2ϵ-non-escaping set of ϵ-safe and forward maximal executions, $E \subset T$. The above reasoning applies to the case where $y^*([0,T])$ has infinite arclength, simply by taking an infinitely long interpolating curve $z^*([0,\infty], v^*)$.

To conclude, observe that at $t = 0$ we have $z(0) = y(0) = z^*(0)$, since all derivatives of $y(0)$ are naught. Hence, the supervisor admits the input $a = a\,(u_\tau(v^*(0), \theta(0)))$ in the interval $[0, \tau]$. By Lemma 5.2, $\|P\theta(\tau) - z^*(\tau, v^*)\|_\infty \leq 2\epsilon$. The above reasoning ensures that the supervisor admits an input $a = a\,(u_\tau(\bar{v}(P\theta(\tau)), \theta(0)))$ such that $P\theta(\tau) + \bar{v}(P\theta(\tau))\tau = z^*(2\tau)$. At $t = k\tau$ with $k > 1$, the process is repeated applying $a = a\,(u_\tau(\bar{v}(P\theta(k\tau)), \theta(k\tau)))$ in each interval $[k\tau, (k+1)\tau]$. We can now bound the distance of the trajectory $y(t, a)$ from the desired trajectory $y^*(t)$. From Lemma 5.2 we have that $\|P\theta(k\tau) - z^*(k\tau, v^*)\|_\infty \leq 2\epsilon$, and from Theorem 3.1 we know that $\|P\theta(k\tau) + \bar{v}(P\theta(k\tau)) - y(t, a)\|_\infty \leq \epsilon$. Then we defined $e(d) := \mathcal{H}(y^*([0,T]), z^*([0, m\tau], v^*))$. Therefore, $\mathcal{H}(y([0, m\tau], a), y^*([0,T])) \leq e(d) + 3\epsilon$, hence for d sufficiently small $\mathcal{H}(y([0, m\tau], a), y^*([0,T])) \leq \delta$. ∎

From the above theorems it follows that the supervisor σ is correct (Problem 2.1) irrespective of the input set V of (4), but it requires V to be a neighbourhood of the origin to be optimal in the limit (Problem 2.2). There are cases where V must not include the origin, for example to force dynamics towards a preferential direction (see, e.g., [10], [11]). In these cases the approximate nature of the bisimilarity between systems (4) and (3) imposes a lower bound on the guaranteed distance between the possible trajectories of (1) and the trajectories that are allowed by σ.

To construct the above defined set T, we need to extend the discrete event system $\bar{G}$ with a set of *uncontrollable* events taking each state q to all states q' such that there exists $y' \in \bar{Y}$ with $\ell(y') = q'$ and $\|y' - q\|_\infty \leq 2\epsilon + \eta/2$. The set T is found as the solution of the *Basic Supervisory Control Problem - Nonblocking Case* (BSCP-NB), for which standard algorithms are provided in the literature (see e.g., [9]).

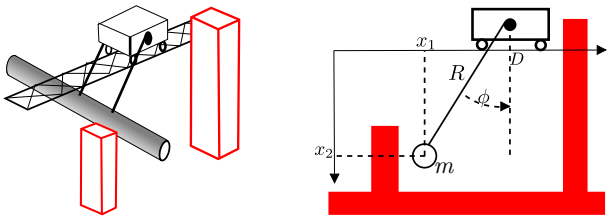

Fig. 1. Schematic representation of the crane model. The crane is supervised to avoid collisions between the mass and the obstacles (in red).

VI. Application

We have tested our algorithm by simulating its performance on the planar cantilever crane described in [15], assuming that the crane is supervised to avoid collisions between the mass and a number of obstacles, as depicted in Fig. 1. The laws of motion of the crane are described implicitly by the following set of differential equations and algebraic constraints:

$$\begin{cases} m\ddot{x}_1 &= -T\sin\phi \\ m\ddot{x}_2 &= -T\cos\phi + mg \\ M\ddot{x}_3 &= \mathcal{F} - \lambda\dot{x}_3 + T\sin\phi \\ \dfrac{J}{\rho^2}\ddot{x}_4 &= \mathcal{C} - \dfrac{\mu}{\rho}\dot{x}_4 - T\rho \\ x_1 &= x_4\sin\phi + x_3 \\ x_2 &= x_4\cos\phi. \end{cases}$$

Here, x_1 and x_2 are the coordinates of the mass, x_3 is the horizontal position of the trolley, x_4 is the length of the rope, T is the tension on the rope, ϕ is the angle of the rope with respect to the vertical axis, $\mathcal{F}$ and $\mathcal{C}$ are the control inputs, and correspond respectively to the force applied to the trolley, and to the torque applied on the rope's pulley, and $(g, J, \lambda, m, M, \mu, \rho)$ are constant parameters. The system is flat with flat output $y := (y_1, y_2) = (x_1, x_2)$. The corresponding "flattened" system is $y^{(4)} = u$, with $u \in U \subset \mathbb{R}^2$. Note that an abstraction based on a straightforward discretization of the full eight-dimensional state space would be computationally intractable, while our flatness-based approximate abstraction only requires the discretization of a two-dimensional set. We defined a supervisor with $\eta = 0.1m$, $\tau = 8s$, and V a hypercube centred on 0 with side $0.1m/s$. The simulation function and interface are defined by the matrices

$$M := \begin{bmatrix} 4.0316 & 5.5010 & 3.5679 & 0.5744 \\ 5.5010 & 10.1676 & 6.9904 & 1.2789 \\ 3.5679 & 6.9904 & 6.4655 & 1.3038 \\ 0.5744 & 1.2789 & 1.3038 & 0.4110 \end{bmatrix} \otimes \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix},$$

where $\otimes$ is the Kronecker product, $R := \begin{bmatrix} 1.3977 & 0 \\ 0 & 1.3977 \end{bmatrix}$, and

$$K := \begin{bmatrix} -8.5690 & -17.3220 & -19.9542 & -4.7978 \end{bmatrix} \otimes \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix},$$

which give $\gamma(v) = 3.9930v$. A sample trajectory of the supervised crane is depicted in Fig. 2. The control inputs $\mathcal{F}$

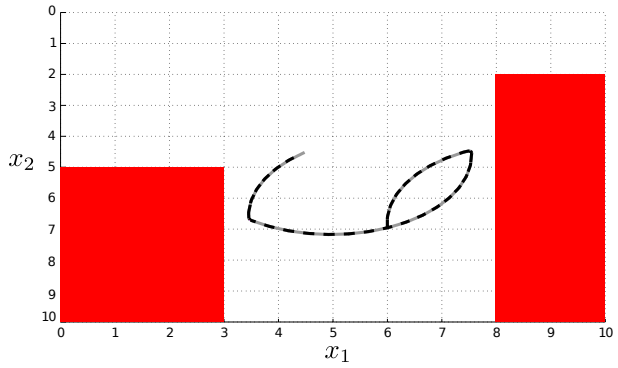

Fig. 2. Simulation of the supervised crane. The input is $v(k\tau) = 0.05[\sin(55k\tau), \cos(55k\tau)]^T$. The red rectangles are the obstacles, the dashed black line is $y(t)$ and the gray line is $z(t)$.

and $\mathcal{C}$ needed to drive the crane's mass along the trajectories in the figure depend on the plant's physical parameters. However, the quantities $\ddot{D}$ and $\ddot{R}$, which are respectively the acceleration of the trolley and of the rope, only depend on g, and thus provide an easy way to compare the performance of the above algorithm with alternative approaches. The maximum values of $\ddot{D}$ and $\ddot{R}$ obtained using the approach described above, when the input v is suddenly changed from $[0.05, 0.05]$ to $-[0.05, 0.05]$, are $\max|\ddot{R}| = 0.04m/s^2$, $\max|\ddot{D}| = 0.13m/s^2$. By comparison, the supervisor described in [10], [11], which forces the mass to follow polynomial trajectories interpolating the points $q \in \hat{Q}$ of a grid identical to the one defined in Section IV, gives $\max|\ddot{R}| = 0.1m/s^2$, $\max|\ddot{D}| = 0.15m/s^2$ under identical conditions. The milder input requirement is obtained at the expense of a greater minimal distance between the allowed trajectories and the obstacles.

VII. Conclusions

We have proposed an extension to the safety enforcing supervisory control algorithm for differentially flat systems originally discussed in [10], [11], obtained by exploiting results found in [12]. We began by considering the flat output dynamics, described by system (3). Following [12], we defined a simulation function and an associated interface, such that the trajectories of (3) with feedback control as in (7) are approximated by the trajectories of a first order integrator. The approximation error is bounded, the bound depending on the design parameters of the interface. Then, we defined a discrete abstraction of the first order integrator, and a safety-enforcing supervisor based on the abstraction. The resulting supervisor is correct, that is, it enforces safety and it is nonblocking. Note that safety is enforced at all times, and not just at the sampling times. Under additional assumptions the supervisor is also proved to be optimal, that is, to allow trajectories that follow any possible safe path, if the discretization step is taken sufficiently small.

By working on a finite abstraction, rather than on the continuous system, the computational complexity of the algorithm is reduced. The algorithm complexity scales exponentially with the dimension of the abstracted system's space, as in most discrete abstraction approaches [4], [5], [6], [8], but since the abstraction is defined on the simple dynamics of the first order integrator, its construction is not affected by the complexity of the nonlinear dynamics of the original system, and its size is proportional to the size of the output space of the system, rather than the size of the full state space. The main advantages with respect to the original algorithm found in [10], [11] are in the properties of the input set returned by the supervisor. Using approximate bisimulations, we were able to define a supervisor that allows inputs that change continuously within some bounded set, rather than restricting inputs to a discrete set of polynomial inputs as in the original algorithm. The new supervisor can thus be overlaid seamlessly to a pre-existing system as a safety-enforcing layer, without requiring to redesign the input set. Moreover the additional degrees of freedom provided by the design parameters of the interface allow for the tuning of the resulting supervisor, trading control effort for restrictiveness.

References

[1] R. Alur, T. Dang, and F. Ivancic, "Predicate abstraction for reachability analysis of hybrid systems," *ACM Trans. on Embedded Computing Systems*, vol. 5, pp. 152–199, 2006.

[2] E. Dallal, A. Colombo, D. Del Vecchio, and S. Lafortune, "Supervisory control for collision avoidance in vehicular networks using discrete event abstractions," *American Control Conference*, 2013.

[3] T. Moor and J. Raisch, *Modelling, analysis, and design of hybrid systems*. Springer-Verlag, 2002, ch. Abstraction based supervisory controller synthesis for high order monotone continuous systems, pp. 247–265.

[4] P. Tabuada, "An approximate simulation approach to symbolic control," *IEEE Trans. Autom. Control*, vol. 53, pp. 1406–1418, 2008.

[5] G. Pola, A. Girard, and P. Tabuada, "Approximately bisimilar symbolic models for nonlinear control systems," *Automatica*, vol. 44, pp. 2508–2516, 2008.

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

[7] 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.*, vol. 43, pp. 1923–1952, 2005.

[8] M. Zamani, G. Pola, M. Mazo Jr., and P. Tabuada, "Symbolic models for nonlinear control systems without stability assumptions," *IEEE Trans. Autom. Control*, in press.

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

[10] A. Colombo and D. Del Vecchio, "Enforcing safety of cyberphysical systems using flatness and abstraction," in *Proceedings of the Work-in-Progress session of ICCPS 2011*, 2011.

[11] ——, "Supervisory control of differentially flat systems based on abstraction," in *50th IEEE Conference on Decision and Control*, 2011.

[12] A. Girard and G. J. Pappas, "Hierarchical control system design using approximate simulation," *Automatica*, vol. 45, pp. 566–571, 2009.

[13] ——, "Approximate bisimulation relations for constrained linear systems," *Automatica*, vol. 43, pp. 1307–1317, 2007.

[14] J. Lygeros, C. Tomlin, and S. Sastry, "Controllers for reachability specifications for hybrid systems," *Automatica*, vol. 35, pp. 349–370, 1999.

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

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

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

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