HAL
archives-ouvertes.fr

Formal Controller Synthesis from Hybrid Programs

Vladimir Sinyakov, Antoine Girard

▶ **To cite this version:**

Vladimir Sinyakov, Antoine Girard. Formal Controller Synthesis from Hybrid Programs. 21st International Conference on Hybrid Systems: Computation and Control, Apr 2018, Porto, Portugal. pp.271-272, 10.1145/3178126.3186998 . hal-02053179

HAL Id: hal-02053179
https://hal.archives-ouvertes.fr/hal-02053179

Submitted on 1 Mar 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Poster: Formal Controller Synthesis from Hybrid Programs

Vladimir Sinyakov
Laboratoire des Signaux et Systèmes (L2S), CNRS,
CentraleSupélec, Université Paris-Sud,
Université Paris-Saclay
Gif-sur-Yvette, France
Vladimir.Sinyakov@l2s.centralesupelec.fr

Antoine Girard
Laboratoire des Signaux et Systèmes (L2S), CNRS,
CentraleSupélec, Université Paris-Sud,
Université Paris-Saclay
Gif-sur-Yvette, France
Antoine.Girard@l2s.centralesupelec.fr

ABSTRACT

We consider a new way of describing complex control problems for dynamic systems called *hybrid programs*. Hybrid program is a finite state automaton whose states describe elementary tasks of reachability and safety defined on a transition system ([3, 5]). The proposed approach to complex control problems description could be seen as an alternative to linear temporal logic (see e.g. [1]). We provide an example to illustrate the approach.

ACM Reference Format:
Vladimir Sinyakov and Antoine Girard. 2018. Poster: Formal Controller Synthesis from Hybrid Programs. In *HSCC '18: 21st International Conference on Hybrid Systems: Computation and Control (part of CPS Week), April 11–13, 2018, Porto, Portugal.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3178126.3186998

1 HYBRID TASKS

In our work we consider the dynamics described by transition systems.

Definition 1. *A transition system S is a tuple (X, U, Y, Δ, X^0) where X is a set of states; U is a set of inputs; Y is a set of outputs; Δ is a transition relation: $\Delta \subseteq X \times U \times X \times Y$; $X^0 \subseteq X$ is a set of initial states.*

Here we assume that $y \in Y$ is a function from $C([0, \theta_y], \mathbb{R}^n)$, $\theta_y > 0$. An input $u \in U$ is *enabled* at state x ($u \in \mathrm{enab}(x)$) if $\Delta(x, u) \neq \emptyset$. The control goal is formally described by a *hybrid program*, the notion of which is introduced in the next section. Hybrid programs consist of elementary hybrid tasks.

Definition 2. *An elementary hybrid task is a tuple $T = (Q, E, I, G, Q^0, Q^1, s)$ consisting of a finite set of states Q; a transition relation $E \subseteq Q \times Q$; a set of invariants $I = \{I_q \subseteq \mathbb{R}^n,\ q \in Q\}$; a set of guards $G = \{G_e \subseteq \mathbb{R}^n,\ e \in E\}$; a set of initial states $Q^0 \subseteq Q$; a set of terminal states $Q^1 \subseteq Q$; a task semantic $s \in$ {"rechability", "safety"}.*

We use elementary hybrid tasks to compose transition systems $S||T$ which are derived from the definition of hybrid automata [4] and satisfy the corresponding safety and reachability requirements.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
HSCC '18, April 11–13, 2018, Porto, Portugal
© 2018 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-5642-8/18/04.
https://doi.org/10.1145/3178126.3186998

Definition 3. *The composition of $S = (X, U, Y, \Delta, X^0)$ with $T = (Q, E, I, G, Q^0, Q^1, s)$ is the transition system $S||T = (X_T, U_T, Y, \Delta_T, X_T^0, X_T^1, s)$ where the set of states is $X_T = Q \times X$; the set of inputs is $U_T = Q^{X \times Y} \times U$; the set of outputs is Y; the set of initial states is $X_T^0 = Q^0 \times X^0$; the set of terminal states is X_T^1. A control $v = (p, u) \in U_T$ is enabled in the state z ($v \in \mathrm{enab}(z)$) if and only if $u \in \mathrm{enab}(x)$ and for all $(x', y) \in \Delta(x, u)$ one of the following conditions holds:*

- *$q = p(x', y)$ and $\forall t \in \mathscr{T}_y$, $y(t) \in I_q$;*
- *there exist $t_i \in \mathscr{T}_y$, $i = 0, \dots, N+1$ with $0 = t_0 \leq t_1 \leq \dots \leq t_{N+1} = \theta_y$ (or $t_{N+1} = N_y$ in the discrete case) and $e_i = (q_i, q_{i+1}) \in E$, $i = 0, \dots, N-1$ with $q_0 = q$, $q_N = p(x', y)$ and*

$$\begin{cases} \forall t \in [t_i, t_{i+1}) \cap \mathscr{T}_y,\ y(t) \in I_{q_i}, & i = 0, \dots, N; \\ y(t_{i+1}) \in G_{e_i}, & i = 0, \dots, N-1. \end{cases}$$

The transition relation is given for $z = (q, x)$, $z' = (q', x') \in X_T$, $v = (p, u) \in U_T$, $y \in Y$ by $(z', y) \in \Delta_T(z, v)$ if and only if $v \in \mathrm{enab}(z)$, $(x', y) \in \Delta(x, u)$, $q' = p(x', y)$.

The set X_T^1 is defined as follows

$$X_T^1 = \Big\{ z = (q, x) \,\Big|\, q \in Q^1 \quad \text{and} \quad \mathrm{enab}(x) \neq \emptyset, \\ \forall u \in \mathrm{enab}(x),\ \forall (x', y) \in \Delta(x, u) \quad \Rightarrow \quad y(0) \in I_q \Big\}.$$

A controller C for the transition system $S||T$ is defined as a subset of $X_T \times U_T$ such that $C(z) \subseteq \mathrm{enab}(z)$. We denote the domain of the controller C as $\mathrm{dom}(C) = \{z \in X_T |\ C(z) \neq \emptyset\}$. The controlled system is given by the transition system $S||T/C = (X_T, U_T, Y, \Delta_{T/C}, X_{T/C}^0, s)$ where the transition relation is given for $z \in X_T$, $v \in U_T$ by $(z', y) \in \Delta_{T/C}(z, v)$ if and only if $v \in C(z)$ and $(z', y) \in \Delta_T(z, v)$; the set of initial states is $X_{T/C}^0 = X_T^0 \cap \mathrm{dom}(C)$.

Next, for each type of tasks we give the definition of the corresponding controller.

Definition 4. *For $S||T = (X_T, U_T, Y, \Delta_T, X_T^0, X_T^1, s)$ with safety semantics s the corresponding controller is a controller such that all maximal trajectories of $S||T/C$ are either infinite or have the form $\{(z_k, v_k, y_k)\}_{k=0}^N$ with $z_N \in X_T^1$.*

Definition 5. *For $S||T = (X_T, U_T, Y, \Delta_T, X_T^0, X_T^1, s)$ with reachability semantics s the corresponding controller is a controller such that all maximal trajectories of $S||T/C$ have the form $\{(z_k, v_k, y_k)\}_{k=0}^N$ with $z_N \in X_T^1$.*

Now we are ready to define hybrid programs and controllers for them.

2 HYBRID PROGRAMS

Definition 6. *A hybrid program* $\mathbb{T}$ *is a tuple* $(\mathcal{T}, \mathcal{E}, \mathcal{R}, \mathcal{T}^0)$ *where* $\mathcal{T}$ *is a finite set of tasks;* $\mathcal{T}^0 \subseteq \mathcal{T}$ *is a set of initial tasks;* $\mathcal{E} \subseteq \mathcal{T} \times \mathcal{T}$ *is a set of transitions;* $\mathcal{R} = \{R_e,\ e \in \mathcal{E}\}$ *is a set of transition relations*

$$R_e \subseteq Q_T^1 \times Q_{T'}^0, \quad \text{for} \quad e = (T, T') \in \mathcal{E}.$$

We also consider the definition of the composition $S||\mathbb{T}$ of the symbolic transition system S and the hybrid program $\mathbb{T}$. Then we use the sets $J_e = \{q|\ R_e(q) \neq \emptyset\}$, $e \in \mathcal{E}$ and $J_T = \bigcup_{T'}\{J_e|\ e = (T, T')\}$ to define a controller for the hybrid program $\mathbb{T}$.

Definition 7. *A controller* C *for the hybrid program* $\mathbb{T}$ *is a set of controllers* $\{C_T,\ T \in \mathcal{T}\}$ *such that*

- C_T *is a controller which solves the task* T*;*
- *For all* $T \in \mathcal{T}$ *for all states* $z = (q, x) \in X_T^1$ *reachable with* C_T *the following property holds: if* $q \in J_T$ *then there exists* $q' \in Q_{T'}^0$ *such that* $q' \in R_e(q)$, $(q', x) \in X_{T'}^0$ *and* $C_{T'}(q', x) \neq \emptyset$ *for some* $e = (T, T')$, $q \in J_e$.

To compute the hybrid program controllers we utilize an algorithm which iteratively updates the individual task controllers ([2, 5]). Algorithm 1 terminates when a fixed point is reached or, in other words, when the maximal safety set of the hybrid program is found.

Algorithm 1: Hybrid program controller

Input: Transition system S, hybrid program $\mathbb{T}$
Output: Controller $C_{\mathbb{T}} = \{C_T,\ T \in \mathcal{T}\}$
Global variables : Value functions V_T, terminal sets X_T^1
Local variables : *controllerUpdated*, *blockingTerminalState*
begin
  controllerUpdated := *true*;
  while *controllerUpdated* **do**
    controllerUpdated := *false*;
    foreach $T \in \mathcal{T}$ **do**
      foreach *reachable* $z = (q, x) \in X_T^1 \cap (J_T \times X)$ **do**
        blockingTerminalState := *true*;
        foreach T' *s.t.* $q \in J_{(T,T')}$ **do**
          if $\left(R_e(q) \times \{x\}\right) \cap X_{T'}^0 \cap \mathrm{dom} C_{T'} \neq \emptyset$ **then**
            blockingTerminalState := *false*;
        if *blockingTerminalState* **then**
          $X_T^1 := X_T^1 \setminus \{z\}$;
          C_T = UpdateController($S||T$, $\{z\}$);
          controllerUpdated := *true*;
  return C;

We then prove correctness of the corresponding *UpdateController* algorithms (for safety and for reachability) using dynamic programming techniques.

3 ILLUSTRATIVE EXAMPLE

Finally, we apply this theory to formulate and solve a control synthesis example. Let us consider the following two-dimensional system

$$\begin{cases} \dot{x}_1 &= 2u_1, \\ \dot{x}_2 &= u_2 + 3vh(x_2 - x_1)u_0 \end{cases}$$

where $u \in [-1, 1]^2$ is the control, $v \in [0, 1]$ is the disturbance, $h(x) = \max\{0, \max\{1, x\}\}$. The state space constraint here is $x \in Q = [0, 10]^2 \setminus [4.44, 7.77] \times [3.33, 4.44]$. The control problem is described by a hybrid program presented on Fig. 2. The safety task requires the system to stay in Q_1 or transition to the reachability task which is to visit Q_2, Q_3, Q_1 (in this particular order) and to transition back to safety task. Here $Q_1 = [0, 1.11] \times [0, 2.22]$, $Q_2 = [5.55, 6.66] \times [0, 2.22]$, $Q_3 = [7.77, 10] \times [6.66, 7.77]$. We compute the hybrid program controller using Algorithm 1. The simulated trajectory with this controller is depicted on Fig. 1.

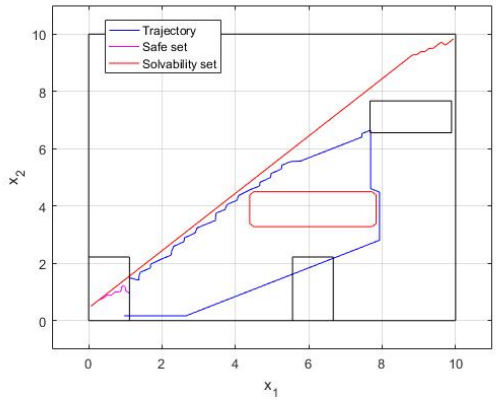

Figure 1: Simulation of the example's system trajectory with the computed controller.

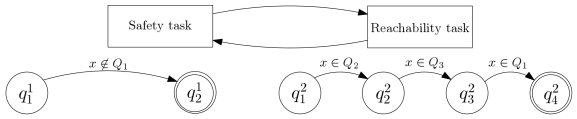

Figure 2: Visual representation of hybrid tasks in the example.

ACKNOWLEDGMENTS

This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No 725144).

REFERENCES

[1] Calin Belta, Boyan Yordanov, and Ebru Aydin Gol. 2017. *Formal Methods for Discrete-Time Dynamical Systems.* Springer.
[2] Antoine Girard. 2012. Controller synthesis for safety and reachability via approximate bisimulation. *Automatica* 48, 5 (2012), 947–953.
[3] Antoine Girard, Gregor Gössler, and Sebti Mouelhi. 2016. Safety controller synthesis for incrementally stable switched systems using multiscale symbolic models. *IEEE Trans. Automat. Control* 61, 6 (2016), 1537–1549.
[4] John Lygeros, Karl Henrik Johansson, Slobodan N. Simić, Jun Zhang, and S. Shankar Sastry. 2003. Dynamical Properties of Hybrid Automata. *IEEE Trans. Automat. Control* 48, 1 (2003), 2–17.
[5] Paulo Tabuada. 2008. *Verification and Control of Hybrid Systems: a symbolic approach.* Springer.