An abstraction and refinement computational approach to safety verification of discrete time nonlinear systems

Simone Smeraldo, Riccardo Desimini, Maria Prandini

***Abstract*— This paper addresses safety verification of nonlinear systems through invariant set computation. More precisely, our goal is verifying if the state of a given discrete time nonlinear system will keep evolving within a safe region, starting from a given set of initial conditions. To this purpose, we introduce a conformant PieceWise Affine (PWA) abstraction of the nonlinear system, which is instrumental to computing a conservative approximation of its maximal invariant set within the safe region. If the obtained set covers the set of initial conditions, safety is proven. Otherwise, subsequent refinements of the PWA abstraction are performed, either on the whole safe region or on some appropriate subset identified through a guided refinement approach and containing the set of initial conditions. Some numerical examples demonstrate the effectiveness of the approach.**

I. Introduction

This paper addresses the problem of verifying if the state of a nonlinear discrete time system keeps evolving within a safe region when initialized in a given set. The problem can be rephrased as that of checking if there exists an invariant set within the safe region including the set of initial conditions.

Invariant set computation has attracted the interest of many researchers since it plays an essential role in model predictive control where recursive feasibility can be granted by properly choosing an invariant set for the terminal state constraint, [1], [2]. In the case of a nonlinear system, the complexity of the problem dramatically increases and approaches that exploit the regularity of the system dynamics [3], [4], or apply some localization procedure [5] have been proposed in the literature.

We propose an alternative approach that relies on the abstraction of the nonlinear system to a PieceWise Affine (PWA) model with additive disturbance so as to exploit an efficient procedure for robust invariant set computation for PWA systems, [6]. If the abstraction is conformant (i.e., it can generate all possible trajectories of the original system), then, an inner approximation of the invariant set for the nonlinear system is obtained.

A counterexample guided abstraction refinement (CEGAR) scheme is adopted in [7], [8] for safety verification of hybrid systems, where a finite ([7]) or infinite ([8]) conformant abstraction on which it is easier to verify safety is built and progressively refined when spurious counterexamples of unsafe behaviors are found. In this work, instead, the abstraction is constructed to the purpose of computing an invariant set within the safe region that contains the set of initial states: if such an invariant is found, then the safety property is proven for the original nonlinear system, otherwise the abstraction is refined to get a more accurate (larger) approximation of the invariant set within the safe region. We derive the sequence of PWA abstractions by means of the hybridization procedure in [9], which, differently from those in [10] and [11], guarantees that all the trajectories that are generated by a finer PWA abstraction can also be generated by the previous rougher one in the sequence while still including all the trajectories of the original nonlinear system. This property, called *refinement inclusion*, allows to progressively get a tighter approximation of the nonlinear system invariant set by subsequently refining the PWA approximant.

In [12], verification of linear temporal logic specifications (including safety) is addressed for PWA systems subject to additive disturbance. An approach for finding the largest set of initial conditions such that the corresponding PWA trajectories satisfy a given specification is proposed. This approach could be adopted for verifying safety of our progressively refined PWA abstractions instead of using them for invariant set computation. However, the admittedly computationally intensive nature of the approach in [12] makes this alternative solution not convenient.

Differently from CEGAR approaches, we do not need to identify and analyze counterexamples to the purpose of refining our model. We in fact adopt a refinement strategy that is based on the abstraction error. However, throughout our procedure, we still search for counterexamples and we do it by simulating within a preassigned finite time horizon the nonlinear system, initialized at a finite number of states extracted uniformly from the subset of initial conditions outside the current invariant estimate. Refinement stops as soon as we are either able to assess safety/unsafety for the nonlinear system or a certain threshold value of the abstraction error is reached. In the latter case, we mitigate undecidability by providing probabilistic finite horizon safety guarantees.

To relieve the combinatorial explosion of the number of modes in the abstraction refinement, we propose an abstraction-based scheme that performs only local guided refinements while searching for the invariant set. We further reduce the computational load by refining the mode partition only outside the currently computed invariant set, since this set remains invariant also for finer abstractions due to the refinement inclusion property.

The authors are with Politecnico di Milano, Piazza Leonardo Da Vinci, 32, 20133 Milano, Italy – e-mail: simone.smeraldo@polimi.it, riccardo.desimini@polimi.it, maria.prandini@polimi.it

II. Problem formulation

Consider a discrete time nonlinear system described by:

$$x_{k+1} = f(x_k), \tag{1}$$

where $x \in \mathbb{R}^n$ denotes the state and function $f : \mathbb{R}^n \to \mathbb{R}^n$ is continuously differentiable up to the second order. Given a compact set $X_s \subset \mathbb{R}^n$ and a set $X_0 \subseteq X_s$, our goal is to determine if system (1) is safe, i.e., if it keeps evolving within X_s when initialized in X_0.

In order to verify safety, we propose a computational abstraction and refinement approach to determine an invariant set I for system (1) satisfying $X_0 \subseteq I \subseteq X_s$. The approach requires that X_s is a convex polytope and X_0 is convex. In order to satisfy this assumption, an inner and an outer approximation of X_s and X_0, respectively, can eventually be adopted.

III. Proposed abstraction and refinement iterative scheme

Our scheme for safety verification starts with the computation of the robust invariant set of a rough PWA abstraction of system (1) over X_s to obtain an inner-approximation of the invariant set of system (1). If this approximation includes X_0, then safety is proven for system (1), otherwise a refinement of the PWA abstraction that reduces the maximal abstraction error over the relevant region is performed, and the procedure of computing a robust invariant set and checking for inclusion of X_0 is repeated for the resulting finer PWA model. We adopt the hybridization proposed in [9] to reduce system (1) to a PWA abstraction where the introduced approximation error is accounted for via an additive disturbance. The refinement inclusion property in [9] makes the resulting inner approximation of the invariant set of the original nonlinear system progressively more and more accurate as its PWA abstraction is refined.

While progressively refining, we also look for a counterexample falsifying safety so as to possibly avoid further useless invariant set computations. To this purpose, we extract a finite number of samples uniformly from X_0 and compute the corresponding trajectories of the nonlinear system over some finite horizon $[0, N]$. If any of them exit X_s, then, we can declare the system unsafe and halt the iterations. Indeed, the described procedure is repeated until one of the following three termination conditions occurs: an invariant including X_0 is found, a witness of unsafety is retrieved, none of the previous conditions is satisfied but the maximal abstraction error gets below a user-chosen threshold. In this last case, we are not able to give a definite answer to the safety verification problem and can only provide a probabilistic finite-time assessment of safety based on the overall number of extracted (safe) trajectories.

The computationally attractive procedure proposed in [6] to determine a polyhedral inner approximation of the maximal robustly positively invariant set inside a polyhedral region for a PWA system is integrated in the iterative scheme.

A key point of the proposed safety verification method is the refinement policy adopted while generating finer PWA models. We propose a *global refinement strategy* where the current PWA abstraction is refined over the entire safe region X_s, and a *local one* where the refinement is confined to a subset of X_s that is an outer bound containing X_0 of the robust invariant set for the current PWA abstraction.

The latter strategy can be implemented using the outer bound computed in the iterative procedure of [6] while determining the invariant set. This may result in a significant reduction of the computational effort, but at the risk of excessively restricting the search domain. A quantitative comparison of the two strategies in terms of computational complexity is hard to obtain, since the size of the outer bound strongly depends on the dynamics at hand. However, numerical examples show the superiority of the local refinement strategy with respect to the global one.

Regardless of the adopted refinement strategy, in order to relieve the computational load, we avoid refining the PWA abstraction within the current estimate of the invariant, since it is an invariant (although not maximal) set also for the original nonlinear system.

We next describe in some detail each step of our scheme.

A. PWA abstraction

Given system (1), evolving within a compact set X, the hybridization process starts considering an outer hyper-rectangle of X with sides parallel to the coordinate axes. Then, X is divided into r closed hyper-rectangles $\{X_i\}_{i=1}^r$, and dynamics (1) is approximated through a PWA model given by:

$$x_{k+1} = A_i x_k + v_i + e_k,\ x_k \in M_i = X \cap X_i,\ i = 1, \ldots, r,$$

where $e \in \mathbb{R}^n$ is an additive error term taking values in a compact set $E_i \subset \mathbb{R}^n$ that depends on X_i and $M_i = X \cap X_i$ denotes mode i, $i = 1, \ldots, r$.

Remark 1: Note that at the boundary of each mode M_i multiple dynamics may be activated. However, this is not an issue since the hybridization process proposed in [9] guarantees that if $x \in M_i$, then, $f(x) \in \{A_i x + v_i + e, e \in E_i\}$, so that if x is a point at the boundary between two modes M_i and M_j, then, this property holds for each activated mode. It is then irrelevant which affine dynamics is actually applied for the PWA model to be a conformant abstraction of the original nonlinear system.

We next define matrix A_i, vector v_i, and set E_i.

Let $c^{(i)}$ denote the center of the hyper-rectangle X_i, which is given by $c^{(i)} = 0.5(\underline{x}^{(i)} + \overline{x}^{(i)})$, where $\underline{x}^{(i)}$ and $\overline{x}^{(i)}$ are, respectively, the vertices with minimal and maximal components of X_i. Then, the j-th column of matrix A_i associated with mode M_i is computed as:

$$A_i^{[j]} = \frac{f(c_1^{(i)}, \ldots, \overline{x}_j^{(i)}, \ldots, c_n^{(i)}) - f(c_1^{(i)}, \ldots, \underline{x}_j^{(i)}, \ldots, c_n^{(i)})}{L_j^{(i)}}$$

where $L_j^{(i)} = \overline{x}_j^{(i)} - \underline{x}_j^{(i)}$ is the size of X_i along coordinate j. The affine term v_i, instead, is computed as $v_i = f(c^{(i)}) - A_i c^{(i)}$, so that the PWA and the nonlinear functions take the same value in the center of X_i. The error support set E_i

associated with mode M_i is characterized in [9] as $E_i = [-\overline{e}^{(i)}, \overline{e}^{(i)}]$, where $\overline{e}^{(i)} \in \mathbb{R}^n$ satisfies elementwise

$$|f(x) - (A_i x + v_i)| \leq \overline{e}^{(i)}, \quad x \in X_i,$$

and is given by

$$\overline{e}^{(i)} = \frac{1}{8} m_H^{(i)} \sum_{j=1}^{n} \sum_{k=1}^{n} L_j^{(i)} L_k^{(i)}, \tag{2}$$

with $m_H^{(i)} \in \mathbb{R}^n$ a vector whose l-th component, $l = 1, \ldots, n$, is defined as:

$$m_{Hl}^{(i)} = \max_{j,k} \max_{x \in X_i} \left| \frac{\partial^2 f_l(x)}{\partial x_j \partial x_k} \right| \tag{3}$$

We shall refer to the components of $m_H^{(i)}$ as the *Lipschitz factors*[1] of f associated with X_i and to the components of $\overline{e}^{(i)}$ as the *error bounds* of f over X_i.

Figure 1 shows an example of the described hybridization procedure applied to a scalar quadratic function with $r = 2$.

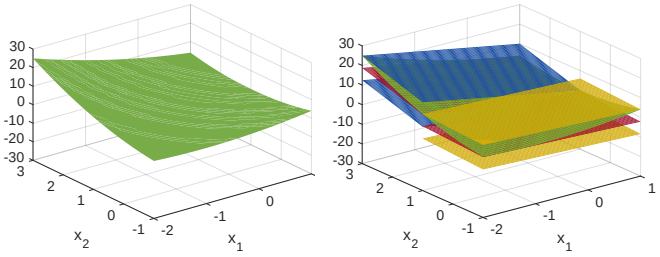

Fig. 1. (a) The nonlinear function $f(x_1, x_2) = (x_1 - x_2)^2$. (b) Its hybridization over $X = [-2, 1] \times [-1, 3]$ for $r = 2$. In green the nonlinear function, in red the PWA approximant, in orange and blue the error bounds.

B. PWA abstraction refinement

The refinement procedure consists in making the rectangular grid of the current PWA abstraction finer, halving some hyper-rectangles along axis-aligned directions and recomputing the affine dynamics and error bounds over the obtained modes as explained in the previous subsection. By applying such a procedure, we increase the accuracy of the PWA abstraction, getting progressively closer to the original nonlinear dynamics as the grid gets finer, due to the refinement inclusion property discussed in the introduction (see [9] for a proof), with the error bounds in (2) tending to zero as the size of the grid elements decreases.

To obtain an efficient safety verification algorithm, we devise a smart refinement procedure where the hyper-rectangles to halve and the axis-aligned directions along which halving are chosen sequentially so as to uniformly reduce the approximation error over the domain of interest, which can either be the safe region X_s or an outer bound of the robust invariant set for the current PWA abstraction.

To this aim, we first select a hyper-rectangle with maximal error bound, which can be interpreted as a measure of the worst combination of its Lipschitz factors and dimensions (see (2)). Once the target hyper-rectangle has been chosen, rather than simply selecting the halving direction that leads to the largest reduction in size, we compute the error bounds associated with the sub-regions obtained by halving the hyper-rectangle in all possible axis-aligned directions and choose the one attaining the overall lowest maximal error bound, so as to guarantee that the abstraction error is reduced as much as possible with a single refinement (recalling equation (2), the error is affected by both the Lipschitz factors and the hyper-rectangle sizes).

[1]Note that $m_{Hl}^{(i)}$ is well defined since the second order derivative of function $f(\cdot)$ is assumed to be continuous and X_i is compact.

Algorithm 1 Safety counterexample generation

```
1  Extract n_s samples {x_in^[k]}_{k=1}^{n_s} from X_in according to the
   uniform distribution
2  for k = 1, ..., n_s do
3     x_0^[k] ← x_in^[k]
4     for t = 1, ..., N do
5        x_t^[k] ← f(x_{t-1}^[k])
6        if x_t^[k] ∉ X_s then
7           System (1) is unsafe
8           return
9        end if
10    end for
11 end for
```

C. Robust invariant set computation and safety verification via PWA abstraction and refinement

Two algorithms for safety verification are formulated, which differ for the adopted refinement strategy:

1) Safety verification through global error-based refinement: A rough PWA abstraction of system (1) is introduced by gridding an outer hyper-rectangle containing X_s. If the robust invariant set within X_s computed for the current PWA abstraction does not contain X_0, the refinement is performed over the portion of the safe region outside the latest invariant I. This approach is summarized in Algorithm 2.

2) Safety verification through local error-based guided refinement: The algorithm in [6] applied within a region of interest computes an outer bound $\hat{C}$ of the maximal robust invariant set within that region. Differently from Algorithm 2, we then refine the PWA abstraction over the set $\hat{C} \setminus I$ in place of $X_s \setminus I$, as long as $\hat{C}$ contains X_0.

Algorithm 3 summarizes the proposed local error-based refinement approach, which proceeds analogously to the global one except for the fact that the refinement domain $\hat{C}$, initialized at X_s (line 2), is progressively updated every time an outer bound C_0 that includes X_0 is computed (lines 5-6). We highlight that, due to the local nature of the invariant search region according to the proposed guided refinement strategy, it may happen that the computed outer bound is too small to get an invariant set covering the set of initial conditions. In this case, when we reach the maximum (local) accuracy without being able to prove or disprove safety, we can still resort to the global refinement strategy.

In both safety verification methods, we try to find a safety counterexample every time we fail at proving that the system

Algorithm 2 Safety via global error-based refinement

Require: $\underline{e}_{max} > 0$, $0 < \alpha < 1$, $N > 0$, $n_s > 0$.

Apply Algorithm 1 setting $X_{in} \leftarrow X_0$
$I \leftarrow \emptyset$
Compute an affine abstraction for system (1) over X_s, and retrieve its maximal error bound $\|\bar{e}^{(1)}\|_\infty$.
$r \leftarrow 1$, $e_{max} \leftarrow \alpha \|\bar{e}^{(1)}\|_\infty$, $X_d \leftarrow X_s$
while $\max\{\|\bar{e}^{(i)}\|_\infty\}_{i=1}^{r} > \underline{e}_{max}$ **do**
  while $\max\{\|\bar{e}^{(i)}\|_\infty\}_{i=1}^{r} > e_{max}$ **do**
    Refine the current PWA grid (Section III-B) choosing among the modes covering X_d and retrieve the new maximal error bounds $\{\|\bar{e}^{(i)}\|_\infty\}_{i=1}^{r+1}$.
    $r \leftarrow r + 1$
  end while
  Compute an outer bound C_o for the maximal robust invariant set within X_s for the current abstraction, [6]
  if $X_0 \subseteq C_o$ **then**
    Compute a robust invariant set $I \subseteq C_o$ for the current abstraction, [6]
    if $X_0 \subseteq I$ **then**
      System (1) is safe
      return
    else
      Collect the maximal error bounds $\{\|\bar{e}^{(i)}\|_\infty\}_{i=1}^{l}$ associated with the modes covering $X_s \setminus I$.
      $X_d \leftarrow X_s \setminus I$, $r \leftarrow l$.
    end if
  end if
  Apply Algorithm 1 setting $X_{in} \leftarrow X_0 \setminus I$
  $e_{max} \leftarrow \alpha\, e_{max}$
end while
Retrieve the number m of samples extracted from $X_0 \setminus I$
$\varepsilon \leftarrow 1 - \beta^{\frac{1}{m}}$
With confidence $1 - \beta$, the set of initial conditions that can drive system (1) outside X_s within $[0, N]$ has a size smaller than or equal to ε times the measure of $X_0 \setminus I$

is safe. We repeat this process until we are able to prove that the system is either safe or unsafe, or until the PWA abstraction reaches the user-chosen accuracy level within the currently explored modes. In this latter case, we provide probabilistic safety guarantees over $X_0 \setminus I$, with I denoting the last computed invariant set.

3) Probabilistic safety certificate: Consider all the samples extracted from X_0 according to the uniform distribution while applying Algorithm 2 or Algorithm 3, and remove the ones included within the last computed invariant set I. The remaining, say m, samples are independent and uniformly distributed in $X_0 \setminus I$. For all these samples the behaviour of the system is safe over $[0, N]$, which allows us to provide a probabilistic safety certificate as follows.

Let p be defined as the probability of extracting from $X_0 \setminus I$ an unsafe initialization over $[0, N]$ and assume that we perform n_e extractions. Then, the random variable representing the number of unsafe extractions is binomially distributed with parameters p and n_e. Given some confidence parameter $\beta \in (0, 1)$, we impose that the probability $(1 - p)^{n_e}$ of extracting from $X_0 \setminus I$ no unsafe initialization out of n_e samples while having p greater than a threshold $\varepsilon \in (0, 1)$ is smaller than or equal to β, which implies, being such probability a decreasing function of p, that $(1 - \varepsilon)^{n_e} \leq \beta$.

Algorithm 3 Safety via local error-based refinement

Require: $\underline{e}_{max} > 0$, $0 < \alpha < 1$, $N > 0$, $n_s > 0$.

Run lines 1-3 in Algorithm 2
$r \leftarrow 1$, $e_{max} \leftarrow \alpha \|\bar{e}^{(1)}\|_\infty$, $X_d \leftarrow X_s$, $\hat{C} \leftarrow X_s$
while $\max\{\|\bar{e}^{(i)}\|_\infty\}_{i=1}^{r} > \underline{e}_{max}$ **do**
  Run lines 6-9 in Algorithm 2
  Compute an outer bound C_o for the maximal robust invariant set within $\hat{C}$ for the current abstraction, [6]
  if $X_0 \subseteq C_o$ **then**
    Collect the maximal error bounds $\{\|\bar{e}^{(i)}\|_\infty\}_{i=1}^{q}$ associated with the modes covering C_o
    $\hat{C} \leftarrow C_o$, $r \leftarrow q$
    Compute a robust invariant set $I \subseteq \hat{C}$ for the current abstraction, [6]
    if $X_0 \subseteq I$ **then**
      System (1) is safe
      return
    else
      Collect the maximal error bounds $\{\|\bar{e}^{(i)}\|_\infty\}_{i=1}^{l}$ associated with the modes covering $\hat{C} \setminus I$.
      $X_d \leftarrow \hat{C} \setminus I$, $r \leftarrow l$.
    end if
  end if
  Run lines 21-22 in Algorithm 2
end while
Run lines 24-26 in Algorithm 2

Turning back again to the uniformly distributed sample of m points within $X_0 \setminus I$ obtained at the end of Algorithm 2 or Algorithm 3, since none of these m extractions led to unsafe trajectories, we can state that p is less than ε with confidence larger than or equal to $1 - \beta$, thus implying that the fraction of the set of initial conditions leading to an unsafe behavior over $[0, N]$ does not exceed the minimum ε satisfying $(1 - \varepsilon)^m \leq \beta$, i.e.:

$$\varepsilon = 1 - \beta^{\frac{1}{m}}.$$

IV. Numerical results

We consider a numerical example presented in [4], where a nonlinear system of the form (1) is considered with $f : \mathbb{R}^2 \mapsto \mathbb{R}^2$ given by

$$f(x) = \begin{bmatrix} 1 + 0.1x_1 + 0.5x_2 - e^{0.1x_1^2} \\ 0.1 + 0.9x_1 - 0.1x_2 - 0.1\cos(x_2) + 0.05x_2^2 \end{bmatrix} \tag{4}$$

Given the region $X_s = [-5, 5]^2 \subset \mathbb{R}^2$, our goal is to verify that the state of the system never exits X_s, starting from some set of initial conditions $X_0 \subset \mathbb{R}^2$.

To this purpose we apply the algorithms in Section III-C, which have been implemented in MATLAB, using the Multi-Parametric Toolbox (MPT), [13], for representing convex polyhedra, and CPLEX, [14], to solve the linear programs required for the procedure in [6].

Results have been obtained on a calculator with processor AMD Ryzen 7 PRO 4750U with Radeon Graphics, 8 Core(s) (1.7 GHz), 16 Logical Processor(s), and 32 GB of RAM.

We present results on safety verification for different sets X_0 and compare global and local refinement strategies, setting in both of them the maximal accuracy parameter $\underline{e}_{max} = 10^{-3}$ and the coefficient rescaling the accuracy

at each iteration of the refinement $\alpha = 0.5$. As for the parameters required for probabilistic assessment, we set $n_s = 7000$ and $\beta = 10^{-3}$, and we choose $N = 50$ as finite horizon length while searching for counterexamples.

For each case, the following performance criteria are considered:

- The maximal error bound $\max\{\|\bar{e}^{(i)}\|_\infty\}_{i=1}^r$ associated to the last obtained PWA abstraction, which represents the final approximation accuracy.
- The number h of region halvings performed to obtain the final mode partition.
- The computing time in seconds, indicative of the computational effort required by each method.

When applying the local refinement strategy, we shall also consider the number q of updates of the outer bound $\hat{C}$.

As for the computation of the Lipschitz factors of f in (3), in this example it is possible to compute them analytically for each region generated during the refinement.

1) $X_0 = [-4, -2]^2$*:* in this case, running any of our two algorithms for safety verification, we are immediately able to find a counterexample. System (4) is thus deemed unsafe. Figure 2 depicts an unsafe trajectory of system (4) starting from X_0.

2) $X_0 = [-1, 1]^2$*:* this set is around the origin, where function (4) is close to be linear. The plots in Figure 3 depict the obtained solutions, whereas Table I provides more quantitative data. While a global refinement leads to a wide invariant set with respect to X_0, with a higher computational load, the local refinement leads very efficiently to an invariant set that is more restricted around X_0 but yet compatible with the safety specification, also retrieving a PWA abstraction that is more accurate.

3) X_0 *is a large polytope around the origin:* we consider a significantly enlarged initial set X_0, which is now the polytope with vertices $\{v_i\}_{i=1}^4$ given by $v_1 = (-2, -3)$, $v_2 = (-3, 4)$, $v_3 = (4, 4)$, $v_4 = (2, -4)$. Through this choice of X_0 we implicitly enforce the local refinement algorithm to compute an invariant set close to the one resulting from the global refinement in the previous example, but with a lower computational effort. The related meaningful information is depicted in Figure 4 and collected in Table II.

4) $X_0 = [2, 4]^2$*:* the considered initial set X_0 is in a region where the nonlinear function in (4) has a stronger nonlinearity. Figure 5, together with Table III, further highlights the computational advantages of a guided refinement with respect to a global refinement.

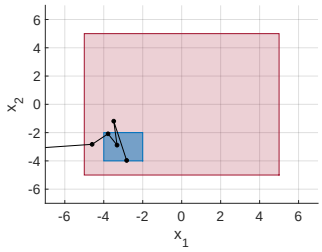

Fig. 2. Unsafe trajectory (black) of system (4), computed within $[0, N]$ by means of Algorithm 1 starting from $X_0 = [-4, -2]^2$ (blue).

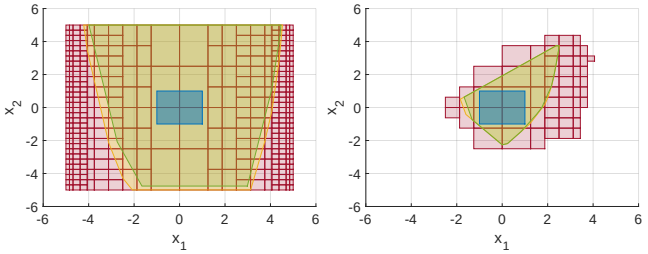

Fig. 3. Global (a) and local (b) refinement for $X_0 = [-1, 1]^2$: in yellow the outer approximation of the invariant, in red the most recent mode partition over X_s (a) and over the latest $\hat{C}$ (b), in green the computed invariant and in cyan the set X_0.

TABLE I

PERFORMANCE INDICES WHEN $X_0 = [-1, 1]^2$

Refinement strategy	$\max\{\|\bar{e}^{(i)}\|_\infty\}_{i=1}^r$	*h*	*q*	*Time (s)*
Global	0.6077	392	-	129
Local	0.3418	122	5	4

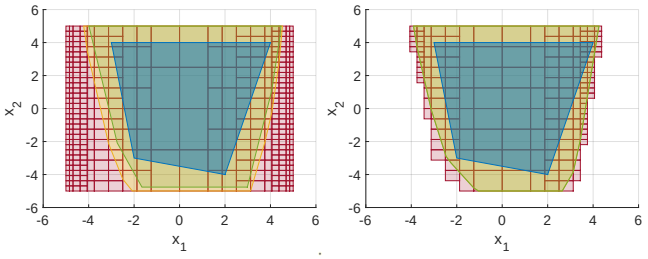

Fig. 4. Global (a) and local (b) refinement for a large polytopic set X_0: in yellow the outer approximation of the invariant, in red the most recent mode partition over X_s (a) and over the latest $\hat{C}$ (b), in green the computed invariant and in cyan the set X_0.

TABLE II

PERFORMANCE INDICES FOR A LARGE POLYTOPIC X_0

Refinement strategy	$\max\{\|\bar{e}^{(i)}\|_\infty\}_{i=1}^r$	*h*	*q*	*Time (s)*
Global	0.6077	392	-	129
Local	0.3197	241	5	83

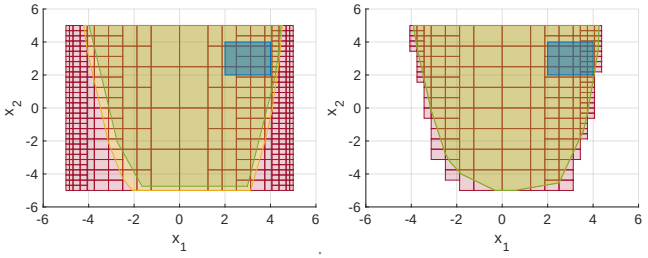

Fig. 5. Global (a) and local (b) refinement for $X_0 = [2, 4]^2$: in yellow the outer approximation of the invariant, in red the most recent mode partition over X_s (a) and over the latest $\hat{C}$ (b), in green the computed invariant and in cyan the set X_0.

TABLE III

PERFORMANCE INDICES WHEN $X_0 = [2, 4]^2$

Refinement strategy	$\max\{\|\overline{e}^{(i)}\|_\infty\}_{i=1}^r$	h	q	*Time (s)*
Global	0.6077	392	-	114
Local	0.3197	260	4	75

5) $X_0 = [-2.5, -0.5] \times [-3.5, -1.5]$*:* we consider a final example with the goal of comparing the performance obtained by running our safety verification algorithms with two alternative refinement domains, i.e., refining also the modes within the latest invariant not including X_0, and refining only the modes not contained within such an invariant. The obtained results are pictorially shown in Figures 6 and 7, as well as in quantitative form in Table IV.

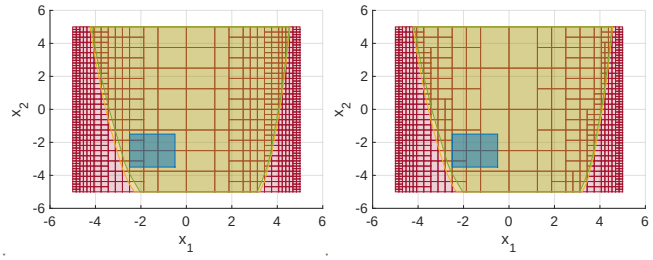

Fig. 6. Variants of Algorithm 2 obtained by refining the modes in X_s (a) and refining the modes in $X_s \setminus I$ (b) for $X_0 = [-2.5, -0.5] \times [-3.5, -1.5]$: in yellow the outer approximation of the invariant, in red the most recent mode partition over X_s, in green the computed invariant and in cyan the set X_0.

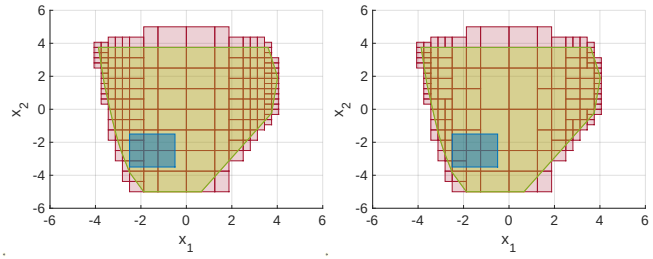

Fig. 7. Variants of Algorithm 3 obtained by refining the modes in $\hat{C}$ (a) and refining the modes in $\hat{C} \setminus I$ (b) for $X_0 = [-2.5, -0.5] \times [-3.5, -1.5]$: in red the most recent mode partition over the latest $\hat{C}$, in green the computed invariant and in cyan the set X_0.

TABLE IV

PERFORMANCE INDICES WHEN $X_0 = [-2.5, -0.5] \times [-3.5, -1.5]$

Refinement domain	$\max\{\|\overline{e}^{(i)}\|_\infty\}_{i=1}^r$	h	q	*Time (s)*
X_s	0.2667	800	-	901
$\hat{C}$	0.2409	277	6	73
$X_s \setminus I$	0.2667	731	-	515
$\hat{C} \setminus I$	0.2409	244	6	62

V. CONCLUSIONS AND FUTURE WORK

In this paper we presented a method for safety verification of nonlinear systems. The proposed approach rests on the introduction of a PieceWise Affine (PWA) conformant abstraction of the nonlinear system, and on the computation of a robust invariant set by means of an efficient algorithm for PWA systems. A refinement procedure allows to progressively improve the estimate of the invariant set, by exploiting the refinement inclusion property of the adopted PWA abstraction method.

In order to mitigate the computational burden associated with a global error-based refinement, we devised a guided refinement procedure that improves the abstraction accuracy locally.

An interesting direction of future work is the extension of the proposed invariant set computation method to the class of nonlinear controlled systems, and its application to model predictive control. Also, scalability of the approach needs to be investigated.

REFERENCES

[1] F. Blanchini, "Set invariance in control," *Automatica*, vol. 35, no. 11, pp. 1747–1767, 1999.

[2] E. C. Kerrigan and J. M. Maciejowski, "Invariant sets for constrained nonlinear discrete-time systems with application to feasibility in model predictive control," in *Proceedings of the 39th IEEE Conference on Decision and Control*, Dec. 2000, pp. 4951–4956.

[3] M. Fiacchini, T. Alamo, and E. F. Camacho, "On the computation of convex robust control invariant sets for nonlinear systems," *Automatica*, vol. 46, no. 8, pp. 1334–1338, 2010.

[4] M. Fiacchini, T. Alamo, and E. Camacho, "On the computation of local invariant sets for nonlinear systems," in *2007 46th IEEE Conference on Decision and Control*, 2007, pp. 3989–3994.

[5] A. P. Krishchenko and A. N. Kanatnikov, "Maximal compact positively invariant sets of discrete-time nonlinear systems," *IFAC Proceedings Volumes*, vol. 44, no. 1, pp. 12 521–12 525, 2011.

[6] T. Alamo, M. Fiacchini, A. Cepeda, D. Limon, J. M. Bravo, and E. F. Camacho, *On the Computation of Robust Control Invariant Sets for Piecewise Affine Systems*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 131–139.

[7] E. Clarke, A. Fehnker, Z. Han, B. Krogh, O. Stursberg, and M. Theobald, "Verification of hybrid systems based on counterexample-guided abstraction refinement," in *Tools and Algorithms for the Construction and Analysis of Systems*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 192–207.

[8] N. Roohi, P. Prabhakar, and M. Viswanathan, "Hybridization based CEGAR for hybrid automata with affine dynamics," in *Tools and Algorithms for the Construction and Analysis of Systems - 22nd International Conference, TACAS 2016, Eindhoven, The Netherlands*, ser. Lecture Notes in Computer Science, vol. 9636. Springer, 2016, pp. 752–769.

[9] P. Tajvar, "Verification of nonlinear systems through hybridization and invariant set computation," Master's thesis, Politecnico di Milano, Milan, Italy, Sep. 2017.

[10] E. Asarin, T. Dang, and A. Girard, "Hybridization methods for the analysis of nonlinear systems," *Acta Informatica*, vol. 43, no. 7, pp. 451–476, 2007.

[11] T. Dang, O. Maler, and R. Testylier, "Accurate hybridization of nonlinear systems," in *Proceedings of the 13th ACM International Conference on Hybrid Systems: Computation and Control*. New York, NY, USA: Association for Computing Machinery, 2010, pp. 11—-20.

[12] B. Yordanov, J. Tůmová, I. Černá, J. Barnat, and C. Belta, "Formal analysis of piecewise affine systems through formula-guided refinement," *Automatica*, vol. 49, no. 1, pp. 261–266, 2013.

[13] M. Herceg, M. Kvasnica, C. N. Jones, and M. Morari, "Multi-parametric toolbox 3.0," in *2013 European Control Conference (ECC)*, Jul. 2013, pp. 502–510.

[14] IBM, *IBM ILOG CPLEX Optimization Studio - CPLEX User's manual*, version 12, release 8 ed., 2017.