Approximate Computation of Reach Sets in Hybrid Systems

D. Ravi∗ and R. K. Shyamasundar

School of Technology and Computer Science
Tata Institute of Fundamental Research
Homi Bhabha Road, Colaba, Mumbai 400 005, India
{ravi, shyam}@tcs.tifr.res.in

Abstract

One of the most important problems in hybrid systems is the reachability problem. The reachability problem has been shown to be undecidable even for a subclass of linear hybrid systems. In view of this, the main focus in the area of hybrid systems has been to find effective semi-decision procedures for this problem. Such an algorithmic approach involves finding methods of computation and representation of reach sets of the continuous variables within a discrete state of a hybrid system. In this paper, after presenting a brief introduction to hybrid systems and reachability problem, we propose a computational method for obtaining the reach sets of continuous variables in a hybrid system. In addition to this, we also describe a new algorithm to over-approximate with polyhedra the reach sets of the continuous variables with linear dynamics and polyhedral initial set. We illustrate these algorithms with typical interesting examples.

1 Introduction

Hybrid systems combine discrete and continuous dynamics. The dynamics of the continuous variables within a discrete state are specified by differential equations or differential inclusions. An important problem in the analysis and synthesis of such systems is the so called reachability problem, which asks, for two sets of configurations of a given hybrid system, say $X_1$ and $X_2$, where a configuration consists of discrete and continuous components, whether or not there is a hybrid trajectory with the initial configuration in $X_1$ and the final configuration in $X_2$. A hybrid trajectory may be described as a trajectory of configurations consisting of discrete state jumps and smooth arcs, where each arc evolves according to the continuous dynamics of a discrete state, with the starting and end points of each arc satisfying the jump conditions of discrete transitions. A more precise definition is presented in Sec. 2, which provides a concise introduction to hybrid systems and the reachability problem.

∗Current Address: Faculty of Mathematics and Computer Science, The Weizmann Institute of Science, 76100 Rehovot, Israel
The reachability problem is undecidable for certain classes of linear hybrid systems (i.e., hybrid systems having linear trajectories within each discrete state and linear or constant reset maps, also called constant slope hybrid systems)\cite{1,2}, although in some cases decidability results have been obtained\cite{3,4} (see also\cite{2}). Therefore, for a general hybrid system, a reasonable alternative appears to be to find semi-decision procedures for the reachability problem\cite{2,5,6}. A computational approach to this problem also requires finding the set of states reached by the continuous variables evolving according to the dynamics of a discrete state. In this paper, we consider the problem of computing and representing the reach sets of the continuous variables within a discrete state when the dynamics of the continuous variables are specified by differential equations with initial conditions belonging to a specified initial set.

In this context, various methods have been proposed in the literature for finding reach sets of continuous variables\cite{5,6,7}. In Sec. 3, we describe a method for computing the reach sets based on the idea that a subset of the boundary of the initial set may be found, such that it is sufficient to compute the solutions with the initial points lying in that set. This method is similar to that in\cite{8} (and also to some extent to that described in\cite{9}). In particular, we present a schematic algorithm, which is somewhat more general in its scope than that in\cite{8} and simpler compared to that in\cite{9}.

Besides these algorithms for reach set computation, an equally important issue the representation of the reach sets for manipulating the sets efficiently. This requires representation of the reach sets in terms of more convenient sets such as, for example polyhedral or subalgebraic sets, that are simple to represent and easy to handle for practical purposes. However, since the representing class of sets may not contain a member that exactly equals the reach set, we may have to find approximations to the reach sets by those that belong to the representing class. (See, for example,\cite{11,12} for approximation with polyhedra, and\cite{13} for approximation using ellipsoids). Typically, over-approximations may be used for verifying whether a safety requirement may potentially be violated by any of the behaviours starting from a given initial set, while under-approximations are needed for characterizing a set of states from which a desirable property is always achievable. We describe in Sec. 4 a method for over-approximating the reach sets by polyhedra when the dynamics are specified by linear differential equations and the initial set is a polyhedron. These algorithms are illustrated with some simple examples in Sec. 5, while Sec. 6 concludes the paper.

2 A Brief Survey of Hybrid Systems

In this section, we present a brief introduction to hybrid systems, and provide motivation for the remaining sections.

2.1 Preliminary Definitions

We begin with a somewhat detailed but general definition of hybrid systems.

**Definition 1** A hybrid system is a tuple $H = (Q, X, \Sigma, G, E, \text{Init}, f)$, where...
1. $Q$ is a finite set of discrete states (also called locations).

2. $X = \mathbb{R}^n$, $n \geq 1$, is the set of continuous states, where $\mathbb{R}$ denotes the set of real numbers. We denote the continuous state variable by $x$.

3. $\Sigma$ is a finite set of discrete events or environment actions. Some events in $\Sigma$ are controllable, while others not. Hence, it is convenient to assume that $\Sigma = \Sigma_c \cup \Sigma_d$, where $\Sigma_c$ is the set of controllable events, and $\Sigma_d$ is the set of uncontrollable or disturbance events.

4. $G \subseteq Q \times X$ is a set of state invariance conditions. When the system is in state $q$, the continuous variables belong to $G(q) = \{ x \in X : (q, x) \in G \}$.

5. $E \subseteq Q \times \mathcal{P}(\mathbb{R}^n) \times \Sigma \times \{ \mathbb{R}^n \rightarrow \mathbb{R}^n \} \times Q$ is the set of transition edges. An edge $e \in E$, where $e = (q_e, X_e, \sigma_e, r_e, q'_e)$, is interpreted as follows:
   - If the continuous state is in $X_e$ and the event $\sigma_e$ occurs, then transition edge $e$ is enabled in state $q_e$. Thus $X_e$ is the set of switching points of the continuous variables from state $q_e$ to $q'_e$. The set $X_e$ is specified by a predicate, and is also called a guard. In case there are many transition edges simultaneously enabled, then the system may select one of the edges nondeterministically.
   - If a transition edge $e$ is selected by the system, then the continuous state is reset using the function $r_e$ when the system enters state $q'_e$. The reset values obey the state invariance condition, $r_e(X_e) \subseteq G(q'_e)$.

6. $\text{Init} \subseteq G$ is a set of initial conditions.

7. $f$ is an $n$-dimensional vector field with real-valued components governing the dynamics of the continuous state $x$. The domain of definition of the function $f$ is a set $D$, where $D = Q \times X$ (if there are no continuous control variables) or $D = Q \times X \times U$ (if $U \subseteq \mathbb{R}^m$, $m \geq 1$, is the range of the continuous control variable, which is a function $u : [0, T_q) \rightarrow U$, where $T_q > 0$ is as large as may be necessary, depending on the discrete state $q \in Q$). When in state $q$, the continuous variables evolve according to the dynamical law
   \[
   \frac{dx}{dt} = f(q, x)
   \]
   or according to the dynamical law
   \[
   \frac{dx}{dt} = f(q, x, u)
   \]
   depending on the presence or absence of the continuous control variables. The initial conditions of the continuous evolution are specified either by the reset maps when the system takes a discrete transition and reaches the state $q$, or by the (nondeterministic) initial conditions of the system given as the set $\text{Init}$.

---

1 Another possibility is to specify the system dynamics of the continuous variables in terms of differential inclusions. In this case, $f(q, x)$ is a set, and the continuous variables evolve according to
   \[
   \frac{dx}{dt} \in f(q, x).
   \]
   But we do not consider this case here.
If, in addition to the set of initial values, $\text{Init}$, the set of final or accepting values in $Q \times X$, denoted by $\text{Final}$, are also specified, then the hybrid system is called a hybrid automaton.

Referring to the system dynamics in the above definition, for our purposes, we assume that there are no continuous control variables. Therefore, in the sequel, we assume that the dynamics of the continuous variables are specified in the form
\[
\frac{dx}{dt} = f(q,x)
\]
without any continuous control variables.

**Definition 2** Let $H = (Q, X, \Sigma, G, E, \text{Init}, f)$ be a hybrid system. Each point $(q,x) \in Q \times X$ is called a configuration of the hybrid system $H$.

In Definitions 3 - 6, we introduce certain terminology, starting with the notion of a step of a hybrid system, leading up to the notions of predecessors and successors of a configuration $(q,x)$. But before that, we mention that predecessors and successors may be defined irrespective of initial conditions, whereas for defining the executions of a hybrid system, we require the specification of the initial set $\text{Init}$, as will be seen later (refer to Definitions 4, 5 and 6).

**Definition 3** Let $H = (Q, X, \Sigma, G, E, \text{Init}, f)$ be a hybrid system, and let $(q,x)$ and $(q',x')$ be two configurations of $H$. Then, the pair of configurations $((q,x),(q',x'))$ is called

1. a time-step, if $q = q'$ and, for some $t \geq 0$, there is a function $y : [0,t] \rightarrow X$, satisfying $y(s) = f(q,y(s))$, $y(s) \in G(q)$, for $s \in [0,t]$, $y(0) = x$ and $y(t) = x'$.
2. an edge-step if there is an edge $e = (q_e, X_e, \sigma_e, r_e, q'_e) \in E$, such that $q = q_e$, $q' = q'_e$, $x \in X_e$ and $x' = r_e(x)$.
3. a $\sigma$-step, where $\sigma \in \Sigma$, if there is an edge $\ell = (q_\ell, X_\ell, \sigma_\ell, r_\ell, q'_\ell) \in E$ with $\sigma = \sigma_\ell$, such that $q = q_\ell$, $q' = q'_\ell$, $x \in X_\ell$ and $x' = r_\ell(x)$.

A step of the hybrid system $H$ is a pair of configurations $((q,x),(q',x'))$, such that $((q,x),(q',x'))$ is either a time-step for some $t \geq 0$, or an edge-step for some $e \in E$, or a $\sigma$-step for some $\sigma \in \Sigma$.

Of course, every edge-step is a $\sigma$-step for some $\sigma \in \Sigma$, and conversely, every $\sigma$-step is an edge-step for some edge $e \in E$. Hence these definitions may seem redundant. However, the distinction between the two types of transitions would become obvious if there is a discrete state controller, defined as a function $C : Q \times X \rightarrow \Sigma_c$, that triggers discrete controllable events to enable a particular transition depending on the present configuration (see for instance [13] for the case of timed automata). But in this work, we shall not have occasion to discuss about discrete state controllers.

**Definition 4** (Trajectories and Executions of a Hybrid System) A hybrid trajectory or simply a trajectory of $H$ is a sequence of configurations $(q_1,x_1),(q_2,x_2),(q_3,x_3), \ldots$, where for each $i \geq 1$, $(q_i,x_i),(q_{i+1},x_{i+1})$ is a step. A trajectory is
1. finite, if the number of steps is finite;

2. an execution, if \((q_1, x_1) \in \text{Init}\), where \((q_1, x_1)\) is the initial configuration; and

3. a finite execution, if it is both.

We now define, for each \(\sigma \in \Sigma\), the set-valued successor function, \(\text{Post}_\sigma: Q \times X \rightarrow 2^Q \times 2^X\). As will be briefly mentioned later (after Definition 6), our definitions lead in a natural way to extract a labeled transition system (see [14]) from a hybrid system.

**Definition 5 (\(\sigma\)-Successors of a Configuration)**

Let \(H = (Q, X, \Sigma, G, E, \text{Init}, f)\) be a hybrid system, \(\sigma \in \Sigma\) and \((q, x) \in Q \times X\). Then, we define \(\text{Post}_\sigma(q, x)\) to be the union of the two sets \(S_1, S_2^\sigma \subseteq Q \times X\), where

1. \(S_1 = \{(q', x') : ((q, x), (q', x'))\}\) is a time-step for some \(t \geq 0\); and

2. \(S_2^\sigma = \{(q'', x'') : ((q', x'), (q'', x''))\}\) is a \(\sigma\)-step for some \((q', x') \in S_1\).

Further, if \((q_s, x_s) \in \text{Post}_\sigma(q, x)\), then \((q_s, x_s)\) is called a \(\sigma\)-successor of \((q, x)\).

The set \(S_1\), as in the first part of this definition, is of main interest to us in the later sections of this paper. Specifically, we will deal with the set \(\text{Reach}_\sigma^G(X_0)\), where \(q \in Q\) and \(X_0 \subseteq G(q)\), defined as

\[
\text{Reach}_\sigma^G(X_0) = \{ x : ((q, x_0), (q, x)) \text{ is a time-step for some } t \geq 0 \text{ and } x_0 \in X_0 \}.
\]

(1)

In the subsequent sections, we will be concerned more with this and a related set. But for now, we shall proceed with our discussion with the following definition:

**Definition 6 (\(\sigma\)-Predecessors of a Configuration)**

Let \(H = (Q, X, \Sigma, G, E, \text{Init}, f)\) be a hybrid system, \(\sigma \in \Sigma\) and \((q, x) \in Q \times X\). Then, we define \(\text{Pre}_\sigma(q, x)\) to be the union of the two sets \(P_1, P_2^\sigma \subseteq Q \times X\), where

1. \(P_1 = \{(q', x') : ((q', x'), (q, x))\}\) is a time-step for some \(t \geq 0\); and

2. \(P_2^\sigma = \{(q'', x'') : ((q'', x''), (q', x'))\}\) is a \(\sigma\)-step for some \((q', x') \in P_1\).

Further, if \((q_p, x_p) \in \text{Pre}_\sigma(q, x)\), then \((q_p, x_p)\) is called a \(\sigma\)-predecessor of \((q, x)\).

Note that the notion of \(\sigma\)-successor generalizes the notion of \(\sigma\)-step by including time-steps. The set valued function \(\text{Post}_\sigma(q, x)\) defines, in a natural way, a transition relation, \(\xrightarrow{\sigma} \subseteq (Q \times X) \times (Q \times X)\), as follows: \((q, x) \xrightarrow{\sigma} (q', x')\), if \((q', x') \in \text{Post}_\sigma(q, x)\). The transition relation \(\xrightarrow{\sigma}\) is also called \(\sigma\)-transition relation. Similarly, the set valued function \(\text{Pre}_\sigma(q, x)\) defines another transition relation, \(\xleftarrow{\sigma} \subseteq (Q \times X) \times (Q \times X)\), as follows: \((q, x) \xleftarrow{\sigma} (q', x')\), if \((q, x) \in \text{Pre}_\sigma(q', x')\).

For two configurations \((q_1, x_1)\) and \((q_2, x_2)\), if \(((q_1, x_1), (q_2, x_2))\) is a \(\sigma\)-step, then \((q_2, x_2) \in \text{Post}(q_1, x_1)\) and \((q_1, x_1) \in \text{Pre}(q_2, x_2)\); hence, in this case, we
have \((q_1, x_1) \xrightarrow{\sigma} (q_2, x_2)\) and \((q_1, x_1) \xleftarrow{\sigma} (q_2, x_2)\). This may mislead us to get the false impression that \(\xrightarrow{\sigma}\) and \(\xleftarrow{\sigma}\) are the same; and to avoid any such possible confusion, we emphasize that, in general, it is not true that \((q_1, x_1) \xrightarrow{\sigma} (q_2, x_2)\) implies \((q_1, x_1) \xleftarrow{\sigma} (q_2, x_2)\); and the same with the converse statement. Hence, the two relations \(\xrightarrow{\sigma}\) and \(\xleftarrow{\sigma}\) are not the same.

The definitions of \(\text{Post}_\sigma\) and \(\text{Pre}_\sigma\) can be extended straightforwardly to subsets of \(Q \times X\), as follows: for \(S \subseteq Q \times X\), define

\[
\text{Post}_\sigma(S) = \bigcup_{(q,x) \in S} \text{Post}_\sigma(q,x), \quad \text{and} \quad \text{Pre}_\sigma(S) = \bigcup_{(q,x) \in S} \text{Pre}_\sigma(q,x).
\]

Finally, define

\[
\text{Post}(S) = \bigcup_{\sigma \in \Sigma} \text{Post}_\sigma(S), \quad \text{and} \quad \text{Pre}(S) = \bigcup_{\sigma \in \Sigma} \text{Pre}_\sigma(S).
\]

We sometimes refer to \(\text{Post}\) and \(\text{Pre}\) as the 1-step transition functions\(^2\). More generally, the \(k\)-step transition functions \(\text{Post}^k\) and \(\text{Pre}^k\) are defined inductively as follows: For \(S \subseteq Q \times X\),

\[
\text{Post}^1(S) = \text{Post}(S), \quad \text{and for} \ k \geq 2, \ \text{Post}^k(S) = \text{Post}(\text{Post}^{k-1}(S)).
\]

Similarly,

\[
\text{Pre}^1(S) = \text{Pre}(S), \quad \text{and for} \ k \geq 2, \ \text{Pre}^k(S) = \text{Pre}(\text{Pre}^{k-1}(S)).
\]

Finally, define \(\text{Post}^*\) and \(\text{Pre}^*\) as

\[
\text{Post}^*(S) = \bigcup_{k \geq 1} \text{Post}^k(S), \quad \text{and} \quad \text{Pre}^*(S) = \bigcup_{k \geq 1} \text{Pre}^k(S).
\]

### 2.2 Reachability Problem for Hybrid Systems

In the notation just discussed, the reachability problem for a hybrid system \(H = (Q, X, \Sigma, G, E, \text{Init}, f)\), where \(\text{Init}\) is not necessarily specified in advance, may be posed as follows:

**ReachProblem1:** For two subsets \(S_1\) and \(S_2\) of \(Q \times X\), is there a finite trajectory, \((q_1, x_1), (q_2, x_2), \ldots, (q_N, x_N)\), for some \(N \geq 1\), such that \((q_1, x_1) \in S_1\) and \((q_N, x_N) \in S_2\)?

This may also be rephrased as

**ReachProblem2:** For two subsets \(S_1\) and \(S_2\) of \(Q \times X\),

\[
\text{whether} \ \text{Post}^*(S_1) \cap S_2 \neq \emptyset \ ? \quad \text{equivalently:} \quad \text{whether} \ S_1 \cap \text{Pre}^*(S_2) \neq \emptyset \ ?
\]

If the initial set \(\text{Init}\) is specified in advance, then, with \(S_1 = \text{Init}\), the reachability problem **ReachProblem1** becomes

\(^2\)This should not be confused with the notion of step as defined in Definition 6.
ReachProblem3: For a subset S of Q × X, is there a finite execution of H with final configuration in S?

There is a counterpart to the reachability problem, called the avoidance problem, which may be posed as follows:

AvoidProblem: For two subsets S_1 and S_2 of Q × X, whether \( Post^*(S_1) \cap S_2 = \emptyset \) ?

\( \text{equivalently: whether } S_1 \cap Pre^*(S_2) = \emptyset \)?

In the sequel, we restrict our attention to ReachProblem1 or ReachProblem2. We observe that the answers to these questions depend, in general, not only on the hybrid system H, but also on the class \( \mathcal{C} \) of subsets of Q × X that are under consideration. Informally, the class \( \mathcal{C} \) is required to

1. include a specified class of sets \( S \), where \( S \) may consist of the initial set, sets defined by state invariance conditions, those defined by guard conditions, and domains and ranges of reset maps of edges (and also final sets, if specified),

2. be closed under the boolean set operations of union and complimentation, and under the functions \( Post_{\sigma} \) and \( Pre_{\sigma} \), and

3. have an effective decision procedure for answering questions such as, for two sets \( S_1, S_2 \in \mathcal{C} \), whether \( S_1 = S_2 \) or not.

A formal presentation of these notions is beyond the scope of this work, although a brief discussion may be found in Appendix A. For more details, the interested reader may refer to the references on Model Theory, such as [19, 20, 21, 22]. But, for our purposes, we will be content with the following (somewhat informal) definition:

**Definition 7** An algorithm for the reachability problem is said to be

1. a decision procedure, if the algorithm stops after a finite number of steps with the correct answer, where the answer can be either yes or no.

2. a semi-decision procedure, if the algorithm

   (a) never stops with an incorrect answer, and

   (b) always stops after a finite number of steps with the correct answer, whenever the answer is yes.

A hybrid system H is decidable, if there is a decision procedure for the reachability problem for H.

Obviously, for a hybrid system H, if there are semi-decision procedures for both reachability problem and avoidance problem, then H is decidable.

We describe below a schematic semi-decision procedure for the reachability problem of a hybrid system:
### Semi-Decision Procedure for ReachProblem2

**Input**: Sets $S_1$ and $S_2$

**Output**: "yes", if there is a trajectory from $S_1$ to $S_2$

**Initialization**

$S := S_1$

**while** $S \cap S_2 = \emptyset$

$S := \text{Post}(S)$

**end while**

return "yes"

In the later part of this section, we shall be mainly concerned with the computational aspects of the Post-operator appearing in the while-loop of the above schematic.

### 2.3 Computation of the Post-operator

We describe here a computational approach to finding $\text{Post}(S)$, $S \subseteq Q \times \mathbb{R}^n$, as in the semi-decision procedure for ReachProblem2 discussed in Sec. 2.2.

Recall that, within a discrete state $q \in Q$, the continuous dynamics of $x$ are specified by

$$\frac{dx}{dt} = f(q, x)$$

with the initial conditions $x(0) = x_0 \in X_0 \subseteq X = \mathbb{R}^n$. We begin our discussion with the following definition.

**Definition 8** Let $\mathbb{R}^+$ denote the set of nonnegative real numbers. Then, for each discrete state $q \in Q$, the flow associated with equation (3) is a function $\phi_q : \mathbb{R}^n \times \mathbb{R}^+ \rightarrow \mathbb{R}^n$, defined as $\phi_q(x, t) = \gamma_x(t)$, $t \geq 0$ and $x \in \mathbb{R}^n$, where the function $\gamma_x : \mathbb{R}^+ \rightarrow \mathbb{R}^n$ satisfies

$$\frac{d\gamma_x(t)}{dt} = f(q, \gamma_x(t)),$$

with the initial condition $\gamma_x(0) = x$.

Recall the set of reachable phases $\text{Reach}^G_q$ as defined in (1). In the following definition, the same is defined in terms of the flow function $\phi_q$:

**Definition 9** Let $q \in Q$ and $X_0 \subseteq \mathbb{R}^n$. Then, the set of reachable continuous phases in the state $q$, is the set $\text{Reach}^G_q(X_0)$ defined as

$$\text{Reach}^G_q(X_0) = \{x \in \mathbb{R}^n : \exists x_0 \in X_0 \exists t \geq 0 \ x = \phi_q(x_0, t) \text{ and } \forall s \ 0 \leq s \leq t \Rightarrow \phi_q(x_0, s) \in G(q)\}$$

We now define another operator, called the projection operator, as follows:

**Definition 10** Let $S \subseteq Q \times X$. Then, the projection of $S$ on $X$, is the set $\pi_x(S) = \{x \in X : (q, x) \in S\}$. 

8
In terms of the operators $\pi_X$ and $\text{Reach}^G_q$, the operator $\text{Post}$ may be rewritten as

$$\text{Post}(S) = \bigcup_{e=(q_e, X_e, \sigma_e, r_e, q'_e) \in E, \; q=q_e} \{q'_e\} \times [r_e \circ \text{Reach}^G_q \circ \pi_X(S)], \quad (4)$$

where, for two operators $g$ and $h$, $g \circ h$ is the composition of the two operators. With regard to computational issues, the set operators $\pi_X$ and $r_e$ may not pose difficulties (provided $r_e$ is suitably specified). Hence the problem reduces to that of computation of $\text{Reach}^G_q$, and effectively representing the resulting set.

Referring to the definition of $\text{Reach}^G_q$ as in (3), the significant and challenging task is the elimination of quantifiers, whenever possible. But, as shown in [23], not all theories of the real number system may admit quantifier elimination (see also the discussion presented in [12] and in Appendix A). This fact provides a motivation for the study of alternative methods (without requiring quantifier elimination method) for computing the set $\text{Reach}^G_q(X_0)$.

Specifically, in the remaining part of the work, we shall be concerned with computation of the operator $\text{Reach}^G_q$ and with representation of the resulting sets. Below is a concise description of the questions that we shall be interested in and of the organization of the rest of the paper:

1. How to effectively compute the set operator $\text{Reach}^G_q$, at least when the global invariance condition is not imposed but a time bound is specified? In this case, we are interested in the set $\text{Reach}^G_q(X_0, [0, \tau])$, $\tau \geq 0$, $X_0 \subset \mathbb{R}^n$, defined as

$$\text{Reach}^G_q(X_0, [0, \tau]) = \{x \in \mathbb{R}^n : \exists x_0 \in X_0 \exists t \leq \tau \text{ and } x = \phi_q(x_0, t)\}.$$

In Sec. 3, we present a schematic algorithm for computing $\text{Reach}^G_q(X_0, [0, \tau])$. This algorithm is based on a generalization of the method described in [8].

2. How to compute – either exactly or approximately – the set operator $\text{Reach}^G_q$ when the global invariance condition is imposed (without time bound). In this case, we have to deal with the set operator $\text{Reach}^G_q$ as defined in (3), and obviously, it would be best to find an algorithm based on quantifier elimination. But, as mentioned earlier, we do not assume that quantifier elimination method is feasible, hence we may have to find an algorithm that computes an approximation to $\text{Reach}^G_q$. In Sec. 3, we indicate how to extend the algorithm for computing $\text{Reach}^G_q(X_0, [0, \tau])$, as mentioned above, to compute an under-approximation or an over-approximation (depending on which one is preferred) to the set $\text{Reach}^G_q(X_0)$.

3. Finally, how to represent the reach sets obtained by the operators $\text{Reach}^G_q$ and $\text{Reach}^G$, i.e. the sets $\text{Reach}^G_q(X_0, [0, \tau])$ or $\text{Reach}^G(X_0)$ as the case may be, such that boolean set operations such as union and complementation can be performed efficiently. This requires representation of the reach sets in terms of sets that are simple and easy to handle, such as polyhedral sets and subalgebraic sets. In [12], an algorithm algorithm for over-approximating the reach sets using polyhedral sets is presented. In Sec. 4, we describe another algorithm for over-approximating the reach
sets with polyhedral sets. The algorithm presented in this paper differs from that of [12], but if these two methods – that described in Sec. 4 and that of [12] – are used together, then better results of over-approximation may be obtained.

The remaining part of the paper is organized as follows. In Sec. 3 and Sec. 4, we restrict our attention to these issues. We illustrate these algorithms with simple examples in Sec. 5, while Sec. 6 summarizes the paper.

3 Computational Approaches for Finding Reach Sets

3.1 Preliminary Discussion on Reach Sets

Let $H = (Q, X, \Sigma, G, E, \text{Init}, f)$ be a hybrid system. For $q \in Q$, consider the equation

$$\frac{dx}{dt} = f(q, x)$$

(5)

with the initial conditions $x(0) = x_0 \in X_0 \subseteq X = \mathbb{R}^n$. It is customary to assume that $f(q, \cdot)$ is Lipschitz continuous in the second variable, in order to ensure existence and uniqueness of solutions to the system of differential equations in (5). Further, we assume that the initial set $X_0$ is closed. Let, as before, the flow (refer to Definition 8) associated with (5) be $\varphi_{q}(x, t)$ and let $\text{Reach}_q(X_0, t) = \{\varphi_q(x_0, t) : x_0 \in X_0\}$ be the set of phases reached at time $t \geq 0$ in state $q$ with initial conditions in $X_0$. Further, let

$$\text{Reach}_q(X_0, [0, \tau]) = \bigcup_{0 \leq t \leq \tau} \text{Reach}_q(X_0, t), \quad \tau \geq 0,$$

and

$$\text{Reach}_q(X_0, [0, \infty)) = \bigcup_{\tau > 0} \text{Reach}_q(X_0, [0, \tau]).$$

We shall now define, for a set $X_q$, such that $X_0 \subseteq X_q \subseteq \mathbb{R}^n$, another set $\text{Reach}'_q(X_0, X_q, [0, \infty))$. For this purpose, we first define, for each $x \in X_0$, a number $\tau_x$ as follows:

$$\tau_x = \inf\{t > 0 : \phi_q(x, t) \not\in X_q\},$$

(6)

where we assume that if $\tau_x = \infty$, then $\phi_q(x, \infty)$ denotes the set of all $\omega$-limit points (see, for instance, [17]) of $\phi_q(x, t)$, for $t \geq 0$, and

$$\{\phi_q(x, t) : 0 \leq t \leq \infty\} = \{\phi_q(x, t) : 0 \leq t < \infty\} \bigcup \phi(x, \infty).$$

Further, let $\Theta_q(x, X_q) \subseteq \mathbb{R}^n$ be defined as follows:

$$\Theta_q(x, X_q) = \begin{cases} 
\{\phi_q(x, t) : 0 \leq t \leq \tau_x\}, & \text{if } \tau_x < \infty \text{ and } \phi(x, \tau_x) \in X_q, \\
\{\phi_q(x, t) : 0 \leq t \leq \infty\}, & \text{if } \tau_x = \infty \text{ and } \phi(x, \infty) \subseteq X_q, \\
\{\phi_q(x, t) : 0 \leq t < \tau_x\}, & \text{otherwise}.
\end{cases}$$

Then the set $\text{Reach}'_q(X_0, X_q, [0, \infty))$ is defined as

$$\text{Reach}'_q(X_0, X_q, [0, \infty)) = \bigcup_{x \in X_0} \Theta_q(x, X_q).$$
Now, by taking $X_q = G(q)$ in the above definition, we have

$$\text{Reach}^G_q(X_0) = \text{Reach}'_q(X_0, X_q, [0, \infty]),$$

where $\text{Reach}^G_q(X_0)$ is defined as in (3), reproduced below for convenience:

$$\text{Reach}^G_q(X_0) = \{x \in \mathbb{R}^n : \exists x_0 \in X_0 \exists t \geq 0 x = \phi_q(x_0, t) \text{ and } \forall s 0 \leq s \leq t \Rightarrow \phi_q(x_0, s) \in G(q)\}.$$

Throughout the rest of the discussion, we fix a state $q$, and consider the problem of computing the sets $\text{Reach}_q(X_0, [0, \tau])$ and $\text{Reach}'_q(X_0, X_q, [0, \infty])$.

### 3.2 A Computational Method: Generalized Face-Lifting Algorithm

When $X_0$ is suitably specified (such as, for example, a rectangle), this problem may be solved by finding the solutions to (3) with $x(0)$ on the boundary of $X_0$. This results in evolving the boundary of $X_0$. More precisely, let $S_0$ be the boundary of $X_0$, and $X(\tau) = X_0 \cup_{0 \leq t \leq \tau} \{\phi_q(x_0, t) : x_0 \in S_0\}$. Then $X(\tau) = \text{Reach}_q(X_0, [0, \tau])$ (see Appendix A for a proof). This is the idea underlying the computational approach, called face lifting method, described in [8].

In this section, we shall study this in considerably more general setting, and describe an algorithm, called generalized face lifting method. It may be noted that the method described in [8] does not assume that a global invariance requirement is imposed; hence the algorithm of [8] computes only $\text{Reach}_q(X_0, [0, \tau])$. However, we shall extend our method for computing $\text{Reach}_q(X_0, [0, \tau])$ for computing an approximation to the set $\text{Reach}^G_q(X_0)$, where the approximation can chosen to be either under-approximation or over-approximation.

Further, let $S_0^+ \subseteq S_0$ be the set of those boundary points of $X_0$, such that the solution with initial point in $S_0^+$ extends into $X_0^+ = \mathbb{R}^n \setminus X_0$, i.e.,

$$S_0^+ = \{x_0 \in S_0 : \exists \epsilon = \epsilon(x_0) > 0 \text{ such that } \phi_q(x_0, t) \in X_0^+, \forall t \in (0, \epsilon)\},$$

(7)

and define $X^+(\tau) = X_0 \cup_{0 \leq t \leq \tau} \{\phi_q(x_0, t) : x_0 \in S_0^+\}$. It turns out that $X^+(\tau) = \text{Reach}_q(X_0, [0, \tau])$ (see Appendix A). Therefore, in order to find $\text{Reach}_q(X_0, [0, \tau])$, we have to find only those solutions for which the initial conditions are in $S_0^+$.

If $S_0^+$ can be found explicitly (by inspection of $f$ and $S_0$), then the problem reduces justifiably to finding the solutions with initial conditions in $S_0^+$. Otherwise, we may have to find a means of obtaining an outer approximation to $S_0^+$, i.e., a set $S^+_1$ satisfying $S_0^+ \subseteq S^+_1 \subseteq S_0$. We suggest a way to find such a set. For this purpose, we assume that $X_0$ is specified as follows: there is a continuously differentiable function, $\ell : \mathbb{R}^n \rightarrow \mathbb{R},$ such that if $x \in \hat{X}_0$ then $\ell(x) < 0$ (where $\hat{X}_0$ denotes the interior of $X_0$, i.e., the largest open set contained in $X_0$), and if $x \in X_0^+$ then $\ell(x) > 0$. Hence $\ell(x) = 0$ on $S_0$. Define

$$S^+_1 = \{x \in S_0 : \nabla \ell(x) \cdot f(q, x) \geq 0\}.$$
It may be easily shown that $S_0^+ \subseteq S_1^+$ (see Appendix A). Further, we observe that the set of points for which $\nabla l(x) \cdot f(q, x) > 0$, constitutes an inner approximation to $S_0^+$. The method of finding an $S_1^+$ as described above can be extended easily to the situation where $X_0$ is specified as the set of intersection of finitely many sets of the form $\ell_i(x) \leq 0$, with the strict inequality in $X_0$.

With this notation, we proceed to describe a schematic algorithm for finding $X^+(\tau)$ for $\tau \geq 0$. In order to exploit the semigroup property of the reach set, i.e., $\text{Reach}_q(X_0, [0, t]) = \text{Reach}_q(\text{Reach}_q(X_0, [0, s]), [0, t - s])$, for any $t$ and $s$, with $0 \leq s \leq t$, we consider a sequence of time intervals $[0, \tau_1], [\tau_1, \tau_2], [\tau_2, \tau_3], \ldots$, where $0 < \tau_1 < \tau_2 < \tau_3 < \ldots$, and $\tau_i \to \infty$, as $i \to \infty$. It may be convenient to choose for some $\tau > 0$, $\tau_i = i\tau$, $i = 0, 1, 2, \ldots$, although we do not require such an assumption in the algorithm.

**Procedure for $\text{Reach}_q(X_0, [0, \tau])$**

initialize: $X^+(0) := X_0$, $F_0 := S_1^+$ and $i := 0$

while $\tau > \tau_i$ and $F_i \neq \emptyset$

if $\tau < \tau_{i+1}$ then

$\Delta_i := \tau - \tau_i$

else

$\Delta_i := \tau_{i+1} - \tau_i$

endif

$T_i := \{\phi_q(x, t) : t \in [0, \Delta_i], \; x \in F_i\}$

/* this computational step requires special attention! */

$X^+(\tau_i + \Delta_i) := X^+(\tau_i) \cup T_i$

$F_{i+1} := \{\phi(x, \Delta_i) : x \in F_i\}\setminus X^+(\tau_i)$

$i := i + 1$

end while

In the above schematic, the initialization step “$F_0 := S_1^+$” could be replaced with “$F_0 := S_0^+$”, as it is not necessary to initialize $F_0$ to $S_1^+$. However, before proceeding further, it must be mentioned that with reference to this algorithm, we assume that the computational steps “$T_i := \{\phi_q(x, t) : t \in [0, \Delta_i], \; x \in F_i\}$” and “$F_{i+1} := \{\phi(x, \Delta_i) : x \in F_i\}\setminus X^+(\tau_i)$” can be performed effectively.

We now describe a method for extending this procedure to another procedure that computes an approximation to $\text{Reach}_q^*(X_0, X_q, [0, \infty))$. But since $\text{Reach}_q^*(X_0) = \text{Reach}^*(X_0, G(q), [0, \infty))$, the procedure to be described below finds an approximation to $\text{Reach}_q^*(X_0)$, when $X_q = G(q)$. The approximation can be chosen to be either under-approximation or over-approximation, depending on a flag “under approximate”, passed as input to the algorithm (refer to the schematic algorithm described below).

To facilitate the discussion, we consider the flow $\psi_q(x, t)$, $t \geq 0$, $x \in X = \mathbb{R}^n$, corresponding to the differential equation

$$\frac{dx}{dt} = -f(q, x)$$

(8)

with the initial conditions $x(0) = x_0 \in X = \mathbb{R}^n$. Now since the flow $\psi_q$ has the opposite direction to that of $\phi_q$, for any $t \geq 0$ and two subsets $X_1$
and $X_2$ of $\mathbb{R}^n$, we have, $X_2 = \phi_q(X_1, t)$ if and only if $X_1 = \psi_q(X_2, t)$. The function $\psi_q$ will be used in the procedure for computing an approximation to $\text{Reach}'(X_0, X_q, [0, \infty))$, for finding, in each iteration, the set of initial conditions in $F_i$ corresponding to which the solutions for the time interval $[0, \Delta_i]$ may violate the global invariance requirement. With this, we present the schematic algorithm as follows:

```
Procedure for Reach'(X_0, X_q, [0, \infty))

precondition: $X_0 \subseteq X_q$

boolean input flag: under_approximate
/* under_approximate = 1, if under-approximation is preferred */
/* by default, the procedure over-approximates the reach set */

initialize: $X^+(0) := X_0$, $F_0 := \mathcal{S}_1^+$ and $i := 0$

while $F_i \neq \emptyset$ do
    $\Delta_i := \tau_{i+1} - \tau_i$
    $T_i := \{ \phi_q(x, t) : t \in [0, \Delta_i], \ x \in F_i \}$

    if $T_i \subseteq X_q$ then /* global invariance not violated */
        $X^+(\tau_i + \Delta_i) := X^+(\tau_i) \cup T_i$
        $F_{i+1} := \{ \phi_q(x, \Delta_i) : x \in F_i \} \setminus X^+(\tau_i)$
    else /* global invariance violated by at least one trajectory */
        $U_i := T_i \setminus X_q$
        $V_i := \{ \psi_q(x, t) : t \in [0, \Delta_i], \ x \in U_i \}$
        $F'_i = F_i \setminus V_i$
        $T'_i := \{ \phi_q(x, t) : t \in [0, \Delta_i], \ x \in F'_i \}$

        if under_approximate = 1 then
            $X^+(\tau_i + \Delta_i) := X^+(\tau_i) \cup U_i$
        else
            $X^+(\tau_i + \Delta_i) := X^+(\tau_i) \cup T'_i$
        end if

        $F_{i+1} := \{ \phi_q(x, \Delta_i) : x \in F'_i \} \setminus X^+(\tau_i)$
    end if

    $i := i + 1$
end while
```

As with the previous algorithm, the initialization step “$F_0 := \mathcal{S}_1^+$” could be replaced with “$F_0 := \mathcal{S}_0^+$”, depending on convenience. Termination of this procedure may be guaranteed, if certain assumptions are satisfied. The required assumptions are as follows:

1. the set $Y_q = \overline{X_q}$ is compact;

13
2. $X_0$ is closed in $\mathbb{R}^n$; and

3. for every $x_0 \in X_0$, $Y_q$ does not contain any $\omega$-limit points of the trajectory of $\phi_q(x_0, t)$, $t \geq 0$, more precisely,

$$\phi_q(x_0, \infty) \cap Y_q = \emptyset, \ x_0 \in X_0.$$ 

Under these assumptions, it can be shown that (see Appendix C) there is a $\tau_{\text{max}} > 0$, such that for every $x \in X_0$, there is a $\tau(x) > 0$ with $0 < \tau(x) \leq \tau_{\text{max}}$ and $\phi_q(x, \tau(x)) \notin X_q$. Obvious, with such a $\tau_{\text{max}}$, we have $\tau_{x} \leq \tau_{\text{max}}$, for each $x \in X_0$, where $\tau_x$ is as in (3). Termination of the algorithm may be deduced from this as follows: Consider a sequence of sets $F_0^{(i)}$, $i \geq 0$, defined as $F_0^{(i)} = \{ x \in S_0 : \phi_q(x, \tau_i) \notin F_i \}$. It is easy to check that $F_0^{(i)}$ is a non-increasing sequence of sets, i.e., $F_0^{(i+1)} \subseteq F_0^{(i)}$. Let $x \in X_0$, and let $k$ be a nonnegative integer such that $\tau_k \leq \tau_x < \tau_{k+1}$. Now, during the $k$th iteration of the while-loop, the global invariance condition (the condition that $T_k \subseteq X_q$) is violated, since $\phi_q(x, t) \notin X_q$, for some $t$ such that $\tau_x \leq t < \tau_{k+1}$. Let $\tau_x' \in [\tau_x, \tau_{k+1})$ be a time instant such that $\phi_q(x, \tau_x') \notin X_0$, and let $t_x = \tau_x' - \tau_x$. With this choice of $t_x$, we have both $0 \leq t_x \leq \Delta$ and $\phi_q(x, \tau_k + t_x) \notin X_q$. Thus, $\phi_q(x, \tau_k + t_x) \in U_k = T_k \setminus X_q$ (refer to the schematic above). Let $y = \phi_q(x, \tau_k + t_x)$, so $\phi_q(x, \tau_k) = \psi_q(y, t_x)$. But, since $y \in U_k$, $\psi_q(y, t_x) \in V_k$ (refer to the schematic). Hence $\phi_q(x, \tau_k) \in V_k$ and $\phi_q(x, \tau_k) \notin F_k = F_k \setminus V_k$. Therefore $\phi_q(x, \tau_{k+1}) \notin F_{k+1}$, implying that $x \notin F_0^{(k+1)}$. By the monotonicity of the sequence of sets $F_0^{(i)}$, we have $x \notin F_0^{(i)}$, for any $i \geq k + 1$. Finally, let $m \geq 0$ be an integer such that $\tau_m \leq \tau_{\text{max}} < \tau_{m+1}$. Now since $\tau_x \leq \tau_{\text{max}}$, $x \notin F_0^{(m+1)}$, for any $x \in X_0$. Hence $F_0^{(m+1)} = \emptyset$, so $F_{m+1} = \emptyset$, satisfying the terminating condition of the while-loop of the algorithm, after at most $m$ number of iterations.

In the rest of the paper, we discuss approximation of the set $T_i$ appearing in the second line of the while-loop of the above schematic (see also the schematic for $\text{Reach}(X_0, [0,\tau])$), when $f(q, x)$ is of the form $f(q, x) = A_q x$, where $A_q$ is a constant $n \times n$ matrix with real entries and the initial set $X_0$ is a polyhedron. For convenience, we drop the subscript $q$ in $A_q$, and deal with the linear system

$$\frac{dx(t)}{dt} = Ax,$$

where $A$ is a constant $n \times n$ matrix with real valued entries.

## 4 Representation of the Reach Sets

### 4.1 Preliminary Discussion

In this section, we discuss representation of the reach set $\text{Reach}(X_0, [0,\tau])$, such that boolean set operations such as union and complementation can be performed efficiently. The most convenient representation schemes appear to be representation in terms of the following classes of subsets of $\mathbb{R}^n$:
1. **Polyhedral Sets**: These are the sets which can be written as union of finitely many polyhedra. A polyhedron in $\mathbb{R}^n$ is a set which may be expressed as the intersection of finitely many closed half spaces, i.e., a finite intersection of sets of the form $\{x \in \mathbb{R}^n : a^Tx \leq b\}$, where $a \in \mathbb{R}^n$ and $b \in \mathbb{R}$. It may be observed this class of sets corresponds to (and includes) the class of definable sets in the theory of linear inequalities of $\mathbb{R}$, i.e., the theory obtained when 0 and 1 are the constant elements, $+$ and $-$ are the (binary) function symbols, and $<$ is the (binary) relation symbol (see Appendix A).

2. **Subalgebraic Sets**: These are the sets which can be written as union of finite number of sets defined by polynomial inequalities, i.e., sets that can be expressed as union of finitely many sets, each of which is the intersection of finitely many sets of the form $\{x \in \mathbb{R}^n : p(x) \leq c\}$, where $p : \mathbb{R}^n \to \mathbb{R}$ is a polynomial function with real-valued coefficients and $c \in \mathbb{R}$. This class of sets corresponds to (and includes) the class of sets that are definable in the theory of $\mathbb{R}$ when viewed as an ordered field. The theory of ordered field of $\mathbb{R}$ is the theory obtained by extending the theory of linear inequalities of $\mathbb{R}$ by including a binary function symbol for representing the product of two real numbers, denoted by $\cdot$ (see Appendix A).

It may be observed that, for a general flow function $\phi_q$, the reach set $\text{Reach}_q(X_0, [0, \tau])$ may not be exactly representable in any one of these classes, even if $X_0$ is. Hence, as an alternative, we may have to settle for an approximation of the reach set by sets that belong to the respective classes. In this context, we shall restrict our attention to over-approximation of the reach set with polyhedral sets.

Specifically, we describe an algorithm for reach set over-approximation with polyhedral sets, when the system dynamics (within a discrete state $q$) are specified as

$$\frac{dx}{dt} = Ax, \quad x(0) \in X_0,$$

where the initial set $X_0$ is a polyhedron and $A$ is a constant real valued $n \times n$ matrix. Based on the discussion of the last section, we consider the problem of approximating the reach set of the solutions of the equation

$$\frac{dx}{dt} = Ax, \quad x(0) \in F_0,$$

where $F_0$ is a face of $X_0$.

The main references on approximate reach set computation using polyhedral sets appear to be [11] and [12]. In [11], the initial set is assumed to be convex (not necessarily a polyhedron), and for each time instance, each slice of the “reach tube” (i.e., for each $t \in [\tau_i, \tau_{i+1}]$, the set $\phi(F_i, t)$) is approximated by polyhedra. The method described in [12] assumes $F_i$ to be a polyhedron, and approximates the tube for the time interval $[\tau_i, \tau_{i+1}]$, i.e., the set $T_i = \bigcup_{0 \leq t \leq \tau_{i+1} - \tau_i} \phi(F_i, t)$, by first constructing a polyhedral over-approximation of the reach tube for the time interval and further over-approximating the resulting polyhedron by a “griddy” polyhedron, i.e., a set that may be expressed as a union of unit hypercubes with
integer left-most vertices \([12]\).

Being a convex set, any polyhedron that over-approximates the reach set contains the convex hull of \(T_i\). An algorithm based on this observation (as given in \([12]\)) is to find a “bloated” convex hull of \(F_i\) and \(F_{i+1}\). This method seems to be simple and straightforward, and gives reasonable approximations in a rather short time (refer to \([12]\)). But we may note that those faces of the convex hull of \(F_i\) and \(F_{i+1}\), for which the solution set \(T_i\) lies on one side, need not be bloated. By shrinking the convex hull, an under-approximation is obtained by the same method.

If the time steps are constant, i.e., \(\tau_k = k\tau\), for some \(\tau > 0\), as in \([12]\), then we have to find approximation \(P_0\) only for \(T_0\), since symbolically, \(T_k = e^{(A\tau)T_k-1}\), for \(k \geq 1\), and therefore, \(T_k = e^{(kA\tau)T_0} = e^{(A\tau_k)T_0}\). Thus, \(P_k = e^{(A\tau_k)P_0}\) is the required approximation, and if \(\lambda_j, j = 1, 2, \ldots, N\), are the normals of the faces \(P_k\), then the normals of the faces of \(P_k\) are given by \(e^{(-A^T\tau_k)\lambda_j}\). From this, we may infer that in the subsequent iterations of the algorithm of \([12]\), the convex hull of \(F_i\) and \(F_{i+1}\) need not be computed.

### 4.2 Over-approximating the Reach Set with Polyhedra

In this section, we consider the problem of over-approximating the reach sets, when the dynamics are specified as

\[
\frac{dx(t)}{dt} = Ax(t), \quad t \geq 0,
\]

with the initial conditions \(x(0) \in X_0\). The solution is given explicitly by \(x(t) = e^{(At)}x(0)\). We assume that the initial set \(X_0\) is a polyhedron, defined by

\[
\lambda_i^T x - h_i \leq 0, \quad i = 1, 2, \ldots, \kappa,
\]

where \(\kappa\) is a fixed positive integer. We assume that \(\lambda_i \in \mathbb{R}^n, |\lambda_i| = 1\) and \(h_i \in \mathbb{R}\), for \(1 \leq i \leq \kappa\). Then the boundary set of \(X_0\) consists of sets \(C_j, 1 \leq j \leq \kappa\), of the form

\[
\lambda_j^T x - h_j = 0, \quad \text{and}
\lambda_i^T x - h_i \leq 0, \quad \text{for} \ 1 \leq i \leq \kappa \text{ and } i \neq j .
\]

Further, we consider only those polyhedral initial sets \(X_0\) and matrices \(A\) that satisfy the following assumptions:

(A1) \(X_0\) is compact.

(A2) The set \(S_0^t\), as defined in \([9]\) corresponding to the flow function \(\phi_\kappa(x, t) = e^{(At)}x\), can be expressed as \(S_0^t = C_{j_1} \cup C_{j_2} \cup \ldots \cup C_{j_m}\), where \(1 \leq j_1 < j_2 < \ldots < j_m \leq \kappa\), such that for each \(i\) with \(1 \leq i \leq m\), \(\lambda_j^T Ax - h_{j_i} \geq \delta_i > 0\), for every \(x \in C_{j_i}\).

Referring to these assumptions, it may be mentioned that the main limitation of this method appears to be the restriction imposed by assumption \(A_2\) on the initial set \(X_0\) and the matrix \(A\). In contrast, the method described in \([12]\) does...
not require such an assumption. But for our purposes, we need this assumption.

Now, let $j$ be an index such that $C_j \subseteq S_0^+$, as in assumption (A$_2$). For simplicity and definiteness, let us assume that $j = \kappa$ and put $F_0 = C_\kappa$. Hence, $F_0$ is described by

$$\lambda_i^T x - h_i \leq 0, \quad 1 \leq i \leq \kappa - 1, \quad \text{and}$$
$$\lambda_\kappa^T x - h_\kappa = 0.$$

After some manipulations and rearrangements, if necessary, the above system of linear constraints can be transformed into an equivalent system of linear constraints, which also describes $F_0$ but is of the following form:

$$\begin{cases}
  a_i^T x - b_i \leq 0, \quad 1 \leq i \leq k - 1, \quad \text{and} \\
  a_k^T x - b_k = 0,
\end{cases} \quad (12)$$

where $k \leq \kappa$, $a_k = \lambda_\kappa$, and for $1 \leq i < k$, $|a_i| = 1$ and $a_i^T a_k = 0$. It may be observed that, since $|\lambda_k| = 1$ initially, we have $|a_k| = 1$ as well. Also, by assumption (A$_2$), there is a $\delta > 0$, such that for all the vectors $x \in F_0$, $a_k^T A x \geq \delta$. Further, we assume that the system of linear constraints in (12) is consistent and that each of the constraints is linearly independent from the remaining constraints.

With this construction, our objective is to describe an algorithm to find a polyhedron $P_0$ as an over-approximation to the set

$$T_0 = \{ e^{(At)} x_0 : x_0 \in F_0, \quad t \in [0, \Delta] \}, \quad (13)$$

where $\Delta$ is a small positive number for which the the following condition holds:

(C$_1$) for some $\delta_0 > 0$, $a_k^T A e^{(At)} x_0 \geq \delta_0$, for all $x_0 \in F_0$ and for all $t \in [-\Delta, \Delta]$.

The existence of such a $\Delta$ can be assured by our assumptions (A$_1$) and (A$_2$). Later, we shall derive an estimate for such a $\Delta$, for any fixed $\delta_0 > 0$ such that $\delta_0 < \delta$. In comparison, the method of [12] does not impose any such restrictions on $\Delta$. But, it may be mentioned that, in both cases, the accuracy of approximation of either method depends on how small a value is chosen for the parameter $\Delta$. The two conditions (C$_2$) and (C$_3$) stated below follow from condition (C$_1$):

(C$_2$) $\alpha_k^T e^{(At)} x_0 - b_k > 0$, for all $x_0 \in F_0$ and for all $t$ such that $0 < t \leq \Delta$; and

(C$_3$) $\alpha_k^T e^{(At)} x_0 - b_k < 0$, for all $x_0 \in F_0$ and for all $t$ such that $-\Delta \leq t < 0$.

Before proceeding to describe our algorithm, we note that the set $F_\Delta = \{ e^{(A \Delta)} x_0 : x_0 \in F_0 \}$ may be described by the system of linear constraints

$$\begin{cases}
  a_i^T (\Delta) x - b_i \leq 0, \quad i = 1, 2, \ldots, k - 1, \\
  a_k^T (\Delta) x - b_k = 0,
\end{cases}$$

where $a_i(\Delta) = e^{(-A^T \Delta)} a_i$, $1 \leq i \leq k$. Dividing throughout the last equation in the above system by $|a_k(\Delta)|$, we get

$$\begin{cases}
  a_i^T (\Delta) x - b_i \leq 0, \quad i = 1, 2, \ldots, k - 1, \\
  b_k^T x - b_k' = 0,
\end{cases}$$
where \( b_k = \frac{a_k(\Delta)}{|a_k(\Delta)|} \) and \( b'_k = \frac{b_k}{|b_k|} \). Now, we observe that, since \( e(-A^T \Delta) \) is invertible and \( |a_i| = 1 \), each \( a_i(\Delta) \) is nonzero, and since \( a_k \) is orthogonal to \( a_i \), for \( 1 \leq i \leq k-1 \), \( a_i(\Delta) \) and \( a_k(\Delta) \) are pair-wise independent. So, after subtracting from each of the inequalities of the above system an appropriate constant times the last equation, and after normalization, the above system of linear constraints can be transformed into the following system of linear constraints:

\[
\begin{align*}
&b_i^T x - b'_i \leq 0, \quad i = 1, 2, \ldots, k-1, \\
&b_k^T x - b'_k = 0,
\end{align*}
\]

(14)

where, for \( 1 \leq i \leq k \), \( |b_i| = 1 \), and for \( 1 \leq i \leq k-1 \), \( b_i^T b_k = 0 \). Now, by the condition \((C_3)\), for \( 0 \leq t < \Delta \) and \( x_0 \in F_0 \), \( a_k^T(\Delta)e^{(At)} - b_k = a_k^T e^{(A(t-\Delta))} - b_k < 0 \). Hence, we also have, for \( 0 \leq t < \Delta \) and \( x_0 \in F_0 \), \( b_k^T e^{(At)} x_0 - b'_k < 0 \).

In this notation, we now describe a schematic algorithm, the output of which is a set of \( 4k \) parameters, consisting of pairs of vectors and constants, \((\mu_i, c_i)\) and \((\nu_i, d_i)\), \( i = 1, 2, \ldots, 2k \), where for each \( i \), \( \mu_i, \nu_i \in \mathbb{R}^n \) and \( c_i, d_i \in \mathbb{R} \), such that the polyhedron \( P_0 \) of intersection of the \( 4k \) half-spaces defined by \( L_i = \{ x \in \mathbb{R}^n : \mu_i^T x - c_i \leq 0 \} \) and \( L'_i = \{ x \in \mathbb{R}^n : \nu_i^T x - d_i \leq 0 \} \) is an over-approximating polyhedron for the set \( T_0 \) as defined in (13).
Schematic Algorithm for Finding Over-approximating Polyhedron

Reach Set : \( T_0 = \{ e^{(At)} x_0 : t \in [0, \Delta], \ x_0 \in F_0 \} \) (refer to (13))

Output : 4k vectors \( \mu_i, \nu_i \) and 4k real numbers \( c_i, d_i, 1 \leq i \leq 2k \)

/* the 4k half-spaces are \( L_i \) and \( L'_i \), \( 1 \leq i \leq 2k \), where */
/* \( L_i = \{ x \in \mathbb{R}^n : \eta_i(x) = \mu_i^T x - c_i \leq 0 \} \), and */
/* \( L'_i = \{ x \in \mathbb{R}^n : \eta'_i(x) = \nu_i^T x - d_i \leq 0 \} \) */

for \( i = 1, 2, \ldots, k - 1, \) do /* to find \( (\mu_i, c_i) \) and \( (\nu_i, d_i) \) */
\( l_i := \inf \{ l : (a_i^T x - b_i) - l(a_i^T x - b_i) \leq 0, \ \forall x \in T_0 \}; \)
\( \mu_i := a_i - l_i a_k; \quad c_i := b_i - l_i b_k; \)
\( l'_i := \inf \{ l' : (b_i^T x - b_i') + l'(b_i^T x - b_i') \leq 0, \ \forall x \in T_0 \}; \)
\( \nu_i := b_i + l'_i b_k; \quad d_i = b'_i + l'_i b_k; \)
end for

\( \mu_k := -a_k^T; \quad c_k = -b_k; \)
\( \nu_k := b_k^T; \quad d_k = b_k^T; \)
\( l_k := \inf \{ l : (a_k^T x - b_k) - l \leq 0, \ \forall x \in T_0 \}; \)
\( \mu_k := a_k; \quad c_{k+1} := b_k + l_k; \)
\( l'_k := \inf \{ l' : (b_k^T x - b_k') - l' \leq 0, \ \forall x \in T_0 \}; \)
\( \nu_k := -b_k; \quad d_{k+1} = -b_k' + l'_k; \)

for \( i = 1, 2, \ldots, k - 1, \) do /* to find \( (\mu_{i+k+1}, c_{i+k+1}) \) and \( (\nu_{i+k+1}, d_{i+k+1}) \) */
\( l_{i+k} := \inf \{ l : (a_i^T x - b_i) - l \leq 0, \ \forall x \in T_0 \}; \)
\( \mu_{i+k} := a_i; \quad c_{i+k+1} := b_i + l_{i+k}; \)
\( l'_{i+k} := \inf \{ l' : (b_i^T x - b_i') - l' \leq 0, \ \forall x \in F_0 \}; \)
\( \nu_{i+k} := b_i; \quad d_{i+k+1} = b_i' + l'_{i+k}; \)
end for

We first observe that both \( l_i > -\infty \) and \( l'_i > -\infty \). If \( a_i^T Ax \geq \delta > 0 \), as in assumption \( A_2 \), then both \( l_i < \infty \) and \( l'_i < \infty \), as we shall see in Sec. 4.3 below. If \( \Delta \) is small, such that the set \( T_0 \) is nearly a polyhedron, then the above method gives reasonable results. Referring to Step 1 of the first for-loop of the above algorithm, the reason for choosing \( l_i \) to be the infimum over all \( l \) for which \( (a_i^T x - b_i) - l(a_i^T x - b_i) \leq 0, \ \forall x \in T_0 \), is obvious: if \( P_1 \) and \( P_2 \) are two polyhedra such that \( P_1 \) is obtained by the above algorithm and \( P_2 \) is obtained by replacing a constraint \( \mu_i^T x - c_i \leq 0 \), where \( \mu_i \) and \( c_i \) are as in Step 2, with another constraint \( \lambda_i^T x - h_i \leq 0 \), where \( \lambda_i = a_i - l a_k \) and \( h_i = b_i - l b_k \), for some \( l > l_i \), then \( P_1 \subset P_2 \). (An analogous statement holds for each of the remaining parameters, \( l'_i \), \( l'_k \), \( c_{i+i} \), as in the algorithm.)

\[ 3 \] This follows from the observation that if both \( (a_i^T x - b_i) - l(a_i^T x - b_i) \leq 0 \) and \( -(a_i^T x - b_i) = \mu_i^T x - c_i \leq 0 \), where \( l \in \mathbb{R} \) and \( x \in \mathbb{R}^n \), then, for any \( h > l \), \( (a_i^T x - b_i) - h(a_i^T x - b_i) \leq 0 \).
It can be shown that the polyhedron included in the \( k + 1 \) half-spaces, specified by \( L_1, L_2, \ldots, L_k, L_{k+1} \), is a bounded polyhedron (see Appendix D). We may also note that the hyperplanes \( \mu_i^T x - c_i = 0 \) and \( \nu_i^T x - d_i = 0 \), \( 1 \leq i \leq k-1 \), are obtained by rotating the hyperplanes \( a_i^T x - b_i = 0 \) and \( b_i^T x - b'_i = 0 \) about their corresponding intersection with the hyperplanes \( a_i^T x - b_k = 0 \) and \( b_i^T x - b'_k = 0 \), respectively; whereas the hyperplanes \( \mu_{k+1}^T x - c_{k+1} = 0 \) and \( \nu_{k+1}^T x - d_{k+1} = 0 \) are obtained by translating the hyperplanes \( a_i^T x - b_i = 0 \) and \( b_i^T x - b'_i = 0 \), respectively. Similarly, for \( i = 1, 2, \ldots, k-1 \), the hyperplanes \( \mu_i^T x - c_i = 0 \) and \( \nu_i^T x - d_i = 0 \) are obtained by translating the hyperplanes \( a_i^T x - b_i = 0 \) and \( b_i^T x - b'_i = 0 \), respectively.

In remaining part of the section (in Sec. 4.3 below), we shall derive upper bounds for the numbers \( l_i \) and \( l'_i \), \( 1 \leq i \leq 2k-1 \), that are defined in the schematic. If all the \( l_i \) and \( l'_i \) are replaced with their corresponding upper bounds, \( l_i \) and \( l'_i \) respectively, then we obtain conservative estimates – \( \hat{l}_i \)'s and \( \hat{l}'_i \)'s. But unfortunately, the conservative upper bounds that we derive for \( \hat{l}_i \)'s and \( \hat{l}'_i \)'s. However, better accuracy may be obtained if the modified algorithm is used in conjunction with that of [12] (see Fig. 5 in Sec. 5). More precisely, the intersection of the polyhedron obtained by the method described here with that obtained by the method of [12] gives a smaller over-approximating polyhedron, as will be discussed in Sec. 5, while illustrating these algorithms with simple examples.

### 4.3 Upper Bounds for \( l_i \) and \( l'_i \)

In this section, we derive upper bounds for the numbers \( l_i \) and \( l'_i \) appearing in the schematic algorithm. We begin with the following result:

**Claim 1.** For \( i = 1, 2, \ldots, k \), with \( a_i \) and \( b_i \) as in [12] and \( x_0 \in F_0 \) and \( t \in (0, \Delta] \), we have

\[
\left| \frac{(a_i^T e^{(At)} x_0 - a_i^T x_0)}{t} \right| \leq M_0 ||a_i^T A||e^{(||A||\Delta)},
\]

where \( M_0 = \max_{x_0 \in F_0} \{||x_0||\} \).

To prove the claim, we first note that \( (a_i^T e^{(At)} x_0 - a_i^T x_0) = t \times a_i^T A e^{(A\theta)} x_0 \), for some \( \theta \) with \( 0 < \theta < t \), where \( \theta \) may depend on \( t \). Hence, we have

\[
\left| \frac{(a_i^T e^{(At)} x_0 - a_i^T x_0)}{t} \right| = \left| \frac{a_i^T A e^{(A\theta)} x_0}{t} \right|, \quad \text{where } \theta \text{ is such that } 0 < \theta < t
\]

\[
\leq \left| a_i^T A e^{(A\theta)} \right| \times ||x_0||
\]

\[
\leq M_0 ||a_i^T A||e^{(||A||\Delta)}.
\]

With \( b_i \) and \( b'_i \), we have an analogous result, but with a slight modification, as in the following:
Claim 2. For \( i = 1, 2, \ldots, k \), with \( b_i \) and \( b_i' \) as in (12), and \( x_0 \in F_0 \) and \( t \in [0, \Delta) \), we have

\[
\frac{\|b_i^T e^{(At)}x_0 - b_i'^T e^{(A\Delta)}x_0\|}{(\Delta - t)} \leq M_0\|b_i^T A\| e^{\|A\|\Delta)},
\]

where \( M_0 = \max_{x_0 \in F_0} \{\|x_0\|\} \).

We have \( (b_i^T e^{(At)}x_0 - b_i'^T e^{(A\Delta)}x_0) = (t - \Delta) \times b_i^T e^{(A\theta)}x_0 \), for some \( \theta \) with \( t < \theta < \Delta \). Hence

\[
\frac{\|b_i^T e^{(At)}x_0 - b_i'^T e^{(A\Delta)}x_0\|}{(\Delta - t)} = \|b_i^T e^{(A\theta)}x_0\|, \quad \text{for some } t < \theta < \Delta
\]

\[
\leq \|b_i^T A\| e^{\|A\|\Delta)}.
\]

Finally, we shall make the following claim, before deriving upper bounds for \( l_i \)'s and \( l_i' \)'s:

Claim 3. Assume, for some \( \delta > 0 \), \( a_i^T Ax \geq \delta \), for every \( x \in F_0 \), and let \( \delta_0 \) be such that \( 0 < \delta_0 < \delta \). Then there is a \( \Delta > 0 \) such that \( a_i^T e^{(A\theta)}x \geq \delta_0 \), for all \( x \in F_0 \) and for all \( t \in [\Delta, \Delta] \).

In order to prove the claim, we shall derive a conservative estimate for \( \Delta > 0 \) for which the claim holds. Fix a \( \delta_0 \) such that \( 0 < \delta_0 < \delta \). We have

\[
a_i^T e^{(A\theta)}x = a_i^T Ax_0 + \int_0^t a_i^T A^2 e^{(As)}x_0 ds, \quad \text{where } x_0 \in F_0,
\]

\[
\geq a_i^T Ax_0 - \int_0^t \|a_i^T \|\|A\|^2 e^{(\|A\|s)}\|x_0\| ds
\]

\[
\geq a_i^T Ax_0 - M_0\|A\| e^{\|A\|\Delta)} - 1,
\]

where \( M_0 = \max_{x_0 \in F_0} \{\|x_0\|\} \). Therefore, if \( \Delta \) is such that

\[
M_0\|A\| e^{\|A\|\Delta)} - 1 \leq \delta - \delta_0,
\]

then the claim holds.

Upper Bounds for \( l_i \) and \( l_i' \), \( 1 \leq i \leq k - 1 \). Recall that, for \( 1 \leq i \leq k - 1 \),

\[
l_i = \inf \{l : (a_i^T x - b_i) - l(a_i^T x - b_k) \leq 0, \forall x \in T_0\}.
\]

Referring to the right hand side of (13), for every \( x_0 \in F_0 \), since \( a_i^T x_0 - b_k = 0 \), for any real number \( l \), we have \( (a_i^T x_0 - b_i) - l(a_i^T x_0 - b_k) = (a_i^T x_0 - b_i) \leq 0 \). Therefore, we may assume \( x \in T_0 \setminus F_0 \), and \( a_i^T x - b_k > 0 \). Thus \( l_i \) has to be chosen such that

\[
l_i = \sup_{x \in T_0 \setminus F_0} \frac{(a_i^T x - b_i)}{(a_i^T x - b_k)}.
\]

Rewriting the above, we have

\[
l_i = \sup_{0 < t \leq \Delta} \frac{(a_i^T e^{(At)}x_0 - b_i)}{(a_i^T e^{(At)}x_0 - b_k)}
\]

\[
\]
Dividing the numerator and denominator by \( t \), this may be written as

\[
l_i = \sup_{0 < t \leq \Delta} \frac{\alpha_i(x_0, t)}{\beta_i(x_0, t)},
\]

where \( \alpha_i(x_0, t) \) and \( \beta_i(x_0, t) \) are given by

\[
\alpha_i(x_0, t) = \frac{(a_i^T e^{(At)} x_0 - b_i)}{t} \quad \text{and} \quad \beta_i(x_0, t) = \frac{(a_i^T e^{(At)} x_0 - b_k)}{t}.
\]

But, since \( a_i^T x_0 - b_i \leq 0 \), for every \( x_0 \in F_0 \), we have

\[
\frac{(a_i^T e^{(At)} x_0 - b_i)}{t} \leq \frac{(a_i^T e^{(At)} x_0 - a_i^T x_0)}{t} \leq \left| (a_i^T e^{(At)} x_0 - a_i^T x_0) \right|.
\]

By Claim 1, we have

\[
\sup_{0 < t \leq \Delta} \alpha_i(x_0, t) \leq M_0 \| a_i^T A\| e^{\| A\| \Delta}.
\]

As to the denominator, since \( a_i^T x_0 - b_k = 0 \), for every \( x_0 \in F_0 \), we have

\[
\frac{(a_i^T e^{(At)} x_0 - b_k)}{t} = \frac{(a_i^T e^{(At)} x_0 - a_i^T x_0)}{t} = a_i^T e^{(A\theta)} x \geq \delta_0,
\]

where, \( \theta \) is some number in the interval \((0, \tau)\), and the last inequality is due to Claim 3. Hence

\[
\inf_{0 < t \leq \Delta} \beta(x_0, t) \geq \delta_0 > 0.
\]

Combining both inequalities, we have

\[
l_i \leq \sup_{0 < t \leq \Delta} \alpha_i(x_0, t) \leq \inf_{0 < t \leq \Delta} \beta_i(x_0, t) \leq \hat{l}_i \leq M_0 \| a_i^T A\| e^{\| A\| \Delta} \delta_0.
\]

Likewise, \( l'_i \) must be chosen such that

\[
l'_i = \sup_{0 \leq t < \Delta} \frac{\alpha'_i(x_0, t)}{\beta'(x_0, t)},
\]

where \( \alpha'_i(x_0, t) \) and \( \beta'(x_0, t) \) are given by

\[
\alpha'_i(x_0, t) = \frac{(b_i^T e^{(At)} x_0 - b'_i)}{(\Delta - t)} \quad \text{and} \quad \beta'_i(x_0, t) = \frac{(b_i^T e^{(A\Delta)} x_0 - b'_k)}{(\Delta - t)}.
\]

Now, since \( b_i^T e^{(A\Delta)} x_0 - b'_i \leq 0 \), for \( x_0 \in F_0 \),

\[
\frac{(b_i^T e^{(At)} x_0 - b'_i)}{(\Delta - t)} \leq \frac{(b_i^T e^{(At)} x_0 - b_i^T e^{(A\Delta)} x_0)}{(\Delta - t)} \leq \left| (b_i^T e^{(At)} x_0 - b_i^T e^{(A\Delta)} x_0) \right| \leq M_0 \| b_i^T A\| e^{\| A\| \Delta}, \quad \text{by Claim 2}.
\]
For the denominator function, since \( b_k^T e^{(A\Delta)} x_0 - b'_k = 0 \), for every \( x_0 \in F_0 \), we have

\[
\frac{(b_k^T e^{(A\Delta)} x_0 - b'_k)}{(\Delta - t)} = \frac{(b_k^T e^{(A\Delta)} x_0 - b'_k)}{(\Delta - t)} = - b_k^T A e^{(A\theta)} x_0,
\]

where \( \theta \) is some number in the interval \((t, \Delta)\). Now,

\[
-b_k^T A e^{(A\theta)} x_0 = \frac{a_k^T e^{(-A\Delta)} A e^{(A\theta)} x_0}{\|a_k^T e^{(-A\Delta)}\|} \geq \delta_1 = \frac{\delta_0}{\|a_k^T e^{(-A\Delta)}\|}.
\]

Hence, we have the following upper bound for \( l_i' \):

\[
l_i' \leq \sup_{0 < t \leq \Delta, \ x_0 \in F_0} a_k' (x_0, t) \leq \frac{M_0 \|a_i^T A\| e(\|A\|\Delta)}{\delta_1}.
\]

Thus, we obtain the following upper bounds:

\[
l_i \leq \hat{l}_i = \frac{M_0 \|a_i^T A\| e(\|A\|\Delta)}{\delta_0}, \quad 1 \leq i \leq k - 1
\]

\[
l_i' \leq \hat{l}_i' = \frac{M_0 \|a_i^T A\| e(\|A\|\Delta)}{\delta_1}, \quad 1 \leq i \leq k - 1,
\]

\[
\delta_1 = \frac{\delta_0}{\|a_k^T e^{(-A\Delta)}\|}.
\]

**Upper Bounds for** \( l_k, l_k', l_{k+1} \) **and** \( l_k'^{\prime}, 1 \leq i \leq k - 1 \). We have \( l_{i+1} = \inf \{ l : (a_i^T x - b_i) - l \leq 0, \ \forall x \in T_0 \} \), and \( l_k = \inf \{ l : (a_k^T x - b_k) - l \leq 0, \ \forall x \in T_0 \} \). Let \( l_{k+1} = l_k \), so we may consider \( l_{k+1} \), for \( 1 \leq i \leq k \). We have to choose \( l_{k+1} \) such that

\[
l_{k+1} = \sup_{x \in T_0} (a_i^T x - b_i) = \sup_{0 < t \leq \Delta} (a_i^T e^{(A\tau)} x_0 - b_i).
\]

Now, since \((a_i^T x_0 - b_i) \leq 0\), for every \( x_0 \in F_0 \), we have

\[
\sup_{0 < t \leq \Delta, \ x_0 \in F_0} (a_i^T e^{(A\tau)} x_0 - b_i) \leq \sup_{0 < t \leq \Delta, \ x_0 \in F_0} (a_i^T e^{(A\tau)} x_0 - a_i^T x_0)
\]

\[
\leq \sup_{0 < t \leq \Delta, \ x_0 \in F_0} t \times \frac{(a_i^T e^{(A\tau)} x_0 - a_i^T x_0)}{t}
\]

\[
\leq \Delta \times \sup_{0 < t \leq \Delta, \ x_0 \in F_0} \frac{|(a_i^T e^{(A\tau)} x_0 - a_i^T x_0)|}{t}
\]

\[
\leq \Delta \times M_0 \|a_i^T A\| e(\|A\|\Delta).
\]

Similarly, for \( 1 \leq i \leq k - 1 \), we have to choose \( l_{k+1}' \) as follows

\[
l_{k+1}' = \sup_{x \in T_0} (b_i^T x - b_i') = \sup_{0 < t \leq \Delta} (b_i^T e^{(A\tau)} x_0 - b_i').
\]
A calculation similar to the above shows
\[
\sup_{0 \leq t < \Delta} \left( b_i^T e^{(A \Delta)} x_0 - b_i' \right) \leq \Delta \times M_0 \| b_i^T A \| e (\| A \| \Delta)
\]
Finally, for \( l'_k \), we have
\[
l'_k = \sup_{x \in T_0} [- (b_i^T x - b'_i)] = \sup_{0 \leq t < \Delta} (b_i^T e^{(A \Delta)} x_0 - b_i' e^{(A \Delta)} x_0)
\]
Another sequence of similar calculations shows
\[
\sup_{0 \leq t < \Delta} (b_i^T e^{(A \Delta)} x_0 - b_i^T e^{(A \Delta)} x_0) \leq \Delta \times M_0 \| b_i^T A \| e (\| A \| \Delta)
\]
So, to collect all the estimates, we have
\[
l_k \leq \hat{l}_k = M_0 \Delta \| a_i^T A \| e (\| A \| \Delta)
\]
\[
l'_k \leq \hat{l'}_k = M_0 \Delta \| b_i^T A \| e (\| A \| \Delta)
\]
\[
l_{k+i} \leq \hat{l}_{k+i} = M_0 \Delta \| a_i^T A \| e (\| A \| \Delta), \quad 1 \leq i \leq k - 1
\]
\[
l'_{k+i} \leq \hat{l'}_{k+i} = M_0 \Delta \| b_i^T A \| e (\| A \| \Delta), \quad 1 \leq i \leq k - 1
\]

5 Illustration

5.1 Example 1
We first illustrate the schematic algorithm presented in Sec. 3 with the help of an example taken from [8]. Let
\[
\begin{align*}
\dot{x} &= a \\
\dot{y} &= b
\end{align*}
\]
a, b > 0, \((x(0), y(0)) \in X_0 = [0, 1] \times [0, 1].\)
So, for \( t \geq 0 \), \( x(t) = x(0) + at \) and \( y(t) = y(0) + bt \). It is easy to check that \( S_0^+ = \{ 1 \} \times [0, 1] \cup [0, 1] \times \{ 1 \}. \) Therefore
\[
X^+(\tau) = X_0 \bigcup_{0 \leq t \leq \tau} \{ (1 + at, c + bt) : 0 \leq c \leq 1 \} \bigcup_{0 \leq t \leq \tau} \{ (c + at, 1 + bt) : 0 \leq c \leq 1 \}.
\]
This is illustrated in Fig. 1.

5.2 Example 2
In this example, we illustrate the over-approximation algorithm. Let \( x(t) = [x_1(t), x_2(t)]^T \in \mathbb{R}^2 \) satisfy
\[
\begin{align*}
\dot{x}_1 &= -x_2 \\
\dot{x}_2 &= x_1
\end{align*}
\]
\((x_1(0), x_2(0)) \in F_0 = [1, \sqrt{2}] \times \{ 0 \}.
\)
The solution is given explicitly by
\[
\begin{bmatrix}
x_1(t) \\
x_2(t)
\end{bmatrix} =
\begin{bmatrix}
\cos t & -\sin t \\
\sin t & \cos t
\end{bmatrix}
\begin{bmatrix}
x_1(0) \\
x_2(0)
\end{bmatrix}
\]
With $\Delta = \pi/6$, for the time interval $[0, \pi/6]$, the solution set in a parametric form is $T_0 = \{(a \cos t, a \sin t) : a \in [1, \sqrt{2}], 0 \leq t \leq \pi/6\}$. For this example, in the notation of Sec. 4, we have

$$a_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad a_2 = \begin{bmatrix} -1 \\ 0 \end{bmatrix}, \quad a_3 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad b_1 = \sqrt{2}, \quad b_2 = -1, \quad \text{and} \quad b_3 = 0,$$

and

$$b_1 = \begin{bmatrix} \sqrt{2} \\ 0.5 \end{bmatrix}, \quad b_2 = \begin{bmatrix} -\sqrt{2} \\ -0.5 \end{bmatrix}, \quad b_3 = \begin{bmatrix} -0.5 \\ \sqrt{2} \end{bmatrix}, \quad b_1' = \sqrt{2}, \quad b_2' = -1, \quad \text{and} \quad b_3' = 0.$$

The set $T_0$ and the result of the algorithm with $l_i$ and $l'_i$ exactly found as in the schematic of Sec. 4 are shown in Fig. 2, where the horizontal axis corresponds to $x_1$ and the vertical axis corresponds to $x_2$.

Now, recalling the notation of Sec. 4, we may take $\delta = 1$, $\delta_0 = \frac{\sqrt{3}}{2}$. It is
easy to check that $M_0 = \sqrt{2}$ and $\|a_i^T A\| = \|b_i^T A\| = 1$, for $1 \leq i \leq 3$. Hence
\[
l_i \leq \hat{l} = \frac{\sqrt{2} \times 1 \times x_i / \sqrt{3}}{2 \sqrt{2} \times e^{x_i / 6}} \approx 2.7566424
\]
and the same upper bound holds for $l'_i$, since $\delta_1 = \delta_0$ in this example. For $l_k$ and $l'_k$ we have the following bounds:
\[
l_k = l'_k \leq \hat{l} = \sqrt{2} \times \frac{\pi}{6} \times e^{x_i / 6} \approx 1.249999
\]
In this example, the same upper bound, given by $\hat{l}$, holds for $l_{k+1}$ and $l'_{k+1}$. Combining all this, the polyhedron is the intersection of the half-spaces given by $\eta_i(x) \leq 0$ and $\eta'_i(x) \leq 0$, $1 \leq i \leq 6$, where
\[
\eta_1(x) = x_1 - 2.7566424 x_2 - \sqrt{2}, \quad \eta'_1(x) = -0.5122958 x_1 + 2.8873223 x_2 - \sqrt{2},
\eta_2(x) = -x_1 - 2.7566424 x_2 + 1.0, \quad \eta'_2(x) = -2.2443466 x_1 + 1.8873223 x_2 + 1.0,
\eta_3(x) = -x_2, \quad \eta'_3(x) = -x_1 + \sqrt{3} x_2,
\eta_4(x) = x_2 - 1.249999, \quad \eta'_4(x) = 0.5 x_1 - \frac{\sqrt{3}}{2} x_2 - 1.249999.
\eta_5(x) = x_1 - \sqrt{2} - 1.249999, \quad \eta'_5(x) = \frac{\sqrt{3}}{2} x_1 + 0.5 x_2 - \sqrt{2} - 1.249999,
\eta_6(x) = -x_1 + 0.249999, \quad \eta'_6(x) = -\frac{\sqrt{3}}{2} x_1 - 0.5 x_2 + 0.249999.
\]
In this example, $\eta'_1(x) \leq 0$, $\eta_6(x) \leq 0$ and $\eta'_6(x) \leq 0$, are redundant.

Fig. 3 illustrates the result of the algorithm with the first 8 half-spaces defined by $\eta_i(x) \leq 0$ and $\eta'_i(x) \leq 0$, $1 \leq i \leq 4$, where the dashed lines correspond to the lines $\eta_i(x) = 0$ and $\eta'_i(x) = 0$, and the polyhedron included in their intersection is shown in thick lines. The point of intersection of $\eta_1(x) = 0$ and $\eta_2(x) = 0$ is $(4.600138, 1.249999)$, of $\eta_3(x) = 0$ and $\eta'_3(x) = 0$ is $(4.2845099, 1.249999)$, and of $\eta_5(x) = 0$ and $\eta'_5(x) = 0$ is $(0.575162, 0.154114)$. So the vertices in the counter-clockwise order are given by $(\sqrt{2}, 0)$, $(4.8600138, 1.249999)$, $(4.2845099, 1.249999)$, $(\sqrt{3}/\sqrt{2}, 1/\sqrt{2})$, $(\sqrt{3}/\sqrt{2}, 0.5)$, $(0.575162, 0.154114)$ and $(1, 0)$.

Fig. 4 shows the result when the remaining two half-spaces defined by $\eta_5(x) \leq 0$ and $\eta'_5(x) \leq 0$ are also used. Finally, Fig. 5 shows the intersection of the polyhedron obtained by the method reported in this paper with that which may possibly be obtained by the method of [12], the latter being shown in dotted lines. The polyhedron corresponding to the method of [12] is obtained as follows: the convex hull of the sets $[1, \sqrt{2}] \times \{0\}$ and $\{(a \cos \frac{\pi}{6}, a \sin \frac{\pi}{6}) : a \in [1, \sqrt{2}]\}$ is the polygon with vertices in the counter-clockwise order $(\sqrt{2}, 0)$, $(\sqrt{3}/\sqrt{2}, 1/\sqrt{2})$, $(\sqrt{3}/\sqrt{2}, 1/2)$ and $(1, 0)$. Therefore the polygon is the intersection of the half-spaces $\zeta_i(x) \leq 0$ and $\zeta'_i(x) \leq 0$, $i = 1, 2$, where
\[
\zeta_1(x) = -x_2, \quad \zeta'_1(x) = -x_1 + \sqrt{3} x_2, \quad \zeta_2(x) = \frac{1}{\sqrt{2}}(x_1 - \sqrt{2}) - \left(\frac{\sqrt{3}}{\sqrt{2}} - \sqrt{2}\right)x_2, \quad \zeta'_2(x) = -\frac{1}{2}(x_1 - 1) + \left(\frac{\sqrt{3}}{2} - 1\right)x_2.
\]
Now if \( \hat{\epsilon} \) is an upper bound for the bloating parameter \( \epsilon \), then the half-spaces of the over-approximating polyhedron corresponding to the method of [12] are given by \( \hat{\zeta}_i(x) \leq 0 \) and \( \hat{\zeta}'_i(x) \leq 0 \), \( i = 1, 2 \), where

\[
\begin{align*}
\hat{\zeta}_1(x) &= -x_2 + \hat{\epsilon}, \\
\hat{\zeta}'_1(x) &= -x_1 + \sqrt{3}x_2 + \hat{\epsilon}, \\
\hat{\zeta}_2(x) &= 0.70710678x_1 + 0.18946869x_2 - 1 + \hat{\epsilon}, \quad \text{and} \\
\hat{\zeta}'_2(x) &= -0.70710678x_1 - 0.1339746x_2 + 0.70710678 + \hat{\epsilon}.
\end{align*}
\]

The upper bound for the bloating parameter \( \epsilon \) as given in [12] works out to be

\[
\epsilon \leq M_0 \left( e^{||A||\Delta} - 1 - ||A||\Delta - \frac{3}{8}||A||^2\Delta^2 \right) = \sqrt{2} \times \left( e^{(\pi/6)} - 1 - \frac{\pi}{6} - \frac{3\pi^2}{8 \times 36} \right) = \sqrt{2} \times 0.06168464 \approx 0.087235255 < 0.09.
\]

Fig. 5 shows the results when an upper bound for the bloating parameter is chosen to be \( \hat{\epsilon} \approx 0.2 \), where the dotted lines correspond to the line \( \hat{\zeta}_1(x) = 0 \), \( \hat{\zeta}'_1(x) = 0 \), \( \hat{\zeta}_2(x) = 0 \) and \( \hat{\zeta}'_2(x) = 0 \). As may be expected, the polyhedron of intersection of the two polyhedra – one polyhedron bounded by the dashed lines corresponding to the method described here and another bounded by the dotted lines corresponding to the method of [12] – gives better results of over-approximation.

Figure 3: Illustration of polyhedral over-approximation for the solution set \( T_0 \) of Example 2: without additional hyperplanes.

6  Discussion and Conclusion

An important issue of hybrid systems appears to be computability of the reach sets of the continuous variables. From a computational point, both the problems of computation and efficient representation of the reach sets of the continuous
variables are difficult, in general, owing to the limitations of quantifier elimination method. In this context, approximation of the reach sets by more convenient sets, such as polyhedra and subalgebraic sets, is discussed in the literature. In this paper, along with a method for finding the reach sets, an algorithm for over-approximation of the reach sets with polyhedra when dynamics of the continuous variables are specified by linear differential equations and the initial set is a polyhedron. A practical version of the over-approximation algorithm is also discussed in this paper. However, it seems that better results of over-approximation may be obtained by taking the intersection of the polyhedron obtained by the method reported here with that obtained by the method given in [12]. It is hoped that the over-approximation method presented here may be extended to systems with more general dynamics and initial sets.
References

[1] R. Alur, C. Courcoubetis, N. Halbwachs, T. A. Henzinger, P.-H. Ho, X. Nicollin, A. Olivero, J. Sifakis and S. Yovine, “The algorithmic analysis of hybrid systems”, *Theoretical Computer Science*, Vol 138, pp. 3-35, 1995.

[2] Y. Kesten, A. Pnueli, J. Sifakis, and S. Yovine, “Decidable Integration Graphs”, Information and Computation, Vol. 150, No. 2, p. 209 - 243, 1999.

[3] G. Lafferriere, G. J. Pappas and S. Yovine, “A New Class of Decidable Hybrid Systems”, *Lecture Notes in Computer Science*, Vol. 1569, pp. 137-151, 1999.

[4] G. Lafferriere, G. J. Pappas and S. Yovine, “Reachability Computation of Linear Hybrid Systems” *Proc. 14th IFAC World Congress*, Vol. E, pp. 7-12, Beijing, PRC, July 1999.

[5] T. A. Henzinger, P.-H. Ho, and H. Wong-Toi, “Algorithmic analysis of nonlinear hybrid systems”, *IEEE Trans. Automatic Control*, Vol. 43, pp. 540-554, 1998.

[6] O. George J. Pappas, S. Sastry, “Semidecidable Controller Synthesis for Classes of Linear Hybrid Systems”, Proc. 39th IEEE Conf. on Decision and Control, Sydney, Australia, Dec. 2000

[7] E. Asarin, O. Bournez, T. Dang, O. Maler, A. Pnueli, “Effective Synthesis of Switching Controllers for Linear Systems”, *Proceedings of the IEEE*, Vol. 88, No. 7, pp. 1011-1025, 2000.

[8] T. Dang and O. Maler, “Reachability analysis via face lifting”, T. A. Henzinger and S. Sastry (Eds.) *Hybrid Systems: Computation and Control*, LNCS 1386, pp. 96-109, Springer Verlag, New York, 1998.

[9] C. Tomlin, J. Lygeros, and S. Sastry, “Synthesizing controllers for nonlinear hybrid systems”, in *Hybrid Systems: Computation and Control*, LNCS 1386, pp. 360–373, Springer Verlag, New York, 1998.

[10] P. Varaiya, “Reach set computation using optimal control”, Proc. *KIT Workshop*, Verimag, Grenoble, 1998.

[11] A. B. Kurzhanski and P. Varaiya, “Ellipsoidal techniques for reachability analysis: internal approximation”, *Systems & Control Letters*, Vol. 41, No. 3, Oct. 2000.

[12] E. Asarin, O. Bournez, T. Dang, and O. Maler, “Approximate reachability analysis of piece-wise linear dynamical systems”, in *Hybrid Systems: Computation and Control*, March 2000, Pittsburgh, USA.

[13] O. Maler, A. Pnueli and J. Sifakis, “On the Synthesis of Discrete Controllers for Timed Systems (An Extended Abstract)”, *Proc. STACS ’95*, LNCS 900, pp. 229-242, Springer, 1995.

[14] T. A. Henzinger, “Hybrid automata with finite bisimulations”, *Proc. ICALP ’95*, LNCS 944, pp. 324-33 Springer-Verlag, 1995.
Appendix A. Decidable Classes of Subsets of $Q \times \mathbb{R}^n$

We introduce here model theoretic concepts for formalizing the notion of decidable class $C$ of subsets of $Q \times \mathbb{R}^n$ that we were interested in Sec. ?? . Since $Q$ is a finite set, for a decidable class $C$ of subsets of $Q \times \mathbb{R}^n$, we may take, for each $q \in Q$, a decidable class of subsets of $\mathbb{R}^n$, $S_q$, $q \in Q$, and choose $C$ to be the product class $\bigcup_{q \in Q} \{ \{q\} \times S_q : S_q \in S_q \}$. Therefore, we restrict our attention to the discussion of classes of subsets of $\mathbb{R}^n$. Most parts of this section are taken from [3] (see also [?], which in turn cites the references [19, 20, 21]).

**Definition 11** A language is a tuple of three sets, $\mathcal{L} = (L_r, L_f, L_c)$, where

1. $L_r$ is a set of relations,
2. $L_f$ is a set of functions, and
3. $L_c$ are a set of constants.

**Definition 12** A model of a language $\mathcal{L} = (L_r, L_f, L_c)$ consists of a nonempty set $S$, together with an interpretation of the relations, functions and constants.

We denote a model by $(S, L_r, L_f, L_c)$, where the interpretation is not made explicit. In the following, let $\mathcal{V} = \{x, y, z, x_0, x_2, \ldots\}$ denote a countable set of variables.
Definition 13 A term of a language, $L = (L_r, L_f, L_c)$, is inductively defined as follows:

1. each variable $\theta \in V$ is a term,
2. each constant $c \in L_c$ is a term, and
3. for an $m$-ary function $g \in L_f$, where $m \geq 1$, and $m$ terms, $\theta_1, \theta_2, \ldots, \theta_m$, $g(\theta_1, \theta_2, \ldots, \theta_m)$ is a term.

Definition 14 An atomic formula of a language $L$ is either

1. $\theta_1 = \theta_2$, where $\theta_1$ and $\theta_2$ are two terms of $L$, or
2. $\rho(\theta_1, \ldots, \theta_n)$, where $n \geq 1$ and $\rho \in L_r$ is an $n$-ary relation.

Definition 15 A first order formula, or simply a formula, of a language $L$ is recursively defined as one of the following:

1. an atomic formula, or
2. $\neg \phi$, where $\phi$ is a formula and $\neg$ is logical negation, or
3. $\phi_1 \land \phi_2$, where $\phi_1$ and $\phi_2$ are formulas and $\land$ is logical and, or
4. $\exists x : \phi$ or $\forall x : \phi$, where $\phi$ is a formula, $x$ is a variable, and $\exists$ (there exists) and $\forall$ (for all) are quantifiers; in this case, each occurrence of the variable $x$ in the formula $\phi$ is called a bound occurrence.

Definition 16 The occurrence of a variable in a formula is free, if it is not bound. A sentence in a language $L$ is a formula with no free variable. A theory of $L$ is a subset of sentences.

For a model $S$ of the language $L$, we shall be particularly interested in the theory defined by the set of all sentences that are true in $S$. To emphasize this, we refer to this theory as the theory of $(S, L_r, L_f, L_c)$.

Definition 17 Let $L$ be a language and $S$ be a model of $L$. A set $X \subseteq S^n$ is said to be definable in the language $L$, if there is an n-ary formula $\phi(x_1, \ldots, x_n)$ such that $X$ can be written as $X = \{(x_1, \ldots, x_n) \in S^n : \phi(x_1, \ldots, x_n)\}$.

Definition 18 Let $L$ be a language, $S$ be a model of $L$, and $C$ be the class of definable sets. Then $C$ is said to be decidable, if the theory of $(S, L_r, L_f, L_c)$ is decidable, i.e., there is a decision procedure that, given an $L$-sentence $\phi$, decides whether $\phi$ belongs to the theory of $(S, L_r, L_f, L_c)$ or not.

Examples

1. The theory $(\mathbb{R}, \{<\}, \{+, -, \}, \{0, 1\})$ is the theory of linear constraints with integer coefficients, denoted by $\text{Lin}(\mathbb{R})$. The sets defined by these formulas are called polyhedral sets.
2. The theory \((\mathbb{R}, \{<\}, \{+,-,\cdot\}, \{0,1\})\) is the theory of polynomial constraints with integer coefficients, denoted by \(\text{OF}(\mathbb{R})\). The sets defined by these formulas are called subalgebraic sets.

The result stated below is due to A. Tarski [22]:

**Theorem 1** The first order theory \(\text{OF}(\mathbb{R})\) is decidable.

**Definition 19** Let \(\mathcal{L}\) be a language and let \(\mathcal{S}\) be a model of \(\mathcal{L}\). We say the theory of \((\mathcal{S}, L_r, L_f, L_c)\) admits quantifier elimination if every first order formula of \((\mathcal{S}, L_r, L_f, L_c)\) is equivalent to a formula of \((\mathcal{S}, L_r, L_f, L_c)\) without quantifiers.

**Examples: Decidability and Quantifier Elimination**

1. The theory \(\text{OF}(\mathbb{R})\) consisting of \((\mathbb{R}, \{<\}, \{+,-,\cdot\}, \{0,1\})\) admits quantifier elimination and is also decidable.

2. Let \(\text{OF}_{\exp}(\mathbb{R})\) be the theory consisting of \((\mathbb{R}, \{<\}, \{+,-,\cdot, \exp\}, \{0,1\})\), where \(\exp\), representing the exponential function, is a new function symbol. This theory does not admit quantifier elimination, and it is not known whether this theory is decidable. (See [13]).

**Appendix B:** \(\text{Reach}_q(X_0, [0, \tau]) = X(\tau) = X^+(\tau)\)

We assume that \(X_0\) is closed and its boundary, denoted by \(S_0\), consists of a finite union of smooth surfaces. Let

\[
X(\tau) = X_0 \bigcup_{0 \leq t \leq \tau} \{\phi_q(x_0, t) : x_0 \in S_0\}.
\]

Also let

\[
S_0^+ = \{x \in S_0 : \exists \varepsilon = \varepsilon(x) > 0 \text{ such that } \phi(x, t) \in X_0^c, \forall t \in (0, \varepsilon)\}, \quad (16)
\]

and \(X^+(\tau) = X_0 \bigcup_{0 \leq t \leq \tau} \{\phi_q(x_0, t) : x_0 \in S_0^+\}\).

**Proposition 1** \(X^+(\tau) = \text{Reach}(X_0, [0, \tau])\).

**Proof.** Note that \(X^+(\tau) \subset X(\tau) \subset \text{Reach}(X_0, [0, \tau])\). So we have to show that \(\text{Reach}(X_0, [0, \tau]) \subset X^+(\tau)\). Let \(z \in \text{Reach}(X_0, t), t > 0\). If \(z \in X_0\), then \(z \in X^+(\tau)\). Now suppose \(z \notin X_0\). So there is a \(z_0 \in X_0\) such that \(z = \phi(z_0, s)\), for some \(s\) with \(0 < s \leq \tau\). If \(z_0 \in S_0^+\), then \(z \in X^+(\tau)\). Otherwise let \(\tau' = \sup\{t : t \leq s \text{ and } \phi(z_0, t) \in X_0\}\), so \(0 < \tau' < s \leq \tau\). Now \(z' = \phi(z_0, \tau') \in S_0^+\). Therefore, \(z = \phi(z', s - \tau') \in X^+(\tau)\).

Now suppose that \(X_0\) is specified as \(X_0 = \{x \in \mathbb{R}^n : \ell(x) \leq 0\}\), where \(\ell : \mathbb{R}^n \to \mathbb{R}\) is continuously differentiable. Further, we assume that if \(x \notin \overset{0}{X_0}\) then \(\ell(x) < 0\) (where \(\overset{0}{X_0}\) denotes the interior of \(X_0\), i.e., the largest open set contained in \(X_0\)). So, obviously, if \(x \in S_0\) then \(\ell(x) = 0\), and if \(x \notin \overset{0}{X_0}\) then \(\ell(x) > 0\). Let \(S_0^+\) be defined as

\[
S_0^+ = \{x \in S_0 : \nabla \ell(x) \cdot f(q, x) \geq 0\}.
\]
We now show that \( S_0^+ \subset S_t^+ \). To this end, we show that \( S_0 \setminus S_t^+ \subset S_0 \setminus S_0^+ \). Let \( x \in S_0 \setminus S_t^+ \), so \( \nabla \ell(x) \cdot f(q, x) < 0 \). Now, since \( \ell(q, 0)) = \ell(x) = 0 \), and since at \( t = 0 \), \( \frac{d\ell(q, 0))}{dt} = \nabla \ell(x) \cdot f(q, x) < 0 \), we have, for a sufficiently small \( \epsilon > 0 \), \( \ell(q, t)) < 0 \), whenever \( 0 < t < \epsilon \), which happens only if \( q, t) \in X_0 \), \( \forall t \in (0, \epsilon) \), implying that \( x \notin S_0^+ \); hence \( x \in S_0 \setminus S_0^+ \). Thus \( S_0 \setminus S_0^+ \subset S_0 \setminus S_0^+ \), as required.

**Appendix C: Reach\( _t \) (\( X_0, X_q, [0, \infty) \))**

Let \( X_q \subset \mathbb{R}^n \) with compact closure, and let \( Y_q = \overline{X_q} \). Let \( f : W \to \mathbb{R}^n \) be a continuous function defined on an open set \( W \) containing \( Y_q \), satisfying a Lipschitz condition on \( W \). Let \( X_0 \) be a closed subset of \( Y_q \). We assume that for each \( x \in X_0 \), a function \( \gamma_x : \mathbb{R}^+ \to \mathbb{R}^n \) exists and satisfies the differential equation

\[
\frac{d\gamma_x(t)}{dt} = f(\gamma_x), \quad t \geq 0,
\]

with the initial condition \( \gamma_x(0) = x \). Hence the flow \( \phi(x, t) \) associated with equation (17) is defined for all \( t \geq 0 \) and \( x \in X_0 \). Further, assume that the \( \omega \)-limit set of the flow \( \phi(x, t), t \geq 0 \), does not intersect \( Y_q \) for any point \( x \in X_0 \).

More precisely, we assume, for \( x \in X_0 \),

\[
L_\omega(x) \cap Y_q = \emptyset,
\]

where

\[
L_\omega(x) = \bigcap_{t \geq 0} \phi(x, [t, \infty)).
\]

(See [6] [7].) In what follows, we show that if (18) holds for every \( x \in X_0 \), then \( \exists \tau_{\text{max}} > 0 \) (depending on \( X_0 \), such that for every \( x \in X_0 \), there is a \( \tau = \tau(x) \) with \( 0 < \tau(x) \leq \tau_{\text{max}} \) and \( \phi(x, \tau(x)) \notin Y_q \). For this, let \( 0 < \tau_1 < \tau_2 < \ldots \) be an increasing sequence such that \( \tau_k \to \infty \), as \( k \to \infty \) (as in Sec. 3), and define the sets \( A(\tau_k, x) = \phi(x, [\tau_k, \infty)) \). We first observe that \( L_\omega(x) = \bigcap_k A(\tau_k, x) \).

**Proposition 2** Let \( x \in X_0 \). If (18) holds, then \( \exists \tau = \tau(x) > 0 \) such that \( \phi(x, t) \notin Y_q, \forall t \geq \tau \).

**Proof.** We have \( L_\omega(x) \cap Y_q = \bigcap_k A(\tau_k, x) \cap Y_q = \bigcap_k [A(\tau_k, x) \cap Y_q] \). Now let \( E(\tau_k, x) = A(\tau_k, x) \cap Y_q \). So, for a fixed \( x \in X_0 \), the sets \( E(\tau_k, x) \), \( k = 1, 2, 3, \ldots \), is a decreasing sequence of closed subsets of \( Y_q \). Since \( Y_q \) is compact, \( \bigcap_k E(\tau_k, x) = \emptyset \) implies \( E(\tau_k, x) = \emptyset \), for all but finitely many \( k \). Therefore, for some \( K \geq 1 \), \( \forall k \geq K, E(\tau_k, x) = \emptyset \). Hence, \( \phi(x, [t, \infty)) \cap Y_q = \emptyset, \forall t \geq \tau_K \). Therefore, with \( \tau = \tau_K \), \( \phi(x, t) \notin Y_q, \forall t \geq \tau \).

We also need the following proposition:

**Proposition 3** Let \( x \in X_0 \) be a point for which there is a \( \tau > 0 \), such that \( \phi(x, \tau) \notin Y_q \). Then there is a \( \delta = \delta_x > 0 \), such that if \( |x - y| < \delta_x \) and \( y \in X_0 \), then \( \phi(y, \tau) \notin Y_q \).
Proof. Let \(|f(x) - f(y)| \leq C|x - y|\). Then \(|\phi(x, \tau) - \phi(y, \tau)| \leq |x - y|e^{C\tau}\) (see, for example, [17]), so for a fixed \(\tau\), \(\phi(\cdot, \tau)\) is continuous in the first variable. Let \(z = \phi(x, \tau)\). Now since \(W \setminus Y_q\) is open, there is an \(\epsilon > 0\) such that \(B(z, \epsilon) \subset W \setminus Y_q\). By the continuity of \(\phi(\cdot, \tau)\) at \(x\), there is a \(\delta > 0\), such that \(|\phi(x, \tau) - \phi(y, \tau)| = |z - \phi(y, \tau)| < \epsilon\), whenever \(|y - x| < \delta\) and \(y \in X_0\). Therefore, \(\phi(y, \tau) \in B(z, \epsilon) \subset W \setminus Y_q\).

From the previous two propositions, we get the following theorem:

**Theorem 2** Let \(Y_q\) be compact, and \(f\) be a continuous function defined on an open set \(W\) containing \(Y_q\) satisfying a Lipschitz condition. Further assume condition (18) holds for every \(x \in X_0\). Then \(\exists \tau_{\text{max}} > 0\) independent of \(x\), such that for each \(x \in X_0\), there is a \(\tau(x)\) with \(0 < \tau(x) \leq \tau_{\text{max}}\) and \(\phi(x, \tau(x)) \not\in Y_q\).

**Proof.** Let \(x \in X_0\). By Proposition 3 there is a \(\tau(x) > 0\), such that \(\phi(x, \tau(x)) \not\in Y_q\). By Proposition 3, there is a \(\delta_x > 0\), such that for any \(y \in X_0\) with \(|y - x| < \delta_x\), \(\phi(y, \tau(x)) \not\in Y_q\). \(\{B(x, \delta_x) : x \in X_0\}\) is an open cover of \(X_0\), containing a finite subcover, say, \(\{B(x_1, \delta_{x_1}), \ldots, B(x_m, \delta_{x_m})\}\). Let \(\tau_{\text{max}} = \max\{\tau(x_1), \ldots, \tau(x_m)\}\). To check whether this choice of \(\tau_{\text{max}}\) is as in the theorem, let \(y \in X_0\) be an arbitrary point. Now \(y \in B(x_i, \delta_{x_i})\), for some \(i\) with \(1 \leq i \leq m\), and by the choice of \(\delta_{x_i}\), \(\phi(y, \tau(x_i)) \not\in Y_q\), concluding the proof.

**Appendix D: Boundedness Results**

In this section, we show that if the initial set \(F\) is bounded then the polyhedron, \(P\), enclosing \(F_{[0, A]}\), as obtained by the algorithm described in Sec. 3 is bounded. We assume that \(F\) is nonempty. Recall that \(F\) is given as the set of points \(x\) in \(\mathbb{R}^n\) which satisfy the following constraints:

\[
\begin{align*}
\mathbf{a}_i^T x - b_i &\leq 0, \quad i = 1, 2, \ldots, k - 1, \\
\mathbf{a}_k^T x - b_k &= 0,
\end{align*}
\]

and \(P\) is included in the set of points \(x \in \mathbb{R}^n\) satisfying

\[
\begin{align*}
\mathbf{a}_i^T x - b_i - l_i (\mathbf{a}_k^T x - b_k) &\leq 0, \quad i = 1, 2, \ldots, k - 1, \\
\mathbf{a}_k^T x - b_k &\geq 0, \quad \text{and} \\
\mathbf{a}_k^T x - b_k - l_k &\leq 0,
\end{align*}
\]

where \(l_i \in \mathbb{R}\) and \(l_k > 0\). Looking at the constraints, one may visualize the set \(P_1\) satisfying (20) as a prism or a truncated pyramid, with its bottom given by (19) and its top given by

\[
\begin{align*}
\mathbf{a}_i^T x - b_i - l_i (\mathbf{a}_k^T x - b_k) &\leq 0, \quad i = 1, 2, \ldots, k - 1, \\
\mathbf{a}_k^T x - b_k - l_k &= 0.
\end{align*}
\]

We assume that \(F\) is nonempty, and wish to show that if \(F\) is bounded then so is the set \(P_1\), defined as the set of points satisfying (20). We start with the following proposition:

**Proposition 4** Assume that \(F\) is nonempty and \(P_1\) is not bounded. Then there is a point \(x_0 \in F\) and a vector \(\lambda \in \mathbb{R}^n\), with \(|\lambda| = 1\), such that \(x_0 + t\lambda \in P_1\), for all \(t \geq 0\).
Proof. Fix a point $x_0 \in F$, and suppose $P_1$ is not bounded. So for each positive integer $k$, there is a point $x_k \in P_1$, such that $|x_k - x_0| \geq k$. Let $\lambda_k = \frac{(x_k - x_0)}{|x_k - x_0|}$. Now, since $F \subset P_1$, $x_0 \in P_1$ and, by convexity, the entire line segment $x_0 + t\lambda_k \in P_1$, for $0 \leq t \leq k$. In particular, the points on these line segments satisfy (20). Now, since for each $k$, $|\lambda_k| = 1$, and the closed unit ball in $\mathbb{R}^n$ is compact, there is a convergent subsequence of $\lambda_k$’s – say, $\lambda_{k_i}$, $i = 1, 2, 3, \ldots$ – such that $\lambda_{k_i} \to \lambda \in \mathbb{R}^n$, as $i \to \infty$; since $|\lambda_{k_i}| = 1$, $|\lambda| = 1$. We show that $x_0 + t\lambda \in P_1$ for $t \in [0, \infty)$. First note that for any positive integer $m$, and for all $i \geq m$, $x_0 + t\lambda_{k_i}$ satisfies the constraints (19), for $t \in [0, k_m]$. Therefore, $x_0 + t\lambda$ satisfies (20), for $t \in [0, k_m]$, and so $x_0 + t\lambda \in P_1$, for $t \in [0, k_m]$. The proposition is concluded by letting $m \to \infty$.

We now show that such an $\lambda$, as in the previous proposition, must be parallel to the hyperplane passing through $F$, the normal of which is given by $a_k$.

**Proposition 5** If $x_0 \in F$ and a unit vector $\lambda \in \mathbb{R}^n$ are such that $x_0 + t\lambda \in P_1$, for all $t \in [0, \infty)$, then $a_k^T \lambda = 0$.

Proof. For a contradiction assume that $h = a_k^T \lambda > 0$. Now since $x + t\lambda$ satisfies (20), we must have

$$a_k^T(x_0 + t\lambda) - b_k - l_k \leq 0,$$

which holds only if $t \leq \frac{(b_k - l_k - a_k^T(x_0))}{h}$, contrary to the hypothesis that, for all $t \in [0, \infty)$, $x_0 + t\lambda \in P_1$. Similarly, if $h = a_k^T \lambda < 0$, then the constraint

$$a_k^T(x_0 + t\lambda) - b_k \geq 0,$$

does not hold for $t > 0$. Therefore we must have $a_k^T \lambda = 0$.

We now show that if $F$ is bounded, then for any unit vector $\lambda$ orthogonal to $a_k$, there is an $i$ with $1 \leq i \leq k - 1$, such that $h_i = a_i^T \lambda > 0$.

**Proposition 6** If $F$ is nonempty and bounded, then for any unit vector $\lambda \in \mathbb{R}^n$, for which $a_k^T \lambda = 0$, there is an $i$, with $1 \leq i \leq k - 1$, such that $a_i^T \lambda > 0$.

Proof. Let $x_0 \in F$, and suppose that there is an $\lambda \in \mathbb{R}^n$, with $|\lambda| = 1$, $a_k^T \lambda = 0$ and $a_i^T \lambda \leq 0$, for $1 \leq i \leq k - 1$. Then for any $t > 0$, the vector $x_0 + t\lambda$ satisfies the constraints (19), and so $x_0 + t\lambda \in F$, contrary to the assumption that $F$ is bounded.

We now combine all the previous propositions to get the following result:

**Theorem 3** If $F$ is nonempty and bounded, then $P_1$ is bounded.

Proof. If $P_1$ is not bounded, then by Proposition 6, there is an $x_0 \in F$ and a unit vector $\lambda \in \mathbb{R}^n$ such that $x_0 + t\lambda \in P_1$, for any $t \geq 0$. By Proposition 6, $a_k^T \lambda = 0$. Now, by the previous proposition, since $F$ is bounded, there is an $i$, with $1 \leq i \leq k - 1$, such that $h_i = a_i^T \lambda > 0$. But then the constraint

$$a_i^T(x_0 + t\lambda) - b_i - l_i(a_k^T(x_0 + t\lambda) - b_k) \leq 0,$$
cannot hold for any $t$ with
\[ t > -\frac{a_i^T x_0 - b_i}{h_i}, \]
contrary to the assumption that $x_0 + t\lambda \in P_1$, for any $t \geq 0$. Hence $P_1$ is bounded.