A Policy Iteration Technique for Time Elapse over Template Polyhedra

(Extended Abstract)

Sriram Sankaranarayanan[1], Thao Dang[2], and Franjo Ivančić[1]

[1] NEC Laboratories America, Princeton, NJ, USA
[2] Verimag, Grenoble, France
{srirams,ivancic}@nec-labs.com, thao.dang@imag.fr

Abstract. We present a technique to compute over-approximations of the time trajectories of an affine hybrid system using template polyhedra. Such polyhedra are obtained by conjoining a set of inequality templates with varying constant coefficients. Given a set of template expressions, we show the existence of a smallest template polyhedron that is a positive invariant w.r.t to the dynamics of the continuous variables, and hence, an over-approximation of the time trajectories. However, the least invariant is hard to compute efficiently. Therefore, we propose a policy iteration technique that iterates over the space of invariant certificates to converge onto a solution that is close to the least solution. We incorporate our ideas in our prototype tool TimePass for safety verification of affine hybrid systems, with promising results on benchmarks.

1 Introduction

The time elapse operator over-approximates the continuous state evolution inside each discrete mode of a hybrid system. In this paper, we investigate the computation of the time elapse over *template polyhedra*. A *template* is a set $H = \{h_1(\boldsymbol{x}), \ldots, h_m(\boldsymbol{x})\}$ of linear expressions over $\boldsymbol{x}$, represented as an $m \times n$ matrix H. Given a template, a family of template polyhedra is obtained by considering conjunctions of the form $\bigwedge_i h_i(\boldsymbol{x}) \leq c_i$.

Definition 1 (Template Polyhedron). *A* template polyhedron *over a template H is a polyhedron of the form $H\boldsymbol{x} \leq \boldsymbol{c}$, wherein $\boldsymbol{c} \in \mathcal{R}_+^m$. Such a polyhedron will be represented as $\langle H, \boldsymbol{c}\rangle$. Further properties of template polyhedra are presented in our previous work [4].*

An instance of the *time elapse* problem consists of an initial region $\langle H, \boldsymbol{c}_0\rangle$, a location invariant $\langle H, \mathsf{inv}\rangle$, and the vector field $\mathbf{D} : \dot{\boldsymbol{x}}_i = f_i(\boldsymbol{x})$ specifying the dynamics of each state variable $\boldsymbol{x}_i$. We assume that $\mathbf{D}$ is an affine vector field. The Lie derivative $\mathcal{L}_{\mathbf{D}}(f)$ for any affine function $f : \boldsymbol{c}^T\boldsymbol{x} + d$ is also affine.

If $\boldsymbol{c}_1 \leq \boldsymbol{c}_2$ (the $\leq$ relation is applied entry-wise) then $\langle H, \boldsymbol{c}_1\rangle \subseteq \langle H, \boldsymbol{c}_2\rangle$. Given a template H, operations such as join, intersection, post-condition, emptiness and containment checks can all be carried out efficiently.

Positive Invariant Sets. Informally, a closed region C is a positive invariant iff at every point on its surface, the vector field "points" back inside the region [1].

M. Egerstedt and B. Mishra (Eds.): HSCC 2008, LNCS 4981, pp. 654–657, 2008.

The polyhedron $\langle H, \boldsymbol{d}\rangle$ s.t. $\boldsymbol{c}_0 \leq \boldsymbol{d} \leq \mathsf{inv}$, is a *positive invariant* w.r.t $\langle H, \mathsf{inv}\rangle$ iff for each row i, either (a) $\langle H, \boldsymbol{d}\rangle \wedge H_i\boldsymbol{x} = \boldsymbol{d}_i \models \mathcal{L}_{\mathbf{D}}(H_i\boldsymbol{x}) > 0$, or (b) $\boldsymbol{d}_i = \mathsf{inv}_i$. The notion of positive invariance can be relaxed using *Lagrangian relaxation.* $\langle H, \boldsymbol{d}\rangle$ is a *relaxed invariant* w.r.t a *scale factor* $\mu \in \mathcal{R}$, iff $\boldsymbol{c}_0 \leq \boldsymbol{d} \leq \mathsf{inv}$, and

$$\forall\ i \in [1, m],\ \text{if}\ \ \boldsymbol{d}_i < \mathsf{inv}_i\ \text{then}\ \ \langle H, \boldsymbol{d}\rangle \models\ \mathcal{L}_{\mathbf{D}}(H_i\boldsymbol{x} - \boldsymbol{d}_i) + \mu(H_i\boldsymbol{x} - \boldsymbol{d}_i) \leq 0\,.$$

Theorem 1. *If $\langle H, \boldsymbol{d}\rangle$ is a relaxed invariant w.r.t some scale factor μ, then it is a positive invariant.*

2 Policy Iteration

We now sketch the salient aspects of our *policy iteration* technique to compute relaxed invariants[1]. The technique presented here extends earlier work by Gaubert et al. to continuous systems [2]. Consider the instance $\langle H, \boldsymbol{c}_0, \mathsf{inv}, \mathbf{D}\rangle$ along with a fixed value for the scale factor μ. Policy iteration starts from an initial relaxed invariant $\boldsymbol{\alpha}(0) = \mathsf{inv}$, and computes a sequence of invariants: $\mathsf{inv} = \boldsymbol{\alpha}(0) > \boldsymbol{\alpha}(1) > \cdots > \boldsymbol{\alpha}(N) = \boldsymbol{\alpha}(N+1) \geq \boldsymbol{c}_0$, eventually converging to a relaxed invariant $\boldsymbol{\alpha}(N)$. For simplicity, we assume that the initial conditions and the invariants are *non-empty* and *bounded.*

Dual Certificate. Let $\langle H, \boldsymbol{\alpha}\rangle$ be a relaxed invariant w.r.t a scale factor μ. We define a *certificate* to verify this fact. The key requirement to be checked is that for each row $j \in [1, m]$, if $\boldsymbol{\alpha}_j < \mathsf{inv}_j$, then $\langle H, \boldsymbol{\alpha}\rangle \models \mu(H_j\boldsymbol{x} - \boldsymbol{\alpha}_j) + \mathcal{L}(H_j\boldsymbol{x}) \leq 0$. This condition is checked by verifying that the linear program L_j has a non-positive solution:

$$L_j\ :\ \mathsf{max}\ \mu(H_j\boldsymbol{x} - \boldsymbol{\alpha}_j) + \mathcal{L}(H_j\boldsymbol{x} - \boldsymbol{\alpha}_j)\ \mathsf{s.t.}\ \langle H, \boldsymbol{\alpha}\rangle\,, \tag{1}$$

Note that since $\langle H, \boldsymbol{\alpha}\rangle$ is feasible and bounded (because $\boldsymbol{c}_0 \leq \boldsymbol{\alpha} \leq \mathsf{inv}$), the optimal solution to L_j exists and is bounded. A row j for which $\boldsymbol{\alpha}_j \geq \mathsf{inv}$ is termed a *frozen row.* The value of $\boldsymbol{\alpha}_j$ is justified by the invariant for such a row. Let $H_j'\boldsymbol{x} + h_j$ denote the Lie derivative of $H_j\boldsymbol{x}$. Dualizing Eqn. 1, we obtain

$$D_j\ :\ \mathsf{min}\ \boldsymbol{\alpha}^T\boldsymbol{\lambda} - \mu\boldsymbol{\alpha}_j + h_j\ \mathsf{s.t.}\ \ H^T\boldsymbol{\lambda} = (\mu(H_j) + H_j')^T\ \ \wedge \boldsymbol{\lambda} \geq 0 \tag{2}$$

The solution to D_j certifies the validity of Eqn. 1 if its optimal value is non-negative: $\boldsymbol{\alpha}^T\boldsymbol{\lambda} - \mu\boldsymbol{\alpha}_j + h_j \leq 0$. The dual solutions can certify a relaxed invariant.

Definition 2 (Invariant Certificate). *An* invariant certificate *is a tuple $\langle F, \Lambda\rangle$ wherein $F \subseteq \{1, \ldots, m\}$ is a set of* frozen *row indices, while Λ is a $m \times m$ matrix with non-negative entries; s.t. for each index $j \in [1, m] - F$, $H^T\Lambda_j = (\mu(H_j) + H_j')^T \wedge \Lambda_j \geq 0$ (Eqn. 2), and for each index $j \in F$, $\Lambda_j = \mathbf{0}$.*

An invariant $\langle H, \boldsymbol{\alpha}\rangle$ is certified by $\langle F, \Lambda\rangle$ iff for each $j \in F$, $\boldsymbol{\alpha}_j = \mathsf{inv}_j$ and for each $j \in [1, m] - F$, $\boldsymbol{\alpha}^T\Lambda_j - \mu\boldsymbol{\alpha}_j + h_j \leq 0$. Given an invariant $\boldsymbol{\alpha}$, we can extract its certificate as follows: First, we solve the LP D_j for each row j. Following

[1] A detailed version of this paper may be obtained by requesting the authors.

L_j, it always has an optimum. If the optimal value is positive, then $j \in F$ and $\Lambda_j = 0$. Otherwise, Λ_j is set to the optimal solution for D_j. The certificates obtained using this procedure will be called *vertex certificates*. Therefore, every relaxed invariant $\langle H, \boldsymbol{\alpha} \rangle$ is certified by some vertex certificate π.

On the flip side, given certificate $\pi : \langle F, \Lambda \rangle$, the relaxed invariants that are certified by it are obtained using the following constraints:

$$L_\pi : \ \boldsymbol{c}_0 \leq \boldsymbol{y} \leq \mathsf{inv} \ \wedge \ \bigwedge_{j \in F} \boldsymbol{y}_j = \mathsf{inv}_j \ \wedge \ \bigwedge_{j \in [1,m]-F} \Lambda_j^T \boldsymbol{y} - \mu \boldsymbol{y}_j + h_j \leq 0 \qquad (3)$$

A certificate π is *feasible* iff the constraint L_π is feasible; i.e, it certifies at least one relaxed invariant.

Lemma 1. *If certificate π is feasible, then it has a* minimal solution*. I.e., $\exists\, \boldsymbol{c} \in [\![L_\pi]\!],\ \mathsf{s.t.}\ \forall \boldsymbol{d} \in [\![L_\pi]\!],\ \boldsymbol{c} \leq \boldsymbol{d}$.*

The minimal solution can be found by solving the LP: $\mathsf{min.}\ \sum_j \boldsymbol{y}_j\ \mathsf{s.t.}\ L_\pi$. The following result forms the basis of our technique:

Theorem 2. *There are finitely many ($O(2^{|H|} \cdot |H|^{2^{|H|}})$) vertex certificates.*

Let $P = \{\pi_1, \ldots, \pi_M\}$ be the set of all feasible vertex certificates, and $C = \{\boldsymbol{c}_i | \boldsymbol{c}_i$ is the least solution to $L_{\pi_i}\}$ be the corresponding least relaxed invariants.

Lemma 2. *For every relaxed invariant $\boldsymbol{c}$, there exists a relaxed invariant $\boldsymbol{c}_j \in C$, s.t. $\boldsymbol{c}_j \leq \boldsymbol{c}$.*

Applying Lemma 2 repeatedly, we show that C has a minimum element. As a result, the least relaxed invariant exists and can be computed algorithmically by enumerating all the elements of the set C, in turn obtained by enumerating P. However, the naive procedure is doubly exponential in the size of the template.

Therefore, we use a *policy iteration* algorithm to converge to a relaxed invariant while exploring a tiny fraction of the set C in practice. However, this solution is not always guaranteed to be the least solution. Starting from $\boldsymbol{\alpha}(0) = \mathsf{inv}$, we repeat the policy improvement steps (shown below) until $\boldsymbol{\alpha}(j+1) = \boldsymbol{\alpha}(j)$.

1. Compute the certificate $\pi(j)$ for $\boldsymbol{\alpha}(j)$ by solving D_j (Eqn. 2).
2. Compute $\boldsymbol{\alpha}(j+1)$ by solving the LP $L_{\pi(j)}$ in Eqn. 3.

Theorem 3. *The policy iteration eventually converges to a relaxed invariant.*

3 Implementation and Experiments

Our prototype tool TimePass implements the techniques described in this paper using template polyhedra for the safety analysis of affine hybrid systems. TimePass primarily uses a flowpipe construction technique for template polyhedra described in an earlier work [4]. The policy iteration algorithm is used to restrict the invariant region for the flowpipe construction. The resulting flowpipe construction is more precise. Surprisingly, the policy iteration technique also leads to fewer flowpipe segment and therefore a non-trivial speedup.

Table 1. Performance of our tool on hybrid systems benchmarks. All timings are in seconds and memory in MBs. Note, **H**: Template size, **T**:Time, **Mem:** memory, **Prf?**: Property proved.

Name	Description	Bench Size				Policy Iter.			FPipe		Comb.	
		#Var	#Loc	#Trs	$\lvert H \rvert$	T	Mem	Prf?	T	Prf?	T	Prf?
nav01	Benchmark [3]	4	8	18	64	10	30	Y	260	Y	22	Y
nav02	-	4	8	18	64	12	25	Y	362	Y	23	Y
nav03	-	4	8	18	64	8	24	Y	390	Y	20	Y
nav04	-	4	8	18	64	2	12	N	1147	Y	18	Y
nav05	-	4	8	18	64	2	10	N	7	N	513	Y
nav06	-	4	8	18	64	5	15	N	45	N	1420	N
nav07	-	4	15	39	64	14	31	Y	1300	N	572	Y
nav08	-	4	15	39	64	12	27	N	139	N	572	Y

Experiments. Table 1 shows the performance of our tool on some hybrid systems benchmarks consisting of small but complex systems, designed to test the accuracy of the flowpipe construction and its propagation. A detailed description is available elsewhere [3]. We compare the performance of three ways for computing the time elapse: (A) policy iteration, (B) flowpipe construction and (C) their combination. Note that policy iteration alone is unable to prove many of the properties. Furthermore, the strengths of the two approaches seem complementary. Together, they can prove properties beyond the reach of either. Our timings are competitive with those reported by tools such as PHaVer, HSolver and a previous version of our tool using full convex polyhedra. Furthermore, we are able to prove more systems using our techniques than previously reported elsewhere.

References

1. Blanchini, F.: Set invariance in control. Automatica 35 11, 1747–1889 (1999)
2. Gaubert, S., Goubault, E., Taly, A., Zennou, S.: Static analysis by policy iteration on relational domains. In: De Nicola, R. (ed.) ESOP 2007. LNCS, vol. 4421, pp. 237–252. Springer, Heidelberg (2007)
3. Fehnker, A., Ivančić, F.: Benchmarks for hybrid systems verification. In: Alur, R., Pappas, G.J. (eds.) HSCC 2004. LNCS, vol. 2993, pp. 326–341. Springer, Heidelberg (2004)
4. Sankaranarayanan, S., Dang, T., Ivančić, F.: Symbolic model checking of hybrid systems using template polyhedra. In: Ramakrishnan, C.R., Rehof, J. (eds.) TACAS 2008. LNCS, vol. 4963, pp. 188–202. Springer, Heidelberg (2008)