Hierarchical Control using Approximate Simulation Relations

Antoine Girard and George J. Pappas

*Abstract***— In this paper we present a new approach for hierarchical control based on the recent notions of approximate simulation relations and simulation functions. Given a complex system that need to be controlled, approximate simulation relations allow to characterize a simple approximation of the system, that can be used for the control design. The controller of this approximate system can be lifted to the complex system using an interface which is completely characterized by a simulation function. Then, the distance between the external trajectories of the complex system and the external trajectories of the approximation is guaranteed to remain bounded by a precision that can be evaluated from the simulation function. This makes our approach suitable for safety critical systems. We show an application to robot motion control.**

I. Introduction

The design of controllers for complex nonlinear systems in order to achieve possibly complex behaviors is a very hard task that rapidly becomes intractable unless a hierarchical approach is used. The simplest hierarchical control system consists of two layers. The first layer consists of a coarse model of the system. A controller is designed so that the coarse model meets the specifications of the problem. Then, the control is lifted to the second layer consisting of a detailed model of the system. One of the main challenges of hierarchical control is to design the interface between the two layers of the system.

An approach using a hierarchy of consistent continuous abstractions of continuous systems has been proposed in [14], [19]. It is based on the notion of simulation relation, widely used in computer science for discrete systems [3], [12], and extended in several works to the continuous and hybrid settings [9], [13], [14], [17], [20].

Recently, the notion of approximate simulation relation has been introduced in [8]. This generalization is more appropriate for systems whose state-space is equipped with a natural topology such as continuous and hybrid systems. In this framework, it is not required that the trajectories of the system and of its abstraction match exactly but only approximately. Consequently, this relaxation enables larger simplifications [7]. The main concept of this approximation framework is the simulation function which defines effectively approximate simulation relations.

In this paper, we use this new framework in the context of hierarchical control. Given a complex system that need to be controlled, approximate simulation relations allow to characterize a simple approximation of the system, that can be used for the control design. The controller of this approximate system can be lifted to the complex system using an interface which is completely characterized by a simulation function. Then, the distance between the external trajectories of the complex system and the external trajectories of the approximation is guaranteed to remain bounded by a precision that can be evaluated from the simulation function.

The organization of the paper is the following. In Section II, we briefly review the approximation framework developed in [8]. In Section III, we show how it can be used to formally design hierarchical controllers. Finally, in Section IV, we present an application to robot motion control.

II. Approximate Simulation Relations

Let us consider two control systems:

$$\Sigma : \left\{ \begin{array}{rcl} \dot{x}(t) & = & f(x(t),u(t)), \\ y(t) & = & h(x(t)) \end{array} \right. \tag{1}$$

where $x(t) \in \mathbb{R}^n$, $x(0) \in I$, $u(t) \in U \subseteq \mathbb{R}^q$, $y(t) \in \mathbb{R}^k$ and

$$\Sigma' : \left\{ \begin{array}{rcl} \dot{z}(t) & = & g(z(t),v(t)), \\ y'(t) & = & k(z(t)) \end{array} \right. \tag{2}$$

where $z(t) \in \mathbb{R}^m$, $z(0) \in J$, $v(t) \in V \subseteq \mathbb{R}^p$, $y'(t) \in \mathbb{R}^k$. Note that both systems have the the same observation space (*i.e.* $\mathbb{R}^k$), but may have different input spaces.

A. From exact to approximate subsystems

We say that the control system Σ' is a *subsystem* of Σ if the external trajectories of Σ' form a subset of the external trajectories of Σ[1]. Let us remark that an external trajectory $y(t)$ can be obtained from Σ and Σ' for different inputs $u(t)$ and $v(t)$ (note that, in general, the sets of inputs U and V are different).

Further, Σ' is a *complete subsystem*, if we can relate any initial state x_0 of Σ to an initial state z_0 of Σ' such that for every state trajectory $z(t)$ of Σ' starting in z_0, there exists a state trajectory $x(t)$ of Σ starting in x_0 and satisfying $h(x(t)) = k(z(t))$, for all $t \geq 0$. Then, for all initial states of Σ, there exists an initial state of Σ' such that it is possible to make Σ behave like Σ'. Thus, hierarchical control of Σ can be achieved by designing a controller for Σ'.

This work was partially supported by the European Community project IST-2003-507219 PROSYD (Property-based System Design), the NSF Presidential Early CAREER (PECASE) Grant 0132716, and the ARO MURI MURI DAAD 19-02-01-0383.

Antoine Girard is with VERIMAG, 2 avenue de Vignate, 38610 Gières, France. Antoine.Girard@imag.fr

George J. Pappas is with the Department of Electrical and Systems Engineering, University of Pennsylvania, Philadelphia, PA 19104, USA. pappasg@seas.upenn.edu

[1] The term *subsystem* is borrowed from [16]. This notion is traditionally referred to as *refinement* in the computer science literature.

An effective way to characterize subsystems is to use the notion of simulation relation, traditionally defined for discrete systems [3], [12] and extended more recently to continuous and hybrid systems [9], [13], [14], [17], [20].

However, when considering continuous systems such as (1) whose state-space is equipped with a natural topology, this notion of *exact subsystem* is quite restrictive and may not allow significant complexity reduction in the control design. Weaker notions such as *approximate subsystems* seem more appropriate.

We will say that the control system Σ' is an *approximate subsystem of precision* δ of Σ if for every external trajectory $y'(t)$ of Σ', there exists an external trajectory $y(t)$ of Σ satisfying for all $t \geq 0$, $\|y(t) - y'(t)\| \leq \delta$. Similar to the exact case, we will say that Σ' is a *complete approximate subsystem of precision* δ, if we can relate any initial state x_0 of Σ to an initial state z_0 of Σ' such that for every trajectory $z(t)$ of Σ' starting in z_0, there exists a state trajectory $x(t)$ of Σ starting in x_0 and satisfying $\|h(x(t)) - k(z(t))\| \leq \delta$, for all $t \geq 0$.

In the following section, we show that approximate subsystems can be characterized using the notion of approximate simulation relation.

B. Approximate simulation relations

Approximate simulation relations have been introduced in [8], in the general framework of so-called metric transition systems. For control systems such as (1) and (2), approximate simulation relations can be defined as follows.

Definition 2.1: A relation $\mathcal{R} \subseteq \mathbb{R}^m \times \mathbb{R}^n$ is an approximate simulation relation of precision δ between Σ' and Σ if for all $(z_0, x_0) \in \mathcal{R}$,

1) $\|k(z_0) - h(x_0)\| \leq \delta$
2) For all state trajectory $z(t)$ of Σ' such that $z(0) = z_0$ there exists a state trajectory $x(t)$ of Σ such that $x(0) = x_0$ and satisfying

$$\forall t \geq 0,\ (z(t), x(t)) \in \mathcal{R}.$$

Let us remark that for $\delta = 0$, we recover the notion of *exact* simulation relation as defined in [13], [20].

The characterization of complete approximate subsystems by approximate simulation relations is given by the following straightforward result.

Proposition 2.2: Let $\mathcal{R} \subseteq \mathbb{R}^m \times \mathbb{R}^n$ be an approximate simulation relation of precision δ between Σ' and Σ. If for all $x_0 \in I$, there exists $z_0 \in J$ such that $(z_0, x_0) \in \mathcal{R}$ and conversely, then Σ' is a complete approximate subsystem of Σ of precision δ.

C. Simulation functions

The construction of approximate simulation relations can be done effectively using a class of functions called simulation functions [8]. Essentially, a simulation function between Σ' and Σ is a positive function bounding the distance between the observations and non-increasing under the parallel evolution of the systems.

Definition 2.3: A function $V : \mathbb{R}^m \times \mathbb{R}^n \to \mathbb{R}^+$ is a simulation function between Σ' and Σ if for all $(z, x) \in \mathbb{R}^m \times \mathbb{R}^n$,

$$V(z, x) \geq \|k(z) - h(x)\|^2, \tag{3}$$

$$\sup_{v \in V} \inf_{u \in U} \left(\frac{\partial V(z,x)}{\partial z} g(z, v) + \frac{\partial V(z,x)}{\partial x} f(x, u) \right) \leq 0\ . \tag{4}$$

Remark 2.4: There are similarities between the class of simulation functions and the class of robust control Lyapunov functions [6], [11]. However, there is an important conceptual difference. Considering the input v as a disturbance and the u as a control input, equation (4) is interpreted as follows: for any disturbance there exists a control such that the function V is non-increasing. In this setting, the control input u can be chosen with the knowledge of the disturbance v. In comparison, a robust control Lyapunov function requires the control input to be chosen independently of the disturbance (*i.e.* the order of the supremum and infimum in equation (4) is reversed) which is a more restrictive condition.

The definition of approximate simulation relations can be done by considering the level sets of a simulation function as stated in the following Theorem.

Theorem 2.5: Let V be a simulation function between Σ' and Σ. Then, for all $\delta \geq 0$, the relation defined by

$$\mathcal{R} = \left\{ (z, x) \in \mathbb{R}^m \times \mathbb{R}^n |\ V(z, x) \leq \delta^2 \right\} \tag{5}$$

is an approximate simulation relation of precision δ between Σ' and Σ.

Proof: Let $(z_0, x_0) \in \mathcal{R}$, then

$$\|k(z_0) - h(x_0)\| \leq \sqrt{V(z_0, x_0)} \leq \delta.$$

Moreover, let $z(t)$ be a state trajectory of Σ' such that $z(0) = z_0$ and $v(t)$ be the associated input. From equation (4) there exists $x(t)$ a state trajectory of Σ satisfying $x(0) = x_0$ and the associated input $u(t)$ such that for all $t \geq 0$,

$$\begin{aligned} \dot{V}(z(t), x(t)) \quad &= \quad \frac{\partial V(z(t), x(t))}{\partial z} g(z(t), v(t)) \\ &\quad + \frac{\partial V(z(t), x(t))}{\partial x} f(x(t), u(t)) \leq 0\ . \end{aligned}$$

Then, for all $t \geq 0$, $V(z(t), x(t)) \leq V(z_0, x_0) \leq \delta^2$ which is equivalent to say that $(z(t), x(t)) \in \mathcal{R}$. ■

The precision of the approximate subsystem of Σ can be evaluated by solving a static game involving a simulation function.

Corollary 2.6: Let V be a simulation function between Σ' and Σ. Let δ be defined by

$$\delta^2 = \max\left(\max_{x_0 \in I} \min_{z_0 \in J} V(z_0, x_0), \max_{z_0 \in J} \min_{x_0 \in I} V(z_0, x_0) \right). \tag{6}$$

Then, Σ' is a complete approximate subsystem of Σ of precision δ.

Proof: Straightforward from Proposition 2.2 and Theorem 2.5. ■

Methods for the computation of simulation functions have been presented for the class of constrained linear systems

[7] where approximate simulation relations have been used in the context of safety verification (*i.e.* the inputs u and v are considered as disturbances rather than controls) to approximate large systems by smaller or simpler ones with guaranteed error bounds.

In the following, we show how approximate subsystems can be used for hierarchical control.

III. Hierarchical Control using Approximate Subsystems

Approximate subsystems are useful for hierarchical control in that they can be used to simplify a control problem by separating the concerns due to the complexity of the system and to the complexity of the specified behavior that we want to achieve.

A. High-level synthesis for low-level specifications

Let us consider a complex, *low-level system* Σ, given by equation (1), that need to be controlled so that it meets some specifications. For simplicity, we will consider only two types of properties, namely *invariance* and *reachability*, but our approach extends to more complex specifications such as those expressed with some temporal logics (*e.g.* LTL [5], [10], [18]).

Definition 3.1: Let $\mathrm{Inv} \subseteq \mathbb{R}^k$, let $y(t)$ be an external trajectory of Σ. Then $y(t)$ satisfies the invariance property with respect to Inv if

$$\forall t \geq 0,\ y(t) \in \mathrm{Inv}. \tag{7}$$

Definition 3.2: Let $\mathrm{Target} \subseteq \mathbb{R}^k$, let $y(t)$ be an external trajectory of Σ. Then $y(t)$ satisfies the reachability property with respect to Target if

$$\exists t \geq 0,\ y(t) \in \mathrm{Target}. \tag{8}$$

Let x_0 be the initial state of the system Σ, our goal is to design a control input $u(t)$ such that the associated external trajectory $y(t)$ satisfies both the invariance and the reachability properties.

Let Σ' be a simple, *high-level system* that is a complete approximate subsystem of precision δ of Σ. For all subsets S of $\mathbb{R}^k$, let

$$\mathcal{I}(S,\delta) = \{s \in S |\ \|s - s'\| \leq \delta \implies s' \in S\}.$$

Let z_0 be the initial state of Σ' related to x_0. Let $v(t)$ be a control input for Σ' such that the associated external trajectory $y'(t)$ satisfies the invariance property with respect to $\mathcal{I}(\mathrm{Inv},\delta)$ and the reachability property with respect to $\mathcal{I}(\mathrm{Target},\delta)$. Since Σ' is an approximate subsystem of precision δ of Σ, there exists an input $u(t)$ for Σ such that the associated external trajectory $y(t)$ satisfies $\|y(t) - y'(t)\| \leq \delta$, for all $t \geq 0$. Then, it is clear that $y(t)$ satisfies the invariance property with respect to Inv and the reachability property with respect to Target.

Synthesizing the control input $v(t)$ for the high-level system Σ' is generally much easier than synthesizing the control input $u(t)$ for the low-level system Σ. In the following we show how $u(t)$ can be obtained from $v(t)$.

B. Effective synthesis of low-level inputs

Let us assume that there exists a simulation function V between Σ' and Σ such that $V(z_0,x_0) \leq \delta^2$. Then, the effective synthesis of the control input $u(t)$ is given by the following result:

Proposition 3.3: Let $u_V : V \times \mathbb{R}^m \times \mathbb{R}^n \to U$ be a continuous function such that for all $(v,z,x) \in V \times \mathbb{R}^m \times \mathbb{R}^n$,

$$\sup_{v\in V}\left(\frac{\partial V(z,x)}{\partial z} g(z,v) + \frac{\partial V(z,x)}{\partial x} f(x,u_V(v,z,x))\right) \leq 0 \tag{9}$$

then for any control input $v(t)$ and associated external trajectory $y'(t)$ of Σ', the external trajectory $y(t)$ of Σ given by

$$\begin{cases} \dot{x}(t) &= f(x(t), u_V(v(t),z(t),x(t))), \\ y(t) &= h(x(t)) \end{cases}$$

satisfies for all $t \geq 0$, $\|y(t) - y'(t)\|^2 \leq V(z_0,x_0)$. The function u_V is called an *interface* between Σ' and Σ.

Proof: From equation (9), we have that $\dot{V}(z(t),x(t)) \leq 0$. Then, from equation (3), for all $t \geq 0$,

$$\|y(t) - y'(t)\|^2 \leq V(z(t),x(t)) \leq V(z_0,x_0).$$

■

The architecture of the controller allowing to synthesize the control input $u(t)$ is shown on Figure 1. Note that the initial state z_0 of the high-level system must be chosen so that it satisfies $V(z_0,x_0) \leq \delta^2$. Therefore, it is not independent of the initial state x_0 of the low-level system.

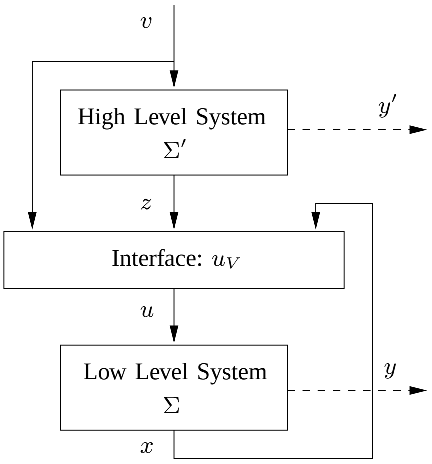

Fig. 1. Architecture of the controller

Remark 3.4: Our approach is related to more classical topics in control theory such as tracking control (see *e.g.* [15]). However, in our work, Σ does not need to be able to track any reference trajectory but only those associated with its approximate subsystem Σ'. Therefore, the tracking controller (*i.e.* the interface) can be designed with the knowledge of the dynamics generating the reference trajectory. Also related is the work on model matching [4]. The main difference is that, in our approach, we do not want to design a controller for Σ so that it matches exactly or asymptotically Σ'. The controller we design guarantees approximate matching with guaranteed and uniform (in the sense of the L^∞-norm) error bounds. This property makes our approach particularly suitable for control of safety critical systems.

IV. Application to Robot Motion Control

We consider a point like planar robot evolving in an environment Inv. Inv can be a complex, non convex set with several holes. The control problem is to drive the robot to a set Target while remaining in the environment Inv. This specification of this standard motion planning problem clearly consists in the conjunction of the invariance property with respect to Inv with the reachability property with respect to Target.

The low-level system Σ that need to be controlled consists of a dynamical model of the robot.

$$\Sigma : \left\{ \begin{array}{rcl} \dot{x}_1(t) & = & x_2(t) \\ \dot{x}_2(t) & = & u(t) \\ y(t) & = & x_1(t) \end{array} \right. \tag{10}$$

where $x_1(t) \in \mathbb{R}^2$, $x_2(t) \in \mathbb{R}^2$. Initially, $x_1(0) \in \mathrm{Init}$ and the velocity of the robot is zero (*i.e.* $x_2(0) = 0$). The input $u(t) \in \mathbb{R}^2$ is such that $\|u(t)\| \leq \mu$.

The high-level system Σ' that we want to use for control synthesis consists of a kinematic model of the robot.

$$\Sigma' : \left\{ \begin{array}{rcl} \dot{z}(t) & = & v(t) \\ y'(t) & = & z(t) \end{array} \right. \tag{11}$$

where $z(t) \in \mathbb{R}^2$, $z(0) \in \mathrm{Init}$. The input $v(t) \in \mathbb{R}^2$ is such that $\|v(t)\| \leq \nu$.

First, let us remark that Σ' is not an exact subsystem of Σ since all the external trajectories of Σ have their initial velocity equal to zero whereas this is clearly not the case of the external trajectories of Σ'. This motivates the use of the notion of approximate subsystem.

A. Approximate subsystem for robot motion control.

In the following, we show that under some assumptions on the parameters μ and ν, Σ' is a complete approximate subsystem of Σ.

Let us assume that the acceleration of the robot is possibly unbounded (*i.e.* $\mu = +\infty$). The bound will be reintroduced later. Then, we have the following result.

Proposition 4.1: Let $\mu = +\infty$, let $Q : \mathbb{R}^2 \times \mathbb{R}^4 \to \mathbb{R}^+$ be defined by

$$Q(z,x) = \frac{4}{3}\left(\|x_1 - z\|^2 + (x_1 - z)\cdot x_2 + \|x_2\|^2\right).$$

Then, $V(z,x) = \max(Q(z,x), 4\nu^2)$ is a simulation function between Σ' and Σ and an interface is given by the linear function

$$u_V(v,z,x) = \frac{v}{2} - x_2 + z - x_1.$$

Proof: Let us start by remarking that

$$\begin{array}{rcl} Q(z,x) & = & \|x_1 - z\|^2 + \\ & & \frac{1}{3}\left(\|x_1 - z\|^2 + 4(x_1 - z)\cdot x_2 + 4\|x_2\|^2\right), \\ & = & \|x_1 - z\|^2 + \frac{1}{3}\|x_1 - z + 2x_2\|^2. \end{array}$$

Then, $V(z,x) \geq Q(z,x) \geq \|x_1 - z\|^2$ and equation (3) is satisfied.

To prove that V is a simulation function between Σ' and Σ, it is sufficient to show that equation (9) holds since this clearly implies that equation (4) holds as well.

If $Q(z,x) \leq 4\nu^2$, then it is clear that equation (9) holds. If $Q(z,x) \geq 4\nu^2$, then

$$\begin{array}{rcl} \frac{\partial V}{\partial z}v + \frac{\partial V}{\partial x_1}x_2 + \frac{\partial V}{\partial x_2}u_V & = & \frac{4}{3}[(-2x_1 + 2z - x_2)\cdot v + \\ & & (2x_1 - 2z + x_2)\cdot x_2 + \\ & & (x_1 - z + 2x_2)\cdot u_V]. \end{array}$$

After the substitution of the expression of u_V and simplification, we arrive to

$$\begin{array}{rcl} \frac{\partial V}{\partial z}v + \frac{\partial V}{\partial x_1}x_2 + \frac{\partial V}{\partial x_2}u_V & = & -Q(z,x) - 2(x_1 - z)\cdot v, \\ & \leq & -Q(z,x) + 2\nu\|x_1 - z\| \end{array}$$

because $\|v\| \leq \nu$. Since $\|x_1 - z\|^2 \leq Q(z,x)$, we have

$$\begin{array}{rcl} \frac{\partial V}{\partial z}v + \frac{\partial V}{\partial x_1}x_2 + \frac{\partial V}{\partial x_2}u_V & \leq & -Q(z,x) + 2\nu\sqrt{Q(z,x)}, \\ & \leq & \sqrt{Q(z,x)}(2\nu - \sqrt{Q(z,x)}). \end{array}$$

Since $Q(z,x) \geq 4\nu^2$, we have

$$\frac{\partial V}{\partial z}v + \frac{\partial V}{\partial x_1}x_2 + \frac{\partial V}{\partial x_2}u_V \leq 0.$$

Then, equation (9) holds. ■

Let us remark that the initial state $z(0)$ of the approximate subsystem can chosen such that $z(0) = x_1(0)$. Then, we have $V(z(0), x(0)) = 4\nu^2$ and from Corollary 2.6, we have the following result.

Corollary 4.2: Let $\mu = +\infty$, then Σ' is a complete approximate subsystem of precision 2ν of Σ.

Thus, it is possible to make the approximate subsystem as precise as desired by choosing the velocity bound ν small enough. The implementation of the controller that makes Σ behave like its approximate subsystem Σ' is very simple since the interface u_V is a linear function.

Generally, the robot has hard constraints on its acceleration (*i.e.* $\mu < +\infty$). Then, we can show that the result of Corollary 4.2 still holds, provided the bound μ is large enough.

Theorem 4.3: Let $\mu \geq (\sqrt{6}+1/2)\nu$, then Σ' is a complete approximate subsystem of precision 2ν of Σ.

Proof: It is clear that it is sufficient to show that for all (z,x) such that $V(z,x) \leq 4\nu^2$, $\|u_V(v,z,x)\| \leq \mu$. First, let us remark that

$$\|u_V(v,z,x)\| \leq \nu/2 + \|-x_2 + z - x_1\|.$$

Moreover, it is easy to show that

$$Q(z,x) = \frac{2}{3}\left(\|x_1 - z + x_2\|^2 + \|x_1 - z\|^2 + \|x_2\|^2\right)$$

which implies that

$$\|-x_2 + z - x_1\|^2 \leq \frac{3}{2}Q(z,x) \leq \frac{3}{2}V(z,x) \leq 6\nu^2.$$

Therefore, $\|u_V(v,z,x)\| \leq (\sqrt{6}+1/2)\nu \leq \mu$. ■

B. Example

We now consider a practical motion planning problem. The environment Inv where the robot evolves can be seen on Figure 2. It consists of a corridor of width 1. At the end of the corridor, there is a room with an obstacle. The Target, a circle of diameter 1, is behind the obstacle.

The velocity bound ν is chosen equal to 0.25 and the accelaration bound μ is assumed large enough (*e.g.* $\mu = 3\nu = 0.75$). Then, from Theorem 4.3, the precision of Σ', the approximate subsystem of Σ is $\delta = 0.5$.

We can design easily a path for Σ' consisting of line segments along which the approximate system evolves at the constant speed ν. The input synthesized for Σ' is lifted to Σ using the controller shown on Figure 1.

On Figure 2, the external trajectory $y'(t)$ (dashed, red) of the approximate subsystem Σ' satisfies the invariance property with respect to $\mathcal{I}(\mathrm{Inv}, \delta)$ and the reachability with respect to $\mathcal{I}(\mathrm{Target}, \delta)$. Then, from the results obtained in section III, it is guaranteed that the external trajectory $y(t)$ (plain, blue) of the controlled system Σ satisfies the invariance property with respect to Inv and the reachability property with respect to Target. This example illustrates how the motion planning problem can be solved very easily using the approximate subsystem Σ'.

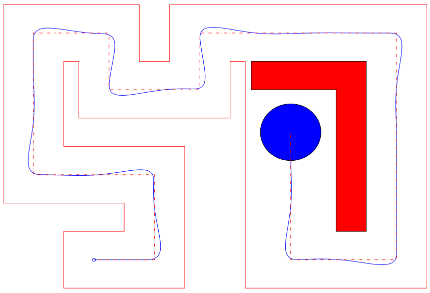

Fig. 2. Trajectories of the system Σ (plain, blue) and of its approximate subsystem Σ' (dashed, red). The trajectory of Σ' satisfies the invariance property with respect to $\mathcal{I}(\mathrm{Inv}, \delta)$ and the reachability with respect to $\mathcal{I}(\mathrm{Target}, \delta)$. This insures that the trajectory of Σ satisfies the invariance property with respect to Inv and the reachability property with respect to Target.

On Figure 3, we represented the evolution of the function $\sqrt{Q(z(t), x(t))}$ for the trajectories of Σ and Σ' represented on Figure 2. Particularly, the value of this function provide a bound on the distance between $y(t)$ and $y'(t)$. We can check that

$$\text{for all } t \geq 0, \ \sqrt{Q(z(t), x(t))} \leq 0.5,$$

which is expected because the precision of the approximate subsystem is 0.5.

Let us remark that the function $\sqrt{Q(z(t), x(t))}$ has twelve local maxima corresponding to the initial phase when the robot starts to move and the eleven changes of direction in the trajectory of the approximate subsystem Σ'. During the period where the trajectory of Σ' is a straight line, it seems that the function stabilizes around 0.25.

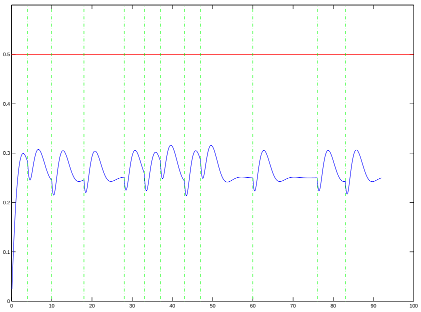

Fig. 3. Value of the function $\sqrt{Q(z(t), x(t))}$ for the trajectories of Σ and Σ' presented on Figure 2. The vertical lines correspond to the times at which the direction of the trajectory of Σ' changes.

Figure 4 is an illustration of what may happen when the motion of the approximate subsystem Σ' is designed independantly of its precision. In this example, the external trajectory $y'(t)$ (dashed, red) of Σ' satisfies the invariance property with respect to Inv and the reachability property with respect to Target. However, it does not satisfy the invariance property with respect to $\mathcal{I}(\mathrm{Inv}, \delta)$. We can see that the external trajectory $y(t)$ (plain, blue) of Σ does not satisfy the invariance property with respect to Inv.

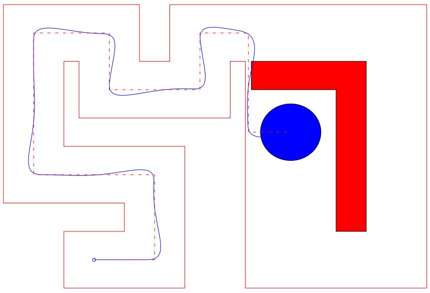

Fig. 4. Trajectories of the system Σ (plain, blue) and of its approximate subsystem Σ' (dashed, red). The trajectory of Σ' does not satisfy the invariance property with respect to $\mathcal{I}(\mathrm{Inv}, \delta)$. Then, the trajectory of Σ does not satisfy the invariance property with respect to Inv.

C. Hierarchical architecture of autonomous robots

Often, we are interested in designing a closed loop controller rather than an open loop controller such as those considered in the previous section. Indeed, a closed loop controller can be used for infinitely many initial states and allows to implement autonomous robots.

The design of control laws for kinematic models such as Σ' is quite well developped. Purely continuous approaches such as those based on potential or navigation functions (see *e.g.* [2]) allows to solve efficiently control problems with invariance and reachability specifications. More recently, methods based on discrete abstractions of Σ' have been proposed [1], [5]. Then, using algorithms for discrete systems (*e.g.* algorithms from model checking [5]), these discrete abstractions allow the design of hybrid controllers for Σ' even for complex specifications such as those expressed in LTL.

Then, the controller designed for the approximate subsystem Σ' can be lifted to Σ the dynamic model of the robot using the hierarchical closed loop controller presented on Figure 5. Then, it is guaranteed that the distance between $y'(t)$ and $y(t)$, the external trajectories of Σ' and of Σ, will be bounded by the precision of Σ'.

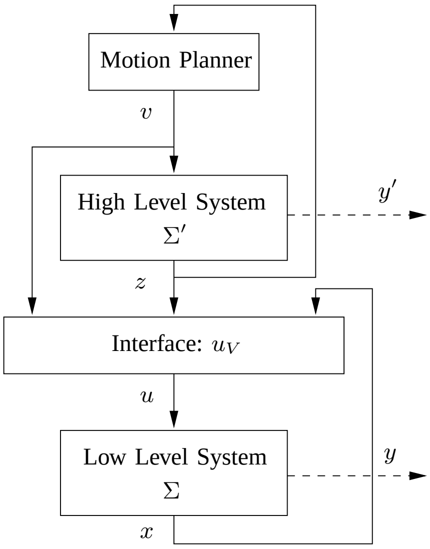

Fig. 5. Hierarchical architecture of the autonomous robot

For illustration, let us consider the example where Σ' is driven by the normalized van der Pol equations:

$$\dot{z}(t) = \frac{f(z(t))}{2\|f(z(t))\|},$$

$$\text{where } f(z_1, z_2) = \begin{bmatrix} z_2 \\ z_2(1 - z_1^2) - z_1 \end{bmatrix}.$$

The velocity bound in the approximate subsystem Σ' given by equation (11) can be chosen equal to 0.5. Then, from Theorem 4.3, the precision of Σ' is bounded by 1. On Figure 6, we represented some trajectories (dashed, red) of the approximate subsystem Σ' and the corresponding trajectories (plain, blue) of system Σ driven by the hierarchical controller of Figure 5.

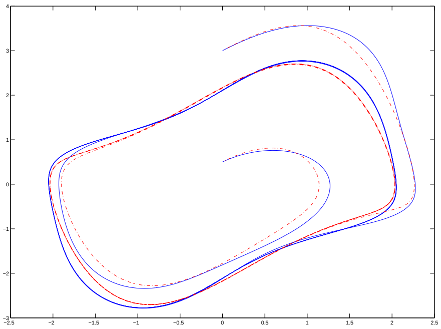

Fig. 6. Examples of trajectories of the system Σ (plain, blue) and of its approximate subsystem Σ' (dashed, red). Σ' is driven by the van der Pol equations.

We can see that our approach allows to lift a possibly complex controller designed for the simple approximate subsystem in order to control the more complex system Σ.

V. Conclusion

In this paper, we presented an approach for hierarchical control based on the notion of approximate subsystem. It allows to use a simplified model of a complex system for control design. The resulting controller is easily lifted to the complex system using an interface given by a simulation function with guaranteed error bounds.

Future work includes the development of algorithmic methods for the computation of simulation functions and of the associated interfaces. An interesting application will consists in coupling our approach with the methods for the design of hybrid controllers [1], [5] for kinematic models in order to control complex systems so that they satisfy specifications expressed in temporal logics.

References

[1] C. Belta, V. Isler, and G. J. Pappas, “Discrete abstractions for robot motion planning and control in polygonal environments,” *IEEE Trans. on Robotics*, vol. 21, no. 5, pp. 864–874, 2005.

[2] H. Choset, K. Lynch, S. Hutchinson, G. Kantor, W. Burgard, L. Kavraki, and S. Thrun, *Principles of Robot Motion: Theory, Algorithms and Implementations*. MIT Press, 2005.

[3] E. M. Clarke, O. Grumberg, and D. A. Peled, *Model Checking*. MIT Press, 2000.

[4] M. Di Benedetto and J. Grizzle, “Asymptotic model matching for nonlinear systems,” *IEEE Trans. on Auto. Cont.*, vol. 39, no. 8, pp. 1539–1549, 1994.

[5] G. E. Fianekos, H. Kress-Gazit, and G. J. Pappas, “Hybrid controllers for path planning: A temporal logic approach.” in *44th IEEE CDC and ECC*, 2005.

[6] R. A. Freeman and P. V. Kokotovic, “Inverse optimality in robust stabilization,” *SIAM J. Cont. and Opt.*, vol. 34, no. 4, pp. 1365–1391, 1996.

[7] A. Girard and G. J. Pappas, “Approximate bisimulations for constrained linear systems.” in *44th IEEE CDC and ECC*, 2005.

[8] ——, “Approximation metrics for discrete and continuous systems,” *IEEE Trans. Auto. Cont.*, 2005, accepted.

[9] E. Haghverdi, P. Tabuada, and G. J. Pappas, “Bisimulation relations for dynamical, control, and hybrid systems,” *Theo. Comp. Sc.*, vol. 342, pp. 229–261, 2005.

[10] M. Kloetzer and C. Belta, “A fully automated framework for control of linear systems from LTL specifications,” in *Hybrid Systems: Computation and Control*, ser. LNCS. Springer, 2006.

[11] D. Liberzon, E. D. Sontag, and Y. Wang, “Universal construction of feedback laws achieving ISS and integral-ISS disturbance attenuation,” *Sys. and Cont. Letters*, vol. 46, pp. 111–127, 2002.

[12] R. Milner, *Communication and Concurrency*. Prentice-Hall, 1989.

[13] G. J. Pappas, “Bisimilar linear systems,” *Automatica*, vol. 39, no. 12, pp. 2035–2047, 2003.

[14] G. J. Pappas, G. Lafferriere, and S. Sastry, “Hierarchically consistent control systems,” *IEEE Trans. on Auto. Cont.*, vol. 45, no. 6, pp. 1144–1160, 2000.

[15] S. Sastry, *Nonlinear Systems: Analysis, Stability and Control*. Springer, 1999.

[16] P. Tabuada, “Symbolic sub-systems and symbolic control of linear systems,” in *44th IEEE CDC and ECC*, 2005.

[17] P. Tabuada and G. J. Pappas, “Bisimilar control affine systems,” *Sys. and Cont. Letters*, vol. 52, no. 1, pp. 49–58, 2004.

[18] ——, “Linear time logic control of discrete time linear systems,” *IEEE Trans. on Auto. Cont.*, 2004, accepted.

[19] ——, “Hierarchical trajectory generation for a class of nonlinear systems,” *Automatica*, vol. 41, no. 4, pp. 701–708, 2005.

[20] A. van der Schaft, “Equivalence of dynamical systems by bisimulation,” *IEEE Trans. Auto. Cont.*, vol. 49, pp. 2160–2172, 2004.